

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 поштового клієнта
електронного документообігу за протоколами POP3 та SMTP”**

Виконав здобувач вищої освіти
IV курсу, групи КІ-20-3СК
ОПІ «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Гуцол В.О.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, доцент
_____ Коваленко О.В.
« ____ » _____ 2023 р.

Рецензент _____

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 17 » січня 2023 року

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

Гуцолу Віталію Олександровичу

(прізвище, ім’я, по батькові)

- Тема роботи *Програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP*
- Керівник роботи *Коваленко Олександр Володимирович, докт. тех. наук, доцент*
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 8-02 від 5.01.2023 року
- Строк подання студентом роботи до захисту *23.05.2023 р.*
- Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Перегляд аналогічних існуючих систем.*
 - Опис і обґрунтування проектних рішень.*
 - Етапи програмування системи.*
 - Впровадження системи в промислову експлуатацію.*
 - Висновки*
- Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

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

7. Дата видачі завдання « 17 » січня 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

Коваленко О.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2023 р.

Підпис здобувача

Гуцол В.О.
(прізвище та ініціали)

АНОТАЦІЯ

Гуцол В.О. Програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

Метою розробки є програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

Результат роботи – програмна реалізація системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

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: комп'ютерна інженерія, поштовий клієнт, POP3, SMTP

ABSTRACT

Hutsol V.O. Software of the mail client system of electronic document management based on POP3 and SMTP protocols. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the first (bachelor) level of higher education, software was developed, which is intended for the mail client system of electronic document management based on the POP3 and SMTP protocols.

The purpose of the development is the software of the mail client system of electronic document management based on the POP3 and SMTP protocols.

The result of the work is the software implementation of the mail client system of electronic document management based on the POP3 and SMTP protocols.

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

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

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10 environment.

Keywords: computer engineering, mail client, POP3, SMTP

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	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	22
2.3 Розгорнута постановка завдання	28
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	30
3.1 Опис функціонування системи	30
3.2 Розробка структурної схеми.....	37
3.3 Розробка функціональної схеми	39
3.4 Розробка діаграми процесів.....	46
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	48
4.2 Захист розробленого програмного забезпечення.....	63
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	67
6 ОСНОВНІ ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75

					ВКРБ-123.23.0014.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Гуцол В.О.</i>					Б	1	84
<i>Перев.</i>	<i>Коваленко О.В.</i>					<i>ЦНТУ КІ-20-3СК</i>		
<i>Н.контр.</i>	<i>Гермак В.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- ПЗ – програмне забезпечення
- ASLR – Випадкове Розміщення в Адресному Просторі
- DEP – Запобігання Виконання Даних
- DNS – Domain Name System
- IMAP – Internet Message Access Protocol
- MIME – Multipurpose Internet Mail Extension
- POP – Post Office Protocol
- NRPC – віддалений виклик процедур Notes
- RM – Менеджером Презавантажень
- RSS – Really Simple Syndication
- SMTP – Simple Mail Transfer Protocol
- SSL – Secure Socket Layer
- TLS – Transport Layer Security
- USENET – User Network

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. 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:

- 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;
- ви помітно заощаджуєте вхідний трафік, тому що не завантажуйте 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.

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.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

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

- Огляд існуючих систем поштового клієнта електронного документообігу за протоколами POP3 та SMTP.
- Дослідження системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.
- Програмна реалізація системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

Таким чином, виходячи з вищеперахованого, програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

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);

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

– інші клієнти Web від постачальників послуг Internet, зокрема GMX або Yahoo!. Однак ці клієнти звичайно мають лише базову функціональність.

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.

1.2 Область застосування

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 киоскових або застосовуватися мобільними співробітниками на підприємстві.

Клієнти на базі Web ведучих систем обміну повідомленнями, навпроти, здатні продемонструвати широкий функціональний охват, що усе більше наближається до функціонального охопту 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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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 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 лише починаючи з версії 6.5.1.

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 поштового клієнта електронного документообігу за протоколами POP3 та SMTP, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Світ поштових клієнтів цікавий тим, що тут не знайти домінуючого продукту. Крім того, чимало користувачів всесвітньої мережі воліють працювати з поштою через Web-Інтерфейс, використовуючи браузер. Проведемо порівняльний аналіз популярних поштових клієнтів.

Outlook Express

Корпорація Microsoft позиціонує свій продукт Outlook Express як додатковий додаток до Internet Explorer для роботи з електронною поштою. Дійсно, інтерфейс поштового клієнта має багато загального з вбудованим в Windows 10/11 браузером.

Потужний Wizard при первісному старті полегшує перше знайомство із програмою для новачків. Як панель інструментів використовуються стандартні засоби Windows, кількість кнопок на панелі зведено до мінімуму, що робить освоєння поштового клієнта максимально простим.

Дуже грамотно реалізована стартова сторінка поштового клієнта, на якій розміщуються посилання на всі основні дії з ним, а також одна довільна корисна рада, пов'язаний з освоєнням програми. Outlook Express має велику україномовну довідку, що описує най докладнішим образом всі нюанси роботи із програмою.

Швидкість

Максимальна інтеграція з операційною системою дозволяє Outlook Express працювати на надзвучових швидкостях. Одне з достоїнств поштового клієнта – це висока швидкість обробки пошти, незалежно від обсягу переданої інформації.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Весь інтерфейс поштового клієнта заснований на стандартних бібліотеках Windows, що забезпечує відмінну швидкість промальовування вікон.

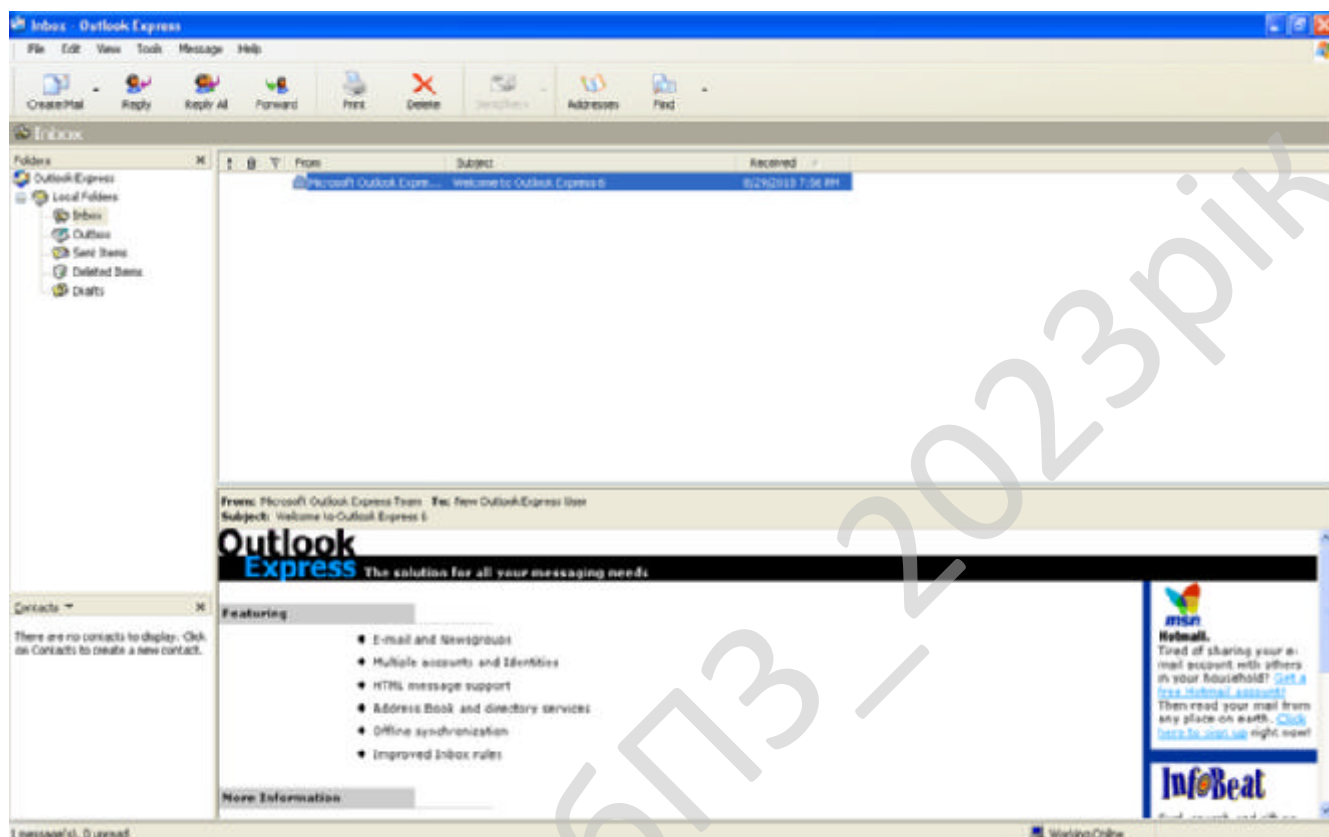


Рисунок 2.1 – Інтерфейс користувача Outlook Express

Редактор повідомлень

Створювати повідомлення у форматі HTML, використовуючи Outlook Express, у край незручно. Ви не можете вставити свої рядки в середину або кінець повідомлення, тому що вертикальна риса цитування нерозривна. Вам доводиться становити своє повідомлення винятково над процитованим тілом оригіналу. HTML-код, генеруємий редактором повідомлень, дуже громіздкий і в багато разів перевищує текстовий еквівалент.

Технологія шаблонів із застосуванням різноманітних змінних в Outlook Express не реалізована.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Функціональність

Можливості Outlook Express є якимось стандартом, від якого відштовхуються інші розроблювачі в надії завоювати користувача чимсь більш функціональним. Outlook Express має стандартний набір функцій, застосовний до багатьох поштових клієнтів.

Поштовий клієнт містить елементи інтеграції з Windows Messenger, що дозволяє синхронізувати панелі контактів обох програм.

Сторонні розроблювачі дозволяють розширити можливості Outlook Express. На сьогоднішній день написана безліч зовнішніх менеджерів повідомлень, які можуть перевіряти пошту на сервері, відкривати заголовки, а потім передавати керування поштовою клієнтові. Також існують програми, що дозволяють вставляти в повідомлення, створювані в Outlook Express смайлики.

Сортування пошти

Сортування пошти обмежується фільтрами, згідно яких ми можемо виконувати які-небудь дії на основі строго заданих умов. Ви можете створювати додаткові папки, у які надалі поміщати повідомлення, що задовольняють особливим критеріям.

Безпека

Outlook Express – самий уразливий поштовий клієнт у світі. Використання движка Trident, а також відсутність яких-небудь інструментів для його "нейтралізації" роблять його небезпечним. Статистика говорить про те, що Outlook Express – основний "провідник" вірусів на ваш комп'ютер. Зараження вірусами через браузер відбуваються набагато рідше.

Висновок

Outlook Express – дуже простий і швидкий інструмент для невибагливих користувачів. Низька захищеність від вірусів – основна проблема Outlook Express, що доводиться вирішувати за допомогою потужної антивірусної оборони.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Outlook

Освоєння

Сьогодні важко знайти користувача персонального комп'ютера, який би жодного разу не бачив Microsoft Office. Розуміння інтерфейсу офісного пакета – ключ до розгадки таїнства освоєння Outlook. Все налаштування програми можна сміло довірити Wizard. На жаль, імпорт пошти можливий тільки з Outlook Express.

Відповіді на безліч питань даються в докладній україномовній довідці, однак прийдеться витратити велику кількість трафіка, тому що основна частина довідкової системи являє собою компактний браузер і завантажує всі дані із сайту компанії розроблювача. Крім того, на сайті розроблювача можна знайти окремі навчальні курси по програмі.

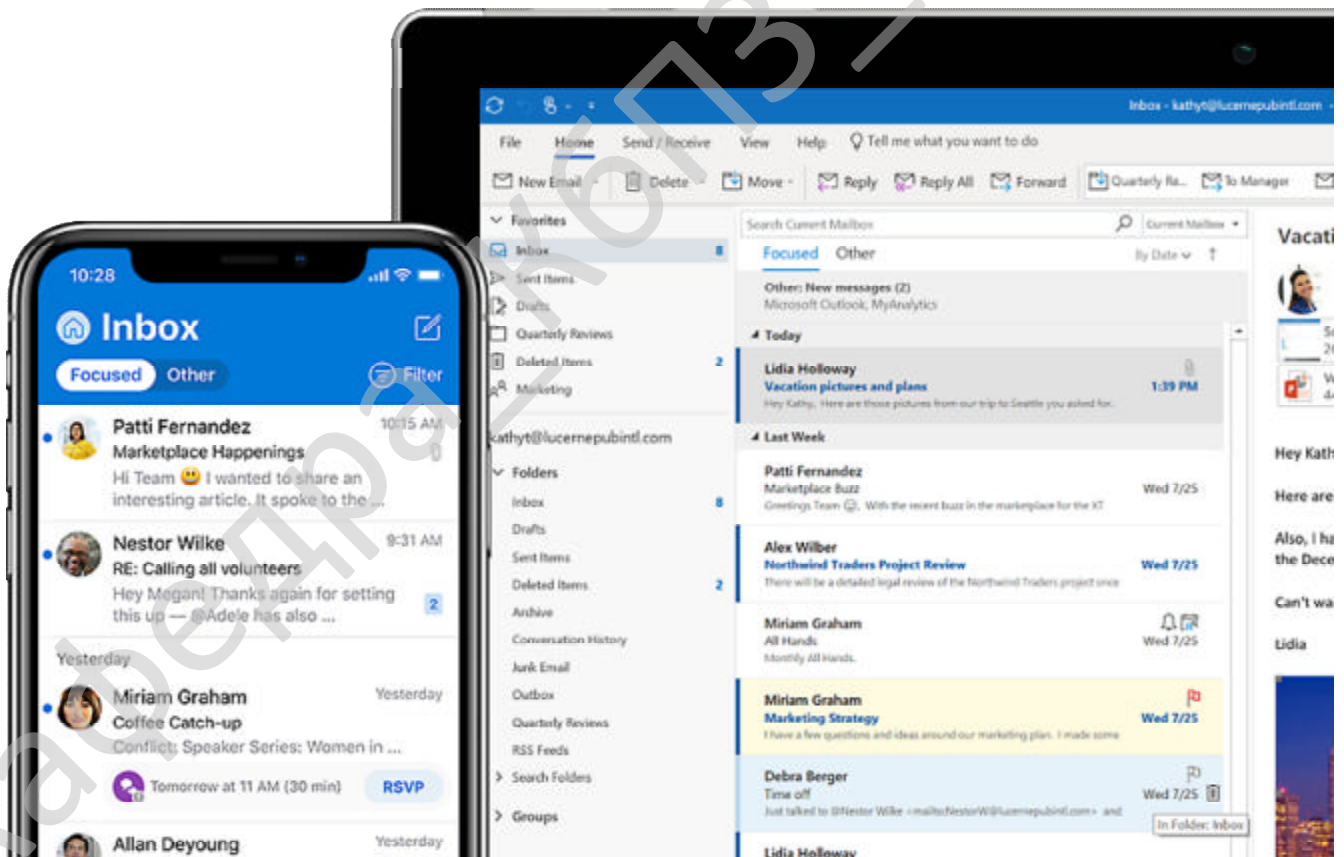


Рисунок 2.2 – Інтерфейс користувача Outlook

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Розкладка вікон за замовчуванням відрізняється від загальноприйнятої, але це легко виправляється. Головне меню програми не перевантажено, незважаючи на широке коло розв'язуваних програмою завдань, розташування елементів логічно продумано.

Швидкість

Швидкість завантаження, системні вимоги й багато чого іншого не сильно відрізняються від інших офісних програм – Word, Excel і ін.

Редактор повідомлень

За замовчуванням Outlook не виділяє цитування. У налаштуваннях можна вказати програмі додавати вертикальну рису при відповіді на повідомлення у форматі HTML, а також вручну вибрати символ для текстового цитування. Вертикальна риса нерозривна, але ви можете виділяти власний текст за допомогою довільного заголовка, що також вказується в налаштуваннях.

Перевірка орфографії на льоту працює тільки при підключенні Word як редактор повідомлень. За допомогою Word ви можете додавати у свої листи елементи ActiveX.

Після підготовки повідомлення, його текст можна перевести на ті мови миру, для яких у вас установлені словники.

Функціональність

Якщо попередні версії Outlook об'єктивно програвали конкурентам по багатьох функціональних критеріях, то в останній версії розроблювачі приділили велику увагу нарощуванню традиційних функцій, властивим популярним сучасним поштовим клієнтам.

В Outlook включена технологія віртуальних папок, що називається "Папки пошуку". Використовуючи останню версію, ви можете завантажувати тільки заголовки повідомлень і лише на другому етапі завантажувати повідомлення цілком.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

В Outlook розроблювачі приділили величезну увагу боротьбі зі спамом. Із простого набору фільтрів, аналогічного Outlook Express, інструментарій блокування небажаної пошти перетворився в справжній форпост.

Корпоративним користувачам надається можливість спільної роботи з органайзером і адресною книгою.

Безпека

HTML движок Trident, Visual Basic у макросах і можливість використання елементів ActiveX у повідомленнях – легка мішень для самих витончених вірусних атак. На щастя, багато антивірусів бачать Outlook у системі й ретельно стежать за небезпекою. Без наявності антивірусу з базою, що регулярно обновляється, використовувати Outlook у край небажано.

Сортування пошти

Outlook підтримує віртуальні папки й стандартні фільтри. Блокування спама здійснюється на основі повного аналізу повідомлень, а не тільки по окремих параметрах (Відправник, Тема, наявність певних слів у тілі повідомлення). Крім інструментів аналізу, можна використовувати списки довірених відправників і адресатів. Сортування повідомлень по даті здійснюється за допомогою списків, що розкриваються (Сьогодні, Учора, Минулого тижня). Групи повідомлень, що є ланцюжком обговорення, можуть також згортатися в один рядок. Подібний принцип відображення обговорень застосовується на Google Mail.

Висновок

Outlook – потужний корпоративний поштовий клієнт, що володіє при цьому зручним інтерфейсом і масою корисних функцій для домашнього застосування. Однак не варто забувати, що без антивірусу зі свіжою базою ви дуже сильно ризикуєте.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

бажаючих уперше перейти на молдавську програму, це може стати серйозною перешкодою. Україномовна термінологія, використувана в програмі, сильно відрізняється від звичних стандартів. Відсутність стартової сторінки також може ускладнити перші кроки. Довідка англійською мовою.

Швидкість

Незважаючи на максимальне дистанціонування від оточення операційної системи, The Bat! показує непогану швидкість роботи. Поштовий клієнт демонструє середній час першого завантаження й досить швидко відображає повідомлення з розсиланнями, що мають HTML-форматування. Володіючи багатопоточністю, The Bat! дозволяє одержувати й відправляти листа одночасно, що підвищує швидкість взаємодії із серверами. Скромні системні вимоги дозволяють кажанові швидко працювати навіть на застарілих робочих станціях.

Редактор повідомлень

Потужний інструмент керування шаблонами дозволяє гнучко набудувати вітання й підписи відповідно до ваших переваг. The Bat! – єдиний поштовий клієнт, у якому ви можете застосувати вирівнювання "по ширині" у текстовому режимі. Крім того, якщо ви переплутали регістр букв, те це можна швидко виправити за допомогою спеціальної групи функцій. Швидкі шаблони дозволяють автоматизувати введення часто вживаємих фраз. Підтримка "девізів" надає додатковий шарм і елементи гумору в особистій переписці.

Серйозні проблеми починаються, коли ви переходите в режим створення повідомлень у форматі HTML. Ви не можете розірвати цитату й змушені цілком уводити свій текст над/під вихідним повідомленням. При зміні формату з HTML у простий текст кількість рядків у повідомленні подвоюється. Слідом за кожним інформативним рядком буде впливати ще один, порожній. Створюване повідомлення у форматі HTML перевищує розмір аналогічного текстового повідомлення більш ніж в 4 рази.

The Bat! орієнтований на серйозного користувача, для якого надається маса сервісних функцій. Кажан має відмінного менеджера листів, що дозволяють

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

відсівати непотрібні повідомлення до переносу на локальний носій. Процес обміну повідомленнями із сервером ретельно візуалізований, ви можете бачити наочний індикатор, що відображає прогрес роботи. Наявність вбудованого планувальника SmartBat дозволяє кажанові конкурувати з Outlook у корпоративному середовищі. Досить корисним буває наявність журналу роботи, де зберігається інформація про всі мережні операції.

The Bat! здатний експортувати/імпортувати пошту, працюючи з безліччю форматів. Кажан непохитно витримує імпорт багатотисячних поштових баз із Outlook, Outlook Express і всіх листонош із зоопарку Mozilla. Повний список поштових клієнтів, з яких ви можете імпортувати базу, складається з восьми пунктів. The Bat! має можливість роботи як сервер у локальній мережі. Ви можете міняти пароль на сервері прямо з поштового клієнта, організовувати локальний чат за допомогою e-mail повідомлень, шифрувати листа, а також задавати вручну тип підключення до мережі для роботи поштового клієнта.

Власний HTML движок, використовуваний в The Bat!, не дозволяє домагатися точної передачі форматування в повідомленнях. Перекручування стосуються, у першу чергу, відображення таблиць. Крім того, кажан не розуміє тег цитування <blockquote>, тому в HTML-повідомленнях цитати часто зливаються із загальним текстом.

У налаштуваннях сортувальника листів можна створювати потужні фільтри, що дозволяють не тільки розподіляти вхідну пошту по потрібних папках, але й заощаджувати вхідний трафік, завантажуючи тільки ті повідомлення, які задовольняють певним умовам. По кожному фільтру ведеться статистика роботи, що дозволяє бачити його ефективність. Підключивши зовнішні модулі, ви можете домогтися ефективного знищення спама.

The Bat! не використовує движок Trident для обробки HTML-коду. Це робить кажана практично невразливим для вірусів. Зрозуміло, якщо ви будете запускати незнайомі прикріплені до повідомлень файли, наприклад, nudegirl.jpg__exe, те можуть виникнути проблеми. Однак навіть у цьому

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

випадку поштовий клієнт покаже попередження про небезпеку. На додаток до всього, The Bat! має можливість автоматичної перевірки вкладень за допомогою популярних антивірусів. The Bat! має власну адресну книгу, не пов'язану з Windows Address Book, що обгороджує користувачів кажана від додаткового джерела проблем з безпекою. Ви можете ставити паролі на окремі поштові скриньки, що підвищує вашу особисту безпеку.

The Bat! складний в освоєнні, але має безліч корисних інструментів. На одній чаші ваг лежить украй неякісна реалізація роботи з HTML-кодом, а на іншій – колосальна безпека.

Mozilla Thunderbird

Mozilla Thunderbird не оптимізований для роботи через модемне з'єднання. Однак при використанні широкополосного підключення ви можете повною мірою відчувати всю міць. Висока безпека й відмінна робота з HTML – візитні картки Mozilla Thunderbird.

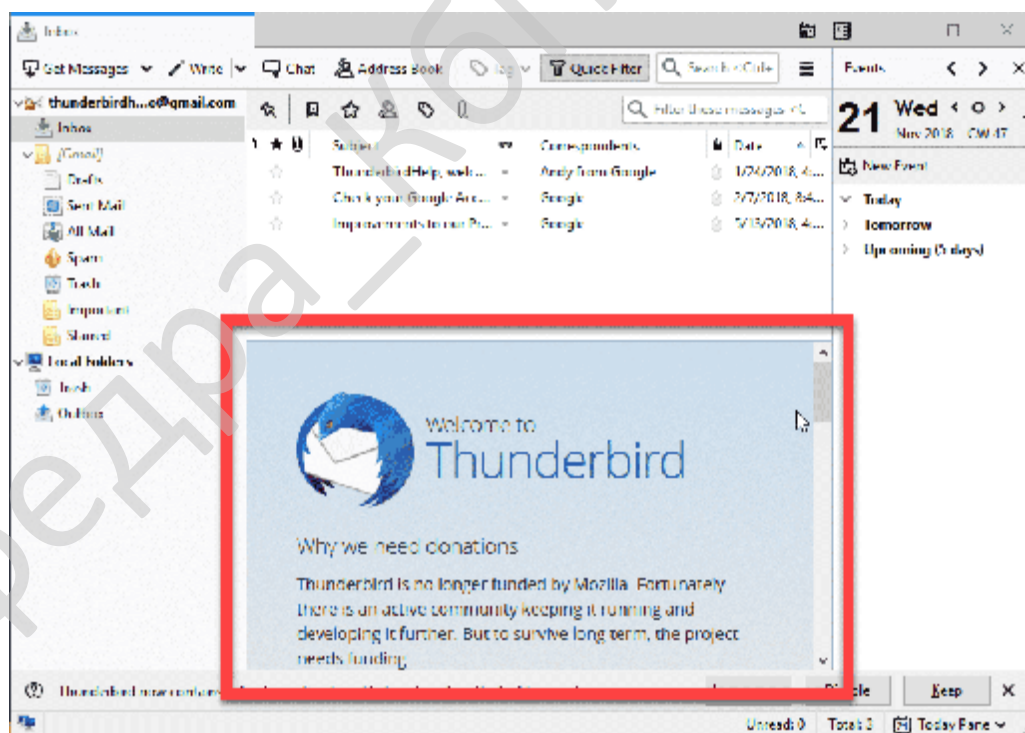


Рисунок 2.4 – Інтерфейс користувача Mozilla Thunderbird

Продовження таблиці 2.1

										
HTML движок	Trident	Trident	без назви	Gecko	KHTML	Trident	Trident	Gecko	Presto	Trident
HTML-редактор	+	+	+	+	+	+	+	+	-	+
Попереднє завантаження заголовків повідомлень	*	+	+	+	+	+	+	-	-	*
Перевірка української орфографії на ходу	-	-	**	**	+	-	**	+	-	-
Розширення	+	+	+	+	-	+	-	+	-	+
Скіни	-	-	*	+	+	-	-	-	+	*
Автоматична навчальна фільтрація спаму	*	+	***	+	+	-	+	+	+	*
Віртуальні папки	-	+	+	+	-	-	-	+	+	-
Роздільні папки для облікових записів / загальна група папок	-/+	-/+	+/-	+/+	-/+	+/-	-/+	-/+	-/+	-/+
NNTP	+	+	*	+	-	-	+	+	+	-
RSS	-	*	*	+	-	-	+	-	+	-
Безпека HTML	-	-	+	+	+	-	-	+	+	-

Продовження таблиці 2.1

										
Шифрування повідомлень	+	+	+	*	+	-	+	+	-	-
Повна підтримка кирилиці	+	+	+	+	+	-	-	+	+	-
Швидкі шаблони (макроси)	-	+	+	*	-	-	+	-	-	-
Бібліотека смайликів	*	-	-	+	-	-	+	+	+	+
Цитування за допомогою ініціалів	-	-	+	-	+	+	-	-	-	-
Довільні кольори для рівнів цитування	-	-	+	*	+	-	+	-	+	-
Спливаючі повідомлення про нові повідомлення	*	+	+	+	+	-	+	-	+	+
Автодозвон	+	+	+	-	-	+	+	-	-	-
Час першого старту, сек	2-3	3-5	3-5	5-7	2-3	2-3	3-5	5-7	3-5	5-7
Витрата оперативної пам'яті, МБ	20-25	30-40	30-40	30-50	30-50	12-15	16-23	60-80	20-25	30-50

Таблиця 2.2 – Позначення значків поштових клієнтів

	Outlook Express
	Outlook
	The Bat!
	Mozilla Thunderbird
	KMail
	Becky!
	PocoMail
	Evolution
	Opera
	IncrediMail

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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

Основні можливості Delphi 10.4.1:

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

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

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

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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

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

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

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

Єдине керування пам'яттю

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

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

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

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

Підвищення якості й швидкодії інструментів

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

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

масштабується під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений. Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

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.

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 поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

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

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

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

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

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

в роботі технологічного обладнання;

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

ВКРБ-123.23.0014.00.00.ПЗ

Арк.

29

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис поштового клієнту

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

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

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

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

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

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

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

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

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

Основне завдання поштового клієнта – одержання із сервера нових повідомлень і відправлення написаних користувачем. Для цього програма повинна підтримувати відповідні протоколи: POP3 або IMAP для одержання й SMTP для відправлення. У випадку якщо пошта зберігається на сервері, призначеному для забезпечення спільної роботи співробітників підприємства необхідна і їхня підтримка. Деякі поштові сервіси дозволяють використовувати для доставки пошти тільки захищені з'єднання; для роботи з такими сервісами програма повинна підтримувати протоколи шифрування TLS і SSL .

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

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

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

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

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

Поштовий клієнт повинен мати наступні достоїнства:

- будь-яка кількість обслуговуваних одночасно POP3 аккаунтів на різних серверах;
- багатомовний інтерфейс із можливістю перемикання російського, українського і багато інших (після завантаження мовного модуля із сервера);

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- диспетчер листів для маніпуляцій з листами безпосередньо на POP3 сервері;
- потужні засоби фільтрації листів;
- абсолютно коректна робота з усіма російськими і європейськими кодуваннями, причому гнучка система налаштувань дозволяє встановлювати свої кодування для кожного ящика й кожного адресата;
- можливість використовувати індивідуальні шаблони листів, створені для кожного ящика, кожної папки й кожного адресата, з урахуванням мови, форми вітання, національних кодувань і т.д. Все це управляється системою макросів і дуже легко програмується;
- система фільтрів, макросів і шаблонів є настільки потужною, що дозволяє перетворити поштовий клієнт у сервер списку розсилання. Програма сама буде перевіряти потрібні аккаунти, фільтрувати, модифікувати поля From, To, і т.д., модифікувати зміст повідомлень і потім розсилати їхнім потрібним адресатам списку;
- повна багатозадачність і багатопоточність – програма може перевіряти аккаунти, вести сортування повідомлень і одночасно дозволяє переглядати й редагувати листа;
- потужні засоби пошуку;
- зручна й дуже потужна записна книжка з фотографіями;
- убудована підтримка PGP, що базується на бібліотеці SSLeay;
- підтримка смарт-карт із секретним ключем PGP – для поштових транзакцій типу "банк-клієнт" і інших сфер, де потрібна підвищена таємність;
- можливість імпорту повідомлень із форматів найбільш популярних поштових клієнтів (Outlook, Netscape Communicator, Netscape Mail, Eudora, Pegasus Mail), а також поштових скриньок Unix;
- підтримка протоколів IMAP4, POP, APOP, SMTP;
- можливість закриття доступу до окремих ящиків паролем;

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

– убудовані функції поштового сервера локальної мережі (TCP/IP – Non-TCP/IP шлюз);

– форми запиту – інструмент для формування запитів, що підлягають автоматичній обробці. Потужний засіб керування електронним документообігом у бізнесі;

– запис журналів роботи для кожного ящика;

– зручний текстовий редактор, автоматична перевірка орфографії кількома мовами, внутрішня функція дозвона, що біжить рядок "Mail Ticker", і безліч інших корисних функцій, опис яких зайняло б ще багато місця й часу.

Після опису поштового клієнту, надамо опис структури поштового повідомлення, яке оброблюється цим клієнтом.

Структура поштового повідомлення

Повідомлення електронної пошти звичайно складається із двох частин:

– 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.).

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.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– 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.

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

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

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

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

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

Тіло повідомлення

Споконвічно передбачалося, що поштові повідомлення можуть містити тільки текст у форматі ASCII. А оскільки можливість передачі нетекстової інформації не передбачалася, то протоколи передачі електронної пошти можуть некоректно обробляти такі повідомлення. У зв'язку із цим у свій час був розроблений спеціальний стандарт, що визначає принципи перетворення

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

нетекстових даних до текстового виду. Цей стандарт одержав назву MIME (Multipurpose Internet Mail Extension, багатоцільове розширення пошти Інтернет).

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

- текст – простий текст у форматі ASCII, а також текст у форматі RTF або HTML;
- графічні зображення – файли у форматі JPEG і GIF;
- аудіо й відео дані;
- дані у форматах різних додатків, наприклад, документи Microsoft Office, а також дані довільного формату (у тому числі, різні файли, що виконуються).

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

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

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

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

– алгоритм "Base64", що здійснює перетворення трьох довільних байт у чотири ASCII-символи.

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

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

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

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

– Блок захисту інформації, фільтрів та антиспаму – призначений для забезпечення конфіденційного електронного документообігу по протоколам POP3 та SMTP та захисту від спаму, тобто листів усілякої реклами, які не потрібні користувачу поштової скриньки.

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

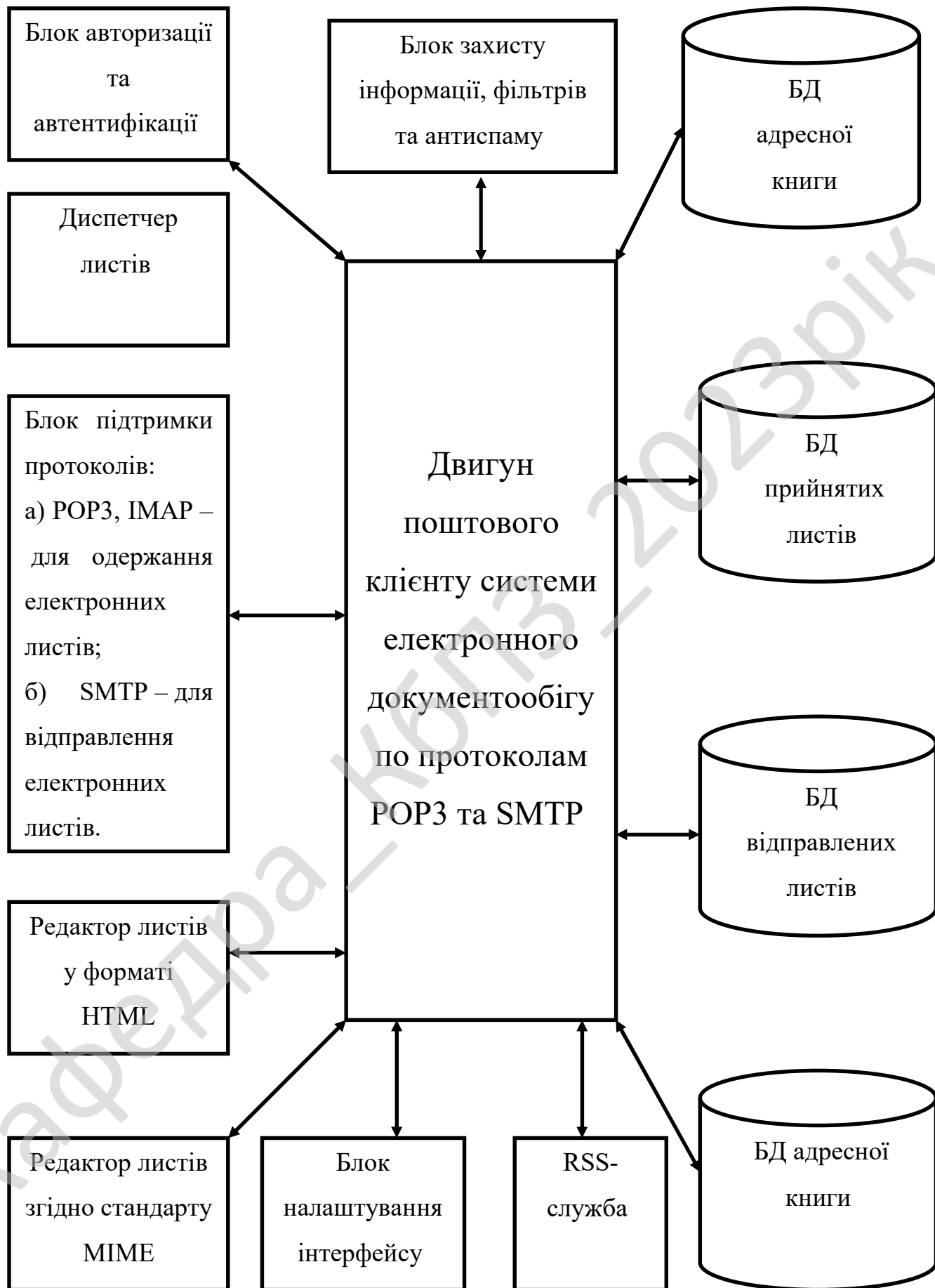


Рисунок 3.1 – Структурна схема розробленої системи

- Блок підтримки протоколів:
 - а) POP3, IMAP – для одержання електронних листів;
 - б) SMTP – для відправлення електронних листів.
- БД адресної книги – призначена для зберігання адрес куди відправлені листи, та звідкіля отримані листи.
- БД прийнятих листів – призначена для зберігання листів надісланих користувачеві поштового клієнта.
- БД відправлених листів – призначена для зберігання листів відправлених користувачем поштового клієнта.
- БД чернеток листів – призначена для зберігання чернеток листів написаних користувачем поштового клієнта;
- Редактор листів у форматі HTML – призначений для написання та редагування текстових листів у вигляді HTML.
- Редактор листів згідно стандарту MIME – призначений для написання та редагування листів у вигляді тексту, картинок, та інших прикріплених до листа файлів.
- Блок налаштування інтерфейсу – призначений для індивідуального налаштування інтерфейсу під конкретного користувача.
- RSS-служба – призначена для реалізації служби новин, як поштового клієнта так і глобальних.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема розробленого поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

Розглянемо більш докладно основні функціональні можливості розробленого поштового клієнту.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

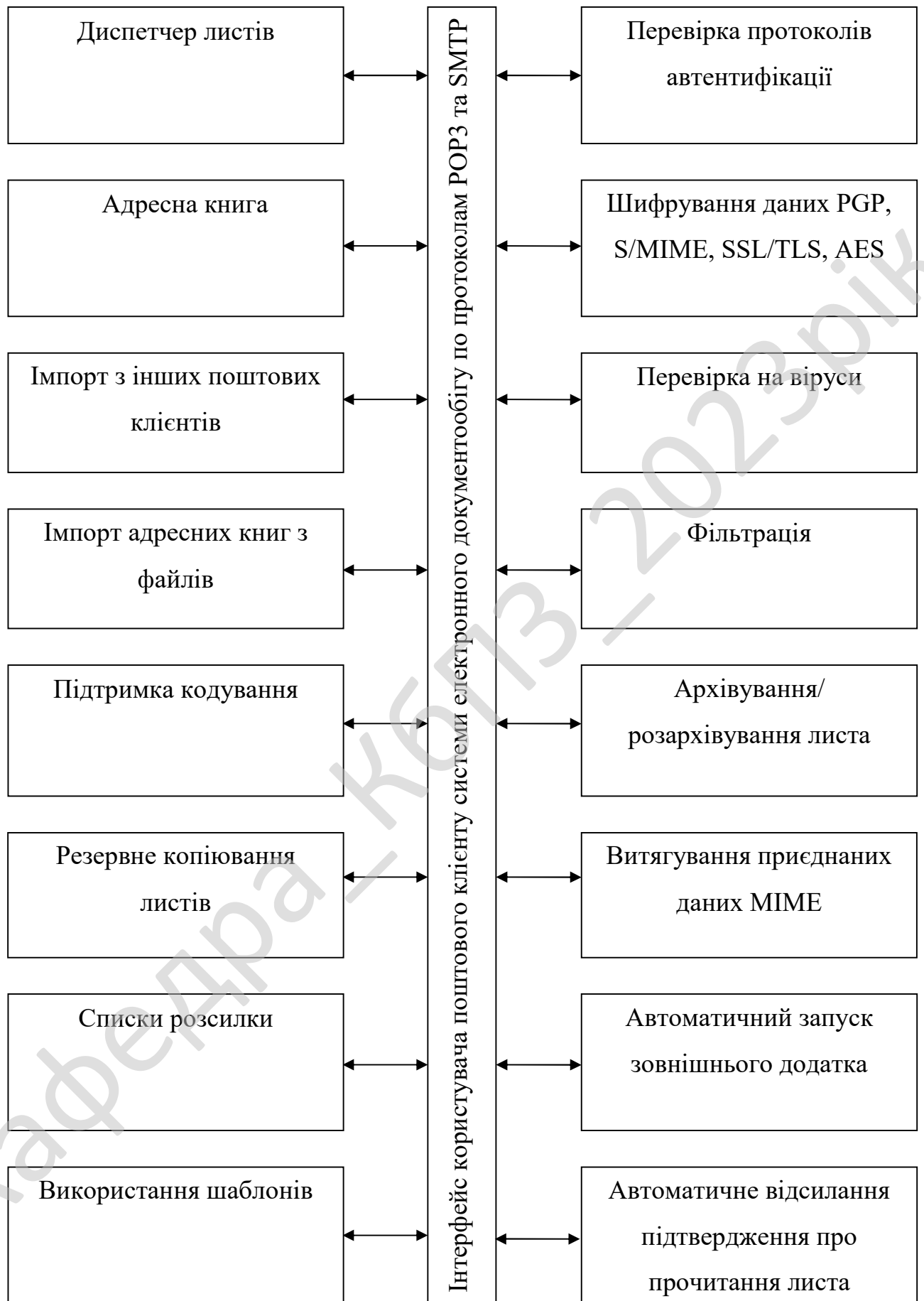


Рисунок 3.2 – Функціональна схема системи

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.

Розглянемо більш докладно вищеперераховані функції.

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 Windows 10/11 повідомляємо, що, на додаток до вищевказаних можливостей,

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

поштовий клієнт задіє такі компоненти Windows 10/11, як механізм Випадкового Розміщення в Адресному Просторі (ASLR) і механізм Запобігання Виконання Даних (DEP) для захисту комп'ютера від вірусів і шкідливих кодів, розповсюджуваних через електронну пошту. У поштовий клієнт вбудований URL менеджер для фонового добування зображень у форматі HTML. Цей менеджер успішно бореться зі шкідливими кодами, що вставляються в зображення.

Фільтрація

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

Автоматична обробка повідомлень

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

Шаблони

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

повний набір макросів, що дозволяє задати будь-яку, навіть саму складну послідовність дій.

Списки розсилання

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

Унікальна функція паркування

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

Резервне копіювання

Всі 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 10/11 – 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 адміністратори (і не тільки вони) по достоїнству оцінять нову можливість “непомітної” установки, що дозволяє інстальювати поштовий клієнт без участі користувача.

Ефективне шифрування

Внутрішнє виконання PGP, засноване на загальновизнаному 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.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Крім того, Поштовий клієнт працює з S/MIME за допомогою Internal Implementation або Microsoft CryptoAPI. Поштовий клієнт – перший поштовий клієнт, що підтримує стиск даних S/MIME. Поштовий 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 поштового клієнта. Синхронізація пошти дозволяє підтримувати дві копії поштового клієнта в однаковому стані. Ви можете вибірково зберігати налаштування поштових скриньок, папок, адресних книг і прикріплених файлів.

Сумісність із Microsoft Exchange

Поштовий клієнт підключається до серверів Microsoft Exchange, задіючи протокол MAPI для виклику або відправлення повідомлень. Необхідно лише встановити Microsoft Office Outlook або клієнт Microsoft Exchange для надання компонентів підключення до Exchange, 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 IMAP сервері, не чекаючи її завантаження. Чудовий засіб для здійснення віддаленого контролю над вашою кореспонденцією. Особливо корисно в тому випадку, якщо необхідно залишити об'ємні листи на сервері, щоб прочитати їх пізніше, або видалити листи, не одержуючи їх.

Адресна книга

Адресна книга містить список ваших адресатів. Ви можете поєднувати їх у групи для класифікації, або для використання адрес як список розсилання, тобто

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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.

Багаті можливості імпорту

Поштовий клієнт дозволяє імпортувати повідомлення з інших поштових клієнтів:

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

Імпорт адресних книг з файлів:

- LDIF-файл.
- Бізнес-Картка VCard.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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

Працює з будь-якою системою Windows.

3.4 Розробка діаграми процесів

На рисунку 3.3 наведена діаграма процесів, які відбуваються в розробленій системі. Процеси покрокового проходження електронної пошти від відправника одержувачеві відбуваються наступним чином:

1. Процес створення листа.
2. Процес з'єднання поштового клієнта з SMTP-сервером.
3. Процес передачі SMTP-серверу інформації про те, кому призначається пошта й хто є відправником.
4. Процес перевірки SMTP-сервером валідності даних про адресата й адресанта й прийняття листа (із заголовками й тілом листа).
5. Процес постановки листа в чергу доставки.
6. Процес DNS запиту про поштові сервери (MX запису) для домену адресата.
7. Процес спроби з'єднання з поштовими серверами, що мають найбільший пріоритет, якщо не вдалося, то спроба передати лист на резервні поштові сервера домену.
8. Процес передачі листа у випадку вдалого з'єднання з поштовим сервером домену адресата, або спроби переслати лист пізніше, у випадку невдачі.
9. Процес прийому поштовим сервером домену адресата листа, передача його модулю, що займається зберіганням листів і видачею їхнім адресатам по протоколу POP3, IMAP або іншим.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

10. Процес з'єднання адресата з POP3 або IMAP сервером, автентифікація й одержання листа адресатом.

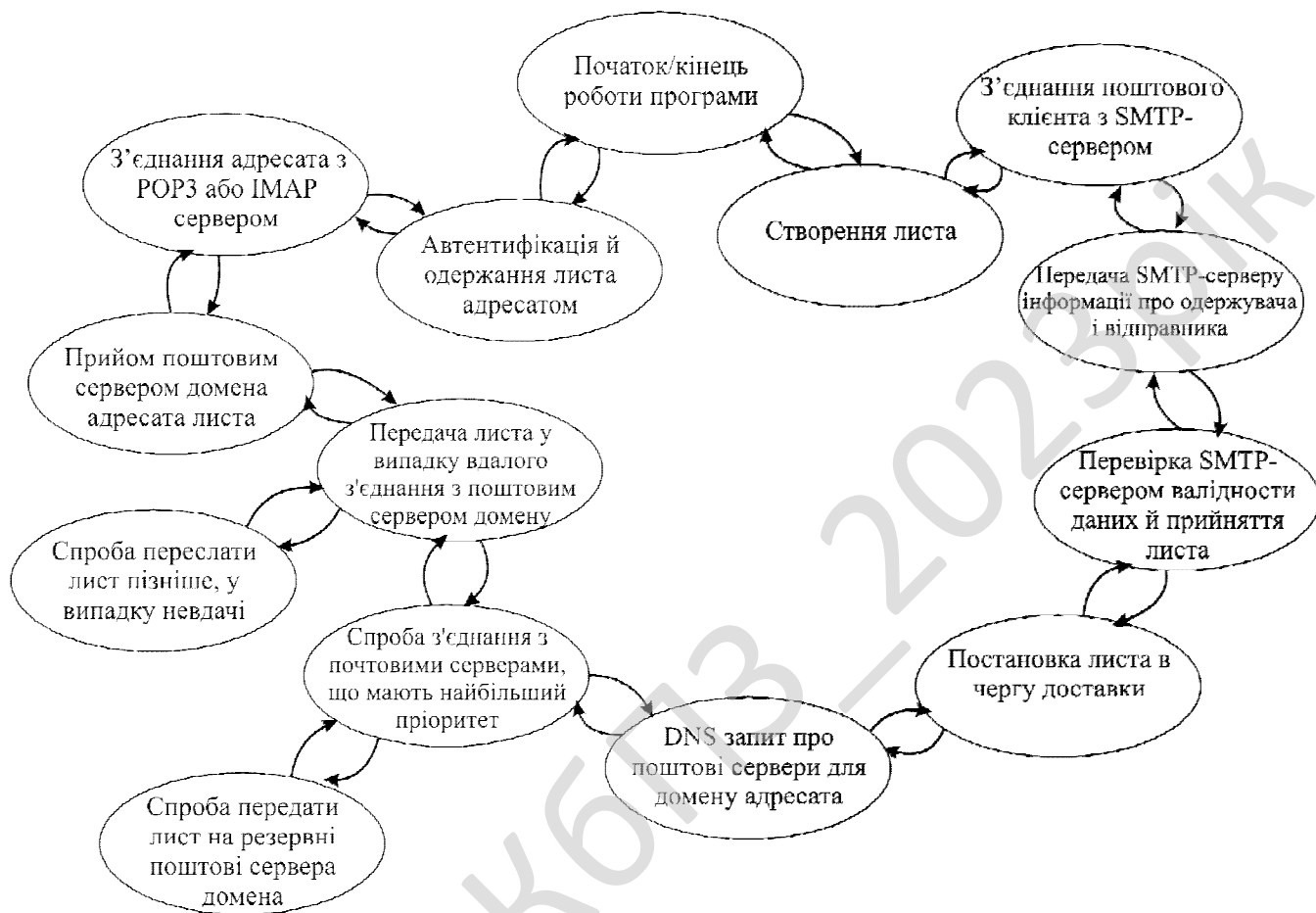


Рисунок 3.3 – Діаграма процесів

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

Розглянемо блок схеми та алгоритми функціонування розробленого програмного забезпечення.

Блок схема основної програми наведена на рисунку 4.1.

З нього ми бачимо, що програма виконує наступні дії.

Спершу відбувається виведення основного вікна програми.

Після цього видається запит на створення нової поштової скриньки.

Якщо вона не існує, то відбувається введення параметрів поштової скриньки.

Після цього поштова скринька створюється.

Якщо ж вона існує, то тоді завантажується вікно пропонування перевірки наявності пошти.

У іншому випадку відбувається перехід до меню відправки листа.

При перевірці наявності пошти, якщо є вхідні листи, то спершу відбувається отримання заголовків листів, після чого, вони виводяться на екран.

Користувач визначає завантажувати листи, або ні.

Якщо треба завантажити листи, то відбувається завантаження отриманих листів на комп'ютер та виведення змісту листа на екран.

Якщо цю дію виконувати не треба, то завантажується меню відправи листа.

Якщо потрібно відправити лист, то відбувається наступна послідовність дій:

– Спершу виводиться вікно створення листа.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

- Після цього відбувається введення адреси отримувача.
- Потім вводиться заголовок та тема листа.
- За цією дією, вводиться текст листа.
- Після чого відбувається відправлення листа.

Якщо ж немає потреби ні в читанні листа, ні в написанні, то існує можливість запуску блоку зміни параметрів програми.

Це дозволяє виконати наступні дії:

- Виведення вікна параметрів.
- Введення нових параметрів.
- Привласнення нових параметрів.

Після виконання усіх вище перерахованих завдань програма або повертається на початок роботи, або закінчує свою роботу, в залежності від вибору користувача.

Розглянемо підпрограму відправки листів, блок-схема якої наведено на рисунку 4.2.

Програма починає працювати зі з'єднання з SMTP-сервером.

Після того, як встановилося з'єднання з SMTP-сервером, йому передається інформація про одержувача та відправника.

Ці дані перевіряються на валідність SMTP-сервером.

Якщо вони не є валідними, тобто не відповідають дійсності, то виводиться повідомлення про помилку, й завантажується вікно повторного введення даних, над якими відбуваються аналогічні дії.

Якщо дані валідні, то відбувається передача листа (із заголовками та тілом листа).

Після цього відбувається постановка листа в чергу доставки.

Потім відбувається DNS запит про поштові сервери для домену адресату.

За цією дією, проводиться спроба з'єднання з поштовими серверами, що мають найбільший пріоритет. Якщо з'єднання не встановлене, то відбувається спроба передати лист на резервні поштові сервери домену.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

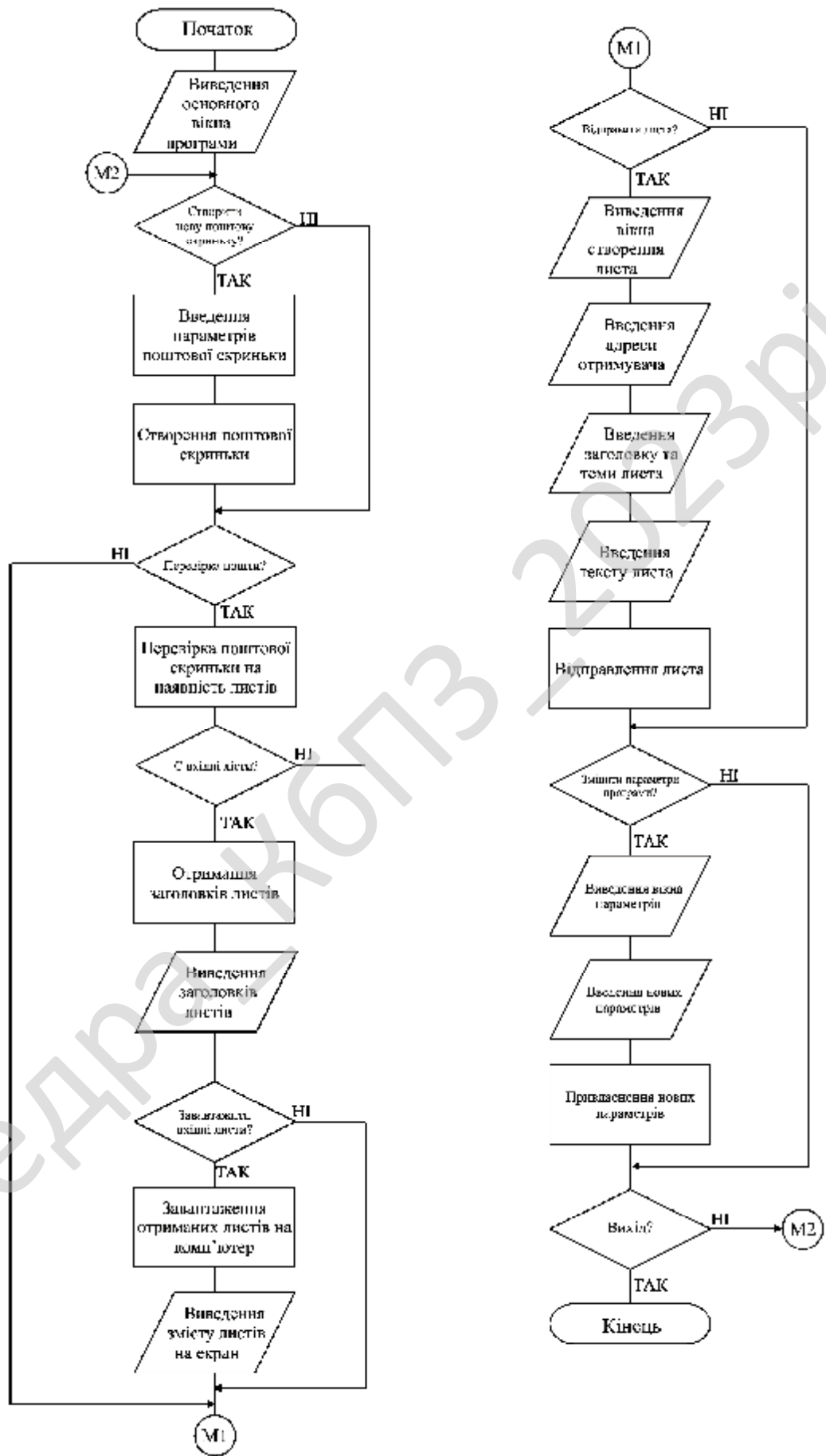


Рисунок 4.1 – Блок-схема роботи основної програми

Якщо з'єднання встановлене, то відбувається передача листа домену адресата.

Після цього якщо з'єднання не встановлене, то лист пересилається пізніше, через вказаний проміжок часу.

Коли лист відіслано, то поштовий сервер домену адресату його приймає.

Після приймання листа поштовим сервером адресату відбувається передача листа модулю, що займається зберіганням листів та видачею їхнім адресатам по протоколу POP3, IMAP або іншим.

При цьому відбувається з'єднання адресата з POP3 або IMAP сервером, під час якої відбувається автентифікація.

Кінцевим пунктом роботи підпрограми відправки листів є одержання листа адресатом.

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

- SMTP.
- POP3.
- IMAP.

Розглянемо ці протоколи більш детально.

SMTP

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

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

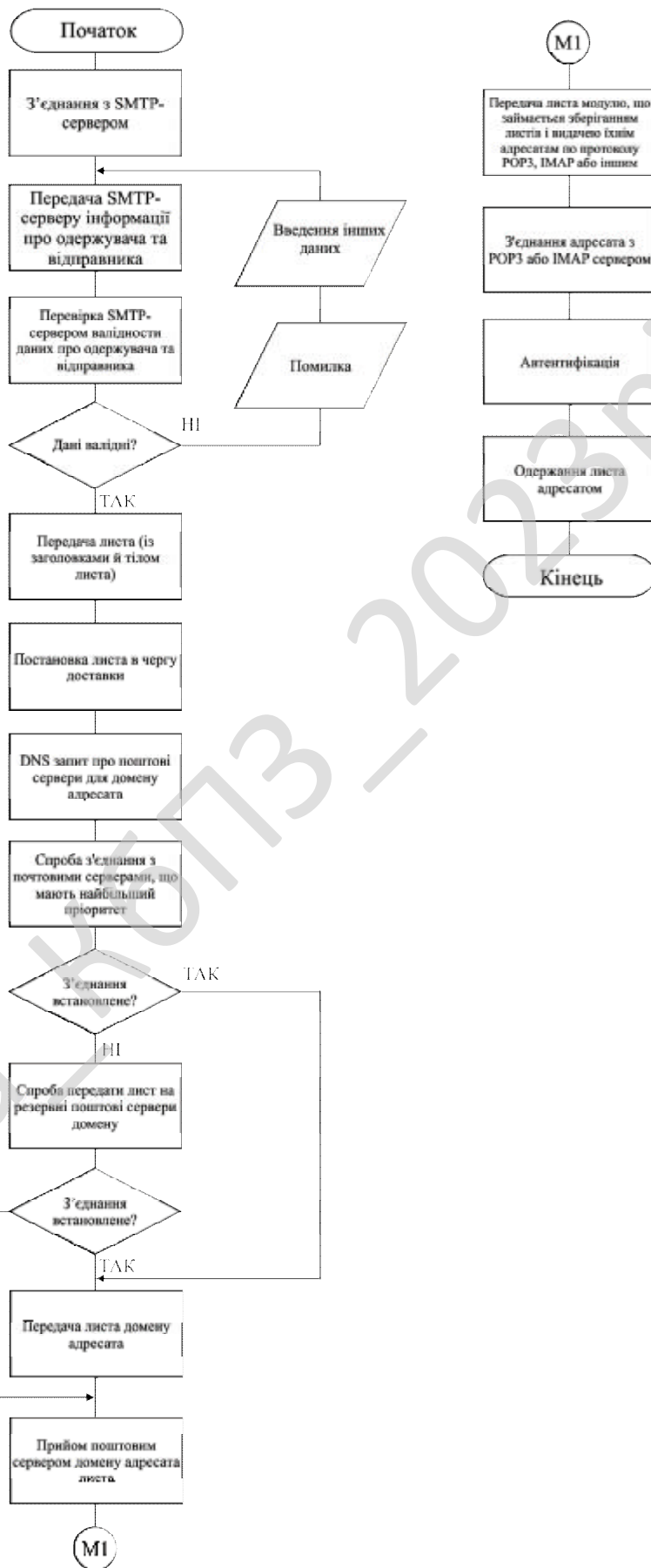


Рисунок 4.2 – Блок-схема роботи підпрограми відправки листів

Огляд протоколу

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

Щоб доставити повідомлення до адресата, необхідно переслати його поштовому серверу домену, у якому перебуває адресат. Для цього звичайно використовується запис типу MX – Mail eXchange – обмін поштою, системи DNS. Якщо MX запис відсутній, то для тих же цілей може бути використаний запис типу A. Деякі сучасні реалізації SMTP-серверів для визначення сервера, що обслуговує пошту в домені адресата, також можуть задіяти SRV-запис (RFC 2782).

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

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

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

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

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

Текстова частина відповіді носить довідковий характер і призначена для людини, а не програми.

ESMTP – розширюваний протокол, на відміну від SMTP. При встановленні з'єднання сервер повідомляє про набір підтримуваних розширень (як відповідь на команду EHLO). Відповідні розширення можуть бути використані клієнтом при роботі. Необхідно пам'ятати, що якщо сесія починається з команди HELO (використовуваної в «класичному» SMTP, RFC 821), то список розширень виводитися не буде.

Безпека SMTP і спам

Споконвічно SMTP не підтримував єдиної схеми авторизації. У результаті цього спам став практично нерозв'язною проблемою, тому що було неможливо визначити, хто насправді є відправником повідомлення – фактично можна відправити лист від імені будь-якої людини. У цей час виробляються спроби вирішити цю проблему за допомогою специфікацій SPF, Sender ID, DKIM. Єдиної специфікації на справжній момент не існує.

Приклад найпростішої SMTP-сесії

C: – клієнт, S: – сервер.

S: (очікує з'єднання)

C: (Підключається до порту 25 сервера)

S:220 mail.company.tld ESMTP CommuniGate Pro 5.1.4i is glad to see you!

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

C:HELO
 S:250 domain name should be qualified
 C:MAIL FROM: <someusername@somecompany.ru>
 S:250 someusername@somecompany.ru sender accepted
 C:RCPT TO:<user1@company.tld>
 S:250 user1@company.tld ok
 C:RCPT TO: <user2@company.tld>
 S:550 user2@company.tld unknown user account
 C:DATA
 S:354 Enter mail, end with "." on a line by itself
 C:from: someusername@somecompany.ru //щоб лист
 C:to: user1@company.tld //не було додано
 C:subject: тема //у категорію спам
 C: //
 C:Hi!
 C:.
 S:250 769947 message accepted for delivery
 C:QUIT
 S:221 mail.company.tld CommuniGate Pro SMTP closing connection
 S: (закриває з'єднання)

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

Розширення ESMTP

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

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

```

ehlo office.company1.tld
250-mail.company2.tld is pleased to meet you
250-DSN
250-SIZE
250-STARTTLS
250-AUTH LOGIN PLAIN CRAM-MD5 DIGEST-MD5 GSSAPI MSN NTLM
250-ETRN
250-TURN
  
```

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

250-ATRN
250-NO-SOLICITING
250-HELP
250-PIPELINING
250 EHLO

POP3

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

Попередні версії протоколу (POP, POP2) застаріли.

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

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

Альтернативним протоколом для збору повідомлень із поштового сервера є IMAP.

Стани сеансу

У протоколі POP3 передбачено 3 стану сеансу:

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

Команди протоколу

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

Аргументи:

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Обмеження:

Її підтримка не є обов'язковою.

Можливі відповіді:

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

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

Аргументи:

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

Обмеження:

Немає.

Можливі відповіді:

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

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

Аргументи:

[пароль] – пароль для поштової скриньки.

Обмеження:

Працює після успішної передачі ім'я поштової скриньки.

Можливі відповіді:

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

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

Аргументи:

[повідомлення] – номер повідомлення.

Обмеження:

Доступна після успішної ідентифікації.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Можливі відповіді:

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

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

Аргументи:

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

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

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

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

Аргументи:

Немає.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

- +OK.

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

Аргументи:

повідомлення – номер повідомлення.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

8. RSET – Цією командою виробляється відкрит транзакцій усередині сесії. Наприклад, якщо користувач випадково позначив на видалення які-небудь повідомлення, він може забрати ці позначки, відправивши цю команду.

Аргументи:

Немає.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

– +OK.

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

Аргументи:

Немає.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

– +OK a b.

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

Аргументи:

– [повідомлення] – номер повідомлення.

– [кількість рядків] – скільки рядків потрібно вивести.

Обмеження:

Доступна після успішної ідентифікації.

Можливі відповіді:

– +OK n octets.

– -ERR no such message.

11. QUIT

Аргументи:

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Немає.

Обмеження:

Немає.

Можливі відповіді:

- +OK

Приклад сесії

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

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

Варіант початку сесії, при якому пароль передається відкритим текстом:

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

S +OK Pass accepted

IMAP

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

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

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

Команди протоколу

Доступні в будь-якому стані:

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

Доступні в неавтентифікованому стані:

- STARTTLS.
- AUTHENTICATE.
- LOGIN.

Доступні в автентифікованому стані:

- SELECT.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

- EXAMINE – аналогічна SELECT, але доступ буде даний тільки для читання.
- CREATE – створити поштову скриньку.
- DELETE – видалити поштову скриньку.
- RENAME.
- SUBSCRIBE.
- UNSUBSCRIBE.
- LIST.
- LSUB.
- STATUS – інформація про поштову скриньку, у тому числі кількість непрочитаних повідомлень.

- APPEND – додати повідомлення в поштову скриньку.

Доступні в стані selected:

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

Експериментальні/розширення:

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

Слід зазначити, що в IMAP немає команди на переміщення повідомлення з одного ящика в іншій. Замість цього, потрібно зробити COPY, а потім відзначити повідомлення на видалення. Деякі IMAP-сервери надають команди з розділу розширених, для переміщення повідомлення: наприклад, сервер GroupWise підтримує команду XGWMOVE¹.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Переваги в порівнянні з POP3

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

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

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

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Threefish – в криптографії симетричний блоковий криптоалгоритм, розроблений автором Blowfish та Twofish, американським криптографом Брюсом

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Шнайером 2008 року для використання в хеш-функції Skein і як універсальну заміну наявним блоковим шифрам. Основними принципами розробки шифру були: мінімальне використання пам'яті, необхідна для використання в хеш-функції стійкість до атак, простота реалізації та оптимізація під 64-розрядні процесори.

Структура алгоритму

Threefish має дуже просту структуру і може бути використаний для заміни алгоритмів блочного шифрування, будучи швидким і гнучким шифром, що працюють в довільному режимі шифрування. Threefish S-блоки не використовує, заснований на комбінації інструкцій виключаючого або, складання і циклічного зсуву.

Як і AES, шифр реалізований у вигляді підстановочно-перестановочної мережі на оборотних операціях, не будучи шифром мережі Фейстел.

Алгоритм передбачає використання tweak-значення, свого роду вектора ініціалізації, дозволяючи змінювати таким чином значення виходу, без зміни ключа, що має позитивний ефект як для реалізації нових режимів шифрування, так і на криптостійкості алгоритму.

Як результат думки авторів, що кілька складних раундів часто гірше ніж застосування великого числа простих раундів, алгоритм має нетрадиційно велику кількість раундів – 72 або 80 при ключі 1024 біт, проте, за заявою творців, його швидкісні характеристики випереджають AES приблизно вдвічі. Варто зауважити, що через 64-бітної структури шифру, дана заява має місцево лише на 64-розрядній архітектурі. Тому, Threefish, як і Skein [1], заснований на ньому, на 32-розрядних процесорах показує значно гірші результати ніж на «рідному» обладнанні.

Ядром шифру є проста функція «MIX», перетворювальна два 64-бітових беззнакових числа, в процесі якої відбувається складання, циклічний зсув (ROL / ROR), і додавання по модулю 2 (XOR).

Нижче представлений код MIX-функції для Threefish-1024 [2]:

```
<syntaxhighlight lang="C">  
// Константи для циклічного зсуву  
int R16 [8] [8] = {  
    {55, 43, 37, 40, 16, 22, 38, 12},  
    {25, 25, 46, 13, 14, 13, 52, 57},  
    {33, 8, 18, 57, 21, 12, 32, 54},  
    {34, 43, 25, 60, 44, 9, 59, 34},  
    {28, 7, 47, 48, 51, 9, 35, 41},  
    {17, 6, 18, 25, 43, 42, 40, 15},  
    {58, 7, 32, 45, 19, 18, 2, 56},  
    {47, 49, 27, 58, 37, 48, 53, 56},  
};  
// D – раунд, j – індекс в таблиці циклічного зсуву  
void mix (int j, int d) {  
    unsigned long long rotl;  
    y [0] = x [0] + x [1];  
    rotl = R16 [d% 8] [j];  
    y [1] = (x [1] << rotl) | (x [1] >> (64 – rotl));  
    y [1] ^ = y [0];  
}  
</Source>
```

Процедура розшифрування обернена процедурі зашифрування і містить зворотну функцію DEMIX.

Кожен з 72 раундів Threefish-256 і Threefish-512 має чотири MIX перетворення, Threefish-1024 – вісім звернень до MIX функції.

Безпека

За заявою авторів, алгоритм має більш високий рівень безпеки, ніж AES. Існує атака на 25 з 72 раундів Threefish, в той час як для AES – на 6 з 10. Threefish має показник фактора безпеки 2.9, в свою чергу, AES всього 1.7 [3]

Для досягнення повної дифузії, шифру Threefish-256 досить 9 раундів, Threefish-512 – 10 раундів і Threefish-1024 – 11 раундів. Виходячи з цього, 72 і 80 раундів відповідно в середньому, забезпечать кращі результати, ніж існуючі шифри. [4]

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

У той же час, алгоритм має набагато простішу структуру і функцію перетворення, проте виконання 72-80 раундів, на думку дослідників, забезпечує необхідну стійкість. Вживаний розмір ключа від 256 до 1024 біт зводить нанівець можливість повного перебору паролів при так званій атаці грубою силою (brute force attack) на сучасному обладнанні.

Кафедра _ КБПЗ _ 2023 рік

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення реалізує поштовий клієнт електронного документообігу по протоколам POP3 та SMTP.

Програмно-апаратні вимоги:

- Загальний обсяг ОЗП: 512 Мбайт.
- Вільний простір на жорсткому диску: 10 Мбайт.
- Операційна система Microsoft Windows 10/11.
- Підключення до мережі Інтернет.

Головне вікно програми зображене на рисунку 5.1.

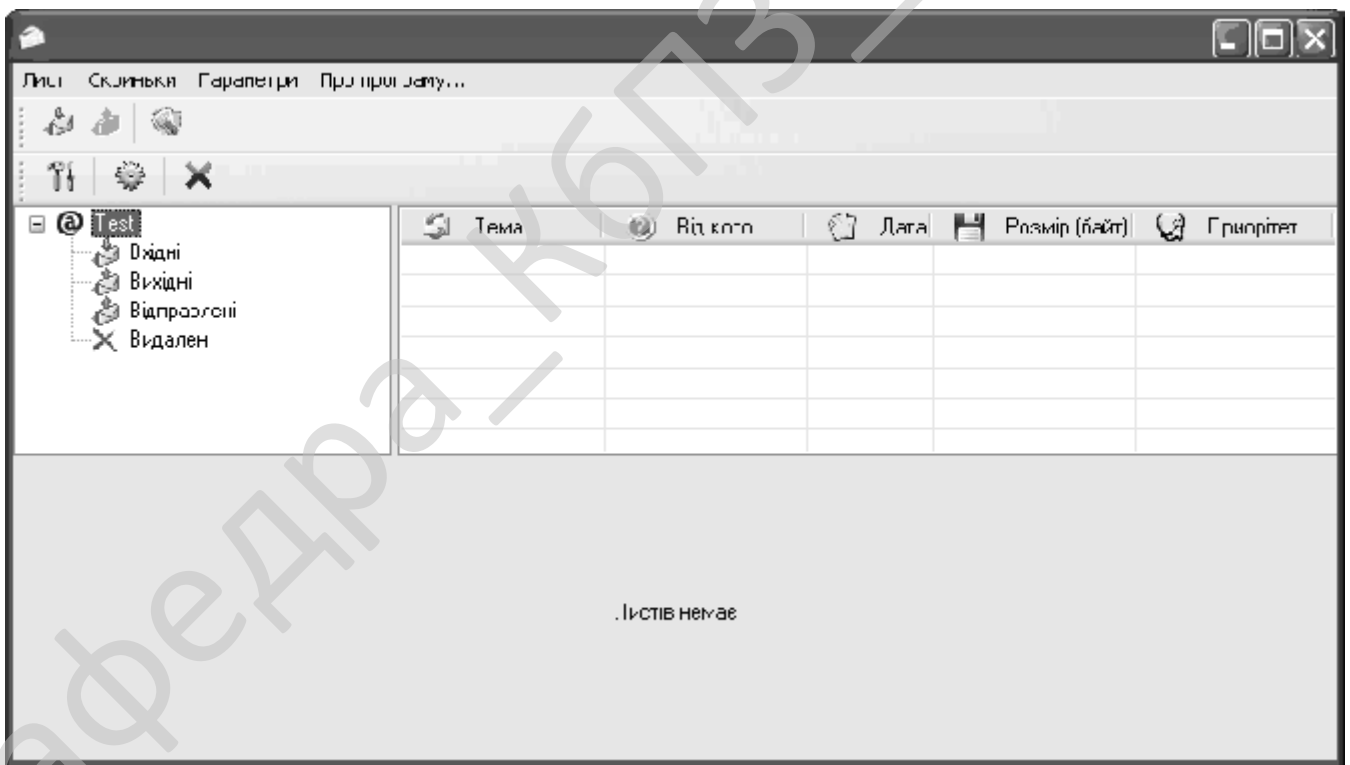


Рисунок 5.1 – Головне вікно програми

Поштовий клієнт зберігає всю кореспонденцію у віртуальних папках, прив'язаних до облікових даних. Папки мають деревоподібну структуру, корінь якої – обліковий запис користувача. Деревоподібна структура папок дозволяє найкращим способом використовувати поштовий клієнт для датамайнінга.

У клієнті присутні 4 папки, у тому числі:

1. Вхідні – для прийому кореспонденції.
2. Вихідні – для посланої Вами, але ще не обробленої кореспонденції.
3. Відправлені – для відправленої й обробленої кореспонденції.
4. Вилучені – для вилучених листів, які перебувають у цій папці, на

випадок якщо буде необхідність все-таки відновити їх.

Для прийому поштової кореспонденції із всіх підключених до поштового клієнта поштових скриньок необхідно:

1. Відкрити поштовий клієнт.
2. Вибрати пункт меню Скриньки->Перевірити пошту.
3. З'явиться діалогове вікно обробки облікових записів пошти. Дочекатися завершення їхньої обробки.
4. У результаті роботи клієнта вся кореспонденція з поштової скриньки буде поміщена в папку «Вхідні» поштового клієнта.

Для відправлення кореспонденції необхідно:

1. Відкрити поштовий клієнт.
2. Вибрати пункт меню Лист->Написати листа.
3. Відкриється порожнє вікно повідомлень. Використовуючи убудований текстовий редактор, наберіть текст повідомлення (рисунок 5.2).
4. Прикріпіть, якщо це необхідно, файл вкладення. Для цього виберіть мишею кнопку зі скріпкою, знайдіть потрібний файл, виберіть його й натисніть кнопку «Вставити».
5. У поле «Тема» введіть основні відомості про зміст листа й про вкладення в лист, які будуть відображатися в заголовку повідомлення.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

6. Введіть адреси електронної пошти адресатів. Заповнюються поля «Кому» і «Копія».

7. Після заповнення всіх полів виберіть кнопку «Відправити». Лист буде переслано в папку «Вихідні».

8. Дочекайтеся відправлення повідомлення. Для прискорення відправлення виберіть кнопку «Відправити/Одержати».



Рисунок 5.2 – Відправка листа

Крім сховища кореспонденції, поштовий клієнт також є сховищем облікових записів електронної пошти. Поштовий клієнт дозволяє збирати кореспонденцію відразу із всіх поштових скриньок, що дуже зручно для користувача.

Обліковий запис електронної пошти – це набір необхідних відомостей, необхідних для підключення зовнішніх програм (поштовий клієнт, антивірусний і

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

антиспамовий модуль і т.п.) до сервера електронної пошти для доступу до поштової скриньки користувача.

Слід зазначити, що облікові записи – це набагато більше, ніж адреса електронної пошти користувача (рисунок 5.3). В облікових записах також утримується:

- протокол доступу до поштової скриньки;
- облікове ім'я й пароль для доступу до ящика;
- адреси серверів електронної пошти;
- і інша необхідна інформація.

Види облікових записів указують, яким образом повинна виробляється завантаження в поштовий клієнт кореспонденції користувача із сервера електронної пошти. Цей обмін може відбуватися по наступних протоколах:

- IMAP. Протокол синхронізації кореспонденції на сервері електронної пошти з усіма папками поштового клієнта.
- POP3. Протокол завантаження (не синхронізації!) кореспонденції із сервера електронної пошти в папку «Вхідні» (“Inbox”) поштового клієнта. Має безліч налаштувань.
- SMTP. Протокол відправлення електронної пошти.
- UUCP. Застарілий протокол зв'язку комп'ютерів. Може використовуватися для одержання й відправлення електронних листів.
- Захищені POP3, SMTP і ін. Ці протоколи перед одержанням або відправленням кореспонденції шифрує її.

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

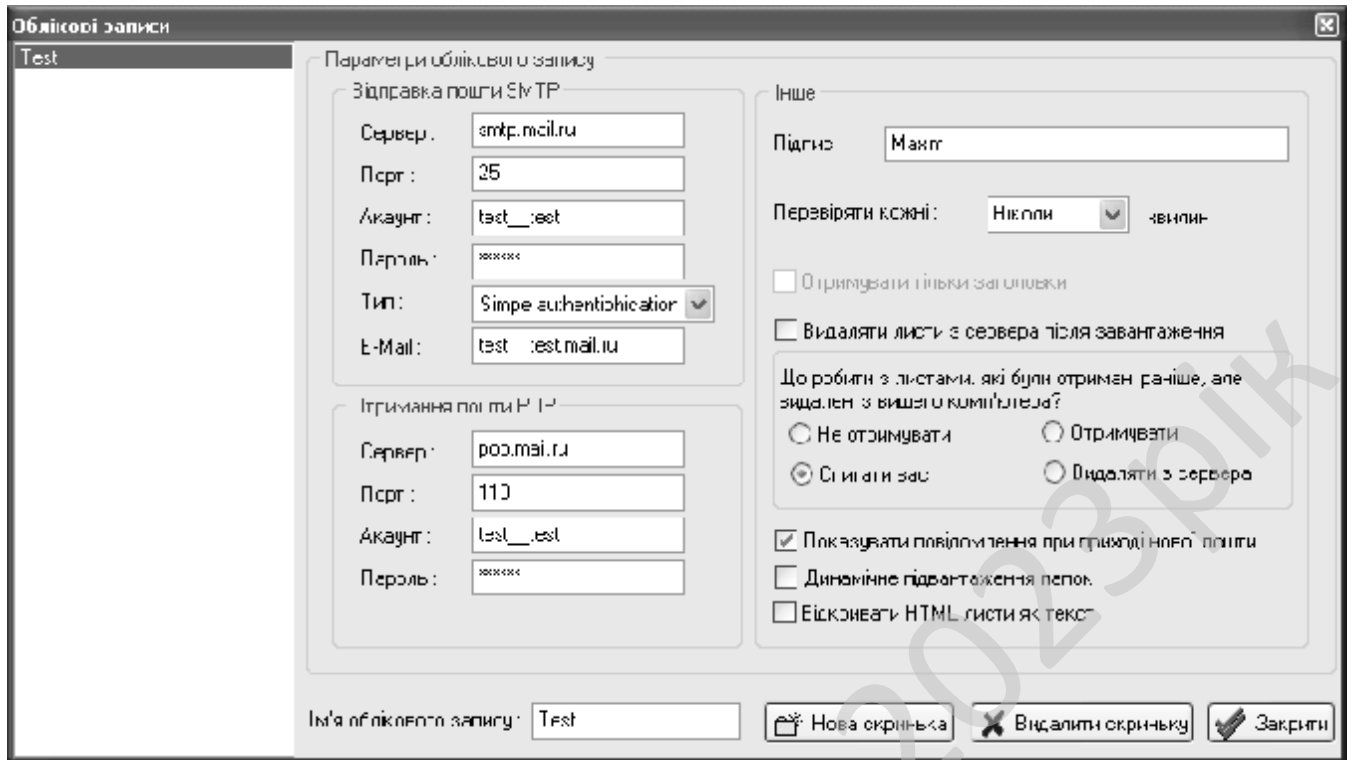


Рисунок 5.3 – Параметри поштової скриньки

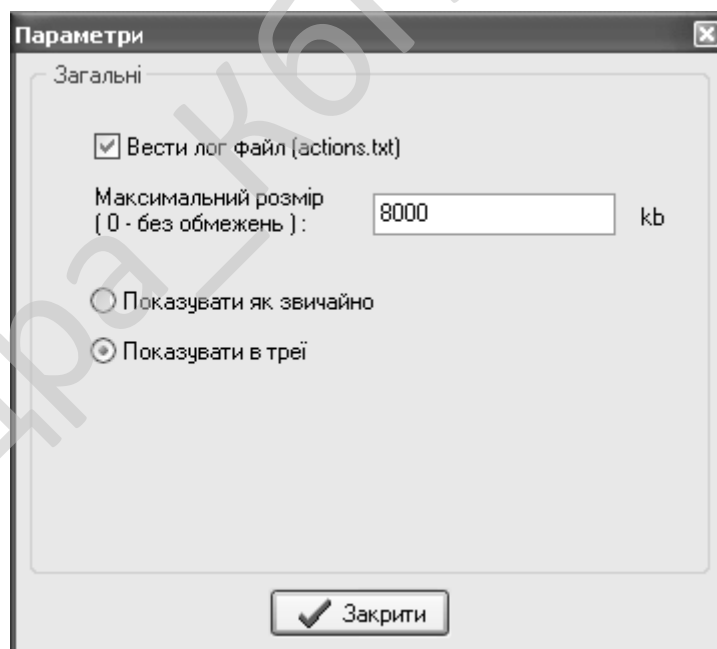


Рисунок 5.4 – Параметри програми

Коротку довідку про розроблену програму можна переглянути натиснувши в меню користувача пункт "Про програму...", після чого з'явиться вікно зображене на рисунку 5.9.

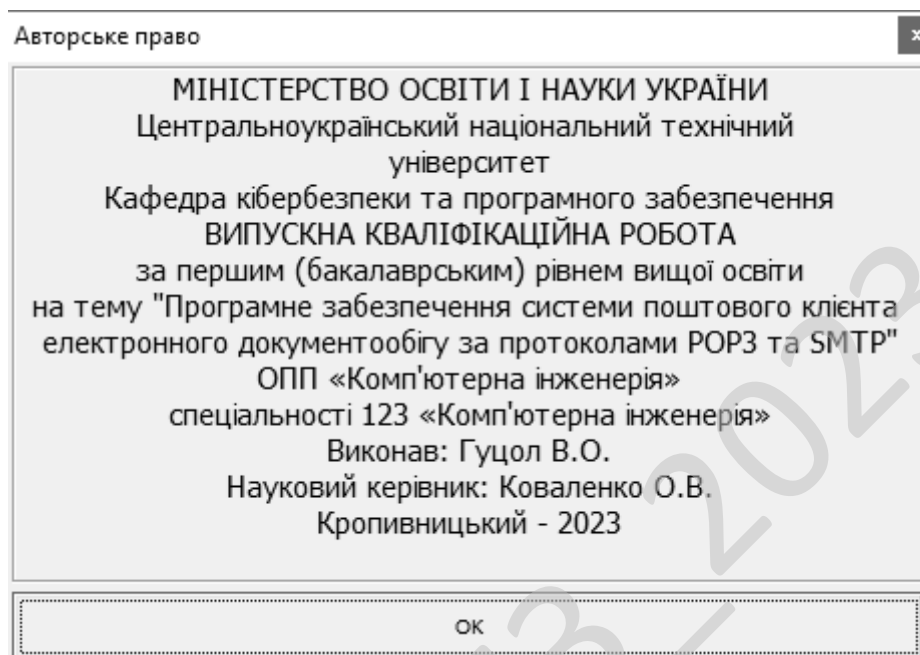


Рисунок 5.9 – Вікно «Про програму...»

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

– Досліджена система поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

– На основі отриманих результатів досліджень створена програмна реалізація системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Threefish.

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

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kovalenko O., Popereshnyak S., Grinenko S., Grinenko O., Radivilova T. «Methods for Assessing the Maturity Levels of Software Ecosystems». *CEUR Workshop Proceedings* Volume 2654, 2019, Pages 251-261. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85083214331&origin=AuthorNamesList&txGid=5633fba897776a6e0f3d5633fbc3d3fbc> (Scopus).

2. Kovalenko O., Drieieva H., Simakhin V., Bondar S., Drieiev O., Zhumadilova M. «Multifractal Properties of Traffic Generator Based on Markov Chains ». *CEUR Workshop Proceedings* Volume 2588, 2019, Pages 567-579. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85083214331&origin=AuthorNamesList&txGid=176e2cada8976a6e0f3d5633fbc3d3fbc> (Scopus).

3. Kovalenko Oleksandr Qualitative risk analysis of software development / Oleksandr Kovalenko, Jamil Al-Azzeh, Oleksii Smirnov, Anna Kovalenko, Serhii Smirnov // *Asian Journal of Information Technology*. – Volume 17 Issue 3. – Medwell Journals. – 2018. – P. 218-230. ISSN: 1682-3915. URL: <http://medwelljournals.com/abstract/?doi=ajit.2018.218.230> Doi: ajit.2018.218.230

4. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A.Kovalenko, S. Smirnov, V. Vialkova // *Scientific & practical cyber security journal (SPCSJ)* Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667. URL: <https://journal.scsa.ge/wp-content/uploads/2018/12/04-3-o.kovalenko-a.kovalenko-o.smirnov-s.smirnov-v.vialkova.pdf>

5. Коваленко А.В. Технология тестирования DOM XSS уязвимости / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // *Scientific & practical cyber security journal (SPCSJ)* Volume 1. Issue 1. P. 38-45 Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

4667. URL: <https://journal.scsa.ge/wp-content/uploads/2018/12/8-dom-xss-testing-technology-vulnerabilities.pdf>

6. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 305 с.

7. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Издавец Рожко С.Г., 2016. – 566 с.

8. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Издавец Рожко С.Г., 2017. – 447 с.

9. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.

10. Коваленко А.В. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 4(120). – Х.: ХУПС – 2014. – С. 161-164.

11. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.

12. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". –

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.

13. Коваленко А.В. Метод качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.

14. Коваленко А.В. Метод количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 128-133.

15. Коваленко А.В. Использование псевдобулевых методов бивалентного программирования для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

16. Коваленко А.В. Метод управления рисками разработки программного обеспечения / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 2 (38). – Полтава: ПолтНТУ. – 2016. – С. 93-100.

17. Коваленко А.В. Технология тестирования уязвимости к SQL инъекциям / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 5 (45). – Полтава: ПолтНТУ. – 2017. – С. 66-71.

18. Коваленко А.В. Масштабирование имитационной модели технологии тестирования безопасности / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 6 (46). – Полтава: ПолтНТУ. – 2017. – С. 181-184.

19. Коваленко А.В. Имитационная модель технологии тестирования безопасности Web-приложений / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 1 (47). – Полтава: ПолтНТУ. – 2018. – С. 114-123.

20. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розроблення програмного забезпечення/ О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – С. 116-125.

21. Коваленко О.В. Управління ризиками розроблення програмного

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

забезпечення за умови обмеженості коштів виділених на усунення помилок безпеки/ О.В. Коваленко // Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Випуск 31. – Кропивницький: ЦНТУ. – 2018. – С. 128-140.

22. Коваленко О.В. GERT-мережевий синтез технології тестування на вразливість WEB-додатків/ О.В. Коваленко // Захист інформації. – Випуск 20(2) – К.: НАУ. – 2018. – С. 89-94.

23. Коваленко О.В. Імітаційна модель технології тестування безпеки на основі положень теорії масштабування / О.В. Коваленко // Безпека інформації. – Випуск 24 (2). – К.: НАУ. – 2018. – С. 110-117.

24. Коваленко О.В. Оцінка ефективності технології тестування безпеки / О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 2, 2018. – С. 137-141

25. Коваленко О.В. Методи та засоби управління безпекою додатків / О.В. Коваленко // Інформаційно-керуючі системи на залізничному транспорті. №4, 2018. – С. 41-44.

26. Коваленко О.В. Розробка інформаційної технології передтестової компіляції та розподілу доступу / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 4 (50). – Полтава: ПолтНТУ. – 2018. – С. 115-119.

27. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення/ О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – С. 131-137.

28. Коваленко О.В. Удосконалений метод управління ризиками розроблення програмного забезпечення на основі напівмарковської моделі прийняття рішень/ О.В. Коваленко // Сучасні інформаційні системи. – Випуск 2(3). – Харків. – 2018. – С. 41-48.

29. Коваленко О.В. Математичні моделі технології тестування DOM XSS

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

вразливості та вразливості до SQL ін'єкцій / О.В. Коваленко // Вісник Черкаського державного технологічного університету. Серія : Технічні науки №4, 2018. – С. 29-36.

30. Коваленко О.В. Математична модель технології тестування вразливості до SQL ін'єкцій / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 6 (58). – Полтава: ПолтНТУ. – 2019. – С. 43-47.

31. Коваленко О.В. Математична модель технології тестування комплексу DOM XSS вразливостей для аналітичної оцінки часових витрат / О.В. Коваленко // Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 173-180, 2019.

32. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць II міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 24-27 лютого 2016 р. – Київ: Європейський університет. – 2016. – С. 138-139.

33. Коваленко А.В. Анализ и оценка рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез «Securitea internationala 2015-2016». Conferenta internationala (editia a XII-a). Chisinau. Moldova. 3 martie 2016. – Chisinau: ADSEM. – 2016. – Р. 96-102.

34. Коваленко А.В. Исследование источников и причин риска разработки программного обеспечения, этапов и работ, при выполнении которых возникает риск / А.В. Коваленко, А.А. Смирнов // Збірник тез VII всеукраїнської науково-практичної конференції "Інформатика та системні науки (ІСН-2016)". м. Полтава. 10-12 березня 2016 р. – Полтава.: ПУЕТ – 2016. – С. 264-266.

35. Коваленко А.В. Оценка показателя чистой приведенной стоимости для количественной оценки рисков проекта разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез науково-практичної конференції “Проблеми кібербезпеки інформаційно-телекомунікаційних

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

систем”. м. Київ. 10-11 березня 2016 р. – Київ: КНУ ім. Тараса Шевченко – 2016. – С. 81-82.

36. Коваленко А.В. Методика структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 24-25 березня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 71-72.

37. Коваленко А.В. Методы качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // 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. – Харків: НТУ «ХПІ». – 2016. – С. 6-7.

38. Коваленко А.В. Структурная идентификация рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 15-16 квітня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 175-182.

39. Коваленко А.В. Исследование разработанной методики структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез VIII міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 28-29 квітня 2016 р. – Харків: ХНЕУ. – 2016. – С. 49.

40. Коваленко А.В. Исследование дерева рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез III міжнародної науково-практичної конференції «Інформаційна та економічна безпека» (INFECO-2016)». м. Харків. 28-30 квітня 2016 р. – Харків: ХННІ ДВНЗ «УБС». – 2016. – С. 174-178.

41. Коваленко А.В. Методы качественного анализа и количественной

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Сборник тезисов XII международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 30 мая – 02 июня 2016 г – Варна. ТУВ. – 2016. – С. 585-589.

42. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Матеріали Всеукраїнської науково-практичної конференції «Кібербезпека в Україні: правові та організаційні питання». м. Одеса, 21 жовтня 2016 р. – Одеса : ОДУВС, 2016. – С.146-148.

43. Коваленко А.В. Метод управления рисками разработки программного обеспечения с использованием псевдобулевых методов бивалентного программирования / А.В. Коваленко, А.А. Смирнов // Матеріали Всеукраїнської науково-практичної конференції «Актуальні задачі та досягнення у галузі кібербезпеки». м. Кропивницький, 23-25 листопада 2016 року – Кропивницький: ЦНТУ, 2016. – С. 162.

44. Коваленко А.В. Псевдобулевые методы бивалентного программирования для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко, С.А. Смирнов // Збірник наукових праць III міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 22-25 лютого 2017 р. – Київ: Європейський університет. – 2017. – С. 158-162.

45. Коваленко А.В. Метод управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез II науково-практичної конференції “Проблеми кібербезпеки інформаційно-телекомунікаційних систем”. м. Київ. 23-24 березня 2017 р. – Київ: КНУ ім. Тараса Шевченко – 2017. – С. 203-205.

46. Коваленко А.В. Алгоритмы анализа уязвимостей при управлении рисками разработки программного обеспечения / А.В. Коваленко,

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

А.А. Смирнов, А.С. Коваленко // Conferenta internationala (editia a XIII-a). «Securitea informationala 2017». Chisinau. Republic of Moldova. 4-5 aprilie 2017. – Chisinau: ADSEM. – 2017. – P. 19-22.

47. Коваленко А.В. Алгоритм анализа DOM XSS уязвимости при управлении рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез дев'ятнадцятого міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кропивницький 7-8 квітня 2017 р. – Кропивницький: ГЛА НАУ. – 2017. – С. 125-127.

48. Коваленко А.В. Алгоритм анализа уязвимости SQL Injection для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез другої міжнародної науково-технічної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2017). м. Харків. 10-12 квітня 2017 р. – Харків: НТУ «ХП». – 2017. – С. 27.

49. Коваленко А.В. Метод управления рисками разработки программного обеспечения на основе алгоритмов анализа уязвимостей / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 20-22 квітня 2017 р. Кіровоград: КНТУ. – 2017. – С. 92.

50. Коваленко А.В. Алгоритмы анализа DOM XSS уязвимости и уязвимости SQL Injection при управлении рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез ІХ міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 20-21 квітня 2017 р. – Харків: ХНЕУ. – 2017. – С. 61.

51. Kovalenko O.V. Method of testing the dom xss vulnerability / Kovalenko Oleksandr, Kovalenko Anna, Smirnov Oleksii, Smirnov Serhii // International

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

Conference «information technologies, systems and networks ITSN-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. – 2017. – P. 7.

52. Коваленко О.В. Метод тестування DOM XSS уразливості / О.В. Коваленко, О.А. Смірнов, А.С. Коваленко, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

53. Коваленко О.В. GERT-модель технології тестування DOM XSS уразливості / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник наукових праць IV міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 21-24 лютого 2018 р. – Київ: Європейський університет. – 2018. – С. 65-70.

54. Коваленко О.В. Технології тестування уразливостей Web-застосунків з використанням GERT-моделі / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної конференції "Комп'ютерні інтелектуальні системи та мережі (КІСМ-2018)". м. Кривий Ріг. 21-23 березня 2018 р. – Кривий Ріг.: ДВНЗ КНУ – 2018. – С. 227-230.

55. Коваленко А.В. Тестирование уязвимости Web-приложений к атаке вида межсайтовый скриптинг / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез «Securitea internationala 2018». Conferenta internationala (editia a XIV-a). Chisinau. Moldova. 20-21 martie 2018. – Chisinau: ADSEM. – 2018. – P. 54-56.

56. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез X міжнародної науково-практичної

конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 19-20 квітня 2018 р. – Харків: ХНЕУ. – 2018. – С. 38.

57. Коваленко О.В. Розробка методу передтестової компіляції й розподілу доступу / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницькій. 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215.

58. Коваленко О.В. Оцінка ефективності технологій тестування безпеки уразливостей DOM XSS й SQL-ін’єкцій / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Сборник тезисов XIV международной конференции "Стратегия качества в промышленности и образовании", Варна, Болгария. 04-07 июня 2018 г – Варна. ТУВ. – 2018. – С. 271-274.

59. Коваленко О.В. Аналіз основних підходів математичного моделювання та методологій для забезпечення максимальних показників безпеки програмного забезпечення / О.В. Коваленко, А.С. Коваленко // Збірник наукових праць всеукр. наук.-практ. конф. здобувачів вищої освіти й молодих учених «Комп’ютерна інженерія і кібербезпека : досягнення та інновації», м. Кропивницькій. 27-29 листопада

					ВКРБ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРБ-123.23.0014.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Гуцол В.О.				Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.						
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20-3СК		
Затв.	Смірнов О.А.						
					Програмне забезпечення системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP		
					Б	1	6

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 8-02 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 поштового клієнта електронного документообігу за протоколами POP3 та SMTP.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-123.23.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- системи поштового клієнта електронного документообігу за протоколами POP3 та SMTP;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРБ-123.23.0014.00.00.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-123.23.0014.00.00.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 – 84 аркуші.

8 Етапи розробки

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

					ВКРБ-123.23.0014.00.00.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 16.06.2023 р.

					ВКРБ-123.23.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Коваленко О.В.

*Програмне забезпечення системи поштового клієнта електронного
документообігу за протоколами POP3 та SMTP*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2023 року

```
program MyMail;

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

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Поштовий клієнт, розроблений Гуцол В.О.';
  Application.CreateForm(Tfrm_Main, frm_Main);
  Application.CreateForm(TDM, DM);
  Application.CreateForm(Tfrm_MailSettings, frm_MailSettings);
  Application.CreateForm(Tfrm_Settings, frm_Settings);
  Application.CreateForm(Tfrm_NewLetter, frm_NewLetter);
  Application.CreateForm(Tfrm_AskOnDuplicate, frm_AskOnDuplicate);
  Application.CreateForm(TForm1, Form1);
  Application.HintPause:=200;
  Application.Run;
end.
```

Файл frmMain.pas - основна програма

```

unit frmMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ToolWin, ComCtrls, ExtCtrls, StdCtrls, XPMan,
  IdBaseComponent, IdComponent, IdTCPConnection, IdTCPClient,
  IdMessageClient, IdPOP3, ActnList, OleCtrls, SHDocVw,
  {HTTPApp, {HTTPProd,}
  LetterStorage, frmMailSettings, 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_MailBoxes: TTreeView;
    ListView_LettersHeaders: TListView;
    ToolBar1: TToolBar;
    btn_CheckMail: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    XPManifest1: TXPManifest;
    Panel_Content: TPanel;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    ActionList1: TActionList;
    aCheckMail: TAction;
    aSendMail: TAction;
    aMailSettings: TAction;
    aExit: TAction;
    ToolBar2: TToolBar;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    aNewLetter: TAction;
    ToolButton7: TToolButton;
    MemoLog: TMemo;
    Splitter3: TSplitter;
    aSettings: TAction;
    ToolButton8: TToolButton;
    ToolButton9: TToolButton;
    N11: TMenuItem;
    Web_LetterViewer: TWebBrowser;
    Splitter4: TSplitter;
    List_Attachmant: TListView;
    Memo_LetterBody: TMemo;
    Popup_Attachment: TPopupMenu;
    mnuSaveAtt: TMenuItem;
    SaveDialog_Att: TSaveDialog;
    PopupMenuHeaders: TPopupMenu;
    mnu_DeleteLetter: TMenuItem;
    mnu_Answer: TMenuItem;
    mnu_Resend: TMenuItem;
    N12: TMenuItem;
  end;

```

```

aAnswer: TAction;
aResend: TAction;
PopupMenu_MailBoxes: TPopupMenu;
N13: TMenuItem;
Jnghfdbnmgjxnel: TMenuItem;
N14: TMenuItem;
About1: TMenuItem;
procedure aMailSettingsExecute(Sender: TObject);
procedure aExitExecute(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure aNewLetterExecute(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure aSettingsExecute(Sender: TObject);
procedure Tree_MailBoxesClick(Sender: TObject);
procedure aCheckMailExecute(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_MailBoxesDragDrop(Sender, Source: TObject; X, Y: Integer);
procedure Tree_MailBoxesDragOver(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_MailBoxesPopup(Sender: TObject);
procedure JnghfdbnmgjxnelClick(Sender: TObject);
procedure About1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
procedure ShowMailBoxesInTree;
procedure LoadLetters;
procedure AddToLog(s : shortstring);
procedure HideLetterBody;
procedure ShowInHtmlViewer(body : TStrings);
end;

const
  LogFileName = 'Actions.log';

var
  frm_Main: Tfrm_Main;

  MailInfoLoaded : boolean=false;

  { Головне сховище електроних листів }
  MailBoxes      : TMailBoxList;

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

implementation

```

```

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

{$R *.dfm}

//параметри облікових записів
procedure Tfrm_Main.aMailSettingsExecute(Sender: TObject);
begin
    frm_MailSettings.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 MailInfoLoaded then
    begin
        { Show Mailboxes }

        ShowMailBoxesInTree;
        LoadLetters;
        MailInfoLoaded:=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.ShowMailBoxesInTree;
var
    i      : word;
    node   : TTreeNode;
begin
    Tree_MailBoxes.Items.Clear;
    for i:=1 to TotalMailBoxes do
    begin
        node:=Tree_MailBoxes.Items.Add(nil,MailBoxesSettings[i].Name);
    end;
end;

```

```

node.ImageIndex:=0;
Tree_MailBoxes.Items.AddChild(node,'Вхідні').ImageIndex:=1;
Tree_MailBoxes.Items.AddChild(node,'Вихідні').ImageIndex:=2;
Tree_MailBoxes.Items.AddChild(node,'Відправлені').ImageIndex:=2;
Tree_MailBoxes.Items.AddChild(node,'Видалені').ImageIndex:=3;

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

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

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

if Tree_MailBoxes.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_MailBoxes.Selected.Parent = nil then
MailBoxName:=Tree_MailBoxes.Selected.Text
else
MailBoxName:=Tree_MailBoxes.Selected.Parent.Text;
frm_NewLetter.Caption:='Новий лист <'+MailBoxName+'>';
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_MailBoxesClick(Sender: TObject);
var
  MailBoxName      : shortstring;
  MailBoxFolder    : shortstring;
  LetterList       : PLetters;
  i                : word;
begin
  if Tree_MailBoxes.Selected = nil then
  begin
    HideLetterBody;
    exit;
  end;
  List_Attachmant.Clear;
  List_Attachmant.Visible:=false;
  Memo_LetterBody.Clear;
  if Tree_MailBoxes.Selected.Parent = nil then
  begin
    if MailBoxes.FindBox(Tree_MailBoxes.Selected.Text).IsBuisyNow then
    aCheckMail.Enabled:=false
    else
    aCheckMail.Enabled:=true;
    ListView_LettersHeaders.Clear;
    HideLetterBody;
    findMailBoxAndSelectIt(Tree_MailBoxes.Selected.Text);
    exit;
  end;

  MailBoxName      := Tree_MailBoxes.Selected.Parent.Text;
  MailBoxFolder    := Tree_MailBoxes.Selected.Text;

  LetterList:=MailBoxes.FindLetters(MailBoxName,MailBoxFolder);

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

  if MailBoxes.FindBox(MailBoxName).IsBuisyNow then aCheckMail.Enabled:=false
  else aCheckMail.Enabled:=true;
  findMailBoxAndSelectIt(MailBoxName);

  for i:=1 to LetterList^.Count do
  with ListView_LettersHeaders.Items.Add do
  begin
    Caption:=LetterList^.Letters[i]^Subject;
    if LetterList^.Letters[i]^Name='' then
    SubItems.Add(LetterList^.Letters[i]^From.Address)
    else
    SubItems.Add(LetterList^.Letters[i]^From.Name);
    SubItems.Add(DateTimeToStr(LetterList^.Letters[i]^Date));
    //Позмір
    SubItems.Add( inttostr(LetterList^.LettersSizes[i]) );
    case LetterList^.Letters[i]^Priority of
      mpHighest : SubItems.Add('Дуже високий');
      mpHigh     : SubItems.Add('Високий');
    end;
  end;
end;

```

```

mpNormal : SubItems.Add('Звичайний');
mpLow    : SubItems.Add('Низький');
mpLowest : SubItems.Add('Дуже низький');
end;
end;

end;

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

(*
procedure RetrieveMail(let_count : word);
var
  mes : TIdMessage;
  i : word;
  LettersNot2beDownloaded : word;
Begin
  try
  LettersNot2beDownloaded:=0;
  for i:=1 to let_count do
    if CurrentMailBoxSettings^.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
MailBoxes.GetByName(CurrentMailBoxSettings^.Name).HasLetterWithID(mes.MsgId)
then
          begin
            inc(LettersNot2beDownloaded);
            continue;
          end;
        if
MailBoxes.GetByName(CurrentMailBoxSettings^.Name).hadLetterWithId(mes.msgId)
then
          begin
            case CurrentMailBoxSettings.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;
              end;
            end;
          end;
        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;

    frm_Main.AddToLog('Отримані листи
('+inttostr(i)+'/'+inttostr(let_count)+' ) от '+mes.From.Text+' по Теме
:"'+mes.Subject+'"');
    DM.POP.Retrieve(i,mes);
    MailBoxes.GetByName(CurrentMailBoxSettings^.Name).Inbox.Add(@mes,false,
DM.POP.RetrieveMsgSize(i));
    if CurrentMailBoxSettings^.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.aCheckMailExecute(Sender: TObject);
var
    CurMailBox      :   PMailBox;
begin
    // Перевірити пошту
    Tree_MailBoxesClick(self);
    if CurrentMailBoxSettings =nil then exit;

    aCheckMail.Enabled:=false;
    CurMailBox:=MailBoxes.GetByName(CurrentMailBoxSettings^.Name);
    if CurMailBox.IsBuisyNow then exit;
    try
        // Імітація переривання таймера для ініціювання процесу перевірки пошти
        CurMailBox.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
    MailBoxes.SaveMailBoxes;
end;

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

procedure Tfrm_Main.List_AttachmantClick(Sender: TObject);
var
    MailBoxName      :   shortstring;
    MailBoxFolder    :   shortstring;
    LetterList       :   PLetters;

```

```

begin
  if Tree_MailBoxes.Selected = nil then exit;
  if Tree_MailBoxes.Selected.Parent = nil then
  begin
    findMailBoxAndSelectIt(Tree_MailBoxes.Selected.Text);
    exit;
  end;
  if (ListView_LettersHeaders.Selected=nil)or
    (List_Attachmant.Selected=nil) then exit;

  MailBoxName := Tree_MailBoxes.Selected.Parent.Text;
  MailBoxFolder := Tree_MailBoxes.Selected.Text;

  LetterList:=MailBoxes.FindLetters(MailBoxName,MailBoxFolder);

  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 CurrentMailBoxSettings^.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
  MailBoxName      : shortstring;
  MailBoxFolder    : shortstring;
  LetterList       : PLetters;
  intIndex         : integer;
  BodyPart         : word;
begin
  if Tree_MailBoxes.Selected = nil then exit;
  if Tree_MailBoxes.Selected.Parent = nil then
    begin
      findMailBoxAndSelectIt(Tree_MailBoxes.Selected.Text);
      exit;
    end;
  if ListView_LettersHeaders.Selected = nil then exit;

  MailBoxName := Tree_MailBoxes.Selected.Parent.Text;
  MailBoxFolder := Tree_MailBoxes.Selected.Text;

  LetterList := MailBoxes.FindLetters(MailBoxName, MailBoxFolder);

  //Список листів
  if pos('TEXT/HTML' ,
ANSIUPPERCASE(LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^
.ContentT
ype)) <> 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]^MessageParts.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]^MessageParts.Items[intIndex].ContentType) = 'TEXT/PLAIN'
then Caption:='Частина #' + inttostr(BodyPart)
else
If
AnsiUppercase(LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^MessageParts.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.Strings[0])=AnsiUppercase('Це повідомлення складається з кількох частин у форматі MIME'))
and
(AnsiUppercase(LetterList^.Letters[ListView_LettersHeaders.ItemIndex+1]^MessageParts.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
    MailBoxName      : shortstring;
    MailBoxFolder    : shortstring;
    LetterList       : PLetters;
Begin
    if Tree_MailBoxes.Selected = nil then exit;
    if Tree_MailBoxes.Selected.Parent = nil then
        begin
            findMailBoxAndSelectIt(Tree_MailBoxes.Selected.Text);
            exit;
        end;
    if (ListView_LettersHeaders.Selected=nil)or
        (List_Attachmant.Selected=nil) then exit;

    MailBoxName      := Tree_MailBoxes.Selected.Parent.Text;
    MailBoxFolder    := Tree_MailBoxes.Selected.Text;

    LetterList:=MailBoxes.FindLetters(MailBoxName,MailBoxFolder);

    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
    MailBoxName,MailBoxFolder : shortstring;
    LetterList                : PLetters;
begin
    if ( Tree_MailBoxes.Selected=nil ) or
        ( Tree_MailBoxes.Selected.Parent = nil ) then exit;
    if ListView_LettersHeaders.Selected = nil then exit;

    MailBoxName := Tree_MailBoxes.Selected.Parent.Text;
    MailBoxFolder := Tree_MailBoxes.Selected.Text;

    LetterList:=MailBoxes.FindLetters(MailBoxName,MailBoxFolder);

MailBoxes.FindBox(MailBoxName).MoveToGarbage(LetterList,ListView_LettersHeaders.
ItemIndex+1);
    ListView_LettersHeaders.DeleteSelected;
end;

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

procedure Tfrm_Main.aAnswerExecute(Sender: TObject);
var
    MailBoxName,MailBoxFolder : shortstring;
    LetterList                : PLetters;
begin
    MailBoxName := Tree_MailBoxes.Selected.Parent.Text;
    MailBoxFolder := Tree_MailBoxes.Selected.Text;

    LetterList:=MailBoxes.FindLetters(MailBoxName,MailBoxFolder);

    frm_NewLetter.ClearForm;

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

procedure Tfrm_Main.aResendExecute(Sender: TObject);
var
    MailBoxName,MailBoxFolder : shortstring;
    LetterList                : PLetters;
    i                        : integer;
begin
    MailBoxName := Tree_MailBoxes.Selected.Parent.Text;
    MailBoxFolder := Tree_MailBoxes.Selected.Text;

    LetterList:=MailBoxes.FindLetters(MailBoxName,MailBoxFolder);

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

    frm_NewLetter.Мемо_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('<--- Пересилаемий лист (кінець) --->');

end;

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

    DestMailBox,
    DestFolderName : shortstring;
    DestLetters : PLetters;

begin
    node:=Tree_MailBoxes.GetNodeAt(x,y);
    if (node = nil) or (node.Parent = nil) then exit;
    if Tree_MailBoxes.Selected.Parent= nil then
SourceMailBoxName:=Tree_MailBoxes.Selected.Text
        else
SourceMailBoxName:=Tree_MailBoxes.Selected.Parent.Text;
    SourceFolderName:=Tree_MailBoxes.Selected.Text;
    SourceLetters:=MailBoxes.FindLetters(SourceMailBoxName,SourceFolderName);

    DestFolderName:=node.Text;
    DestMailBox:=node.Parent.Text;
    DestLetters:=MailBoxes.FindLetters(DestMailBox, DestFolderName);
    DestLetters.Add(SourceLetters.Letters[ListView_LettersHeaders.ItemIndex+1],

false,SourceLetters.LettersSizes[ListView_LettersHeaders.ItemIndex+1]);

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

procedure Tfrm_Main.Tree_MailBoxesDragOver(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_MailBoxes.Selected = nil) or (Tree_MailBoxes.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;
end;

```

```

    //Application.Minimize;
end;

// Delete All Letters in the folder
procedure Tfrm_Main.N13Click(Sender: TObject);
var
    Letters : PLetters;
    i       : word;
begin
    if Tree_MailBoxes.Selected=nil then exit;
    if Tree_MailBoxes.Selected.Parent = nil then exit;
    Letters:=MailBoxes.FindLetters(Tree_MailBoxes.Selected.Parent.Text,
Tree_MailBoxes.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_MailBoxesClick(self);
end;

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

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

// Відправити всі листи в поштовій скриньці
procedure Tfrm_Main.Jnghfdbnmjgxn1Click(Sender: TObject);
var
    BoxName : shortstring;
    MailBox : PMailBox;
    Letters : PLetters;
    i       : word;
begin
    if Tree_MailBoxes.Selected=nil then exit;
    if Tree_MailBoxes.Selected.Parent=nil then
        BoxName:=Tree_MailBoxes.Selected.Text
    else
        BoxName:=Tree_MailBoxes.Selected.Parent.Text;
    MailBox:=MailBoxes.FindBox(BoxName);
    if MailBox=nil then exit;
    Letters:=@MailBox.Outbox;
    if Letters=nil then exit;

    Tree_MailBoxesClick(self);

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

```
MailBox.Sent.Add(Letters.Letters[1], false, Letters.LettersSizes[1]); //
Додати до папки
MailBox.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.
```

Кафедра _ КБПЗ _ 2023 рік

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,
  frmMailSettings, 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;
  MailBoxName : shortstring;
  CURR_mailbox_SETTINGS : TMailBoxSettings;
  CURR_MAIL_BOX : PMailBox;
  function CreateLetterFromForm:PidMessage;
  procedure UpdateStatusLine;
  procedure ClearForm;
end;

procedure FillSMTPwithMailBoxSettings(mbSettings : PMailBoxSettings);

var
  frm_NewLetter: Tfrm_NewLetter;

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

procedure FillSMTPwithMailBoxSettings(mbSettings : PMailBoxSettings);
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 CurrentMailBoxSettings = 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
    FillSMTPwithMailBoxSettings(CurrentMailBoxSettings);
    frm_Main.AddToLog('Негайна відправка листа від
'+CurrentMailBoxSettings.Name+' до '+mes.Recipients.EmailAddresses);
    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_BOX.Sent.Add(@mes^), false, 0);
    frm_Main.Tree_MailBoxesClick(self);
except
    on E:Exception do
        begin
            Application.MessageBox(pchar(E.message), 'Не вдалося відправити пошту');
            CURR_MAIL_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_BOX.Outbox.Add( @mes, false, 0);
    frm_Main.Tree_MailBoxesClick(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 -> Видалити 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_MailBoxesClick(self);
  if CurrentMailBoxSettings =nil then exit;

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

  PLet^.Subject:=Ed_Subject.Text;
  PLet^.CCList.EmailAddresses:=Ed_Copy.Text;
  PLet^.BccList.EmailAddresses:=Ed_HiddenCopy.Text;
  PLet^.Body.Assign(Memo_Body.Lines);
  PLet^.From.Text:= CURR_mailbox_SETTINGS.POPEmail; raises error on some servers
  PLet^.From.Text:=CurrentMailBoxSettings^.POPEmail;
  if Ed_ReplyTo.Text='' then
    PLet^.ReplyTo.EmailAddresses:=CurrentMailBoxSettings.POPEmail
      else PLet^.ReplyTo.EmailAddresses:=Ed_ReplyTo.Text;
  PLet^.Recipients.EmailAddresses:=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;

  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;

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

```

```
    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_mailbox_SETTINGS:= CurrentMailBoxSettings^;  
    CURR_MAIL_BOX:=MailBoxes.FindBox(CurrentMailBoxSettings.Name);  
end;
```

```
end.
```

Кафедра _ КБПЗ _ 2023рік

Файл LetterStorage.pas - збереження листів

```

unit LetterStorage;

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

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

type
{
TMyAttachments = object
  public
    MailBoxName,
    FolderName   : shortstring;
    Count        : word;
  constructor Init(_mailboxname, _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
    MailBoxName,
    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(MailBox, 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;
TGetMailThread = class(TThread)
  private
    POP : TIdPOP3;
    MailBoxSettings : TMailBoxSettings;
    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 : TMailBoxSettings;
IsBuisyFlag : Pboolean );
        end;

PMailBox = ^TMailBox;
TMailBox = 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      : TGetMailThread;

        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;

TMailBoxList =object
    private
        MailBoxes : array [1..100] of PMailBox;
    public
        TotalBoxes: word;
        function  Add(name : shortstring)      : PMailBox;
        function  GetByName (name : shortstring) : PMailBox;
        function  GetByIndex(index: word)      : PMailBox;
        function  FindLetters( mailbox, mailfolder : shortstring ):PLetters;
        function  FindBox( boxname : shortstring): PMailBox;
        procedure SaveMailBoxes;
        end;

var
    LettersPath      :   string;
    CS                :   TCriticalSection;

implementation

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

{ TLetter }

```

```

constructor TLetters.Init(MailBox, Folder : shortstring);
Begin
  MailBoxName := MailBox;
  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
  findMailBoxAndSelectIt(MailBoxName);
  if CurrentMailBoxSettings.DynamicFolders then exit;

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

  try
    stCompressed:=TFileStream.Create(SourceFile, fmOpenRead);
  except
    CreateEmptyLellerArchive(SourceFile);
    frm_Main.AddToLog('Не удалось найти архивные файлы ящика <'+MailBoxName+'>.
Они будут созданы заново. ');
    Application.MessageBox(pchar('Архів листів "'+SourceFile+'" не може бути
відкритий і розархівований! Exception ='+E.Message), 'Помилка завантаження за
архівованих 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+MailBoxName+' '+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;
  mailbx: PMailBox;
  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);

  mailbx:=MailBoxes.FindBox(MailBoxName);
  if mailbx = nil then exit;
  IdIndex:=mailbx.DeletedIDs.IndexOf(mesID);
  if IdIndex<>-1 then exit
    else mailbx.DeletedIDs.Add(mesID);
end;

```

```

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

```

```

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

    mailbx:=MailBoxes.FindBox (MailBoxName);
    if mailbx = nil then exit;
    IdIndex:=mailbx.DeletedIDs.IndexOf (mesID);
    if IdIndex<>-1 then exit
        else mailbx.DeletedIDs.Add (mesID);
end;

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

constructor TMailBox.Init (name : shortstring);
Begin
    BoxName:=name;
    IsBuisyNow:=false;
    findMailBoxAndSelectIt (name);
    CheckThread:=TGetMailThread.Create (true);
    CheckThread.Init (CurrentMailBoxSettings^, @IsBuisyNow);

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

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

```

End;

```

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

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

```

```

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

```

```

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

```

```

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

```

```

function TMailBoxList.Add(name : shortstring):PMailBox;
Begin
  inc (TotalBoxes);
  new (MailBoxes[TotalBoxes]);
  MailBoxes[TotalBoxes]^ .init (name);
  Add:=MailBoxes[TotalBoxes];
End;

```

```

function TMailBoxList.GetByName (name : shortstring):PMailBox;

```

```

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

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

procedure TMailBoxList.SaveMailBoxes;
var
  i : word;
Begin
  for i:=1 to TotalBoxes do MailBoxes[i].Save;
End;

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

```

```

function TMailBoxList.GetByIndex(index : word):PMailBox;
Begin
  if (index>=1)and(index<=100) then GetByIndex:=MailBoxes[index]
  else GetByIndex:=nil;
End;

{ TMyAttachments }

constructor TMyAttachments.Init(_mailboxname,_foldername : shortstring);
Begin
  Count:=0;
  MailBoxName:=_MailBoxName;
  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;

{ TGetMailThread }

```

```

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

  MailBoxSettings:=mbSettings;
  IsBuisy:=IsBuisyFlag;
end;

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

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

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

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

procedure TGetMailThread.Execute;
var
  CurMailBox      :   PMailBox;
  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 MailBoxSettings.ShowBalloonOnNewMes then
            CallShowBalloon('Помилка підключення до сервера:'#10#13'<'+pop.host+
:'+inttostr(POP.port)+'>');
        end;
      end;
    end;
  end;
end;

```

```

    IsBuisy^:=false;
    exit;
end;
end; // of try

CurMailBox:=MailBoxes.GetByName(MailBoxSettings.Name);
LettersOnServer := POP.CheckMessages;
CurMailBox^.UsedSizeOnServer := POP.RetrieveMailBoxSize;

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

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

LettersNot2beDownloaded:=0;
for i:=1 to LettersOnServer do
if MailBoxSettings.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
{MailBoxes.GetByName(MailBoxSettings.Name)}CurMailBox^.HasLetterWithID(mes.MsgId
) then
begin
inc(LettersNot2beDownloaded);
continue;
end;
if
{MailBoxes.GetByName(MailBoxSettings.Name)}CurMailBox^.hadLetterWithId(mes.msgId
) then
begin
CS.Enter;
case MailBoxSettings.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;

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 MailBoxSettings.ShowBalloonOnNewMes then
        CallShowBalloon('Вхідний лист.'#10#13+'Від:
'+mes.From.Text+#10#13+'Тема: '+mes.Subject);
        POP.Retrieve(i,mes);

{MailBoxes.GetByName(MailBoxSettings.Name)}CurMailBox^.Inbox.Add(@mes,false,
POP.RetrieveMsgSize(i));
    if MailBoxSettings.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_MailBoxesClick(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 frmMailSettings.pas - параметри облікових записів та створення нових поштових скриньок

```

unit frmMailSettings;

interface

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

type
  PMailBoxSettings = ^TMailBoxSettings;

  Tfrm_MailSettings = class(TForm)
    ListBox_MailBoxes: 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_NewMailBox: TBitBtn;
    GroupBox4: TGroupBox;
    Label11: TLabel;
    Ed_Subscript: TEdit;
    Ed_MailBoxName: 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: TEdit;
    Check_DynamicFolders: TCheckBox;
    Check_ShowBalloon: TCheckBox;

    procedure btn_NewMailBoxClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Combo_CheckEveryExit(Sender: TObject);
  end;

```

```

procedure btn_ApplyClick(Sender: TObject);
procedure ListBox_MailBoxesClick(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 : PMailBoxSettings);
  procedure SaveFormToBox(box : PMailBoxSettings);
  procedure ClearForm;
end;

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

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

  POPServer  :shortstring;
  POPPort    :integer;
  POPAccount :shortstring;
  POPPass    :shortString;
  POPEmail   :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 findMailBoxAndSelectIt(name : shortstring);

const
  MailBoxSettingsFileName = 'mailboxes.dat';

var
  frm_MailSettings      : Tfrm_MailSettings;
  TotalMailBoxes       : word=0;
  MailBoxesSettings    : array [1..300] of PMailBoxSettings;
  CurrentMailBoxSettings : PMailBoxSettings=nil;
  PrevMailBoxSettings  : PMailBoxSettings=nil;

implementation

uses frmMain;
{$R *.dfm}

procedure findMailBoxAndSelectIt(name : shortstring);

```

```

var
  i : word;
Begin
  CurrentMailBoxSettings := nil;
  for i:=1 to TotalMailBoxes do
    if MailBoxesSettings[i]^Name = name then
      begin
        CurrentMailBoxSettings:=MailBoxesSettings[i];
        exit;
      end;
  End;

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

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

  st.Read(TotalMailBoxes, sizeof(TotalMailBoxes));
  for i:=1 to TotalMailBoxes do
    begin
      new(MailBoxesSettings[i]);
      st.Read(MailBoxesSettings[i]^, sizeof(TMailBoxSettings));
    end;
  st.Free;
End;

procedure Tfrm_MailSettings.LoadFormFromBox(box : PMailBoxSettings);
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.Text    := box.POPEmail;

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:='Ніколи';

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

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

procedure Tfrm_MailSettings.SaveFormToBox(box : PMailBoxSettings);
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  := Ed_POPEmail.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)<>'HIKOJII' 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_MailSettings.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.Text := '<nick>@<server>.ru';
Ed_Subscript.Text := 'Вася';
```

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

```
procedure Tfrm_MailSettings.LoadSettingsFromRecords;
```

```
Begin
  if TotalMailBoxes <> 0 then
  begin
    if MailBoxesSettings[1]<>nil then
    begin
      CurrentMailBoxSettings:=MailBoxesSettings[1];

      LoadFormFromBox (CurrentMailBoxSettings);
      ListBox_MailBoxes.Selected[0]:=true;
      PrevMailBoxSettings:=CurrentMailBoxSettings;
    end;
  end
  else CurrentMailBoxSettings:=nil;
End;
```

```
// створення нової поштової скриньки
```

```
procedure Tfrm_MailSettings.btn_NewMailBoxClick(Sender: TObject);
var
  i : word;
begin
  if Ed_MailBoxName.Text<>' ' then
  begin
    for i:=1 to ListBox_MailBoxes.Count do
      if AnsiUpperCase(ListBox_MailBoxes.Items[i-1])=AnsiUpperCase(Ed_MailBoxName.Text) then
      begin
        Application.MessageBox('Таке ім'я облікового запису вже існує!', 'Помилка');
        exit;
      end;
    ListBox_MailBoxes.Items.Add(Ed_MailBoxName.Text);
    GroupBox1.Visible:=true;
    Radio_Ask.Checked:=true;
    inc(TotalMailBoxes);
    new(MailBoxesSettings[TotalMailBoxes]);
    CurrentMailBoxSettings:=MailBoxesSettings[TotalMailBoxes];
    CurrentMailBoxSettings.Name:=Ed_MailBoxName.Text;
    if PrevMailBoxSettings<>nil then SaveFormToBox(PrevMailBoxSettings);
    PrevMailBoxSettings:=CurrentMailBoxSettings;
    ClearForm;
    SaveFormToBox (CurrentMailBoxSettings);
    ListBox_MailBoxes.Selected[ListBox_MailBoxes.Count-1]:=true;
    {Тепер створити і заповнити нову поштову скриньку}
    MailBoxes.Add(Ed_MailBoxName.Text);
```

```

frm_Main.AddToLog('Створена поштова скринька <'+Ed_MailBoxName.Text+'>');
frm_Main.ShowMailBoxesInTree;
end
else
Application.MessageBox('Спочатку введіть ім'я вашої поштової
скриньки', 'Помилка');
end;

procedure Tfrm_MailSettings.FormShow(Sender: TObject);
var
i : word;
begin
if (ListBox_MailBoxes.Count=0) and (TotalMailBoxes<>0) then
for i:=1 to TotalMailBoxes do
ListBox_MailBoxes.Items.Add(MailBoxesSettings[i].Name);
LoadSettingsFromRecords;
if CurrentMailBoxSettings=nil then GroupBox1.Visible:=false
else GroupBox1.Visible:=true;
end;

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

procedure Tfrm_MailSettings.btn_ApplyClick(Sender: TObject);
var
i : word;
begin
if ListBox_MailBoxes.ItemIndex>=0 then
begin
SaveFormToBox(CurrentMailBoxSettings);
end;
SaveSettingsToFile;
self.Hide;
for i:=1 to TotalMailBoxes do
if MailBoxes.GetByIndex(i)<>nil then
MailBoxes.GetByIndex(i).SetTimerInterval(MailBoxesSettings[i].CheckEvery);
end;

function FindMailBox(name : shortstring):PMailBoxSettings;
var
i : word;
Begin
FindMailBox:=nil;

for i:=1 to TotalMailBoxes do
if MailBoxesSettings[i].Name = name then
begin
FindMailBox:=MailBoxesSettings[i];
exit;
end;

End;

```

```

procedure Tfrm_MailSettings.ListBox_MailBoxesClick(Sender: TObject);
begin
  if ListBox_MailBoxes.ItemIndex >=0 then
    begin
      CurrentMailBoxSettings:=FindMailBox(ListBox_MailBoxes.Items[ListBox_MailBoxes.itemindex]);
      if CurrentMailBoxSettings=nil then
        CurrentMailBoxSettings:=PrevMailBoxSettings;
      if PrevMailBoxSettings<>nil then SaveFormToBox(PrevMailBoxSettings);
      PrevMailBoxSettings:=CurrentMailBoxSettings;
      LoadFormFromBox(CurrentMailBoxSettings);
      end;
    end;

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

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

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

procedure Tfrm_MailSettings.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);
    end;
end;

```

```

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

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

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

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

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

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

```

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

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.
```