

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
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
віртуалізованого відеоспостереження”

КБПЗ - 2024

Виконав здобувач вищої освіти
II курсу, групи КІ-23Мз
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Мороз А.С.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Центр *Заочної та дистанційної освіти*
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

Морозу Андрію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи	<i>Дослідження та програмна реалізація системи віртуалізованого відеоспостереження</i>
2. Керівник роботи	<i>Смірнова Тетяна Віталіївна, канд. техн. наук</i> (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 21-13 від 07.08.2024 року	
3. Строк подання студентом роботи до захисту	<i>2.12.2024 р.</i>
4. Мета та завдання випускної кваліфікаційної роботи:	<i>Метою розробки є дослідження та програмна реалізація системи віртуалізованого відеоспостереження</i>
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 проєктних 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

Мороз А.С. Дослідження та програмна реалізація системи віртуалізованого відеоспостереження. 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 віртуалізованого відеоспостереження.

Метою розробки є дослідження та програмна реалізація системи віртуалізованого відеоспостереження.

Об'єктом дослідження є процес віртуалізованого відеоспостереження.

Предметом дослідження є методи віртуалізованого відеоспостереження.

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

Результат роботи – програмна реалізація системи віртуалізованого відеоспостереження.

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, відеоспостереження

ABSTRACT

Moroz A.S. Research and software implementation of the virtual video surveillance system. 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 virtual video surveillance system.

The goal of the development is the research and software implementation of the virtualized video surveillance system.

The object of research is the process of virtualized video surveillance.

The subject of research is the methods of virtualized video surveillance.

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

The result of the work is the software implementation of the virtualized video surveillance system.

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 Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer engineering, video surveillance

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	16
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми	30
3.4 Розробка діаграми процесів.....	33
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	35
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	35
4.2 Захист розробленого програмного забезпечення.....	48
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	50
6 НАУКОВА НОВИЗНА	55

					ВКРМ-123.24.0001.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи віртуалізованого відеоспостереження	Літ.	Аркуш	Аркушів
Розроб.	Мороз А.С.					М	1	82
Перев.	Смірнова Т.В.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23Мз		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	56
7.1	Визначення цільової аудиторії кінцевого готового продукту	56
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	57
7.3	Вибір методу оцінки вартості ПЗ	59
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	60
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	62
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	63
7.7	Визначення ключових факторів успіху конкретного проєкту.....	64
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	65
8.1	Вступ.....	65
8.2	Пожежна безпека	67
8.3	Пропозиції щодо підвищення працездатності ІТ-фахівців.....	68
8.4	Розробка заходів з умов поліпшення охорони праці	70
8.5	Розрахункова частина	71
9	ОСНОВНІ ВИСНОВКИ.....	74
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76

КБПЗ-2024

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БПД	–	бездротова передача даних
ПЗ	–	програмне забезпечення
СЗД	–	системи зберігання даних
СПД	–	системи передачі даних
ACK	–	повідомлення підтвердження прийому
ARQ	–	протокол повторної передачі даних
BPSK	–	Binary phase-shift keying
FFD	–	повнофункціональний пристрій
GFSK	–	Gaussian frequency-shift keying
MAC	–	шар механізму доступу
NACK	–	повідомлення непідтвердження прийому
OSI	–	мережна модель
P2P	–	однорангові мережі
PAN	–	персональна мережа
PPS	–	Portable Protocol Stack
RFD	–	пристрій з полегшеними функціями
TDMA	–	часовий поділ
Wi-Fi	–	бездротова технологія

ВСТУП

Актуальність теми. Віртуалізація є важливою частиною ІТ-індустрії вже майже 20 років. Дивно, але багато хто в галузі не знайомі з технологією, її призначенням або тим, як вона пов'язана з постійно зростаючим ринком ІР-відеоспостереження. Перш ніж ми глибше зануримося у віртуалізацію та її роль у індустрії безпеки/ІР-відео, нам спочатку потрібно зрозуміти, що це таке, а потім ми зможемо переглянути, як це працює.

Простіше кажучи, віртуалізація – це обчислювальна технологія, яка створює віртуальне обладнання для функціонального запуску системних служб на основі програмного забезпечення, таких як програми, сервери обміну файлами, сховища, мережі та багато іншого. Створення віртуальних екземплярів цих служб і операційних систем дає змогу максимально використати ваші апаратні ресурси в повній мірі.

Віртуальні машини створюються та встановлюються на програмне забезпечення, яке називається гіпервізорами. Можливо, ви вже чули про деякі з найбільш популярних гіпервізорів у галузі, таких як VMware ESXi, Microsoft Hyper-V або Citrix XenServer. Є два основних типи гіпервізорів: голі або розміщені. Обидва дозволяють створювати кілька віртуальних машин в одній фізичній системі, яка також відома як хост. Якщо у вас є кілька хостів в одному середовищі, це називається кластером, і це надає можливість групувати всі комп'ютерні ресурси хоста в один гігантський пул, який можна ефективно та ефективно розподіляти. Після встановлення гіпервізор розподіляє ресурси цим віртуальним машинам, щоб вони могли працювати з максимальною ефективністю. Ці параметри також можна налаштувати вручну для кожного екземпляра віртуальної машини, що можна виконати, не виводячи пристрої в автономний режим. Наявність такого типу еластичного розподілу ресурсів, а також можливість запускати кілька систем на одній фізичній машині робить ІТ-

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

операції набагато ефективнішими та знижує витрати на капітальні та операційні витрати, звільняючи простір у стійці та зменшуючи споживання електроенергії та нагрівання. і витрати на охолодження.

Оскільки вимоги до пропускнуої здатності камер високої роздільної здатності постійно зростають, завжди буде попит на можливість швидкого та ефективного додавання додаткових ресурсів до системи. Завдяки універсальності та гнучкості «на льоту», які досягаються завдяки використанню та розгортанню віртуальних машин, ринок відеоспостереження відчуває більший попит на такі типи розгортань. Розробка рішення системи відеоспостереження з віртуалізованим середовищем ще більше покращує масштабованість і можливості для архітектури високої доступності, яка інакше була б неможлива з попередньо встановленою системою фізичного сервера.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи віртуалізованого відеоспостереження.

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

- Огляд існуючих систем віртуалізованого відеоспостереження.
- Дослідження системи віртуалізованого відеоспостереження.
- Програмна реалізація системи віртуалізованого відеоспостереження.

Об'єктом дослідження є процес віртуалізованого відеоспостереження.

Предметом дослідження є методи віртуалізованого відеоспостереження.

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

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

- Удосконалено метод віртуалізованого відеоспостереження.
- Розроблено вітчизняний продукт віртуалізованого відеоспостереження, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віртуалізованого відеоспостереження.

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 віртуалізованого відеоспостереження, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					VKPM-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Наявність можливості віртуалізації фізичного середовища безпеки вашого клієнта дозволяє йому консолідувати своє апаратне забезпечення для відео, контролю доступу та інших різноманітних платформ безпеки та об'єднати їх усі в одне кероване середовище. Віртуалізація також надає можливість керувати та виконувати технічне обслуговування всіх ваших серверів шляхом входу в один інтерфейс гіпервізора замість входу на 10 або 15 різних серверів.

Віртуалізація вашого ІТ-середовища дає вам і вашому клієнту можливість отримати вигоду з ефективності ресурсів. У індустрії відеоспостереження майже ніколи не використовується повний потенціал навантаження ЦП на голий металевий пристрій; тоді як пам'ять того самого пристрою майже миттєво вичерпується через постійне збільшення часу зберігання. Завдяки віртуалізації у вас тепер є операційний контроль, щоб точно визначити, скільки обчислювальних ресурсів і ресурсів зберігання віртуальна машина отримує від ЦП, оперативної пам'яті, розміру жорсткого диска, мережевих адаптерів і відеокарт. Таким чином, надання кінцевому користувачеві ще більших можливостей і контролю над своїм середовищем, ніж сценарій встановлення на «голому металі», не обов'язково дозволяє.

Віртуалізація має свої недоліки, і не всі проекти відеоспостереження можуть отримати користь від встановлення віртуальної системи. Вищі початкові витрати на впровадження, обмеження пропускної здатності, можливі проблеми сумісності та безпеки даних – це деякі з причин, які можуть змусити клієнтів піти шляхом віртуального середовища. З огляду на таку кількість факторів, вибір може здатися величезним. Навчання основам обох може стати ключем до вибору відповідного варіанту для вашого проекту.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Ринок відеоспостереження перспективний, що розвивається, хоча в поточній економічній ситуації робити які-небудь прогнози про його подальшу долю в нашій країні складно. Серед факторів росту виділимо вимоги, що підсилюються, до безпеки, інвестиції в інфраструктуру, перехід від аналогових систем до ІР-відеоспостереження, зміна законодавчих і регулюючих норм. Розширюються сфери застосування ІР-камер: вони всі частіше використовуються не тільки для спостереження, але й для попередження інцидентів і їхнього розслідування. Обмежуючим фактором залишаються оптичні системи, які еволюціонують далеко не так швидко, як процесори й ПЗ.

Вартість устаткування для відеоспостереження (у доларовому вираженні) буде поступово знижуватися, а його частка в загальному обсязі ринку відеоспостереження скорочуватися. Разом з тим через складність, що збільшується, завдань, що стоять перед відеоспостереженням, підвищується попит на ПЗ для відеоспостереження й відеоаналітики. Світовий ринок послуг відеоспостереження й відеоаналітики – самий швидкозростаючий.

1.2 Область застосування

У нашій країні основними користувачами технологій відеоспостереження й відеоаналітики є організації з державного сектора, банки, підприємства транспорту й роздрібною торгівлі, керування нерухомістю. У світі спостерігається схожа картина. Разом з тим український ринок відеоспостереження слабо структурований і перебуває в стадії формування. Найбільшу активність проявляють численні дрібні гравці, що пропонують локальні рішення на базі аналогових камер.

До числа основних тенденцій ринку відеоспостереження віднесемо наступні:

– ріст якості зображення при одночасному зниженні вартості камер (за прогнозами IMS Research, до 2026 року більше 70% всіх поставок мережевих камер буде доводитися на камери з мегапіксельним розрішенням);

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- збільшення попиту на хмарні системи відеоспостереження (за прогнозами, сегмент VSaaS буде рости більш ніж на 20% щорічно);
- «інтелектуальне» відеоспостереження й розвиток можливостей убудованої й серверної відеоаналітики;
- спрощення користувальницького керування системою відеоспостереження (інтуїтивно зрозумілі користувальницькі інтерфейси).

Активний розвиток одержать додатки для мобільних платформ, а збір відеоінформації всі частіше буде здійснюватися за допомогою бездротових камер і каналів LTE.

У числі важливих технологічних факторів – збільшення продуктивності процесорів IP-камер, оснащення останніх картами пам'яті великої ємності, ріст попиту на камери високого розрішення (HD), поява нових продуктів з розрішенням 4/8K. Все більшою популярністю користуються тепловізори, камери кругового огляду й з автофокусом, а застосування спеціальних технологій і інфрачервоного підсвічування дозволяє одержати інформативне зображення в умовах поганої освітленості.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи віртуалізованого відеоспостереження, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Відеореєстратори з великою глибиною архіву Avigard NVR5416, NVR5816 і NVR5832

Якщо Ви плануєте створити професійну систему відеоспостереження у своєму офісі або на підприємстві, то Вам ніяк не обійтися без професійного відео встаткування. Як вибрати дійсно якісне встаткування й на що звертати увагу в першу чергу ми розглянемо детальніше.

Плануючи й установлюючи системи відеоспостереження звичайно обертають більше уваги на відеокамери, приділяючи зовсім небагато часу вибору відеореєстраторів. Такий підхід – у корені невірний. Відеореєстратор – основа, стовп на якому коштує ваша система відеоспостереження. Від його параметрів і характеристик прямо залежить її надійність і функціональність.

Перед установкою системи відеоспостереження необхідно розуміти які завдання вона повинна виконувати. Якщо ви встановлюєте невелику систему на кілька відеокамер для спостереження, наприклад, за паркуванням, то вам підійдуть недорогі рішення, наприклад: кілька камер IPC-HDW4100S і відеореєстратор Avigard AVG-NVR2208. Вони відмінно впораються з покладеними на них завданнями. Однак, якщо ви плануєте встановити професійну систему відеоспостереження, то до підбора встаткування необхідно підходити набагато серйозніше. Для забезпечення безперебійного якісного відеоспостереження на великій території виробництва або офісів потрібно здобувати потужні відеореєстратори з підтримкою великої кількості відеокамер і передових технологій у відеоспостереженні.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Вони мають міцний металевий корпус зі спеціально передбаченою можливістю установки відеореєстратора в серверну стійку. Така установка допоможе Вам убезпечити свої відеореєстратори від можливих фізичних впливів, забезпечити їм поліпшене охолодження й циркуляцію повітря й уникнути плутанини проводів, що неминуче виникне при установці серйозної системи відеоспостереження. Подібна плутанина сильно ускладнює роботу з відеореєстратором і навіть може привести до його ушкоджень і виходу з ладу, тому що за проведення легко чим-небудь зачепитися й потягнути їх за собою. У найкращому разі це ушкодить самі проведення, у гіршому ушкодить рознімання реєстратора або навіть виведе його з ладу.

У даних відеореєстраторів на передній панелі присутня повноцінне кнопочне меню, за допомогою якого можна управляти всіма його функціями. Також на передній панелі передбачене місце для установки DVD RW привода. Установивши його ви зможете легко й швидко записати будь-який відеофайл на диск, якщо буде потреба передачі його кому або.

Значно більше надійними дані моделі робить убудований блок живлення. Він запобігає ушкодженням при перепадах напруги й забезпечує надійну роботу. Що б зрозуміти наочно про збільшення надійності даної лінійки відеореєстраторів, досить заглянути до них під корпус. У порівнянні з іншими реєстраторами в NVR 5416, NVR 5816 і NVR 5832 материнська плата у два рази більше по розмірі. Це через те, що розроблювачі вмонтували на плату додаткові системи охолодження й електронних складових для більшої надійності й відказостійкості.

Для професійної системи відеоспостереження великий архів відеоданих у високому розрішенні – нагальна потреба. В Avigard NVR 5416, NVR 5816 і NVR 5832 для цього реалізована підтримка до 4 жорстких дисків E SATA по 4 терабайта. Також можливо підключити до 4 зовнішніх жорстких дисків по 4 терабайта кожного. При такому обсязі пам'яті створити великий архів відеоданих не складе для Вас ніяких проблем.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Одне з найважливіших питань для системи відеоспостереження є надійність зберігання архіву відеоданих і захист його від ушкоджень і втрати. У даній лінійці відеореєстраторів є відповідь і на нього.

Для захисту від втрати даних реалізований режим Raid1 – дублювання відеоданих на другий жорсткий диск у реальному часі. Запис із ваших камер відеоспостереження буде автоматично дублюватися в реальному часі й у випадку форс мажору у Вас буде наготові копія всіх відеоматеріалів.

І для завершення розроблювачі встановили 2 RJ45 порти й два мережевих контролери для значного збільшення пропускної здатності відеореєстратора. І якщо у випадку непередбачених обставин у Вас ушкодиться одне рознімання, завжди є другий, дублюючий.

Завдяки всім перерахованим вище складовим Avigard NVR 5416, NVR 5816 і NVR 5832 стануть надійною основою для самих професійних і більших систем відеоспостереження. Здобуваючи ці відеореєстратори Ви можете бути впевненими в тім, що вони впораються з усіма покладеними на них завданнями й забезпечать якісну, надійну, і безперебійну роботу Вашої системи відеоспостереження.

Модуль відеокамер Avigard серії HD

Всі елементи модуля мають дуже акуратний вид, пайка й кріплення дивують свої добірністю, компактністю й продуманим розташуванням на платі.

І, звичайно ж, одним із самих головних елементів модуля, без якого він був би просто шматком «мертвої» плати є процесор, що обробляє відеосигнал і керівна вся робота камери, тобто, по суті, її мозок.

Avigard використовує в камерах з індексом HD винятково процесори компанії RJ Electronics – одного зі світових лідерів по виробництву й поставкам високотехнологічних апаратних компонентів для систем безпеки.

RJ Electronics EOS RJ-8 – саме цей 76-піновий процесор використовується у всій лінійці камер Avigard з індексом HD. Основна його перевага – підтримка найвищого розрішення в 700 телевізійних ліній (поки максимально можливого

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

розрішення для аналогового відеоспостереження) і найширшого цифрового динамічного діапазону, що забезпечує відмінні показники зйомки в умовах слабкого висвітлення. Крім цього, процесор оснащений чотирьохзоною компенсацією від засвітлення яскравого світла, системою цифрового шумозаглушення, 8-зонною полігональною маскою конфіденційності, функцією дзеркального відображення й набудовується функцією, що, компенсації дефектів зображення. Взагалі про всі характеристики й функції процесора можна говорити годинами, єдине, на чому варто особливо заострити свою увагу – оптимальне співвідношення ціни й технічних характеристик.

Якщо ми заговорили про найважливіший складовий модуль відеокамер Avigard і згадали про «мозок», то неодмінно потрібно зупинитися й на «очах», а саме на відеоматриці. У цьому модельному ряді використовується матриця SONY з технологією EXview HAD CCD II. Про це говорить маркування самої матриці – ICX673AKA.

Відразу необхідно зробити невеликий відступ і сказати про те, що якщо Avigard заявляє про використання матриць SONY у своєму встаткуванні, те так воно і є. Справа в тому, що в цей час вітчизняні торговельні марки, та й просто продавці люблять «прикрашувати» технічні характеристики встаткування, «підмінюючи» матриці SHARP на SONY або просто додаючи зайву сотню телевізійних ліній до розрішення відеокамери. Перевірити це практично не представляється можливим, тому що маркування наноситься на зворотну сторону матриці й для перевірки доводиться розпаювати її, ну а про перевірку кількості телевізійних ліній і зовсім говорити не доводиться – без спеціальної вимірювальної техніки кінцевий клієнт не зможе відрізнити 540 ТВЛ від 600, а 600 ТВЛ від 700. Особливо, це стосується тих випадків, коли до моменту покупки й установки, клієнт не мав справи з якісним устаткуванням для відеоспостереження.

Але повернемося до матриці. Маркування SONY EXview HAD CCD II говорить про загальну високу продуктивність і світлочутливість сенсора. Він

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

проектувався спеціально під роботу з високим розрешенням і використовується в дорогому встаткуванні у зв'язуванні із процесорами Effio, також виробництва SONY. Однак, як показує практика, із процесорами сторонніх виробників матриця працює не гірше й показує відмінні результати.

Конкретніше про технічні характеристики: діагональ самого сенсора становить 6 мм (1/3"). Загальний дозвіл становить 1020 x 596 пікселів, а ефективний дозвіл – 976 x 582 пікселів або 700 телевізійних ліній по горизонталі, при цьому розмір самого пікселя – 5.0 x 6.25 мкм.

Спільне використання високоякісного процесора RJ Electronics і матриці SONY говорить про те, що союз дійсно вийшов удалим і практичним. Відносно невисока вартість камер Avigard і простота керування й налаштування говорить про те, що виробник дійсно піклується не тільки й не стільки про професійних монтажників, як про свого кінцевого клієнта, неперевершена якість зображення й відеозапису дозволить запам'ятати будь-які нюанси, щоб забезпечити тотальну безпеку.

Виробники камер для відеоспостереження, орієнтуючись на низький і середній ціновий діапазон, віддають переваги не сучасним технологіям, а використовують добре перевірені часом комплектуючі. З однієї сторони це звичайно плюс, але при цьому розрив поколінь між недорогими й дуже дорогими камерами, які оснащені якісними компонентами й передовими технологіями може бути дуже великий. В Avigard вирішили не йти цим шляхом, а спробувати використовувати дорогі і якісні компоненти в досить доступному сегменті, у зв'язку із чим мають дуже більшу популярність, а поставки цих камер закінчуються з величезною швидкістю, не залежуючись на складах і не піддаючись моральному старінню.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об'єктноорієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

- Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.
- Об'єктноорієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.
- Високий рівень: зручний для людини, а не для комп'ютера.
- Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веброзробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

потрібно витратити багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов'язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок, тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати,

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

- Фахівець з даних.
- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веброзробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

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 віртуалізованого відеоспостереження.

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

негативних якостей. Результати аналізу врахувати в подальших розробках;

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

В даний час віртуальні машини масово використовуються окремими особами та компаніями по всьому світу з різних причин і в різних сценаріях, включаючи, але не обмежуючись установками відеоспостереження. Якщо ви ще не чули про віртуальні машини, наступний розділ допоможе вам зрозуміти, що таке віртуальна машина, і чому мільйони користувачів і компаній уже користуються перевагами технології.

Віртуальна машина (яка також скорочено називається віртуальною машиною) – це емуляція (або віртуалізація) комп'ютерної системи, яка існує на фізичній машині, використовує її ресурси та апаратне забезпечення та забезпечує функціональність справжнього комп'ютера. На одній фізичній машині може бути кілька таких «дублікатів», які працюють від одного чи кількох програм гіпервізора, ізольованих і працюючих незалежно один від одного. Ви коли-небудь орендували хмарний сервер? Швидше за все, це віртуальна машина всередині фізичної машини, яка знаходиться десь у центрі обробки даних.

Технологія виникла на основі принципу *розподілу часу*, введеного в 1960-х роках, який дозволяв кільком користувачам використовувати комп'ютер одночасно, що пояснює одну з цілей використання віртуальної машини – *спільне використання ресурсів* для виконання різних завдань всередині однієї машини. Ця мета зазвичай є основною, коли віртуальна машина використовується в корпоративних системах.

Приватні особи, як правило, використовують віртуальні машини переважно з міркувань безпеки, завдяки *ізолюваній роботі* віртуальних машин.

Існує третя причина використання віртуальних машин, яка зараз може представляти великий інтерес як для приватного, так і для бізнес-сектору –

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

віртуальні машини дозволяють емулювати чужі операційні системи всередині іншої. Наприклад, коли деякі основні галузеві комп'ютерні програми працюють лише в ОС Windows, а робочими машинами є комп'ютери Mac.

Docker – це тип реалізації віртуальної машини на рівні операційної системи, що дозволяє працювати декільком ізольованим і безпечним віртуалізованим серверам на одному фізичному сервері. Docker використовує програмне забезпечення, що постачається в *контейнерах* – ізольовані комплекти програмного забезпечення, конфігураційних файлів і необхідних бібліотек – які обмінюються даними один з одним через певні визначені канали. Зазвичай докери використовують менше системних ресурсів, ніж звичайні віртуальні машини.

Якщо віртуальні машини настільки ефективні, ізольовані та безпечні речі, то чому не всі ними користуються?

Як ми встановили вище, віртуальні машини та докери не є *реальними* машинами, вони є емуляцією одного, середовища. Це збиває з пантелику багато програм, яким потрібно отримати інформацію про машину, на якій вони працюють, або використовувати системні ресурси, до яких вони не мають доступу. Наприклад, якщо програму розроблено для роботи із зображеннями веб-камери, вона може не робити того, що повинна робити у віртуальній машині, яка не має прав доступу до USB-пристроїв.

Крім того, на деяких слабших комп'ютерах запуск віртуальних машин може інтенсивно навантажувати ЦП, тому, мабуть, не буде чудовою ідеєю створювати віртуальну машину для запуску топових відеоредакторів, сумно відомих своїми голодними вимогами.

Що стосується докерів, то не кожна програма підтримує роботу в одній, оскільки їй структурно може знадобитися доступ до ресурсів, бібліотек, файлів, які не входять до їх власного контейнера. У такому випадку він може чудово працювати на реальному комп'ютері та не працювати належним чином у контейнері – і не кожен виробник витрачає дорогоцінні ресурси та час на

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

розробку обхідного шляху для Dockers, якщо це не те, що становить суттєву частку їх ринку.

Саме тому у даній роботі розробляється система для відеоспостереження, яка можуть працювати як у віртуальному середовищі, так і на фізичних машинах.

3.2 Розробка структурної схеми

Система віртуалізованого відеоспостереження – віртуальний сервер відеоспостереження. Його можна встановити на комп'ютері з Windows, Mac або Linux на всіх стандартних платформах віртуалізації, таких як VMware, Hyper-V, KVM або Virtualbox, і працювати постійно. Одна або кілька віртуальних машин системи віртуалізованого відеоспостереження встановлено на одному або кількох серверах віртуалізації:

Система віртуалізованого відеоспостереження легко масштабується завдяки технології віртуалізації.

Усі переваги віртуалізації можна використати за допомогою системи віртуалізованого відеоспостереження, яка розробляється в даній роботі. До них належать:

- Безкоштовний розподіл ядер ЦП за потреби.
- Збільшити дисковий простір.
- Збільште обсяг оперативної пам'яті.
- Просте використання VLAN (віртуальних локальних мереж) для відділення мережевого трафіку від відеоспостереження.
- Продовження ліцензій або додавання ліцензій на камеру.
- Побудова інфраструктури до 90 камер з кількома віртуальними машинами системи віртуалізованого відеоспостереження, яка розробляється в даній роботі.
- Встановлення в існуючих середовищах віртуалізації, тому не потрібно купувати обладнання

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

– Переміщення віртуальної машини системи віртуалізованого відеоспостереження на інші фізичні сервери віртуалізації, також для балансування навантаження

Встановіть систему віртуалізованого відеоспостереження на існуючу ІТ-інфраструктуру, платформу віртуалізації VMware, Hyper-V або KVM. Тому системі віртуалізованого відеоспостереження не потрібно купувати новий сервер. У великій платформі системи віртуалізованого відеоспостереження просто встановлюється. Для цього можна розгорнути попередньо налаштовані образи для VMware або Hyper-V або встановити систему віртуалізованого відеоспостереження за допомогою образу ISO. Монтаж системи відеоспостереження системи віртуалізованого відеоспостереження виконується в найкоротші терміни.

Система віртуалізованого відеоспостереження підтримує засоби керування інтерфейсом віртуалізації. Наприклад, системою віртуалізованого відеоспостереження можна керувати за допомогою інструментів VMware у VMware vCenter для завантаження або автоматичного вимкнення та перезапуску у вікні обслуговування vCenter. Підтримка адміністративних інструментів також працює в Hyper-V і KVM або VirtualBox. Також підтримуються менші середовища віртуалізації

Система віртуалізованого відеоспостереження встановлюється не лише у великих середовищах ІТ-інфраструктури, таких як кластери VMware, Hyper-V або KVM. Систему віртуалізованого відеоспостереження також можна легко встановити на окремих ПК або Mac. Систему віртуалізованого відеоспостереження можна встановити на ПК з Windows 10 під керуванням Hyper-V або на Mac із VirtualBox. Подібним чином ПК з ОС Linux можна використовувати для встановлення системи віртуалізованого відеоспостереження під VirtualBox або KVM і керування ним. Завдяки попередньо створеним інсталяційним образам інсталяція у VirtualBox на ПК або Mac завершується

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

всього за кілька хвилин. Просто двічі клацніть файл OVA, який можна безкоштовно завантажити, щоб зручно, легко та швидко встановити у VirtualBox.

Систему віртуалізованого відеоспостереження оптимізовано для браузера Firefox, але також може використовуватися з іншими браузерами. Браузер Safari на iPhone і Chrome на Android також використовуються на мобільних пристроях.

Клієнти системи віртуалізованого відеоспостереження користуються багатьма перевагами віртуалізації. Система віртуалізованого відеоспостереження є масштабованою. Потужність процесора, простір на жорсткому диску та оперативну пам'ять можна розподілити за потреби. Систему віртуалізованого відеоспостереження можна перемістити на інший апаратний сервер або навіть працювати в надлишковому кластері. Систему віртуалізованого відеоспостереження також можна легко захистити через Veeam. I: Ви можете розширити камери за потреби та почати свій відеопроєкт з малого.

Система віртуалізованого відеоспостереження повністю управляється через Інтернет. Окрім великого веб-сайту для настільних комп'ютерів і планшетів, мобільні пристрої, такі як iPhone або телефони Android, можуть використовувати зрозумілі мобільні сторінки. Тривога, електронні листи з відео тривоги, альтернативно SMS-тривога, є однією зі стандартних функцій відеоспостереження. Тут розробники надають великого значення тому, що відео можна відтворювати без встановлення додаткового програмного забезпечення на звичайних пристроях, таких як Windows, Mac, Linux, iPhone, Android.

Система віртуалізованого відеоспостереження робить знімки на основі виявлення руху на зображенні. У веб-інтерфейсі можна встановити багато параметрів, які викликають тривогу залежно від налаштувань. За допомогою масок зображень можна оптимізувати виявлення руху. Порогові значення, які також встановлюються через веб-інтерфейс, зрештою запускають відеозаписи та відповідні електронні листи тривоги. Крім того, можна інтегрувати зовнішні детектори руху, щоб відеосигнали спрацьовували не через зміни пікселів у відеозображенні, а, наприклад, через світлові бар'єри або ультразвукові

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

детектори руху, які вбудовані в деякі моделі камер. Будильники також можна налаштувати за допомогою часових таблиць. Тож ви отримуєте відеосигнали лише в тому випадку, якщо це актуально для вас! Також можна вільно визначати свята або свята, щоб надати їм власний профіль часу для відеозаписів і відеосигналів. На додаток до сповіщень електронною поштою з вкладеним відео, SMS-повідомлення можна надсилати на мобільний телефон одночасно, щоб ви отримували сповіщення про виявлення руху та спрацьовування сповіщення.

У системі віртуалізованого відеоспостереження ви можете надати кожному користувачу права на камери, записи та параметри налаштувань. Це робить можливим, наприклад, що користувач, який отримує лише повільніше представлення прямого ефіру через мобільний телефонний зв'язок, все ще може отримати значущий доступ до системи віртуалізованого відеоспостереження, не зупиняючи відео. Система віртуалізованого відеоспостереження пропонує прямі трансляції за допомогою технології, яка працює у веб-браузерах на комп'ютерах, а також на iPhone та смартфонах Android.

Залежно від програми, система віртуалізованого відеоспостереження може включати потоки MJPEG з високою якістю, але високою пропускну здатністю, або відео MPEG4 з меншою пропускну здатністю та нижчою якістю. Користувач сам вирішує, яку поточкову технологію використовувати в залежності від своїх вимог і моделі камери.

У системі віртуалізованого відеоспостереження база даних камер керується через відомі моделі камер, які можна вибрати та інтегрувати. База даних постійно поповнюється. Крім того, користувачі системи віртуалізованого відеоспостереження мають можливість інтегрувати вільно визначені камери. Немає жодних перешкод для інтеграції камер, якщо ви використовуєте загальні стандарти.

Важливою особливістю при використанні системи віртуалізованого відеоспостереження є підключення систем сигналізації. Систему віртуалізованого відеоспостереження можна підключити до систем сигналізації. У цьому випадку

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

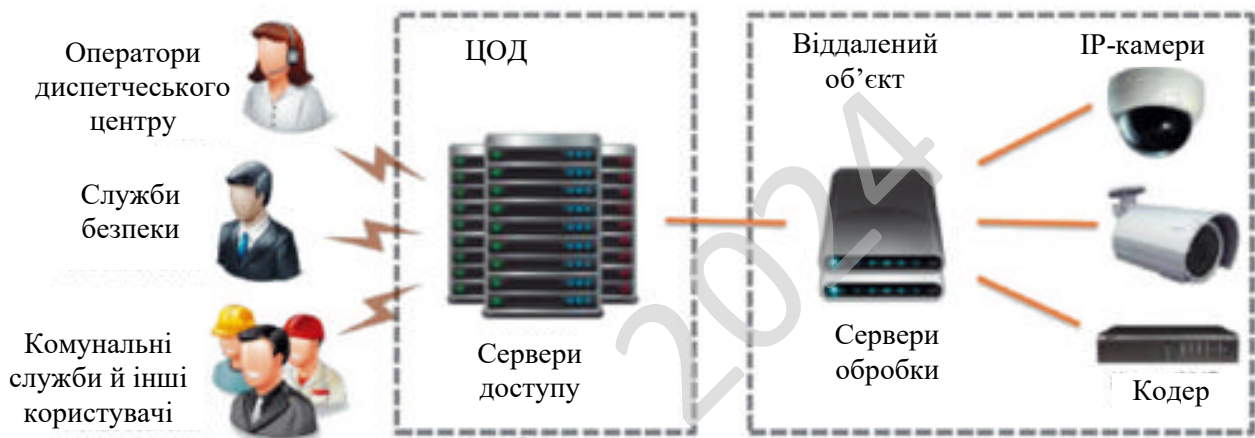
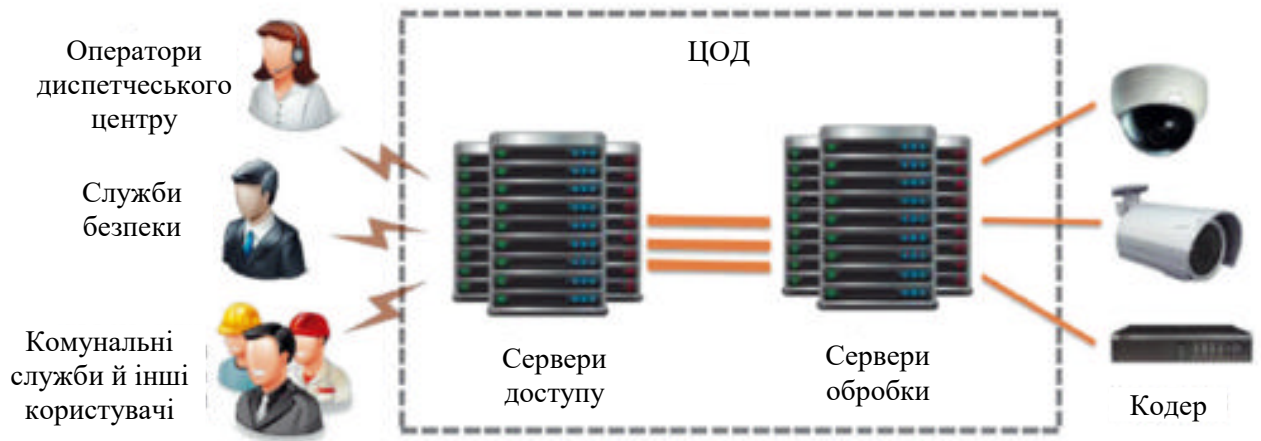


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Опишемо функції розробленої системи відеоспостереження, побудованої на базі розроблених алгоритмів спільного калібрування камер і виділення областей інтересу.

Функціональна схема системи складається з наступних блоків:

- Блок функцій оператора віртуалізованої системи відеоспостереження.
- Блок функцій інженера віртуалізованої системи відеоспостереження.
- Блок функцій адміністратора віртуалізованої системи відеоспостереження.

Блок функцій оператора віртуалізованої системи відеоспостереження

Блок функцій оператора віртуалізованої системи відеоспостереження включає в себе наступні функції:

1. Перегляд відео з камер.
2. Оцінка ситуації по плану:
 - Відображення областей спостереження камер.
 - Відображення доданих об'єктів.
 - Відображення знайдених об'єктів.
3. Керування камерами:
 - Клік по відео.
 - Обертання камери.
 - Клік по карті.
 - Режим супроводу.
4. Зміна налаштувань системи.

Блок функцій інженера віртуалізованої системи відеоспостереження

Блок функцій інженера віртуалізованої системи відеоспостереження включає в себе наступні функції:

1. Завантаження плану контролюемого простору.
2. Додавання заборонених областей на план.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30



Рисунок 3.2 – Функціональна схема системи

3. Додавання об'єктів на план.
4. Калібрування камер.
5. Додавання та видалення камер.
6. Калібрування плану.

Блок функцій адміністратора віртуалізованої системи відеоспостереження

Блок функцій адміністратора віртуалізованої системи відеоспостереження включає в себе наступні функції:

1. Додавання нових користувачів.

Система дозволяє не тільки контролювати обстановку по відеоряду, але й відслідковувати всі зміни на двомірному плані охоронюваного простору. На плані відображаються області видимості камер і виявлені об'єкти, що полегшує сприйняття загальної обстановки.

Так, наприклад, на плані легко помітити області контрольованого простору не видимі ні однією відеокамерою.

Крім того система надає користувальницький інтерфейс для керування камерами за допомогою кліків миші на компоненті головного вікна, із зображенням плану.

Система відеоспостереження має розвитий графічний інтерфейс і істотну обчислювальну частину.

Система має більш багату функціональність ніж існуючі комерційні розробки. Використання поворотних камер, розмаїтість можливостей по керуванню камерами й спостереженню за обстановкою дозволяє ефективніше вирішувати завдання відеоспостереження, ніж при використанні традиційних систем відеоспостереження з нерухливими камерами.

Таких можливостей удалося досягти завдяки використанню розроблених методів калібрування камер і виявлення областей інтересу.

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

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.

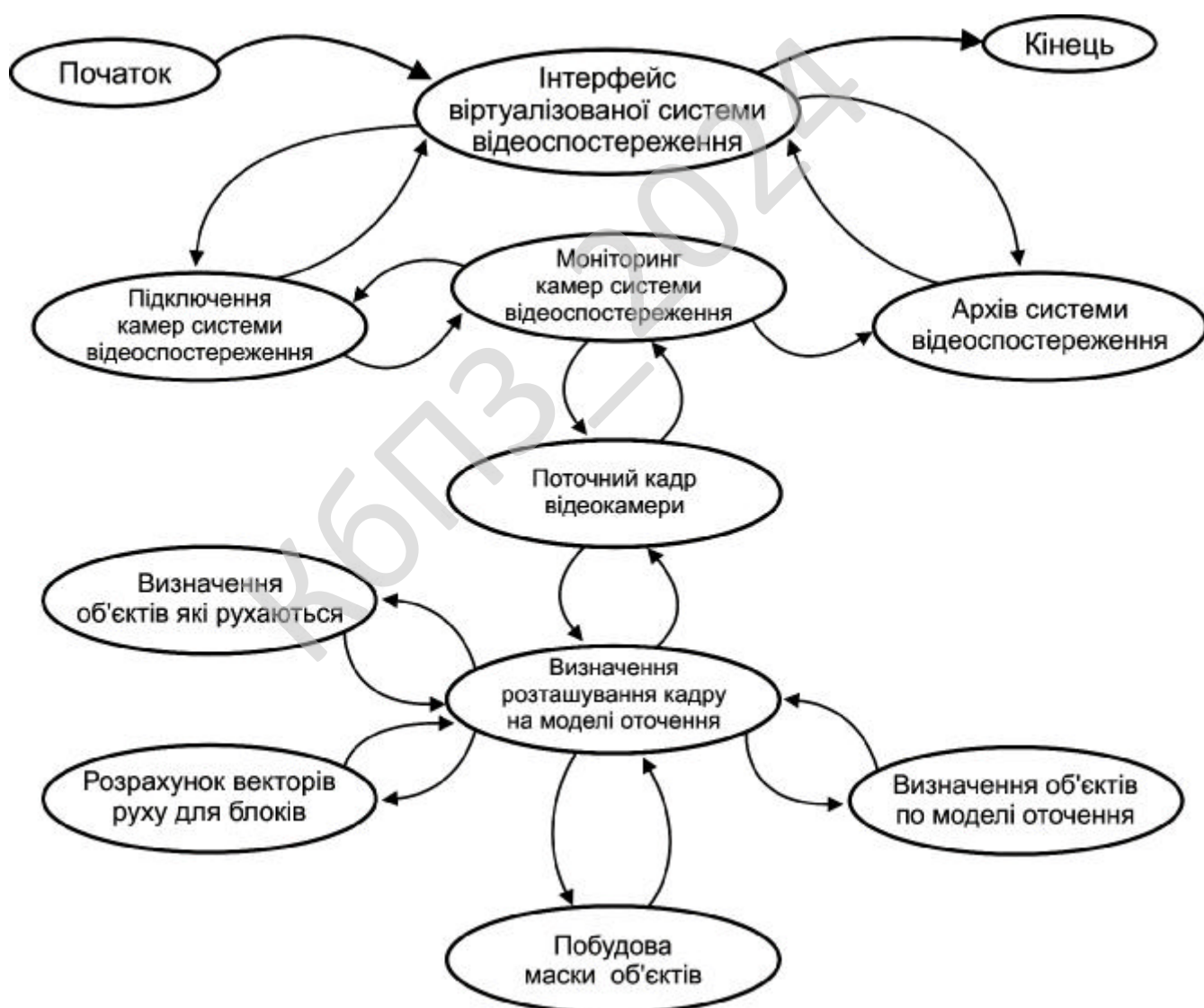


Рисунок 3.3 – Діаграма взаємодії процесів

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

Діаграми потоків даних містять чотири типи елементів:

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

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ – 2024

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

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 віртуалізованої системи відеоспостереження.

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-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

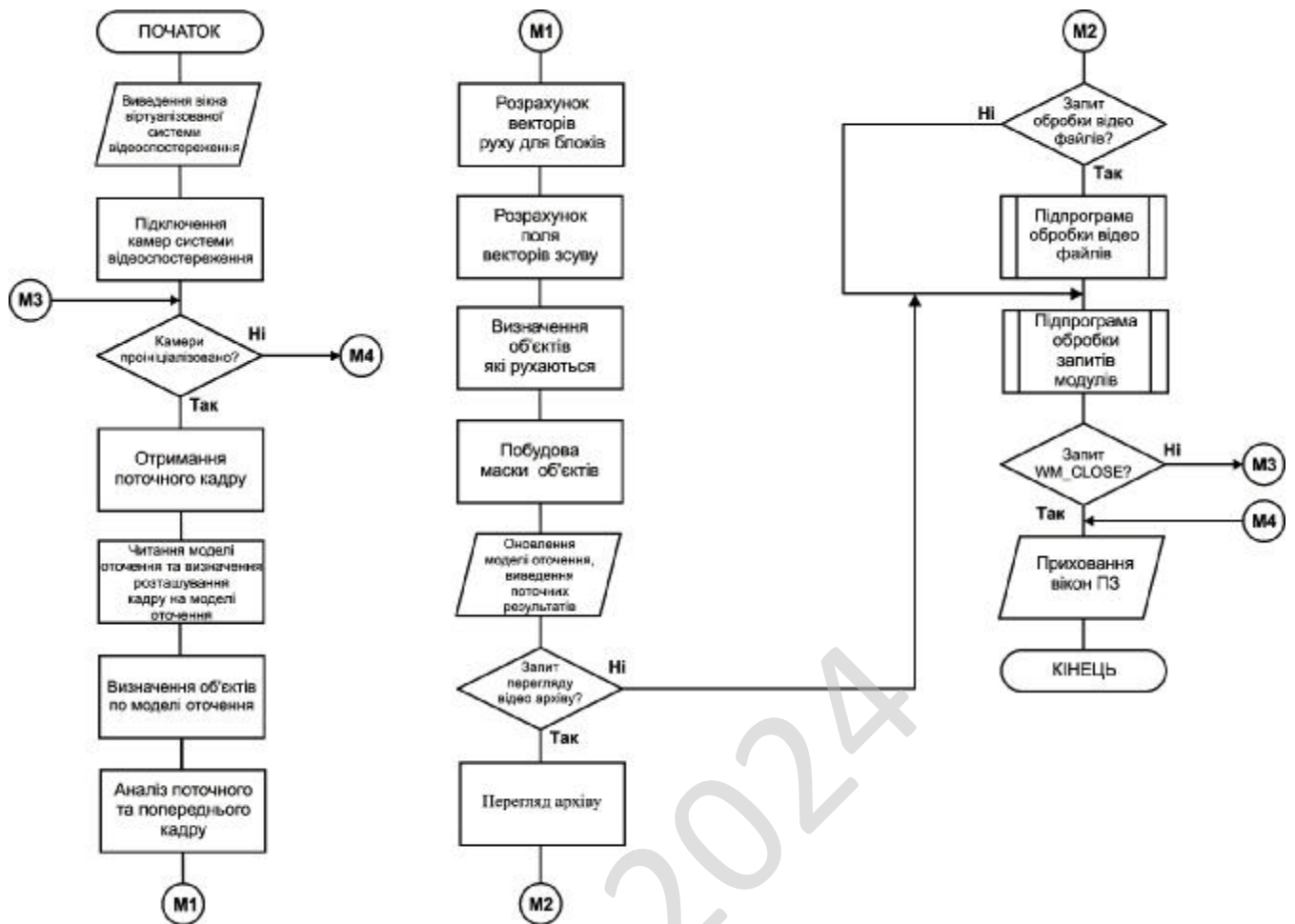


Рисунок 4.1 – Блок-схема основної програми

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

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.

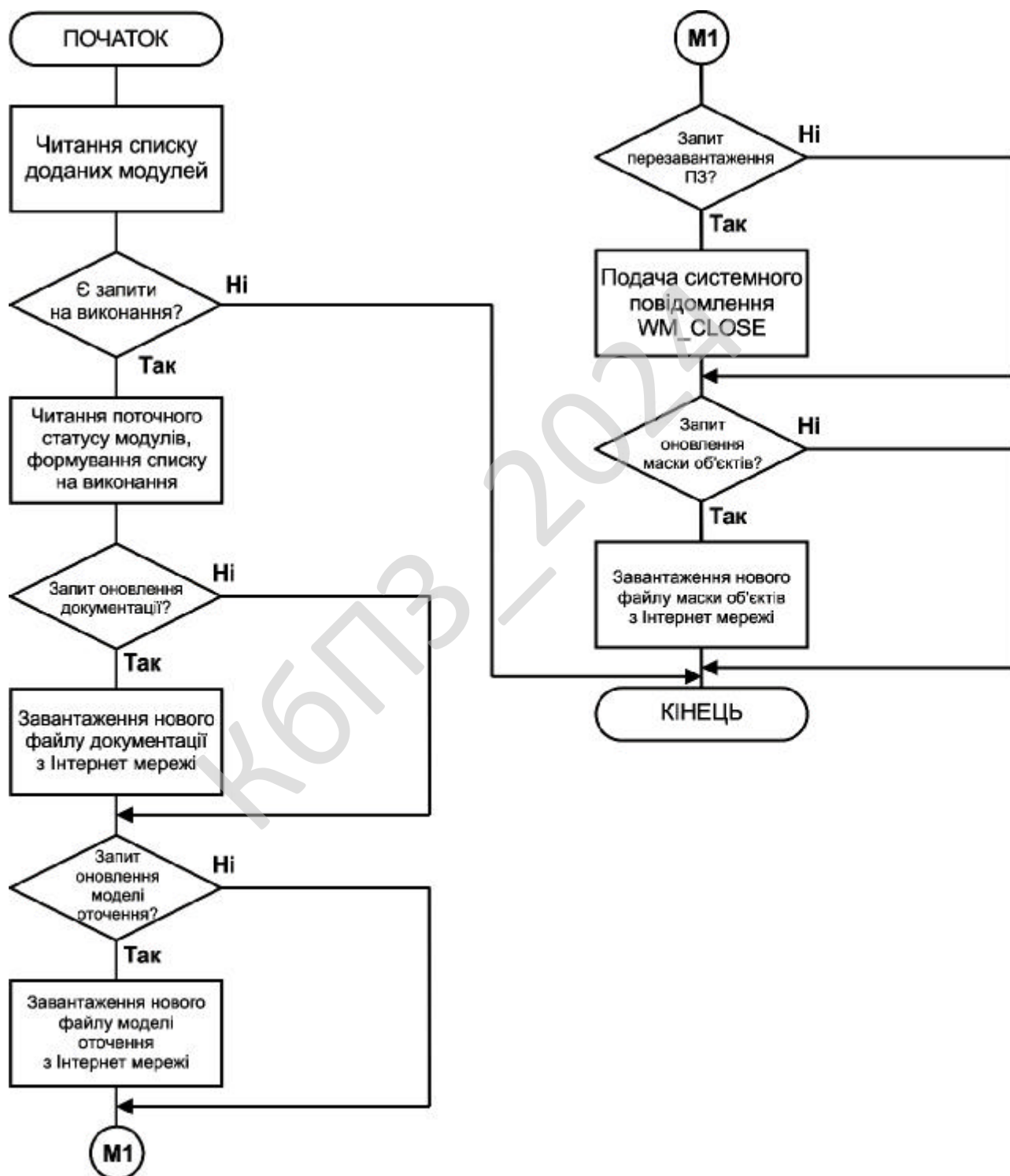


Рисунок 4.2 – Блок-схема роботи підпрограми

– медіа-формати (наприклад кодек), які можуть застосовуватися під час сесії;

– часу старту й зупинки. Використовується у випадку широкомовних сесій, наприклад, телевізійних або радіопрограм.

Можна внести час початку, завершення й часи повторів сесії. Незважаючи на те, що Session Description Protocol надає можливість опису мультимедіаданих, у ньому не вистачає механізмів узгодження параметрів сесії.

Розглянемо приклад надання моделі узгодження на основі механізму відгуку. У прикладі вузли обмінюються SDP повідомленнями з метою дійти згоди щодо формату даних.

```
v=0
o=- 1815849 0 IN IP4 194.67.15.181
s=Cisco SDP 0
c=IN IP4 194.67.15.181
t=0 0
m=audio 20062 RTP/AVP 99 18 101 100
a=rtpmap:99 G.729b/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=rtpmap:100 X-NSE/8000
a=fmtp:100 200-202
a=sqn:0
a=cdsc: 1 audio RTP/AVP 99 18 101 100
a=cdsc: 5 image udpt1 t38
a=cpar: a=T38Faxversion:0
a=cpar: a=T38Faxratemanagement:transferredtcf
a=cpar: a=T38Faxmaxdatagram:160
a=X-sqn:0
a=X-cap: 1 image udpt1 t38
```

Специфікація RTSP значною мірою перегукується зі специфікацією HTTP/1.1. На верхньому рівні обидва протоколи переслідують одну мету – дати можливість клієнтові запитувати зміст із сервера на основі URI запити.

Історично розробка RTSP почалася після того, як була виконана частина робіт над специфікацією HTTP.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Оскільки HTTP послужив основою для RTSP, мети й завдання цих протоколів багато в чому збігаються.

Доставка мультимедійних потоків використовує таку ж саму модель, що й HTTP: ресурси, URI що ідентифікуються, взаємодія запит/відповідь, а також можливість передачі даних через одне або декількох проміжних ланок на шляху між клієнтом і сервером.

Використання готових концепцій, включення спеціальних заголовків протоколу й кодів відповідей зменшило витрати на розробку й реалізацію RTSP. Крім того, обидва протоколи відіграють роль в ініціюванні передачі мультимедійних потоків через WEB.

Подібність між двома протоколами забезпечує значну гнучкість при ухваленні рішення, який протокол буде обслуговувати конкретну функцію.

Розглянемо запит на інформацію з описом сеансу (наприклад, <http://www.foo.com/bar.sdp>). Відповідь Web-сервера буде включати інформацію, у форматі SDP:

```
HTTP/1.1 200 OK Content-Type: application/sdp
v=0
o=- 2890844526 2890842807 IN IP4 192.16.24.202 s=RTSP Session
m=audio 0 RTP/AVP 0
a=control:
rtsp://foo.com/bar/audio
m=video 0 RTP/AVP 31
a=control:
rtsp://foo.com/bar/video
```

У цей момент Web-браузер уже може активізувати медіаплейер для виконання інших дій. Крім того, медіаплейер може надати користувачеві інтерфейс для вибору мультимедійних сеансів.

У цьому випадку медіаплейер може прямо взаємодіяти з Rtp-сервером для уточнення описів без залучення Web-браузера.

Методи запитів RTSP. RTSP-сервери повинні підтримувати чотири основні методи, використовуваних клієнтами для добування мультимедійних сеансів: OPTIONS, SETUP, PLAY і TEARDOWN.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

На верхньому рівні заголовок OPTIONS дозволяє клієнтові визначати функціональні можливості сервера, такі як номер версії RTSP і підтримувані методи; цей метод поводить себе так само, як метод OPTIONS HTTP/1.1.

Інші три методи маніпулюють збереженою на сервері інформацією про стан для координації передачі мультимедійних даних. Клієнт використовує метод SETUP для встановлення транспортного з'єднання для кожного потоку в сеансі. Метод PLAY використовується для ініціювання передачі потоку (потоків).

Метод TEARDOWN використовується для завершення передачі. Більшість методів запитів пов'язане з підтримкою клієнта, який запитує відтворення потоку з мультимедійного сервера.

Однак RTSP – сервер може також допускати запис мультимедійного змісту за допомогою необов'язкових методів ANNOUNCE і RECORD.

Допустимо, користувач прагне записати телеконференцію на Rtsп-сервері для подальшого відтворення. У цьому випадку клієнт відправляє опис конференції серверу в повідомленні ANNOUNCE.

Потім клієнт повинен відправити повідомлення SETUP для кожного з потоків у сеансі, щоб проінформувати сервер про транспортні параметри, таких як Ір-адреси й номери портів. URI запити в повідомленні SETUP указує ім'я, яке клієнт прагне асоціювати із записуваним змістом.

Після цього клієнтові слід відправити запит RECORD для початку запису потоків. Через якийсь час клієнт може зв'язатися із сервером і переглянути записану конференцію.

Обробка відео потоку і виведення на екран в середовищі Windows при застосуванні основних методів виведення відео інформації на екран лінійки операційних систем Windows.

Виникає гостра проблема в швидкості обробки потокового кадру, що приводить до уповільнення процесу виведення інформації на екран. Як відомо для перегляду відеопотоку необхідно не менше 24 кадрів в секунду.

Перед розглядом подробиць схеми роботи програми розглянемо виконані основні напрацювання у вигляді коментованого вихідного коду системи.

```
import cv2
import threading
from flask import Flask, Response

# Ініціалізація Flask веб-додатку
app = Flask(__name__)

# Клас для управління відеопотоками
class VideoStream:
    def __init__(self, rtsp_url):
        self.rtsp_url = rtsp_url
        self.capture = cv2.VideoCapture(self.rtsp_url)
        self.lock = threading.Lock()
        self.current_frame = None
        self.running = True
        thread = threading.Thread(target=self.update_frame)
        thread.start()

    # Оновлення поточного кадру з камери
    def update_frame(self):
        while self.running:
            ret, frame = self.capture.read()
            if ret:
                with self.lock:
                    self.current_frame = frame

    # Отримання поточного кадру для трансляції
    def get_frame(self):
        with self.lock:
            return self.current_frame

    # Зупинка відеопотоку
    def stop(self):
        self.running = False
        self.capture.release()

# Ініціалізація відеопотоків для декількох камер
camera1 = VideoStream('rtsp://camera1_url')
camera2 = VideoStream('rtsp://camera2_url')
```

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

```

# Функція для трансляції відеопотоку через HTTP
def generate_video(camera):
    while True:
        frame = camera.get_frame()
        if frame is not None:
            ret, jpeg = cv2.imencode('.jpg', frame)
            if ret:
                yield (b'--frame\r\n'
                       b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() +
                       b'\r\n\r\n')

# Маршрут для отримання відеопотоку з першої камери
@app.route('/camera1')
def stream_camera1():
    return Response(generate_video(camera1), mimetype='multipart/x-mixed-replace;
boundary=frame')

# Маршрут для отримання відеопотоку з другої камери
@app.route('/camera2')
def stream_camera2():
    return Response(generate_video(camera2), mimetype='multipart/x-mixed-replace;
boundary=frame')

# Запуск Flask сервера
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)

# Основна архітектура включає використання OpenCV для доступу до RTSP-потоків
# Flask для відтворення потокового відео через веб-інтерфейс
# Обробка потоків відбувається в окремих потоках для забезпечення плавності
трансляції
# При використанні багатьох камер необхідно забезпечити достатню пропускну
здатність мережі та ресурси CPU/GPU
# Для перевірки коректності можна вимірювати пропускну здатність каналу та
затримки

```

Система включає в себе використання відеопотоків з декількох камер через RTSP-протокол, обробку відео, а також збереження та відтворення відео в режимі реального часу.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Основні модулі включають OpenCV для обробки зображень, Flask для веб-інтерфейсу, а також бібліотеки для збереження та трансляції відео.

Архітектура

1. Система отримує відеопотоки від IP-камер через протокол RTSP. Для цього використовується бібліотека OpenCV, яка підтримує обробку таких відеопотоків у реальному часі.

2. Відео обробляється в режимі реального часу, кожна камера працює в окремому потоці. Це дозволяє уникати затримок при відтворенні, що особливо важливо при одночасній роботі декількох камер.

3. Виведення та відтворення Відеопотоки виводяться через веб-інтерфейс за допомогою Flask. Кожен відеопотік має свій маршрут, що дозволяє легко додати нові камери або переглядати кілька камер одночасно.

Розрахункові дані

Для оцінки продуктивності системи необхідно враховувати:

1. Пропускнну здатність мережі. Для однієї камери 720p зі швидкістю 30 кадрів на секунду і бітрейтом 4 Мбіт/с, необхідно приблизно 500 КБ/с трафіку.

2. Одночасна обробка відео з декількох камер вимагає значних ресурсів CPU/GPU для декодування відеопотоків.

3. Затримка. Залежно від якості з'єднання та обробки відео, затримка може варіюватися від 100 мс до 1 секунди.

Далі повернемося до схеми, функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами. Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання. Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу. Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань. Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дампи пам'яті або скріншот.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

- Виправлено (виправлення включені у версію таку-то).
- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).
- «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SEED – у криптографії симетричний блоковий криптоалгоритм на основі Мережі Фейстеля, розроблений Корейським агентством інформаційної безпеки (Korean Information Security Agency, KISA) в 1998 році. В алгоритмі використовується 128-бітний блок і ключ довжиною 128 біт. Алгоритм одержав широке поширення й використовується фінансовими й банківськими структурами, виробничими підприємствами й бюджетними установами Південної Кореї, оскільки 40-бітний SSL не забезпечує на даний момент мінімально необхідного рівня безпеки. Агентством по теорії кодування та комп'ютерних мереж специфіковане використання шифру SEED у протоколах TLS і S/MIME. У той же час, алгоритм SEED не реалізований у більшості сучасних браузерів і інтернет-додатків, що утрудняє його використання в даній сфері поза межами Південної Кореї.

SEED являє собою мережу Фейстеля з 16 раундами, 128-бітовими блоками й 128-бітовим ключем. Алгоритм використовує дві 8×8 таблиці підстановки, які, як такі з Safer, виведені з дискретного зведення в ступінь (у цьому випадку, x^{247} і x^{251} – плюс деякі «несумісні операції»). Це є деякою подібністю с MISTY1 у рекурсивності його структури: 128-бітовий повний шифр – мережа Фейстеля з F-функцією, що впливає на 64-бітові половини, у той час як сама F-функція – Мережа Фейстеля, складена з G-функції, що впливає на 32-розрядні половини. Однак рекурсія не простягнеться далі, тому що G-функція – не Мережа Фейстеля.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

В G-функції 32-розрядне слово розглядають як чотири 8-бітових байта, кожний з яких проходить через одну або іншу таблицю підстановки, потім поєднується в помірковано комплексному наборі булевих функцій таким чином, що кожний біт виводу залежить від 3 з 4 вхідних байтів.

SEED має складний ключовий розклад, генеруючи тридцять два 32-розрядних додаткових символу, використовуючи G-функції на серіях обертань вихідного неопрацьованого ключа, комбінованого зі спеціальними раундовими константами (як в TEA) від «Золотого співвідношення» (англ. Golden ratio).

Згідно з дослідженнями KISA, алгоритм SEED «надійно протистоїть відомим атакам».

КБПЗ_2024

					VKPM-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ віртуалізованої системи відеоспостереження яке зображено на рисунку 5.1. Консольний додаток відповідає за виконання обчислень, а результати пересилаються в інтерфейс для подальшого використання. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; Допомога; Архівні дані; Налаштування; Довідка.
- Вікно виведення результату роботи системи – сигнал.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.



Рисунок 5.1 – Головне вікно ПЗ

					VKPM-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Ринок відеоспостереження перспективний, що розвивається, хоча в поточній економічній ситуації робити які-небудь прогнози про його подальшу долю в нашій країні складно. Серед факторів росту виділимо вимоги, що підсилюються, до безпеки, інвестиції в інфраструктуру, перехід від аналогових систем до ІР-відеоспостереження, зміна законодавчих і регулюючих норм. Розширюються сфери застосування ІР-камер: вони всі частіше використовуються не тільки для спостереження, але й для попередження інцидентів і їхнього розслідування. Обмежуючим фактором залишаються оптичні системи, які еволюціонують далеко не так швидко, як процесори й ПЗ.

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.

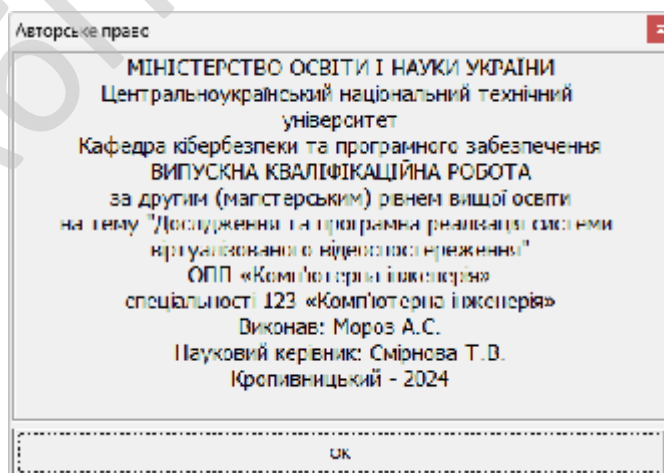


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

- Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

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-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віртуалізованого відеоспостереження.

Метою розробки є дослідження та програмна реалізація системи віртуалізованого відеоспостереження.

Об'єктом дослідження є процес віртуалізованого відеоспостереження.

Предметом дослідження є методи віртуалізованого відеоспостереження.

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

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

- Удосконалено метод віртуалізованого відеоспостереження.
- Розроблено вітчизняний продукт віртуалізованого відеоспостереження, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи віртуалізованого відеоспостереження можуть бути корисні для таких груп, наведених на рисунку 7.1.

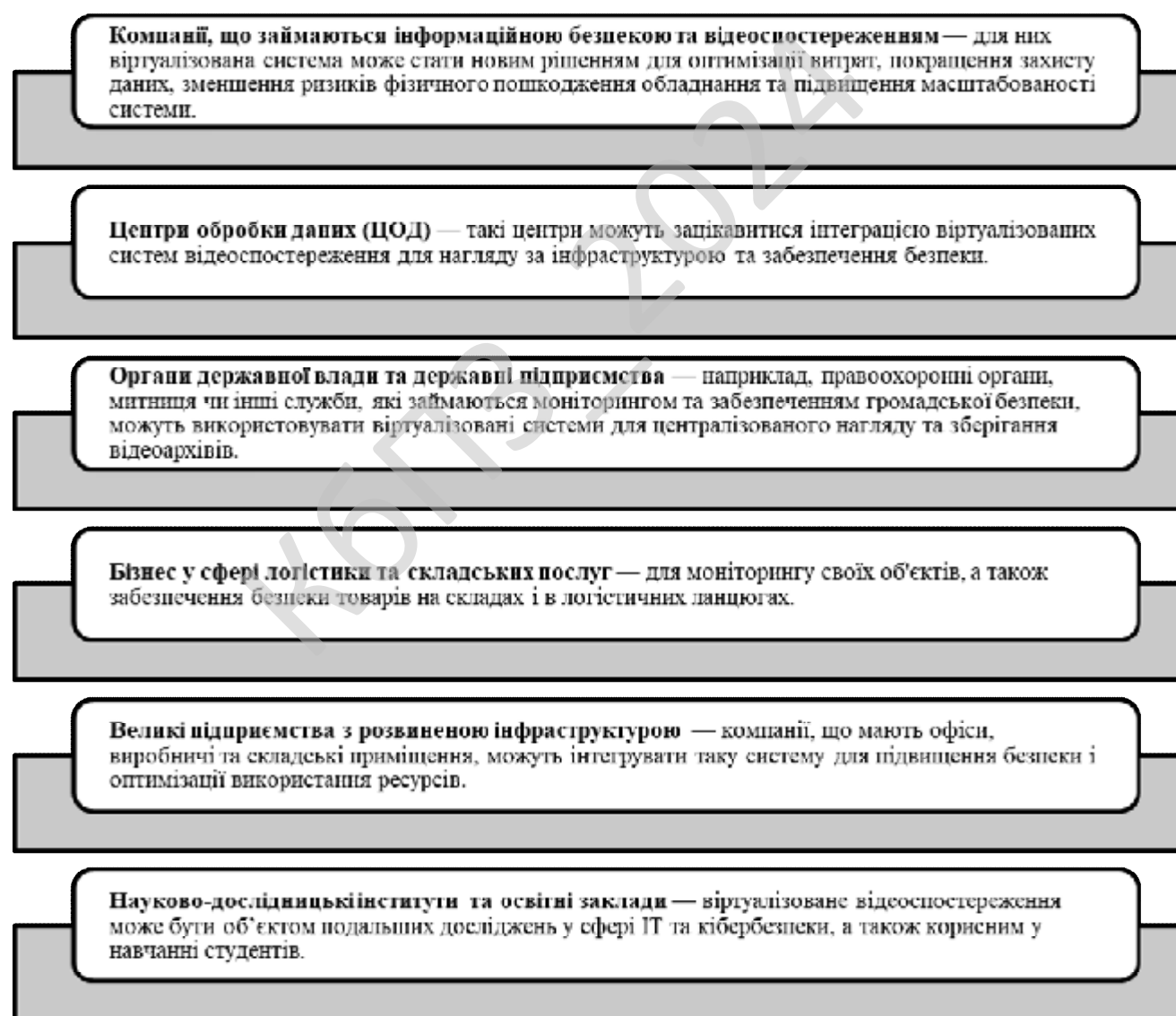


Рисунок 7.1 – Цільова аудиторія

Ця технологія може знайти застосування у будь-якій сфері, де важливо забезпечити високий рівень безпеки, масштабованість, зменшення витрат на обслуговування обладнання та підвищену гнучкість у використанні ресурсів.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи віртуалізованого відеоспостереження можна використати метод експертних оцінок, наприклад метод парних порівнянь або метод бальної оцінки. Проводимо оцінку привабливості системи на основі кількох критеріїв (рис. 7.2).

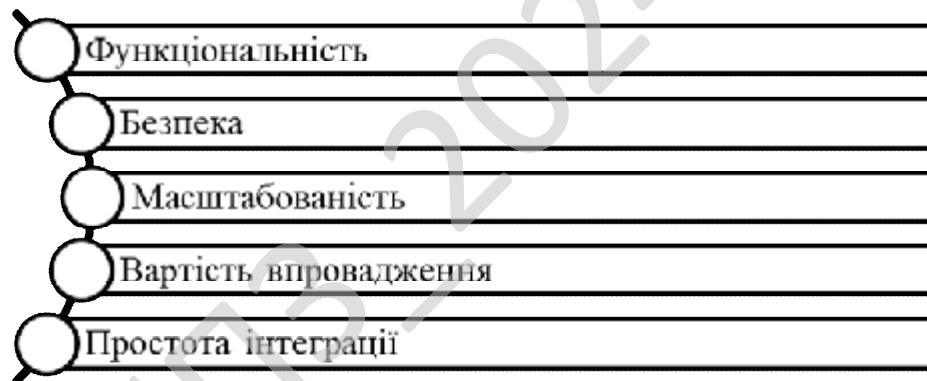


Рисунок 7.2 – Основні критерії оцінки привабливості

Далі проходимо 5 кроків реалізації методу бальної оцінки.

Крок 1: Вибір експертів. Залучаємо групу експертів у сферах ІТ, кібербезпеки, відеоспостереження та віртуалізації. Оптимальна кількість експертів – 5–7 осіб для забезпечення надійності оцінок.

Крок 2: Визначення ваг критеріїв. Експерти мають визначити вагу кожного критерію (в сумі всі ваги мають дорівнювати 1): функціональність – 0,3, безпека – 0,25, масштабованість – 0,2, вартість впровадження – 0,15, простота інтеграції – 0,1.

Крок 3: Оцінка критеріїв. Кожен експерт оцінює систему за кожним критерієм за бальною шкалою від 1 до 10, де 10 – максимальна привабливість. Результати зводимо до таблиці 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Експерт 4	Експерт 5	Середня оцінка
Функціональність	8	9	7	8	8	8,0
Безпека	9	8	8	9	7	8,2
Масштабованість	7	8	9	7	8	7,8
Вартість впровадження	6	7	6	6	5	6,0
Простота інтеграції	8	7	7	8	8	7,6

Крок 4: Розрахунок зваженої оцінки. Обчислюємо зважену оцінку для кожного критерію, помноживши середню оцінку на вагу критерію:

Функціональність: $8,0 \times 0,3 = 2,4$

Безпека: $8,2 \times 0,25 = 2,05$

Масштабованість: $7,8 \times 0,2 = 1,56$

Вартість впровадження: $6,0 \times 0,15 = 0,9$

Простота інтеграції: $7,6 \times 0,1 = 0,76$

Крок 5: Розрахунок загальної оцінки привабливості. Сума всіх зважених оцінок дає загальну оцінку привабливості системи: $2,4 + 2,05 + 1,56 + 0,9 + 0,76 = 7,67$. Загальна оцінка привабливості програмної реалізації системи віртуалізованого відеоспостереження становить 7,67 бала з 10. На основі цієї оцінки можна зробити висновок про потенційну привабливість системи для впровадження.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи віртуалізованого відеоспостереження краще використовувати метод, який враховує специфічні вимоги до розробки, підтримки, та потенційного масштабування системи. Для програмної реалізації системи віртуалізованого відеоспостереження рекомендується поєднати метод функціональних точок (FPA) для визначення вартості програмного забезпечення та метод оцінки за основою витрат (Cost-Based Estimation) для детального розрахунку інфраструктури, інтеграції та підтримки. Метод оцінки вартості за основою витрат (Cost-Based Estimation). Цей метод передбачає детальний розрахунок усіх витрат, пов'язаних із розробкою, впровадженням та підтримкою системи. Він включає оцінку витрат на: заробітну плату команди розробників, інженерів та менеджерів проєкту, закупівлю або оренду серверів і програмного забезпечення, інтеграцію з іншими системами, забезпечення безпеки та управління доступом, тестування, обслуговування та підтримку після запуску. Метод заснований на створенні докладної специфікації проєкту, розрахунку обсягу робіт і часу, необхідного для завершення. Він особливо корисний, якщо система має нестандартну архітектуру або вимагає індивідуальної адаптації для клієнтів.

Метод оцінки на основі функціональних точок (Function Point Analysis, FPA). Метод функціональних точок корисний, якщо потрібно оцінити вартість розробки програмного забезпечення на основі його функціональних вимог. FPA дозволяє оцінити обсяг функціональних компонентів, які необхідно розробити, таких як: введення та виведення даних, обробка транзакцій, зв'язок із зовнішніми системами. Функціональні точки переводяться у зусилля розробки (людино-години) та оцінюються у грошовому еквіваленті. Цей метод добре підходить для оцінки вартості програмного забезпечення з великою кількістю чітко визначених функцій, особливо для розрахунків з масштабованою архітектурою, такою як система віртуалізованого відеоспостереження.

Це дозволить отримати найбільш точну оцінку вартості, враховуючи як функціональні потреби, так і технічну реалізацію системи.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

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

Економічна ефективність від впровадження системи віртуалізованого відеоспостереження може бути розрахована через оцінку прямих і непрямих вигод, отриманих унаслідок її застосування, порівняно з традиційними системами відеоспостереження. Вхідні дані та розрахунки зведемо до таблиці 7.2.

Таблиця 7.2 – Економічна ефективність від реалізації

Вихідні умови:
Класична система відеоспостереження має витрати на фізичні сервери, оновлення обладнання та високі витрати на обслуговування. Віртуалізована система зменшує потребу в обладнанні, використовуючи централізовану інфраструктуру та оптимізовані серверні потужності.
Зниження витрат на обладнання Класична система: потребує значної кількості серверів і накопичувачів даних для кожного об'єкта відеоспостереження. Віртуалізована система: використовує менше фізичних серверів, що скорочує витрати на закупівлю обладнання.
Вартість обладнання для класичної системи = \$100,000 Вартість обладнання для віртуалізованої системи = \$50,000 Економія: \$100,000 - \$50,000 = \$50,000
Зниження витрат на технічне обслуговування Віртуалізована система потребує менше фізичних ресурсів, що зменшує витрати на обслуговування та ремонти.
Річне обслуговування класичної системи = \$15,000 Річне обслуговування віртуалізованої системи = \$8,000 Економія: \$15,000 - \$8,000 = \$7,000

Продовження таблиці 7.2

Оптимізація витрат на зберігання даних

Замість великих фізичних накопичувачів даних віртуалізована система може використовувати централізовані або хмарні сховища, що скорочує витрати на зберігання даних.

Річні витрати на зберігання даних для класичної системи = \$20,000

Річні витрати для віртуалізованої системи (враховуючи хмарні сховища) = \$12,000

Економія: \$20,000 - \$12,000 = \$8,000

Зниження витрат на електроенергію

Завдяки меншій кількості фізичних серверів віртуалізована система споживає менше електроенергії.

Річні витрати на електроенергію для класичної системи = \$5,000

Річні витрати на електроенергію для віртуалізованої системи = \$2,500

Економія: \$5,000 - \$2,500 = \$2,500

Підвищення продуктивності персоналу

Віртуалізована система може автоматизувати деякі процеси моніторингу та аналітики, скорочуючи навантаження на персонал і дозволяючи їм зосередитися на більш складних задачах.

Зменшення потреби в операторах = 2 працівники (зарплата \$30,000 на рік кожен)

Економія: 2 × \$30,000 = \$60,000

Сумарна економія від переходу на віртуалізовану систему відеоспостереження: 50,000+7,000+8,000+2,500+60,000=127,500 доларів на рік

Впровадження віртуалізованої системи відеоспостереження дозволяє зекономити \$127,500 на рік за рахунок зменшення витрат на обладнання, технічне обслуговування, зберігання даних, електроенергію та зниження навантаження на персонал.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Процес просування проєкту програмної реалізації системи віртуалізованого відеоспостереження можна представити графічно на рисунку 7.3.

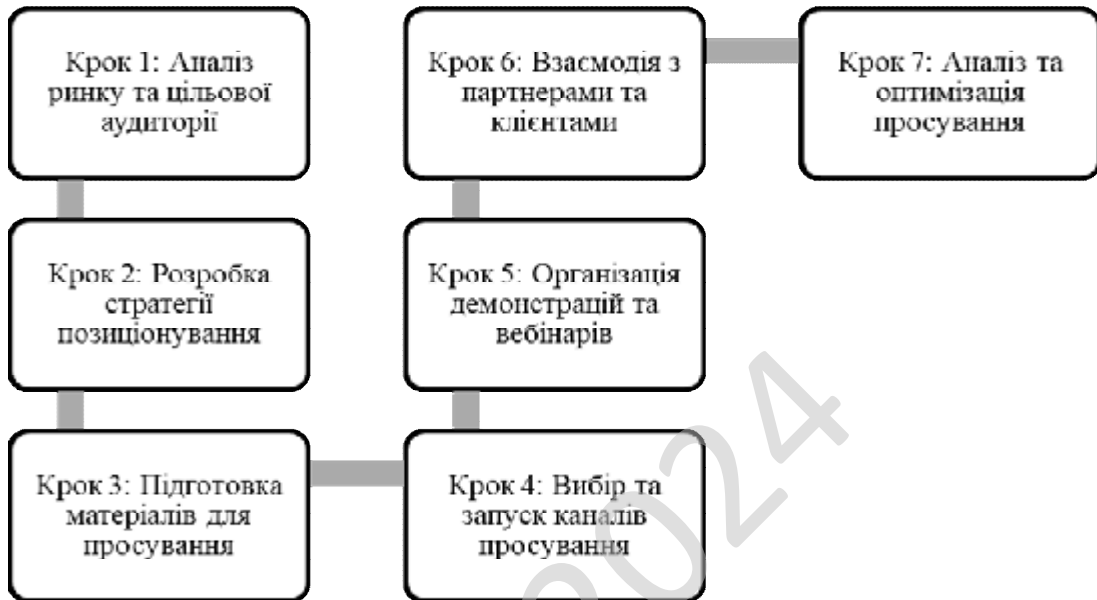


Рисунок 7.3 – Алгоритм просування проєкту

Цей план допоможе ефективно позиціонувати проєкт на ринку, залучити потенційних клієнтів і продемонструвати переваги системи. Цей алгоритм допоможе систематизувати та оптимізувати процес просування, підвищуючи ймовірність успіху проєкту та залучення клієнтів для програмної реалізації системи віртуалізованого відеоспостереження.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи віртуалізованого відеоспостереження можна застосувати комплексний підхід, що враховує специфіку продукту та цільової аудиторії (рисунок 7.4).

1. сегментація каналів збуту

- **прямі продажі:** створіть спеціалізовану команду з продажів, яка буде працювати з великими корпоративними клієнтами, такими як бізнес-центри, державні установи та компанії з високими вимогами до безпеки, такі клієнти часто потребують індивідуального підходу, і прямі продажі допоможуть налаштувати рішення відповідно до їхніх потреб.
- **інтегратори систем безпеки:** залучіть партнерів, які спеціалізуються на інтеграції систем відеоспостереження, кібербезпеки, або мережних рішень, інтегратори можуть надавати готові комплексні рішення клієнтам, що підвищить охоплення і дозволить ефективніше досягати нових сегментів ринку.
- **онлайн-канали:** враховуючи сучасні тенденції, забезпечте присутність продукту на спеціалізованих маркетплейсах для й-рішень, де компанії шукають продукти з функціями віртуалізації та відеоспостереження.

2. моделі цінової оптимізації

- **програмне забезпечення як послуга (SAAS):** розгляньте можливість надання системи як saas-рішення, така модель дозволить клієнтам сплачувати лише за використані ресурси або за час роботи системи, що зробить продукт доступнішим для невеликих компаній і скоротить витрати клієнтів на апаратне забезпечення.
- **гнучке ціноутворення:** запропонуйте різні цінові пакети на основі потреб замовника — від базового плану з основними функціями до преміального з розширеними можливостями безпеки та підтримкою, це дозволить охопити ширшу аудиторію.

3. автоматизація маркетингу та управління лідями

- **автоматизовані інструменти для залучення лідів:** використовуйте платформи автоматизації маркетингу (наприклад, HubSpot, Marketo), щоб збирати лідів з різних каналів і сегментувати їх для подальшої роботи, налаштуйте автоматичні кампанії, які будуть послідовно підводити потенційних клієнтів до ухвалення рішення.
- **ретаргетинг:** запровадьте ретаргетингові кампанії, щоб повернути на сайт відвідувачів, які вже проявили інтерес до системи, але не завершили покупку.

4. розширення партнерських програм

- **партнерства з компаніями-постачальниками хмарних рішень:** співпраця з провайдерами хмарних послуг, такими як AWS, Microsoft Azure або Google Cloud, може покращити доступність вашої системи для компаній, які вже використовують ці платформи.
- **співпраця з постачальниками обладнання:** залучайте партнерів, які пропонують рішення для відеоспостереження (наприклад, виробників камер або мережних пристроїв), для спільного просування інтегрованих продуктів.

5. оптимізація каналу підтримки клієнтів

- **цілодобова технічна підтримка:** надання цілодобової технічної підтримки підвищить довіру клієнтів, особливо тих, для кого безперебійна робота системи є критичною, можна впровадити чат-боти для швидкої відповіді на типові запити.
- **навчання клієнтів та інструктаж:** запропонуйте детальні інструкції, відеосюроки та вебінари з налаштування та використання системи, що знизить навантаження на підтримку та полегшить адптацію клієнтів.

6. активна рг-кампанія та робота з відгуками

- **публікації в медіа:** опублікуйте статті в провідних виданнях на тему безпеки, технології віртуалізації та відеоспостереження, щоб підвищити впізнаваність бренду.
- **відгуки клієнтів:** збирайте та публікуйте реальні відгуки клієнтів, кейси з позитивним досвідом використання вашої системи, це створює додаткову цінність і підвищує довіру.

7. впровадження системи аналітики для моніторингу збуту

- **аналіз каналів збуту:** використовуйте аналітичні інструменти, щоб відстежувати результати роботи кожного каналу, такі як кількість лідів, вартість конверсії та прибутковість, це дозволить своєчасно коригувати стратегію продажів.
- **оцінка результатів та адаптація стратегії:** регулярно аналізуйте ефективність збутових каналів і шляхів реалізації, щоб оптимізувати стратегію збуту та адаптувати її до змін на ринку.

8. демо-пропозиції для великих клієнтів

- **тестові періоди:** надавайте потенційним клієнтам можливість безкоштовно протестувати систему або використати її на певних умовах протягом певного часу, це дозволить клієнтам оцінити систему в дії і збільшить ймовірність покупки.
- **індивідуальні презентації:** організуйте презентації системи для великих замовників із демонстрацією її можливостей на конкретних кейсах і завданнях, що відображають їхні потреби.

Рисунок 7.4 – Оптимізація каналів збуту та шляхів реалізації

Ці кроки можуть допомогти ефективно оптимізувати канали збуту, покращити взаємодію з цільовою аудиторією та підвищити рентабельність проєкту системи віртуалізованого відеоспостереження.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи віртуалізованого відеоспостереження охоплюють технічні, управлінські та ринкові аспекти, які забезпечують конкурентоспроможність, ефективність і стабільність роботи системи (рисунк 7.5).

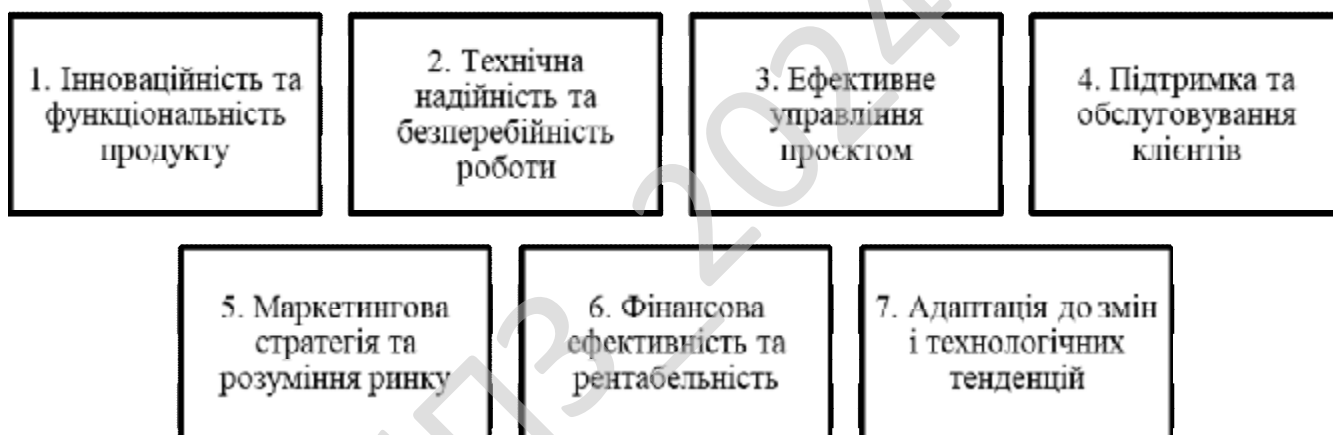


Рисунок 7.5 – Ключові фактори успіху проєкту

Ці ключові фактори забезпечать конкурентоспроможність та ефективність системи віртуалізованого відеоспостереження, підвищуючи її привабливість для клієнтів і сприяючи успішному просуванню на ринку.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

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

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

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

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

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

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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

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

Ці шкідливі фактори можуть привести до професійних захворювань.

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

Умови праці програміста включають наступні фактори:

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

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

8.2 Пожежна безпека

Вимог пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам. Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

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

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

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

- евакуаційних шляхів;
- місць для паління;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

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

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами. Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

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

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

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

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

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

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

8.5 Розрахункова частина

Для захисного штучного заземлення будемо застосовувати вертикальні електроди з сталевого прокату круглого перерізу діаметром 45 мм., довжиною $L=2$ м., та горизонтальний електрод – металева полоса з перетином 45·5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення:

– тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір $\rho_2 = 40$ Ом·м);

– умовна товщина верхнього шару ґрунту: $H=0,55$ м;

– відстань між вертикальними заземлювачами (електродами) $A=3$ м;

– глибина закладення горизонтального контура заземлення $t=0,75$ м;

– опір заземлювача, який нормується: $R_{3Н} = 4$ Ом.

Необхідно визначити потрібну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача) (рис. 8.1).

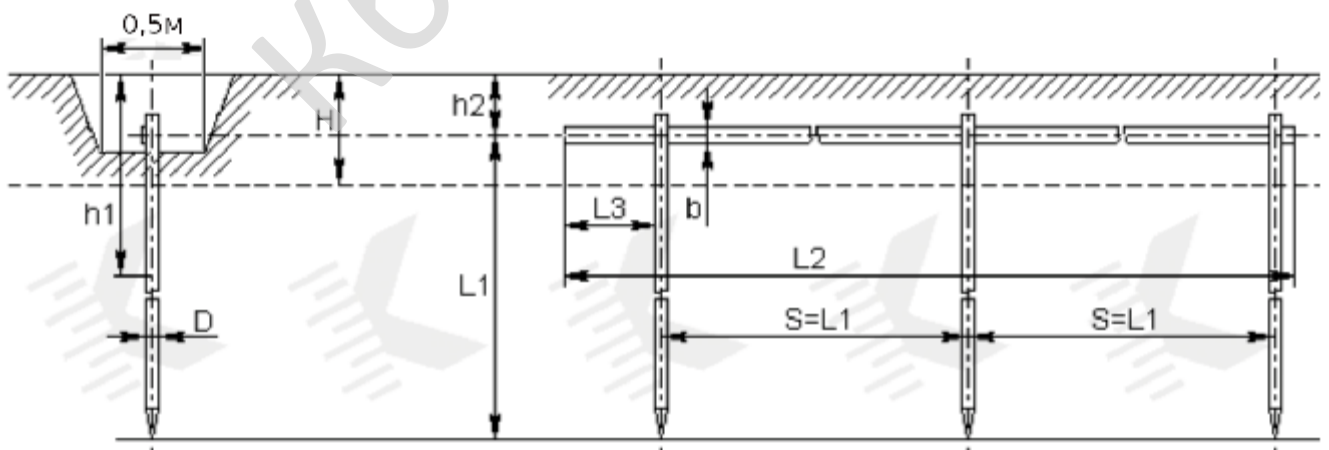


Рисунок 8.1 – Схема штучного заземлення

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

$B = 45 \text{ мм} = 0,045 \text{ м.}$ - ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [11]:

$$R = (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) = \\ = (20,6 \cdot 11,57) / (20,6 \cdot 0,55 + 10 \cdot 11,57 \cdot 0,53) = 3,3 \text{ Ом.}$$

де $\eta_{\Pi} = 0,55$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

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

Оскільки R суттєво більше $R_{3Н}$, зменшимо кількість вертикальних електродів до 8 і виконаємо перерахунок. У результаті остаточно отримали: кількість вертикальних електродів дорівнює 8 при $R = 3,9 \text{ Ом}$.

Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд питань пожежної безпеки, небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи віртуалізованого відеоспостереження.

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 віртуалізованого відеоспостереження.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем віртуалізованого відеоспостереження.

– Досліджена система віртуалізованого відеоспостереження.

– На основі отриманих результатів досліджень створена програмна реалізація системи віртуалізованого відеоспостереження.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віртуалізованого відеоспостереження.

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-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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 SEED.

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

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мороз А.С. Дослідження та програмна реалізація системи віртуалізованого відеоспостереження // 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. 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.
11. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

12. 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.

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

14. 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.

15. 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>

16. 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.

17. 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.

18. 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.

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

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

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

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

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

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

40. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

“Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

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

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

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

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

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

46. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI’2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

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

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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

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

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

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

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

КБГІЗ 2024

					ВКРМ-123.24.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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-123.24.0001.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Мороз А.С.				Дослідження та програмна реалізація системи віртуалізованого відеоспостереження	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23Мз			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи віртуалізованого відеоспостереження.

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. № 21-13 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 віртуалізованого відеоспостереження.

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-123.24.0001.00.00.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 віртуалізованого відеоспостереження;
- 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-123.24.0001.00.00.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-123.24.0001.00.00.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 повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівців.

					ВКРМ-123.24.0001.00.00.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 – 82 аркуші.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 17.12.2024 р.

					ВКРМ-123.24.0001.00.00.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

_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи віртуалізованого відеоспостереження*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 17

Літера: РП

Кропивницький – 2024 року

Основна програма

```

import cv2
import threading
import time
from flask import Flask, Response, render_template
import os

# Створюємо Flask веб-додаток
app = Flask(__name__)

# Клас для обробки та керування відеопотоком
class VideoStream:
    def __init__(self, rtsp_url, save_dir):
        self.rtsp_url = rtsp_url
        self.capture = cv2.VideoCapture(self.rtsp_url)
        self.lock = threading.Lock()
        self.current_frame = None
        self.running = True
        self.save_dir = save_dir
        self.recording = False
        thread = threading.Thread(target=self.update_frame)
        thread.start()

    # Оновлюємо кадри для потокової передачі
    def update_frame(self):
        while self.running:
            ret, frame = self.capture.read()
            if ret:
                with self.lock:
                    self.current_frame = frame
                if self.recording:
                    self.save_frame(frame)

    # Отримуємо поточний кадр
    def get_frame(self):
        with self.lock:
            return self.current_frame

    # Зупиняємо відеопотік
    def stop(self):
        self.running = False
        self.capture.release()

    # Починаємо запис відео
    def start_recording(self):
        self.recording = True

    # Зупиняємо запис відео
    def stop_recording(self):
        self.recording = False

    # Зберігаємо кадр у файл
    def save_frame(self, frame):
        timestamp = time.strftime("%Y%m%d-%H%M%S")
        file_path = os.path.join(self.save_dir, f"frame_{timestamp}.jpg")
        cv2.imwrite(file_path, frame)

# Ініціалізуємо відеопотоки
camera1 = VideoStream('rtsp://camera1_url', './recordings/camera1')
camera2 = VideoStream('rtsp://camera2_url', './recordings/camera2')

# Функція для потокової передачі відео
def generate_video(camera):
    while True:
        frame = camera.get_frame()
        if frame is not None:

```

```

        ret, jpeg = cv2.imencode('.jpg', frame)
        if ret:
            yield (b'--frame\r\n'
                  b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() +
                  b'\r\n\r\n')

# Маршрут для перегляду камери 1
@app.route('/camera1')
def stream_camera1():
    return Response(generate_video(camera1), mimetype='multipart/x-mixed-replace; boundary=frame')

# Маршрут для перегляду камери 2
@app.route('/camera2')
def stream_camera2():
    return Response(generate_video(camera2), mimetype='multipart/x-mixed-replace; boundary=frame')

# Головна сторінка для вибору камер
@app.route('/')
def index():
    return render_template('index.html')

# Маршрут для запуску запису камери 1
@app.route('/start_recording_camera1')
def start_recording_camera1():
    camera1.start_recording()
    return "Recording started for camera 1"

# Маршрут для зупинки запису камери 1
@app.route('/stop_recording_camera1')
def stop_recording_camera1():
    camera1.stop_recording()
    return "Recording stopped for camera 1"

# Маршрут для запуску запису камери 2
@app.route('/start_recording_camera2')
def start_recording_camera2():
    camera2.start_recording()
    return "Recording started for camera 2"

# Маршрут для зупинки запису камери 2
@app.route('/stop_recording_camera2')
def stop_recording_camera2():
    camera2.stop_recording()
    return "Recording stopped for camera 2"

# Функція для запуску сервера
if __name__ == '__main__':
    if not os.path.exists('./recordings/camera1'):
        os.makedirs('./recordings/camera1')
    if not os.path.exists('./recordings/camera2'):
        os.makedirs('./recordings/camera2')
    app.run(host='0.0.0.0', port=5000)

# Додатковий клас для запису аудіо
class AudioStream:
    def __init__(self, audio_source, save_dir):
        self.audio_source = audio_source
        self.save_dir = save_dir
        self.recording = False

    # Початок запису аудіо
    def start_recording(self):
        self.recording = True
        thread = threading.Thread(target=self.record_audio)
        thread.start()

    # Зупинка запису аудіо

```

```

def stop_recording(self):
    self.recording = False

# Запис аудіо у файл
def record_audio(self):
    while self.recording:
        # Тут реалізація аудіозапису
        pass

# Додатковий функціонал для керування аудіо записом
audio1 = AudioStream('microphone1', './audio/camera1')
audio2 = AudioStream('microphone2', './audio/camera2')

@app.route('/start_audio_camera1')
def start_audio_camera1():
    audio1.start_recording()
    return "Audio recording started for camera 1"

@app.route('/stop_audio_camera1')
def stop_audio_camera1():
    audio1.stop_recording()
    return "Audio recording stopped for camera 1"

@app.route('/start_audio_camera2')
def start_audio_camera2():
    audio2.start_recording()
    return "Audio recording started for camera 2"

@app.route('/stop_audio_camera2')
def stop_audio_camera2():
    audio2.stop_recording()
    return "Audio recording stopped for camera 2"

# Додаткові функції для інтеграції з системами моніторингу
class MonitoringSystem:
    def __init__(self):
        self.alerts = []

    # Додаємо тривогу
    def add_alert(self, message):
        timestamp = time.strftime("%Y%m%d-%H%M%S")
        self.alerts.append(f"{timestamp}: {message}")
        print(f"Alert added: {message}")

    # Отримуємо список всіх тривог
    def get_alerts(self):
        return self.alerts

# Ініціалізуємо систему моніторингу
monitoring_system = MonitoringSystem()

# Маршрут для отримання всіх тривог
@app.route('/alerts')
def get_alerts():
    return "<br>".join(monitoring_system.get_alerts())

# Функція для перевірки статусу камер
def check_camera_status():
    while True:
        if not camera1.capture.isOpened():
            monitoring_system.add_alert("Camera 1 disconnected")
        if not camera2.capture.isOpened():
            monitoring_system.add_alert("Camera 2 disconnected")
        time.sleep(10)

# Запускаємо окремий потік для перевірки статусу камер
thread = threading.Thread(target=check_camera_status)
thread.start()

```

```
# Додатковий функціонал для управління користувачами
class UserManager:
    def __init__(self):
        self.users = {}

    # Додаємо нового користувача
    def add_user(self, username, password):
        self.users[username] = password

    # Перевіряємо чи існує користувач
    def user_exists(self, username):
        return username in self.users

    # Авторизація користувача
    def authenticate(self, username, password):
        return self.users.get(username) == password

# Ініціалізуємо менеджер користувачів
user_manager = UserManager()

# Додаємо декілька користувачів
user_manager.add_user("admin", "admin123")
user_manager.add_user("user", "user123")
```

КБПЗ_2024

User Authentication and Authorization (auth.py)

```
# auth.py
class UserManager:
    def __init__(self):
        self.users = {}

    # Реєстрація нового користувача
    def register_user(self, username, password):
        if username in self.users:
            return False # Користувач вже існує
        self.users[username] = password
        return True

    # Логін користувача
    def login_user(self, username, password):
        if self.users.get(username) == password:
            return True
        return False

    # Перевірка прав доступу
    def check_permissions(self, user_role, resource):
        # Простий приклад розмежування доступів
        if user_role == "admin":
            return True
        if user_role == "user" and resource != "admin_area":
            return True
        return False

# Приклад використання
user_manager = UserManager()
user_manager.register_user("admin", "admin123")
user_manager.register_user("user", "user123")
```

Monitoring and Alerts (monitoring.py)

```
# monitoring.py
import time

class MonitoringSystem:
    def __init__(self):
        self.alerts = []

    # Додаємо тривогу
    def add_alert(self, message):
        timestamp = time.strftime("%Y%m%d-%H%M%S")
        self.alerts.append(f"{timestamp}: {message}")
        print(f"Alert added: {message}")

    # Отримуємо список всіх тривог
    def get_alerts(self):
        return self.alerts

    # Перевірка підключення камери
    def check_camera_connection(self, camera):
        if not camera.capture.isOpened():
            self.add_alert(f"Camera {camera.rtsp_url} disconnected")

# Приклад використання
monitoring_system = MonitoringSystem()
```

Video Recording Management (recording.py)

```
# recording.py
import os
import time
import cv2

class VideoRecorder:
    def __init__(self, camera, save_dir):
        self.camera = camera
        self.save_dir = save_dir
        self.recording = False

    # Початок запису відео
    def start_recording(self):
        self.recording = True
        thread = threading.Thread(target=self.record_video)
        thread.start()

    # Зупинка запису відео
    def stop_recording(self):
        self.recording = False

    # Запис відео у файл
    def record_video(self):
        while self.recording:
            ret, frame = self.camera.capture.read()
            if ret:
                timestamp = time.strftime("%Y%m%d-%H%M%S")
                file_path = os.path.join(self.save_dir,
f"video_{timestamp}.avi")
                cv2.imwrite(file_path, frame)
                time.sleep(1) # Запис кожну секунду

# Приклад використання
camera = VideoStream('rtsp://camera_url', './recordings')
recorder = VideoRecorder(camera, './recordings/camera1')
```

Audio Recording Management (audio_recording.py)

```
# audio_recording.py
import os
import time

class AudioRecorder:
    def __init__(self, audio_source, save_dir):
        self.audio_source = audio_source
        self.save_dir = save_dir
        self.recording = False

    # Початок запису аудіо
    def start_recording(self):
        self.recording = True
        thread = threading.Thread(target=self.record_audio)
        thread.start()

    # Зупинка запису аудіо
    def stop_recording(self):
        self.recording = False

    # Запис аудіо у файл
    def record_audio(self):
        while self.recording:
            # Симуляція запису аудіо, можна додати бібліотеку для реального
            аудіозапису
            timestamp = time.strftime("%Y%m%d-%H%M%S")
            file_path = os.path.join(self.save_dir, f"audio_{timestamp}.wav")
            with open(file_path, 'w') as file:
                file.write(f"Audio recorded at {timestamp}")
            time.sleep(1) # Запис кожену секунду

# Приклад використання
audio_recorder = AudioRecorder('microphone1', './audio/camera1')
```

Streaming Management (streaming.py)

```
# streaming.py
from flask import Response
import cv2

class StreamManager:
    def __init__(self, camera):
        self.camera = camera

    # Генерація відеопотоку
    def generate_video_stream(self):
        while True:
            frame = self.camera.get_frame()
            if frame is not None:
                ret, jpeg = cv2.imencode('.jpg', frame)
                if ret:
                    yield (b'--frame\r\n'
                           b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes()
                           + b'\r\n\r\n')

# Приклад використання
stream_manager = StreamManager(camera)
@app.route('/stream_camera1')
def stream_camera1():
    return Response(stream_manager.generate_video_stream(),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
```

Scheduling and Automation (scheduling.py)

```
# scheduling.py
import threading
import time

class Scheduler:
    def __init__(self):
        self.scheduled_tasks = []

    # Плануємо запис
    def schedule_recording(self, camera_id, start_time, end_time, recorder):
        task = threading.Thread(target=self._run_scheduled_recording,
args=(camera_id, start_time, end_time, recorder))
        self.scheduled_tasks.append(task)
        task.start()

    # Відміна запланованого запису
    def cancel_scheduled_recording(self, task_id):
        if task_id < len(self.scheduled_tasks):
            self.scheduled_tasks[task_id].cancel()

    # Виконання запланованого запису
    def _run_scheduled_recording(self, camera_id, start_time, end_time,
recorder):
        time.sleep(max(0, start_time - time.time())) # Чекаємо до початку
запису
        recorder.start_recording()
        time.sleep(end_time - start_time) # Чекаємо до завершення запису
        recorder.stop_recording()

    # Приклад використання
scheduler = Scheduler()
scheduler.schedule_recording('camera1', time.time() + 5, time.time() + 20,
recorder)
```

File Management (file_management.py)

```
# file_management.py
import os
import time

class FileManager:
    def __init__(self, directory):
        self.directory = directory

    # Показуємо всі файли в директорії
    def list_recordings(self):
        return [f for f in os.listdir(self.directory) if
os.path.isfile(os.path.join(self.directory, f))]

    # Видаляємо старі записи
    def delete_old_recordings(self, days_to_keep):
        cutoff_time = time.time() - (days_to_keep * 86400)
        for filename in self.list_recordings():
            file_path = os.path.join(self.directory, filename)
            if os.path.getmtime(file_path) < cutoff_time:
                os.remove(file_path)

    # Отримуємо метадані файлу
    def get_file_metadata(self, file_path):
        if os.path.exists(file_path):
            file_stat = os.stat(file_path)
            return {
                'size': file_stat.st_size,
                'last_modified': time.ctime(file_stat.st_mtime),
                'created': time.ctime(file_stat.st_ctime)
            }
        else:
            return None

# Приклад використання
file_manager = FileManager('./recordings/cameral')
print(file_manager.list_recordings())
```

Notification System (notifications.py)

```
# notifications.py
import smtplib
from email.mime.text import MIMEText

class NotificationManager:
    def __init__(self, smtp_server, smtp_port, smtp_user, smtp_password):
        self.smtp_server = smtp_server
        self.smtp_port = smtp_port
        self.smtp_user = smtp_user
        self.smtp_password = smtp_password

    # Відправляємо email-повідомлення
    def send_email_notification(self, recipient_email, subject, message):
        msg = MIMEText(message)
        msg['Subject'] = subject
        msg['From'] = self.smtp_user
        msg['To'] = recipient_email
        with smtplib.SMTP(self.smtp_server, self.smtp_port) as server:
            server.starttls()
            server.login(self.smtp_user, self.smtp_password)
            server.sendmail(self.smtp_user, recipient_email, msg.as_string())

    # Записуємо лог повідомлень
    def create_notification_log(self, message, timestamp):
        with open('notification_log.txt', 'a') as log_file:
            log_file.write(f"{timestamp}: {message}\n")

# Приклад використання
notification_manager = NotificationManager('smtp.example.com', 587,
    'user@example.com', 'password')
notification_manager.send_email_notification('recipient@example.com', 'Alert',
    'Camera disconnected!')
```

Settings and Configuration Management (config.py)

```
# config.py
import json

class ConfigManager:
    def __init__(self, config_file):
        self.config_file = config_file
        self.config_data = self.load_config()

    # Завантаження конфігурації з файлу
    def load_config(self):
        if os.path.exists(self.config_file):
            with open(self.config_file, 'r') as file:
                return json.load(file)
        return {}

    # Збереження конфігурації до файлу
    def save_config(self, new_config):
        self.config_data.update(new_config)
        with open(self.config_file, 'w') as file:
            json.dump(self.config_data, file)

    # Отримання конкретного налаштування
    def get_setting(self, setting_name):
        return self.config_data.get(setting_name, None)

# Приклад використання
config_manager = ConfigManager('config.json')
config_manager.save_config({'camera1_url': 'rtsp://camera1'})
print(config_manager.get_setting('camera1_url'))
```

Analytics and Reporting (analytics.py)

```
# analytics.py
import csv
from datetime import datetime

class Analytics:
    def __init__(self):
        self.data = []

    # Додаємо подію в аналітичні дані
    def log_event(self, camera_id, event_type, timestamp=None):
        if timestamp is None:
            timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        self.data.append({'camera_id': camera_id, 'event_type': event_type,
'timestamp': timestamp})

    # Генеруємо звіт по трафіку камер
    def generate_traffic_report(self, camera_id, start_date, end_date):
        report = []
        for entry in self.data:
            if entry['camera_id'] == camera_id and start_date <=
entry['timestamp'] <= end_date:
                report.append(entry)
        return report

    # Генеруємо звіт по подіях
    def generate_event_report(self, event_type, start_date, end_date):
        report = []
        for entry in self.data:
            if entry['event_type'] == event_type and start_date <=
entry['timestamp'] <= end_date:
                report.append(entry)
        return report

    # Експортуємо звіт у CSV файл
    def export_report_to_csv(self, report_data, file_path):
        with open(file_path, 'w', newline='') as csvfile:
            fieldnames = ['camera_id', 'event_type', 'timestamp']
            writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
            writer.writeheader()
            for entry in report_data:
                writer.writerow(entry)

    # Приклад використання
    analytics = Analytics()
    analytics.log_event('camera1', 'motion_detected')
    analytics.log_event('camera2', 'disconnected')

start_date = '2024-10-01 00:00:00'
end_date = '2024-10-21 23:59:59'
```

```
traffic_report = analytics.generate_traffic_report('camera1', start_date,  
end_date)  
event_report = analytics.generate_event_report('disconnected', start_date,  
end_date)
```

```
analytics.export_report_to_csv(traffic_report, 'traffic_report.csv')  
analytics.export_report_to_csv(event_report, 'event_report.csv')
```

К6П3_2024

Camera Management (camera_management.py)

```
# camera_management.py
class CameraManager:
    def __init__(self):
        self.cameras = {}

    # Додаємо нову камеру
    def add_camera(self, camera_id, camera_name, rtsp_url):
        if camera_id not in self.cameras:
            self.cameras[camera_id] = {
                'name': camera_name,
                'rtsp_url': rtsp_url,
                'status': 'offline'
            }
            return True
        return False # Камера з таким ID вже існує

    # Видаляємо камеру
    def remove_camera(self, camera_id):
        if camera_id in self.cameras:
            del self.cameras[camera_id]
            return True
        return False # Камери з таким ID немає

    # Оновлюємо дані камери
    def update_camera(self, camera_id, new_data):
        if camera_id in self.cameras:
            self.cameras[camera_id].update(new_data)
            return True
        return False # Камери з таким ID немає

    # Отримуємо інформацію про камеру
    def get_camera_info(self, camera_id):
        return self.cameras.get(camera_id, None)

    # Отримуємо список всіх камер
    def list_cameras(self):
        return self.cameras

# Приклад використання
camera_manager = CameraManager()
camera_manager.add_camera('camera1', 'Front Door', 'rtsp://camera1_url')
camera_manager.add_camera('camera2', 'Backyard', 'rtsp://camera2_url')

print(camera_manager.get_camera_info('camera1'))

camera_manager.update_camera('camera1', {'status': 'online'})
print(camera_manager.list_cameras())

camera_manager.remove_camera('camera2')
```