

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
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему

“Програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення”

Виконав здобувач вищої освіти
IV курсу, групи КБ-19
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Колесник Є.В.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
« ____ » _____ 2023 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 "Інформаційні технології"
Спеціальність 125 "Кібербезпека"
Освітньо-професійна (освітньо-наукова) програма "Кібербезпека"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2023 року

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

Колеснику Євгену Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення*

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

затверджені наказом вищого навчального закладу № 12-02 від 5.01.2023 року

3. Строк подання студентом роботи до захисту *23.05.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення*

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 кібербезпеки в промислову експлуатацію.

6. Висновки

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

Структурна схема системи кібербезпеки *1 аркуш*

Функціональна схема системи кібербезпеки *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

7. Дата видачі завдання « 17 » січня 2023 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	10.03.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2023 р.	
3.	Розробка моделі компонента	20.03.2023 р.	
4.	Розробка структур даних	25.03.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2023 р.	
6.	Програмування алгоритмів	10.04.2023 р.	
7.	Оформлення ПЗ	17.04.2023 р.	
8.	Попередній захист роботи	23.05.2023 р.	

Дата видачі завдання
« 17 » січня 2023 р.

Підпис керівника

Смірнов О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2023 р.

Підпис здобувача

Колесник Є.В.
(прізвище та ініціали)

АНОТАЦІЯ

Колесник Є.В. Програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

Метою розробки є програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

Результат роботи – програмна реалізація системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

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 архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.

Ключові слова: кібербезпека, захищена електронна пошта

ABSTRACT

Kolesnyk E.V. Cyber security system software for a special purpose secure e-mail client. 125 Cyber security. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

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 client of special-purpose protected e-mail.

The purpose of the development is the cyber security system software of the client of a special purpose protected e-mail.

The result of the work is a software implementation of a cyber security system for a client of special purpose protected e-mail.

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 PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10 environment.

Keywords: cyber security, secure e-mail

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	24
2.3 Розгорнута постановка завдання	30
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	31
3.1 Опис функціонування системи	31
3.2 Розробка структурної схеми.....	43
3.3 Розробка функціональної схеми	49
3.4 Розробка діаграми процесів.....	55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	58
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	58
4.2 Захист розробленого програмного забезпечення.....	74
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	79
6 ОСНОВНІ ВИСНОВКИ.....	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	84

						ВКРБ-125.23.0010.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
<i>Розроб.</i>	<i>Колесник Є.В.</i>				<i>Програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення</i>	Літ.	Аркуш	Аркушів
<i>Перев.</i>	<i>Смірнов О.А.</i>					Б	1	91
Н.контр.	<i>Гермак В.С.</i>				ЦНТУ КБ-19			
Затв.	<i>Смірнов О.А.</i>							

ВСТУП

Актуальність теми. Програми для поштових скриньок або поштові клієнти – це програми для роботи з поштою на локальному комп'ютері, які значно полегшують роботу з поштою. Завдяки спеціалізованим поштовим програмам-клієнтам користувачі можуть одержувати й відправляти електронні листи, зберігати й працювати з листами, а також багато чого іншого. Багато програм можуть у мінімізованому виді працювати й регулярно перевіряти поштову скриньку на наявність нових листів на сервері, тим самим про нові листи користувач довідається вчасно.

Популярність поштових клієнтів поступово знижується. Сучасна просунута молодь воліє працювати з поштою через веб-інтерфейс, використовуючи браузер. Дійсно, навіщо використовувати дві програми, коли досить освоїти одну. Підручних засобів, надаваних інтерфейсом популярних поштових сервісів, у загальному-то, досить для повсякденних потреб.

Однак використання спеціалізованих програм, у кожному разі, дає кілька приємних бонусів. Спілкуючись із поштовим клієнтом, ви ведете роботу з електронною кореспонденцією так, як зручно саме вам, а не веб-дизайнеру. При наявності декількох адрес електронної пошти освоєння однієї нової програми найчастіше виявляється менш трудомістким, ніж освоєння тонкостей веб-інтерфейсу кожного сервісу. Крім того, ви помітно заощаджуєте вхідний трафік, тому що не завантажуєте сайти сервісів. Поштові клієнти, найчастіше, надають гнучкі засоби сортування повідомлень, дозволяють вибрати шаблон з готового набору, роблять перевірку орфографії в міру набору тексту й мають інші корисні можливості, що робить роботу з електронною кореспонденцією більше зручною й ефективною.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

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

- Огляд існуючих систем клієнта захищеної електронної пошти спеціального призначення.
- Дослідження системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.
- Програмна реалізація системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі клієнта захищеної електронної пошти спеціального призначення.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Поштова програма (клієнт електронної пошти, поштовий клієнт, мейл-клієнт, мейлер) – програмне забезпечення, установлене на комп'ютері користувача й призначене для одержання, написання, відправлення й зберігання повідомлень електронної пошти одного або декількох користувачів (у випадку, наприклад, декількох облікових записів на одному комп'ютері) або декількох облікових записів одного користувача. Поштові клієнти дозволяють виконувати наступні дії у світі електронного документообігу:

- ви помітно заощаджуєте вхідний трафік, тому що не завантажуєте сайт сервісу;
- грамотне налаштування анти-спам фільтрів дозволяє ще більше заощадити трафік і зберегти нервову систему;
- гнучкі засоби сортування повідомлень, шаблони, перевірка орфографії на льоту й інші можливості поштових клієнтів роблять вашу роботу більш зручною й ефективною;
- використання поштового клієнта дозволяє вести роботу з електронною кореспонденцією так, як зручно саме вам, а не Web-дизайнерові;
- при наявності декількох адрес електронної пошти освоєння однієї нової програми найчастіше виявляється менш трудомістким, ніж вникання в тонкості Web-інтерфейсу кожного сервісу.

Всі сучасні популярні поштові клієнти вміють працювати через протоколи POP і IMAP. Деякі розроблювачі додають можливість роботи із групами новин USENET і читання RSS-новин.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1.2 Область застосування

Через зростаюче значення електронної пошти як засобу комунікації зросли й очікування відносно функціональності поштових клієнтів. Найважливіші вимоги, які сьогодні пред'являються до них, припускають, серед іншого, доступ до декількох поштових скриньок з одного клієнта, мобільну перевірку пошти, включення груп користувачів, що не мають власних комп'ютерів у корпоративну комунікацію, за допомогою так званих кіоскових систем і простий доступ для тимчасових співробітників або провайдерів послуг. Поряд з повноцінною інтеграцією функціональності електронної пошти в корпоративні портали важлива роль приділяється питанням безпеки електронної пошти й простоті обігу з великою кількістю вхідних повідомлень.

Очевидно, що жодна технологія не може однаково добре відповідати всім вимогам. Однак відповідність деяким з них приводить до росту популярності клієнтів електронної пошти на базі Web. Сьогодні на ринку можна виділити чотири групи клієнтів:

- "традиційні" (Microsoft Outlook, IBM Lotus Notes або клієнт GroupWise від Novell) – поряд з функціями електронної пошти й календаря вони в різному обсязі пропонують і додаткові послуги. Як приклади можна привести функції колективної роботи Lotus Notes або підтримку керування документообігом;
- Microsoft Outlook Express або Mozilla Thunderbird забезпечують меншу функціональність, але все-таки працюють як локальні додатки під керуванням Windows або інших операційних систем;
- клієнти на базі Web великих платформ обміну повідомленнями, тобто відповідні розширення для Microsoft Exchange Server (Outlook Web Access), IBM Lotus Domino (Domino Web Access) і Novell GroupWise (GroupWise Web Access);
- інші клієнти Web від постачальників послуг Internet, зокрема GMX або Yahoo!. Однак ці клієнти звичайно мають лише базову функціональність.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Проміжне положення займає IBM Workplace Messaging. Цей продукт може використовуватися як через браузер Web, так і через Rich Client на базі рішення Eclipse з відкритими кодами. Тим самим його функціональність перевищує можливості простого клієнта на базі Web, однак він не порівнянний, наприклад, з Lotus Notes. По продуктивності його можна віднести до другої групи – до таких програм, як Outlook Express або Thunderbird.

Прості клієнти, надавані провайдерами Internet, застосовуються найчастіше приватними користувачами або в якості «обхідного шляху» для доступу до електронної пошти, коли інші механізми доступу, такі, як POP3, не функціонують. Однак вони мало пристосовані для обробки більших обсягів електронної пошти. Будучи обмеженими винятково функціональністю електронної пошти, не підтримуючий календар або керування завданнями, що вимагають порівняно складного обслуговування, ці системи навряд чи можуть використовуватися в якості кіоскових або застосовуватися мобільними співробітниками на підприємстві.

Клієнти на базі Web ведучих систем обміну повідомленнями, навпроти, здатні продемонструвати широкий функціональний охоплення, що усе більше наближається до функціонального охоплення товстих клієнтів – Microsoft Outlook, наприклад. Вони підходять для різних сценаріїв застосування, однак завжди припускають наявність власної інфраструктури обміну повідомленнями, оскільки є частиною таких систем, як Microsoft Exchange, Lotus Domino або Novell GroupWise.

Звичайно вони затребувані, коли користувачам потрібний доступ до всіх функцій електронної пошти, але можливість локального запуску товстих клієнтів відсутня. У випадку поштового клієнта на базі Web можна обійтися без локальної інсталяції, навіть якщо (почасти це залежить від бажаної функціональності) потрібна установка локальних компонентів. Для кіоскових систем і зовнішніх співробітників, які, приміром, входять у корпоративну мережу із власних ноутбуків, такі клієнти являють собою раціональну опцію. З ними можуть

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

працювати й користувачі інших операційних систем, а не тільки Windows, при цьому необхідно щораз перевіряти можливість роботи із застосовуваним браузером. Lotus Domino Web Access підтримує, наприклад Mozill, а тим самим і істотною незалежністю від платформи.

Коли доступ не може бути зроблений із власного персонального комп'ютера або ноутбука, клієнти Web становлять інтерес і для мобільних користувачів. Вони пропонують майже такого ж функціонала, як і відповідні клієнти на базі Windows – тільки через Web. Якщо співробітник, навпроти, працює на ноутбуці або ПК будинку, він може використовувати стандартні клієнти – Microsoft Outlook або Lotus Notes. У цьому випадку доступ відбувається по Post Office Protocol (POP3 – поштовий протокол) або Internet Message Access Protocol (IMAP – протокол доступу до повідомлень в Internet), через VPN або такі протоколи, як Notes Remote Procedure Calls (NRPC – віддалений виклик процедур Notes). Використання Microsoft Outlook через POP3, приміром, помітно ефективніше, ніж доступ через клієнта Web, коли в першу чергу завантажуються тільки заголовки листів.

Внаслідок відсутності необхідності в розподілі програмного забезпечення в багатьох з'являється бажання повністю перейти на клієнти такого типу. Однак перш, ніж зважитися на подібний крок, треба оцінити, з одного боку, функціональність систем, а з іншого боку – їхню безпеку.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

бути зрозумілий і знайомий користувачеві Linux, але далекий людині, все життя проробило в Windows. Панелі інструментів не можна набудувати. Це означає, що ви не можете додавати або видаляти кнопки, приховувати текстові підписи, міняти дизайн піктограм. Це відмінна риса всіх додатків GNOME, і із цим прийде упокоритися. Інтерфейс працює нестерпно повільно. Відкриття будь-якого діалогового вікна змушує програму задуматися на кілька секунд. У всіх діалогових вікнах відсутні кнопки «ОК» і «Застосувати», що традиційно для робітничого середовища GNOME. Як тільки ви внесете яку-небудь зміну в налаштування програми, ваші коректування відразу набувають чинності – додаткового застосування або підтвердження дій користувача не потрібно. Процедура відправлення й одержання повідомлень докладно відображається усередині спеціальних інформаційних вікон: ви бачите поточний статус з'єднань, кількість прийнятої й переданої інформації, обчислювальної одночасно в кількості повідомлень і кілобайтах.

Для обробки кореспонденції у форматі HTML використовується движок Gecko, застосовуваний у програмних продуктах сімейства Mozilla. Він забезпечує високу точність відображення інформації, але не є причиною появи серйозних проломів у безпеці. Evolution відмінно працює з кирилицею. Не тільки тіла повідомлень, але і їхні теми завжди відображаються коректно, без яких-небудь проблем, навіть без необхідності ручної зміни кодування.

Створення нових повідомлень відбувається в повноцінному HTML-редакторі. У налаштуваннях програми можна вказувати формат повідомлень за замовчуванням відповідно до власних переваг, щоб постійно не перемикати режим у налаштуваннях. Ви можете додавати до листів підпису. Крім підпису, використовуюваної за замовчуванням, можна вибирати додаткові елементи зі спеціального списку. Підпис може бути представлена у вигляді простого тексту або у вигляді HTML і навіть містити спеціальні сценарії, що дозволяють автоматично міняти її вид залежно від яких-небудь параметрів. Наприклад, можна призначити вивід унікального текстового рядка при відповіді на

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Evolution підтримує сортування пошти по окремих папках. Ви можете вручну створювати вкладені папки й набудувувати фільтри, за допомогою яких листи будуть копіюватися або переноситися в зазначені вами файлові структури. При цьому на жорсткому диску створюються нові файли, пошта фізично сортується в папках. Крім того, поштовий клієнт підтримує створення віртуальних папок. Ви робите пошук повідомлень у якій-небудь фізичній папці, а його результати, у свою чергу, зберігаєте як папку. Виходить, що інформація в цьому другому випадку в другій папці групується лише логічно, віртуально, поштова база при цьому не міняється, що забезпечує дуже високу швидкість пошуку й сортування вхідної кореспонденції. Програма містить потужних інструмент ручного сортування листів. Ви можете створювати складні умови фільтрації повідомлень, що включають у себе різні логічні вираження. Evolution також має вбудований навчаємий інструмент виявлення рекламних повідомлень. Крім того, поштовий клієнт може взаємодіяти зі спеціальним програмним забезпеченням, установленим на поштових серверах. Ґрунтуючись на даних, отриманих від віддаленого додатка, можна робити фільтрацію реклами без необхідності її завантаження на локальну робочу станцію.

Крім чисто поштових функцій Evolution має широкі можливості спільної роботи над проектами, дозволяє планувати завдання, вести замітки, використовувати мережні календарі. Контакти ви можете зберігати не тільки на локальній робочій станції, але й на серверах LDAP. Evolution може бути використаний і як повноцінний планувальник подій. При цьому вам буде доступна робота з мережними календарями, що дозволяє здійснювати єдине планування для великої кількості користувачів. Для цього є відповідний і досить розвитий інструментарій. Функціональність Evolution може бути збільшена за допомогою модулів, що підключаються. У базовий дистрибутив поштового клієнта включені інструменти публікації календарів, роботи із прогнозами погоди й багато чого іншого. Довгий час Evolution поширювався тільки для операційної системи Linux, це одна із причин низької популярності настільки потужного

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

програмного продукту, що становить реальну конкуренцію Microsoft Outlook. Втім, для серйозного завоювання ринку Evolution не вистачає продуктивності й гнучкості інтерфейсу. Перед нами потужний, але не дуже ергономічний продукт.

Foxmail

Foxmail – поштова програма, створена китайськими розроблювачами. Її офіційні збірки поширюються на китайській і англійській мовах. Файл русифікації інтерфейсу існує, однак завантажується й встановлюється окремо.

Foxmail володіє традиційним для багатьох поштових клієнтів інтерфейсом. Ліва бічна панель містить список поштових папок, а також групу передплат на новинні RSS-стрічки. Основна частина робочого вікна поділена на дві частини: зверху перебуває список повідомлень у поточній папці, а знизу відображається вміст обраного в цей момент листа. Дане компонування робочого вікна може легко мінятися за допомогою групи кнопок, розташованих у правій частині статусного рядка. Програма дозволяє самому вибрати колір інтерфейсу. За замовчуванням пропонується синій, але ви можете позбавити її «особу» яскравості, зробивши його, наприклад, сірим. Зміна дизайну кнопок не підтримується. Поштовий клієнт здійснює прийом і відправлення пошти з використанням протоколів POP3 і SMTP. Підтримка поштових скриньок IMAP відсутня. Ви можете імпортувати налаштування з попередньої версії Foxmail, а також з Outlook Express. Допускається також перенос облікових записів і поштових баз. При створенні нового облікового запису програма дозволяє автоматично заповнювати деякі поля налаштувань. Наприклад, при уведенні даних ящика Gmail відразу вказуються коректні порти POP3 і SMTP, типи з'єднань, а також прописуються імена, необхідні для доступу до серверів. Крім Gmail програма окремо підтримує поштові скриньки Hotmail, MSN і Yahoo!.

Адресна книга може зберігатися на віддаленому сервері, до якого відкритий доступ тільки певним користувачам. Подібне рішення дозволяє використовувати поштовий клієнт у корпоративних мережах. Можливий імпорт адрес із текстових файлів, а також з файлів *.WAB і *.CSV. Крім того, при

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

необхідності ви без особливої праці імпортуєте адресні книги Outlook Express, MS Outlook, Eudora і всіх поштових програм Mozilla. Допускається синхронізація даних із зазначеної в налаштуваннях папкою, підтримуються облікові записи LDAP.

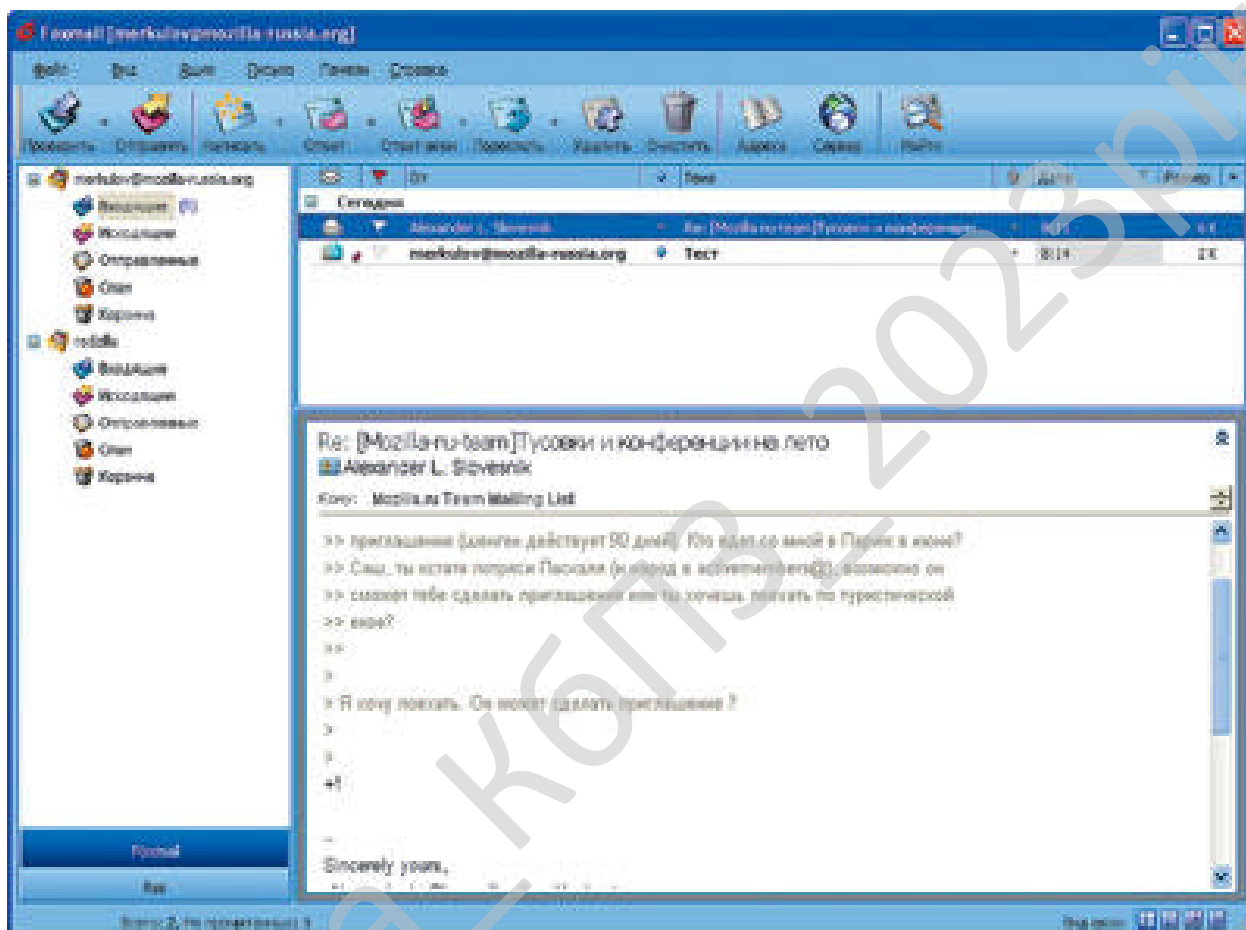


Рисунок 2.2 – Интерфейс користувача Foxmail

Замість одержання нових повідомлень можна просто перевірити наявність пошти на сервері. Програма має зручного менеджера роботи з віддаленим сервером, що дозволяє переглядати заголовки, указувати окремі повідомлення для завантаження, видаляти листи, блокувати адресатів.

Для перегляду повідомлень використовуються компоненти Internet Explorer. Ви можете читати листа в будь-якому кодуванні, що використовує кирилицю, – автоматичне визначення кодування тексту працює без збоїв.

Невелика проблема полягає в тому, що, якщо кодування заголовка повідомлення і його тіл відрізняються, у списку повідомлень текст теми буде відображений з перекручуваннями. Ручне перемикання кодування відновлює тему, однак порушує читабельність самого листа. Всі повідомлення сортуються в групи по даті їхнього одержання. Допускається приховання або, навпроти, розкриття будь-яких груп. Ви можете переглядати листи, отримані сьогодні, учора, на цьому тижні, минулого тижня, а також повідомлення, вік яких перевищує зазначені строки. Швидкий пошук інформації в поштової базі здійснюється за допомогою виклику спеціальної панелі, розташованої над списком кореспонденції. Як критерій пошуку може виступати тема, відправник, сполучення перших двох полів, а також тіло повідомлення. Складний пошук викликається за допомогою традиційного сполучення Ctrl + F, він реалізований у вигляді окремого вікна. При його виклику допускається додаткова вказівка якоїсь певної поштової папки (за замовчуванням пошук ведеться у всій базі), а також тимчасового діапазону, у якому були отримані потрібні вам повідомлення. Ви можете також шукати листи з певним прапором, що містять вкладені файли або позначені як реклама.

Створення нових повідомлень виробляється за допомогою потужного HTML-редактора. Швидко перемикання між режимами HTML і простого тексту здійснюється прямо з панелі інструментів. Ви можете не просто здійснювати форматування тексту, але також додавати в листи таблиці, фоновий рисунок, звичайні зображення, фонову музику. Крім того, редактор містить інструмент створення знімків робочого стола, а також дозволяє захоплювати окремі робочі вікна. Після створення графічного файлу він автоматично прикріплюється до повідомлення. Допускається ручний вибір кодування тексту, за замовчуванням листа створюються з використанням CP1251 – кодування Windows. Редактор підтримує стилістичне оформлення тексту. При створенні відповідей вихідне поштове повідомлення цитується, однак традиційного виділення цитат символом «>>» не відбувається. Втім, не варто звертати пильну увагу на тонкостях цитування й на інших деталях оформлення листів. Важливою перевагою Foxmail

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

є розвинені інструменти створення користувальницьких шаблонів. Одна справа – формально реалізувати цю функцію, інше – зробити це якісно, але доступно для рядового користувача. Редактор шаблонів Foxmail дозволяє гнучко управляти шаблонами нових листів, відповідями, пересиланнями. Ви можете зробити довільну кількість шаблонів і підписів, а потім вибрати їх зі списку при створенні повідомлень. Редактор підтримує велику кількість макросів, користувальницьких змінних. У вікні попереднього перегляду шаблону ви відразу бачите графічну структуру майбутніх листів. Комбінуючи простий текст і макроси, можна створювати, наприклад, окремі шаблони для особистої переписки й для ділових повідомлень. Крім того, тим, хто багато спілкується з іноземними партнерами, зручно створити дві групи шаблонів – для переписки українською мовою й на якому-небудь іншому. Підписи можуть бути створені у форматі HTML і містити в собі не тільки відформатований текст, але й графіку.

Foxmail має кілька цікавих мережних можливостей. Ви можете збирати листи з декількох POP3-серверів у загальну поштову скриньку в рамках єдиного облікового запису. Допускається пряме відправлення повідомлень, мінаючи сервер SMTP. Для цього в налаштуваннях поштового клієнта необхідно вказати адреси спеціальних серверів DNS. Програма містить вбудований інструмент дозвону до провайдеру, необхідний при використанні з'єднання, що комутирується. Foxmail володіє відразу декількома інструментами сортування й фільтрації вхідної пошти. Виявлення рекламних повідомлень може відбуватися на основі жорстко описаних правил, а також за допомогою алгоритму Байєса. У другому випадку необхідно попередньо натренувати програму, після чого, ґрунтуючись на отриманому досвіді, вона буде самостійно виявляти небажану пошту. Рекламні повідомлення можуть відразу віддалятися в сміттєвий кошик, а також попередньо міститися в спеціальну папку «Спам». Фільтрація реклами на основі набору правил має на увазі збірки рейтингу небажаності для кожного повідомлення. У випадку перевищення критичного значення рейтингу лист буде автоматично позначено як спам. Навчання механізму фільтрації не обов'язково

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

повинне бути сполучене із тривалим сортуванням рекламних повідомлень вручну. Поштовий клієнт містить інструмент швидкого навчання, при якому аналізується папка «Вхідні», де ви вказуєте рекламні повідомлення, що різко підвищує ефективність фільтрації. Foxmail постачений також традиційними «чорним» і «білим» списками. В «білий список» автоматично заносяться електронні адреси тих кореспондентів, яким відправляли листа ви.

Сортування повідомлень може вироблятися автоматично в папках «Вхідні», «Вихідні», а також вручну після виділення групи листів. Допускається створення складних умов, у яких бере участь кілька параметрів. Умови можуть зв'язуватися декількома логічними змінними. Як результат сортування пропонується великий вибір дій: ви можете копіювати або переносити повідомлення в інші папки, автоматично створювати відповіді, пересилати листа. Допускаються й деякі специфічні дії, наприклад створення довільної мітки, відтворення унікального звукового сигналу, запуск зовнішнього додатка, автоматичний показ повідомлення.

Foxmail дозволяє не тільки працювати з електронною поштою, але й підписуватися на RSS-новини. При перемиканні в режим RSS міняється панель інструментів і головне меню програми. Всі новинні стрічки можуть бути довільно згруповані. Ви можете підписатися на новинну стрічку, указавши адресу, місце розташування OPML або ключові слова. Foxmail володіє однією особливістю, що багатьом стала відома завдяки високій популярності молдавського поштового клієнта The Bat!. Повідомлення про прихід нової кореспонденції можуть відображатися усередині рядка, що біжить, що може розташовуватися в будь-якому місці робочого стола. Спочатку на ній відображається кількість нових повідомлень, а потім перед очима повільно пропливають їхні заголовки, що включають у себе вміст полів «Від» і «Тема». Налаштування інструмента дозволяють указувати висоту рядка, швидкість руху тексту, дизайн її підкладки, колір шрифту. При видаленні повідомлень із бази відбувається лише їхня позначка, розмір файлів не міняється. Програма дозволяє здійснювати стиск

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

поштової бази з метою економії дискового простору. Крім того, можете створювати архіви з поштовими повідомленнями. У випадку порушення цілісності бази Foxmail дозволяє швидко відновити дані з резервної копії.

Foxmail – компактний, швидкий поштовий клієнт. Він не перевантажений складними, рідко затребуваними функціями. Основні його переваги полягають у гнучкій, але дуже простій роботі із шаблонами, потужних алгоритмах фільтрації рекламних повідомлень, якісній підтримці HTML. Програма працює з кирилицею, хоча далеко не ідеально, з огляду на проблеми з відображенням заголовків листів. Зате приємним бонусом є зручний інструмент читання RSS-новин. Недоліки Foxmail зводяться скоріше до відсутності певних функцій, чим до помилок роботи програми. Ви не можете створювати поштові скриньки IMAP. Програма не підтримує віртуальні папки, що змушує здійснювати тільки тверду фільтрацію пошти. В Foxmail відсутня перевірка орфографії українською мовою. Незважаючи на перераховані недоліки, в особі Foxmail ми маємо програмний продукт, що якісно виконує закладені в нього функції.

Mozilla Thunderbird

Mozilla Thunderbird з'явилася в 2004 році в результаті розпаду інтегрованого набору інструментів Mozilla Suite на незалежні компоненти. Поштовий клієнт дозволяє працювати з електронною кореспонденцією через протоколи POP, SMTP і IMAP, брати участь у конференціях Usenet, а також здійснює передплату на новинні стрічки RSS. При першому старті програми вам пропонується налаштувати обліковий запис. Якщо ви маєте поштову скриньку на сервісі Gmail, то все налаштування відбувається автоматично – вам потрібно лише ввести ім'я облікового запису й пароль. При необхідності допускається імпорт пошти, настановь і адресної книги із програм Outlook, Outlook Express і Eudora. В основі поштового клієнта Thunderbird лежить движок Gecko, використовуваний всіма додатками сімейства Mozilla. Завдяки йому Thunderbird не тільки без проблем відображає будь-яке форматування HTML, але й має високу безпеку. Програма має потужний інтерфейс, що набудовується. Ви

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

реалізована без допомоги якогось окремого менеджера, прямо в рамках єдиного робочого вікна додатка. У налаштуваннях може бути зазначена необхідність попереднього завантаження повідомлень. Ви відкриваєте який-небудь лист, і якщо його заголовок не несе в собі ознак спама, клацаєте по посиланню й завантажуєте тіло повідомлення.

Поштовий клієнт відображає цитування за допомогою вертикальної риси. Даний спосіб застосуємо не тільки до HTML, але й до текстового режиму. Програма допускає виділення довільним кольором до п'яти рівнів цитування. При цьому можна міняти не тільки колір шрифту, але й тло, а також колір вертикальних смуг. Thunderbird звичайно не має проблем з відображенням кирилиці як у самих повідомленнях, так і в їхніх темах. Якщо раптом з'явилися якісь порушення в читабельності заголовків, то досить один раз перемкнути кодування папки, що у українській версії за замовчуванням має значення KOI 8-R.

Створення повідомлень відбувається в HTML-редакторі, що створює дуже компактний HTML-Код. І це не дивно, адже ядро модуля писалося в часи Mozilla Suite, до складу якого входив Composer, спеціальний інструмент для створення HTML-сторінок. Thunderbird перевіряє орфографію в міру набору тексту. При цьому можуть використовуватися як англійський, так і український словники. Крім того, можна окремо завантажити спільний англо-український словник, що дозволить на льоту виділяти помилки відразу на двох мовах.

Зміна шаблонів повідомлень доступно лише за допомогою виправлення конфігураційних файлів. Втім, ви можете скористатися модулями розширення, серед яких неважко знайти програми, що дозволяють без праці створювати різноманітні шаблони. Завдяки відкритості Thunderbird багато хто з популярних функцій, відсутніх у базовому дистрибутиві поштового клієнта, можуть бути реалізовані за допомогою плагінів. Ніхто адже не переживає, що Miranda не підтримує вкладки. Для цього є спеціальний плагін. У цьому випадку ситуація аналогічна. Для поштового клієнта написана безліч модулів розширення. За

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

допомогою їх можна автоматично витягати вкладення з повідомлень, створювати гарні шаблони й строкато розфарбовувати рівні цитування. Розширення дозволяють виводити яскраві інформативні повідомлення, набудовувати комбінації клавіш, використовувати зовнішні редактори й робити багато інше.

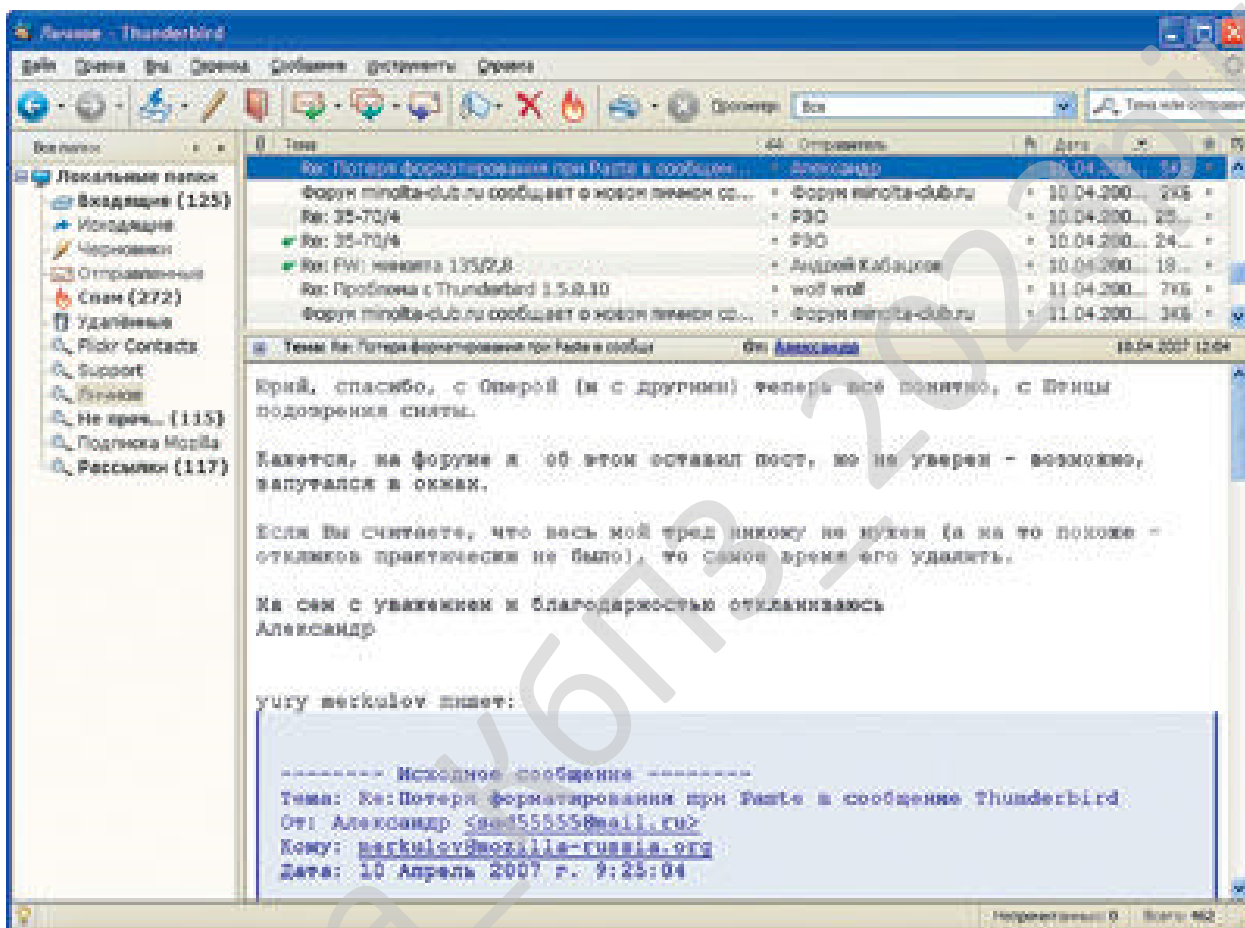


Рисунок 2.3 – Інтерфейс користувача Mozilla Thunderbird

Підключення підпису відбувається в налаштуваннях програми. Підпис може братися з текстового файлу або з HTML-документа. Поштовий клієнт підтримує сортування листів по фізичних і віртуальних папках. У другому випадку на жорсткому диску не відбувається створення додаткових структур, пошта зберігається в єдиній базі, а папки лише є результатами дій певних фільтрів. Ви можете також скористатися третім способом, що не має аналогів в інших поштових програмах. Thunderbird підтримує створення

користувальницьких міток для повідомлень. Їхня кількість не обмежена, кожному листу можна привласнити кілька міток. У результаті з'являється новий спосіб сортування пошти, пошуку повідомлень. Виявлення листів рекламного характеру здійснюється на основі того, якого навчають, алгоритму. Яких-небудь налаштувань твердості й коректності перевірок поштової клієнт не має.

Thunderbird дозволяє також перевіряти повідомлення на наявність у них ознак шахрайства (fishing). Навколо включення даної функції у свій час було багато шуму, обіцялася небачена досі безпека. Але, чесно говорячи, користі від подібного інструмента обмаль. Поштової клієнт часто підозрює в шахрайстві навіть популярні українські розсилання, і відучити його від подібних дій не так-те просто. На відміну від засобів блокування реклами, тут не працює навченість.

The Bee

Працьовита бджілка The Bee є маленьким поштовою клієнтом, який можна запускати з будь-якого носія, навіть із дискети. Програма не вимагає установки в системі. Після її запуску в ОС не відбувається ніяких змін: The Bee не торкає реєстр і нічого не пише в системний розділ жорсткого диска. Програма дозволяє працювати з електронною поштою через протоколи POP, SMTP і IMAP. Ви можете також створювати облікові записи груп новин Usenet і використовувати «бджілку» як засіб спілкування через ICQ. Відзначимо, що The Bee має дуже простий інтерфейс, позбавлений навіть головного меню. Одна із кнопок на панелі інструментів може налаштуватися й виконувати довільну дію. Ви можете також запланувати виконання цієї дії через певний інтервал часу. При звертанні до сервера спочатку завантажуються заголовки листів. Подвійне клацання мишею по заголовку завантажує тіло повідомлення цілком. Програма дозволяє вручну міняти кодування CP1251 і KOI 8-R. Втім, поштової клієнт без проблем читає й повідомлення, що мають кодування UTF-8. Стосовно до такої маленької програми варто окремо відзначити можливість читання листів, закодованих в Base64, UUCode і Quoted-Printable. При звертанні до сервера ви не тільки можете завантажувати всі повідомлення відразу, але також обмежуватися п'ятьома

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

останніми листами. Розроблювачі попереджають, що ця функція працює не з усіма серверами. При одержанні повідомлень через ICQ відтворюється звуковий сигнал. Поточний статус підключення відображається усередині головного вікна додатка, що дозволяє закривати вікно чату, не втрачаючи з'єднання із сервером.

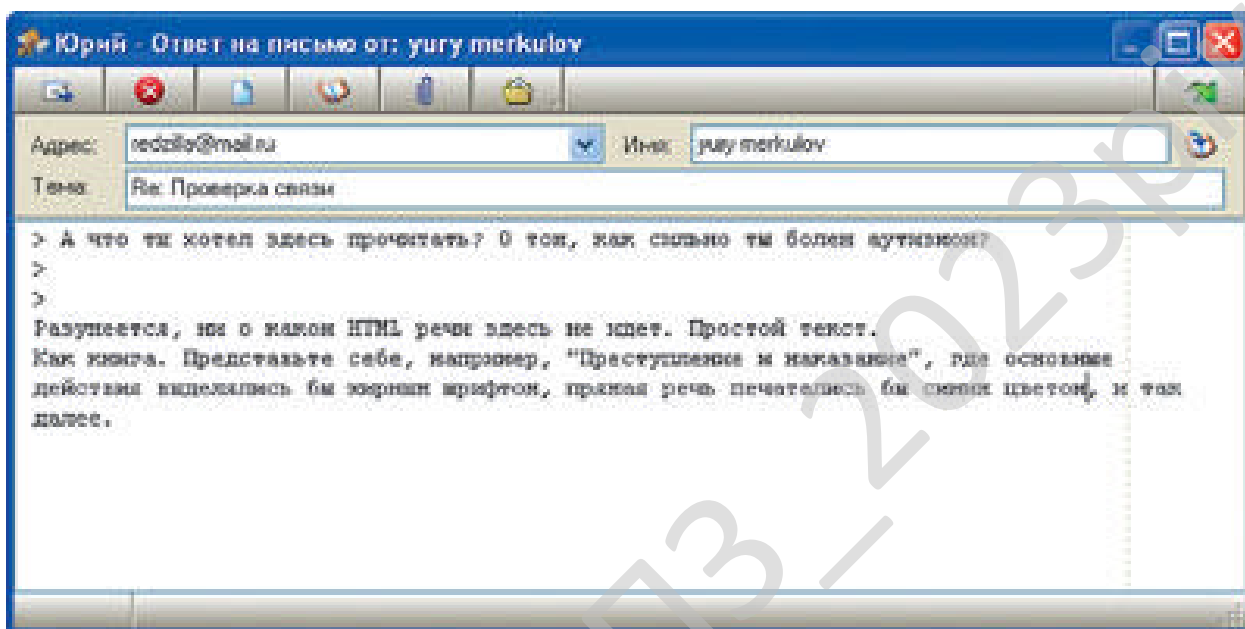


Рисунок 2.4 – Інтерфейс користувача The Bee

Редактор The Bee працює лише із простим текстом. Було б дивним очікувати від настільки маленької програми підтримки якого-небудь складного форматування. Головне призначення The Bee полягає не в тому, щоб замінити ваш улюблений поштовий клієнт на робочій станції. Основна користь «бджілки» – у її мобільності. Ви можете заходити в інтернет з будь-якого комп'ютера, фактично носячись всю пошту із собою.

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

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

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

– Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TMemo 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.

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

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

RAD Studio 10.4 Короткий огляд:

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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

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.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, 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.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку 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.

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

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 клієнта захищеної електронної пошти спеціального призначення.

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 кібербезпеки контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Базові функції поштового клієнта

Звичайно системи електронної пошти складаються із двох підсистем: користувальницьких агентів, що дозволяють користувачам читати й відправляти електронну пошту, і агентів передачі повідомлень, які пересилають повідомлення від відправника до одержувача. Користувальницькі агенти являють собою локальні програми, що надають різні методи взаємодії користувача з поштовою системою. Ці методи (або інтерфейси) можуть бути командними, графічними або заснованими на меню. Агенти передачі повідомлень звичайно є системними демонами, що працюють у фоновому режимі й переміщують електронну пошту по системі. При розгляді поштового клієнта можна виділити:

- базові функції (основні);
- додаткові функції (розширені);
- спеціальні функції.

Базові функції поштового клієнта призначені для виконання найпростіших операцій по відправленню й прийому повідомлень електронної пошти:

– Прийом повідомлень і автономний перегляд. Це основна функція поштового клієнта. При підключенні до сервера POP3 виробляється автоматичне копіювання всіх повідомлень, що надійшли, у базу даних поштового клієнта, після чого їх можна читати в автономному режимі (тобто, відключившись від мережі).

– Створення нових повідомлень. Це друга найважливіша функція поштового клієнта. Для її реалізації програма може мати вбудований текстовий редактор. Функціональність редактора залежить від властивостей конкретної програми. Для поштового клієнта, що працюють в операційних системах

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Windows, загальноприйнятою вважається підтримка операцій з використанням буфера обміну й можливість вибору кодування тексту.

– Упорядкування повідомлень. Упорядкування вхідних і вихідних повідомлень виробляється шляхом угруповання й сортування. Угруповання виконується розподілом повідомлень по логічних папках. Нагадаємо, що фізично всі повідомлення зберігаються в одному файлі загальної бази даних. Їхній розподіл по папках – чисто логічна операція, еквівалентна фільтрації записів бази. Кожна папка – це фільтр. Так, наприклад, у папці Вхідні відображаються тільки прийняті повідомлення (інші відфільтровуються й не візуалізуються), у папці «Відправлені» – відправлені повідомлення, а в папці «Вихідні» – повідомлення, підготовлені для відправлення, але поки ще не відправлені за якимись причинами. Поштовий клієнт надає спеціальну папку «Чернетки» для зберігання повідомлень, не готових для відправлення.

– Сортування після фільтрації використовуються для впорядкування повідомлень усередині папок. Звичайний порядок сортування – по даті, але його можна й змінити, якщо поштовий клієнт має відповідні функції.

– Автоматизація підготовки відповідних повідомлень. При підготовці відповідних повідомлень поштовий клієнт у стані використовувати дані, узяті з полів заголовка вихідного повідомлення. Це дозволяє спростити заповнення полів «Кому», «Тема» й т.п. Відпадає також необхідність у ручному введенні адреси одержувача. Зручною є функція цитування вихідного повідомлення.

– Операції із вкладеними файлами. По своїй природі електронна пошта призначена для пересилання чисто текстових повідомлень, переважно виконаних англійською мовою, тобто символами, що входять у стандарт ASCII. Відмінна риса цих символів полягає в тому, що їхні коди не перевищують значення 127. У той же час, у довільних файлах (програмних, архівних, мультимедійних, у файлах форматуваних документів і інших) можуть із рівною ймовірністю зустрічатися й байти зі значеннями від 128 до 255. Для пересилання їх разом з повідомленням електронної пошти потрібне спеціальне перетворення, у результаті якого символи

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

з кодуванням від 128 до 255 , замінюються комбінаціями символів з кодуванням від 0 до 127. На стороні адресата поштовий клієнт виконує зворотне перетворення. Такий механізм пересилання довільних файлів одержав назву поштових вкладень. Підготувавши повідомлення електронної пошти, відправник вказує місце розташування файлу, що повинен бути приєднаний. При цьому поштовий клієнт робить відповідний запис в одному з полів заголовка повідомлення, де вказує ім'я файлу й використаний метод кодування. Основними методами кодування є: MIME, BinHex і UUEncode. Поштовий клієнт здатний розуміти всі ці методи. При прийманні повідомлення поштовий клієнт дозволяє витягти поштове вкладення й зберегти його в заданому місці у вигляді автономного файлу.

Додаткові функції поштового клієнта

Додаткові функції клієнтів електронної пошти призначені для автоматизації основних операцій або для підвищення зручності роботи зі службою:

– Підтримка множинних ідентифікаційних записів. Ідентифікаційним записом називається сукупність налаштувань програми на конкретного користувача. Якщо із програмою працюють кілька людей, то деякі поштові клієнти дозволяють кожному з них створити власний ідентифікаційний запис і, тим самим, одержати власний комплект логічних папок. Передбачається, що при підключенні до поштового сервера відбувається завантаження тільки тих повідомлень електронної пошти, які адресовані власникові поточного ідентифікаційного запису. При зміні користувача відбувається зміна ідентифікаційного запису, і тоді можливе копіювання інших повідомлень, що надійшли. З міркувань безпеки операція зміни ідентифікаційного запису може супроводжуватися запитом пароля.

– Підтримка множинних облікових записів. Обліковим записом називається сукупність налаштувань на конкретного постачальника послуг електронної пошти. Якщо користувач має декілька «поштових скриньок»,

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

відкритих на різних серверах, то деякі поштові клієнти дозволяють вибрати поточний обліковий запис і легко перемикаються між ними.

– Підтримка облікових записів WWW. Якщо поштовий клієнт підтримує роботу з обліковими записами Web-Mail, то його можна використовувати одночасно й для роботи з «звичайною» електронною поштою, і для звертання до «поштових скриньок» служб електронної пошти, заснованої на Web. Ця функція характерна для поштового клієнта, що володіють функціями Web-браузерів або інтегруються з Web-браузерами.

– Підтримка формату HTML. Якщо поштовий клієнт підтримує формат HTML, то це дає можливість готувати, відправляти, одержувати й переглядати повідомлення, у яких є елементи форматування, використовується шрифтове й колірне оформлення, є вбудовані мультимедійні об'єкти.

– Підтримка адресної книги. Адресна книга – це зручний засіб для роботи з адресами електронної пошти. Це засіб керування базою даних, звичайно вбудований в поштову програму, що дозволяє вести облік контактів. Контактими називаються записи адресної книги, що відповідають регулярним кореспондентам і містять дані про людей і їхні адреси електронної пошти.

Стандартними функціями роботи з адресною книгою є:

- а) автоматизоване створення контакту на підставі даних, узятих з полів заголовка повідомлення, що надійшло;
- б) автоматизоване створення шаблону нового повідомлення після вибору потрібного контакту;
- в) упорядкування списку контактів шляхом угруповання й сортування.

– Розширені функції вбудованого текстового редактора. Найбільш бажаним розширенням властивостей текстового редактора є наявність засобів перевірки правопису, особливо багатомовних. Додатковими функціями можуть бути макрокоманди для уведення обігів і підписів, а також засоби генерації підписів, у тому числі й шляхом випадкового вибору із заздалегідь заготовленого зовнішнього текстового файлу. Цим прийомом користуються для «пожвавлення»

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Фільтрація повідомлень. Описаний вище аналіз полів заголовків може виконуватися автоматично за допомогою програмних фільтрів. Так, наприклад, поштові клієнти, що володіють засобами фільтрації, дозволяють виконувати автоматичне видалення небажаних повідомлень – у цьому випадку адресат їх навіть ніколи не побачить, а відправник не одержить повідомлення про те, що його повідомлення отримане й прочитане. Фільтрацію використовують для боротьби з негативним явищем, що називається спамом. Спам – це розсилання невикликаної кореспонденції. Існують різні підходи до кваліфікації спама. Зокрема, перший обіг, як правило, спамом не вважається – це нерідко використовують рекламні служби електронної торгівлі, але невмотивовані повторні звернення вже є порушенням правил користування електронної пошти й можуть бути оскаржені. Ще одна корисна функція фільтрації полягає в тому, що вступні повідомлення можна автоматично групувати в заданих тематичних або персональних папках. Так, наприклад, аналізуючи вміст поля Кому, програма може розподіляти вступник пошту в різні папки, що відповідають різним користувачам. Цю функцію використовують у малих підприємствах, якщо трохи співробітників використовують один загальний обліковий запис електронної пошти.

– Підтримка «чорного» і «білого» списків. Засоби фільтрації можуть працювати із заздалегідь заготовленими списками поштових адрес. «Чорним» називається список адрес електронної пошти, повідомлення від яких автоматично блокуються й знищуються безпосередньо на сервері без завантаження на локальний комп'ютер. «Білий список» використовують, щоб пропускати вибрані повідомлення в тих випадках, коли поштовий клієнт настроєний на блокування всіх вступні повідомлень. Слід зазначити, що «чорний» список може бути розподілений. Так, наприклад, сьогодні в Інтернеті є служби, що ведуть облік відомих спаммерів і рекламних служб, що порушують етикет електронної пошти. Поштовий клієнт може автоматично звіряти адреси відправників вхідних повідомлень із даними цих мережних служб і блокувати вступник кореспонденцію на підставі їхніх рекомендацій.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

– Функції спостереження й контролю за виконанням робіт. Для організацій, що одержують по електронній пошті великий потік звернень від громадян або інших організацій, важливим фактором є ефективний контроль за рухом і виконанням вхідних документів. У дуже великих організаціях, що обробляє тисячі звернень до року, для цієї мети використовують спеціальні системи керування діловодством. У малих і середніх організаціях у багатьох випадках вдається обмежитися поштовим клієнтом, що має розширені функції для контролю за рухом вхідних повідомлень по робочих місцях виконавців.

– Функції резервування й архівації. У тих випадках, коли характер електронної пошти, що проходить через організацію, відрізняється особливо важливою документальною цінністю, використовують засоби для роботи з електронною поштою, що володіють функціями автоматичного резервування й архівації. Це досягається вбудовуванням у поштовий клієнт програмного засобу резервування даних у стислому форматі, а також системою роботи з розкладу. Наявність цих функцій значно підвищує надійність і стійкість поштової програми. Так, наприклад, у випадку її виходу з ладу відкривається можливість повного відновлення всієї переписки, включаючи дані адресної книги.

– Функції автоматичної генерації відповіді й переадресації. Автоматична генерація відповіді на поштове повідомлення, що надійшло, дозволяє дотримати етикету електронної пошти й оперативно відповісти на повідомлення, що надійшло, навіть перебуваючи в тривалій відсутності. Звичайно на всі вступні повідомлення генерується стандартна відповідь із викладом причин тимчасової відсутності зв'язку. Якщо при цьому використовувати описані вище функції фільтрації, то можна класифікувати вхідні повідомлення по ім'ю відправника або темі повідомлення й генерувати різні автовідповіді на різні звернення.

– Ті, хто багато часу проводять у роз'їздах, поряд з функцією автовідповіді користуються також функцією автоматичної переадресації. Загальноприйнята практика складається в створенні тимчасового «поштової скриньки» на сервері однієї зі служб Web-Mail і переадресації на нього

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

кореспонденції, що надходить на облікові записи серверів POP3. Оскільки для роботи з електронною поштою, заснованої на Web, не потрібно ні наявності поштового клієнта, ні його налаштування, робота з тимчасовим «поштовою скринькою» виробляється за допомогою звичайного броузера з будь-якого місця зв'язку. Де б не перебував користувач, він завжди має можливість переглянути всю вступник електронну пошту на одному легкодоступному Web-вузлі.

– Експортно-імпортні функції. Формати, у яких різні поштові клієнти зберігають повідомлення електронної пошти, можуть розрізнятися. Розрізняються й структури логічних папок, і формати даних внутрішніх адресних книг. Найбільш потужні програми для роботи з електронною поштою дозволяють експортувати дані у формати інших програм або, навпаки, імпортувати їх.

Спеціальні функції засобів електронної пошти

Ці функції використовуються, насамперед, у бізнесі. Вони дозволяють автоматизувати операції, пов'язані з обслуговуванням більших інформаційних потоків:

– Функції масового розсилання. Звичайні засоби електронної пошти дозволяють відправити одне повідомлення в кілька адрес шляхом вказівки додаткових адресатів у полі «Копія». Цього недостатньо, якщо мова йде про відправлення тисяч поштових відправлень, до того ж не завжди бажано, щоб клієнти могли визначити адреси інших одержувачів по записам цього поля. У таких випадках поштова програма інтегрується із засобами обслуговування списку розсилання. Список розсилання – це текстовий файл, що представляє найпростішу базу даних, у якій окремі записи розділені стандартним роздільником, наприклад крапкою або символом табуляції. Розсилання повідомлення виробляється по всіх адресах, наведеним у списку. Додаткові засоби програми дозволяють також:

а) автоматично вносити адреси в список розсилання на основі аналізу вступні повідомлень (наприклад, на прохання клієнта про передплату на інформаційний бюлетень, електронний каталог і т.п.);

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- b) автоматично видаляти адреси зі списку розсилання (наприклад на прохання клієнта про припинення розсилання реклами на його адресу);
- c) групувати запису в списках розсилання; багато фірм відправляють новим клієнтам листа одного змісту (із вдячністю за увагу), а постійним клієнтам – іншого (з новими пропозиціями);
- d) розшукувати й усувати дублікати записів у списку розсилання;
- e) установлювати й видаляти записи, що втратили актуальність (наприклад, при зміні адреси електронної пошти клієнта).

Варто розрізнити масове розсилання й «поштове бомбардування». Поштове бомбардування – це цілеспрямована зловмисна акція по переповненню «поштової скриньки» жертви шляхом масового відправлення невикликаної кореспонденції. Це грубе порушення правил користування Інтернетом, адміністративні заходи для якого може прийняти сервіс-провайдер. У деяких випадках навмисне порушення функціонування систем зв'язку може кваліфікуватися і як карний злочин. До такої кваліфікації прибігають, якщо є можливість довести наявність прямого збитку. Масове розсилання відрізняється тим, що повідомлення відправляються в різні адреси. Воно не є правопорушенням, хоча й не завжди відповідає договору із сервіс-провайдером. Звичайна практика полягає в тому, що для проведення масових розсилок варто передбачити відповідний пункт у договорі й обмовити фінансову сторону заходу.

– Функції локального SMTP-сервера. Проведення масового відправлення кореспонденції через SMTP-сервер постачальника послуг сполучено з великими витратами часу, оскільки на сервері утвориться черга повідомлень, отриманих від різних клієнтів. У спеціальних випадках використовують поштові системи, що сполучають функції поштового клієнта й SMTP-сервера. Вони дозволяють відправляти кореспонденцію в обхід SMTP-сервера постачальника послуг і забезпечують високу продуктивність (до декількох тисяч відправлень у годину).

– Функції персоналізації. Персоналізація окремих повідомлень має важливе значення при проведенні масових розсилок. Її зміст полягає в тому, що кожне повідомлення має індивідуальні елементи, наприклад звернення (пан, пані, добродію й т.п.), ім'я адресата і якісь персональні відомості про нього. Персоналізація виконується шляхом взаємодії поштової програми з базою даних і зі списком розсилання. Повідомлення готується у вигляді шаблону, що має поля злиття. Дані в ці поля підставляються з бази даних відповідно до параметрів, зазначеними в списку розсилання. Завдяки технології персоналізації вдається різко підвищити відгук у рекламно-маркетингових кампаніях, при проведенні анкетування клієнтів, у встановленні довірчих відносин із замовниками.

– Функції віддаленого адміністрування. Деякі поштові клієнти здатні не тільки відправляти сигнали оповіщення на віддалені засоби зв'язку, але й приймати від них керуючі сигнали. Віддалене керування поштовими системами виробляється за допомогою телефонних апаратів, що мають тональний набір номера. Після встановлення з'єднання система сприймає сигнали цифрового набору як команди. Подібні системи дозволяють переадресовувати повідомлення, що надійшли, або автоматично відправляти один із заздалегідь заготовлених варіантів відповіді. Можливе також відправлення голосового повідомлення, продиктованого з віддаленого телефонного апарата.

Технологія функціонування електронної пошти

Розглянемо технологію функціонування електронної пошти.

Послідовність процедур для організації передачі інформації з електронної пошти:

- встановити зв'язок зі своїм хост-комп'ютером;
- запустити програму керування електронною поштою (наприклад, клацнути на робочому столі на ярлику поштового клієнта або скористатися командою в головному меню);
- у діалоговому вікні поштового клієнта натиснути кнопку «Створити повідомлення» або виконати команду меню «Повідомлення > Створити»;

– у діалоговому вікні, що з'явилося, у полі «Кому» вказати через крапку з комою доменні адреси (адреса) електронної пошти одержувачів (одержувача);

– у поле введення «Копія» перелічити через крапку з комою адреси електронної пошти всіх одержувачів, кому повинна бути доставлена копія повідомлення;

– у поле «Тема» описати короткий зміст повідомлення;

– у робочій області ввести текст повідомлення;

– вставити в повідомлення електронний підпис, виконавши команду «Вставка > Підпис». Електронний підпис містить звичайно ПІБ відправника, його посаду, назву фірми, телефон і т.д., але може містити в собі й оригінальний придуманий автором текст. Електронний підпис підготовляється заздалегідь і зберігається в папці. Створити її простіше всього по команді «Сервіс > Параметри», і на вкладці «Підпис» натиснути кнопку «Створити»;

– якщо потрібно, прикріпити файл вкладення за допомогою команди «Вставка > Вкладення» файлу або натисканням відповідної кнопки на панелі інструментів з наступним вибором вложеного файлу. Файл-вкладення може бути будь-яким створеним заздалегідь файлом (текстовим, графічним, що виконується й т.д.);

– «покласти» створений лист у папку «Вихідні», нажавши на панелі інструментів кнопку Відправити;

– відправити повідомлення по електронній пошті з папки «Вихідні» натисканням кнопки «Доставити повідомлення».

Через кілька секунд ваше повідомлення надійде хост-комп'ютеру одержувача по зазначеній адресі.

Адреси всіх одержувачів можуть бути введені вручну, або взяті з адресної книги. Щоб скористатися наявною адресною книгою, варто виконати команду «Сервіс > Вибрати одержувачів». У вікні, що з'явилося, зі списку вибрати потрібних одержувачів, супроводжуючи вибір кожного з них для вказівки категорії одержувачів наступним натисканням клавіш: «Кому», «Копія»,

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

«Схована копія». До категорії «Схована копія» варто віднести тих одержувачів, відсилення дубліката в чю адресу потрібно зберегти в таємниці.

В Інтернеті для доставки електронної пошти машина-джерело встановлює TCP-з'єднання з портом 25 машини-приймача. Цей порт прослуховується поштовим демоном, і їхнє спілкування відбувається за допомогою протоколу SMTP (Simple Mail Transfer Protocol – простий протокол електронної пошти). Цей демон приймає вхідні з'єднання й копіює повідомлення з них у відповідні поштові скриньки. Якщо лист неможливо доставити, відправникові вертається повідомлення про помилку, що містить першу частину цього листа.

Послідовність процедур для одержання повідомлення, що надійшло у вашу адресу:

- встановити зв'язок зі своїм хост-комп'ютером;
- запустити програму керування електронною поштою (наприклад, клацнути на робочому столі на ярлику поштового клієнта або скористатися командою в головному меню);
- у діалоговому вікні відкрити папку «Вхідні». У цій папці перебувають всі повідомлення, що надійшли на адресу користувача.

У відведеній для цього області вікна з'явиться список всіх повідомлень, що надійшли (ще не прочитані повідомлення позначені – звичайно виділені жирним шрифтом або супроводжуються словом new). У списку повідомлень будуть зазначений: порядковий номер повідомлення, відправник, дата й час одержання, а також рядок «зміст» (тема), заповнена відправником. Щоб прочитати потрібне повідомлення, варто двічі клацнути на ньому лівою кнопкою миші.

POP3 (Post Office Protocol v. 3 – поштовий протокол, 3-я версія) – спеціальний протокол, що дозволяє користувальницькому агенту (на машині клієнта) з'єднатися з агентом передачі повідомлень провайдеру (на машині провайдеру) і скопіювати пошту, що зберігається для нього.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Послідовність дій для відповіді на отримане повідомлення:

– У діалоговому вікні натиснути кнопку «Відповісти» або виконати команду «Повідомлення > Відповісти відправникові». Відкриється вікно створення повідомлення, але в рядку «Кому» вже буде перебувати адреса ініціатора повідомлення, на яке створюється відповідь, а в рядку «Тема» – тема вихідного повідомлення із префіксом Re (що означає Reply – відповідь). У рядку «Копія» можна додати адреси, по яких направляться копії відповіді. У робочому полі буде видний текст вихідного повідомлення (його можна підкоротити або, відповідно до мережної етики, зберегти тільки окремі цитати), до якого варто додати текст відповіді.

– Натисканням кнопки «Відправити відповідь» відсилається по зазначених адресах.

3.2 Розробка структурної схеми

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

Програмне забезпечення поштового клієнту складається з наступних структурних блоків:

– Двигун поштового клієнту системи електронного документообігу по протоколам POP3 та SMTP – основне ядро на якому будується поштовий клієнт.

– 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.

– Блок авторизації та автентифікації – призначений для здійснення входу до поштової скриньки згідно логіну та пароллю.

– 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.

– 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.

Структура поштового повідомлення

Повідомлення електронної пошти, що пересилається по мережі, складається із трьох частин:

– кодового конверта;

– заголовка (header), що містить службову інформацію, що управляє доставкою й обробкою повідомлення;

– тіла (body), що містить безпосередньо користувальницьке повідомлення: текст і прикріплені дані (графічні, звукові файли й т.п.).

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

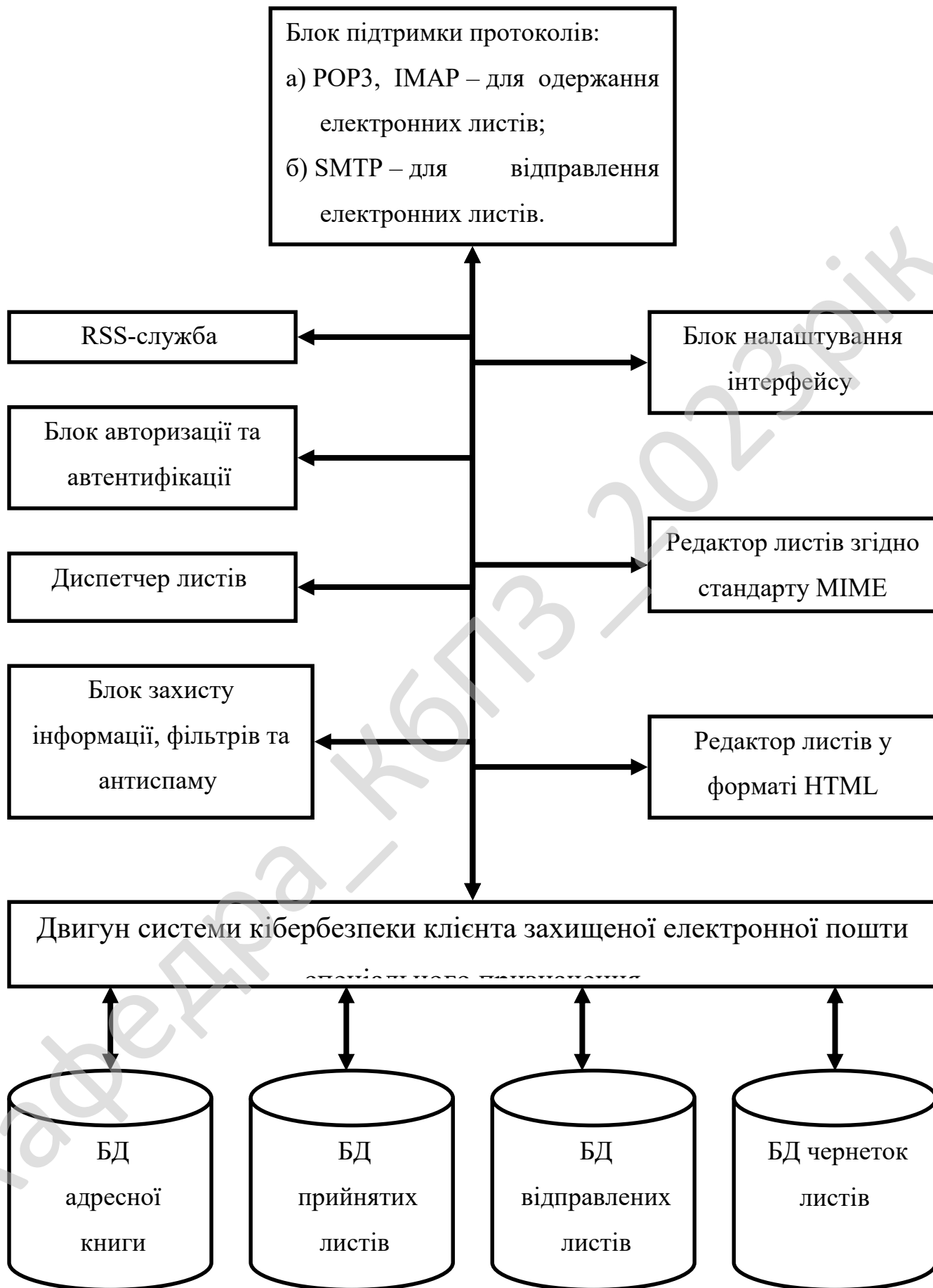


Рисунок 3.1 – Структурна схема розробленої системи

Існують два основних стандарти оформлення заголовка: стандарт RFC 822 і стандарт MIME. Стандарт RFC-822 був у основі електронної пошти й закріпив угоду про формат електронного листа і його керуючих заголовків, розраховуючи на простий текст. Але користувачі захотіли писати листа в національних кодуваннях, пересилати графіку, програми й ін. У результаті, після інших невдалих спроб, з'явився стандарт MIME, описаний в RFC 1341 і прийнявший остаточну форму в RFC 1521, що регламентує формат заголовка й тіла повідомлення, а також можливості використання в ньому інформаційних елементів різного типу.

Заголовок повідомлення

Поштове повідомлення – це простий текст у форматі ASCII. Тому заголовок повідомлення являє собою послідовність текстових рядків виду:

Стандартом поштових повідомлень передбачена велика кількість полів. Частина з них є обов'язковими, тобто без них неможливо здійснити правильну доставку повідомлень, а частина – необов'язковими. Нижче перераховані найбільше часто використовувані поля:

- Message-ID – унікальний ідентифікатор повідомлення. Унікальність значення цього поля гарантується програмним забезпеченням вузла-відправника, тому воно генерується автоматично.

- Date – поле "Дата". Містить дату відправлення повідомлення. Значення поля встановлюється автоматично поштовим клієнтом при відправленні повідомлення.

- From – поле "Від". Містить адресу, що відправник повідомлення вказує як вихідну адресу.

- Sender – поле "Відправник". Містить адресу, з якої було реально відправлене повідомлення. Це поле може бути відсутнім у заголовку, якщо поле "From" містить адресу реального відправника.

- To – поле "Кому". Містить адресу основного одержувача повідомлення.

– Сс – поле "Копія". Містить адреси додаткових одержувачів повідомлення.

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

У всіх полях, призначених для вказівки адрес одержувачів ("То", "Сс" і "Всс") 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;

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

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

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 сформулювати 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 текстового 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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

поля вказує на конкретний тип даних, або інформує про те, що тіло містить кілька різнотипних блоків.

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

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема розробленого поштового клієнту електронного документообігу по протоколам POP3 та SMTP.

Розглянемо більш докладно основні функціональні можливості розробленого поштового клієнту.

До основних функцій розробленого поштового клієнта відносяться наступні:

- Перевірка протоколів автентифікації.
- Шифрування даних PGP, S/MIME, SSL/TLS, AES.
- 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.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49



Рисунок 3.2 – Функціональна схема системи

Розглянемо більш докладно вищеперераховані функції.

Автоматична обробка повідомлень

Вищезгаданий Сортувальник листів дозволяє автоматизувати обробку повідомлень: налаштувати 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 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.

Безпека особистої інформації

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

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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Windows Vista повідомляємо, що, на додаток до вищевказаних можливостей, поштовий клієнт задіє такі компоненти Windows Vista, 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.

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 налаштування 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 Vista – ви зможете обновляти 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-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

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 загально визнаному 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 налаштування поштової скриньок, папок, адресних книг і прикріплених файлів.

Багаті можливості імпорту

Поштової клієнт дозволяє імпортувати повідомлення з інших поштової 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.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

- Текст, розділений комами.
- Текст, розділений табуляціями.
- INI-файл.
- Адресна книга поштового клієнта.
- Адресна книга Eudora.
- Файл-Шаблон Pegasus.

Працює з будь-якою системою Windows.

Сумісність із Microsoft Exchange

Поштовий клієнт підключається до серверів Microsoft Exchange, задіючи протокол MAPІ для виклику або відправлення повідомлень. Необхідно лише встановити Microsoft Office Outlook або клієнт Microsoft Exchange для надання компонентів підключення до Exchange, які буде використовувати поштовий клієнт.

Національні кодування

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

Диспетчер листів

Диспетчер дозволяє обробляти пошту на POP3 або ІМАР 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

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

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.

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

3.4 Розробка діаграми процесів

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

З нього видно, що процеси взаємодіють наступним чином.

Спершу запускається процес виведення головного вікна програми.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Він взаємодіє з наступними процесами:

- Процес створення листа.
- Процес автентифікації одержання листа адресатом.

Процес створення листа взаємодіє з процесом з'єднання поштового клієнта з SMTP-сервером.

Процес з'єднання поштового клієнта з SMTP-сервером взаємодіє з процесом передачі SMTP-серверу інформації про одержувача та відправника.

Процес передачі SMTP-серверу інформації про одержувача та відправника взаємодіє з процесом перевірки SMTP-сервером валідності даних та прийняття листа.

Процес перевірки SMTP-сервером валідності даних та прийняття листа взаємодіє з процесом постановки листа у чергу доставки.

Процес постановки листа у чергу доставки взаємодіє з процесом DNS-запиту про поштові сервери для домену адресату.

Процес DNS-запиту про поштові сервери для домену адресату взаємодіє з процесом спроби з'єднання з поштовими серверами, що мають найбільший пріоритет.

Процес спроби з'єднання з поштовими серверами, що мають найбільший пріоритет взаємодіє з наступними процесами:

- Процес спроби передати лист на резервні поштові сервери домену.
- Процес передачі листа у випадку вдалого з'єднання з поштовим сервером домену.

Процес передачі листа у випадку вдалого з'єднання з поштовим сервером домену взаємодіє з наступними процесами:

- Процес спроби переслати лист у випадку невдачі.
- Процес прийому поштовим сервером домену адресата листа.

Процес прийому поштовим сервером домену адресата листа взаємодіє з процесом з'єднання адресата з POP3 або IMAP сервером. Останній процес взаємодіє з процесом автентифікації одержання листа адресатом.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

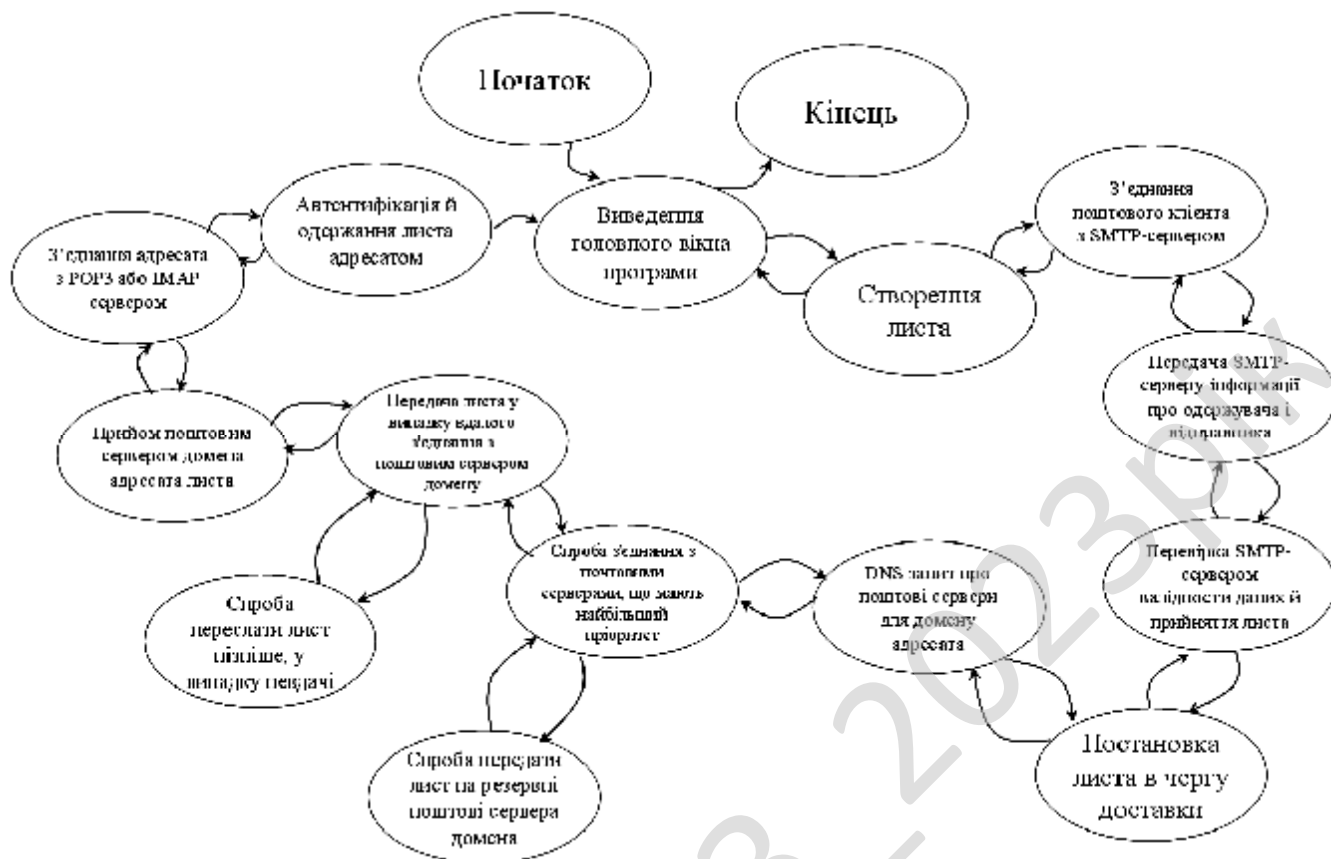


Рисунок 3.3 – Діаграма процесів системи

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.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків.

Спершу відбувається виведення основного вікна програми. Після цього користувач обирає яку дію йому зробити:

- Створити нову поштову скриньку.
- Перевірити пошту.
- Перевірити наявність вхідних листів.
- Завантажити вхідні листи.
- Відправити лист.
- Змінити параметри програми.

Якщо користувач обирає таку дію, як створити нову поштову скриньку, то виконуються наступні ітерації:

- Вводяться параметри поштової скриньки.
- Створюється поштова скринька.

Якщо користувач обирає таку дію, як перевірити пошту, то виконується перевірка поштової скриньки на наявність листів.

Якщо користувач обирає таку дію, як перевірити наявність вхідних листів, то виконуються наступні ітерації:

- Отримуються заголовки вхідних листів.
- Виводяться заголовки вхідних листів.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Якщо користувач обирає таку дію, як завантажити вхідні листи, то виконуються наступні ітерації:

- Завантажуються отримані листи на комп'ютер.
- Виводиться зміст листів на екран.

Якщо користувач обирає таку дію, як відправити лист, то виконуються наступні ітерації:

- Виводиться вікно створення листа.
- Вводяться адреси отримувача.
- Вводяться заголовки та теми листів.
- Вводиться текст листа.
- Відбувається відправка листа.

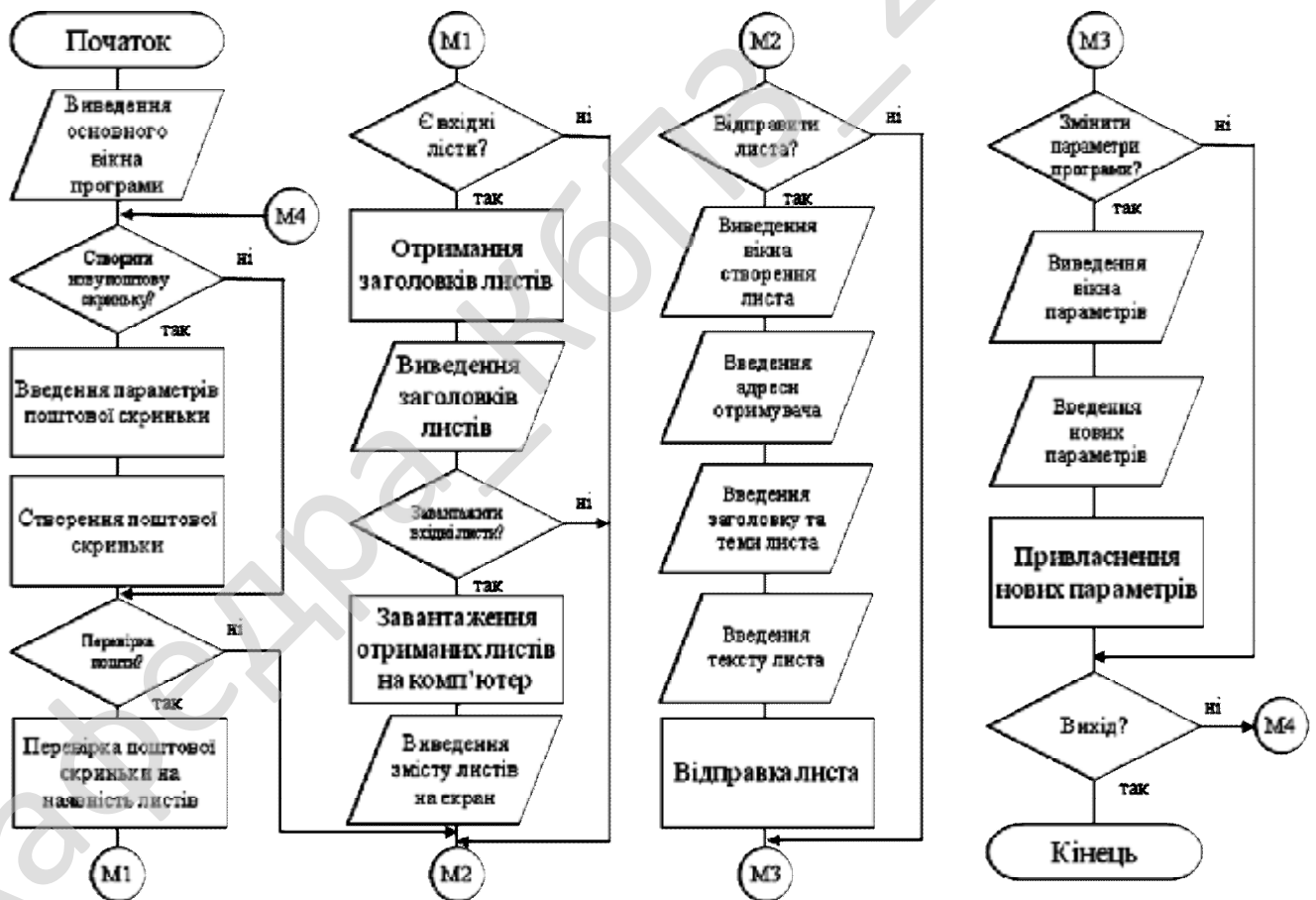


Рисунок 4.1 – Блок-схема роботи основної програми

Якщо користувач обирає таку дію, як змінити параметри програми, то виконуються наступні ітерації:

- Виводиться вікно параметрів.
- Вводяться нові параметри.
- Відбувається привласнення нових параметрів.

Після виконання усіх вищеперерахованих дій, користувач обирає, чи працювати йому далі з програмою, або ні.

На рисунку 4.2 зображена блок-схема роботи підпрограми відправки листів. Вона працює наступним чином.

Спершу відбувається з'єднання з SMTP-сервером.

Після цього відбувається передача SMTP-серверу інформації про одержувача та відправника.

Наступним етапом є перевірка SMTP-сервером валідності даних про одержувача та відправника. Якщо дані не є валідними, тоді виконуються наступні дії:

- Виводиться повідомлення про помилку.
- Відбувається введення інших даних.

Якщо ж дані є валідними, тоді виконуються наступні дії:

- Передається лист (із заголовками та вмістом листа).
- Відбувається постановка листа в чергу доставки.
- Відбувається DNS запит про поштові сервери для домену адресата.
- Відбувається спроба з'єднання з поштовими серверами, які мають найбільший пріоритет.

Якщо з'єднання не встановлено, тоді відбувається спроба передати лист на резервні поштові сервери домену.

Якщо з'єднання з резервними поштовими серверами домену не встановлено, тоді відбувається спроба переслати лист пізніше.

У іншому випадку виконуються наступні дії:

- Передача листа домену адресату.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

- Прийом поштовим сервером домену адресату листа.
 - Передача листа модулю, який займається зберіганням листів і видачею їх адресатам по протоколу IMAP, POP3 або іншим.
 - З'єднання адресата з IMAP або POP3 сервером.
 - Автентифікація адресата.
 - Одержання листа адресатом.
- На цьому підпрограма відправки листів закінчує свою роботу.

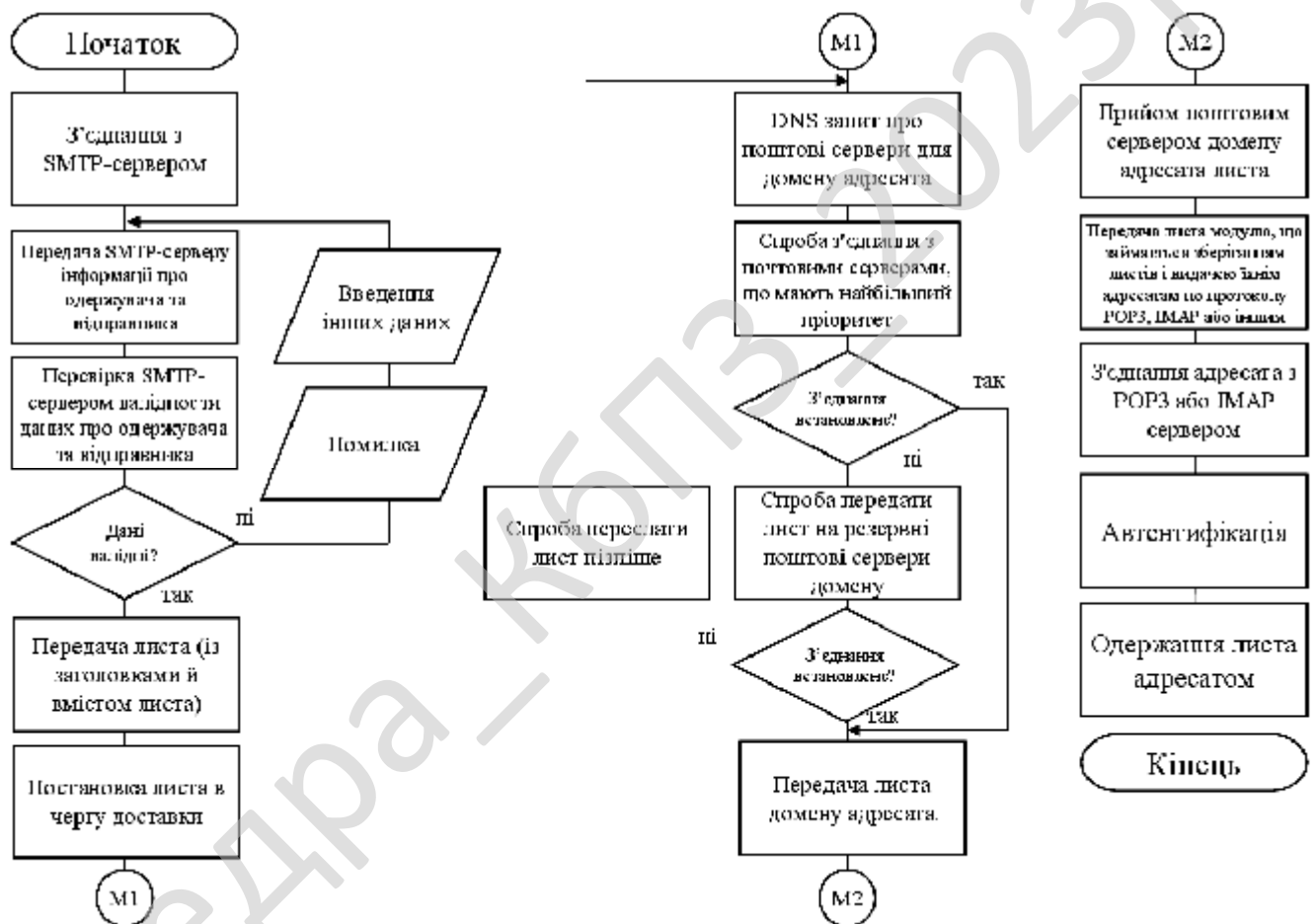


Рисунок 4.2 – Блок-схема роботи підпрограми відправки листів

Для роботи з мережею, в Delphi використовуються Indy компоненти, які можна знайти на декількох вкладках середовища розробки. Знайдіть на вкладці Indy Clients – компонент IdPOP3, на вкладці Indy Misc – компонент IdMessage. За допомогою компонента IdPOP3 ми зможемо здійснити підключення до

поштового сервера за протоколом POP3 і одержати необхідне нам кіл-у повідомлень. Компонент IdMessage буде використовуватися, як буфер для одержуваного листа. Ще нам буде потрібно мемо для відображення тексту листа, і кнопка для завантаження чергового повідомлення.

Одержання одного листа

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  POP3.Host:='mail.58r.ua'; // адреса поштового сервера
  POP3.Port:=110; // порт по якому буде здійснюватися підключення
  POP3.Username:='test+58r.ua'; // Логін користувача
  POP3.Password:='12345666?'; // пароль користувача
  IdMessage.Clear; // очищення буфера для повідомлення
  POP3.Connect; // підключення за протоколом POP3, по налаштуваннях у
компоненті IdPOP3, з ім'ям POP3
  Memo1.Clear; // очищення компонента мемо для відображення тексту листа.
  POP3.Retrieve(1,IdMessage); // одержання одного повідомлення
  Memo1.Lines.AddStrings(IdMessage.Body); // передача повідомлення з
компонента IdMessage в мемо
  POP3.Delete(1); // видалення із сервера, отриманого повідомлення
  POP3.Disconnect; // розрив зв'язку
end;
```

Звичайно потрібно завантажити кілька повідомлень, всі що є на сервері. Я вирішив обмежитися 10 листами, щоб не ускладнювати код. Одержувані повідомлення зберігаються в тій же папці що й поштова програма.

Одержання 10 листів

```
procedure TForm1.Button2Click(Sender: TObject);
label f;
var
  mailcicl:integer; // лічильник одержуваних листів
  addr:string; // папка в якій будуть зберігатися одержувані листи
begin
  POP3.Host:='mail.58r.ua'; // адреса поштового сервера
  POP3.Port:=110; // порт по якому буде здійснюватися підключення
  POP3.Username:='test+58r.ua'; // Логін користувача
  POP3.Password:='12345666?'; // пароль користувача
  addr:=ExtractFilePath(Application.ExeName); // одержання адреси папки, у
якій перебувати наша програма
  POP3.Connect; // підключення до сервера
```

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

```

for mailcicl:=1 to 10 do // цикл одержання листів
begin
if POP3.CheckMessages<1 then goto f ; // Якщо листів ні, те вийти й циклу
IdMessage.Clear; // очищення буфера для повідомлення
Memo1.Clear; // очищення компонента мемо для відображення тексту листа.
POP3.Retrieve(1,IdMessage); // одержання одного повідомлення
Memo1.Lines.AddStrings(IdMessage.Body); // передача повідомлення з
компонента IdMessage в мемо
Memo1.Lines.SaveToFile(addr+inttostr(mailcicl)+'.txt'); // збереження листа
в папці, де перебуває наша програма
POP3.Delete(mailcicl); // видалення листа на сервері
end;
f: // мітка виходу із циклу
POP3.Disconnect; // розрив з'єднання із сервером
end;

```

Кожне повідомлення має атрибути – це майл відправника, тема повідомлення, важливість і інші елементи. Вивчіте компонент IdMessage, щоб знати всі можливі атрибути.

Одержання листа з атрибутам

```

procedure TForm1.Button3Click(Sender: TObject);
var
i,numPosts: Integer;
begin
POP3.Host:='mail.58r.ua'; // адреса поштового сервера
POP3.Port:=110; // порт по якому буде здійснюватися підключення
POP3.Username:='test+58r.ua'; // логін користувача
POP3.Password:='12345666?'; // пароль користувача
IdMessage.Clear; // очищення буфера для повідомлення
POP3.Connect; // підключення за протоколом POP3, по налаштуваннях у
компоненті IdPOP3, з ім'ям POP3
Memo1.Clear; // очищення компонента мемо для відображення тексту листа.
POP3.Retrieve(1,IdMessage); // одержання одного повідомлення
Memo1.Lines.AddStrings(IdMessage.Body); // передача повідомлення з
компонента IdMessage в мемо
// вивід у компоненти Label інформації про повідомлення.
Label1.Caption := IdMessage.From.Text;
Label2.Caption := IdMessage.Recipients.EmailAddresses;
Label3.Caption := IdMessage.CCList.EMailAddresses;
Label4.Caption := IdMessage.Subject;
Label5.Caption := FormatDateTime('dd mmm yyyy hh:mm:ss', IdMessage.Date);

```

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

```

Label6.Caption := IdMessage.ReceiptRecipient.Text;
Label7.Caption := IdMessage.Organization;
POP3.Delete(1); // видалення із сервера, отриманого повідомлення
POP3.Disconnect; // розрив зв'язку
end;

```

Відправлення пошти здійснюється за протоколом SMTP, компонент IdSMTP, ви зможете знайти його на вкладці Indy Clients. У даному прикладі відправлення пошти буде здійснюватися після авторизації сервері, лист відправляється з майла test@58r.ua, на майл test@58r.ua .

Відправлення одного листа

```

procedure TForm1.Button4Click(Sender: TObject);
begin
SMTP.Host:='mail.58r.ua'; // адреса поштового сервера
SMTP.Port:=25; // порт по якому буде здійснюватися підключення
SMTP.Username:='test+58r.ua'; // логін користувача
SMTP.Password:='12345666?'; // пароль користувача
SMTP.AuthenticationType:=atLogin; // тип підключення до сервера - з
авторизацією
with IdMessage do
begin
Body.Assign(Memo1.Lines); // передача даних з мемо у компонент IdMessage,
для наступного відправлення
From.Text := 'test@58r.ua'; // майл відправника листа
Recipients.EmailAddresses := 'test@58r.ua'; // майл одержувача листа
Subject := 'Programmersclub.ua'; // теми листа
end;
SMTP.Connect; // з'єднання з поштовим сервером
try
showmessage('Відбулося підключення до сервера'); // повідомлення про вдале
підключення до сервера
SMTP.Send(IdMessage); // відправлення листа на сервер
finally
SMTP.Disconnect; // розрив з'єднання
end; end;

```

Відправлення листа й вказівка теми, тексту повідомлення

Для відправлення листа за допомогою зареєстрованого клієнта використовується функція Windows – ShellExecute, де як аргумент передається рядок протоколу Mailto. Для цього зробіть наступне:

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64


```

public
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
    function Send(str: string):boolean; //...відправлення повідомлення
protected
    procedure Loaded; override;
    procedure ErrorCatch(Sender : TObject; Exc : Exception);
published
    property ServerName: string read FServerName write FServerName;
    property MailSlotName: string read FMailSlotName write FMailSlotName;
end;
procedure register;
implementation
procedure register;
begin
    RegisterComponents('Gl          Components', [TglMailSlotServer,
TglMailSlotClient]);
end;
constructor TglMailSlotServer.Create(AOwner: TComponent);
begin
    inherited;
    FEnabled := true;
    FMailSlotName := 'MailSlot';
    Timer := TTimer.Create(nil);
    Timer.Enabled := false;
    Timer.OnTimer := OnTimer;
end;
destructor TglMailSlotServer.Destroy;
begin
    Timer.Free;
    // закриття каналу
    Close;
    inherited;
end;
procedure TglMailSlotServer.Loaded;
begin
    inherited;
    Open;
end;
procedure TglMailSlotServer.Open;
begin
    // створення каналу з ім'ям MailSlotName - по цьому імені до нього

```

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71


```

    CloseHandle (h) ;
end;
Result := h <> INVALID_HANDLE_VALUE;
end;
end.

```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Serpent – симетричний блочний алгоритм шифрування, розроблений Россом Андерсоном, Елі Біхамом та Ларсом Кнудсенем. Алгоритм був одним з фіналістів 2-го етапу конкурсу AES. Як і інші алгоритми, які брали участь у конкурсі AES, Serpent має розмір блоку 128 біт і можливі довжини ключа 128, 192 або 256 біт. Алгоритм являє собою 32-раундовий шифр на основі SP-мережі, і працює з блоком з чотирьох 32-бітових слів. Serpent був розроблений так, що всі операції можуть бути виконані паралельно, використовуючи 32-а 1-бітних «потоків».

При розробці Serpent використовувався консервативніший підхід до безпеки, ніж у інших фіналістів AES, проектувальники шифру вважали, що 16 раундів достатньо, щоб протистояти відомим видам криптоаналізу, але збільшили число раундів до 32, щоб алгоритм міг краще протистояти ще не відомим методам криптоаналізу.

Шифр Serpent не запатентований і є громадським надбанням.

Алгоритм створювався під гаслом «криптографічний алгоритм 21 століття» для участі в конкурсі AES. При створенні нового алгоритму Serpent його автори дотримувалися консервативних поглядів на проектування, що підтверджується первісним рішенням про використання таблиць підстановки з відомого багато років раніше алгоритму шифрування DES, який протягом довгого часу вивчався провідними фахівцями в області криптографії та захисту інформації і чиї властивості і особливості були добре відомі науковому світу. Одночасно з цим до нового алгоритму міг бути застосований вичерпний аналіз,

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Початкова перестановка IP

Дана перестановка IP задається таблицею, де вказується позиція, на яку перейде відповідний біт (наприклад, біт 1 перейде на 32 позицію):

S-бокси (таблиці замін)

В алгоритмі Serpent таблиці замін є 4-бітовими перестановками з властивостями стійкості до диференціального криптоаналізу, до лінійного криптоаналізу і такою властивістю, що порядок вихідних біт, як функції вхідних повинен бути максимальний, тобто бути рівним 3.

Таблиця підстановки генерується з відомих і добре вивчених таблиць для алгоритму DES в ітераційному процесі, поки не будуть отримані бажані диференціальні й лінійні властивості. Таким чином, створюється 8 таблиць підстановки.

Лінійне перетворення LT

Лінійне перетворення LT задається таблицею, де біти перераховані від 0 до 127 (наприклад, вихідний 2 біт утворений 2, 9, 15, 30, 76, 84, 126 бітами, складеними за модулем 2) . В кожному рядку описується 4 вихідних біти, які разом складають вхідні дані на одну таблицю замін в наступному раунді. Варто зазначити, що даний набір являє собою таблицю $IP(LT(FP(x)))$, де LT і є те лінійне перетворення.

Таблиця зворотного лінійного перетворення, яке використовується при розшифровці PLT.

Кінцева перестановка FP

Дана перестановка є зворотною до початкової, тобто $FP=IP^{-1}$ і задається наступною таблицею.

Ефективна реалізація алгоритму

Бажання авторів зробити алгоритм саме таким, яким він є стає зрозумілим при розгляді його ефективної низькорівневої реалізації.

Serpent був створений таким чином, щоб всі операції в процесі шифрування і розшифрування одного блоку могли бути виконані паралельно в 32

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

потоках. До того ж низькорівневий опис алгоритму набагато простіший, ніж стандартний опис. Ніяких початкових і кінцевих перестановок не потрібно.

Шифрування складається з 32 раундів. Відкритий текст є першими проміжними даними $V_0 = P$. Потім виконується 32 раунди, кожен i -й раунд складається з:

- Змішування з ключем. Проводиться побітове виключаюче «або» проміжних даних V_i з ключем довжиною 128 біт.

- Застосування таблиць підстановки. Вхідні дані довжиною 128 біт поділяються на 4 слова по 32 біта. Таблиця підстановки, реалізована послідовністю логічних операцій (як якщо це було б реалізовано апаратно), застосовується до цих 4 слів. В результаті виходить 4 вихідних слова. Таким чином, центральний процесор виконує підстановку по 32 копії таблиці одночасно.

- Лінійне перетворення. 32-бітові слова перетворюються заданим порядком.

Першою причиною вибору такого лінійного перетворення є максимізація лавинного ефекту. Такі таблиці підстановки мають властивість, що зміна кожного вхідного біта призведе до зміни 2 вихідних бітів. Таким чином, кожен вхідний біт відкритого тексту вже через 3 раунди впливає на всі вихідні біти. Аналогічно кожен біт ключа впливає на результат шифрування.

Друга причина полягає в простоті перетворення. Воно може бути реалізоване на будь-якому сучасному процесорі з мінімальними витратами.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Блок меню складається з наступних елементів:

- Лист.
- Скриньки.
- Параметри.
- Довідка.

Блок клавіш швидкого доступу до функцій програми складається з наступних елементів:

- Прийняти лист.
- Відправити лист.
- Написати лист.
- Додати користувача.
- Зробити розсилки.
- Параметри.

Блок вибору дій користувача складається з наступних елементів:

- Вхідні листи.
- Вихідні листи.
- Чернетки.
- Спам.
- Видалені.

Вікно відображення даних про лист складається з наступних елементів:

- Від кого.
- Тест.
- Дата.
- Розмір.

На рисунку 5.2 зображено вікно створення листа. З нього видно, що для створення листа необхідно написати лист, заповнити адресу одержувача, тему листа.

На рисунку 5.3 зображено вікно довідки, у якому наведено тему, автора, керівника та місце виконання бакалаврського проектування.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

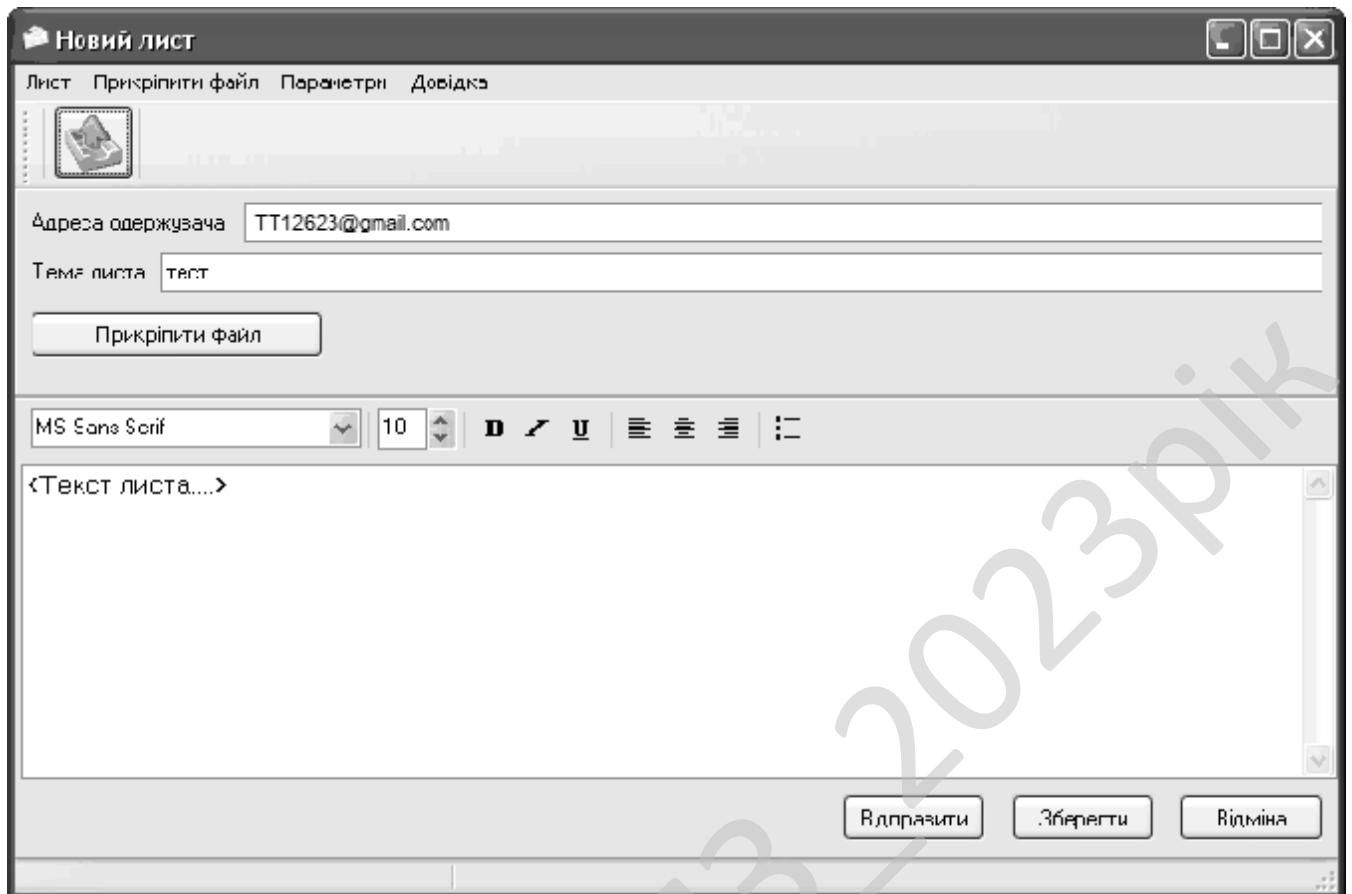


Рисунок 5.2 – Створення листа

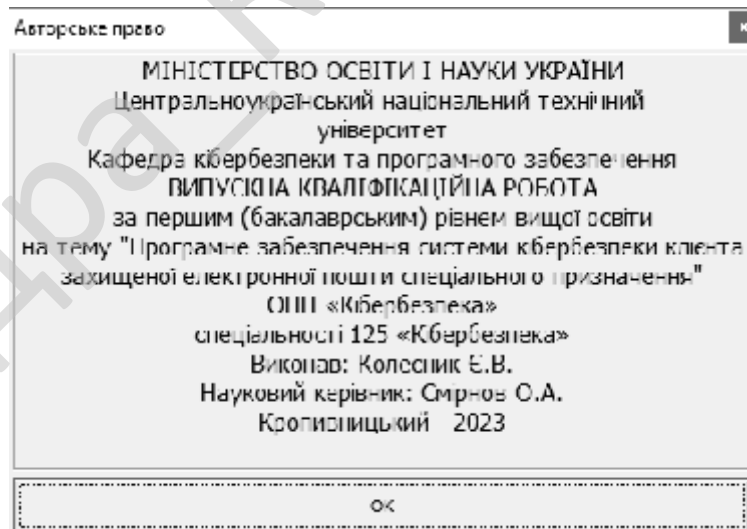


Рисунок 5.3 – Довідка

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем клієнта захищеної електронної пошти спеціального призначення.

– Досліджена система клієнта захищеної електронної пошти спеціального призначення.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання клієнта захищеної електронної пошти спеціального призначення.

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. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки клієнта захищеної електронної пошти спеціального

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

призначення. 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 кібербезпеки й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Serpent.

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-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.

2. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

3. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

4. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

5. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

6. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2014. – № 4(17). – С. 90-95.

7. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.

8. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

9. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.

10. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. – № 3(20). – С. 134-141.

11. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. - практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

12. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

13. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.

14. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.

15. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.

16. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.

17. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных вирусов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.

18. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.

19. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани,

С. А. Смирнов // Проблемы і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.

20. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

21. Смирнов С. А. технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.

22. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.

23. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.

24. Смирнов С. А. Метод управления доступом к облачным телекоммуни-кационным ресурсам для обеспечения защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.

25. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.

26. Смирнов С. А. Разработка комплекса gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та взаємодії» (IT & I): зб. тез II міжнар. наук.-практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.

27. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II междунар. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

28. Смирнов С. А. gert-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: зб. тез Круглого столу, м. Київ, 16 грудня 2015 р. – К.: Європейський університет, 2015. – С.41-43.

29. Смирнов С. А. Алгоритмы формирования множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць II Міжнар. наук.-практ. конф., м. Київ, 24-27 лютого 2016 р. – К.: Європейський університет, 2016. – С. 140-142.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

30. Смирнов С. А. Разработка и реализация метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Securitea informationala 2015-2016: Conferenta internationala (editia a XII-a), Chisinau, Moldova, 3 martie 2016. – Chisinau: ADSEM, 2016. – С. 90-96.

31. Смирнов С. А. Алгоритм формирования базового множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информатика та системні науки (ICN-2016): зб. тез VII всеукр. наук.-практ. конф., м. Полтава, 10-12 березня 2016 р. – Полтава: ПУЕТ, 2016. – С. 261-263.

32. Смирнов С. А. Система обработки и формирования начального состояния маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. тез наук.-практ. конф., м. Київ, 10-11 березня 2016 р. – К.: КНУ ім. Тараса Шевченка, 2016. – С. 81-82.

33. Смирнов С. А. Алгоритм безопасной маршрутизации на базовом множестве путей передачи метаданных в программный сервер облачной антивирусной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна безпека та комп'ютерні технології (IS&CT): зб. тез міжнар. наук.-практ. конф., м. Кіровоград, 24-25 березня 2016 р. – Кіровоград: КНТУ, 2016. – С. 73.

34. Смирнов С. А. Исследование способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016), м. Харків, 30 березня - 1 квітня 2016 р. – Х.: НТУ «ХП», 2016. – С. 14.

35. Смирнов С. А. Разработка способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов,

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

С. А. Смирнов, А. К. Дидык // Матеріали XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» (м. Кіровоград, 15-16 квітня 2016 р.). –Кіровоград: КНТУ, 2016. – С. 182-186.

36. Смирнов С. А. Разработка и исследование способа контроля линий связи телекоммуникационных сетей для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми і перспективи розвитку ІТ-індустрії: VIII міжнар. наук.-практ. конф., м. Харків, 28-29 квітня 2016 р.: зб. тез. – Х.: ХНЕУ, 2016. – С. 48.

37. Смирнов С. А. Модель системы нейросетевых экспертов безопасной маршрутизации для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна та економічна безпека (INFECO-2016): зб. тез III міжнар. наук.-практ. конф., м. Харків, 28-30 кві. 2016 р. – Х.: ХННІ ДВНЗ «УБС», 2016. – С. 178-182.

38. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Сборник тезисов XII международной конференции «Стратегия качества в промышленности и образовании» (г. Варна, Болгария, 30 мая - 02 июня 2016 г.). – Варна: ТУВ, 2016. – С. 581-585.

39. Смирнов С. А. Оценка эффективности метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. С. Коваленко // РадіоЕлектроніка та ІнфоКомунікації: зб. тез першої наук. - техн. конф., м. Київ, 11-16 вересня 2016 р. – К.: НТУУ «КПІ», 2016. – С. 17.

40. Современные телекоммуникации. Технологии и экономика / [В.Л. Банкет, О.В. Бондаренко, П.П. Воробьенко и др.]; под ред. С.А. Довгого. – М.: Эко-Трендз, 2003. – 320 с.

41. Столлингс В. Современные компьютерные сети / Вильям Столлингс. –СПб.: Питер, 2003. – 778 с.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

42. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.
43. Телекоммуникационные системы и сети: учебное пособие. В 3 томах / [В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев]; под ред. В.П. Шувалова. – М.: Горячая линия-Телеком, 2005, т. 3 – 592 с.
44. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1103 с.
45. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.
46. Elwalid, D. Mitra, I. Saniee, and I. Widjaja. Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs // Journal of lightwave technology. – 2003. – №21(11), P. 2828-28-38.
47. A.V. Bagula, M. Botha, and A.E Krzesinski. Online Traffic Engineering: The Least Interference Optimization Algorithm // IEEE Communications Society – 2004, P. 1232-1236.
48. Anees Shaikh, Jennifer Rexford, and Kang G. Shin. Evaluating the Impact of Stale Link State on Quality-of-Service Routing // IEEE/ACM Transactions on Networking. – 2001. – №9(2), P. 162-176.
49. Basabi Chakraborty. Simultaneous Search for Multiple Routes using Genetic Algorithm / IEEE International Conference on Computational Intelligence for Measurement System and Applications Boston. MA, USA, 14-16, July 2004, P. 77-80/
50. Barakat, E. Altman, and W. Dabbous. On TCP performance in a heterogeneous network: a survey // IEEE Communications Magazine. – 2000. – №38(1). – P. 40 - 46.
51. Carpenter G., A., Grossberg S. Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MA, MIT Press, 1991.

					ВКРБ-125.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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-125.23.0010.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Колесник С.В.				Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.						
Н. Контр.	Гермак В.С.				ЦНТУ КБ-19		
Затв.	Смірнов О.А.						
<i>Програмне забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення</i>							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 12-02 від 5.01.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення.

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-125.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- системи кібербезпеки клієнта захищеної електронної пошти спеціального призначення;
- 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-125.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.

					ВКРБ-125.23.0010.00.00.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 – 91 аркуш.

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-125.23.0010.00.00.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 23.05.2023 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 8.06.2023 р.

					ВКРБ-125.23.0010.00.00.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 клієнта захищеної
електронної пошти спеціального призначення*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2023 року

```
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_clientSettings, 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_clientBoxes: TTreeView;
    ListView_LettersHeaders: TListView;
    ToolBar1: TToolBar;
    btn_CheckMail_client: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    XPManifest1: TXPManifest;
    Panel_Content: TPanel;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    ActionList1: TActionList;
    aCheckMail_client: TAction;
    aSendMail_client: TAction;
    aMail_clientSettings: 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;
```

```

aAnswer: TAction;
aResend: TAction;
PopupMenu_Mail_clientBoxes: TPopupMenu;
N13: TMenuItem;
Jnghfdbnmgjxnel: TMenuItem;
N14: TMenuItem;
About1: TMenuItem;
procedure aMail_clientSettingsExecute(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_clientBoxesClick(Sender: TObject);
procedure aCheckMail_clientExecute(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_clientBoxesDragDrop(Sender, Source: TObject; X, Y:
Integer);
    procedure Tree_Mail_clientBoxesDragOver(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_clientBoxesPopup(Sender: TObject);
    procedure JnghfdbnmgjxnelClick(Sender: TObject);
    procedure About1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    procedure ShowMail_clientBoxesInTree;
    procedure LoadLetters;
    procedure AddToLog(s : shortstring);
    procedure HideLetterBody;
    procedure ShowInHtmlViewer(body : TStrings);
end;

const
    LogFileName    = 'Actions.log';

var
    frm_Main: Tfrm_Main;

    Mail_clientInfoLoaded : boolean=false;

    { Головне сховище електронних листів }
    Mail_clientBoxes      : TMail_clientBoxList;

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

```

```

implementation

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

{$R *.dfm}

//параметри облікових записів
procedure Tfrm_Main.aMail_clientSettingsExecute(Sender: TObject);
begin
    frm_Mail_clientSettings.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_clientInfoLoaded then
        begin
            { Show Mail_clientboxes }

            ShowMail_clientBoxesInTree;
            LoadLetters;
            Mail_clientInfoLoaded:=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_clientBoxesInTree;
    var
        i      : word;
        node   : TTreeNode;
    begin
        Tree_Mail_clientBoxes.Items.Clear;
        for i:=1 to TotalMail_clientBoxes do

```

```

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

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

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

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

    if Tree_Mail_clientBoxes.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_clientBoxes.Selected.Parent = nil then
      Mail_clientBoxName:=Tree_Mail_clientBoxes.Selected.Text
    else
      Mail_clientBoxName:=Tree_Mail_clientBoxes.Selected.Parent.Text;
    frm_NewLetter.Caption:='Новий лист <'+Mail_clientBoxName+'>';
  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_clientBoxesClick (Sender: TObject);
var
    Mail_clientBoxName      : shortstring;
    Mail_clientBoxFolder    : shortstring;
    LetterList              : PLetters;
    i                       : word;
begin
    if Tree_Mail_clientBoxes.Selected = nil then
    begin
        HideLetterBody;
        exit;
    end;
    List_Attachmant.Clear;
    List_Attachmant.Visible:=false;
    Memo_LetterBody.Clear;
    if Tree_Mail_clientBoxes.Selected.Parent = nil then
    begin
        if Mail_clientBoxes.FindBox (Tree_Mail_clientBoxes.Selected.Text).IsBuisyNow
        then aCheckMail_client.Enabled:=false
        else
            aCheckMail_client.Enabled:=true;
            ListView_LettersHeaders.Clear;
            HideLetterBody;
            findMail_clientBoxAndSelectIt (Tree_Mail_clientBoxes.Selected.Text);
            exit;
        end;

        Mail_clientBoxName := Tree_Mail_clientBoxes.Selected.Parent.Text;
        Mail_clientBoxFolder := Tree_Mail_clientBoxes.Selected.Text;

        LetterList:=Mail_clientBoxes.FindLetters (Mail_clientBoxName,Mail_clientBoxFolder
        );

        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_clientBoxes.FindBox (Mail_clientBoxName).IsBuisyNow then
            aCheckMail_client.Enabled:=false
        else
            aCheckMail_client.Enabled:=true;
            findMail_clientBoxAndSelectIt (Mail_clientBoxName);

            for i:=1 to LetterList^.Count do
                with ListView_LettersHeaders.Items.Add do
                    begin
                        Caption:=LetterList^.Letters[i]^Subject;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

    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));
    //Розмір
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_clientBoxSettings(mbSettings :
PMail_clientBoxSettings);
Begin
    DM.POP.Host :=mbSettings.POPServer;
    DM.POP.Port :=mbSettings.POPPort;
    DM.POP.Username :=mbSettings.POPAccount;
    DM.POP.Password :=mbSettings.POPPass;
End;

(*
procedure RetrieveMail_client(let_count : word);
var
    mes : TIdMessage;
    i : word;
    LettersNot2beDownloaded : word;
Begin
    try
    LettersNot2beDownloaded:=0;
    for i:=1 to let_count do
    if CurrentMail_clientBoxSettings^.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_clientBoxes.GetByName(CurrentMail_clientBoxSettings^.Name).HasLetterWithID(
mes.MsgId) then
    begin
    inc(LettersNot2beDownloaded);
    continue;
    end;
    if
Mail_clientBoxes.GetByName(CurrentMail_clientBoxSettings^.Name).hadLetterWithId(
mes.msgId) then
    begin
    case CurrentMail_clientBoxSettings.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_clientBoxes.GetByName(CurrentMail_clientBoxSettings^.Name).Inbox.Add(@mes,f
alse, DM.POP.RetrieveMsgSize(i));
            if CurrentMail_clientBoxSettings^.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_clientExecute(Sender: TObject);
var
    CurMail_clientBox : PMail_clientBox;
begin
    // Перевірити пошту
    Tree_Mail_clientBoxesClick(self);
    if CurrentMail_clientBoxSettings =nil then exit;

    aCheckMail_client.Enabled:=false;

    CurMail_clientBox:=Mail_clientBoxes.GetByName(CurrentMail_clientBoxSettings^.Nam
e);
    if CurMail_clientBox.IsBuisyNow then exit;
    try
        // Імітація переривання таймера для ініціювання процесу перевірки пошти
        CurMail_clientBox.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_clientBoxes.SaveMail_clientBoxes;
end;

```

```

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

procedure Tfrm_Main.List_AttachmantClick(Sender: TObject);
var
  Mail_clientBoxName      : shortstring;
  Mail_clientBoxFolder    : shortstring;
  LetterList              : PLetters;
begin
  if Tree_Mail_clientBoxes.Selected = nil then exit;
  if Tree_Mail_clientBoxes.Selected.Parent = nil then
  begin
    findMail_clientBoxAndSelectIt(Tree_Mail_clientBoxes.Selected.Text);
    exit;
  end;
  if (ListView_LettersHeaders.Selected=nil)or
    (List_Attachmant.Selected=nil)  then exit;

  Mail_clientBoxName      := Tree_Mail_clientBoxes.Selected.Parent.Text;
  Mail_clientBoxFolder    := Tree_Mail_clientBoxes.Selected.Text;

  LetterList:=Mail_clientBoxes.FindLetters(Mail_clientBoxName,Mail_clientBoxFolder
);

  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_clientBoxSettings^.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_clientBoxName      : shortstring;
    Mail_clientBoxFolder    : shortstring;
    LetterList               : PLetters;
    intIndex                 : integer;
    BodyPart                 : word;
begin
    if Tree_Mail_clientBoxes.Selected = nil then exit;
    if Tree_Mail_clientBoxes.Selected.Parent = nil then
        begin
            findMail_clientBoxAndSelectIt(Tree_Mail_clientBoxes.Selected.Text);
            exit;
        end;
    if ListView_LettersHeaders.Selected=nil then exit;

    Mail_clientBoxName      := Tree_Mail_clientBoxes.Selected.Parent.Text;
    Mail_clientBoxFolder    := Tree_Mail_clientBoxes.Selected.Text;

    LetterList:=Mail_clientBoxes.FindLetters(Mail_clientBoxName, Mail_clientBoxFolder
);

    //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[intIndex] 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
rings[0])=AnsiUppercase('Це повідомлення складається з кількох частин у форматі
MIME') )

```

```

                                and

(AnsiUppercase (LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^).Message
Parts.Items[intIndex].ContentType) = 'TEXT/HTML')
                                then begin
                                    Selected:=true;
                                    List_AttachmantClick(self);
                                end;
                                end;

                                end
                                end;
                                end;
                                if List_Attachmant.Items.Count=0 then List_Attachmant.Visible:=false;
end;

procedure Tfrm_Main.List_AttachmantMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var
    it : TListItem;
begin
if Button = mbRight then
    begin
        it:=List_Attachmant.GetItemAt(x,y);
        if it=nil then exit;
        it.Selected:=true;
        end;
    end;

const
    invalid :set of char=['\','/',':','?','*','"','<','>','|'];

function isValidFileName(s:string):boolean;
var
    i : word;
Begin
    isValidFileName:=false;
    for i:=1 to length(s) do
        if s[i] in invalid then exit;
    isValidFileName:=true;
End;

procedure Tfrm_Main.mnuSaveAttClick(Sender: TObject);
var
    Mail_clientBoxName      : shortstring;
    Mail_clientBoxFolder    : shortstring;
    LetterList               : PLetters;
Begin
    if Tree_Mail_clientBoxes.Selected = nil then exit;
    if Tree_Mail_clientBoxes.Selected.Parent = nil then
        begin
            findMail_clientBoxAndSelectIt(Tree_Mail_clientBoxes.Selected.Text);
            exit;
        end;
    if (ListView_LettersHeaders.Selected=nil)or
        (List_Attachmant.Selected=nil) then exit;

    Mail_clientBoxName      := Tree_Mail_clientBoxes.Selected.Parent.Text;
    Mail_clientBoxFolder    := Tree_Mail_clientBoxes.Selected.Text;

LetterList:=Mail_clientBoxes.FindLetters(Mail_clientBoxName,Mail_clientBoxFolder
);

```

```

    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_clientBoxName,Mail_clientBoxFolder : shortstring;
    LetterList : PLetters;
begin
    if ( Tree_Mail_clientBoxes.Selected=nil ) or
        ( Tree_Mail_clientBoxes.Selected.Parent = nil ) then exit;
    if ListView_LettersHeaders.Selected = nil then exit;

    Mail_clientBoxName := Tree_Mail_clientBoxes.Selected.Parent.Text;
    Mail_clientBoxFolder := Tree_Mail_clientBoxes.Selected.Text;

LetterList:=Mail_clientBoxes.FindLetters(Mail_clientBoxName,Mail_clientBoxFolder
);

Mail_clientBoxes.FindBox(Mail_clientBoxName).MoveToGarbage(LetterList,ListView_L
ettersHeaders.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_clientBoxName,Mail_clientBoxFolder : shortstring;
    LetterList : PLetters;
begin
    Mail_clientBoxName := Tree_Mail_clientBoxes.Selected.Parent.Text;
    Mail_clientBoxFolder := Tree_Mail_clientBoxes.Selected.Text;

LetterList:=Mail_clientBoxes.FindLetters(Mail_clientBoxName,Mail_clientBoxFolder
);

    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_clientBoxName+'> [відповідь]';
  frm_NewLetter.Show;
end;

procedure Tfrm_Main.aResendExecute(Sender: TObject);
var
  Mail_clientBoxName,Mail_clientBoxFolder : shortstring;
  LetterList          : PLetters;
  i                   : integer;
begin
  Mail_clientBoxName := Tree_Mail_clientBoxes.Selected.Parent.Text;
  Mail_clientBoxFolder := Tree_Mail_clientBoxes.Selected.Text;

  LetterList:=Mail_clientBoxes.FindLetters(Mail_clientBoxName,Mail_clientBoxFolder
);

  frm_NewLetter.ClearForm;
  frm_NewLetter.Ed_Subject.Text:='Fw:
'+LetterList.Letters[ListView_LettersHeaders.itemindex+1].Subject;
  frm_NewLetter.Caption:='Новий лист с <'+Mail_clientBoxName+'> [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_clientBoxesDragDrop(Sender, Source: TObject; X,
Y: Integer);
var
  node : TTreeNode;
  SourceMail_clientBoxName : shortstring;
  SourceFolderName : shortstring;
  SourceLetters : PLetters;

  DestMail_clientBox,
  DestFolderName : shortstring;
  DestLetters : PLetters;
begin
  node:=Tree_Mail_clientBoxes.GetNodeAt(x,y);
  if (node = nil)or(node.Parent = nil) then exit;
  if Tree_Mail_clientBoxes.Selected.Parent= nil then
SourceMail_clientBoxName:=Tree_Mail_clientBoxes.Selected.Text
    else
SourceMail_clientBoxName:=Tree_Mail_clientBoxes.Selected.Parent.Text;
  SourceFolderName:=Tree_Mail_clientBoxes.Selected.Text;

  SourceLetters:=Mail_clientBoxes.FindLetters(SourceMail_clientBoxName,SourceFolde
rName);

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

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

```

```

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

procedure Tfrm_Main.Tree_Mail_clientBoxesDragOver(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_clientBoxes.Selected =
nil)or(Tree_Mail_clientBoxes.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_clientBoxes.Selected=nil then exit;
    if Tree_Mail_clientBoxes.Selected.Parent = nil then exit;

Letters:=Mail_clientBoxes.FindLetters(Tree_Mail_clientBoxes.Selected.Parent.Text
, Tree_Mail_clientBoxes.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_clientBoxesClick(self);
end;

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

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

```

```

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

  Tree_Mail_clientBoxesClick(self);

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

Кафедра — КЮБ — 2023 рік

```

      for i:=1 to Letters.Count do
        begin
          frm_Main.AddToLog('Відправка листа від
'+CurrentMail_clientBoxSettings.Name+' к
'+Letters.Letters[1].Recipients.EMail_clientAddresses+' за темою:
'+Letters.Letters[1].Subject);
          DM.SMTP.Send( Letters.Letters[1]^ ); // Відправка листа
          Mail_clientBox.Sent.Add(Letters.Letters[1],false,Letters.LettersSizes[1]);
// Додати до папки
          Mail_clientBox.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 MyMail_client.dpr - файл проекту

```
program MyMail_client;

uses
  Forms,
  frmMain in 'frmMain.pas' {frm_Main},
  DataModule in 'DataModule.pas' {DM: TDataModule},
  frmMail_clientSettings in 'frmMail_clientSettings.pas'
{frm_Mail_clientSettings},
  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 := 'Поштовый клиент, розроблений Гушан М.М.';
  Application.CreateForm(Tfrm_Main, frm_Main);
  Application.CreateForm(TDM, DM);
  Application.CreateForm(Tfrm_Mail_clientSettings, frm_Mail_clientSettings);
  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 LetterStorage.pas - збереження листів

```

unit LetterStorage;

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

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

type
{
TMyAttachments = object
  public
    Mail_clientBoxName,
    FolderName   : shortstring;
    Count       : word;
  constructor Init( Mail_clientboxname, _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_clientBoxName,
    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_clientBox, 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_clientThread = class(TThread)
  private
    POP : TIdPOP3;
    Mail_clientBoxSettings : TMail_clientBoxSettings;
    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_clientBoxSettings;
IsBuisyFlag : Pboolean );
        end;

PMail_clientBox = ^TMail_clientBox;
TMail_clientBox = 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,lfDeleted,lfAll);

        CheckTimer       : TTimer;
        IsBuisyNow        : boolean;
        CheckThread       : TGetMail_clientThread;

        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_clientBoxList =object
    private
        Mail_clientBoxes :   array [1..100] of PMail_clientBox;
    public
        TotalBoxes:      word;
        function Add(name : shortstring)      : PMail_clientBox;
        function GetByName (name : shortstring) : PMail_clientBox;
        function GetByIndex(index: word)      : PMail_clientBox;
        function FindLetters( Mail_clientbox, Mail_clientfolder : shortstring
):PLetters;
        function FindBox( boxname : shortstring): PMail_clientBox;
        procedure SaveMail_clientBoxes;
        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_clientBox, Folder : shortstring);
Begin
  Mail_clientBoxName := Mail_clientBox;
  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_clientBoxAndSelectIt(Mail_clientBoxName);
  if CurrentMail_clientBoxSettings.DynamicFolders then exit;

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

  try
    stCompressed:=TFileStream.Create(SourceFile, fmOpenRead);
  except
    CreateEmptyLellerArchive(SourceFile);
    frm_Main.AddToLog('Не удалось найти архивные файлы ящика
<'+Mail_clientBoxName+'.>. 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_clientBoxName+' '+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_clientbx: PMail_clientBox;
  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_clientbx:=Mail_clientBoxes.FindBox(Mail_clientBoxName);
  if Mail_clientbx = nil then exit;
  IdIndex:=Mail_clientbx.DeletedIDs.IndexOf(mesID);
  if IdIndex<>-1 then exit
  else Mail_clientbx.DeletedIDs.Add(mesID);
end;

```

```

procedure TLetters.DeleteLetterWithoutFreeing(index: integer);
var
  i : word;
  temp : PidMessage;
  mesID : shortstring;
  Mail_clientbx: PMail_clientBox;
  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;

```

```

        end;
    dec (TotalLetters);
    temp.Free;
    dispose (temp);

    Mail_clientbx:=Mail_clientBoxes.FindBox (Mail_clientBoxName);
    if Mail_clientbx = nil then exit;
    IdIndex:=Mail_clientbx.DeletedIDs.IndexOf (mesID);
    if IdIndex<>-1 then exit
        else Mail_clientbx.DeletedIDs.Add (mesID);
end;

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

constructor TMail_clientBox.Init (name : shortstring);
Begin
    BoxName:=name;
    IsBuisyNow:=false;
    findMail_clientBoxAndSelectIt (name);
    CheckThread:=TGetMail_clientThread.Create (true);
    CheckThread.Init (CurrentMail_clientBoxSettings^, @IsBuisyNow);

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

    CheckTimer:=TTimer.Create (frm_Main);
    CheckTimer.OnTimer:=onTimer_Check;
    if CurrentMail_clientBoxSettings.CheckEvery<>0 then
    begin
        CheckTimer.Interval:=CurrentMail_clientBoxSettings.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_clientBox.GetName : shortstring;
Begin
    GetName:=BoxName;

```

End;

```

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

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

```

```

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

```

```

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

```

```

function TMail_clientBox.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 смітцевого 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_clientBoxList.Add(name : shortstring):PMail_clientBox;
Begin
  inc(TotalBoxes);
  new(Mail_clientBoxes[TotalBoxes]);
  Mail_clientBoxes[TotalBoxes]^ .init(name);
  Add:=Mail_clientBoxes[TotalBoxes];
End;

```

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

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

```
procedure TMail_clientBoxList.SaveMail_clientBoxes;
var
    i : word;
Begin
    for i:=1 to TotalBoxes do Mail_clientBoxes[i].Save;
End;
```

```
function TMail_clientBoxList.FindBox(boxname : shortstring):PMail_clientBox;
var
    i : word;
Begin
    FindBox:=nil;
    for i:=1 to TotalBoxes do
        if Mail_clientBoxes[i].BoxName = boxname then
            begin
                FindBox:=Mail_clientBoxes[i];
                break;
            end;
    End;
```

```

function TMail_clientBoxList.GetByIndex(index : word):PMail_clientBox;
Begin
  if (index>=1)and(index<=100) then GetByIndex:=Mail_clientBoxes[index]
  else GetByIndex:=nil;
End;

{ TMyAttachments }

constructor TMyAttachments.Init(_Mail_clientboxname,_foldername : shortstring);
Begin
  Count:=0;
  Mail_clientBoxName:=_Mail_clientBoxName;
  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_clientThread }

procedure TGetMail_clientThread.Init(var mbSettings : TMail_clientBoxSettings;
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_clientBoxSettings:=mbSettings;
  IsBuisy:=IsBuisyFlag;
end;

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

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

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

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

procedure TGetMail_clientThread.Execute;
var
  CurMail_clientBox      :   PMail_clientBox;
  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_clientBoxSettings.ShowBalloonOnNewMes then

```

```

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

CurMail_clientBox:=Mail_clientBoxes.GetByName(Mail_clientBoxSettings.Name);
LettersOnServer := POP.CheckMessages;
CurMail_clientBox^.UsedSizeOnServer := POP.RetrieveMail_clientBoxSize;

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

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

LettersNot2beDownloaded:=0;
for i:=1 to LettersOnServer do
    if Mail_clientBoxSettings.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_clientBoxes.GetByName(Mail_clientBoxSettings.Name)}CurMail_clientBox^.HasL
etterWithID(mes.MsgId) then
                    begin
                        inc(LettersNot2beDownloaded);
                        continue;
                    end;
                if
{Mail_clientBoxes.GetByName(Mail_clientBoxSettings.Name)}CurMail_clientBox^.hadL
etterWithId(mes.msgId) then
                    begin
                        CS.Enter;
                        case Mail_clientBoxSettings.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;
                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_clientBoxSettings.ShowBalloonOnNewMes then
        CallShowBalloon('Вхідний лист.#10#13+Від:
'+mes.From.Text+#10#13+Тема: '+mes.Subject);
        POP.Retrieve(i,mes);

{Mail_clientBoxes.GetByName(Mail_clientBoxSettings.Name)}CurMail_clientBox^.Inbo
x.Add(@mes,false, POP.RetrieveMsgSize(i));
    if Mail_clientBoxSettings.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_clientBoxesClick(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 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_clientSettings, 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;
  end;

```

```

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_clientBoxName : shortstring;
  CURR_Mail_clientbox_SETTINGS : TMail_clientBoxSettings;
  CURR_MAIL_CLIENT_BOX : PMail_clientBox;
  function CreateLetterFromForm:PidMessage;
  procedure UpdateStatusLine;
  procedure ClearForm;
end;

procedure FillSMTPwithMail_clientBoxSettings(mbSettings :
PMail_clientBoxSettings);

var
  frm_NewLetter: Tfrm_NewLetter;

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

procedure FillSMTPwithMail_clientBoxSettings(mbSettings :
PMail_clientBoxSettings);
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_clientBoxSettings = 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_clientBoxSettings(CurrentMail_clientBoxSettings);
    frm_Main.AddToLog('Негайна відправка листа від
'+CurrentMail_clientBoxSettings.Name+' до
'+mes.Recipients.EMail_clientAddresses);

```

```

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

```

// 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_clientBoxesClick(self);
  if CurrentMail_clientBoxSettings =nil then exit;

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

  PLet^.Subject:=Ed_Subject.Text;
  PLet^.CCList.Email_clientAddresses:=Ed_Copy.Text;
  PLet^.BccList.Email_clientAddresses:=Ed_HiddenCopy.Text;
  PLet^.Body.Assign(Memo_Body.Lines);
  PLet^.From.Text:= CURR_Mail_clientbox_SETTINGS.POPEmail_client; raises error
on some servers
  PLet^.From.Text:=CurrentMail_clientBoxSettings^.POPEmail_client;
  if Ed_ReplyTo.Text='' then
  PLet^.ReplyTo.Email_clientAddresses:=CurrentMail_clientBoxSettings.POPEmail_client
  nt
      else
  PLet^.ReplyTo.Email_clientAddresses:=Ed_ReplyTo.Text;
  PLet^.Recipients.Email_clientAddresses:=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_clientbox_SETTINGS:= CurrentMail_clientBoxSettings^;

CURR_MAIL_CLIENT_BOX:=Mail_clientBoxes.FindBox(CurrentMail_clientBoxSettings.Nam
e);
end;

end.
```

Кафедра _ КБПЗ _ 2023 рік

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

```

unit frmMail_clientSettings;

interface

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

type
  PMail_clientBoxSettings = ^TMail_clientBoxSettings;

  Tfrm_Mail_clientSettings = class(TForm)
    ListBox_Mail_clientBoxes: 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_clientBox: TBitBtn;
    GroupBox4: TGroupBox;
    Label11: TLabel;
    Ed_Subscript: TEdit;
    Ed_Mail_clientBoxName: 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_client: TEdit;
    Check_DynamicFolders: TCheckBox;
    Check_ShowBalloon: TCheckBox;

    procedure btn_NewMail_clientBoxClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  end;

```

```

procedure Combo_CheckEveryExit(Sender: TObject);
procedure btn_ApplyClick(Sender: TObject);
procedure ListBox_Mail_clientBoxesClick(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_clientBoxSettings);
  procedure SaveFormToBox(box : PMail_clientBoxSettings);
  procedure ClearForm;
end;

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

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

  POPServer    :shortstring;
  POPPort      :integer;
  POPAccount   :shortstring;
  POPPass      :shortString;
  POPMail_client :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_clientBoxAndSelectIt(name : shortstring);

const
  Mail_clientBoxSettingsFileName = 'Mail_clientboxes.dat';

var
  frm_Mail_clientSettings      : Tfrm_Mail_clientSettings;
  TotalMail_clientBoxes       : word=0;
  Mail_clientBoxesSettings    : array [1..300] of PMail_clientBoxSettings;
  CurrentMail_clientBoxSettings : PMail_clientBoxSettings=nil;
  PrevMail_clientBoxSettings  : PMail_clientBoxSettings=nil;

implementation

uses frmMain;
{$R *.dfm}

```

```

procedure findMail_clientBoxAndSelectIt(name : shortstring);
var
  i : word;
Begin
  CurrentMail_clientBoxSettings := nil;
  for i:=1 to TotalMail_clientBoxes do
    if Mail_clientBoxesSettings[i]^Name = name then
      begin
        CurrentMail_clientBoxSettings:=Mail_clientBoxesSettings[i];
        exit;
      end;
  End;

procedure SaveSettingsToFile;
var
  SFile : shortstring;
  st : TFileStream;
  i : word;
Begin
  SFile:=ExtractFilePath(Application.ExeName)+Mail_clientBoxSettingsFileName;
  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 могу відкрити файл "'+SFile+'" для запису !'), 'Помилка збереження параметрів скриньки');
      exit;
    end;
  end;
  st.Write(TotalMail_clientBoxes, sizeof(TotalMail_clientBoxes));
  for i:=1 to TotalMail_clientBoxes do
    st.write(Mail_clientBoxesSettings[i]^, sizeof(TMail_clientBoxSettings));
  st.Free;
End;

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

  st.Read(TotalMail_clientBoxes, sizeof(TotalMail_clientBoxes));
  for i:=1 to TotalMail_clientBoxes do
    begin
      new(Mail_clientBoxesSettings[i]);
      st.Read(Mail_clientBoxesSettings[i]^, sizeof(TMail_clientBoxSettings));
    end;
  st.Free;
End;

procedure Tfrm_Mail_clientSettings.LoadFormFromBox(box :
PMail_clientBoxSettings);
Begin

```

```

Ed_SMTPServer.Text := box.SMTPServer;
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_client.Text := box.POPEmail_client;

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

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

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_clientSettings.SaveFormToBox(box : PMail_clientBoxSettings);
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_client := Ed_POPEmail_client.Text;

  box.Subscript := Ed_Subscript.Text;
  case Combo_SMTPType.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_clientSettings.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_client.Text := '<nick>@<server>.ru';
Ed_Subscript.Text := 'Вася';

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

procedure Tfrm_Mail_clientSettings.LoadSettingsFromRecords;
Begin
if TotalMail_clientBoxes <> 0 then
begin
if Mail_clientBoxesSettings[1]<>nil then
begin
CurrentMail_clientBoxSettings:=Mail_clientBoxesSettings[1];

LoadFormFromBox(CurrentMail_clientBoxSettings);
ListBox_Mail_clientBoxes.Selected[0]:=true;
PrevMail_clientBoxSettings:=CurrentMail_clientBoxSettings;
end;
end
else CurrentMail_clientBoxSettings:=nil;
End;

// створення нової поштової скриньки
procedure Tfrm_Mail_clientSettings.btn_NewMail_clientBoxClick(Sender: TObject);
var
i : word;
begin
if Ed_Mail_clientBoxName.Text<>' ' then
begin
for i:=1 to ListBox_Mail_clientBoxes.Count do
if AnsiUpperCase(ListBox_Mail_clientBoxes.Items[i-
1])=AnsiUpperCase(Ed_Mail_clientBoxName.Text) then
begin
Application.MessageBox('Таке ім'я облікового запису вже
існує!', 'Помилка');
exit;
end;
ListBox_Mail_clientBoxes.Items.Add(Ed_Mail_clientBoxName.Text);
GroupBox1.Visible:=true;
Radio_Ask.Checked:=true;
inc(TotalMail_clientBoxes);
new(Mail_clientBoxesSettings[TotalMail_clientBoxes]);

CurrentMail_clientBoxSettings:=Mail_clientBoxesSettings[TotalMail_clientBoxes];
CurrentMail_clientBoxSettings.Name:=Ed_Mail_clientBoxName.Text;
if PrevMail_clientBoxSettings<>nil then
SaveFormToBox(PrevMail_clientBoxSettings);
PrevMail_clientBoxSettings:=CurrentMail_clientBoxSettings;
ClearForm;

```

```

    SaveFormToBox(CurrentMail_clientBoxSettings);
    ListBox_Mail_clientBoxes.Selected[ListBox_Mail_clientBoxes.Count-1]:=true;
    {Тепер створити і заповнити нову поштову скриньку}
    Mail_clientBoxes.Add(Ed_Mail_clientBoxName.Text);
    frm_Main.AddToLog('Створена поштова скринька
<'+Ed_Mail_clientBoxName.Text+'>');
    frm_Main.ShowMail_clientBoxesInTree;
end
else
    Application.MessageBox('Спочатку введіть ім'я вашої поштової
скриньки', 'Помилка');
end;

procedure Tfrm_Mail_clientSettings.FormShow(Sender: TObject);
var
    i : word;
begin
    if (ListBox_Mail_clientBoxes.Count=0)and(TotalMail_clientBoxes<>0) then
        for i:=1 to TotalMail_clientBoxes do
            ListBox_Mail_clientBoxes.Items.Add(Mail_clientBoxesSettings[i].Name);
            LoadSettingsFromRecords;
            if CurrentMail_clientBoxSettings=nil then GroupBox1.Visible:=false
                else GroupBox1.Visible:=true;
end;

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

procedure Tfrm_Mail_clientSettings.btn_ApplyClick(Sender: TObject);
var
    i : word;
begin
    if ListBox_Mail_clientBoxes.ItemIndex>=0 then
        begin
            SaveFormToBox(CurrentMail_clientBoxSettings);
            SaveSettingsToFile;
            self.Hide;
            for i:=1 to TotalMail_clientBoxes do
                if Mail_clientBoxes.GetByIndex(i)<>nil then
                    Mail_clientBoxes.GetByIndex(i).SetTimerInterval(Mail_clientBoxesSettings[i].Chec
kEvery);
            end;
end;

function FindMail_clientBox(name : shortstring):PMail_clientBoxSettings;
var
    i : word;
begin
    FindMail_clientBox:=nil;
    for i:=1 to TotalMail_clientBoxes do
        if Mail_clientBoxesSettings[i].Name = name then
            begin

```

```

        FindMail_clientBox:=Mail_clientBoxesSettings[i];
        exit;
    end;

End;

procedure Tfrm_Mail_clientSettings.ListBox_Mail_clientBoxesClick(Sender:
TObject);
begin
    if ListBox_Mail_clientBoxes.ItemIndex >=0 then
    begin

CurrentMail_clientBoxSettings:=FindMail_clientBox(ListBox_Mail_clientBoxes.Items
[ListBox_Mail_clientBoxes.itemindex]);
        if CurrentMail_clientBoxSettings=nil then
CurrentMail_clientBoxSettings:=PrevMail_clientBoxSettings;
        if PrevMail_clientBoxSettings<>nil then
SaveFormToBox(PrevMail_clientBoxSettings);
        PrevMail_clientBoxSettings:=CurrentMail_clientBoxSettings;
        LoadFormFromBox(CurrentMail_clientBoxSettings);
        end;
    end;

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

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

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

procedure Tfrm_Mail_clientSettings.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
    btn_ApplyClick(self);
end;

end.

```

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);
    
```

```

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.
```

Кафедра _ КБПЗ _ 2023рік

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.
```