

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи аналізу та**  
**консолідації розподіленої інформації уніфікованих комунікацій**  
**з підтримкою відеозв’язку”**

Виконав здобувач вищої освіти  
II курсу, групи КН-23М  
ОПП «Комп’ютерні науки»  
спеціальності 122 «Комп’ютерні науки»  
\_\_\_\_\_ Величко Є.Д.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Коваленко А.С.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 "Інформаційні технології"  
Спеціальність 122 "Комп'ютерні науки"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

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

Величку Євгенію Дмитровичу

(прізвище, ім'я, по батькові)

- |  |   |  |                            |   |  |  |  |  |                     |  |  |
|--|---|--|----------------------------|---|--|--|--|--|---------------------|--|--|
| 1. Тема роботи   | <i>Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку</i>   |  |                            |   |  |  |  |  |                     |  |  |
| 2. Керівник роботи   | <i>Коваленко Анна Степанівна, канд. техн. наук, доцент</i><br>(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)<br>затверджені наказом вищого навчального закладу № 18-13 від 07.08.2024 року   |  |                            |   |  |  |  |  |                     |  |  |
| 3. Строк подання студентом роботи до захисту   | <u>2.12.2024 р.</u>   |  |                            |   |  |  |  |  |                     |  |  |
| 4. Мета та завдання випускної кваліфікаційної роботи:                                | <i>Метою розробки є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку</i>  |  |                            |   |  |  |  |  |                     |  |  |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</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> |  |
| <i>1. Призначення та область використання.</i>                                       | <i>6. Наукова новизна.</i>  |  |                            |   |  |  |  |  |                     |  |  |
| <i>2. Перегляд аналогічних існуючих систем.</i>                                      | <i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</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   | Доренська А.О.                            | 05.10.2024     | 14.11.2024       |
| Охорона праці | Марченко К.М., к.т.н., доцент             | 06.10.2024     | 16.11.2024       |
|               |   |                |                  |

7. Дата видачі завдання « 6 » вересня 2024 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  | 10.10.2024 р.   |          |
| 2.    | Постановка задачі, оформлення ТЗ  | 15.10.2024 р.   |          |
| 3.    | Розробка моделі компонента  | 20.10.2024 р.   |          |
| 4.    | Розробка структур даних   | 25.10.2024 р.   |          |
| 5.    | Розробка алгоритмів зв'язку та відображення   | 30.10.2024 р.   |          |
| 6.    | Програмування алгоритмів  | 10.11.2024 р.   |          |
| 7.    | Розрахунок економічної ефективності   | 13.11.2024 р.   |          |
| 8.    | Розрахунки з охорони праці та техніки безпеки   | 15.11.2024 р.   |          |
| 9.    | Оформлення ПЗ   | 17.11.2024 р.   |          |
| 10.   | Попередній захист роботи  | 2.12.2024 р.  |          |
|       |   |   |          |

Дата видачі завдання  
« 6 » вересня 2024 р.

Підпис керівника

\_\_\_\_\_  
(прізвище та ініціали)Завдання прийнято до виконання  
« 6 » вересня 2024 р.

Підпис здобувача

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

**Величко Є.Д. Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

Метою розробки є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

Об'єктом дослідження є процес аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

Предметом дослідження є методи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

Методи дослідження базуються на методах теорії кодування та теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

**Ключові слова:** Комп'ютерні науки, уніфіковані комунікації, відеозв'язок

## ABSTRACT

**Velichko Ye.D. Research and software implementation of the system for analyzing and consolidating distributed information of unified communications with video support. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the system for analyzing and consolidating distributed information of unified communications with video support.

The purpose of the development is the research and software implementation of the system for analyzing and consolidating distributed information of unified communications with video support.

The object of the research is the process of analyzing and consolidating distributed information of unified communications with video support.

The subject of the research is the methods of analyzing and consolidating distributed information of unified communications with video support.

The research methods are based on the methods of coding theory and computer network theory, methods of mathematical statistics, and methods of software development.

The result of the work is a software implementation of a system for analyzing and consolidating distributed information of unified communications with video support.

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 are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

**Keywords:** Computer science, unified communications, video communication

## 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.....   | 7  |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....  | 9  |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... | 9  |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....  | 12 |
| 2.3 Розгорнута постановка завдання .....  | 18 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....   | 19 |
| 3.1 Опис функціонування системи .....   | 19 |
| 3.2 Розробка структурної схеми.....   | 22 |
| 3.3 Розробка функціональної схеми .....   | 32 |
| 3.4 Розробка діаграми процесів.....   | 34 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....  | 36 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....   | 36 |
| 4.2 Захист розробленого програмного забезпечення.....   | 47 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....  | 52 |
| 6 НАУКОВА НОВИЗНА .....   | 58 |

|                 |                       |                 |              |             |   |                    |              |                |
|-----------------|-----------------------|-----------------|--------------|-------------|---|--------------------|--------------|----------------|
|                 |                       |                 |              |             | <b>БКРМ-122.24.0004.00.00.ПЗ</b>  |                    |              |                |
| <b>Вим</b>      | <b>Арк.</b>           | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> | <i>Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку</i> | <b>Літ.</b>        | <b>Аркуш</b> | <b>Аркушів</b> |
| <i>Розроб.</i>  | <i>Величко Є.Д.</i>   |                 |              |             |   | <b>М</b>           | 1            | 83             |
| <i>Перев.</i>   | <i>Коваленко А.С.</i> |                 |              |             |   | <b>ЦНТУ КН-23М</b> |              |                |
| <i>Н.контр.</i> | <i>Коваленко А.С.</i> |                 |              |             |   |                    |              |                |
| <i>Затв.</i>    | <i>Смірнов О.А.</i>   |                 |              |             |   |                    |              |                |

|     |  |    |
|-----|--|----|
| 7   | МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....  | 59 |
| 7.1 | Визначення цільової аудиторії кінцевого готового продукту .....  | 59 |
| 7.2 | Оцінка привабливості шляхом застосування методів експертних оцінок ...                                   | 60 |
| 7.3 | Вибір методу оцінки вартості ПЗ .....  | 61 |
| 7.4 | Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості..... | 62 |
| 7.5 | Пропозиція алгоритму просування проєкту розробки ПЗ .....  | 63 |
| 7.6 | Оптимізація каналів збуту та шляхів реалізації ПЗ .....  | 64 |
| 7.7 | Визначення ключових факторів успіху конкретного проєкту.....   | 65 |
| 8   | ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....  | 66 |
| 8.1 | Вступ.....   | 66 |
| 8.2 | Шкідливі і небезпечні фактори при роботі з комп'ютером.....  | 67 |
| 8.3 | Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...                                | 68 |
| 8.4 | Розробка заходів з умов поліпшення охорони праці .....   | 71 |
| 8.5 | Розрахункова частина .....   | 72 |
| 9   | ОСНОВНІ ВИСНОВКИ.....  | 75 |
|     | СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....   | 77 |

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 2    |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

DS – DirectShow мультимедійне розширення Microsoft;

TrFl – TransformFilter -Трансформ-фільтр;

RnFl – RendererFilter – Рендер-фільтр;

SourceFilter – SourceFilter – Фільтр джерела;

IC – Intelligent Connect – інтелектуальне з'єднання;

ООП – об'єктно-орієнтоване програмування;

DSP – бібліотека компонентів для розробки мультимедійних програм;

VWINDOW – вертикальна синхронізація;

HWINDOW – горизонтальна синхронізація;

PLL – Цикл Блокування Стадії;

USB – Universal Serial Bus – універсальна послідовна шина;

URL – universal resource locator – локатор ресурсів інтернет;

HTML – мова розмітки гіпертекстових документів;

ОС – операційна система;

ПЕОМ – персональна електронно обчислювальна машина;

КС – комп'ютерна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 3    |

## ВСТУП

**Актуальність теми.** Термін «уніфіковані комунікації» (УфК) уже давно й міцно ввійшов у вживання. Усе починалося з інтеграції електронної й голосової пошти, у результаті чого користувачі одержали можливість із єдиного інтерфейсу звертатися до повідомлень обох типів. Такі рішення назвали уніфікованою поштою. Пізніше, в основному завдяки активному впровадженню ІР-телефонії, стало можливим з того ж інтерфейсу ініціювати й приймати телефонні виклики. Подібний комунікаційний функціонал і дав привід «розширити» термін і перейти до поняття УфК. Поступово функціональність систем УфК розширювалася, однак цей процес не був підкріплений масовим попитом на них у корпоративному секторі. Розвиток УфК гальмувало ту обставину, що багато комунікаційних систем базувалися на закритих пропрієтарних архітектурах, через що можливості інтеграції виявлялися сильно обмеженими. При цьому кожний виробник або група виробників розуміли під УфК щось відмінне від інших. Точка зору визначалася «точкою сидіння»: виробники офісних АТМ у главу кута ставили функції голосового зв'язку, гравці ринку відео-конференц-зв'язку (ВКЗ) уважали, що відео – «усьому голова», а молоді й активні постачальники сервісів Web-конференцій переконували всіх, що саме їхнього рішення і є щирі УфК, і т.д., і т.п. Перехід галузі до загальних технологічних знаменників, таким як протоколи ІР і SIP, спростив інтеграцію різних систем, а хвиля покупок і об'єднань виробників привела до того, що в розпорядженні великих постачальників з'явилися всі необхідні компоненти уніфікованих комунікацій. Однак чітке погоджене розуміння того, які функціональні блоки повинні бути обов'язковими компонентами системи УфК, довгий час був відсутній.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 4    |

– Огляд існуючих систем аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

– Дослідження системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

– Програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

*Об'єктом дослідження є процес аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.*

*Предметом дослідження є методи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.*

*Методи дослідження базуються на методах теорії кодування та теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

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

– Удосконалено метод аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

– Розроблено вітчизняний продукт аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

**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.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 5    |

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ\_2024

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPM-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 6    |

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Наступні блоки УфК варто вважати основними:

- голосовий зв'язок (телефонія);
- відеозв'язок (ВКЗ);
- веб-конференції;
- електронна пошта;
- керування статусом користувачів (presence);
- миттєвий обмін повідомленнями (чати).

Хоча консенсус щодо функціонального наповнення систем УфК, схоже, досягнуть, рівень проникнення відповідних рішень невеликий. Тільки 13% респондентів указали на те, що в їхній компанії вже розгорнута єдина (уніфікована) комунікаційна система, що забезпечує голосову й відеозв'язок, Web-конференції й інші функції. У більшості ж випадків для різних комунікаційних функцій – телефонія, ВКЗ, Web-конференції – застосовують різні системи.

Але сумувати постачальникам уніфікованих систем не треба. 66% опитаних нами компаній указали на те, що саме впровадження таких систем стане пріоритетною метою на найближчі роки. Таких значно більше, ніж тих, хто буде акцентувати увагу на впровадженні ВКЗ в «чистому» виді (30%) або Web-конференцій з підтримкою відеозв'язку (45%).

## 1.2 Область застосування

Основним спонукальним мотивом для впровадження УфК для компаній є прагнення скоротити загальні витрати на корпоративні комунікаційні системи

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 7    |

(67%). Крім того, багато хто (55%) сподіваються, що завдяки єдиному користувальницькому інтерфейсу значно підвищиться зручність використання комунікаційних сервісів. Більше просте керування (одна система замість декількох) залучає 48% респондентів. Чимало замовників (41%) розраховують і на те, що збільшення числа каналів взаємодії між співробітниками, а також між співробітниками й клієнтами підвищить ефективність бізнесу й дасть інші додаткові конкурентні переваги.

У ході дослідження нами були більш докладно вивчені питання застосування відео-конференц-зв'язку. Рівень проникнення відповідних рішень виявився досить високим, але в більшості випадків термінали ВКЗ установлені тільки в декількох конференц-кімнатах і кабінетах керівників і використовуються в основному топ-менеджерами. Саме така ситуація в 40% опитаних компаній. При цьому в тих же компаніях часто співробітники задіють безкоштовні сервіси на зразок Skype.

У кожній п'ятій компанії корпоративною системою ВКЗ охоплені «широкі маси» співробітників, які можуть брати участь у конференціях зі своїх робочих місць. Цей показник ми оцінюємо як дуже високий. 9% компаній планують розгорнути систему ВКЗ протягом найближчого року. Приблизно 12% учасників не змогли вибрати жоден із запропонованих нами варіантів для оцінки ситуації з наявністю системи ВКЗ. Якщо додати до них тих, хто не використовує й не збирається впроваджувати ВКЗ, виходить, що чверть опитаних, видимо, поки не знаходять мотивації для організації в себе системи відео-конференц-зв'язку.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

## 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

Головним призначенням відеоконференц-системи є економія часу й грошей користувача.

Значна вартість такої системи не є перешкодою для великих компаній, у яких часто один груповий термінал ВКЗ не перевищує вартості двох-трьох відряджень співробітників. Середній же й малий бізнес, не маючи таких же фінансових можливостей, обходиться в основному Skype-зв'язком. Однак стає неможливим миритися з досить умовною якістю зв'язку, у той час як твої партнери по бізнесу вже перейшли на системи відеоконференц-зв'язку.

Останнім часом ринок ВКЗ швидко міняється, створюючи різноманіття різних підходів до побудови систем ВКЗ. Для кінцевого користувача це означає максимальне рішення поставлених завдань винятково за наявний у розпорядженні бюджет.

На даний момент можна виділити кілька ключових типів ВКЗ, що зустрічаються на ринку:

- Повністю хмарне рішення.
- Хмарно-апаратне рішення.
- Хмарне або хмарно-апаратне рішення з підтримкою публічних засобів відео спілкування.
- Програмне серверне рішення (для установки в мережі замовника або в дата-центрі провайдеру).
- Повністю апаратне рішення (класична інфраструктура).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 9    |

Розглянемо докладніше переваги й недоліки кожного типу рішень і приведемо приклади вдалої реалізації.

### **Повністю хмарне рішення**

Рішення покликане об'єднати всіх співробітників компанії в єдину ВКЗ-платформу. На 100% реалізує концепцію BYOD (Bring Your Own Device), що уможливорює використання особистих або корпоративних комп'ютерів/ноутбуків, смартфонів і планшетів як пристрої ВКЗ-зв'язку.

Переваги хмарного рішення: відсутні витрати на апаратну інфраструктуру, а також не потрібна участь технічного фахівця для налаштування й обслуговування системи. Звичайно ліцензується передплата на 1 рік із продовженням.

Відмінний приклад успішного хмарного рішення: Lifesize Cloud.

Вартість мінімального рішення на 10 постійних учасників = \$5624 у рік (\$47 на 1 чоловік на місяць).

Lifesize Cloud заміняє апаратну інфраструктуру в 7-10 разів більшої вартості початкової закупівлі. А також виключає витрати на регулярне обслуговування, відновлення сервісних ліцензій (10% вартості встаткування в рік), зарплату системному адміністраторові.

Таким чином, витрати на хмарне рішення прийнятні при всіх перевагах такого підходу.

### **Хмарно-апаратне рішення**

Для інсталяції в конференц-зали й переговорні кімнати часто потрібні якісні зовнішні камери – потрібний гарний оптичний зум, керування наведенням камери на спікера, а також якісні мікрофонні системи з можливістю охопту середнього й великого приміщення. У таких випадках неможливо обійтись без апаратних класичних кодеків ВКЗ із якісними комплектними камерами й мікрофонами.

При цьому, такі системи можуть прекрасно працювати із хмарними рішеннями, поєднуючись із ними в єдине ціле. Хмарне рішення в такому випадку

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 10   |

створює можливість багатобічної конференції й підключає учасників з мобільними пристроями.

Вартість Lifesize Icon 400 для переговорної кімнати або конференц-залу становить \$3770 (у вартість включена сервісна ліцензія на 1 рік). Додаємо хмарне рішення Lifesize Cloud на 10 учасників \$5624.

Такий підхід поєднує зручність класичних апаратних систем і простоту з розширеною функціональністю хмарного рішення.

### **Хмарне або хмарно-апаратне рішення з підтримкою публічних засобів відеоспілкування**

Самий новий і самий затребуваний на сьогодні тип ВКЗ-системи. До всіх переваг, описаних у п. 1-3, додається можливість підтримки публічних засобів відеоспілкування (Microsoft Lync, Google Hangouts, Cisco Jabber, Skype, Lifesize Cloud, WebEx, GoToMeeting і т.п.)

Для того, щоб усе це працювало, до хмарного рішення Lifesize Cloud на 10 учасників \$5624 додаємо кодек (з камерою й мікрофоном): Lifesize Icon Flex \$3040 (у вартість включена сервісна ліцензія на 1 рік).

### **Програмне серверне рішення (для установки в мережі замовника або в дата-центрі провайдеру)**

Таке рішення звичайно вимагає наявності власного сервера в замовника, куди можна встановити ПЗ для ВКЗ (звичайно на основі віртуальної машини).

Альтернативні варіанти: покупка серверних віртуальних ресурсів у провайдеру й розміщення там віртуальної машини або установка власного сервера в дата-центрі провайдеру. Можуть бути використані обидва варіанти одночасно.

Фактично, при такому підході замовник одержує повністю контрольовану хмару у власній мережі.

Використовуючи кілька віртуальних машин, як усередині мережі, так і зовні, можна поєднувати їхні ресурси й гнучко управляти інтернет-трафіком і маршрутизацією.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 11   |

Переваги такого підходу оцінять великі корпорації, сервіс-провайдери, компанії із сотнями людей персоналу, які повинні бути об'єднані відеозв'язком. А також ті, хто активно використовує Microsoft Lync і хоче розширити його функціональність якісним відеозв'язком. Всі ці переваги використовує рішення Pexip Infinity.

Pexip Infinity також сумісно із традиційними апаратними системами по стандартних протоколах SIP / H.323. Працює з найвідомішими публічними засобами відеоспілкування Microsoft Lync і Skype.

Pexip Infinity ліцензується залежно від кількості учасників або використовуваних портів. Розрахунок надається індивідуально під поставлене завдання.

### **Повністю апаратне рішення (класична інфраструктура)**

Доступний великий вибору різних кодеків, серверів і іншого устаткування, що дозволяє побудувати повністю апаратний комплекс. Устаткування Lifesize відмінно вирішує дані завдання.

Таким чином, важливо зрозуміти особливості експлуатації системи замовником і запропонувати найбільш підходяще рішення під індивідуальні особливості бізнесу. Використання хмарних і програмних рішень гарантує оперативне відновлення для підтримки всіх нових функцій, що з'являються, нових версій відеокодеків, протоколів і т.п. Використання ж окремих апаратних елементів додає зручності експлуатації в переговорні кімнати й конференц-зали.

## **2.2 Обґрунтування вибору засобів для побудови системи та мови програмування**

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 12   |

мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

### **Швидкість виконання коду Python**

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

### **Використання Python**

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 13   |

2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.

3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.

4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.

5. World of Tanks.

6. Вільна енциклопедія Вікіпедія.

7. Пошукова система Google.

8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.

9. YouTube – популярне відеосховище.

### **Версії Python**

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з'являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп'ютері. Одна з основних функцій процесора – це обробка даних згідно комп'ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп'ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп'ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 14   |

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

### **Отримати**

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

### **Декодувати**

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

### **Виконати**

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 15   |

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

### **Завантаження Python**

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

### **Середовище програмування для Python**

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 16   |

Текстові редактори та інтегровані середовища програмування для Python:

– IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.

– Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.

– Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.

– PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.

– Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.

– Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

– PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

– Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

– Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 17   |

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

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.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 18   |

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Протягом всієї історії ВКЗ одним з головних векторів розвитку було підвищення якості зображення. В останні роки з'являлися повідомлення про розробку систем з підтримкою тривимірного відео (3D), а з недавніх пор на ринку стали з'являтися засоби відображення з дозволом 4k. Однак відповідні системи ВКЗ поки не пішли «у тираж». Ми вирішили з'ясувати, яке якість зображення влаштовує замовників. Виявилось, що половині респондентів цілком достатньо якості HD 720p. 16% улаштовує навіть якість, типове для сервісу Skype. Про існування потреби в такій «екзотиці» (для систем ВКЗ), як дозвіл 4k і підтримка 3D, заявили по 5% замовників. Цього явно недостатньо для того, щоб виробники кинули основні сили на розробку (ну або доробку) систем з такими можливостями.

Для них доцільніше зосередитися на підтримці інших тенденцій, таких як мобілізація й софтверизація ВКЗ. Майже 60% опитаних як найбільше важлива тенденція (для впровадження рішень ВКЗ у своїй компанії) назвали поява в користувачів мобільних пристроїв можливості брати участь у сеансах ВКЗ, у першу чергу зі смартфонів і планшетів на базі iOS і Android. Це стало можливим в останні роки завдяки, по-перше, значному підвищенню продуктивності таких пристроїв, а по-друге – розвитку мереж бездротового зв'язку, включаючи впровадження технологій 3G і LTE, а також збільшення числа зон Wi-Fi, розширення їхнього покриття, підвищення швидкості і якості зв'язку в них. Відповідно, основні виробники реалізували у своїх рішеннях ВКЗ підтримку найбільш популярних мобільних платформ. Головними перешкодами на шляху масового використання мобільного відеозв'язку залишаються психологічна неготовність користувачів, а також побоювання, пов'язані з безпекою при використанні моделі BYOD.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 19   |

Програмні клієнти на мобільних пристроях – це одна із численних граней процесу софтверизації рішень ВКЗ. Ще однієї є поява програмних серверів багатоточечного відеозв'язку (MCU) – інфраструктурних продуктів, які традиційно, через високе навантаження при перекодуванні відеопотоків, реалізовувалися у вигляді апаратних рішень. Майже половина опитаних нами фахівців відзначили важливість софтверизації інфраструктури ВКЗ, оскільки це знижує загальну вартість системи й підвищує гнучкість впровадження й масштабування. Програмні MCU сьогодні пропонують як традиційні постачальники систем ВКЗ, такі як Polycom і LifeSize, так і нові гравці – наприклад, норвезька компанія Рехір, заснована вихідцями з Tandberg.

Можливість програмної реалізації серверів MCU з'явилася багато в чому завдяки підвищенню продуктивності серверів стандартної архітектури. Як затверджують представники Рехір, сервери із всім числом, що збільшується, гіперпотоків ядер на одному кристалі й розширених системах команд, такими як SSE і AVX, забезпечують навіть кращу продуктивність, ніж спеціалізовані апаратні пристрої, що використовують процесори ASIC і DSP. Наприклад, на базі одного стандартного сервера висотою 1U будь-якого провідного виробника (HP, Dell, IBM, Cisco і ін.) зі здвоєними центральними процесорами Intel серії E 5-2600 по 8 ядер із частотою 2,7 ГГц кожний система Рехір здатна підтримувати 32 порту сполучення відео-конференц-зв'язку високої чіткості. По параметрі «число HD-портів на одне непохитно-місце» такий MCU не уступає спеціалізованим апаратним рішенням. При використанні блейд-серверів можна домогтися щільності більше 1000 портів HD-відеозв'язку на 10U у стійці.

Важливою перевагою програмної інфраструктури ВКЗ є й те, що відповідні продукти можуть бути розгорнуті у віртуальному середовищі. У результаті замовник автоматично одержує всі переваги роботи в такому середовищі, включаючи масштабованість і відказостійкість. Наприклад, МС може «переїхати» з одного ЦОД в іншій у випадку виникнення яких-небудь проблем або з метою зниження навантаження на канали зв'язку (коли нова площадка розташовується ближче до учасників сеансу ВКЗ).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 20   |

Говорячи про перспективи розвитку наявної системи ВКЗ, велика кількість респондентів (60%) відзначили важливість інтеграції з корпоративної АТМ. Це, по суті, означає, що на підприємстві з'явиться комунікаційна система з високим ступенем уніфікації. Традиційно багато фахівців (44%) зацікавлені в забезпеченні можливості підключення до системи ВКЗ клієнтів сервісів класу Skype. Це й не дивно, оскільки 30% опитаних взагалі називають Skype оптимальним сервісом для проведення відеоконференцій у своїй компанії.

Відносно невелике число респондентів (22%) вважають найважливішим для розвитку (впровадження) системи ВКЗ перехід на хмарну модель. Далі про це докладніше.

Майже три чверті (точніше, 71%) учасників повністю ігнорують хмарну модель як спосіб надання комунікаційних сервісів. Приблизно кожний десятий одержує із хмари сервіси електронної пошти й Web-конференцій. (Загалом кажучи, Web-конференції реалізуються переважно саме по хмарній моделі.) Відповідний показник для ВКЗ значно нижче – 6% респондентів споживають «ВКЗ із хмари». Нарешті, як з'ясувалося, з опитаних нами тільки одна компанія використовує віртуальну АТМ. Це можна пояснити тим, що основна частина наших респондентів представляє досить великі (більше 500 комп'ютерів) компанії, тоді як віртуальними АТМ в основному користується малий і середній бізнес.

Традиційна більшість замовників в Україні воліють, щоб все встаткування й ПЗ перебували «усередині» компанії. Як показало наше опитування, половина компаній вважають кращим сценарій, коли всі компоненти комунікаційної системи (АТМ, ВКЗ, УфК) встановлюються в компанії, а також управляються й обслуговуються її фахівцями. 13% готові передати керування провайдерів-сервісу-провайдеру при збереженні інфраструктури у власності. Разом з тим стрімко набирає популярність гібридна модель, при якій частина інфраструктури залишається в компанії, а при епізодичному сплеску навантаження задіюються зовнішні ресурси (публічна хмара).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 21   |

Повністю відмовитися від власної інфраструктури й одержувати комунікаційні послуги із хмари в майбутньому планують лише 10% опитаних. Як головна перешкода на шляху в хмару очікувано називаються «побоювання щодо несанкціонованого доступу до переданої інформації» (66%). Інші названі перешкоди теж досить передбачувані: проблеми з каналами зв'язку; відсутність пропозицій із прийнятним співвідношенням ціна/функціональність; труднощі інтеграції з іншими корпоративними системами. При цьому слід зазначити досить високий ступінь довіри замовників до самих хмарних платформ – у їхню надійність не вірить лише кожний п'ятий наш респондент.

Підбиваючи підсумок, ще раз підкреслю, що саме впровадженню (розвитку) систем уніфікованих комунікацій з підтримкою відеозв'язку опитані нами компанії мають намір приділяти найбільшу увагу. Крім цього, гарні перспективи в Web-конференцій, теж з підтримкою відеозв'язку. Інтерес до ВКЗ «у чистому виді» поступово падає: якщо рік назад за пріоритетний розвиток цього напрямку висловилися 38% компаній, то в цьому – тільки 30%. Що ж стосується моделей розгортання, те найбільше що стрімко набирає популярність варіант – це гібридна модель. Відповідно, постачальникам і системним інтеграторам варто приділити максимум уваги забезпеченню можливості «безшовно» переводити навантаження із частки ЦОД у публічний, а також у зворотному напрямку.

### 3.2 Розробка структурної схеми

Уніфіковані комунікації – це багатофункціональне рішення, що, як правило, складається з декількох. Далеко не всі виробники здатні надати всі ці компоненти, тому найчастіше для одержання повнофункціонального рішення УфК потрібно інтегрувати продукти декількох компаній. Базовим процесом такої інтеграції стає сполучення систем відеозв'язку (ВКЗ) і телефонії (УАТМ).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 22   |









засобів зв'язку, менш сприйнятливих до якості з'єднання (приміром, електронної пошти), додатка реального часу пред'являють цілий ряд вимог до локальних мереж і з'єднань WAN. Так, їм необхідна стабільна пропускна здатність (залежно від якості, звичайно потрібно від 0,5 до 2 Мбіт/с, а передача зображень в Full HD можлива при швидкості з'єднання від 1 Мбіт/с). Крім того, вони дуже чутливі до більших затримок і їхніх варіацій (Jitter), втраті пакетів або інших перешкод, що виникають при передачі пакетів даних, тому мають потребу в досить високій якості сервісу (Quality of Service, QoS).

Ситуація збільшується тим обставиною, що мережний адміністратор обмежений у своїх можливостях впливати на ці параметри. Усередині корпоративної мережі на базі протоколу IP або технології MPLS відділ IT ще може регулювати розподіл пропускної здатності мережі, якість послуг і тривалість затримок. Якщо ж задіяні лінії далекого зв'язку, які орендуються в стороннього сервісу-провайдеру, то мережні адміністратори стають неспроможні. А для інтернет-з'єднань взагалі можна гарантувати лише передачу даних «у міру можливості» (Best Effort).

Для оптимального використання засобами відеозв'язку доступної пропускної здатності мережі не обійтися без алгоритмів стиску, таких як H.264, H.264 High Profile і H.264 SVC (Scalable Video Coding). У якості стандартних комунікаційних протоколів повсюдно затвердилися стандарти SIP і H.323.

У випадку проведення відеоконференцій з якістю HD, дозволом 720p і частотою зміни зображення 30 кадрів у секунду, для кодека H.264 потрібна пропускна здатність від 700 Кбіт/с, а для Full HD з 1080p – від 1 до 5 Мбіт/с. Реальна потреба в мережних ресурсах залежить від декількох факторів, у тому числі від можливостей устаткування для відеоконференцій, системних налаштувань, частоти зміни кадрів, а також від того, чи будуть крім відео передаватися інші типи даних.

Як показує досвід, для передачі відеозображень у гарній якості при дозволі 720p необхідно не менш 1 Мбіт/с, а для 1080p – близько 2 Мбіт/с. Якщо

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 27   |

потрібно забезпечити дуже гарна якість для зображення з високою динамікою, варто розраховувати вже на 2 Мбіт/с (для 720p) і на 4 Мбіт/с (для 1080p). Помітимо, кодек H.264 High Profile забезпечує дуже висока якість при помірному навантаженні на мережу.

Додаткову складність створює вся зростаюча кількість користувачів, що бажають одержати доступ до додатків для UCC і спільної роботи зі своїх мобільних пристроїв (смартфонів і планшетних ПК). Оскільки ці пристрої, як правило, мають порівняно невеликі екрани, відеодані доводиться під них підбудовувати. Крім того, комунікація за допомогою подібних пристроїв здійснюється через мережі стільникового зв'язку, WLAN і Інтернет, а виходить, адміністратор корпоративної мережі не може вплинути на якість з'єднання й сервісу.

Типові проблеми таких неадмініструємих з'єднань – висока завантаженість, занадто більша втрата пакетів (Packet Loss), що часом перевищує 1%, а також тривалі затримки при передачі даних (200 мс і вище). Зазначені проблеми можна вирішити за допомогою систем, що підтримують стандарт H.264 SVC. Він забезпечує створення масштабованого відеопотоку, що підбудовується під продуктивність пристрою, розмір екрана і його програмне оснащення. Крім того, ця технологія здатна враховувати параметри пропускної здатності й рівень (у процентному співвідношенні) втрати пакетів. У результаті під час сеансів спільної роботи з відеозв'язку вдається підтримувати максимальне – для конкретного кінцевого пристрою при наявному мережному підключенні – якість зображення.

Наприклад, SVC може перекодувати вихідний відеопотік з дозволом 720p і частотою 30 кадрів у секунду в дозвіл 480p при 30 кадрах у секунду або у формат qCIF, придатний для невеликих мобільних пристроїв (розмір зображення 176 x 144 пікселей, частота 15 кадрів у секунду). Крім того, SVC має механізми, що знижують наслідку від втрати пакетів. Виміру показують, що стандартні системи для відеоконференцій переносять втрату пакетів максимум до 1%. У свою чергу, механізми виправлення помилок і багаторівнева структура кодека

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 28   |

H.264 SVC дозволяють забезпечити прийнятна якість відображення відео, навіть якщо втрата пакетів перевищує 20%.

Важливий аспект інфраструктури для UCC і спільної роботи з відеозв'язку на базі IP – її масштабованість. Такі рішення можна використовувати скрізь, де є з'єднання IP. При цьому підтримуються різні категорії систем, починаючи від окремих кімнат для віртуальних переговорів до настільних клієнтів і мобільних пристроїв. До такої інфраструктури можна підключити користувачів, що перебувають як усередині корпоративної локальної мережі, так і поза нею. Крім того, підхід на базі IP дозволяє підприємствам здійснювати безперервну міграцію спочатку з офісних VoIP-УАТМ на настільні системи для відеоконференцій, а потім на інтегровані рішення UCC. Ще один позитивний аспект середовища на базі протоколу IP – у ній відсутня прив'язка до певного апаратного забезпечення.

Відповідно до дослідження, проведеному консалтинговим агентством Forrester Consulting за замовленням Avaya, однієї з основних проблем, що утрудняють впровадження відеокommунікаційних рішень, є недостатня сумісність мережної й програмної інфраструктури з компонентами рішень для відеозв'язку. Скориставшись радами експертів, фахівці IT зможуть обійти цю перешкоду:

Переклад протоколів. Всі кінцеві пристрої, відеоплюзи й сервери багатоточечного відеозв'язку варто набувати таким чином, щоб вони «розуміли» використовувані протоколи або могли перетворити їх у формат, зрозумілий для відеосистем. Такими протоколами є, приміром, H.323, SIP і кодеки H.264 або більше старий H.263. Для специфічних протоколів окремих виробників, таких як Telepresence Interoperability Protocol (TIP) або RTVideo, можна застосовувати шлюзи. Варто також брати до уваги, що існують різні версії протоколів, не сумісні між собою (приміром, H.264 і H.264 Annex G).

Оптимізація пропускної здатності. Шлюзи й пристрої MCU повинні бути настроєні таким чином, щоб адаптувати відеопотік з урахуванням можливостей кінцевих пристроїв, на які він направляєється. Так, для перегляду відео в гарній якості (15 кадрів у секунду) на порівняно невеликому екрані смартфона підійде

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 29   |

навіть канал із пропускною здатністю 200 Мбіт/с, якщо для стиску буде використовуватися кодек H.264. А от для системи відеозв'язку з передачею зображення якості HD, 60-дюймовим екраном і дозволом 1920 x 1080 пікселів при 30 кадрах у секунду буде потрібно вже не менш 1 Мбіт/с у випадку H.264 High Profile.

Стандартизація адресації. Щоб адресація в ІТ-середовищах, де розгорнуті рішення для відеоконференцій і спільної роботи з відеозв'язку від різних виробників, функціонувала без проблем, варто використовувати стандартизовані протоколи (SIP або H.323). А для підключення кінцевих пристроїв, що перебувають за межами корпоративної мережі, ці протоколи повинні підтримуватися архітектурою брандмауера.

Одне з найбільш складних завдань для адміністраторів – інтеграція відеозв'язку в інфраструктуру уніфікованих комунікацій (Unified Communication, UC), що поєднує мобільні пристрої, мультимедійні елементи й засоби колективної роботи (Social Collaboration). Центральними елементами такої інфраструктури є відеоконференції (їх можна проводити за допомогою найрізноманітнішого кінцевого встаткування), рішення UCS, а також корпоративна мережа, що дозволяє інтегрувати мобільні пристрої співробітників.

Одне із уже реалізованих рішень для спільної роботи з використанням відеозв'язку створено компанією Avaya для системи уніфікованих комунікацій IP Office. Воно являє собою віртуальний конференц-зал, у якому користувачі можуть бачити інших учасників, спілкуватися з ними й спільно розглядати документи. Його базовими елементами є сервер відео^-конференц-зв'язку (підтримують клієнти під керуванням Windows і Mac, а також мобільні клієнти під Android і IOS) і система для створення віртуальних переговорних кімнат (якість HD). Багатоточечний керуючий модуль HD-MCU (Multipoint Control Unit) дозволяє об'єднати різні кінцеві пристрої в єдиній конференції й забезпечити підтримку систем стандарту H.323 і SIP.

Важливо, що за допомогою такого рішення можна здійснювати комунікацію й з тими учасниками, які перебувають за межами корпоративної

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 30   |

мережі (приміром, з мобільними співробітниками або діловими партнерами). Тому воно оснащено інтегрованими функціями брандмауера й NAT Traversal. Крім того, підприємствам варто звертати увагу на наявність можливості інтеграції служб каталогів, зокрема Active Directory, і номерних планів.

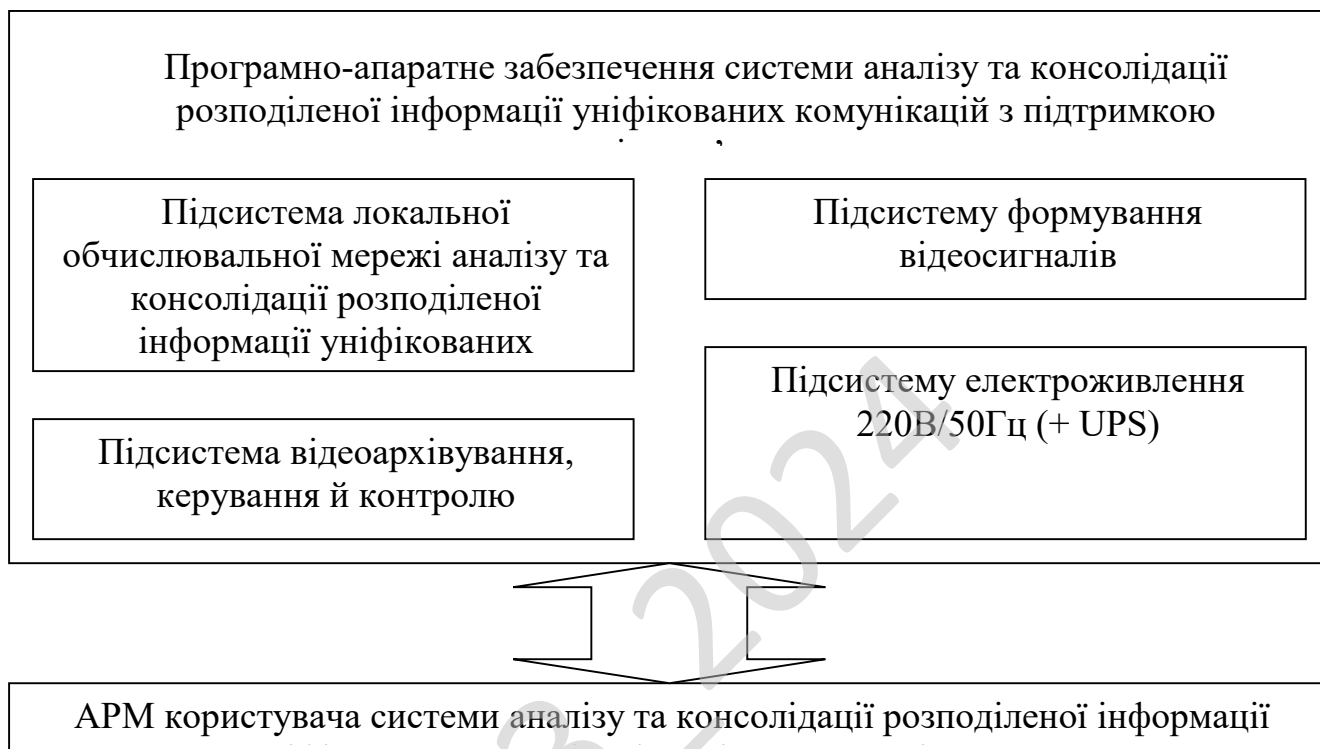


Рисунок 3.1 – Структурна схема роботи системи

Система аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку містить:

– Підсистему формування відеосигналів, що складається з камер аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

– Підсистему відеоархівування, керування й контролю, що містить сервера, автоматизовані робочі місця – комп'ютери й відеомонітори.

– Підсистему локальної обчислювальної мережі безпеки, що поєднує АРМ операторів і сервера системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

– Підсистему електроживлення 220В/50Гц, що містить джерела безперебійного живлення (UPS) для гарантованого електроживлення встаткування системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку протягом заданого часу.

– Програмно-апаратне забезпечення системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

### 3.3 Розробка функціональної схеми

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

Комунікації:

– Відеоконференції – Багатоточечні Full HD конференції, – 140 одночасних інтерактивних відео учасників.

– Відеодзвінки – Миттєві двосторонні голосові й відеодзвінки (P2P) усередині корпоративної мережі.

– Індикація доступності (presence) – Відображення онлайн-статусу (у мережі, немає на місці й т.д.).

– Підтримка SIP – Сумісність із апаратними ВКЗ і MCU.

– Підтримка WebRTC – Вхід у конференцію через будь-який браузер на основі Chrome без установки плагіна.

– Обмін ІМ-повідомленнями – Миттєве відправлення й одержання повідомлень.

Спільна робота:

– Показ робочого стола – Трансляція робочого стола.  
– Трансляція додатків – Показ будь-яких документів і мультимедійних файлів.

– Електронна дошка – Редагування текстових і графічних даних.

– Обмін файлами – Обмін файлами між учасниками конференції.

– Протокол – Мультифункціональний режим ведення протоколів нарад.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 32   |

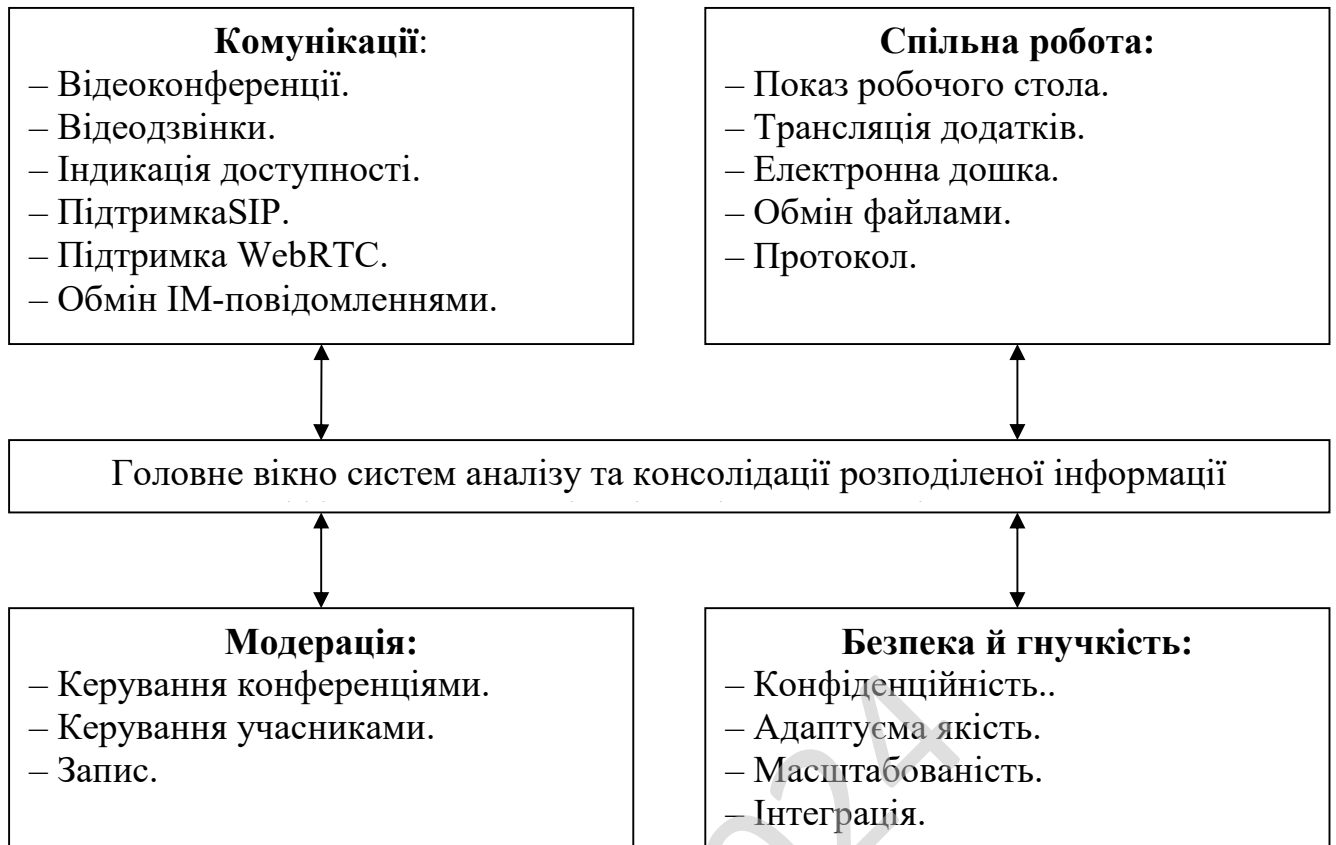


Рисунок 3.2 – Функціональна схема системи

**Модерація:**

- Керування конференціями – Зручні інструменти для створення й керування конференціями, запрошення учасників.
- Керування учасниками – Передача прав модератора, надання слова, відключення мікрофона, відключення від конференції.
- Запис – Трансляція, запис і перегляд відеоконференцій.

**Безпека й гнучкість:**

- Конфіденційність – Відеоконференції на серверах під вашим контролем, підтримка шифрування.
- Адаптуєма якість – Кожний учасник одержує максимально можливе для нього – якість відео зображення, що залежить від потужності ПК і – якості зв'язку й не залежить від інших учасників.



При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPM-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 35   |

## 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.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

#### Опис алгоритмів функціонування системи

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю систем аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку. При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 36   |

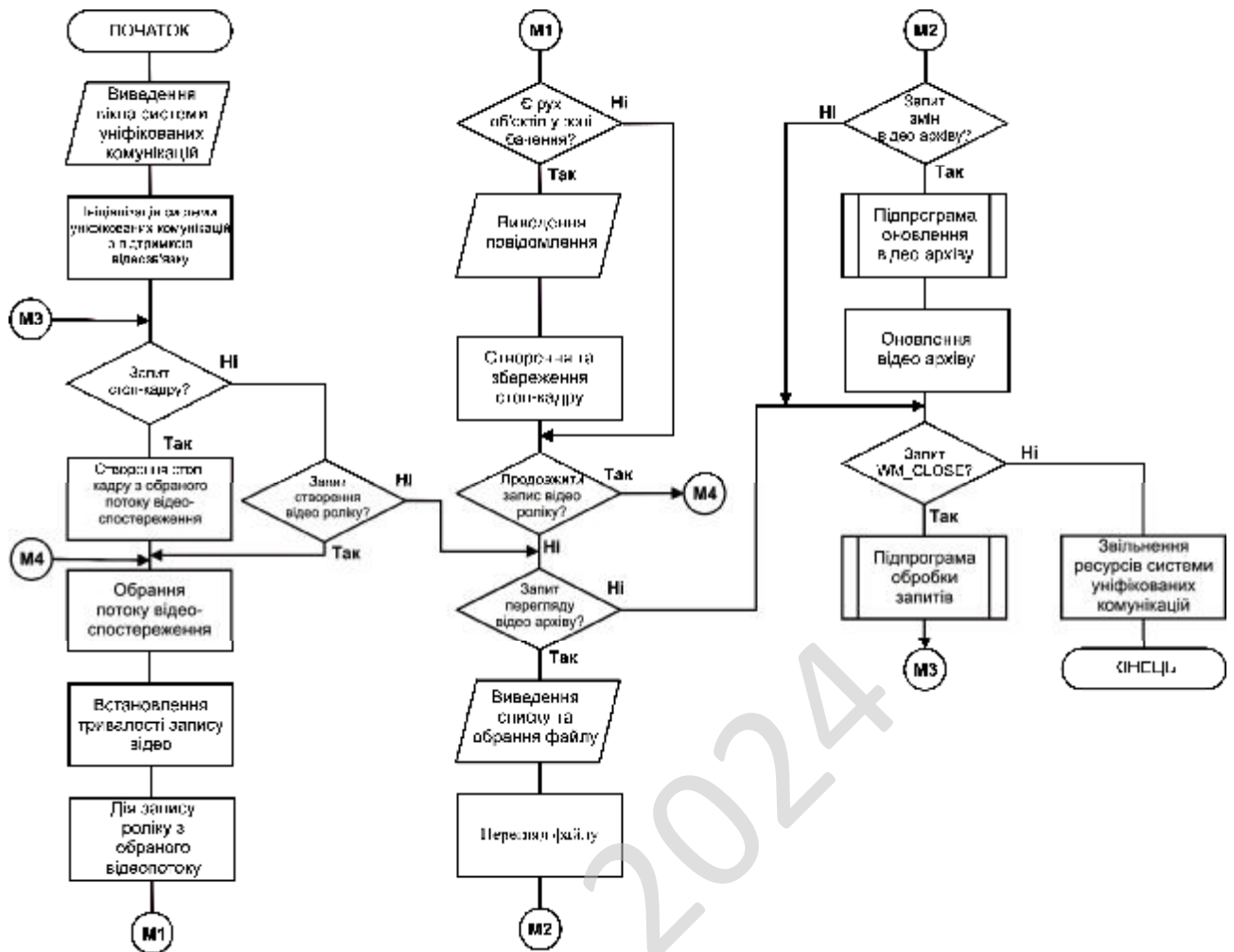


Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які

підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

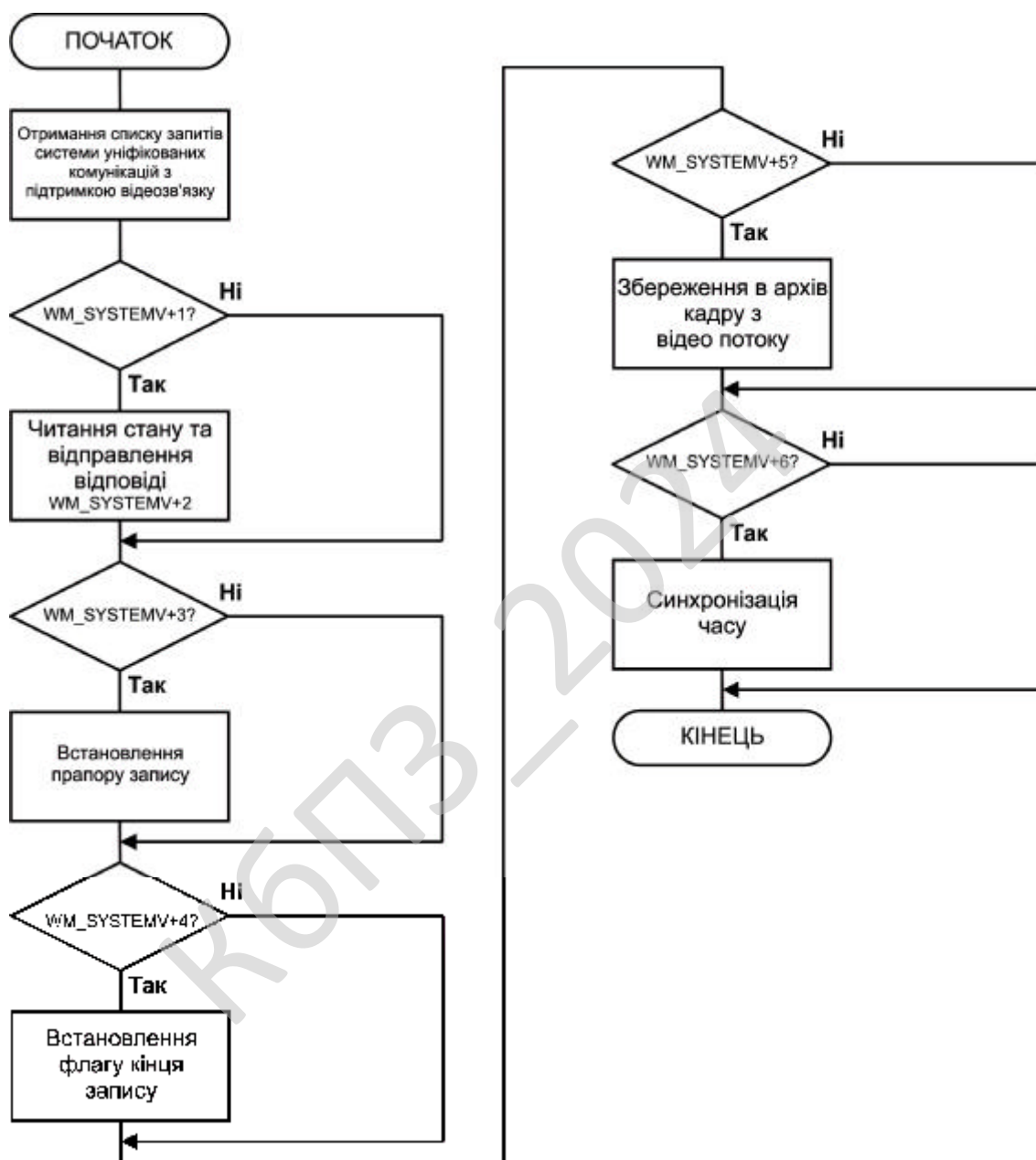


Рисунок 4.2 – Блок-схема роботи підпрограми

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників

між собою. Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

- Керівникам проектів, які керують розподілом завдань і контролем за проектом.
- Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.
- Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.
- Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

Також при розробці магістерської дипломної роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів.



При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>41</b> |



діапазон: "нуль або одиниця" (0..1), "багато" (0 .. \*), "одиниця або більше" (1 .. \*).  
Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. \*, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

При розробці ПЗ мені необхідно було точно знати розмір пікселя, область бачення, співвідношення сторін і кількість крапок на дюйм для різних діагоналей і роздрукованих дозволів моніторів.

Реалізований стиск зображень у розробленому ПЗ. Було реалізовано стиск зображень за допомогою відомого та найпоширенішого алгоритму JPEG. Система забезпечує аналіз, консолідацію розподіленої інформації та підтримку уніфікованих комунікацій з функцією відеозв'язку.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 43   |

Система складається з таких основних модулів:

1. Модуль збору даних. Забезпечує отримання розподіленої інформації від клієнтів.
2. Модуль обробки даних. Аналізує отримані дані та проводить їх консолідацію.
3. Модуль комунікацій. Реалізує функціонал відеозв'язку.
4. Модуль зберігання даних. Забезпечує запис та збереження результатів уніфікації.

Архітектура відповідає принципу розподілених обчислень з використанням клієнт-серверної моделі.

```
import threading
import socket
import cv2
import pickle
import struct
import json
import os
from queue import Queue

class DataCollector:
    def __init__(self, data_source):
        self.data_source = data_source
        self.data_queue = Queue()

    def collect_data(self):
        while True:
            data = self.data_source.get_data()
            self.data_queue.put(data)

class DataProcessor:
    def __init__(self):
        self.consolidated_data = {}

    def process_data(self, data):
        if data['id'] not in self.consolidated_data:
            self.consolidated_data[data['id']] = []
        self.consolidated_data[data['id']].append(data['value'])
```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 44   |

```

def consolidate_data(self):
    return {k: sum(v)/len(v) for k, v in
self.consolidated_data.items()}

class VideoCommunication:
    def __init__(self, host, port):
        self.host = host
        self.port = port

    def start_server(self):
        server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server_socket.bind((self.host, self.port))
        server_socket.listen(5)
        print("Server started. Waiting for connections...")
        while True:
            client_socket, addr = server_socket.accept()
            threading.Thread(target=self.handle_client,
args=(client_socket,)).start()

    def handle_client(self, client_socket):
        payload_size = struct.calcsize("Q")
        data = b""
        while True:
            while len(data) < payload_size:
                packet = client_socket.recv(4 * 1024)
                if not packet:
                    break
                data += packet
            packed_msg_size = data[:payload_size]
            data = data[payload_size:]
            msg_size = struct.unpack("Q", packed_msg_size)[0]
            while len(data) < msg_size:
                data += client_socket.recv(4 * 1024)
            frame_data = data[:msg_size]
            data = data[msg_size:]
            frame = pickle.loads(frame_data)
            cv2.imshow('Receiving Video', frame)
            if cv2.waitKey(1) == 13:
                break
        client_socket.close()

    def start_client(self, server_ip):

```

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>БКРМ-122.24.0004.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>45</b> |

```

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((server_ip, self.port))
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    data = pickle.dumps(frame)
    message_size = struct.pack("Q", len(data))
    client_socket.sendall(message_size + data)
    cv2.imshow('Sending Video', frame)
    if cv2.waitKey(1) == 13:
        break
client_socket.close()
cap.release()
cv2.destroyAllWindows()

class DataStorage:
    def __init__(self, file_path):
        self.file_path = file_path

    def save_data(self, data):
        with open(self.file_path, 'w') as f:
            json.dump(data, f)

    def load_data(self):
        if os.path.exists(self.file_path):
            with open(self.file_path, 'r') as f:
                return json.load(f)
        return {}

if __name__ == "__main__":
    collector = DataCollector(data_source=None) # Placeholder for data
source implementation
    processor = DataProcessor()
    storage = DataStorage(file_path="data.json")

    # Start video communication server in a separate thread
    video_comm = VideoCommunication(host='0.0.0.0', port=9999)
    threading.Thread(target=video_comm.start_server).start()
    # Placeholder for data collection and processing logic
    # Collect and process data

```

## 4.2 Захист розробленого програмного забезпечення

Дані у програмному забезпеченні захищаю за допомогою MISTY1. MISTY1 – блоковий алгоритм шифрування, створений для компанії Mitsubishi Electric криптологом Міцурю Мацую. Назва є аббревіатурою Mitsubishi Improved Security Technology. Алгоритм був розроблений в 1995-1996 рр. Відомі також дві модифікації алгоритму MISTY1: MISTY2 і KASUMI

Шифр став переможцем на Європейському конкурсі NESSIE. У результаті аналізу алгоритму експерти зробили вивід, що ніяких серйозних уразливостей даний алгоритм не має (переважно, завдяки вкладеним мережам Фейстеля, що суттєво утрудняє криптоаналіз). У нього високий запас криптостійкості, алгоритм має високу швидкість шифрування й досить ефективний для апаратної реалізації.

Алгоритм був розроблений на основі теорії «підтвердженої безпеки» проти диференціального й лінійного криптоаналізу. Цей алгоритм був спроектований, щоб протистояти криптоатакам, відомим на момент створення.

З моменту публікації MISTY1 було проведено багато досліджень, щоб оцінити його рівень безпеки.

Диференціальний і неможливий диференціальний криптоаналіз високого порядку ефективно застосовується до блокових шифрів з малим ступенем. Найкращі результати для обох варіантів були отримані для 5-рівневого алгоритму MISTY1 без FL функцій.

Саме FL функції й широкобітні AND/OR операції в сильно утрудняють використання диференціального криптоаналізу, що не заважає проведенню в цьому напрямку всі нових досліджень і досягненню усе більш близьких до розв'язку результатів.

### Параметри вихідних даних

MISTY1 – це шифр на основі вкладених мереж Фейстеля з варюємим числом раундів. Рекомендоване використання 8-раундової версії, але може

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 47   |

використовуватися будь-яка кількість раундів, кратне 4-м. Розмір блоку вихідного тексту – 64 біта, розмір ключа – 128 біт.

Для роботи алгоритму також попередньо виконується процедура розширення ключа, яка для 8-мі раундів обчислює 1216 бітів ключової інформації з 128-бітного ключа шифрування.

### **Структура алгоритму**

Для задоволення вимогам конкурсу NESSIE, а також для задоволення завдання мультиплатформеності, в алгоритмі MISTY1 використовувалися наступні методи шифрування:

- Логічні операції.
- Арифметичні операції.
- Операції зрушення.
- Таблиці перестановок.

Як говорилося вище, алгоритм MISTY1 заснований на «вкладених» мережах Фейстеля. Спочатку блок вихідного тексту розбивається на два 32-бітних субблоки, після чого виконується  $r$  раундів наступних перетворень[1]:

- У кожному непарному раунді обидва субблоки обробляються операцією FL
- Над обробленим субблоком виконується операція FO.
- Результат цих операцій накладається логічною операцією «, що виключає або» (XOR) на неопрацьований субблок.
- Субблоки міняються місцями. Після заключного раунду обидва субблоки ще раз обробляються операцією FL.

### **Операція FL**

Оброблюваний 32-бітний субблок розбивається на два 16-бітних фрагмента, до яких застосовуються операції, де:

- $L$  і  $R$  – вхідні значення лівого й правого фрагментів відповідно;
- $L'$  і  $R'$  – вихідні значення;

–  $i$  – фрагменти  $j$ -го підключа  $i$ -го раунду для функції FL (процедура розширення ключа докладно описана далі);

–  $i$  – побітві логічні операції «і» і «або» відповідно.

### **Операція FO**

Саме ця функція є вкладеною мережею Фейстеля. Тут, як і раніше, виконується розбивка вхідного значення на два 16-бітних фрагмента, що проходять 3 раунду наступних перетворень:

– На лівий фрагмент операцією XOR накладається фрагмент ключа, де  $k$  – номер раунду функції FO.

– Лівий фрагмент обробляється операцією FI.

– На лівий фрагмент накладається операцією XOR значення правого фрагмента.

– Фрагменти міняються місцями.

Після третього раунду операції FO на лівий фрагмент накладається операцією XOR додатковий фрагмент ключа.

### **Операція FI**

Дана операція також представляє собою третій рівень вкладеності мережі Фейстеля. На відміну від двох верхніх рівнів, дана мережа є незбалансованою: оброблюваний 16-бітний фрагмент ділиться на дві частини: 9-бітну ліву й 7-бітну праву. Потім виконуються 3 раунду перетворень, що впливають:

– Ліва частина зазнає обробці S-box. 9-бітна частина (в 1-м і 3-м раундах) обробляється таблицею S9, а 7-бітна (в 2-м раунді) – таблицею S7. Дані таблиці описані нижче.

– На ліву частину операцією XOR накладається поточне значення правої частини. При цьому, якщо праворуч 7-бітна частина, вона доповнюється нулями ліворуч, а в 9-бітної частини віддаляються ліворуч два біти.

– У другому раунді на ліву частину операцією XOR накладається фрагмент ключа раунду, а на праву – фрагмент. В інших раундах ці дії не виконуються.

– Ліва й права частини міняються місцями.

Для оптимального розв'язку завдання мультиплатформеності, таблиці S7 і S9 алгоритму MISTY1 можуть бути реалізовані як за допомогою обчислень, так і безпосередньо таблицями.

### **Розширення ключа**

Для 8 раундів алгоритму результатом процедури розширення ключа буде наступний набір ключових значень:

- 20 фрагментів ключа (), кожний з яких має розмір по 16 бітів;
- 32 16-бітних фрагмента ();
- 24 7-бітних фрагмента (при  $k=4$ , тобто в 4-м раунді функції FO, операція FI не виконується);
- 24 9-бітних фрагмента.

Виконується дане обчислення в такий спосіб:

1. 128-бітний ключ ділиться на 8 фрагментів ... по 16 бітів кожний.
2. Формуються значення: у якості використовується результат обробки значення функцією FI, яка в якості ключа (тобто сукупності необхідних 7- і 9-бітних фрагментів) використовує значення (якщо індекс  $n$  фрагмента ключа перевищує 8, то замість нього використовується індекс  $n-8$ ).

Необхідні фрагменти розширеного ключа «набираються» у міру виконання перетворень із відповідних масивів і згідно з відповідними таблицями 16-бітний фрагмент ділиться на 7-бітний фрагмент і 9-бітний .

### **Розшифрування**

Розшифрування проводиться виконанням тих же операцій, що й при зашифруванні, але з наступними змінами:

- фрагменти розширеного ключа використовуються у зворотній послідовності,
- замість операції FL використовується зворотна їй операція – FLI.

Схеми виконання функції FLI і процедури розшифрування наведено на малюнках 6 і 7 відповідно:

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 50   |

## Методи аналізу

Як говорилося на початку розділу, диференціальний і неможливий диференціальний аналізи виявилися ефективні лише до версій шифру з меншою кількістю раундів і без операції FL [2][3]. Проте, на даний момент цей напрямок аналізу, особливе використання слабких ключів, найбільше перспективно, тому що наближене до реальних можливих допущень при використанні алгоритму.

Так само, ученим з Японії був проведений інтегральний аналіз повного алгоритму, використовуючи відкритих текстів зі складністю обчислення, рівної [4].

Лінійний аналіз дав результати тільки для 7-раундової версії шифру, і також без операції FL[5].

Так як MISTY1 створювався, у тому числі, з розрахунку на апаратну реалізацію, має сенс диференціальний аналіз, заснований на використанні атаки по помилках обчислень, що в цьому випадку наближене до реальності.

## Висновок

Таким чином, була докладно описана структура алгоритму шифрування MISTY1 і розглянуті методи його аналізу, найбільш прагматичні напрямки дослідження. Далі має бути створення програмної реалізації для більш детального розгляду алгоритму й набір статистичних даних для повного дослідження й пошуку оптимального підходу до аналізу MISTY1.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 51   |



Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

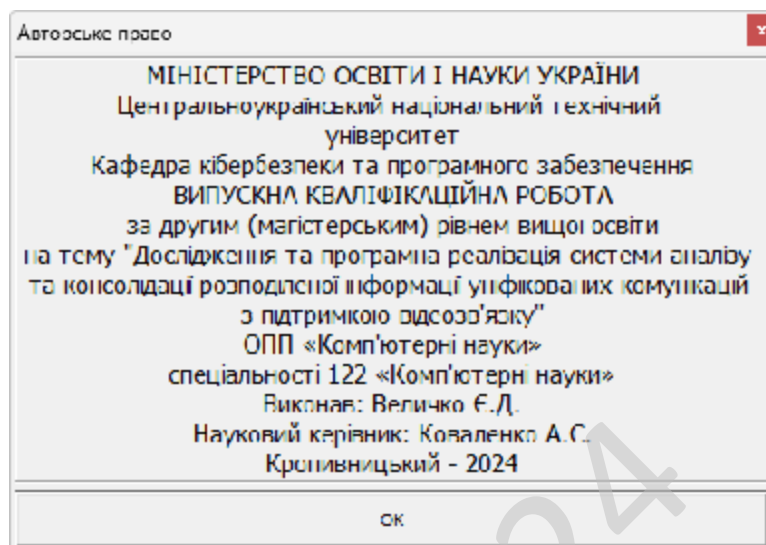


Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 53   |

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки та чорної скриньки.

Тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

- Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 54   |



Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;
- Помилки інтерфейсу;
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права.

Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію.

Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт.

Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення.

Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

КБПЗ\_2024

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPM-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 57   |

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

*Метою розробки є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.*

*Об'єктом дослідження є процес аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.*

*Предметом дослідження є методи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.*

*Методи дослідження базуються на методах теорії кодування та теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

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

– Удосконалено метод аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

– Розроблено вітчизняний продукт аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку, який має більш широкі можливості, на відміну від існуючих аналогів.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPM-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 58   |

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку можуть зацікавити (рисунок 7.1).

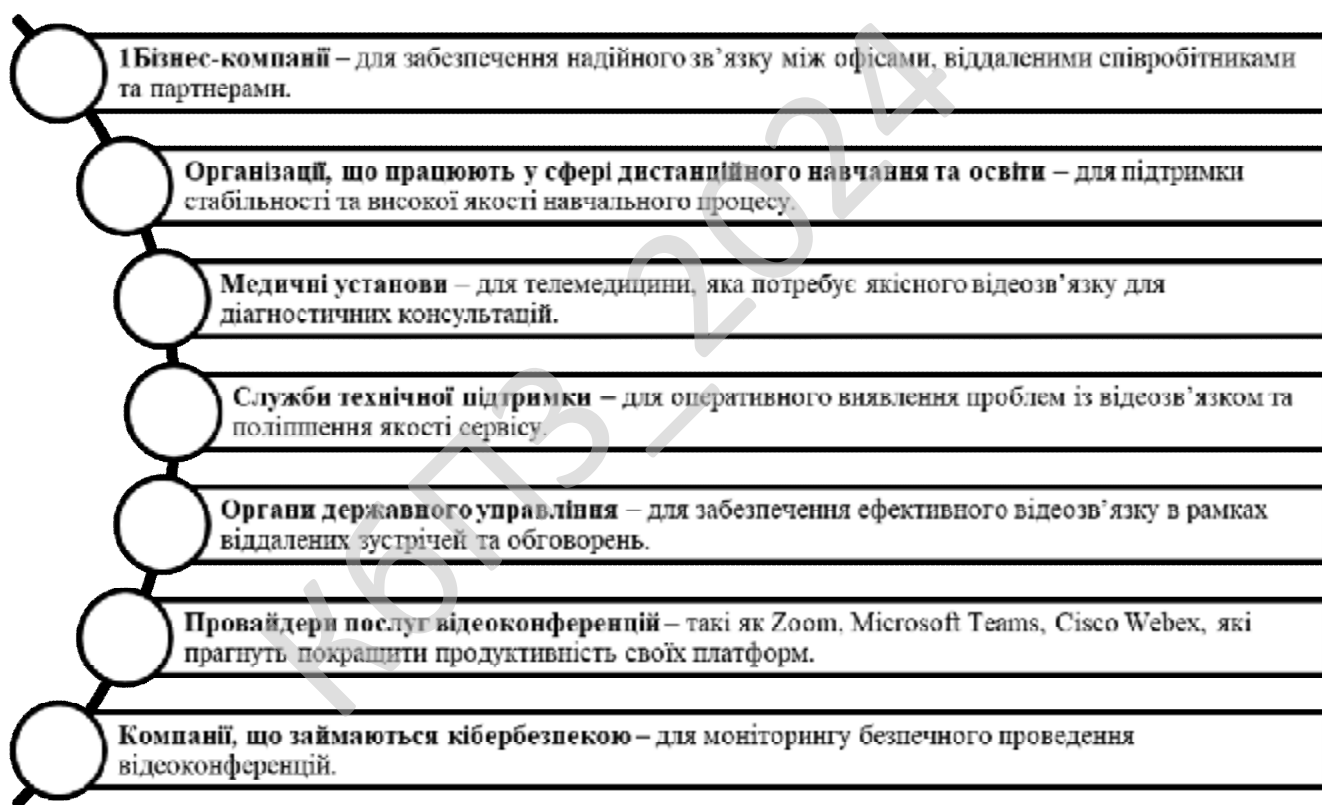


Рисунок 7.1 – Цільова аудиторія

Широка цільова аудиторіє може стати запорукою успішного просування проекту.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку за допомогою методу експертних оцінок може включати такі кроки: визначення критеріїв, формування експертної групи, розробка шкали оцінювання, оцінку експертами, обробка результатів.

Спочатку потрібно визначити ключові критерії, за якими експерти оцінюватимуть систему (рисунок 7.2).

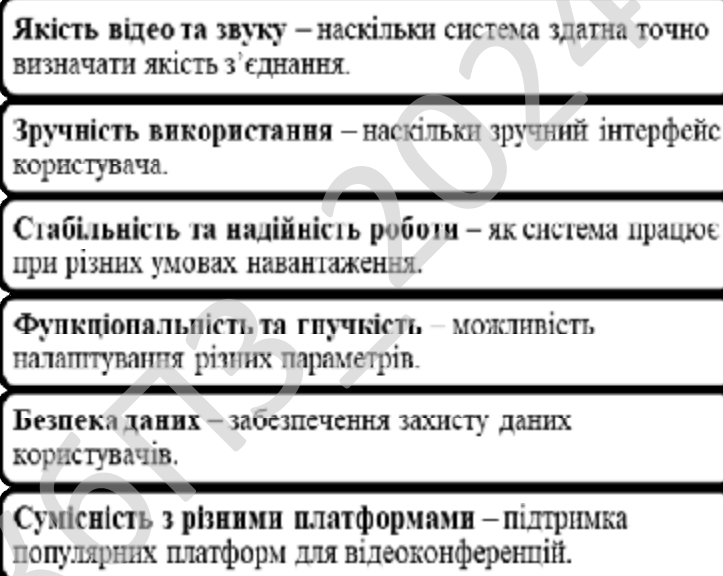


Рисунок 7.2 – Критерії експертної оцінки

Залучаються експерти у сфері відеоконференцій, інформаційних технологій, UI/UX дизайну, кібербезпеки та підтримки клієнтів. Обрано групу з 2 експертів.

Слід використовувати 5-бальну шкалу, де 1 – дуже низький рівень привабливості, а 5 – дуже високий.





## 7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Алгоритм просування проекту програмної реалізації системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку може включати етапи, наведені нижче (рисунок 7.4).

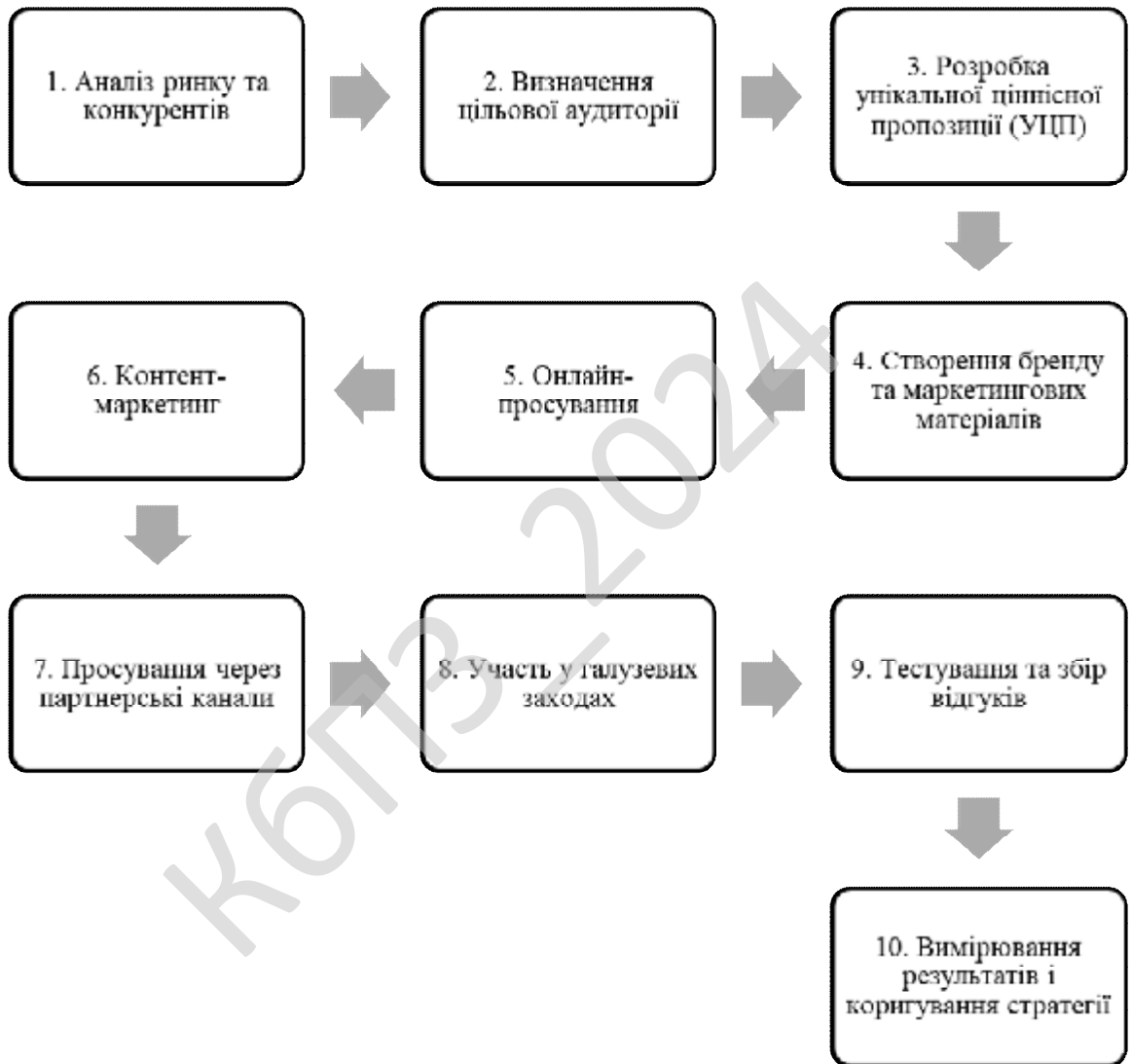


Рисунок 7.4 – Алгоритм просування проекту

Такі кроки дозволяють комплексно підійти до просування, охоплюючи ключові канали та сегменти цільової аудиторії, що підвищує ймовірність успішного впровадження та розширення проекту на ринку.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту для проєкту програмної реалізації аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку може включати такі стратегічні кроки (рисунок 7.5).

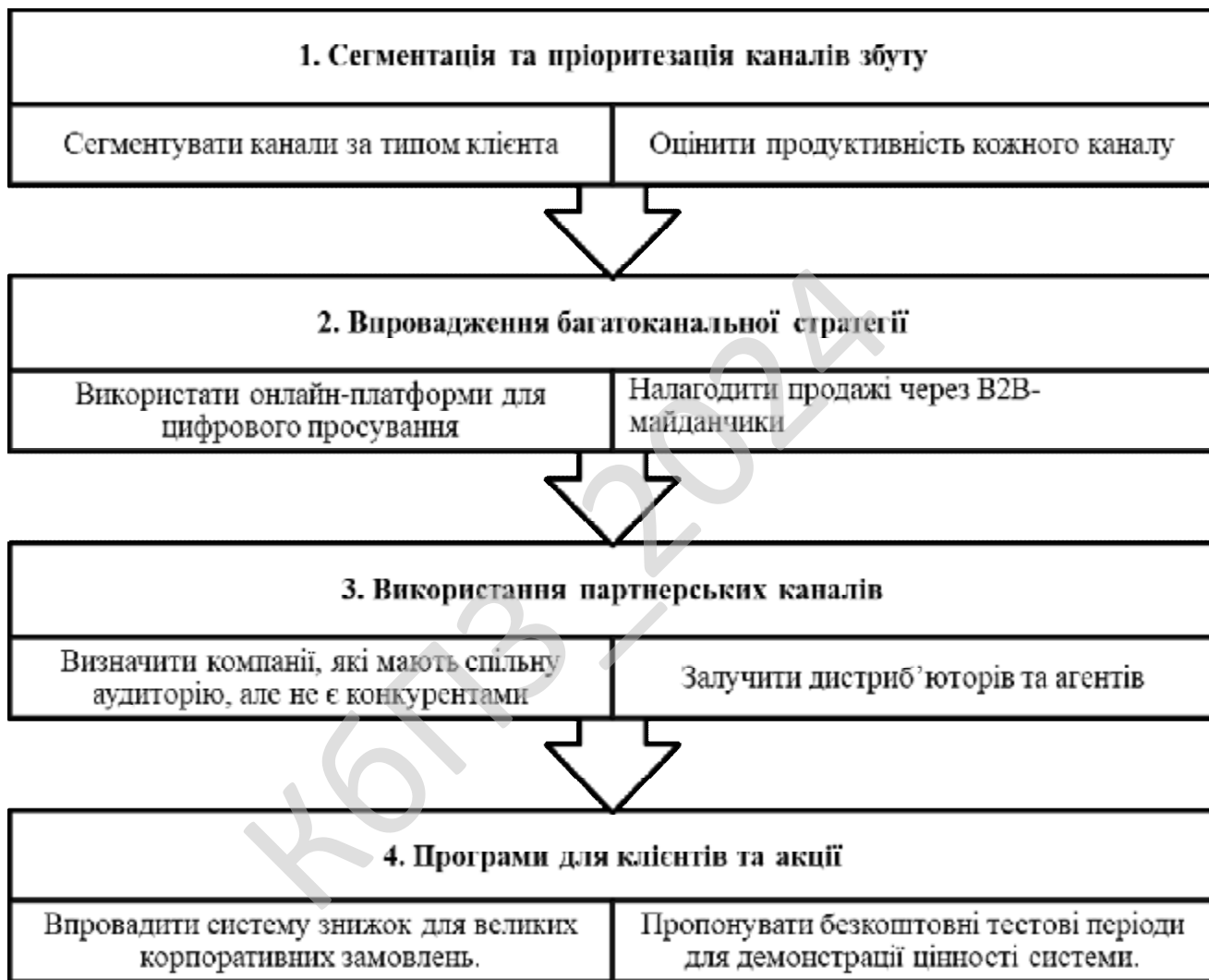


Рисунок 7.5 – Шляхи оптимізації каналів збуту

Ці кроки дозволять створити гнучку та продуктивну систему збуту, що підвищить обсяг продажів, розширить охоплення аудиторії та забезпечить стійкий ріст проєкту.

## 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху для проєкту програмної реалізації системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку включають (рисунок 7.6). Ці фактори створюють основу для успішного виходу продукту на ринок і забезпечують його стабільне зростання та високу репутацію серед користувачів.



Рисунок 7.6 – Шляхи оптимізації каналів збуту

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

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 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.

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 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 66   |

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.

## 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у тому числі високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 67   |

- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

| Найменування | Значення, м |
|--------------|-------------|
| Ширина       | 5,38        |
| Довжина      | 5,95        |
| Висота       | 2,8         |

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого\*

| Геометрична характеристика | Одиниця виміру | Нормативне значення* | Фактичне значення |
|----------------------------|----------------|----------------------|-------------------|
| Площа, S                   | м <sup>2</sup> | не менше 6.0         | 8                 |
| Об'єм, V                   | м <sup>3</sup> | не менше 20.0        | 22,4              |

\* Згідно ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

У зазначеному приміщенні працюють четверо людей. За даними, які наведено у табл. 8.1- 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин». Таним чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають нормативним вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація.

Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 69   |



(під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк [1]/ Крім того, все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

#### **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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 71   |

наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

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 напрузі 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.

### 8.5 Розрахункова частина

Початкові дані для розрахунку захисного штучного заземлення: опір заземлювача, який нормується:  $R_{3Н} = 4 \text{ Ом}$ .

Для захисного штучного заземлення застосовуються вертикальні електроди з металевого прутка діаметром 30 мм ( $D = 30 \text{ мм} = 0,03 \text{ м}$ ) довжиною  $L=2,5 \text{ м}$  та горизонтальний електрод – металева полоса з перетином 40x4 мм. тип ґрунту – глина (питомий опір 40 Ом·м). Відстань між вертикальними заземлювачами (електродами)  $A=3 \text{ м}$ .

Глибина закладення горизонтального контура заземлення  $t = 0,8 \text{ м}$ .

Умовна товщина верхнього шару ґрунту:  $H=0,4 \text{ м}$ . Напруга – 220/380 В.

Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 8.1).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 72   |



Визначаємо коефіцієнт екранування вертикальних електродів  $K_{ев}=0,8$  при попередній (орієнтовній) кількості вертикальних електродів, яке дорівнює 4 [8].

Визначаємо необхідну кількість вертикальних заземлювачів (без врахування горизонтального заземлювача), при  $R_{3Н} = 4 \text{ Ом}$  :

$$N=R_0 / ( K_{ев} \cdot R_{3Н} ) = 20,1 / (0,8 \cdot 4) = 5,87 \approx 6 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси [8]:

$$L_{П} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 6 = 18,8 \approx 19 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси [8]:

$$\begin{aligned} R_{П} &= 0,366 \cdot (\rho_2 \cdot K_{П} / L_{П} ) \cdot \lg(2(L_{П} \cdot L_{П} ) / (K \cdot t)) = \\ &= 0,366 \cdot (40 \cdot 5 / 16) \cdot [\lg(2 \cdot 16 \cdot 16) / (0,04 \cdot 0,8)] = 16,5 \text{ Ом.} \end{aligned}$$

де  $K_{П} = 5$  – табличне значення коефіцієнта сезонності для відповідної кліматичної зони з'єднуючої полоси [8].

Загальний опір розтіканню електричного струму заземлювача [8]:

$$\begin{aligned} R &= (R_0 \cdot R_{П}) / ( R_0 \cdot \eta_{П} + N \cdot R_{П} \cdot K_{ев} ) = \\ &= (20,1 \cdot 16,5) / (20,1 \cdot 0,75 + 6 \cdot 16,5 \cdot 0,8) = 3,32 \text{ Ом.} \end{aligned}$$

де  $\eta_{П} = 0,75$  – табличне значення коефіцієнта екранування з'єднуючої полоси [8].

Умова  $R \leq R_{3Н}$  виконується ( $3,32 \leq 4 \text{ Ом}$ ).

### Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 74   |

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

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 аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.
- Досліджена система аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.
- На основі отриманих результатів досліджень створена програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

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.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 75   |

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

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

Програма реалізована на мові високого рівня Python. 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 Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм MISTY1.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 76   |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Величко Є.Д. Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024. Маценко В.Г.
2. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
3. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
4. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. - Д.: НГУ, 2016. - 187 с.
5. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
6. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
7. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.
8. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447
9. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0004.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 77   |

software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

10. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

11. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

12. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

13. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

14. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science*, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

15. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

16. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>БКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 78   |

17. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.
18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
19. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.
20. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.
21. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.
22. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
23. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

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

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

26. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

27. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

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

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

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

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

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

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

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

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

36. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

37. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

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

39. 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.

40. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

41. 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.

42. 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.

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

44. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

45. 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.

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

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

47. 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

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

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

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

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., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0004.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 83   |

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 |

|           |                |             |        |      |   |                    |       |         |
|-----------|----------------|-------------|--------|------|---|--------------------|-------|---------|
|           |                |             |        |      | <b>ВКРМ-122.24.0004.00.00.ТЗ</b>  |                    |       |         |
| Вим.      | Арк.           | № документа | Підпис | Дата |   |                    |       |         |
| Розробив  | Величко Є.Д.   |             |        |      | <i>Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку</i> | Літ.               | Аркуш | Аркушів |
| Перевірів | Коваленко А.С. |             |        |      |   | М                  | 1     | 6       |
| Н. Контр. | Коваленко А.С. |             |        |      |   | <b>ЦНТУ КН-23М</b> |       |         |
| Затв.     | Смірнов О.А.   |             |        |      |   |                    |       |         |

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

## 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. № 18-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку.

## 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;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-122.24.0004.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 2    |

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи аналізу та консолідації розподіленої інформації уніфікованих комунікацій з підтримкою відеозв'язку;
- 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.

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРМ-122.24.0004.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 3    |

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Python.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-122.24.0004.00.00.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 маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-122.24.0004.00.00.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                 | – 83 аркуші. |

## 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 Маркетингове та економічне обґрунтування ІТ-проєкту.

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 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2024 р.

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРМ-122.24.0004.00.00.ТЗ</b> | 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 А.С.

*Дослідження та програмна реалізація  
системи аналізу та консолідації розподіленої інформації уніфікованих  
комунікацій з підтримкою відеозв'язку*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 18

Літера: РП

Кропивницький – 2024 року

## Основна програма

```
import threading
import socket
import cv2
import pickle
import struct
import json
import os
import datetime
from queue import Queue
from cryptography.fernet import Fernet

# Генерація ключа для шифрування даних
encryption_key = Fernet.generate_key()
cipher = Fernet(encryption_key)

class DataCollector:
    def __init__(self, data_source):
        self.data_source = data_source
        self.data_queue = Queue()

    def collect_data(self):
        while True:
            data = self.data_source.get_data()
            encrypted_data = cipher.encrypt(json.dumps(data).encode())
            self.data_queue.put(encrypted_data)

class DataProcessor:
    def __init__(self):
        self.consolidated_data = {}

    def process_data(self, data):
        decrypted_data = json.loads(cipher.decrypt(data).decode())
        if decrypted_data['id'] not in self.consolidated_data:
            self.consolidated_data[decrypted_data['id']] = []

        self.consolidated_data[decrypted_data['id']].append(decrypted_data['value'])

    def consolidate_data(self):
        return {k: sum(v) / len(v) for k, v in self.consolidated_data.items()}

class VideoCommunicationServer:
    def __init__(self, host, port):
        self.host = host
        self.port = port
```

```

def start_server(self):
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((self.host, self.port))
    server_socket.listen(5)
    while True:
        client_socket, addr = server_socket.accept()
        threading.Thread(target=self.handle_client,
args=(client_socket,)).start()

def handle_client(self, client_socket):
    payload_size = struct.calcsize("Q")
    data = b""
    while True:
        while len(data) < payload_size:
            packet = client_socket.recv(4 * 1024)
            if not packet:
                break
            data += packet
        packed_msg_size = data[:payload_size]
        data = data[payload_size:]
        msg_size = struct.unpack("Q", packed_msg_size)[0]
        while len(data) < msg_size:
            data += client_socket.recv(4 * 1024)
        frame_data = data[:msg_size]
        data = data[msg_size:]
        frame = pickle.loads(frame_data)
        cv2.imshow('Server Receiving Video', frame)
        if cv2.waitKey(1) == 13:
            break
    client_socket.close()

class VideoCommunicationClient:
    def __init__(self, server_ip, port):
        self.server_ip = server_ip
        self.port = port

    def start_client(self):
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client_socket.connect((self.server_ip, self.port))
        cap = cv2.VideoCapture(0)
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break
            data = pickle.dumps(frame)

```

```

        message_size = struct.pack("Q", len(data))
        client_socket.sendall(message_size + data)
        cv2.imshow('Client Sending Video', frame)
        if cv2.waitKey(1) == 13:
            break
    client_socket.close()
    cap.release()
    cv2.destroyAllWindows()

class DataStorage:
    def __init__(self, file_path):
        self.file_path = file_path

    def save_data(self, data):
        encrypted_data = cipher.encrypt(json.dumps(data).encode())
        with open(self.file_path, 'wb') as f:
            f.write(encrypted_data)

    def load_data(self):
        if os.path.exists(self.file_path):
            with open(self.file_path, 'rb') as f:
                encrypted_data = f.read()
            return json.loads(cipher.decrypt(encrypted_data).decode())
        return {}

class Logger:
    def __init__(self, log_file):
        self.log_file = log_file

    def log(self, message):
        timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        with open(self.log_file, 'a') as f:
            f.write(f"{timestamp} - {message}\n")

class SystemOrchestrator:
    def __init__(self):
        self.collector = DataCollector(data_source=None) # Placeholder
        self.processor = DataProcessor()
        self.storage = DataStorage(file_path="data.json")
        self.logger = Logger(log_file="system.log")
        self.video_server = VideoCommunicationServer(host="0.0.0.0", port=9999)
        self.video_client = VideoCommunicationClient(server_ip="127.0.0.1",
port=9999)

    def start_system(self):
        threading.Thread(target=self.video_server.start_server).start()

```

```
threading.Thread(target=self.run_data_collection).start()

def run_data_collection(self):
    while True:
        if not self.collector.data_queue.empty():
            data = self.collector.data_queue.get()
            self.processor.process_data(data)
            self.logger.log("Data processed")
            consolidated_data = self.processor consolidate_data()
            self.storage.save_data(consolidated_data)

if __name__ == "__main__":
    orchestrator = SystemOrchestrator()
    orchestrator.start_system()
```

K6П3\_2024

## Файл ai\_video\_analysis.py

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model

class VideoAnalyzer:
    def __init__(self, model_path):
        self.model = load_model(model_path)
        self.classes = ["Person", "Vehicle", "Other"]

    def analyze_frame(self, frame):
        resized_frame = cv2.resize(frame, (224, 224))
        normalized_frame = resized_frame / 255.0
        batch = np.expand_dims(normalized_frame, axis=0)
        predictions = self.model.predict(batch)
        predicted_class = self.classes[np.argmax(predictions)]
        return predicted_class

if __name__ == "__main__":
    analyzer = VideoAnalyzer("model.h5")
    cap = cv2.VideoCapture(0)
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        label = analyzer.analyze_frame(frame)
        cv2.putText(frame, label, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 255, 0), 2)
        cv2.imshow("AI Video Analysis", frame)
        if cv2.waitKey(1) == 13:
            break
    cap.release()
    cv2.destroyAllWindows()
```

## Файл multi\_factor\_auth.py

```
import pyotp
import qrcode

class MultiFactorAuth:
    def __init__(self, user_email):
        self.secret = pyotp.random_base32()
        self.user_email = user_email

    def generate_qr_code(self):
        otp = pyotp.TOTP(self.secret)
        qr = qrcode.make(otp.provisioning_uri(self.user_email,
issuer_name="DistributedSystem"))
        qr.save(f"{self.user_email}_qrcode.png")

    def validate_otp(self, otp_code):
        otp = pyotp.TOTP(self.secret)
        return otp.verify(otp_code)

if __name__ == "__main__":
    mfa = MultiFactorAuth("user@example.com")
    mfa.generate_qr_code()
    user_code = input("Enter the OTP code: ")
    if mfa.validate_otp(user_code):
        print("Authentication successful")
    else:
        print("Authentication failed")
```

```
import socket
import threading

class ChatServer:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.clients = []

    def start_server(self):
        server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server_socket.bind((self.host, self.port))
        server_socket.listen(5)
        print("Chat server started")
        while True:
            client_socket, addr = server_socket.accept()
            self.clients.append(client_socket)
            threading.Thread(target=self.handle_client,
args=(client_socket,)).start()

    def handle_client(self, client_socket):
        while True:
            try:
                message = client_socket.recv(1024).decode()
                self.broadcast(message, client_socket)
            except:
                self.clients.remove(client_socket)
                client_socket.close()
                break

    def broadcast(self, message, sender_socket):
        for client in self.clients:
            if client != sender_socket:
                client.send(message.encode())

class ChatClient:
    def __init__(self, server_ip, port):
        self.server_ip = server_ip
        self.port = port

    def start_client(self):
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client_socket.connect((self.server_ip, self.port))
        threading.Thread(target=self.receive_messages,
args=(client_socket,)).start()
        while True:
            message = input("")
            client_socket.send(message.encode())

    def receive_messages(self, client_socket):
        while True:
```

```
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            client_socket.close()
            break

if __name__ == "__main__":
    role = input("Start as server (s) or client (c)? ")
    if role == "s":
        server = ChatServer("0.0.0.0", 12345)
        server.start_server()
    elif role == "c":
        client = ChatClient("127.0.0.1", 12345)
        client.start_client()
```

K6П3\_2024

```
import cv2
import threading

class MultiStreamViewer:
    def __init__(self, streams):
        self.streams = streams

    def view_stream(self, stream_url):
        cap = cv2.VideoCapture(stream_url)
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break
            cv2.imshow(f"Stream: {stream_url}", frame)
            if cv2.waitKey(1) == 13:
                break
        cap.release()
        cv2.destroyAllWindows()

    def start_streams(self):
        for stream in self.streams:
            threading.Thread(target=self.view_stream, args=(stream,)).start()

if __name__ == "__main__":
    viewer = MultiStreamViewer(["stream1.mp4", "stream2.mp4"])
    viewer.start_streams()
```

```
import sqlite3

class DatabaseStorage:
    def __init__(self, db_name="system_data.db"):
        self.connection = sqlite3.connect(db_name)
        self.create_table()

    def create_table(self):
        cursor = self.connection.cursor()
        cursor.execute("""
            CREATE TABLE IF NOT EXISTS data (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                timestamp TEXT,
                data TEXT
            )
        """)
        self.connection.commit()

    def save_data(self, data):
        cursor = self.connection.cursor()
        cursor.execute("INSERT INTO data (timestamp, data) VALUES
(datetime('now'), ?)", (data,))
        self.connection.commit()

    def load_all_data(self):
        cursor = self.connection.cursor()
        cursor.execute("SELECT * FROM data")
        return cursor.fetchall()

if __name__ == "__main__":
    db = DatabaseStorage()
    db.save_data("Sample data")
    print(db.load_all_data())
```

**Файл mobile\_access.py**

```
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/data', methods=['POST'])
def receive_data():
    data = request.json
    print("Received data from mobile device:", data)
    return jsonify({"status": "success", "message": "Data received"}), 200

@app.route('/status', methods=['GET'])
def get_status():
    return jsonify({"status": "system operational"}), 200

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

**Файл web\_interface.py**

```
from flask import Flask, render_template, request, jsonify
import sqlite3

app = Flask(__name__)

def get_all_data():
    connection = sqlite3.connect("system_data.db")
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM data")
    data = cursor.fetchall()
    connection.close()
    return data

@app.route('/')
def home():
    data = get_all_data()
    return render_template('index.html', data=data)

@app.route('/add', methods=['POST'])
def add_data():
    new_data = request.form['data']
    connection = sqlite3.connect("system_data.db")
    cursor = connection.cursor()
    cursor.execute("INSERT INTO data (timestamp, data) VALUES (datetime('now'),
?)", (new_data,))
    connection.commit()
    connection.close()
    return jsonify({"status": "success", "message": "Data added"}), 200

if __name__ == "__main__":
    app.run(debug=True)
```

## Файл report\_generator.py

```

from fpdf import FPDF

class ReportGenerator:
    def __init__(self, report_file="report.pdf"):
        self.pdf = FPDF()
        self.pdf.set_auto_page_break(auto=True, margin=15)
        self.report_file = report_file

    def add_section(self, title, content):
        self.pdf.add_page()
        self.pdf.set_font("Arial", size=12)
        self.pdf.cell(200, 10, txt=title, ln=True, align='C')
        self.pdf.ln(10)
        self.pdf.multi_cell(0, 10, content)

    def save_report(self):
        self.pdf.output(self.report_file)

if __name__ == "__main__":
    generator = ReportGenerator()
    generator.add_section("System Data Report", "This is a generated report with
detailed system data.")
    generator.save_report()
    print("Report generated and saved as 'report.pdf'")

```

## Файл encrypted\_video\_stream.py

```

import cv2
import socket
import pickle
import struct
from cryptography.fernet import Fernet

key = Fernet.generate_key()
cipher = Fernet(key)

class EncryptedVideoServer:
    def __init__(self, host, port):
        self.host = host
        self.port = port

    def start_server(self):
        server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server_socket.bind((self.host, self.port))
        server_socket.listen(5)
        print("Encrypted Video Server started")
        while True:
            client_socket, addr = server_socket.accept()
            self.handle_client(client_socket)

    def handle_client(self, client_socket):

```

```

payload_size = struct.calcsize("Q")
data = b""
while True:
    while len(data) < payload_size:
        packet = client_socket.recv(4 * 1024)
        if not packet:
            break
        data += packet
    packed_msg_size = data[:payload_size]
    data = data[payload_size:]
    msg_size = struct.unpack("Q", packed_msg_size)[0]
    while len(data) < msg_size:
        data += client_socket.recv(4 * 1024)
    encrypted_frame_data = data[:msg_size]
    data = data[msg_size:]
    frame_data = cipher.decrypt(encrypted_frame_data)
    frame = pickle.loads(frame_data)
    cv2.imshow('Encrypted Video Server', frame)
    if cv2.waitKey(1) == 13:
        break
client_socket.close()

class EncryptedVideoClient:
    def __init__(self, server_ip, port):
        self.server_ip = server_ip
        self.port = port

    def start_client(self):
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client_socket.connect((self.server_ip, self.port))
        cap = cv2.VideoCapture(0)
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break

            frame_data = pickle.dumps(frame)
            encrypted_frame_data = cipher.encrypt(frame_data)
            message_size = struct.pack("Q", len(encrypted_frame_data))
            client_socket.sendall(message_size + encrypted_frame_data)
            cv2.imshow('Encrypted Video Client', frame)
            if cv2.waitKey(1) == 13:
                break

        client_socket.close()
        cap.release()
        cv2.destroyAllWindows()

if __name__ == "__main__":
    role = input("Start as server (s) or client (c)? ")
    if role == "s":
        server = EncryptedVideoServer("0.0.0.0", 9999)
        server.start_server()
    elif role == "c":
        client = EncryptedVideoClient("127.0.0.1", 9999)
        client.start_client()

```

## Файл cloud\_backup.py

```

import boto3

class CloudBackup:
    def __init__(self, bucket_name, aws_access_key, aws_secret_key):
        self.bucket_name = bucket_name
        self.s3 = boto3.client('s3', aws_access_key_id=aws_access_key,
aws_secret_access_key=aws_secret_key)

    def upload_file(self, file_path, key_name):
        self.s3.upload_file(file_path, self.bucket_name, key_name)
        print(f"File {file_path} uploaded to {self.bucket_name}/{key_name}")

    def download_file(self, key_name, file_path):
        self.s3.download_file(self.bucket_name, key_name, file_path)
        print(f"File {key_name} downloaded to {file_path}")

if __name__ == "__main__":
    cloud_backup = CloudBackup("my-bucket", "AWS_ACCESS_KEY", "AWS_SECRET_KEY")
    cloud_backup.upload_file("data.json", "backup/data.json")

```

## Файл critical\_event\_notifications.py

```

import smtplib
from email.mime.text import MIMEText

class CriticalEventNotifier:
    def __init__(self, smtp_server, port, login, password):
        self.smtp_server = smtp_server
        self.port = port
        self.login = login
        self.password = password

    def send_notification(self, recipient_email, subject, message):
        msg = MIMEText(message)
        msg["Subject"] = subject
        msg["From"] = self.login
        msg["To"] = recipient_email
        with smtplib.SMTP(self.smtp_server, self.port) as server:
            server.starttls()
            server.login(self.login, self.password)
            server.sendmail(self.login, recipient_email, msg.as_string())
        print(f"Notification sent to {recipient_email}")

if __name__ == "__main__":
    notifier = CriticalEventNotifier("smtp.gmail.com", 587,
"your_email@gmail.com", "your_password")
    notifier.send_notification("recipient@example.com", "Critical Event", "A
critical event has occurred in the system.")

```

## Файл big\_data\_integration.py

```

from pyspark.sql import SparkSession

class BigDataAnalyzer:
    def __init__(self):
        self.spark = SparkSession.builder \
            .appName("BigDataIntegration") \
            .getOrCreate()

    def load_data(self, file_path):
        return self.spark.read.json(file_path)

    def analyze_data(self, data):
        data.createOrReplaceTempView("system_data")
        result = self.spark.sql("SELECT id, AVG(value) as avg_value FROM
system_data GROUP BY id")
        result.show()

if __name__ == "__main__":
    analyzer = BigDataAnalyzer()
    data = analyzer.load_data("data.json")
    analyzer.analyze_data(data)

```

## Файл auto\_scaling.py

```

import psutil
import threading
import time

class AutoScaler:
    def __init__(self, threshold=80):
        self.threshold = threshold

    def monitor_resources(self):
        while True:
            cpu_usage = psutil.cpu_percent(interval=1)
            memory_usage = psutil.virtual_memory().percent
            if cpu_usage > self.threshold or memory_usage > self.threshold:
                self.scale_up()
                time.sleep(5)

    def scale_up(self):
        print("Scaling up resources due to high usage.")

if __name__ == "__main__":
    scaler = AutoScaler(threshold=75)
    monitoring_thread = threading.Thread(target=scaler.monitor_resources)
    monitoring_thread.start()

```

**Файл user\_roles.py**

```
class UserRoleManager:
    def __init__(self):
        self.roles = {"admin": ["view", "edit", "delete"], "user": ["view"]}

    def assign_role(self, username, role):
        print(f"Assigned role '{role}' to user '{username}'.")

    def check_permission(self, username, action):
        role = self.get_user_role(username)
        if role and action in self.roles.get(role, []):
            return True
        return False

    def get_user_role(self, username):
        # Placeholder for retrieving user roles from a database
        return "admin" if username == "admin_user" else "user"

if __name__ == "__main__":
    manager = UserRoleManager()
    print(manager.check_permission("admin_user", "edit"))
```

**Файл network\_optimization.py**

```
class NetworkOptimizer:
    def __init__(self):
        self.max_bandwidth = 1000 # Placeholder bandwidth limit in kbps

    def optimize_stream(self, current_bandwidth):
        if current_bandwidth > self.max_bandwidth:
            print("Reducing video quality to optimize bandwidth.")
        else:
            print("Current stream quality is optimal.")

if __name__ == "__main__":
    optimizer = NetworkOptimizer()
    optimizer.optimize_stream(current_bandwidth=1200)
```

**Файл api\_integration.py**

```
import requests

class ExternalAPIClient:
    def __init__(self, base_url):
        self.base_url = base_url

    def fetch_data(self, endpoint):
        response = requests.get(f"{self.base_url}/{endpoint}")
```

```
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error: {response.status_code}")
        return None

if __name__ == "__main__":
    client = ExternalAPIClient("https://jsonplaceholder.typicode.com")
    data = client.fetch_data("posts")
    print(data)
```

#### Файл `offline_mode.py`

```
class OfflineModeManager:
    def __init__(self, storage_file="offline_data.json"):
        self.storage_file = storage_file

    def save_data_offline(self, data):
        with open(self.storage_file, "w") as f:
            json.dump(data, f)
        print("Data saved for offline use.")

    def load_offline_data(self):
        if os.path.exists(self.storage_file):
            with open(self.storage_file, "r") as f:
                return json.load(f)
        return {}

if __name__ == "__main__":
    manager = OfflineModeManager()
    manager.save_data_offline({"example": "data"})
    print(manager.load_offline_data())
```