

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 відео-
конференц-зв’язку на основі серверів MCU”**

Виконав здобувач вищої освіти
II курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Подплетній Є.О.
« ____ » _____ 2024 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.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 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

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

Подплетньому Євгенію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU*

2. Керівник роботи *Усік Павло Сергійович, доктор філософії (PhD)*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU*

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

- | | |
|--|--|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. 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

Подплетній Є.О. Дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи відео-конференц-зв'язку на основі серверів MCU.

Метою розробки є дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU.

Об'єктом дослідження є процес відео-конференц-зв'язку на основі серверів MCU.

Предметом дослідження є методи відео-конференц-зв'язку на основі серверів MCU.

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

Результат роботи – програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU.

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'ютерна інженерія, відео-конференц-зв'язок, MCU

ABSTRACT

Podpletnyi E.O. Research and software implementation of a video conferencing system based on MCU servers. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for a video-conferencing system based on MCU servers.

The goal of the development is the research and software implementation of a video-conferencing system based on MCU servers.

The object of research is the process of video conferencing based on MCU servers.

The subject of the study is the methods of video conferencing based on MCU servers.

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

The result of the work is a software implementation of a video-conferencing system based on MCU servers.

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

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

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

The program was developed in the Python environment.

Keywords: computer engineering, video conferencing, MCU

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	26
3.3 Розробка функціональної схеми	31
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.....	44
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	47
6 НАУКОВА НОВИЗНА	52

						ВКРМ-123.24.0031.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.		Подплетній Є.О.			Дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU	Літ.	Аркуш	Аркушів
Перев.		Усік П.С.				М	1	78
Н.контр.		Коваленко А.С.			ЦНТУ КІ-23М			
Затв.		Смірнов О.А.						

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	53
7.1	Визначення цільової аудиторії кінцевого готового продукту	53
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	54
7.3	Вибір методу оцінки вартості ПЗ	56
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	56
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	58
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	58
7.7	Визначення ключових факторів успіху конкретного проєкту.....	60
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	61
8.1	Вступ.....	61
8.2	Аналіз умов праці	62
8.3	Техніка безпеки та протипожежна профілактика	66
8.4	Розробка заходів з охорони праці	68
8.5	Висновки до розділу.....	69
9	ОСНОВНІ ВИСНОВКИ.....	70
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72

КБПЗ-2024

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВКЗ	–	відеоконференцзв'язок
ПЗ	–	програмне забезпечення
CMP	–	Codian Management Platform
DNS	–	Domain Name System
GMS	–	Global Management System
GRE	–	Generic Routing Encapsulation
ISDN	–	Integrated Services Digital Network
MCU	–	Multipoint Control Unit
MPLS	–	Multiprotocol Label Switching
MXM	–	Media eXchange Manager
SSL	–	Secure Socket Layer
TLS	–	Transport Layer Security
TMS	–	Tandberg Management Suite
VPN	–	Virtual Private Network
USENET		User Network

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Зміни на ринку відео-конференц-зв'язку (ВКЗ) багато в чому обумовлені тим, що ВКЗ поступово переходить у зону відповідальності ІТ-підрозділів компаній, тоді як раніше цими системами в основному займалися фахівці з аудіовізуального (AV) рішення або зв'язківці. З переходом ВКЗ під крило ІТ міняються й самі технічні рішення. Спеціалізовані програмно-апаратні комплекси, що домінували раніше, користуються на ринку всі меншим попитом, тоді як популярність набирають програмні рішення – як клієнти, так і сервери, включаючи MCU. Сьогодні вже складно знайти виробника серверів MCU, що не мав би програмних варіантів таких продуктів. Ціла низка придбань (Brother купила компанію Nefsis, LifeSize – ClearSea, Radvision – Scopia (згодом сама Radvision увійшла до складу Avaya), ClearOne – Spontania, 3CX – e-works) також показує, що виробники останнім часом спішно заповнювали відсутню експертизу в області програмних рішень. Міняється й модель ціноутворення на ринку ВКЗ-рішень: якщо раніше вартість таких рішень розраховували виходячи з кількості портів, то сьогодні все частіше розрахунок виробляється по числу користувачів, що обслуговуються, що відповідає програмній суті рішень нового покоління.

Впровадження систем ВКЗ почалося ще на початку нульових років. Однак через високу вартість традиційних рішень ВКЗ розвиток системи йшов надзвичайно повільно. Було ухвалене рішення про впровадження більше дешевого програмного сервера ВКЗ. ПЗ було розгорнуто на фізичному сервері, що перебуває безпосередньо в установі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU.

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

- Огляд існуючих систем відео-конференц-зв'язку на основі серверів MCU.
- Дослідження системи відео-конференц-зв'язку на основі серверів MCU.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU.

Об'єктом дослідження є процес відео-конференц-зв'язку на основі серверів MCU.

Предметом дослідження є методи відео-конференц-зв'язку на основі серверів MCU.

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

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

- Удосконалено метод відео-конференц-зв'язку на основі серверів MCU.
- Розроблено вітчизняний продукт відео-конференц-зв'язку на основі серверів MCU, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі відео-конференц-зв'язку на основі серверів MCU.

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.

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 вищеперахованого, дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Перехід на менш дороге (бюджетне) рішення дозволяє прискорити розвиток системи відео-конференц-зв'язку й підключити до неї основні підрозділи. Розгорнута система ВКЗ успішно застосовується для проведення відеоселекторних нарад, дистанційної атестації й телелекцій, з її допомогою проводяться конференції. Головний результат впровадження ВКЗ – підвищення ефективності системи керування й прийняття рішень у центрі. Крім того, застосування відеозв'язку дозволило раціоналізувати використання робочого часу співробітників, дало додаткові інструменти для підвищення їхньої кваліфікації, сприяло зниженню витрат на відрядні витрати й на міжміські переговори.

Разом з тим слід зазначити, що розгорнута система ВКЗ використовується в основному для рішення офісних завдань. При всіх перевагах програмних систем, у рішень на основі спеціалізованих апаратних комплексів залишається більша область застосування – зокрема, у вертикальних додатках.

1.2 Область застосування

У цей час системи ВКЗ активно доповнюються різними елементами того, що прийнято називати уніфікованими комунікаціями (УфК). Як показують дослідження, проведені OSP Data, ВКЗ у чистому виді стає усе менш затребуваною. Користувачі хочуть, щоб цей вид комунікацій супроводжувався такими зручними функціями, як чат, статус присутності абонента й т.д. Обов'язкової, з погляду більшості користувачів, стає й підтримка системою ВКЗ популярних мобільних платформ. Слід зазначити, що все вищезгадане вже є в рішеннях більшості виробників, головне питання – як реалізація й зручності

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

використання цих функцій. Ще однією важливою тенденцією й у той же час певною погрозою для традиційних гравців ринку ВКЗ стає ріст популярності системи Lync (Skype for Business) компанії Microsoft. Це рішення за замовчуванням є присутнім у програмних системах Microsoft, установлених у переважній більшості великих організацій. Відповідно, як тільки в Lync з'явилася підтримка відеодзвінків, на ринку ВКЗ з'явився новий потужний гравець із величезною інстальованою базою. Не дивно, що всі традиційні гравці ВКЗ хочуть дружити з Microsoft, забезпечуючи стикування своїх рішень із системою Lync. Але, як вважають експерти, проблема полягає в тому, що в рішенні Microsoft використовується дуже багато пропрієтарних (нестандартних) технологій.

У більшості розвинених країн світу ВКЗ, як, втім, і інші ІКТ-сервіси, йде в хмари. Однак в Україні хмари приживаються дуже повільно. Головна проблема – побоювання замовників щодо безпеки сервісу, одержуваного із хмари. Відповідно до дослідження, проведеному в 2016 році OSP Data, цього остерігаються більше 60% українських замовників. Схоже, стіна недовіри (до хмарної моделі) досить міцна, пробити її найближчим часом буде дуже важко.

Але навіть без хмар проблема безпеки відеодзвінків стає усе більше актуальною. ВКЗ активно виходить за межі переговорних кімнат, всі частіше використовується на персональних і мобільних пристроях користувачів, які ніяк не можна віднести до надійного з погляду безпеки.

Відповімо на два важливих питання: «Потрібно захищати відеотрафік?» і «Як його захищати?».

Відповідь на перше питання багато в чому залежить від того, для яких завдань використовується відеозв'язок на конкретному підприємстві й наскільки критично важлива передана через систему ВКЗ інформація. При цьому не треба забувати, що впроваджені на програмному рівні закладки дозволяють знімати відео безпосередньо з камер і серверів, тому одного захисту трафіку недостатньо для забезпечення безпеки. А наскільки б надійною і довіреною не була платформа, на якій побудована система ВКЗ, її злом – лише справа часу й

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

грошей. Коротко говорячи, загальної відповіді на питання «Потрібно захищати відеотрафік?» не існує – все повинне вирішуватися в кожному конкретному випадку. Якщо ж ви вважаєте, що такий захист вам потрібен, варто прислухатися до рекомендацій фахівця. Він нагадує, що забезпечення інформаційної безпеки в цілому й безпеці систем ВКЗ зокрема вимагає комплексного підходу, що повинен передбачати адміністративні, організаційні, технічні й фізичні міри захисту. Що стосується технічних методів, те основні – це забезпечення контролю доступу до системи й властиво закриття переданої інформації. Як правило, останнє здійснюється шляхом інкапсуляції відеотрафіку в захищені канали VPN.

Однак системи ВКЗ мають ряд специфічних особливостей, які необхідно враховувати при проектуванні й виборі засобів захисту: необхідність пріоритизації трафіку, забезпечення малого часу затримки і її варіацій і т.д. При виборі системи захисту для ВКЗ, крім загальних вимог, таких як наявність сертифікатів регуляторів, виникає ряд технологічних умов. Зокрема, система захисту повинна підтримувати різні схеми роботи, використовувані в системах ВКЗ (клієнт – сервер, сервер – сервер, клієнт – сервер), проходження міжмережних екранів і засобів NAT, а також об'єднання мереж з однаковою адресацією. Важлива умова – мінімізація затримок і їхньої варіації, а також мінімальний обсяг службової інформації. Додавання такої інформації, особливо при невеликому розмірі пакетів, може істотно збільшувати вимоги до каналів зв'язку, які у випадку ВКЗ і так досить високі. Крім властиво VPN-шлюзів, бажано використовувати й інші системи захисту: міжмережні екрани, системи визначення й запобігання вторгнень, комплексну систему керування й моніторингу.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

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

BigBlueButton

Що представляє із себе система організації ВКЗ BigBlueButton? Чим вона відрізняється від інших подібних систем? Чи підійде вона для тих завдань, які стоять перед Вами? Пропоную короткий огляд можливостей системи.

Огляд можливостей

Показ презентації. Завантаження й демонстрація документів.

Підтримувані формати:

- документи у форматі PDF;
- різні типи документів Microsoft Office;
- різні типи документів OpenOffice;
- перемикання ведучого: право на завантаження презентації;

Керування презентацією. Проведення презентації й створення своїх позначок:

- показ слайдів: послідовний або довільний;
- розмір слайда: по розмірі вікна або по ширині вікна;
- масштабування: збільшення зображення;
- позначки на слайді: малювання маркером, малювання окружностей, прямокутників, вибір кольору й товщини ліній;
- перемикання ведучого: керування презентацією;

Аудіоконференція. Можливість організації повноцінної аудіоконференції з керуванням:

- кожний учасник може управляти своїм мікрофоном;

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

– модератор може включити або виключити мікрофон учасника в одне натискання;

– модератор може включити або виключити мікрофон групи учасників в одне натискання.

Відеоконференція. Можливість організації повноцінної відеоконференції:

– трансляція відео з камери (tv-тюнера, плати відеозахвата й т.д.) ведучого;

– трансляція відео з камери учасників для живого спілкування;

– керування відеоконференцією;

Чат. Обговорення питань у форматі чату:

– організація загального чату для питань/відповідей або обговорення;

– організація персонального чату для особистого спілкування;

– можливість вибору кольору для своїх повідомлень.

Запис і перегляд відеоконференцій:

– запис і показ демонструємих слайдів;

– запис і відтворення позначок ведучого на слайдах;

– запис і відтворення аудіоконференції;

– запис і відтворення спілкування в чаті;

– керування записами.

Трансляція робочого стола:

– трансляція усього робочого стола ведучого;

– трансляція області робочого стола ведучого;

Організація опитування:

– створення опитувань, попередній перегляд;

– швидке проведення опитування;

– проведення веб-опитування;

– підведення підсумків опитування й вивід результатів.

Інтерфейс. Простий і зручний для учасників інтерфейс:

– доступ через браузер (непотрібно ніяких програм, налаштувань портів);

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

- підтримка більше 40 мов інтерфейсу;
- адаптація (ребрендинг) інтерфейсу.

Інтеграція. Взаємодія Ваших проектів із сервером відео-конференц-зв'язку:

- наявність API для JSP;
- наявність API для PHP;
- наявність API для Ruby;

Інтеграція. Готові модулі інтеграції з розповсюдженими системами:

- Moodle;
- WordPress;
- Drupal;
- Joomla;
- Canvas LMS;
- Sakai;
- Tiki Wiki CMS Groupware;
- eFront;
- Chamilo LMS.

TrueConf

Функціональні можливості програми для відео-конференц-зв'язку дозволяють організовувати різні режими відеозв'язку, під час яких учасники можуть не тільки чути й бачити один одного, але й обмінюватися різними даними, демонструвати один одному презентації й спільно працювати над документами за допомогою інструментів електронної дошки.

Існує кілька режимів відеозв'язку: персональні відеоконференції (відеодзвінки), багатоточечні відеоконференції й селекторні наради.

Персональна відеоконференція – можливості

Сеанс персональної відео-конференц-зв'язку має на увазі онлайн-спілкування двох користувачів, які можуть бачити й чути один одного, а також користуватися інструментами спільної роботи. Співрозмовники можуть

						ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			11

перебувати в різних точках земної кулі, якість відеозв'язку при цьому не постраждає. Звичайно така послуга, як відеодзвінок, надається провайдерами зовсім безкоштовно.

Багатоточечні (групові) відеоконференції

Як правило, під час групових відеоконференцій кількість взаємодіючих між собою учасників варіюється в розмірі 10-15 користувачів. Така відеоконференція вважається симетричною, тому як абсолютно всі учасники мають можливість висловлюватися й вільно обговорювати один з одним які-небудь питання.

Можливості селекторної наради

На відміну від групової відеоконференції, селекторна нарада дозволяє зібрати у віртуальній аудиторії до сотні учасників, однак активними можуть бути тільки троє з них. Даний тип ставиться до розряду асиметричних відеоконференцій, тому як можливості слухачів обмежені – вони не можуть бачити один одного й переговорюватися між собою під час сеансу. Однак за допомогою таких інструментів спільної роботи, як текстовий чат, передача файлів і демонстрація слайдшоу, користувачі можуть обмінюватися інформацією.

Додаткові можливості:

- Текстовий чат. Ця функція особливо зручна під час проведення асиметричних відеоконференцій, коли не всі учасники можуть висловлюватися в процесі сеансу.
- Передача файлів. Співрозмовники мають можливість обмінюватися аудіо й відеоданими, а також різними документами й графічними малюнками.
- Запис відеоконференції. Можливість запису сеансу відеозв'язку.
- Слайдшоу. Демонстрація презентацій практично завжди є невід'ємною частиною виступу учасників відеоконференції. За допомогою слайдшоу що віщає може підкріплювати свої слова фотографіями, діаграмами й таблицями.
- Електронна дошка. Користувачі можуть спільно працювати над документами в окремому вікні свого комп'ютера.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Показ віддаленого робочого стола. Використовуючи цю можливість, відео-співрозмовники можуть допомагати один одному під час роботи над документами.

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 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.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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.
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

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

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.

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.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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.

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).

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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.

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.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

– 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.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи відео-конференц-зв'язку на основі серверів MCU.

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 методикку 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.

КБПЗ_2024

					VKPM-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Напевно в кожному ІТ-відділі, рано або пізно, виникає завдання: «забезпечити можливість проведення відеоконференцій (вебінарів)». Пошук по мережі запропонує Вам велика кількість рішень: апаратні, програмні, хмарні сервіси. Що вибрати й що не забути врахувати при такому виборі? На що звернути увагу при виборі системи? Адже на неї прийде витратити чимало часу (а можливо й засобів) і буде кривдно з'ясувати через рік-два, що обрано тупикову галузь розвитку.

Критерії вибору

Що вибрати – саме складне питання, і універсальної відповіді на нього не існує. Відповідь для кожного свій, тому що залежить від великої кількості параметрів:

- **Гроші:** первинний і головний параметр, коли їх менше 2-х (3-х, 5-ти .. 30-ти .. 50-ти) мільйонів грн. це накладає відчутні обмеження на волю вибору.
- **Устаткування:** наявність раніше придбаного встаткування для організації ВКЗ, наявність вільних обчислювальних потужностей у ЦОДі.
- **Канали зв'язку:** ширина Вашого каналу зв'язку, ширина і якість каналів у клієнтів.
- **Якість:** а раптом начальство бажає картинку високого дозволу (див. Пункт 3).
- **Кількість клієнтів:** скільки учасників Ви хочете бачити одночасно, бути присутнім (слухати й дивитися) звичайно можуть багато хто.
- **Завдання:** Вам потрібно переважно обговорення (відеозв'язок), або бізнес-презентація (демонстрація чого-небудь), а може навчання?
- **Захист:** закритий внутрішній зв'язок або доступний усім сервіс.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

– Кадри: одна справа купити або скачати, цим же хтось повинен займатися, якщо кадри слабкі – краще платити за рішення «під ключ» і не скупитися.

– Час: гроші є, але презентація вже на днях. устаткування не встигне приїхати, можна узяти під оренду встаткування або сервіс, або підняти програмний сервер.

– Мобільність: Вам потрібні компактні пристрої в кабінетах або система для конференц-залів, чи будуть у Вас мобільні клієнти (ноутбук з камерою).

– Запис і трансляція: не всі системи ВКЗ уміють робити все.

– Гнучкість: а як щодо доробки або інтеграції з іншими використовуваними рішеннями.

– Клієнти: яке є встаткування в них або їм його треба закуповувати, а якщо програмне рішення те хто його буде набувати на стороні клієнтів, а яка в них операційна система, як щодо сумісності. а може їм простіше забезпечити доступ через браузер.

– Додатково: Ваші корпоративні вимоги....

Всі ці параметри Вам краще визначити як можна точніше й заздалегідь, так сказати «на березі», відкладати їх на потім – занадто дороге задоволення.

Рішення існує

Дуже часто бажані Вами параметри ВКЗ можуть суперечити один одному. Якщо у Вас так, то потрібно шукати золоту середину: якимись «хочу» прийде пожертвувати ну або серйозно переглянути бюджет проекту. У моєму випадку відповіді вийшли приблизно такі:

– Гроші: 0.00 грн (ти спершу запусти, покажи, проведи 10-20 заходів...).

– Устаткування: є запас потужності на ЦОДі й xPoint з керованою камерою (ні про яке додаткове встаткування мова вже не йде, див. Пункт 1).

– Канали зв'язку: є клієнти з 128-256 кбіт/сек.

– Якість: не критично (див. Пункт 3).

– Кількість клієнтів: до 35 учасників, хоча б шість-десять на відеозв'язку.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- Завдання: в основному презентація й навчання.
- Захист: відкритий доступ.
- Кадри: «не хочеш – змусимо, не можеш – навчимо».
- Час: строки підтискають (грошей не дамо, а подивитися прийдемо).
- Мобільність: мобільні клієнти й ПК із веб-камерами.
- Запис і трансляція: запис обговорення й демонстрації (слайдів).
- Гнучкість: інтеграція зі своїм сервером LDAP і веб-порталом.
- Клієнти: тільки через браузер інакше кожному прийде по годині все

пояснювати.

- Додатково: немає вимог.

Відео-конференц-зв'язок (ВКЗ) – це технологія інтерактивної взаємодії двох і більше віддалених користувачів, при якій між ними можливий обмін відео- і аудіоінформацією у режимі реального часу.

Відео-конференц-зв'язок (ВКЗ) – сучасний високотехнологічний інструмент спільної роботи, за допомогою якого можна оперативно організувати онлайн-наради й зустрічі, а також проводити вебінари й онлайн-трансляції на велику кількість учасників. Сучасні системи ВКЗ поступово досягають такого рівня якості передачі відеозображення й звуку, що в процесі спілкування створюється «ефект присутності», а обсяг сприйманої за допомогою ВКЗ інформації досягає порядку 90%. При цьому, на відміну від особистих зустрічей, ВКЗ має ряд переваг, таких як:

- Швидкість і простота організації заходу (наради або онлайн-семінару).
- Можливість миттєво ділитися документами, робочим столом і програмами свого комп'ютера з усіма учасниками, спільно їх обговорювати й редагувати.
- Записувати захід.
- При необхідності підключати інших учасників, що перебувають онлайн, до заходу, де б вони не перебували.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

- Брати участь у заходах дистанційно, у тому числі з мобільних пристроїв.
- Підключатися до заходів з телефону.
- Проводити опитування й голосування просто й швидко з автоматичним аналізом результатів у процесі.
- Використовувати чат, у тому числі приватний між окремими учасниками заходу.

Переваги ВКЗ:

- Підвищення ефективності ведення бізнесу.
- Зниження витрат на відрядження й поїздки.
- Прискорення процесу прийняття рішень.
- Економія часу на зустрічах поза офісом.
- Підвищення залученості співробітників, клієнтів і партнерів у робочий процес.

Кейси використання ВКЗ:

- Організація онлайн-зустрічей і нарад між співробітниками компаній (особливо в територіально-розподілених структурах).
- Проведення навчальних / експертних вебінарів, у тому числі в рамках корпоративного навчання.
- Трансляції виступів керівників на співробітників компаній.
- Проведення онлайн-співбесід при підборі персоналу.
- Проведення маркетингових заходів і прес-конференцій онлайн.
- Презентації ринку / продажу продуктів і послуг компаній.
- Обслуговування клієнтів онлайн / відеоконсультації.
- Телемедицина.

Підхід до вибору рішення ВКЗ повинен формуватися, виходячи з відповіді на питання – яким образом організація буде застосовувати відеотехнології? Потрібно розуміти, що за допомогою рішень ВКЗ можна

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

вдосконалити практично всі робочі процеси, які в цей момент припускають використання телефонії, електронної пошти або відряджень.

Наприклад, якщо основне завдання – віддалене навчання співробітників з робочих місць (комп'ютерів), те тут буде досить бюджетного рішення веб-ВКЗ. Якщо ж є потреба забезпечити якісні комунікації між віддаленими офісами компанії на рівні керівництва, між групами співробітників, коли в першу чергу необхідно забезпечити ефект присутності в декількох місцях, то логічно буде використовувати рішення рівня Telepresence (апаратні ВКЗ).

Різні за класом рішення ВКЗ (софтверні й апаратні) можуть гармонійно співіснувати один з одним в одній компанії, закриваючи різні бізнес-процеси й потреби у відео-спілкуванні й спільній роботі співробітників.

Відповідність розв'язуваних завдань у комунікаціях, спільній роботі й існуючих рішеннях ВКЗ

Після визначення – які завдання буде закривати ВКЗ, необхідно визначити модель розгортання мережі ВКЗ на підприємстві.

Сьогодні існують кілька моделей розгортання мережі ВКЗ:

- Інвестування в побудову власної мережі ВКЗ.
- Модель «Відео-конференц-зв'язок як послуга».
- Гібридна модель. Коли частина функцій винесена в зону відповідальності Оператора послуг ВКЗ.

Перед визначенням моделі розгортання, необхідно розуміти, що побудова мережі ВКЗ є комплексним проектом, що містить у собі наступні складові:

Канали

Організація мережної зв'язності між філіями компанії, каналами з необхідними параметрами швидкості і якості для пропуску критичного до затримок відеотрафіку від прикінцевих пристроїв(відеотерміналів).

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

«Широка смуга» для відеосервера

У рамках проектування каналної складової, необхідно передбачити в місці (філії) розташування відеосервера забезпечення широкої каналної смуги для агрегації відеотрафіку від всіх відеоточок філій.

ЦОД

Виділення приміщення з необхідними параметрами по електриці, заземленню й ін. для розміщення встаткування ядра мережі ВКЗ.

Устаткування ядра мережі ВКЗ

Покупка, проектування, монтаж і пуско-налагодження встаткування ядра мережі відео-конференц-зв'язку, включаючи відеосервера, сервера керування, моніторингу, запису, реєстрації прикінцевих пристроїв. Також, для отказоустойчивости рішення, необхідно передбачити резервування системи.

Співробітники

Найм штату дефіцитних фахівців для обслуговування ядра мережі ВКЗ і створення служби тех.підтримки користувачів Найм штату фахівців для обслуговування відеотерміналів у філіях компанії.

Прикінцеві точки

Покупка й, при необхідності, проектування, монтаж і пусконалагодження відеотерміналів і програмних клієнтів.

Порівняння способів організації Мережі ВКЗ

З огляду на вищесказане, можна зробити наступні висновки на вибір моделі розгортання мережі ВКЗ: Якщо підприємство вирішує інвестувати у власну мережу ВКЗ, то потрібно бути готовим до чималих витрат в устаткування інфраструктури. Потрібно приділити особлива увага питанням нарощування компетенцій обслуговуючого персоналу, керування й аудита системи ВКЗ. Якщо всі ці завдання здійсненні й компанія має необхідний бюджет, то можна бути впевненим, що модель буде успішна, тому що в першу чергу володіння всією інфраструктурою ВКЗ забезпечує підприємству відносно повну незалежність і безпеку від зовнішніх постачальників.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

При виборі моделі «ВКЗ як Послуга» підприємство перекладе витрати по побудові, обслуговуванню й керуванню мережею відеоконференцій на оператора зв'язку, а також зменшить бюджет на непрофільні витрати. У цьому випадку підприємству не прийде інвестувати засобу в модернізацію через 2-3 роки й, якщо формат ВКЗ не буде користуватися попитом – відмовитися від послуги, тим самим заощадивши засобу. Звичайно, всі перераховані переваги моделі виконуються залежно від пропозиції й гарантій провайдеру послуг ВКЗ.

Гібридна модель повинна бути цікава Замовникам у яких уже розгорнута мережа ВКЗ і вона вимагає посилення. У цьому випадку підприємство зберігає інвестиції у вже придбане встаткування й може «малою кров'ю» наростити необхідну потужність, не витрачаючи величезні засоби на дозакупівлю інфраструктури ВКЗ, а платячи тільки абонентську плату за оренду ресурсів в оператора ВКЗ. Успішність моделі також заздрості від пропозиції провайдеру ВКЗ

3.2 Розробка структурної схеми

Архітектури систем відео-конференц-зв'язку

Будь-яка система відео-конференц-зв'язку складається з:

- Терміналів ВКЗ – це можуть бути абсолютно будь-які пристрої, що підтримують передачу відео й аудіо: апаратні, мобільні, програмні, складні системи телеприсутності або навіть устаткування, убудоване в холодильник.
- Сервера ВКЗ, що необхідний для проведення групових відеоконференцій.
- Інфраструктури – канали зв'язку, пристрою для передачі даних, допоміжне устаткування, наприклад для трансляції й запису відеоконференції.
- Периферійного встаткування – спікерфони, мікшери, мікрофони, PTZ-камери.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Які типи відеоконференцій існують?

– Відеодзвінок – даний тип відео-конференц-зв'язку передбачає з'єднання між двома (не більше) абонентами, при цьому наявність ВКЗ-сервера не обов'язково.

– Групова відеоконференція – сеанс багатоточечного відео-конференц-зв'язку при участі трьох і більше абонентів. Для того, щоб створити групову відеоконференцію, необхідна наявність сервера, що буде комутирувати дзвінки, а також передачу відео й аудіо даних. Такі групові відеоконференції можна розділити на симетричне й асиметричні (відеовіщання, селекторна нарада).

Архітектура сервера ВКЗ

Всі рішення діляться на дві групи:

– Апаратні рішення – звичайно перекоднують відеопотоки для кожного з учасників конференції, що вимагає багато процесорних ресурсів.

– Програмні рішення – не перекоднують відеопотоки, ресурси процесора витрачаються тільки на передачу даних.

– Апаратні рішення – для реалізації великої кількості обчислень їх часто будують на основі модулів з DSP-Процесорами, причому рішення дозволяють нарощувати потужність за рахунок збільшення кількості модулів. Такі рішення більше дорогі.

– Програмні ВКЗ-рішення – не вимагають перекодування відео, тому можуть бути побудовані на базі процесорів загального призначення, улаштовані набагато простіше й, отже, більше доступні.

Моделі впровадження ВКЗ системи

1. Виділене рішення. Вся інфраструктура належить і обслуговується замовником самостійно.

2. Хмарне рішення. Сервер ВКЗ десь в інтернеті. Термінали ВКЗ у мережі замовника.

3. Віддалене керування. Сервера й термінали замовника перебувають під контролем сервісу-провайдеру

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Принципи роботи класичної системи апаратного сервера ВКЗ

MCU (сервер ВКЗ) приймає відеопотоки від кожного терміналу, декодує, мікшує в один потік, потім кожному терміналу кодує цей потік і відсилає.

Переваги й недоліки апаратних рішень

– Переваги:

– Економія каналів зв'язку.

– Відсутність проблеми «слабкого встаткування» за рахунок використання MCU і технології масштабованого відеокодування (SVC).

– Стандартизовані протоколи. Всі апаратні рішення створювалися за загальними стандартами й специфікаціям – за рахунок цього вони глибоко сумісні між собою. Як мінімум – аудіо й відео між різними пристроями завжди сумісні.

– Централізоване керування.

Недоліки:

– Вимагає кваліфікованої підтримки, тому що всі сучасні апаратні рішення досить складні й мають потребу в обслуговуванні висококваліфікованих і сертифікованих фахівців.

– Складності з масштабуванням. Вони виникають при обслуговуванні апаратної системи, коли з'являється потреба в збільшенні кількості серверів і користувачів.

– Складності з розширенням функціонала – інструменти спільної роботи, запис відеоконференцій. Для цього також необхідні додаткові модулі (“залізо”), що позначається на вартості системи ВКЗ.

Архітектура програмних рішень ВКЗ

Програмні рішення бувають різних типів:

– Клієнт-серверні: зв'язок між терміналами здійснюється через єдиний сервер. Це централізована система, має гарну керованість, висока надійність досягається дублюванням серверів. Недоліки – всі дані йдуть через єдину точку (сервер) і якість зв'язку залежить від того, де розташований сервер.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

– P2P рішення (peer-to-peer, P2P – рівний до рівного). Децентралізована система, має високу надійність, але низкою керованістю. Трафік проходить по різних вузлах мережі й різні точки зв'язку (звичайно термінали). Недолік – якість зв'язки може мінятися від шляху проходження даних і залежить також від потужності вузлів.

– Гібридні – поєднує переваги обох систем. Трафік між терміналами йде або прямо між терміналами, або через сервер, залежно від аналізу каналів зв'язку.

Класична архітектура ВКЗ на основі ПЗ

Програмна платформа комутирує вхідні потоки від учасників, розмножує їх відповідно до кількості приймаючих учасників і відсилає кожному користувачеві.

Тобто, у симетричній груповій конференції з 4 учасниками користувач, що відправляє відео з дозволом 640x360 (SD – стандартний дозвіл), одержить у підсумку відео з дозволом $4 * (640x360) = 1280x720$ (HD – високої чіткості).

Користувач одержує максимальне значення – суму всіх дозволів учасників груповий ВКЗ.

Поліпшена архітектура ВКЗ без перекодування на основі ПЗ

За рахунок використання технології масштабованого відеокодування (SVC), кожний учасник зможе одержати картинку, адаптовану під його термінал. Так наприклад, якщо термінал – це телефон з дозволом екрана 800x480, то дозвіл 1280x720 уже надлишково для нього й сервер зменшує дозволу потоків, що відсилаються, відео.

Такі рішення випускають уже сьогодні, принцип їхньої роботи полягає в тім, що термінали ВКЗ кодують відео спеціальним образом, так, що стислий потік можна розділити на декілька підпотоків різного дозволу або різної частоти кадрів без декодування відео. ВКЗ сервер виступає в ролі розподільника якості – “проріджує” потоки під умови, що змінюються, каналу або потужності терміналу вибираючи оптимальний на даний момент підпотік таким чином, щоб ні у вас, ні у ваших співрозмовників не виникало проблем у спілкуванні. В ідеалі, все це

									ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата						29

дозволяє на основі програмного рішення побудувати повну емуляцію апаратного MCU, тільки тут замість перекодування використовується проріджені пакети, що набагато ефективніше з погляду процесорних ресурсів.

Які клієнтські додатки бувають?

- Десктопні.
- Мобільні.
- Браузерні (WebRTC).
- Smart TV (домашні телевізори).

Переваги й недоліки програмних рішень

Переваги:

– Можлива побудова сервера на базі процесорів загального призначення й стандартних серверів.

- Швидкість впровадження.
- Можливість тестування.
- Легка розширюваність.
- Можливість модернізації інтерфейсу/функціонала.
- Можливість використання моделі SaaS (зовнішній сервер).
- Використання вже наявних пристроїв
- Відсутність проблеми «слабкого встаткування» – за рахунок

використання SVC.

- Легко масштабується – все залежить від ширини каналу на сервері.
- Весь функціонал уже на борті.

Недоліки:

- Як термінали ВКЗ використовуються звичайні ПК.
- Периферійне встаткування потрібно здобувати окремо.

Виходячи з усього вищенаведеного на рисунку 3.1 зображена структурна схема системи.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

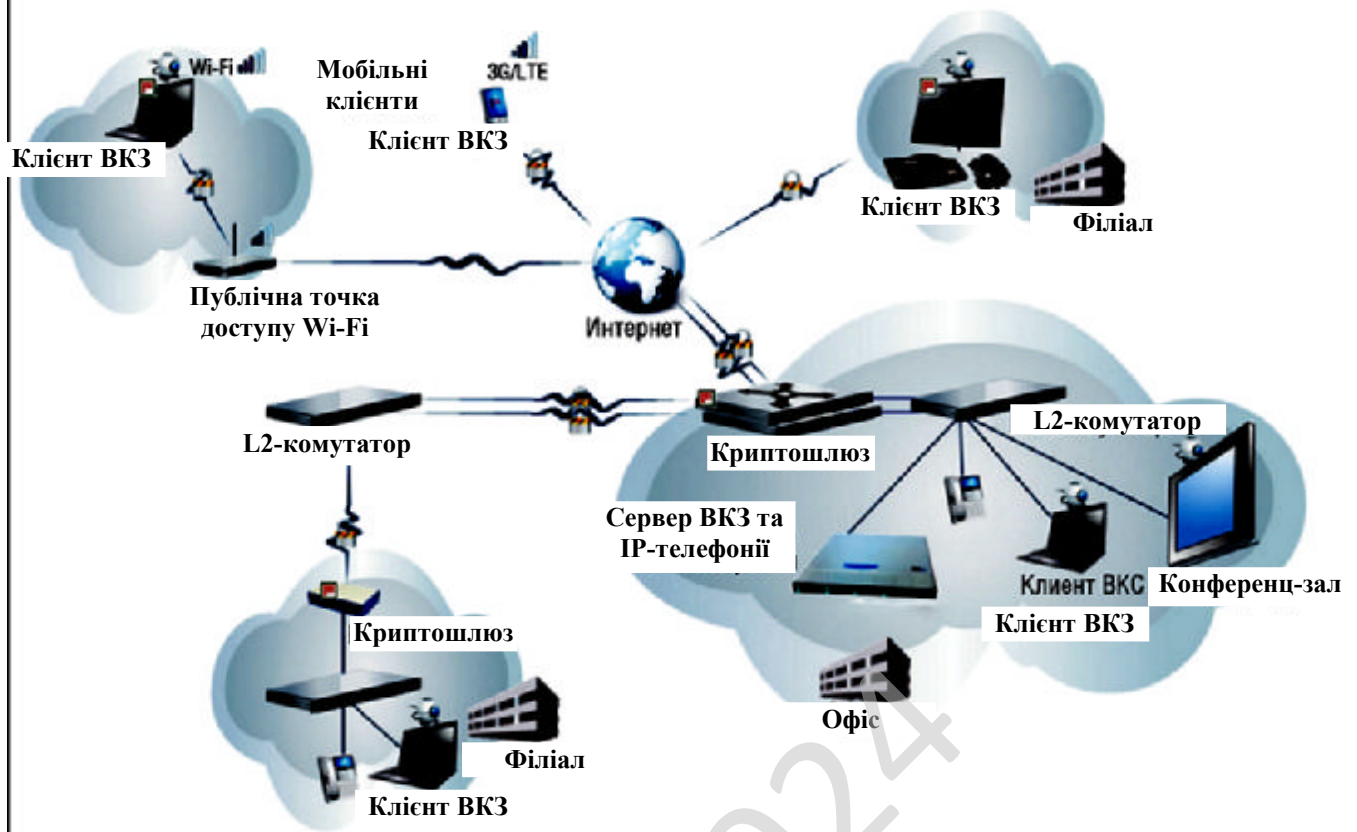


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

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

1. Головне вікно системи відео-конференц-зв'язку на основі серверів MCU.

2. Блок показу презентації. Завантаження й демонстрація документів.

Підтримувані формати:

- документи у форматі PDF;
- різні типи документів Microsoft Office;
- різні типи документів OpenOffice;
- перемикання ведучого: право на завантаження презентації.

3. Блок керування презентацією. Проведення презентації й створення своїх позначок:

- показ слайдів: послідовний або довільний;
- розмір слайда: по розмірі вікна або по ширині вікна;
- масштабування: збільшення зображення;
- позначки на слайді: малювання маркером, малювання окружностей, прямокутників, вибір кольору й товщини ліній;
- перемикач ведучого: керування презентацією.

4. Блок аудіоконференції. Можливість організації повноцінної аудіоконференції з керуванням:

- кожний учасник може управляти своїм мікрофоном;
- модератор може включити або виключити мікрофон учасника в одне натискання;
- модератор може включити або виключити мікрофон групи учасників в одне натискання.

5. Блок відеоконференції. Можливість організації повноцінної відеоконференції:

- трансляція відео з камери (tv-тюнера, плати відеозахвата й т.д.) ведучого;
- трансляція відео з камери учасників для живого спілкування;
- керування відеоконференцією.

6. Блок чату. Обговорення питань у форматі чату:

- організація загального чату для питань/відповідей або обговорення;
- організація персонального чату для особистого спілкування;
- можливість вибору кольору для своїх повідомлень.

7. Блок запису і перегляду відеоконференцій:

- запис і показ демонструємих слайдів;
- запис і відтворення позначок ведучого на слайдах;
- запис і відтворення аудіоконференції;

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- запис і відтворення спілкування в чаті;
- керування записами.

8. Блок трансляції робочого стола:

- трансляція усього робочого стола ведучого;
- трансляція області робочого стола ведучого.

9. Блок організації опитування:

- створення опитувань, попередній перегляд;
- швидке проведення опитування;
- проведення веб-опитування;
- підведення підсумків опитування й вивід результатів.

10. Блок інтерфейсу. Простий і зручний для учасників інтерфейс:

- доступ через браузер (непотрібно ніяких програм, налаштувань портів);
- підтримка більше 40 мов інтерфейсу;
- адаптація (ребрендинг) інтерфейсу.

11. Блок інтеграції. Взаємодія Ваших проектів із сервером відео-конференц-зв'язку:

- наявність API для JSP;
- наявність API для PHP;
- наявність API для Ruby.

Інтеграція. Готові модулі інтеграції з розповсюдженими системами:

- Moodle;
- WordPress;
- Drupal;
- Joomla;
- Canvas LMS;
- Sakai;
- Tiki Wiki CMS Groupware;
- eFront;
- Chamilo LMS.

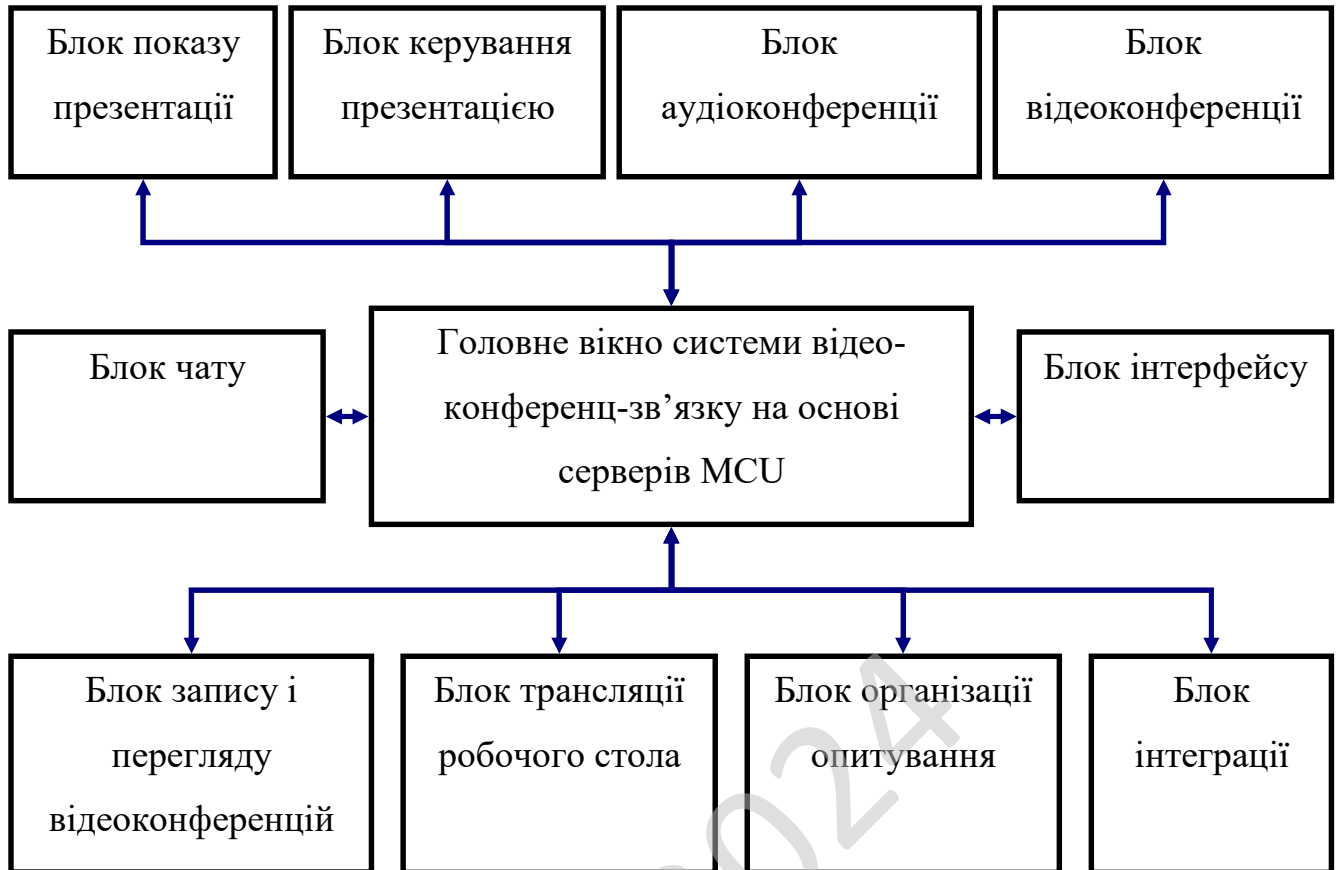


Рисунок 3.2 – Функціональна схема системи

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

3.4 Розробка діаграми процесів

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. Діаграма процесів розробленої системи зображена на рисунку 3.3.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки 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, також те, що при розробці програми слід надати особливу увагу модулю системи відео-конференц-зв'язку на основі серверів MCU.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

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 – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

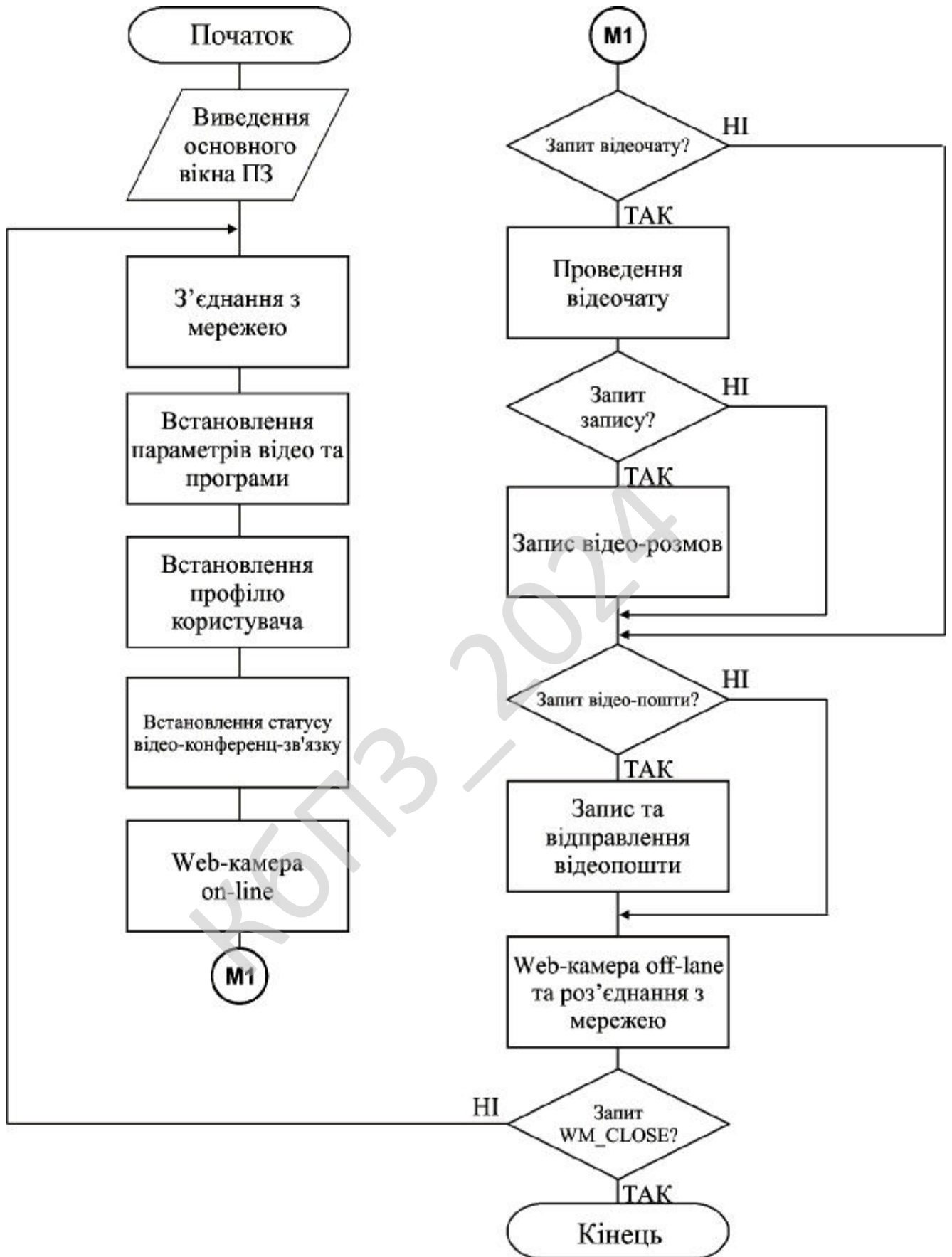


Рисунок 4.1 – Блок-схема основної програми

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

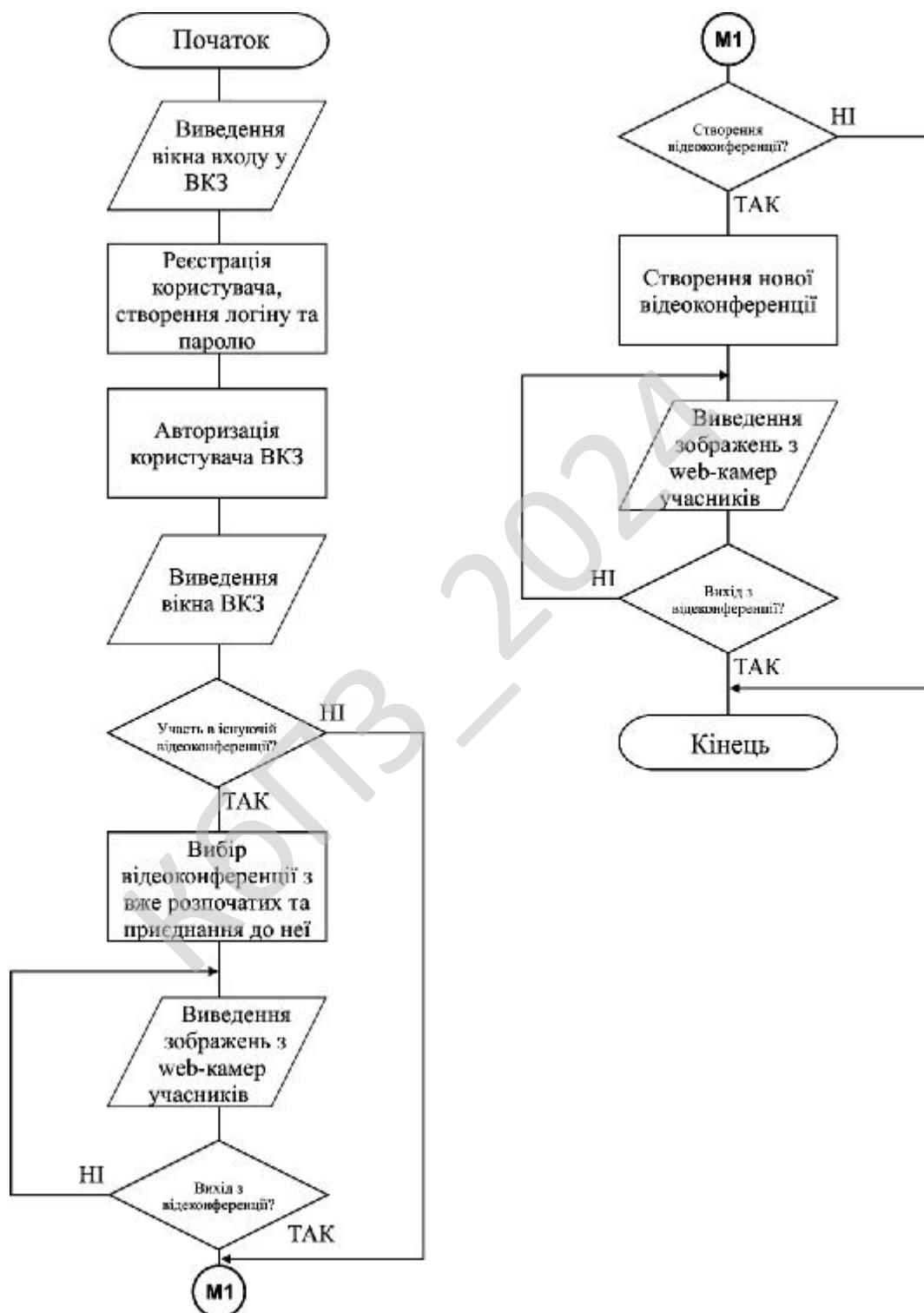


Рисунок 4.2 – Блок-схема роботи підпрограми

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.

Система відео-конференц-зв'язку на основі серверів MCU (Multipoint Control Unit) є комплексним рішенням для організації багатокористувацьких відеоконференцій у реальному часі. Нижче наведено структуру системи, основні модулі та архітектуру, а також реалізацію на Python. Розрахунки та обґрунтування технічних рішень додаються після основного коду.

Основні компоненти системи:

1. Клієнтський модуль. Відповідає за обробку аудіо та відео з пристроїв користувача (камера, мікрофон). Забезпечує відправку даних на сервер MCU. Приймає та відтворює відеопотоки з сервера.

2. Сервер MCU. Основний елемент, який обробляє всі відеопотоки. Об'єднує кілька відеопотоків в єдиний потік для кожного учасника. Забезпечує комунікацію між клієнтами, зменшує затримку та оптимізує трафік.

3. База даних користувачів. Зберігає дані про користувачів конференції, їх права доступу та конфігурацію.

4. Контролер конференцій. Відстежує стан конференцій, підключення/відключення учасників, керує створенням та завершенням сесій.

Основний код системи:

```
import socket
import threading
import cv2
import numpy as np
```

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

```

class VideoClient:
    def __init__(self, server_ip, server_port):
        self.server_ip = server_ip
        self.server_port = server_port
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        self.socket.connect((self.server_ip, self.server_port))
        threading.Thread(target=self.receive_video).start()
        self.send_video()

    def send_video(self):
        cap = cv2.VideoCapture(0)
        while True:
            ret, frame = cap.read()
            if not ret:
                break
            data = cv2.imencode('.jpg', frame)[1].tobytes()
            self.socket.sendall(data)

    def receive_video(self):
        while True:
            data = self.socket.recv(1024*64)
            np_data = np.frombuffer(data, np.uint8)
            frame = cv2.imdecode(np_data, cv2.IMREAD_COLOR)
            cv2.imshow("Video Stream", frame)
            if cv2.waitKey(1) == 27:
                break
        cv2.destroyAllWindows()

class MCU:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.clients = []
        self.server_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

    def start(self):
        self.server_socket.bind((self.host, self.port))
        self.server_socket.listen(5)
        print("MCU Server started")

```

					БКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

```

while True:
    client_socket, addr = self.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:
            data = client_socket.recv(1024*64)
            self.broadcast(data, client_socket)
        except:
            self.clients.remove(client_socket)
            client_socket.close()
            break

def broadcast(self, data, sender_socket):
    for client in self.clients:
        if client != sender_socket:
            client.sendall(data)

if __name__ == "__main__":
    mcu = MCU("0.0.0.0", 5000)
    threading.Thread(target=mcu.start).start()

    client1 = VideoClient("localhost", 5000)
    threading.Thread(target=client1.connect).start()

```

Обґрунтування проектних рішень

1. Розрахунок ширини каналу зв'язку. Сервер MCU отримує відеопотоки з усіх клієнтів та об'єднує їх в один, який потім передає іншим клієнтам. Для обчислення мінімальної ширини каналу необхідно враховувати такі параметри: Роздільна здатність відео: 720р (1280x720 пікселів); Кількість кадрів: 30 fps; Стиснення: H.264.

Формула обчислення ширини каналу:

*Ширина_каналу = Роздільна_здатність * Кількість_кадрів * Бітрейт*

(При середньому бітрейті 3 Мбіт/с ширина каналу для кожного потоку складає приблизно 3 Мбіт/с.)

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

2. Підтримка одночасних користувачів. Сервер MCU у тестовій конфігурації підтримує до 10 одночасних клієнтів. При збільшенні кількості клієнтів необхідно масштабувати сервер або використовувати додаткові сервери MCU.

3. Обґрунтування використання багатопоточності. Для обробки відеопотоків у реальному часі багатопоточність дозволяє розподілити навантаження між різними потоками, що зменшує затримку та покращує якість зв'язку.

Це базова реалізація системи відео-конференц-зв'язку з використанням серверів MCU. Для подальшого масштабування та оптимізації можна додати функціонал автоматичного масштабування та використання більш складних алгоритмів компресії.

При роботі над ПЗ що розробляється використовувалась система Jira. Це комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами.

Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gozilla» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру.

Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скріншотами) або коментарями.

Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настраюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens).

Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи російську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- Вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;
- програмне забезпечення від Atlassian є на веб-сайті проекту.

4.2 Захист розробленого програмного забезпечення

Tiny Encryption Algorithm (TEA) [1] – блочний алгоритм шифрування типу «Мережі Фейстеля». Алгоритм був розроблений на факультеті комп'ютерних наук Кембриджського університету Девідом Вілером (David Wheeler) і Роджером Нідгемом (Roger Needham) та вперше представлений в 1994 році [2] на симпозиумі зі швидкими алгоритмами шифрування в Льовені (Бельгія).

Шифр не патентований, широко використовується в ряді криптографічних додатків і широкому спектрі апаратного забезпечення, завдяки вкрай низькими вимогами до пам'яті й простоті реалізації. Алгоритм має як програмну реалізацію на різних мовах програмування, так і апаратну реалізацію на інтегральних схемах типу FPGA.

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму TEA, який заснований на бітових операцій з 64-бітним блоком, має 128-бітний ключ шифрування. Стандартна кількість раундів мережі Фейстеля біля 64 (32 циклу), однак, для досягнення найкращої продуктивності або шифрування, число циклів можна варіювати від 8 (16 раундів) до 64 (128

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

раундів). Мережа Фейстеля несиметрична через використання в якості операції накладення додавання за модулем ²³².

Перевагами шифру є його простота в реалізації, невеликий розмір коду й досить висока швидкість виконання, а також можливість оптимізації виконання на стандартних 32-бітних процесорах, так як в якості основних операцій використовуються операції виключна «АБО» (XOR), побітового зсуву й додавання за модулем ²³². Оскільки алгоритм не використовує таблиць підстановки і раундова функція досить проста, алгоритму потрібно не менше 16 циклів (32 раундів) для досягнення ефективної дифузії, хоча повна дифузія досягається вже через 6 циклів (12 раундів).

Алгоритм має відмінну стійкість до лінійного криптоаналізу і досить гарну до диференціального криптоаналізу. Головним недоліком цього алгоритму шифрування є його вразливість до атак «на пов'язаних ключах» (англ. Related-key attack). Через простий розклад ключів кожен ключ має 3 еквівалентних ключа. Це означає, що ефективна довжина ключа складає всього 126 біт [3] [4], тому даний алгоритм не слід використовувати в якості геш-функції.

Опис алгоритму

Вихідний текст розбивається на блоки по 64 біта кожен. 128-бітний ключ K ділиться на чотири 32-бітних підключа $K[0]$, $K[1]$, $K[2]$ і $K[3]$. На цьому підготовчий процес закінчується, після чого кожен 64-бітний блок шифрується протягом 32 циклів (64 раундів) за нижченаведеним алгоритмом. [5]

Припустимо, що на вхід n -го раунду ($1 \leq n \leq 64$) надходять права й ліва частини (L_n, R_n) , тоді на виході n -го раунду будуть ліва й права частини (L_{n+1}, R_{n+1}) , які обчислюються за такими правилами:

$$L_{n+1} = R_n.$$

Якщо $n = 2 * i - 1$ для $1 \leq i \leq 32$ (непарні раунди), то:

$$R_{n+1} = L_n(\{ [R_n 4] K[0] \} \{ R_n i * \delta \} \{ [R_n 5] K[1] \}).$$

Якщо $n = 2 * i$ для $1 \leq i \leq 32$ (парні раунди), то:

$$R_{n+1} = L_n(\{ [R_n 4] K[2] \} \{ R_n i * \delta \} \{ [R_n 5] K[3] \}).$$

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Де

$X \oplus Y$ – операція додавання чисел X і Y за модулем 232.

$X \oplus Y$ – побітове виключне АБО» (XOR) чисел X і Y , яке в мові програмування Сі позначається як $X \wedge Y$

$X \ll Y$ і $X \gg Y$ – операції побітового зсуву числа X на Y біт вліво й вправо відповідно.

Константа δ була виведена з Золотого перерізу:

$$\delta = (-1) * 2^{31} = 2654435769 = 9E3779B9_h.$$

У кожному раунді константа множиться на номер циклу i . Це було зроблено для запобігання простих атак, заснованих на симетрії раундів.

Також очевидно, що в алгоритмі шифрування TEA немає як такого алгоритму розкладу ключів. Замість цього в непарних раундах використовуються підключі $K [0]$ та $[1]$, у парних – $K [2]$ і $[3]$.

Так як це блочний шифроалгоритм, де довжина блоку 64-біт, а довжина даних може бути не кратна 64-біт, значення всіх байтів, які доповнюють блок до кратності в 64-біт, встановлюється в 0x01.

КБПЗ-2024

					VKPM-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи відео-конференц-зв'язку на основі серверів MCU яке зображено на рисунку 5.1а та налаштування на рисунку 5.1б.

З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні блоки:

- Навігаційне меню: Дії; Сервіс; Довідка.
- Блоку виведення роботи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

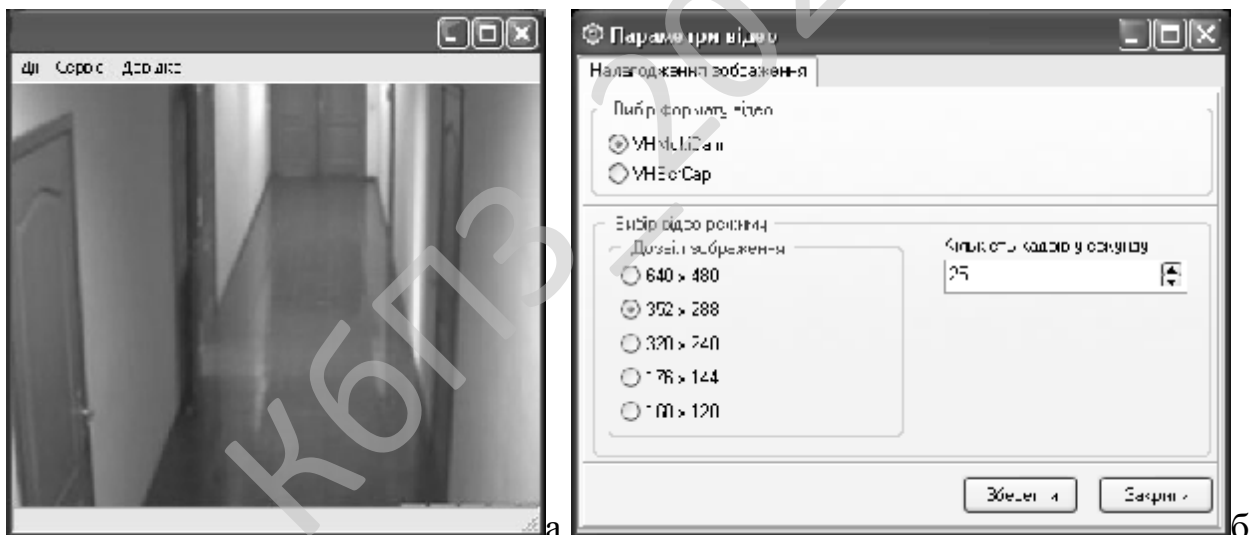


Рисунок 5.1 – Інтерфейс ПЗ: а- головне вікно ПЗ, б – вікно налаштувань

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

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

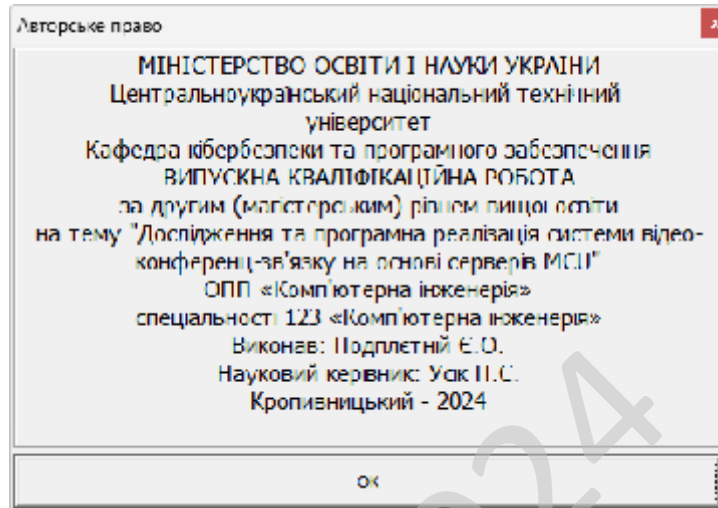


Рисунок 5.2 – Авторське право

Перехід на менш дороге (бюджетне) рішення дозволяє прискорити розвиток системи відео-конференц-зв'язку й підключити до неї основні підрозділи. Розгорнута система ВКЗ успішно застосовується для проведення відеоселекторних нарад, дистанційної атестації й телелекцій, з її допомогою проводяться конференції. Головний результат впровадження ВКЗ – підвищення ефективності системи керування й прийняття рішень у центрі. Крім того, застосування відеозв'язку дозволило раціоналізувати використання робочого часу співробітників, дало додаткові інструменти для підвищення їхньої кваліфікації, сприяло зниженню витрат на відрядні витрати й на міжміські переговори.

Разом з тим слід зазначити, що розгорнута система ВКЗ використовується в основному для рішення офісних завдань. При всіх перевагах програмних систем, у рішень на основі спеціалізованих апаратних комплексів залишається більша область застосування – зокрема, у вертикальних додатках.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

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.

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

Проводилось чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

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.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

КБПЗ-2024

					VKPM-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи відео-конференц-зв'язку на основі серверів MCU.

Метою розробки є дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU.

Об'єктом дослідження є процес відео-конференц-зв'язку на основі серверів MCU.

Предметом дослідження є методи відео-конференц-зв'язку на основі серверів MCU.

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

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

- Удосконалено метод відео-конференц-зв'язку на основі серверів MCU.
- Розроблено вітчизняний продукт відео-конференц-зв'язку на основі серверів MCU, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU можуть бути цікавими для різних груп та організацій, які шукають рішення для ефективно організації відеоконференцій та онлайн-зв'язку (рисунок 7.1).



Рисунок 7.1 – Цільова аудиторія

Отже, результати дослідження та програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU будуть цікавими для широкого кола професіоналів, компаній і організацій, які зацікавлені у вдосконаленні своїх комунікаційних можливостей через надійні, масштабовані та високоякісні відеоконференційні системи.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU може бути виконана з використанням методів експертних оцінок. Перш ніж проводити оцінку, необхідно визначаємо критерії, за якими експерти оцінюватимуть проект (рисунок 7.2).

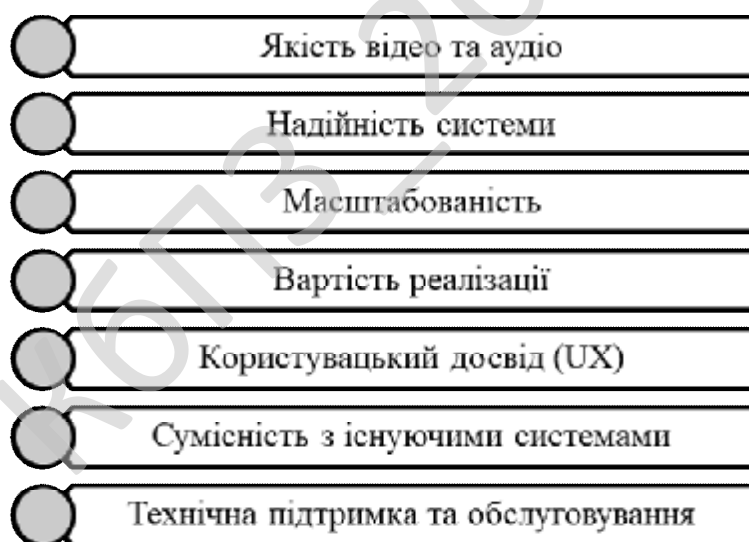


Рисунок 7.2 – Критерії експертної оцінки

Для проведення оцінки залучаємо групу експертів, які мають досвід у сфері відео-конференц-зв'язку, інформаційних технологій та бізнесу. Експертами обираємо фахівців з: інформаційних технологій, відео-комунікацій, бізнес-аналізу, маркетингу.

Один з методів експертних оцінок – це метод "Дельфі". Експерти анонімно оцінюють кожен з критеріїв за шкалою, наприклад, від 1 до 5, де 1 – дуже погано, 5 – дуже добре. Результати зводимо до таблиці 7.1.

Таблиця 7.1 – Середнє значення експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Середнє значення
Якість відео та аудіо	5	4	5	4.67
Надійність системи	4	5	4	4.33
Масштабованість	3	4	3	3.33
Вартість реалізації	4	3	4	3.67
Користувацький досвід	5	5	4	4.67
Сумісність	4	3	4	3.67
Технічна підтримка	5	4	5	4.67

На основі отриманих середніх значень експертних оцінок можна провести подальший аналіз, наприклад, порівняти з конкурентами, визначити сильні та слабкі сторони системи. Для підсумкової оцінки привабливості проекту вводимо вагові коефіцієнти для кожного критерію залежно від їх важливості. Зведені результати вносимо в таблицю 7.2.

Таблиця 7.2 – Зведені результати оцінювання

Критерій	Середнє значення	Ваговий коефіцієнт	Вагомий бал
Якість відео та аудіо	4.67	0.25	1.17
Надійність системи	4.33	0.20	0.87
Масштабованість	3.33	0.15	0.50
Вартість реалізації	3.67	0.15	0.55
Користувацький досвід	4.67	0.15	0.70
Сумісність	3.67	0.05	0.18
Технічна підтримка	4.67	0.05	0.23

Сумарний бал проекту підраховуємо шляхом додавання всіх вагомих балів. Загальний вагомий бал = $1.17 + 0.87 + 0.50 + 0.55 + 0.70 + 0.18 + 0.23 = 3.50$

Залежно від отриманого підсумкового балу, можна визначити привабливість проекту: висока привабливість: 4.0 - 5.0, середня привабливість: 2.5 - 4.0, низька привабливість: 0 - 2.5.

В даному прикладі, отримане значення 3.50 може вказувати на середню привабливість проекту, що вказує на необхідність подальшої роботи над поліпшенням окремих критеріїв, таких як масштабованість або вартість реалізації. Цей підхід дозволяє систематизувати думки експертів та приймати обгрунтовані рішення щодо подальшого розвитку проекту.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU можна використовувати кілька методів (рис.7.3).

Для програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU доцільно використовувати комбінований підхід, поєднавши запропоновані Це дозволить отримати всебічну картину вартості проекту та його привабливості з точки зору інвесторів і бізнесу.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Оцінка економічної ефективності впровадження системи відео-конференц-зв'язку на основі серверів MCU може включати кілька ключових етапів: оцінка витрат, потенційних вигод і порівняння їх для визначення чистої вигоди. Економічна ефективність від впровадження проекту для клієнта схематично подана на рисунку 7.4.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

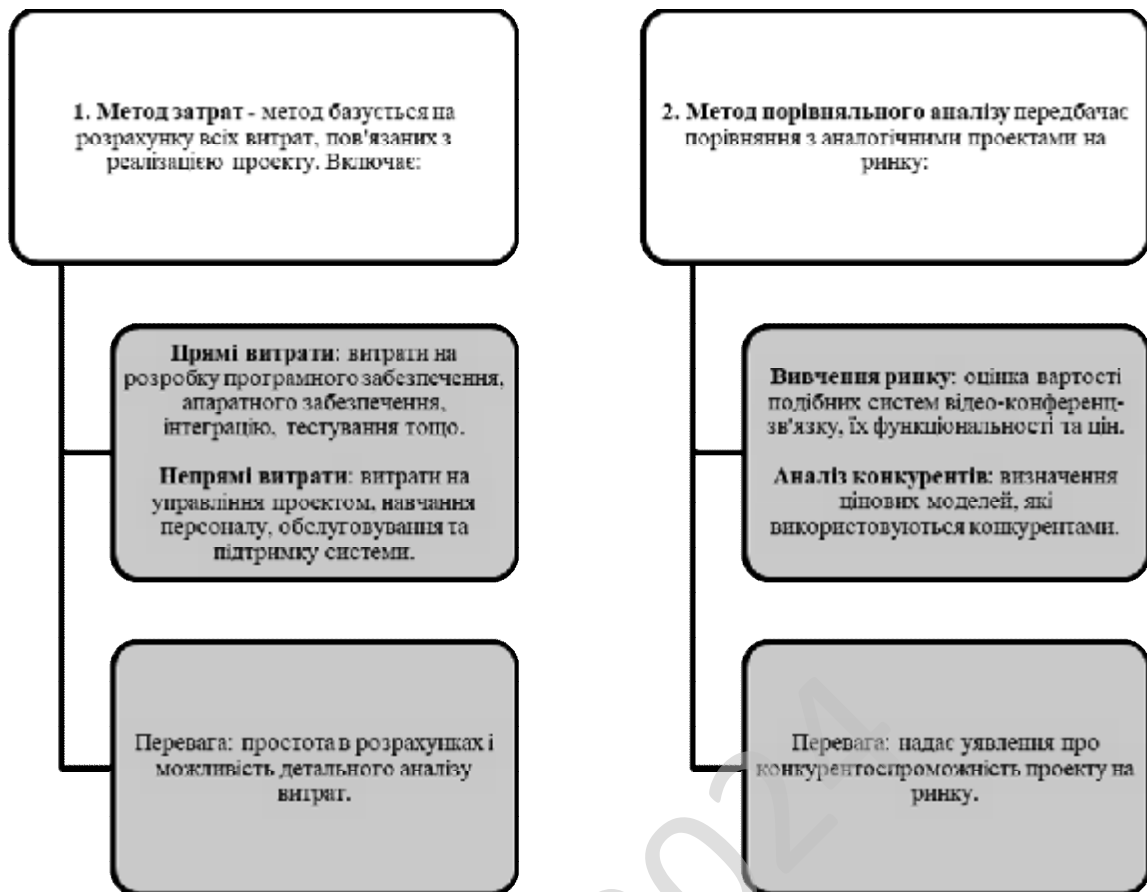


Рисунок 7.3 – Методи оцінки вартості



Рисунок 7.4 – Економічна ефективність від реалізації проєкту для клієнта

Проект з впровадження системи відео-конференц-зв'язку на основі серверів MCU є економічно ефективним, з позитивним ROI в 69% і терміном окупності менше ніж 2 роки. Це дозволяє компанії значно знизити витрати на поїздки та підвищити ефективність внутрішньої і зовнішньої комунікації.

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Алгоритм просування проекту програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU може бути розроблений у кілька етапів. Структурований підхід, який допоможе ефективно просунути проект представлено на рисунку 7.5.

Цей алгоритм просування дозволяє систематизувати процес просування проекту програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU, зосередившись на цільовій аудиторії, унікальних перевагах продукту та ефективних маркетингових інструментах.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU є важливим етапом для забезпечення успішного виходу на ринок та підвищення ефективності продажів. Корисним буде:

- мультиканальна стратегія збуту;
- зміцнення відносин з клієнтами;
- програми лояльності та стимулювання;
- інтеграція з іншими рішеннями;
- маркетинг і просування;
- навчання та підтримка;
- аналіз та корекція.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

1. Аналіз ринку та конкурентів

- Дослідження потреб
- Аналіз конкурентів

2. Розробка стратегії позиціонування

- Унікальна пропозиція цінності (USP)
- Цільовий сегмент

3. Створення маркетингових матеріалів

- Презентації та брошури
- Веб-сайт
- Відео-демонстрації

4. Запуск маркетингової кампанії

- Цифровий маркетинг
- E-mail маркетинг
- Блог та контент-маркетинг

5. Участь у виставках та конференціях

- Виставки та заходи
- Презентації та майстер-класи

6. Пробна версія або демонстраційна версія

- Доступ до безкоштовної версії
- Зворотній зв'язок

7. Партнерства та співпраця

- Співпраця з IT-компаніями
- Програми лояльності

8. Оцінка та аналіз результатів

- Моніторинг KPI

Корекція стратегії

Рисунок 7.5 – Алгоритм просування проекту

Оптимізація каналів збуту та шляхів реалізації системи відео-конференц-зв'язку на основі серверів MCU вимагає комплексного підходу, що поєднує ефективні дистрибуційні стратегії, зміцнення відносин з клієнтами, маркетингові ініціативи та постійний аналіз результатів. Цей підхід допоможе вам підвищити конкурентоспроможність та забезпечити успішне впровадження вашого продукту на ринку.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU включають в себе кілька важливих аспектів, які можуть суттєво вплинути на його результативність і популярність серед користувачів (рисунок 7.6). Ось основні з них:



Рисунок 7.6 – Ключові фактори успіху проєкту

Ключові фактори успіху проєкту програмної реалізації системи відео-конференц-зв'язку на основі серверів MCU є взаємопов'язаними і їхнє комплексне врахування допоможе забезпечити успішне впровадження і конкурентоспроможність продукту на ринку. Сфокусування на технічній надійності, зручності використання, безпеці, підтримці клієнтів і ефективному маркетингу – основа для досягнення успіху.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Комп'ютерні мережі вриваються у життя людей в їх професійну діяльність найнесподіванішим і масовим чином. Вони породили істотно нові технології обробки інформації – мережні технології. У найпростішому разі мережні технології дозволяють спільно використовувати ресурси – нагромаджувачі великий ємності, друкують устрою, доступ в Internet, бази й банки даних. Найбільш сучасні і перспективні підходи до мереж пов'язані з допомогою колективного поділу праці – розробці різних документів і майже проектів, управлінні установою чи підприємством, і т. д.

Впровадження комп'ютерних технологій принципово змінило характер праці різних категорій фахівців. Працівники, використовують комп'ютерну техніку, на своєму досвіді оцінили її величезні можливості. Одночасно виникла певна безтурботність при її експлуатації.

Недотримання вимог безпеки призводить до того, що й через кілька днів роботи за комп'ютером співробітник починає відчувати певний дискомфорт: в нього виникає головний біль і різь у власних очах, з'являються почуття виснаження й дратівливості. В окремих людей порушується сон, погіршується зір, занедужують руки, шия, поперек тощо.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці.

8.2 Аналіз умов праці

Приміщення розташовано на третьому поверсі п'ятиповерхового будинку. У приміщенні розташовано 3 робочих місць з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м², об'єм не менше 20 м³. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м². Необхідна площа на 3 робочих місця із установленими ПК складає 18 м², що не перевищує фактичну. Обсяг кабінету на одного працюючого складає 31,5м³, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м³.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

По категорії вибухо- і пожежонебезпеки, згідно дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії пожежонебезпеки і поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [3] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

По ступені небезпеки поразки людей електричним струмом відділ, згідно, класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціонування в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціонування» [4]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам відповідно до ДСанПіНЗ.3.2.007 – 98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно – обчислювальних машин».

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м³.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [5]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

Одним з найбільш поширеніших чинників зовнішнього середовища, який несприятливо впливає на людину, є шум. Вплив шуму на організм людини

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

залежить від рівня звукового тиску, частотних характеристик, тривалості дії, а також індивідуальних особливостей людини.

При тривалій дії шуму у оператора ЕОМ виявляються симптоми утомленості, нервового збудження, що сприяють погіршенню працездатності і допущенні помилок при роботі. Для уникнення шкідливої дії шуму на організм працюючого, необхідне дотримання нормованих параметрів, які не повинні перевищувати допустимих величин. При роботі на комп'ютері рівень шуму не повинен перевищувати 50 дБА. Приміщення розташоване вікнами у двір і знаходиться далеко від проїжджої частини вулиці. Основними джерелами шуму в приміщенні є устаткування і люди. Розглянута кімната не призначена для прийому відвідувачів і тому в ній не спостерігається великого скупчення людей. Тому основним джерелом шуму є комп'ютерна техніка.

Джерелами шуму при роботі ЕОМ є механічні частини принтера, що рухаються, і вентилятори ($L_{pk} = 35$ дБА, $L_{rpm} = 48$ дБА) При роботі вентиляційної системи, що забезпечує оптимальний температурний режим електронних блоків ЕОМ і вмонтована в задню панель, створюється аеродинамічний шум. Шум, створюваний працюючим комп'ютером, може бути охарактеризований як широко смуговий постійний з аперіодичним посиленням при роботі принтера. Час роботи ПЕОМ – 6 – 8 год. за добу; принтери працюють не більш 1,5-2 год. за добу.

При наявності великої кількості джерел шуму еквівалентне значення шуму $L_{\text{екв}}$, дБА розраховують по наступній формулі:

$$L_{\text{екв}} = 10 \cdot \lg \left(\frac{1}{T} \sum_{i=1}^n (t_i \cdot 10^{0.1 L_i}) \right) \quad (8.1)$$

де L_i – рівень шуму i -го джерела (пристрою),

t_i – час роботи i -го джерела (пристрою),

T – загальний час роботи,

n – кількість джерел шуму даного типу;

Для даного приміщення необхідні змінні складають:

Загальний час роботи – робочий день, тобто $T=8$ годин.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Для фонового шуму (вентиляторів):

$$L_1 = 35 \text{ дБА}, T_1 = 8 \text{ годин}, n_1=15 (5 \times 3);$$

Для лазерного принтера Lexmark Jet:

$$L_2 = 48 \text{ дБА}, T_2 = 2 \text{ години}, n_2=1, \text{ для сканера } L_3 = 46 \text{ дБА}, T_3 = 2 \text{ години}.$$

Підставляємо отримані величини у формулу (8.1):

$$L_{\text{СКВ}} = 10 \cdot \lg \left(\frac{1}{8} \cdot (15 \cdot 8 \cdot 10^{0,1 \cdot 35} + 1 \cdot 2 \cdot 10^{0,1 \cdot 48} + 1 \cdot 2 \cdot 10^{0,1 \cdot 46}) \right) = 46,3 \text{ дБА}$$

Таким чином, еквівалентний рівень шуму в приміщенні за робітник день $L_{\text{екв}} = 46,3 \text{ дБА}$, тобто не перевищує норму 50 дБА.

8.3 Техніка безпеки та протипожежна профілактика

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізольований кабель. Висота проводки становить 2,2м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення.

Всі обчислювальні машини з'єднані із клемми заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 V. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежобезпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки.

До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту.

Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропонується використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості, проведених відділом охорони праці.

В якості мір по зниженню шуму можна запропонувати:

- облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 до);
- екранування робочого місця (встановленням перегородок, діафрагм);
- установка в комп'ютерних приміщеннях устаткування, що викликає

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

мінімальний шум;

- раціональне планування приміщення.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

8.5 Висновки до розділу

У даному розділі магістерської роботи були виконано аналіз умов праці користувачів ПК, які працюють у зазначеному приміщенні. Проведено перевірку організації робочого місця із відповідними замірами параметрів мікроклімату, освітлення, рівня шуму та розрахунком рівня шуму.

Розроблені заходи щодо поліпшення умов праці дотримання техніки безпеки та проведення протипожежної профілактики дозволить створити умови, які будуть забезпечувати більш комфортну роботу.

КБПЗ - 2024

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи відео-конференц-зв'язку на основі серверів MCU.

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 відео-конференц-зв'язку на основі серверів MCU.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем відео-конференц-зв'язку на основі серверів MCU.
- Досліджена система відео-конференц-зв'язку на основі серверів MCU.
- На основі отриманих результатів досліджень створена програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання відео-конференц-зв'язку на основі серверів MCU.

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

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

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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 TEA.

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 IT-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Подплетній Є.О. Дослідження та програмна реалізація системи відео-конференц-зв'язку на основі серверів MCU // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
3. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
4. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
5. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
6. 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.
7. 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.
8. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
9. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
10. 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.
11. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447

12. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

14. 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.

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

16. 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.

17. 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>

18. 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.

					BKPM-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

19. 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.

20. 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.

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

22. 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.

23. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

24. Smirnov O., 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

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

33. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of

					БКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

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

47. 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.

48. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

49. 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).

50. 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

51. 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.

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

					ВКРМ-123.24.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.24.0031.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Подплетний Є.О.				Літ.	Аркуш	Аркушів
Перевірів	Усік П.С.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи відео-конференц-зв'язку на основі серверів MCU.

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. № 19-13 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 відео-конференц-зв'язку на основі серверів MCU.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0031.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи відео-конференц-зв'язку на основі серверів MCU;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.24.0031.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0031.00.00.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 повинна бути розглянута техніка безпеки та протипожежна профілактика.

					ВКРМ-123.24.0031.00.00.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 – 78 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 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 18.12.2024 р.

					ВКРМ-123.24.0031.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Усік П.С.

*Дослідження та програмна реалізація
системи відео-конференц-зв'язку на основі серверів MCU*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

Основна програма

```

import socket
import threading
import cv2
import numpy as np
import time
import json

class VideoClient:
    def __init__(self, server_ip, server_port, username):
        self.server_ip = server_ip
        self.server_port = server_port
        self.username = username
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        self.socket.connect((self.server_ip, self.server_port))
        threading.Thread(target=self.receive_video).start()
        self.send_video()

    def send_video(self):
        cap = cv2.VideoCapture(0)
        while True:
            ret, frame = cap.read()
            if not ret:
                break
            data = cv2.imencode('.jpg', frame)[1].tobytes()
            self.socket.sendall(data)
            time.sleep(0.033) # Затримка для 30fps

    def receive_video(self):
        while True:
            data = self.socket.recv(1024*64)
            np_data = np.frombuffer(data, np.uint8)
            frame = cv2.imdecode(np_data, cv2.IMREAD_COLOR)
            if frame is not None:
                cv2.imshow(f"Video Stream - {self.username}", frame)
                if cv2.waitKey(1) == 27:
                    break
        cv2.destroyAllWindows()

class MCU:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.clients = []
        self.client_usernames = {}
        self.server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def start(self):
        self.server_socket.bind((self.host, self.port))
        self.server_socket.listen(5)
        print("MCU Server started")
        while True:
            client_socket, addr = self.server_socket.accept()
            username = client_socket.recv(1024).decode('utf-8')
            self.client_usernames[client_socket] = username
            self.clients.append(client_socket)
            threading.Thread(target=self.handle_client,
args=(client_socket,)).start()

    def handle_client(self, client_socket):
        while True:
            try:

```

```

        data = client_socket.recv(1024*64)
        if not data:
            break
        self.broadcast(data, client_socket)
    except:
        self.clients.remove(client_socket)
        username = self.client_usernames.pop(client_socket, None)
        client_socket.close()
        self.notify_disconnect(username)
        break

    def broadcast(self, data, sender_socket):
        for client in self.clients:
            if client != sender_socket:
                client.sendall(data)

    def notify_disconnect(self, username):
        if username:
            print(f"User {username} has disconnected")

class ConferenceManager:
    def __init__(self):
        self.conferences = {}

    def create_conference(self, conference_id, host_socket):
        self.conferences[conference_id] = {"host": host_socket, "participants":
[]}
        print(f"Conference {conference_id} created")

    def join_conference(self, conference_id, participant_socket):
        if conference_id in self.conferences:
            self.conferences[conference_id]["participants"].append(participant_socket)
            print(f"Participant joined conference {conference_id}")

    def leave_conference(self, conference_id, participant_socket):
        if conference_id in self.conferences:
            self.conferences[conference_id]["participants"].remove(participant_socket)
            print(f"Participant left conference {conference_id}")

    def end_conference(self, conference_id):
        if conference_id in self.conferences:
            for participant in self.conferences[conference_id]["participants"]:
                participant.close()
            del self.conferences[conference_id]
            print(f"Conference {conference_id} ended")

class AuthenticationModule:
    def __init__(self):
        self.users_db = {"admin": "password", "user1": "password123"}

    def authenticate(self, username, password):
        if username in self.users_db and self.users_db[username] == password:
            print(f"User {username} authenticated successfully")
            return True
        print(f"Authentication failed for user {username}")
        return False

class LoggingModule:
    def __init__(self, log_file="server_log.txt"):
        self.log_file = log_file

    def log_event(self, event):
        with open(self.log_file, "a") as f:
            f.write(f"{time.ctime()} - {event}\n")
        print(f"Logged event: {event}")

class VideoRecorder:

```

```

def __init__(self, output_file="conference_recording.avi"):
    self.output_file = output_file
    self.writer = None
    self.recording = False

def start_recording(self, frame_shape):
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    self.writer = cv2.VideoWriter(self.output_file, fourcc, 30,
(frame_shape[1], frame_shape[0]))
    self.recording = True
    print("Recording started")

def record_frame(self, frame):
    if self.recording and self.writer is not None:
        self.writer.write(frame)

def stop_recording(self):
    if self.writer is not None:
        self.writer.release()
    self.recording = False
    print("Recording stopped")

if __name__ == "__main__":
    logging_module = LoggingModule()
    auth_module = AuthenticationModule()
    conference_manager = ConferenceManager()

    mcu = MCU("0.0.0.0", 5000)
    threading.Thread(target=mcu.start).start()

    username = input("Enter your username: ")
    password = input("Enter your password: ")

    if auth_module.authenticate(username, password):
        client = VideoClient("localhost", 5000, username)
        threading.Thread(target=client.connect).start()

    conference_id = "conf123"
    conference_manager.create_conference(conference_id, client.socket)

    # Додаємо логування подій
    logging_module.log_event("Server started")
    logging_module.log_event(f"User {username} connected")

    # Запускаємо запис конференції
    recorder = VideoRecorder()
    recorder.start_recording((720, 1280))

    # Симуляція додавання та видалення учасників
    conference_manager.join_conference(conference_id, client.socket)
    conference_manager.leave_conference(conference_id, client.socket)
    conference_manager.end_conference(conference_id)

    # Зупиняємо запис
    recorder.stop_recording()

```

user_registration.py

```
import json

class UserManager:
    def __init__(self, db_file="users.json"):
        self.db_file = db_file
        self.users = self.load_users()

    def load_users(self):
        try:
            with open(self.db_file, "r") as f:
                return json.load(f)
        except FileNotFoundError:
            return {}

    def save_users(self):
        with open(self.db_file, "w") as f:
            json.dump(self.users, f, indent=4)

    def register_user(self, username, password):
        if username in self.users:
            print("Username already exists")
            return False
        self.users[username] = {"password": password, "profile": {}}
        self.save_users()
        print(f"User {username} registered successfully")
        return True

    def update_profile(self, username, profile_data):
        if username not in self.users:
            print("User does not exist")
            return False
        self.users[username]["profile"].update(profile_data)
        self.save_users()
        print(f"Profile for {username} updated")
        return True
```

conference_types.py

```
class ConferenceType:
    def __init__(self):
        self.conferences = {}

    def create_conference(self, conference_id, conference_type="public"):
        self.conferences[conference_id] = {
            "type": conference_type,
            "participants": [],
        }
        print(f"Conference {conference_id} of type {conference_type} created")

    def get_conference_type(self, conference_id):
        if conference_id in self.conferences:
            return self.conferences[conference_id]["type"]
        return None

    def set_conference_type(self, conference_id, conference_type):
        if conference_id in self.conferences:
            self.conferences[conference_id]["type"] = conference_type
            print(f"Conference {conference_id} type set to {conference_type}")
```

```
class ChatModule:
    def __init__(self):
        self.messages = {}

    def send_message(self, conference_id, user, message):
        if conference_id not in self.messages:
            self.messages[conference_id] = []
        self.messages[conference_id].append({"user": user, "message": message})
        print(f"Message from {user}: {message}")

    def get_messages(self, conference_id):
        return self.messages.get(conference_id, [])
```

video_recording_and_streaming.py

```
import cv2

class VideoService:
    def __init__(self):
        self.recording = False
        self.stream_target = None
        self.writer = None

    def start_recording(self, output_file="output.mp4", frame_shape=(720,
1280)):
        fourcc = cv2.VideoWriter_fourcc(*'mp4v')
        self.writer = cv2.VideoWriter(output_file, fourcc, 30, (frame_shape[1],
frame_shape[0]))
        self.recording = True
        print("Recording started")

    def stop_recording(self):
        if self.writer is not None:
            self.writer.release()
        self.recording = False
        print("Recording stopped")

    def start_streaming(self, target_url):
        self.stream_target = target_url
        print(f"Streaming started to {target_url}")

    def stop_streaming(self):
        self.stream_target = None
        print("Streaming stopped")

    def record_frame(self, frame):
        if self.recording:
            self.writer.write(frame)

    def stream_frame(self, frame):
        if self.stream_target:
            # Логіка трансляції відео у соціальні мережі
            pass
```

user_roles.py

```
class UserRoles:
    def __init__(self):
        self.roles = {}

    def assign_role(self, username, role):
        self.roles[username] = role
        print(f"User {username} assigned role {role}")

    def get_role(self, username):
        return self.roles.get(username, "participant")

    def is_moderator(self, username):
        return self.get_role(username) == "moderator"

    def is_presenter(self, username):
        return self.get_role(username) == "presenter"
```

K6П3_2024

platform_support.py

```
import platform

class PlatformSupport:
    def check_platform(self):
        current_platform = platform.system()
        print(f"Current platform: {current_platform}")
        if current_platform in ["Windows", "Linux", "Darwin"]:
            print("Platform supported")
            return True
        print("Platform not supported")
        return False
```

encryption_module.py

```
from cryptography.fernet import Fernet

class EncryptionModule:
    def __init__(self):
        self.key = Fernet.generate_key()
        self.cipher = Fernet(self.key)

    def encrypt(self, data):
        encrypted_data = self.cipher.encrypt(data)
        return encrypted_data

    def decrypt(self, encrypted_data):
        decrypted_data = self.cipher.decrypt(encrypted_data)
        return decrypted_data
```

chat_and_scaling.py

```
class ChatService:
    # Реалізація модуля чату, об'єднана з існуючими функціями
    pass

class MCUScaler:
    def __init__(self):
        self.servers = []

    def add_server(self, server):
        self.servers.append(server)
        print(f"Server {server} added for scaling")

    def remove_server(self, server):
        self.servers.remove(server)
        print(f"Server {server} removed from scaling")

    def scale_up(self):
        # Логіка масштабування серверів MCU
        pass
```

calendar_mobile_file_sharing.py

```
import os

class CalendarIntegration:
    def sync_with_calendar(self, user, calendar_type):
        # Логіка синхронізації з Google Calendar або Outlook
        pass

class MobileSupport:
    def check_mobile_compatibility(self, os_type):
        supported_os = ["iOS", "Android"]
        return os_type in supported_os

class FileSharing:
    def send_file(self, file_path, recipient):
        if os.path.exists(file_path):
            # Логіка відправки файлу
            pass

    def receive_file(self, file_path):
        # Логіка отримання файлу
        pass
```

extended_features.py

```
import os
import time
import cv2

class NotesRecorder:
    def __init__(self):
        self.notes = {}

    def add_note(self, conference_id, note):
        if conference_id not in self.notes:
            self.notes[conference_id] = []
            timestamp = time.ctime()
            self.notes[conference_id].append(f"{timestamp}: {note}")
            print(f>Note added to conference {conference_id}")

    def get_notes(self, conference_id):
        return self.notes.get(conference_id, [])

class ExternalDeviceSupport:
    def connect_device(self, device_type):
        if device_type in ["document_camera", "interactive_board"]:
            print(f"{device_type} connected successfully")
        else:
            print(f"Device {device_type} not supported")

class SpeakerDetection:
    def __init__(self):
        self.current_speaker = None

    def detect_speaker(self, audio_stream):
        # Логіка розпізнавання активного доповідача
        # Наприклад, аналіз рівня гучності звуку
        pass

    def highlight_speaker(self, video_frame):
        if self.current_speaker:
            # Виділення активного доповідача на відео
            pass

class CorporateIntegration:
    def integrate_with_surveillance(self, user, camera_id):
        # Інтеграція з системами відеоспостереження
        print(f"Surveillance system integrated for user {user} with camera {camera_id}")

class CloudStorage:
    def __init__(self, storage_path="cloud_storage"):
        self.storage_path = storage_path
        if not os.path.exists(self.storage_path):
            os.makedirs(self.storage_path)
```

```

def save_recording(self, file_name):
    destination = os.path.join(self.storage_path, file_name)
    # Логіка збереження запису конференції у хмару
    print(f"Recording saved to {destination}")

def retrieve_recording(self, file_name):
    path = os.path.join(self.storage_path, file_name)
    if os.path.exists(path):
        print(f"Recording {file_name} retrieved from cloud storage")
        return path
    print(f"Recording {file_name} not found in cloud storage")
    return None

class NetworkQualityMonitor:
    def __init__(self):
        self.quality_data = {}

    def analyze_network(self, conference_id):
        # Логіка аналізу якості підключення
        # Наприклад, перевірка затримок, втрат пакетів
        print(f"Network quality analyzed for conference {conference_id}")

    def optimize_stream(self, video_stream):
        # Логіка оптимізації відеопотоку на основі аналізу якості мережі
        pass

class PollingSystem:
    def __init__(self):
        self.polls = {}

    def create_poll(self, conference_id, poll_question):
        if conference_id not in self.polls:
            self.polls[conference_id] = []
        self.polls[conference_id].append({"question": poll_question, "votes":
{}))
        print(f"Poll created in conference {conference_id}: {poll_question}")

    def vote(self, conference_id, poll_index, user, choice):
        if conference_id in self.polls and 0 <= poll_index <
len(self.polls[conference_id]):
            poll = self.polls[conference_id][poll_index]
            poll["votes"][user] = choice
            print(f"User {user} voted in poll: {choice}")

    def get_results(self, conference_id, poll_index):
        if conference_id in self.polls and 0 <= poll_index <
len(self.polls[conference_id]):
            poll = self.polls[conference_id][poll_index]
            return poll["votes"]
        return {}

class ConferenceReports:

```

```
def generate_report(self, conference_id, participants, duration):
    report = {
        "conference_id": conference_id,
        "participants": participants,
        "duration": duration,
        "timestamp": time.ctime()
    }
    print(f"Report generated for conference {conference_id}")
    return report

class AccessibilitySupport:
    def enable_subtitles(self, video_stream):
        # Логіка підтримки субтитрів у реальному часі
        print("Subtitles enabled for the video stream")

    def enable_sign_language(self, video_stream):
        # Логіка підтримки жестової мови
        print("Sign language support enabled for the video stream")

class AugmentedRealityIntegration:
    def apply_ar_overlay(self, video_frame, ar_elements):
        # Логіка додавання елементів доповненої реальності
        print("AR elements applied to the video frame")

class LiveTranslation:
    def translate_stream(self, audio_stream, source_lang, target_lang):
        # Логіка перекладу аудіопотоку в режимі реального часу
        print(f"Live translation from {source_lang} to {target_lang} enabled")

class NoiseReduction:
    def apply_noise_reduction(self, audio_stream):
        # Логіка зменшення шуму у реальному часі
        print("Noise reduction applied to the audio stream")

class RemoteConferenceControl:
    def __init__(self):
        self.sessions = {}

    def start_remote_session(self, session_id, host_user):
        self.sessions[session_id] = host_user
        print(f"Remote control session {session_id} started by {host_user}")

    def end_remote_session(self, session_id):
        if session_id in self.sessions:
            print(f"Remote control session {session_id} ended")
            del self.sessions[session_id]

class FacialRecognition:
    def authenticate_participant(self, video_frame, user_id):
        # Логіка автентифікації учасників за допомогою біометричних даних
        print(f"User {user_id} authenticated using facial recognition")

class ConferenceNotification:
```

```
def notify_event(self, user, event):
    # Логіка сповіщень про події конференції
    print(f"Notification sent to {user}: {event}")

class VRSupport:
    def enable_vr_mode(self, video_stream):
        # Логіка підключення та налаштування VR-режиму
        print("VR mode enabled for the video stream")

class IntegrationWithLearningPlatforms:
    def sync_with_lms(self, platform_name, course_id):
        # Логіка інтеграції з навчальними платформами
        print(f"Conference synchronized with LMS platform {platform_name},
course {course_id}")
```

КБПЗ_2024

advanced_conference_features.py

```
import json
import os
import time
import cv2

class ARIntegration:
    def apply_augmented_reality(self, video_frame, ar_object):
        # Логіка додавання елементів доповненої реальності на відео
        print("Applying AR object to the video frame")

    def apply_text_overlay(self, video_frame, text):
        # Логіка накладання текстового оверлею на відео
        print(f"Text '{text}' overlay applied to the video frame")

class ConferenceTranslation:
    def translate_audio(self, audio_stream, source_lang, target_lang):
        # Логіка перекладу аудіопотоків у реальному часі
        print(f"Translating audio from {source_lang} to {target_lang}")

    def translate_text(self, message, source_lang, target_lang):
        # Логіка перекладу текстових повідомлень
        print(f"Translating message from {source_lang} to {target_lang}")
        return f"Translated {message}"

class NoiseSuppression:
    def reduce_background_noise(self, audio_stream):
        # Логіка зменшення фонових шумів у аудіопотоці
        print("Background noise reduced")

class ConferenceControl:
    def __init__(self):
        self.active_sessions = {}

    def start_control_session(self, session_id, controller):
        self.active_sessions[session_id] = controller
        print(f"Control session {session_id} started by {controller}")

    def end_control_session(self, session_id):
        if session_id in self.active_sessions:
            print(f"Control session {session_id} ended")
            del self.active_sessions[session_id]

    def adjust_video_quality(self, session_id, quality_level):
        if session_id in self.active_sessions:
            # Логіка зміни якості відео у реальному часі
            print(f"Video quality set to {quality_level} for session {session_id}")

class FacialRecognitionModule:
    def __init__(self):
```

```
self.recognized_faces = {}

def authenticate_user(self, video_frame, user_id):
    # Логіка автентифікації користувачів за допомогою розпізнавання облич
    print(f"User {user_id} authenticated using facial recognition")
    self.recognized_faces[user_id] = True

def check_authentication(self, user_id):
    return self.recognized_faces.get(user_id, False)

class EventNotificationSystem:
    def notify_event(self, user, event_type):
        # Логіка сповіщень користувачів про події конференції
        print(f"User {user} notified about event: {event_type}")

    def schedule_notification(self, user, event_type, delay):
        # Логіка планування відкладених сповіщень
        time.sleep(delay)
        self.notify_event(user, event_type)

class VRModeSupport:
    def enable_vr(self, video_stream):
        # Логіка включення VR-режиму
        print("VR mode enabled for the video stream")

    def adjust_vr_settings(self, video_stream, settings):
        # Налаштування параметрів VR для оптимізації
        print(f"VR settings adjusted: {settings}")

class LMSIntegration:
    def sync_with_learning_platform(self, platform_name, course_id, user_id):
        # Логіка інтеграції з навчальними платформами (Moodle, Blackboard)
        print(f"User {user_id} synchronized with course {course_id} on
{platform_name}")

    def upload_conference_materials(self, platform_name, course_id, file_path):
        # Логіка завантаження матеріалів конференції на навчальну платформу
        if os.path.exists(file_path):
            print(f"Materials uploaded to course {course_id} on
{platform_name}")
        else:
            print("File not found")

class AutoConferenceEnd:
    def set_auto_end_time(self, conference_id, end_time):
        # Логіка автоматичного завершення конференції
        print(f"Conference {conference_id} will end at {end_time}")

    def check_end_time(self, conference_id, current_time):
        # Перевірка поточного часу для завершення конференції
        print(f"Checking end time for conference {conference_id}")

class ConferenceArchiving:
```

```
def __init__(self, archive_dir="conference_archives"):
    self.archive_dir = archive_dir
    if not os.path.exists(self.archive_dir):
        os.makedirs(self.archive_dir)

def archive_conference(self, conference_id, video_file):
    archive_path = os.path.join(self.archive_dir, f"{conference_id}.zip")
    # Логіка архівації відео
    print(f"Conference {conference_id} archived at {archive_path}")

def retrieve_archive(self, conference_id):
    archive_path = os.path.join(self.archive_dir, f"{conference_id}.zip")
    if os.path.exists(archive_path):
        print(f"Archive retrieved for conference {conference_id}")
        return archive_path
    print("Archive not found")
    return None

class KeywordSearch:
    def search_in_conference(self, conference_id, keyword):
        # Логіка пошуку у записах конференцій за ключовими словами
        print(f"Searching for '{keyword}' in conference {conference_id}")

class GraphicOverlay:
    def add_graphics(self, video_stream, graphics_file):
        if os.path.exists(graphics_file):
            # Логіка додавання графіки (діаграм, презентацій) на відеопотік
            print(f"Graphics from {graphics_file} added to video stream")
        else:
            print("Graphics file not found")

class MultiLanguageSupport:
    def __init__(self, supported_languages):
        self.supported_languages = supported_languages

    def switch_language(self, user, language):
        if language in self.supported_languages:
            print(f"Language switched to {language} for user {user}")
        else:
            print(f"Language {language} not supported")

class CloudStorageIntegration:
    def __init__(self):
        self.cloud_services = {}

    def integrate_cloud_service(self, service_name, credentials):
        # Логіка інтеграції з хмарними сервісами (Google Drive, Dropbox)
        self.cloud_services[service_name] = credentials
        print(f"Cloud service {service_name} integrated")

    def upload_to_cloud(self, service_name, file_path):
        if service_name in self.cloud_services:
            # Логіка завантаження файлів у хмару
```

```

        print(f"File {file_path} uploaded to {service_name}")
    else:
        print(f"Cloud service {service_name} not configured")

class APIIntegration:
    def connect_to_external_api(self, api_name, api_key):
        # Логіка підключення до API сторонніх сервісів
        print(f"Connected to external API: {api_name}")

    def retrieve_data(self, api_name, endpoint):
        # Логіка отримання даних з API
        print(f>Data retrieved from {api_name} at endpoint {endpoint}")

class AccessControl:
    def __init__(self):
        self.access_policies = {}

    def set_access_policy(self, user, policy):
        self.access_policies[user] = policy
        print(f"Access policy '{policy}' set for user {user}")

    def check_access(self, user):
        return self.access_policies.get(user, "default")

class AutomaticUpdate:
    def check_for_updates(self):
        # Логіка перевірки наявності оновлень для системи
        print("Checking for updates")

    def perform_update(self):
        # Логіка автоматичного оновлення програмного забезпечення
        print("System update performed")

class BackupAndRecovery:
    def __init__(self, backup_dir="backup_data"):
        self.backup_dir = backup_dir
        if not os.path.exists(self.backup_dir):
            os.makedirs(self.backup_dir)

    def backup_conference_data(self, conference_id, data):
        backup_path = os.path.join(self.backup_dir,
f"{conference_id}_backup.json")
        with open(backup_path, "w") as f:
            json.dump(data, f, indent=4)
        print(f"Conference data for {conference_id} backed up at {backup_path}")

    def restore_conference_data(self, conference_id):
        backup_path = os.path.join(self.backup_dir,
f"{conference_id}_backup.json")
        if os.path.exists(backup_path):
            with open(backup_path, "r") as f:
                data = json.load(f)
            print(f"Conference data for {conference_id} restored")

```

```
    return data
print("Backup file not found")
return None
```

К6П3_2024