

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи обміну миттєвими
повідомленнями у мережі Інтернет”**

Виконав здобувач вищої освіти
IV курсу, групи КІ-20
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Гарбар В.В.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Дресев О.М.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 17 » січня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Гарбару Вадиму Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет

2. Керівник роботи Дреєв Олександр Миколайович, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 131-02 від 01.04.2024 року

3. Строк подання студентом роботи до захисту 23.05.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 17 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання
« 17 » січня 2024 р.

Підпис керівника

Дреєв О.М.

(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2024 р.

Підпис здобувача

Гарбар В.В.

(прізвище та ініціали)

АНОТАЦІЯ

Гарбар В.В. Програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обміну миттєвими повідомленнями у мережі Інтернет.

Метою розробки є програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет.

Результат роботи – програмна реалізація системи обміну миттєвими повідомленнями у мережі Інтернет.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерна інженерія, обміну миттєвими повідомленнями

ABSTRACT

Harbar V.V. Software of the system of exchange of instant messages on the Internet. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the first (bachelor) level of higher education, software is developed, which is intended for the system of instant messaging on the Internet.

The goal of the development is the software of the instant messaging system on the Internet.

The result of the work is the software implementation of the instant messaging system on the Internet.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Delphi 10.4 environment.

Keywords: computer engineering, instant messaging

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	12
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми	28
3.4 Розробка діаграми процесів.....	53
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	55
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	55
4.2 Захист розробленого програмного забезпечення.....	66
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	68
6 ОСНОВНІ ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72

					ВКРБ-123.24.0017.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Гарбар В.В.				Програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет	Літ.	Аркуш	Аркушів
Перев.	Дресв О.М.					Б	1	78
Н.контр.	Коваленко А.С.				ЦНТУ КІ-20			
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- ОС – операційна система
- BMP – графічний формат
- JPEG – графічний формат
- GIF – графічний формат
- IP – інтернет протокол
- OSCAR – протокол обміну інформацією
- P2P – peer-to-peer – безпосереднє інтернет-з'єднання двох комп'ютерів, минаючи сервер
- SQL – мова управління базами даних
- UIN – Universal Identification Number – унікальний для кожного облікового запису номер

КБПЗ-2024

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Обмін миттєвими повідомленнями, як впливає з назви, – це інтернет-служба, яка відповідає за просте, легке та миттєве надсилання повідомлень від одного користувача до іншого користувача такої служби. Ви також можете назвати це текстовою розмовою між кількома особами в Інтернет-чаті за допомогою їхніх мобільних пристроїв або інших електронних пристроїв зв'язку.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем обміну миттєвими повідомленнями у мережі Інтернет.
- Дослідження системи обміну миттєвими повідомленнями у мережі Інтернет.
- Програмна реалізація системи обміну миттєвими повідомленнями у мережі Інтернет.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі обміну миттєвими повідомленнями у мережі Інтернет.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для обміну миттєвими повідомленнями у мережі Інтернет.

Щоб текстові системи, які ми використовуємо сьогодні, закріпилися, нам потрібні були дві речі. По-перше, обмін повідомленнями має стати більш оптимізованим, а стандартизація забезпечить засоби та сумісність для різних систем для спілкування одна з одною. По-друге, нам потрібен був спосіб надати цю послугу масам, забезпечивши швидкий і простий спосіб впровадження служб обміну миттєвими повідомленнями без технічних поразок попередніх систем.

Еволюція цих потреб остаточно відбулася наприкінці 80-х із появою системи Internet Relay Chat. Базуючись на системі кімнатного типу та використовуючи чистий графічний інтерфейс, IRC надав користувачам можливість зустрічатися та спілкуватися в кімнатах, а також надаючи засоби доступу до служб прямого обміну повідомленнями.

Використовуючи TCP для підключення та працюючи на різних мережевих серверах, IRC викликав вибухове зростання серед користувачів Інтернету у форматі, який залишається популярним донині. Однак із розвитком технологій спрощений характер IRC-з'єднань став вразливим до атак типу «відмова в обслуговуванні» та інших форм експлуатації.

Коли технологія змінилася від нової до старої, 90-ті принесли нового конкурента грі месенджерів із запуском ICQ у 1996 році. Спочатку розроблена в Ізраїлі ICQ була унікальною, оскільки не покладалася на традиційний маркетинг, і ICQ стала популярною в рекомендації і нічого більше. Хоча ICQ мав можливість надавати групові чати, спочатку він був розроблений для спілкування один на один із базою користувачів у всьому світі, яку можна було шукати та взаємодіяти.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

ICQ залучила десятки мільйонів користувачів, на піку свого розвитку було зареєстровано понад 100 мільйонів облікових записів. ICQ також заклав би основу для того, щоб багато популярних сьогодні форм соціальних медіа стали популярними. Бум дот-ком також забезпечив багато інших претендентів на поєднання з месенджерами AOL і Yahoo, які мали велику кількість користувачів на початку 2000-х років.

Остання частина головоломки з'явилася в 2000 році з появою формату XMPP. Проект із відкритим кодом і відкритим стандартом, який працював через додаток під назвою Jabber (це також було початковою назвою для XMPP), XMPP забезпечив можливість міжстандартного спілкування, заклавши основу для швидкої еволюції соціальних медіа в наше повсякденне життя. XMPP залишається популярним форматом для приватних серверів і програм для чату навіть сьогодні в світі, де домінують повідомлення в соціальних мережах.

Коли зірки нарешті зійшлися і всі наші потреби задоволені, ми переходимо до кінця 2000-х із запуском Myspace, а пізніше Facebook і Twitter. Надання користувачам спільноти та мережових функцій, а також обміну миттєвими повідомленнями Соціальні медіа були унікальними, оскільки вони забезпечили перші спроби взагалі в одній програмі, що дало користувачам централізований простір для їх цифрового життя.

Хоча Myspace був першим, він не був довгим, і перша справжня успішна спроба, яку ми побачили в цьому просторі, відбулася із запуском Facebook у 2004 році. Хоча все починалося з простих початків, невдовзі було запущено месенджер Facebook, користувачі засоби для миттєвого обміну текстовими повідомленнями за межами своєї скриньки Facebook.

Оскільки використання текстових служб повільно починає замінювати традиційні SMS-повідомлення, наприкінці 2010-х років буде розроблено нові протоколи, а також запущено кілька помітних гравців, які продовжують працювати й сьогодні. У 2009 році у нас був Whatsapp, у 2013 році – Telegram, а в 2014 році – Signal. Запущений для того, щоб забезпечити справжню можливість

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

зашифрованого чату, запуск Signal ознаменував нову сферу, де концепція конфіденційності знову буде в центрі уваги громадськості.

1.2 Область застосування

Областю застосування є інтернет-спілкування. З тих пор, як Інтернет став загальнодоступним, люди, мало-мальськи знайомі з комп'ютером, воліють спілкуватися за допомогою електронних комунікацій, усе рідше прибігаючи до традиційних засобів зв'язку. Переваги цього способу незаперечні: швидко, зручно й просто можна обмінятися інформацією з будь-яким жителем миру, де б він не перебував, у сусідньому будинку або на іншому континенті.

Найбільш популярні онлайн-месенджери має власного клієнта (програму), але більше розповсюджений альтернативний софт, що дозволяє підключити кілька протоколів до однієї оболонки (мультипротокольні клієнти). Останній варіант зручний тим, що не вимагає установки декількох клієнтів і освоєння різних інтерфейсів, заощаджує ресурси комп'ютера, прискорює його запуск (одна програма стартує швидше, ніж дві-три), а також не перевантажує інтерфейс зайвими об'єктами (ті ж іконки в системному треї). Крім того, подібні програми володіють «архітектурними надмірностями» начебто надбудов, плагінів, прикрашень, що розширюють функціональні можливості. Незручність доставляє хіба що не завжди коректна підтримка протоколів, через яку рветься зв'язок, і необхідність постійно відслідковувати відновлення. Проблеми виникають і через те, що інтернет-провайдери блокують доступ до деяких ресурсів, а багато роботодавців навмисно роблять те ж саме – щоб співробітники не витрачали робочий час на особисті потреби.

Існує й другий вид онлайн-месенджерів – веб-сервіси, їх можна запустити на будь-якому комп'ютері будь-яким браузером і навіть у середовищі мобільної ОС (Android, iOS). Вони підтримують всі популярні інтернет-комунікатори, причому найчастіше в особистому кабінеті користувача

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

запам'ятовуються налаштування для кожного з них, і для кожного входу не потрібно вводити трохи логінів і паролів. Цей варіант кращий для тих, кому не потрібні красивості, а досить практичного, надійного й простого засобу спілкування, доступного скрізь, де є комп'ютер і підключення до Інтернету. Та й турботи про свіжість версії або протоколу лежать на плечах розроблювачів сервісів, споживання ресурсів комп'ютера мінімально, а зручність безсумнівно: запустив браузер – одержав повний доступ; закрив – робоче місце прибрано.

Зрозуміло, тут теж є свої особливості: спартанський інтерфейс, відсутність плагінів/надбудов, необхідність тримати відкритою окрему вкладку (стежити, щоб випадково її не закрити), відсутність всіх сервісів окремого онлайн-месенджера роблять таке рішення не занадто привабливим для деяких користувачів.

Узагальнюючи, можна сказати, що всі найпоширеніші месенджери мають однакові характеристики. Якщо функція мала більшу популярність в одному месенджері, то інші месенджери також незабаром додадуть її в список своїх можливостей – так, після того, як стало популярно використовувати месенджери для того, щоб робити безкоштовні телефонні дзвінки іншим користувачам при використанні навушників, всі месенджери стали включати цю функцію.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Проведемо дослідження існуючих програм обміну миттєвими повідомленнями в мережі Інтернет.

Signal

Signal – інтернет-пейджер. Крім швидкого обміну повідомленнями, програма підтримує також можливості, як IP-телефонія, відеодзвінки, онлайн-ігри й безкоштовне відправлення SMS, а також повідомляє про нову пошту прийшла по email.

Основні можливості Signal:

- Спілкування зі співрозмовниками в режимі реального часу.
- Міжміські й міжнародні телефонні дзвінки за низькими цінами.
- Безкоштовне відправлення SMS на мобільні телефони.
- Голосове спілкування з іншими користувачами Signal.
- Онлайн ігри – шахи, шашки, морський бій.
- Можливість пошуку співрозмовників по регіонах і анкетних даних.
- Система «Антиспам» захистить вас від непрошених співрозмовників.
- Відеодзвінки за допомогою веб-камери.
- Налаштування кольору вікон програми на свій смак.
- Пересилання фотографій і інших файлів співрозмовникам через Signal.
- Перевірка орфографії «на льоту».
- Можливість використання декількох облікових записів (логінів) з однієї

програми.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8



Рисунок 2.1 – Інтерфейс користувача Signal

Raidcall

RaidCall – безкоштовна програма для групового голосового спілкування в багатокористувальницьких іграх, особливо в потребуючих командної участі (наприклад шутери або MMORPG). Спілкування по RaidCall відрізняється якісним звуком при мінімальних затримках. При цьому немає необхідності орендувати сервер або встановлювати його самостійно.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

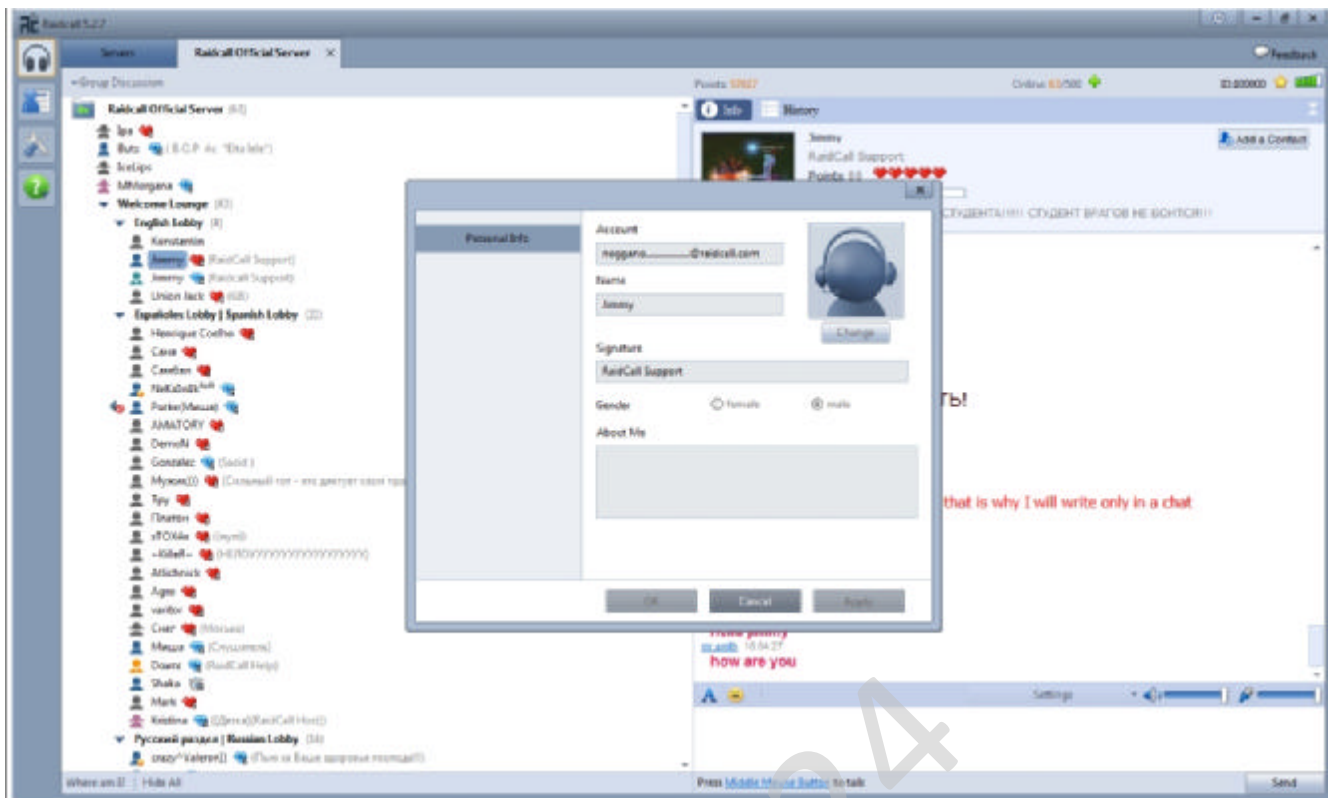


Рисунок 2.2 – Інтерфейс користувача RaidCall

RaidCall використовує модель передачі даних «клієнт-сервер». При використанні такої моделі, RaidCall відправляє дані від користувача на сервер, а сервер у свою чергу відправляє дані всім іншим користувачам на каналі. Це дозволяє знизити потреба в ширині каналу даних і може підтримувати до 1000 слотів на кожний канал.

Основні особливості RaidCall:

- Аудіо сервіс RaidCall заснований на системі хмарних обчислень.
- Система інтеграції інтерфейсу дозволяє вільно спілкуватися під час гри. Ви можете спілкуватися, говорити й навіть дивитися відео не залишаючи гри.
- RaidCall заснований на популярному й потужному движку Sreex, що дозволяє максимально знизити рівень шуму й підвищити якість аудіо.
- Використання UDP як комунікаційний протокол значно скорочує затримки в порівнянні з TCP протоколом, використовуваним іншими системами VoIP.

– Працює швидко й не вимагає великих ресурсів системи, дозволяючи не відволікатися від процесу гри.

– Має різні убудовані параметри конфіденційності й системи надання прав доступу, що дозволяє призначати різні повноваження різних учасників на основі каналів.

Windows Live Messenger

Windows Live Messenger – клієнт для обміну миттєвими повідомленнями, що надає безліч цікавих і корисних можливостей, включаючи звичайний чат, аудіо й відео конференції, спільні папки, дошки оголошень і т.п.

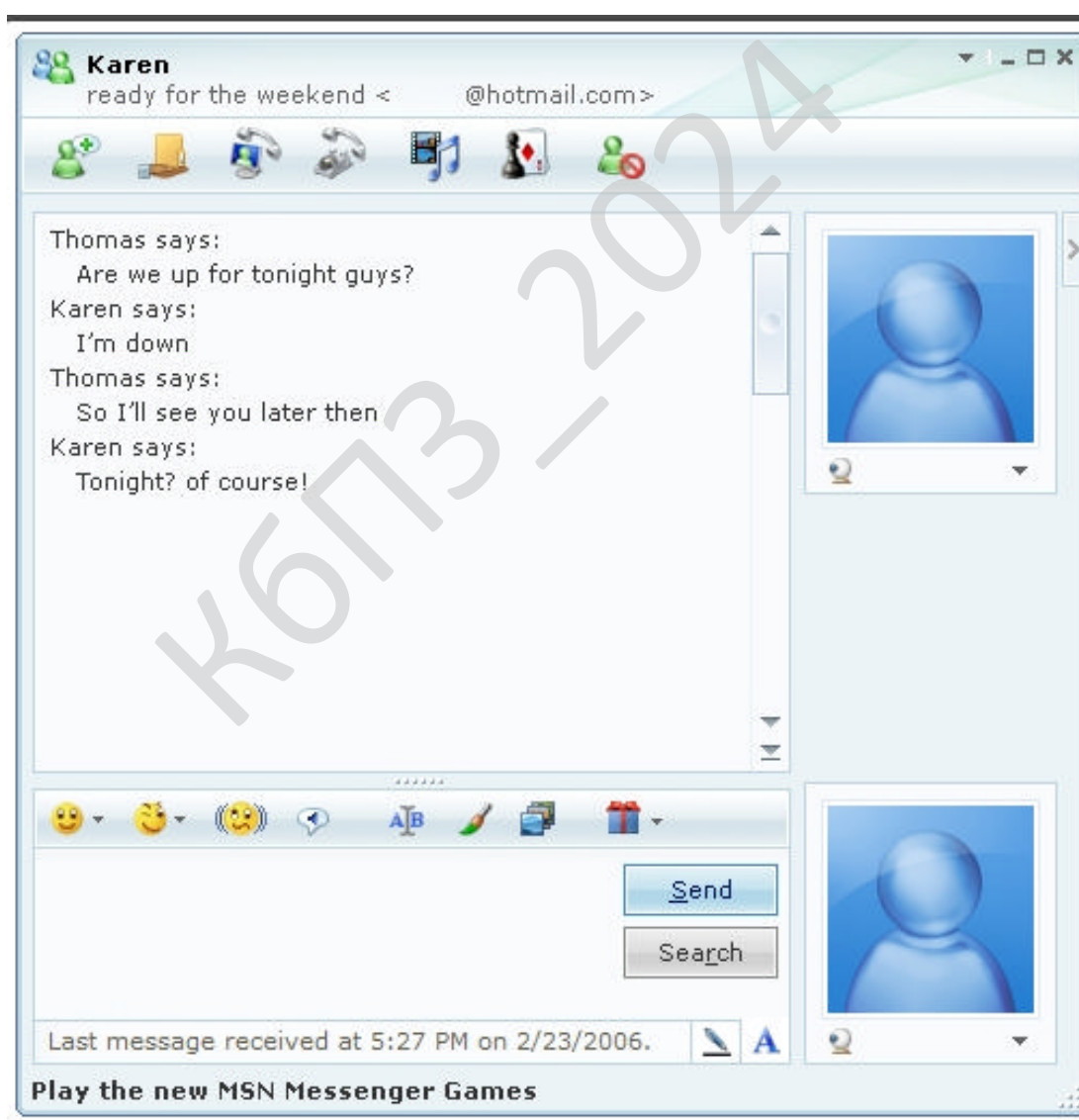


Рисунок 2.3 – Інтерфейс користувача Windows Live Messenger

Windows Live Messenger пропонує весь спектр стандартних послуг подібних клієнтів: робота з контакт-аркушем, смайлики, пересилання миттєвих повідомлень друзям, розмова голосом, відеокамера. Серед додаткових можливостей – телефонні дзвінки через систему телефонії MCI Web Calling for Windows Live Call. Крім того, є можливість пересилання документів і зображень.

Основні можливості Windows Live Messenger:

- Відправити контакту повідомлення можна, навіть якщо його немає в мережі. Повідомлення буде доставлено, коли контакт увійде в Messenger.
- Вибір різних ігор і додатків, до яких можна одержати доступ з вікна бесіди.
- Скіни, анімовані аватарки, звукові оповіщення. Колірна схема може бути обрана для всього додатка, включаючи вікно стану.
- Спільне використання папок. Функція передачі файлів.
- Дзвінки комп'ютера на комп'ютер і з комп'ютера на телефон.
- Веб або мобільний інтерфейс.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

- Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

- Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

- Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

- Тип даних Delphi «record» тепер підтримуватиме довільні ініціалізацію, фіналізацію й операції копіювання.

- Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

- Відладник Win 64 (на LLDB) і збирач для C++.

- Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

- Підтримка Metal Driver GPU для macOS і iOS.

- Вбудований Fmxlinux.

- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкістю. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів. У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків. ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCL, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізовані компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки. Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи обміну миттєвими повідомленнями у мережі Інтернет.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ - 2024

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

На початку існувало кілька основних інтернет-сервісів, таких як AOL, скорочення від America Online, CompuServe і Prodigy. Це були деякі зі способів, якими люди могли передавати інформацію туди-сюди кілька десятиліть тому. На початку 1990-х років було проведено опитування, і було виявлено, що 80% людей проводять більше часу в Інтернеті, працюючи або спілкуючись. Тим не менш, служби чату не були найкращими до листопада 1996 року, коли на сцену з'явився ICQ (Я шукаю тебе).

ICQ, служба обміну миттєвими повідомленнями, яка використовує програмне забезпечення під назвою «Клієнт». Ця програма була безкоштовною, і кожен міг легко нею користуватися. Цей так званий клієнт зазвичай розміщується на будь-якому електронному пристрої, де встановлено ICQ, і з тих пір і дотепер модель ICQ використовувалася як прецедент для більшості програм обміну миттєвими повідомленнями. Отже, використовуючи цей «клієнт» для ілюстрації, ось як працює обмін миттєвими повідомленнями.

Перш за все, вам потрібно завантажити копію «клієнта» прикладного програмного забезпечення на свій комп'ютер. Наступним кроком є встановлення клієнта та відкриття його на вашому електронному пристрої, після чого клієнт підключається до сервера за власним протоколом.

Після підключення клієнта до сервера ви вводите своє ім'я та пароль, щоб зареєструватися на сервері. Після цього першого разу вам потрібно буде увійти зі своїми даними під час наступних відвідувань. Після перевірки ви миттєво входите в систему.

Тепер клієнт надішле серверу іншу інформацію про підключення, наприклад кількість портів, імена людей у вашому списку контактів та IP-адресу

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

комп'ютера, який ви використовуєте. Імена людей відобразатимуться на вашому пристрої.

Сервер створює один тимчасовий файл, який містить інформацію про підключення та список ваших контактів. Він також дізнається, чи хтось із користувачів у вашому контакті ввійшов у систему. Якщо вони ввійшли, сервер надсилає клієнту повідомлення з контактною інформацією цього користувача, і те саме робиться з вашою контактною інформацією. Людям у вашому списку контактів, які знаходяться в мережі.

Ви можете клацнути ім'я будь-якої особи у своєму списку контактів, з якою хочете поспілкуватися, і одразу відкриється вікно, у яке ви можете ввести текст. Після введення повідомлення необхідно натиснути «Надіслати», якщо ви хочете, щоб воно було доставлено іншій стороні.

Оскільки ваше клієнтське програмне забезпечення має номер порту та IP-адресу комп'ютера чи іншого пристрою особи, якій ви надсилаєте повідомлення, ваше повідомлення буде надіслано безпосередньо до клієнтського програмного забезпечення на пристрої цієї особи, тобто сервер більше не задіяний у цей момент, оскільки зв'язок тепер відбувається лише між двома клієнтськими програмами.

Інша сторона отримала ваше миттєве повідомлення та відповідає. Оскільки розмова стає довшою, вікно стає довшим, тому ви можете прокручувати вгору та вниз, щоб переглянути попередні повідомлення та відповіді на поточні повідомлення.

Коли ви закінчите розмову, ви можете закрити вікно повідомлення, перейти в автономний режим і вийти. Як тільки ви це зробите, іншого користувача також відразу помітять. Це так просто, але чи існують деякі небезпеки, пов'язані з роботою миттєвих повідомлень.

Небезпека обміну миттєвими повідомленнями

Незважаючи на те, що миттєві повідомлення мають численні переваги, у них є ряд недоліків:

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

1. Недотримання протоколу на робочому місці

Це може змусити співробітників на робочому місці почати передавати неприйнятний контент, особисту інформацію та навіть використовувати ворожу мову. Подібні дії призведуть лише до надзвичайного зниження продуктивності та навіть до сексуальних чи інших видів домагань, що, у свою чергу, призведе до кількох судових позовів. Найкраще, що можуть зробити компанії, коли йдеться про спілкування на робочому місці, це створити політику для використання миттєвих повідомлень на робочому місці, а також політику попередження для працівників, які не відповідають вимогам.

2. Крадіжки особистих даних

Слід знати, що системи обміну миттєвими повідомленнями можуть дозволити будь-кому створювати анонімні або підроблені особи, які не співвідносяться з електронною адресою. Це погано тим, що Ідентифікаційні картки інших осіб можуть бути створені без відома справжнього власника. Насправді домен та ідентифікаційні дані власника не обов'язково повинні бути у власності шахрая, перш ніж він створить підробку.

3. Витік безпеки даних

Одним із ключових недоліків обміну миттєвими повідомленнями є те, що кілька файлів або документів (маленьких чи великих) можна передати без жодних слідів. Після надсилання інформації навіть можливості відстеження ІТ-відділу виявляться марними, оскільки технологія обміну миттєвими повідомленнями не розвинена настільки, щоб фільтрувати та відхиляти повідомлення чи файли, які можуть створити для компанії проблеми, скажімо, розкриваючи свої секрети. Таким чином, будь-яка інформація, яку потрібно надіслати, має бути відфільтрована або перевірена ІТ-департаментом або тим, хто відповідальний, щоб переконатися, що все є недоторканим.

4. Спам-повідомлення

Через те, що інформацію, що передається через систему миттєвих повідомлень, неможливо відфільтрувати, багато спаму потрапляють до

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

електронних листів користувачів у всьому світі. Якось дослідження показало, що з кожних 10 електронних листів на тиждень принаймні 2 з них мають бути спамом. Це дуже неприємні та неприйнятні повідомлення, але ви можете заблокувати їх надходження.

5. Віруси

Якими б непроникними не були служби обміну миттєвими повідомленнями, насправді вони дуже схильні до вірусних атак. Недавнє дослідження показало, що протягом шести місяців із 50 найпопулярніших вірусів, які проникли на комп'ютери чи інші мобільні пристрої людей, 19 з них використовували програми обміну миттєвими повідомленнями для здійснення своїх дій. Ці віруси можна легко надіслати під час передачі файлів, оминаючи безпеку або будь-яку іншу форму перевірки. Найкращий спосіб уникнути вірусів – перевірити будь-який файл, який хтось може надіслати вам.

Нові інновації в миттєвих повідомленнях

Які найпопулярніші та найновіші інновації в обміні миттєвими повідомленнями в місті? Раніше не було нічого схожого на спільний доступ до файлів, лише звичайний метод спілкування в чаті, але тепер майже всі служби обміну миттєвими повідомленнями пропонують обмін файлами. Насправді Google тепер включив служби обміну миттєвими повідомленнями у свою службу електронної пошти, що полегшує користувачам отримання сповіщень електронною поштою навіть під час використання миттєвих повідомлень. MSN також наслідував цей приклад.

Більшість постачальників миттєвих повідомлень тепер дозволяють своїм користувачам обмінюватися файлами під час обміну миттєвими повідомленнями з будь-якої точки світу та в будь-який час, якщо є мережа. Якщо ви користуєтеся Google Talk, є функція, яка дозволяє архівувати ваші розмови, перетягувати папки та файли до чату, вибирати зображення, надсилати їх іншим користувачам тощо. Насправді Windows Live Messenger було запрограмовано на можливість обміну миттєвими повідомленнями між геймерами Xbox 360.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Завдяки чудовим можливостям персоналізації IMers тепер можуть повністю налаштовувати особисті значки, привітання та інтерфейс користувача. Цей самий Windows Live Messenger разом із Google Talk та іншими програмами для обміну миттєвими повідомленнями тепер дозволяє користувачам робити щось, відоме як голосова нотатка та відеодзвінок, де ви можете записати, як щось говорите, і надіслати це їм для першого, і навіть побачити їх віч-на-віч під час телефонної розмови. для останнього.

Перспективи розвитку

Хоча ніхто не обговорює зручність цих послуг і переваги продуктивності, які вони забезпечують, багато хто висловив законне занепокоєння щодо управління даними та порушень, а також небезпек, пов'язаних із наданням великих обсягів даних користувача великим корпораціям. Управління цими зібраними даними також стає предметом суперечок, оскільки ми бачимо широкомасштабний збір і застосування аналітичних даних і подальше неправильне управління цими даними через такі речі, як скандал із Cambridge Analytica.

Ми також бачимо еволюцію зашифрованих програм для обміну повідомленнями, що значно полегшує спілкування кіберзлочинців і організованої злочинності, що значно ускладнює перехоплення комунікацій правоохоронним органам. Тож у майбутньому ми можемо очікувати, що суперечки навколо таких речей, як наскрізне шифрування в програмах для прямого обміну повідомленнями, будуть продовжуватися. Ми також можемо побачити нові функції, щоб спробувати пом'якшити це, з новим фокусом на таких речах, як повідомлення, що самознищуються, і емпіричний вміст.

Однак очікується, що розвиток технології штучного інтелекту стане місцем справжньої еволюції, коли інтеграція чат-ботів, віртуальних помічників та інших функцій вдосконалення стане звичним явищем. Ми починаємо бачити перші початки цього в 2023 році із запуском Snapchat MyAI. Незважаючи на те, що AI

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

тренд лише зароджується, він також дає розуміння деяких змін, які відбудуться в нашому майбутньому підключеному світі.

3.2 Розробка структурної схеми

Структурна схема системи зображена на рисунку 3.1. Розроблене програмне забезпечення складається з наступних блоків.

Блок протоколів:

- OSCAR.
- XMPP.
- MRA.
- SIP.

Клієнтська частина:

- Блок встановлення статусів.
- Блок пошуку.
- Блок обміну текстовими повідомленнями.
- Блок обміну мовною інформацією.
- Блок обміну відеоінформацією.
- Блок історії повідомлень по контактам.
- Головне меню.
- Блок авторизації.
- Блок відображення контактів.
- Блок налаштувань.

Серверна частина:

- База даних користувачів.
- База даних інформації про користувачів.
- База даних оффлайн повідомлень.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

У основі програмного забезпечення лежить протокол OSCAR. OSCAR – відкритий (з 5 березня 2008 року), але не вільний мережний протокол, що забезпечує обмін миттєвими й оффлайнними текстовими повідомленнями.

Стани:

- Online – доступний.
- Free for chat (F4C) – вільний.
- Away – вдалині від комп'ютера (довго не працював).
- Not available (N/A) – недоступний.
- Occupied – зайнятий.
- Do not disturb (DND) – не турбувати.
- Invisible – не бачимий.
- Offline – відключений.

У програмах-клієнтах сторонніх розроблювачів деякі стани можуть бути відсутніми або мати місце додаткові.

Розглянемо опис протоколу OSCAR.

Клієнти для AIM і ICQ використовують протокол OSCAR для спілкування з базою даних AIM/ICQ, також для багатьох інших речей як одержання списку контактів, відправлення й одержання повідомлень. Ви можете використовувати протокол OSCAR для написання свого IM-клієнта.

Офіційна документація не підтримує ніяких бібліотек і рекомендує використовувати одну з готових бібліотек доступних на їхньому сайті. Звичайно використовуючи дану документацію, ви порушуєте правила «Open AIM Platform Developer Agreement and Terms of Service». Але ця документація писалася й розвивалася заради інтересу, не використовуючи декомпіляторів. Ціль документації небагато доповнити офіційну документацію про AIM, додавши туди що не дістають частини й додати опис про сегмент ICQ.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

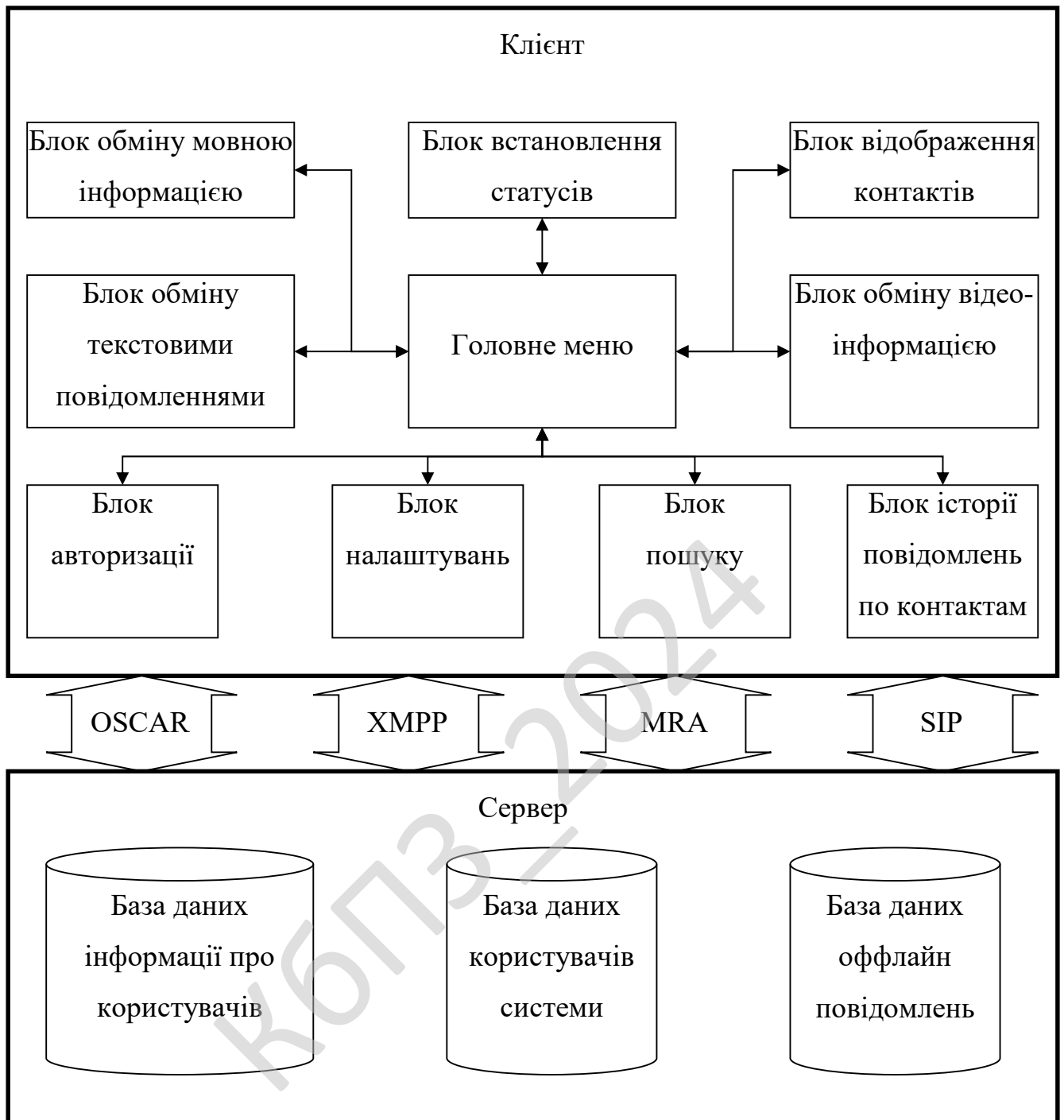


Рисунок 3.1 – Структурна схема системи

Протокол OSCAR – протокол двійкової синхронної передачі даних. Стік протоколу складається з TCP, для забезпечення передачі FLAP і SNAC повідомлення протоколу, що представляють собою, OSCAR. Багато серверів

підтримують TLS-з'єднання між FLAP і шаром TCP. У більшості випадків потік даних створений зі структур описаних нижче.

3.3 Розробка функціональної схеми

Типи серверів, що забезпечують роботу AIMICQ

ari.screenname.com – сервера даного типу забезпечують автентифікацію на веб-сервіси, відому як «AOL Open Auth». Перший крок на цих серверах – використання clientLogin для автентифікації користувача.

BUCP – сервера даного типу є серверами авторизації, що забезпечують автентифікацію клієнта через протокол OSCAR.

SBUCP – даний тип серверів відрізняється від BUCP-серверів тільки тим, що працює через SSL.

BOSS – опис BOSS-серверів.

ROOM – опис ROOM-серверів.

CHAT – опис CHAT-серверів.

ADMIN – опис ADMIN-серверів.

BART – Buddy Art (або BART) сервера забезпечують доступ клієнтові для завантаження зображень, звуків, і XML.

Той самий користувач може використовувати трохи клієнтів на тому самому або на різних пристроях. Створюючи клієнт або бібліотеку, потрібно передбачити те, що той самий користувач може бути онлайн багаторазово. Загалом, всі ІМ-повідомлення й повідомлення про зміну статусів відправляються на всі активні сесії, хоча, є деякого правила напрямку ІМ, які набирають чинності, коли користувачі в стані «відійшов» або в простій.

Похідні базові типи

Протокол побудований на типах даних описаних нижче.

Базові типи:

– u08 – 1 байт – байт;

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- u16 – 2 байт – два байти;
- u32 – 4 байти – чотири байти;
- f32 – 4 байти – чотири байти зі знаком;
- t70 – 4 байти – чотирьохбайтова мітка часу, з епохи UNIX (час у секундах з 1970 року);
- UUID – 16 байт – Шістнадцять байт які представляють собою «UUID» відомий як «GUID»;
- blob – довга в байтах – використовується в TLV, тип даних і розмір використовуваний зовнішніми значеннями;
- empty – 0 байт – використовується в TLV, наявність його означає позначку, дані ігноруються.

Типи рядків

Звичайні рядки не завершуються символом NULL і зберігаються в кодуванні UTF8. Рядок називається стислий, якщо вилучені пробіли, і всі букви у верхньому регістрі переведені в нижній:

- string – дані – усередині TLV рядок успадковує довжину від TLV-контейнера;
- string08 – u08 + дані – один байт довжини, супроводжувані даними зазначеної довжини;
- string16 – u16 + дані – два байти довжини, супроводжувані даними зазначеної довжини.

Тип даних: TLV

TLV – самий використовуваний тип даних у протоколі OSCAR. При розборі пакетів потрібно ігнорувати невідомі значення, щоб не порушити роботу старих клієнтів, коли нові речі будуть додані до протоколу. Доступні значення для тегів певні в протоколі; ці доступні значення належать до «TLV Class» (TLV класу). Тип даних іноді збігається зі значеннями, по цьому дана документація використовує числові значення (десятькова система вираження) для асоціювання з TLV і типом, як типом даних асоційованих із цим значенням:

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- tag – u16 – числові дані, припустимі значення описані в класі TLV даної групи TLV;

- len – u16 – довжина вмісту;

- value – blob – дані усередині TLV довжини len.

TLV звичайно використовуються в безлічі форм TLV, дозволяючи протоколу бути легко розширюваним. Наявність тільки одного TLV, без масивів приносить мало користі, тому що описує тільки один предмет. Є дві загальноприйняті методики для того, щоб додавати масиви TLV до типів даних і SNAC. Також існує додатковий метод для додавання масивів TLV до SNAC. Самим використовуваним є tlvBlock, що містить у собі число (u16) означаюче число TLV, супроводжуваного цими TLV. Менш використовуваним є tlvLBlock, що замість того, щоб використовувати кількість TLV, вважає розмір всіх TLV. Третій метод, що доступний тільки в SNAC, є tlvRestBlock, що означає, що будь-які байти, що залишаються, в SNAC – TLV:

- tlvBlock – u16 + дані – два байти означаючу кількість елементів, супроводжувані цими елементами;

- tlvLBlock – u16 + дані – два байти означаючу довжину елементів, супроводжувані цими елементами.

UUID – універсальний унікальний ідентифікатор (Universally Unique Identifier)

UUID використовуються в протоколі, для визначення якими можливостями володіє клієнт. Плюси UUID, це те, що вони унікальні коли створюються. Є багато веб сайтів, які можуть створити вам їх. Також UUID відомі як GUID. UUID це 16 байт двійкових даних, але в документаціях вони записані у формі послідовності. (Наприклад: 09460000-4C7F-11D1-8222-444553540000)

У певний момент було вирішено, що повні UUID занадто великі, і зробили Short UUID (Короткі UUID) – є дійсними як і звичайні UUID. (Наприклад: В

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

UUID 0946 XXYY-4C7F-11D1-8222-444553540000, XXYY – Short UUID). Клієнт одержує короткі UUID, якщо вкаже, що підтримує цю можливість.

Розроблювачі клієнтів можуть створювати нові UUID, але не Short UUID:

– SHORT_CAPS – 09460000-4C7F-11D1-8222-444553540000 – клієнт підтримує «Short UUID».

– SECURE_IM – 09460001-4C7F-11D1-8222-444553540000 – підтримка SECURE_IM.

– XHTML_IM – 09460002-4C7F-11D1-8222-444553540000 – підтримка XHTML-профілю й повідомлень усередині AOLRTF.

– RTCVIDEO – 09460101-4C7F-11D1-8222-444553540000 – підтримка SIP/RTP (Аудіо/Відео) розмови.

– HAS_CAMERA – 09460102-4C7F-11D1-8222-444553540000 – у користувача є веб-камера.

– HAS_MICROPHONE – 09460103-4C7F-11D1-8222-444553540000 – у користувача є мікрофон.

– RTCAUDIO – 09460104-4C7F-11D1-8222-444553540000 – підтримка RTCAUDIO.

– HOST_STATUS_TEXT_AWARE – 0946010A-4C7F-11D1-8222-444553540000 – клієнт підтримує нові статусні повідомлення.

– RTIM – 0946010 B-4C7F-11D1-8222-444553540000 – Client support "see as I type" IMs

– SMART_CAPS – 094601 FF-4C7F-11D1-8222-444553540000 – Client only asserts caps for services it is participating in

– FILE_TRANSFER – 09461343-4C7F-11D1-8222-444553540000 – підтримка передачі файлів.

– DIRECT_ICBM – 09461345-4C7F-11D1-8222-444553540000 – відправлення повідомлень прямо.

– FILE_SHARING – 09461348-4C7F-11D1-8222-444553540000 – File sharing.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- SUPPORT_ICQ – 0946134D-4C7F-11D1-8222-444553540000 – клієнт підтримує діалог з користувачами ICQ.

FLAP – Frame Layer Protocol

У протоколі OSCAR, FLAP забезпечує роботу із серверами на рівні TCP або шаруючи TLS. Кожне повідомлення, послане в або від клієнта, укладено в структурі FLAP, що легко ідентифікувати, використовуючи 6-ти байтовий заголовок, що супроводжується даними. Дані в FLAP – у більшості випадків містять SNAC.

Тип даних: FLAP__Header

Заголовок містить тип каналу, число послідовності й довжину наступного сегмента даних. Число послідовності незалежно в кожному напрямку, тобто в пакетів від сервера до клієнта є одне число послідовності, у той час як у пакетів від клієнта до сервера – інше число. З кожним відправленим пакетом це число збільшується. Якщо сервер одержить число послідовності не в неправильному порядку, то він розірве зв'язок із клієнтом. Краще використовувати, щоб використовувати u08, щоб створити випадкове число послідовності, тому що при досягненні максимального значення числа послідовності, сервер відключить клієнта. Також деякі люди вважають, що потрібно генерувати певні числа послідовності (відноситься тільки для ICQ), що б сервер «дозволив» відправляти ICBM:

- startMarker – u08 – зірочка (символ ASCII '*');
- frameType – u08 – [Клас: FLAP__FRAME_TYPE] – тип каналу;
- sequenceNumber – u16 – ініціалізоване випадкове значення, з кожним відправленням збільшується на один;
- payloadLength – u16 – довга даних, без 6-ти байт заголовка.

Клас: FLAP__FRAME_TYPE

Є кілька різних каналів, найбільш використовуваний з них «DATA»:

- SIGNON – 1 – ініціалізація підключення.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- DATA – 2 – повідомлення, що використовують FLAP підключення, звичайно це SNAC.

- ERROR – 3 – помилка FLAP – рідкість.

- SIGNOFF – 4 – закриття FLAP-підключення.

- KEEP_ALIVE – 5 – канал для відправлення повідомлень на сервер, щоб підтримувати або перевіряти зв'язок.

Клас: FLAP__MULTICONN_FLAGS

Дані прапори вказують серверам, як обробляти вже відкриті сесії, коли нова сесія починається:

- OLD_CLIENT – 0x00 – не використовувати мульти-сесії.

- RECENT_CLIENT – 0x01 – Цей клієнт використовує мульти-сесії.

- SINGLE – 0x03 – Цей клієнт підтримує мульти-сесії, але не хоче використовувати їх.

TLV Клас: FLAP__SIGNON_TAGS

Ці ознаки використовуються в каналі FLAP__SIGNON_FRAME. Вони додаються після 4-ьох байтового flapVersion:

- CLIENT_NAME – 3 – string – назва клієнта.

- LOGIN_COOKIE – 6 – blob – «Cookie» з startOSCARSession або BUCP__LOGIN_REPLY.

- MAJOR_VERSION – 23 – u16 – версія клієнта: «1», якщо версія клієнта «1.2.3».

- MINOR_VERSION – 24 – u16 – версія клієнта: «2», якщо версія клієнта «1.2.3».

- POINT_VERSION – 25 – u16 – версія клієнта: «3», якщо версія клієнта «1.2.3».

- BUILD_NUM – 26 – u16 – номер складання клієнта, що звичайно постійно підвищується.

- MULTICONN_FLAGS – 74 – u08 – [Клас: FLAP_MULTICONN_FLAGS]. У більшості випадків повинен бути RECENT_CLIENT (для ICQ: OLD_CLIENT).

- CLIENT_RECONNECT – 148 – u08 – інформація про те, що клієнт підключається після кого, як його відключило.

FLAP: FLAP_SIGNON_FRAME

При підключенні й сервер і клієнт посилають структуру SIGNON_FRAME на каналі FLAP для ініціалізації підключення. Число послідовності повинне бути згенероване випадковим образом. Містить чотири байти версії FLAP, тепер рівне одиниці. Також може містити будь-яка кількість необхідних TLV-блоків:

- header – FLAP_Header – заголовок FLAP, тип повинен бути FLAP_FRAME_TYPE_SIGNON;

- flapVersion – u32 – завжди «1»;

- tlvs – Масив з TLV довжини payloadLength – 4 – [Клас: FLAP_SIGNON_TAGS] TLV.

FLAP: FLAP_KEEPA_LIVE_FRAME

Безліч систем і мереж працюють із антивірусами, фаєрволами й подібними програмами, дані сервіси мають недоліки, наприклад, такі як втрату довго підключених підключень. Рекомендується використовувати KEEPA_LIVE пакет на каналі FLAP, так само існує його старий аналог OSERVICE_PING (радиться використання першого). Пакет потрібно посилати не частіше разу в хвилину, і коли немає реакції від користувача. Сервер ніяк не реагує на даний запит і ніколи не посилає подібний до клієнта:

- header – FLAP_Header – заголовок FLAP, тип повинен бути FLAP_FRAME_TYPE_KEEPA_LIVE.

FLAP: FLAP_DATA_FRAME

Дана структура даних завжди містить у собі один SNAC:

- header – FLAP_Header – заголовок FLAP, тип повинен бути FLAP_FRAME_TYPE_KEEPA_DATA;

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

- snacHeader – SNAC__Header – заголовок SNAC;
- snac – blob – дані присутні в SNAC міняються від типу SNAC описаних у його заголовку.

FLAP: FLAP__SIGNOFF_FRAME

На даному каналі FLAP повідомлення може відправити клієнт і сервер по різних причинах. Клієнт посилає його, коли хоче відключитися остаточно. Не варто відправляти його, якщо хочете перепідключитися. Сервер коли вимикається. Сервер якщо клієнт порушував які-небудь правила і його відключають:

- header – FLAP__Header – заголовок FLAP, тип повинен бути FLAP__FRAME_TYPE_SIGNOFF.

SNAC – Simple Network Atomic Communication

SNAC представляють повідомлення протоколу, які посилають між собою клієнт і сервер. Ці повідомлення розділені на різні групи, у даній документації називаємі «foodgroup». У межах кожної із групи є різні види повідомлень, пов'язаних із групою.

Тип: SNAC__Header

Заголовок FLAP відразу ж супроводжується заголовком SNAC який описує яке повідомлення містить сам SNAC. Заголовок складається з 10 байт утримуючих у собі групу, тип, прапори й номер запиту (requestId). Номер запиту використовує для об'єднання запитів і відповідей і так само може використовуватися для визначення напрямку трафіку. Клієнт не повинен установлювати верхній біт номера запиту або використовувати число 0. Сервер завжди встановлює верхній біт номера запиту, якщо даний SNAC не є відповіддю клієнтові, у цьому випадку сервер використовує цей же номер запиту який використовував клієнт. Прапори ж використовуються для повідомлення про деякі відповіді на запит або про опціональні TLV:

- id – SNAC__ID – група й тип повідомлення;
- flags – u16 – [Клас: SNAC__HEADER_FLAGS];

– requestId – u32 – використовуються для зледеніння запитів і відповідей.

Тип: SNAC__ID

Описує групу й тип, які представляє SNAC:

– foodgroup – u16 – числове значення, визначає до якої групи належить цей SNAC;

– type – u16 – числове значення, визначає тип SNAC.

Клас: SNAC__HEADER_FLAGS

Дані прапори пояснюють клієнтові деякі деталі про SNAC:

– OPT_TLV_PRESENT – 0x8000 – tlvLBlock треба відразу за заголовком SNAC.

– MORE_REPLIES_FOLLOW – 0x0001 – більше відповідей треба за цим повідомленням використовуючи цей же номер запиту. Остання відповідь легко визначити – на ньому не виставлений цей прапор.

Помилки

У кожній групі є помилки, і тип SNAC помилки – 1.

TLV Клас: ERROR__TAGS

Клас: ERROR__CODE

Foodgroup: OSERVICE

Група OSERVICE містить основні операції й типи даних, які використовуються в багатьох інших групах і навіть на різних серверах. Це також використовується для запитів обслуговування, коли клієнт повинен з'єднатися із ще одним сервером.

Клас: OSERVICE__NICK_FLAGS

Клас: OSERVICE__DCINFO_TYPES

Клас: OSERVICE__DCINFO_VERSIONS

Тип даних: OSERVICE__DCINFO

Клас: OSERVICE__CLIENT_TYPE

Клас: OSERVICE__ICQ_BROAD_TYPE

Клас: OSERVICE__PARENTAL_CONTROLS

Клас: OSERVICE__MY_BOT_INFO
 # Клас: OSERVICE__MY_SUBSCRIPTIONS
 # TLV Клас: OSERVICE__NICK_INFO_TAGS
 # Тип даних: OSERVICE__NickwInfo
 # Тип даних: OSERVICE__Version
 # Тип даних: OSERVICE__ShortVersion
 # SNAC: OSERVICE__ERROR – Foodgroup: 1 Type: 1
 # SNAC: OSERVICE__CLIENT_ONLINE – Foodgroup: 1 Type: 2
 # SNAC: OSERVICE__HOST_ONLINE – Foodgroup: 1 Type: 3

Rate Limiting (Ліміт обігів)

Щоб захистити сервер і інших користувачів від їхніх клієнтів, що навантажують, існують ліміти обігів. Це обмеження зроблене простою формулою, що обчислює середній час між послідовними SNAC від клієнта на сервер. Клієнт може підписатися на повідомлення про ліміти для того, щоб можна було попередити користувача. Навіть якщо клієнт не підпишеться, то буде одержувати повідомлення, коли ліміт був досягнутий і сервер почав ігнорувати SNAC. Якщо клієнт продовжить посилати SNAC, то з'єднання в остаточному підсумку буде розірвано.

Кожний SNAC перебуває в групах ліміту, які управляють параметрами формули в межах цього ліміту. Більшість SNAC перебуває в самому поблажливому ліміті за замовчуванням, а деякі SNAC, такі як IM-повідомлення, перебувають у більше строгих лімітах. Формула лімітів – $currentAvg = ((currentAvg * (windowSize - 1)) + delta) / windowSize$. $currentAvg$ падаючий нижче певних оцінок, описаних у класі обігів, попереджає клієнта, що ліміт прийшов у нормальний стан, обмежений або наближається до роз'єданого. Коли клієнт досяг обмеженого стану, ліміт повинен піднятися до значення CLEAR, перш ніж варто почати посилати SNAC знову.

Фактичні параметри для формул не відомі, тому що вони можуть час від часу змінюватися й різні залежно від поточного рівня попередження й інших

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

речей. Клієнт може відправляти біля одного ІМ-повідомлення кожні дві секунди, не будучи обмеженим лімітами.

Тип даних: OSERVICE__Rate_Parameters

Тип даних: OSERVICE__Rate_Class_Members

Клас: OSERVICE__RATE_CODE

SNAC: OSERVICE__RATE_PARAMS_QUERY – Foodgroup: 1 Type: 6

SNAC: OSERVICE__RATE_PARAMS_REPLY – Foodgroup: 1 Type: 7

SNAC: OSERVICE__RATE_ADD_PARAM_SUB – Foodgroup: 1 Type: 8

SNAC: OSERVICE__RATE_DEL_PARAM_SUB – Foodgroup: 1 Type: 9

SNAC: OSERVICE__RATE_PARAM_CHANGE – Foodgroup: 1 Type: 10

Migration

OSCAR підтримує поняття міграції, AOL може виконати обслуговування сервера, не відключаючи сесії користувача. Коли сервер починає переміщення, він повідомляє клієнта, що він повинен мігрувати на інший сервер. Як тільки клієнт готовий мігрувати до іншого сервера, він повідомляє про це й сервер дає дані для міграції, по яких клієнт може повторно з'єднатися, щоб продовжити його сесію знову.

SNAC: OSERVICE__PAUSE – Foodgroup: 1 Type: 11

SNAC: OSERVICE__PAUSE_ACK – Foodgroup: 1 Type: 12

SNAC: OSERVICE__RESUME – Foodgroup: 1 Type: 13

TLV Клас: OSERVICE__MIGRATE_TAGS

SNAC: OSERVICE__MIGRATE_GROUPS – Foodgroup: 1 Type: 18

Secondary Service Requests

Є додаткові сервера, які реалізують специфічні можливості. Для підключення до таких серверів клієнт повинен відправити SERVICE_REQUEST і одержує SERVICE_RESPONSE з інформацією для підключення. Додаткові сервера включають:

ACCOUNTADMIN – керування аккаунтами (переважно AIM).

CHAT – чати (для кожної кімнати окреме підключення).

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

CHATADMIN – керування чатами.

BART – завантаження XML, аватарів і смайликів.

TLV Клас: OSERVICE__SERVICE_REQUEST_TAGS

Клас: OSERVICE__SERVICE_RESPONSE_SSL_STATE

TLV Клас: OSERVICE__SERVICE_RESPONSE_TAGS

SNAC: OSERVICE__SERVICE_REQUEST – Foodgroup: 1 Type: 4

SNAC: OSERVICE__SERVICE_RESPONSE – Foodgroup: 1 Type: 5

Різне

SNAC: OSERVICE__NICK_INFO_QUERY – Foodgroup: 1 Type: 14

SNAC: OSERVICE__NICK_INFO_UPDATE – Foodgroup: 1 Type: 15

SNAC: OSERVICE__EVIL_NOTIFICATION – Foodgroup: 1 Type: 16

SNAC: OSERVICE__IDLE_NOTIFICATION – Foodgroup: 1 Type: 17

Клас: OSERVICE__MOTD_TYPE

TLV Клас: OSERVICE__MOTD_TAGS

SNAC: OSERVICE__MOTD – Foodgroup: 1 Type: 19

Клас: OSERVICE__PRIVACY_FLAGS

SNAC: OSERVICE__SET_PRIVACY – Foodgroup: 1 Type: 20

SNAC: OSERVICE__WELL_KNOWN_URLS – Foodgroup: 1 Type: 21

SNAC: OSERVICE__PING – Foodgroup: 1 Type: 22

SNAC: OSERVICE__CLIENT_VERSIONS – Foodgroup: 1 Type: 23

SNAC: OSERVICE__HOST_VERSIONS – Foodgroup: 1 Type: 24

SNAC: OSERVICE__MAX_CONFIG_QUERY – Foodgroup: 1 Type: 25

SNAC: OSERVICE__MAX_CONFIG_REPLY – Foodgroup: 1 Type: 26

SNAC: OSERVICE__STORE_CONFIG – Foodgroup: 1 Type: 27

SNAC: OSERVICE__CONFIG_QUERY – Foodgroup: 1 Type: 28

SNAC: OSERVICE__CONFIG_REPLY – Foodgroup: 1 Type: 29

SNAC: OSERVICE__SET_NICKINFO_FIELDS – Foodgroup: 1 Type: 30

SNAC: OSERVICE__VERIFICATION_QUERY – Foodgroup: 1 Type: 31

SNAC: OSERVICE__VERIFICATION_REPLY – Foodgroup: 1 Type: 32

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

SNAC: OSERVICE__BART_REPLY – Foodgroup: 1 Type: 33
SNAC: OSERVICE__BART_QUERY2 – Foodgroup: 1 Type: 34
SNAC: OSERVICE__BART_REPLY2 – Foodgroup: 1 Type: 35
SNAC: OSERVICE__CLIENT_MIGRATE_REQUEST – Foodgroup: 1

Type: 36

SNAC: OSERVICE__CLIENT_MIGRATE_ACK – Foodgroup: 1 Type: 37

Foodgroup: LOCATE

Група «LOCATE» дозволяє перегляд таких речей, як профіль AIM, інформація про аккаунт AIM/ICQ, статус повідомлення й пошук AIM-ніка по E-mail.

SNAC: LOCATE__ERROR – Foodgroup: 2 Type: 1

TLV Клас: LOCATE__RIGHTS_REPLY_TAGS

SNAC: LOCATE__RIGHTS_QUERY – Foodgroup: 2 Type: 2

SNAC: LOCATE__RIGHTS_REPLY – Foodgroup: 2 Type: 3

TLV Клас: LOCATE__TAGS

SNAC: LOCATE__SET_INFO – Foodgroup: 2 Type: 4

Клас: LOCATE__QUERY_TYPE

SNAC: LOCATE__USER_INFO_QUERY – Foodgroup: 2 Type: 5

SNAC: LOCATE__USER_INFO_REPLY – Foodgroup: 2 Type: 6

SNAC: LOCATE__WATCHERS_QUERY – Foodgroup: 2 Type: 7

SNAC: LOCATE__WATCHERS_REPLY – Foodgroup: 2 Type: 8

SNAC: LOCATE__UPDATE_DIRINFO_QUERY – Foodgroup: 2 Type: 9

SNAC: LOCATE__UPDATE_DIRINFO_REPLY – Foodgroup: 2 Type: 10

SNAC: LOCATE__GROUP_CAPABILITY_QUERY – Foodgroup: 2 Type:

13

SNAC: LOCATE__GROUP_CAPABILITY_REPLY – Foodgroup: 2 Type:

14

SNAC: LOCATE__KEYWORD_INFO_QUERY – Foodgroup: 2 Type: 17

SNAC: LOCATE__KEYWORD_INFO_REPLY – Foodgroup: 2 Type: 18

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Тип: LOCATE__NickwInfo_Email

SNAC: LOCATE__FIND_LIST_BY_EMAILS – Foodgroup: 2 Type: 19

SNAC: LOCATE__FIND_LIST_BY_EMAILS_REPLYS – Foodgroup: 2

Type: 20

Клас: LOCATE__QUERY2_TYPE

TLV Клас: LOCATE__GEOLOCATION_TAGS

TLV Клас: LOCATE__GEOLOCATION_REPLY_TAGS

Тип даних: LOCATE__GEOLOCATION_Info

SNAC: LOCATE__GEOLOCATION_QUERY – Foodgroup: 2 Type: 22

SNAC: LOCATE__GEOLOCATION_REPLY – Foodgroup: 2 Type: 23

SNAC: LOCATE__USER_INFO_QUERY2 – Foodgroup: 2 Type: 21

SNAC: LOCATE__LOOKUP_LOCATION_QUERY – Foodgroup: 2 Type:

24

SNAC: LOCATE__LOOKUP_LOCATION_REPLY – Foodgroup: 2 Type:

25

Foodgroup: BUDDY

За допомогою групи «BUDDY» OSCAR-сервера повідомляють клієнта про зміну статусу інших користувачів у його контакт-аркуші. Дані можуть містити інформацію про інших користувачів, наприклад, доданих у тимчасовий список контактів. Тимчасові списки зберігаються тільки на час сесії.

SNAC: BUDDY__ERROR – Foodgroup: 3 Type: 1

Клас: BUDDY__RIGHTS_QUERY_FLAGS

TLV Клас: BUDDY__RIGHTS_QUERY_TAGS

SNAC: BUDDY__RIGHTS_QUERY – Foodgroup: 3 Type: 2

TLV Клас: BUDDY__RIGHTS_REPLY_TAGS

SNAC: BUDDY__RIGHTS_REPLY – Foodgroup: 3 Type: 3

SNAC: BUDDY__ADD_BUDDIES – Foodgroup: 3 Type: 4

SNAC: BUDDY__DEL_BUDDIES – Foodgroup: 3 Type: 5

SNAC: BUDDY__WATCHERS_QUERY – Foodgroup: 3 Type: 6

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

SNAC: BUDDY__WATCHERS_REPLY – Foodgroup: 3 Type: 7
 # Тип: BUDDY__WATCHER
 # SNAC: BUDDY__WATCHER_REQUEST – Foodgroup: 3 Type: 8
 # SNAC: BUDDY__WATCHER_NOTIFICATION – Foodgroup: 3 Type: 9
 # SNAC: BUDDY__REJECT_NOTIFICATION – Foodgroup: 3 Type: 10
 # SNAC: BUDDY__ARRIVED – Foodgroup: 3 Type: 11
 # SNAC: BUDDY__DEPARTED – Foodgroup: 3 Type: 12
 # SNAC: BUDDY__ADD_TEMP_BUDDIES – Foodgroup: 3 Type: 15
 # SNAC: BUDDY__DEL_TEMP_BUDDIES – Foodgroup: 3 Type: 16
 # SNAC: BUDDY__LIFESTREAM_ADDED – Foodgroup: 3 Type: 17
 # SNAC: BUDDY__LIFESTREAM_UPDATED – Foodgroup: 3 Type: 18
 # SNAC: BUDDY__LIFESTREAM_DELETED – Foodgroup: 3 Type: 19

Foodgroup: ICBM

ICBM або Inter Client Basic Message – група протоколу, відповідальна за передачу повідомлень між користувачами або клієнтами.

ICBM__Cookie – 8 байт – дані, використовувані для об'єднання запитів і відповідей на них.

Клас: ICBM__ERROR_SUBCODE
 # SNAC: ICBM__ERROR – Foodgroup: 4 Type: 1
 # Клас: ICBM__CHANNELS
 # Клас: ICBM__PARAMETER_FLAGS
 # SNAC: ICBM__ADD_PARAMETERS – Foodgroup: 4 Type: 2
 # SNAC: ICBM__DELETE_PARAMETERS – Foodgroup: 4 Type: 3
 # SNAC: ICBM__PARAMETER_QUERY – Foodgroup: 4 Type: 4
 # SNAC: ICBM__PARAMETER_REPLY – Foodgroup: 4 Type: 5
 # Клас: ICBM__IM_SECTION_ENCODINGS
 # Тип даних: ICBM__IM_SECTION
 # TLV Клас: ICBM__IM_DATA_TAGS
 # Клас: ICBM__RENDEZVOUS_MESSAGE

Клас: ICBM__RENDEZVOUS_CANCEL_REASONS
 # TLV Клас: ICBM__RENDEZVOUS_TAGS
 # Тип даних: ICBM__IM_RENDEZVOUS
 # TLV Клас: ICBM__TAGS
 # SNAC: ICBM__CHANNEL_MSG_TOHOST – Foodgroup: 4 Type: 6
 # SNAC: ICBM__CHANNEL_MSG_TOCLIENT – Foodgroup: 4 Type: 7
 # Клас: ICBM__EVIL_REQUEST_FLAGS
 # SNAC: ICBM__EVIL_REQUEST – Foodgroup: 4 Type: 8
 # SNAC: ICBM__EVIL_REPLY – Foodgroup: 4 Type: 9
 # Клас: ICBM__MISSED_CALL_REASONS
 # Тип даних: ICBM__Missed_Call
 # SNAC: ICBM__MISSED_CALLS – Foodgroup: 4 Type: 10
 # Клас: ICBM__RENDEZVOUS_NAK
 # Клас: ICBM__CLIENT_ERRORS
 # SNAC: ICBM__CLIENT_ERR – Foodgroup: 4 Type: 11
 # TLV Клас: ICBM__ACK_TLV_TAGS
 # SNAC: ICBM__HOST_ACK – Foodgroup: 4 Type: 12
 # SNAC: ICBM__OFFLINE_QUERY – Foodgroup: 4 Type: 14
 # SNAC: ICBM__OFFLINE_REPLY – Foodgroup: 4 Type: 15
 # SNAC: ICBM__OFFLINE_RETRIEVE – Foodgroup: 4 Type: 16
 # SNAC: ICBM__OFFLINE_DELETE – Foodgroup: 4 Type: 17
 # SNAC: ICBM__NOTIFY_REQUEST – Foodgroup: 4 Type: 18
 # SNAC: ICBM__NOTIFY_REPLY – Foodgroup: 4 Type: 19
 # Клас: ICBM__EVENTS
 # SNAC: ICBM__CLIENT_EVENT – Foodgroup: 4 Type: 20
 # SNAC: ICBM__TEMP_DISABLE_IMF – Foodgroup: 4 Type: 21
 # SNAC: ICBM__RE_ENABLE_IMF – Foodgroup: 4 Type: 22
 # SNAC: ICBM__OFFLINE_RETRIEVE_REPLY – Foodgroup: 4 Type: 23
 # SNAC: ICBM__QUERY_SMS_INFO – Foodgroup: 4 Type: 24

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

SNAC: ICBM__QUERY_SMS_INFO_REPLY – Foodgroup: 4 Type: 25

Foodgroup: INVITE

Дана група протоколу відповідає за запрошення нових користувачів в AIM.

TLV Клас: INVITE__TAGS

SNAC: INVITE__ERROR – Foodgroup: 6 Type: 1

SNAC: INVITE__REQUEST – Foodgroup: 6 Type: 2

SNAC: REQUEST_ACK – Foodgroup: 6 Type: 3

TLV Клас: BUCP__LOGIN_REPLY_TAGS

Foodgroup: ADMIN

Група протоколу відповідальна за керуванням аккаунтом і реалізуюча можливості адміністратора.

SNAC: ADMIN__ERROR – Foodgroup: 7 Type: 1

Клас: ADMIN__ACCOUNT_PERMISSIONS_FLAGS

Тип: ADMIN__ACCOUNT_INFO

SNAC: ADMIN__ACCOUNT_INFO_QUERY – Foodgroup: 7 Type: 2

SNAC: ADMIN__ACCOUNT_INFO_REPLY – Foodgroup: 7 Type: 3

SNAC: ADMIN__UPDATE_ACCOUNT_INFO_QUERY – Foodgroup: 7
Type: 4

SNAC: ADMIN__UPDATE_ACCOUNT_INFO_REPLY – Foodgroup: 7
Type: 5

SNAC: ADMIN__ACCOUNT_CONFIRM_REQUEST – Foodgroup: 7
Type: 6

Клас: ADMIN__ACCOUNT_CONFIRM_FLAGS

TLV Клас: ADMIN__ACCOUNT_CONFIRM_TAGS

SNAC: ADMIN__ACCOUNT_CONFIRM_RESPONCE – Foodgroup: 7
Type: 7

SNAC: ADMIN__ACCOUNT_DELETE_REQUEST – Foodgroup: 7 Type:
8

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

SNAC: ADMIN__ACCOUNT_DELETE_RESPONCE – Foodgroup: 7 Type:

9

Foodgroup: PD

Група PD дозволяє настроїти відмови й дозволи.

SNAC: PD__ERROR – Foodgroup: 9 Type: 1

TLV Клас: PD__RIGHTS_REPLY_TAGS

SNAC: PD__RIGHTS_QUERY – Foodgroup: 9 Type: 2

SNAC: PD__RIGHTS_REPLY – Foodgroup: 9 Type: 3

SNAC: PD__SET_PRIVACY – Foodgroup: 9 Type: 4

SNAC: PD__ADD_VISIBLE – Foodgroup: 9 Type: 5

SNAC: PD__DEL_VISIBLE – Foodgroup: 9 Type: 6

SNAC: PD__ADD_INVISIBLE – Foodgroup: 9 Type: 7

SNAC: PD__DEL_INVISIBLE – Foodgroup: 9 Type: 8

SNAC: PD__SERVICE_ERROR – Foodgroup: 9 Type: 9

SNAC: PD__ADD_TEMP_PERMIT_LIST_ENTRIES – Foodgroup: 9 Type:

10

SNAC: PD__DEL_TEMP_PERMIT_LIST_ENTRIES – Foodgroup: 9 Type:

11

Foodgroup: LOOKUP

Дана група протоколу відповідає за пошук користувача по E-Mail. Вона поверне всі AIM-аккаунти користувача, які зареєстровані на цей E-Mail.

SNAC: LOOKUP__ERROR – Foodgroup: 10 Type: 1

SNAC: LOOKUP__QUERY – Foodgroup: 10 Type: 2

SNAC: LOOKUP__REPLY – Foodgroup: 10 Type: 3

Foodgroup: ROOM

Група протоколу яка використовує додатковий сервер, запитується через OSERVICE__SERVICE_REQUEST. Відповідає за створення й пошук чат кімнат.

SNAC: ROOM__ERROR – Foodgroup: 13 Type: 1

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

SNAC: ROOM__RIGHTS_QUERY – Foodgroup: 13 Type: 2

Тип даних: ROOM__RoomInfo

SNAC: ROOM__CREATE – Foodgroup: 13 Type: 8

TLV Клас: ROOM__INFO_TLV_TAGS

SNAC: ROOM__INFO_REPLY – Foodgroup: 13 Type: 9

TLV Клас: ROOM__TLV_TAGS

Foodgroup: BART

Група протоколу яка використовує додатковий сервер, запитується через OSERVICE__SERVICE_REQUEST. Відповідає за завантаження зображень і інших емоцій себе й користувачів зі списку контактів. Після завантаження даних, бажано, що б клієнт їх кешував.

Всі дані в BART мають свій унікальний BART_ID. BART_ID займає від 5 до 20 байт. Сервер BART поєднує однакові елементи, тобто якщо різні користувачі завантажать ідентичні дані на сервер, створиться тільки один елемент на сервер, і кожний з користувачів одержати однаковий BART_ID.

Клас: BART__ID_FLAGS

Клас: BART__ID_TYPES

Тип даних: BART__ID

Тип даних: BART__IDs_wName

Клас: BART__REPLY_CODES

Тип даних: BART__QUERY_REPLY_ID

SNAC: BART__ERROR – Foodgroup: 16 Type: 1

SNAC: BART__UPLOAD – Foodgroup: 16 Type: 2

SNAC: BART__UPLOAD_REPLY— Foodgroup: 16 Type: 3

SNAC: BART__DOWNLOAD – Foodgroup: 16 Type: 4

SNAC: BART__DOWNLOAD_REPLY – Foodgroup: 16 Type: 5

SNAC: BART__DOWNLOAD2 – Foodgroup: 16 Type: 6

SNAC: BART__DOWNLOAD_REPLY2 – Foodgroup: 16 Type: 7

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Foodgroup: FEEDBAG

FEEDBAG це список контактів і налаштувань AIM і ICQ які зберігаються на сервері для користувача. Клієнт має прямий доступ до керування FEEDBAG. Сервер здійснює деякі перевірки, для запобігання руйнування цього списку, але розроблювач клієнта повинен бути акуратним, що б не допустити псування FEEDBAG.

Class

Кожний запис в FEEDBAG має свій клас, що визначає характеристики цього запису, і як з нею працювати. Клас представлений числовим classId для кожного запису.

Item

Це основне складове FEEDBAG користувача. Елемент може бути групою, контактом, налаштуванням або чим-небудь ще. Всі елементи мають ім'я (яке може бути порожнім), itemId повинен бути унікальним. groupId і itemId можуть мати значення від 0 до 32767, значення від 32768 до 65535 зарезервовані. Клієнт сам вибирає значення для groupId і itemId (видавати їх можна в будь-якому порядку).

Group

Група це особливий елемент, що використовує itemId рівним нулю й classId рівним GROUP. Група з groupId 0 є кореневою групою. Всі групи мають ORDER атрибут, у якому описується в якому порядку повинні стояти групи або елементи FEEDBAG.

Замітки

Основне

FEEDBAG може містити нуль елементів. Він може мінятися як клієнтом, так і сервером. Всі рядки в FEEDBAG утримуються в UTF8 кодуванню.

Тільки користувач або сервер може видалити предмети з FEEDBAG. Клієнт не повинен видаляти атрибути або предмети зі списку якщо він не розуміє їхнього призначення, допускається видалення ушкоджених елементів.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Всі елементи в FEEDBAG є не обов'язковими, якщо клієнт не бачить елемент, або атрибут в елементі, то він повинен використовувати стандартне значення.

Коренева група

У будь-якому FEEDBAG утримуючої групи є «коренева група» яка має порожнє ім'я, groupId 0, itemId 0 і classId 1 (GROUP). Ця група містить атрибут ORDER який описує у якому порядку повинні йти групи. Якщо кореневої групи ні, то при додаванні нової групи, клієнт спочатку повинен створити кореневу групу. Кореневу групу не можна видалити, але якщо надіслати запит на її видалення (або очистити ORDER атрибут), те з FEEDBAG будуть вилучені всі групи й контакти.

Зверніть увагу, що деякі елементи можуть бути тільки в кореневій групі, а деякі тільки у звичайних групах.

Порядок в FEEDBAG

В FEEDBAG_REPLY елементи будуть упорядковані. Тобто, спочатку буде йти коренева група, потім різні елементи які знаходяться в ній (їх немає в ORDER), потім буде йти перша в ORDER атрибуті група, а потім контакти в цій групі.

Відображення списку контактів

Клієнт може використовувати кілька варіантів відображення списку контактів. Один із яких повинен бути «природним сортуванням», у якому контакти відображаються по ORDER-атрибуті, якщо користувач переміщає контакт, потрібно змінити ORDER на сервері.

Назви елементів

Назва кореневої групи повинне бути порожнім.

Назви інших груп повинні бути унікальними. Якщо є група «abc», можна створити групу «a b c», але назву «aBc» використовувати не можна.

Імена в елементів у кореневій групі повинні бути стислі й унікальні. Є їсти елемент із ім'ям «abc» і classId=PERMIT, не можна створити елемент із ім'ям

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

«aBc» або «a b c» с classId=PERMIT, але можна створити елемент із ім'ям «abc» і classId=DENY.

Елементи максимальна кількість яких лімітовано одним, повинні мати порожнє ім'я.

Клас: FEEDBAG__RIGHTS_QUERY_FLAGS

TLV Клас: FEEDBAG__RIGHTS_QUERY_TAGS

SNAC: FEEDBAG__ERROR – Foodgroup: 19 Type: 1

SNAC: FEEDBAG__RIGHTS_QUERY – Foodgroup: 19 Type: 2

TLV Клас: FEEDBAG__RIGHTS_REPLY_TAGS

SNAC: FEEDBAG__RIGHTS_REPLY – Foodgroup: 19 Type: 3

SNAC: FEEDBAG__QUERY – Foodgroup: 19 Type: 4

SNAC: FEEDBAG__CHECKOUT – Foodgroup: 19 Type: 5

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Вона складається з наступних блоків:

1. Блок обміну текстовими повідомленнями:

- Вікно користувачів.
- Вікно обміну повідомленнями.
- Вікно набору нового повідомлення.
- Вікно відображення даних про користувача.
- Вікно відображення аватарки.
- Вікно вибору виду зв'язку.
- Налаштування кольору та стилю шрифту повідомлень.
- Вікно повідомлень про набір тексту.

2. Блок авторизації:

- Введення логіну.
- Введення паролю.

3. Блок згортання у трей:

- Показати програму.
- Ваш статус.

- Налаштування.
- Змінити профіль.
- Закрити програму.
- 4. Блок відображення контактів, з розбиттям по групам і категоріям.
- 5. Блок обміну мовною інформацією.
- 6. Блок історії повідомлень по контактам.
- 7. Блок обміну відеоінформацією.
- 8. Головне меню:
 - Пристикувати список контактів.
 - Приватні списки.
 - Показати/змінити мої дані.
 - Налаштування.
 - Повідомлення анти-спам бота.
 - Сервісні повідомлення.
 - Мастер імпорту.
 - Перевірити наявність оновлень.
 - Про програму.
 - Додати обліковий запис.
 - Змінити профіль.
 - Закрити програму.
- 9. Блок налаштувань:
 - Загальні.
 - Список контактів.
 - Події.
 - Повідомлення.
 - Історія.
 - Статуси.
 - Анти-спам.
 - Звуки.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- Інтерфейс.
- Гарячі клавіші.
- Модулі.
- Підключення.
- Відео та звук.
- Облікові записи.

10. Блок встановлення статусів:

- Готовий спілкуватися.
- Злий.
- Депресія.
- Дома.
- На роботі.
- Ї'м.
- Відійшов.
- Недосяжний.
- Зайнятий.
- Не турбувати.
- У мережі.
- Невидимий.
- Невидимий для усіх.
- Не у мережі.

11. Блок пошуку.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3. Першим процесом, який завантажується у системі, є процес виведення вікна авторизації/реєстрації. Він взаємодіє з процесом введення логіна та паролю, який, у свою чергу, взаємодіє з процесом підключення до сервера. Процес підключення до сервера взаємодіє з процесом виведення головного вікна програми.

Процес виведення головного вікна програми взаємодіє з наступними процесами:

- Процес виведення списку контактів.
- Процес виведення списку функцій.
- Процес вибору співрозмовника зі списку контактів.
- Процес відключення від сервера.

Процес виведення списку контактів взаємодіє з наступними процесами:

- Процес пошуку контакту.
- Процес редагування списку контактів.

Процес виведення списку функцій взаємодіє з наступними процесами:

- Процес редагування своїх даних.
- Процес зміни свого статусу.
- Процес перегляду історії повідомлень.
- Процес роботи анти-спам фільтру.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 приведено блок-схему основної програми. Розглянемо по крокам роботу цієї програми.

Спочатку відбувається виведення вікна авторизації або реєстрації. Якщо користувач не зареєстрований, виконується процедура реєстрації нового користувача.

Далі, після введення логіну та паролю, програма здійснює пошук користувача за вказаним паролем у базі даних, після чого здійснюється перевірка введеного паролю.

Якщо логін та пароль введено вірно, здійснюється виведення головного вікна програми, в протилежному випадку, виконується виведення повідомлення про помилку авторизації. Також в головному вікні виводиться список контактів.

В процесі роботи можливе редагування списку контактів. При виборі такої опції виконується процедура пошуку нових контактів, після чого вибрані контакти додаються до бази даних. Також можливе видалення існуючих контактів та створення груп.

Основним режимом є робота із повідомленнями, яка здійснюється шляхом виконання наступних операцій:

- Вибір контакту із списку.
- Виведення вікна діалогів.
- Обмін повідомленнями.

Після цього за бажанням користувача обмін повідомленнями припиняється і програма завершує свою роботу.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55


```

        (Remplir = TRUE) and (LoginExisteDejaDesole = FALSE) then
begin
    StructureOrdinateur[j].LoginConnecte := login;
    StructureOrdinateur[j].NomOrdinateur := NomOrdi;
    StructureOrdinateur[j].Iden := NombreIdentifiant;
    StructureOrdinateur[j].img := 'Можу розмовляти';
    Remplir := FALSE;
end;
end;
MessageInfoArrive := 'Сервер'+>>' +Login+ ('+NomOrdi+')'+
з'єднаний...'+''';
for i:=0 to NumeroArriveConnexion-1 do
begin
    if LoginExisteDejaDesole then
        PasLeDroitDeSeConnecter(login,NomOrdi,NombreIdentifiant);
end;

```

Розглянемо більш детально реалізацію процесу обміну повідомленнями.

На рисунку 4.2 приведено блок-схему підпрограми обміну повідомленнями.

Робота підпрограми розпочинається виведенням вікна діалогу.

Для відправлення текстового повідомлення виконується така послідовність операцій:

- Введення текстового повідомлення.
- Здійснення форматування тексту.
- Додавання смайлів.
- Відправлення введеного повідомлення.

Якщо надійшло текстове повідомлення, відбувається виведення отриманого повідомлення в текстовому полі.

У випадку отримання аудіоповідомлення відбувається передача звуку на аудіопристрій та його програвання.

Для відправлення аудіоповідомлення виконується наступна послідовність операцій:

- Ввімкнення мікрофону.
- Запис аудіоповідомлення.
- Відправлення аудіоповідомлення.

Окремою підпрограмою реалізовано обмін файлами за вимогою користувача.

Після цього підпрограма завершує роботу.

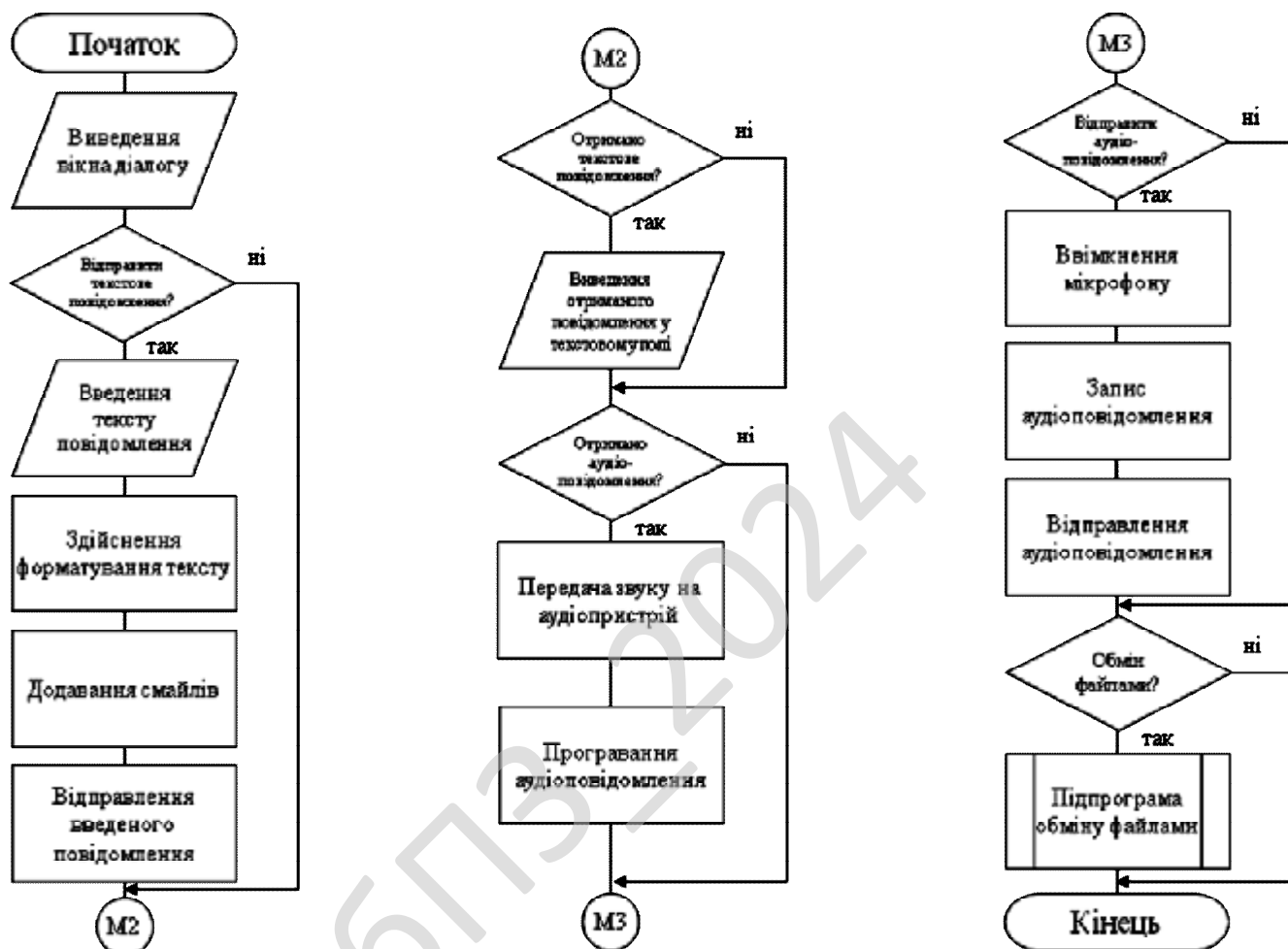


Рисунок 4.2 – Блок-схема підпрограми обміну повідомленнями

Для відправлення особистого повідомлення в залежності від статусу приймача використовується наступний код:

```

DS_SOURCE.DataSet.First;
while not DS_SOURCE.DataSet.Eof do
begin
    if (ClientDataSet1.NOMPrenom.Value = LoginEnvoi) then
    begin
        ClientDataSet1.Edit;
    end
end
    
```

```

        if MessageRecuClient = 'Зайнятий' then
ClientDataSet1IMAGE.Assign(IMG_PAUSE.Picture.Bitmap);
        if MessageRecuClient = 'Можу розмовляти' then
ClientDataSet1IMAGE.Assign(IMG_LIGNE.Picture.Bitmap);
        if MessageRecuClient = 'Не турбувати' then
ClientDataSet1IMAGE.Assign(IMG_OCCUPE.Picture.Bitmap);
        if MessageRecuClient = 'Розмовляю по телефону' then
ClientDataSet1IMAGE.Assign(IMG_TEL.Picture.Bitmap);
        if MessageRecuClient = 'Тимчасово відсутній' then
ClientDataSet1IMAGE.Assign(IMG_1MN.Picture.Bitmap);
        if MessageRecuClient = 'Зайнятий' then
ClientDataSet1IMAGE.Assign(IMG_ABS.Picture.Bitmap);
        ClientDataSet1.Post;
        end;
        DS_SOURCE.DataSet.Next;
    end;
end;
end;
if CompareStr(copy(MessageRecuClient, 1, 8), 'MsgPrive') = 0 then
begin
    MessageRecuClient := Gauche('#ЗАВЕРШЕННЯ РОБОТИ#', MessageRecuClient);
    MessageRecuClient := Droite('MsgPrive#DE#', MessageRecuClient);
    LoginEnvoi := Gauche('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#', MessageRecuClient);
    MessageRecuClient := Droite('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#',
MessageRecuClient);
    LoginRecoi := Gauche('#ТІЛО ПОВІДОМЛЕННЯ#', MessageRecuClient);
    if (Comparestr(LoginEnvoi, EditLogin) = 0) or (Comparestr(LoginRecoi,
EditLogin) = 0) then
        begin
            MessageTexte := Droite('#ТІЛО ПОВІДОМЛЕННЯ#', MessageRecuClient);
            if beep then
                MessageBeep(MB_OK);
            RendreVisiblePremierPlan;
            if (Memo_MITTEVI_POVIDOMLENNYA_.Lines.Count > 10) then
                Memo_MITTEVI_POVIDOMLENNYA_.Lines.Clear;
            if (LoginEnvoi <> UserName) then
                begin
                    if assigned(MsgPerso) then
                        begin
                            if (MsgPerso.Caption <> 'Хочете відправити особисте повідомлення ' -
LoginEnvoi + ' :') then
                                begin
                                    MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);

```

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

```

        MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :';
        MsgPerso.cxGroupBox2.Caption := ' A : ' + LoginEnvoi + ' ';
        MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
        MsgPerso.Show;
        FlashWindow(Application.Handle, true);
    end
else
begin
    MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
    FlashWindow(Application.Handle, true);
end;
end
else
begin
    MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
    MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :';
    MsgPerso.cxGroupBox2.Caption := ' A : ' + LoginEnvoi + ' ';
    MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
    MsgPerso.Show;
    FlashWindow(Application.Handle, true);
end;
end
else
    MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
end;
end;

```

Наведемо частину програмного коду для виставлення статусу клієнта:

```

beep := False;
Registre := TRegistry.Create;
Registre.RootKey := HKEY_LOCAL_MACHINE;
Registre.OpenKey('\Software\IntraMSN\Couleur\', True);
Registre.OpenKey('\Software\IntraMSN\Son\', True);
if Registre.ValueExists('Son') then
    if Registre.ReadInteger('Son') = 1 then
        beep := True;
    MessageRecuClient := Msg;
    if (CompareStr(copy(MessageRecuClient, 1, 5), '') = 0) and
(CompareStr(copy(MessageRecuClient, length(MessageRecuClient) - 5, 6),
'¤@*!@¤') = 0) then

```

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

```

begin
  k := 0;
  fillchar(TableauLocal, sizeof(TableauLocal), 0);
  for j := 1 to length(MessageRecuClient) - 5 do
  begin
    if MessageRecuClient[j] = 'µ' then
    begin
      inc(k);
      TableauLocal[k] := j;
    end;
  end;
  if k <> 0 then
  begin
    for l := 1 to k - 1 do
    begin
      Personnage := copy(MessageRecuClient, TableauLocal[l] + 1,
(TableauLocal[l + 1]) - TableauLocal[l] - 1);
      if (CompareStr(Personnage, '' ) <> 0) and (comparestr(Personnage, ' ')
<> 0) then
      begin
        Image := Gauche('#ORDI#', Droite('#IMG#', Personnage));
        Ordi := Droite('#ORDI#', Personnage);
        Personnage := Gauche('#IMG#', Personnage);
        Ajouter := TRUE;
        DS_SOURCE.DataSet.First;
        while not (DS_SOURCE.DataSet.Eof) do
        begin
          if comparestr(ClientDataSet1NOMPrenom.Value, Personnage) = 0 then
          begin
            Ajouter := FALSE;
          end;
          DS_SOURCE.DataSet.Next;
        end;
        DS_SOURCE.DataSet.First;
        if Ajouter then
        begin
          DS_SOURCE.DataSet.Insert;
          nbpersonne := nbpersonne + 1;
          ClientDataSet1N_ID.Value := nbpersonne;
          if Image = 'Зайнятий' then
            ClientDataSet1IMAGE.Assign(IMG_PAUSE.Picture.Bitmap);

```

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

```

        if Image = 'Можу розмовляти' then
ClientDataSet1IMAGE.Assign(IMG_LIGNE.Picture.Bitmap);
        if Image = 'Не турбувати' then
ClientDataSet1IMAGE.Assign(IMG_OCCUPE.Picture.Bitmap);
        if Image = 'Розмовляю по телефону' then
ClientDataSet1IMAGE.Assign(IMG_TEL.Picture.Bitmap);
        if Image = 'Тимчасово відсутній' then
ClientDataSet1IMAGE.Assign(IMG_LMN.Picture.Bitmap);
        if Image = 'Зайнятий' then
ClientDataSet1IMAGE.Assign(IMG_ABS.Picture.Bitmap);
        ClientDataSet1NOMPrenom.Value := Personnage;
        ClientDataSet1ORDI.Value := Ordi;
        ClientDataSet1N_GROUPE.Value := 1;
        DS_SOURCE.DataSet.Post;
        end;
    end;
end;
end;
end;
end;

```

Для вибору відправлення файлу виконуються наступний код:

```

newlabel := TLabel.Create(MsgPerso.Mem01);
with newlabel do
begin
Parent := MsgPerso.Mem01;
Caption := 'testttt';
end;

MsgPerso.Mem01.Lines.InsertObject(0, 'Test', SpeedButton3);
MsgPerso.Show;
FlashWindow(Application.Handle, true);
end
else
begin
MsgPerso.Mem01.Lines.Add(LoginEnvoi + ' Ви відправили файл: ' +
MessageTexte);
FlashWindow(Application.Handle, true);
end;
end
else
begin
MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
MsgPerso.Caption := + LoginEnvoi + ' :';

```

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63


```

    ComboBox6.Items.Add (Mixer.ProductName);
If (ComboBox6.Items.Count > 0) then
    ComboBox6.ItemIndex := 0;
ComboBox6Change (Sender);
If (ComboBox6.ItemIndex >= 0) AND (ComboBox6.ItemIndex < Mixer.MixerCount)
then
    Mixer.MixerId := ComboBox6.ItemIndex;
ComboBox4.Items.Clear;
If Mixer.MixerCount>0 then
begin
    For A:=1 to Mixer.Destinations.Count-1 do
        ComboBox4.Items.Add (Mixer.Destinations[A].Data.szName);
    If ComboBox4.Items.Count>0 then
        begin
            ComboBox4.ItemIndex:=0;
            ComboBox4Change (Self);
        end;
    end
else
begin
    ComboBox4.OnChange:=nil;
    ComboBox5.OnChange:=nil;
    TrackBar.OnChange:=nil;
    CheckBox.OnClick:=nil;
    MessageDlg ('Немає змішувача у системі!', mtError, [mbOK], 0);
end;
Setting:=False;
end;

```

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 9041:2020. Його повна назва: ДСТУ 9041:2020. Інформаційні технології. Криптографічний захист інформації. Алгоритм шифрування коротких повідомлень, що ґрунтується на скручених еліптичних кривих Едвардса.

Цей алгоритм призначений для шифрування коротких (до 616 біт) повідомлень для будь-яких алгоритмів шифрування, в тому числі визначених національними стандартами України.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Як і стандарт цифрового підпису ДСТУ 4145:2002, новий алгоритм використовує криптографічні перетворення у групі точок еліптичних кривих, використовуючи замість кривих у формі Вейерштрасса найновітніші розробки у галузі еліптичної криптографії – криві у формі Едвардса. Це дає суттєві переваги у швидкодії більш ніж у 3 рази. Новий стандарт розроблений з урахуванням усіх найсучасніших вимог до стійкості криптографічних алгоритмів. Так, нижня межа стійкості криптосистем у цьому стандарті дорівнює 2127 (≈ 1042) (це більш ніж у півтора рази вище, ніж у ДСТУ 4145) і можуть бути обрані інші рівні, такі як 2255 (≈ 1085), 2383 (≈ 10127) та 2767 (≈ 10255); крім того, строго обґрунтована його стійкість як до атак на відновлення відкритого тексту, так і до розрізняючих атак.

Проект алгоритму шифрування, що ліг в основу цього стандарту, пройшов апробацію як в Україні, так і за її межами (Центрально-Європейська конференція з криптографії (червень 2020 року) – форум ведучих криптологів з усього світу).

Стандарт ДСТУ-9041 узгоджений з усіма діючими в Україні національними стандартами. Новиною стандарту є його сфера застосування – інкапсуляція ключів, найсучасніший математичний апарат, а також новий алгоритм генерації псевдовипадкових послідовностей, який, на відміну від аналогічного алгоритму генерації з ДСТУ 4145, використовує виключно національні криптографічні алгоритми національних стандартів та не містить посилань на відповідні пост-радянські стандарти, термін дії яких вже практично вичерпався.

Новий стандарт не належить до так званих постквантових стандартів. Але його стійкість буде під загрозою лише тоді, коли з'являться квантові комп'ютери з 700 і більше кубітами (на даний час кількість "робочих" кубітів, які вдалося створити, – близько 50). Його перевагою перед постквантовими алгоритмами є відносно невелика довжина ключа (у десятки або навіть у сотні разів менша, ніж у постквантових алгоритмах).

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо інтерфейс програми. Головне вікно розробленого програмного забезпечення приведено на рисунку 5.1.

З правого боку виводиться список співрозмовників із графічним зображенням їх статусів. Основну частину займає вікно виводу діалогу з поточним співрозмовником, реалізоване за допомогою вкладок. Нижче розташоване поле текстового вводу із кнопками «Цитувати», «Смайли», «Відправити повідомлення», «Закрити вкладку».

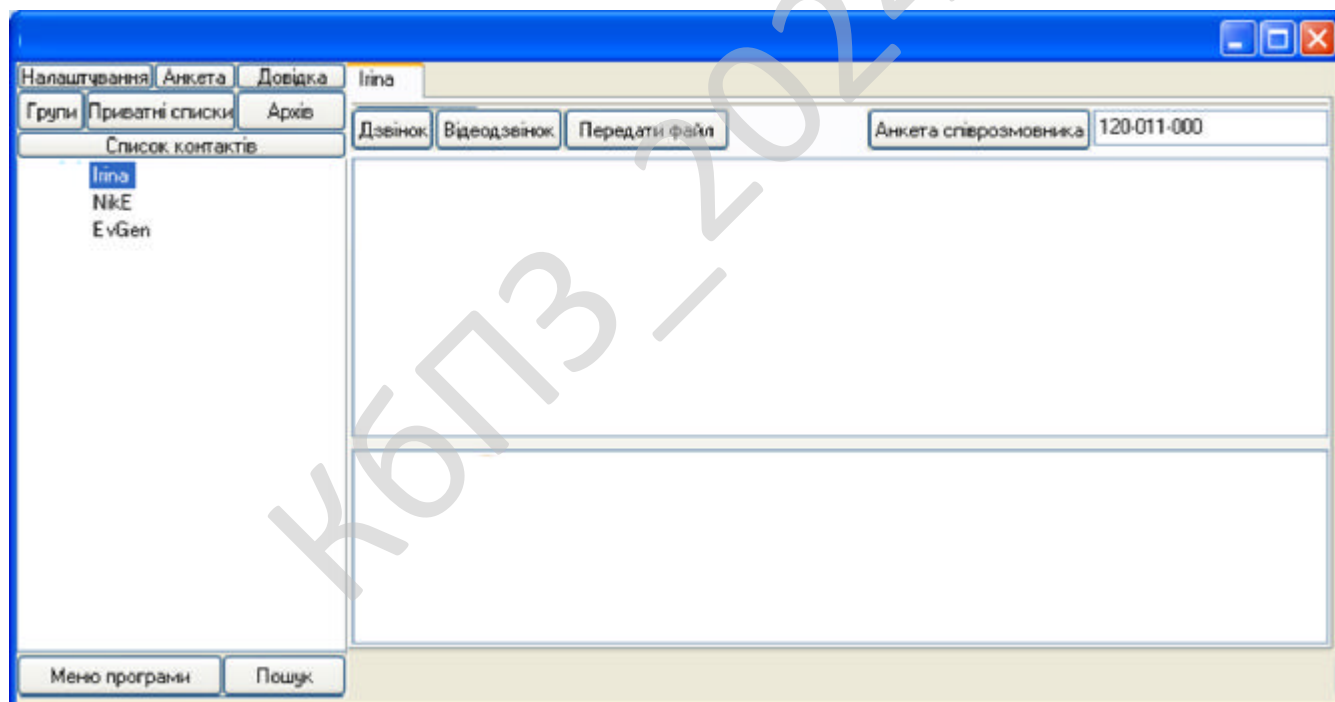


Рисунок 5.1 – Головне вікно програми

Для зміни параметрів програми призначені кнопки:

- «Меню програми».
- «Налаштування».

- «Приватні списки».
- «Змінити статус».

Для передачі файлів призначена кнопка «Передати файл». Здійснення аудіодзвінка відбувається натисненням на кнопку «Аудіодзвінок». Також є функція «Відеодзвінок».

Для перегляду інформації про співрозмовників призначені наступні кнопки:

- «Анкета співрозмовника».
- «Архів» (архів текстових повідомлень).

Пошук співрозмовників здійснюється за допомогою кнопки «Пошук». Редагувати та структурувати список контактів можна за допомогою функцій, що вмикаються кнопками «Список контактів» та «Групи» відповідно.

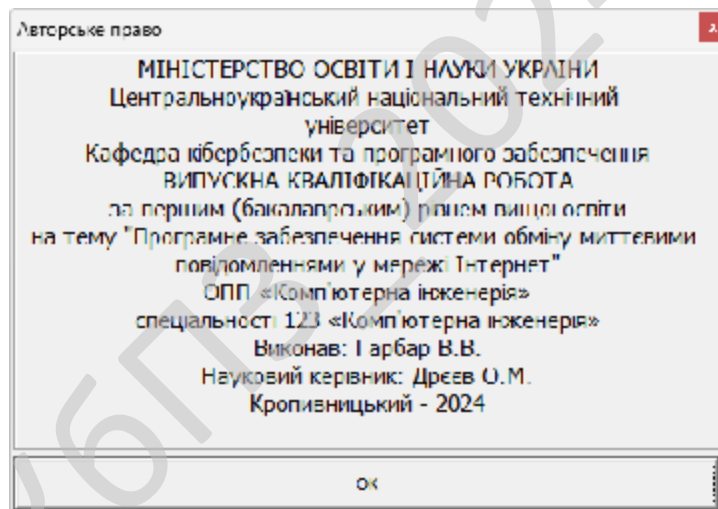


Рисунок 5.2 – Вікно «Про програму...»

На рисунку 5.2 зображено вікно «Про програму...», яке виводиться після натискання кнопки «Довідка».

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи обміну миттєвими повідомленнями у мережі Інтернет.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем обміну миттєвими повідомленнями у мережі Інтернет.
- Досліджена система обміну миттєвими повідомленнями у мережі Інтернет.
- На основі отриманих результатів досліджень створена програмна реалізація системи обміну миттєвими повідомленнями у мережі Інтернет.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання обміну миттєвими повідомленнями у мережі Інтернет.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи обміну миттєвими повідомленнями у мережі Інтернет. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 9041:2020.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

КБПЗ_2024

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.

2. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

3. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

4. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

5. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

6. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

7. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science*, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

8. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems:

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

9. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

10. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

11. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

12. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

13. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

14. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

15. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

16. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

17. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

18. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

19. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

20. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

21. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

22. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

23. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

24. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

25. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

26. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

27. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

28. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

29. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

30. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

31. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

32. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

33. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

34. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

35. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

36. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» *Комп'ютерні науки та кібербезпека*. № 4. С. 30-37. 2019.

37. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. *Проектування комп'ютерних систем та мереж*. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

38. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

39. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

40. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

41. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

42. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

43. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

44. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67

45. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

науково-практичної конференції “Проблеми та перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. - 2014. - С. 240.

46. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лєвошко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.

47. Смірнов О.А., Дреєв О.М., Дорєнський О.П. «Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі. Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.

48. Смірнов О.А., Дорєнський О.П., Дреєв О.М. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів. Наука і техніка Повітряних сил Збройних Сил України. – Випуск 3(12). – Х.: ХУПС. – 2013. – С.122-127.

49. Смірнов О.А., Мєлєшко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

50. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мєлєшко Є.В. Технології і стандарти комп'ютерних мереж. – Кіровоград: КНТУ 2012. – 454 с

					ВКРБ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

					ВКРБ-123.24.0017.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Гарбар В.В.				<i>Програмне забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.					Б	1	6
Н. Контр.	Коваленко А.С				ЦНТУ КІ-20			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи обміну миттєвими повідомленнями у мережі Інтернет.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 131-02 від 01.04.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи обміну миттєвими повідомленнями у мережі Інтернет.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-123.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи обміну миттєвими повідомленнями у мережі Інтернет;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.4.

					ВКРБ-123.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 78 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					ВКРБ-123.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 14.06.2024 р.

					ВКРБ-123.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти

_____ Дреєв О.М.

*Програма забезпечення системи обміну миттєвими повідомленнями у
мережі Інтернет*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 54

Літера: РП

Кропивницький – 2024 року

Interface_korystuvacha.pas - параметри інтерфейсу клієнтської частини

```

unit Interface_korystuvacha;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, Registry, WinSkinStore, WinSkinData,
  ComCtrls, ExtDlgs;

type
  TChoixCouleur = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox1: TGroupBox;
    RadioGroupSons: TRadioGroup;
    RadioGroupVisible: TRadioGroup;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    GroupBox4: TGroupBox;
    Image5: TImage;
    GroupBox5: TGroupBox;
    StaticText3: TStaticText;
    Couleur: TComboBox;
    Panell: TPanel;
    BitBtn1: TBitBtn;
    GroupBox6: TGroupBox;
    GroupBox7: TGroupBox;
    GroupBox10: TGroupBox;
    GroupBox11: TGroupBox;
    CheckBox4: TCheckBox;
    CheckBox5: TCheckBox;
    GroupBox12: TGroupBox;
    Absent: TCheckBox;
    Temps: TEdit;
    OpenTextFileDialog1: TOpenTextFileDialog;
    SpeedButton1: TSpeedButton;
    GroupBox8: TGroupBox;
    Label1: TLabel;
    Fichier: TEdit;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Image1Click(Sender: TObject);
    procedure CouleurClick(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
  private
    { Private declarations }
    procedure SystemeCommande(var Msg: TMessage);
      message WM_SysCommand; // перехоплення повідомлень SysCommand

    procedure ReadSkinfile( apath : string );

  public
    { Public declarations }
  end;

var
  ChoixCouleur: TChoixCouleur;

implementation

```

```

uses Unit_MITTEVI_POVIDOMLENNYA_RL, Unit_MsgPerso;

{$R *.dfm}

procedure TChoixCouleur.FormCreate(Sender: TObject);
var
  Registre : TRegistry;
begin
  readskinfile(path);

  Registre:=TRegistry.Create;
  Registre.RootKey:=HKEY_LOCAL_MACHINE;

  Registre.OpenKey('\Software\IntraMSN\Couleur\',True);
  if Registre.ValueExists('Couleur') then
    Couleur.Text:=Registre.ReadString('Couleur');

  Registre.OpenKey('\Software\IntraMSN\Son\',True);
  if Registre.ValueExists('Son') then
    RadioGroupSons.ItemIndex := Registre.ReadInteger('Son');

  Registre.OpenKey('\Software\IntraMSN\Afficher\',True);
  if Registre.ValueExists('Afficher') then
    RadioGroupVisible.ItemIndex := Registre.ReadInteger('Afficher');

  Registre.OpenKey('\Software\IntraMSN\Image\',True);
  if Registre.ValueExists('Image') then
  begin
    Image5.Width := Registre.ReadInteger('Image');
    Image5.Height := Registre.ReadInteger('Image');
  end;

  Registre.OpenKey('\Software\IntraMSN\Transfert\',True);
  if Registre.ValueExists('Fichier') then
    Fichier.Text := Registre.ReadString('Fichier')
  else
    Fichier.Text := root;

  Registre.OpenKey('\Software\IntraMSN\Absent\',True);
  if Registre.ValueExists('Activer') then
    Absent.Checked := Registre.ReadBool('Activer');
  if Registre.ValueExists('Activer') then
    Temps.Text := Registre.ReadString('Temps');

  Registre.CloseKey;
  Registre.Free;
end;

procedure TChoixCouleur.BitBtn1Click(Sender: TObject);
var
  Registre : TRegistry;
begin
  Registre:=TRegistry.Create;
  Registre.RootKey:=HKEY_LOCAL_MACHINE;

  Registre.OpenKey('\Software\IntraMSN\Couleur\',True);
  Registre.WriteString('Couleur', _MITTEVI_POVIDOMLENNYA_RL.sd1.SkinFile);

  Registre.OpenKey('\Software\IntraMSN\Son\',True);
  Registre.WriteInteger('Son', RadioGroupSons.ItemIndex);

  Registre.OpenKey('\Software\IntraMSN\Afficher\',True);
  Registre.WriteInteger('Afficher', RadioGroupVisible.ItemIndex);

  Registre.OpenKey('\Software\IntraMSN\Image\',True);
  Registre.WriteInteger('Image', Image5.Width);

```

```

_MITTEVI_POVIDOMLENNYA_RL.ClientDataSet1IMAGE.DisplayWidth :=
Round(Image5.Width / 6);

Registre.OpenKey('\Software\IntraMSN\Absent\ ', True);
Registre.WriteBool('Activer', Absent.Checked);
Registre.WriteString('Temps', Temps.Text);

Registre.OpenKey('\Software\IntraMSN\Transfert\ ', True);
Registre.WriteString('Fichier', Fichier.Text);

Registre.CloseKey;
Registre.Free;

ChoixCouleur.Close;
end;

procedure TChoixCouleur.SpeedButton1Click(Sender: TObject);
begin
  if (OpenTextFileDialog1.Execute) Then
    Fichier.Text := OpenTextFileDialog1.FileName;
end;

procedure TChoixCouleur.SystemeCommande(var Msg: TMessage);
begin
  if Msg.wParam = sc_Close then ;
end;

procedure TChoixCouleur.Image1Click(Sender: TObject);
var
  i : integer;
begin
  Image5.Width := (Sender as TImage).Width;
  Image5.Height := (Sender as TImage).Height;
end;

procedure TChoixCouleur.CouleurClick(Sender: TObject);
begin
  _MITTEVI_POVIDOMLENNYA_RL.sdl.SkinFile:=path+Couleur.Text;
  if not _MITTEVI_POVIDOMLENNYA_RL.sdl.Active then
    _MITTEVI_POVIDOMLENNYA_RL.sdl.Active:=true;
end;

procedure TChoixCouleur.ReadSkinfile( apath : string );
var
  sts: Integer ;
  SR: TSearchRec;
  list: Tstringlist;

  procedure AddFile;
  begin
    list.add(sr.name);
  end;
begin
  list:=Tstringlist.create;
  sts := FindFirst( apath + '*.skn' , faAnyFile , SR );
  if sts = 0 then begin
    if ( SR.Name <> '.' ) and ( SR.Name <> '..' ) then begin
      if pos('.', SR.Name) <> 0 then
        Addfile;
    end;
    while FindNext( SR ) = 0 do begin
      if ( SR.Name <> '.' ) and ( SR.Name <> '..' ) then begin
        if Pos('.', SR.Name) <> 0 then Addfile;
      end;
    end;
  end ;
  FindClose( SR ) ;
  list.sort;
  Couleur.items.assign(list);
  list.free;
end;
end;

```

КБПЗ_2024

Unit_MsgPerso.pas - передача текстових повідомлень

```

unit Unit_MsgPerso;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, cxGraphics, cxControls, cxLookAndFeels, cxLookAndFeelPainters,
  cxContainer, cxEdit, IdAntiFreezeBase, IdAntiFreeze, IdCustomTCPServer,
  IdTCPServer, IdCmdTCPServer, IdExplicitTLSClientServerBase, IdFTPServer,
  IdUserAccounts, IdComponent, IdTCPConnection, IdTCPClient, IdFTP,
  IdBaseComponent, IdZLibCompressorBase, IdCompressorZLibEx, ExtCtrls,
  cxProgressBar, cxSplitter, StdCtrls, cxGroupBox, Buttons,
  IdFTPListOutput, Registry, IdFTPList, IdWinSock2 ;

type
  TMsgPerso = class(TForm)
    Panell: TPanel;
    SpeedButton5: TSpeedButton;
    Login: TLabel;
    SpeedButton1: TSpeedButton;
    BitBtnEnvoyermsgPrive: TBitBtn;
    BitBtn1: TBitBtn;
    Panel2: TPanel;
    cxGroupBox1: TcxGroupBox;
    LabeledEdit1: TEdit;
    cxGroupBox2: TcxGroupBox;
    Memo1: TMemo;
    cxSplitter1: TcxSplitter;
    FontDialog1: TFontDialog;
    Ouvrir: TOpenDialog;
    TimerWizz: TTimer;
    IdCompressorZLibEx2: TIdCompressorZLibEx;
    IdFTP1: TIdFTP;
    IdUserManager1: TIdUserManager;
    IdFTPServer1: TIdFTPServer;
    IdAntiFreeze1: TIdAntiFreeze;
    Timer1: TTimer;
    Transfert: TPanel;
    cxProgressBar1: TcxProgressBar;
    Taux_Transfert: TLabel;
    Niveau_Transfert: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    procedure IdFTPServer1StoreFile(ASender: TIdFTPServerContext;
      const AFileName: string; AAppend: Boolean; var VStream: TStream);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure BitBtnEnvoyermsgPriveClick(Sender: TObject);
    procedure TimerWizzTimer(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure IdFTPServer1RetrieveFile(ASender: TIdFTPServerContext;
      const AFileName: string; var VStream: TStream);
    procedure IdFTPServer1RemoveDirectory(ASender: TIdFTPServerContext;
      var VDirectory: string);
    procedure IdFTPServer1MakeDirectory(ASender: TIdFTPServerContext;
      var VDirectory: string);
    procedure IdFTPServer1ListDirectory(ASender: TIdFTPServerContext;
      const APath: string; ADirectoryListing: TIdFTPListOutput; const ACmd,
      ASwitches: string);
    procedure IdFTPServer1GetFileSize(ASender: TIdFTPServerContext;
      const AFilename: string; var VFileSize: Int64);
    procedure IdFTPServer1DeleteFile(ASender: TIdFTPServerContext;
      const APathName: string);
    procedure IdFTPServer1ChangeDirectory(ASender: TIdFTPServerContext;

```

```

    var VDirectory: string);
    procedure IdFTP1Work(ASender: TObject; AWorkMode: TWorkMode;
        AWorkCount: Integer);
    procedure IdFTP1WorkBegin(ASender: TObject; AWorkMode: TWorkMode;
        AWorkCountMax: Integer);
    procedure IdFTP1WorkEnd(ASender: TObject; AWorkMode: TWorkMode);
    procedure FontDialog1Apply(Sender: TObject; Wnd: HWND);
private
    function ReplaceChars(APath: string): string;
    function GetSizeOfFile(AFile: string): Integer;
public
    FileSize: integer;
    FileName: string;
    STime: extended;
    AbortTransfer: boolean;
end;

var
    MsgPerso: TMsgPerso;
    Bouge: Boolean;
    cpt: integer = 0;
    AppDir: string;

implementation

uses IdFTPCommon, Unit_MITTEVI_POVIDOMLENNYA_RL;

{$R *.dfm}

var
    ServeurEnReception: Boolean = False;
    ServeurAdresseClient: string = '';
    ClientFichier: file;

function GaucheNDroite(substr: string; s: string; n: integer): string;
var i: integer;
begin
    S := S + substr;
    for i := 1 to n do
        begin
            S := copy(s, pos(substr, s) + length(substr), length(s) - pos(substr, s) +
                length(substr));
        end;
    result := copy(s, 1, pos(substr, s) - 1);
end;

function TrouverNomLoginQuiRecoitMsg: string;
var
    Login: string;
begin
    Login := MsgPerso.Caption;
    Login := GaucheNDroite(' ', Login, 7);
    result := Login;
end;

procedure TMsgPerso.FontDialog1Apply(Sender: TObject; Wnd: HWND);
begin
    //LabeledEdit1.Font := FontDialog1.Font;
end;

procedure TMsgPerso.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    if ServeurEnReception and
        (messagedlg('Ви впевнені, що хочете покинути програму?', mtconfirmation,
            [mbYes, mbNo], 0) = mrNo) then
        begin
            //нічого не робить
        end
end

```

```

else
begin
  IdFTP1.Disconnect;
  MsgPerso.Destroy;
  MsgPerso := nil;
end;
end;

function TMsgPerso.ReplaceChars(APath: string): string;
var
  s: string;
begin
  s := StringReplace(APath, '/', '\', [rfReplaceAll]);
  s := StringReplace(s, '\\', '\', [rfReplaceAll]);
  Result := s;
end;

//Створення форми
procedure TMsgPerso.FormCreate(Sender: TObject);
var
  Registre: TRegistry;
  tmpCouleur: integer;
begin
  IdFTPServer1.Active := True;
  if IdFTP1.Connected then IdFTP1.Disconnect;
end;

//Визначення розміру файлу
function TMsgPerso.GetSizeOfFile(AFile: string): Integer;
var
  FStream: TFileStream;
begin
  try
    FStream := TFileStream.Create(AFile, fmOpenRead);
    try
      Result := FStream.Size;
    finally
      FreeAndNil(FStream);
    end;
  except
    Result := 0;
  end;
end;

procedure TMsgPerso.IdFTP1Work(ASender: TObject; AWorkMode: TWorkMode;
  AWorkCount: Integer);
var
  S: string;
  TotalTime: TDateTime;
  H, M, Sec, MS: Word;
  DLTime: Double;
  AverageSpeed: extended;
begin
  Application.ProcessMessages;
  TotalTime := Now - STime;
  DecodeTime(TotalTime, H, M, Sec, MS);
  Sec := Sec + M * 60 + H * 3600;
  DLTime := Sec + MS / 1000;
  if DLTime > 0 then
  begin
    AverageSpeed := (AWorkCount / 1024) / DLTime;
    S := FormatFloat('0.00 Kb/s', AverageSpeed);
    Taux_Transfert.Caption := S;
  end;
  // if AbortTransfer then IdFTP1.Abort;
  cxProgressBar1.EditValue := AWorkCount/FileSize * 100;

```

```

    Niveau_Transfert.Caption := IntToStr(AWorkCount) + '/' + IntToStr(FileSize) +
    ' octets';
end;

procedure TMsgPerso.IdFTP1WorkBegin(ASender: TObject; AWorkMode: TWorkMode;
    AWorkCountMax: Integer);
begin
    ServeurEnReception := True;
    AbortTransfer := false;
    STime := Now;
    Memo1.Lines.Add(FileName);
    Taux_Transfert.Caption := '0.00 Kb/s';
    if FileSize < AWorkCountMax then FileSize := AWorkCountMax;
    Niveau_Transfert.Caption := '0 / ' + IntToStr(FileSize) + ' octets';
end;

procedure TMsgPerso.IdFTP1WorkEnd(ASender: TObject; AWorkMode: TWorkMode);
begin
    // SpeedButton2.Visible := false;
    //Memo1.Lines.Clear;
    //if AbortTransfer then Memo2.Lines.Add('Скасування надсилання: ' + FileName)
    // else
    if (FileName <> '') then Memo1.Lines.Add('Передача даних: ' + FileName);
    FileSize := 0;
    FileName := '';
    IdFTP1.Disconnect;
    ServeurEnReception := False;
end;

procedure TMsgPerso.IdFTPServer1ChangeDirectory(ASender: TIdFTPServerContext;
    var VDirectory: string);
begin
    ASender.CurrentDir := VDirectory;
end;

procedure TMsgPerso.IdFTPServer1DeleteFile(ASender: TIdFTPServerContext;
    const APathName: string);
begin
    DeleteFile(ReplaceChars(AppDir + ASender.CurrentDir + '\' + APathname));
end;

procedure TMsgPerso.IdFTPServer1GetFileSize(ASender: TIdFTPServerContext;
    const AFilename: string; var VFileSize: Int64);
var
    LFile: string;
begin
    LFile := ReplaceChars(AppDir + AFilename);
    try
        if FileExists(LFile) then
            VFileSize := GetSizeOfFile(LFile)
        else
            VFileSize := 0;
    except
        VFileSize := 0;
    end;
end;

procedure TMsgPerso.IdFTPServer1ListDirectory(ASender: TIdFTPServerContext;
    const APath: string; ADirectoryListing: TIdFTPListOutput; const ACmd,
    ASwitches: string);
var
    LFTPItem: TIdFTPListItem;
    SR: TSearchRec;
    SRI: Integer;
begin
    ADirectoryListing.DirFormat := doUnix;
    SRI := FindFirst(AppDir + APath + '\*.*', faAnyFile - faHidden - faSysFile,
    SR);
    while SRI = 0 do

```

```

begin
  LFTPItem := ADirectoryListing.Add;
  LFTPItem.FileName := SR.Name;
  LFTPItem.Size := SR.Size;
  LFTPItem.ModifiedDate := FileDateToDateTime(SR.Time);
  if SR.Attr = faDirectory then
    LFTPItem.ItemType := ditDirectory
  else
    LFTPItem.ItemType := ditFile;
  SRI := FindNext(SR);
end;
FindClose(SR);
SetCurrentDir(AppDir + APath + '\..');
end;

procedure TMsgPerso.IdFTPServer1MakeDirectory(ASender: TIdFTPServerContext;
  var VDirectory: string);
begin
  if not ForceDirectories(ReplaceChars(AppDir + VDirectory)) then
    begin
      raise Exception.Create('Неможливо створити директорію');
    end;
end;

procedure TMsgPerso.IdFTPServer1RemoveDirectory(ASender: TIdFTPServerContext;
  var VDirectory: string);
var
  LFile: string;
begin
  LFile := ReplaceChars(AppDir + VDirectory);
end;

procedure TMsgPerso.IdFTPServer1RetrieveFile(ASender: TIdFTPServerContext;
  const AFileName: string; var VStream: TStream);
begin
  VStream := TFileStream.Create(ReplaceChars(AppDir + AFilename), fmOpenRead);
end;

procedure TMsgPerso.IdFTPServer1StoreFile(ASender: TIdFTPServerContext;
  const AFileName: string; AAppend: Boolean; var VStream: TStream);
begin
  if not AAppend then
    VStream := TFileStream.Create(ReplaceChars(AppDir + AFilename), fmCreate)
  else
    VStream := TFileStream.Create(ReplaceChars(AppDir + AFilename), fmOpenWrite)
end;

function TrouverIP(ordinateur: string): string;
var
  WSADATA: TWSADATA;
  Name, Address: string;
  Phe: PHostEnt;
begin
  WSASStartup(2, WSADATA);
  SetLength(Name, 255);
  Phe := GetHostByName(PChar(ordinateur));
  with Phe^ do
    Address := Format('%d.%d.%d.%d', [Byte(h_addr^[0]), Byte(h_addr^[1]),
    Byte(h_addr^[2]), Byte(h_addr^[3])]);
  WSACleanup;
  TrouverIP := Address;
end;

procedure TMsgPerso.SpeedButton1Click(Sender: TObject);
var
  MessageBouger: string;
begin

```

```

    MessageBouger := 'MsgBouger' + '#DE#' + EditLogin + '#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#'
+ TrouverNomLoginQuiRecoitMsg + '#ТІЛО ПОВІДОМЛЕННЯ#vient de te rйveiller !
:'))#ЗАВЕРШЕННЯ РОБОТИ#';
    _MITTEVI_POVIDOMLENNYA_RL.ClientSocket.Socket.SendText(#13 + MessageBouger);
    Bouge := True;
end;

procedure TMsgPerso.SpeedButton5Click(Sender: TObject);
var
    MessageTransfert: string;
begin
    if not Ouvrir.Execute then Exit;
    AssignFile(ClientFichier, Ouvrir.FileName);
    MessageTransfert := 'MsgTransfert' + '#DE#' + EditLogin + '#ЗАГОЛОВОК
ПОВІДОМЛЕННЯ#' + TrouverNomLoginQuiRecoitMsg + '#ТІЛО ПОВІДОМЛЕННЯ#' +
ExtractFileName(Ouvrir.FileName) + '#ЗАВЕРШЕННЯ РОБОТИ#';
    _MITTEVI_POVIDOMLENNYA_RL.ClientSocket.Socket.SendText(#13 +
MessageTransfert);
    if not IdFTP1.Connected then
        begin
            IdFTP1.Host := TrouverIP(Login.Caption);
            IdFTP1.UserName := 'root';
            IdFTP1.Password := '';
            IdFTP1.Connect;
        end;
    if (IdFTP1.Connected) then
        begin
            Transfert.Visible := True;
            IdFTP1.TransferType := ftBinary;
            IdFTP1.Put(Ouvrir.FileName, ExtractFileName(Ouvrir.FileName));
            Reset(ClientFichier, 1);
        end;
end;

procedure TMsgPerso.TimerWizzTimer(Sender: TObject);
begin
    if Bouge then
        begin
            if cpt = 10 then
                begin
                    Bouge := False;
                    cpt := 0;
                end
            else
                begin
                    if (cpt mod 2) = 0 then
                        MsgPerso.Left := MsgPerso.Left + 5
                    else
                        MsgPerso.Left := MsgPerso.Left - 5;
                    end;
                    cpt := cpt + 1;
                end;
            Exit;
        end;
end;

procedure TMsgPerso.BitBtn1Click(Sender: TObject);
begin
    if (FontDialog1.Execute) then
        begin
            LabeledEdit1.Font := FontDialog1.Font;
            Mem1.Font := FontDialog1.Font;
        end;
end;

procedure TMsgPerso.BitBtnEnvoyermsgPriveClick(Sender: TObject);
var
    MessagePrive: string;
begin

```

```
if LabeledEdit1.Text <> '' then
begin
  if ClientConnector = TRUE then
  begin
    MessagePrive := 'MsgPrive' + '#DE#' + EditLogin + '#ЗАГОЛОВОК
ПОВІДОМЛЕННЯ#' + TrouverNomLoginQuiRecoitMsg + '#ТІЛО ПОВІДОМЛЕННЯ#' +
LabeledEdit1.Text + '#ЗАВЕРШЕННЯ РОБОТИ#';
    _MITTEVI_POVIDOMLENNYA_RL.ClientSocket.Socket.SendText(#13 +
MessagePrive);
    end;
  end;
  LabeledEdit1.Clear;
end;

end.
```

КБПЗ_2024

AlertMsg.pas - передача програмних повідомлень

```

unit AlertMsg;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons, ImgList;

type
  TAlertMsgF = class(TForm)
    ExecutionTimer: TTimer;
    IconImg: TImage;
    IconsList: TImageList;
    FondImg: TImage;
    TitleLbl: TLabel;
    TextLbl: TLabel;
    procedure ExecutionTimerTimer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    Oghigh: integer; // Оригінальний розмір форми
    WaitBeforeDownCounter: integer;
    WaitBeforeDownTime: integer;
    AlertWindowState : byte; // 0 = змонтувати; 1 = зупинка; 2 = походження; 3 =
інактивзації - див. «Const»
    IsRunning: boolean;
  end;

procedure AlertMsgBox(ATitle, AText: string; AIcon:integer=0;
ABeep:boolean=false; WaitBeforeDown:integer=1000; StepTime:integer=10;
OnTextClick:TNotifyEvent=nil);

var
  AlertMsgF: TAlertMsgF;

const // ВИЗНАЧЕННЯ КОНСТАНТ ДЛЯ ПОЛЕГШЕННЯ ВИКОРИСТАННЯ ПРОЦЕДУР АПЕЛЯЦІЇ
  ICON_NONE=0; ICON_INFO=1; ICON_QUESTION1=2; ICON_QUESTION2=3; ICON_WARNING1=4;
  ICON_WARNING2=5; ICON_QUIT=6; ICON_STOP=7; ICON_GO=8; ICON_SECURE=9;
  ICON_VALIDATE=10; ICON_SEARCH=11; ICON_SENDMAIL=12; ICON_NEWMAIL=13;
  ICON_USER=14;
  // КОНСТАНТИ ДЛЯ КРАЩОГО РОЗУМІННЯ КОДУ
  WS_UP=0; WS_STOP=1; WS_DOWN=2; WS_DISABLED=3;

implementation

{$R *.dfm}

// ПАРАМЕТРИ ІНІЦІАЛІЗАЦІЇ
procedure TAlertMsgF.FormCreate(Sender: TObject);
var Rect: TRect;
begin
  DoubleBuffered := true;
  Oghigh := ClientHeight;
  FormStyle := fsStayOnTop;
  Left:=Screen.Width-ClientWidth;
  SystemParametersInfo(SPI_GETWORKAREA, 0, @Rect, 0);
  Top := Screen.Height - (Screen.Height - Rect.Bottom)-1; //
  ClientHeight := 1; //
  IsRunning := false; //
end;

// ЗВЕРНЕННЯ ДО MESSAGEBOX
procedure AlertMsgBox(ATitle, AText: string; AIcon:integer=0;
ABeep:boolean=false; WaitBeforeDown:integer=1000; StepTime:integer=10;
OnTextClick:TNotifyEvent=nil);
begin

```

with AlertMsgF do begin

```

    IsRunning := true;

    // УСТАНОВКА ВІКНА (ТЕКСТ І ЗНАЧОК)
    Caption := ATitle;
    TitleLbl.Caption := ATitle;
    TextLbl.Caption := AText;
    if AIcon = 0 then begin
        IconImg.Picture := nil;
        TitleLbl.Left := 3; TitleLbl.Width := 173;
    end else begin
        IconImg.Picture := nil;
        IconsList.GetBitmap(AIcon-1, IconImg.Picture.Bitmap);
        TitleLbl.Left := 21; TitleLbl.Width := 155;
    end;

    // ПОРЯДОК НАЛАШТУВАННЯ ОДНОГО КЛИКА НА ПОВІДОМЛЕННЯ
    if Assigned(OnTextClick) then begin
        TextLbl.Cursor := crHandPoint;
        TextLbl.OnClick := OnTextClick;
    end else begin
        TextLbl.Cursor := crDefault;
        TextLbl.OnClick := nil;
    end;

    WaitBeforeDownTime := WaitBeforeDown div StepTime;
    ExecutionTimer.Interval := StepTime;
    AlertWindowState := WS_UP;
    Show;
    if ABeep then Beep;
    ExecutionTimer.Enabled := true;
end;

procedure TAlertMsgF.ExecutionTimerTimer(Sender: TObject);
begin
    if AlertWindowState = WS_DISABLED then begin
        ExecutionTimer.Enabled := false; exit;
    end else if AlertWindowState = WS_UP then begin
        ClientHeight := ClientHeight + 1;
        Top := Top - 1;
        if ClientHeight=OHigh then begin
            WaitBeforeDownCounter := 0;
            AlertWindowState:=WS_STOP;
        end;
    end else if AlertWindowState = WS_STOP then begin
        inc(WaitBeforeDownCounter);
        if WaitBeforeDownCounter = WaitBeforeDownTime then begin
            WaitBeforeDownCounter := 0;
            AlertWindowState:=WS_DOWN;
        end;
    end else if AlertWindowState = WS_DOWN then begin
        ClientHeight := ClientHeight - 1;
        Top := Top + 1;
        if ClientHeight=1 then begin
            AlertWindowState:=WS_DISABLED;
        end;
    end;
end;

```

```
    ExecutionTimer.Enabled := false;  
    Visible := false;  
    IsRunning := false;  
end;  
end;
```

```
    FormStyle := fsStayOnTop;  
    Application.ProcessMessages;  
end;
```

```
end.
```

К6ПЗ_2024

Серверна частина
Unit_MITTEVI_POVIDOMLENNYA_RL.pas - основна частина серверного додатку

```

unit Unit_MITTEVI_POVIDOMLENNYA_RL;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, WinSock, Buttons, ScktComp, ExtCtrls, Grids,
  ValEdit, Menus, Math, ShellAPI;

const
  WM_CallBack = WM_USER;

type
  T_MITTEVI_POVIDOMLENNYA_RL = class(TForm)
    PageControl: TPageControl;
    TabSheetServeur: TTabSheet;
    GroupBox1: TGroupBox;
    LabelIP: TLabel;
    ServerSocket: TServerSocket;
    TimerInformations_MITTEVI_POVIDOMLENNYA_eur: TTimer;
    BitBtnLancerS: TBitBtn;
    TimerNombresClientsActuels: TTimer;

    procedure TabSheetServeurShow(Sender: TObject);
    function TrouverIP(ordinateur : string) : string;
    function NomPcActuel : string;
    function MessageInfo : string;
    procedure FormCreate(Sender: TObject);
    procedure ServerSocketAccept(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocketClientConnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BitBtnLancerSClick(Sender: TObject);
    procedure ServerSocketClientDisconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocketClientRead(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure AnalysePremiereInformation(MessageRecu : string);
    procedure AnalyseDerniereInformation(MessageRecu : string);
    procedure TimerInformations_MITTEVI_POVIDOMLENNYA_eurTimer(Sender: TObject);
    procedure NomClientParti;
    procedure PageControlChanging(Sender: TObject;
      var AllowChange: Boolean);

    procedure LabeledEdit1KeyPress(Sender: TObject; var Key: Char);
    procedure Fermer1Click(Sender: TObject);
    procedure PasLeDroitDeSeConnecter(Login : string; Ordi : string; iden :
integer);
    procedure VientDeSeDeconnecterVolontairement(MessageRecu : string);
    procedure VientDeSeDeconnecterCarVire(MessageRecu : string);
    procedure TropDeMondeSurLeServeur(MessageRecu : string);
    procedure ReponseDeMessageStatut(MessageRecu: string; MessageEnvoi: string);
    procedure ReponseDeMessagePrive(MessageRecu : string; MessageEnvoi :
string);
    procedure AnalyseMessageRecuParServeur(Msg : String);
    procedure ServerSocketClientError(Sender: TObject; Socket: TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);

  private

```

```

public
protected
    IsServer: Boolean;
end;

type
    TStructureListeConnecte = record
        LoginConnecte : string[30];
        NomOrdinateur : string[30];
        Iden : integer;
        img : string;
    end;

var
    _MITTEVI_POVIDOMLENNYA_RL: T_MITTEVI_POVIDOMLENNYA_RL;
    EditPortServeur : string;
    StructureOrdinateur, StructureOrdinateur002 :array[1..52] of
TStructureListeConnecte;

ComboBoxNombresClients, NumeroArriveConnexion, NombresMaximumClients, NombreSecret
: integer;
    ServeurActif, ClientConnecter : boolean;
    Present: TDateTime;
    Hour, Min, Sec, MSec: Word;

    _MITTEVI_POVIDOMLENNYA_HauteurDeDebut, _MITTEVI_POVIDOMLENNYA_LargeurDeDebut, Page
HauteurDebut, PageLargeurDebut : integer;
    TrayIcon : TNotifyIconData;
    blah : HICON;
    mvt : Integer;

function UserName():string;

implementation

{$R *.dfm}

//Функція визначення часу
function QuelHeureEstIl : string;
begin
    Present:= Now;
    DecodeTime(Present, Hour, Min, Sec, MSec);
    result := '['+IntToStr(Hour)+':'+IntToStr(Min)+':'+IntToStr(Sec)+']';
end;
function droite(substr: string; s: string): string;
begin
    if pos(substr,s)=0 then result:='' else
        result:=copy(s, pos(substr, s)+length(substr), length(s)-pos(substr,
s)+length(substr));
    end;
function gauche(substr: string; s: string): string;
begin
    result:=copy(s, 1, pos(substr, s)-1);
end;

//Функція пошуку IP-адреси

function T_MITTEVI_POVIDOMLENNYA_RL.TrouverIP(ordinateur : string) : string;
var
    WSADATA : TWSADATA;
    Name,Address : String;
    Phe : PHostEnt;
begin
    WSASStartup(2, WSADATA);
    SetLength(Name, 255);
    Phe := GetHostByName(PChar(ordinateur));
    with Phe^ do

```

```

    Address := Format ('%d.%d.%d.%d' , [Byte(h_addr^[0]),Byte(h_addr^[1]),
                                         Byte(h_addr^[2]),Byte(h_addr^[3])]);
    WSACleanup;
    TrouverIP := Address;
end;

function T_MITTEVI_POVIDOMLENNYA_RL.NomPcActuel : string;
var
    Buffer : array[0..255] of char;
    BufferSize : DWORD;
begin
    BufferSize := sizeof(Buffer);
    GetComputerName(@buffer, BufferSize);
    result := buffer;
end;

//процедура визначення списку серверів

procedure T_MITTEVI_POVIDOMLENNYA_RL.TabSheetServeurShow(Sender: TObject);
begin
    LabelIP.Caption := TrouverIP(NomPcActuel);
    GroupBox1.Caption := 'Вибрати сервер: ' + NomPcActuel;
    ComboBoxNombresClients := 999;
end;

//Створення форми чату

//створення форми

procedure T_MITTEVI_POVIDOMLENNYA_RL.FormCreate(Sender: TObject);
var
    SysMenu: hMenu;
begin
    SysMenu := GetSystemMenu(Handle, False);
    ModifyMenu(SysMenu, sc_Close, mf_ByCommand, sc_Close, 'Покинути програмне
забезпечення обміну інформацією у мережі під керуванням ОС
Windows!!!'#9'Alt+F4');
    SysMenu := GetSystemMenu(application.handle,false);
    ModifyMenu(SysMenu, sc_Close, mf_ByCommand, sc_Close, 'Покинути програмне
забезпечення обміну інформацією у мережі під керуванням ОС
Windows!!!'#9'Alt+F4');

    Randomize;
    NombreSecret := RandomRange(1000,9999);

    _MITTEVI_POVIDOMLENNYA_RL.Left := screen.Width -
_MITTEVI_POVIDOMLENNYA_RL.Width ;
    _MITTEVI_POVIDOMLENNYA_RL.top := screen.height -
_MITTEVI_POVIDOMLENNYA_RL.height-30 ;

    blah := application.Icon.Handle;
    Trayicon.cbSize := SizeOf(TNotifyIconData);
    Trayicon.Wnd := handle;
    Trayicon.szTip := 'Обмін повідомленнями';
    Trayicon.uID := 1;
    TrayIcon.hIcon := blah;
    TrayIcon.uCallbackMessage := WM_Callback;
    Trayicon.uFlags := NIF_MESSAGE or NIF_ICON or NIF_TIP;
    Shell_NotifyIcon(NIM_ADD,@trayicon);

    _MITTEVI_POVIDOMLENNYA_RL.Visible := true;

    _MITTEVI_POVIDOMLENNYA_RL.Paint;

    EditPortServeur := '2879';

```

```

end;

//Введення імені користувача

function UserName():string;
const
  cnMaxUserNameLen = 254;
var
  UserName2 : string;
  nSize : DWord;
begin
  nSize := cnMaxUserNameLen - 1;
  SetLength(UserName2, cnMaxUserNameLen);

  GetUserName(Pchar(UserName2), nSize);

  SetLength(UserName2, nSize -1);

  result := UserName2;
end;

//Прийняття даних з сокету серверу

procedure T_MITTEVI_POVIDOMLENNYA_RL.ServerSocketAccept(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  IsServer := True;
end;

//З'єднання сокетів сервера та клієнта програмного забезпечення обміну
інформацією у мережі під керуванням ОС Windows

procedure T_MITTEVI_POVIDOMLENNYA_RL.ServerSocketClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  inc(NúmeroArriveConnexion);
  TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Enabled := TRUE;
end;

//Закриття форми

//Закриття форми

procedure T_MITTEVI_POVIDOMLENNYA_RL.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  Fermer1Click(_MITTEVI_POVIDOMLENNYA_RL);
end;

//Запуск та відключення серверу

procedure T_MITTEVI_POVIDOMLENNYA_RL.BitBtnLancerSClick(Sender: TObject);
var
  i : integer;
begin
  if (CompareStr('Запуск серверу',BitBtnLancerS.Caption)=0) then
  begin
    if (EditPortServeur <> '') and ((strtoint(EditPortServeur)) > 0) then
    begin
      try
        NombresMaximumClients := ComboBoxNombresClients;
        ServerSocket.Port := strtoint(EditPortServeur);
        ServerSocket.Active := True;
        BitBtnLancerS.Caption := 'Відключення серверу';
        ServeurActif := TRUE;
        NombresMaximumClients := ComboBoxNombresClients;
      except on ESocketError do
        begin

```

```

MessageDlg('Ви не можете запустити 2 сервери на одному комп'ютері.',
mtInformation, [mbOK], 0);
BitBtnLancerS.Caption := 'Запуск серверу';
TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Enabled := FALSE;
TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Interval := 3000;
fillchar(StructureOrdinateur, sizeof(StructureOrdinateur), 0);
fillchar(StructureOrdinateur002, sizeof(StructureOrdinateur002), 0);
NumeroArriveConnexion := 0;
ServerSocket.Active := FALSE;
ServeurActif := FALSE;
end;
end;
end;
end
else if (CompareStr(BitBtnLancerS.Caption, 'Відключення серверу') = 0) then
begin
for i:=0 to NumeroArriveConnexion-1 do
begin
ServerSocket.Socket.Connections[i].SendText(#13+'Підключено'); //
відправлення даних клієнту
end;
BitBtnLancerS.Caption := 'Запуск серверу';
TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Enabled := FALSE;
TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Interval := 3000;
fillchar(StructureOrdinateur, sizeof(StructureOrdinateur), 0);
fillchar(StructureOrdinateur002, sizeof(StructureOrdinateur002), 0);
NumeroArriveConnexion := 0;
ServerSocket.Active := FALSE;
ServeurActif := FALSE;
end;
end;

// Роз'єднання сокетів сервера та клієнта програмного забезпечення обміну
інформацією у мережі під керуванням ОС Windows

procedure T_MITTEVI_POVIDOMLENNYA_RL.ServerSocketClientDisconnect(Sender:
TObject;
Socket: TCustomWinSocket);
begin
dec(NumeroArriveConnexion);
if NumeroArriveConnexion = 0 then
begin
TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Enabled := FALSE;
TimerInformations_MITTEVI_POVIDOMLENNYA_eur.Interval := 3000;
end;
end;

// Помилки з'єднання сокетів сервера та клієнта програмного забезпечення обміну
інформацією у мережі під керуванням ОС Windows

procedure T_MITTEVI_POVIDOMLENNYA_RL.ServerSocketClientError(Sender: TObject;
Socket: TCustomWinSocket; ErrorEvent: TErrorEvent; var ErrorCode: Integer);
var Rapport: string;
begin
case ErrorEvent of
eeGeneral: Rapport := 'Несподівана помилка' + Socket.RemoteAddress;
eeSend: Rapport := 'Помилка з'єднання сокетів' + Socket.RemoteAddress;
eeReceive: Rapport := 'Помилка читання з'єднання сокетів' +
Socket.RemoteAddress;
eeConnect: Rapport := 'Запит на з'єднання неможливий ' +
Socket.RemoteAddress;
eeDisconnect: Rapport := 'Помилка закриття з'єднання ' +
Socket.RemoteAddress;
eeAccept: Rapport := 'Помилка прийняття запиту з'єднання клієнта ' +
Socket.RemoteAddress;
else
end;
ErrorCode := 0;

```

end;

// Читання з'єднання сокетів сервера та клієнта програмного забезпечення обміну інформацією у мережі під керуванням ОС Windows

```
procedure T_MITTEVI_POVIDOMLENNYA_RL.ServerSocketClientRead(Sender:
TObject;Socket: TCustomWinSocket);
```

```
var
```

```
    TEMPO : string;
```

```
begin
```

```
    Application.ProcessMessages;
```

```
    TEMPO := socket.ReceiveText;
```

```
    while (pos(#13,TEMPO) <> 0) do
```

```
    begin
```

```
        AnalyseMessageRecuParServeur (gauche (#13, TEMPO));
```

```
        TEMPO := droite (#13, TEMPO);
```

```
    end;
```

```
    AnalyseMessageRecuParServeur (TEMPO);
```

```
end;
```

//Аналіз повідомлення отриманого сервером

```
procedure T_MITTEVI_POVIDOMLENNYA_RL.AnalyseMessageRecuParServeur(Msg : String);
```

```
var
```

```
    MessageRecuServeur, MessagePrivePourLesClient: string;
```

```
begin
```

```
    MessageRecuServeur := Msg;
```

```
    if copy(MessageRecuServeur,1,1) = 'µ' then
```

```
    begin
```

```
        if NombresMaximumClients+1 > ServerSocket.Socket.ActiveConnections
```

```
    then
```

```
        begin
```

```
            MessageRecuServeur := Gauche ('µ',MessageRecuServeur);
```

```
        AnalysePremiereInformation (copy (MessageRecuServeur,2,length (MessageRecuServeur)-
1));
```

```
        end
```

```
    else
```

```
    begin
```

```
        MessageRecuServeur := Gauche ('µ',MessageRecuServeur);
```

```
        TropDeMondeSurLeServeur (copy (MessageRecuServeur,2,length (MessageRecuServeur)-
1));
```

```
        end;
```

```
    end;
```

```
    if Comparestr (copy (MessageRecuServeur,1,5), '@DECO') = 0 then
```

```
    begin
```

```
        MessageRecuServeur := Gauche ('K',MessageRecuServeur);
```

```
        MessageRecuServeur := Droite ('@DECO',MessageRecuServeur);
```

```
        AnalyseDerniereInformation (MessageRecuServeur);
```

```
        NomClientParti;
```

```
        VientDeSeDeconnecterVolontairement (MessageRecuServeur);
```

```
    end;
```

```
    if Comparestr (copy (MessageRecuServeur, 1, 8), '#CTATYC#') = 0 then
```

```
    begin
```

```
        MessagePrivePourLesClient := MessageRecuServeur;
```

```
        MessageRecuServeur := Gauche ('#ЗАБЕПШЕННЯ РОБОТИ#', MessageRecuServeur);
```

```
        MessageRecuServeur := Droite ('#CTATYC#', MessageRecuServeur);
```

```
        ReponseDeMessageSTATUT (MessageRecuServeur, MessagePrivePourLesClient);
```

```
    end;
```

```
    if Comparestr (copy (MessageRecuServeur,1,8), 'MsgPrive')=0 then
```

```
    begin
```

```
        MessagePrivePourLesClient := MessageRecuServeur;
```

```
        MessageRecuServeur := Gauche ('#ЗАБЕПШЕННЯ РОБОТИ#',MessageRecuServeur);
```

```
        MessageRecuServeur := Droite ('MsgPrive#DE#',MessageRecuServeur);
```

```
        ReponseDeMessagePrive (MessageRecuServeur,MessagePrivePourLesClient);
```

```
    end;
```

```

if Comparestr(copy(MessageRecuServeur,1,12),'MsgTransfert')=0 then
begin
    MessagePrivePourLesClient := MessageRecuServeur;
    MessageRecuServeur := Gauche('#ЗАБЕРШЕННЯ РОБОТИ#',MessageRecuServeur);
    MessageRecuServeur := Droite('MsgTransfert#DE#',MessageRecuServeur);
    ReponseDeMessagePrive(MessageRecuServeur,MessagePrivePourLesClient);
end;

if Comparestr(copy(MessageRecuServeur,1,9),'MsgBouger')=0 then
begin
    MessagePrivePourLesClient := MessageRecuServeur;
    MessageRecuServeur := Gauche('#ЗАБЕРШЕННЯ РОБОТИ#',MessageRecuServeur);
    MessageRecuServeur := Droite('MsgBouger#DE#',MessageRecuServeur);
    ReponseDeMessagePrive(MessageRecuServeur,MessagePrivePourLesClient);
end;
end;

// Визначення статусу повідомлення

procedure T_MITTEVI_POVIDOMLENNYA_RL.ReponseDeMessageStatut(MessageRecu: string;
MessageEnvoi: string);
var
    LoginEnvoi: string;
    i: integer;
begin
    LoginEnvoi := Droite('#DE#', MessageRecu);
    MessageRecu := Gauche('#DE#', MessageRecu);

    for i := 0 to NumeroArriveConnexion - 1 do
    begin
        ServerSocket.Socket.Connections[i].SendText(#13 + MessageEnvoi);
    end;
end;

//Відповіді на особисте повідомлення

procedure T_MITTEVI_POVIDOMLENNYA_RL.ReponseDeMessagePrive(MessageRecu : string;
MessageEnvoi : string);
var
    LoginEnvoi, LoginRecoi, MessageTexte : string;
    i : integer;
begin
    LoginEnvoi := Gauche('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#',MessageRecu);
    MessageRecu := Droite('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#',MessageRecu);
    LoginRecoi := Gauche('#ТІЛО ПОВІДОМЛЕННЯ#',MessageRecu);
    MessageTexte := Droite('#ТІЛО ПОВІДОМЛЕННЯ#',MessageRecu);

    for i:=0 to NumeroArriveConnexion-1 do
    begin
        ServerSocket.Socket.Connections[i].SendText(#13+MessageEnvoi);
    end;
end;

end;
procedure T_MITTEVI_POVIDOMLENNYA_RL.TropDeMondeSurLeServeur(MessageRecu :
string);
var
    i,iden : integer;
    Login,NomOrdi,MessageInfoDepart : string;
begin
    //EditLogin.Text + 'µ' + NomPcActuel+'«/\»'+inttostr(NombreSecret)
    Application.ProcessMessages;
    Login := Gauche('µ',MessageRecu);
    MessageRecu := Droite('µ',MessageRecu);
    NomOrdi := Gauche('«/\»',MessageRecu);
    iden := strtoint(Droite('«/\»',MessageRecu));

    MessageInfoDepart :=
'TuEsVirЙ°J°'+Login+'µ'+NomOrdi+'«/\»'+inttostr(iden)+'003';

```

```

    for i:=0 to NumeroArriveConnexion-1 do
    begin
        ServerSocket.Socket.Connections[i].SendText(#13+MessageInfoDepart);
    end;
end;

procedure
T_MITTEVI_POVIDOMLENNYA_RL.VientDeSeDeconnecterVolontairement(MessageRecu :
string);
var
    Login,NomOrdi : string;
    m : integer;
begin
    Application.ProcessMessages;
    Login := Gauche('µ',MessageRecu);
    MessageRecu := Droite('µ',MessageRecu);
    NomOrdi := Gauche('«/\»',MessageRecu);

    for m:=0 to NumeroArriveConnexion-1 do
    begin
        ServerSocket.Socket.Connections[m].SendText(#13+'Сервер'+ ' >>  '+Login+'
('+NomOrdi+')'+ ' поз'єднаний...'+'');
    end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.VientDeSeDeconnecterCarVire(MessageRecu :
string);
var
    m : integer;
    Login,NomOrdi : string;
begin
    //EditLogin.Text + 'µ' + NomPcActuel+'«/\»'+inttostr(NombreSecret)
    Application.ProcessMessages;
    Login := Gauche('µ',MessageRecu);
    MessageRecu := Droite('µ',MessageRecu);
    NomOrdi := Gauche('«/\»',MessageRecu);

    for m:=0 to NumeroArriveConnexion-1 do
    begin
        ServerSocket.Socket.Connections[m].SendText(#13+'Сервер'+ ' >>  '+Login+'
('+NomOrdi+')'+ ' з'єднаний...'+'');
    end;
end;

//Первиний аналіз даних

procedure T_MITTEVI_POVIDOMLENNYA_RL.AnalysePremiereInformation(MessageRecu :
string);
var
    i,j,k,NombreIdentifiant : integer;
    login,NomOrdi,MessageInfoArrive : string;
    Remplir,LoginExisteDejaDesole : boolean;
begin
    //EditLogin.Text + 'µ' + NomPcActuel + '«/\»' + NombreSecret
    Application.ProcessMessages;

    Login := Gauche('µ',MessageRecu);
    NomOrdi := Droite('µ',MessageRecu);
    NomOrdi := Gauche('«/\»',NomOrdi);
    NombreIdentifiant := StrToInt(Droite('«/\»',MessageRecu));

    Remplir := TRUE;
    LoginExisteDejaDesole := FALSE;

    for k := 0 to length(StructureOrdinateur)-1 do
    begin
        if CompareStr(Login,StructureOrdinateur[k].LoginConnecte)=0
then

```

```

begin
    LoginExisteDejaDesole := TRUE;
end;
end;

for j := 1 to NumeroArriveConnexion + 3 do
begin
    if (StructureOrdinateur[j].LoginConnecte = '') and
        (StructureOrdinateur[j].NomOrdinateur = '') and
        (Remplir = TRUE) and (LoginExisteDejaDesole = FALSE) then
    begin
        StructureOrdinateur[j].LoginConnecte := login;
        StructureOrdinateur[j].NomOrdinateur := NomOrdi;
        StructureOrdinateur[j].Iden := NombreIdentifiant;
        StructureOrdinateur[j].img := 'Можу розмовляти';
        Remplir := FALSE;
    end;
end;

if not Remplir then
begin
    MessageInfoArrive := 'Сервер'+>> '+Login+' ('+NomOrdi+')'+
з'єднаний...'+''';
    for i:=0 to NumeroArriveConnexion-1 do
    begin
ServerSocket.Socket.Connections[i].SendText(#13+MessageInfoArrive);
        end;
    end;

    if LoginExisteDejaDesole then
        PasLeDroitDeSeConnecter(login,NomOrdi,NombreIdentifiant);
end;

//Помилка з'єднання

Procedure T_MITTEVI_POVIDOMLENNYA_RL.PasLeDroitDeSeConnecter(Login : string;
Ordi : string; iden : integer);
var
    MessageInfoDepart : string;
    i : integer;
begin
    MessageInfoDepart :=
'TuEsViriy°J°'+Login+'µ'+Ordi+'«/\»'+inttostr(iden)+'001';

    for i:=0 to NumeroArriveConnexion-1 do
    begin
        ServerSocket.Socket.Connections[i].SendText(#13+MessageInfoDepart);
    end;
end;

// Останній аналіз інформації

procedure T_MITTEVI_POVIDOMLENNYA_RL.AnalyseDerniereInformation(MessageRecu :
string);
var
    j,k,l,UnCranDeMoins,Identifiant : integer;
    login,NomOrdi : string;
begin
    Application.ProcessMessages;
    Login := Gauche('µ',MessageRecu);
    MessageRecu := Droite('µ',MessageRecu);
    NomOrdi := Gauche('«/\»',MessageRecu);
    Identifiant := strtoint(Droite('«/\»',MessageRecu));

    for j:=1 to 52 do
    begin
        if (CompareStr(StructureOrdinateur[j].LoginConnecte, Login)=0)

```

```

and (CompareStr(StructureOrdinateur[j].NomOrdinateur,
NomOrdi)=0)
and (Identifiant = StructureOrdinateur[j].Iden) then
begin
StructureOrdinateur[j].LoginConnecte := '';
StructureOrdinateur[j].NomOrdinateur := '';
StructureOrdinateur[j].Iden := 0;
StructureOrdinateur[j].Img := 'Можу розмовляти';

fillchar(StructureOrdinateur002,sizeof(StructureOrdinateur002),0); // Розмір
структури дорівнює нулю
UnCranDeMoins :=0;

for k :=1 to 52 do
begin
if (StructureOrdinateur[k].LoginConnecte = '')
and (StructureOrdinateur[k].NomOrdinateur = '')
and (StructureOrdinateur[k].Iden = 0) then
begin
inc(UnCranDeMoins);
end
else
begin
StructureOrdinateur002[k-UnCranDeMoins].LoginConnecte
:= StructureOrdinateur[k].LoginConnecte;
StructureOrdinateur002[k-UnCranDeMoins].NomOrdinateur
:= StructureOrdinateur[k].NomOrdinateur;
StructureOrdinateur002[k-UnCranDeMoins].Iden :=
StructureOrdinateur[k].Iden;
StructureOrdinateur002[k-UnCranDeMoins].img :=
StructureOrdinateur[k].img;
end;
end;

fillchar(StructureOrdinateur,sizeof(StructureOrdinateur),0);
for l :=1 to 52 do
begin
if (StructureOrdinateur002[l].LoginConnecte <> '')
and (StructureOrdinateur002[l].NomOrdinateur <>'')
and (StructureOrdinateur002[l].Iden <> 0) then
begin
StructureOrdinateur[l].LoginConnecte :=
StructureOrdinateur002[l].LoginConnecte;
StructureOrdinateur[l].NomOrdinateur :=
StructureOrdinateur002[l].NomOrdinateur;
StructureOrdinateur[l].Iden :=
StructureOrdinateur002[l].Iden;
StructureOrdinateur[l].img :=
StructureOrdinateur002[l].img;
end;
end;
end;
end;

//Запис часу введення повідомлення

procedure
T_MITTEVI_POVIDOMLENNYA_RL.TimerInformations_MITTEVI_POVIDOMLENNYA_eurTimer(Sender: TObject);
var
i : integer;
begin
{відправляє повідомлення з з'єднань усіх ПК клієнта до сервера}
for i:=0 to NumeroArriveConnexion-1 do
begin
ServerSocket.Socket.Connections[i].SendText(#13+MessageInfo); //
відправлення даних клієнту

```

```

    end;
end;

//Інформація про повідомлення

function T_MITTEVI_POVIDOMLENNYA_RL.MessageInfo : string;
var
    i : integer;

MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur,MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl : string;
begin
    Application.ProcessMessages;
    MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl := '';
    MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur := '';
    MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur := '@ш*@';
    for i:=1 to NumeroArriveConnexion + 3 do
    begin
        if StructureOrdinateur[i].LoginConnecte <> '' then
        begin
            MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl :=
            MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl + 'µ'
            + StructureOrdinateur[i].LoginConnecte + '#IMG#' +
            StructureOrdinateur[i].img +
            '#ORDI#'+StructureOrdinateur[i].NomOrdinateur;
        end;
    end;

    if MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl <> '' then
    begin
        MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur :=
        MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur +
        MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl + 'µш@*!@ш';
        result:= MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur;
    end
    else
    begin
        result:= '??';
    end;
end;

end;

//Виведення імені користувача, з яким спілкуємся

procedure T_MITTEVI_POVIDOMLENNYA_RL.NomClientParti;
var
    i : integer;
    MessageInfoDepart : string;
begin
    MessageInfoDepart := ` `;
    for i:=0 to NumeroArriveConnexion-1 do
    begin
        ServerSocket.Socket.Connections[i].SendText(#13+MessageInfoDepart); //
        відправлення даних клієнту
    end;
end;

end;

{Зміна вкладки}
procedure T_MITTEVI_POVIDOMLENNYA_RL.PageControlChanging(Sender: TObject;
    var AllowChange: Boolean);
begin
    if ((Sender as TPageControl).ActivePage = TabSheetServeur) and (ServeurActif =
    TRUE) then
        AllowChange := FALSE
    end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.LabeledEdit1KeyPress(Sender: TObject; var
    Key: Char);
var

```

```

    i : integer;
    MessageAEnvoyer : string;
begin
    if (Key = #13) and (ServeurActif = TRUE) then
    begin
        key := #0;
        for i:=0 to NumeroArriveConnexion-1 do
        begin
            ServerSocket.Socket.Connections[i].SendText(#13+'Сервер'+ ' >>
'+надсилає повідомлення'); // відправлення даних клієнту
            end;
        end;
        if (Key = #13) then
            key := #0;
    end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.Fermer1Click(Sender: TObject);
var
    i : integer;
begin
    ClipCursor(nil); // Мишка знаходиться у полі програмного забезпечення.
    if (not ClientConnector) and (not ServeurActif) then
        Application.Terminate;

    if ServeurActif = TRUE then
    begin
        TimerNombresClientsActuels.Enabled := FALSE;
        TimerNombresClientsActuels.Interval := 100;
        for i:=0 to NumeroArriveConnexion-1 do
        begin
            ServerSocket.Socket.Connections[i].SendText(#13+'Підключено'); //
відправлення даних клієнту
            end;
            ServerSocket.Active := FALSE;
            Application.Terminate;
        end;
    end;

end;

{Управління звуком}
initialization
    NumeroArriveConnexion :=0;

end.

```

Unit_MITTEVI_POVIDOMLENNYA_RL.pas - параметри інтерфейсу серверної частини

```

unit Interface_korystuvacha;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls;

type
  TChoixCouleur = class(TForm)
    GroupBox1: TGroupBox;
    BitBtn1: TBitBtn;
    RadioGroupSons: TRadioGroup;
    RadioGroupVisible: TRadioGroup;
    GroupBoxCouleur: TGroupBox;
    ColorDialog1: TColorDialog;
    Panell: TPanel;
    StaticText1: TStaticText;
    Button1: TButton;
    StaticText2: TStaticText;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure PanellClick(Sender: TObject);
  private
    { Private declarations }
    procedure SystemeCommande(var Msg : TMessage);
    message WM_SysCommand; // перехоплення повідомлень SysCommand
  public
    { Public declarations }
  end;

var
  ChoixCouleur: TChoixCouleur;

implementation

uses Unit_MITTEVI_POVIDOMLENNYA_RL;

{$R *.dfm}

procedure TChoixCouleur.FormCreate(Sender: TObject);
var
  Style : LongInt; // видалити рядок заголовка
begin
  {Наступні 4 рядка можуть зробити невидимим заголовок}
  Style := GetWindowLong(Handle,GWL_STYLE); // Встановлення поточного стилю
  Style := Style and not WS_CAPTION; // Видаляє поточний стиль для
відображення заголовка
  SetWindowLong(Handle,GWL_STYLE,Style); // Робить зміни
  SetWindowPos(Handle,0,0,0,0,SWP_FRAMECHANGED or SWP_NOMOVE or SWP_NOSIZE or
SWP_NOZORDER); // Останій рядок
end;

procedure TChoixCouleur.BitBtn1Click(Sender: TObject);
var
  n: Integer; // Служить для активації і CtrlAltShift та AltTab
begin
  SystemParametersInfo(SPI_SCREENSAVER_RUNNING, 0, @n, 0); // Дозволяє закрити
CtrlAltShift та AltTab
  ClipCursor(nil); // Мишка знаходиться у полі програмного забезпечення.

```

```

    Couleur := ChoixCouleur.Panell.Color;
    _МИТТЕВИ_POVIDOMLENNYA_RL.Colorier;
    ChoixCouleur.Close;
end;

procedure TChoixCouleur.SystemeCommande(var Msg : TMessage);
begin
    //успадкований;
    // Інструкція так, що повідомлення обробляються зазвичай
    if Msg.wParam = sc_Close then ; // Закривається по Alt-F4
end;

procedure TChoixCouleur.FormShow(Sender: TObject);
var
    n: Integer; // Служить для активації і CtrlAltShift та AltTab
    Rect: TRect; // Обмеження зони для мишки
begin
    application.ProcessMessages;
    SystemParametersInfo(SPI_SCREENSAVERERRUNNING, Integer(TRUE), @n, 0); //
    Служить для деактивації і CtrlAltShift та AltTab

    { Перетворює координати листа на екран}
    Rect.TopLeft:= ClientToScreen(ClientRect.TopLeft);
    Rect.BottomRight:= ClientToScreen(ClientRect.BottomRight);
    ClipCursor(@Rect); // Обмеження переміщення миші у клієнтської області
    плагіна.

    ChoixCouleur.Panell.Color := Couleur;
    ChoixCouleur.GroupBox1.Color := Couleur;
    ChoixCouleur.RadioGroupSons.Color := Couleur;
    ChoixCouleur.RadioGroupVisible.Color := Couleur;
    ChoixCouleur.GroupBoxCouleur.Color := Couleur;

end;

procedure TChoixCouleur.Button1Click(Sender: TObject);
begin
    Panell.Color := clBtnFace;
end;

procedure TChoixCouleur.PanellClick(Sender: TObject);
var
    n: Integer; // Служить для активації і CtrlAltShift та AltTab
    Rect: TRect; // Обмеження зони для мишки
begin
    SystemParametersInfo(SPI_SCREENSAVERERRUNNING, 0, @n, 0); // Дозволяє закрити
    CtrlAltShift та AltTab
    ClipCursor(nil); // Мишка знаходиться у полі програмного забезпечення.

    If (ColorDialog1.Execute=True) Then
    Begin
        Panell.Color:=ColorDialog1.Color;
        Repaint;
    End;

    SystemParametersInfo(SPI_SCREENSAVERERRUNNING, Integer(TRUE), @n, 0); //
    Служить для деактивації і CtrlAltShift та AltTab
    { Перетворює координати листа на екран}
    Rect.TopLeft:= ClientToScreen(ClientRect.TopLeft);
    Rect.BottomRight:= ClientToScreen(ClientRect.BottomRight);
    ClipCursor(@Rect); // Обмеження переміщення миші у клієнтської області
    плагіна.
end;
end.

```

Клієнтська частина

Unit_MITTEVI_POVIDOMLENNYA_RL.pas - основна частина клієнтського додатку

```

unit Unit_MITTEVI_POVIDOMLENNYA_RL;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, WinSock, Buttons, ScktComp, ExtCtrls, Grids,
  ValEdit, Menus, Math, ShellAPI, cxControls, cxSplitter, cxStyles,
  cxCustomData, cxGraphics, cxFilter, cxData, cxDataStorage, cxEdit, cxImage,
  cxGridCustomTableView, cxGridTableView, cxGridCustomView, cxClasses,
  cxGridLevel, cxGrid, cxBlobEdit, ImgList, cxTextEdit, cxGridBandedTableView,
  DB, DBClient, dxGDIPlusClasses, cxDBData, cxGridDBTableView, cxImageComboBox,
  ActnList, XPStyleActnCtrls, ActnMan, cxLookAndFeels, cxLookAndFeelPainters,
  Registry,
  WinSkinData, WinSkinStore, dxBar;

const
  WM_Callback = WM_USER;
  WM_MYMESSAGE = WM_USER + 100;

type
  T_MITTEVI_POVIDOMLENNYA_RL = class(TForm)
    ClientSocket: TClientSocket;
    Memo_MITTEVI_POVIDOMLENNYA_: TMemo;
    cxSplitter1: TcxSplitter;
    ImageListold: TImageList;
    ClientDataSet1: TClientDataSet;
    ClientDataSet1N_ID: TIntegerField;
    ClientDataSet1IMAGE: TGraphicField;
    ClientDataSet1NOMPrenom: TStringField;
    DS_SOURCE: TDataSource;
    ActionManager1: TActionManager;
    ActionAjouterGroupe: TAction;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    IMG_LIGNE: TImage;
    IMG_OCCUPE: TImage;
    IMG_PAUSE: TImage;
    ClientDataSet3: TClientDataSet;
    DS_GROUPE: TDataSource;
    ClientDataSet3N_ID: TIntegerField;
    ClientDataSet3GROUPE: TStringField;
    ClientDataSet1N_GROUPE: TIntegerField;
    Panel2: TPanel;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    ComboBox1: TComboBox;
    IMG_1MN: TImage;
    IMG_TEL: TImage;
    IMG_ABS: TImage;
    ActionEMAIL: TAction;
    traymenu: TPopupMenu;
    Image1: TImage;
    SpeedButton4: TSpeedButton;
    ActionNavigateur: TAction;
    NavigateurWeb1: TMenuItem;
    BoitederceptionMail1: TMenuItem;
    N1: TMenuItem;
    ActionQuitter: TAction;
    Quitter1: TMenuItem;
    Statut1: TMenuItem;
    EnLignel: TMenuItem;
    Occup1: TMenuItem;
    Autlphonel: TMenuItem;
    EnPausel: TMenuItem;
  end;

```

```

Abs1mn1: TMenuItem;
Absent1: TMenuItem;
ClientDataSet1ORDI: TStringField;
SkinStore1: TSkinStore;
sdl: TSkinData;
Action1: TAction;
ActionFermer: TAction;
Action4: TAction;
Action5: TAction;
Action6: TAction;
Action7: TAction;
Action8: TAction;
Action9: TAction;
Action10: TAction;
Action11: TAction;
Action12: TAction;
Action13: TAction;
dxBarManager1: TdxBarManager;
dxBarManager1Bar1: TdxBar;
dxBarSubItem1: TdxBarSubItem;
dxBarSubItem2: TdxBarSubItem;
dxBarSubItem3: TdxBarSubItem;
dxBarSubItem4: TdxBarSubItem;
dxBarSubItem5: TdxBarSubItem;
dxBarButton1: TdxBarButton;
dxBarButton2: TdxBarButton;
dxBarButton3: TdxBarButton;
dxBarButton4: TdxBarButton;
dxBarSubItem6: TdxBarSubItem;
dxBarSubItem7: TdxBarSubItem;
dxBarButton5: TdxBarButton;
dxBarButton6: TdxBarButton;
dxBarButton7: TdxBarButton;
dxBarButton8: TdxBarButton;
dxBarButton9: TdxBarButton;
dxBarButton10: TdxBarButton;
dxBarButton11: TdxBarButton;
dxBarButton12: TdxBarButton;
dxBarButton13: TdxBarButton;
dxBarButton14: TdxBarButton;
TimerReconnexion: TTimer;
cxGrid3DBTableView1: TcxGridDBTableView;
cxGrid3Level1: TcxGridLevel;
cxGrid3: TcxGrid;
cxGrid3Level2: TcxGridLevel;
cxGrid3DBTableView2: TcxGridDBTableView;
cxGrid3DBTableView2N_ID: TcxGridDBCColumn;
cxGrid3DBTableView2GROUPE: TcxGridDBCColumn;
cxGrid3DBTableView1N_ID: TcxGridDBCColumn;
cxGrid3DBTableView1NOMPrenom: TcxGridDBCColumn;
cxGrid3DBTableView1N_GROUPE: TcxGridDBCColumn;
cxGrid3DBTableView1ORDI: TcxGridDBCColumn;
cxGrid3DBTableView1IMAGE: TcxGridDBCColumn;
ImageList1: TImageList;
procedure WM_CALLBACKPRO(var msg: TMessage); message wm_callBack;
procedure mvtFenetre(i: Integer);
function NomPcActuel: string;
function MessageInfo: string;
procedure FormCreate(Sender: TObject);
procedure TimerNombresClientsActuelsTimer(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ClientSocketConnect(Sender: TObject;
    Socket: TCustomWinSocket);
procedure ClientSocketDisconnect(Sender: TObject;
    Socket: TCustomWinSocket);
procedure ClientSocketError(Sender: TObject; Socket: TCustomWinSocket;
    ErrorEvent: TErrorEvent; var ErrorCode: Integer);
procedure AnalysePremiereInformation(MessageRecu: string);
procedure AnalyseDerniereInformation(MessageRecu: string);

```

```

procedure ClientSocketRead(Sender: TObject; Socket: TCustomWinSocket);
procedure LabeledEditMessageEcritKeyPress(Sender: TObject;
  var Key: Char);
procedure Fermer1Click(Sender: TObject);
procedure RendreVisiblePremierPlan;
procedure PourquoiDeconnecte(raison: string; iden: integer);

procedure DireQueOnSeDeconnecte;
procedure BitBtnDeconnexionClick(Sender: TObject);
procedure FormResize(Sender: TObject);

procedure Minimize(Sender: TObject);

procedure AnalyseMessageRecuParClient(Msg: string);
procedure ActionAjouterGroupeExecute(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure ActionEMAILExecute(Sender: TObject);
procedure ActionNavigateurExecute(Sender: TObject);
procedure ActionQuitterExecute(Sender: TObject);
procedure EnLigne1Click(Sender: TObject);
procedure Action13Execute(Sender: TObject);
procedure Action14Execute(Sender: TObject);
procedure ActionFermerExecute(Sender: TObject);
procedure Action10Execute(Sender: TObject);
procedure TimerReconnexionTimer(Sender: TObject);
procedure cxGrid3DBTableView1CellDbClick(Sender: TcxCustomGridTableView;
  ACellViewInfo: TcxGridTableDataCellViewInfo; AButton: TMouseButton;
  AShift: TShiftState; var AHandled: Boolean);

private

public

end;

type
  TStructureListeConnecte = record
    LoginConnecte: string[30];
    NomOrdinateur: string[30];
    Iden: integer;
    img: string;
  end;

var
  Affichage: Boolean = False;
  root, path: string;
  _MITTEVI_POVIDOMLENNYA_RL: T_MITTEVI_POVIDOMLENNYA_RL;
  EditIPConnexion, EditPortClient, EditLogin: string;
  StructureOrdinateur, StructureOrdinateur002: array[1..52] of
  TStructureListeConnecte;
  mvt, GroupeNB, NumeroArriveConnexion, NombresMaximumClients, NombreSecret:
  integer;
  ServeurActif, ClientConnecter: boolean;
  Present: TDateTime;
  Hour, Min, Sec, MSec: Word;
  nbpersonne, _MITTEVI_POVIDOMLENNYA_HauteurDeDebut,
  _MITTEVI_POVIDOMLENNYA_LargeurDeDebut, PageHauteurDebut, PageLargeurDebut:
  integer;
  TrayIcon: TNotifyIconData;
  blah: HICON;

function UserName(): string;

implementation

uses Interface_korystuvacha, Unit_MsgPerso, AlertMsg, AudioVideo;

```

```

{$R *.dfm}

function QuelHeureEstIl: string;
begin
  Present := Now;
  DecodeTime(Present, Hour, Min, Sec, MSec);
  result := '[' + IntToStr(Hour) + ':' + IntToStr(Min) + ':' + IntToStr(Sec) +
  ']';
end;

function droite(substr: string; s: string): string;
begin
  if pos(substr, s) = 0 then result := '' else
    result := copy(s, pos(substr, s) + length(substr), length(s) - pos(substr,
s) + length(substr));
end;

function gauche(substr: string; s: string): string;
begin
  result := copy(s, 1, pos(substr, s) - 1);
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.RendreVisiblePremierPlan;
begin
  if (ChoixCouleur.RadioGroupVisible.ItemIndex = 0) then
  begin
    Application.Restore;
    Application.BringToFront;
  end;
end;

function T_MITTEVI_POVIDOMLENNYA_RL.NomPcActuel: string;
var
  Buffer: array[0..255] of char;
  BufferSize: DWORD;
begin
  BufferSize := sizeof(Buffer);
  GetComputerName(@buffer, BufferSize);
  result := buffer;
end;

//Зменшення у трей форми

  procedure T_MITTEVI_POVIDOMLENNYA_RL.Minimize;
begin
  mvtFenetre(1);
  _MITTEVI_POVIDOMLENNYA_RL.Visible := False;
end;

//створення форми

  procedure T_MITTEVI_POVIDOMLENNYA_RL.FormCreate(Sender: TObject);
var
  Registre: TRegistry;
  SysMenu: hMenu;
  IPServeur: string;
  I: Integer;
begin
  // Application.OnMinimize := Minimize;
  root := ExtractFilePath(ParamStr(0));
  path := root + 'vsskin\';

  Registre := TRegistry.Create;
  Registre.RootKey := HKEY_LOCAL_MACHINE;

  Registre.OpenKey('\Software\IntraMSN\Couleur\', True);
  if Registre.ValueExists('Couleur') then

```

```

sdl.SkinFile := Registre.ReadString('Couleur');

Registre.OpenKey('\Software\IntraMSN\Image\', True);
if (Registre.ValueExists('Image')) then
  ClientDataSet1IMAGE.DisplayWidth := Round(Registre.ReadInteger('Image') /
6);

Registre.CloseKey;
Registre.Free;

ClientDataSet1.CreateDataSet;
GroupeNB := 1;
EditIPConnexion := '10.1.1.27';
EditPortClient := '2879';
ComboBox1.Text := 'Можу розмовляти';
SysMenu := GetSystemMenu(Handle, False);
ModifyMenu(SysMenu, sc_Close, mf_ByCommand, sc_Close, '&Вихід з програми
!!!'#9'Alt+F4');
SysMenu := GetSystemMenu(application.handle, false);
ModifyMenu(SysMenu, sc_Close, mf_ByCommand, sc_Close, '&Вихід з програми
!!!'#9'Alt+F4');
_MITTEVI_POVIDOMLENNYA_RL.Height := Round(500 * (Screen.Width / 1024) / 1.3);
_MITTEVI_POVIDOMLENNYA_RL.Width := Round(300 * (Screen.height / 768) / 1.3);
_MITTEVI_POVIDOMLENNYA_HauteurDeDebut := _MITTEVI_POVIDOMLENNYA_RL.Height;
_MITTEVI_POVIDOMLENNYA_LargeurDeDebut := _MITTEVI_POVIDOMLENNYA_RL.Width;
ChoixCouleur := TChoixCouleur.Create(_MITTEVI_POVIDOMLENNYA_RL);
CAudioVideo := TCAudioVideo.Create(_MITTEVI_POVIDOMLENNYA_RL);
Randomize;
NombreSecret := RandomRange(1000, 9999);
_MITTEVI_POVIDOMLENNYA_RL.Left := screen.Width -
_MITTEVI_POVIDOMLENNYA_RL.Width;
_MITTEVI_POVIDOMLENNYA_RL.top := screen.height -
_MITTEVI_POVIDOMLENNYA_RL.height - 30;
blah := application.Icon.Handle;
Trayicon.cbSize := SizeOf(TNotifyIconData);
Trayicon.Wnd := handle;
Trayicon.szTip := 'eMessenger';
Trayicon.uID := 1;
TrayIcon.hIcon := blah;
TrayIcon.uCallbackMessage := WM_CALLBACK;
Trayicon.uFlags := NIF_MESSAGE or NIF_ICON or NIF_TIP;
Shell_NotifyIcon(NIM_ADD, @trayicon);

mvtFenetre(1);
_MITTEVI_POVIDOMLENNYA_RL.Visible := true;
mvtFenetre(-1);
_MITTEVI_POVIDOMLENNYA_RL.Paint;

EditLogin := UserName();

if (EditIPConnexion <> '') and (EditPortClient <> '') and
((strtoint(EditPortClient)) > 0) then
begin
  IPServeur := EditIPConnexion;
  ClientSocket.Host := IPServeur;
  ClientSocket.Port := strtoint(EditPortClient);
  ClientSocket.Active := TRUE;
end;

ClientDataSet3.Insert;
ClientDataSet3N_ID.Value := 1;
ClientDataSet3GROUPE.Value := 'Гінйрал';
ClientDataSet3.Post;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.WM_CALLBACKPRO(var msg: TMessage);
begin

```

```

case msg.LParam of WM_LBUTTONDOWN:
begin
  if _MITTEVI_POVIDOMLENNYA_RL.Visible = true then
  begin
    mvtFenetre(1);
    _MITTEVI_POVIDOMLENNYA_RL.Visible := False;
  end
  else
  begin
    mvt := 1;
    _MITTEVI_POVIDOMLENNYA_RL.Visible := True;
    mvtFenetre(-1);
  end;
  end;
  WM_RBUTTONDOWN: traymenu.Popup(mouse.CursorPos.X, mouse.CursorPos.y);
end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.mvtFenetre(i: Integer);
var
  k, fin, offset: integer;
begin
  offset := screen.Height - GetSystemMetrics(SM_CYFULLSCREEN);
  mvt := 1;
  _MITTEVI_POVIDOMLENNYA_RL.Left := screen.Width -
_MITTEVI_POVIDOMLENNYA_RL.Width;
  if (i < 0) then
    fin := screen.Height - _MITTEVI_POVIDOMLENNYA_RL.Height - offset
  else
    fin := screen.Height;
  while (_MITTEVI_POVIDOMLENNYA_RL.Top <> fin) do
  begin
    _MITTEVI_POVIDOMLENNYA_RL.Top := _MITTEVI_POVIDOMLENNYA_RL.Top + i;
    for k := 1 to 1000000 do
      mvt := 1;
    end;
    if (i < 0) then
      _MITTEVI_POVIDOMLENNYA_RL.Top := screen.Height -
_MITTEVI_POVIDOMLENNYA_RL.Height - offset
    else
      _MITTEVI_POVIDOMLENNYA_RL.Top := screen.Height;
    mvt := 0;
  end;
end;

function UserName(): string;
const
  cnMaxUserNameLen = 254;
var
  UserName2: string;
  nSize: DWord;
begin
  nSize := cnMaxUserNameLen - 1;
  SetLength(UserName2, cnMaxUserNameLen);
  GetUserName(Pchar(UserName2), nSize);
  SetLength(UserName2, nSize - 1);
  result := UserName2;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.TimerNombresClientsActuelsTimer(Sender:
TObject);
begin
  application.ProcessMessages;
end;

//Час перез'єднання

procedure T_MITTEVI_POVIDOMLENNYA_RL.TimerReconnexionTimer(Sender: TObject);
var
  IPServeur: string;

```

```

begin
  if not ClientConnector then
    begin
      if (EditIPConnexion <> '') and (EditPortClient <> '') and
        ((strtoint(EditPortClient)) > 0) then
        begin
          IPServeur := EditIPConnexion;
          ClientSocket.Host := IPServeur;
          ClientSocket.Port := strtoint(EditPortClient);
          ClientSocket.Active := TRUE;
          ClientConnector := True;
        end;
      end
    end;
end;

//Закриття форми

procedure T_MITTEVI_POVIDOMLENNYA_RL.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  ActionFermerExecute(Sender);
end;

//З'єднання з сокетом клієнта

procedure T_MITTEVI_POVIDOMLENNYA_RL.ClientSocketConnect(Sender: TObject;
Socket: TCustomWinSocket);
var
  MessageInitial: string;
begin
  Application.ProcessMessages;
  ClientConnector := TRUE;
  MessageInitial := 'µ' + EditLogin + 'µ' + NomPcActuel + '«/\»' +
  intostr(NombreSecret) + 'ч';
  ClientSocket.Socket.SendText(#13 + MessageInitial);
  Application.ProcessMessages;
end;

//Роз'єднання з сокетом клієнта

procedure T_MITTEVI_POVIDOMLENNYA_RL.ClientSocketDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
begin
  Application.ProcessMessages;
end;

//Помилка у сокеті клієнта

procedure T_MITTEVI_POVIDOMLENNYA_RL.ClientSocketError(Sender: TObject;
Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
var ErrorCode: Integer);
begin
  Application.ProcessMessages;
  ErrorCode := 0;
  Application.ProcessMessages;
  ClientSocket.Active := FALSE;
  ClientSocket.Close;
  ClientConnector := FALSE;
end;

//Первинний аналіз даних

procedure T_MITTEVI_POVIDOMLENNYA_RL.AnalysePremiereInformation(MessageRecu:
string);
var
  j, k, NombreIdentifiant: integer;
  login, NomOrdi: string;
  Remplir, LoginExisteDejaDesole: boolean;
begin

```

```

Application.ProcessMessages;

Login := Gauche('µ', MessageRecu);
NomOrdi := Droite('µ', MessageRecu);
NomOrdi := Gauche('«/\»', NomOrdi);
NombreIdentifiant := StrToInt(Droite('«/\»', MessageRecu));

Remplir := TRUE;
LoginExisteDejaDesole := FALSE;

for k := 0 to length(StructureOrdinateur) - 1 do
begin
  if CompareStr(Login, StructureOrdinateur[k].LoginConnecte) = 0 then
  begin
    LoginExisteDejaDesole := TRUE;
  end;
end;

for j := 1 to NumeroArriveConnexion + 3 do
begin
  if (StructureOrdinateur[j].LoginConnecte = '') and
    (StructureOrdinateur[j].NomOrdinateur = '') and
    (Remplir = TRUE) and (LoginExisteDejaDesole = FALSE) then
  begin
    StructureOrdinateur[j].LoginConnecte := login;
    StructureOrdinateur[j].NomOrdinateur := NomOrdi;
    StructureOrdinateur[j].Iden := NombreIdentifiant;
    StructureOrdinateur[j].Img := ComboBox1.Text;
    Remplir := FALSE;
  end;
end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.Action10Execute(Sender: TObject);
begin
  CAudioVideo.Show;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.Action13Execute(Sender: TObject);
begin
  ChoixCouleur.Show;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.Action14Execute(Sender: TObject);
var
  IPServeur: string;
begin
  if (EditIPConnexion <> '') and (EditPortClient <> '') and
    ((strtoint(EditPortClient)) > 0) then
  begin
    IPServeur := EditIPConnexion;
    ClientSocket.Host := IPServeur;
    ClientSocket.Port := strtoint(EditPortClient);
    ClientSocket.Active := TRUE;
  end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.ActionFermerExecute(Sender: TObject);
begin
  Shell_NotifyIcon(Nim_DELETE, @trayicon);
  Fermer1Click(_MITTEVI_POVIDOMLENNYA_RL);
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.ActionAjouterGroupeExecute(Sender:
TObject);
begin
  ClientDataSet3.Insert;
  GroupeNB := GroupeNB + 1;
  ClientDataSet3N_ID.Value := GroupeNB;

```

```

ClientDataSet3GROUPE.Value := 'Groupe Temp';
ClientDataSet3.Post;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.ActionEMAILExecute(Sender: TObject);
begin
  if FileExists('C:\Program Files\Mozilla Thunderbird\thunderbird.exe') then
    ShellExecute(handle, 'open', 'C:\Program Files\Mozilla
Thunderbird\thunderbird.exe', '', '', 0);
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.ActionNavigateurExecute(Sender: TObject);
begin
  if FileExists('C:\Program Files\Mozilla Firefox\firefox.exe') then
    ShellExecute(handle, 'open', 'C:\Program Files\Mozilla Firefox\firefox.exe',
'', '', 0)
  else
    ShellExecute(handle, 'open', 'C:\Program Files\Internet
Explorer\iexplorer.exe', '', '', 0);
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.ActionQuitterExecute(Sender: TObject);
begin
  Shell_NotifyIcon(Nim_DELETE, @trayicon);
  Fermer1Click(_MITTEVI_POVIDOMLENNYA_RL);
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.AnalyseDerniereInformation(MessageRecu:
string);
var
  j, k, l, UnCranDeMoins, Identifiant: integer;
  login, NomOrdi: string;
begin
  Application.ProcessMessages;
  Login := Gauche('µ', MessageRecu);
  MessageRecu := Droite('µ', MessageRecu);
  NomOrdi := Gauche('«/\»', MessageRecu);
  Identifiant := strtoint(Droite('«/\»', MessageRecu));

  for j := 1 to 52 do
  begin
    if (CompareStr(StructureOrdinateur[j].LoginConnecte, Login) = 0)
      and (CompareStr(StructureOrdinateur[j].NomOrdinateur, NomOrdi) = 0)
      and (Identifiant = StructureOrdinateur[j].Iden) then
      begin
        StructureOrdinateur[j].LoginConnecte := '';
        StructureOrdinateur[j].NomOrdinateur := '';
        StructureOrdinateur[j].Iden := 0;
        StructureOrdinateur[j].Img := ComboBox1.Text;

        fillchar(StructureOrdinateur002, sizeof(StructureOrdinateur002), 0); //
        Розмір структури дорівнює нулю
        UnCranDeMoins := 0;

        for k := 1 to 52 do
        begin
          if (StructureOrdinateur[k].LoginConnecte = '')
            and (StructureOrdinateur[k].NomOrdinateur = '')
            and (StructureOrdinateur[k].Iden = 0) then
            begin
              inc(UnCranDeMoins);
            end
          else
            begin
              StructureOrdinateur002[k - UnCranDeMoins].LoginConnecte :=
StructureOrdinateur[k].LoginConnecte;
              StructureOrdinateur002[k - UnCranDeMoins].NomOrdinateur :=
StructureOrdinateur[k].NomOrdinateur;

```

```

        StructureOrdinateur002[k - UnCranDeMoins].Iden :=
StructureOrdinateur[k].Iden;
        StructureOrdinateur002[k - UnCranDeMoins].Img :=
StructureOrdinateur[k].Img;
        end;
    end;

    fillchar(StructureOrdinateur, sizeof(StructureOrdinateur), 0);
    for l := 1 to 52 do
    begin
        if (StructureOrdinateur002[l].LoginConnecte <> '')
            and (StructureOrdinateur002[l].NomOrdinateur <> '')
            and (StructureOrdinateur002[l].Iden <> 0) then
        begin
            StructureOrdinateur[l].LoginConnecte :=
StructureOrdinateur002[l].LoginConnecte;
            StructureOrdinateur[l].NomOrdinateur :=
StructureOrdinateur002[l].NomOrdinateur;
            StructureOrdinateur[l].Iden := StructureOrdinateur002[l].Iden;
            StructureOrdinateur[l].Img := StructureOrdinateur002[l].Img;
        end;
    end;
    end;
    end;
    end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.ComboBox1Change(Sender: TObject);
var
    j: integer;
    MessageStatut: string;
    Icon: TIcon;
    Image : TBitmap;
begin
    Icon := TIcon.Create;
    Image1.Picture := nil;
    ImageList1.GetIcon(ComboBox1.ItemIndex, Icon);
    ImageList1.GetBitmap(ComboBox1.ItemIndex, Image1.Picture.Bitmap);
    Trayicon.hIcon := Icon.Handle;
    Shell_NotifyIcon(Nim_Modify, @Trayicon);

    for j := 1 to 52 do
    begin
        if StructureOrdinateur[j].LoginConnecte = UserName then
            StructureOrdinateur[j].img := ComboBox1.Text;
        end;

        if ClientConnector = TRUE then
        begin
            MessageStatut := '#CTATYC#' + ComboBox1.Text + '#DE#' + EditLogin +
'#ЗАБЕРШЕЕННЯ РОБОТИ#';
            _MITTEVI_POVIDOMLENNYA_RL.ClientSocket.Socket.SendText(#13 +
MessageStatut);
        end;
    end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.cxGrid3DBTableView1CellDblClick(
    Sender: TcxCustomGridTableView; ACellViewInfo: TcxGridTableDataCellViewInfo;
    AButton: TMouseButton; AShift: TShiftState; var AHandled: Boolean);
var
    Login: string;
begin
    if
    ((ACellViewInfo.GridRecord.Values[TcxGridDBTableView(Sender).GetColumnByFieldNam
e('NOMPRENOM').Index] <> '')) then
    begin
        Login :=
ACellViewInfo.GridRecord.Values[TcxGridDBTableView(Sender).GetColumnByFieldName(
'NOMPRENOM').Index];
        if CompareStr(Login, EditLogin) <> 0 then

```

```

begin
  if assigned(MsgPerso) then
    begin
      MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' + Login +
      ' :';
      MsgPerso.Login.Caption :=
      ACellViewInfo.GridRecord.Values[TcxGridDBTableView(Sender).GetColumnByFieldName(
      'ORDI').Index];
      MsgPerso.LabeledEdit1.clear;
      MsgPerso.Show;
    end
  else
    begin
      MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
      MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' + Login +
      ' :';
      MsgPerso.Login.Caption :=
      ACellViewInfo.GridRecord.Values[TcxGridDBTableView(Sender).GetColumnByFieldName(
      'ORDI').Index];
      MsgPerso.cxGroupBox2.Caption := ' A : ' + Login + ' ';
      MsgPerso.Show;
    end;
  end
  else
    begin
      MessageDlg('Неможливо відправити повідомлення!', mtInformation, [mbOK],
      0);
    end;
  end;
end;

function T_MITTEVI_POVIDOMLENNYA_RL.MessageInfo: string;
var
  i: integer;
  MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur,
  MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl: string;
begin
  Application.ProcessMessages;
  MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl := '';
  MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur := '';
  MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur := '@щ*@';
  for i := 1 to NumeroArriveConnexion + 3 do
    begin
      if StructureOrdinateur[i].LoginConnecte <> '' then
        begin
          MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl :=
          MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl + 'µ' +
          StructureOrdinateur[i].LoginConnecte
          + '#IMG#' + StructureOrdinateur[i].img + '#ORDI#' +
          StructureOrdinateur[i].NomOrdinateur;
        end;
      end;
    end;

    if MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl <> '' then
      begin
        MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur :=
        MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur +
        MessageInfoListe_MITTEVI_POVIDOMLENNYA_eurl + 'µ@*!@µ';
        result := MessageInfoListe_MITTEVI_POVIDOMLENNYA_eur;
      end
    else
      begin
        result := '??';
      end;
    end;
  end;

  //Читання з сокету клієнта

  procedure T_MITTEVI_POVIDOMLENNYA_RL.ClientSocketRead(Sender: TObject;

```

```

    Socket: TCustomWinSocket);
var
    TEMPO: string;
begin
    Application.ProcessMessages;
    TEMPO := socket.ReceiveText;

    while (pos(#13, TEMPO) <> 0) do
    begin
        AnalyseMessageRecuParClient(gauche(#13, TEMPO));
        TEMPO := droite(#13, TEMPO);
    end;
    AnalyseMessageRecuParClient(TEMPO);
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.AnalyseMessageRecuParClient(Msg: string);
var
    MessageRecuClient, MsgTemp, LoginEnvoi, LoginRecoi, MessageTexte: string;
    j, k, l: integer;
    TableauLocal: array[1..52] of integer;
    Personnage, Ordi, Image: string;
    Ajouter, beep: Boolean;
    Registre: TRegistry;

    newlabel : TLabel;
begin
    beep := False;
    Registre := TRegistry.Create;
    Registre.RootKey := HKEY_LOCAL_MACHINE;
    Registre.OpenKey('\Software\IntraMSN\Couleur\', True);
    Registre.OpenKey('\Software\IntraMSN\Son\', True);
    if Registre.ValueExists('Son') then
        if Registre.ReadInteger('Son') = 1 then
            beep := True;
    MessageRecuClient := Msg;
    if (CompareStr(copy(MessageRecuClient, 1, 5), '') = 0) and
(CompareStr(copy(MessageRecuClient, length(MessageRecuClient) - 5, 6), 'α@*!@α')
= 0) then
    begin
        k := 0;
        fillchar(TableauLocal, sizeof(TableauLocal), 0);

        for j := 1 to length(MessageRecuClient) - 5 do
        begin
            if MessageRecuClient[j] = 'μ' then
            begin
                inc(k);
                TableauLocal[k] := j;
            end;
        end;

        if k <> 0 then
        begin
            for l := 1 to k - 1 do
            begin
                Personnage := copy(MessageRecuClient, TableauLocal[l] + 1,
(TableauLocal[l + 1]) - TableauLocal[l] - 1);
                if (CompareStr(Personnage, '') <> 0) and (comparestr(Personnage, ' ') <>
0) then
                begin
                    Image := Gauche('#ORDI#', Droite('#IMG#', Personnage));
                    Ordi := Droite('#ORDI#', Personnage);
                    Personnage := Gauche('#IMG#', Personnage);

                    Ajouter := TRUE;
                    DS_SOURCE.DataSet.First;

```



```

    end;
  end;
end;

if CompareStr(copy(MessageRecuClient, 1, 8), '#СТАТУС#') = 0 then
begin
  MessageRecuClient := Gauche('#ЗАВЕРШЕННЯ РОБОТИ#', MessageRecuClient);
  MessageRecuClient := Droite('#СТАТУС#', MessageRecuClient);
  LoginEnvoi := Droite('#DE#', MessageRecuClient);
  MessageRecuClient := Gauche('#DE#', MessageRecuClient);
  if beep then
    MessageBeep(MB_OK);
  RendreVisiblePremierPlan;

  DS_SOURCE.DataSet.First;
  while not DS_SOURCE.DataSet.Eof do
  begin
    if (ClientDataSet1NOMPrenom.Value = LoginEnvoi) then
    begin
      ClientDataSet1.Edit;
      if MessageRecuClient = 'Зайнятий' then
      ClientDataSet1IMAGE.Assign(IMG_PAUSE.Picture.Bitmap);
      if MessageRecuClient = 'Можу розмовляти' then
      ClientDataSet1IMAGE.Assign(IMG_LIGNE.Picture.Bitmap);
      if MessageRecuClient = 'Не турбувати' then
      ClientDataSet1IMAGE.Assign(IMG_OCCUPE.Picture.Bitmap);
      if MessageRecuClient = 'Розмовляю по телефону' then
      ClientDataSet1IMAGE.Assign(IMG_TEL.Picture.Bitmap);
      if MessageRecuClient = 'Тимчасово відсутній' then
      ClientDataSet1IMAGE.Assign(IMG_1MN.Picture.Bitmap);
      if MessageRecuClient = 'Зайнятий' then
      ClientDataSet1IMAGE.Assign(IMG_ABS.Picture.Bitmap);
      ClientDataSet1.Post;
    end;
    DS_SOURCE.DataSet.Next;
  end;
end;

if CompareStr(copy(MessageRecuClient, 1, 8), 'MsgPrive') = 0 then
begin
  MessageRecuClient := Gauche('#ЗАВЕРШЕННЯ РОБОТИ#', MessageRecuClient);
  MessageRecuClient := Droite('MsgPrive#DE#', MessageRecuClient);
  LoginEnvoi := Gauche('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#', MessageRecuClient);
  MessageRecuClient := Droite('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#', MessageRecuClient);
  LoginRecoi := Gauche('#ТІЛО ПОВІДОМЛЕННЯ#', MessageRecuClient);

  if (Comparestr(LoginEnvoi, EditLogin) = 0) or (Comparestr(LoginRecoi,
  EditLogin) = 0) then
  begin
    MessageTexte := Droite('#ТІЛО ПОВІДОМЛЕННЯ#', MessageRecuClient);
    if beep then
      MessageBeep(MB_OK);
    RendreVisiblePremierPlan;
    if (Memo_MITTEVI_POVIDOMLENNYA_.Lines.Count > 10) then
      Memo_MITTEVI_POVIDOMLENNYA_.Lines.Clear;

    if (LoginEnvoi <> UserName) then
    begin
      if assigned(MsgPerso) then
      begin
        if (MsgPerso.Caption <> 'Хочете відправити особисте повідомлення ' +
        LoginEnvoi + ' :') then
        begin
          MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
          MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
          LoginEnvoi + ' :';
          MsgPerso.cxGroupBox2.Caption := ' A : ' + LoginEnvoi + ' ';
          MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
        end;
      end;
    end;
  end;
end;

```

```

        MsgPerso.Show;
        FlashWindow(Application.Handle, true);
    end
    else
    begin
        MsgPerso.Memol.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
        FlashWindow(Application.Handle, true);
    end;
end
else
begin
    MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
    MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :';
    MsgPerso.cxGroupBox2.Caption := ' А : ' + LoginEnvoi + ' ';
    MsgPerso.Memol.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
    MsgPerso.Show;
    FlashWindow(Application.Handle, true);
end;
end
else
    MsgPerso.Memol.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
end;
end;

if CompareStr(copy(MessageRecuClient, 1, 12), 'MsgTransfert') = 0 then
begin
    MessageRecuClient := Gauche('#ЗАВЕРШЕННЯ РОБОТИ#', MessageRecuClient);
    MessageRecuClient := Droite('MsgTransfert#DE#', MessageRecuClient);
    LoginEnvoi := Gauche('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#', MessageRecuClient);
    MessageRecuClient := Droite('#ЗАГОЛОВОК ПОВІДОМЛЕННЯ#', MessageRecuClient);
    LoginRecoi := Gauche('#ТІЛО ПОВІДОМЛЕННЯ#', MessageRecuClient);

    if (Comparestr(LoginEnvoi, EditLogin) = 0) or (Comparestr(LoginRecoi,
EditLogin) = 0) then
    begin
        MessageTexte := Droite('#ТІЛО ПОВІДОМЛЕННЯ#', MessageRecuClient);
        if beep then
            MessageBeep(MB_OK);
            RendreVisiblePremierPlan;
            if (Memo_MITTEVI_POVIDOMLENNYA_.Lines.Count > 10) then
                Memo_MITTEVI_POVIDOMLENNYA_.Lines.Clear;

            if (LoginEnvoi <> UserName) then
            begin
                if assigned(MsgPerso) then
                begin
                    if (MsgPerso.Caption <> 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :') then
                    begin
                        MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
                        MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :';
                        MsgPerso.cxGroupBox2.Caption := ' А : ' + LoginEnvoi + ' ';
                        MsgPerso.Memol.Lines.Add(LoginEnvoi + ' Ви відправити файл: ' +
MessageTexte);
                        //////////////////////////////////
                    end;
                end;
            end;
        end;

        newlabel := TLabel.Create(MsgPerso.Memol);
        with newlabel do
        begin
            Parent := MsgPerso.Memol;
            Caption := 'testtt';
        end;
    end;
end;

```



```

else
begin
  MsgPerso := TMsgPerso.Create(_MITTEVI_POVIDOMLENNYA_RL);
  MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :';
  MsgPerso.cxGroupBox2.Caption := ' A : ' + LoginEnvoi + ' ';
  MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' ' + MessageTexte);
  MsgPerso.Show;
  Bouge := True;
  FlashWindow(Application.Handle, true);
end;
end
else
  MsgPerso.Memo1.Lines.Add('Доступний тільки для тих, хто повинен
бачити!');
end;
end;
if CompareStr(copy(MessageRecuClient, 1, 5), 'Підключено') = 0 then
begin
  Memo_MITTEVI_POVIDOMLENNYA_.Lines.Add('Сервер роз'єднаний... ' +
QuelHeureEstIl);
  Application.ProcessMessages;
  cxGrid3DBTableView1.ClearItems;

  ClientSocket.Active := FALSE;
  ClientSocket.Close;
  ClientConnector := FALSE;
end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.PourquoiDeconnecte(raison: string; iden:
integer);
begin
  case StrToInt(raison) of
    001: begin
      MessageDlg('Введіть логін та пароль.', mtInformation, [mbOK], 0);
      end;
    003: begin
      MessageDlg('«Сервер зайнятий. Будь ласка, повторіть спробу пізніше.',
mtInformation, [mbOK], 0);
      end;
  end;
  end;
  Application.ProcessMessages;
  cxGrid3DBTableView1.ClearItems;
  ClientSocket.Active := FALSE;
  ClientSocket.Close;
  ClientConnector := FALSE;
end;

{Зміна вкладки}

procedure T_MITTEVI_POVIDOMLENNYA_RL.LabeledEditMessageEcritKeyPress(Sender:
TObject;
  var Key: Char);
var
  DroitEcrire: boolean;
  MessageAEnvoyer: string;
begin
  if (Key = #13)
  and (ClientConnector = TRUE) then
  begin
    key := #0;
    DroitEcrire := FALSE;
    DS_SOURCE.DataSet.First;

    while not DS_SOURCE.DataSet.Eof do
    begin
      if CompareStr(EditLogin, ClientDataSet1NOMPrenom.Value) = 0 then
      begin

```



```

    Image1.Picture.Bitmap := IMG_OCCUPE.Picture.Bitmap;
    ClientDataSet1IMAGE.Assign(IMG_OCCUPE.Picture.Bitmap);
end;
if MessageStatut = '&Розмовляю по телефону' then
begin
    ImageList1.GetIcon(2, Icon);
    Image1.Picture.Bitmap := IMG_TEL.Picture.Bitmap;
    ClientDataSet1IMAGE.Assign(IMG_TEL.Picture.Bitmap);
end;
if MessageStatut = 'На перерві' then
begin
    ImageList1.GetIcon(3, Icon);
    Image1.Picture.Bitmap := IMG_PAUSE.Picture.Bitmap;
    ClientDataSet1IMAGE.Assign(IMG_PAUSE.Picture.Bitmap);
end;
if MessageStatut = 'Тимчасово відсутній' then
begin
    ImageList1.GetIcon(4, Icon);
    Image1.Picture.Bitmap := IMG_1MN.Picture.Bitmap;
    ClientDataSet1IMAGE.Assign(IMG_1MN.Picture.Bitmap);
end;
if MessageStatut = 'Відсутній' then
begin
    ImageList1.GetIcon(5, Icon);
    Image1.Picture.Bitmap := IMG_ABS.Picture.Bitmap;
    ClientDataSet1IMAGE.Assign(IMG_ABS.Picture.Bitmap);
end;
ClientDataSet1.Post;
Trayicon.hIcon := Icon.Handle;
Shell_NotifyIcon(Nim_Modify, @Trayicon);

for j := 1 to 52 do
begin
    if StructureOrdinateur[j].LoginConnecte = UserName then
        StructureOrdinateur[j].img := ComboBox1.Text;
    end;
    if ClientConnector = TRUE then
    begin
        MessageStatut := '#СТАТУС#' + ComboBox1.Text + '#DE#' + EditLogin +
        '#ЗАБЕПШЕННЯ РОБОТИ#';
        _MITTEVI_POVIDOMLENNYA_RL.ClientSocket.Socket.SendText(#13 +
        MessageStatut);
    end;
    end;
    DS_SOURCE.DataSet.Next;
end;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.BitBtnDeconnexionClick(Sender: TObject);
begin
    DireQueOnSeDeconnecte;
end;

procedure T_MITTEVI_POVIDOMLENNYA_RL.FormResize(Sender: TObject);
begin
    Application.OnMinimize := Minimize;
end;

initialization
    NumeroArriveConnexion := 0;

end.
```

AudioVideo.pas - відео- та аудіозв'язок

```

unit AudioVideo;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, Registry, WinSkinStore, WinSkinData,
  AMixer, MMSystem, ComCtrls, dxGDIPlusClasses, Camera;

type
  TCAudioVideo = class(TForm)
    Mixer: TAudioMixer;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox2: TGroupBox;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label3: TLabel;
    Label2: TLabel;
    Label4: TLabel;
    LabelStereo: TLabel;
    Image2: TImage;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    ComboBox3: TComboBox;
    TrackBar: TTrackBar;
    CheckBox: TCheckBox;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Image1: TImage;
    ComboBox4: TComboBox;
    ComboBox5: TComboBox;
    ComboBox6: TComboBox;
    TrackBar1: TTrackBar;
    CheckBox1: TCheckBox;
    GroupBox5: TGroupBox;
    GroupBox6: TGroupBox;
    Label5: TLabel;
    Image3: TImage;
    TrackBar2: TTrackBar;
    Camera1: TCamera;
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    TrackBar3: TTrackBar;
    Label10: TLabel;
    Label11: TLabel;
    TrackBar4: TTrackBar;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    procedure TrackBarChange(Sender: TObject);
    procedure CheckBoxClick(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure TrackBar2Change(Sender: TObject);
    procedure TabSheet2Show(Sender: TObject);
    procedure TabSheet1Show(Sender: TObject);
    procedure ComboBox6Change(Sender: TObject);
    procedure ComboBox4Change(Sender: TObject);
  end;

```

```

    procedure ComboBox5Change(Sender: TObject);
    procedure TrackBar3Change(Sender: TObject);
    procedure TrackBar4Change(Sender: TObject);
private
    { Private declarations }
    Setting:Boolean;
public
    { Public declarations }
end;

var
    CAudioVideo: TCAudioVideo;

implementation

uses Unit_MITTEVI_POVIDOMLENNYA_RL, Unit_MsgPerso;

{$R *.dfm}

procedure TCAudioVideo.FormCreate(Sender: TObject);
var A:Integer;
begin
    For A := 0 to Mixer.MixerCount - 1 do
        ComboBox3.Items.Add (Mixer.ProductName);
    If (ComboBox3.Items.Count > 0) then
        ComboBox3.ItemIndex := 0;
    ComboBox3Change (Sender);

    For A := 0 to Mixer.MixerCount - 1 do
        ComboBox6.Items.Add (Mixer.ProductName);
    If (ComboBox6.Items.Count > 0) then
        ComboBox6.ItemIndex := 0;
    ComboBox6Change (Sender);

end;

procedure TCAudioVideo.MixerControlChange(Sender: TObject; MixerH, ID: Integer);
begin
    ComboBox2Change (Self);
end;

procedure TCAudioVideo.TabSheet1Show(Sender: TObject);
begin
    if(Camera1.Actif)Then
        Camera1.Actif := False;
end;

procedure TCAudioVideo.TabSheet2Show(Sender: TObject);
begin
    if(not Camera1.Actif)Then
        Camera1.Actif := True;
end;

procedure TCAudioVideo.TrackBar2Change(Sender: TObject);
begin
    Camera1.FramesPreview := TrackBar2.Position;
end;

procedure TCAudioVideo.TrackBar3Change(Sender: TObject);
begin
    Camera1.FramesCaptura := TrackBar3.Position;
end;

procedure TCAudioVideo.TrackBar4Change(Sender: TObject);
begin
    Camera1.Secondes:=TrackBar4.Position;
end;

```

```

procedure TCAudioVideo.TrackBarChange(Sender: TObject);
begin
  If (not Setting) then
  begin
    Setting:=True;
    Mixer.SetVolume (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;

procedure TCAudioVideo.BitBtn1Click(Sender: TObject);
begin
  Close;
end;

procedure TCAudioVideo.CheckBoxClick(Sender: TObject);
begin
  If not Setting then
  begin
    Setting:=True;
    Mixer.SetVolume (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;

procedure TCAudioVideo.ComboBox1Change(Sender: TObject);
var A:Integer;
begin
  ComboBox2.Items.Clear;
  For A:=0 to Mixer.Destinations[ComboBox1.ItemIndex].Connections.Count-1 do
  ComboBox2.Items.Add(Mixer.Destinations[ComboBox1.ItemIndex].Connections[A].Data.
szName);
  If ComboBox2.Items.Count>0 then
  begin
    ComboBox2.ItemIndex:=0;
    ComboBox2Change (Self);
  end;
end;

procedure TCAudioVideo.ComboBox2Change(Sender: TObject);
var L,R,M:Integer;
    VD,MD:Boolean;
    Stereo:Boolean;
    IsSelect:Boolean;
begin
  Mixer.GetVolume (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,L,R,M,Stereo,VD,MD,IsSelect);
  Setting:=True;
  TrackBar.Visible:=not VD;
  Label1.Visible:=not VD;
  Label3.Visible:=VD;
  If TrackBar.Visible then
    TrackBar.Position:=L;
  CheckBox.Visible:=not MD;
  Label2.Visible:=not MD;
  Label4.Visible:=MD;
  If CheckBox.Visible then
    CheckBox.Checked:=M<>0;
  If (Stereo) then
    LabelStereo.Caption := '- Stereo -'
  else
    LabelStereo.Caption := '- Mono -';
  Setting:=False;
end;

procedure TCAudioVideo.ComboBox3Change(Sender: TObject);

```

```

var A:Integer;
begin
  If (ComboBox3.ItemIndex >= 0) AND (ComboBox3.ItemIndex < Mixer.MixerCount)
  then
    Mixer.MixerId := ComboBox3.ItemIndex;
    ComboBox1.Items.Clear;
    If Mixer.MixerCount>0 then
      begin
        For A:=0 to Mixer.Destinations.Count-2 do
          ComboBox1.Items.Add (Mixer.Destinations[A].Data.szName);
          If ComboBox1.Items.Count>0 then
            begin
              ComboBox1.ItemIndex:=0;
              ComboBox1Change (Self);
            end;
          end
        else
          begin
            ComboBox1.OnChange:=nil;
            ComboBox2.OnChange:=nil;
            TrackBar.OnChange:=nil;
            CheckBox.OnClick:=nil;
            MessageDlg ('Немає змішувача у системі!',mtError,[mbOK],0);
          end;
          Setting:=False;
        end;

procedure TCAudioVideo.ComboBox4Change(Sender: TObject);
var A:Integer;
begin
  ComboBox5.Items.Clear;
  For A:=0 to Mixer.Destinations[ComboBox4.ItemIndex + 1].Connections.Count-1 do
    ComboBox5.Items.Add(Mixer.Destinations[ComboBox4.ItemIndex +
1].Connections[A].Data.szName);
  If ComboBox5.Items.Count>0 then
    begin
      ComboBox5.ItemIndex:=0;
      ComboBox5Change (Self);
    end;
  end;

procedure TCAudioVideo.ComboBox5Change(Sender: TObject);
var L,R,M:Integer;
    VD,MD:Boolean;
    Stereo:Boolean;
    IsSelect:Boolean;
begin
  Mixer.GetVolume (ComboBox4.ItemIndex,ComboBox5.ItemIndex-
1,L,R,M,Stereo,VD,MD,IsSelect);
  Setting:=True;
  TrackBar1.Visible:=not VD;
  Label6.Visible:=not VD;
  Label8.Visible:=VD;
  If TrackBar1.Visible then
    TrackBar1.Position:=L;
  CheckBox1.Visible:=not MD;
  Label7.Visible:=not MD;
  Label9.Visible:=MD;
  If CheckBox1.Visible then
    CheckBox1.Checked:=M<>0;
  Setting:=False;
end;

procedure TCAudioVideo.ComboBox6Change(Sender: TObject);
var A:Integer;
begin
  If (ComboBox6.ItemIndex >= 0) AND (ComboBox6.ItemIndex < Mixer.MixerCount)
  then
    Mixer.MixerId := ComboBox6.ItemIndex;

```

```
ComboBox4.Items.Clear;
If Mixer.MixerCount>0 then
begin
  For A:=1 to Mixer.Destinations.Count-1 do
    ComboBox4.Items.Add (Mixer.Destinations[A].Data.szName);
  If ComboBox4.Items.Count>0 then
  begin
    ComboBox4.ItemIndex:=0;
    ComboBox4Change (Self);
  end;
end
else
begin
  ComboBox4.OnChange:=nil;
  ComboBox5.OnChange:=nil;
  TrackBar.OnChange:=nil;
  CheckBox.OnClick:=nil;
  MessageDlg ('Немає змішувача у системі!',mtError,[mbOK],0);
end;
Setting:=False;
end;

end.
```

КБПЗ_2024

About.pas - довідка

```
unit About;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls;

type
  TForm2 = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    Image5: TImage;
    Image6: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.FormCreate(Sender: TObject);
begin
  Memo1.Clear;
  Memo1.Lines.Add('БАКАЛАВРСЬКИЙ ПРОЕКТ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Програмне забезпечення системи обміну миттєвими повідомленнями  
у мережі Інтернет');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Дреєв О.М. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент  Гарбар Вадим Володимирович');
  Memo1.Lines.Add('                                     гр. КІ-20');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Кропивницький 2024');
  Memo1.Lines.Add('');

end;

procedure TForm2.Button1Click(Sender: TObject);
begin
  Form2.Close;
end;

end.
```