

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 другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи клієнта
електронної пошти під ОС Android”

Виконав здобувач вищої освіти
II курсу, групи КІ-22М-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Щерба О.В.
« ____ » _____ 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*
Рівень вищої освіти *магістр*
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
« 6 » вересня 2023 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Щербі Олексію Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android*

2. Керівник роботи *Коваленко Олександр Володимирович, докт. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 34-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android*

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

- | | |
|--|---|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> |
| <i>5. Впровадження системи в промислову експлуатацію</i> | |

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

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <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> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Щерба О.В. Дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи клієнта електронної пошти під ОС Android.

Метою розробки є дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android.

Об'єктом дослідження є процес клієнта електронної пошти під ОС Android.

Предметом дослідження є методи клієнта електронної пошти під ОС Android.

Методи дослідження базуються на методах теорії телекому, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи клієнта електронної пошти під ОС Android.

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 мобільних пристроях під керуванням ОС Android.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерна інженерія, електронна пошта, ОС Android

ABSTRACT

Shcherba O.V. Research and software implementation of the e-mail client system for the Android OS. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the second (master's) level of higher education, software is developed, which is intended for the e-mail client system under the Android OS.

The purpose of the development is the research and software implementation of the e-mail client system for the Android OS.

The object of the research is the process of the e-mail client under the Android OS.

The subject of the research is methods of the e-mail client for the Android OS.

Research methods are based on telecom theory methods, mathematical statistics methods, and software development methods.

The result of the work is a software implementation of the e-mail client system under the Android OS.

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 mobile devices running the Android OS.

The program was developed in the Delphi 10.4 environment.

Keywords: computer engineering, e-mail, Android OS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	28
3.1 Опис функціонування системи	28
3.2 Розробка структурної схеми.....	36
3.3 Розробка функціональної схеми	45
3.4 Розробка діаграми процесів.....	61
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	63
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	63
4.2 Захист розробленого програмного забезпечення.....	78
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	80
6 НАУКОВА НОВИЗНА	82

					ВКРМ-123.23.0025.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android	Літ.	Аркуш	Аркушів
Розроб.	Шерба О.В.					М	1	123
Перев.	Коваленко О.В.					ЦНТУ КІ-22М-1		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	83
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	83
7.2 Розрахунок трудомісткості розробки програмної продукції.....	85
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	87
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	92
7.5 Визначення собівартості розробки та ціни програмної продукції.....	96
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	99
7.7 Визначення експлуатаційних витрат.....	99
7.8 Визначення економічної ефективності програмної продукції.....	101
7.9 Висновок.....	103
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	104
8.1 Вступ.....	104
8.2 Пожежна безпека.....	106
8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців.....	107
8.4 Розробка заходів з умов поліпшення охорони праці.....	109
8.5 Розрахункова частина	110
9 ОСНОВНІ ВИСНОВКИ.....	115
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	117

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 Перезавантажень
- RSS – Really Simple Syndication
- SMTP – Simple Mail Transfer Protocol
- SSL – Secure Socket Layer
- TLS – Transport Layer Security
- USENET – User Network

КБГЗ-2023

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Мабуть, самим затребуваним сервісом у мобільному телефоні є електронна пошта. Це можливість залишатися на зв'язку із друзями й колегами. Пошта в смартфоні дозволяє вивчити переписку у дорозі або у випадках, коли під рукою немає робітника комп'ютера або ноутбука. Можна вивчити документ, відправити свої зауваження й все це в маленькому кишеньковому пристрої.

Поштові аккаунти відіграють величезну роль у повсякденному житті сучасних інтернет-користувачів. Без прив'язки облікового запису до електронного ящика практично неможливо зареєструватися в жодному з популярних у Мережі сервісів.

Як і більшість програмних продуктів, вони, звичайно, мають цілком робочі стоківі аналоги від виробників пристроїв. Однак у більшості випадків налаштування й прив'язка декількох аккаунтів в убудованих поштових клієнтів реалізована незручно.

То необхідно окремо прописувати сервера вхідних і вихідних повідомлень, то обов'язково потрібно ввести ідентифікатор порту. Та й сама робота з ними найчастіше сполучена з набором труднощів – від нелогічної автоматизації процесу синхронізації до цілком прозаїчної появи зайвих папок. Результат – плутанина з робочими листами. Тими ж недоліками страждають і послуги збору повідомлень із різних аккаунтів, які пропонують самі поштові сервіси.

Тому якщо на комп'ютерах часи поштових клієнтів проходять, то на смартфонах, планшетах і комунікаторах вони ще можуть послужити добру службу своїм власникам.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android.

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

- Огляд існуючих систем клієнта електронної пошти під ОС Android.
- Дослідження системи клієнта електронної пошти під ОС Android.
- Програмна реалізація системи клієнта електронної пошти під ОС Android.

Об'єктом дослідження є процес клієнта електронної пошти під ОС Android.

Предметом дослідження є методи клієнта електронної пошти під ОС Android.

Методи дослідження базуються на методах теорії телекому, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод клієнта електронної пошти під ОС Android.
- Розроблено вітчизняний продукт клієнта електронної пошти під ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі клієнта електронної пошти під ОС Android.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГПЗ - 2023

					VKPM-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для реалізації програмного забезпечення клієнта електронної пошти під ОС Android. У більшості сучасних смартфонів підтримуються самі популярні поштові сервіси. Але навіть якщо потрібно налаштувати робочу пошту (включаючи пошту з використання Microsoft Exchange), це нескладно зробити вручну. Варто сказати, що користувачі Android мають деяку перевагу – їм досить один раз увести пароль до свого облікового запису Google, після чого не тільки буде синхронізовані всі контакти, але й автоматично налаштована електронна пошта Gmail.

Якщо у вас немає облікового запису Gmail, але ви вже стали щасливим власником Android-смартфона, то її можна створити прямо в меню телефону.

Для переходу до облікових записів пошти в Android потрібно відкрити головне меню й потім перейти до пунктів «Налаштування» – «Облікові записи й синхронізація». А там уже нажати на кнопку «Додати обліковий запис».

Звичайно ж, Android дозволяє використовувати не тільки пошту Gmail у всіх її іпостасях, але й будь-яку іншу електронну пошту, включаючи корпоративну через Microsoft Exchange. Налаштувати її можна в додатку E-mail. Для початку спробуйте ввести просто адресу пошти й пароль. Якщо Android не знаком з налаштуваннями вашої поштової служби, не лихो – потрібно вибрати режим «Ручне налаштування» у вікні, що з'явилося, нажати на кнопку відповідному типу використовуваного поштового сервісу – POP3, IMAP, MS Exchange.

Довідатися всю необхідну інформацію для налаштування будь-якого поштового сервісу можна у вашого системного адміністратора, якщо мова йде про робочу пошту. І в довідковій інформаційній системі вашої пошти, якщо ви

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

користуєтеся одним з доступних інтернет-сервісів. Як правило, потрібні адреси POP3 / SMTP або IMAP-серверів, номери портів (часто вони використовуються за замовчуванням) і тип використовуваного захисту (якщо він є).

При використанні IMAP-сервісу (наприклад, якщо ви працюєте з поштою Gmail) можна переходити до папок з листами. Для цього потрібно на сторінці зі списком листів натиснути на назву папки, після чого відкриється список всіх поштових папок. Щоб налаштувати пошту через MS Exchange в Android досить увести ім'я користувача, сервер і пароль. Всі ці дані можна взяти в системного адміністратора на роботі.

Крім уведення поштових сервісів, можна налаштувати пошту так, щоб листи, завантажені в телефон, видалялися на сервері. Як правило, телефон використовується як допоміжний засіб і на сервері варто залишити копію листа для того, щоб завантажити всю пошту, коли ви будете за комп'ютером.

1.2 Область застосування

Областю застосування системи є ОС Android.

Android – операційна система для смартфонів, планшетів і нетбуков. Компанія Google придбала розроблювача програмного забезпечення Android inc. в 2005 році. Операційна система Android заснована на модифікованому ядрі Linux. Згодом, Google і інші учасники Open Headset Alliance співробітничали для спільної розробки цієї нової операційної системи. Далі Android Open Source Project (AOSP) доручені підтримка й подальший розвиток платформи. В Android є велике співтовариство розроблювачів, які розширюють функціональність пристроїв. OS Android має свій офіційний магазин із продажу додатків – Android Market. Включає він у себе як платні додатки, так і безкоштовні. У цей момент, для України доступні для завантаження програми й ігри. Так як OS Android є відкритою, користувачеві надається можливість завантажувати додатки й з інших ресурсів.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Під Android розроблювачі, в основному, пишуть додатки на мовах Java, Delphi 10, керуючі пристроєм через розроблені Google бібліотеки.

Офіційно про OS Android стало відомо 5 листопада 2007 року, коли була оголошена підстава Open Headset Alliance – консорціум з 80 компаній. Більшу частину коду Android була випущена під ліцензією Apache.

Android додатка містять у собі java-додатки й бібліотеки, які запускаються віртуальною машиною Dalvik з JIT компілятором. Бібліотеки містять у собі систему керування, графікові OpenGL ES 2.0, движок WebKit, графічний движок SGL, SSL і бібліотеки Bionic. OS Android складається з 12 мільйонів рядків коду, у тому числі 3-х мільйонів рядків XML, 2.8 мільйонів рядків на C, 2.1 мільйона рядків на Java і 1.75 мільйона рядків на C++.

Android Market

Android Market – інтернет-магазин додатків для смартфонів на базі Android, що просуває альянс Open Handset Alliance (ОНА) на чолі з Google. Містить у собі гри, клієнти соціальних мереж, офісні додатки, додатки для читання новин, керування фінансами й інші.

22 жовтня 2008 року Google оголосила про відкриття цього онлайн-магазину додатків для Android.

В Україні на смартфонах перший час був відсутній офіційний магазин додатків. Ситуація була виправлена 12 січня 2010 року, коли про це повідомили в Samsung Україна, природно, що пізніше Android Market з'явився й в інших вендорів – Motorola, HTC, LG і Sony Ericsson, що значно підігріло інтерес до Android-смартфонам в Україні.

Першим офіційним смартфоном з Android Market в Україні став Samsung i5700 Galaxy Spica.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Cloud Magic

Клієнт, дуже схожий на стоковий. Єдина відмінність – при прив'язці аккаунту для деяких сервісів не обов'язково прописувати налаштування серверів вхідних і вихідних повідомлень. Gmail, Google Apps, Office 365, Yahoo, Hotmail і iCloud мають базову підтримку в додатку, і при підключенні до них всього їхнього налаштування (фільтри, папки, архіви й «чорні списки») автоматично інтегруються з Cloud Magic. Крім вищезгаданих поштових служб, підтримуються будь-які ІМАР-сервіси. Але є один істотний мінус: не всі з них одержать можливість працювати із передвстановленими директоріями (Вхідні, Вихідні, Кошик).

Взаємодія з елементами керування не викличе труднощів навіть у починаючих користувачів: зовнішній вигляд і панелі інструментів (як для планшетів, так і для смартфонів) зроблені максимально схожими на стокові поштові клієнти. Для аматорів секретних переписок буде корисний убудований парольний захист доступу до облікових записів.

Переваги:

- повна підтримка популярних поштових сервісів;
- простий і зручний інтерфейс;
- захист облікових записів паролем.

Недоліки:

- не всі поштові сервіси коректно працюють із передвстановленими папками.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

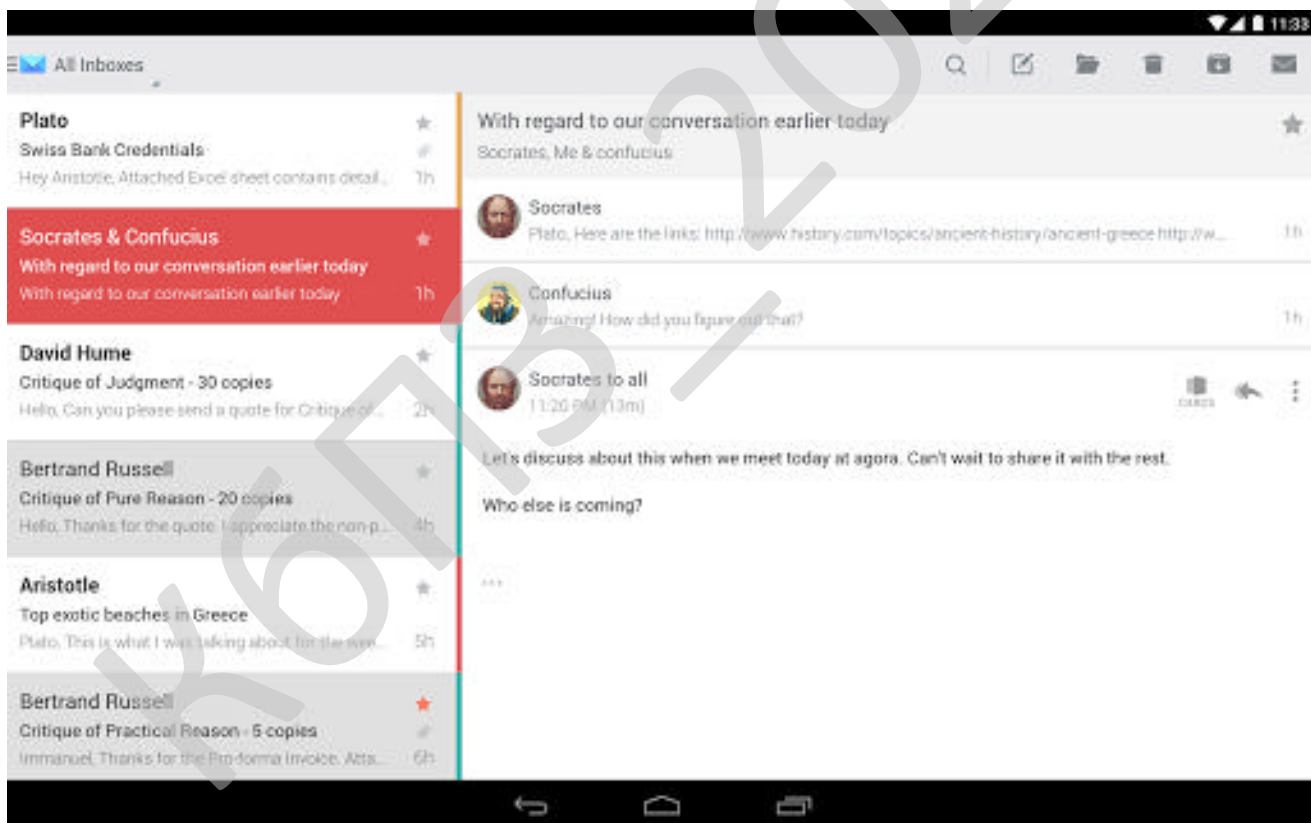
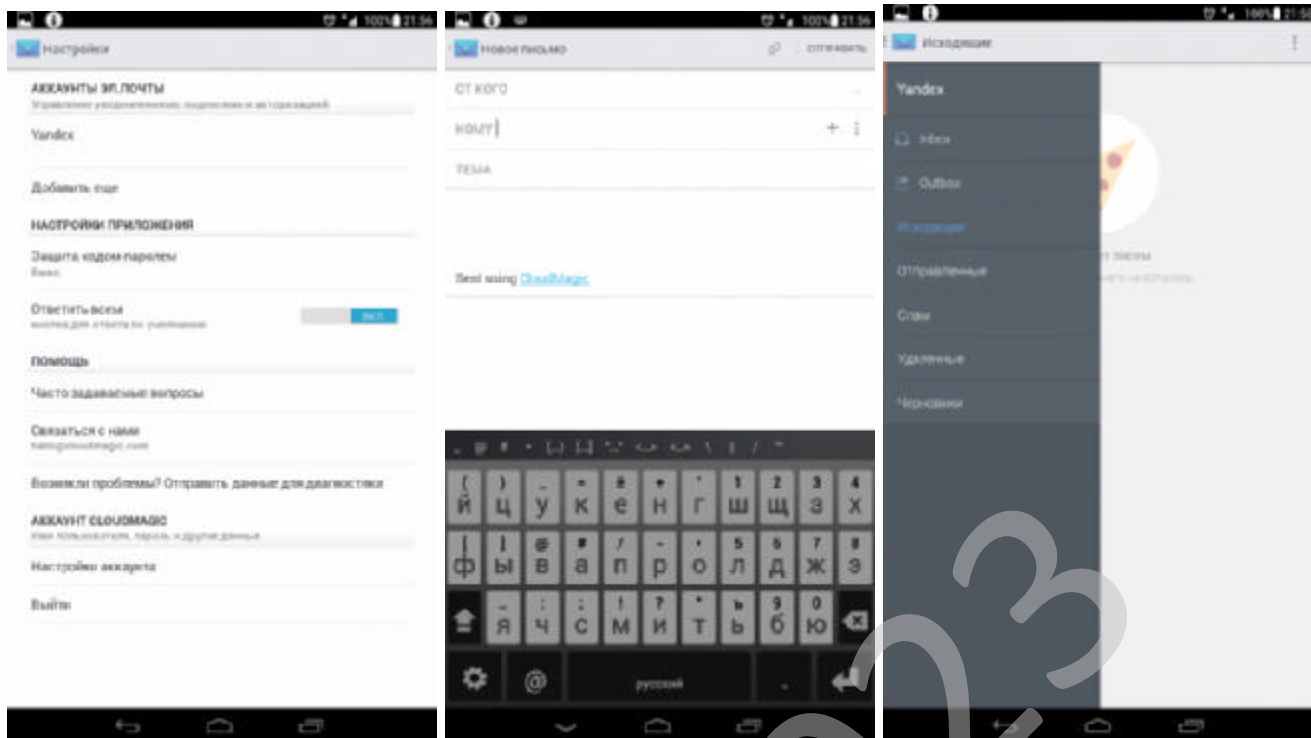


Рисунок 2.1 – Інтерфейс користувача

Пошта К-9

Додаток, що зарекомендував себе як дуже надійна поштова програма. «Пошта К-9» – повнофункціональний клієнт, що працює практично з усіма відомими поштовими сервісами.

На початку роботи з додатком необхідно ввести дані ваших аккаунтів... і, властиво, усе. Розроблювачі позбавили користувачів від метушні з ручним прописуванням серверів і портів підключення, залишивши тільки можливості редагування списку папок і створення окремих каталогів для різних облікових записів. Додаток уміє працювати як з окремими директоріями аккаунтів, так і збирати все на стартовому екрані – для швидкого доступу до редагування повідомлень. З додаткових можливостей не можна не відзначити тонке налаштування повідомлень (приховання теми, звук, вібрація), керування жестами, керування вкладеннями й, звичайно, підтримку різних типів поштових серверів (IMAP, POP3, Exchange, WebDAV).

Недолік у додатка тільки один: не зовсім зручне керування, що вимагає для роботи з листами з різних аккаунтів входу зі стартового екрана навіть на планшетах.

Переваги:

- великий список підтримуваних поштових сервісів;
- підтримка стандартів IMAP, POP3, Exchange, WebDAV;
- тонке налаштування повідомлень.

Недоліки:

- не дуже зручне керування.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

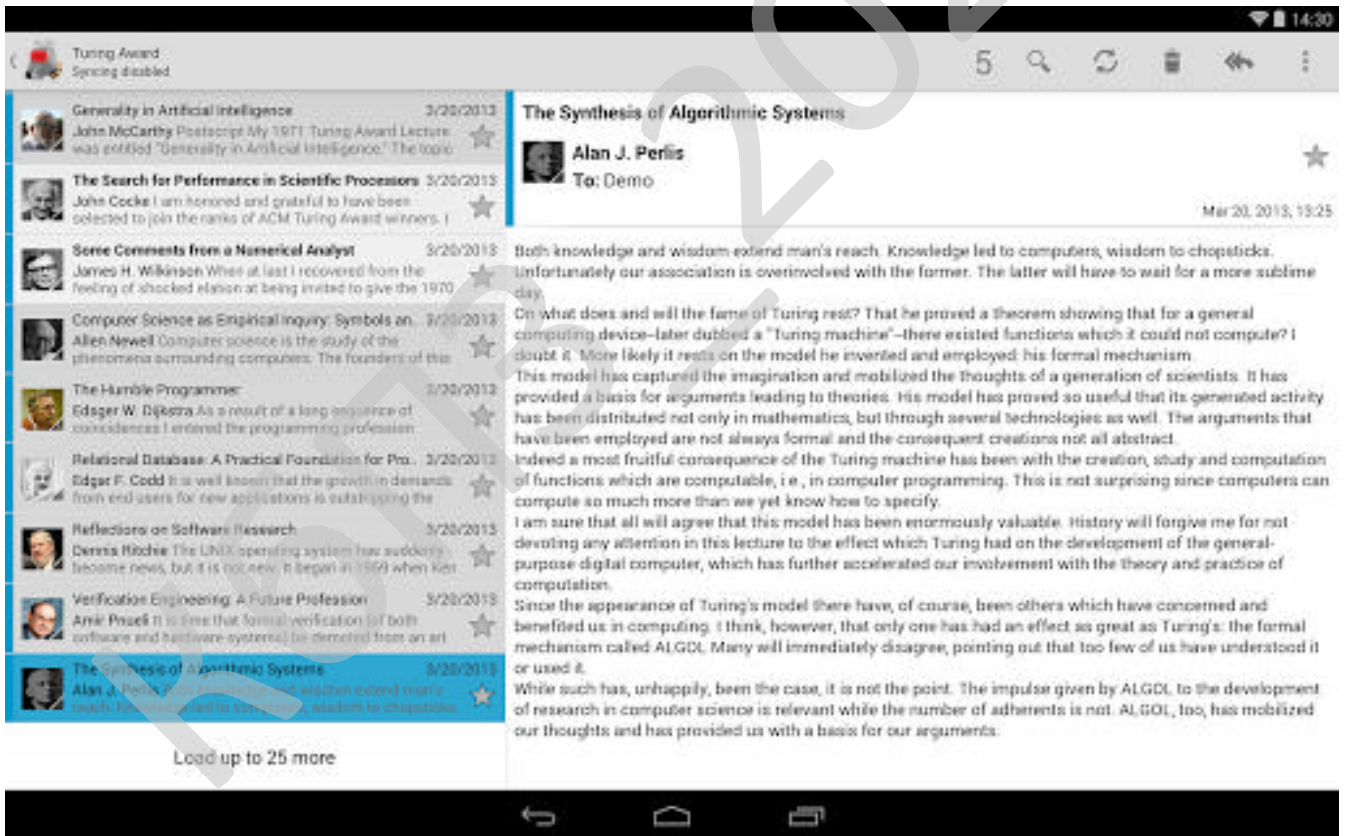
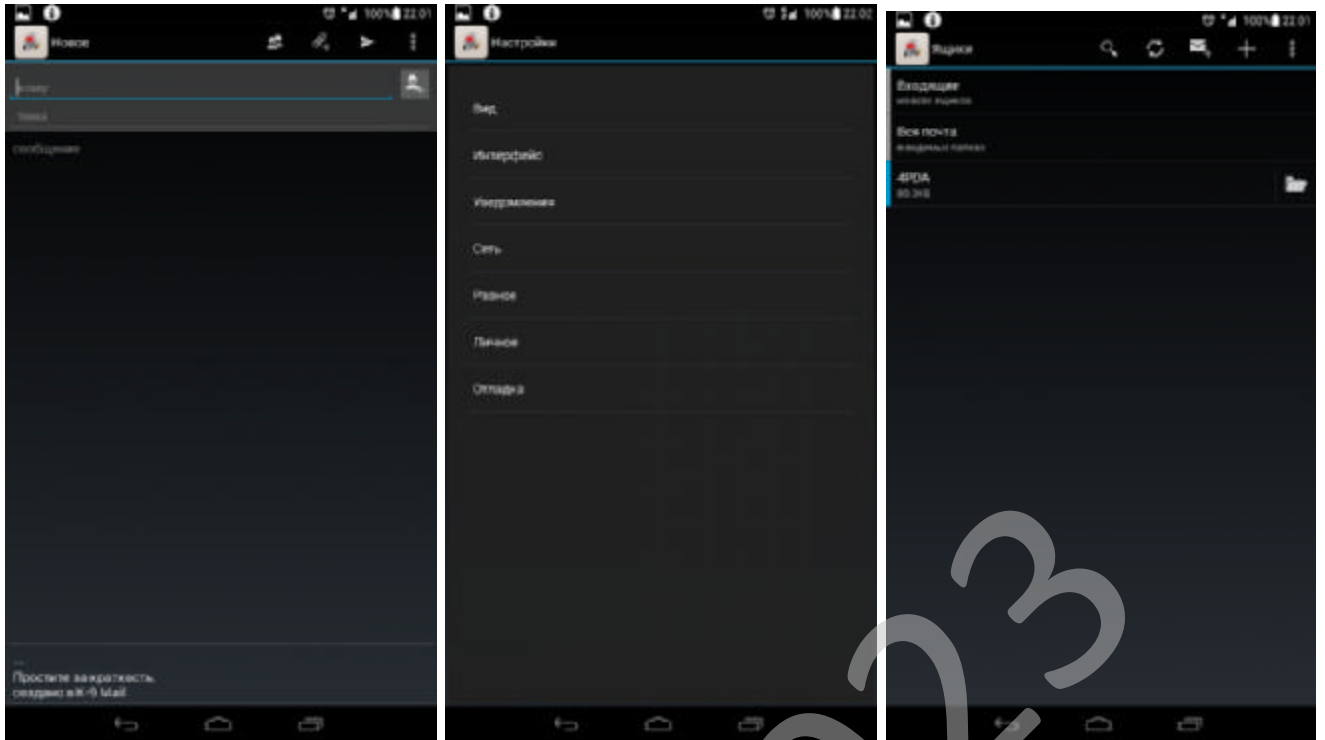


Рисунок 2.2 – Інтерфейс користувача Пошта К-9

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Blue Mail

Додаток з підтримкою величезної кількості поштових сервісів – як міжнародних (Yahoo, Google, Outlook, etc.), так і орієнтованих на українського користувача. Базова підтримка IMAP-серверів гарантує зручну роботу із передвстановленими директоріями й не вимагає додаткових маніпуляцій із чищенням зайвих папок, що дублюють оригінальні каталоги сервісів.

Сам додаток має набір зручних інструментів: фільтри, розклад для відправлення й розсилання листів, налаштування автоматизації, дві схеми інтерфейсу (темна й світла). Робота з аккаунтами може здійснюватися як з окремих облікових записів, так і із загальної папки вхідних повідомлень, що істотно спрощує перегляд, створення й відправлення пошти.

Яких-небудь критичних недоліків у додатка немає. Більше того: розроблювачі продовжують активну доробку проекту, і незабаром обіцяють додати підтримку більшої кількості сервісів за рахунок взаємодії з POP3-серверами.

Переваги:

- підтримка всіх популярних поштових сервісів (у тому числі Mail.ru і Yandex);
- повна підтримка передвстановлених директорій;
- загальна (програмна) папка «Вхідні».

Недоліки:

- об'єктивних недоліків немає.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

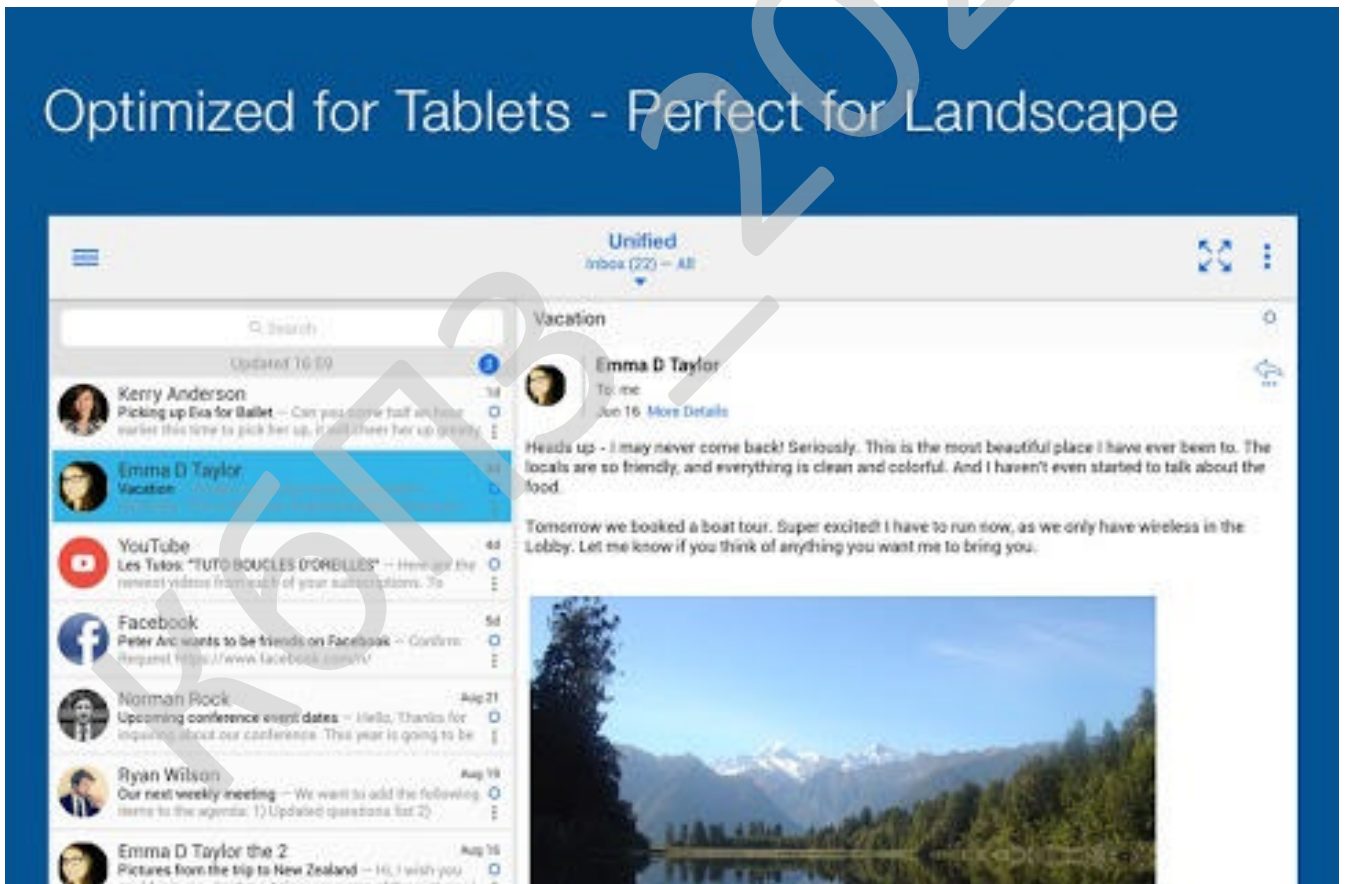
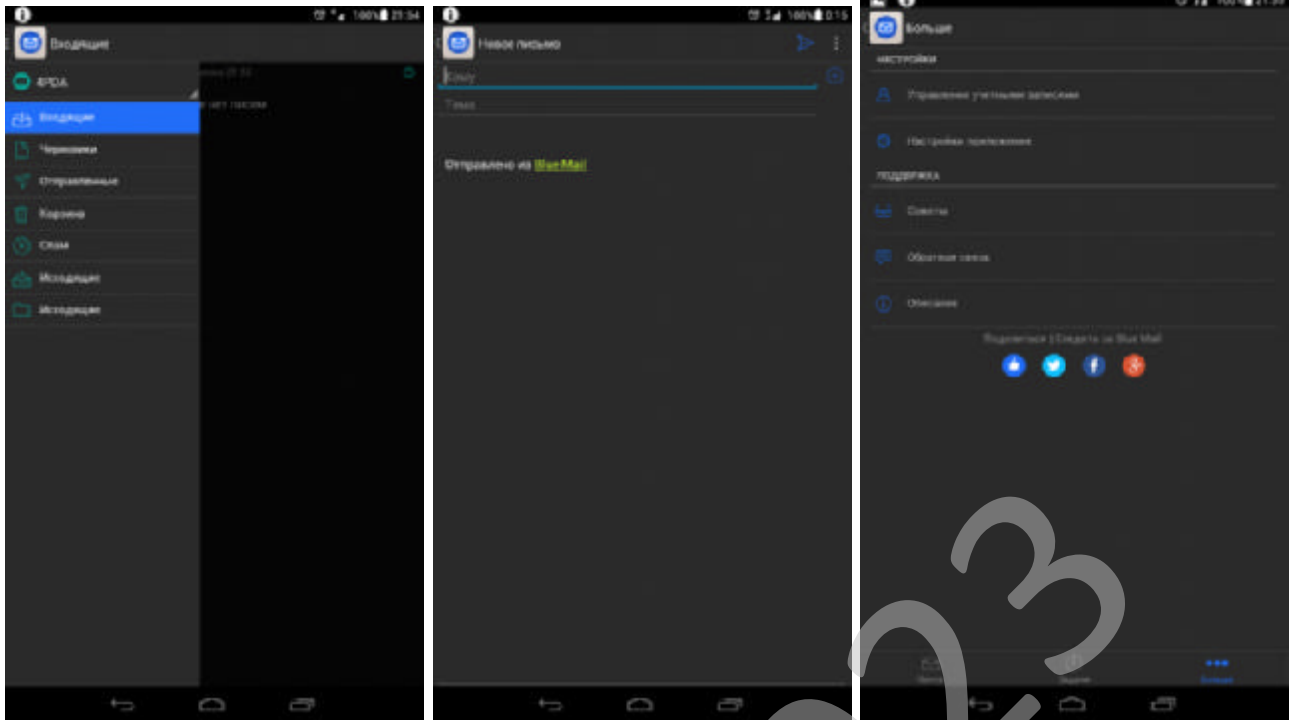


Рисунок 2.3 – Інтерфейс користувача Blue Mail

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Mailbox

Нестандартне рішення для керування обліковими записами поштових сервісів. На жаль, у додатка є один мінус – розроблювачі передбачили підтримку тільки Gmail і iCloud.

Однак далі – тільки переваги: інтуїтивно зрозумілий трьохвіконий інтерфейс (Недавні, Вхідні, Вихідні), висувна панель швидкого доступу для переходу між обліковими записами й додатковими директоріями з листами, передвстановленні шаблони швидких відповідей, автосортування, тонке налаштування панелі повідомлень і сигналу подій, інтеграція з Dropbox.

Головна ж фішка додатка – керування жестами. Кожний запис легко позначити, як прочитану, видалити, архівувати або викликати форму відповіді простим свайпом. Але саме головне: на відміну від того ж убудованого в прошивання практично всіх Android-пристроїв Gmail-клієнта, в Mailbox дуже просто працювати з текстом – як тільки що написаним, так і з отриманим у повідомленні. І все це – без збоїв і затримок, на що скаржаться багато користувачів мобільного Gmail.

Переваги:

- керування жестами;
- зручна робота з текстом;
- інтуїтивно зрозумілий інтерфейс.

Недоліки:

- на жаль, підтримуються тільки аккаунти Gmail і iCloud.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

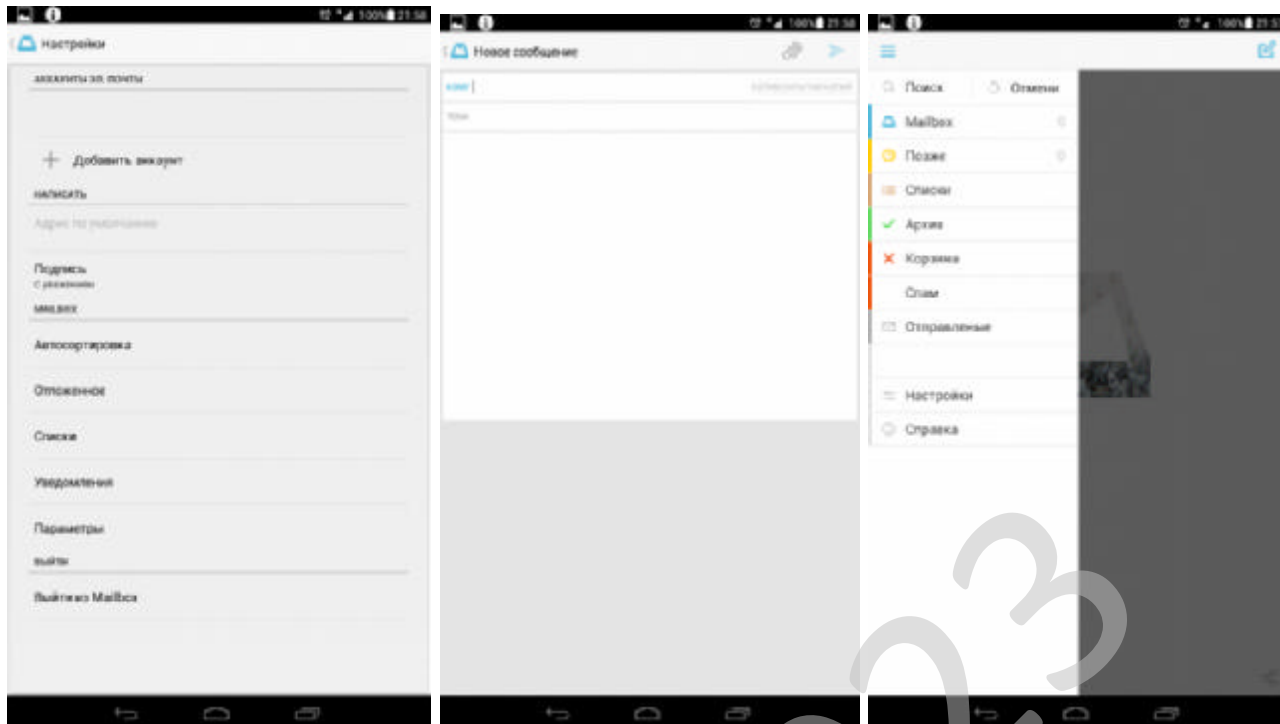


Рисунок 2.4 – Интерфейс користувача Mailbox

myMail

Дуже зручний поштовий клієнт із підтримкою великої кількості популярних сервісів (Google, Yahoo, Hotmail, Outlook і ін.). По великому рахунку, myMail мало чим відрізняється від описаних вище додатків, але це і є головна його особливість.

При першому знайомстві із цим поштовим клієнтом потрібно один раз увести свої дані про всі аккаунти поштових служб, якими ви користуєтеся – і можна забути про інших клієнтів. До цього розташовують і зручний інтерфейс із висувною панеллю швидких переходів, і можливість швидкої відповіді заданими шаблонами, і налаштування безпеки/конфіденційності. І, звичайно, різні фільтри для різних облікових записів. Панель швидкого запуску влаштована так, що кожний обліковий запис можна підбудувати незалежно від іншої. При цьому – без зайвих переходів до налаштувань самого поштового сервісу, – додаток дозволяє робити швидку заміну підписів і аватарок.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Робота з електронною кореспонденцією теж нічим особливим не виділяється: підтримка передвстановлених директорій без створення дублюючих папок, створення схованих копій, швидке редагування підпису й, звичайно, прив'язка вкладень (музика, фото, відео, документи). Словом, розроблювачі вирішили зробити максимально універсальний поштовий клієнт – і поки в них це непогано виходить.

Переваги:

- зручний інтерфейс;
- швидкий доступ до всіх облікових записів;
- тонке незалежне налаштування кожного ящика;
- відсутні дублюючі папки при роботі з поштовими сервісами.

Недоліки:

- серйозних недоліків не виявлено.

У наш огляд потрапили як прості, зручні й популярні поштові клієнти, так і нові рішення з таки оригінальним способом керування електронною кореспонденцією. Ми не випадково не включили до списку традиційні додатки від розроблювачів, що прямо залежать від поштових ресурсів («Пошта.Mail.ru», «Яндекс.Пошта», Gmail і ін.). При всій їхній зручності тримати на пристрої кілька окремих клієнтів разом – надто ресурсозатратно.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

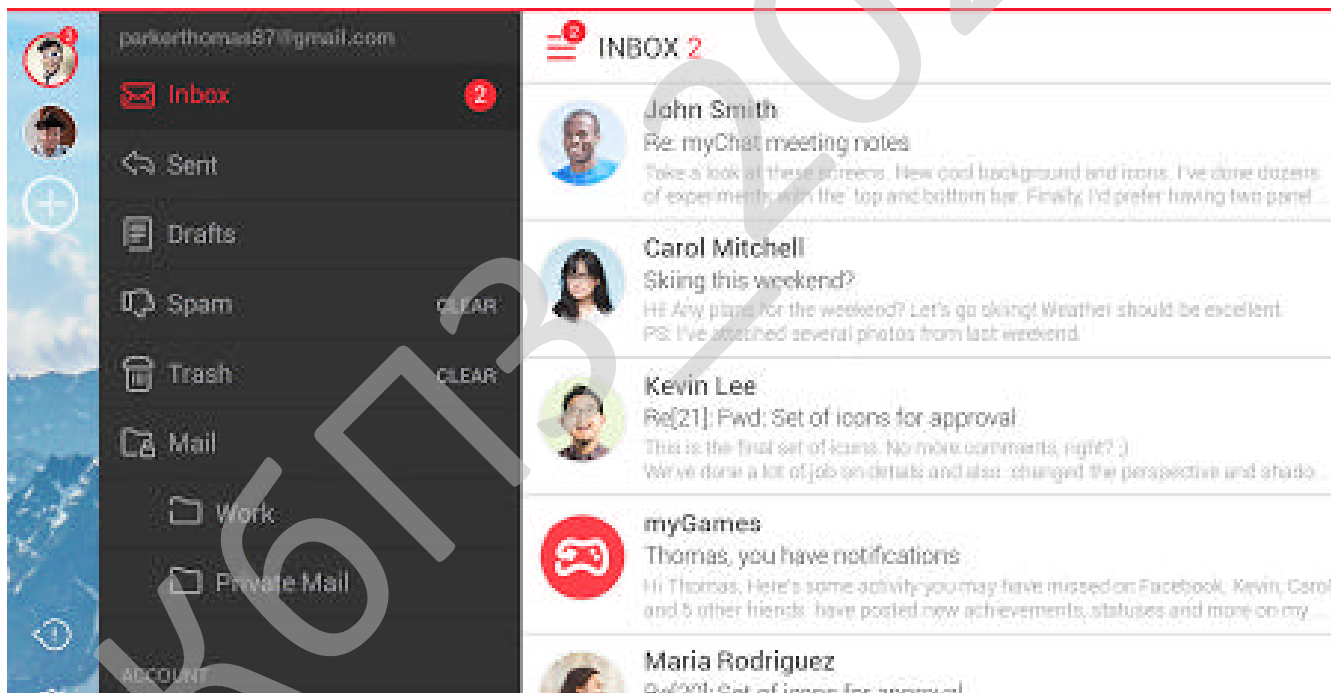
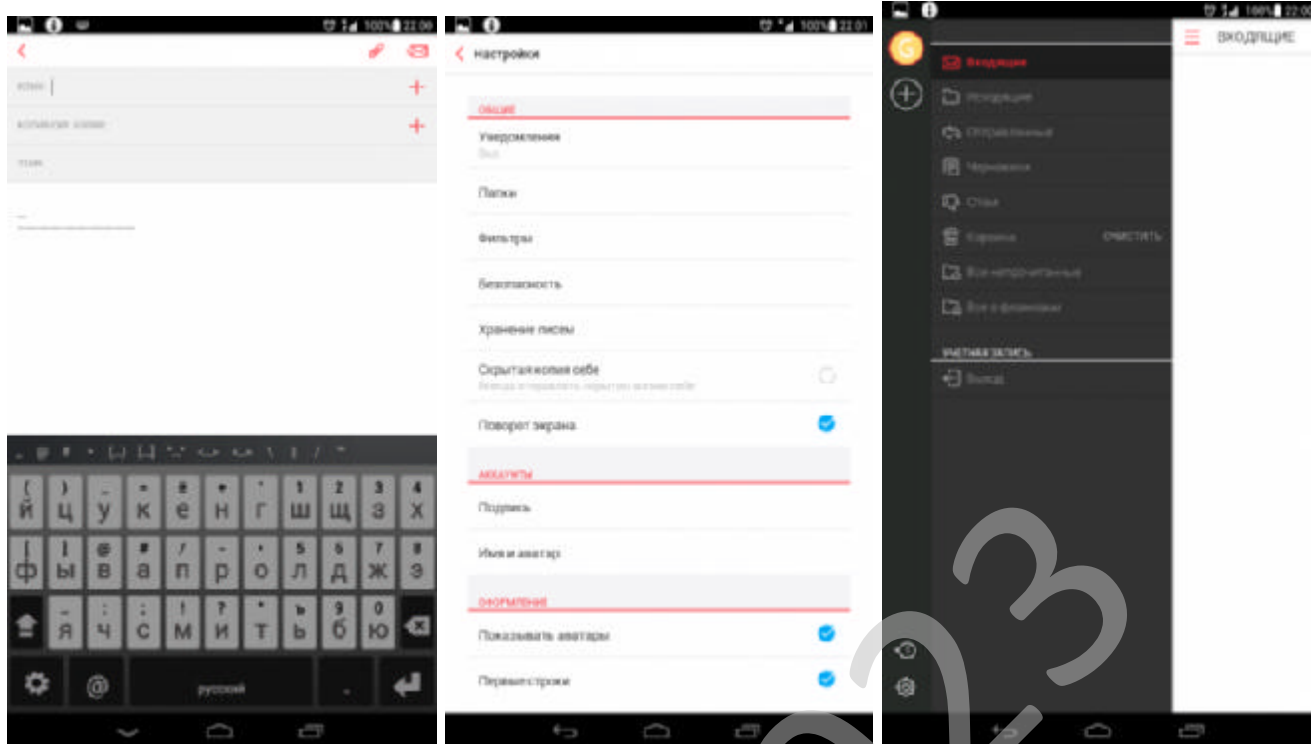


Рисунок 2.5 – Интерфейс користувача myMail

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

- Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.
- Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.
- Відладник Win 64 (на LLDB) і збирач для C++.
- Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.
- Підтримка Metal Driver GPU для macOS і iOS.
- Вбудований Fmxlinux.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).
- Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.
- Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services
- У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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 – використовуючи класичну реалізацію керування пам'яттю об'єктів.

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

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

Розширена підтримка бібліотек C++

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

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, 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.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Підвищення якості й швидкодії інструментів

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

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи клієнта електронної пошти під ОС Android.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розробка програмного забезпечення

Додатки для Android є програмами в нестандартному байт-кодi для віртуальної машини Dalvik. Google пропонує для вільного завантаження інструментарій для розробки (Android SDK), що призначений для x 86-машин під операційними системами Windows XP, Android, Mac OS X (10.4.8 або вище) і Linux. Для розробки потрібно JDK 5 або JDK 6.

Розробку додатків для Android можна вести мовою Java (не нижче Java 1.5). Існує плагін для Eclipse – «Android Development Tools» (ADT), призначений для Eclipse версій 3.3-3.5. Для IntelliJ IDEA також існує плагін, що полегшує розробку Android-Додатків. Повідомляється, що для середовища розробки NetBeans IDE розроблений експериментальний плагін.

Архітектура OS Android

Рівень додатків (Applications)

До складу Android входить комплект базових додатків: клієнти електронної пошти й SMS, календар, різні карти, браузер, програма для керування контактами й багато інше. Всі додатки, що запускаються на платформі Android написані мовою Java.

Рівень каркаса додатків (Application Framework)

Android дозволяє використовувати всю міць API, використовуваного в додатках ядра. Архітектура побудована таким чином, що будь-який додаток може використовувати вже реалізовані можливості іншого додатка за умови, що останнє відкриє доступ на використання своєї функціональності. Таким чином, архітектура реалізує принцип багаторазового використання компонентів ОС і додатків.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Основою всіх додатків є набір систем і служб:

1. Система подань (View System) – це багатий набір подань із розширеною функціональністю, що служить для побудови зовнішнього вигляду додатків, що включає такі компоненти, як списки, таблиці, поля уведення, кнопки й т.п.

2. Контент-провайдери (Content Providers) – це служби, які дозволяють додаткам одержувати доступ до даних інших додатків, а також надавати доступ до своїх даних.

3. Менеджер ресурсів (Resource Manager) призначений для доступу до строкових, графічних і інших типів ресурсів.

4. Менеджер повідомлень (Notification Manager) дозволяє будь-якому додатку відображати користувацькі повідомлення в рядку статусу.

5. Менеджер дій (Activity Manager) управляє життєвим циклом додатків і надає систему навігації по історії роботи з діями.

Рівень бібліотек (Libraries)

Платформа Android включає набір C/C++ бібліотек, використовуваних різними компонентами ОС. Для розроблювачів доступ до функцій цих бібліотек реалізований через використання Application Framework. Нижче представлені деякі з них:

1. System C library – BSD-Реалізація стандартної системної бібліотеки C (libc) для пристроїв, що вбудовуються, заснованих на Linux.

2. Media Libraries – бібліотеки, засновані на PacketVideo's OpenCORE, призначені для підтримки програвання й записи популярних аудіо- і відео-форматів (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG і т.п.).

3. Surface Manager – менеджер поверхонь управляє доступом до підсистеми відображення 2D-і 3D-графічних шарів.

4. LibWebCore – сучасний движок web-браузера, що надає всю міць убудованого Android-Браузера.

5. SGL – движок для роботи з 2D-графікою.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

6. 3D libraries – движок для роботи з 3D-графікою, заснований на OpenGL ES 1.0 API.

7. FreeType – бібліотека, призначена для роботи зі шрифтами.

8. SQLite – потужний легковагий движок для роботи з реляційними БД.

Рівень середовища виконання (Android Runtime)

До складу Android входить набір бібліотек ядра, які надають більшу частину функціональності бібліотек ядра мови Java.

Платформа використовує оптимізовану, реєстр-орієнтовану віртуальну машину Dalvik, у відмінності від її стандартна віртуальна машина Java – стек-орієнтована. Кожний додаток запускається у своєму власному процесі, зі своїм власним екземпляром віртуальної машини. Dalvik використовує формат Dalvik Executable (*.dex), оптимізований для мінімального використання пам'яті додатком. Це забезпечується такими базовими функціями ядра Linux, як організація потокової обробки й низькорівневе керування пам'яттю. Байт-код Java, на якому написані ваші додатки, компілюються в dex-формат за допомогою утиліти dx, що входить до складу SDK.

Рівень ядра Linux (Linux Kernel)

Android заснований на ОС Linux версії 2.6, тим самим платформі доступні системні служби ядра, такі як керування пам'яттю й процесами, забезпечення безпеки, робота з мережею й драйверами. Також ядро служить шаром абстракції між апаратним і програмним забезпеченням.

Доступні бібліотеки:

- Bionic – бібліотека стандартних функцій, несумісна з libc.
- SSL – шифрування.
- Media Framework (PacketVideo OpenCORE, MPEG4, H.264, MP3, AAC, AMR, JPG, PNG).
- Surface Manager.
- LibWebCore (на базі WebKit).
- SGL – 2D-графіка.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- OpenGL ES – 3D-Бібліотека.
- FreeType – шрифти.
- SQLite – легковага СУБД.

У порівнянні зі звичайними додатками Linux, додатка Android підкоряються додатковим правилам:

- Content Providers – обмін даними між додатками.
- Resource Manager – доступ до таких ресурсів, як файли XML, PNG, JPEG.
- Notification Manager – доступ до рядка стану.
- Activity Manager – керування активними додатками.

Для Android був розроблений формат архівних файлів-додатків, що виконуються .apk

Android

Material Design

Яскравий і чуйний дизайн операційної системи ідеально сполучається з інтуїтивним користувальницьким інтерфейсом на всіх ваших пристроях. Колірне оформлення завдяки Material Design стало більше реалістичним завдяки світловим ефектам, тіням і новим квітам. Більше раціональне використання всього екрана й нові шрифти дозволяють детально концентруватися на важливих аспектах, не відволікаючись на дріб'язку.

Повідомлення

В Android реалізований абсолютно новий спосіб взаємодії з повідомленнями. Тепер всю систему можна налаштувати таким чином, що повідомлення будуть надходити лише в той момент, коли це зручно власникові пристрою. У цілому ж їхнє використання вийшло на абсолютно новий рівень. По-перше, справа отут у згадані вище можливостях, пов'язаних з локскріном. По-друге, в ОС з'явився особливий режим, що зветься Priority mode. Суть його полягає в тому, що користувач вибирає групу додатків і користувачів, від яких він хоче одержувати повідомлення під час роботи цього режиму. Всі інші

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

повідомлення блокуються. По-третє, є сенс ще раз відзначити повідомлення, які з'являються поверх інших додатків – це дуже зручно й практично.

Час роботи

Android може похвастатися відразу декількома нововведеннями, які пов'язані із зарядкою й часом роботи. По-перше, це новий енергозберігаючий режим, що повинен забезпечити до 90 хвилин додаткового часу. По-друге, вказівка часу, що залишилося до повної зарядки акумулятора. По-третє, зразкове брешемо, через яке акумулятор буде виряджений повністю (відображається в налаштуваннях батареї).

Безпека

Убезпечити Android пристрій тепер можна за допомогою фічі Android Smart Lock. Суть її полягає в тому, що розблокування смартфона або планшета здійснюються за допомогою обраного пристрою, що підключається по Bluetooth (смарт-годинник або приладова панель автомобіля). Крім того, в Android відбувається автоматичне шифрування даних. Додатковий захист від шкідливих додатків забезпечує SELinux.

Новий доступ до пристроїв

В Android за замовчуванням реалізований вилучений доступ до пристроїв. Якщо ви забули свій смартфон будинку, то для доступу до контенту, який зберігається на ньому, досить скористатися іншим пристроєм з версією ОС Android 5.0. Для цього потрібно просто увійти зі свого аккаунта. Також в ОС є доступ для гостей.

Нові швидкі налаштування

Швидкі налаштування тепер запускаються за допомогою двох свайпов долілиць. Нові функції: спалах, точка доступу, положення екрана, налаштування трансляції на великий екран. Крім того, включати/виключати різні налаштування тепер набагато зручніше.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Підключення

Тепер при перемиканні від WiFi до мобільного Інтернету не відбувається ніяких збоїв. Підключення здійснюється тільки до перевірених WiFi мережам, а пошук Bluetooth пристроїв став менш енергозатратним.

Продуктивність

З Android повною мірою реалізоване нове середовище виконання додатків ART. Завдяки цьому у випадку з деякими додатками відзначається чотириразовий ріст продуктивності. Крім того, ОС підтримує 64-бітні процесори (ARM, x86, і MIPS).

Медіа

У новій версії ОС зменшилася затримка відтворення. Крім того, з'явилася підтримка USB аудіопристроїв і багатоканального відтворення (системи 5.1 і 7.1). Що стосується графіки, те завдяки OpenGL ES 3.1 Android стає в один ряд з комп'ютерами й ігровими консолями. Не обійшлося й без нововведень в області фото й відео, але вони менш значні.

OK Google

Команда "OK Google" працює тепер і на екрані блокування (поки тільки для Nexus 6 і 9).

Android TV

Новий інтерфейс краще адаптований під великий екран. Крім цього в Android TV є голосовий пошук для Google Play, YouTube і деяких інших додатків (список буде розширюватися).

Опції безпеки для Android

Нові опції безпеки для Android – це Smart Lock, шифрування пристрою й SELinux. Smart Lock дозволяє знімати блокування з екрана пристрою при підключенні до зазначених Bluetooth і NFC пристроям. Крім того, надбудова містить у собі поліпшене розблокування за допомогою фронтальної камери й селфшотів, а також можливість залишати на локскріні тільки повідомлення. Що стосується шифрування пристрою, те цей процес вийшов на абсолютно новий

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

рівень. Справа в тому, що на кожному пристрої шифрування відбувається по унікальному алгоритмі, що істотно утрудняє доступ зловмисників до користувальницьких даних. Якщо ж говорити про SELinux, то основне нововведення тут – це режим SELinux Enforcing, що виконується для всіх додатків на пристрої з Lollipop. Чи досить цього, щоб забезпечити надійний захист для Android смартфонів і планшетів? Ділимося думками в коментарях.

Нові принципи роботи з SD картами для Android

Розроблювачі дуже довго чекали цього моменту, і от він настав: в Android додатки по-новому взаємодіють не тільки з різними розділами пристрою, але й зі знімними SD картами. Користувач може призначити папку, що зберігається на SD карті, що буде перебувати в повнім розпорядженні додатка. Це дозволить додатку створювати, зберігати, змінювати й видаляти файли, що зберігаються в цій папці. Такий доступ може бути відкритий до всієї карти, а також до будь-якої іншої папки, що зберігається на пристрої. Це помітно спростить роботу з додатками й дозволить уніфікувати зберігання різного контенту. Розроблювачам же прийде оновити свої додатки, щоб вони могли працювати по новому алгоритмі.

Видалення передвстановлених додатків

В Android зміниться похід до передвстановлених оператором додаткам. Вся справа в тому, що на даний момент їх можна лише приховувати, але так як всі вони розміщені в системному розділі, то видалити їх можна, одержавши root-права. З новою версією операційної системи Android цей принцип повинен істотно помінятися. По-перше, всі додатки, які оператор звичайно ставить на смартфон (від 5 до 20), будуть установлені після того, як користувач активує SIM-карту цього оператора. Так як встановлені вони будуть не в системний розділ, те всі непотрібні додатки можна буде оперативно видалити.

Готове відновлення до Android для смартфонів і планшетів Nexus.

Відновлення до Android незабаром з'явиться на всіх Nexus пристроях, так що власники Nexus 5 і Nexus 7 зможуть прилучитися до нової версії ОС.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		33

10 основних проблем Android і їхнього рішення

На жаль, крім великої кількості приємних доповнень і нововведень в Android існують і численні помилки. Усього основних проблем у новій версії ОС 10, і далі ми розповімо про те, як їх усунути.

Помилка Missing system.img

Перша проблема, з якої можна зштовхнутися при переході на Android – це помилка, що інформує користувача про те, що файл system.img не знайдений. Зштовхнутися з даною проблемою можна лише в тому випадку, якщо ви встановлюєте нову версію операційної системи вручну. Щоб неї вирішити, необхідно встановити всі файли окремо, а не у вигляді єдиного flash-all.bat файлу.

Проблеми з підключення до WiFi

Повністю усунути помилку вдасться тільки розроблявачам разом із черговим відновленням операційної системи. Користувачам же залишається постійно зіштовхуватися із цією проблемою. Щоб підключитися до WiFi, не дивлячись на помилку, потрібно:

- включити/виключити режим польоту;
- спробувати повторно перепідключитися до мережі;
- перезавантажити смартфон;
- перезавантажити роутер;
- перемкнутися між смугами WiFi.

Проблеми з батареєю

Так як Android доводиться встановлювати поверх існуючої ОС, те це може привести до появи таких проблем, як надмірне енергоспоживання. Щоб усунути цю помилку, пристрій необхідно відкотити до заводських налаштувань. Попередньо не забудьте створити резервні копії всіх важливих файлів і даних.

Проблеми з відтворенням відео

Деякі користувачі Nexus 7 (2013) скаржаться на проблеми з відтворенням відео. Помилка актуальна не тільки для YouTube, але й для багатьох інших додатків. Рішень у цієї проблеми трохи, але їхня ефективність варіюється. По-

						ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			34

перше, можна почистити кеш додатка, що видає помилку. По-друге, додаток можна переустановити. По-третє, можна зробити відкіт до заводських налаштувань. Як і у випадку з попередньою проблемою, не забудьте створити важливі бекапи.

Проблеми з OTA відновленнями

Поки проблеми з OTA відновленнями виникали лише у власників Nexus 9. Але не виключено, що дана помилка може вразити й інші пристрої. У цій ситуації рішення всього два: установити офіційні відновлення вручну або дочекатися, поки розроблювачі вирішать проблему.

Проблеми зі спалахом

Іноді використання спалаху приводить до блокування камери й, властиво, самого спалаху. Тут може допомогти тільки перезавантаження. Проблема існує ще з версії ОС, що була випущена спеціально для розроблювачів, але ніхто так і не вирішив її.

Відсутність беззвучного режиму, Power меню й прокручування тексту повідомлень

На жаль, все це нововведення, а не помилка. Чомусь творці Android вирішили відмовитися від цих зручних особливостей. Сподіваємося, що тимчасово.

Проблеми з установкою OTA на модифікованих пристроях

Звичайно ж, ця особливість характерна й всім іншим версія ОС Android. Google не хоче, щоб користувачі масово модифікували свої смартфони й планшети, тому для установки OTA відновлень вам доведеться знову повернутися до всіх заводських обмежень.

Проблеми з перемиканням між додатками

В Android з'явився новий спосіб перемикання між додатками, в основі якого лежить використання карток. У деяких користувачів виникають різні проблеми, вирішити які не допомагає навіть перезавантаження. Якщо раптом ви теж зштовхнулися із чимсь подібним, то знайте, що проблема може бути вирішена тільки разом із черговим відновленням.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Проблема із вкладками Chrome

У новій версії ОС вкладки Chrome відображаються разом з недавніми додатками. Це не помилка, а особливість Android, яку можна дуже просто виправити, повернувши звичний інтерфейс: у налаштуваннях браузера потрібно відключити опцію, що відповідає за об'єднання вкладок з додатками.

3.2 Розробка структурної схеми

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

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

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Технологія функціонування електронної пошти

Розглянемо технологію функціонування електронної пошти.

Послідовність процедур для організації передачі інформації з електронної пошти:

- встановити зв'язок зі своїм хост-смартфоном;
- запустити програму керування електронною поштою (наприклад, клацнути на робочому столі на ярлику поштового клієнта або скористатися командою в головному меню);
- у діалоговому вікні поштового клієнта натиснути кнопку «Створити повідомлення» або виконати команду меню «Повідомлення > Створити»;
- у діалоговому вікні, що з'явилося, у полі «Кому» вказати через крапку з комою доменні адреси (адреса) електронної пошти одержувачів (одержувача);
- у поле введення «Копія» перелічити через крапку з комою адреси електронної пошти всіх одержувачів, кому повинна бути доставлена копія повідомлення;
- у поле «Тема» описати короткий зміст повідомлення;
- у робочій області ввести текст повідомлення;
- вставити в повідомлення електронний підпис, виконавши команду «Вставка > Підпис». Електронний підпис містить звичайно ПІБ відправника, його посаду, назву фірми, телефон і т.д., але може містити в собі й оригінальний придуманий автором текст. Електронний підпис підготовляється заздалегідь і зберігається в папці. Створити її простіше всього по команді «Сервіс > Параметри», і на вкладці «Підпис» натиснути кнопку «Створити»;
- якщо потрібно, прикріпити файл вкладення за допомогою команди «Вставка > Вкладення» файлу або натисканням відповідної кнопки на панелі інструментів з наступним вибором вкладеного файлу. Файл-вкладення може бути будь-яким створеним заздалегідь файлом (текстовим, графічним, що виконується й т.д.);

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

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

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

Через кілька секунд ваше повідомлення надійде хост-смартфону одержувача по зазначеній адресі.

Адреси всіх одержувачів можуть бути уведені вручну, або взяті з адресної книги. Щоб скористатися наявною адресною книгою, варто виконати команду «Сервіс > Вибрати одержувачів». У вікні, що з'явилося, зі списку вибрати потрібних одержувачів, супроводжуючи вибір кожного з них для вказівки категорії одержувачів наступним натисканням клавіші: «Кому», «Копія», «Схована копія». До категорії «Схована копія» варто віднести тих одержувачів, відсилення дубліката в чюю адресу потрібно зберегти в таємниці.

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

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

– встановити зв'язок зі своїм хост-смартфоном;

– запустити програму керування електронною поштою (наприклад, клацнути на робочому столі на ярлику поштового клієнта або скористатися командою в головному меню);

– у діалоговому вікні відкрити папку «Вхідні». У цій папці перебувають всі повідомлення, що надійшли на адресу користувача.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

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

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

Послідовність дій для відповіді на отримане повідомлення:

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

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

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

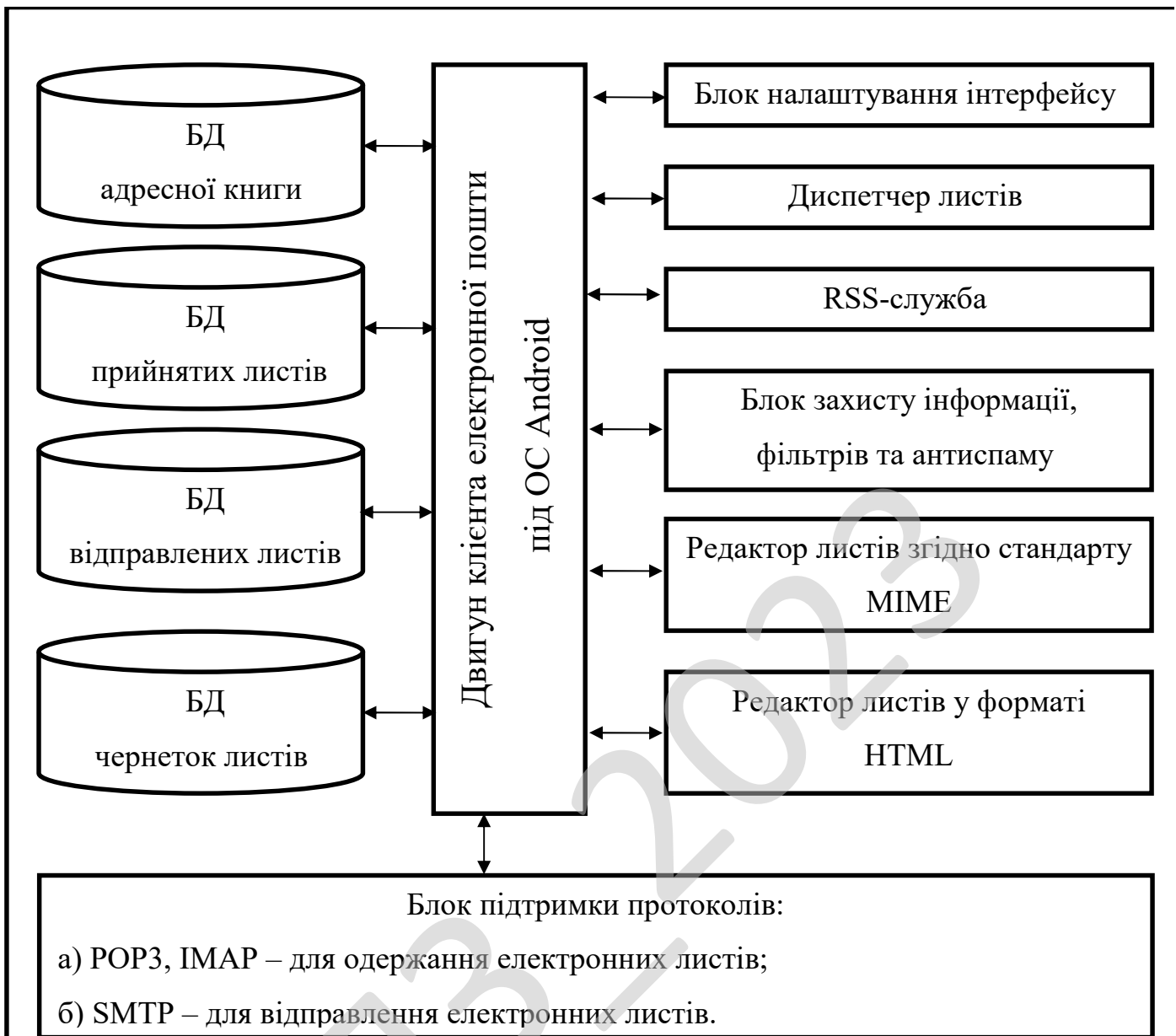


Рисунок 3.1 – Структурна схема розробленої системи

Програмне забезпечення поштового клієнту складається з наступних структурних блоків:

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

– Блок підтримки протоколів:

а) POP3, ІМАР – для одержання електронних листів;

б) SMTP – для відправлення електронних листів.

– БД адресної книги – призначена для зберігання адрес куди відправлені

листи, та звідкіля отримані листи.

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

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

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

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

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

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

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

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

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

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

Структура поштового повідомлення

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

– кодового конверта;

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

– заголовок (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.).

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

Тіло повідомлення

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

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

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

– графічні зображення – файли у форматі JPEG і GIF;

– аудіо й відео дані;

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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

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

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

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

Заголовок повідомлення

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

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

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

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

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

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

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

– 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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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.

3.3 Розробка функціональної схеми

Базові функції поштового клієнта

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

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

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

– Прийом повідомлень і автономний перегляд. Це основна функція поштового клієнта. При підключенні до сервера POP3 виробляється автоматичне копіювання всіх повідомлень, що надійшли, у базу даних поштового клієнта,

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

після чого їх можна читати в автономному режимі (тобто, відключившись від мережі).

– Створення нових повідомлень. Це друга найважливіша функція поштового клієнта. Для її реалізації програма може мати вбудований текстовий редактор. Функціональність редактора залежить від властивостей конкретної програми. Для поштового клієнта, що працюють в операційних системах Windows, загальноприйнятою вважається підтримка операцій з використанням буфера обміну й можливість вибору кодування тексту.

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

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

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

– Операції із вкладеними файлами. По своїй природі електронна пошта призначена для пересилання чисто текстових повідомлень, переважно виконаних

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

англійською мовою, тобто символами, що входять у стандарт ASCII. Відмінна риса цих символів полягає в тому, що їхні коди не перевищують значення 127. У той же час, у довільних файлах (програмних, архівних, мультимедійних, у файлах форматуваних документів і інших) можуть із рівною ймовірністю зустрічатися й байти зі значеннями від 128 до 255. Для пересилання їх разом з повідомленням електронної пошти потрібне спеціальне перетворення, у результаті якого символи з кодуванням від 128 до 255, заміняються комбінаціями символів з кодуванням від 0 до 127. На стороні адресата поштової клієнт виконує зворотне перетворення. Такий механізм пересилання довільних файлів одержав назву поштових вкладень. Підготувавши повідомлення електронної пошти, відправник вказує місце розташування файлу, що повинен бути приєднаний. При цьому поштової клієнт робить відповідний запис в одному з полів заголовка повідомлення, де вказує ім'я файлу й використаний метод кодування. Основними методами кодування є: MIME, BinHex і UUEncode. Поштової клієнт здатний розуміти всі ці методи. При прийманні повідомлення поштової клієнт дозволяє витягти поштове вкладення й зберегти його в заданому місці у вигляді автономного файлу.

Додаткові функції поштової клієнта

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

– Підтримка множинних ідентифікаційних записів. Ідентифікаційним записом називається сукупність налаштувань програми на конкретного користувача. Якщо із програмою працюють кілька людей, то деякі поштової клієнти дозволяють кожному з них створити власний ідентифікаційний запис і, тим самим, одержати власний комплект логічних папок. Передбачається, що при підключенні до поштової сервера відбувається завантаження тільки тих повідомлень електронної пошти, які адресовані власникові поточного ідентифікаційного запису. При зміні користувача відбувається зміна

						ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			47

ідентифікаційного запису, і тоді можливе копіювання інших повідомлень, що надійшли. З міркувань безпеки операція зміни ідентифікаційного запису може супроводжуватися запитом пароля.

– Підтримка множинних облікових записів. Обліковим записом називається сукупність налаштувань на конкретного постачальника послуг електронної пошти. Якщо користувач має декілька «поштових скриньок», відкритих на різних серверах, то деякі поштові клієнти дозволяють вибрати поточний обліковий запис і легко перемикаються між ними.

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

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

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

Стандартними функціями роботи з адресною книгою є:

а) автоматизоване створення контакту на підставі даних, узятих з полів заголовка повідомлення, що надійшло;

б) автоматизоване створення шаблону нового повідомлення після вибору потрібного контакту;

в) упорядкування списку контактів шляхом угрупування й сортування.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

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

– Функції оповіщення. Ці функції звичайно реалізуються вбудованою додатковою системою, призначеної для контролю стану облікових записів на серверах POP3 і Web-Mail. Розрізняють мережні й автономні засоби оповіщення. Мережні засоби працюють під час сеансів зв'язку з Інтернетом і займаються періодичним опитуванням облікових записів. Автономні засоби оповіщення працюють за заданим розкладом, при цьому вони можуть самостійно встановлювати з'єднання віддаленого доступу й після перевірки облікових записів розривати його. Найбільш розвинені засоби оповіщення можуть аналізувати заголовки повідомлень, що надійшли, і переадресовувати повідомлення на іншу адресу електронної пошти (той, де в цей момент перебуває власник облікового запису) або на інші засоби зв'язку (звичайний або стільниковий телефон, пейджер, факс-машину). Деякі програми при цьому здатні відтворити прийнятий текст голосом. Як сигнал оповіщення може використовуватися звуковий або візуальний сигнал (діалогове вікно). Більшість засобів оповіщення можуть сигналізувати про надходження нової пошти запуском заданої програми, а деякі й досить екзотичними засобами, наприклад миготінням світлодіодів клавіатури.

– Засоби керування «поштовою скринькою». Характерна відмінність даних, що втримуються в полях заголовка, від даних «тіла» повідомлення полягає в їхньому невеликому розмірі. У тих випадках, коли щоденний обсяг вступних повідомлень значний, доцільно використовувати поштового клієнта, що володіє

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

функціями попереднього перегляду полів заголовків до завантаження повідомлень із сервера. Це дозволяє відмовитися від завантаження невикликаної кореспонденції, уникнути прийому небажаних файлів поштових вкладень, видалити повідомлення, що мають незаповнені поля. Всі операції відбору й видалення виконуються безпосередньо на сервері, що знижує навантаження на канали зв'язку й зменшує витрати на їхню експлуатацію.

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

– Підтримка «чорного» і «білого» списків. Засоби фільтрації можуть працювати із заздалегідь заготовленими списками поштових адрес. «Чорним» називається список адрес електронної пошти, повідомлення від яких автоматично блокуються й знищуються безпосередньо на сервері без завантаження на локальний смартфон. «Білий список» використовують, щоб пропускати вибрані повідомлення в тих випадках, коли поштовий клієнт настроєний на блокування

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

всіх вступні повідомлень. Слід зазначити, що «чорний» список може бути розподілений. Так, наприклад, сьогодні в Інтернеті є служби, що ведуть облік відомих спаммерів і рекламних служб, що порушують етикет електронної пошти. Поштовий клієнт може автоматично звіряти адреси відправників вхідних повідомлень із даними цих мережних служб і блокувати вступник кореспонденцію на підставі їхніх рекомендацій.

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

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

– Функції автоматичної генерації відповіді й переадресації. Автоматична генерація відповіді на поштове повідомлення, що надійшло, дозволяє дотримати етикету електронної пошти й оперативно відповісти на повідомлення, що надійшло, навіть перебуваючи в тривалій відсутності. Звичайно на всі вступні повідомлення генерується стандартна відповідь із викладом причин тимчасової відсутності зв'язку. Якщо при цьому використовувати описані вище функції

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

фільтрації, то можна класифікувати вхідні повідомлення по ім'ю відправника або темі повідомлення й генерувати різні автовідповіді на різні звернення.

– Ті, хто багато часу проводять у роз'їздах, поряд з функцією автовідповіді користуються також функцією автоматичної переадресації. Загальноприйнята практика складається в створенні тимчасового «поштової скриньки» на сервері однієї зі служб Web-Mail і переадресації на нього кореспонденції, що надходить на облікові записи серверів POP3. Оскільки для роботи з електронною поштою, заснованої на Web, не потрібно ні наявності поштового клієнта, ні його налаштування, робота з тимчасовим «поштовою скринькою» виробляється за допомогою звичайного броузера з будь-якого місця зв'язку. Де б не перебував користувач, він завжди має можливість переглянути всю вступник електронну пошту на одному легкодоступному Web-вузлі.

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

Спеціальні функції засобів електронної пошти

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

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

роздільником, наприклад крапкою або символом табуляції. Розсилання повідомлення виробляється по всіх адресах, наведеним у списку. Додаткові засоби програми дозволяють також:

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

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

– Функції локального SMTP-сервера. Проведення масового відправлення кореспонденції через SMTP-сервер постачальника послуг сполучено з великими

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

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

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

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

Розглянемо більш докладно основні функціональні можливості розробленого поштового клієнту.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

До основних функцій розробленого поштового клієнта відносяться наступні:

- Адресна книга.
- 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.
- Імпорт з інших поштових клієнтів.
- Імпорт адресних книг з файлів.

Розглянемо більш докладно вищеперераховані функції.

Адресна книга

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

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.

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

оцініте ці шаблони. У швидких шаблонах, так само як і у звичайних, доступний повний набір макросів, що дозволяє задати будь-яку, навіть саму складну послідовність дій.

Списки розсилання

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

Безпека особистої інформації

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

Захист від вірусів

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

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

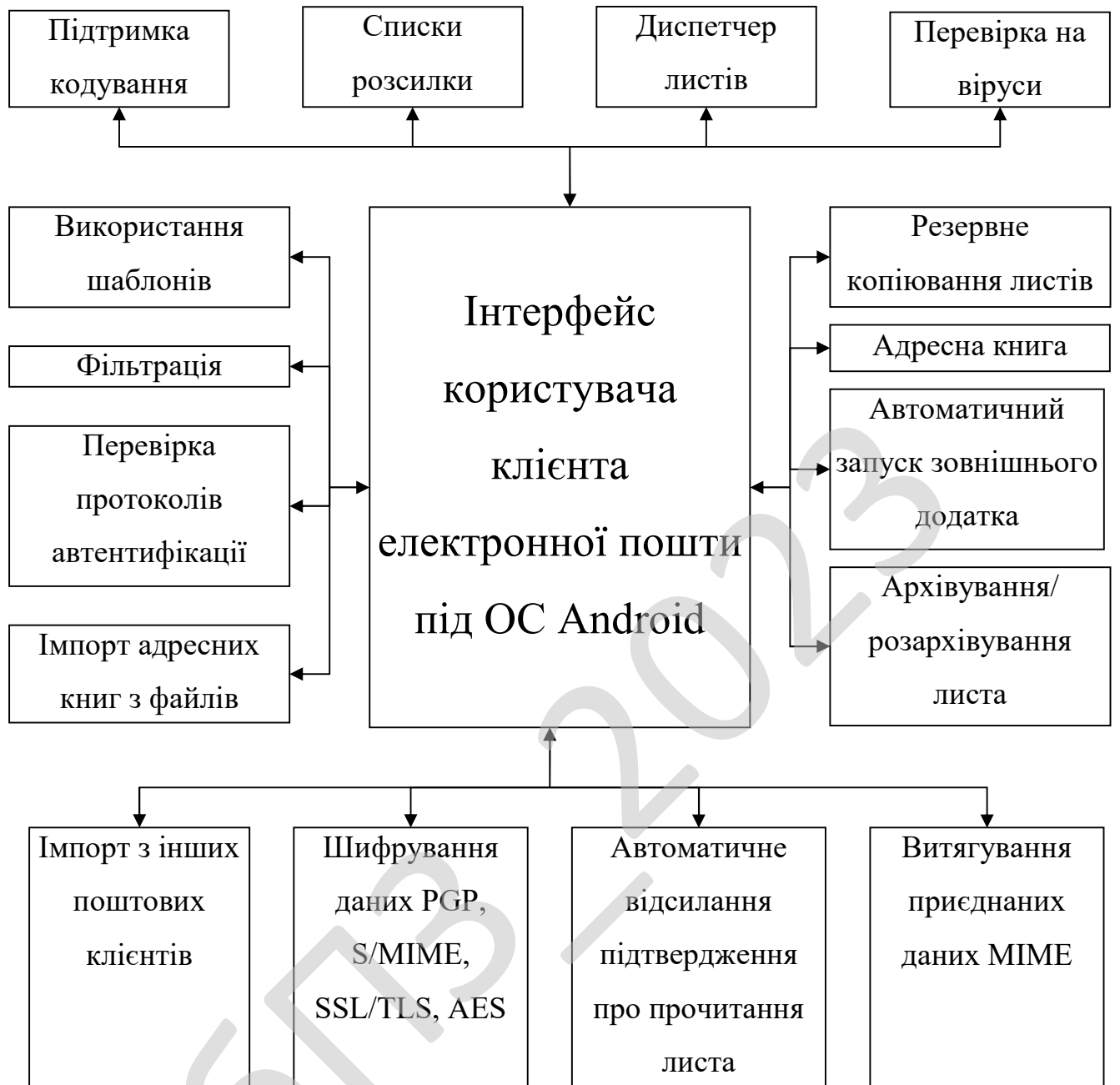


Рисунок 3.2 – Функціональна схема системи

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 перекидання листів з папки в папку.

Унікальна функція паркування

За допомогою паркування ви зможете закріпити лист за певною папкою й зберігати його від випадкового видалення або переміщення. Ви ніколи не втратите важливий лист, якщо збережете 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) Android – ви зможете оновлювати компоненти Windows, необхідні для роботи поштового клієнта, не перезавантажуючи ні поштовий клієнт, ні Windows. Ніколи ще процес відновлення не був таким простим і непомітним. Що стосується можливостей установки, системні адміністратори (і не тільки вони) по достоїнству оцінять нову можливість “непомітної” установки, що дозволяє інсталиувати поштовий клієнт без участі користувача.

Ефективне шифрування

Внутрішнє виконання PGP, засноване на загальноновизнаному механізмі OpenSSL, дозволяє вам шифрувати повідомлення й підписувати їхнім цифровим підписом. Зручний менеджер цифрових ключів додається. Безкоштовні доповнення до PGP v5.5, v6.0.2, 6.5, v7.x & v8.x GnuPG також підтримуються. Крім того, 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.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

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.

Багаті можливості імпорту

Поштовий клієнт дозволяє імпортувати повідомлення з інших поштових клієнтів:

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

Імпорт адресних книг з файлів:

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

Працює з будь-якою системою Windows.

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.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

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.

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

3.4 Розробка діаграми процесів

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

- Виведення вікна ПЗ на моб. пристрої.
- Створення листа з врахуванням особливостей версії ОС Android.
- З'єднання поштового клієнта з SMTP-сервером.
- Передача SMTP-серверу інформації про одержувача і відправника.
- Перевірка SMTP-сервером валідності даних й прийняття листа.
- DNS запит про поштові сервери для домену адресата.
- Спроба з'єднання з поштовим сервером.
- Спроба передати лист на резервні поштові сервера домену.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

- Передача листа.
- Спроба переслати лист пізніше, у випадку невдачі.
- З'єднання адресата з POP3 або IMAP сервером (відповідно до налаштувань).
- Автентифікація й одержання листа адресатом.

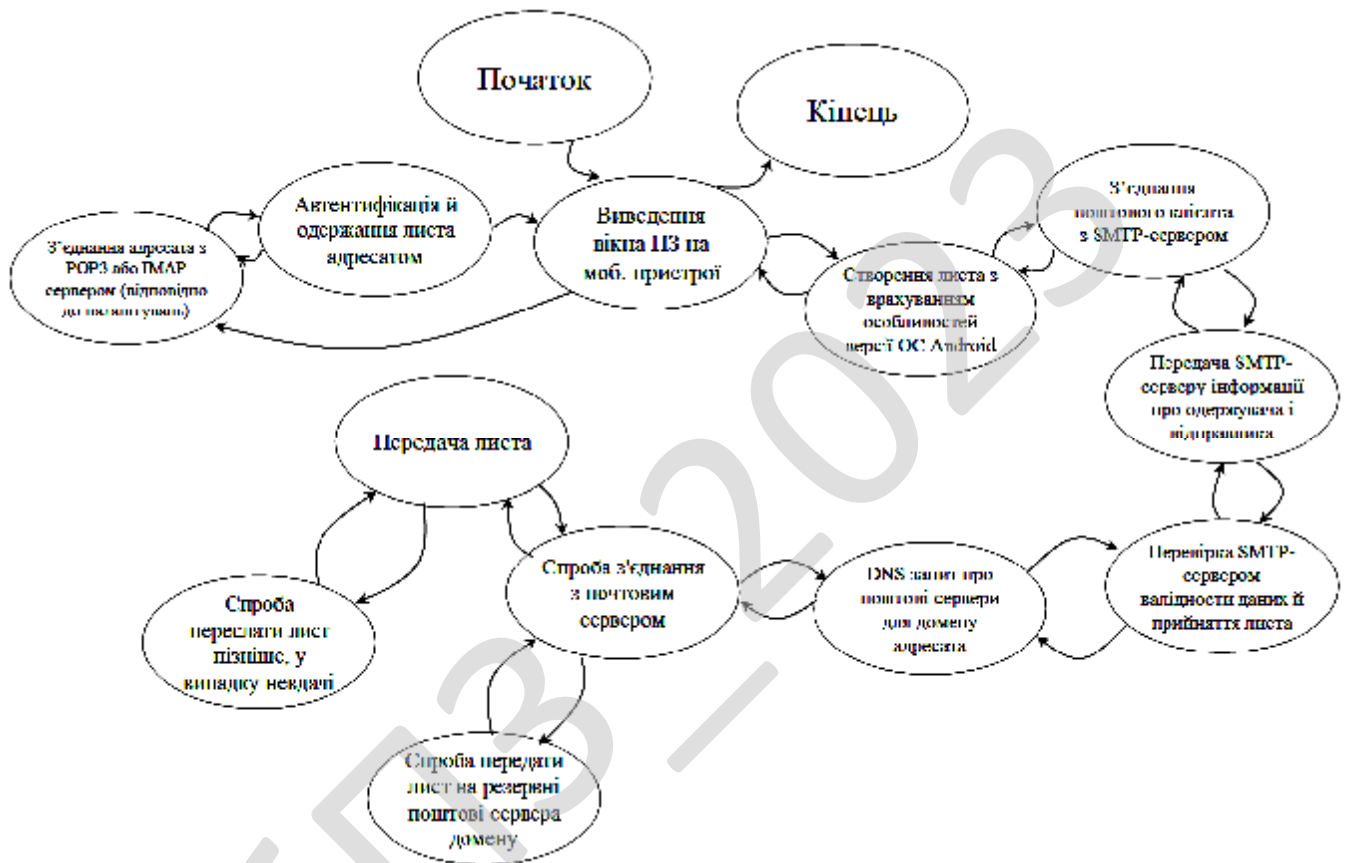


Рисунок 3.3 – Діаграма взаємодії процесів

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

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

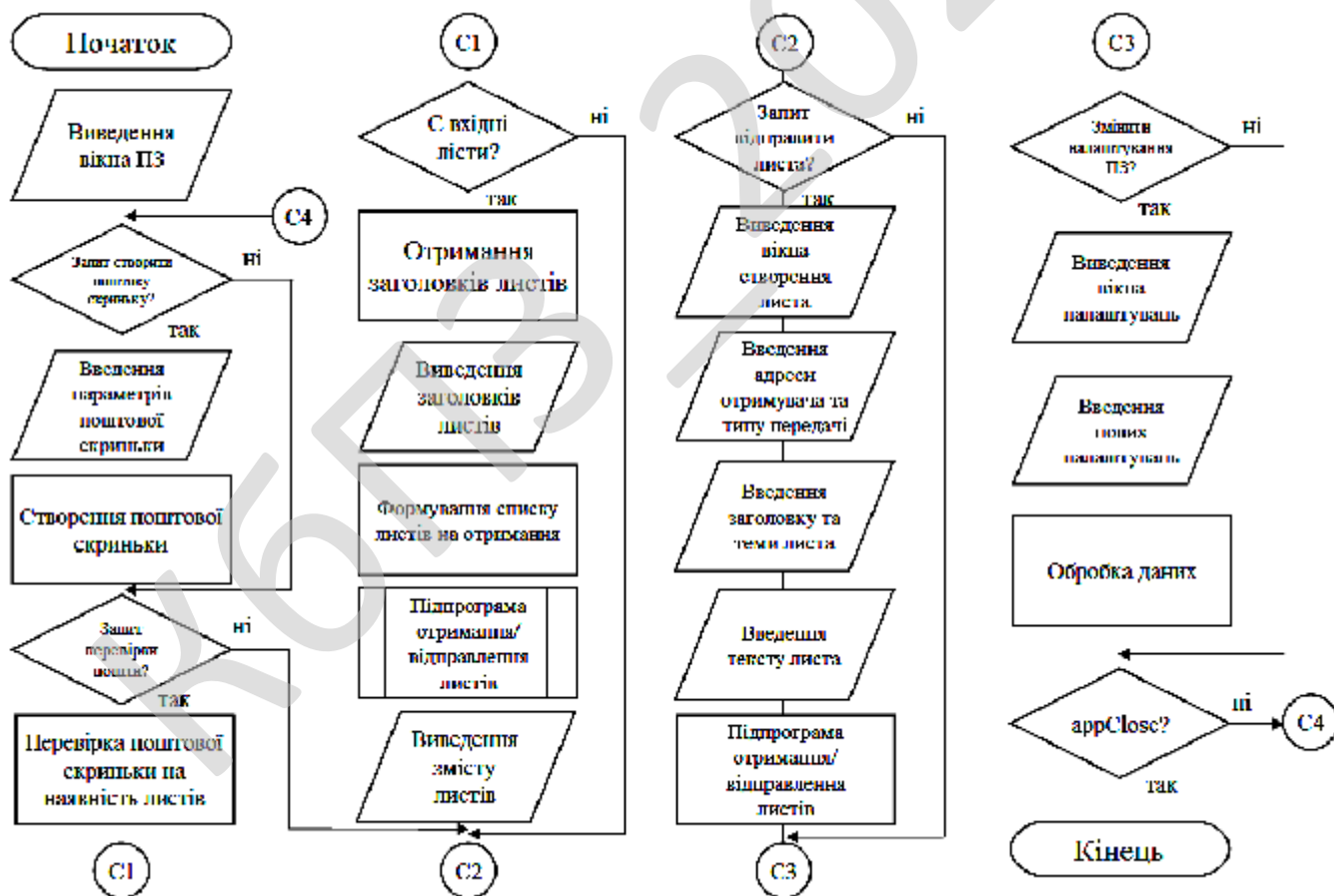


Рисунок 4.1 – Блок схема основної програми

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

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

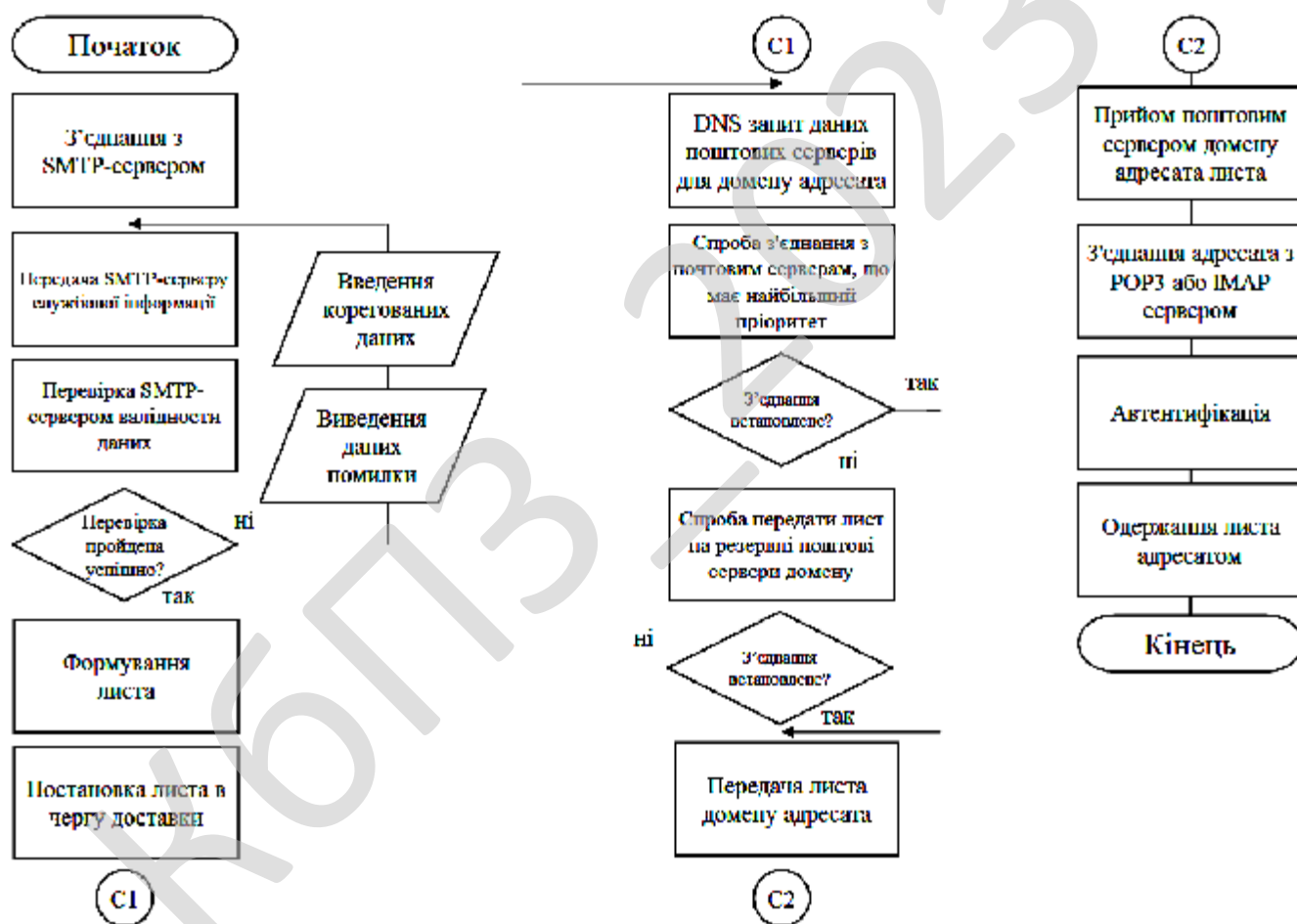


Рисунок 4.2 – Блок схема підпрограми

При розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльності.

Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Опис алгоритмів функціонування системи

Для роботи з мережею, в Delphi використовуються Indy компоненти, які можна знайти на декількох вкладках середовища розробки.

Знайдіть на вкладці Indy Clients – компонент IdPOP3, на вкладці Indy Misc – компонент IdMessage. За допомогою компонента IdPOP3 ми зможемо здійснити підключення до поштового сервера за протоколом POP3 і одержати необхідне нам кіл-у повідомлень.

Компонент IdMessage буде використовуватися, як буфер для одержуваного листа. Ще нам буде потрібно тето для відображення тексту листа, і кнопка для завантаження чергового повідомлення.

Одержання листа з атрибутами.

```
procedure TForm1.Button3Click(Sender: TObject);  
var  
i,numPosts: Integer;  
begin  
POP3.Host:='mail.ua';  
// адреса поштового сервера
```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

```

POP3.Port:=110;
// порт по якому буде здійснюватися підключення
POP3.Username:='test+58r.ua';
// логін користувача
POP3.Password:='12345666?';
// пароль користувача
IdMessage.Clear;
// очищення буфера для повідомлення
POP3.Connect;
// підключення за протоколом POP3, по налаштуваннях
// у компоненті IdPOP3, з ім'ям POP3
Memo1.Clear;
// очищення компонента мемо для відображення тексту листа.
POP3.Retrieve(1,IdMessage);
// одержання одного повідомлення
Memo1.Lines.AddStrings(IdMessage.Body);
// передача повідомлення з компонента IdMessage в мемо
// вивід у компоненти Label інформації про повідомлення.
Label1.Caption := IdMessage.From.Text;
Label2.Caption := IdMessage.Recipients.EmailAddresses;
Label3.Caption := IdMessage.CCList.EMailAddresses;
Label4.Caption := IdMessage.Subject;
Label5.Caption := FormatDateTime('dd mmm yyyy hh:mm:ss', IdMessage.Date);
Label6.Caption := IdMessage.ReceiptRecipient.Text;
Label7.Caption := IdMessage.Organization;
POP3.Delete(1);
// видалення із сервера, отриманого повідомлення
POP3.Disconnect;
// розрив зв'язку
end;

```

Відправлення пошти здійснюється за протоколом SMTP, компонент IdSMTP, ви зможете знайти його на вкладці Indy Clients.

У даному прикладі відправлення пошти буде здійснюватися після авторизації сервері, лист відправляється з майла test@58r.ua, на майл test@58r.ua .

Відправлення одного листа.

```

procedure TForm1.Button4Click(Sender: TObject);
begin
SMTP.Host:='mail.58r.ua';
// адреса поштового сервера
SMTP.Port:=25;

```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

```

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

```

Опис алгоритму одержання одного листа.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
POP3.Host:='mail.58r.ua';
// адреса поштового сервера
POP3.Port:=110;
// порт по якому буде здійснюватися підключення
POP3.Username:='test+58r.ua';
// Логін користувача
POP3.Password:='12345666?';
// пароль користувача
IdMessage.Clear;

```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

```

// очищення буфера для повідомлення
POP3.Connect;
// підключення за протоколом POP3, по налаштуваннях
// у компоненті IdPOP3, з ім'ям POP3
Memo1.Clear;
// очищення компонента мемо для відображення тексту листа.
POP3.Retrieve(1,IdMessage);
// одержання одного повідомлення
Memo1.Lines.AddStrings(IdMessage.Body);
// передача повідомлення з компонента IdMessage в мемо
POP3.Delete(1);
// видалення із сервера, отриманого повідомлення
POP3.Disconnect;
// розрив зв'язку
end;

```

Звичайно потрібно завантажити кілька повідомлень, всі що є на сервері. Було вирішено обмежитися 10 листами, щоб не ускладнювати код.

Одержувані повідомлення зберігаються в тій же папці що й поштова програма. Одержання 10 листів.

```

procedure TForm1.Button2Click(Sender: TObject);
label f;
var
mailcicl:integer;
// лічильник одержуваних листів
addr:string;
// папка в якій будуть зберігатися одержувані листи
begin
POP3.Host:='mail.58r.ua';
// адреса поштового сервера
POP3.Port:=110;
// порт по якому буде здійснюватися підключення
POP3.Username:='test+58r.ua';
// Логін користувача
POP3.Password:='12345666?';
// пароль користувача
addr:=ExtractFilePath(Application.ExeName);
// одержання адреси папки, у якій перебувати наша програма
POP3.Connect;
// підключення до сервера
for mailcicl:=1 to 10 do

```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

```

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

```

Кожне повідомлення має атрибути – це майл відправника, тема повідомлення, важливість і інші елементи. Вивчіте компонент IdMessage, щоб знати всі можливі атрибути. Відправлення листа й вказівка теми, тексту повідомлення.

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

1. У розділі uses підключите ShellAPI.
2. В оброблювачі OnClick мітки або клавіші введіть наступний код.

```

ShellExecute(Handle, 'open', 'mailto:lalala@lala.ua?par1=value1&par1=
value1&...', nil, nil, SW_SHOWNORMAL);

```

Третій параметр функції ShellExecute() – це рядок відповідно до протоколу mailto і правилами оформлення URL:

– mailto: тип протоколу (може бути http: у цьому випадку оставшая URL і параметри запиту).

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

– lalala@lala.ua – адреса одержувача, можна включати кілька адрес, розділяючи із символом ";".

– ? – роздільник параметрів від адреси.

– par1=value1 – ім'я параметра і його значення.

– & – роздільник параметрів.

Протокол Mailto має наступну форму.

MAILTO:Recipients&Parameters

– Поле Recipients може бути порожньою, одиночною адресою й складатися з декількох адрес, розділених символом ";"

– Поле Parameters додатково і якщо воно є те повинне бути відділене символом "&". Параметри повинні з'являтися у формі пари name/value. Наступний список описує можливі параметри:

PARAMETER DESCRIPTION:

– CC = Carbon copy (додаткові одержувачі).

– BCC = Blind carbon copy (додаткові одержувачі, адреси яких не показуються іншим одержувачам).

– SUBJECT = Subject text (тема).

– BODY = Body text (текст).

Всі дані вказуються в параметрах повинні бути так звані Internet safe characters. Використовуйте %0d для символу переклад рядка (LF), %20 для пробілу й так далі.

Приклад:

```
mailto:email1;email2&cc=email3?subject=Це%20тема&
body=це%20текст%20листа%0dце%20інший%20рядок
```

Відправлення листа потай від користувача за протоколом SMTP.

```
{
dest - адреса одержувача
subject - тема листа
body - текст листа
smtp - ip адреса smtp сервера
port - порт smtp сервера, за замовчуванням 25
from - адреса відправника
```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Повертає True якщо лист був успішно відправлено...

```
}  
  
function mail(smtp: string; port: integer; from, dest, subject,  
  body: string): bool;  
const  
  cl = #13#10;  
var  
WSAData: TWSAData;  
  Host: TSocketAddrIn;  
  Sock: TSocket;  
  res: Integer;  
  buff: array[1..255] of Char;  
  { відправляємо дані через сокет }  
  procedure senddata(str: string);  
  var  
    i: integer;  
  begin  
    for i := 1 to Length(str) do  
      if send(Sock, str[i], 1, 0) = SOCKET_ERROR then  
        exit;  
  end;  
  { одержуємо відповідь від команди }  
  function recvdata(accept: string): bool;  
  var  
    buff: array[1..255] of Char;  
  begin  
    res := recv(Sock, buff, SizeOf(buff), 0);  
    Result := (Res = SOCKET_ERROR) or (Copy(buff, 1, 3) = accept);  
  end;  
begin  
  try  
    result := false;  
    { ініціалізація сокета }  
    WSASStartUp(257, WSAData);  
    Sock := socket(AF_INET, SOCK_STREAM, IPPROTO_IP);  
    if Sock = INVALID_SOCKET then  
      Exit;  
    { встановлюємо хост і порт сервера }  
    res := inet_addr(PChar(smtp));  
    if res <= 0 then  
      exit;  
    Host.sin_family := AF_INET;
```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		71

```

Host.sin_port := htons(port);
Host.sin_addr.S_addr := res;
{ підключаємося до сервера }
if connect(Sock, Host, SizeOf(Host)) > 0 then
    Exit;
{ вітання сервера }
if not recvdata('220') then
    Exit;
{ EHLO }
senddata('EHLO' + cl);
if not recvdata('250') then
    Exit;
{ MAIL FROM: }
senddata('MAIL FROM:' + from + cl);
if not recvdata('250') then
    Exit;
{ RCPT TO: }
senddata('RCPT TO:' + dest + cl);
if not recvdata('250') then
    Exit;
{ DATA }
senddata('DATA' + cl);
if not recvdata('354') then
    Exit;
{ відправляємо текст повідомлення }
senddata('Subject:' + subject + cl + cl + body + cl + '.');
if not recvdata('250') then
    Exit;
{ відключаємося від сервера }
senddata('QUIT' + cl);
result := true;
finally
    { убиваємо сокет }
    closesocket(sock);
    WSACleanup;
end;
end;

```

Опис перевірки правильності E-mail адреси.

```

function IsValidEmail(const Value:string):boolean;
    function CheckAllowed(const s:string):boolean;
    var

```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

```

    i: integer;
begin
    Result:= false;
    for i:= 1 to Length(s) do
    begin
        if not (s[i] in ['a'..'z', 'A'..'Z', '0'..'9', '_', '-', '.']) then
            Exit;
        end;
        Result:= true;
    end;
end;
var
    i: integer;
    namePart, serverPart: string;
begin
    Result:= false;
    i:= Pos('@', Value);
    if i = 0 then
        Exit;
    namePart:= Copy(Value, 1, i - 1);
    serverPart:= Copy(Value, i + 1, Length(Value));
    if (Length(namePart) = 0) or ((Length(serverPart) < 5)) then
        Exit;
    i:= Pos('.', serverPart);
    if (i = 0) or (i > (Length(serverPart) - 2)) then
        Exit;
    Result:= CheckAllowed(namePart) and CheckAllowed(serverPart);
end;

```

Опис обміну інформацією між програмами в мережі по поштових каналах.

Як реалізувати обмін інформацією між додатками в мережі – ОС Android пропонує кілька технологій. Опишемо один дуже простий і надійний спосіб.

Обмін текстовими даними в локальній мережі дуже простий. Для цього необхідні функції:

- GetMailslotInfo – визначення наявності повідомлення в каналі;
- ReadFile – читання повідомлення з каналу, як з файлу;
- CreateMailslot – створення поштового каналу;
- WriteFile – запис повідомлення в канал, як у файл.

Функції роботи з поштовими каналами присутні як в Windows 7, так і в ОС Android. Розглянемо створення поштового каналу (сервер).

```
//... створення каналу з ім'ям MailSlotName - по цьому ім'ю до нього
// будуть звертатися клієнти
h := CreateMailSlot(PChar('\\.\mailslot\' + MailSlotName),
0, MAILSLLOT_WAIT_FOREVER,nil);
if h = INVALID_HANDLE_VALUE then
begin
    raise Exception.Create('MailSlotServer: Помилка створення каналу !');
// Відправлення повідомлень по поштовому каналі (клієнту)
if not GetMailSlotInfo(h,nil,DWORD(MsgNext),@MsgNumber,nil) then
begin
    raise Exception.Create('TglMailSlotServer: Помилка збору інформації!');
end;
if MsgNext <> MAILSLLOT_NO_MESSAGE then
begin
    beep;

// читання повідомлення з каналу й додавання в текст протоколу
if ReadFile(h,str,200,DWORD(read),nil) then
    MessageText := str
else
    raise Exception.Create('TglMailSlotServer: Помилка читання
    повідомлення !');
end;
Всі дуже просто. Тепер для зручності використання створимо два компоненти -
клієнт і сервер

unit glMSlots;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls,
    Forms, Dialogs, extctrls;
type
TOnNewMessage = procedure (Sender: TObject; MessageText: string) of object;
TglMailSlotServer = class(TComponent)
private
    FMailSlotName, FLastMessage: string;
    FOnNewMessage: TOnNewMessage;
    Timer: TTimer;
//...таймер для прослуховування каналу
```

```

    h : THandle;
    str : string[250];
    MsgNumber,MsgNext,read : DWORD;
public
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
    procedure Open;
//...створення каналу
    procedure Close;
//...закриття каналу
    protected
        procedure Loaded; override;
        procedure OnTimer(Sender: TObject);
    published
        property MailSlotName: string read FMailSlotName write FMailSlotName;
//...подія одержання повідомлення
    property OnNewMessage: TOnNewMessage read FOnNewMessage write
FOnNewMessage;
    end;
    TglMailSlotClient = class(TComponent)
    private
        FMailSlotName, FServerName, FLastMessage: string;
    public
        constructor Create(AOwner: TComponent); override;
        destructor Destroy; override;
        function Send(str: string):boolean;
//...відправлення повідомлення
    protected
        procedure Loaded; override;
        procedure ErrorCatch(Sender : TObject; Exc : Exception);
    published
        property ServerName: string read FServerName write FServerName;
        property MailSlotName: string read FMailSlotName write FMailSlotName;
    end;

    procedure register;
implementation

    procedure register;
begin
    RegisterComponents('Gl Components', [TglMailSlotServer,TglMailSlotClient]);
end;

```

					ВКРМ-123.23.0025.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		75

```

constructor TglMailSlotServer.Create(AOwner: TComponent);
begin
    inherited;
    FEnabled := true;
    FMailSlotName := 'MailSlot';
    Timer := TTimer.Create(nil);
    Timer.Enabled := false;
    Timer.OnTimer := OnTimer;
end;

destructor TglMailSlotServer.Destroy;
begin
    Timer.Free;

    // закриття каналу
    Close;
    inherited;
end;

procedure TglMailSlotServer.Loaded;
begin
    inherited;
    Open;
end;

function TglMailSlotClient.Send(str: string):boolean;
var
    strMess: string[250];
    UserName: array[0..99] of char;
    h: THandle;
    i: integer;
begin
    // відкриття каналу : MyServer - ім'я сервера
    // (\\.\mailslot\xxx - монітор працює на цьому ж ПК)
    // xxx - ім'я каналу
    if FServerName = '' then
        FServerName := '.';
    h:=CreateFile( PChar('\\' + FServerName + '\mailslot\' + FMailSlotName),
        GENERIC_WRITE, FILE_SHARE_READ, nil, OPEN_EXISTING, 0, 0);
    if h <> INVALID_HANDLE_VALUE then
        begin
            strMess := str;

            // передача тексту помилки (запис у канал і закриття каналу)

```

						ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			76

```

        WriteFile(h, strMess, Length(strMess) + 1, DWORD(i), nil);
        CloseHandle(h);
    end;

    Result := h <> INVALID_HANDLE_VALUE;
end;

procedure TglMailSlotServer.Open;
begin
    // створення каналу з ім'ям MailSlotName - по цьому імені до нього
    // будуть звертатися клієнти
    h := CreateMailSlot(PChar('\\.\mailslot\' + MailSlotName),
        0, MAILSLOT_WAIT_FOREVER, nil);
    //h:=CreateMailSlot('\\.\mailslot\MailSlot', 0, MAILSLOT_WAIT_FOREVER, nil);
    if h = INVALID_HANDLE_VALUE then
        raise Exception.Create('TglMailSlotServer: Помилка створення каналу !');
    Timer.Enabled := true;
end;

procedure TglMailSlotServer.Close;
begin
    if h <> 0 then
        CloseHandle(h);
    h := 0;
end;

procedure TglMailSlotServer.OnTimer(Sender: TObject);
var
    MessageText: string;
begin
    MessageText := '';
    // визначення наявності повідомлення в каналі
    if not GetMailSlotInfo(h, nil, DWORD(MsgNext), @MsgNumber, nil) then
        raise Exception.Create('TglMailSlotServer: Помилка збору інформації!');
    if MsgNext <> MAILSLOT_NO_MESSAGE then
        begin
            beep;
            // читання повідомлення з каналу й додавання в текст протоколу
            if ReadFile(h, str, 200, DWORD(read), nil) then
                MessageText := str
            else
                raise Exception.Create('TglMailSlotServer:Помилка читання повідомлення !');
        end;
        if (MessageText<>'') and Assigned(OnNewMessage) then
            OnNewMessage(self, MessageText);
    end;
end;

```

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

```

    FLastMessage := MessageText;
end;

constructor TglMailSlotClient.Create(AOwner: TComponent);
begin
    inherited;
    FMailSlotName := 'MailSlot';
    FServerName := '';
end;

destructor TglMailSlotClient.Destroy;
begin
    inherited;
end;

procedure TglMailSlotClient.Loaded;
begin
    inherited;
    Application.OnException := ErrorCatch;
end;

procedure TglMailSlotClient.ErrorCatch(Sender : TObject; Exc : Exception);
var
    UserName: array[0..99] of char;
    i: integer;
begin
    // одержання ім'я користувача
    i:=SizeOf(UserName);
    GetUserName(UserName, DWORD(i));
    Send('/'+UserName+'/' + FormatDateTime ('hh:mm', Time) + '/' + Exc.message);
    // вивід повідомлення про помилку користувачеві
    Application.ShowException(Exc);
end;
end.

```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Sinople – симетричний блоковий криптоалгоритм, побудований на основі незбалансованої «мережі Фейстеля». Алгоритм розроблено у 2003 році.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Основні вимоги до алгоритму при його розробці:

- Можливість програмної і апаратної реалізації.
- Висока швидкість.
- Простота.
- Низькі вимоги до пам'яті.
- Високий рівень безпеки.

Алгоритм заснований на 32-розрядних операціях і має 64 раунду, серед яких два типи – С і D. D раунди спроектовані для досягнення максимальної дифузії, С раунди – для досягнення перемішування. F-функція D раунду використовує один з елементів блоку даних ($D[3]$) та поточного з'єднання ($K[r]$) для трансформації 3-х елементів блоку даних. F-функція С раунду, навпаки, використовує перші три елемента блоку даних і поточний з'єднання ($K[r]$) для трансформації останнього елемента блоку даних ($D[3]$). Раунди D-типу виконуються до раундів С-типу. Додавання ключів з даними проводиться тільки через таблиці замін. Операції XOR (додавання по модулю 2) обов'язково поєднуються з операціями ADD (додавання по модулю 2^{32}).

Таблиці замін спочатку запозичені з алгоритму MARS і містять 512 32-розрядних елементів, проте були жорстко проаналізовані на предмет посилення.

Ключове розклад було спроектовано з урахуванням вимог:

- Простота
- Використовується та ж процедура, що і при шифруванні та розшифруванні
- Установка ключа займає менше часу, ніж зашифрування
- Виключення еквівалентних ключів
- Виключення слабких ключів

Алгоритм, згідно із заявою авторів, стійкий до лінійного і диференціального аналізу.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

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

- Розділ верхнього меню (визивається при довгому натисненні): Лист; Скриньки; Параметри; Довідка.
- Розділ часу останньої синхронізації.
- Розділ поточного дати та часу.
- Розділ відображення листів: Від кого; Текст (скорочений); Дата; Розмір.

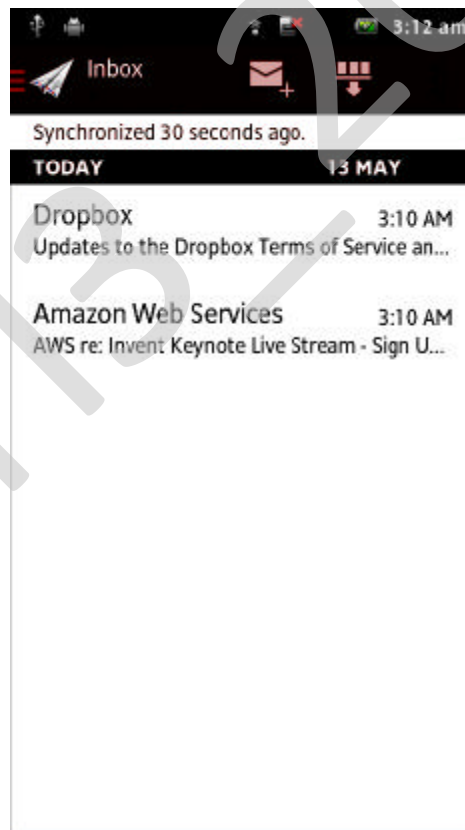


Рисунок 5.1 – Головне вікно програми

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

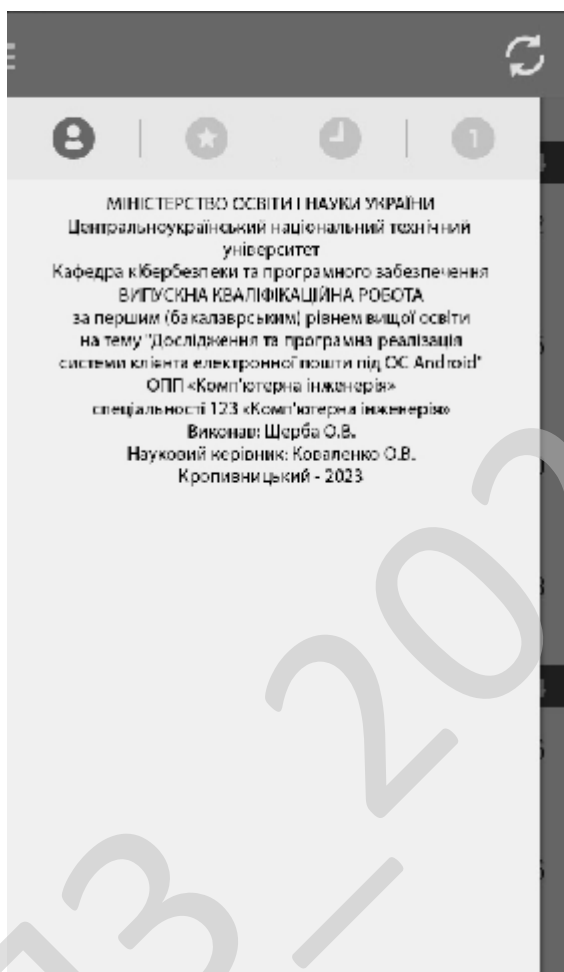


Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи клієнта електронної пошти під ОС Android.

Метою розробки є дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android.

Об'єктом дослідження є процес клієнта електронної пошти під ОС Android.

Предметом дослідження є методи клієнта електронної пошти під ОС Android.

Методи дослідження базуються на методах теорії телекому, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод клієнта електронної пошти під ОС Android.
- Розроблено вітчизняний продукт клієнта електронної пошти під ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи клієнта електронної пошти під ОС Android.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	100
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 73 = 123 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	123	Ф 7.1-7.4
Впровадження	13	Д13
Всього	164	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{164 \cdot 1}{60 - 5} = 3 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	39,49

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{др}^c = \frac{3_ч \cdot n_{міс}}{1,2}, \quad (7.6)$$

$$\Phi_{др}^c = \frac{39,49 \cdot 3}{1,2} = 99 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$C_{ел} = 99/(60 \cdot 8) = 0,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,25	9500	7125
Продакт-менеджер	0,25	8000	6000
Інженер-програміст	3	8000	72000
Інженер - електронщик	0,2	7000	4200
Інженер-системотехнік	0,25	7000	5250
Адміністратор мережі	0,5	7000	10500
Системний програміст	0,25	7000	5250
Дизайнер WEB	0,25	7000	5250
Інженер-верстальник	0,25	7000	5250
Бухгалтер-економіст	0,5	7000	10500
Всього за період розробки	$R_{cn} = 5,7$	-	$\Phi_{роб} = 131325$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{131325}{5,7 \cdot 60} = 384 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

$$B_{y\partial} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Комп'ютерторг за 20.10.23 – джерело <http://computorg.ua/ru/price.html>

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	AMD A8-8670E PRO (AD867BAHM44AB) AM4, 4 ядра, 4 потоки, 2.8 GHz, 3.3 GHz, Максимальний TDP - 35 Вт, 28nm, L1: 128KB, L2: 2x1MB, AMD Radeon Graphics, Carrizo, Tray	-
Системна плата	Biostar A520MH 3.0 - AM4, чипсет - AMD A520, DDR4, макс. об'єм оперативної пам'яті - 64 ГБ, макс. частота ОП - 4400 MHz, LAN - 1 Гбіт/с, D-Sub (VGA), HDMI, внутрішній - 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Відеокарта	GeForce GT710 1024Mb Afox PCI-Express 2.0, 1 ГБ, GDDR3, 64 Bit,	-
Жорсткий диск	SSD 2.5" 240GB Mibrand	-
Оперативна пам'ять	DDR4 8GB 2400 MHz eXceleram (E408247D) DDR4, 4 ГБ, В наборі - 2, Частота пам'яті - 2400 МГц,	-
DVD-привод	відсутній	-
Корпус	Vinga CS105B-450W Класичний, Miditower, ATX, Micro - ATX, Mini - ITX, PSU - 450 Вт	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	Atcom TD2027 USB 2.0 ALL IN 1 - (Memory Stick, Secure Digit зовнішній USB 2.0, micro-SD, SDHC, mini-SD, SDHC, RS-MMC, SD, SDHC, SDXC	-
інше	Клавіатура, мишка	-
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	199177

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$K_p = 2523008$		$A_p = 249636,25$

Примітка: вартість автомобіля Renault Scenic 2006 взята за даними електронного ресурсу, джерело https://auto.ria.com/uk/auto_renault_scenic_33598032.html, складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 384 \cdot 164 / 100 = 629 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 629 \cdot 10 \cdot 0,01 = 63 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(629+63) = 256 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 629 \cdot 15 \cdot 0,01 = 94 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджей, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

Згідно прийнятих норм на підприємстві $n_{\text{вип}}$ приймаємо 1,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=206$ грн., визначаємо вартість паперу за період розробки:

$$З_{M1} = Ц_n \cdot N_m. \quad (7.16)$$

$$З_{M1} = 206 \cdot 1,5 = 309 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 50):

$$З_{M2} = \sum Ц_{\delta}, \quad (7.17)$$

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR box – 24 грн./шт., DVD-R box – 39 грн./шт.

$$З_{M2} = 49 \cdot 24 + 1 \cdot 39 = 1215 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{M3} = \sum Ц_{з}, \quad (7.18)$$

де: $Ц_{з}$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$З_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$З_M = (309 + 1215 + 1702) / 100 = 32 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = З_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 629 \cdot 15 \cdot 0,01 = 94 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 100$ прим.):

$$A_m = \frac{A_p \cdot N_{\text{міс}}}{N_e \cdot 12}, \quad (7.20)$$

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 249636 \cdot 3 / (100 \cdot 12) = 624 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	3_o	629
2. Додаткова зарплата виконавців	3_δ	63
3. Відрахування на соціальні потреби	C_{oc}	256
4. Загальногосподарські витрати	Γ_{ocn}	94
5. Витрати на матеріали	3_M	32
6. Освоєння нових операційних систем, мов програмування	O_n	94
7. Амортизація основних фондів	A_m	624
8. Повна собівартість програмного забезпечення	C_n	1792
9. Плановий прибуток	Π_p	896
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	2688
11. Податок на додану вартість $\text{ПДВ} = 0.01 \cdot H_{ob} \cdot C_n$	ПДВ	537,6
12. Відпускна ціна програмної продукції $C = C_n + \text{ПДВ}$	C	3225,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = 3_o + 3_\delta + C_{oc} + \Gamma_{ocn} + 3_M + O_n + A_m. \quad (7.21)$$

$$C_n = 629 + 63 + 256 + 94 + 32 + 94 + 624 = 1792 \text{ грн.}$$

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 1792 = 896 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3226
Всього капітальних витрат	–	3226

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування)	Z_p	33818	14091
2. Витрати на електроенергію	$Z_{ел}$	276	58
3. Витрати на амортизацію	$Z_{ам}$	0	807
Всього витрат за рік	I	34094	14956

Витрати на профілактичні роботи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 100 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 105 \cdot 1,1 \cdot 1,22 = 33818 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 100 \cdot 105 \cdot 1,1 \cdot 1,22 = 14091 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,5 \cdot 240 \cdot 2,3 = 276 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,5 \cdot 50 \cdot 2,3 = 58 \text{ грн}.$$

$$T_e = \frac{479208}{(2688-1792) \cdot 100 \cdot 12 / 3} = 1,3 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	100
2. Повна собівартість розробленої програми	Грн.	1792
3. Ціна розробленої програми	Грн.	2688
4. Плановий прибуток від реалізації розробленої програми	Грн.	896
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	89600
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	32170
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3226
11. Величина економічного ефекту у користувача програмної продукції	Грн.	18332
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,17

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (34094 - 14956) - 0,25 \cdot 3226 = 18332 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{3226}{34094 - 14956} = 0,17 \text{ року.}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

В охорону праці включають санітарно-гігієнічні, лікувально-профілактичні та організаційно-технічні системи правових і соціально-економічних заходів.

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

8.2 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на декілька категорій:

Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів, які пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимбілдінг, спортивні змагання і естафети. Також, сюди

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог. Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9]. Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при нарузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

8.5 Розрахункова частина

Для захисного штучного заземлення будемо застосовувати вертикальні електроди з сталевого прокату круглого перерізу діаметром 35 мм., довжиною $L=1,5$ м., та горизонтальний електрод – металева полоса з перетином 35·4 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,65$ м. Відстань між вертикальними заземлювачами (електродами) $A=3$ м. Глибина закладення горизонтального контура заземлення $t=0,7$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,7+1,5/2=1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho_2 = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [11];

$\rho_2 = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта (глина) [11].

Діаметр вертикального електрода (задан):

$$D_v = 35 \text{ мм.} = 0,035 \text{ м.}$$

Відношення $A/L=3/1,5=2$.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.22).

4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення 19.10.22).

6. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.09.22).

7. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://spo.stu.cn.ua> (дата звернення 19.10.22).

8. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

9. Оришака О.В. Охорона праці в галузі та цивільний захист / О.В Оришака, Г.П. Горбачова, О.М. Мезенцева , К.М. Марченко, К.О. Буравченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: Видавець Лисенко В.Ф., 2019. – 226 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/9258> (дата звернення 19.09.22).

10. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

"Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.22).

11. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : "Колос", 1973. – 238 с.

КБГІЗ - 2023

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи клієнта електронної пошти під ОС Android.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів клієнта електронної пошти під ОС Android.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем клієнта електронної пошти під ОС Android.
- Досліджена система клієнта електронної пошти під ОС Android.
- На основі отриманих результатів досліджень створена програмна реалізація системи клієнта електронної пошти під ОС Android.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання клієнта електронної пошти під ОС Android.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

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.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

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

Програма реалізована на мові високого рівня Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані. 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 Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Sinople.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 18332 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,17 роки.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Щерба О.В. Дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
3. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
4. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
5. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
6. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
7. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
8. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.
9. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

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

11. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

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

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

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

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

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

					БКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

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

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

19. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

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

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

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

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

					BKPM-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

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

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

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

27. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

28. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

29. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

30. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

31. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

32. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

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

34. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

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

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

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

38. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		121

and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

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

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

41. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральнотраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

42. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

43. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

44. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

45. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

					ВКРМ-123.23.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		122

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

47. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

48. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

49. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

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

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

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

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 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.23.0025.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Щерба О.В.				Дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android	Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-1			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи клієнта електронної пошти під ОС Android.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 34-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи клієнта електронної пошти під ОС Android.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) 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;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.23.0025.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи клієнта електронної пошти під ОС Android;
- 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.

					ВКРМ-123.23.0025.00.00.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 мобільних пристроях під керуванням ОС Android 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 Android.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.

					ВКРМ-123.23.0025.00.00.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 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівців.

					ВКРМ-123.23.0025.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 123 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 13.12.2023 р.

					ВКРМ-123.23.0025.00.00.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
другим (магістерським) рівнем вищої освіти

_____ Коваленко О.В.

*Дослідження та програмна реалізація
системи клієнта електронної пошти під ОС Android*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2023 року

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

```
unit frmMail_clientSettings;

interface

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

type
  PMail_clientBoxSettings = ^TMail_clientBoxSettings;

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

    procedure btn_NewMail_clientBoxClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Combo_CheckEveryExit(Sender: TObject);
  end;
end.
```

```

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

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

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

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

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

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

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

const
  Mail_clientBoxSettingsFileName = 'Mail_clientboxes.dat';

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

implementation

uses frmMain;
{$R *.dfm}

procedure findMail_clientBoxAndSelectIt(name : shortstring);

```

```

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

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

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

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

procedure Tfrm_Mail_clientSettings.LoadFormFromBox(box :
PMail_clientBoxSettings);
Begin
  Ed_SMTPServer.Text := box.SMTPServer;

```

```

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

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

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

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

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

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

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

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

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

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

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

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

box.DynamicFolders      := Check_DynamicFolders.Checked;
box.ShowBalloonOnNewMes := Check_ShowBalloon.Checked;

```

End;

```

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

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

```

```

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

```

```

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

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

```

```

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

ListBox_Mail_clientBoxes.Items.Add(Ed_Mail_clientBoxName.Text);
GroupBox1.Visible:=true;
Radio_Ask.Checked:=true;
inc(TotalMail_clientBoxes);
new(Mail_clientBoxesSettings[TotalMail_clientBoxes]);

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

```

```

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

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

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

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

function FindMail_clientBox(name : shortstring):PMail_clientBoxSettings;
var
    i : word;
begin
    FindMail_clientBox:=nil;

for i:=1 to TotalMail_clientBoxes do
    if Mail_clientBoxesSettings[i].Name = name then
        begin
            FindMail_clientBox:=Mail_clientBoxesSettings[i];

```

```

        exit;
    end;

End;

procedure Tfrm_Mail_clientSettings.ListBox_Mail_clientBoxesClick(Sender:
TObject);
begin
    if ListBox_Mail_clientBoxes.ItemIndex >=0 then
    begin
        CurrentMail_clientBoxSettings:=FindMail_clientBox(ListBox_Mail_clientBoxes.Items
[ListBox_Mail_clientBoxes.itemindex]);
        if CurrentMail_clientBoxSettings=nil then
        CurrentMail_clientBoxSettings:=PrevMail_clientBoxSettings;
        if PrevMail_clientBoxSettings<>nil then
        SaveFormToBox(PrevMail_clientBoxSettings);
        PrevMail_clientBoxSettings:=CurrentMail_clientBoxSettings;
        LoadFormFromBox(CurrentMail_clientBoxSettings);
        end;
    end;

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

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

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

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

end.

```

Файл 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,
  {HTTApp, {HTTProd,}
  LetterStorage, frmMail_clientSettings, CoolTrayIcon, about;

type
  Tfrm_Main = class(TForm)
    MainMenu: TMainMenu;
    Splitter1: TSplitter;
    Splitter2: TSplitter;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    CoolBar: TCoolBar;
    Tree_Mail_clientBoxes: TTreeView;
    ListView_LettersHeaders: TListView;
    ToolBar1: TToolBar;
    btn_CheckMail_client: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    XPMANifest1: TXPMANifest;
    Panel_Content: TPanel;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    ActionList1: TActionList;
    aCheckMail_client: TAction;
    aSendMail_client: TAction;
    aMail_clientSettings: TAction;
    aExit: TAction;
    ToolBar2: TToolBar;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    aNewLetter: TAction;
    ToolButton7: TToolButton;
    MemoLog: TMemo;
    Splitter3: TSplitter;
    aSettings: TAction;
    ToolButton8: TToolButton;
    ToolButton9: TToolButton;
    N11: TMenuItem;
    Web_LetterViewer: TWebBrowser;
    Splitter4: TSplitter;
    List_Attachmant: TListView;
    Memo_LetterBody: TMemo;
    Popup_Attachment: TPopupMenu;
    mnuSaveAtt: TMenuItem;
    SaveDialog_Att: TSaveDialog;
    PopupMenuHeaders: TPopupMenu;
    mnu_DeleteLetter: TMenuItem;
    mnu_Answer: TMenuItem;
    mnu_Resend: TMenuItem;
    N12: TMenuItem;
    aAnswer: TAction;
  end;

```

```

aResend: TAction;
PopupMenu_Mail_clientBoxes: TPopupMenu;
N13: TMenuItem;
Jnghfdbnmgjxne1: TMenuItem;
N14: TMenuItem;
About1: TMenuItem;
procedure aMail_clientSettingsExecute(Sender: TObject);
procedure aExitExecute(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure aNewLetterExecute(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure aSettingsExecute(Sender: TObject);
procedure Tree_Mail_clientBoxesClick(Sender: TObject);
procedure aCheckMail_clientExecute(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure List_AttachmantClick(Sender: TObject);
procedure Web_LetterViewerBeforeNavigate2(Sender: TObject; const pDisp:
IDispatch; var URL, Flags, TargetFrameName, postData, Headers: OleVariant; var
Cancel: WordBool);
    procedure ListView_LettersHeadersSelectedItem(Sender: TObject; Item:
TListItem; Selected: Boolean);
    procedure List_AttachmantMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
    procedure mnuSaveAttClick(Sender: TObject);
    procedure Popup_AttachmentPopup(Sender: TObject);
    procedure mnu_DeleteLetterClick(Sender: TObject);
    procedure PopupMenuHeadersPopup(Sender: TObject);
    procedure aAnswerExecute(Sender: TObject);
    procedure aResendExecute(Sender: TObject);
    procedure Tree_Mail_clientBoxesDragDrop(Sender, Source: TObject; X, Y:
Integer);
    procedure Tree_Mail_clientBoxesDragOver(Sender, Source: TObject; X, Y:
Integer; State: TDragState; var Accept: Boolean);
    procedure TrayClick(Sender: TObject);
    procedure TrayMinimizeToTray(Sender: TObject);
    procedure N13Click(Sender: TObject);
    procedure PopupMenu_Mail_clientBoxesPopup(Sender: TObject);
    procedure Jnghfdbnmgjxne1Click(Sender: TObject);
    procedure About1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    procedure ShowMail_clientBoxesInTree;
    procedure LoadLetters;
    procedure AddToLog(s : shortstring);
    procedure HideLetterBody;
    procedure ShowInHtmlViewer(body : TStrings);
end;

const
    LogFileName    = 'Actions.log';

var
    frm_Main: Tfrm_Main;

    Mail_clientInfoLoaded : boolean=false;

    { Головне сховище електроних листів }
    Mail_clientBoxes      : TMail_clientBoxList;

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

```

implementation

```

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

{$R *.dfm}

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

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

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

            ShowMail_clientBoxesInTree;
            LoadLetters;
            Mail_clientInfoLoaded:=true;

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

procedure Tfrm_Main.ShowMail_clientBoxesInTree;
var
    i      : word;
    node   : TTreeNode;
begin
    Tree_Mail_clientBoxes.Items.Clear;
    for i:=1 to TotalMail_clientBoxes do
        begin

```

```

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

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

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

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

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

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

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

procedure Tfrm_Main.AddToLog(s : shortstring);
var
    st      : TFileStream;
    SLog    : shortstring;
    Mess    : shortstring;
    mem     : TMemoryStream;
Begin
    if (s[1]=#10)and(s[2]=#13) then

```

```

begin
delete(s,1,2);
Mess:=#10#13+DateTimeToStr(Now)+' -=> '+s+#10#13;
end
else Mess:=DateTimeToStr(Now)+' -=> '+s+#10#13;

MemoLog.Lines.Add(Mess);

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

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

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

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

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

procedure Tfrm_Main.FormCreate(Sender: TObject);
begin
frm_Settings.LoadGlobalSettingsFromFile;
DeleteFiles(
IncludeTrailingPathDelimiter(ExtractFilePath(Application.ExeName))+ 'temp_*.html'
);

```

```

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

// параметри програми

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

procedure Tfrm_Main.Tree_Mail_clientBoxesClick(Sender: TObject);
var
  Mail_clientBoxName      : shortstring;
  Mail_clientBoxFolder    : shortstring;
  LetterList              : PLetters;
  i                       : word;
begin
  if Tree_Mail_clientBoxes.Selected = nil then
  begin
    HideLetterBody;
    exit;
  end;
  List_Attachmant.Clear;
  List_Attachmant.Visible:=false;
  Memo_LetterBody.Clear;
  if Tree_Mail_clientBoxes.Selected.Parent = nil then
  begin
    if Mail_clientBoxes.FindBox(Tree_Mail_clientBoxes.Selected.Text).IsBuisyNow
    then aCheckMail_client.Enabled:=false
    else
aCheckMail_client.Enabled:=true;
    ListView_LettersHeaders.Clear;
    HideLetterBody;
    findMail_clientBoxAndSelectIt(Tree_Mail_clientBoxes.Selected.Text);
    exit;
  end;

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

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

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

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

  for i:=1 to LetterList^.Count do
  with ListView_LettersHeaders.Items.Add do
  begin
    Caption:=LetterList^.Letters[i]^Subject;
    if LetterList^.Letters[i]^Name='' then
SubItems.Add(LetterList^.Letters[i]^From.Address)

```

```

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

end;

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

(*
procedure RetrieveMail_client(let_count : word);
var
  mes : TIdMessage;
  i : word;
  LettersNot2beDownloaded : word;
Begin
  try
  LettersNot2beDownloaded:=0;
  for i:=1 to let_count do
  if CurrentMail_clientBoxSettings^.RetrieveOnlyHeaders then
  // Отримати лише заголовки
  begin
  mes:=TIdMessage.Create(frm_Main);
  DM.POP.RetrieveHeader(i,mes);
  end
  else
  // Отримати вес лист
  begin
  mes:=TIdMessage.Create(frm_Main);
  DM.POP.RetrieveHeader(i,mes);
  if
Mail_clientBoxes.GetByName(CurrentMail_clientBoxSettings^.Name).HasLetterWithID(
mes.MsgId) then
  begin
  inc(LettersNot2beDownloaded);
  continue;
  end;
  if
Mail_clientBoxes.GetByName(CurrentMail_clientBoxSettings^.Name).hadLetterWithId(
mes.msgId) then
  begin
  case CurrentMail_clientBoxSettings.ifDuplicate of
    Ignore : continue;
    Recieve : ;
    Ask : begin
      frm_AskOnDuplicate.FillInInfo(@mes);
      frm_AskOnDuplicate.ShowModal;
      case frm_AskOnDuplicate.ModalResult of
        mrIgnore : continue; // не отримувати
        mrOk : ; // отримувати
        mrAbort : begin // видаляти з сервера

```

```

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

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

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

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

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

procedure Tfrm_Main.aCheckMail_clientExecute(Sender: TObject);
var
CurMail_clientBox : PMail_clientBox;
begin
// Перевірити пошту
Tree_Mail_clientBoxesClick(self);
if CurrentMail_clientBoxSettings =nil then exit;

aCheckMail_client.Enabled:=false;

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

end;

procedure Tfrm_Main.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Mail_clientBoxes.SaveMail_clientBoxes;
end;

```

```

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

    //Список листів

```

```

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

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

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

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

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

```

```

                                and

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

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

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

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

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

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

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

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

```

```

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

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

End;

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

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

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

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

Mail_clientBoxes.FindBox(Mail_clientBoxName).MoveToGarbage(LetterList,ListView_L
ettersHeaders.ItemIndex+1);
    ListView_LettersHeaders.DeleteSelected;
end;

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

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

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

    frm_NewLetter.ClearForm;

```

```

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

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

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

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

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

end;

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

```

```

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

  Tree_Mail_clientBoxesClick(self);

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

2023

```

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

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

end;

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

end.
```

Файл MyMail_client.dpr - файл проекту

```
program MyMail_client;

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

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Поштовый клиент, разработаний Гушан М.М.';
  Application.CreateForm(Tfrm_Main, frm_Main);
  Application.CreateForm(TDM, DM);
  Application.CreateForm(Tfrm_Mail_clientSettings, frm_Mail_clientSettings);
  Application.CreateForm(Tfrm_Settings, frm_Settings);
  Application.CreateForm(Tfrm_NewLetter, frm_NewLetter);
  Application.CreateForm(Tfrm_AskOnDuplicate, frm_AskOnDuplicate);
  Application.CreateForm(TForm1, Form1);
  Application.HintPause:=200;
  Application.Run;
end.
```

Файл LetterStorage.pas - збереження листів

```

unit LetterStorage;

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

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

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

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

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

PBoolean = ^boolean;
TGetMail_clientThread = class(TThread)
  private
    POP : TIdPOP3;
    Mail_clientBoxSettings : TMail_clientBoxSettings;
    LogString : string;
    BalloonMes : string;
    IsBuisy : PBoolean;
  protected
    procedure Execute; override;
    procedure CallAddToLog(s:string);

```

```

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

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

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

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

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

TMail_clientBoxList =object
    private
        Mail_clientBoxes : array [1..100] of PMail_clientBox;
    public
        TotalBoxes: word;
        function Add(name : shortstring) : PMail_clientBox;
        function GetByName (name : shortstring) : PMail_clientBox;
        function GetByIndex(index: word) : PMail_clientBox;
        function FindLetters( Mail_clientbox, Mail_clientfolder : shortstring
):PLetters;
        function FindBox( boxname : shortstring): PMail_clientBox;
        procedure SaveMail_clientBoxes;
        end;

var
    LettersPath      :   string;
    CS                :   TCriticalSection;

implementation

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

{ TLetter }

```

```

constructor TLetters.Init(Mail_clientBox, Folder : shortstring);
Begin
  Mail_clientBoxName := Mail_clientBox;
  FolderName := Folder;
End;

{ *** Структура файлу TLetters ***}

{ 4 байти      - розмір наступного листа  }                               //(* { 256(5) байт
- ідентифікатор} *)
{ ... .. }
{ .. Лист ..}
{ ... .. }

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

  TotalLetters2Load : word;
  sizeofALetter    : word;
Begin
  findMail_clientBoxAndSelectIt(Mail_clientBoxName);
  if CurrentMail_clientBoxSettings.DynamicFolders then exit;

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

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

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

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

      zstCompressed.Read(IsOnlyHeader, sizeof(IsOnlyHeader));
      zstCompressed.Read(LettersSizes, sizeof(LettersSizes));
      TotalLetters:=0;

```

```

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

End;

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

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

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

```

```

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

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

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

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

zst.Free;
st.Free;

end;

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

16132023

```

    exit;
  end;
end;

  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.');
```

16132023

```

        exit;
      end;
    end;
  end;

  // Загальна кількість листів
  TotalLetters:=0;
  fillchar(IsOnlyHeader, sizeof(IsOnlyHeader), 0);
  zst.Write(TotalLetters, sizeof(TotalLetters));
  zst.Write(IsOnlyHeader, sizeof(IsOnlyHeader));

  zst.Free;
  st.Free; }
end;

```

```

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

```

```

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

```

```

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

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

```

```

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

```

```

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

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

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

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

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

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

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

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

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

End;

function TMail_clientBox.GetName : shortstring;
Begin
    GetName:=BoxName;

```

End;

```

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

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

{ TMyAttachments }

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

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

constructor TMyAttachments.Load(st : TStream);
var
  i : word;
  s : shortstring;
Begin
  Count:=0;
  fillchar (FileNames,sizeof(FileNames),0);
  st.Read(Count,sizeof(Count));
  for i:=1 to Count do
  begin
    st.Read(s,sizeof(s));
    Add(s);
  end;
End;

procedure TMyAttachments.Save(st : TStream);
var
  i : word;
Begin
  st.write(Count,sizeof(Count));
  for i:=1 to Count do st.write(FileNames[i]^,sizeof(FileNames[i]^));
End;

destructor TMyAttachments.Done;
var
  i : word;
Begin
  for i:=1 to Count do dispose(FileNames[i]);
End;

procedure TMyAttachments.Add(fname : shortstring);
Begin
  inc(Count);
  new(FileNames[Count]);
  FileNames[Count]^:=fname;
End;

function TMyAttachments.Get(index: word): string;
begin
  if index<=Count then Get:=FileNames[index]^;
end;

```

```

{ TGetMail_clientThread }

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

  Mail_clientBoxSettings:=mbSettings;
  IsBuisy:=IsBuisyFlag;
end;

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

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

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

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

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

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

```

```

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

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

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

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

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

```

```

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

    end;
    CS.Leave;
end;

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

{Mail_clientBoxes.GetByName(Mail_clientBoxSettings.Name)}CurMail_clientBox^.Inbo
x.Add(@mes,false, POP.RetrieveMsgSize(i));
    if Mail_clientBoxSettings.DeleteFromServer then
        if not POP.Delete(i) then frm_Main.AddToLog('Помилка видалення листа з
сервера. Від: '"+mes.From.Address+'" Тема: '"+mes.Subject+'");
        end;

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

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

            self.Suspend;
        end;
    END;
end;

begin

    CS:=TCriticalSection.Create;

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

end.

finalization
    CS.Free;
end.

```

Файл frmNewLetter.pas - створення та відправка нового листа

```

unit frmNewLetter;

interface

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

type
  PidMessage = ^TidMessage;

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

```

```

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

procedure FillSMTPwithMail_clientBoxSettings(mbSettings :
PMail_clientBoxSettings);

var
  frm_NewLetter: Tfrm_NewLetter;

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

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

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

```

```

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

```

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

```

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

```

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

```

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

```

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

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

```

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

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

function Tfrm_NewLetter.CreateLetterFromForm: PidMessage;
var
  PLet : PidMessage;
  i : integer;
  s : ^shortstring;
begin
  frm_Main.Tree_Mail_clientBoxesClick(self);
  if CurrentMail_clientBoxSettings =nil then exit;

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

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

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

```

```

    CreateLetterFromForm:=PLet;
end;

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

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

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

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

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

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

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

```

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

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

```
procedure Tfrm_NewLetter.FormShow(Sender: TObject);
begin
    CURR_Mail_clientbox_SETTINGS:= CurrentMail_clientBoxSettings^;

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

end.
```

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.
```

К6П3 - 2023

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.
```