

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 інформацією у мережі
під керуванням ОС Windows 11/Windows Server 2022”**

КБГЗ-2024

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

Керівник проекту
кандидат технічних наук, доцент
_____ Коваленко А.С.
« ____ » _____ 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

Маламужу Владиславу Сергійовичу

(прізвище, ім'я, по батькові)

- Тема роботи *Програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022*
- Керівник роботи *Коваленко Анна Степанівна, 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
- Строк подання студентом роботи до захисту *23.05.2024 р.*
- Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Перегляд аналогічних існуючих систем.*
 - Опис і обґрунтування проектних рішень.*
 - Етапи програмування системи.*
 - Впровадження системи в промислову експлуатацію.*
 - Висновки*
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

<i>Структурна схема системи</i>	<i>1 аркуш</i>
<i>Функціональна схема системи</i>	<i>1 аркуш</i>
<i>Діаграма процесів</i>	<i>1 аркуш</i>
<i>Блок-схема алгоритму роботи додатку</i>	<i>2 аркуша</i>

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

Коваленко А.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2024 р.

Підпис здобувача

Маламуж В.С.
(прізвище та ініціали)

АНОТАЦІЯ

Маламуж В.С. Програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022. 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 інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

Метою розробки є програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

Результат роботи – програмна реалізація системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

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 Sydney.

Ключові слова: комп'ютерна інженерія, система обміну інформацією

ABSTRACT

Malamuzh V.S. Network Information Exchange System Software under OC Windows 11/Windows Server 2022. 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 information exchange in the network under the management of OC Windows 11/Windows Server 2022.

The purpose of development is the software of the information exchange system in the network under the management of OC Windows 11/Windows Server 2022.

The result of the work is the software implementation of the information exchange system in the network under Windows 11/Windows Server 2022 OC control.

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 Sydney environment.

Keywords: computer engineering, information exchange system

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	22
2.3 Розгорнута постановка завдання	28
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	29
3.1 Опис функціонування системи	29
3.2 Розробка структурної схеми.....	43
3.3 Розробка функціональної схеми	46
3.4 Розробка діаграми процесів.....	50
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	52
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	52
4.2 Захист розробленого програмного забезпечення.....	61
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	66
6 ОСНОВНІ ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71

					ВКРБ-123.24.0009.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022	Літ.	Аркуш	Аркушів
Розроб.	Маламуж В.С.					Б	1	77
Перев.	Коваленко А.С.							
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-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Програмне забезпечення обміну інформацією в мережі під керуванням ОС Windows 11/Windows Server 2022 – це зручний в обігу Інтернет-засіб, що інформує в будь-який час про тих, хто перебуває в он-лайнному режимі й дозволяє при бажанні спілкуватися з ними. Програмне забезпечення шукає в Мережі друзів або співрозмовників, попереджаючи в реальному часі про їхню появу в Інтернеті. Усунуто необхідність проводити пошук у каталогах щораз, коли хочете спілкуватися з певною людиною. За допомогою програмного забезпечення можливо вести бесіду (чат), відправляти й приймати повідомлення, файли й веб-посилання URL, грати в ігри або просто спілкуватися зі своїми приятелями.

Програмне забезпечення дозволяє вибирати саме той режим спілкування, що хочете використовувати. Поза залежністю від додатка, будь то чат, дошка повідомлень, обмін даними або голосовими повідомленнями, пересилання файлів або Інтернет-ігри, програмне забезпечення буде повністю приймати повідомлення в реальному часі. Програмне забезпечення підтримує безліч популярних Інтернет-додатків і служить універсальною базою, звідки можливо запуснути програми для спілкування, такі як Microsoft NetMeeting або Netscape CoolTalk). Вона також може бути використана в багатокористувальницькому режимі, де групи користувачів можуть проводити конференції. Ця програма працює в фоні, забираючи мінімум пам'яті й ресурсів. Під час роботи з іншими додатками програмне забезпечення буде попереджати, коли ваші друзі будуть перебувати в Інтернеті, дозволяючи тим самим ефективно працювати, маючи під рукою великий набір Інтернет-інструментів. Серед яких – чат, обмін повідомлення й електронною поштою, пересилання URL і файлів. Всі ці функції об'єднані в одну зручну в обігу програму, що відмінно вписується в систему.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

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

– Огляд існуючих систем обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

– Дослідження системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

– Програмна реалізація системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

У цей момент у комунікаційному середовищі більшу популярність одержує такий мережний сервіс, як служба миттєвих повідомлень (передачі даних у мережі), а також IP-телефонія. Це обумовлено тим, що при використанні даних сервісів можлива більша економія засобів у спілкуванні з людьми на далеких відстанях і в інших країнах.

Передача даних у мережі залишається вибором способу спілкування для багатьох людей, тому що надає більше «особисте» спілкування, ніж електронна пошта, але менш нав'язливе, чим телефон.

Передача даних у мережі також корисна при спілкуванні в надзвичайних ситуаціях, тому що Інтернет може функціонувати під час неполадок на телефонній лінії.

Сервіси миттєвих повідомлень поступово витісняють електронну пошту в результаті наступних недоліків останньої:

- Спам.
- Низький рівень безпеки.
- Незручності при активній переписці.
- Відносно повільна доставка повідомлень.
- Відсутність інформації про стан абонентів.

Основна причина спаму для електронної пошти, це відсутність надійної системи ідентифікації відправника повідомлення. У сьогоdnшній системі електронної пошти спамери можуть поставити в поле «Відправник» будь-яку інформацію. Це ж є однією з основних причин низької безпеки.

Незручність активної переписки полягає в складності контролю всієї історії спілкування. У результаті, активним користувачам електронної пошти доводиться самим організувати способи зберігання логів.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1.2 Область застосування

Областю застосування розроблюємого програмного забезпечення є обмін інформацією в мережі під керуванням ОС Windows 11/Windows Server 2022. Для цього запропоновано використовувати протоколи сервісів миттєвої передачі повідомлень. Доставка повідомлень по електронній пошті повільна, якщо порівнювати саме із системами миттєвих повідомлень. Затримка в 1-10 секунд (офіційна специфікація затверджує про можливість доставки пошти в плинні 24 годин) утрудняє спілкування в режимі реального часу. Крім того, занадто часта перевірка пошти може трактуватися сервером як мережна атака, приводячи до тимчасового блокування користувача.

Експерти міжнародної дослідницької компанії TNS з'ясували, що SMS і електронна пошта на мобільному телефоні поступово втрачають свою популярність. «Неголосове» спілкування тепер усе більше будується на обміні миттєвими повідомленнями. За даними дослідження, у якому взяли участь 17 тис. респондентів з 30 країн миру, серед власників мобільних телефонів, що використовують при спілкуванні служби миттєвих повідомлень, 61% віддає перевагу саме цьому способу зв'язку, 55% використовують SMS і тільки 12% – електронну пошту.

Вибір передачі даних у мережі замість SMS цілком логічний: «Як тільки власники мобільних телефонів одержують доступ в інтернет і мають можливість використовувати служби миттєвих повідомлень, вартість текстових послань різко знижується й прагне практично до нуля, тому що користувачі в цьому випадку платять тільки за трафік, а самі повідомлення нічого їм не коштують. Крім того, велика кількість людей звикло спілкуватися за допомогою служб миттєвих повідомлень через інтернет на персональних комп'ютерах, тому вони з великим бажанням перемикаються на аналогічне спілкування й на своїх мобільних телефонах. А так як усе більше операторів стільникового зв'язку пропонують своїм клієнтам необмежений доступ в інтернет, то ми можемо із упевненістю

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

говорити про подальший ріст популярності миттєвих повідомлень і зниження ролі SMS і повідомлень по електронній пошті», – пояснюють експерти TNS.

На сьогоднішній день в усьому світі за допомогою миттєвих повідомлень по мобільному телефоні спілкуються 8% людей. Найбільше популярно таке спілкування в Гонконгу, де цей показник досягає 23%. Далі впливають Китай (16%), Саудівська Аравія (15%), ПАР (12%), Індія (11%) і Бразилія (10%).

Недоліки й слабкі сторони миттєвої передачі даних у мережі

Незважаючи на всі свої переваги, сервіси передачі даних у мережі не позбавлені недоліків. От головні з них:

– Спам. Сервіс передачі даних у мережі також є об'єктом небажаної реклами (передачі даних у мережі спам іноді називають spam). Спам – найбільш дратівна проблема для користувачів сервісів миттєвих повідомлень. Багато організацій навмисно використовують можливості сервісу для «безкоштовної реклами» шляхом масового розсилання спаму великій кількості користувачів. Крім того, такий тип розсилання одержав популярність у груп людей, метою яких є розкрадання інформації або нанесення навмисної шкоди системі користувача.

– Надмірність переданої інформації. Дана проблема характерна для таких протоколів, як XMPP (Jabber). Якщо в корпоративній мережі багато працівників, залучених у чат, це може вдарити по завантаженості мережі й продуктивності.

– Недостатні обчислювальні можливості сервера. Нерідкі відключення від сервера користувачів у таких протоколах, як ICQ. Багато користувачів зустрічалися з такими повідомленнями, як «ліміт підключень перевищений». Також були замічені періодичні відключення від серверів Skype.

– Низький рівень мережної безпеки. Особливо даний недолік небезпечний у корпоративних мережах, які дозволили використовувати сервіси миттєвих повідомлень серед персоналу. Даний пункт заслуговує на особливу увагу й більш детально розглядається нижче.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Мережна безпека служб миттєвих повідомлень

Різні передачі даних у мережі додатка використовують різні пропріетарні протоколи, і стандартна конфігурація брандмауера може не виявити їх. Багато передач даних у мережі програми можуть обходити системи автентифікації. Деякі клієнти передачі даних у мережі можуть використовувати порти відмінні від тих, які асоціюються з передачею даних у мережі, навіть звичайно відкриті порти, такі як 80.

Основні проблеми безпеки сервісу передачі даних у мережі:

– P2P обмін. Даний вид обміну відбувається при передачі файлів такими протоколами, як ICQ. P2P обмін дає можливість довідатися IP адресу клієнта й, прямо підключившись до комп'ютера, використовувати його уразливості.

– Шифрування даних і безпечна автентифікація. Багато протоколів на сьогоднішній день часто не використовують шифрування під час чату й/або передають паролі від облікових записів у відкритому виді. Даний недолік неприпустимий у корпоративних мережах, які мають на увазі пересилання конфіденційної інформації й конфіденційне спілкування.

Виходячи із всіх статистичних досліджень, можна побачити популярність, що підвищується, сервісів миттєвих повідомлень. Актуальність роботи полягає в підвищенні якості роботи сервісів шляхом рішення зазначених вище недоліків.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

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

Програми обміну інформацією у мережі називають месенджерами. Месенджери (Instant messengers) уже давно не виправдують свою назву. По-перше вони мають безліч корисних функцій, таких як IP-телефонія, відео-конференції, інструменти для спільної роботи. По-друге вони дозволяють спілкуватися не тільки в синхронному (instant) режимі, але й в асинхронному. Основною перевагою месенджерів над використанням e-mail й телефону є індикація статусу співрозмовника, тобто завжди видно, що він в онлайні або зайнятий, або в офф-лайн. Крім того, якщо раніше месенджери вимагали установки на комп'ютер спеціальної програми-клієнта, то тепер можливо спілкуватися абсолютно з будь-якого комп'ютера, підключеного до інтернету. Тому месенджери стали чи не основним засобом для комунікації у віртуальних робочих групах.

Розглянемо найбільш популярні месенджери в ОС Windows. До них відносяться наступні:

- Windows Live Messenger.
- ICQ.
- Yahoo Messenger.
- Google Talk.

Проведемо дослідження перерахованих вище месенджерів.

Windows Live Messenger

Windows Live Messenger – програма миттєвого обміну повідомленнями для Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Server 2008 і Windows Mobile. Є спадкоємцем програми MSN Messenger і випущена під новим ім'ям компанією Microsoft 13 грудня 2005 року. Є одним з компонентів Windows Live – набору мережних служб від Microsoft.

Клієнт підключається до Microsoft .NET Messenger Service. Корпорації також можуть інтегрувати власний Live Communication Server і Active Directory у робочу мережу для своїх клієнтів. Головні клієнти з мультипротоколами також можуть підключитися до служби.

Програма, раніше відома як MSN Messenger, знайшла нове ім'я й приєдналася до сімейства Windows Live. Додаток не тільки перемінив зовнішній вигляд, але й стало більше функціональним. При інсталяції пропонується встановити не тільки саму програму, але й вибрати msn.com у якості стартової інтернет-сторінки, завантажити Windows Live Toolbar та інші дріб'язки. Це за замовчуванням відзначено галочками, і неухважний користувач може ненавмисно їх активізувати, сам того не бажаючи. Для використання Live Messenger необхідний аккаунт Windows Live, який можна відкрити за адресою get.live.com. Доступ на цей ресурс можливий із самої програми, але краще було б, якби реєстраційна система вже була в неї інтегрована.

Після завершення реєстрації на екрані монітора з'являється список контактів, упорядкований по групах або ж за абеткою. Правий «клік» на контактні викликає меню з комунікаційними й іншими можливостями. При спілкуванні в режимі чату у верхній частині вікна доступні всі опції зв'язку, а в нижньої – вікно для введення тексту й вибір емотиконів (смайликів). Недосвідчений користувач, можливо, стикнеться із труднощами при пошуку звичного меню у верхній частині вікна (File, Tools і т.д.). Підкоряючись задуму дизайнера, це меню ховається за кнопками, розташованими поруч із символом мінімізації.

Цей засіб спілкування має дуже широкі можливості персоніфікації. У кожного вікна можна змінити схему кольорів (щоб, наприклад, при діалозі з колегами букви були синіми, а з дітьми – червоними). Точно так само для кожного вікна можна встановити своє тло. Програма має у своєму розпорядженні

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

досить широкий вибір емотиконів (смайликів), однак на відміну від Skype, Windows Live Messenger дозволяє приєднати до їхнього списку будь-яке зображення, що будуть бачити всі співрозмовники. Можна робити й більш барвисті знаки уваги, відсилаючи невеликі flash-анімації (winks), доповнені звуковими ефектами. Голосове спілкування в Live Messenger відбувається в тому ж вікні, де й чат. Складається таке враження, що програма більше орієнтована на комунікації за допомогою букв, ніж голосу. На своїй домашній сторінці творці продукту запевняють, що за допомогою Live Messenger можна дзвонити абонентам мобільної й фіксованої мережі. Однак ретельно дослідивши програму й інструкцію виробника, нам так і не вдалося знайти відповідне меню й здійснити телефонні дзвінки. Можливо, ця послуга просто недоступна користувачам певних країн, тому що Windows Live Messenger передбачений, головним чином, для ринку США.

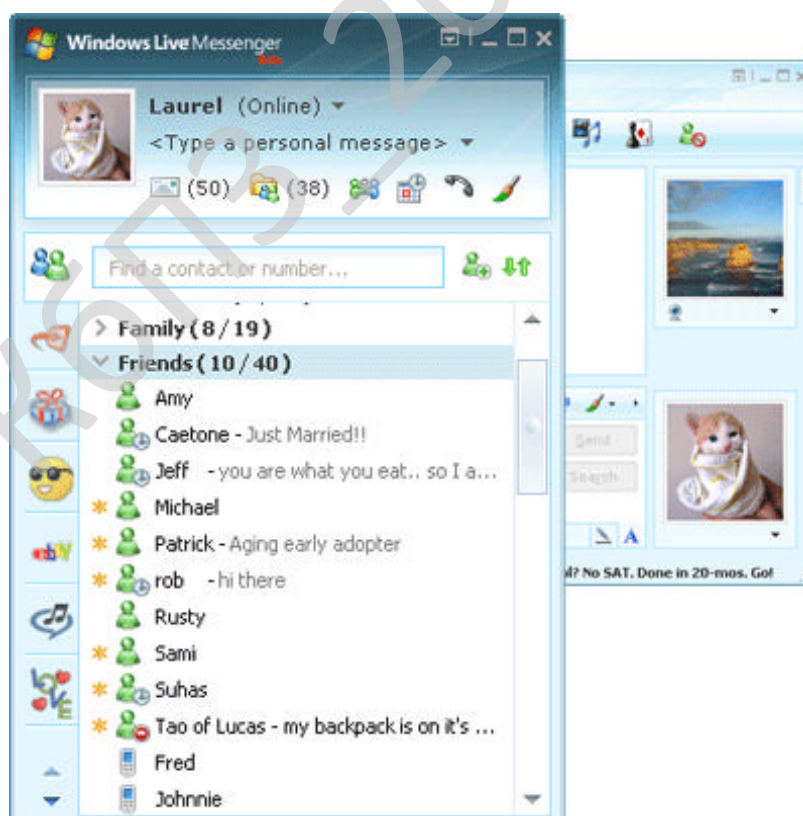


Рисунок 2.1 – Інтерфейс користувача Windows Live Messenger

За допомогою веб-камери програма підтримує й відеорозмови. Вони здійснюються в тому же вікні, що й чат, хіба що фотографія в контактах заміняється на передане камерою відеозображення. Користувачам надається також можливість пересилати файли, грати в спільні ігри й відсилати SMS на незареєстрованні в контактах номери мобільних телефонів.

Одна з найбільш корисних функцій Windows Live Messenger – Remote assistance, або можливість віддаленої допомоги. Вона дозволяє звернутися по допомогу до іншого користувача, з його згоди одержати віддалений доступ до його комп'ютера й працювати з ним, як зі своїм власним. Безумовно, Remote assistance буде корисна у випадках, коли потрібен доступ до файлів або документів, які перебувають на відстані. Варто додати, що Live Messenger дозволяє спілкуватися й з користувачами Yahoo! Messenger.

На жаль, при використанні Windows Live Messenger ми кілька разів зіштовхувалися із проблемами при спробі комусь подзвонити. Програма може «підвиснути» і, оскільки володіє дуже візуально насиченим інтерфейсом, дуже вимоглива до технічних ресурсів комп'ютера. Однак з обміном повідомленнями все в порядку – послання завжди знаходить свого адресата. Якість звуку теж гарне, якщо ви володієте досить високошвидкісним інтернет-каналом.

Виводи: Windows Live Messenger призначений в основному для чату – складається враження, що функція «Подзвонити на інший комп'ютер» є всього лише факультативною. Програма візуально багата й оснащена безліччю доповнень, пов'язаних з текстовим спілкуванням. На жаль, їй не вистачає можливості просто подзвонити абонентів мобільної або фіксованої мережі.

ICQ

ICQ – комп'ютерна прикладна програма із графічним інтерфейсом користувача, офіційний клієнт служби миттєвого обміну повідомленнями ICQ. Перша версія програми була випущена в листопаді 1996 року й стала однією з перших інтернет-чатів. Сьогодні ICQ використовують більше 38 млн користувачів в усьому світі. Працює за протоколом OSCAR.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12



Рисунок 2.2 – Інтерфейс користувача ICQ

З кожним з контактів можна вести особисту переписку – для цього треба відкрити вікно діалогу. Вікно діалогу розділене на дві частини: поле для введення й вікно чату. У поле для введення користувач набирає своє повідомлення, після чого відправляє його іншому. Вся переписка показується у вікні чату із вказівкою ніка відправника й часу відправлення повідомлення. Якщо відправник не відключив цю можливість, то у вікні діалогу буде відображатися повідомлення про набір.

У вікні діалогу й списку контактів показуються рекламні баннери формату Adobe Flash (Macromedia Flash).

Логотип ICQ використовується для візуалізації процесу підключення клієнта до сервера, а також як ідеограма статусів присутності. Для позначення статусу «у мережі» використовується ідеограма повністю зеленої квітки, «не в

мережі» – повністю червоного. Статус «відійшов» позначається зеленою квіткою із прикріпленою до нього білим папірцем, на якій або написано «Away», або не написано нічого; статус «недоступний» – зеленою квіткою з великим білим папірцем поперек нього, на якій написано «N/A». Якщо користувач перебуває в режимі невидимості, то до ідеограм будь-яких статусів додаються чорні окуляри, «надягнуті» на квітку.

На жаль, хазяї проекту з метою його подальшої комерціалізації й добування прибутків, останнім часом стали неодноразово міняти протокол передачі повідомлень, що найчастіше не дозволяє нормально використовувати альтернативні програми-клієнти без нав'язливої реклами.

Візуальний інтерфейс програми і її настроювання також викликають дорікання користувачів, що звикли до компактних і зручних програм-конкурентам. Незручно переглядати профілі користувачів чату, тому що програма увесь час підключається для цього в інтернет, відкриваючи окрему сторінку у веб-браузері. В "рідній" програмі ICQ не виходить відправляти навіть відносно не довгі текстові повідомлення (обмеження по довжині), але які можна було відправляти з інших програм-клієнтів.

Yahoo Messenger

Yahoo! Messenger (скорочено Y!M) – програма миттєвого обміну повідомленнями компанії Yahoo!, що використовує власний протокол. Існує для платформ Windows, Mac, Linux (Unix). Поширюється безкоштовно. Для використання потрібна попередня реєстрація на порталі Yahoo!

Програма призначена для спілкування в режимі реального часу з іншими користувачами інтернету. Передбачено наступні можливості: текстове спілкування, голосове спілкування, зокрема багатокористувальницький голосовий чат, відеоконференції, дзвінки на мобільні й стаціонарні телефони, обмін файлами, онлайнві гри, що набудовуються онлайнві трансляції музики, що набудовується інтегрований доступ до сервісів порталу Yahoo! Не має версії з русифікованим інтерфейсом, але розробки в цьому напрямку вже ведуться.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Використовуючи новітню версію, її користувач може зв'язатися й із власником Windows Live Messenger, що істотно розширює комунікаційні можливості. Тому що ця комунікаційна програма передбачена, в основному, для ринку США, то російською мовою меню програми недоступно. На жаль, творець Yahoo! Messenger – компанія Yahoo – хоче, щоб користувач не тільки встановив програму, але й зробив Yahoo! стартовою сторінкою й пошуковою системою за замовчуванням. Тому треба бути уважним і відзначити, що хочеш від цього відмовитися. При виборі типового пакета автоматично інсталюється й Yahoo! Toolbar, і Yahoo! PopUp Blocker. Це дуже дратує, тому як більшість користувачів вибирають не Typical, а Custom інсталяцію. Сам процес установки займає набагато більше часу, ніж у випадку з іншими VoIP програмами. Варто додати, що інсталяційна програма повідомляє про неможливість працювати разом з пошуковою системою Firefox, що теж треба враховувати. Після неймовірно тривалого процесу установки можна, нарешті, приступитися до звичної реєстрації серед користувачів. Для роботи з Yahoo! Messenger необхідний Yahoo ID. Якщо ви вже зареєстрували свою поштову адресу на Yahoo.com, той же ідентифікатор можна використовувати, реєструючись у системі. Якщо ж ідентифікатора ні, його можна одержати на вищезгаданому ресурсі. При першому запуску програми пропонується імпортувати контакти з інших місць, наприклад, з адресної книги електронної пошти. Після цього можна починати роботу. Відразу стає помітно, що програма перезавантажена, тому орієнтуватися в меню досить важко. У нижній частині вікна перебувають установлені доповнення (plug-ins), вікно пошуку Yahoo і, на жаль, реклама. У верхній частині програмного вікна розташоване звичне меню – Contacts, Actions і т.д. Як і в інших програм, всі дії з Yahoo Messenger можна здійснювати через правий клік на обраному ім'ї в списку контактів. Клацнувши на Send an instant message, відкриваємо вікно обміну повідомленнями. Це просто й зрозуміло – меню у верхній частині, відображення розмови посередині, а поле введення повідомлення, вибір емотикона (смайликів) і різні налаштування – у нижній частині.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15



Рисунок 2.3 – Інтерфейс користувача Yahoo! Messenger

Як і будь-яка VoIP програма, Yahoo Messenger дозволяє дзвонити на комп'ютер іншого користувача за допомогою списку контактів, а також на мобільні й фіксовані телефони всього миру. Те саме що Windows Live Messenger, ця програма запитує згоди на контакт того абонента, з яким ви хочете зв'язатися.

Тільки після одержання дозволу можна дзвонити й посилати повідомлення. Для голосових дзвінків потрібно абонувати Yahoo Voice. Подзвонити дуже просто, треба всього лише вибрати відповідний номер і натиснути кнопку «Дзвонити». Приблизно так само можна подзвонити на інший комп'ютер, тільки для цього треба вибрати ім'я користувача в контактному аркуші. У добавок до звичних функцій, Yahoo Messenger пропонує й різні додаткові. Багато хто з них доступні за замовчуванням – наприклад, можна слухати інтернет-радіо за допомогою Lounchcast, дізнаватися прогноз погоди по усьому світі на Yahoo Weather і навіть створити свою сторінку на Yahoo360, де можна публікувати свої особисті мемуари, фотографії або розповідати про себе. Варто відзначити й

широкі можливості персоніфікації Yahoo Messenger – можна встановити різні теми, поміняти тло й т.д. Програма укомплектована додатковими можливостями, такими як відеодзвінок, відеоконференція, одержання й відсилання файлів, мережні ігри, послуги електронної пошти й SMS.

Yahoo Messenger дозволяє без проблем дзвонити на інший комп'ютер або передавати повідомлення іншим способом. Хіба що при контактах з користувачами Windows Live Messenger повідомлення іноді відсилаються досить довго, а в контактному аркуші присутній онлайн користувач відображався, як відсутній. Радує те, що при рівних можливостях Yahoo займає набагато менше ресурсів, чим Live Messenger або Skype.

Виводи: Yahoo Messenger – багатофункціональна програма, що споживає набагато менше ресурсів, ніж аналогічні продукти конкурентів. Однак досить неприємна необхідність установки зовсім непотрібних програм і неймовірно тривалий процес інсталяції.

Google Talk

Google Talk – пропріетарна програма миттєвого обміну повідомленнями із закритим вихідним кодом, розроблена компанією Google.

Google Talk дозволяє спілкуватися за допомогою голосового чату й текстових повідомлень. Особливістю Google Talk є тісна інтеграція з поштовою службою Gmail (наприклад, по Google Talk приходять повідомлення про нові повідомлення). Для використання Google Talk обов'язкова наявність облікового запису Gmail.

Як клієнт можливий використання сторонніх додатків таких, як Psi, Miranda IM, iChat і інших. Користувачі Google Talk можуть спілкуватися з користувачами інших XMPP-серверів відповідно до загальної архітектури протоколу XMPP.

Як заявляють власники компанії Google, ціль їхньої роботи це організація всієї світової інформації для того, щоб зробити її більше доступній і корисної кожній людині. Однак, дивлячись на те, якими кроками розвивається ця

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

компанія, здається, що щирі її мети куди більше масштабні, Google скуповує багато популярних мережних сервісів, створює свої різні служби й додатки, не боїться вступати в конфронтації з такими гігантами як Microsoft і навіть запускає на космічну орбіту власні супутники.

Один з успішних проектів компанії в області комунікації це месенджер за назвою Google Talk. Він дає можливість спілкуватися людям як за допомогою коротких текстових повідомлень, так і встановлюючи голосовий зв'язок. Дана програма працює через протокол Jabber/XMPP, не самий популярний у Росії, але досить солідний приріст, що показує останнім часом, користувачів. Тому що технологія Jabber відкрита, можна використовувати для обміну повідомленнями й клієнт Google Talk, і будь-який сторонній додаток, що працює на Jabber.

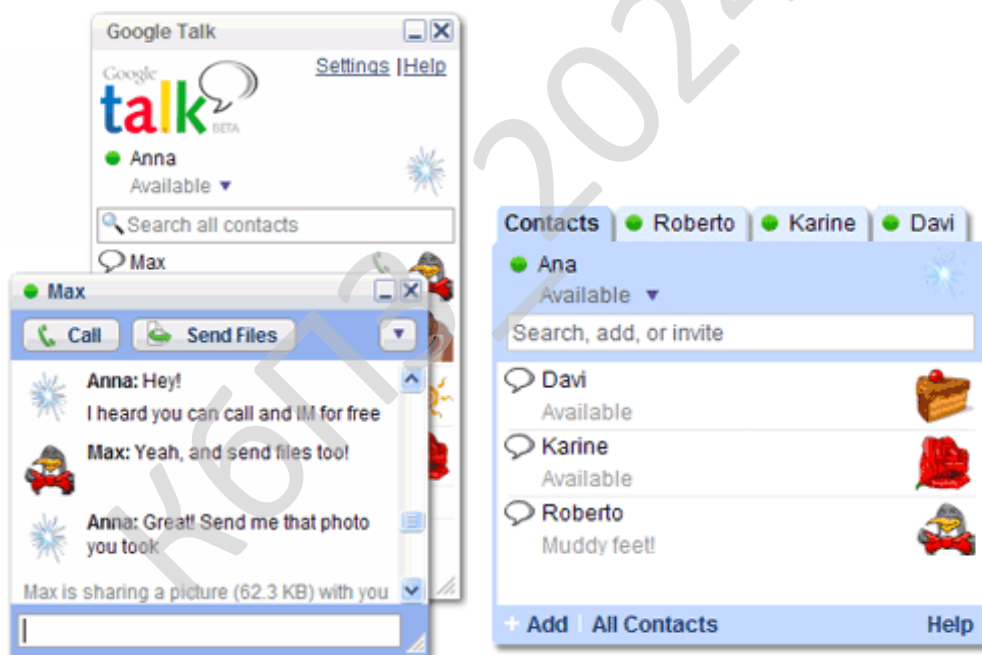


Рисунок 2.4 – Інтерфейс користувача Google Talk

Google Talk тісно пов'язаний з іншим популярним сервісів від компанії Google, з поштовим клієнтом Gmail. При реєстрації акаунта в Gmail заводиться й обліковий запис в Google Talk, причому в останній включена функція автоматичної перевірки поштової скриньки. Крім цього, реалізована можливість

синхронізації контактів в Google Talk з адресною книгою в GMail. Також з особливостей хочеться відзначити зберігання історії переписки на віддаленому сервері, чого так не вистачає користувачам ICQ. Адже іноді при некоректній переустановці ICQ-клієнта ми можемо втратити настільки коштовні «щоденники», та й, користуючись для доступу в мережу різними комп'ютерами, хочеться завжди мати під рукою повну історію переписки, а не тільки її окремі клаптики.

Цікава й дуже зручна реалізація роботи клієнта Google Talk. Вам не потрібно завантажувати і встановлювати дистрибутив програми, досить лише зайти на сторінку Google і натиснути кнопку «Установити». Клієнт працює як звичайне Flash-додаток, приліплене до будь-якої сторінки браузера. Незвичайно, але дуже зручно, особливо тим, хто не любить встановлювати на свій комп'ютер «зайві» програми й запускати непотрібні процеси. До того ж, можливо розмістити віконце додатка в будь-якій частині сторінки, наприклад, аж унизу свого блогу. У цьому випадку люди зможуть спілкуватися прямо звідти. Одним з головних переваг Google Talk є можливість безкоштовного голосового спілкування з кожним з контактів з вашої адресної книги, також можна дзвонити за гроші на звичайні телефони, як і в інших програмах, що забезпечують з'єднання за допомогою VoIP.

Програма Google Talk абсолютно безкоштовна, вона коректно працює на системах Windows старше версії 2000, з'єднання з інтернетом необхідно хоча б на швидкості 56 Кбіт/с. Шанувальники Mac OS і Linux поки обділені розроблявачами від Google, але в них досить інших додатків, які працюють по тій же протоколі, що й Google Talk. Підтримує «інтернет-звонилку» від Google функцію голосової пошти, тобто вхідний дзвінок при вашій відсутності зберігається в поштової скриньці, про що неодмінно відправляється повідомлення. Загалом, всі як у пристойних будинках, необхідні функції для голосового зв'язку присутні, однак є й недоліки, характерні для всіх подібних сервісів, такі як проблеми з якістю переданого звуку.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Як уже було сказано, Google Talk працює разом з поштовиком GMail, тому якщо у вас немає зареєстрованого аккаунта, то прийдеться його завести. Однак цей не весь сервіси, з якими може синхронно працювати програма. Як відомо, компанії Google належить площадка для розміщення відеороликів YouTube, і за допомогою нашого месенджера можна переглядати ролики безпосередньо в його діалоговому вікні. Крім того, Google Talk має можливість перегляду слайдшоу з веб-сервісів Flickr і Picasa.

Судячи з активної політики завоювання різних ринків в інтернеті, месенджер від Google буде й далі нарощувати потужності й одержувати всі нових користувачів. Це й не дивно, коли функції програми спрямовані, насамперед, на задоволення потреб користувачів, і розроблювачі не бояться освоювати нові цікаві додатки, то успіх напевно буде гарантований.

Дослідивши найбільш відомі програми обміну повідомленнями й мультимедійними даними зробимо наступні висновки. Типовий набір характеристик месенджеру містить у собі:

1. Індикатор статусу.
2. Список контактів.
3. Чат.
4. Історія розмов у чаті (Log).
5. Інтегруємість із іншими ІМ-мережами (Int).
6. Організація чат-конференцій (дискусія відразу з кількома людьми).
7. ІР-телефонія (VoIP).
8. Підтримка передачі відео за допомогою веб-камери (Video).
9. Дзвінки на звичайні телефони (PSTN).
10. Передача файлів (File).
11. Відправлення SMS (SMS).
12. Трансляція картинок або робочого стола.
13. Підтримка захищеного з'єднання.

А тепер розглянемо відповідність месенджери мережним службам.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Таблиця 2.1 – Найбільш популярні месенджери

	Web	VoIP	PSTN	Video	SMS	Log	File
Yahoo Messenger	✓	✓		✓	✓	✓	✓
Windows Live Messenger	✓	✓		✓	✓	✓	✓
AIM Pro		✓		✓			✓
Skype		✓	✓	✓	✓	✓	✓
Google Talk	✓	✓				✓	
ICQ		✓		✓	✓	✓	✓
Gizmo		✓	✓				
Apple iChat				✓		✓	✓
Qnext		✓		✓		✓	✓
Trillian		✓		✓		✓	✓
Gaim					✓		✓
Ineen		✓		✓			
Jabber		✓				✓	✓
Jabbin							✓
Meebo	✓						
KoolIM	✓						
AIM Triton	✓	✓	✓	✓	✓	✓	
Invite		✓		✓		✓	✓
WebMessenger		✓	✓			✓	

Провівши дослідження існуючого програмного забезпечення обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022, перейдемо до вибору мови програмування, на якій буде, у ході виконання бакалаврського проектування, реалізоване програмне забезпечення.

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 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.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– Тип даних Delphi «record» тепер підтримуть 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.

– 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 4k 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.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

– 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 швидкодією. 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 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 звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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 Snake.
- 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.

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 нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

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 TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- 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.

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.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

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 інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

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;

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.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Програмне забезпечення обміну інформацією в мережі під керуванням ОС Windows 11/Windows Server 2022 є централізованою службою миттєвого обміну інформацією, що використовує протокол OSCAR.

Чат і обмін миттєвими повідомленнями є ключовою частиною того, скільки програм сприяють прийняттю продукту, залученню та утриманню користувачів. Як власник продукту або розробник, ви майже розпещені вибором, коли справа доходить до рішення, як застосувати чат у вашому продукті.

У цьому розділі ми докладніше розглянемо протоколи, які забезпечують живлення чату. Ми почнемо з огляду ландшафту протоколів веб-чату, потім огляду восьми найпоширеніших протоколів, а потім – того, як розпочати роботу з миттєвими повідомленнями та чатом.

Протоколи обміну миттєвими повідомленнями та чату: огляд

Однією з труднощів перегляду протоколів обміну миттєвими повідомленнями та чату є те, що багато з них вирішують різні проблеми або різні частини проблеми. За своєю суттю протокол чату повинен передавати повідомлення між учасниками. Але є два виміри, які відокремлюють один протокол від іншого:

1. Які частини проблеми охоплює цей протокол.
2. Наскільки ретельно він розглядає ці частини проблеми.

Перше легко. Деякі протоколи мають на меті подбати про все та забезпечити наскрізне рішення, тоді як інші, як лазер, зосереджені на вирішенні лише одного аспекту.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Другий вимагає трохи більше уваги. Наприклад, IRC – це протокол, який, здається, охоплює все, але йому також бракує багатьох сучасних зручностей, на які очікують користувачі.

Один корисний спосіб подумати про це – порівняти з семирівневою мережевою моделлю OSI. Ось коротка інформація: для впровадження комп'ютерної мережі потрібні протоколи, які працюють на кожному рівні від оптоволокна чи міді фізичного рівня аж до прикладного рівня.

TCP і UDP, наприклад, є двома протоколами, які працюють на транспортному рівні, кожен із різними компромісами. TCP повільніший, але обіцяє, що дані надходитимуть у правильному порядку, тоді як UDP швидший і не дає таких гарантій. Протоколи чату та обміну миттєвими повідомленнями пропонують подібні компроміси залежно від вашого випадку використання.

Вісім протоколів обміну миттєвими повідомленнями та чату

Вибір правильного протоколу або протоколів є одним із перших архітектурних рішень, які ви приймете під час планування чату чи служби обміну миттєвими повідомленнями.

Розглядаючи протокол, корисно враховувати:

- **Як клієнти підключаються один до одного:** клієнт-сервер або одноранговий.
- **Типи даних, які він може підтримувати:** Чи підтримує він лише текст або потокове аудіо та відео.
- **Надійність і затримка:** заходи, які вживає протокол для забезпечення доставки, і як це впливає на швидкість обміну повідомленнями.
- **Масштабованість:** наскільки добре протокол справляється зі збільшенням потоків даних, клієнтів тощо.
- **Безпека:** чи має протокол вбудований захист, наприклад шифрування повідомлень під час передавання, чи вимагає використання інших інструментів для захисту.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Давайте розглянемо деякі з найпопулярніших протоколів для обміну миттєвими повідомленнями.

1. WebSocket

WebSocket – це керований подіями протокол потокової передачі даних із низькою затримкою, який ідеально підходить для потокової передачі даних у реальному часі. Це робить його придатним для чату та обміну миттєвими повідомленнями, оскільки він розроблений з нуля для підтримки постійних двосторонніх з'єднань між серверами та клієнтами. Однак варто зазначити, що WebSocket сам по собі не є протоколом чату. Натомість це протокол обміну повідомленнями в реальному часі, на основі якого ви можете створити чат.

Отже, як це формується?

Переваги WebSocket

– **Двонаправлений і повний дуплекс:** ідеально підходить для чату, повідомлення використовують одне з'єднання для передачі в обох напрямках і одночасно між клієнтом і сервером.

– **Ефективне використання ресурсів сервера:** постійні з'єднання WebSocket дозволяють уникнути накладних витрат на повторні цикли запитів/відповідей HTTP.

– **Низька затримка:** як протокол, керований подіями, WebSocket надсилає дані клієнту або серверу в момент, коли вони доступні. На відміну від HTTP, якому потрібні заголовки та інші метадані, після встановлення з'єднання WebSocket надсилає лише самі дані.

– **Широке розповсюдження:** усі основні браузерери та різноманітні бібліотеки мають вбудовану підтримку WebSocket, що полегшує використання для ваших кінцевих користувачів.

Недоліки WebSocket

– **Менш підходить для аудіо та відео:** хоча можна передавати аудіо та відео за допомогою з'єднання WebSocket, висока пропускну здатність цих типів даних робить їх кращими для протоколів, які встановлюють кілька одночасних

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

з'єднань. Подібним чином, хоча WebSocket пропонує гарний компроміс між затримкою та гарантіями доставки, потокове відео та аудіо зазвичай можуть втратити деякі дані, а це означає, що найнижча затримка мультимедійних протоколів пропонує меншу затримку в обмін на менше гарантій того, що окремі пакети даних будуть досягти клієнта.

– **Немає автоматичного повторного підключення:** якщо з'єднання WebSocket розривається, вам потрібно буде відновити його вручну. Це означає написання коду, який відстежує підключення та ініціює повторне підключення, якщо це необхідно.

– **Дещо важче горизонтально масштабувати :** WebSockets зберігають стан з'єднання, що може ускладнити їх використання у великих системах із кількома серверами WebSocket. Підтримка стану з'єднання потребує синхронізації даних кожного з'єднання на всіх серверах, які потенційно можуть обслуговувати це з'єднання.

2. WebRTC

У той час як WebSocket побудовано на моделі клієнт-сервер, WebRTC покладається на однорангові з'єднання. Це може зробити його кращим для робочих навантажень, де введення сервера було б непотрібним. Наприклад, передача файлів або спеціальний відеодзвінок між двома людьми, які вже підключені в іншій формі, як-от спільна участь в онлайн-чаті.

Які переваги та недоліки WebRTC?

Переваги WebRTC

– **Добре підходить для відео- та аудіоконтенту:** на відміну від WebSocket, WebRTC розроблено спеціально для потокового мультимедіа.

– **Підвищена стійкість у деяких ситуаціях:** у моделі клієнт-сервер проблема на сервері може порушити передачу даних кільком клієнтам. Оскільки підключення WebRTC є рівноправними, немає центральної точки, де збій міг би вплинути на кілька потоків даних однаково. Однак багато служб WebRTC покладаються на централізовану серверну частину для каталогів та інших служб.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– **Полегшена інфраструктура:** без необхідності розробляти, розгортати та підтримувати центральний сервер, можна легше створювати програми, які залежать від WebRTC.

– **Адаптивний підхід до затримки:** WebRTC може перемикатися між TCP (для надійної доставки) та UDP (для менших затримок і більших одночасних обсягів даних) залежно від того, що важливіше в цей момент.

Недоліки WebRTC

– **Створено для видалення даних у разі потреби.** Зворотною стороною меншої затримки є те, що WebRTC більше турбується про швидкість даних, ніж про цілісність даних. Це добре для потокового відео та аудіо, але менш підходить для тексту чи даних JSON.

3. MQTT

MQTT – це легкий протокол, призначений насамперед для міжмашинного зв'язку, особливо в ситуаціях, коли низька пропускну здатність і низька обчислювальна потужність. Клієнти підписуються на канали, якими керує центральний брокер повідомлень. Коли клієнт публікує повідомлення, він надсилає його брокеру, який потім розповсюджує його передплатникам.

Як правило, MQTT можна знайти в додатках Інтернету речей (IoT), таких як надсилання інформації про нові тарифи на розумні лічильники комунальних послуг.

Однак ті самі характеристики, які підходять MQTT для IoT, також підходять для чату та обміну миттєвими повідомленнями. Facebook Messenger – це одна з платформ чату, яка використовує MQTT, особливо тому, що вона відносно невелике навантаження на пропуску здатність і батарею пристрою.

Але чи це правильний протокол для вашої програми чату?

Переваги MQTT

– **Легкість:** MQTT має дуже мало накладних витрат, що означає, що він може працювати в середовищах з обмеженими ресурсами, наприклад у стільникових мережах із незначним покриттям.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

– **Надійна впорядкована доставка:** з трьома рівнями якості обслуговування (QoS) MQTT пропонує максимально один, принаймні один і максимальний режим доставки, щоб ви могли адаптуватися до складних умов мережі.

– **Контроль потоку:** MQTT допомагає запобігти перевантаженню та перевантаженню центрального брокера повідомлень, обмежуючи кількість повідомлень, які може надіслати кожен клієнт.

Недоліки MQTT

– **Посередник повідомлень обмежує масштабування:** використання центрального посередника повідомлень ускладнює масштабованість MQTT, а також створює єдину точку відмови.

– **Дещо важче розробляти:** як протокол, що використовує пропускну здатність, MQTT обмежує зручності для розробників. Наприклад, існує п'ять стандартних повідомлень про помилку з'єднання, які можуть ускладнити налагодження, якщо помилка виходить за рамки, передбачені розробниками протоколу.

– **Непридатний для відео та аудіо:** MQTT не оптимізований для потокового передавання аудіо чи відео, оскільки створено з урахуванням випадків використання низької пропускну здатності.

– **За замовчуванням не захищено:** облікові дані передаються у вигляді звичайного тексту, тому вам знадобиться вторинний механізм безпеки, наприклад SSL, поверх MQTT.

4. XMPP

Зародившись у 1990-х роках, XMPP – це протокол на основі XML, розроблений для забезпечення інструментів обміну миттєвими повідомленнями з відкритим кодом. Сьогодні це стандарт IETF, а зменшена версія під назвою FunXMPP є протоколом, який лежить в основі WhatsApp.

XMPP використовує децентралізовану архітектуру клієнт-сервер. Кожен сервер XMPP може підключатися до інших на одноранговій основі, підвищуючи

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

стійкість і масштабованість. Хоча, можливо, незвично, XMPP не вказує транспортний рівень.

Які переваги та недоліки XMPP як спеціального протоколу обміну миттєвими повідомленнями?

Переваги XMPP

– **Відмовостійкість і масштабованість.** Завдяки децентралізованій архітектурі простіше масштабувати додатки для чату та обміну миттєвими повідомленнями на основі XMPP, а також захищені від збоїв окремих серверів.

– **Гнучкість транспортування:** XMPP не залежить від транспорту, тобто він може обходити брандмауери та інші обмеження. Як правило, XMPP працює на TCP, але також може використовувати HTTP, WebSocket та інші механізми доставки.

– **Вбудована безпека:** протокол пропонує кілька способів захисту зв'язку, наприклад шифрування з'єднань за допомогою безпеки транспортного рівня (TLS) і додаткового підпису повідомлень.

Недоліки XMPP

– **Складність:** гнучкість XMPP, наприклад у тому, як транспортуються дані, і його розширюваність означають, що вбудовування його у вашу програму чату може вимагати більше зусиль для розробки.

– **На основі XML:** XMPP використовує XML для структурування даних. Більш сучасні протоколи чату засновані на JSON, що робить їх потенційно кращими для існуючих інструментів.

– **Менш ефективний:** XMPP є відносно важким протоколом, що означає, що одні й ті самі дані можуть займати більшу пропускну здатність і страждати від більших затримок, ніж такі протоколи, як WebSocket.

5. AMQP

Розширений протокол черги повідомлень (AMQP) – це протокол обміну повідомленнями, який базується на черзі публікації та підписки. Він

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

реальному часі. Використовуючи мости до інших сервісів, Matrix також може взаємодіяти з іншими інструментами, такими як Slack, Skype і WeChat.

Інструмент чату з відкритим кодом Rocket.chat використовує Matrix. Але чи це хороший вибір для програми чату?

Переваги Matrix

– **Децентралізація та федерація:** Matrix використовує досить радикальний підхід до масштабованості та стійкості. Замість того, щоб покладатися на єдиний центральний сервер або кластер, вони створюють мережу об'єднаних екземплярів. Це добре для масштабованості та довговічності ширшої мережі, але це має певні компроміси, як ми побачимо за мить.

– **Розширюваність:** Matrix може підключатися до інших мереж і інструментів за допомогою своєї концепції мосту.

Недоліки Matrix

– **Важче масштабувати:** хоча децентралізація може сприяти зростанню великої мережі, вона також створює проблеми з масштабуванням. Однорангова федерація ускладнює розповсюдження повідомлень у мережі, особливо в міру того, як зростає кількість одночасних користувачів. Це може ускладнити доставку повідомлень у режимі реального часу програмам чату на основі Matrix.

– **Відсутність стандартизації:** з кількома незалежними реалізаціями клієнта Matrix існує ймовірність неузгодженості в тому, як працює протокол від одного до іншого. Крива навчання: для кінцевих користувачів об'єднання ускладнює навчання користування системою. Наприклад, кімнати для чатів і користувачі розміщені на кількох серверах, що ускладнює виявлення.

7. WebSocket

До WebSocket серверні події були ранньою відповіддю на проблему відсутності стану мережі. У той час оновлення інформації на веб-сторінці вимагало б оновлення цієї сторінки або використання такої техніки, як довге опитування, обидва з яких є дещо неефективними. Події, надіслані сервером, розташовані поверх HTTP, щоб відкрити постійне одностороннє з'єднання між

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

клієнтом (зазвичай веб-браузером) і сервером. Коли сервер має нові дані для клієнта, він надсилає їх через з'єднання.

Як односторонній протокол, Server Sent Events не дуже підходить для програм чату та обміну миттєвими повідомленнями. Але розглянемо більш детально переваги і недоліки.

Переваги WebSocket

– **На основі HTTP:** Завдяки тому, що на основі HTTP, брандмауерами буде менш імовірно, що з'єднання, надіслані сервером, будуть заблоковані. Це також спрощує впровадження, оскільки базова інфраструктура як на клієнті, так і на сервері вже існує.

– **Зосереджено на одній справі:** Server Sent Events є хорошим вибором, якщо вам потрібен простий, легкий протокол, який надсилатиме лише відносно прості дані із сервера на клієнт.

– **Підтримує повторне з'єднання:** якщо з'єднання розривається, сервер надісланих подій автоматично відновлює його. Це економить зусилля на розробку порівняно з WebSocket, де потрібно відстежувати роз'єднання та повторно підключатися вручну.

Недоліки WebSocket

– **Односторонній зв'язок.** Перешкодою для додатків чату є те, що серверні події є одностороннім протоколом. Він може підходити лише для широкомовних повідомлень, але немає можливості відповісти за допомогою самих подій, надісланих сервером.

– **Лише текст:** Події, надіслані сервером, підтримують лише повідомлення UTF-8, тобто неможливо надсилати розширений вміст, наприклад зображення чи аудіо.

8. IRC

Починаючи з кінця 1980-х років, IRC зіграв важливу роль в еволюції спільнот програмного забезпечення з відкритим кодом, кожна з мереж присвячена різним темам або місцевостям. Однак IRC мало змінився протягом десятиліть і

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

виглядає застарілим у порівнянні з сучасними альтернативами, такими як WebSocket, XMPP і Matrix.

Переваги IRC

– **Перевірено** : IRC використовується щодня більше трьох десятиліть, а це означає, що багато помилок було виправлено.

– **Широке поширення**: незважаючи на те, що клієнти IRC уже не такі популярні, як колись, вони доступні майже для кожної платформи, навіть включаючи 8-розрядні домашні комп'ютери, такі як Sinclair ZX Spectrum.

– **Простий**: IRC зосереджений на створенні текстового чату в реальному часі, тобто сам протокол відносно простий.

Недоліки IRC

– **Ефемерність**: щоб отримувати повідомлення, окремий користувач має бути підключений до відповідного сервера IRC. Будь-які повідомлення, надіслані під час від'єднання, фактично втрачаються.

– **Лише текст**: IRC не підтримує зображення, спільний доступ до файлів чи інші мультимедійні файли. Відсутність безпеки: IRC походить із того часу, коли Інтернет використовувався здебільшого як інструмент для наукових кіл. Таким чином, він має мінімальні функції безпеки.

Як вибрати правильний протокол чату для вашого бізнесу

З такої різноманітності, як вибрати правильний протокол чату для ваших конкретних бізнес-потреб? Ось п'ять сфер, які вам потрібно розглянути.

Де це знаходиться в архітектурі

Кожен із розглянутих нами протоколів фокусується на певній області проблемного простору, а потім використовує власний підхід до вирішення. Наприклад, IRC – це комплексне рішення для спілкування в чаті, хоча його й не вистачає за сучасними стандартами. З іншого боку, WebSocket забезпечує базову інфраструктуру для створення власної програми чату.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Масштабованість

Розробка архітектури для масштабованості підготує вашу програму чату до зростання. Протоколи, які ми розглянули, поділяються на три великі категорії масштабування:

– **Обмежена масштабованість:** деякі аспекти протоколу істотно ускладнюють масштабування. Наприклад, може бути складніше масштабувати централізований брокер повідомлень MQTT.

– **Горизонтальне масштабування:** додавання додаткових екземплярів сервера для роботи один з одним дає змогу збільшити пропускну здатність вашої чат-платформи. XMPP і WebSocket дозволяють масштабувати таким чином.

– **Інтегрований:** окремі екземпляри протоколу чату існують на окремих серверах, але вони спілкуються один з одним для обміну повідомленнями. Ось як працює Matrix.

Надійність

Гарантії доставки повідомлень, відмовостійкість і відновлення після помилок значно відрізняються від одного протоколу до іншого. Наприклад, MQTT розроблено для роботи з обмеженими мережевими умовами, що дає вам можливість контролювати, який рівень гарантій доставки повідомлень у порівнянні з додатковими мережевими витратами є відповідним. Підвищена складність об'єднаного протоколу, такого як Matrix, може ускладнити створення надійного рішення.

Безпека та конфіденційність

Деякі протоколи мають кілька рівнів безпеки та конфіденційності, тоді як інші, наприклад IRC, практично не мають. Перше запитання: «Що обробляє сам протокол?» Ось деякі з факторів безпеки та конфіденційності, які вам потрібно враховувати:

– **Шифрування з'єднання:** чи протокол шифрує з'єднання між клієнтом і сервером (або вузлами), як у випадку з AMQP? Або це залишилося на

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

– **Створення повністю власними силами:** створити власне рішення може бути спокусливо. Якщо ви відповідаєте за все, то здається, що ви можете побудувати саме те, що вам потрібно. Досвід багатьох інженерних команд свідчить про те, що власне будівництво призводить до більших компромісів, оскільки більше часу потрібно витратити на створення та підтримку базових технологій, а не на інновації. Подібним чином довгострокові витрати можуть швидко зрости, оскільки технічне обслуговування, заходи безпеки та розробка функцій потребують обмежених ресурсів.

– **Використовуйте платформу як послугу (PaaS):** Замість того, щоб створювати все всередині компанії, використання PaaS у реальному часі, наприклад Ably, дає вам інструменти та інфраструктуру для створення власного рішення, але без додаткового тягаря створення та підтримки кожної частини технічний стек. Це повертає вам інженерні ресурси для інновацій там, де це важливо.

– **Використовуйте систему чату з відкритим кодом:** існують сервери та клієнти з відкритим кодом для таких протоколів, як IRC і Matrix, які ви можете розгорнути та налаштувати відповідно до своїх потреб. Однак ви залишаєтесь пов'язаними планом розвитку функцій, інженерною майстерністю та чуйністю у сфері безпеки спільнот розробників, які стоять за цими проектами. Залежно від ваших потреб, налаштування може бути дорогим або навіть зовсім неекономічним.

– **Використовуйте чат як послугу:** аналогічно існують комерційні служби, які надають послугу чату білої мітки. Параметри налаштування відрізняються, але, принаймні, ви можете додати власний бренд і певною мірою інтегруватись із існуючими службами. Незважаючи на те, що це один із способів швидко налагодити роботу, один недолік полягає в тому, що ви зобов'язані дотримуватися плану ціноутворення та функцій обраного вами постачальника.

Який би варіант ви не вибрали, вибір, який ви зробите, і основний(і) протокол(и), які використовують ваш чат або обмін миттєвими повідомленнями,

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

матимуть суттєвий вплив на роботу кінцевого користувача, вашу здатність надавати нові функції та поточну вартість обслуговування.

3.2 Розробка структурної схеми

Структурна схема розробленого програмного забезпечення обміну інформацією в мережі під керуванням ОС Windows 11/Windows Server 2022 зображена на рисунку 3.1. У основі програмного забезпечення лежить протокол OSCAR.

OSCAR (Open System for Communication in Realtime) – це протокол обміну миттєвими повідомленнями, розроблений AOL для AOL Instant Messenger (AIM) і використовується після придбання ICQ для обох Messenger. Однак є кілька безкоштовних клієнтів ICQ, які нормально працювали до публікації протоколу.

Протокол довго не публікувався, навіть якщо «відкритий» припускав протилежне в назві. Тому вся документація не була специфікацією, а була лише створена шляхом зворотного проектування. З представленням AOL платформи розробників AIM Open AIM 2.0 у березні 2008 року було розкрито протокол OSCAR. Це призвело до меншої кількості проблем сумісності з багатопроколовими месенджерами, які часто траплялися під час зміни протоколу.

Позначення версій

Falsely OSCAR також називається ICQv7, v8, v9 і v10 відповідно. Це неправильно, оскільки оригінальний протокол ICQ завжди використовує прямі з'єднання між клієнтами, однак OSCAR побудований з нуля по-іншому та покладається на з'єднання через сервер. Крім того, версії 7-10 не відрізняються протоколом, а лише послугами, які пропонує сервер, який підтримується дуже ліберально OSCAR.

OSCAR працює над існуючим з'єднанням TCP і сам складається з двох рівнів протоколу.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Нижчий, називається протоколом прикладного рівня кадру (FLAP), заснований безпосередньо на TCP. Заголовок FLAP починається з шістнадцяткового значення 2A для виявлення пакета FLAP. За ним йде канал (див. нижче), порядковий номер для виявлення помилок і розмір вкладених даних.

Канал подібний до портів для TCP і UDP, однак завжди призначений для певної операції. Можливі канали:

– Якщо використовується канал, відмінний від 2, зазвичай монтується дані у форматі Тип – Довжина-Значення (TLV).

– На каналі 2 слідує пакет типу Simple Network Atomic Communication (SNAC), друга основна частина протоколу OSCAR. Типи пакетів SNAC організовані в так звані Food Group. Кожна група продуктів має ідентифікаційний номер і організована в інший ідентифікатор типу повідомлення. Як один із прикладів, Food Group 4, тип 20 для повідомлення, клієнт інформує іншого, у якому його користувач щойно ввів текст або зупинив його. В принципі, інформуйте цей тип повідомлення через інші процеси, але визначає лише два. Тип 1 для кожної Food Group представляє окреме повідомлення про помилку

Після входу сервер повинен вказати, які групи продуктів він підтримує; тільки вони можуть / повинні використовувати клієнт. Після вказівки слід надсилання іншого з'єднання було перервано сервером. Food Group в принципі можуть бути розширені програмістами.

Подається задокументовано на відміну від специфікації, Food Group 5, 7, 8, 10-15, 21 і 23

Наведені нижче клієнти підтримують протокол OSCAR або старий протокол ICQ. Деякі з них є багатопрокольніми клієнтами, які підтримують лише ICQ, наприклад AIM, IRC, XMPP, MSN, Yahoo! і більше.

Використання альтернативних клієнтів часто прямо забороняється в Умовах Постачальника послуг, наприклад в Умовах використання ICQ або AIM.

Розглянувши протокол перейдемо до структурної схеми.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

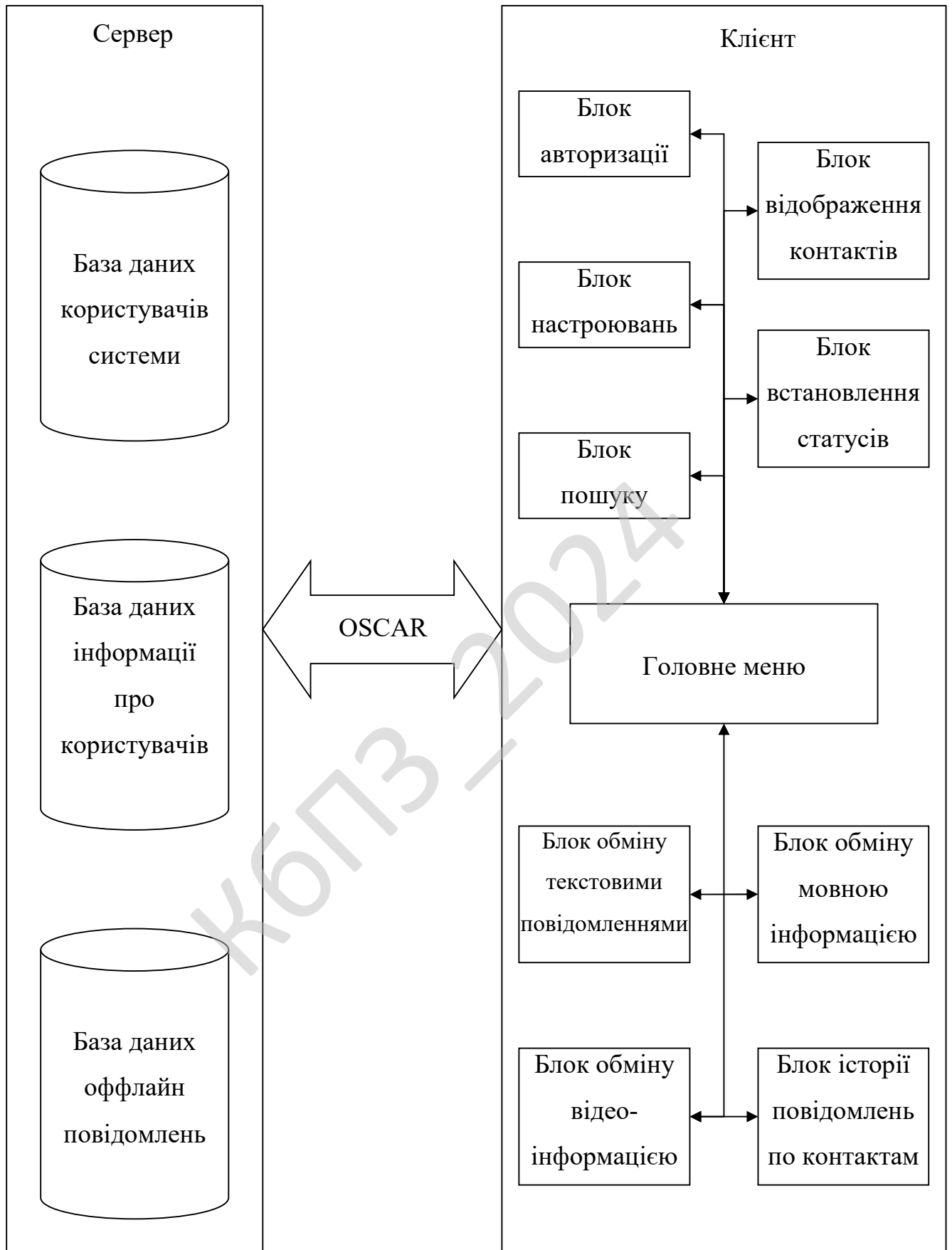


Рисунок 3.1 – Структурна схема системи

Розроблене програмне забезпечення складається з наступних блоків:

Серверна частина:

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

Клієнтська частина:

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

3.3 Розробка функціональної схеми

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

1. Головне меню:

- Пристикувати список контактів.
- Приватні списки.
- Показати/змінити мої дані.
- Налаштування.
- Повідомлення анти-спам бота.
- Сервісні повідомлення.
- Мастер імпорту.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

– Перевірити наявність оновлень.

– Про програму.

– Додати обліковий запис.

– Змінити профіль.

– Закрити програму.

2. Блок авторизації:

– Введення логіну.

– Введення паролю.

3. Блок налаштувань:

– Загальні.

– Список контактів.

– Події.

– Повідомлення.

– Історія.

– Статуси.

– Анти-спам.

– Звуки.

– Інтерфейс.

– Гарячі клавіші.

– Модулі.

– Підключення.

– Відео та звук.

– Облікові записи.

4. Блок обміну текстовими повідомленнями:

– Вікно користувачів.

– Вікно обміну повідомленнями.

– Вікно набору нового повідомлення.

– Вікно відображення даних про користувача.

– Вікно відображення аватарки.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

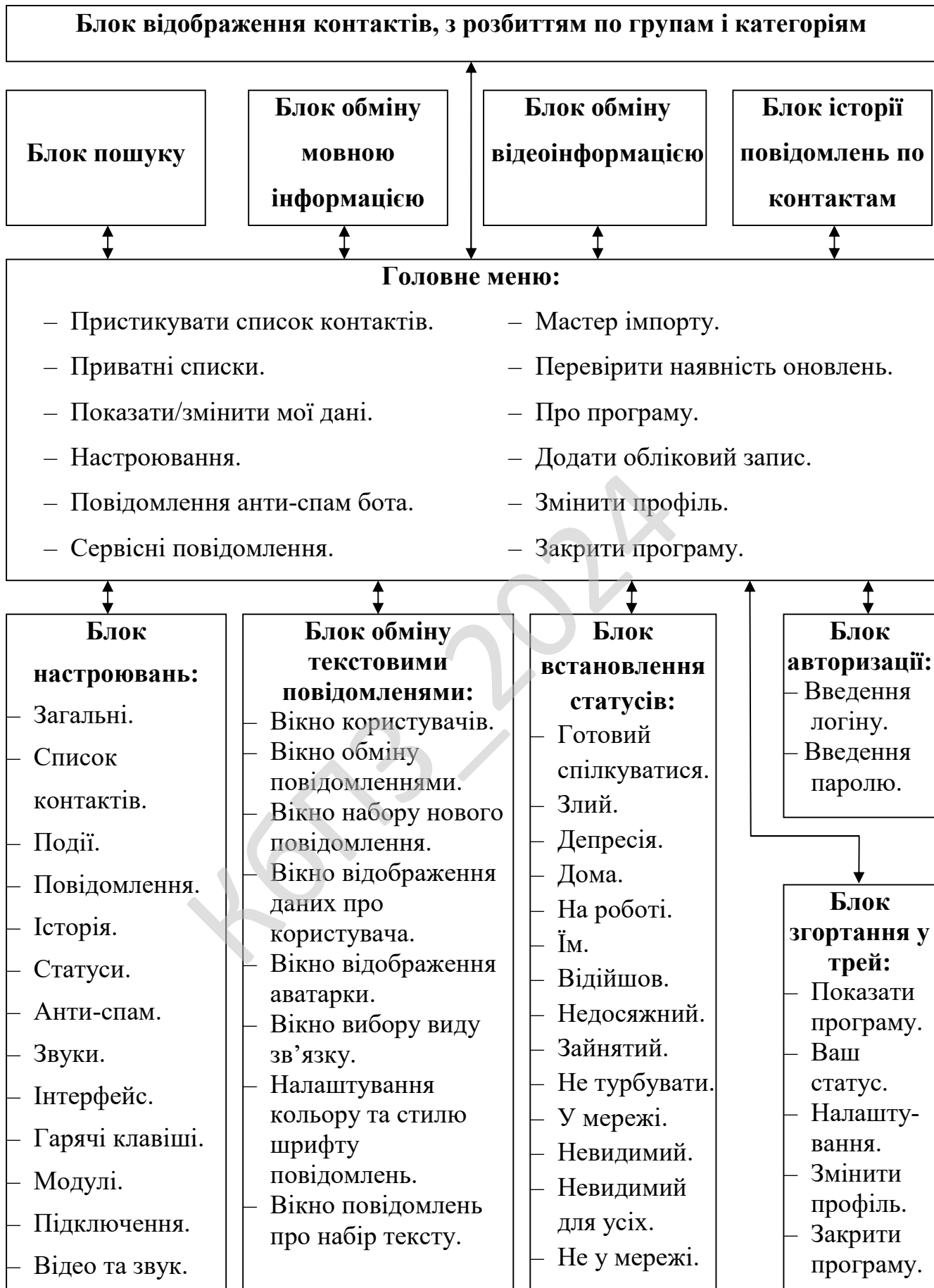


Рисунок 3.2 – Функціональна схема системи

- Вікно вибору виду зв'язку.
- Налаштування кольору та стилю шрифту повідомлень.
- Вікно повідомлень про набір тексту.

5. 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.

6. Блок згортання у трей:

- Показати програму.
- Ваш статус.
- Налаштування.
- Змінити профіль.
- Закрити програму.

7. Блок відображення контактів, з розбиттям по групам і категоріям.

8. Блок пошуку.

9. Блок обміну мовною інформацією.

10. Блок обміну відеоінформацією.

11. Блок історії повідомлень по контактам.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3.

Першими процесами, які запускаються у системі є наступні:

- Процес авторизації.
- Процес реєстрації.

Процес реєстрації взаємодіє з процесом авторизації. Останній процес взаємодіє з процесом підключення до сервера.

Процес підключення до сервера взаємодіє з процесом виведення основного вікна програми.

Процес виведення основного вікна програми взаємодіє з наступними процесами:

- Процес зміни статусу.
- Процес вибору контакту.
- Процес виведення списку контактів.
- Процес зміни параметрів програми.
- Процес відключення від сервера.

Процес зміни параметрів програми взаємодіє з процесом додавання/видалення нових контактів, який у свою чергу, взаємодіє з процесом пошуку контактів.

Процес вибору контакту взаємодіє з процесом відкриття вікна діалогу.

Останній процес взаємодіє з наступними процесами:

- Процес запиту зв'язку.
- Процес обміну текстовими повідомленнями.
- Процес перегляду історії повідомлень.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

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 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків.

Спершу відбувається виведення вікна авторизації та реєстрації користувача, який входить у систему. Після цього користувач обирає, що йому проходити, авторизацію, якщо є обліковий запис, або реєстрацію, якщо облікового запису немає.

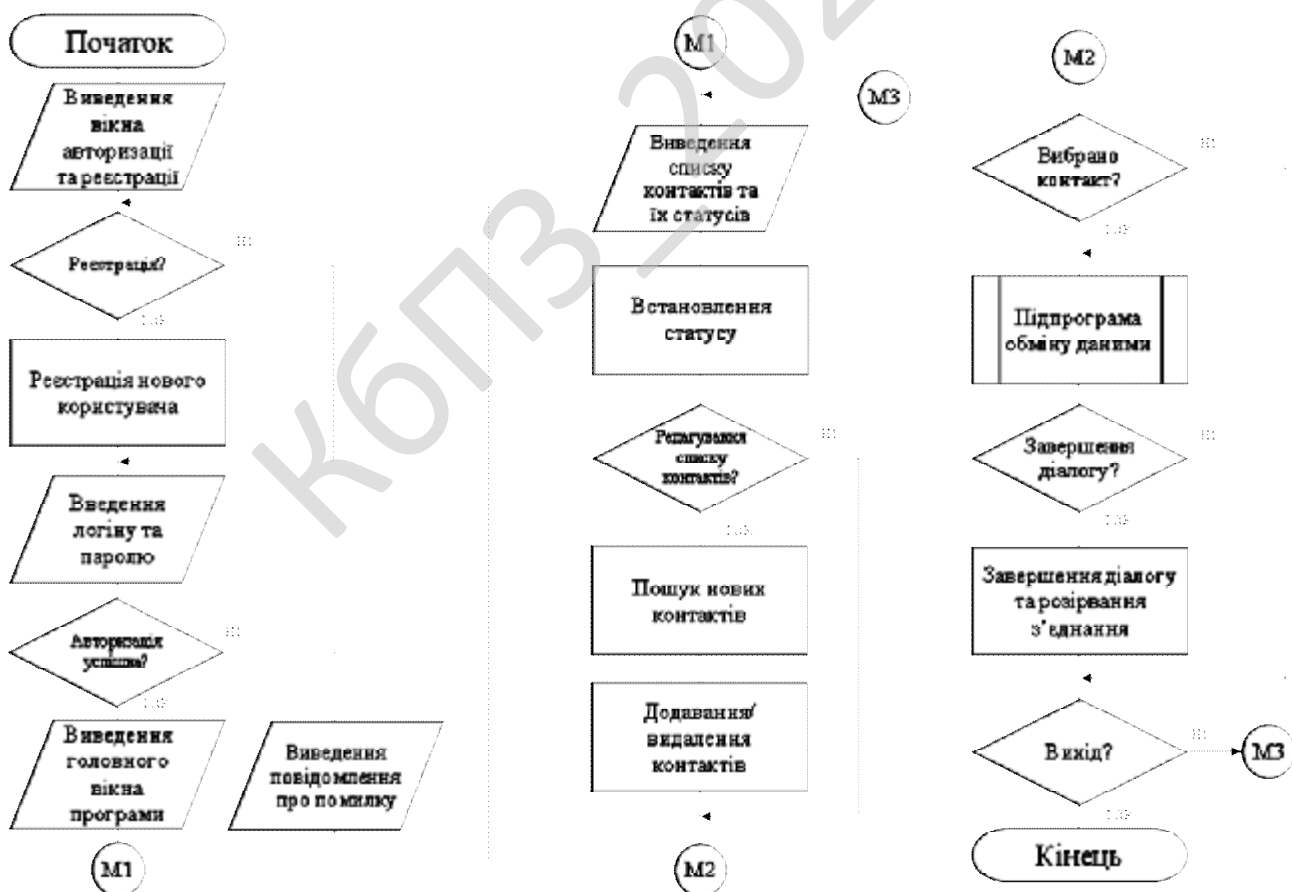


Рисунок 4.1 – Блок-схема основної програми

Якщо користувач обирає реєстрацію, тоді відбувається виконання наступних дій:

- Реєстрація нового користувача.
- Введення логіну та пароля користувача.

Після цього користувач проходить процедуру авторизації. Якщо він не пройшов цю процедуру, тоді відбувається виведення вікна про помилку, й програма переходить у початок.

Якщо ж авторизація пройдена успішно, тоді виводиться головне вікно програми.

Після цього виводиться список контактів та їх статусів, й відбувається встановлення статусу користувача.

Після цього користувач обирає, чи редагувати йому список контактів. Якщо редагувати, тоді виконуються наступні дії:

- Пошук нових контактів.
- Додавання/видалення контактів.

Якщо користувач обирає контакт, з яким він збирається спілкуватися, тоді викликається підпрограма обміну даними.

Після завершення діалогу відбувається розірвання з'єднання.

Після цього, користувач обирає, працювати йому далі з програмою, або ні.

На рисунку 4.2 зображено блок-схему підпрограми обміну даними.

З неї ми бачимо, що для реалізації обміну даними необхідно виконання наступних кроків.

Спершу відбувається виведення вікна діалогу. Після цього користувач обирає які дії він буде виконувати.

Якщо він обирає обмін текстовими повідомленнями, тоді він входить у відповідне вікно, й обмінюється текстовими повідомленнями.

При обирає перегляду історії повідомлень, користувач може переглянути усі діалоги з усіма користувачами.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

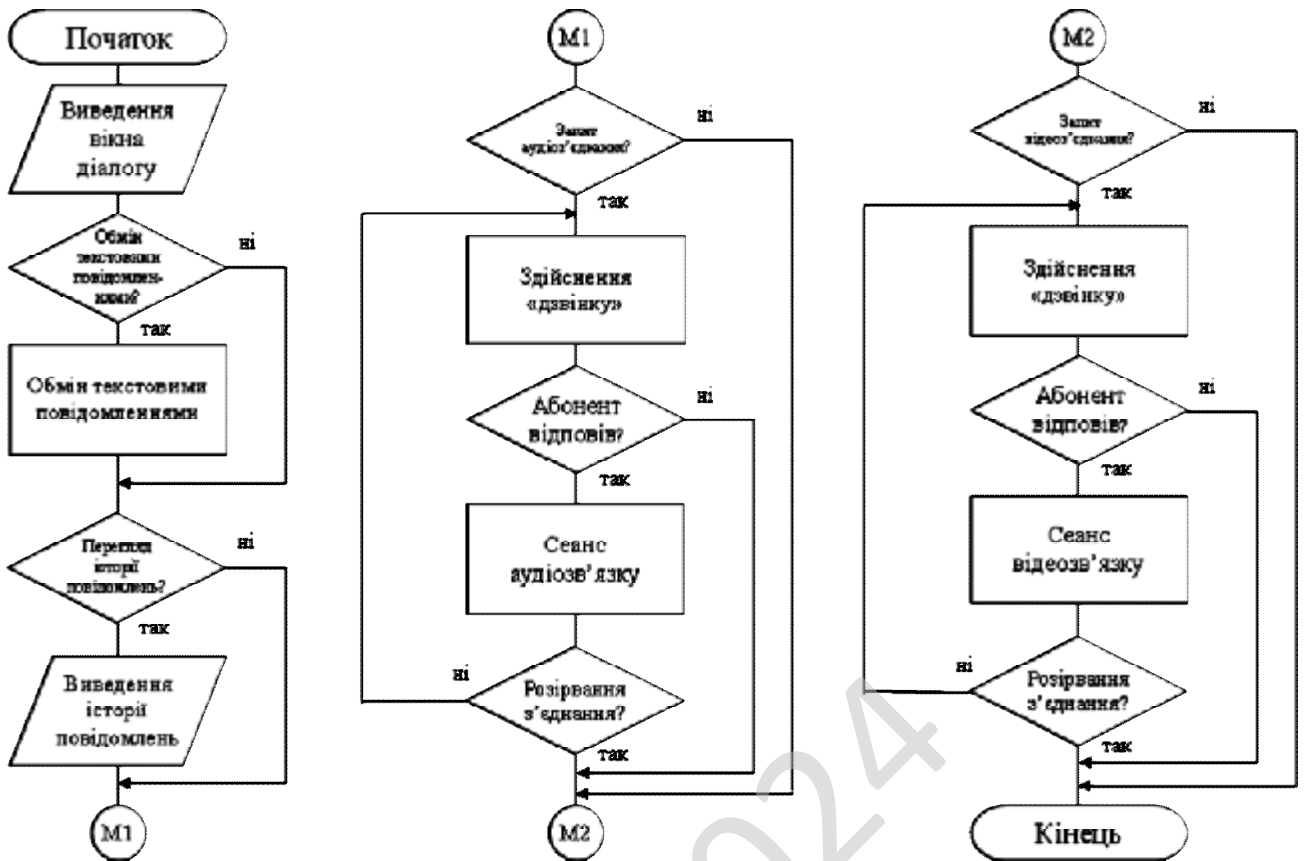


Рисунок 4.2 – Блок-схема підпрограми обміну даними

Якщо він посилає запит на аудіозв'язок тоді відбувається здійснення «дзвінку». При цьому, якщо абонент до якого він дзвонить, відповів, то відбувається сеанс розмови по мережі.

Якщо він посилає запит на відеозв'язок тоді відбувається здійснення «дзвінку». При цьому, якщо абонент до якого він дзвонить, відповів, то відбувається сеанс розмови по мережі, при цьому користувачі бачать один одного у вікні, куди передається зображення з веб-камери..

У кожного користувача є мережний месенджер. Усі користуються ним: хтось рідко, хтось завжди в онлайні. В основному з його допомогою обмінюються повідомленнями, набагато рідше пересилають файли й грають в ігри. І далеко не всі використовують його для чогось іншого. У даному розділі опишемо як на Delphi за допомогою компонента TICQclient написати бот. У програмах обміну інформацією, бот – комп'ютерна програма, що виконує різні

функції, і використовує для введення й виводу повідомлення, підтримувані протоколом даної програми обміну інформацією.

Боти бувають для протоколів IRC, сервісів миттєвого обміну повідомленнями й інших чатів.

Принцип роботи: бот приймає вхідні повідомлення, аналізує їх і відсилає результат виконання й/або виконує команду.

Основні функції таких ботів:

Службові:

– Ведення логів чату.

– Облік прав учасників. Наприклад, служба IRC ChanServ на IRC-серверах дозволяє зареєструвати канал, і знімає статус оператора із вхідних незареєстрованих користувачів, автоматично видаючи статуси тим, кому це дозволив, за допомогою спеціальних команд, творець каналу або уповноважені їм користувачі.

– Забезпечення безпеки чату. Наприклад, «викидає» користувачів, якщо вони повторюють те саме багато разів (захист від флуду), або по якихось параметрах (наприклад, ім'я складається з декількох заголовних букв, або являє собою нецензурне вираження) схожі на «небажаного» учасника.

– Забезпечення можливості конференції між більш, ніж двома користувачами, у протоколах без такої функції.

Інформаційно-розважальні:

– Довідкова – інтерфейс до бази даних, у якій зберігаються слова й відповіді на них («фактоїди»). Відповідей може бути декілька, тоді з них може вибиратися випадкова. Один з таких ботів – IRC-бот Infobot. Фактоїди можуть створюватися як користувачами прямо, так і автоматично, шляхом «підслуховування» їхніх розмов між собою (наприклад, з повідомлень із рядком «– це»). Окремий випадок – словник.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Для початку перелічу самі основні властивості, методи й події компонента, без яких нам ніяк не обійтися:

- властивість UIN – номер месенджера для бота;
- властивість Password – пароль від номера;
- властивість ConverToPlaintext визначає, чи потрібно перетворювати отримані RTF-повідомлення у звичайний текст; для того щоб можна було спілкуватися з ботом за допомогою офіційного клієнта ICQ, потрібно встановити значення true;
- метод Login – підключення до сервера;
- метод Disconnect – відключення від сервера;
- метод SendMessage(UIN, msg) – відправлення повідомлення msg на номер UIN;
- подія MessageRecv – викликається при одержанні повідомлення.

Напишемо бот-шлюз. Після включення по нашій команді він буде пересилати всі отримані від нас повідомлення на зазначений номер. Зрозуміло, те ж саме буде відбуватися з повідомленнями нашого співрозмовника. Фішка в тому, що він не побачить твого номера, а тільки номер бота. Отже, кидаємо на форму компонентів, встановлюємо властивості UIN і Password. Двічі клацаємо за формою й пишемо от такий оброблювач:

```
MasterUIN:='<номер «хазяїна»>';  
work:=false;  
ICQclient1.Login;
```

Піднімаємося ледве вище й під рядком «Form1: TForm1;» пишемо:

```
OpponentUIN, MasterUIN:string;  
work:boolean;
```

Це змінні для зберігання номера співрозмовника й хазяїна бота, а також статусу роботи. Тепер почнемо писати обробку вхідних повідомлень – подія OnMessageRecv. Спочатку – команда створення з'єднання:

Створення каналу зв'язку

```
if (pos('/start ',Msg)=1)and(UIN=MasterUIN) then begin  
    delete(Msg,1,length('/start '));  
    while pos(#10,Msg)<>0 do delete(Msg,pos(#10,Msg),1);
```

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

```

while pos(#13,Msg)<>0 do delete(Msg,pos(#13,Msg),1);
OpponentUIN:=Msg;
ICQclient1.SendMessage(StrToInt(MasterUIN),'Канал створений. ');
work:=true;
exit;
end;

```

Бот, одержавши від хазяїна команду виду «/start <номер>», збереже номер співрозмовника, відправить підтвердження хазяїнові й змінить статус роботи. Два цикли потрібні спеціально для офіційного клієнта, що любить наприкінці повідомлення вставляти переклад рядка. Ми його вирізуємо, щоб виділити номер.

Фрагмент обробки команди завершення з'єднання повністю аналогічний.

Розрив зв'язку

```

if (pos('/stop',Msg)=1) and (UIN=MasterUIN) then begin
    ICQclient1.SendMessage(StrToInt(MasterUIN),'Канал знищений. ');
    work:=false;
    exit;
end;

```

Ну, і останні рядки – пересилання повідомлень.

Пересилання

```

if not work then ICQclient1.SendMessage(StrToInt(UIN),'немає з'єднання.')
else
    if UIN=MasterUIN then
        ICQclient1.SendMessage(StrToInt(OpponentUIN),Msg)
    else ICQclient1.SendMessage(StrToInt(MasterUIN),Msg);

```

Отут спочатку робиться перевірка статусу (якщо з'єднання розірване, то відправляється повідомлення про помилку), потім визначається відправник повідомлення, що пересилається іншому співрозмовникові.

Записна книжка

Будинок, робота, інтернет-кафе...І скрізь – свій клієнт інтернет месенджера. А якщо потрібно зберегти важливе повідомлення? Записати на папірець? Є рішення краще: написати свого бота – усюди доступну записну книжку. Він буде зберігати отримані повідомлення й по команді показувати їх. Створюємо новий проект, розташовуємо компонент, виставляємо властивості UIN і Password.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

```
MasterUIN:='<номер «хазяїна»>';  
ICQclient1.Login;
```

Тепер будемо обробляти вхідні повідомлення. У події OnMessageRecv повідомляємо чотири змінні:

```
fs:TFileStream;  
i:integer;  
ch:char;  
send:string;
```

Це файловий потік, куди ми будемо зберігати замітки, лічильник для циклу, зчитаний з потоку символ і буфер для відправлення. Принцип роботи такий: одержуємо повідомлення тільки від хазяїна; якщо отримано повідомлення «/show», то виводимо всі замітки; інакше зберігаємо повідомлення. У ньому як роздільник заміток використовується символ з кодом 0. При записі ми додаємо символи перекладу рядка.

Код записної книжки

```
if UIN<>MasterUIN then exit;  
send:='';  
if FileExists('notes.txt') then  
    fs:=TFileStream.Create('notes.txt',fmOpenReadWrite)  
else  
    fs:=TFileStream.Create('notes.txt',fmCreate);  
if pos('/show',Msg)=1 then begin  
    fs.Seek(0,soFromBeginning);  
    for i:=1 to fs.Size do begin  
        fs.Read(ch,1);  
        if ch<>#0 then send:=send+ch  
        else begin  
            ICQclient1.SendMessage(StrToInt(MasterUIN),send);  
            send:='';  
        end;  
    end;  
end;  
else begin  
    fs.Seek(0,soFromEnd);  
    send:=DateTimeToStr(now)+#10#13+Msg+#10#13#0;  
    fs.Write(send[1],length(send));  
    ICQclient1.SendMessage(StrToInt(MasterUIN),'Запис збережений.');
```

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

створює тільки одну консоль і перед будь-якою операцією робить перевірку її існування, що рятує нас від зайвих проблем. Всі його методи дивися в коментарях у модулі uCmdPipe. Тепер нам потрібно відіслати результат команд назад. Є одна тонкість – ми ніколи не можемо бути впевнені в часі приходу даних у пайп. Тому ми будемо робити його в окремому потоці, не намагатися одержати відразу після відправлення команди. Отже, кидаємо на форму TTimer, встановлюємо інтервал поменше, а властивість Enable в true і пишемо оброблювач OnTimer:

```
if p.DataPresent then begin
    send:=p.ReadStringCmdPipe;
    ICQclient1.SendMessage(StrToInt(MasterUIN), send);
end;
```

Ми перевіряємо наявність даних у пайпі (і заодно сам пайпа) і, якщо вони є, зчитуємо й відсилаємо. У цьому ж оброблювачі потрібно оголосити змінну send типу string.

Альтернативи

У компонента TICQclient є одна незаперечна перевага – простота. Головний же його недолік полягає в тому, що він може не підтримуватися. У випадку зміни протоколу ніхто не гарантує випуск патча. Є альтернатива – використовувати бібліотеку IcqOscar. Це один з основних компонентів Miranda IM, доступний у вихідних кодах, і з документацією. Він регулярно обновляється. Можна реалізувати у своїй програмі автоматичне завантаження бібліотеки із сайту проекту. От тільки заголовні модулі написані на C++.

Іншим варіантом може бути використання бібліотеки, що розвивається зараз, ICQkid2 – зручної кроссплатформеної реалізації протоколу ICQ. Знову ж хедери написані на C++.

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7624:2014 («Калина»). При розробці національного стандарту ДСТУ 7624:2014

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

(блоковий шифр «Калина» і режими його роботи) було ухвалене рішення забезпечити прозорість проектування й використовувати консервативний підхід із застосуванням добре досліджених конструкцій, що забезпечують запас стійкості для безпечного застосування алгоритму в умовах істотного прогресу криптоаналітичних технік і засобів обробки даних.

Національний стандарт підтримує розмір блоку й довжину ключа шифрування 128, 256 і 512 біт (довжина ключа дорівнює розміру блоку або у два рази перевищує його), забезпечуючи нормальний, високий і надвисокий рівень стійкості (зараз це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі). Різні варіанти забезпечують гнучкість вибору параметрів для розроблювачів систем криптографічного захисту, що дозволяє одержати як найвищий рівень швидкодії, так і найбільший запас стійкості перетворення.

Високорівнева конструкція використовує добре досліджену Square-подібну SPN-структуру, застосовувану в алгоритмах AES/Rijndael, Whirlpool, «Стрибог», «Коник» і багатьох інших. Циклове перетворення побудоване на базі таблиць підстановки (S-блоків) і множення на МДР-матрицю над кінцевим полем, забезпечуючи необхідні криптографічні властивості. Застосування саме такої конструкції дозволяє забезпечити доказову стійкість до диференціального, лінійному й ін. видам криптоаналізу, одночасно забезпечуючи ефективну реалізацію на широкому спектрі програмних і програмно-апаратних платформ. При виборі розміру МДР-матриці був прийнятий в увагу розмір кешу L1 сучасних і перспективних процесорів, що дозволило оптимізувати швидкодія програмної реалізації шифру.

Альтернативний варіант, який розглядався при розробці циклової функції, – ARX перетворення (Addition-rotation-xor). Цей підхід реалізований у шифрі SPECK (розроблений Агентством національної безпеки США й переданий у відкритий доступ), у блокових алгоритмах, на основі яких побудовано сімейство геш-функцій SHA-0,1,2 і ін. Перевагою походу є компактність і

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

швидкодія перетворення. Але, у той же час, є й істотний недолік, пов'язаний з відсутністю методів, що дозволяють виконати строге аналітичне обґрунтування криптографічної стійкості таких розв'язків. Навіть із наймогутнішими у світі можливостями для аналізу, США кілька раз були змушено модифікувати свої стандарти гешування через знайдені уразливості: з 1993 по 1995 рр. діяв SHA-0, з 1995 по 2001 рр. застосовувався SHA-1, з тих пор використовується SHA-2. Через питання, що піднімаються, до стійкості й цієї версії, у США з 2008 по 2012 рр. був проведений міжнародний конкурс SHA-3. До теперішнього часу розроблений проект стандарту FIPS 202, що описує нову криптографічну геш-функцію.

Таким чином, консервативний і прозорий підхід до проектування нового національного стандарту України обумовив вибір добре перевіреної конструкції на базі S-блоків і лінійного перетворення, для якої можливо забезпечити доказову стійкість до різних видів криптоаналізу.

При порівнянні характеристик S-блоків і інших симетричних перетворень, у т.ч. нових білоруських і російських стандартів, можна відзначити, що саме національний стандарт України забезпечує найбільшу нелінійність булевих функцій, що дає додатковий запас стійкості до лінійного криптоаналізу. Ще більше значення нелінійності для взаємо-однозначної підстановки можна забезпечити, застосовуючи, наприклад, афінно-тквівалентні статечні функції в кінцевому полі, але такі перетворення, використані в AES, Camellia і ін. алгоритмах, ставлять шифр під погрозу реалізації алгебраїчної атаки (цей метод криптоаналізу був успішно застосований проти шифру Keeloq, використовуваного в системах автомобільної безпеки).

У якості схеми розгортання ключів була запропонована нова конструкція з наступними властивостями:

– забезпечення криптографічної стійкості до відомих методів аналізу, відсутність «слабких» ключів, які можуть погіршити властивості перетворення;

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

– зручність програмної й програмно-апаратної реалізації (для формування циклових ключів застосовуються тільки операції, використовувані при шифруванні);

– висока обчислювальна складність відновлення ключа шифрування по одному або декільком цикловим ключам.

Остання властивість забезпечує додатковий захист до атак на реалізацію, коли зловмисник намагається атакувати інженерні розв'язки (вимірюючи споживаний пристроєм струм, навмисне викликаючи збої в роботі через навмисний перегрів шифратора та ін.). Ця особливість є істотною перевагою для ряду додатків, зокрема, при реалізації шифрування на смарт-картах, USB-токенах та ін., коли ключ прошитий у пристрої й повинен бути захищений від зовнішнього доступу (наприклад, у модулях доступу до платних цифрових ТБ-каналів і ін.)

Кількість циклів шифрування залежить від довжини ключа: 10 циклів для 128-бітового, 14 циклів для 256-бітового й 18 циклів для 512-бітового ключа шифрування.

У порівнянні з іншими алгоритмами на основі Square-подібної SPN-структури, блоковий шифр «Калина» має наступні істотні конструктивні відмінності:

– початкове й кінцеве забілювання з використанням модульного додавання (264) для підвищення складності криптоаналітичних атак;

– застосування чотирьох різних S-блоків (замість одного) із властивостями для захисту від алгебраїчних атак, і при порівнянні характеристик з іншими блоковими й потоковими шифрами забезпечують найбільшу нелінійність булевих функцій (104), що дає додатковий запас стійкості перетворення;

– збільшений розмір МДР-перетворення, що поліпшує криптографічні властивості й дозволяє оптимізувати швидкодія на сучасних 64-бітових платформах;

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

– нову односпрямовану схему формування циклових ключів, що забезпечує захист від атак, ефективність програмної й програмно-апаратної реалізації, разом з додатковою стійкістю до методів аналізу спеціального виду.

Оцінка криптографічної стійкості до диференціального, лінійному, алгебраїчному, інтегральному й іншим методам аналізу (практичний критерій) показала, що шифр є стійким при 6 циклах для 128-бітового блоку, 7 циклах для 256-бітового й 9 циклах для 512-бітового (кожний додатковий цикл забезпечує експонентний ріст складності криптоаналізу). Таким чином, шифр, що містить 10, 14 і 18 циклів для блоку 128, 256 і 512 біт відповідно, забезпечує захист від розглянутих видів аналізу й має істотний запас стійкості.

Крім блокового шифру, ДСТУ 7624:2014 визначає режими роботи, що відповідають як ISO 10116:2006, так і додаткові, призначені для сучасних систем криптографічного захисту IP-трафіка, прозорого шифрування носіїв інформації й ін. У стандарті визначені обсяги повідомлень, після обробки яких потрібна обов'язкова зміна ключа. Крім того, приводяться рекомендації розроблювачам, що обертають увагу на необхідність запобігання атак з використанням особливостей реалізації засобів шифрування. Зокрема, враховані особливості, що дозволяли організацію атак BEAST і CRIME/BREACH у протоколах SSL/TLS, повторне приймання повідомлення, відновлення конфіденційних параметрів на основі залежності часу шифрування від оброблюваних даних (через промахи кешу при табличній реалізації) і ін.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс головного вікна складається з наступних блоків:

- Визначення користувача.
- Пошук.
- Дані про профіль.
- Довідка про програму.
- Вікно користувачів, які є у системі, розбиті по різним групам.
- Статус.
- Меню.
- Дзвінок.
- Відеодзвінок.

На рисунку 5.2 зображене вікно діалогу. На ньому відображено аватарки користувачів, які ведуть діалог у системі обміну інформацією.

Також наведені вікна, у яких користувачі друкують повідомлення.

Крім того на формі є наступні кнопки:

- Дзвінок.
- Відео дзвінок.
- Історія повідомлень між обраними користувачами.
- Анкета користувача.

На рисунку 5.3 зображено вікно довідки про програму, де наведені дані про автора програмного проекту, його керівник, тему проекту та місце виконання проекту.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

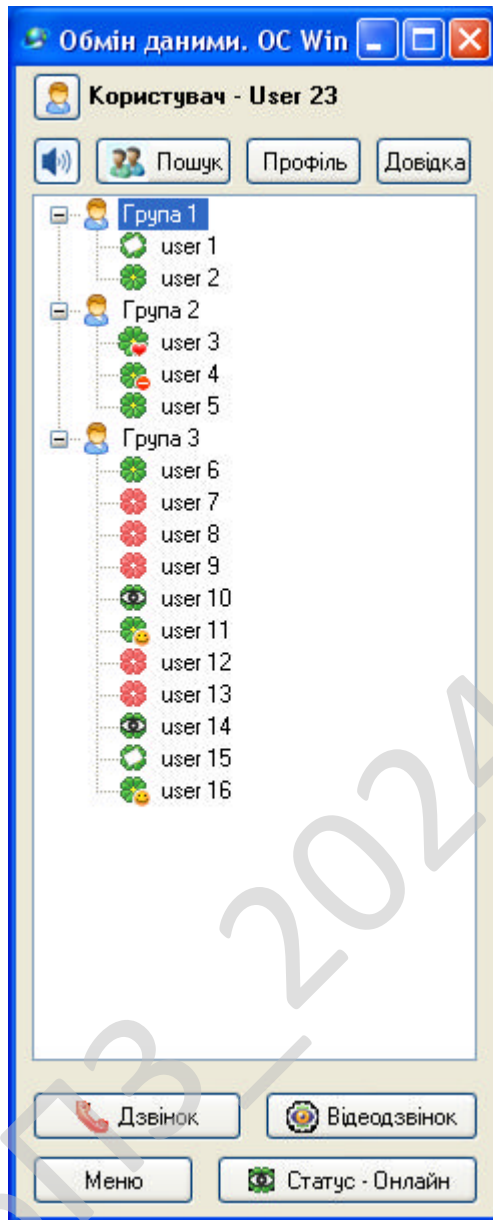


Рисунок 5.1 – Головне вікно програми

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

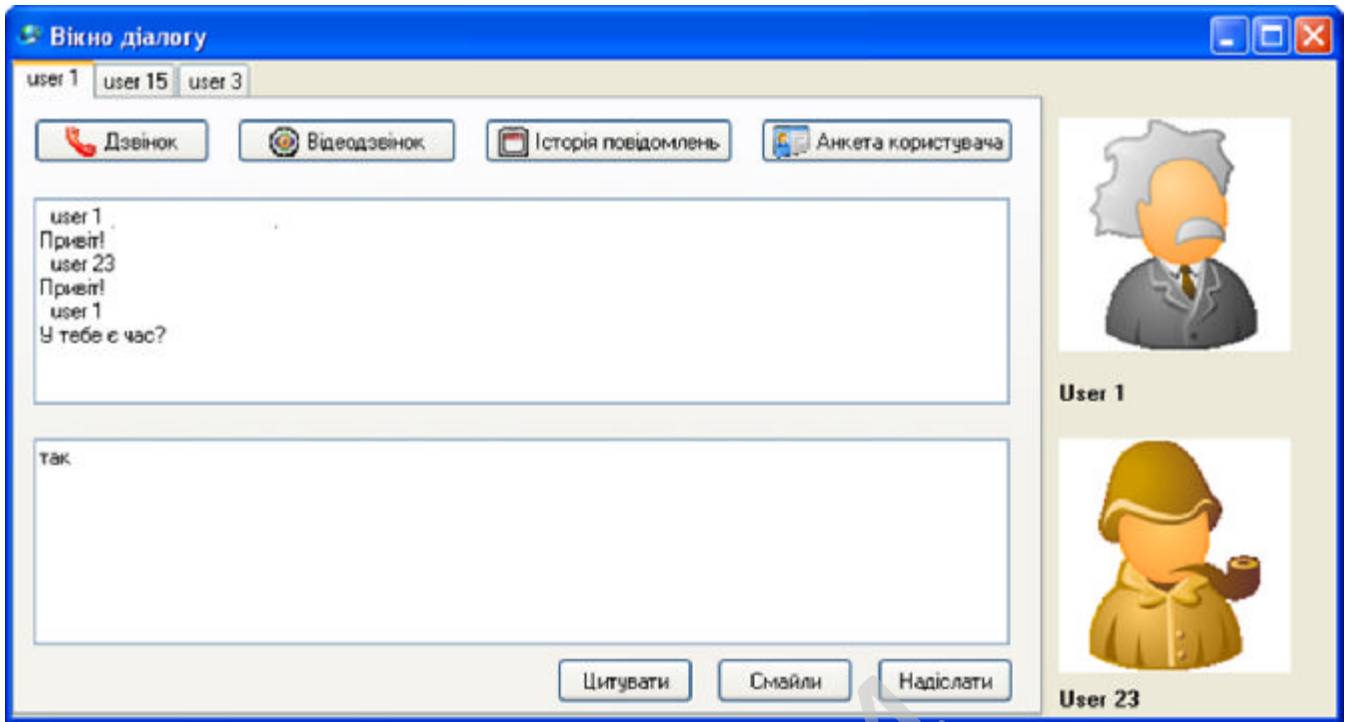


Рисунок 5.2 – Вікно діалогу

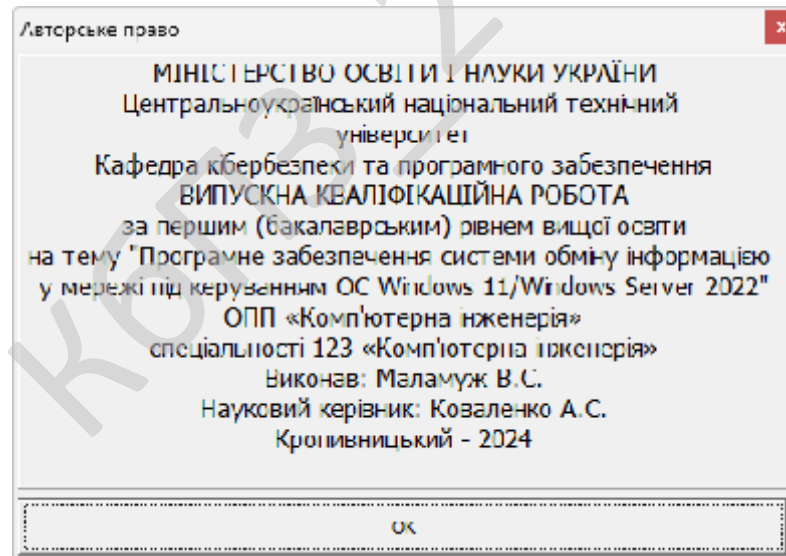


Рисунок 5.3 – Довідка

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

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 інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

– Досліджена система обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

– На основі отриманих результатів досліджень створена програмна реалізація системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

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 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

системи обміну інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на 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 11/Windows Server 2022.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7624:2014.

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.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
2. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
3. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
4. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
5. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
6. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. - Д.: НГУ, 2016. - 187 с.
7. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
8. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
9. 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.
10. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
11. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

12. 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,

13. 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.

14. 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>

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

16. 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.

17. 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.

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

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

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

38. 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.

39. 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.

40. 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.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

41. 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.

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

43. 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.

44. 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.

45. 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).

46. 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

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

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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

49. 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.

50. 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.

51. 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

52. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції “Проблеми та перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. - 2014. - С. 240.

					ВКРБ-123.24.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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-123.24.0009.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Маламуж В.С.				Літ.	Аркуш	Аркушів
Перевірів	Коваленко А.С.						
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 інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

2 Підстава для розробки

Підставою для розробки служить завдання на випуск 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 інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022.

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-123.24.0009.00.00.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 інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022;
- 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-123.24.0009.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		

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, працювати в ОС Windows 11/Windows Server 2022 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 11/Windows Server 2022.

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 Sydney.

					ВКРБ-123.24.0009.00.00.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 – 77 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-123.24.0009.00.00.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-123.24.0009.00.00.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

_____ Коваленко А.С.

*Програмне забезпечення системи обміну інформацією у мережі під
керуванням ОС Windows 11/Windows Server 2022*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 53

Літера: РП

Кропивницький – 2024 року

Серверна частина
Unit_ICQ_OSCAR_RL.pas - основна частина серверного додатку

```

unit Unit_ICQ_OSCAR_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_ICQ_OSCAR_RL = class(TForm)
    PageControl: TPageControl;
    TabSheetServeur: TTabSheet;
    GroupBox1: TGroupBox;
    LabelIP: TLabel;
    ServerSocket: TServerSocket;
    TimerInformations_ICQ_OSCAR_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_ICQ_OSCAR_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
    _ICQ_OSCAR_RL: T_ICQ_OSCAR_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;

_ICQ_OSCAR_HauteurDeDebut, _ICQ_OSCAR_LargeurDeDebut, PageHauteurDebut, PageLargeur
Debut : 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_ICQ_OSCAR_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_ICQ_OSCAR_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_ICQ_OSCAR_RL.TabSheetServeurShow(Sender: TObject);
begin
    LabelIP.Caption := TrouverIP(NomPcActuel);
    GroupBox1.Caption := 'Вибрати сервер: '+ NomPcActuel;
    ComboBoxNombresClients := 999;
end;

//Створення форми чату

//створення форми

procedure T_ICQ_OSCAR_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 11/Windows
Server 2022!!!'#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 11/Windows
Server 2022!!!'#9'Alt+F4');

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

    _ICQ_OSCAR_RL.Left := screen.Width - _ICQ_OSCAR_RL.Width ;
    _ICQ_OSCAR_RL.top := screen.height - _ICQ_OSCAR_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);

    _ICQ_OSCAR_RL.Visible := true;

    _ICQ_OSCAR_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_ICQ_OSCAR_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 11/Windows Server 2022

procedure T_ICQ_OSCAR_RL.ServerSocketClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  inc(NumeroArriveConnexion);
  TimerInformations_ICQ_OSCAR_eur.Enabled := TRUE;
end;

//Закриття форми

//Закриття форми

procedure T_ICQ_OSCAR_RL.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Fermer1Click(_ICQ_OSCAR_RL);
end;

//Запуск та відключення серверу

procedure T_ICQ_OSCAR_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_ICQ_OSCAR_eur.Enabled := FALSE;
        end;
      end;
    end;
  end;
end;

```

```

TimerInformations_ICQ_OSCAR_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_ICQ_OSCAR_eur.Enabled := FALSE;
TimerInformations_ICQ_OSCAR_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 11/Windows Server 2022

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

// Помилки з'єднання сокетів сервера та клієнта програмного забезпечення обміну
інформацією у мережі під керуванням ОС Windows 11/Windows Server 2022

procedure T_ICQ_OSCAR_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 11/Windows Server 2022

```

```

procedure T_ICQ_OSCAR_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_ICQ_OSCAR_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 ('ч',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), '#CTATVC#') = 0 then
        begin
            MessagePrivePourLesClient := MessageRecuServeur;
            MessageRecuServeur := Gauche ('#ЗАБЕРШЕЕННЯ РОБОТИ#', MessageRecuServeur);
            MessageRecuServeur := Droite ('#CTATVC#', MessageRecuServeur);
            ReponseDeMessageSTATUT (MessageRecuServeur, MessagePrivePourLesClient);
        end;

        if Comparestr (copy (MessageRecuServeur,1,8), 'MsgPrive')=0 then
        begin
            MessagePrivePourLesClient := MessageRecuServeur;
            MessageRecuServeur := Gauche ('#ЗАБЕРШЕЕННЯ 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 ('#ЗАБЕРШЕЕННЯ 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('#ЗАВЕРШЕННЯ РОБОТИ#',MessageRecuServeur);
        MessageRecuServeur := Droite('MsgBouger#DE#',MessageRecuServeur);
        ReponseDeMessagePrive(MessageRecuServeur,MessagePrivePourLesClient);
    end;
end;

// Визначення статусу повідомлення

procedure T_ICQ_OSCAR_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;

//Відповісти на особисте повідомлення

procedure T_ICQ_OSCAR_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('#ТИПО ПОВІДОМЛЕННЯ#',MessageRecu);
    MessageTexte := Droite('#ТИПО ПОВІДОМЛЕННЯ#',MessageRecu);

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

end;

procedure T_ICQ_OSCAR_RL.TropDeMondeSurLeServeur(MessageRecu : string);
var
    i,iden : integer;
    Login,NomOrdi,MessageInfoDepart : string;
begin
    //EditLogin.Text + 'µ' + NomPcActuel+'«/\»'+inttostr(NumeroSecret)
    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;
end;

```

```

procedure T_ICQ_OSCAR_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_ICQ_OSCAR_RL.VientDeSeDeconnecterCarVire(MessageRecu : string);
var
  m : integer;
  Login,NomOrdi : string;
begin
  //EditLogin.Text + 'µ' + NomPcActuel+'«/\»'+inttostr(NumeroSecret)
  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_ICQ_OSCAR_RL.AnalysePremiereInformation(MessageRecu : string);
var
  i,j,k,NumeroIdentifiant : integer;
  login,NomOrdi,MessageInfoArrive : string;
  Remplir,LoginExisteDejaDesole : boolean;
begin
  //EditLogin.Text + 'µ' + NomPcActuel + '«/\»' + NumeroSecret
  Application.ProcessMessages;

  Login := Gauche('µ',MessageRecu);
  NomOrdi := Droite('µ',MessageRecu);
  NomOrdi := Gauche('«/\»',NomOrdi);
  NumeroIdentifiant := 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_ICQ_OSCAR_RL.PasLeDroitDeSeConnecter(Login : string; Ordi : string;
iden : integer);
var
    MessageInfoDepart : string;
    i : integer;
begin
    MessageInfoDepart :=
'TuEsVirй°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_ICQ_OSCAR_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;

//Запис часу введення повідомлення

procedure T_ICQ_OSCAR_RL.TimerInformations_ICQ_OSCAR_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_ICQ_OSCAR_RL.MessageInfo : string;
var
    i : integer;
    MessageInfoListe_ICQ_OSCAR_eur,MessageInfoListe_ICQ_OSCAR_eurl : string;
begin
    Application.ProcessMessages;

```

```

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

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

end;

//Виведення імені користувача, з яким спілкуємся

procedure T_ICQ_OSCAR_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;

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

procedure T_ICQ_OSCAR_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;
end;

```

```
procedure T_ICQ_OSCAR_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;

  {Управління звуком}
  initialization
    NumeroArriveConnexion :=0;

end.
```

КБПЗ_2024

Unit_ICQ_OSCAR_RL.pas - параметри інтерфейсу серверної частини

```

unit ChoixCouleurPanel;

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_ICQ_OSCAR_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_SCREENSAVERRUNNING, 0, @n, 0); // Дозволяє закрити
  CtrlAltShift та AltTab
  ClipCursor(nil); // Мишка знаходиться у полі програмного забезпечення.

```

```

    Couleur := ChoixCouleur.Panell.Color;
    _ICQ_OSCAR_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_ICQ_OSCAR_RL.pas - основна частина клієнтського додатку

```

unit Unit_ICQ_OSCAR_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_ICQ_OSCAR_RL = class(TForm)
    ClientSocket: TClientSocket;
    Memo_ICQ_OSCAR_: 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;
    EnLigne1: TMenuItem;
    Occup1: TMenuItem;
    Autlphonel: TMenuItem;
    EnPause1: 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;
  _ICQ_OSCAR_RL: T_ICQ_OSCAR_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, _ICQ_OSCAR_HauteurDeDebut, _ICQ_OSCAR_LargeurDeDebut,
  PageHauteurDebut, PageLargeurDebut: integer;
  TrayIcon: TNotifyIconData;
  blah: HICON;

function UserName(): string;

implementation

uses ChoixCouleurPanel, 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_ICQ_OSCAR_RL.RendreVisiblePremierPlan;
begin
  if (ChoixCouleur.RadioGroupVisible.ItemIndex = 0) then
  begin
    Application.Restore;
    Application.BringToFront;
  end;
end;

function T_ICQ_OSCAR_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_ICQ_OSCAR_RL.Minimize;
begin
  mvtFenetre(1);
  _ICQ_OSCAR_RL.Visible := False;
end;

//створення форми

procedure T_ICQ_OSCAR_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');
_ICQ_OSCAR_RL.Height := Round(500 * (Screen.Width / 1024) / 1.3);
_ICQ_OSCAR_RL.Width := Round(300 * (Screen.height / 768) / 1.3);
_ICQ_OSCAR_HauteurDeDebut := _ICQ_OSCAR_RL.Height;
_ICQ_OSCAR_LargeurDeDebut := _ICQ_OSCAR_RL.Width;
ChoixCouleur := TChoixCouleur.Create(_ICQ_OSCAR_RL);
CAudioVideo := TCAudioVideo.Create(_ICQ_OSCAR_RL);
Randomize;
NombreSecret := RandomRange(1000, 9999);
_ICQ_OSCAR_RL.Left := screen.Width - _ICQ_OSCAR_RL.Width;
_ICQ_OSCAR_RL.top := screen.height - _ICQ_OSCAR_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);
_ICQ_OSCAR_RL.Visible := true;
mvtFenetre(-1);
_ICQ_OSCAR_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_ICQ_OSCAR_RL.WM_CALLBACKPRO(var msg: TMessage);
begin
  case msg.LParam of WM_LBUTTONDOWN:
    begin
      if _ICQ_OSCAR_RL.Visible = true then

```

```

begin
    mvtFenetre(1);
    _ICQ_OSCAR_RL.Visible := False;
end
else
begin
    mvt := 1;
    _ICQ_OSCAR_RL.Visible := True;
    mvtFenetre(-1);
end;
end;
WM_RBUTTONDOWN: traymenu.Popup(mouse.CursorPos.X, mouse.CursorPos.y);
end;
end;

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

//Час перез'єднання

procedure T_ICQ_OSCAR_RL.TimerReconnexionTimer(Sender: TObject);
var
    IPServeur: string;
begin
    if not ClientConnecter 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;

//Закриття форми

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

//З'єднання з сокетом клієнта

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

//Роз'єднання з сокетом клієнта

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

//Помилка у сокеті клієнта

procedure T_ICQ_OSCAR_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_ICQ_OSCAR_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;

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

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

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

procedure T_ICQ_OSCAR_RL.ActionAjouterGroupeExecute(Sender: TObject);
begin
  ClientDataSet3.Insert;
  GroupeNB := GroupeNB + 1;
  ClientDataSet3N_ID.Value := GroupeNB;
  ClientDataSet3GROUPE.Value := 'Groupe Temp';
  ClientDataSet3.Post;
end;

procedure T_ICQ_OSCAR_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_ICQ_OSCAR_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_ICQ_OSCAR_RL.ActionQuitterExecute(Sender: TObject);
begin
  Shell_NotifyIcon(Nim_DELETE, @trayicon);
  Fermer1Click(_ICQ_OSCAR_RL);
end;

procedure T_ICQ_OSCAR_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); //
        Позмір 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[1].NomOrdinateur <> '')
        and (StructureOrdinateur002[1].Iden <> 0) then
    begin
        StructureOrdinateur[1].LoginConnecte :=
StructureOrdinateur002[1].LoginConnecte;
        StructureOrdinateur[1].NomOrdinateur :=
StructureOrdinateur002[1].NomOrdinateur;
        StructureOrdinateur[1].Iden := StructureOrdinateur002[1].Iden;
        StructureOrdinateur[1].Img := StructureOrdinateur002[1].Img;
    end;
end;
end;
end;
end;

procedure T_ICQ_OSCAR_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 +
'#ЗАБЕПШЕННЯ РОБОТИ#';
            _ICQ_OSCAR_RL.ClientSocket.Socket.SendText(#13 + MessageStatut);
        end;
    end;

procedure T_ICQ_OSCAR_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(_ICQ_OSCAR_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_ICQ_OSCAR_RL.MessageInfo: string;
var
  i: integer;
  MessageInfoListe_ICQ_OSCAR_eur, MessageInfoListe_ICQ_OSCAR_eurl: string;
begin
  Application.ProcessMessages;
  MessageInfoListe_ICQ_OSCAR_eurl := '';
  MessageInfoListe_ICQ_OSCAR_eur := '';
  MessageInfoListe_ICQ_OSCAR_eur := '@ш*@';
  for i := 1 to NumeroArriveConnexion + 3 do
  begin
    if StructureOrdinateur[i].LoginConnecte <> '' then
    begin
      MessageInfoListe_ICQ_OSCAR_eurl := MessageInfoListe_ICQ_OSCAR_eurl + 'µ' +
      StructureOrdinateur[i].LoginConnecte
      + '#IMG#' + StructureOrdinateur[i].img + '#ORDI#' +
      StructureOrdinateur[i].NomOrdinateur;
    end;
  end;

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

//Читання з сокету клієнта

procedure T_ICQ_OSCAR_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_ICQ_OSCAR_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), 'x@*!@x')
= 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;
          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;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        if Image = 'Зайнятий' then
ClientDataSet1IMAGE.Assign(IMG_PAUSE.Picture.Bitmap);
        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_1MN.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;

if CompareStr(copy(MessageRecuClient, 1, 3), ' ') = 0 then
begin
    cxGrid3DBTableView1.ClearItems;
    if length(MessageRecuClient) > 5 then
    begin
        MessageRecuClient := Droite(' ', MessageRecuClient);
    end;
end;

if CompareStr(copy(MessageRecuClient, 1, 5), ' ') = 0 then
begin
    if CompareStr(copy(MessageRecuClient, length(MessageRecuClient) - 4, 5), ' ') = 0 then
    begin
        MessageRecuClient := Gauche(' ', MessageRecuClient);
        MessageRecuClient := Droite(' ', MessageRecuClient);
        if beep then
            MessageBeep(MB_OK);
        RendreVisiblePremierPlan;

        if CompareStr(copy(MessageRecuClient, 1, 4), ' ') = 0 then
        begin
            MessageRecuClient := Droite(' ', MessageRecuClient);
            MsgTemp := Droite('>>', Gauche('З'єднання', MessageRecuClient));
            MsgTemp := MsgTemp + #13 + 'З'єднання';
            AlertMsgBox('З'єднання', MsgTemp, 0, false, 1000, 10, nil);
            Memo_ICQ_OSCAR_.Lines.Add(MessageRecuClient + ' ' + QuelHeureEstIl);
        end
        else
        begin
            AlertMsgBox('Поз'єднання', MessageRecuClient, 0, false, 1000, 10, nil);
            Memo_ICQ_OSCAR_.Lines.Add(MessageRecuClient);
        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;
end;

```

```

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_ICQ_OSCAR_.Lines.Count > 10) then
      Memo_ICQ_OSCAR_.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(_ICQ_OSCAR_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
        else
        begin
          MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' dit : ' + MessageTexte);
          FlashWindow(Application.Handle, true);
        end;
      end
      else
      begin
          MsgPerso := TMsgPerso.Create(_ICQ_OSCAR_RL);
          MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :';
          MsgPerso.cxGroupBox2.Caption := ' A : ' + 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_ICQ_OSCAR_.Lines.Count > 10) then
            Memo_ICQ_OSCAR_.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(_ICQ_OSCAR_RL);
                    MsgPerso.Caption := 'Хочете відправити особисте повідомлення ' +
                    LoginEnvoi + ' :';
                    MsgPerso.cxGroupBox2.Caption := ' А : ' + LoginEnvoi + ' ';
                    MsgPerso.Memol.Lines.Add(LoginEnvoi + ' Ви відправити файл: ' +
                    MessageTexte);
                    ///////////////////////////////////

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

                    MsgPerso.Memol.Lines.InsertObject(0, 'Test', SpeedButton3);
                    MsgPerso.Show;
                    FlashWindow(Application.Handle, true);
                end
                else
                begin
                    MsgPerso.Memol.Lines.Add(LoginEnvoi + ' Ви відправити файл: ' +
                    MessageTexte);
                    FlashWindow(Application.Handle, true);
                end;
            end
            else
            begin
                MsgPerso := TMsgPerso.Create(_ICQ_OSCAR_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('Ви пропонуєте передачу файлів');
end;
end;

if CompareStr(copy(MessageRecuClient, 1, 9), 'MsgBouger') = 0 then
begin
    MessageRecuClient := Gauche('#ЗАВЕРШЕННЯ РОБОТИ#', MessageRecuClient);
    MessageRecuClient := Droite('MsgBouger#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_ICQ_OSCAR_.Lines.Count > 10) then
        Memo_ICQ_OSCAR_.Lines.Clear;

    if (LoginEnvoi <> UserName) then
begin
    if assigned(MsgPerso) then
begin
begin
    if (MsgPerso.Caption <> 'Хочете відправити особисте повідомлення ' +
LoginEnvoi + ' :') then
begin
        MsgPerso := TMsgPerso.Create(_ICQ_OSCAR_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
    else
begin
        MsgPerso.Memo1.Lines.Add(LoginEnvoi + ' ' + MessageTexte);
        Bouge := True;
        FlashWindow(Application.Handle, true);
    end;
end
end
else
begin
    MsgPerso := TMsgPerso.Create(_ICQ_OSCAR_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
end
else
begin
    MsgPerso.Memo1.Lines.Add('Доступний тільки для тих, хто повинен
бачити!');

```

```

    end;
end;

if CompareStr(copy(MessageRecuClient, 1, 5), 'Підключено') = 0 then
begin
Memo_ICQ_OSCAR_.Lines.Add('Сервер роз'єднаний... ' + QuelHeureEstIl);
Application.ProcessMessages;
cxGrid3DBTableView1.ClearItems;

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

procedure T_ICQ_OSCAR_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;

    Application.ProcessMessages;

    cxGrid3DBTableView1.ClearItems;

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

{Зміна вкладки}

procedure T_ICQ_OSCAR_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
                    DroitEcrire := TRUE;
                    end;
                DS_SOURCE.DataSet.Next;
            end;
            if DroitEcrire then
            begin
                MessageAEnvoyer := ' ' + EditLogin + ' ';
                ClientSocket.Socket.SendText(#13 + MessageAEnvoyer);
            end;
            end;
            if (Key = #13) then
                key := #0;
        end;
end;

```

```

procedure T_ICQ_OSCAR_RL.Fermer1Click(Sender: TObject);
begin
  ClipCursor(nil);
  if (not ClientConnector) then
    Application.Terminate;

  if ClientConnector = TRUE then
  begin
    DireQueOnSeDeconnecte;
    ClientSocket.Active := FALSE;
    sleep(1000);
    Application.Terminate;
  end;

end;

procedure T_ICQ_OSCAR_RL.DireQueOnSeDeconnecte;
var
  MessageFinal: string;
begin
  Application.ProcessMessages;

  cxGrid3DBTableView1.ClearItems;

  if ClientConnector then
  begin
    MessageFinal := '@DECO' + EditLogin + 'µ' + NomPcActuel + '«/\»' +
    intostr(NombreSecret) + 'K';
    ClientSocket.Socket.SendText(#13 + MessageFinal);
  end;

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

end;

procedure T_ICQ_OSCAR_RL.EnLigne1Click(Sender: TObject);
var
  j: integer;
  MessageStatut: string;
  Icon: TIcon;
begin
  MessageStatut := (Sender as TMenuItem).Caption;
  Icon := TIcon.Create;
  DS_SOURCE.dataset.First;
  while not DS_SOURCE.DataSet.Eof do
  begin
    if UserName = ClientDataSet1NOMPrenom.Value then
    begin
      ClientDataSet1.Edit;
      if MessageStatut = '&У мережі' then
      begin
        ImageList1.GetIcon(0, Icon);
        Image1.Picture.Bitmap := IMG_LIGNE.Picture.Bitmap;
        ClientDataSet1IMAGE.Assign(IMG_LIGNE.Picture.Bitmap);
      end;
      if MessageStatut = '&Зайнятий' then
      begin
        ImageList1.GetIcon(1, Icon);
        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;
    end;
  end;
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 ClientConnecter = TRUE then
begin
  MessageStatut := '#СТАТУС#' + ComboBox1.Text + '#ДЕ#' + EditLogin +
'#ЗАБЕПШЕННЯ РОБОТИ#';
  _ICQ_OSCAR_RL.ClientSocket.Socket.SendText(#13 + MessageStatut);
end;
end;
DS_SOURCE.DataSet.Next;
end;

end;

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

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

initialization
  NumeroArriveConnexion := 0;

end.

```

ChoixCouleurPanel.pas - параметри інтерфейсу клієнтської частини

```

unit ChoixCouleurPanel;

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;
    Panell1: 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_ICQ_OSCAR_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', _ICQ_OSCAR_RL.sdl.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);
  _ICQ_OSCAR_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
  _ICQ_OSCAR_RL.sdl.SkinFile:=path+Couleur.Text;
  if not _ICQ_OSCAR_RL.sdl.Active then _ICQ_OSCAR_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.

```

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_ICQ_OSCAR_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
  IdFTP1.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.IdFTP1Server1ChangeDirectory(ASender: TIdFTP1ServerContext;
    var VDirectory: string);
begin
    ASender.CurrentDir := VDirectory;
end;

procedure TMsgPerso.IdFTP1Server1DeleteFile(ASender: TIdFTP1ServerContext;
    const APathName: string);
begin
    DeleteFile(ReplaceChars(AppDir + ASender.CurrentDir + '\' + APathname));
end;

procedure TMsgPerso.IdFTP1Server1GetFileSize(ASender: TIdFTP1ServerContext;
    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.IdFTP1Server1ListDirectory(ASender: TIdFTP1ServerContext;
    const APath: string; ADirectoryListing: TIdFTP1ListOutput; const ACmd,
    ASwitches: string);
var
    LFTPItem: TIdFTP1ListItem;
    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('Unable to create directory');
    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#';
    _ICQ_OSCAR_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#';
    _ICQ_OSCAR_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#';
      _ICQ_OSCAR_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
    OHigh: integer; // Оригінальний розмір форми
    WaitBeforeDownCounter: integer;
    WaitBeforeDownTime: integer;
    AlertWindowState : byte; // 0 = змонтувати; 1 = зупинка; 2 = походження; 3 =
інактивації - 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;
  OHigh := 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;
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.
```

К6П3_2024

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_ICQ_OSCAR_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 міксеру у системі!',mtError,[mbOK],0);
          end;
          Setting:=False;
        end;
      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;
    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;
    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 міксеру у системі!',mtError,[mbOK],0);
end;
Setting:=False;
end;

end.
```

К6ПЗ_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('БАКІЯВРСЬКИЙ ПРОЕКТ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Програмне забезпечення обміну інформацією у мережі  
під керуванням ОС Windows 11/Windows Server 2022');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Коваленко А.С. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Маламуж Владислав Сергійович');
  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.
```