

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 кібербезпеки поштового
клієнта захищеного документообігу з використанням
протоколів TLS/SSL”**

КБПЗ-2024

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

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
« ____ » _____ 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

Алексєєвському Олексію Владиславовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL*

2. Керівник роботи *Смірнов Олексій Анатолійович, докт. техн. наук, професор*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

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

3. Строк подання студентом роботи до захисту *23.05.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 17 » січня 2024 р.

Підпис керівника

Смірнов О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2024 р.

Підпис здобувача

Алексеевський О.В.
(прізвище та ініціали)

АНОТАЦІЯ

Алексєєвський О.В. Програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL. 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 кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Метою розробки є програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Результат роботи – програмна реалізація системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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, поштовий клієнт, документообіг, TLS/SSL

ABSTRACT

Aliexieievskiy O.V. Software of the cyber security system of the email client of secure document circulation using TLS/SSL protocols. 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 cyber security system of a mail client of secure document circulation using TLS/SSL protocols.

The purpose of the development is the software of the cyber security system of the mail client of secure document circulation using TLS/SSL protocols.

The result of the work is the software implementation of the cyber security system of the mail client of secure document circulation using TLS/SSL protocols.

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, mail client, document management, TLS/SSL

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	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	26
3.3 Розробка функціональної схеми	33
3.4 Розробка діаграми процесів.....	43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	60
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	62
6 ОСНОВНІ ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66

					ВКРБ-123.24.0001.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Алексєєвський О.І</i>					Б	1	72
<i>Перев.</i>	<i>Смірнов О.А.</i>					<i>ЦНТУ КІ-20</i>		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

ВСТУП

Актуальність теми. Організації переміщують все більше і більше бізнес-процесів в Інтернет, відповідно захист конфіденційності та безпеки інформації, яка використовується в цих процесах, а також забезпечення автентичності та цілісності є надзвичайно важливими. Оскільки багато автоматизованих процесів покладаються на електронні документи, що містять конфіденційну інформацію, організації повинні належним чином захищати ці документи. Багато рішень із захисту інформації намагаються захистити електронні документи лише тоді, коли вони зберігаються або передаються. Однак ці рішення не забезпечують захист протягом усього життєвого циклу електронного документа. Коли документ досягає одержувача, захист втрачається, і документ може бути навмисно чи ненавмисно пересланий і переглянутий неавторизованими одержувачами.

Набагато ефективнішим рішенням є захист документа шляхом призначення параметрів безпеки, які передаються разом з ним. Щоб забезпечити більш ефективний захист електронного документа протягом усього його життєвого циклу, необхідно виконати шість критеріїв: конфіденційність, авторизація, підзвітність, цілісність, автентичність і неспростовність. Двома основними методами безпеки, які використовуються для встановлення цих шести критеріїв безпеки документів, є керування документами та цифрові підписи.

Організації можуть легко інтегрувати рішення безпеки електронних документів у свої поточні бізнес-процеси та корпоративну інфраструктуру для підтримки широкого спектру простих і складних процесів. Рішення динамічно захищають електронні документи як у мережі, так і поза нею, онлайн і офлайн, забезпечуючи наскрізну безпеку протягом усього життєвого циклу електронного документа.

Одним з елементів систем електронного документообігу є поштові клієнти.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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

- Огляд існуючих систем кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.
- Дослідження системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.
- Програмна реалізація системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для реалізації програмного забезпечення поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Сьогодні більшість компаній переходять від документообігу до електроніки. Це економить час і гроші організації. Електронний документообіг зменшує паперову роботу. Інформацію можна ділитися 24 години на добу з офісу чи з іншого місця. Для цього достатньо мати електронний підпис і комп'ютер, підключений до мережі Інтернет.

Крім того, система електронного документообігу дозволяє контролювати документообіг в режимі реального часу, керувати рухом паперових та електронних документів із збереженням історії роботи з документами.

Простіше кажучи, електронний документообіг – це обмін електронними документами через локальні документи, Інтернет чи інші канали. Електронний документообіг може відбуватися як всередині організації, так і між компаніями.

Електронний документообіг – це сукупність процесів надсилання та отримання електронних документів через інформаційну систему. Електронний документообіг може використовуватися для укладення угод (у тому числі договорів), розрахунків, офіційного та неофіційного листування та іншої інформації.

Система електронного документообігу виконує такі функції:

- реєстрація документів;
- контроль за виконанням документів;
- створення каталогів;
- створювати та редагувати реквізити документів;
- формування звітів про документообіг підприємства;

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

- імпортувати документи з файлової системи та Інтернету;
- створити документ безпосередньо з системи за шаблоном;
- працювати з версіями документів, складними багатокomпонентними та багатоформатними документами, вкладеннями;
- електронний розподіл документів;
- працювати з документами в папках;
- отримання документів шляхом сканування та розпізнавання;
- зменшити витрати на доступ до даних та обробку документів.

1.2 Область застосування

Областю застосування є системи електронного документообігу. Налаштувати систему електронного документообігу можна двома способами.

По-перше, є можливість укласти договір електронного документообігу з контрагентами та обмінюватися документами, підписаними електронним підписом, через електронну пошту.

По-друге, налаштувати ЄДІ можна через приватного оператора. У цьому випадку потрібно долучитися до регулювання електронного документообігу. Після цього відбудеться обмін офіційними документами та неофіційними документами з контрагентами.

Електронний документообіг – це повна інформаційна система, що включає спеціалізоване програмне забезпечення, електронну пошту, Інтернет, локальну мережу тощо.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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

Працювати з електронною поштою можна не тільки за допомогою інтерфейсу поштового сервісу – наприклад, ukr.net, Gmail і т.д. Для роботи з електронною поштою набагато зручніше використовувати десктопні поштові клієнти – установлені в операційну систему програми, які можуть підключатися до поштового сервера, завантажувати пошту й відображати електронний ящик уже у своєму інтерфейсі. Поштові клієнти більше функціональні, ніж веб-інтерфейс поштових сервісів. У числі їхніх переваг – можливість роботи з декількома поштовими аккаунтами в єдиному програмному інтерфейсі, що набудовуються параметри зовнішнього вигляду, додатковий функціонал для роботи з потоком кореспонденції, інтеграція з іншими програмами типу органайзерів, оповіщення надходження нових листів, різні способи їхнього сортування й т.д.

Нижче розглянемо п'ять поштових клієнтів для Windows. Троє перших – дітища творця цієї операційної системи, компанії Microsoft. А два останніх поштових клієнти – продукти розроблювачів популярних браузерів Mozilla Firefox і Opera.

Перш ніж приступитися до огляду, відзначимо один важливий нюанс. Не всі поштові сервіси за замовчуванням надають десктопним поштовим клієнтам доступ до завантаження пошти. Цей момент необхідно уточнити в налаштуваннях кожного окремого поштового сервісу.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Як і всі Metro-додатка, «Пошта» – мінімалістична, у ній є присутнім лише необхідний функціонал поштового клієнта, включаючи можливість роботи з декількома аккаунтами електронної пошти.

Якщо на комп'ютері встановлена збірка Windows 10/11 без базового комплекту Metro-додатків, «Пошту» можна встановити в магазині Windows безкоштовно.

«Пошта Windows Live»

Поштовий клієнт «Пошта Windows Live» в 2007 році прийшов на зміну Windows Mail, убудованому в Windows 8/10, що, у свою чергу, перемінив на пості убудований в Windows XP Outlook Express. «Пошта Windows Live» входить до складу програмного пакета Windows Live. При установці поштового клієнта від інших компонентів можна відмовитися, забравши в майстру інсталяції їхньої галочки.

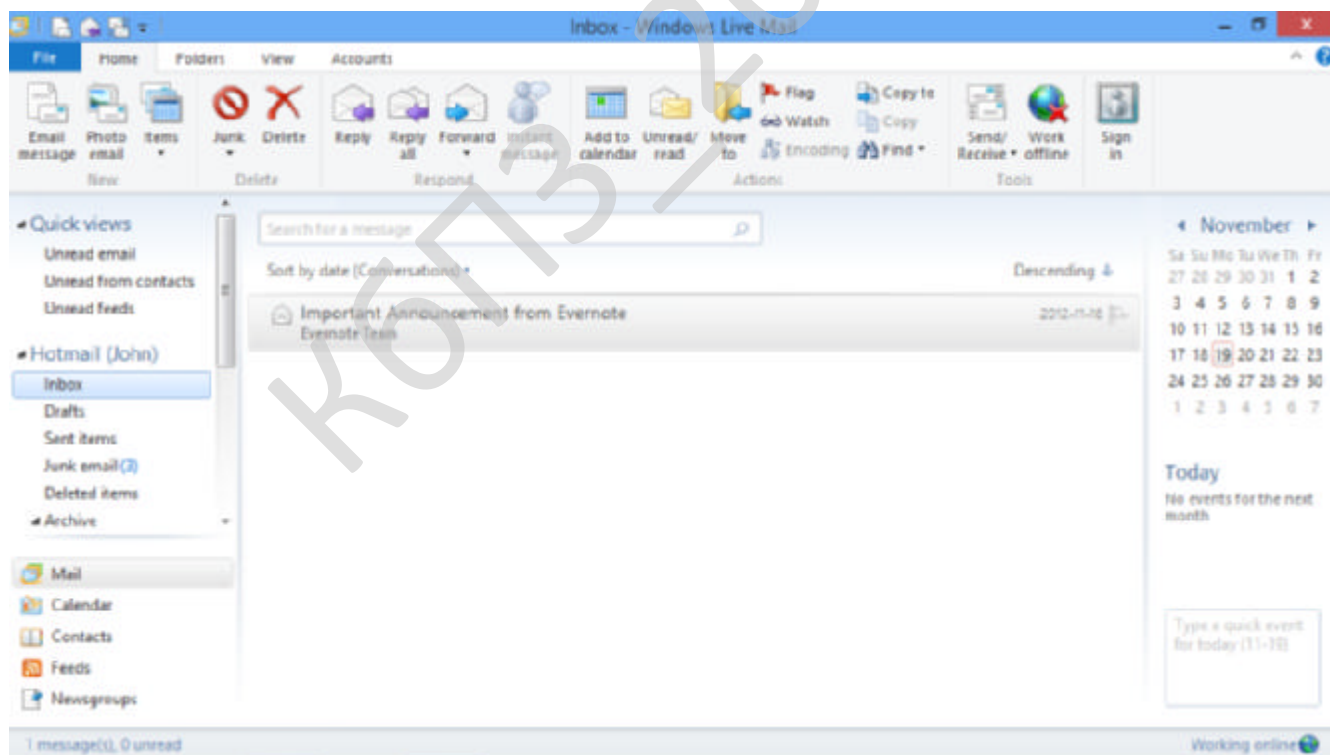


Рисунок 2.3 – Інтерфейс користувача «Пошта Windows Live»

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Із числа переваг поштовика – аскетичний, але досить зручний інтерфейс, легковагість, наймогутніший спам-фільтр для небажаної кореспонденції, якому можна в процесі роботи з поштою «навчати». Mozilla Thunderbird, як і браузер Firefox, можна доповнювати функціональними розширеннями й «прикрашати» темами оформлення. До складу програми Mozilla Thunderbird входить також RSS-рідер і адресна книга для зручності роботи з контактами.

Mozilla Thunderbird – кроссплатформене рішення. Привикнув до інтерфейсу програми на одній операційній системі, приміром, на Windows, у випадку зміни системи, скажемо, на Linux, не потрібно піклуватися про пошук нового поштовика й звикати до нового дизайну, до нової організації функціональних команд. Mozilla Thunderbird можна встановити й на Linux, і на Mac OS, і на Windows. Поштовий клієнт безкоштовний.

Opera Mail

Після кардинальних змін у житті норвезького браузера Opera, коли в 2013 році власний движок Presto розроблювач Opera Software перемінив на платформу Chromium, з Опери зникли убудований поштовий клієнт і RSS-рідер. Але не безвісти. Прибраний із браузера функціонал норвежці реалізували в окремий програмний продукт у стилі Opera 12 – поштовий клієнт Opera Mail.

В Opera Mail дуже спрощений дизайн. Немає навіть можливості установки тим оформлення. У той же час – це ідеальний інструмент для новачків, не перевантажений просунутим функціоналом, із простою організацією й зрозумілим керуванням. Крім поштового клієнта, програма Opera Mail обладнана RSS-ридером, є присутнім можливість окремого відображення електронних листів по призначенню раніше мітках і типу вкладених у листи файлів.

Opera Mail стане практичним рішенням для малопотужних комп'ютерних пристроїв. Споживання системних ресурсів цим поштовим клієнтів мінімальне.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

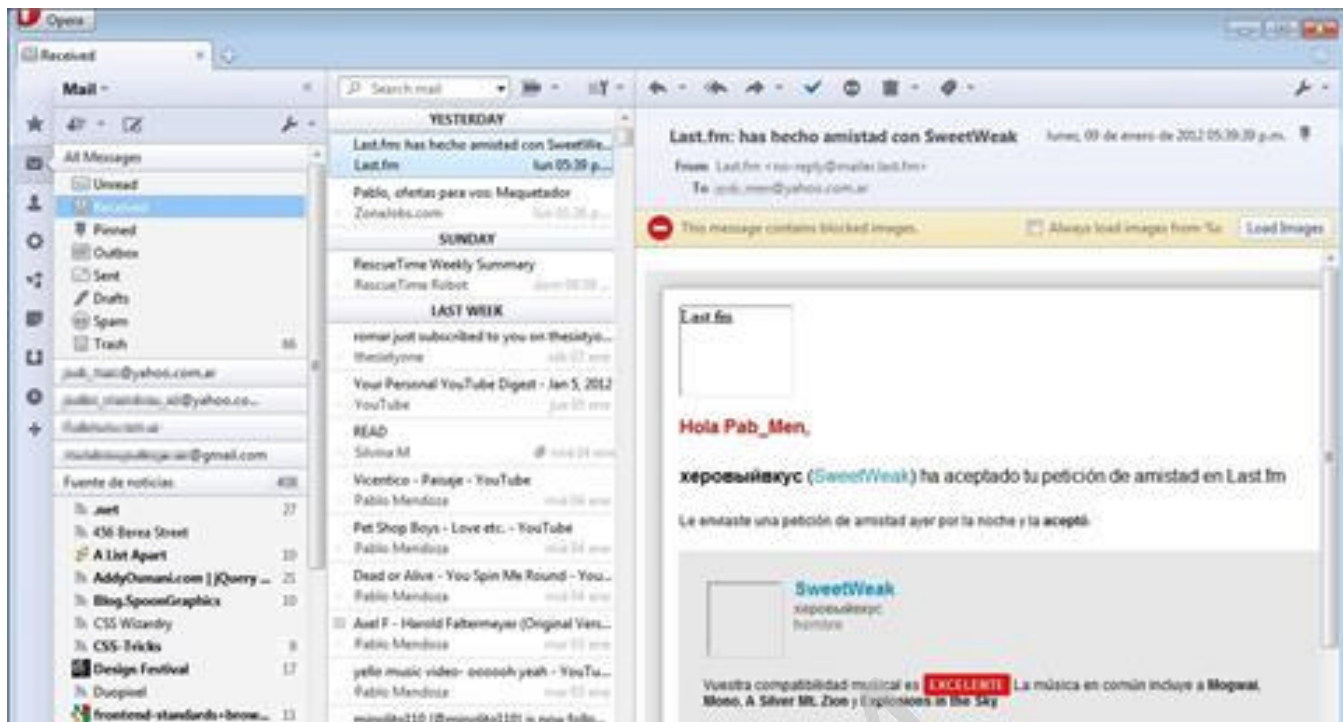


Рисунок 2.5 – Інтерфейс користувача Opera Mail

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

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

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

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

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

– Тип даних Delphi «record» тепер підтримуватимуть 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 IME.

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

– 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.

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

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

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

Єдине керування пам'яттю

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

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

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

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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

Win 64-відладник і збирач для C++

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

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка 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.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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.

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 стилізуємі FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБПЗ - 2024

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Системи електронного документообігу, з'явившись відносно недавно, у цей час міцне зайняли своє місце в інформаційній структурі підприємства. Спочатку в їхнє завдання входила винятково проста автоматизація документообігу, тобто допомога діловодам, але ні в якій мірі не заміна документних потоків, що циркулюють на підприємстві.

Багато чого змінилося тоді, коли виробники СЕД почали не просто копіювати процеси підприємств, що забезпечують рух паперових документів, але враховувати завдання, що коштують як перед цими процесами зокрема, так і перед всієї документообігом у цілому.

Розуміння того, що документообіг підприємства призначений для підтримки його системи керування, прийшло в СЕД далеко не відразу. Більше того, багато сучасних федеральних і муніципальних органів влади й по справжній момент використовують найпростіші СЕД, у завдання яких входить лише дублювання паперового документообігу, але не заміна його й не оптимізація й підтримка структури керування.

Розділяючи хронологічно етапи еволюції систем електронного документообігу, варто виділити перший з них, що почався в 80-х роках ХХ століття й який продовжувався фактично до його кінця. Цей період характерний розрізненістю рішень, що з'явилася внаслідок того, що підприємства, керівництво яких уже розуміло необхідність автоматизації документообігу, ще не мали уніфікованого інструментарію, необхідного для рішення завдань такого роду.

Розглянемо причини, які спочатку привели до необхідності автоматизувати документопотоки. В останній чверті двадцятого століття обсяги управлінської документації підприємств ще не досягли критичних показників,

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

однак наявна територіальна розрізненість структурних підрозділів приводила до того, що процедури узгодження керівниками різних рівнів могли проходити тижнями й в окремих випадках навіть місяцями. Введення на таких підприємствах служб кільцевої пошти дозволяло оптимізувати тільки механізми послідовного узгодження, віялове ж, або як його називають зараз, паралельне узгодження, вимагало дуже великих часових витрат, внаслідок чого його використання зводилося до мінімуму й застосовувалося, по можливості, винятково межах одного будинку.

Коли ж у багатьох великих промислових підприємствах і органах влади обсяги документообігу стали підходити до критичним значенням, стало зрозуміло, що процеси документообігу потрібно піддати кардинальному реінжинірингу, тобто перебудові, у результаті якої при збереженні загальних вимог до результатів процесів самі процеси повинні були бути оптимізовані й, як наслідок, їхнє проходження повинне було займати значно менше часу.

Таким чином, перші СЕД були повністю індивідуалізовані, розроблялися безпосередньо на підприємствах, для яких створювалися силами внутрішніх ресурсів. Очевидні переваги такого підходу (чітка відповідність побудованої СЕД фактичним регламентованим процесам документообігу, загальна інтегрованість в інформаційну інфраструктуру організації) були з лишком компенсовані одним принциповим недоліком: така система, розроблена в самій організації, звичайно була немасштабованою, а змінити структуру автоматизуємих процесів було практично неможливо, що приводило до неможливості розвитку такої системи. А оскільки отримана в такий спосіб СЕД була нерозривно пов'язана з моделлю керування організацією, то наступав момент, коли розвиток компанії повністю зупинялося внаслідок того, що використовувана система електронного документообігу сама гальмувала те, що повинна була вдосконалювати: ефективність прийнятої в організації системи керування починала швидко падати й компанія була змушена прикладати зусилля для того, щоб створити адекватну

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

СЕД, яка б змогла підтримати керування й зробити розвиток компанії більше ефективним.

Рішення цих завдань прийняли на себе одночасно кілька компаній, створених у середині 90-х років: вони почали створювати універсальні СЕД, які легко масштабувалися під потреби практично будь-яких замовників, а сама технологія створення кінцевого рішення СЕД стала двоетапною: на першому етапі компанія створювала уніфіковане ядро СЕД, на другому етапі відбувалося впровадження – припасування процесів під потреби конкретного замовника. Такий підхід дозволив знизити вартість кінцевих рішень, а будучи одночасно значно більше функціональним, забезпечував можливості організаційного й функціонального масштабування системи.

З кінця ХХ-го століття підхід до автоматизації документообігу почав змінюватися якісно: у Росію прийшли нові підходи до керування, що дозволили підвищити його ефективність уже методологічно. Саме так до нас прийшло поняття менеджменту якості, а разом з ним – процесний підхід до керування. Інформаційні системи практично відразу відреагували на ці віяння, як результат у них прийшло поняття Workflow – потоків робіт, що представляли із себе одиничні екземпляри ділових документоорієнтованих процесів. Сучасні СЕД відгукнулися на їхню появу реалізацією процесно-орієнтованих движків (Workflow-engines), які стали одночасно новим методологічним кроком, що дозволяє сучасним СЕД ще швидше й ефективніше адаптуватися під потреби компаній, які швидко росли. Прихід на український ринок західних розроблювачів (IBM Lotus, Documentum), що робила платформи для цих систем, увів поняття ECM – enterprise content management (керування інформацією підприємства). Методично СЕД, що ставляться до категорії ECM, відрізнялися від інших тем, що в них з'явився чіткий поділ: сам документ, його інформаційна складова (контент). ECM-система працювала з контентом документів за допомогою його метаданих – виділеної інформації, що має певну важливість для організації. Так, для вихідного листа метаданими могли бути адресат, дата

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

підписання й посадова особа, що підписує, реєстраційний номер і відомості про відправлення й доставку.

Використання workflow дозволило вже самим фахівцям компаній, що експлуатують системи електронного документообігу, забезпечити безперервна зміна автоматизуємого процесу з метою його максимальної відповідності процесам підприємства. Безумовно, це був величезний стрибок уперед.

Змінилася й архітектура рішень: якщо спочатку СЕД являла собою приклад дволанкової архітектури (СУБД – додаток), то до 2005-му року вимога триланкової архітектури СУБД – сервер додатків – інтерфейс користувача стало фактичним стандартом галузі, а організації, що проводять тендери на поставку СЕД, стали вказувати саме таку вимогу до системної архітектури як краще або обов'язкове.

Благодатна економічна ситуація цього часу сприяла бурхливому росту підприємств всіх галузей, одночасно створюючи сприятливі умови для виводу на ринок нових продуктових рішень СЕД. У цей час остаточно сформувалося коло виробників СЕД, що працюють на українському ринку й ринках найближчого зарубіжжя.

Сучасна функціональність СЕД

Корпоративні інформаційні системи й системи електронного документообігу зокрема являють собою рішення, переважно призначені для керування інформаційними ресурсами середніх і великих підприємств. Основними причинами такого розшарування є, насамперед досить висока ціна “вхідного квитка” на цей ринок. Придбання повноцінної, якісної й безпечної системи керування базами даних (СУБД), забезпечення її сучасними засобами криптографічного захисту інформації, а також необхідність залучення до проектів впровадження висококваліфікованого персоналу аналітиків, консультантів і внедренців привели до того, що компанія, що прийшла до необхідності одержання такого потужного засобу керування корпоративним контентом, як СЕД, була змушена одноразово понести досить більші фінансові

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

витрати, яких не могли собі дозволити індивідуальні підприємці й малі підприємства. Таким чином, СЕД ставали інструментами вдосконалювання керування саме у великих компаніях, а забезпечити максимальну віддачу від впровадження могли територіально-розподілені комерційні структури холдингового типу.

Розвиток функціональності СЕД привело до того, що усе більше документоорієнтованих процедур керування було автоматизовано й всі більше областей повсякденної діяльності компаній були покриті їхнім функціоналом.

До теперішнього часу фактичним стандартом сучасних СЕД є наступні області діяльності:

– загальне діловодство – обробка вхідної кореспонденції й звернень громадян, підготовка вихідних листів і документів, внутрішня й організаційно-розпорядницька документація, контроль виконання доручень;

– кадрове діловодство – жорстко регламентована область приватного документообігу, що дозволяє здійснювати підтримку процедур прийому на роботу й звільнення працівників, напрямку працівників у відрядження, надання планової відпустки й інших видів відпусток. Такі системи забезпечують ведення всіх затверджених форм кадрового діловодства: наказ про прийом працівника на роботу, наказ про розірвання трудового договору, наказ про переклад працівника на іншу роботу, особиста картка працівника, штатний розклад, графік відпусток;

– архівне діловодство – область приватного діловодства, що завершує життєвий цикл документів організації й підтримуючої наступної процедур: формування описів по затверджених формах, передача справ на архівне зберігання, експертиза цінності документів і справ у цілому, знищення документів і справ, що більше не представляють цінності для організації, її співробітників, держави й суспільства, здача окремих справ на зберігання в структури архівного агентства;

– колегіальна робота – дозволяє автоматизувати документне забезпечення діяльності колегіальних органів керування – нарад, засідань, рад директорів,

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

зборів акціонерів. Також дозволяє здійснювати підготовку, узгодження й твердження повісток дня (планової документації) і протоколів (результуючої документації);

– керування взаєминами із клієнтами – функціонал CRM (consumer relationship management) зараз у тім або іншому ступені реалізований у всіх СЕД, крім самі старого, у користувачам яких, втім, такий функціонал не особливо затребуваний.

І хоча в цілому комплекс завдань електронного документообігу досить зрозумілий, способи їхньої реалізації сильно відрізняються. Виходить, що одне з головних вимог, пропонованих до розроблювачів сучасної СЕД, – запропонувати адекватне за ціною, якості й строкам впровадження рішення незалежно від специфіки роботи замовника (іншими словами – задовольняючій будь-якій специфіці).

3.2 Розробка структурної схеми

У цей момент обмін повідомленнями електронної пошти став основним засобом обміну інформацією й документами для ведення бізнесу. По електронній пошті приходять замовлення від клієнтів, виставляються рахунки, підтверджуються зустрічі, а також здійснюється документообіг усередині й між організаціями.

Перш ніж описати роботу поштового клієнта, розглянемо основні засади застосування поштового серверу.

Раніше малі й середні компанії для роботи з поштою користувалися послугами хостинг-провайдерів, т.к. це дозволяло легко й просто одержати поштовий сервіс, не замислюючись про його адміністрування й підтримку. Але з розвитком інформаційних технологій усе більше й більше компаній стали переходити на власні поштові сервери.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Власний поштовий сервер має ряд переваг у порівнянні з використанням послуг хостинг-провайдерів:

1. Безпека. Ваш поштовий сервер знаходиться у вас, на території вашої компанії й ніхто, крім вас не має до нього доступу. Ніхто крім вас не зможе читати вашу секретну інформацію й не одержить доступу до інформації, що ставиться до комерційної таємниці.

2. При використанні пошти, розташованої на хостинзі, ваша конфіденційна інформація розташована на серверах хостинг-компанії, де працюють сотні людей і всі з них при бажанні можуть одержати доступ до вашої пошти й розкрити комерційну таємницю, при цьому ви не зможете не розкрити не довести факт крадіжки.

3. Індивідуальність. Оскільки ваш поштовий сервер цілком і повністю перебуває під вашим контролем, ви самі розпоряджаєтесь правилами його використання. Самі вирішуєте, який повинен бути обсяг ящика в кожного користувача, самі вибираєте максимальний розмір поштового повідомлення, самі набудовуєте параметри спама й вирішуєте яка пошта потрібна, яка ні. При яких-небудь проблемах з поштою, ви можете самі їх вирішити.

4. При використанні хостинга ви найчастіше повинні задовольнитися тим, що вам дали без можливості зміни тих або інших параметрів. А при виникненні проблем з поштою, вам доводиться годинниками додзвонюватися до служби підтримки, які спочатку обіцяють усе перевірити, потім через години, коли ви дзвоните в другий раз, вас повідомляють, що все перевірили й у них все працює (хоча найчастіше це не так) і т.д., а проблема залишається не вирішеною.

5. Зниження витрат. Поштовий сервер ви здобуваєте раз і назавжди, а за послуги хостинга доводиться платити щорічно або щомісяця. При використанні свого поштового сервера внутрішня пошта не виходить за межі вашої локальної мережі, отже, ви не платите провайдеру за інтернет трафік.

Основне завдання поштового клієнта – одержання із сервера нових повідомлень і відправлення написаних користувачем. Для цього програма

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

повинна підтримувати відповідні протоколи: POP3 або IMAP для одержання й SMTP для відправлення. У випадку якщо пошта зберігається на сервері, призначеному для забезпечення спільної роботи співробітників підприємства необхідна і їхня підтримка. Деякі поштові сервіси дозволяють використовувати для доставки пошти тільки захищені з'єднання; для роботи з такими сервісами програма повинна підтримувати протоколи шифрування TLS і SSL .

Важливо, щоб отримані повідомлення відображалися коректно. Це значить, що клієнт не повинен помилятися при визначенні кодування тексту й повинен «розуміти» листа у форматі HTML (саме він використовується для збереження форматування тексту електронних листів). Крім того, необхідно й наявність інструментів для створення повідомлень у форматі HTML .

Неодмінний атрибут будь-якого поштового клієнта – адресна книга. Вона повинна дозволяти зберігати не тільки e-mail-адреси, але й додаткову контактну інформацію: номери телефонів (домашні, робочі й мобільного), поштові адреси (домашній і робітник), адреса особистої веб-сторінки й т.ін. Важлива й можливість імпорту й експорту записів для забезпечення схоронності інформації при переході на інший комп'ютер.

Крім того, необхідні інструменти для підвищення безпеки переписки. Гарним помічником буде вбудований у програму спам-фільтр, а інтеграція з антивірусними програмами дозволить «на лету» перевіряти прикріплені до повідомлень файли. При відправленні конфіденційної інформації не зайвої виявиться можливість шифрування листів.

Крім роботи з кореспонденцією, більшість клієнтів для роботи з електронною поштою надає користувачам і інші можливості. Часто в поштових клієнтах передбачена функція завантаження новин по протоколі NNTP: робота з ними здійснюється точно так само, як і із вхідними повідомленнями. Небагато відрізняється процес завантаження стрічок новин у форматах RSS і Atom. Якщо e-mail-клієнт підтримує ці протоколи, він цілком підійде й на роль новинного агрегатора.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Оскільки поштовий клієнт, як правило, працює постійно, він легко впорається з функціями іншого програмного забезпечення першої необхідності, наприклад календаря-органайзера. Зустрічаються й комбіновані програми для роботи в Інтернеті, що сполучають у собі веб-браузер, поштовий клієнт, агрегатор новин, менеджер завантажень і інші модулі.

Поштовий клієнт повинен мати наступні достоїнства:

- будь-яка кількість обслуговуваних одночасно POP3 аккаунтів на різних серверах;
- багатомовний інтерфейс із можливістю перемикання кириличного, українського і багато інших (після завантаження мовного модуля із сервера);
- диспетчер листів для маніпуляцій з листами безпосередньо на POP3 сервері;
- потужні засоби фільтрації листів;
- абсолютно коректна робота з усіма кириличними і європейськими кодуваннями, причому гнучка система налаштувань дозволяє встановлювати свої кодування для кожного ящика й кожного адресата;
- можливість використовувати індивідуальні шаблони листів, створені для кожного ящика, кожної папки й кожного адресата, з урахуванням мови, форми вітання, національних кодувань і т.д. Все це управляється системою макросів і дуже легко програмується;
- система фільтрів, макросів і шаблонів є настільки потужною, що дозволяє перетворити поштовий клієнт у сервер списку розсилання. Програма сама буде перевіряти потрібні аккаунти, фільтрувати, модифікувати поля From, To, і т.д., модифікувати зміст повідомлень і потім розсилати їхнім потрібним адресатам списку;
- повна багатозадачність і багатопоточність – програма може перевіряти аккаунти, вести сортування повідомлень і одночасно дозволяє переглядати й редагувати листа;
- потужні засоби пошуку;

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- зручна й дуже потужна записна книжка з фотографіями;
- убудована підтримка PGP, що базується на бібліотеці SSLeay;
- підтримка смарт-карт із секретним ключем PGP – для поштових транзакцій типу "банк-клієнт" і інших сфер, де потрібна підвищена таємність;
- можливість імпорту повідомлень із форматів найбільш популярних поштових клієнтів (Outlook, Netscape Communicator, Netscape Mail, Eudora, Pegasus Mail), а також поштових скриньок Unix;
- підтримка протоколів IMAP4, POP, APOP, SMTP;
- можливість закриття доступу до окремих ящиків паролем;
- убудовані функції поштового сервера локальної мережі (TCP/IP – Non-TCP/IP шлюз);
- форми запиту – інструмент для формування запитів, що підлягають автоматичній обробці. Потужний засіб керування електронним документообігом у бізнесі;
- запис журналів роботи для кожного ящика;
- зручний текстовий редактор, автоматична перевірка орфографії кількома мовами, внутрішня функція дозвона, що біжить рядок "Mail Ticker", і безліч інших корисних функцій, опис яких зайняло б ще багато місця й часу.

Структурна схема розробленого, у ході виконання дипломного проектування, програмного забезпечення зображена на рисунку 3.1.

Програмне забезпечення поштового клієнту складається з наступних структурних блоків:

- Двигун забезпечення поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL – основне ядро на якому будується поштовий клієнт.
- Блок авторизації та автентифікації – призначений для здійснення входу до поштової скриньки згідно логіну та паролю.
- Блок захисту інформації, фільтрів та антиспаму – призначений для забезпечення конфіденційного електронного документообігу по протоколам

POP3 та SMTP та захисту від спаму, тобто листів усілякої реклами, які не потрібні користувачу поштової скриньки.

– Диспетчер листів – призначений для управління отриманими, написаними, відправленими листами та чернеток листів.

– Блок підтримки протоколів:

а) POP3, IMAP – для одержання електронних листів;

б) SMTP – для відправлення електронних листів.

– БД адресної книги – призначена для зберігання адрес куди відправлені листи, та звідкіля отримані листи.

– БД прийнятих листів – призначена для зберігання листів надісланих користувачеві поштового клієнта.

– БД відправлених листів – призначена для зберігання листів відправлених користувачем поштового клієнта.

– БД чернеток листів – призначена для зберігання чернеток листів написаних користувачем поштового клієнта;

– Редактор листів у форматі HTML – призначений для написання та редагування текстових листів у вигляді HTML.

– Редактор листів згідно стандарту MIME – призначений для написання та редагування листів у вигляді тексту, картинок, та інших прикріплених до листа файлів.

– Блок налаштування інтерфейсу – призначений для індивідуального налаштування інтерфейсу під конкретного користувача.

– RSS-служба – призначена для реалізації служби новин, як поштового клієнта так і глобальних.

Надамо опис структури поштового повідомлення, яке оброблюється цим клієнтом.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

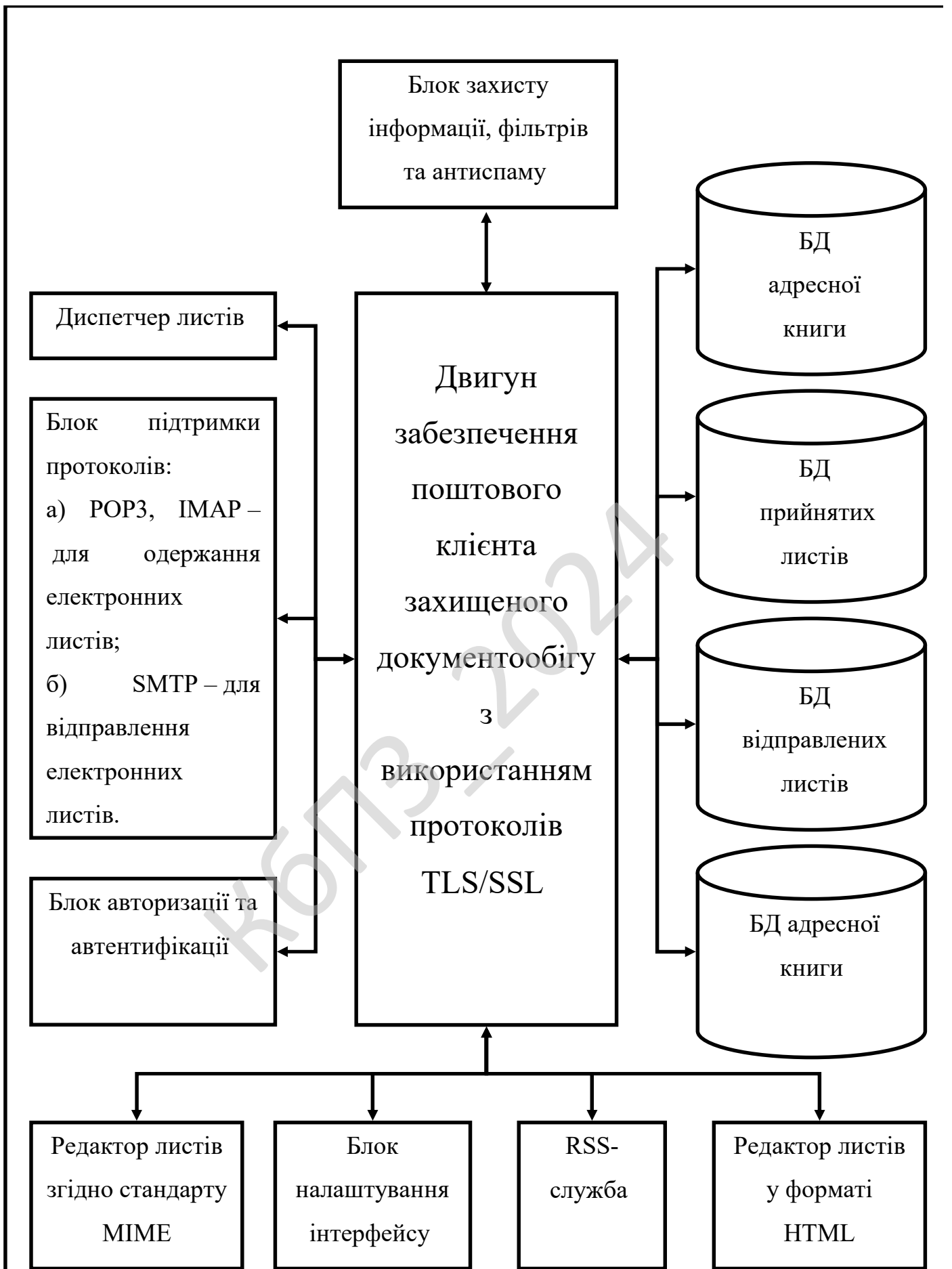


Рисунок 3.1 – Структурна схема розробленої системи

– Bcc – поле "Схована копія". Містить адреси додаткових одержувачів повідомлення. Одержувачі, перераховані в полях "To" і "Cc", не будуть знати, що абоненти зі списку "Bcc" одержали копію повідомлення.

У всіх полях, призначених для вказівки адрес одержувачів ("To", "Cc" і "Bcc") може вказуватися як одна, так і кілька адрес. Поля можуть також бути відсутні у повідомленні. Однак очевидно, що для відправлення повідомлення повинен бути зазначена хоча б одна адреса одержувача в кожному із цих полів:

– Reply-to – поле "Відповісти". Містить адресу, по якому одержувач повинен направляти відповідь. Це поле є необов'язковим: у випадку його відсутності відповіді направляються за адресою, зазначеному в поле "From".

– Subject – поле "Тема повідомлення". У цьому полі звичайно вказується короткий опис (тема) повідомлення.

Тіло повідомлення

Споконвічно передбачалося, що поштові повідомлення можуть містити тільки текст у форматі ASCII. А оскільки можливість передачі нетекстової інформації не передбачалася, то протоколи передачі електронної пошти можуть некоректно обробляти такі повідомлення. У зв'язку із цим у свій час був розроблений спеціальний стандарт, що визначає принципи перетворення нетекстових даних до текстового виду. Цей стандарт одержав назву MIME (Multipurpose Internet Mail Extension, багатоцільове розширення пошти Інтернет).

MIME припускає, що в тілі повідомлення можуть передаватися наступні види інформації:

– текст – простий текст у форматі ASCII, а також текст у форматі RTF або HTML;

– графічні зображення – файли у форматі JPEG і GIF;

– аудіо й відео дані;

– дані у форматах різних додатків, наприклад, документи Microsoft Office, а також дані довільного формату (у тому числі, різні файли, що виконуються).

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

В одному поштовому повідомленні можуть бути передані дані різних типів. Такі повідомлення являють собою структуру із загальним заголовком і декількома блоками усередині тіла, кожний з яких містить інформацію свого типу.

Це широко використовується при відправленні повідомлень із вкладеннями (attachments) – додатковими "прикріпленими" файлами, які можуть містити різноманітну інформацію. Наприклад, до текстового повідомлення можна прикріпити графічний файл з фотографією відправника.

Крім того, це може виявитися корисним, коли текст повідомлення необхідно передати в різних форматах. Наприклад, відправлене повідомлення у форматі HTML, що містить яке-небудь оформлення, може некоректно сприйматися клієнтською програмою одержувачів. Щоб уникнути подібних проблем поштовий клієнт відправника може сформувати альтернативне подання повідомлення у вигляді простого тексту.

Для забезпечення коректної передачі повідомлень з нетекстовими даними в MIME передбачені два алгоритми перекодування, що здійснюють перетворення таких даних до тестового виду:

- алгоритм "Quoted-printable", призначений для заміни байтів, що не є ASCII-символами, на групу із трьох байт, що представляють собою тільки стандартні символи;
- алгоритм "Base64", що здійснює перетворення трьох довільних байт у чотири ASCII-символи.

Для забезпечення коректної інтерпретації даних стандартом MIME у заголовках повідомлення вводяться додаткові спеціальні поля:

- Content-type – поле "Тип вмісту". Відповідає за коректне визначення типу даних, що втримуються в повідомленні заголовка повідомлення. Значення поля вказує на конкретний тип даних, або інформує про те, що тіло містить кілька різноманітних блоків.

– Content-Transfer-Encoding – поле "Тип кодування вмісту". Визначає спосіб перетворення (перекодування) вихідних даних у текстовий вид.

На рисунку 3.2 зображена функціональна схема розробленого поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Розглянемо більш докладно основні функціональні можливості розробленого поштового клієнту.

До основних функцій розробленого поштового клієнта відносяться наступні:

- Перевірка протоколів автентифікації.
- Шифрування даних SSL/TLS.
- Перевірка на віруси.
- Фільтрація.
- Автоматичний запуск зовнішнього додатка.
- Автоматичне відсилання підтвердження про прочитання листа.
- Архівування/розархівування листа.
- Витягування приєднаних даних MIME.
- Використання шаблонів.
- Списки розсилки.
- Резервне копіювання листів.
- Підтримка кодування.
- Диспетчер листів.
- Адресна книга.
- Імпорт з інших поштових клієнтів.
- Імпорт адресних книг з файлів.

Розглянемо більш докладно вищеперераховані функції.

Безпека особистої інформації

Відмітна риса поштового клієнта – сама широка підтримка протоколів автентифікації й шифрування при роботі з поштовими серверами.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36



Рисунок 3.2 – Функціональна схема системи

Захист від вірусів

Друга сторона безпеки – захист від шкідливих кодів. На відміну від інших поштових програм, поштовий клієнт не запускає скрипти автоматично. Поштовий клієнт використовує власний механізм перегляду HTML – Robin HTML viewer, власний модуль перегляду зображень, що захищає вас від вірусів, спрямованих на уразливості механізмів операційної системи. Користувачам Windows 8/10 повідомляємо, що, на додаток до вищевказаних можливостей, поштовий клієнт задіє такі компоненти Windows 8/10, як механізм Випадкового Розміщення в Адресному Просторі (ASLR) і механізм Запобігання Виконання Даних (DEP) для захисту комп'ютера від вірусів і шкідливих кодів, розповсюджуваних через електронну пошту. У поштовий клієнт вбудований URL менеджер для фонового добування зображень у форматі HTML. Цей менеджер успішно бореться зі шкідливими кодами, що вставляються в зображення.

Фільтрація

Для обробки кореспонденції застосовується вбудований Сортувальник листів, що автоматично розподіляє вхідну, вихідну, прочитану й оброблену пошту по відповідних папках. Завдання правил сортування займе всього кілька хвилин, після чого ви можете назавжди забути про ручне перекладання листів з папки в папку.

Автоматична обробка повідомлень

Вищезгаданий Сортувальник листів дозволяє автоматизувати обробку повідомлень: настроїти автовідповідь, запуск зовнішнього додатка, додати адресатів в адресну книгу, експортувати або архівувати повідомлення, витягти приєднані файли й багато чого іншого.

Шаблони

Поштовий клієнт не тільки може додати підпис до повідомлення, але й дозволяє створювати шаблони листів, використовуючи безліч спеціальних макросів. Будь-який ящик, папка або адресат можуть мати власний шаблон для нового листа, відповіді або пересилання. Застосування шаблонів дозволяє істотно

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

прискорити підготовку пошти, особливо коли лист містить у собі стандартний або часто повторюваний текст.

Швидкі шаблони

Швидкі шаблони дозволяють автоматизувати процес створення листів і заощадити робочий час, вставляючи в лист заздалегідь набраний текст. Якщо щодня вам доводиться відповідати в листах на ті самі питання, ви по достоїнству оціните ці шаблони. У швидких шаблонах, так само як і у звичайних, доступний повний набір макросів, що дозволяє задати будь-яку, навіть саму складну послідовність дій.

Списки розсилання

Система фільтрації листів настільки потужна, що дозволяє реалізувати засобами поштового клієнта список розсилання, де учасники можуть самостійно додавати або видаляти себе зі списку.

Унікальна функція паркування

За допомогою паркування ви зможете закріпити лист за певною папкою й зберігати його від випадкового видалення або переміщення. Ви ніколи не втратите важливий лист, якщо припаркуєте його в підходящій папці.

Резервне копіювання

Всі настроювання програми, адресні книги й поштові папки можуть бути збережені в окремому файлі й потім відновлені з резервної копії у випадку небажаної зміни інформації або втрати даних. Архів можна постачити коментарем і захистити паролем. Крім того, поштовий клієнт дозволяє створювати окрему резервну копію для кожної поштової скриньки.

Відновлення й установка

Поштовий клієнт повністю сумісний з новим Менеджером Перезавантажень (RM) Windows 8/10 – ви зможете оновлювати компоненти Windows, необхідні для роботи поштового клієнта, не перезавантажуючи ні поштовий клієнт, ні Windows. Ніколи ще процес відновлення не був таким простим і непомітним. Що стосується можливостей установки, системні

						ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			39

адміністратори (і не тільки вони) по достоїнству оцінять нову можливість “непомітної” установки, що дозволяє інсталювати поштової клієнт без участі користувача.

Ефективне шифрування

Внутрішнє виконання PGP, засноване на загальноновизнаному механізмі OpenSSL, дозволяє вам шифрувати повідомлення й підписувати їхнім цифровим підписом. Зручний менеджер цифрових ключів додається. Безкоштовні доповнення до PGP v5.5, v6.0.2, 6.5, v7.x & v8.x GnuPG також підтримуються. Крім того, Поштової клієнт працює з S/MIME за допомогою Internal Implementation або Microsoft CryptoAPI. Поштової клієнт – перший поштової клієнт, що підтримує стиск даних S/MIME. Поштової клієнт працює з Secure Socket Layer (SSL) v3.1 / Transport Layer Security (TLS) v1.0 з алгоритмом AES.

Синхронізація пошти

Ви можете синхронізувати вміст декількох екземплярів поштової клієнта. Синхронізація пошти дозволяє підтримувати дві копії поштової клієнта в однаковому стані. Ви можете вибірково зберігати настроювання поштової скриньок, папок, адресних книг і прикріплених файлів.

Сумісність із Microsoft Exchange

Поштової клієнт підключається до серверів Microsoft Exchange, задіючи протокол MAPI для виклику або відправлення повідомлень. Необхідно лише встановити Microsoft Office Outlook або клієнт Microsoft Exchange для надання компонентів підключення до Exchange, буде використовувати поштової клієнт.

Національні кодування

Поштової клієнт коректно працює з усіма кодуваннями, включаючи всі українські й східноєвропейські: KOI 8-R, Windows-1251, DOS, ISO і т.д. Гнучка система настроювань дозволяє встановлювати свої кодування для кожного ящика й кожного адресата. Крім того, починаючи з версії Поштової клієнт підтримує кодування Unicode. Також підтримуються всі системні шрифти й набори символів.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Диспетчер листів

Диспетчер дозволяє обробляти пошту на POP3 або IMAP сервері, не чекаючи її завантаження. Чудовий засіб для здійснення віддаленого контролю над вашою кореспонденцією. Особливо корисно в тому випадку, якщо необхідно залишити об'ємні листи на сервері, щоб прочитати їх пізніше, або видалити листи, не одержуючи їх.

Адресна книга

Адресна книга містить список ваших адресатів. Ви можете поєднувати їх у групи для класифікації, або для використання адрес як список розсилання, тобто відправити лист відразу всім членам групи. До кожного адресного запису можна приєднати фотографію, указати особисті дані, відомості про місце роботи, шаблони листів до даного адресата, список S/MIME-сертифікатів абонента, а також кодування за замовчуванням. У поштовому клієнті присутня опція Історії адрес, що дозволяє переглядати й швидко знаходити будь-яке повідомлення кожного з кореспондентів.

Інтерфейс кількома мовами

Поштовий клієнт дозволяє змінювати мову інтерфейсу «на льоту», без перезавантаження програми.

Модуль перегляду зображень

Поштовий клієнт оснащений власним модулем перегляду зображень. Це означає, що доступ до вкладених зображень набагато швидше, ніж через зовнішній додаток. Користувачі можуть перемикатися між всіма зображеннями, вкладеними в лист. Модуль підтримує поворот, алгоритми зміни розміру й масштабу, повноекранний режим.

Форми запиту

Форми являють собою інструмент для формування запитів, що підлягають автоматичній обробці. Форма запиту призначена для віддаленого доступу до інформації й керування їй у режимі оффлайн, витримуючи вимоги безпеки.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Багаті можливості імпорту

Поштовий клієнт дозволяє імпортувати повідомлення з інших поштових клієнтів:

- Microsoft Outlook Express v4.xx.
- Microsoft Outlook Express v5.0.
- Netscape Communicator v4.xx.
- Netscape Mail v2.xx/3.xx.
- Eudora Lite/Pro.
- Pegasus Mail v2.xx або 3.xx.

Імпорт адресних книг з файлів:

- LDIF-файл.
- Бізнес-картка VCard.
- Текст, розділений комами.
- Текст, розділений табуляціями.
- INI-файл.
- Адресна книга поштового клієнта.
- Адресна книга Eudora.
- Файл-шаблон Pegasus.

Працює з будь-якою системою Windows.

TLS/SSL

TLS (англ. Transport Layer Security – захист на транспортному рівні), як і його попередник SSL – криптографічний протокол, що надає можливості безпечної передачі даних в Інтернет для навігації, отримання пошти, спілкування та передачі інших даних. Використовує асиметричне шифрування і сертифікати X.509. Існують незначні відмінності між TLS та SSL, але, в основі,

TLS надає можливості автентифікації і безпечної передачі даних через Інтернет з використанням криптографічних засобів. Часто відбувається лише автентифікація сервера, а клієнт залишається не автентифікованим. Для взаємної автентифікації кожна з сторін мусить підтримувати інфраструктуру відкритого

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

ключа (PKI), яка дозволяє захистити клієнт-серверні додатки від перехоплення, редагування повідомлень або ж створення підроблених.

SSL включає три основні фази:

- Діалог між сторонами, метою якого є вибір алгоритму шифрування.
- Обмін ключами на основі криптосистем з відкритим ключем або ж автентифікація на основі сертифікатів.
- Передача даних, що шифруються за допомогою симетричних алгоритмів шифрування

В даній поточній версії протоколу доступні такі алгоритми:

– Для обміну ключами і перевірки їх справжності використовують комбінації алгоритмів: RSA (асиметричний шифр), Diffie-Hellman (безпечний обмін ключами), DSA (алгоритм цифрового підпису) і алгоритми технології Fortezza.

– Для симетричного шифрування: RC2, RC4, IDEA, DES, Triple DES або AES.

– Для хеш-функцій: MD5 або SHA.

Алгоритми можуть доповнюватися в залежності від версії протоколу. Розглянувши всі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

– Система захищеного документообігу з використанням протоколів TLS/SSL.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

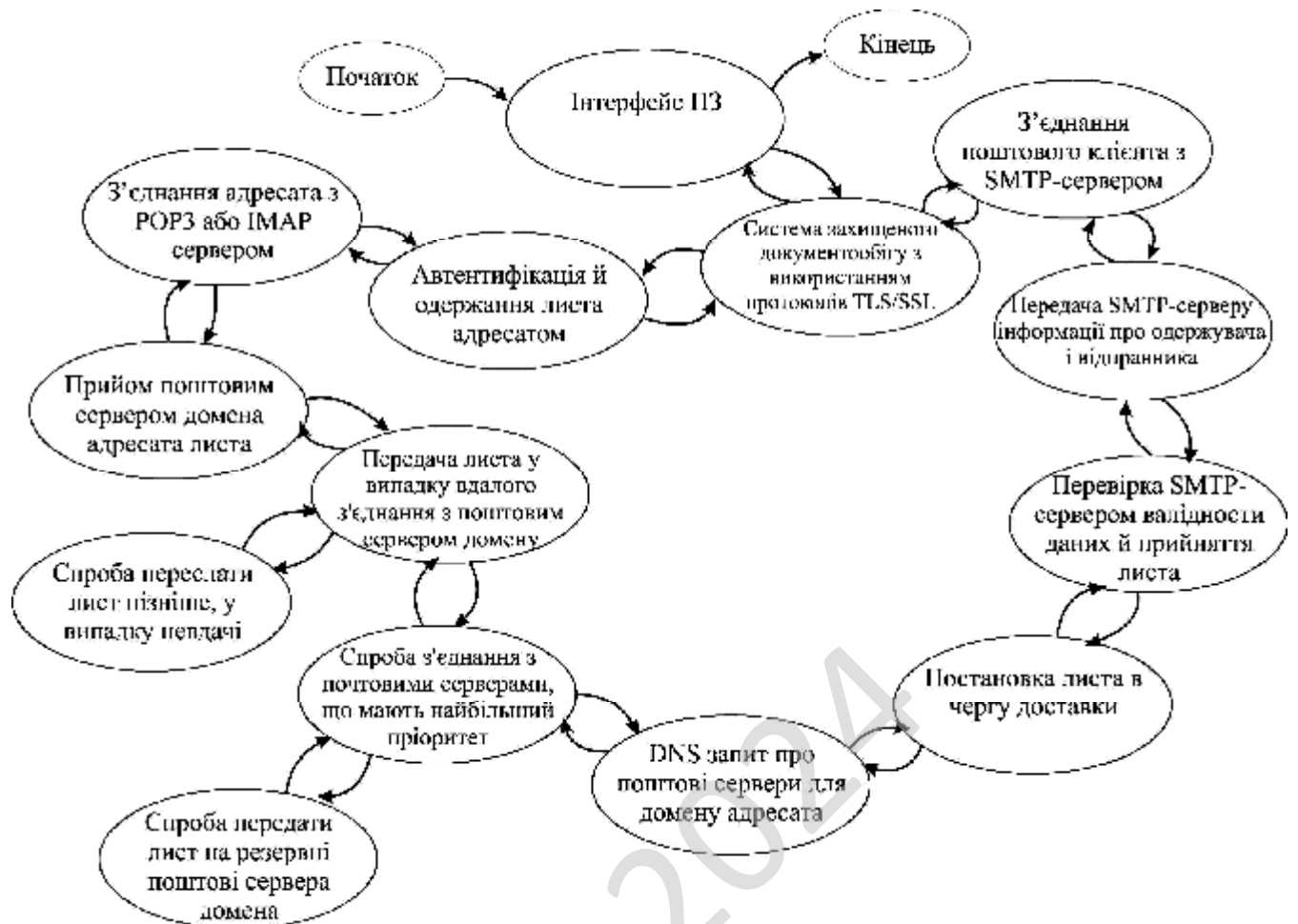


Рисунок 3.3 – Діаграма взаємодії процесів

- З'єднання поштового клієнта з SMTP-сервером.
- Передача SMTP-серверу інформації про одержувача і відправника.
- Перевірка SMTP-сервером валідності даних й прийняття листа.
- Постанова листа в чергу доставки DNS запит про поштові сервери для домену адресата.
- Спроба з'єднання з поштовими серверами, що мають найбільший пріоритет.
- Спроба передати лист на резервній поштовій сервера домену.
- Передача листа у випадку вдалого з'єднання з поштовим сервером домену адресата.
- Спроба переслати лист пізніше, у випадку невдачі.

- Прийом поштовим сервером домену адресата листа.
- З'єднання адресата з POP3 або IMAP сервером.
- Автентифікація й одержання листа адресатом.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ - 2024

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

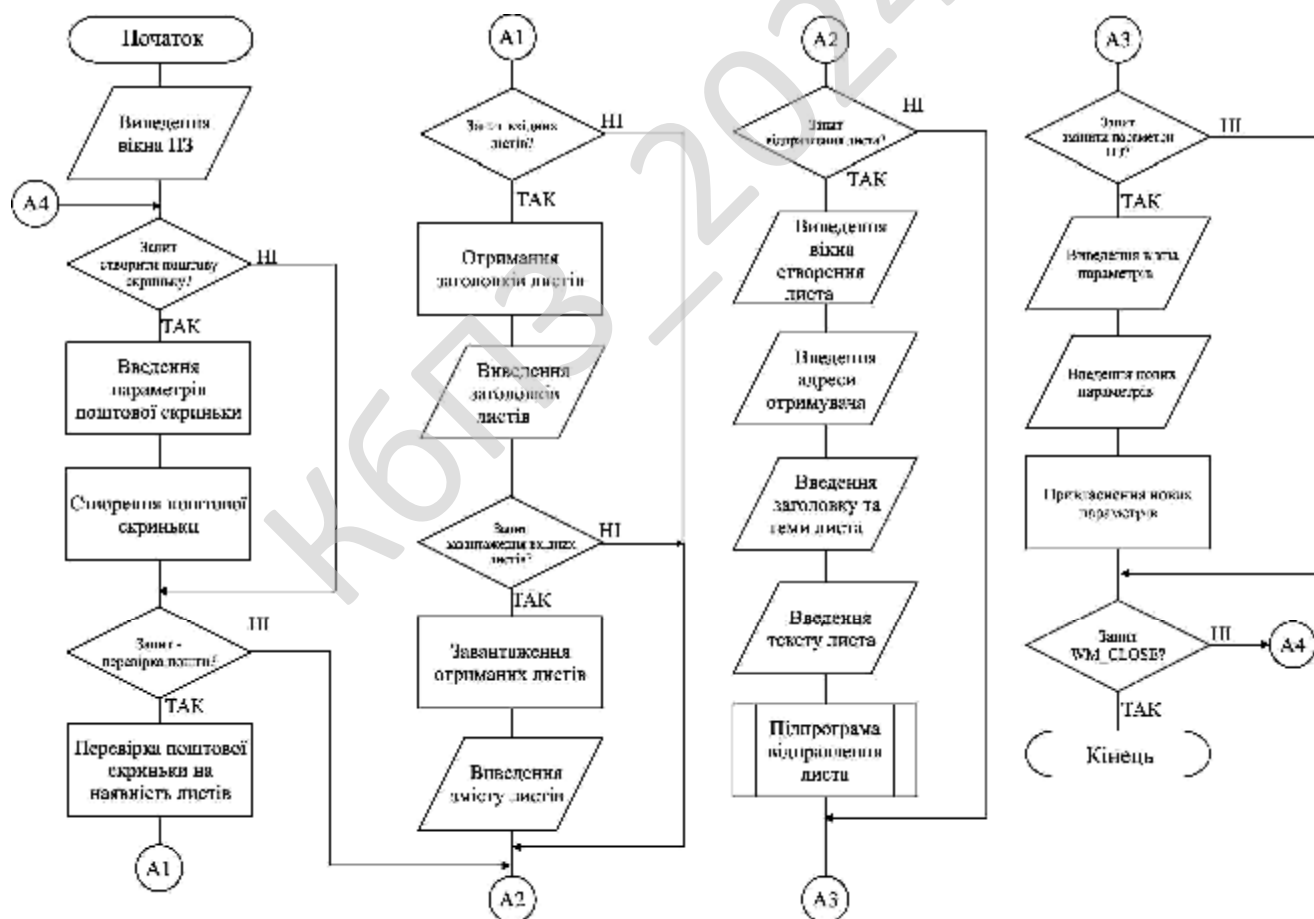


Рисунок 4.1 – Блок схема основної програми

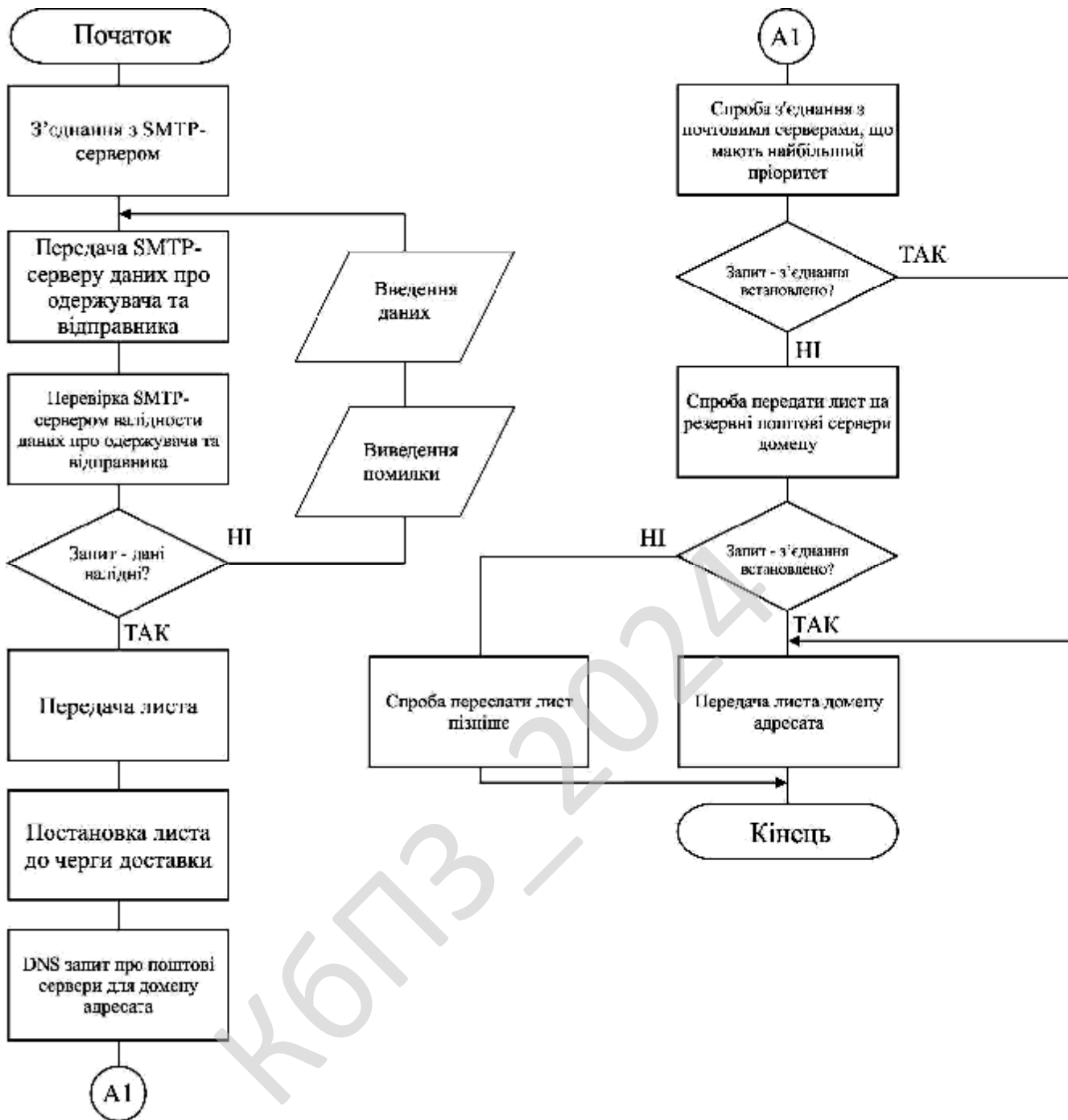


Рисунок 4.2 – Блок схема підпрограми

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Опис алгоритмів функціонування системи

Розглянемо вихідний код головного модуля, та його складових.

```
program MyMail_TLS_SSL;
// назва програми
Uses
// підключення бібліотек та модулів
  Forms,
  frmMain in 'frmMain.pas' {frm_Main},
// Головна
  DataModule in 'DataModule.pas' {DM: TDataModule},
// дані
frmMail_TLS_SSLSettings in 'frmMail_TLS_SSLSettings.pas',
// налаштування TLS_SSL
  frmSettings in 'frmSettings.pas' {frm_Settings},
// налаштування ПЗ
  LetterStorage in 'LetterStorage.pas',
// пошта
  frmNewLetter in 'frmNewLetter.pas' {frm_NewLetter},
// створення нового письма
  frmAskOnDuplicate in 'frmAskOnDuplicate.pas' {frm_AskOnDuplicate},
  about in 'about.pas' {Form1};
// авторські дані
{$R *.res}
// ресурси
Begin
  Application.Initialize;
// початок та ініціалізація
  Application.Title := 'Поштовий клієнт, розроблений Яценко О.Ю.';
// Титульна назва ПЗ
  Application.CreateForm(Tfrm_Main, frm_Main);
// створення форми frm_Main
  Application.CreateForm(TDM, DM);
// створення форми DM
Application.CreateForm(Tfrm_Mail_TLS_SSLSettings,
  frm_Mail_TLS_SSLSettings);
// створення форми frm_Mail_TLS_SSLSettings
  Application.CreateForm(Tfrm_Settings, frm_Settings);
```

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

// створення форми frm_Settings
Application.CreateForm(Tfrm_NewLetter, frm_NewLetter);
// створення форми frm_NewLetter
Application.CreateForm(Tfrm_AskOnDuplicate, frm_AskOnDuplicate);
// створення форми frm_AskOnDuplicate
Application.HintPause:=200;
Application.Run;
// початок роботи
end.

```

Розглянемо роботу з протоколом POP3 зі застосуванням протоколів TLS/SSL

POP3 – Post Office Protocol Version 3 – протокол поштової відділення, версія 3 – використовується поштовим клієнтом для одержання повідомлень електронної пошти із сервера. Звичайно використовується в парі із протоколом SMTP.

Попередні версії протоколу (POP, POP2) застаріли.

Стандарт протоколу POP3 визначений в RFC 1939. Розширення й методи авторизації визначені в RFC 2195, RFC 2449, RFC 1734, RFC 2222, RFC 3206, RFC 2595.

Існують реалізації POP 3-серверів, що підтримують TLS і SSL.

Альтернативним протоколом для збору повідомлень із поштового сервера є IMAP.

Опис станів сеансу

У протоколі POP3 передбачено 3 стану сеансу:

- Авторизація – Клієнт проходить процедуру автентифікації.
- Транзакція – клієнт одержує інформацію про стан поштової скриньки, приймає й видаляє пошту.
- Відновлення – Сервер видаляє обрані листи й закриває з'єднання.

Опис команд протоколу

1. APOP [ім'я] [digest] – Команда служить для передачі серверу ім'я користувача й зашифрованого пароля (digest).

Аргументи:

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- [ім'я] – рядок, що вказує ім'я поштової скриньки.
- [digest] – хеш-сума тимчасової мітки, конкатенированої з паролем користувача, обчислена по алгоритму MD5. У випадку підтримки цієї команди тимчасова мітка виходить при з'єднанні із сервером.

Обмеження:

Її підтримка не є обов'язковою.

Можливі відповіді:

- +OK maildrop has n message.
- -ERR password supplied for [ім'я] is incorrect.

2. USER [ім'я] – Передає серверу ім'я користувача.

Аргументи:

[ім'я] – рядок, що вказує ім'я поштової скриньки.

Обмеження:

Немає.

Можливі відповіді:

- +OK name is a valid mailbox.
- -ERR never heard of mailbox name.

3. PASS [пароль] – Передає серверу пароль поштової скриньки.

Аргументи:

[пароль] – пароль для поштової скриньки.

Обмеження:

Працює після успішної передачі ім'я поштової скриньки.

Можливі відповіді:

- +OK maildrop locked and ready.
- -ERR invalid password.
- -ERR unable to lock maildrop.

4. DELE [повідомлення] – Сервер позначає зазначене повідомлення для видалення. Повідомлення, позначені на видалення, реально віддаляються тільки після закриття транзакції (закриття транзакцій відбувається звичайно після посилання команди QUIT, крім цього, на серверах закриття транзакцій може відбуватися після закінчення певного часу, встановленого сервером).

Аргументи:

[повідомлення] – номер повідомлення.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK message deleted.
- -ERR no such message.

5. LIST [повідомлення] – Якщо був переданий аргумент, то сервер видає інформацію про зазначене повідомлення. Якщо аргумент не був переданий, то сервер видає інформацію про всі повідомлення, що перебувають у поштової скриньці. Повідомлення, позначені для видалення, не перераховуються.

Аргументи:

[повідомлення] – номер повідомлення (необов'язковий аргумент).

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK scan listing follows.
- -ERR no such message.

6. NOOP – Сервер нічого не робить, завжди відповідає позитивно.

Аргументи:

Немає.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK.

7. RETR повідомлення – Сервер передає повідомлення із зазначеним номером.

Аргументи:

повідомлення – номер повідомлення.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK message follows.
- -ERR no such message.

8. RSET – Цією командою виробляється відкрит транзакцій усередині сесії.

Наприклад, якщо користувач випадково позначив на видалення які-небудь повідомлення, він може забрати ці позначки, відправивши цю команду.

Аргументи:

Немає.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK.

9. STAT – Сервер повертає кількість повідомлень у поштової скриньці плюс розмір, займаними цими повідомленнями на поштової скриньці

Аргументи:

Немає.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK a b.

10 TOP [повідомлення] [кількість рядків] – Сервер повертає заголовки зазначеного повідомлення, порожній рядок і зазначену кількість перших рядків тіла повідомлення.

Аргументи:

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

- [повідомлення] – номер повідомлення.
- [кількість рядків] – скільки рядків потрібно вивести.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK n octets.
- -ERR no such message.

11. QUIT

Аргументи:

Немає.

Обмеження:

Немає.

Можливі відповіді:

- +OK

Розглянемо приклад сесії

Це приклад сесії з підтримкою зашифрованих паролів (APOP, RFC 1939):

```
S: <Сервер очікує вхідних з'єднань на порту 110>
C: <підключається до сервера>
S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <сервер передає повідомлення 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
```

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```
S: <сервер передає повідомлення 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <закриває з'єднання>
S: <продовжує чекати вхідні з'єднання>
```

Варіант початку сесії, при якому пароль передається відкритим текстом:

```
C: USER mrose
S +OK User accepted
C: PASS mrosepass
S +OK Pass accepted
```

Розглянемо роботу з протоколом IMAP зі застосуванням протоколів TLS/SSL

IMAP – Internet Message Access Protocol – Протокол доступу до електронної пошти Інтернету – протокол прикладного рівня для доступу до електронної пошти.

Аналогічно POP3, служить для роботи із вхідними листами, однак забезпечує додаткові функції, зокрема, можливість пошуку по ключовому слову без збереження пошти в локальній пам'яті.

IMAP надає користувачеві багаті можливості для роботи з поштовими скриньками, що перебувають на центральному сервері. Поштова програма, що використовує цей протокол, одержує доступ до сховища кореспонденції на сервері так, начебто ця кореспонденція розташована на комп'ютері одержувача. Електронними листами можна маніпулювати з комп'ютера користувача (клієнта) без постійного пересилання із сервера й назад файлів з повним змістом листів. Для відправлення листів використовується протокол SMTP.

Опис команди протоколу

Доступні в будь-якому стані:

- CAPABILITY.
- NOOP – порожня команда, використовується щоб не розірвати сесію по тайм-ауті.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

- LOGOUT.
- STARTTLS.

Доступні в неавтентифікованому стані:

- STARTTLS.
- AUTHENTICATE.
- LOGIN.

Доступні в автентифікованому стані:

- SELECT.
- EXAMINE – аналогічна SELECT, але доступ буде даний тільки для читання.

- CREATE – створити поштову скриньку.
- DELETE – видалити поштову скриньку.
- RENAME.
- SUBSCRIBE.
- UNSUBSCRIBE.
- LIST.
- LSUB.
- STATUS – інформація про поштову скриньку, у тому числі кількість

непрочитаних повідомлень.

- APPEND – додати повідомлення в поштову скриньку.

Доступні в стані selected:

- CHECK.
- CLOSE.
- EXPUNGE.
- SEARCH – пошук повідомлень у поштовій скриньці.
- FETCH – одержати повідомлення або його частину.
- STORE.
- COPY.

- UID.

Експериментальні/розширення:

- X<atom> – будь-які команди, що починаються з «X» – експериментальні.

Слід зазначити, що в ІМАР немає команди на переміщення повідомлення з одного ящика в іншій. Замість цього, потрібно зробити COPY, а потім відзначити повідомлення на видалення. Деякі ІМАР-сервери надають команди з розділу розширених, для переміщення повідомлення: наприклад, сервер GroupWise підтримує команду XGWMOVE.

Опис переваг в порівнянні з протоколом POP3

ІМАР був розроблений для заміни більше простого протоколу POP3 і має наступні переваги в порівнянні з останнім:

- Листи зберігаються на сервері, а не на клієнті. Можливий доступ до тієї самої поштової скриньки з різних клієнтів. Підтримується також одночасний доступ декількох клієнтів. У протоколі є механізми, за допомогою яких клієнт може бути проінформований про зміни, зроблених іншими клієнтами.
- Підтримка декількох поштових скриньок (або папок). Клієнт може створювати, видаляти й перейменовувати поштові скриньки на сервері, а також переміщати листа з однієї поштової скриньки в іншій.
- Можливе створення загальних папок, до яких можуть мати доступ кілька користувачів.
- Інформація про стан листів зберігається на сервері й доступна всім клієнтам. Листи можуть бути позначені як прочитані, важливі й т.п.
- Підтримка пошуку на сервері. Немає необхідності скачувати із сервера безліч повідомлень для того, щоб знайти одне потрібне.
- Підтримка онлайн-роботи. Клієнт може підтримувати із сервером постійне з'єднання, при цьому сервер у реальному часі інформує клієнта про зміни в поштових скриньках, у тому числі про нові листи.
- Передбачено механізм розширення можливостей протоколу.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Поточна версія протоколу має позначення IMAP4rev1 (IMAP, версія 4, ревізія 1). Протокол підтримує передачу пароля користувача в зашифрованому виді. Крім того, IMAP-трафік можна зашифрувати за допомогою SSL.

Опис системи

У програмі використовуються наступні протоколи передачі та приймання поштових повідомлень й листів:

- SMTP.
- POP3.
- IMAP.

Розглянемо ці протоколи більш детальніше.

Розглянемо роботу з протоколом SMTP зі застосуванням протоколів TLS/SSL

SMTP – Simple Mail Transfer Protocol – простий протокол передачі пошти – це мережний протокол, призначений для передачі електронної пошти в мережах TCP/IP.

ESMTP – Extended SMTP – масштабоване розширення протоколу SMTP. У цей час під «протоколом SMTP», як правило, мають на увазі ESMTP і його розширення.

Огляд протоколу

SMTP використовується для відправлення пошти від користувачів до серверів і між серверами для подальшого пересилання до одержувача. Для прийому пошти поштовий клієнт повинен використовувати протоколи POP3 або IMAP.

Щоб доставити повідомлення до адресата, необхідно переслати його поштовому серверу домену, у якому перебуває адресат. Для цього звичайно використовується запис типу MX – Mail eXchange – обмін поштою, системи DNS. Якщо MX запис відсутній, то для тих же цілей може бути використаний запис типу A. Деякі сучасні реалізації SMTP-серверів для визначення сервера, що

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

обслуговує пошту в домені адресата, також можуть задіяти SRV-запис (RFC 2782).

Широке поширення SMTP одержав на початку 1980-х років. До нього використовувався протокол UUCP, що жадав від відправника знання повного маршруту до одержувача і явної вказівки цього маршруту в адресі одержувача, або наявності прямого що комутирується або постійного з'єднання між комп'ютерами відправника й одержувача.

Sendmail був одним з перших (якщо не першим) агентом відправлення повідомлень, що почав працювати з SMTP. У цей час протокол SMTP є стандартним для електронної пошти і його використовують всі клієнти й сервери.

Протокол був розроблений для передачі тільки тексту в кодуванні ASCII, крім того, перші специфікації вимагали обнуління старшого біта кожного переданого байта. Це не дає можливості відсилати текст на національних мовах (наприклад, кирилиці), а також відправляти двійкові файли (такі як зображення, відеофайли, програми або архіви). Для зняття цього обмеження був розроблений стандарт MIME, що описує спосіб перетворення двійкових файлів у текстові. У цей час більшість серверів підтримують 8BITMIME, що дозволяє відправляти двійкові файли так само просто, як текст.

Сервер SMTP – це кінцевий автомат із внутрішнім станом. Клієнт передає на сервер рядок команда<пробіл>параметри<переклад рядка>. Сервер відповідає на кожну команду рядком, що містить код відповіді й текстове повідомлення, відділене пробілом. Код відповіді – число від 100 до 999, представлене у вигляді рядка, що трактується в такий спосіб:

- 2XX – команда успішно виконана;
- 3XX – очікуються додаткові дані від клієнта;
- 4XX – тимчасова помилка, клієнт повинен зробити наступну спробу через якийсь час;
- 5XX – непереборна помилка.


```
C:.  
S:250 769947 message accepted for delivery  
C:QUIT  
S:221 mail.company.tld CommuniGate Pro SMTP closing connection  
S: (закриває з'єднання)
```

У результаті такої сесії лист буде доставлено адресатові user1@company.tld, але не буде доставлено адресатові user2@company.tld, тому що такої адреси не існує.

Розширення ESMTP

RFC 1869 пропонує починати сесію не командою HELO, а командою EHLO. У випадку, якщо сервер не підтримує розширень, то він відповість на EHLO помилкою, у цьому випадку клієнт повинен послати команду HELO і не використовувати розширення протоколу.

Якщо ж сервер підтримує ESMTP, то крім вітання він повідомить список підтримуваних розширень протоколу SMTP, наприклад:

```
ehlo office.company1.tld  
250-mail.company2.tld is pleased to meet you  
250-DSN  
250-SIZE  
250-STARTTLS  
250-AUTH LOGIN PLAIN CRAM-MD5 DIGEST-MD5 GSSAPI MSN NTLM  
250-ETRN  
250-TURN  
250-ATRN  
250-NO-SOLICITING  
250-HELP  
250-PIPELINING  
250 EHLO
```

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SEED – у криптографії симетричний блоковий криптоалгоритм на основі Мережі Фейстеля, розроблений Корейським агентством інформаційної безпеки (Korean Information Security Agency, KISA) в

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

1998 році. В алгоритмі використовується 128-бітний блок і ключ довжиною 128 біт. Алгоритм одержав широке поширення й використовується фінансовими й банківськими структурами, виробничими підприємствами й бюджетними установами Південної Кореї, оскільки 40-бітний SSL не забезпечує на даний момент мінімально необхідного рівня безпеки. Агентством по захисту інформації специфіковане використання шифру SEED у протоколах TLS і S/MIME. У той же час, алгоритм SEED не реалізований у більшості сучасних браузерів і інтернет-додатків, що утрудняє його використання в даній сфері поза межами Південної Кореї.

SEED являє собою мережу Фейстеля з 16 раундами, 128-бітовими блоками й 128-бітовим ключем. Алгоритм використовує дві 8×8 таблиці підстановки, які, як такі з Safer, виведені з дискретного зведення в ступінь (у цьому випадку, x^{247} і x^{251} – плюс деякі «несумісні операції»). Це є деякою подібністю с MISTY1 у рекурсивності його структури: 128-бітовий повний шифр – мережа Фейстеля з F-функцією, що впливає на 64-бітові половини, у той час як сама F-функція – Мережа Фейстеля, складена з G-функції, що впливає на 32-розрядні половини. Однак рекурсія не простягнеться далі, тому що G-функція – не Мережа Фейстеля. В G-функції 32-розрядне слово розглядають як чотири 8-бітових байта, кожний з яких проходить через одну або іншу таблицю підстановки, потім поєднується в помірковано комплексному наборі булевих функцій таким чином, що кожний біт виводу залежить від 3 з 4 вхідних байтів.

SEED має складний ключовий розклад, генеруючи тридцять два 32-розрядних додаткових символу, використовуючи G-функції на серіях обертань вихідного неопрацьованого ключа, комбінованого зі спеціальними раундовими константами (як в TEA) від «Золотого співвідношення» (англ. Golden ratio).

Згідно з дослідженнями KISA, алгоритм SEED «надійно протистоїть відомим атакам».

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

– Розділ меню: Лист; Скриньки; Параметри; Про програму.

– Розділ опису листів папки: Тема; Від кого; Дата; Розмір; Пріоритет; Примітки; Пратори.

– Розділ папок: Вхідні – для прийому кореспонденції; Вихідні – для посланої Вами, але ще не обробленої кореспонденції; Відправлені – для відправленої й обробленої кореспонденції; Видалені – для вилучених листів, які перебувають у цій папці, на випадок якщо буде необхідність все-таки відновити їх.

– Розділ інформаційної панелі.

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

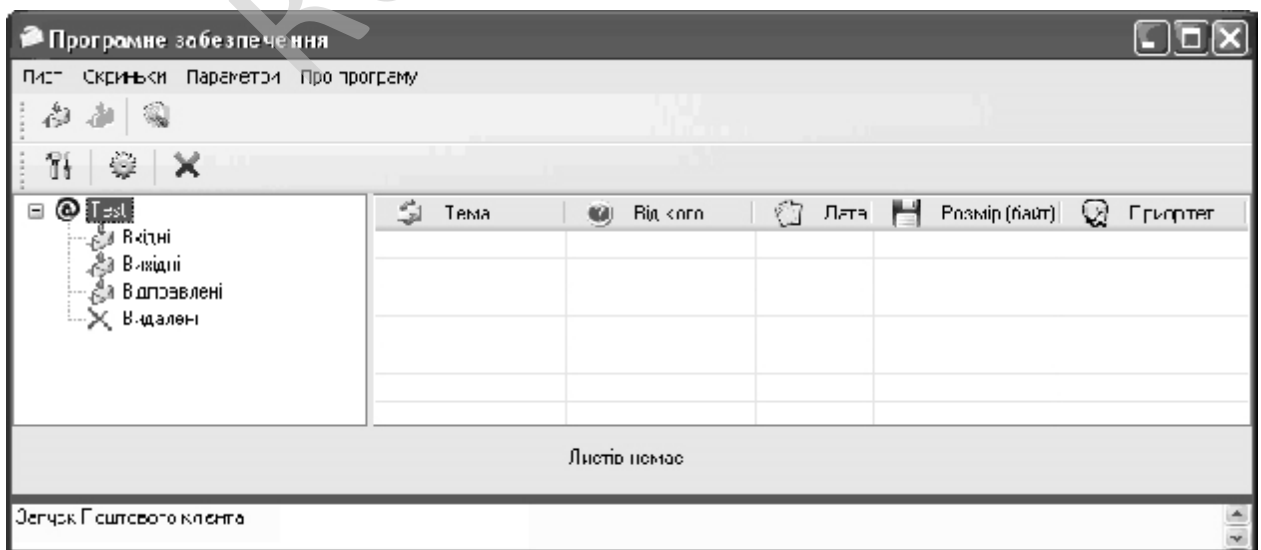


Рисунок 5.1 – Головне вікно програми

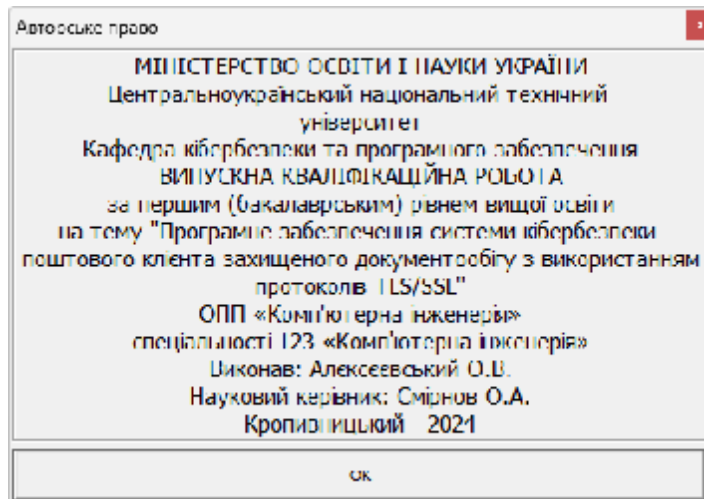


Рисунок 5.2 – Авторське право

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів. Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення. Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються. Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

– Досліджена система кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL. 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 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SEED.

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-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
2. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
3. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
4. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
5. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p.
6. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
7. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
8. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.
9. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
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. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.
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,

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

13. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskiy, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) *Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

14. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

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., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages 44-58.

19. 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.

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

21. 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.

22. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiyчук A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

23. 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.

24. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

25. Smirnov O., Kuznetsov A., Onikiyчук 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.

26. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

27. 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.

28. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko 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.

29. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 125-136.

30. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.

31. 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.

32. 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.

33. 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.

34. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during

Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

35. 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.

36. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

37. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

38. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P. 707-712.

39. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.701-706.

40. Smirnov, O., Hu, Z., Vasiliu, Y., Sydorenko, V., Polishchuk, Y., «Abstract Model of Eavesdropper and Overview on Attacks in Quantum Cryptography Systems», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.399-405.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

41. 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.

42. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019, Lviv, Ukraine, 2-6 July, 2019*, P. 129-134.

43. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

44. 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.

45. 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.

46. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 873-884.

47. 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.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

48. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

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

50. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

51. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

					ВКРБ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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-123.24.0001.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Алексєєвський О.В.				Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-20		
Затв.	Смірнов О.А.						
					Програмне забезпечення системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL		

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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 кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL.

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-123.24.0001.00.00.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 кібербезпеки поштового клієнта захищеного документообігу з використанням протоколів TLS/SSL;
- 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-123.24.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРБ-123.24.0001.00.00.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 – 72 аркуші.

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-123.24.0001.00.00.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 4.06.2024 р.

					ВКРБ-123.24.0001.00.00.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

_____ Смірнов О.А.

*Програмне забезпечення системи кібербезпеки поштового клієнта
захищеного документообігу з використанням протоколів TLS/SSL*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 50

Літера: РП

Кропивницький – 2024 року

Файл MyMail_TLS_SSL.dpr - файл проекту

```
program MyMail_TLS_SSL;

uses
  Forms,
  frmMain in 'frmMain.pas' {frm_Main},
  DataModule in 'DataModule.pas' {DM: TDataModule},
  frmMail_TLS_SSLSettings in 'frmMail_TLS_SSLSettings.pas'
{frm_Mail_TLS_SSLSettings},
  frmSettings in 'frmSettings.pas' {frm_Settings},
  LetterStorage in 'LetterStorage.pas',
  frmNewLetter in 'frmNewLetter.pas' {frm_NewLetter},
  frmAskOnDuplicate in 'frmAskOnDuplicate.pas' {frm_AskOnDuplicate},
  about in 'about.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Поштовий клієнт, розроблений Алексеевський Олексій  
Владиславович';
  Application.CreateForm(Tfrm_Main, frm_Main);
  Application.CreateForm(TDM, DM);
  Application.CreateForm(Tfrm_Mail_TLS_SSLSettings, frm_Mail_TLS_SSLSettings);
  Application.CreateForm(Tfrm_Settings, frm_Settings);
  Application.CreateForm(Tfrm_NewLetter, frm_NewLetter);
  Application.CreateForm(Tfrm_AskOnDuplicate, frm_AskOnDuplicate);
  Application.CreateForm(TForm1, Form1);
  Application.HintPause:=200;
  Application.Run;
end.
```

Файл frmMain.pas - основна програма

```

unit frmMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ToolWin, ComCtrls, ExtCtrls, StdCtrls, XPMan,
  IdBaseComponent, IdComponent, IdTCPConnection, IdTCPClient,
  IdMessageClient, IdPOP3, ActnList, OleCtrls, SHDocVw,
  {HTTPApp, {HTTPProd,}
  LetterStorage, frmMail_TLS_SSLSettings, CoolTrayIcon, about;

type
  Tfrm_Main = class(TForm)
    MainMenu: TMainMenu;
    Splitter1: TSplitter;
    Splitter2: TSplitter;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    CoolBar: TCoolBar;
    Tree_Mail_TLS_SSLBoxes: TTreeView;
    ListView_LettersHeaders: TListView;
    ToolBar1: TToolBar;
    btn_CheckMail_TLS_SSL: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    XPManifest1: TXPManifest;
    Panel_Content: TPanel;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    ActionList1: TActionList;
    aCheckMail_TLS_SSL: TAction;
    aSendMail_TLS_SSL: TAction;
    aMail_TLS_SSLSettings: TAction;
    aExit: TAction;
    ToolBar2: TToolBar;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    aNewLetter: TAction;
    ToolButton7: TToolButton;
    MemoLog: TMemo;
    Splitter3: TSplitter;
    aSettings: TAction;
    ToolButton8: TToolButton;
    ToolButton9: TToolButton;
    N11: TMenuItem;
    Web_LetterViewer: TWebBrowser;
    Splitter4: TSplitter;
    List_Attachmant: TListView;
    Memo_LetterBody: TMemo;
    Popup_Attachment: TPopupMenu;
    mnuSaveAtt: TMenuItem;
    SaveDialog_Att: TSaveDialog;
    PopupMenuHeaders: TPopupMenu;
    mnu_DeleteLetter: TMenuItem;
    mnu_Answer: TMenuItem;
    mnu_Resend: TMenuItem;
    N12: TMenuItem;
  end;

```

```

aAnswer: TAction;
aResend: TAction;
PopupMenu_Mail_TLS_SSLBoxes: TPopupMenu;
N13: TMenuItem;
Jnghfdbnmgjxnel: TMenuItem;
N14: TMenuItem;
About1: TMenuItem;
procedure aMail_TLS_SSLSettingsExecute(Sender: TObject);
procedure aExitExecute(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure aNewLetterExecute(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure aSettingsExecute(Sender: TObject);
procedure Tree_Mail_TLS_SSLBoxesClick(Sender: TObject);
procedure aCheckMail_TLS_SSLExecute(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure List_AttachmantClick(Sender: TObject);
procedure Web_LetterViewerBeforeNavigate2(Sender: TObject; const pDisp:
IDispatch; var URL, Flags, TargetFrameName, postData, Headers: OleVariant; var
Cancel: WordBool);
    procedure ListView_LettersHeadersSelectedItem(Sender: TObject; Item:
TListItem; Selected: Boolean);
    procedure List_AttachmantMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
    procedure mnuSaveAttClick(Sender: TObject);
    procedure Popup_AttachmentPopup(Sender: TObject);
    procedure mnu_DeleteLetterClick(Sender: TObject);
    procedure PopupMenuHeadersPopup(Sender: TObject);
    procedure aAnswerExecute(Sender: TObject);
    procedure aResendExecute(Sender: TObject);
    procedure Tree_Mail_TLS_SSLBoxesDragDrop(Sender, Source: TObject; X, Y:
Integer);
    procedure Tree_Mail_TLS_SSLBoxesDragOver(Sender, Source: TObject; X, Y:
Integer; State: TDragState; var Accept: Boolean);
    procedure TrayClick(Sender: TObject);
    procedure TrayMinimizeToTray(Sender: TObject);
    procedure N13Click(Sender: TObject);
    procedure PopupMenu_Mail_TLS_SSLBoxesPopup(Sender: TObject);
    procedure JnghfdbnmgjxnelClick(Sender: TObject);
    procedure About1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    procedure ShowMail_TLS_SSLBoxesInTree;
    procedure LoadLetters;
    procedure AddToLog(s : shortstring);
    procedure HideLetterBody;
    procedure ShowInHtmlViewer(body : TStrings);
end;

const
    LogFileName    = 'Actions.log';

var
    frm_Main: Tfrm_Main;

    Mail_TLS_SSLInfoLoaded : boolean=false;

    { Головне сховище електронних листів }
    Mail_TLS_SSLBoxes      : TMail_TLS_SSLBoxList;

function ExecuteFile(const FileName, Params, DefaultDir: string; ShowCmd:
Integer): THandle;
procedure FillPOPwithMail_TLS_SSLBoxSettings(mbSettings :
PMail_TLS_SSLBoxSettings);

```

```

implementation

uses shellapi, frmSettings, IdMessage,
    DataModule, frmNewLetter,
    ComObj, frmAskOnDuplicate, IdEmail_TLS_SSLAddress;

{$R *.dfm}

//параметри облікових записів
procedure Tfrm_Main.aMail_TLS_SSLSettingsExecute(Sender: TObject);
begin
    frm_Mail_TLS_SSLSettings.Show;
end;

// вихід
procedure Tfrm_Main.aExitExecute(Sender: TObject);
begin
    halt;
end;

procedure Tfrm_Main.FormShow(Sender: TObject);
{var
    IE: Variant;
begin
    IE := CreateOleObject('InternetExplorer.Application');
    IE.Navigate('C:\MyHTML.html');
    While IE.Busy do begin end;
    Memo1.Text:= IE.Document.Body.innerText;
}end; }
begin
    if HideCaret(frm_Main.Handle) then Application.MessageBox('', '');
    if not Mail_TLS_SSLInfoLoaded then
        begin
            { Show Mail_TLS_SSLboxes }

            ShowMail_TLS_SSLBoxesInTree;
            LoadLetters;
            Mail_TLS_SSLInfoLoaded:=true;

            Memo_LetterBody.Visible:=false;
            Web_LetterViewer.Visible:=false;
            if Settings.ShowInTray then
                begin
                    Tray.IconVisible:=true;
                    Tray.HideTaskbarIcon;
                    frm_Main.Tray.MinimizeToTray:=true;
                end
            else
                begin
                    Tray.IconVisible:=false;
                    Tray.ShowMainForm;
                    frm_Main.Tray.MinimizeToTray:=false;
                end;
            end;
            frm_AskOnDuplicate.ShowModal;
            if frm_AskOnDuplicate.ModalResult = mrIgnore then
                Application.MessageBox('', '');
        end;

procedure Tfrm_Main.ShowMail_TLS_SSLBoxesInTree;
var
    i      : word;
    node   : TTreeNode;
begin
    Tree_Mail_TLS_SSLBoxes.Items.Clear;
    for i:=1 to TotalMail_TLS_SSLBoxes do

```

```

begin
node:=Tree_Mail_TLS_SSLBoxes.Items.Add(nil,Mail_TLS_SSLBoxesSettings[i].Name);
node.ImageIndex:=0;
Tree_Mail_TLS_SSLBoxes.Items.AddChild(node,'Вхідні').ImageIndex:=1;
Tree_Mail_TLS_SSLBoxes.Items.AddChild(node,'Вихідні').ImageIndex:=2;
Tree_Mail_TLS_SSLBoxes.Items.AddChild(node,'Відправлені').ImageIndex:=2;
Tree_Mail_TLS_SSLBoxes.Items.AddChild(node,'Видалені').ImageIndex:=3;

AddToLog('Завантажені параметри скриньки
<'+Mail_TLS_SSLBoxesSettings[i].Name+'>');
end;
AddToLog('Всього облікових записів : '+inttostr(TotalMail_TLS_SSLBoxes));
end;

procedure Tfrm_Main.LoadLetters;
var
i : word;
Begin
Mail_TLS_SSLBoxes.TotalBoxes:=0;
for i:=1 to TotalMail_TLS_SSLBoxes do
begin
Mail_TLS_SSLBoxes.Add(Mail_TLS_SSLBoxesSettings[i].Name);
AddToLog('Завантажені листи для скриньки
<'+Mail_TLS_SSLBoxesSettings[i].Name+'>');
end;
End;

// написати листа
procedure Tfrm_Main.aNewLetterExecute(Sender: TObject);
var
Mail_TLS_SSLBoxName : shortstring;
begin
frm_NewLetter.ClearForm;
frm_NewLetter.Show;

if Tree_Mail_TLS_SSLBoxes.Selected=nil then
begin
Application.MessageBox('Ви повинні виділити яку-небудь скриньку, щоб
відправити з неї листа','З якої поштової скриньки відправляти?');
exit;
end;

if Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil then
Mail_TLS_SSLBoxName:=Tree_Mail_TLS_SSLBoxes.Selected.Text
else
Mail_TLS_SSLBoxName:=Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
frm_NewLetter.Caption:='Новий лист <'+Mail_TLS_SSLBoxName+'>';
end;

function ExecuteFile(const FileName, Params, DefaultDir: string; ShowCmd:
Integer): THandle;
var
zFileName, zParams, zDir: array[0..79] of Char;
begin
Result := ShellExecute(Application.MainForm.Handle, nil,
StrPCopy(zFileName, FileName), StrPCopy(zParams, Params),
StrPCopy(zDir, DefaultDir), ShowCmd);
end;

procedure Tfrm_Main.AddToLog(s : shortstring);
var
st : TFileStream;
SLog : shortstring;
Mess : shortstring;
mem : TMemoryStream;

```

```

Begin
  if (s[1]=#10)and(s[2]=#13) then
    begin
      delete(s,1,2);
      Mess:=#10#13+DateTimeToStr(Now)+' -=> '+s+#10#13;
      end
    else Mess:=DateTimeToStr(Now)+' -=> '+s+#10#13;

MemoLog.Lines.Add(Mess);

if Settings.AllowLog then
begin
  SLog:=ExtractFilePath(Application.ExeName)+LogFileName;
  try
    st:=TFileStream.Create(SLog,fmOpenReadWrite or fmShareDenyNone);
  except
    try
      st:=TFileStream.Create(SLog,fmCreate);
    except
      Application.MessageBox( pchar('Не могу відкрити файл "'+SLog+'" для
запису !'), 'Помилка збереження лог файлу');
      System.exit;
    end;
  end;

  { Перевіряємо, якщо виходимо за границю MaxLogSize }
  { та обрізаємо лог файл у цьому випадку }
  { в випадку вибору більшого MaxLogSize значення ми можемо уповільнити ти
програму}
  { так як кожний раз при записі лог файл буде копіюватися до пам'яті та
обратно }
  if (Settings.MaxLogSize<>0)and(st.Size > Settings.MaxLogSize * 1024) then
    try
      mem:=TMemoryStream.Create;
      mem.Size:=Settings.MaxLogSize*1024;
      mem.seek(0,soFromBeginning);

      st.Seek(-Settings.MaxLogSize*1024,soFromEnd);
      mem.CopyFrom(st,Settings.MaxLogSize*1024);
      st.Seek(0,soFromBeginning);
      mem.Seek(0,soFromBeginning);
      st.CopyFrom(mem,Settings.MaxLogSize*1024);
      st.Size:=Settings.MaxLogSize*1024;
      finally mem.Free; end;

    st.Seek(0,soFromEnd);
    st.Write(Mess[1],length(Mess));
    st.Free;
  end;
End;

procedure DeleteFiles(s : shortstring);
var
  sr : TSearchRec;
Begin
  try
    findfirst(s, faAnyFile, sr);
    deletefile( IncludeTrailingPathDelimiter(ExtractFilePath(s))+sr.Name );
    while findnext(sr)=0 do deletefile(
IncludeTrailingPathDelimiter(ExtractFilePath(s))+sr.Name );
  except end;
End;

procedure Tfrm_Main.FormCreate(Sender: TObject);
begin
  frm_Settings.LoadGlobalSettingsFromFile;

```

```

DeleteFiles (
IncludeTrailingPathDelimiter(ExtractFilePath(Application.ExeName))+ 'temp_*.html'
);

AddToLog(#10#13' *** Запуск Поштового клієнта'#10#13);
end;

// параметри програми

procedure Tfrm_Main.aSettingsExecute(Sender: TObject);
begin
    frm_Settings.show;
end;

procedure Tfrm_Main.Tree_Mail_TLS_SSLBoxesClick(Sender: TObject);
var
    Mail_TLS_SSLBoxName      : shortstring;
    Mail_TLS_SSLBoxFolder    : shortstring;
    LetterList                : PLetters;
    i                         : word;
begin
    if Tree_Mail_TLS_SSLBoxes.Selected = nil then
    begin
        HideLetterBody;
        exit;
    end;
    List_Attachmant.Clear;
    List_Attachmant.Visible:=false;
    Memo_LetterBody.Clear;
    if Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil then
    begin
        if
Mail_TLS_SSLBoxes.FindBox(Tree_Mail_TLS_SSLBoxes.Selected.Text).IsBuisyNow then
aCheckMail_TLS_SSL.Enabled:=false
                                                    else
aCheckMail_TLS_SSL.Enabled:=true;
        ListView_LettersHeaders.Clear;
        HideLetterBody;
        findMail_TLS_SSLBoxAndSelectIt(Tree_Mail_TLS_SSLBoxes.Selected.Text);
        exit;
    end;

    Mail_TLS_SSLBoxName      := Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
    Mail_TLS_SSLBoxFolder   := Tree_Mail_TLS_SSLBoxes.Selected.Text;

LetterList:=Mail_TLS_SSLBoxes.FindLetters(Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFol
der);

    ListView_LettersHeaders.Items.Clear;
    if LetterList = nil then
    begin
        Panel_Content.Caption:='Немає такої папки.'+
        ' Або вона була завантажена з помилкою. ';
        exit;
    end;

    if Mail_TLS_SSLBoxes.FindBox(Mail_TLS_SSLBoxName).IsBuisyNow then
aCheckMail_TLS_SSL.Enabled:=false
                                                    else
aCheckMail_TLS_SSL.Enabled:=true;
        findMail_TLS_SSLBoxAndSelectIt(Mail_TLS_SSLBoxName);

    for i:=1 to LetterList^.Count do
    with ListView_LettersHeaders.Items.Add do
    begin

```

```

Caption:=LetterList^.Letters[i]^Subject;
if LetterList^.Letters[i]^Name='' then
SubItems.Add(LetterList^.Letters[i]^From.Address)
else
SubItems.Add(LetterList^.Letters[i]^From.Name);
SubItems.Add(DateTimeToStr(LetterList^.Letters[i]^Date));
//Позміп
SubItems.Add( inttostr(LetterList^.LettersSizes[i]) );
case LetterList^.Letters[i]^Priority of
  mpHighest : SubItems.Add('Дуже високий');
  mpHigh    : SubItems.Add('Високий');
  mpNormal  : SubItems.Add('Звичайний');
  mpLow     : SubItems.Add('Низький');
  mpLowest  : SubItems.Add('Дуже низький');
end;
end;

end;

procedure FillPOPwithMail_TLS_SSLBoxSettings(mbSettings :
PMail_TLS_SSLBoxSettings);
Begin
  DM.POP.Host :=mbSettings.POPServer;
  DM.POP.Port :=mbSettings.POPPort;
  DM.POP.Username :=mbSettings.POPAccount;
  DM.POP.Password :=mbSettings.POPPass;
End;

(*
procedure RetrieveMail_TLS_SSL(let_count : word);
var
  mes : TIdMessage;
  i : word;
  LettersNot2beDownloaded : word;
Begin
  try
  LettersNot2beDownloaded:=0;
  for i:=1 to let_count do
  if CurrentMail_TLS_SSLBoxSettings^.RetrieveOnlyHeaders then
  // Отримати лише заголовки
  begin
  mes:=TIdMessage.Create(frm_Main);
  DM.POP.RetrieveHeader(i,mes);
  end
  else
  // Отримати вес лист
  begin
  mes:=TIdMessage.Create(frm_Main);
  DM.POP.RetrieveHeader(i,mes);
  if
Mail_TLS_SSLBoxes.GetByName(CurrentMail_TLS_SSLBoxSettings^.Name).HasLetterWithI
D(mes.MsgId) then
  begin
  inc(LettersNot2beDownloaded);
  continue;
  end;
  if
Mail_TLS_SSLBoxes.GetByName(CurrentMail_TLS_SSLBoxSettings^.Name).hadLetterWithI
d(mes.msgId) then
  begin
  case CurrentMail_TLS_SSLBoxSettings.ifDuplicate of
    Ignore : continue;
    Recieve : ;
    Ask : begin
  frm_AskOnDuplicate.FillInInfo(@mes);
  frm_AskOnDuplicate.ShowModal;
  case frm_AskOnDuplicate.ModalResult of
    mrIgnore : continue; // не отримувати

```

```

mrOk      : ;           // отримувати
mrAbort   : begin      // видаляти з сервера

                if not DM.POP.Delete(i) then
frm_Main.AddToLog('Помилка видалення листа як отриманого раніше, але видаленого
з комп'ютера (див. параметри) Від: "'+mes.From.Address+'" Тема:
"' +mes.Subject+'"');

                continue;
                end;

            end;

        DeleteFromServer : begin
                if not DM.POP.Delete(i) then
frm_Main.AddToLog('Помилка видалення листа як отриманого раніше, але видаленого
з комп'ютера (див. параметри) Від: "'+mes.From.Address+'" Тема:
"' +mes.Subject+'"');

                continue;
                end;

            end;

        end;

        frm_Main.AddToLog('Отримані листи
('+inttostr(i)+'/'+inttostr(let_count)+' ) от '+mes.From.Text+' по Теме
: "'+mes.Subject+'"');
        DM.POP.Retrieve(i,mes);

Mail_TLS_SSLBoxes.GetByName(CurrentMail_TLS_SSLBoxSettings^.Name).Inbox.Add(@mes
,false, DM.POP.RetrieveMsgSize(i));
        if CurrentMail_TLS_SSLBoxSettings^.DeleteFromServer then
            if not DM.POP.Delete(i) then frm_Main.AddToLog('Помилка видалення листа
з сервера. Від: "'+mes.From.Address+'" Тема: "'+mes.Subject+'"');
            end;
            finally {mes.Free;} end;

        if LettersNot2beDownloaded<>0 then
            frm_Main.AddToLog('На сервері лежить(ать)' + inttostr
(LettersNot2beDownloaded) + 'листи, які не були завантажені помому що вони були
завантажені раніше і лежать в папках поточної скриньки');
        End;
        *)

procedure Tfrm_Main.aCheckMail_TLS_SSLExecute(Sender: TObject);
var
    CurMail_TLS_SSLBox : PMail_TLS_SSLBox;
begin
    // Перевірити пошту
    Tree_Mail_TLS_SSLBoxesClick(self);
    if CurrentMail_TLS_SSLBoxSettings =nil then exit;

    aCheckMail_TLS_SSL.Enabled:=false;

    CurMail_TLS_SSLBox:=Mail_TLS_SSLBoxes.GetByName(CurrentMail_TLS_SSLBoxSettings^.
Name);
    if CurMail_TLS_SSLBox.IsBuisyNow then exit;
    try
        // Імітація переривання таймера для ініціювання процесу перевірки пошти
        CurMail_TLS_SSLBox.onTimer_Check(self);
    except
        on e:exception do Application.MessageBox(pchar(E.Message), 'Помилка при
запуску служби перевірки скриньки');
    end;

end;

procedure Tfrm_Main.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Mail_TLS_SSLBoxes.SaveMail_TLS_SSLBoxes;

```

```

end;

procedure Tfrm_Main.HideLetterBody;
begin
  Memo_LetterBody.Hide;
  Web_LetterViewer.Hide;
end;

procedure Tfrm_Main.List_AttachmantClick(Sender: TObject);
var
  Mail_TLS_SSLBoxName      : shortstring;
  Mail_TLS_SSLBoxFolder    : shortstring;
  LetterList                : PLetters;
begin
  if Tree_Mail_TLS_SSLBoxes.Selected = nil then exit;
  if Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil then
  begin
    findMail_TLS_SSLBoxAndSelectIt(Tree_Mail_TLS_SSLBoxes.Selected.Text);
    exit;
  end;
  if (ListView_LettersHeaders.Selected=nil)or
    (List_Attachmant.Selected=nil)  then exit;

  Mail_TLS_SSLBoxName      := Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
  Mail_TLS_SSLBoxFolder    := Tree_Mail_TLS_SSLBoxes.Selected.Text;

  LetterList:=Mail_TLS_SSLBoxes.FindLetters(Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFolder);

  if
  LetterList.Letters[ListView_LettersHeaders.ItemIndex+1]^ .MessageParts.Items[List
_Attachmant.ItemIndex].ContentType = 'text/plain' then
    // text/plain
    begin
      Web_LetterViewer.Hide;
      Memo_LetterBody.Clear;

      Memo_LetterBody.Lines.AddStrings(TIdText(LetterList^.Letters[ListView_LettersHea
ders.ItemIndex+1]^ .MessageParts.Items[List_Attachmant.ItemIndex]).Body);
      Memo_LetterBody.Show;
    end;
  if
  LetterList.Letters[ListView_LettersHeaders.ItemIndex+1]^ .MessageParts.Items[List
_Attachmant.ItemIndex].ContentType = 'text/html' then
    // text/html
    begin
      Memo_LetterBody.Hide;

      ShowInHtmlViewer(TIdText(LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]
)^ .MessageParts.Items[List_Attachmant.ItemIndex]).Body);
      Web_LetterViewer.Show;
    end;
end;

// Вхід: рядок з кодом HTML; Вихід: в Web_LetterViewer
procedure Tfrm_Main.ShowInHtmlViewer(body: TStrings);
var
  tempFileName  : shortstring;
  st             : TFileStream;
  i              : integer;
  s              : string;
begin
  if not CurrentMail_TLS_SSLBoxSettings^.OpenHTMLasTEXT then
  BEGIN

```

```

try
    randomize;

tempFileName:=IncludeTrailingPathDelimiter(ExtractFilePath(Application.ExeName))
+'temp_'+inttostr(random(999))+'.html';
    deletefile(tempFileName);
    st:=TFileStream.Create(tempFileName, fmCreate);
except
    on E:Exception do Application.MessageBox(pchar('Тимчасовий файл
    "'+tempFileName+'" Не може бути створений :=> '+e.message), 'Помилка створення
    тимчасового файлу');
end;

    st.Seek(0, soFromBeginning);
    for i:=0 to body.Count-1 do
    begin
        s:=body.Strings[i]+#10#13;
        st.Write(s[1], length(s) );
    end;
    st.Free;

    Web_LetterViewer.Navigate('file://'+tempFileName);
END
ELSE
BEGIN
    Web_LetterViewer.Hide;
    Memo_LetterBody.Clear;
    Memo_LetterBody.Lines.AddStrings(Body);
    Memo_LetterBody.Show;
END;
end;

procedure Tfrm_Main.Web_LetterViewerBeforeNavigate2(Sender: TObject;
const pDisp: IDispatch; var URL, Flags, TargetFrameName, PostData,
Headers: OleVariant; var Cancel: WordBool);
begin
    try
        if (pos('temp_', URL)=0) and (URL[2]<>':') then
            Cancel:=true;
    except end;
end;

procedure Tfrm_Main.ListView_LettersHeadersSelectItem(Sender: TObject;
Item: TListItem; Selected: Boolean);
var
    Mail_TLS_SSLBoxName      : shortstring;
    Mail_TLS_SSLBoxFolder    : shortstring;
    LetterList                : PLetters;
    intIndex                  : integer;
    BodyPart                  : word;
begin
    if Tree_Mail_TLS_SSLBoxes.Selected = nil then exit;
    if Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil then
        begin
            findMail_TLS_SSLBoxAndSelectIt(Tree_Mail_TLS_SSLBoxes.Selected.Text);
            exit;
        end;
    if ListView_LettersHeaders.Selected=nil then exit;

    Mail_TLS_SSLBoxName      := Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
    Mail_TLS_SSLBoxFolder    := Tree_Mail_TLS_SSLBoxes.Selected.Text;

    LetterList:=Mail_TLS_SSLBoxes.FindLetters(Mail_TLS_SSLBoxName, Mail_TLS_SSLBoxFolder);

    //Список листів

```

```

    if pos('TEXT/HTML' ,
ANSIUPPERCASE (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.ContentType))<>0 then
    begin
        Memo_LetterBody.Hide;

ShowInHtmlViewer (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.Body)
;
    Web_LetterViewer.Show;
    end
    else
    begin

Memo_LetterBody.Lines:=LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^
.Body;
    Memo_LetterBody.visible:=true;
    end;

    BodyPart:=0;
    List_Attachmant.Items.Clear;
    for intIndex := 0 to
Pred (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageParts.Count) do
    begin
        if
(LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageParts.Items[intIndex] is TIdAttachment) then
            begin //загальні вкладення
                List_Attachmant.visible := true;
                with List_Attachmant.Items.Add do
                    begin
                        StateIndex:=0;
                        Caption :=
TIdAttachment (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageP
arts.Items[intIndex]).Filename;

SubItems.Add (TIdAttachment (LetterList^.Letters[ListView_LettersHeaders.ItemIndex
+1]^.MessageParts.Items[intIndex]).ContentType);
                    end;
                end
            else
                begin //текст листа
                    if
LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageParts.Items[int
Index] is TIdText then
                        begin
                            List_Attachmant.visible := true;
                            inc (BodyPart);
                            with List_Attachmant.Items.Add do
                                begin
                                    StateIndex:=1;
                                    if
AnsiUppercase (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageP
arts.Items[intIndex].ContentType) = 'TEXT/PLAIN'
                                        then Caption:='Частина #' +inttostr (BodyPart)
                                        else
                                            If
AnsiUppercase (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageP
arts.Items[intIndex].ContentType) = 'TEXT/HTML'
                                                then Caption:='HTML #' +inttostr (Bodypart)
                                                else
Caption:=LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.MessageParts.
Items[intIndex].ContentType;
                                    if
LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.Body.Count>0 then
                                        if
(AnsiUppercase (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^.Body.St

```



```

    if
LetterList.Letters[ListView_LettersHeaders.ItemIndex+1]^ .MessageParts.Items[List
_Attachmant.ItemIndex] is TIdAttachment then
    begin
        SaveDialog_Att.FileName:=List_Attachmant.selected.Caption;
        if not isValidFileName(SaveDialog_Att.FileName) then
SaveDialog_Att.FileName:='Вкладення '+inttostr(random(99))+'.txt';
        if not SaveDialog_Att.Execute then exit;

TIdAttachment(LetterList.Letters[ListView_LettersHeaders.ItemIndex+1]^ .MessagePa
rts.Items[List_Attachmant.ItemIndex]).SaveToFile(SaveDialog_Att.FileName);
        end else Application.MessageBox('Ця частина листа не є додатком!', 'Це не
допустимо');

End;

procedure Tfrm_Main.Popup_AttachmentPopup(Sender: TObject);
begin
    if List_Attachmant.Selected = nil then mnuSaveAtt.Enabled:=false
        else mnuSaveAtt.Enabled:=true;
end;

procedure Tfrm_Main.mnu_DeleteLetterClick(Sender: TObject);
var
    Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFolder : shortstring;
    LetterList : PLetters;
begin
    if ( Tree_Mail_TLS_SSLBoxes.Selected=nil ) or
        ( Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil ) then exit;
    if ListView_LettersHeaders.Selected = nil then exit;

    Mail_TLS_SSLBoxName := Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
    Mail_TLS_SSLBoxFolder := Tree_Mail_TLS_SSLBoxes.Selected.Text;

LetterList:=Mail_TLS_SSLBoxes.FindLetters(Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFol
der);

Mail_TLS_SSLBoxes.FindBox(Mail_TLS_SSLBoxName).MoveToGarbage(LetterList,ListView
_LettersHeaders.ItemIndex+1);
    ListView_LettersHeaders.DeleteSelected;
end;

procedure Tfrm_Main.PopupMenuHeadersPopup(Sender: TObject);
var
    i : word;
begin
    if ListView_LettersHeaders.Selected=nil then
        for i:=0 to PopupMenuHeaders.Items.Count-1 do
            PopupMenuHeaders.Items[i].Enabled:=false
        else for i:=0 to PopupMenuHeaders.Items.Count-1 do
            PopupMenuHeaders.Items[i].Enabled:=true;
end;

procedure Tfrm_Main.aAnswerExecute(Sender: TObject);
var
    Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFolder : shortstring;
    LetterList : PLetters;
begin
    Mail_TLS_SSLBoxName := Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
    Mail_TLS_SSLBoxFolder := Tree_Mail_TLS_SSLBoxes.Selected.Text;

LetterList:=Mail_TLS_SSLBoxes.FindLetters(Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFol
der);

    frm_NewLetter.ClearForm;

```

```

frm_NewLetter.Ed_Reciever.Text:=LetterList.Letters[ListView_LettersHeaders.itemindex+1].From.Text;
  frm_NewLetter.Ed_Subject.Text:='Re:
'+LetterList.Letters[ListView_LettersHeaders.itemindex+1].Subject;
  frm_NewLetter.Caption:='Новий лист с <'+Mail_TLS_SSLBoxName+'> [відповідь]';
  frm_NewLetter.Show;
end;

procedure Tfrm_Main.aResendExecute(Sender: TObject);
var
  Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFolder : shortstring;
  LetterList : PLetters;
  i : integer;
begin
  Mail_TLS_SSLBoxName := Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
  Mail_TLS_SSLBoxFolder := Tree_Mail_TLS_SSLBoxes.Selected.Text;

LetterList:=Mail_TLS_SSLBoxes.FindLetters(Mail_TLS_SSLBoxName,Mail_TLS_SSLBoxFolder);

  frm_NewLetter.ClearForm;
  frm_NewLetter.Ed_Subject.Text:='Fw:
'+LetterList.Letters[ListView_LettersHeaders.itemindex+1].Subject;
  frm_NewLetter.Caption:='Новий лист с <'+Mail_TLS_SSLBoxName+'> [пересилка]';
  frm_NewLetter.Show;

  frm_NewLetter.Memo_Body.Lines.Add('<--- Пересилаємий лист (початок) --->');
  for i:=0 to
LetterList.Letters[ListView_LettersHeaders.itemindex+1].Body.Count-1 do
    frm_NewLetter.Memo_Body.Lines.Add('>
'+LetterList.Letters[ListView_LettersHeaders.itemindex+1]^Body.Strings[i]);
  frm_NewLetter.Memo_Body.Lines.Add('<--- Пересилаємий лист (кінець) --->');

end;

procedure Tfrm_Main.Tree_Mail_TLS_SSLBoxesDragDrop(Sender, Source: TObject; X,
Y: Integer);
var
  node : TTreeNode;
  SourceMail_TLS_SSLBoxName : shortstring;
  SourceFolderName : shortstring;
  SourceLetters : PLetters;

  DestMail_TLS_SSLBox,
  DestFolderName : shortstring;
  DestLetters : PLetters;

begin
  node:=Tree_Mail_TLS_SSLBoxes.GetNodeAt(x,y);
  if (node = nil)or(node.Parent = nil) then exit;
  if Tree_Mail_TLS_SSLBoxes.Selected.Parent= nil then
SourceMail_TLS_SSLBoxName:=Tree_Mail_TLS_SSLBoxes.Selected.Text
  else
SourceMail_TLS_SSLBoxName:=Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
  SourceFolderName:=Tree_Mail_TLS_SSLBoxes.Selected.Text;

SourceLetters:=Mail_TLS_SSLBoxes.FindLetters(SourceMail_TLS_SSLBoxName,SourceFolderName);

  DestFolderName:=node.Text;
  DestMail_TLS_SSLBox:=node.Parent.Text;

DestLetters:=Mail_TLS_SSLBoxes.FindLetters(DestMail_TLS_SSLBox, DestFolderName);
  DestLetters.Add(SourceLetters.Letters[ListView_LettersHeaders.ItemIndex+1],
false,SourceLetters.LettersSizes[ListView_LettersHeaders.ItemIndex+1]);

```

```

SourceLetters.DeleteLetterWithoutFreeing(ListView_LettersHeaders.ItemIndex+1);
    Tree_Mail_TLS_SSLBoxesClick(self);
end;

procedure Tfrm_Main.Tree_Mail_TLS_SSLBoxesDragOver(Sender, Source: TObject; X,
    Y: Integer; State: TDragState; var Accept: Boolean);
begin
    if not (Source is TListView) then exit;
    if ListView_LettersHeaders.GetItemAt(x,y) = nil then exit;
    if (Tree_Mail_TLS_SSLBoxes.Selected =
nil)or(Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil) then exit;
    Accept:=true;
end;

procedure Tfrm_Main.TrayClick(Sender: TObject);
begin
    // Tray.ShowMainForm;
end;

procedure Tfrm_Main.TrayMinimizeToTray(Sender: TObject);
begin
    if Settings.ShowInTray then
    begin
        // frm_Main.Tray.IconVisible:=true;
        // frm_Main.Tray.HideTaskbarIcon;
    end
    else
    begin
        //// frm_Main.Tray.IconVisible:=false;
        // frm_Main.Tray.ShowMainForm;
    end;
    //Application.Minimize;
end;

// Delete All Letters in the folder
procedure Tfrm_Main.N13Click(Sender: TObject);
var
    Letters : PLetters;
    i       : word;
begin
    if Tree_Mail_TLS_SSLBoxes.Selected=nil then exit;
    if Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil then exit;

Letters:=Mail_TLS_SSLBoxes.FindLetters(Tree_Mail_TLS_SSLBoxes.Selected.Parent.Te
xt, Tree_Mail_TLS_SSLBoxes.Selected.Text);
    if Letters = nil then exit;
    if MessageDlg('Ви впевнені, що хочете безповоротно видалити ВСІ листи в цій
папці?',
        mtConfirmation,[mbYes, mbNo], 0)<> mrYes then exit;

    for i:=1 to Letters.Count do Letters.DeleteLetter(1);
    Tree_Mail_TLS_SSLBoxesClick(self);
end;

procedure Tfrm_Main.PopupMenu_Mail_TLS_SSLBoxesPopup(Sender: TObject);
begin
    PopupMenu_Mail_TLS_SSLBoxes.Items[0].Enabled:=false;
    PopupMenu_Mail_TLS_SSLBoxes.Items[1].Enabled:=false;
    PopupMenu_Mail_TLS_SSLBoxes.Items[2].Enabled:=false;
    if Tree_Mail_TLS_SSLBoxes.Selected=nil then exit;

    PopupMenu_Mail_TLS_SSLBoxes.Items[0].Enabled:=true;
    if Tree_Mail_TLS_SSLBoxes.Selected.Parent = nil then exit;
    PopupMenu_Mail_TLS_SSLBoxes.Items[1].Enabled:=true;
    PopupMenu_Mail_TLS_SSLBoxes.Items[2].Enabled:=true;
end;

```

```

// Відправити всі листи в поштовій скриньці
procedure Tfrm_Main.JnghfdbnmgjxnelClick(Sender: TObject);
var
  BoxName : shortstring;
  Mail_TLS_SSLBox : PMail_TLS_SSLBox;
  Letters : PLetters;
  i       : word;
begin
  if Tree_Mail_TLS_SSLBoxes.Selected=nil then exit;
  if Tree_Mail_TLS_SSLBoxes.Selected.Parent=nil then
    BoxName:=Tree_Mail_TLS_SSLBoxes.Selected.Text
           else
    BoxName:=Tree_Mail_TLS_SSLBoxes.Selected.Parent.Text;
  Mail_TLS_SSLBox:=Mail_TLS_SSLBoxes.FindBox(BoxName);
  if Mail_TLS_SSLBox=nil then exit;
  Letters:=@Mail_TLS_SSLBox.Outbox;
  if Letters=nil then exit;

  Tree_Mail_TLS_SSLBoxesClick(self);

  try
    try
      FillSMTPwithMail_TLS_SSLBoxSettings(CurrentMail_TLS_SSLBoxSettings);
      DM.SMTP.Connect(20000);
      frm_Main.AddToLog('З'єднання з сервером
'+DM.SMTP.host+':'+inttostr(DM.SMTP.port)+' успішне.');
```

КБІБ-2024

```

      for i:=1 to Letters.Count do
        begin
          frm_Main.AddToLog('Відправка листа від
'+CurrentMail_TLS_SSLBoxSettings.Name+' к
'+Letters.Letters[1].Recipients.Email_TLS_SSLAddresses+' за темою:
'+Letters.Letters[1].Subject);
          DM.SMTP.Send( Letters.Letters[1]^ ); // Відправка листа

Mail_TLS_SSLBox.Sent.Add(Letters.Letters[1],false,Letters.LettersSizes[1]); //
Додати до папки
          Mail_TLS_SSLBox.Outbox.DeleteLetterWithoutFreeing(1);
// Видалити з папки Вихідні
        end;

        finally
          DM.SMTP.Disconnect;
        end;
      except
        on E:Exception do
          begin
            Application.MessageBox(pchar('Помилка при відправці пошти зі скриньки
<'+BoxName+'> :'),pchar(E.Message) );
            AddToLog('Помилка при відправці пошти зі скриньки <'+BoxName+'>
:'+E.Message);
          end;
        end;
      end;

end;

// відкриття вікна "про програму..."
procedure Tfrm_Main.About1Click(Sender: TObject);
begin
  Form1.Show;
end;

end.
```

Файл frmNewLetter.pas - створення та відправка нового листа

```
unit frmNewLetter;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Buttons, ToolWin, ComCtrls, Menus, HTTPApp, HTTPProd, OleCtrls,
  SHDocVw, StdCtrls, ExtCtrls, ImgList, ActnList,
  IdMessage,
  frmMail_TLS_SSLSettings, LetterStorage;

type
  PidMessage = ^TidMessage;

  Tfrm_NewLetter = class(TForm)
    MainMenu: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    StatusBar: TStatusBar;
    CoolBar: TCoolBar;
    ToolBar1: TToolBar;
    Panell1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Ed_Reciever: TEdit;
    Ed_Copy: TEdit;
    Ed_Subject: TEdit;
    Memo_Body: TMemo;
    Label4: TLabel;
    Ed_HiddenCopy: TEdit;
    Actions_NewLetter: TActionList;
    SendNewLetter: TAction;
    Images_SendLetter: TImageList;
    ToolButton1: TToolButton;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    MoveNewLetter2Outbox: TAction;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    Splitter1: TSplitter;
    List_NewAttachments: TListView;
    ToolButton4: TToolButton;
    NewAttachment: TAction;
    Dialog_OpenAttachment: TOpenDialog;
    PopupM_Attachment: TPopupMenu;
    N6: TMenuItem;
    Ghbrhtgbnmafqk1: TMenuItem;
    Panel_AdditionlSettings: TPanel;
    Group_AdditionalSettings: TGroupBox;
    Label6: TLabel;
    Label7: TLabel;
    Label5: TLabel;
    Combo_Priority: TComboBox;
    Ed_Sender: TEdit;
    Memo_AdditionalHeaders: TMemo;
    ToolButton5: TToolButton;
    Btn_AdvancedPage: TBitBtn;
    Splitter2: TSplitter;
    N7: TMenuItem;
    N8: TMenuItem;
    Label8: TLabel;
    Ed_ReplyTo: TEdit;
    Label9: TLabel;
```

```

Combo_ContextType: TComboBox;
procedure SendNewLetterExecute(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure MoveNewLetter2OutboxExecute(Sender: TObject);
procedure NewAttachmentExecute(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure List_NewAttachmentsMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure Btn_AdvancedPageClick(Sender: TObject);
procedure N8Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormShow(Sender: TObject);
private
  { Private declarations }
  function CheckIfFormIsFilled : boolean;
public
  { Public declarations }
  IsToBeResend : boolean;
  ResendMes : TidMessage;
  Mail_TLS_SSLBoxName : shortstring;
  CURR_Mail_TLS_SSLbox_SETTINGS : TMail_TLS_SSLBoxSettings;
  CURR_MAIL_TLS_SSL_BOX : PMail_TLS_SSLBox;
  function CreateLetterFromForm:PidMessage;
  procedure UpdateStatusLine;
  procedure ClearForm;
end;

procedure FillSMTPwithMail_TLS_SSLBoxSettings(mbSettings :
PMail_TLS_SSLBoxSettings);

var
  frm_NewLetter: Tfrm_NewLetter;

implementation
uses frmMain, DataModule;
{$R *.dfm}

procedure FillSMTPwithMail_TLS_SSLBoxSettings(mbSettings :
PMail_TLS_SSLBoxSettings);
Begin
  DM.SMTP.Host :=mbSettings.SMTPServer;
  DM.SMTP.Port :=mbSettings.SMTPPort;
  DM.SMTP.Username :=mbSettings.SMTPAccount;
  DM.SMTP.Password :=mbSettings.SMTPPass;
  DM.SMTP.AuthenticationType:=mbSettings.AuthType;
End;

// відправити лист
procedure Tfrm_NewLetter.SendNewLetterExecute(Sender: TObject);
var
  mes : PidMessage;
begin
  if CurrentMail_TLS_SSLBoxSettings = nil then
  begin
    Application.MessageBox('Ви повинні вибрати скриньку з якої хочете ', 'Помилка
відправлення');
    exit;
  end;
  if not CheckIfFormIsFilled then exit;
  mes:=CreateLetterFromForm;
  SendNewLetter.Enabled:=false;
  try
    FillSMTPwithMail_TLS_SSLBoxSettings(CurrentMail_TLS_SSLBoxSettings);
    frm_Main.AddToLog('Негайна відправка листа від
'+CurrentMail_TLS_SSLBoxSettings.Name+' до
'+mes.Recipients.EMail_TLS_SSLAddresses);

```

```

    DM.SMTP.Connect(20000);
    frm_Main.AddToLog('З'єднання з сервером
'+DM.SMTP.host+':'+inttostr(DM.SMTP.port)+' успішне.');
```

```

    try
        DM.SMTP.Send(mes^);
    finally
        DM.SMTP.Disconnect;
    end;
    frm_Main.AddToLog('Відключення від сервера успішне, повідомлення
відправлене.');
```

```

    CURR_MAIL_TLS_SSL_BOX.Sent.Add(@mes^),false,0);
    frm_Main.Tree_Mail_TLS_SSLBoxesClick(self);
except
    on E:Exception do
        begin
            Application.MessageBox(pchar(E.message),'Не вдалося відправити пошту');
            CURR_MAIL_TLS_SSL_BOX.Outbox.Add(@mes^),false,0);
        end;
    end;
hide;
end;
```

```

procedure Tfrm_NewLetter.N4Click(Sender: TObject);
begin
    self.Hide;
end;
```

```

// Помістити лист в папку вихідні
procedure Tfrm_NewLetter.MoveNewLetter2OutboxExecute(Sender: TObject);
var
    mes : TIdMessage;
begin
    Application.MessageBox('','');
    mes:=CreateLetterFromForm^;
    CURR_MAIL_TLS_SSL_BOX.Outbox.Add(@mes,false,0);
    frm_Main.Tree_Mail_TLS_SSLBoxesClick(self);
    hide;
end;
```

```

// Додати файл
procedure Tfrm_NewLetter.NewAttachmentExecute(Sender: TObject);
var
    s : ^shortstring;
    st: TFileStream;
begin
    if not Dialog_OpenAttachment.Execute then exit;
    with List_NewAttachments.Items.Add do
        begin
            caption:=ExtractFileName(Dialog_OpenAttachment.FileName);
            try
                st:=TFileStream.Create(Dialog_OpenAttachment.FileName,fmOpenRead or
fmShareDenyNone);
                SubItems.Add(inttostr(st.Size));
                ImageIndex:=2;
                st.Free;
            except
                Application.MessageBox('Не вдалося отримати розмір файлу!','Помилка');
                exit;
            end;
            new(s);
            s^:=Dialog_OpenAttachment.FileName;
            data:=s;

            Splitter1.Visible:=true;
            List_NewAttachments.Visible:=true;
            UpdateStatusLine;
        end;
    end;
end;
```

```

// Контекстне меню -> Видалили з програми
procedure Tfrm_NewLetter.N6Click(Sender: TObject);
begin
  if List_NewAttachments.ItemIndex<0 then exit;
  dispose(List_NewAttachments.Items[List_NewAttachments.ItemIndex].Data);
  List_NewAttachments.DeleteSelected;
  UpdateStatusLine;
  if List_NewAttachments.Items.Count<=0 then
  begin
    Splitter1.Visible:=false;
    List_NewAttachments.Visible:=false;
  end;
end;

procedure Tfrm_NewLetter.List_NewAttachmentsMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var
  Item : TListItem;
begin
  Item:=List_NewAttachments.GetItemAt(x,y);
  if Item<>nil then Item.Selected:=true;
end;

function Tfrm_NewLetter.CreateLetterFromForm: PidMessage;
var
  Plet : PidMessage;
  i : integer;
  s : ^shortstring;
begin
  frm_Main.Tree_Mail_TLS_SSLBoxesClick(self);
  if CurrentMail_TLS_SSLBoxSettings =nil then exit;

  new(Plet);
  Plet^:=TIdMessage.Create(self);

  Plet^.Subject:=Ed_Subject.Text;
  Plet^.CCList.Email_TLS_SSLAddresses:=Ed_Copy.Text;
  Plet^.BccList.Email_TLS_SSLAddresses:=Ed_HiddenCopy.Text;
  Plet^.Body.Assign(Memo_Body.Lines);
  Plet^.From.Text:= CURR_Mail_TLS_SSLbox_SETTINGS.POPEmail_TLS_SSL; raises error
  on some servers
  Plet^.From.Text:=CurrentMail_TLS_SSLBoxSettings^.POPEmail_TLS_SSL;
  if Ed_ReplyTo.Text='' then
  Plet^.ReplyTo.Email_TLS_SSLAddresses:=CurrentMail_TLS_SSLBoxSettings.POPEmail_TLS
  S_SSL
  else
  Plet^.ReplyTo.Email_TLS_SSLAddresses:=Ed_ReplyTo.Text;
  Plet^.Recipients.Email_TLS_SSLAddresses:=Ed_Reciever.Text;
  case Combo_Priority.ItemIndex of
    0 : Plet^.Priority:=mpHighest;
    1 : Plet^.Priority:=mpHigh;
    2 : Plet^.Priority:=mpNormal;
    3 : Plet^.Priority:=mpLow;
    4 : Plet^.Priority:=mpLowest;
  end;
  Plet^.Sender.Text:=Ed_Sender.Text;
  Plet^.ExtraHeaders.Assign(Memo_AdditionalHeaders.Lines);
  Plet^.ContentType:=Combo_ContextType.Text;

  for i:=1 to List_NewAttachments.Items.Count do
  begin
    s:=List_NewAttachments.Items.Item[i-1].data;
    TIdAttachment.Create(Plet^.MessageParts, s^);
  end;
end;

```

```

    CreateLetterFromForm:=PLet;
end;

procedure Tfrm_NewLetter.UpdateStatusLine;
var
    AttSize : longint;
    i       : word;
begin
    AttSize:=0;
    for i:=1 to List_NewAttachments.Items.Count do
    inc(AttSize, strtointdef(List_NewAttachments.Items[i-1].SubItems[0],0));
    if AttSize<>0 then StatusBar.Panels.Items[1].Text:='Розмір вкладених файлів :
'+inttostr(AttSize div 1024)+' Kb'
    else StatusBar.Panels.Items[1].Text:='';
end;

// відкрити/приховати додаткові поля
procedure Tfrm_NewLetter.Btn_AdvancedPageClick(Sender: TObject);
begin
    if Btn_AdvancedPage.Caption<>'Прибрати' then
    begin
        Panel_AdditionlSettings.Show;
        Splitter2.Show;
        Btn_AdvancedPage.Caption:='Прибрати';
        Btn_AdvancedPage.Width:=230;
    end
    else
    begin
        Splitter2.Hide;
        Panel_AdditionlSettings.Hide;

        Btn_AdvancedPage.Caption:='Додатково';
        Btn_AdvancedPage.Width:=170;
    end;
end;

procedure Tfrm_NewLetter.ClearForm;
begin
    Ed_Reciever.Clear;
    Ed_Copy.Clear;
    Ed_Subject.Clear;
    Ed_HiddenCopy.Clear;
    Ed_Sender.Clear;
    Memo_Body.Clear;
    Memo_AdditionalHeaders.Clear;
    List_NewAttachments.Clear;
    Ed_ReplyTo.Clear;
    Combo_Priority.ItemIndex:=2;

    IsToBeResend:=false;
    ResendMes.Clear;
    SendNewLetter.Enabled:=true;
end;

procedure Tfrm_NewLetter.N8Click(Sender: TObject);
begin
    ClearForm;
end;

function Tfrm_NewLetter.CheckIfFormIsFilled: boolean;
begin
    CheckIfFormIsFilled:=false;
    if Ed_Reciever.Text='' then
    begin
        Application.MessageBox('Ви забули вказати Кому ви хочете відправити цей
лист.', 'Помилка');
        exit;
    end;
end;

```

```
    if Ed_Subject.Text='' then
    begin
        if Application.MessageBox('Ви дійсно хочете залишити повідомлення без
Теми?', 'Підтвердіть...', MB_YESNO )=IDNo
            then exit;
        end;
        CheckIfFormIsFilled:=true;
    end;

procedure Tfrm_NewLetter.FormCreate(Sender: TObject);
begin
    frm_NewLetter.ResendMes:=TIdMessage.Create(self);
end;

procedure Tfrm_NewLetter.FormShow(Sender: TObject);
begin
    CURR_Mail_TLS_SSLbox_SETTINGS:= CurrentMail_TLS_SSLBoxSettings^;

CURR_MAIL_TLS_SSL_BOX:=Mail_TLS_SSLBoxes.FindBox(CurrentMail_TLS_SSLBoxSettings.
Name);
end;

end.
```

КБПЗ_2024

Файл LetterStorage.pas - збереження листів

```

unit LetterStorage;

interface
uses idMessage, Classes, ExtCtrls { TTimer }, IdPOP3, frmMail_TLS_SSLSettings,
SyncObjs{CS};

const
  LetterDir      = 'Letters\';
  LetterBaseExt  = '.let';

type
{
TMyAttachments = object
  public
  Mail_TLS_SSLBoxName,
  FolderName    : shortstring;
  Count         : word;
  constructor   Init( _Mail_TLS_SSLboxname, _foldername : shortstring);
  constructor   Load(st : TStream);
  destructor    Done;
  procedure     Save(st : TStream);
  procedure     Add(fname : shortstring);
  function      Get(index : word):string;
  private
  Filenames    : packed array [1..512] of ^shortstring;
  end;
}

PIdMessage = ^TIdMessage;
PLetters = ^TLetters;
TLetters = object
  private
  Mail_TLS_SSLBoxName,
  FolderName      : shortstring;

  TotalLetters   : word;
  public
  IsOnlyHeader   : {packed} array [1..1024*32] of boolean;
  LettersSizes   : {packed} array [1..1024*32] of longint;
  Letters        : array [1..1024*32] of PIdMessage;
  constructor    Init(Mail_TLS_SSLBox, Folder : shortstring);
  procedure      Add( mes:PIdMessage; IsHeader: boolean; Size : longint);
  procedure      Load;
  procedure      Save;
  procedure      CreateEmptyLellerArchive(stName : shortstring);
  function       Count:word;
  function       HasLetterWithId( id : string ): boolean;
  procedure      DeleteLetter(index : integer);
  procedure      DeleteLetterWithoutFreeing(index : integer);
  procedure      PrepareLetter(Index : integer);
  end;

PBoolean      = ^boolean;
TGetMail_TLS_SSLThread = class(TThread)
  private
  POP : TIdPOP3;
  Mail_TLS_SSLBoxSettings : TMail_TLS_SSLBoxSettings;
  LogString      : string;
  BalloonMes     : string;
  IsBuisy        : PBoolean;
  protected
  procedure Execute; override;
  procedure CallAddToLog(s:string);

```

```

        procedure CallShowBalloon(s:string);
        procedure _ShowBalloon;
        procedure _Add2log;
    public
        procedure Init( var mbSettings : TMail_TLS_SSLBoxSettings;
IsBuisyFlag : Pboolean );
        end;

PMail_TLS_SSLBox = ^TMail_TLS_SSLBox;
TMail_TLS_SSLBox = object
    private
        BoxName      :   shortstring;
    public
        Inbox,
        Outbox,
        Sent,
        Deleted      :   TLetters;
        { Для збереження з іншою частиною інформації } { Не забувайте зберігати це
також }
        TotalLetters      : word;
        UsedSizeOnServer  : integer;
        DeletedIDs        : TStringList;
        LoadedFolder      : (lfNone,lfInbox,lfOutBox,lfSent,lf,Deleted,lfAll);

        CheckTimer       : TTimer;
        IsBuisyNow        : boolean;
        CheckThread       : TGetMail_TLS_SSLThread;

        procedure onTimer_Check(Sender: TObject);
        procedure SetTimerInterval( i : word );

        constructor Init(name : shortstring);
        function  GetName      : shortstring;
        function  HasLetterWithID( id : string): boolean;
        function  HadLetterWithID( id : string): boolean;
        procedure Save;
        function  MoveToGarbage( folder : PLetters; index : integer ):boolean;
        end;

TMail_TLS_SSLBoxList =object
    private
        Mail_TLS_SSLBoxes :   array [1..100] of PMail_TLS_SSLBox;
    public
        TotalBoxes:   word;
        function Add(name : shortstring)          : PMail_TLS_SSLBox;
        function GetByName (name : shortstring)   : PMail_TLS_SSLBox;
        function GetByIndex(index: word)         : PMail_TLS_SSLBox;
        function FindLetters( Mail_TLS_SSLbox, Mail_TLS_SSLfolder : shortstring
):PLetters;
        function FindBox( boxname : shortstring): PMail_TLS_SSLBox;
        procedure SaveMail_TLS_SSLBoxes;
        end;

var
    LettersPath      :   string;
    CS                :   TCriticalSection;

implementation

uses SysUtils,frmMain, frmAskOnDuplicate,Forms, dzlib, Controls {4modal
results},
    Dialogs {for MessageDlg}, CoolTrayIcon;

```

```

{ TLetter }

constructor TLetters.Init(Mail_TLS_SSLBox, Folder : shortstring);
Begin
  Mail_TLS_SSLBoxName := Mail_TLS_SSLBox;
  FolderName := Folder;
End;

{ *** Структура файлу TLetters ***}

{ 4 байти          - розмір наступного листа }
- ідентифікатор} *)
{ ... .. }
{ .. Лист ..}
{ ... .. }

procedure TLetters.Load;
var
  SourceFile      : shortstring;
  stCompressed    : TFileStream;
  zstCompressed   : TDecompressionStream;
  stM             : TMemoryStream;

  TotalLetters2Load : word;
  sizeOfALetter     : word;
Begin
  findMail_TLS_SSLBoxAndSelectIt(Mail_TLS_SSLBoxName);
  if CurrentMail_TLS_SSLBoxSettings.DynamicFolders then exit;

  SourceFile:=LettersPath+Mail_TLS_SSLBoxName+' '+FolderName+LetterBaseExt;

  try
    stCompressed:=TFileStream.Create(SourceFile, fmOpenRead);
  except
    CreateEmptyLellerArchive(SourceFile);
    frm_Main.AddToLog('Не удалось найти архивные файлы ящика
<'+Mail_TLS_SSLBoxName+'>. Они будут созданы заново. ');
    Application.MessageBox(pchar('Архів листів "'+SourceFile+'" не може бути
відкритий. '), 'Помилка завантаження листів. ');
    exit;
  end;

  try
    zstCompressed:=TDecompressionStream.Create(stCompressed);
  except
    on E:Exception do
      begin
        stCompressed.Free;
        Application.MessageBox(pchar('Архів листів "'+SourceFile+'" не може бути
відкритий і розархівований! Exception =' +E.Message), 'Помилка завантаження за
архівованих листів. ');
        exit;
      end;
  end;

  // Тут починається читання архівних даних
  try
    try
      zstCompressed.Seek(0, soFromBeginning);
      zstCompressed.Read(TotalLetters2Load, sizeof(TotalLetters2Load));

      zstCompressed.Read(IsOnlyHeader, sizeof(IsOnlyHeader));
      zstCompressed.Read(LettersSizes, sizeof(LettersSizes));

```

```

TotalLetters:=0;
while (TotalLetters < TotalLetters2Load) do
try
  zstCompressed.Read(sizeOfALetter, sizeof(sizeOfALetter));
  stM:=TMemoryStream.Create;
  stM.Seek(0, soFromBeginning);
  stM.CopyFrom(zstCompressed, sizeOfALetter);
  stM.Seek(0, soFromBeginning);
  inc(TotalLetters);
  new(Letters[TotalLetters]);
  Letters[TotalLetters]^:=TIdMessage.Create(nil);
  Letters[TotalLetters]^..LoadFromStream(stM, false);
finally
  stM.Free;
end;
finally
  zstCompressed.Free;
  stCompressed.Free;
end;
except
  Application.MessageBox(pchar('Виникла помилка при завантаженні листів з файлу
"' + SourceFile + '".'+
                                # 10 # 13 'Можливо він пошкоджений. Спробуйте видалити
його .'), ' Помилка завантаження листів. ')
  мети;

End;

function TLetters.Count: word;
begin
  Count:=TotalLetters;
end;

procedure TLetters.Save;
var
  i      : word;
  st     : TFileStream;
  stName : shortstring;
  zst    : TCompressionStream;
  stM    : TMemoryStream;
  LetterSize : word;
begin
  try
    stName:=LettersPath+Mail_TLS_SSLBoxName+' '+FolderName+LetterBaseExt;
    st:=TFileStream.Create(stName, fmOpenWrite);
  except
    try
      st:=TFileStream.Create(stName, fmCreate);
    except
      Application.MessageBox(pchar('Архів листів "'+stName+'" не може бути
створений. '), 'Помилка створення листів. ');
      exit;
    end;
  end;
  st.Size:=0;

  try
    zst:=TCompressionStream.Create(clMax, st);
  except
    on E:Exception do
      begin
        st.Free;
        Application.MessageBox(pchar('Архів листів "'+stName+'" не може бути
стиснутий. Exception='+E.Message), 'Помилка завантаження листів. ');
        exit;
      end;
    end;
  end;
end;

```

```

{ Написати загальну кількість листів }
zst.Write(TotalLetters, sizeof(TotalLetters));
zst.Write(IsOnlyHeader, sizeof(IsOnlyHeader));
zst.Write(LettersSizes, sizeof(LettersSizes));

for i:=1 to TotalLetters do
try
  stM:=TMemoryStream.Create;
  Letters[i].SaveToStream(stM, false);

  stM.Seek(0, soFromBeginning);
  LetterSize:=stM.size;
  zst.Write(LetterSize, sizeof(LetterSize));

  stM.Seek(0, soFromBeginning);
  zst.CopyFrom(stM, stM.Size);
finally
  stM.Free;
end;

zst.Free;
st.Free;

end;

procedure TLetters.CreateEmptyLellerArchive(stName: shortstring);
var
  st : TFileStream;
  zst : TCompressionStream;
begin
{ try
  st:=TFileStream.Create(stName, fmOpenWrite);
except
  try
    st:=TFileStream.Create(stName, fmCreate);
  except
    Application.MessageBox(pchar('Архів листів "'+stName+'" не може бути
створений.'), 'Помилка збереження листів.');
```

```

procedure TLetters.Add(mes: PIdMessage; IsHeader: boolean; Size: longint);
begin
  inc(TotalLetters);
  new(Letters[TotalLetters]);
  //mes.MessageParts.
  Letters[TotalLetters]^:=TIdMessage.Create(nil);
  Letters[TotalLetters]^:=mes^;
  IsOnlyHeader[TotalLetters]:=IsHeader;
  LettersSizes[TotalLetters]:=Size;
end;

```

```

function TLetters.HasLetterWithId(id: string): boolean;
var
  i : word;
begin
  HasLetterWithId:=true;
  for i:=1 to TotalLetters do
    if Letters[i]^MsgId = id then exit;
  HasLetterWithId:=false;
end;

```

```

procedure TLetters.DeleteLetter(index: integer);
var
  i : word;
  temp : PIdMessage;
  mesID : shortstring;
  Mail_TLS_SSLbx: PMail_TLS_SSLBox;
  IdIndex : integer;
begin
  temp:=Letters[index];
  mesID:=temp^.MsgId;
  for i:=index to TotalLetters-1 do
    begin
      Letters[i]:=Letters[i+1];
      LettersSizes[i]:=LettersSizes[i+1];
      IsOnlyHeader[i]:=IsOnlyHeader[i+1];
    end;
  dec(TotalLetters);
  temp.Free;
  dispose(temp);

  Mail_TLS_SSLbx:=Mail_TLS_SSLBoxes.FindBox(Mail_TLS_SSLBoxName);
  if Mail_TLS_SSLbx = nil then exit;
  IdIndex:=Mail_TLS_SSLbx.DeletedIDs.IndexOf(mesID);
  if IdIndex<>-1 then exit
    else Mail_TLS_SSLbx.DeletedIDs.Add(mesID);
end;

```

```

procedure TLetters.DeleteLetterWithoutFreeing(index: integer);
var
  i : word;
  temp : PIdMessage;
  mesID : shortstring;
  Mail_TLS_SSLbx: PMail_TLS_SSLBox;
  IdIndex : integer;
begin
  temp:=Letters[index];
  mesID:=temp^.MsgId;
  for i:=index to TotalLetters-1 do
    begin
      Letters[i]:=Letters[i+1];
      LettersSizes[i]:=LettersSizes[i+1];
    end;
end;

```

```

        IsOnlyHeader[i]:=IsOnlyHeader[i+1];
    end;
dec (TotalLetters);
temp.Free;
dispose (temp);

Mail_TLS_SSLbx:=Mail_TLS_SSLBoxes.FindBox (Mail_TLS_SSLBoxName);
if Mail_TLS_SSLbx = nil then exit;
IdIndex:=Mail_TLS_SSLbx.DeletedIDs.IndexOf (mesID);
if IdIndex<>-1 then exit
    else Mail_TLS_SSLbx.DeletedIDs.Add (mesID);
end;

{ Викликається, коли прийшов час перевірити цю поштову скриньку }
procedure TMail_TLS_SSLBox.onTimer_Check(Sender: TObject);
Begin
    if IsBuisyNow then exit;
    IsBuisyNow:=true;
    frm_Main.aCheckMail_TLS_SSL.Enabled:=false;
    CheckThread.Resume;
End;

constructor TMail_TLS_SSLBox.Init (name : shortstring);
Begin
    BoxName:=name;
    IsBuisyNow:=false;
    findMail_TLS_SSLBoxAndSelectIt (name);
    CheckThread:=TGetMail_TLS_SSLThread.Create (true);
    CheckThread.Init (CurrentMail_TLS_SSLBoxSettings^, @IsBuisyNow);

    if CurrentMail_TLS_SSLBoxSettings.DynamicFolders then LoadedFolder:=lfNone
        else LoadedFolder:=lfAll;

    CheckTimer:=TTimer.Create (frm_Main);
    CheckTimer.OnTimer:=onTimer_Check;
    if CurrentMail_TLS_SSLBoxSettings.CheckEvery<>0 then
    begin
        CheckTimer.Interval:=CurrentMail_TLS_SSLBoxSettings.CheckEvery*6000;
        CheckTimer.Enabled:=true;
    end
    else CheckTimer.Enabled:=false;

    Inbox.Init (name, 'Вхідні');
    Outbox.Init (name, 'Вихідні');
    Sent.Init (name, 'Відправлені');
    Deleted.Init (name, 'Видалені');

    Inbox.Load;
    Outbox.Load;
    Sent.Load;
    Deleted.Load;

    if not FileExists (LettersPath+BoxName+' DeletedIDs.dat') then
    FileClose (FileCreate (LettersPath+BoxName+' DeletedIDs.dat'));
    DeletedIDs:=TStringList.Create;           { Ми повинні визволити це у кінці }
    DeletedIDs.LoadFromFile (LettersPath+BoxName+' DeletedIDs.dat');

End;

function TMail_TLS_SSLBox.GetName : shortstring;
Begin

```

```

    GetName:=BoxName;
End;

procedure TMail_TLS_SSLBox.Save;
begin
    Inbox.Save;
    OutBox.Save;
    Sent.Save;
    Deleted.Save;

    deletefile(LettersPath+BoxName+' DeletedIDs.dat');
    DeletedIDs.SaveToFile(LettersPath+BoxName+' DeletedIDs.dat');
end;

function TMail_TLS_SSLBox.HasLetterWithID(id: string): boolean;
begin
    if (Inbox.HasLetterWithId(id) or Outbox.HasLetterWithId(id) or
        Sent.HasLetterWithId(id) or Deleted.HasLetterWithId(id) ) then
        HasLetterWithID:=true
    else
        HasLetterWithID:=false;
end;

function TMail_TLS_SSLBox.HadLetterWithID(id: string): boolean;
begin
    HadLetterWithID:= DeletedIDs.IndexOf(id)<>-1 ;
end;

procedure TMail_TLS_SSLBox.SetTimerInterval(i: word);
begin
    CheckTimer.Interval:=i*60000;
    if i=0 then CheckTimer.Enabled:=false
    else CheckTimer.Enabled:=true;
end;

function TMail_TLS_SSLBox.MoveToGarbage(folder: PLetters; index:
integer):boolean;
begin
    if folder.FolderName<>'Удаленные' then
    begin
        Deleted.Add( Folder.Letters[index], false, Folder.LettersSizes[index]);
        folder.DeleteLetterWithoutFreeing(index);
        MoveToGarbage:=true;
    end
    else // Якщо ви хочете видалити лист зі сміттевого кошику
    if MessageDlg('Ви впевнені(!), що хочете безповоротно видалити цей
лист?',mtConfirmation,[mbYes, mbNo], 0) <> mrNo then
    begin
        folder.DeleteLetter(index);
        MoveToGarbage:=true;
    end
    else MoveToGarbage:=false;
end;

function TMail_TLS_SSLBoxList.Add(name : shortstring):PMail_TLS_SSLBox;
Begin
    inc(TotalBoxes);
    new(Mail_TLS_SSLBoxes[TotalBoxes]);
    Mail_TLS_SSLBoxes[TotalBoxes]^ .init(name);
    Add:=Mail_TLS_SSLBoxes[TotalBoxes];
End;

```

```

function TMail_TLS_SSLBoxList.GetByName (name : shortstring):PMail_TLS_SSLBox;
var
  i : word;
Begin
  GetByName:=nil;
  for i:=1 to TotalBoxes do
    if AnsiUpperCase( Mail_TLS_SSLBoxes[i]^ .BoxName ) = AnsiUpperCase(name)
then
  begin
    GetByName:=Mail_TLS_SSLBoxes[i];
    exit;
  end;
End;

```

```

function TMail_TLS_SSLBoxList.FindLetters( Mail_TLS_SSLbox, Mail_TLS_SSLfolder
: shortstring ):PLetters;
var
  i : word;
Begin
  FindLetters:=nil;
  for i:=1 to TotalBoxes do
    if Mail_TLS_SSLBoxes[i].BoxName = Mail_TLS_SSLbox then
begin
  if Mail_TLS_SSLfolder = 'Вхідні' then
begin
  FindLetters:=@Mail_TLS_SSLBoxes[i].Inbox;
  exit;
  end;
  if Mail_TLS_SSLfolder = 'Вихідні' then
begin
  FindLetters:=@Mail_TLS_SSLBoxes[i].OutBox;
  exit;
  end;
  if Mail_TLS_SSLfolder = 'Відправлені' then
begin
  FindLetters:=@Mail_TLS_SSLBoxes[i].Sent;
  exit;
  end;
  if Mail_TLS_SSLfolder = 'Видалені' then
begin
  FindLetters:=@Mail_TLS_SSLBoxes[i].Deleted;
  exit;
  end;
end;
end;
End;

```

```

procedure TMail_TLS_SSLBoxList.SaveMail_TLS_SSLBoxes;
var
  i : word;
Begin
  for i:=1 to TotalBoxes do Mail_TLS_SSLBoxes[i].Save;
End;

```

```

function TMail_TLS_SSLBoxList.FindBox(boxname : shortstring):PMail_TLS_SSLBox;
var
  i : word;
Begin
  FindBox:=nil;
  for i:=1 to TotalBoxes do
    if Mail_TLS_SSLBoxes[i].BoxName = boxname then
begin
  FindBox:=Mail_TLS_SSLBoxes[i];
  break;
  end;
end;

```

End;

```
function TMail_TLS_SSLBoxList.GetByIndex(index : word):PMail_TLS_SSLBox;
Begin
  if (index>=1)and(index<=100) then GetByIndex:=Mail_TLS_SSLBoxes[index]
  else GetByIndex:=nil;
```

End;

```
{ TMyAttachments }
```

```
constructor TMyAttachments.Init(_Mail_TLS_SSLboxname,_foldername : shortstring);
Begin
  Count:=0;
  Mail_TLS_SSLBoxName:=_Mail_TLS_SSLBoxName;
  FolderName :=_foldername;
```

```
  fillchar(FileNames,sizeof(FileNames),0);
End;
```

```
constructor TMyAttachments.Load(st : TStream);
var
```

```
  i : word;
  s : shortstring;
```

Begin

```
  Count:=0;
  fillchar(FileNames,sizeof(FileNames),0);
  st.Read(Count,sizeof(Count));
  for i:=1 to Count do
  begin
    st.Read(s,sizeof(s));
    Add(s);
  end;
```

End;

```
procedure TMyAttachments.Save(st : TStream);
var
```

```
  i : word;
```

Begin

```
  st.write(Count,sizeof(Count));
  for i:=1 to Count do st.write(FileNames[i]^,sizeof(FileNames[i]^));
```

End;

```
destructor TMyAttachments.Done;
```

```
var
```

```
  i : word;
```

Begin

```
  for i:=1 to Count do dispose(FileNames[i]);
```

End;

```
procedure TMyAttachments.Add(fname : shortstring);
```

Begin

```
  inc(Count);
  new(FileNames[Count]);
  FileNames[Count]^:=fname;
```

End;

```
function TMyAttachments.Get(index: word): string;
```

begin

```
  if index<=Count then Get:=FileNames[index]^;
```

end;

```

{ TGetMail_TLS_SSLThread }

procedure TGetMail_TLS_SSLThread.Init(var mbSettings : TMail_TLS_SSLBoxSettings;
IsBuisyFlag : PBoolean );
begin
  POP:=TIdPOP3.Create(frm_Main);
  POP.Host :=mbSettings.POPServer;
  POP.Port :=mbSettings.POPPort;
  POP.Username :=mbSettings.POPAccount;
  POP.Password :=mbSettings.POPPass;

  Mail_TLS_SSLBoxSettings:=mbSettings;
  IsBuisy:=IsBuisyFlag;
end;

procedure TGetMail_TLS_SSLThread.CallAddToLog(s:string);
Begin
  LogString:=s;
  Synchronize(_Add2log);
End;

procedure TGetMail_TLS_SSLThread._Add2log;
Begin
  frm_Main.AddToLog(LogString);
End;

procedure TGetMail_TLS_SSLThread.CallShowBalloon(s:string);
Begin
  BalloonMes:=s;
  Synchronize(_ShowBalloon);
End;

procedure TGetMail_TLS_SSLThread._ShowBalloon;
Begin
  if BalloonMes[1]='B' then frm_Main.Tray.ShowBalloonHint(
'MyMail_TLS_SSL',BalloonMes,bitInfo,30)
  else frm_Main.Tray.ShowBalloonHint(
'MyMail_TLS_SSL',BalloonMes,bitError,30)
End;

procedure TGetMail_TLS_SSLThread.Execute;
var
  CurMail_TLS_SSLBox      :   PMail_TLS_SSLBox;
  LettersOnServer, i,
  LettersNot2beDownloaded :   word;
  mes                     :   TIdMessage;
begin
  inherited;
  while not Terminated do
  BEGIN
  // Перевірка пошти
  if POP.Connected then Pop.Disconnect;

  try
    POP.Connect(20000);           // 20 секунд
  except
    on E:Exception do
      begin
        CallAddToLog('Помилка підключення до сервера <'+pop.host+
:'+inttostr(POP.port)+'> Exception='+E.message);

```

```

    if Mail_TLS_SSLBoxSettings.ShowBalloonOnNewMes then
        CallShowBalloon('Помилка підключення до сервера:'#10#13'<' + pop.host +
: '+inttostr(POP.port)+'>');
        IsBuisy^:=false;
        exit;
    end;
end; // of try

CurMail_TLS_SSLBox:=Mail_TLS_SSLBoxes.GetByName(Mail_TLS_SSLBoxSettings.Name);
LettersOnServer := POP.CheckMessages;
CurMail_TLS_SSLBox^.UsedSizeOnServer := POP.RetrieveMail_TLS_SSLBoxSize;

CallAddToLog('Підключення до'+POP.Host+' для
<' +Mail_TLS_SSLBoxSettings.Name+'> успішне. На сервері
'+inttostr(LettersOnServer)+' (нових або непрочитаних) листів
'+inttostr(CurMail_TLS_SSLBox^.UsedSizeOnServer)+' байт');

try
{ Частина коду, яка відповідає за завантаження кожного повідомлення }

LettersNot2beDownloaded:=0;
for i:=1 to LettersOnServer do
    if Mail_TLS_SSLBoxSettings.RetrieveOnlyHeaders then
        // Отримати лише заголовки
        begin { витягнути тільки заголовки}
            mes:=TIdMessage.Create(frm_Main);
            POP.RetrieveHeader(i,mes);
        end
    else
        // Отримати всі листи
        begin
            mes:=TIdMessage.Create(frm_Main);
            POP.RetrieveHeader(i,mes);
            if
{Mail_TLS_SSLBoxes.GetByName(Mail_TLS_SSLBoxSettings.Name)}CurMail_TLS_SSLBox^.H
asLetterWithID(mes.MsgId) then
                begin
                    inc(LettersNot2beDownloaded);
                    continue;
                end;
            if
{Mail_TLS_SSLBoxes.GetByName(Mail_TLS_SSLBoxSettings.Name)}CurMail_TLS_SSLBox^.h
adLetterWithId(mes.msgId) then
                begin
                    CS.Enter;
                    case Mail_TLS_SSLBoxSettings.ifDuplicate of
                        Ignore : begin
                                    continue;
                                end;
                        Recieve : ;
                        Ask : begin
                                frm_AskOnDuplicate.FillInInfo(@mes);
                                while IsShown do ;
                                    IsShown:=true;
                                    frm_AskOnDuplicate.ShowModal;
                                    IsShown:=false;
                                    case frm_AskOnDuplicate.ModalResult of
                                        mrIgnore : continue;
                                        mrOk : ;
                                        mrAbort : begin server
                                                    if not POP.Delete(i) then CallAddToLog('Ошибка
удаления письма как полученного ранее, но удаленного с компьютера (см. настройки)
От: "' +mes.From.Address+'"' Тема: "' +mes.Subject+'"' );
                                                    CS.Leave;
                                                    continue;
                                                end;
                                    end;
                                end;
                            end;
                end;
            end;
        end;
    end;
end;

```

```

        end;
        DeleteFromServer : begin
            if not POP.Delete(i) then CallAddToLog('Ошибка
удаления письма как полученного ранее, но удаленного с компьютера (см. настройки)
От: "' + mes.From.Address + '" Тема: "' + mes.Subject + '"');
                CS.Leave;
                continue;
            end;
        end;
        CS.Leave;
    end;

    CallAddToLog('Отримання листа
(' + inttostr(i) + '/' + inttostr(LettersOnServer) + ') від ' + mes.From.Text + ' по Теме
: "' + mes.Subject + '"');
    if Mail_TLS_SSLBoxSettings.ShowBalloonOnNewMes then
        CallShowBalloon('Вхідний лист. '#10#13+'Від:
' + mes.From.Text + '#10#13+'Тема: ' + mes.Subject);
        POP.Retrieve(i, mes);

{Mail_TLS_SSLBoxes.GetByName(Mail_TLS_SSLBoxSettings.Name)}CurMail_TLS_SSLBox^.I
nbox.Add(@mes, false, POP.RetrieveMsgSize(i));
    if Mail_TLS_SSLBoxSettings.DeleteFromServer then
        if not POP.Delete(i) then frm_Main.AddToLog('Помилка видалення листа з
сервера. Від: "' + mes.From.Address + '" Тема: "' + mes.Subject + '"');
        end;

    if LettersNot2beDownloaded <> 0 then
        CallAddToLog('На сервері лежить (ать) ' + inttostr(LettersNot2beDownloaded) + '
листи, які не були завантажені, тому що вони вже були завантажені раніше і
лежать у папках поточної скриньки');

        finally
            IsBuisy^ := false;
            CS.Enter;
            frm_Main.Tree_Mail_TLS_SSLBoxesClick(self);
            CS.Leave;
            if POP.Connected then Pop.Disconnect;
            CallAddToLog('Отключение от ' + POP.Host + ' - ОК. ');

            self.Suspend;
        end;
    END;
end;

begin

    CS := TCriticalSection.Create;

    LettersPath := IncludeTrailingPathDelimiter(ExtractFilePath(Application.exename));
    LettersPath := LettersPath + LetterDir;
    if not DirectoryExists(LettersPath) then ForceDirectories(LettersPath);

end.

finalization
    CS.Free;
end.

```

Файл frmMail_TLS_SSLSettings.pas - параметри облікових записів та створення нових поштових скриньок

```

unit frmMail_TLS_SSLSettings;

interface

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

type
  PMail_TLS_SSLBoxSettings = ^TMail_TLS_SSLBoxSettings;

Tfrm_Mail_TLS_SSLSettings = class(TForm)
  ListBox_Mail_TLS_SSLBoxes: TListBox;
  GroupBox1: TGroupBox;
  GroupBox3: TGroupBox;
  GroupBox2: TGroupBox;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Combo_SMTPType: TComboBox;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Ed_SMTPServer: TEdit;
  Ed_SMTPPort: TEdit;
  Ed_SMTPAccount: TEdit;
  Ed_SMTPPassword: TEdit;
  Ed_POPServer: TEdit;
  Ed_POPPort: TEdit;
  Ed_POPAccount: TEdit;
  Ed_POPPassword: TEdit;
  btn_Apply: TBitBtn;
  btn_NewMail_TLS_SSLBox: TBitBtn;
  GroupBox4: TGroupBox;
  Label11: TLabel;
  Ed_Subscript: TEdit;
  Ed_Mail_TLS_SSLBoxName: TEdit;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Combo_CheckEvery: TComboBox;
  BitBtn1: TBitBtn;
  Check_RetrieveHeadersOnly: TCheckBox;
  Check_DeleteFromServer: TCheckBox;
  Check_DontOpenHTML: TCheckBox;
  GroupBox5: TGroupBox;
  Label15: TLabel;
  Radio_ignore: TRadioButton;
  Radio_Recieve: TRadioButton;
  Radio_Ask: TRadioButton;
  RadioDelete: TRadioButton;
  Label10: TLabel;
  Ed_POPEmail_TLS_SSL: TEdit;
  Check_DynamicFolders: TCheckBox;
  Check_ShowBalloon: TCheckBox;

  procedure btn_NewMail_TLS_SSLBoxClick(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure Combo_CheckEveryExit(Sender: TObject);

```

```

procedure btn_ApplyClick(Sender: TObject);
procedure ListBox_Mail_TLS_SSLBoxesClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
  procedure LoadSettingsFromRecords;
  procedure LoadFormFromBox(box : PMail_TLS_SSLBoxSettings);
  procedure SaveFormToBox(box : PMail_TLS_SSLBoxSettings);
  procedure ClearForm;
end;

TDuplicateAction = (Ask, Ignore, Recieve, DeleteFromServer);

TMail_TLS_SSLBoxSettings = record
  SMTPServer :shortstring;
  SMTPPort   :integer;
  SMTPAccount :shortstring;
  SMTPPass   :shortstring;
  AuthType   :TAuthenticationType;

  POPServer  :shortstring;
  POPPort    :integer;
  POPAccount :shortstring;
  POPPass    :shortString;
  POPEmail_TLS_SSL :shortstring;

  Subscript  :shortstring;
  Name       :shortstring;
  CheckEvery :word;

  RetrieveOnlyHeaders : boolean;
  DeleteFromServer    : boolean;
  OpenHTMLasTEXT      : boolean;
  ifDuplicate         : TDuplicateAction;
  DynamicFolders       : boolean;
  ShowBalloonOnNewMes : boolean;
end;

procedure SaveSettingsToFile;
procedure LoadSettingsFromfile;
procedure findMail_TLS_SSLBoxAndSelectIt(name : shortstring);

const
  Mail_TLS_SSLBoxSettingsFileName = 'Mail_TLS_SSLboxes.dat';

var
  frm_Mail_TLS_SSLSettings : Tfrm_Mail_TLS_SSLSettings;
  TotalMail_TLS_SSLBoxes  : word=0;
  Mail_TLS_SSLBoxesSettings : array [1..300] of PMail_TLS_SSLBoxSettings;
  CurrentMail_TLS_SSLBoxSettings : PMail_TLS_SSLBoxSettings=nil;
  PrevMail_TLS_SSLBoxSettings : PMail_TLS_SSLBoxSettings=nil;

implementation

uses frmMain;
{$R *.dfm}

procedure findMail_TLS_SSLBoxAndSelectIt(name : shortstring);

```

```

var
  i : word;
Begin
  CurrentMail_TLS_SSLBoxSettings := nil;
  for i:=1 to TotalMail_TLS_SSLBoxes do
    if Mail_TLS_SSLBoxesSettings[i]^Name = name then
      begin
        CurrentMail_TLS_SSLBoxSettings:=Mail_TLS_SSLBoxesSettings[i];
        exit;
      end;
  End;

procedure SaveSettingsToFile;
var
  SFile : shortstring;
  st     : TFileStream;
  i      : word;
Begin
  SFile:=ExtractFilePath(Application.ExeName)+Mail_TLS_SSLBoxSettingsFileName;
  try
    deletefile(SFile);
  except end;
  try
    st:=TFileStream.Create(SFile, fmOpenWrite or fmShareDenyNone);
  except
    try
      st:=TFileStream.Create(SFile, fmCreate);
    except
      Application.MessageBox( pchar('Не можу відкрити файл "'+SFile+'" для
запису !'), 'Помилка збереження параметрів скриньки');
      exit;
    end;
  end;
  st.Write(TotalMail_TLS_SSLBoxes, sizeof(TotalMail_TLS_SSLBoxes));
  for i:=1 to TotalMail_TLS_SSLBoxes do
    st.write(Mail_TLS_SSLBoxesSettings[i]^, sizeof(TMail_TLS_SSLBoxSettings));
  st.Free;
End;

procedure LoadSettingsFromfile;
var
  SFile : shortstring;
  st     : TFileStream;
  i      : word;
Begin
  SFile:=ExtractFilePath(Application.ExeName)+Mail_TLS_SSLBoxSettingsFileName;
  try
    st:=TFileStream.Create(SFile, fmOpenRead or fmShareDenyNone);
  except
    Application.MessageBox( pchar('Не можу відкрити файл "'+SFile+'" для запису
!'), 'Error loading Mail_TLS_SSLboxes settings');
    exit;
  end;

  st.Read(TotalMail_TLS_SSLBoxes, sizeof(TotalMail_TLS_SSLBoxes));
  for i:=1 to TotalMail_TLS_SSLBoxes do
    begin
      new(Mail_TLS_SSLBoxesSettings[i]);
      st.Read(Mail_TLS_SSLBoxesSettings[i]^, sizeof(TMail_TLS_SSLBoxSettings));
    end;
  st.Free;
End;

procedure Tfrm_Mail_TLS_SSLSettings.LoadFormFromBox(box :
PMail_TLS_SSLBoxSettings);
Begin
  Ed_SMTPTServer.Text := box.SMTPTServer;

```

```

Ed_SMTPPort.Text      := inttostr(box.SMTPPort);
Ed_SMTPAccount.Text  := box.SMTPAccount;
Ed_SMTPPassword.Text:= box.SMTPPass;

Ed_POPServer.Text    := box.POPServer;
Ed_POPPort.Text      := inttostr(box.POPPort);
Ed_POPAccount.Text   := box.POPAccount;
Ed_POPPassword.Text  := box.POPPass;
Ed_POPEmail_TLS_SSL.Text    := box.POPEmail_TLS_SSL;

Ed_Subscript.Text    := box.Subscript;
case box.AuthType of
  atNone : Combo_SMTPTType.ItemIndex:=0;
  atLogin: Combo_SMTPTType.ItemIndex:=1;
end;

if box.CheckEvery<>0 then Combo_CheckEvery.Text:=inttostr(box.CheckEvery)
  else Combo_CheckEvery.Text:='Ніколи';

Check_RetrieveHeadersOnly.Checked := box.RetrieveOnlyHeaders;
Check_DeleteFromServer.Checked    := box.DeleteFromServer;

Check_DontOpenHTML.Checked:=box.OpenHTMLasTEXT;
case box.ifDuplicate of
  Ignore : Radio_ignore.Checked:=true;
  Recieve : Radio_Recieve.Checked:=true;
  Ask      : Radio_Ask.Checked:=true;
  DeleteFromServer : RadioDelete.Checked:=true;
end;
Check_DynamicFolders.Checked := box.DynamicFolders;
Check_ShowBalloon.Checked    := box.ShowBalloonOnNewMes;
End;

procedure Tfrm_Mail_TLS_SSLSettings.SaveFormToBox(box :
PMail_TLS_SSLBoxSettings);
Begin
  box.SMTPServer := Ed_SMTPServer.Text;
  box.SMTPPort   := strtoint(Ed_SMTPPort.Text);
  box.SMTPAccount := Ed_SMTPAccount.Text;
  box.SMTPPass:= Ed_SMTPPassword.Text;

  box.POPServer := Ed_POPServer.Text;
  box.POPPort   := strtoint(Ed_POPPort.Text);
  box.POPAccount := Ed_POPAccount.Text;
  box.POPPass   := Ed_POPPassword.Text;
  box.POPEmail_TLS_SSL := Ed_POPEmail_TLS_SSL.Text;

  box.Subscript := Ed_Subscript.Text;
  case Combo_SMTPTType.ItemIndex of
    0 : box.AuthType:=atNone;
    1 : box.AuthType:=atLogin;
  end;

  if AnsiUpperCase(Combo_CheckEvery.Text)<>'НИКОЛИ' then
    box.CheckEvery:=strtoint(Combo_CheckEvery.Text)
  else
    box.CheckEvery:=0;

  box.RetrieveOnlyHeaders := Check_RetrieveHeadersOnly.Checked;
  box.DeleteFromServer    := Check_DeleteFromServer.Checked;
  box.OpenHTMLasTEXT      := Check_DontOpenHTML.Checked;

  if Radio_ignore.Checked then box.ifDuplicate:=Ignore;
  if Radio_Recieve.Checked then box.ifDuplicate:=Recieve;
  if Radio_Ask.Checked then box.ifDuplicate:=Ask;
  if RadioDelete.Checked then box.ifDuplicate:=DeleteFromServer;

  box.DynamicFolders := Check_DynamicFolders.Checked;

```

```

box.ShowBalloonOnNewMes := Check_ShowBalloon.Checked;

End;

procedure Tfrm_Mail_TLS_SSLSettings.ClearForm;
Begin
Ed_SMTPServer.Text := 'smtp.<servername>.ru';
Ed_SMTPPort.Text := '25';
Ed_SMTPAccount.Text := '';
Ed_SMTPPassword.Text := '';

Ed_POPServer.Text := 'pop.<servername>.ru';
Ed_POPPort.Text := '110';
Ed_POPAccount.Text := '';
Ed_POPPassword.Text := '';
Ed_POPEmail_TLS_SSL.Text := '<nick>@<server>.ru';
Ed_Subscript.Text := 'Вася';

Combo_SMTPType.ItemIndex:=1;
Combo_CheckEvery.ItemIndex:=0;
End;

procedure Tfrm_Mail_TLS_SSLSettings.LoadSettingsFromRecords;
Begin
if TotalMail_TLS_SSLBoxes <> 0 then
begin
if Mail_TLS_SSLBoxesSettings[1]<>nil then
begin
CurrentMail_TLS_SSLBoxSettings:=Mail_TLS_SSLBoxesSettings[1];

LoadFormFromBox(CurrentMail_TLS_SSLBoxSettings);
ListBox_Mail_TLS_SSLBoxes.Selected[0]:=true;
PrevMail_TLS_SSLBoxSettings:=CurrentMail_TLS_SSLBoxSettings;
end;
end
else CurrentMail_TLS_SSLBoxSettings:=nil;
End;

// створення нової поштової скриньки
procedure Tfrm_Mail_TLS_SSLSettings.btn_NewMail_TLS_SSLBoxClick(Sender:
TObject);
var
i : word;
begin
if Ed_Mail_TLS_SSLBoxName.Text<>' ' then
begin
for i:=1 to ListBox_Mail_TLS_SSLBoxes.Count do
if AnsiUpperCase(ListBox_Mail_TLS_SSLBoxes.Items[i-
1])=AnsiUpperCase(Ed_Mail_TLS_SSLBoxName.Text) then
begin
Application.MessageBox('Таке ім'я облікового запису вже
існує!', 'Помилка');
exit;
end;

ListBox_Mail_TLS_SSLBoxes.Items.Add(Ed_Mail_TLS_SSLBoxName.Text);
GroupBox1.Visible:=true;
Radio_Ask.Checked:=true;
inc(TotalMail_TLS_SSLBoxes);
new(Mail_TLS_SSLBoxesSettings[TotalMail_TLS_SSLBoxes]);

CurrentMail_TLS_SSLBoxSettings:=Mail_TLS_SSLBoxesSettings[TotalMail_TLS_SSLBoxes
];
CurrentMail_TLS_SSLBoxSettings.Name:=Ed_Mail_TLS_SSLBoxName.Text;
if PrevMail_TLS_SSLBoxSettings<>nil then
SaveFormToBox(PrevMail_TLS_SSLBoxSettings);

```

```

PrevMail_TLS_SSLBoxSettings:=CurrentMail_TLS_SSLBoxSettings;
ClearForm;
SaveFormToBox(CurrentMail_TLS_SSLBoxSettings);
ListBox_Mail_TLS_SSLBoxes.Selected[ListBox_Mail_TLS_SSLBoxes.Count-
1]:=true;
{Тепер створити і заповнити нову поштову скриньку}
Mail_TLS_SSLBoxes.Add(Ed_Mail_TLS_SSLBoxName.Text);
frm_Main.AddToLog('Створена поштова скринька
<'+Ed_Mail_TLS_SSLBoxName.Text+'>');
frm_Main.ShowMail_TLS_SSLBoxesInTree;
end
else
Application.MessageBox('Спочатку введіть ім'я вашої поштової
скриньки', 'Помилка');
end;

procedure Tfrm_Mail_TLS_SSLSettings.FormShow(Sender: TObject);
var
i : word;
begin
if (ListBox_Mail_TLS_SSLBoxes.Count=0)and(TotalMail_TLS_SSLBoxes<>0) then
for i:=1 to TotalMail_TLS_SSLBoxes do
ListBox_Mail_TLS_SSLBoxes.Items.Add(Mail_TLS_SSLBoxesSettings[i].Name);
LoadSettingsFromRecords;
if CurrentMail_TLS_SSLBoxSettings=nil then GroupBox1.Visible:=false
else GroupBox1.Visible:=true;
end;

procedure Tfrm_Mail_TLS_SSLSettings.Combo_CheckEveryExit(Sender: TObject);
begin
if AnsiUpperCase(Combo_CheckEvery.Text)<>'НИКОЛИ' then
begin
try
strtoint(Combo_CheckEvery.Text);
except
Application.MessageBox('Невірне число', 'Помилка');
Combo_CheckEvery.Text:='Ніколи';
exit;
end;
end;

procedure Tfrm_Mail_TLS_SSLSettings.btn_ApplyClick(Sender: TObject);
var
i : word;
begin
if ListBox_Mail_TLS_SSLBoxes.ItemIndex>=0 then
begin
SaveFormToBox(CurrentMail_TLS_SSLBoxSettings);
end;
SaveSettingsToFile;
self.Hide;
for i:=1 to TotalMail_TLS_SSLBoxes do
if Mail_TLS_SSLBoxes.GetByIndex(i)<>nil then
Mail_TLS_SSLBoxes.GetByIndex(i).SetTimerInterval(Mail_TLS_SSLBoxesSettings[i].Ch
eckEvery);
end;

function FindMail_TLS_SSLBox(name : shortstring):PMail_TLS_SSLBoxSettings;
var
i : word;
Begin
FindMail_TLS_SSLBox:=nil;

```

```

for i:=1 to TotalMail_TLS_SSLBoxes do
  if Mail_TLS_SSLBoxesSettings[i].Name = name then
  begin
    FindMail_TLS_SSLBox:=Mail_TLS_SSLBoxesSettings[i];
    exit;
  end;

End;

procedure Tfrm_Mail_TLS_SSLSettings.ListBox_Mail_TLS_SSLBoxesClick(Sender:
TObject);
begin
  if ListBox_Mail_TLS_SSLBoxes.ItemIndex >=0 then
  begin

CurrentMail_TLS_SSLBoxSettings:=FindMail_TLS_SSLBox(ListBox_Mail_TLS_SSLBoxes.It
ems[ListBox_Mail_TLS_SSLBoxes.itemindex]);
  if CurrentMail_TLS_SSLBoxSettings=nil then
CurrentMail_TLS_SSLBoxSettings:=PrevMail_TLS_SSLBoxSettings;
  if PrevMail_TLS_SSLBoxSettings<>nil then
SaveFormToBox(PrevMail_TLS_SSLBoxSettings);
  PrevMail_TLS_SSLBoxSettings:=CurrentMail_TLS_SSLBoxSettings;
  LoadFormFromBox(CurrentMail_TLS_SSLBoxSettings);
  end;
end;

procedure Tfrm_Mail_TLS_SSLSettings.FormCreate(Sender: TObject);
begin
  LoadSettingsFromfile;
end;

// видалення поштової скриньки
procedure Tfrm_Mail_TLS_SSLSettings.BitBtn1Click(Sender: TObject);
var
  i,j : word;
begin
  if ListBox_Mail_TLS_SSLBoxes.ItemIndex >= 0 then
  begin
    if MessageDlg('Ви впевнені, що хочете видалити обліковий запис
"' + ListBox_Mail_TLS_SSLBoxes.Items[ListBox_Mail_TLS_SSLBoxes.itemindex] + '" ?',
mtConfirmation, [mbYes, mbNo], 0) = mrNo then exit;

    for i:=1 to TotalMail_TLS_SSLBoxes do
      if Mail_TLS_SSLBoxesSettings[i]^ .Name =
ListBox_Mail_TLS_SSLBoxes.Items[ListBox_Mail_TLS_SSLBoxes.itemindex] then
        begin
          for j:=i to TotalMail_TLS_SSLBoxes-1 do
            Mail_TLS_SSLBoxesSettings[j]:=Mail_TLS_SSLBoxesSettings[j+1];
          break;
        end;
        frm_Main.AddToLog('Видалено скриньку
<' + ListBox_Mail_TLS_SSLBoxes.Items[ListBox_Mail_TLS_SSLBoxes.itemindex] + '>');
        ListBox_Mail_TLS_SSLBoxes.DeleteSelected;
        dec(TotalMail_TLS_SSLBoxes);
        try
          ListBox_Mail_TLS_SSLBoxes.Selected[0]:=true;
          ListBox_Mail_TLS_SSLBoxesClick(self);
          frm_Main.ShowMail_TLS_SSLBoxesInTree;
        except end;
      end;
    end;
  end;

procedure Tfrm_Mail_TLS_SSLSettings.FormClose(Sender: TObject;
var Action: TCloseAction);
begin

```

```
    btn_ApplyClick(self);  
end;  
  
end.
```

К6П3_2024

Файл frmSettings.pas - параметри програми

```

unit frmSettings;

interface

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

type
  Tfrm_Settings = class(TForm)
    GroupBox1: TGroupBox;
    Check_AllowLog: TCheckBox;
    L_MaxLogSize: TLabel;
    Ed_MaxLogSize: TEdit;
    L_kb: TLabel;
    btn_Close: TBitBtn;
    RadioShowInTaskBar: TRadioButton;
    RadioShowInTray: TRadioButton;
    procedure Check_AllowLogClick(Sender: TObject);
    procedure Ed_MaxLogSizeChange(Sender: TObject);
    procedure Ed_MaxLogSizeKeyPress(Sender: TObject; var Key: Char);
    procedure FormCreate(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btn_CloseClick(Sender: TObject);
    procedure FormHide(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure LoadGlobalSettingsToForm;
    procedure LoadGlobalSettingsFromFile;
    procedure SaveGlobalSettingsFromForm;
    procedure SaveGlobalSettingsToFile;
  end;

  TSettings = record
    AllowLog : boolean;
    MaxLogSize: longint; // kb
    ShowInTray: boolean;
  end;

const
  GlobalSettingsFileName = 'globalsettings.dat';

var
  frm_Settings: Tfrm_Settings;
  Settings : TSettings=(AllowLog:true);

implementation

uses frmMain;

{$R *.dfm}

procedure Tfrm_Settings.FormCreate(Sender: TObject);
begin
  // Не працює { Нам потрібно це в frmMain.onCreate }
  LoadGlobalSettingsFromFile;
end;

procedure Tfrm_Settings.FormShow(Sender: TObject);
begin

```

```

    LoadGlobalSettingsToForm;
end;

procedure Tfrm_Settings.Check_AllowLogClick(Sender: TObject);
var
    b : boolean;
begin
    if Check_AllowLog.Checked then b:=true
        else b:=false;

    L_MaxLogSize.Visible:=b;
    Ed_MaxLogSize.Visible:=b;
    L_kb.Visible:=b;
    Settings.AllowLog:=b;
    Settings.MaxLogSize:=strtointdef(Ed_MaxLogSize.text,500);
end;

procedure Tfrm_Settings.Ed_MaxLogSizeChange(Sender: TObject);
var
    maxs : longint;
begin
    try
        if Ed_MaxLogSize.Text<>' ' then
            maxs:=strtoint(Ed_MaxLogSize.text);
        except
            Application.MessageBox('Розмір лог файлу може бути від 0 до 2147483647
kb', 'Невірно вказане значення');
            Ed_MaxLogSize.Text:='500';
            exit;
        end;
        Settings.MaxLogSize:=maxs;
    end;
end;

const
    Digits :set of char =['1','2','3','4','5','6','7','8','9','0','#'];

procedure Tfrm_Settings.Ed_MaxLogSizeKeyPress(Sender: TObject; var Key: Char);
begin
    if not (key in Digits) then Key:=#0;
end;

procedure Tfrm_Settings.LoadGlobalSettingsToForm;
begin
    Ed_MaxLogSize.Text := inttostr(Settings.MaxLogSize);
    Check_AllowLog.Checked := Settings.AllowLog;
    RadioShowInTray.Checked := Settings.ShowInTray;
end;

procedure Tfrm_Settings.SaveGlobalSettingsFromForm;
begin
    Settings.AllowLog := Check_AllowLog.Checked;
    Settings.MaxLogSize := strtointdef(Ed_MaxLogSize.text,500);
    Settings.ShowInTray := RadioShowInTray.Checked;
end;

procedure Tfrm_Settings.LoadGlobalSettingsFromFile;
var
    st : TFileStream;
    SFName : shortstring;
begin
    SFName:=ExtractFilePath(Application.ExeName)+GlobalSettingsFileName;
    try
        st:=TFileStream.Create(SFName, fmOpenRead or fmShareDenyNone);
    end;
end;

```

```

except
  try
    st:=TFileStream.Create(SFName, fmCreate or fmShareDenyNone);
  except
    Application.MessageBox(pchar('Неможливо відкрити/створити файл
''+SFName+'''), 'Помилка завантаження глобальних параметрів');
    exit;
  end;
  fillchar(Settings, sizeof(Settings), 0);
  LoadGlobalSettingsToForm;
  st.Free;
  exit;
end;
fillchar(Settings, sizeof(Settings), 0);
st.Read(Settings, sizeof(Settings));
st.Free;
end;

procedure Tfrm_Settings.SaveGlobalSettingsToFile;
var
  st      : TFileStream;
  SFName  : shortstring;
begin
  SFName:=ExtractFilePath(Application.ExeName)+GlobalSettingsFileName;
  try
    st:=TFileStream.Create(SFName, fmOpenWrite or fmShareDenyNone);
  except
    try
      st:=TFileStream.Create(SFName, fmCreate or fmShareDenyNone);
    except
      Application.MessageBox(pchar('Неможливо відкрити/створити файл
''+SFName+'''), 'Помилка завантаження глобальних параметрів');
      exit;
    end;
  end;

  st.Write(Settings, sizeof(Settings));
  st.Free;
end;

procedure Tfrm_Settings.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  SaveGlobalSettingsFromForm;
  SaveGlobalSettingsToFile;
end;

procedure Tfrm_Settings.btn_CloseClick(Sender: TObject);
begin
  self.Hide;
end;

procedure Tfrm_Settings.FormHide(Sender: TObject);
begin
  SaveGlobalSettingsFromForm;
  SaveGlobalSettingsToFile;
  if Settings.ShowInTray then
  begin
    frm_Main.Tray.IconVisible:=true;
    frm_Main.Tray.HideTaskbarIcon;
    frm_Main.Tray.MinimizeToTray:=true;
  end
  else
  begin
    frm_Main.Tray.IconVisible:=false;
    frm_Main.Tray.ShowMainForm;
    frm_Main.Tray.ShowTaskbarIcon;
  end;
end;

```

```
    frm_Main.Tray.MinimizeToTray:=false;  
end;  
end;  
end.
```

КБПЗ_2024

Файл about.pas - файл довідки

```
unit about;

interface

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

type
  TForm1 = class(TForm)
    Image1: TImage;
    BitBtn1: TBitBtn;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form1.Close;
end;

end.
```