

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
відеоспостереження за технологією VSaaS ”

КБПЗ - 2024

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

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
« ____ » _____ 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

Кравчук Оксани Вікторівні

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS*

2. Керівник роботи *Якименко Наталія Миколаївна, канд. фіз. мат. наук, доцент*
(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 *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS*

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

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

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

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	7.12.2024 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

Кравчук О.В. Дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS. 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 відеоспостереження за технологією VSaaS.

Метою розробки є дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS.

Об'єктом дослідження є процес відеоспостереження за технологією VSaaS.

Предметом дослідження є методи відеоспостереження за технологією VSaaS.

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

Результат роботи – програмна реалізація системи відеоспостереження за технологією VSaaS.

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

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

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Delphi XE8.

Ключові слова: комп'ютерна інженерія, відеоспостереження, технологія VSaaS.

ABSTRACT

Kravchuk O.V. Research and program realization of the system surveillance technology VSaaS. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi 2024

In this final qualification work on the second (master's) level of higher education the software which is intended for system surveillance technology VSaaS.

The purpose of the development is the research and program implementation of the system surveillance technology VSaaS.

The object of the study is the process surveillance technology VSaaS.

The subject of the study is the methods surveillance technology VSaaS.

Methods of research are based on methods surveillance technology VSaaS.

The result of the work is the software implementation of the system surveillance technology VSaaS.

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

User-friendly user interface is developed. These are instructions for working with software.

The program can be used on on a computer of architecture IBM PC from OS Windows XP/Vista/7/8/10.

The program was developed in the Delphi XE8 environment.

Keywords: computer engineering, video surveillance, technology VSaaS.

ЗМІСТ

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

					ВКРМ-123.24.0004.0000.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS</i>	Лім.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Кравчук О.В.</i>					М	1	103
<i>Перев.</i>	<i>Якименко Н.М.</i>							
<i>Н.контр.</i>	<i>Коваленко А.С.</i>					<i>ЦНТУ КІ-23Мз</i>		
<i>Затв.</i>	<i>Смірнов О.А.</i>							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

DS – DirectShow мультимедійне розширення Microsoft;

TrF1 – TranformFilter – Трансформ-фільтр;

RnF1 – RendererFilter – рендер-фільтр;

SourceFilter – SourceFilter – фільтр джерела;

IC – Intelligent Connect – інтелектуальне з'єднання;

ООП – об'єктно-орієнтоване програмування;

DSP – бібліотека компонентів для розробки мультимедійних програм;

VWINDOW – вертикальна синхронізація;

HWINDOW – горизонтальна синхронізація;

PLL – цикл Блокування Стадії;

USB – Universal Serial Bus – універсальна послідовна шина;

URL – universal resource locator – локатор ресурсів інтернет;

HTML – мова розмітки гіпертекстових документів;

ОС – операційна система;

ПЕОМ – персональна електронно обчислювальна машина;

КС – комп'ютерна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Щоб забезпечити безпеку приватної власності від кримінальних посягань використовуються три види пристроїв: пристрої виявлення, фізичні перешкоди і служби охорони. Звичайно, відеоспостереження на відміну від інших, має значні переваги: здатність оцінювати територію, виявлення та фіксація кримінальних фактів на об'єкті охорони. Значним досягненням системи охоронного телебачення стало застосування в 80-х роках камер на базі ПЗЗ-матриць, що зробило реальним зменшення розмірів та вдосконалення надійності функціонування відеокамер. До того ж, ПЗЗ-камери мають високочутливі в інфрачервоному спектрі, завдяки чому можна вести спостереження в умовах абсолютної темряви. Цілком логічно, що системи охоронного відеоспостереження безперервно вдосконалюються, що й призвело до активного використання їх основними засобами охорони фізичних осіб, приміщень, територій. Найкомпактніша модель відеоспостереження це: монітор та відеокамери. Стабільний розвиток комп'ютерної індустрії сприяє й зростанню якості систем відеоспостереження. Тому сучасні системи відеоспостереження мають таку надважливу властивість: висока ефективність за умови мінімальних фінансових витрат. Саме завдяки власній унікальності відеоспостереження через Інтернет користується популярністю при організації комплексних моделей охорони. Частково це можна пояснити тим, що, використовуючи одне і те ж обладнання, можна оформити, як відеоспостереження за фізичною особою/особами, так і за значною територією корпоративної власності.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS.

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

- Огляд існуючих систем відеоспостереження за технологією VSaaS.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

другим (магістерським) рівнем вищої освіти.

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Якість картинки є головним фактором, і доки тривають обговорення розрішення 4К, конкретна задача полягає в оптимізації цієї якості для поставлених цілей відеоспостереження, навіть за найскладніших умов при мінімальному освітленні. Відповідні найсучасніші технології мають стати фундаментальними в процесі розвитку галузі.

Проте, одночасно з вдосконаленням якості картинки, стає складнішим і сам процес керування ресурсами відеоспостереження. Тобто, можуть виникати розбіжності між пропускнуою здатністю мережі та обсягом пам'яті, що, відповідно, вимагає використання досконаліших методів стискання.

Відеоспостереження як сервіс і хмарні обчислення

Відеоспостереження як сервіс (VSaaS) зручний для зберігання та архівування відеозаписів у хмарних середовищах. За допомогою систем відеоспостереження, встановлених в громадських закладах, місцях відпочинку, транспорті, можливим стало запобігання деяким правопорушенням та ефективніше розслідування скоєних. Зростання попиту на системи відеоспостереження у всіх сферах зумовлене потребою вдосконалення процесу забезпечення охорони.

Згідно з останніми дослідженнями, ринок VSaaS підніметься до 4,7 млрд доларів до 2025 року. Окремі експерти вважають хмарне середовище вирішальним фактором впливу на ринок, особливо, в корпоративній сфері. Значна частина крупних корпорацій обирають стратегію «спочатку хмара» для вирішення термінових проблем. Не зважаючи на те, що більшість цих організації інвестують в свої поточні технології, гібридне середовище пропонує елементарний та економічний шлях до застосування переваг хмари й до повного

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

переходу згодом. Останні роки термін «хмарні обчислення» є широкоживаним у сфері ІТ- фахівців, але все з часом модернізується. Чи це загальнодоступна хмара – середовище, яке за запитом розповсюджується через інтернетмережу постачальником послуг, чи приватна хмара , у якій всі ресурси зарезервовані та доступні лише одному клієнту, – хмарні технології надають кілька надзвичайно вагомих плюсів – надмірність, масштабованість і перенесення витрат з капітальних вкладень на поточні операційні витрати.

В залежності від місця розташування хмари (хостинговий ресурс чи внутрішня система), вагомою перевагою є можливість делегування технічного сервісу (апгрейди, установка відновлень і виправлень) третім особам. Значно дешевше використовувати спільну інфраструктуру, якою керує третя сторона, ніж мати такий самий масштаб налаштування всередині компанії.

Масштабованість реалізується не лише додаванням кількості камер. Вона дає можливість збільшувати обчислювальну потужність та ресурси зберігання, якщо, наприклад, необхідно змінити розрішення або частоту кадрів. Наприклад, для проведення аналізу отриманих відеоматеріалів по виявленню порушень на території громадського закладу, можна в короткі строки мати миттєвий доступ до обчислювальної потужності й фінансувати її за необхідністю.

В найбільш відповідальних інфраструктурних додатках можуть використовуватися ефективніші аналітичні прийоми, які працюють на периферії системи. Наприклад, додаткові обчислювальні потужності можна використовувати для зберігання півмільйона автомобільних номерів і фіксації номерних знаків автомобілів, що рухаються зі швидкістю 40 км/год.

Ще одна надважлива функція технології VSaaS – клієнт має доступ до додаткових сервісів відеоспостереження, це охорона чи дистанційне спостереження та доступ до відеоданих.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

1.2 Область застосування

Отже, технології вирішення аналітичних завдань все ширше використовуються в області відеоспостереження. Досить ефективними вони є для отримання потрібних даних з величезних об'ємів структурованої та неструктурованої інформації (Великі дані).

Дякуючи можливості мережевих камер здійснювати зйомку з високим розрішенням дистанційно і пересилати матеріал з будь-якої географічної точки в будь-який час, охоронні сервіси мають доступ до якісних даних з максимально доступної кількості джерел. Цілком логічним є факт, що дані поступають хаотичні (неструктуровані), тобто фото, відеоматеріали разом. То ж для адаптації до практичного використання просто необхідні інтелектуальні додатки, які розфасують за категоріями та інтерпретують для доступності.

У критичному стані «три V» Великих даних – обсяг (Volume), швидкість (Velocity) і розмаїтість (Variety) – дають можливість доставляти необхідні дані в певний час.

Маючи висновки за рахунок отриманої максимально об'ємної інформації: конкретного відеоспостереження, фізичного контролю доступу й дані про кіберактивності – і з'ясовуючи між ними логічні зв'язки, цілком реально значно мінімізувати витрати шляхом знаходження більш доречної інформації. І провідну роль тут відіграє ІР-революція, яка успішно трансформує камеру відоспостереження з приладу-помічника у розслідуваннях, в ефективний механізм попередження негативних ситуацій та злочинів.

Відеоапаратура, встановлена, наприклад, на перехрестях доріг чи аварійно небезпечних ділянках проїжджих частин, автоматично спонукає водіїв бути максимально обережними та уважними, що попереджає аварійні ситуації.

Надзвичайно вигідно мати відеоспостереження в торгових центрах, адже за допомогою отриманих відеоматеріалів та супутніх аналітичних

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

обчислювальних засобів, можна аналізувати тенденції пересування клієнтів, вибір ними товару.

Об'єднавши та проаналізувавши дані, кваліфікований фахівець надасть запропонує варіанти збільшення об'ємів продажу, підвищення конкурентоспроможності, зростання прибутків підприємства. Все це є важливими перевагами.

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

КБПЗ_2024

					VKPM-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

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

Smartec: 16 каналний NVR із записом 5 MP відео й функцією SMART-пошуку

Новий відеореєстратор Smartec STNR-1660 розрахований на роботу з 3 MP/5 MP IP-камерами через 12/4 окремих порти й надає їх живлення через вбудований PoE комутатор. Дякуючи прямому підключенню IP-камер, наявності 2 слотів для установки жорстких дисків, інтерфейсів eSATA, iSCSI і USB для підключення зовнішніх сховищ/накопичувачів і HDMI/VGA для виводу поточного/архівного відео на екран монітора, відеореєстратор 16 каналний може функціонувати як повноцінна робоча станція.

До того ж, новий і 4/8-каналні відеореєстратори цієї серії стануть ядром великої системи спостереження спільно з віртуальною матрицею STNR-6462 та іншими NVR, які мають функцію WARP. Налаштування й керування STNR-1660 і всіх його 16 каналів реально здійснювати із ПК/смартфона, застосовуючи, як варіант, ПЗ Titanium 2.0 і додаток Tive 2.0, чи через безплатний хмарний сервіс P2P.

Фешенебельний зручний корпус із замірами 378x47x317 мм дає можливість ставити відеореєстратор на столі або консолі, ще можливе кріплення його на стійку при експлуатації адаптерів. Окрім цього, сучасний відеореєстратор 16 каналний має досить просту систему керування й логічно-зрозумілий інтерфейс користувача, а також низку програмних ліній, які кардинально спрощують налаштування й сервісні роботи відеосистеми. Для високошвидкісного з'єднання з Ethernet-Мережею STNR-1660 постачений

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

гігабітним портом, а через 16 роз'ємів (Ethernet, 10/100 Мбіт/с) надає можливість прямо підключати IP-камери, а ще пускати на них живлення потужністю 130 Вт (PoE IEEE802.3af) від вбудованого 16-канального PoE-комутатора. Варто зазначити, що відеореєстратор має відеовходи HDMI/VGA і тривожні вхідні/вихідні інтерфейси 8/1, відповідно.

Запис 5-мегапіксельної картинки в H.264 відеореєстратор може виконувати по 4-м каналам із загальним фреймрейтом 120 к/с, а по іншим 12-ти максимально підтримуване розрішення запису становить 3 МР при 360 к/с, тобто по 30 к/с на канал, навіть за умови повного завантаження приладу. Незалежно від кількості використовуваних IP-камер, 16-канальний відеореєстратор підтримує паралельно запис двох відеопотоків від обох з них з високим і низьким розрішенням. Це необхідно, щоб під час перегляду архівного відео одночасно по всім 16 каналам, і місцево, і дистанційно, була можливість виводити потоки з низьким розрішенням, окрім перевантаження системи. Зазначимо, що в цій ситуації при дистанційному доступі до відео можна використовувати канали зв'язку навіть із низькою пропускну здатністю.

Сприяння модерновій технології WARP дозволяє логічно поєднувати між собою 4-, 8- і 16-канальні відеореєстратори Smartec Delta останнього покоління, для прикладу, можливість використання в межах єдиної великої чи середньої системи відеоспостереження. Дякуючи тому, що архітектура інтеграції в даному поєднанні ієрархічна, всі налаштування й функції управління доступні через визначений адміністратором Head NVR (т.зв. термінал). Наприклад, практикуючи технологію WARP, досить легко через відеореєстратор 16-канальний STNR-1660 підключитися до якогось зовсім іншого подібного пристрою, без обмежень дивитися відео й конструювати його налаштування. А також, при застосуванні WARP для регулювання роботою відеосистеми з NVR Smartec на 4, 8 і/або 16 каналів зовсім не обов'язкова зовнішня робоча станція й дороговартісне ПЗ відеоспостереження. Організацію архіву відеореєстратор здійснює на базі одного або двох локальних жорстких дисків (HDD SATA) необхідного об'єму, також

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

може фіксувати запис відеофайлів на мережне сховище, яке під'єднується шляхом інтерфейсу eSATA. Для запасного копіювання інформації можна застосовувати iSCSI і/або USB накопичувачі. Окрім цього, завдяки підтримці P2P сервісу, відеореєстратор 16організовує взаємодію з «хмарами», продуктивне використання каналів з різною смугою пропускання, високі показники відказостійкості й миттєвий доступ до налаштувань та відео від IP-камер з будь-якої частини світу. Робота з відеоархівом в STNR-1660 максимально спрощена, наявності функції SMART-пошуку, що фіксує всі об'єкти на відеозаписах від всіх 16 IP-камер, навіть не утримуючих метадані.

Штатне програмне забезпечення Titanium версії 2.0, яким оснащені всі відеореєстратори лінійки Delta, призначене для роботи в операційних середовищах MS Windows і Apple Mac і дозволяє за потребою централізовано здійснювати моніторинг до 128 відеоканалів. Доступний для пересічного користувача інтерфейс ПЗ дозволяє налаштовувати 16 каналний відеореєстратор STNR-1660 та інші додатки, використовувати матеріали відео архіву, коригувати IP-відеосистему, компонуючи будь-які з 16 камер, що обслуговуються, у зручні групи, які потім будуть застосовані для налаштування параметрів відеозйомки й реагування на позаштатні випадки. Швидкий доступ до STNR-1660 можливий через смартфон чи планшет з ОС iOS або Android з встановленим додатком Tive 2.0. Новітні 16-канальні відеореєстратори Smartec STNR-1660 з технологією WARP уже з'явилися на українському ринку та є цілком доступними для аудиторії за роздрібною ціною 590 UDS.

Wisenet Samsung: відказостійкий відеореєстратор для 64-канальної відеосистеми

Нові IP-відеореєстратори PRN-4011P марки Wisenet Samsung – це пристрої високої потужності для використання у відеоспостереженнях середнього та великого масштабу: гіпермаркети, аеропорти, стадіони, промислові підприємства.

Дякуючи можливості такого відеореєстратора забезпечувати доступ до 64 IP-камер будь-яких вендорів з кодеками H.265, H.264 і/або MPEG-4, а також

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

зовнішніх охоронних датчиків і виконавчих пристроїв, а через аудіоінтерфейси надає можливість слухати й записувати звук по всім 64 каналам від камер з мікрофоном.

Сучасний 64-канальний відеореєстратор PRN-4011P марки Wisenet Samsung вже можна придбати на український ринку за роздрібною ціною 6 166 USD.

AXIS: тепловізор для цілодобового відеоконтролю на відстанях до 5,6 км

У палітрі товарів компанії Axis Communications присутні камери Q 1942-E і Q 1942-E PT Mount, які виконують тепловізійну відеозйомку за будь-яких погодних умов та освітлення з розрешенням до 800x600 пікс. при 8,3 к/с, передавають звук, підтримують відеоаналітику, технологію Zipstream та електронну стабілізацію картинки.

Кожен тепловізор має сенсор із чутливістю менше 50 мк, об'єтив на 10, 19, 35 або 60 мм і доставляється в закритому від зовнішніх факторів термокожусі (IP66/IP67 і NEMA). Маючи здатність виявляти тепловипромінюючі об'єкти на відстані (до 5,6 км), ці тепловізори максимально ефективні для безперервного відеоспостереження в приміщеннях та зовні на архіважливих об'єктах, а управління ними в межах IP-відеосистем забезпечується засобами як фірмового, так і якогось іншого ONVIF-сумісного ПЗ.

При резервування камери-тепловізора AXIS клієнт має змогу обрати модель із одним з 4 видів оптики, що визначається площею охоронної території. Так, дальність фіксації людини/транспортного засобу в AXIS Q 1942-E може сягати 1 833/5 623 м, а в Q 1942-E PT Mount – 1 774/5 441 м, то ж, при експлуатації 60-міліметрового об'єктиву, за умови потреби меншої дальності бачення, буде доречним тепловізор з об'єктивом на 35 мм (17°), 19 мм (32°) або 10 мм (63°). Аналогічно IP-камерам AXIS видимого діапазону, тепловізори нового покоління можуть мати живлення за технологією PoE, і також цілком поєднані із джерелами живлення напругою 8-28 V DC або 20-24 V AC.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

живлення при налаштуванні. За потреби, тепловізор Q 1942-E, на відміну від Q 1942-E PT Mount, можливо доукомплектувати опціональним кріпленням для монтажної коробки (T98A 17-VE) або для опори із круглим перетином (T91A47). Для сталої роботи на зовнішньому об'єкті сучасні тепловізори підтримують електронну стабілізацію зображення (EIS) .

Bosch Security: 16 каналний IP відеореєстратор з підтримкою 8 MP/H.265 камер

Модернізовані настільні IP відеореєстратори DIVAR network 2000 компанії Bosch Security Systems призначені для прийому відео від 16 камер відеоспостереження, у тому числі й поворотних, мають здатність проводити запис відео з розрешенням до 4K Ultra HD і загальним вхідною бітрейтом до 256 Мбіт/с. Маючи адекватну вартість, кожен 16-каналний IP відеореєстратор цієї лінії максимально підходить для організації систем відеоспостереження в малих магазинах, офісах, костелах, освітніх закладах, а за рахунок підтримки відеокодека H.265 у поєднанні з передовими мережними протоколами – гарантує найвищий ступінь надійності й безпеки інформації. Доналаштувати відеореєстратор і керувати відеопотоками по кожному з 16-ти каналів можна, використовуючи функціональний мобільний додаток DIVAR Viewer App, сконструйований фахівцями Bosch.

Лінійка пристроїв DIVAR network 2000 об'єднує різні за базовим оснащенням відеореєстратори. Моделі DDN-2516-212Nxx постачаються з перед вбудованим 2-терабайтним накопичувачем, DDN-2516-112Dxx – з DVD-приводом, а 16-ти каналний IP відеореєстратор DDN-2516-200N00 підтримує роботу з 2-терабайтними жорсткими дисками, для розташування яких він має дві «посадкових» позиції. Зазначимо що, крім одного з HDD SATA цей NVR допускає встановлення приводу для DVD дисків. Варто відмітити, що всі відео реєстратори оснащені PoE комутатором на 8 або 16 каналів (залежно від модифікації) і забезпечують безперервну експлуатацію IP камер потужністю споживання до 30 Вт.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Функція запису 4K Ultra HD відео по кожному каналу дає можливість робити на базі Bosch DIVAR network 2000 системи відеоспостереження ультрависокого розрішення. Тому новий 16-ти каналний IP відеореєстратор може забезпечувати сервісом IP-камери як Bosch, так і інших виробників з якістю картини різних стандартів, у тому числі HD 720p і HD 1080p (Full HD), надаючи швидкість передачі даних по мережі в діапазоні від 16 Кбіт/с до 24 Мбіт/с. Через інтерфейси VGA і HDMI по будь-якому 16-ти каналному пристрої серії реально підключити монітор для організації локального робочого місця оператора й переглядати відео в 4-, 9- і 16-віконному режимі або по черзі від кожного каналу.

Фіксувати та демонструвати відео від IP камер сучасні відеореєстратори Bosch можуть не лише в H.264 і M-JPEG, але й у сучасному форматі H.265. Застосування H.265 надзвичайно полегшує компресію відеосигналу, особливо ультрависокого розрішення, дякуючи чому 16-ти каналний IP відеореєстратор серії DIVAR network 2000 буде зберігати відео ідеальної якості, «забираючи» при цьому мінімум дискової території. Наприклад, порівняно з H.264 алгоритм H.265 дає змогу економити до 50% мережного трафіку й місця на HDD, і також забезпечує передачу потоків в 4K Ultra HD по 16 IP каналах з малою пропускнуою здатністю.

Різні варіанти налаштування й управління DIVAR network 2000 можуть бути проведені через спеціальний додаток DIVAR Viewer App, розроблений компанією Bosch для платформ Android або iOS. Цей додаток робить можливим конфігурувати 16-ти каналний IP відеореєстратор і проводити дії з поточним, архівним відео з будь-якої точки світу, управляти панорамуванням, нахилом, масштабуванням підключених до NVR поворотних IP камер у режимі поточного часу та здійснювати безліч різноманітних дій. Окрім всього, 16-ти каналний відеореєстратор дає доступ до відео й зі стаціонарних ПК через ПЗ Bosch Video Client чи інші ONVIF-сумісні програми.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Разом з підтримкою відеоаналітики IP камер Bosch, включаючи фірмові системи Essential Video Analysis (EVA), Intelligent Video Analysis (IVA) і Motion+, новітні IP відеореєстратори вмiють «збирати» дані про тривоги від зовнішніх охоронних датчиків (до 4-х). При фіксуванні тривоги 16-ти канальний IP відеореєстратор може направити юзерам інформацію по електронній пошті й/або на FTP-сервер, активувати 1 або 2 виходи тривоги, включити зумер і вивести повідомлення на екран монітору. Для свідчення дійсності запису по 16-ти каналам використовуються «водяні знаки» (Watermake), а для відтворення захищених відеофайлів – програма Archive Player.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero® Delphi XE8 – це миттєве вирішення для розробки програмного забезпечення на Delphi та C++, що пришвидшує проектування, програмування й формування інтегрованих додатків для Windows, Mac, iOS, Android і IoT. Надає можливість вирішувати проблеми корпоративного класу, залучаючи до себе native-платформи користувачів, мобільні розширення, інтелектуальні пристрої, хмарні сервіси, корпоративні й вбудовані бази даних. FireUI пришвидшує розробку клієнтських інтерфейсів для різноманітних форм-факторів на базі однієї форми. Наявні додатки реально доукомплектувати новими можливостями IoT, включаючи інформування про наближення. Кінцеве рішення Enterprise Mobility Services надає доступ до корпоративних баз даних та можливість самостійної публікації створеної REST API з аналітичними можливостями.

Embarcadero® Delphi XE8 Architect – це швидке вирішення для розробки програмного забезпечення на Delphi, що пришвидшує проектування, програмування й формування інтегрованих додатків для Windows, Mac, iOS,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Android і IoT, які співпрацюють з корпоративними СУБД. Delphi Architect містить середовище візуального моделювання даних ER/Studio Developer Edition, що дозволяє розроблявачам максимально функціонально застосовувати корпоративні структури й бази даних.

Embarcadero® Delphi XE8 Ultimate – це швидке вирішення для розробки програмного забезпечення, орієнтоване на розробників ПЗ та робочих груп, які здійснюють створення, адміністрування й налаштування взаємодіючих додатків, що активно співпрацюють з базами даних та інтегрованими з корпоративними СУБД. Delphi XE8 Ultimate підтримує всі можливості редакції Enterprise, а також включає засоби розробки, налаштування й профілювання коду SQL і засоби управління трансформаціями в базі даних, здійснювані DB PowerStudio® (редакція Developer).

Embarcadero® Delphi XE8 Enterprise – це миттєве вирішення для розробки програмного забезпечення, сформоване для самостійних та корпоративних розробників. Забезпечує створення клієнт-серверних і багаторівневих взаємодіючих додатків, що під'єднуються до масштабного кола корпоративних баз даних та хмарних площадок, маючи в складі Microsoft SQL Server, DB2, Oracle, Sybase, InterBase, Amazon і Microsoft Azure. Delphi Enterprise підтримує всі види редакції Professional, а ще забезпечує підключення до корпоративних даних використанням DataSnap SDK і має у складі ліцензію розроблявача на Enterprise Mobility Services.

Embarcadero® Delphi XE8 Professional – це миттєве вирішення для розробки програмного забезпечення, створене для швидкого проектування, шифрування й розширення взаємодіючих додатків для Windows, Mac, iOS, Android і IoT. Допомогає створювати нові додатки для кількох платформ у межах єдиного проекту розробки, використовуючи багаточисельні можливості мови Delphi.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Пакет Mobile Add-On Pack для Delphi XE8 Professional збільшує функції цього середовища, дозволяючи винаходити підключені мобільні додатки для Android і iOS з однієї кодової бази Delphi. Потрібно: Delphi XE8 Professional.

Пакет FireDAC Client/Server Add-On Pack здійснює клієнт-серверне доєднання до баз даних і гарантує додаткові корпоративні бази даних в Delphi XE8 Professional. За рахунок унікальної та високоефективної архітектури FireDAC дозволяє прямий доступ додатків до Oracle, SQL Server, InterBase, DB2, Firebird, SQLite, MySQL, PostgreSQL, SQL Anywhere, Advantage DB, Access, Informix, DataSnap і т.д. Включає ліцензію розроблювача на Enterprise Mobility Services.

Delphi® XE8 Starter – це вдали варіант стартувати конструювання високопродуктивних додатків для Windows на Delphi. У комплект Delphi Starter включене полегшене інтегроване середовище розробки (IDE), інтенсивні модулі інтерфейсу користувача, що завантажуються, Microsoft ISAPI Delphi (32- і 64-розрядні), інтегрований відладник, двосторонні візуальні конструктори для пришвидшення проектування, сотні візуальних деталей та лімітована ліцензія на фінансове застосування. Delphi Starter не включає елементи баз даних і драйвери.

Повний список нових можливостей Delphi XE8

Інтегровані компілятори й набори:

- Компілятор і комплект інструментів Object Pascal для 64-розрядної iOS.
- Підтримування універсальних додатків iOS (включення 32- і 64-розрядних двійкових файлів у комплект додатка).

Платформа додатків FMX (FIREMONKEY):

- Передперегляд FireUI на мобільних платформах.
- Передперегляд на різному обладнанні.
- Розгорнуті перспективи роботи зі стилями в дизайнері єдиної панелі управління, включаючи універсальні стилі й налагодження стилів Windows.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

- Диспетчер пристроїв FireUI дає можливості міняти специфікації відкритих пристроїв та використовувати нові.
 - Елементи керування Мето для платформи iOS.
 - Елементи керування ListView для платформи iOS.
 - Елементи керування Calendar для платформи iOS.
 - Елементи керування Switch для платформи iOS.
 - Компонент MapView для мобільних платформ (iOS і Android), який підтримує спеціалізовані бібліотеки інтерактивних карт для кожної платформи (розроблені відповідно Apple і Google).
 - Компонент WebBrowser для Windows.
 - Компонент WebBrowser для OS X.
 - Удосконалений комплект компонентів платформи й компонентів, що настраюються індивідуально, включаючи нові варіанти стилю для iOS і покращена робота з Z-порядком (властивість ControlType для нестандартних елементів керування).
 - Компонент ImageList для FireMonkey.
 - Вдосконалена підтримка мультимедіа в FireMonkey.
 - Підтримка вищого розрішення в деталях камери.
 - Автоматично включений стиль Android Lollipop (тільки для Android).
 - Підтримка варіантів аналізу додатків FireMonkey. Для збереження даних аналізу, отриманих під час роботи, та доступу до них необхідний відповідний контракт із оплатою в залежності від обсягу даних.
 - Розширення й доповнені протоколи для зв'язку між додатками.
 - Вдосконалена платформа Bluetooth і новий компонент TBluetooth для стандартного Bluetooth.
 - Підтримка наближення з використанням технології маяків (у тому числі стандарти iBeacon і AltBeacon) для iOS, Android і OS X.
- Бібліотека візуальних компонентів (VCL):
- Удосконалене масштабування шрифтів VCL.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

– Підтримка інструментів аналізу додатків VCL. Для збереження даних аналізу, отриманих під час роботи, і доступу до них необхідний відповідний контракт із оплатою в залежності від обсягу даних.

– Розширення й доповнені протоколи для зв'язування додатків VCL.

– Вдосконалена платформа Bluetooth і новий компонент TBluetooth для стандартного Bluetooth.

Характеристики мови й робота з бібліотеками області виконання:

– Заголовки API для 64-розрядної архітектури iOS і інтеграція з RTL для Delphi і C++.

– Фізичний движок Vox2D.

– Інтерфейси Object Pascal для фізичного движка Vox2D.

– Новий модуль RTL System.Hash з геш-значеннями для підтримки нової платформи HTTP.

– Нова клієнтська платформа HTTP, що включає в себе еквіваленти для бібліотек на всіх підтримуваних платформах.

– Нові компоненти NetHTTPClient і NetHTTPRequest, що дають прямий доступ до клієнтської платформи HTTP.

– Підтримка власних клієнтських бібліотек HTTP/HTTPS в Windows.

– Підтримка власних клієнтських бібліотек HTTP/HTTPS в OS X.

– Підтримка власних клієнтських бібліотек HTTP/HTTPS на мобільних платформах (iOS і Android).

Особиста продуктивність розроблювачів:

– Диспетчер бібліотек GetIt для комфортного пошуку, завантаження й відбудови бібліотек вихідного коду, деталей та інших ресурсів із серверів Embarcadero GetIt.

– Засіб переносу параметрів конфігурації для переносу конфігурації зі попередньої версії продукту в нову, а також між різними екземплярами однієї версії.

– Вдосконалена початкова сторінка.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

– Комплексна підтримка бази даних SQLite версії 3 і вище, включаючи "інтелектуальне" розпізнавання видів даних, підтримка динамічних і статичних посилань.

Сервіси ENTERPRISE MOBILITY SERVICES:

– Вдосконалена інтеграція FireDAC/EMS, основним чином дотичного управління установкою відновлень.

– Підтримка сервера push-повідомлень EMS для iOS і Android.

– Підтримка зовнішніх облікових даних в EMS.

– Розширений адміністративний API EMS в даний час підтримує установку EMS і push-ресурси EMS.

– Пул підключень до бази даних і інші способи оптимізації EMS.

– Новий елемент EMSClientAPI, який оптимізує розробку клієнтської частини EMS.

– Способи аналізу й складання звітів по юзерах, групах, сеансах і викликах API через консоль EMS з веб-інтерфейсом.

– Змога експорту даних з консолі EMS в CSV-Файли.

– Розширена аналітика для користувачів і груп.

– Клієнтський додаток EMS для управління обліковими записами юзерів.

З'єднувачі DATASNAP:

– Мобільні з'єднувачі DataSnap для Objective-C з підтримкою SDK для iOS 7 і iOS 8.

Включені бази даних:

– InterBase XE7 Developer Edition – до 20 користувачів і 80 логічних підключень.

Ліцензування й управління засобами:

– У мережних ліцензіях підтримуються попередні версії (Delphi 2007-XE7, C++Builder 2007-XE5, HTML5 Builder XE3, RadPHP XE2). Ліцензії на Delphi 7 і C++Builder 6 отримуються окремо.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

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 відеоспостереження за технологією VSaaS.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

За очевидними причинами якість картинки в охоронному відеоспостереженні не треба плутати з якістю рівня 4K або HD при художній фотозйомці чи сфері розваг. Ось, для розуміння, можливість записати чітке зображення за умова вкрай мінімального освітлення є архіважливим показником для збереження відповідного контролю за подіями, які відбувається навколо. Висока якість картинки надається засобами шумозаглушення (за останній рік їхня ефективність стрімко виросла) у поєднанні з досконалою світлочутливістю відеодатчиків камер.

Теперішні мережні камери надають можливість спостерігати деталізовані кольорові картини навіть в нічний час. Найскладнішою задачею є збереження передачі кольорів в умовах відеозйомки без штучного освітлення. Дякуючи модернізованим світлочутливим відео датчикам, якісна мережна система відеоспостереження може передавати подібне зображення.

Наступним результативним способом підвищити якість зображення в абсолютній темряві є інфрачервоне (ІЧ) підсвічування. У надмірно вузькому коридорі підсвічування всередині з'явиться так звана біла імла чи відблиски, а деякі області будуть освітлюватися некоректно. З іншої сторони, при освітленні дуже великої області виявляються яскраво помітні й сторонні об'єкти, а дальність спостереження стає коротшою. В залежності від моделі пристрою площа охоплення інфрачервоним підсвічуванням може перевищувати 40 м при мінімальному рівні енергоспоживання (електроживлення постачається звичними засобами Power over Ethernet). Це стане корисним організаціям, які турбуються про мінімалізацію енергозатрат з міркувань економії, і також, впливу на екологію.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Наприклад, після регулювання сектора огляду камери, обладнаної дистанційним управлінням зумом та оптимізованим інфрачервоним підсвічуванням, кут підсвічування автоматично підлаштовується під рівень масштабування відеокартинки. Кут підсвічування потрібно ставити за фокусуванням камери таким чином, щоб на зображення у фокусі подавалась найбільш можлива кількість світла. Наступна характеристика автоматичного підстроювання нової технології оптимізованого інфрачервоного підсвічування яскраво визначається, коли об'єкт спостереження знаходиться на великій відстані від камери в освітленій ділянці спостереження. По мірі приближення об'єкта до камери експозиція автоматично міняється. Об'єкт, наближаючись до камери, стає достатньо освітленим, але без переекспозиції.

Технологія максимально динамічного діапазону (WDR) розбирається з найрізноманітнішими умовами освітлення. Рядова камера не може надати чітке відображення темних об'єктів там, де є і надзвичайно світлі, так і затемнені площини чи світло падає позаду, – наприклад, якщо особа знаходиться на фоні сонячного вікна. Технологія WDR гарантує найкращу видимість і добре висвітлених, і затемнених об'єктів, що містяться в зоні прегляду.

Наступна функція технології WDR – захоплення зображення для використання в криміналістичних експертизах. Багато хто вважає її ключовою з огляду гарантії якості картинки в абсолютній темряві, оскільки вона оптимізує відеозйомку в інтересах криміналістичної експертизи, максимально підвищуючи деталізацію як у затемнених, так і в яскраво освітлених місцях сектору спостереження. Довгі роки наукових досліджень і проектно-конструкторських напрацювань надали позитивні результати – сьогодні клієнти можуть скористатися перевагами сучасних алгоритмів стиску відеоданих для оптимізації якості зображення, включаючи можливість плавного переходу з технології WDR на зйомку в умовах слабого освітлення й навпаки.

Всі ці досягнення являють собою знаменні віхи на шляху до бездоганної якості зображення навіть в умовах повної темряви.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

експлуатації даних. Останні, у свою чергу, можуть стати джерелом перспективних ідей не лише в сфері безпеки, але й для збільшення ефективності й отримання козирів у бізнесі. Це бажано брати до уваги всім, хто зацікавлений у впровадженні прогресивних рішень у сфері відеоспостереження.

CCTV

CCTV (система відеоспостереження за об'єктом охорони) – це набір обладнання, яке надає можливість використовувати камери системи безпеки для збору відеоінформації про об'єкт, зображення з яких поступають на єдину спільну систему спостереження. Камери й комплексні рішення по відеоспостереженню на сьогодні стають все більш популярними за рахунок падіння на них вартості, цим же стають максимально доступними. Дякуючи їм, клієнти можуть відчувати себе комфортніше, можуть бути впевненими, що їхнє майно захищене, чи то невелика комерційна діяльність, чи масштабна, чи інша приватна власність. Дана інформація дозволить дізнатися про особливості роботи відеоспостережень та зробити вибір при покупці.

Визначте завдання для вашої системи спостереження

Головна потреба спостерігати за тим, хто приходить і йде? Чи необхідність спостереження за об'єктом у цілодобово? Необхідний дистанційний доступ до відеоінформації? Важливий відеоархів? Хочете спостерігати за працівниками, товарами, бачити загальну картину? Є умови для провідної чи лише для бездротової системи відеоспостереження? Маєте намір встановити камери всередині приміщення чи зовні? Яка кількість камер потрібна? На який бюджет розраховуєте?

Оберіть тип камери

В залежності від мети та місця встановлення камери, потрібно обирати таку, яка буде максимально підходити за всіма параметрами. Одним із основних критеріїв є – провідна чи бездротова. Плюсами бездротових відеокамер є: простота встановлення бездротового пристрою, відсутня потреба в прокладці сигнального кабелю, такі камери виділяються своєю мобільністю, їх можна

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

переміщати по всій території. Мінуси: бездротові відеокамери функціонують на радіохвилях, то ж цілком логічно, що будь-які радіоперешкоди чи перебої можуть знижувати або ж взагалі зупиняти роботу. Бездротова камера отримує живлення не через мережу електропостачання, а від аккумулятора, тому для забезпечення безперебійного функціонування, потрібно контролювати рівень заряду. Як приклад, IP-камера D-Link DCS-2230 FullHD Cube, WiFi 802.11n.

Провідні камери відеоспостереження

Підключаються в систему безпеки за допомогою проводів. На сьогодні найпопулярнішим видом камер є – IP камери. Щоб поєднати їх із системою спостереження застосовується звичайна кручена пара, а методом використання технології POE постачання на камеру поступає теж по кручений парі. Це дає підключитися камері до вже наявної домашньої мережі чи мережі закладу, і аналогічним способом отримати доступ до них через Інтернет. Для правильного встановлення провідних камер відеоспостереження найкраще звертатися за допомогою до спеціалістів, оскільки зробити це вірно й продумано без відповідної кваліфікації та досвіду практично не можливо. IP-камера Panasonic Dome 1280x720 60fsp SD IR LED PoE. IP-камера Panasonic HD Fixed Dome network Wide coverage Horizontal camera 1280x960 PoE. IP-камера Panasonic HD Pan-tilting Network Camera 1280x720.

Основною та вагомою перевагою провідних відеокамер відеоспостереження є те, що вони не потребують постійного обслуговування. Також, можуть досить просто монтуватися у важкодоступних місцях всередині й поза приміщеннями. Провідні камери значно менше піддаються зовнішнім впливам та перешкодам. Є й низка недоліків. Встановлені не професійно та всупереч вимогам, можлива нестабільна робота, неякісна картинка та інші проблеми. Також логічно, що провідні камери не можна рухати. Тому обирати місце для монтажу відеокамери потрібно скрупульозно, бо змінити його потім буде достатньо складно.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

із загальної картини спостереження, проте, якщо у вашої системи відеоспостереження немає оператора або прямої потреби охоплювати більшу територію онлайн, не варто купувати PTZ камеру. Відмінне рішення IP-камера Panasonic Weatherproof HD network PTZ camera 1280x960 PoE Plus.

Якість зображення

За нинішніми розцінками, рекомендується купувати IP камери з якістю запису картини менше 720 p. Запис вищої якості просто дасть загальну картину про ситуацію. Якщо потрібно бачити, хто приходив за час вашої відсутності, у чому був одягнений, куди дивився, за необхідності – роздрукувати фото відвідувача. В цьому випадку економити вже не варто: чим вищою буде якість запису, тим чіткішим буде зображення. Відповідно, можна детально розглянути необхідні моменти та ситуації. Бюджетніший варіант IP-камера D-Link DCS-6112V. Чи більш просунута IP-камера Panasonic Full-HD Fixed Dome network wide coverage Horizontal camera 1920x1080 PoE.

Що робити з отриманим зображенням?

Залежить від кінцевої мети. Можна зберігати лише важливі чи потрібні події, можна зберігати відеоархів за якийсь конкретний період, а можна взагалі нічого не зберігати, а просто продивлятися все в режимі он-лайн. Найпримітивніше, встановити камери з SD картами та зберігати важливі події лиш там. Та все ж, якщо стоїть мета: зберігати дані, то найкраще для цього спеціалізований пристрій – відеореєстратор. Відрізняються вони за різними характеристиками: кількість підтримуваних пристроїв, об'єм збережених даних, спосіб підключення й управління. Для прикладу, IP-відеореєстратор Panasonic Network Disk Recorder Full HD. Запис 16 HD камер в поточному часі, встроєна функція розпізнавання осіб, підтримка розрішення запису до 1920x1080, швидке налагодження роботи без ПК, є вихід для підключення Full HD монітора, управляється за допомогою миші, підтримує технологію Onvif. Об'єм і тривалість збережених даних залежить від об'єму встановленого накопичувача. Для цього пристрою бажано організувати прихований монтаж та по максимально обмежити

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

кількість осіб, обізнаних щодо місця його дислокації на охоронному об'єкті. Це гарантує збереження даних за будь-яких непередбачуваних ситуацій.

3.2 Розробка структурної схеми

IP відеокамери, IP системи відеоспостереження стають все більш поширеними. І цілком спрогнозовано, IP відеосистеми витісняють аналогові. Переваги цілком передбачувані та зрозумілі: високе розрішення, картинка чіткіша в 4-16 разів в порівняно з аналоговими камерами, широкі можливості програмного забезпечення: визначення номерів, розпізнавання осіб, встановлення пожеж, ідентифікація поведінки в натовпі, паніка, бійка, виявлення залишених предметів, стабільне спостереження за об'єктом, інтелектуальне управління поворотними відеокамерами та багато іншого. При цьому вартість IP відеокамер приблизно така ж, як аналогових. І на сьогодні вже зрозуміло, що, обираючи систему відеоспостереження, значно вигідніше встановлювати IP відеоспостереження.

Переваги IP відеосистем

IP система відеоспостереження, подібно до аналогової має в складі: відеокамери, середовище передачі даних (відеосервер), пристрої відображення. Але відрізняється від аналогових систем можливістю використання кількості IP відеокамер: від 1-ї до кількох сотень. Ще один величезний плюс – ймовірність одночасного використання відеокамер з різними параметрами, різними дозволами матриці, від різних виробників. Таким чином, на головні точки спостереження раціонально встановити камери високого розрішення (2-3 мегапікселя), а в інших: проходах, коридорах, по периметру, цілком достатньо буде відеокамер з розрішенням 1,3-2 мегапікселя, які, відповідно, значно дешевші.

Середовище передачі даних IP відеосистем.

На відміну від аналогових, де від кожної камери тягнеться окреме проведення, дальність якого досить обмежена і з відстанню сигнал погіршується

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

чи зникає, а чіткості немає, в IP системах відстань не має значення й може сягати від 10 метрів до безкінечності. При цьому якість якості та чіткість картинки залишаються високими, бо сигнал подається в цифровому вигляді. Чисельність відеокамер у системі на відміну від аналогових – не обмежена. Всі блоки IP відеосистеми можна поєднувати в одну систему. При цьому IP відеокамери можуть бути зовсім різними за характеристиками, марками, типами. Не зважаючи на це, вони злагоджено працюють в одній системі. Це стає можливим дякуючи використанню спільних стандартизованих протоколів передачі даних та гнучкості програмного забезпечення.

Віддалений відеоархів або відеосервер

Наступна перевага IP систем відеоспостереження – це можливість розміщення відеоархіву віддалено. Відеокамера може знаходитись на об'єкті, а фіксуючий пристрій абсолютно в іншому місці, можливо, засекреченому. Цілком реально подвійне та потрійне синхронічне збереження відеоданих з однієї відеокамери. За умов критичних ситуацій, якщо раптом один з серверів вийде з ладу, другий чи третій збережуть всю інформацію. Таким чином, можна зробити висновок, що IP відеоспостереження значно надійніше за аналогове та більше відповідає стандартам безпеки.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

загальної довжини проводки, від її якості та від загальної схеми розгалуження мережі.

Пропускна спроможність – це не лише кількість Мбіт/сек, це також пінг, максимальна кількість синхронно працюючих пристроїв: відеокамер, відеосерверів, перетворювачів і т.д.

«Відеоаналітика як сервіс» знаходиться на перехресті цих двох ринків і фіксує фокус на відеоаналізі без залучення оператора, в той час як термін VSaaS нині часто припускає лише ймовірність віддаленого перегляду й запису без будь-якої аналітики.

Тут аналізується стан ринку й технологій VSaaS, а також розглядається варіант вирішення основної проблеми розвитку ринку VSaaS – обмеження вихідного каналу абонента – методом використання відеоаналітики.

Як наголошує IMS Research, зростання ринку VSaaS зумовлено попитом зі сторони одноосібних користувачів, малого та середнього бізнесів, й держави. Для певної частини юзерів послуга VSaaS є привабливішою, ніж звичні рішення на базі мережних відеореєстраторів (NVR) й систем керування відео (VMS). Так, модель комерціалізації VSaaS допускає, що замість ціни апаратно-програмного рішення без гарантій повернення інвестицій клієнт сплачує одну конкретну послугу, наприклад, запис відео, автоматичний виклик служби охорони, збір відомостей, підготовку аналітичних звітів. Послуга VSaaS – добре масштабована з погляду об'єму збереженого відео, кількості точок спостереження й чисельності споживачів.

На сьогодні зафіксовано біля 30 компаній, які активно займаються розвитком VSaaS, і їхня кількість постійно зростає. Цільовими ринками таких компаній є мережі роздрібних продажів, будівельники, малі офіси й домашні господарства.

Серед найкрупніших можна відмітити компанію Axis, вона успішно розробила інфраструктуру Axis AVHS для надання послуг VSaaS через

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

посередників. Стримуючим фактором Axis AVHS є те, що сервіс VSaaS працює лише з мережними камерами, виготовленими Axis.

В Україні лиш декілька компаній вийшли вихід на ринок VSaaS. Наприклад, вітчизняна компанія iVideon надає комфортний та зрозумілий сервіс відеоспостереження через Інтернет, компанія Profinegro активно впроваджує послуги бізнес аналітики в торгівельні мережі. Компанія DSSL анонсувала сервіс Trassir Cloud для автоматичного моніторингу систем відеоспостереження на першому етапі та певні плани розвитку спектру хмарних послуг, безпосередньо пов'язаних з відеоспостереженням.

Отже, на ринок VSaaS виходять компанії таких типів:

- молоді компанії (стартапи), сконцентровані на певних секторах ринку;
- виробники систем відеоспостереження (VMS);
- виробники камер;
- оператори зв'язку.

Найбільшою перешкодою масштабного поширення послуг VSaaS у світі й, зокрема, в Україні є невисока пропускну спроможність каналів зв'язку за межами локальної мережі. Порівняємо потоки даних, створені камерами, із середніми швидкостями підключення абонентів.

З однієї сторони, навіть при застосуванні сучасних алгоритмів стискання, таких як H.264, камери стандартної чіткості (0,4 мегапікселя) формують потік даних від 0,5 до 4 Мбіт/с, а камери високої чіткості (13 мегапікселя) від 1 до 10 Мбіт/с за хороших спостереження й до 50 Мбіт/с – при гірших. Для систем з максимальною кількістю камер затрати на передачу, зберігання й аналіз інформації стають критичними.

З другої сторони, середня швидкість вихідного каналу відповідає 4 Мбіт/с у світі й 13 Мбіт/с в Україні за даним NetIndex. Під час застосування несиметричних технологій доступу (наприклад, LTE, 4G, ADSL чи кабельний модем) вихідний канал від абонента до додатка VSaaS в 410 разів менший вхідного. Таким способом, послуга VSaaS не дає можливості віддаленого

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

перегляду та запису відео з більшої кількості камер, особливо, якщо це камери високого розрішення (більше 1 мегапікселя).

За нашою оцінкою, щорічний ріст об'єму відеоданих, отриманих камерами, є більшим за 50%, а збільшення пропускної спроможності каналів лишень 20%. Саме тому, відстань між потребами VSaaS та перспективами каналів зв'язку кожного року зростає на 30%.

Відеоаналітика – єдина технологія, яка здатна вирішити проблему вихідного каналу абонента, і також проблему зберігання відео в хмарі. Хоча є досить економічні способи зберігання відео, такі як LAID (LinearArrayofIdleDisks), збереження більших об'ємів відео в хмарі є максимально затратною складовою послуги VSaaS.

Відеоаналітику можна кваліфікувати спеціалізованим кодером, що залишає у відео виключно дані, які потребує користувач. Всеосяжний кодер, такий як H.264, «не розуміє» міру важливості кожного елемента на картинці, тому не здатен продуктивно фільтрувати зайву інформацію для забезпечення послуги VSaaS. Для прикладу, шаблонний кодер не в змозі відрізнити мініатюрну фігурку людини на дальньому плані та багаточисельні краплі дощу на ближньому. Якщо фігура й кожна крапля кодується з аналогічною деталізацією, то потік буде у вагомій мірі надмірний для передавання й збереження.

Зазвичай, універсальний кодер та відеоаналітика застосовуються синхронно, саме це і надає можливість використати всі плюси кожного.

Логічно, що для зменшення завантаженості каналів зв'язку, відеоаналітика має функціонувати на боці абонента. До того ж, для окремих видів відеоаналітики, як визначання осіб, потребується відеоаналіз нестиснутого потоку для здобування максимально точних показників. Через ці причини хмарна інфраструктура не може бути економічно ефективно застосована для попереднього опрацювання відео без певного обладнання на боці абонента.

З другої сторони, хмарну інфраструктуру можна результативно застосовувати для масштабування системи відеоспостереження в таких вимірах:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- збереження відео й метаданих відеоаналітики;
- під'єднання нових об'єктів спостереження (це можуть бути торговельні точки);
- запровадження нових можливостей аналізу метаданих і розшукування в архіві;
- надання послуг сервісу для значного числа користувачів.

Чим вищою є точність відеоаналітики, тим меншим стає навантаження на лінії зв'язку й хмарне сховище. Отже, коли показники точності відеоаналітики відомі, надавач послуги чи користувач мають можливість досить швидко та без затрати особливих зусиль прорахувати економічний прибуток від застосування відеоаналітики.

Для прикладу. Для охорони території грандіозної сонячної станції (геліотермічної установки) встановлено 300 відеокамер високого розрішення (1,2 Мп). За стабільних погодних умов сумарний потік даних становить 1,8 Гбіт/с. За поганих умов, тобто, якщо сигнал приглушується, наприклад, після заходу сонця, потік піднімається майже у два рази до 3,5 Гбіт/с. Використання стандартного детектора руху дає можливість мінімізувати об'єм відеоданих у середньому на 80%, тобто до 0,40,7 Гбіт/с.

Надзвичайно прикро, але пікове навантаження часом у багато разів перебільшує цей діапазон тому, що детектори руху відгукуються на тотальні зміни освітлення й погодних умов одразу на усіх камерах.

Застосування професійної відеоаналітики для фіксації осіб на площі сонячної станції дало можливість мінімізувати середню завантаженість до 0,010,02 Гбіт/с, а максимальну – до 0,05 Гбіт/с. Таким способом, порівняно із звичним детектором руху, відеоаналітика дала можливість мінімізувати навантаження на лінію зв'язку та хмарне сховище більше ніж в 40 разів.

Місцеве сховище відео на флеш-карті в камері чи на жорсткому диску в сервері-шлюзі (gateway) на боці споживача надає можливість продуктивніше задіювати вихідний канал за рахунок буферизації інформації. При створенні події

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

відеоаналітики камера чи локальний сервер-шлюз фіксує відео в місцеву пам'ять, що потім транспортується в хмарне сховище по звичному каналу зв'язку. При подібному режимі постійного запису швидкість записування може бути вищою за граничну швидкість передачі відеосигналів.

Місцеве сховище є також джерелом гарантії відказостійкості системи відеоспостереження на обставини з тимчасовим відключенням каналу сполучення між камерою (локальним сервером) та хмарним додатком.

Зразком більше складного способу використання відеоаналітики в складі додатка VSaaS є технологія ранжирування відеосигналів. В цьому випадку транспортування відео по джерелах з'єднання з лімітованою пропускнуою спроможністю досягається не за рахунок пониження якості відео, як у звичайних відеореєстраторах, а за рахунок пріоритетної передачі важливішої для клієнта інформації, причому пріоритет фрагментів відео автоматично визначається відеоаналітикою. Цей підхід також збільшує продуктивність застосування дискового простору хмари: період зберігання відеоінформації в місцевому сховищі чи хмарі може залежати від міри їх значення.

На завершення розберемо головні відмінності між класичною системою адміністрування відео (VMS) і хмарним додатком для забезпечення послуги VSaaS:

– VSaaS допускає застосування значно складніших технологій для управління каналом зв'язку між об'єктом слідкування й хмарою, і також між хмарою та абонентом. Важливо брати до уваги ліміт по швидкості передачі, і ще присутність складних топологій із трансляцією адрес (NAT) і мережними екранами.

– VMS допускає присутність стабільних операторів перед екранами охоронного телебачення, у той час коли абоненти VSaaS здебільшого під'єднуються до системи через браузер тільки після поступання тривожного знака або для аналізу відеоархіву (звітів).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

– Як наслідок пунктів 1 і 2, відеоаналітика відіграє важливішу функцію в VSaaS, чим VMS. Відеоаналітику використовують для управління транспортуванням даних, компонування тривожних сигналів та звітів.

– Інтерфейс користувачів VSaaS має бути втілений на основі браузера чи мобільного телефону, у той час як VMS в основному застосовує інсталюємий додаток.

– Протекція локального сховища має більшу вагу в додатках VSaaS, чим VMS.

– VSaaS має на меті забезпечити сервіс значно більшої кількості користувачів і камер, ніж VMS.

– В основному, додаток VMS налаштовують спеціалізовані системні інтегратори, а додаток VSaaS – кінцеві юзери, що передбачає доступність та простоту інтерфейсу VSaaS.

– Оператори послуги VSaaS чи/або їх абоненти чутливіші до ціни кінцевого устаткування (камер, локальних відеосерверів), ніж користувачі VMS.

Прогнозовано, що для забезпечення послуг VSaaS або «Відеоаналітика як сервіс» береться хмарна інфраструктура.

3.3 Розробка функціональної схеми

На рисунку 3.2 відтворена функціональна схема системи. Пізніше розберемо докладніше.

Систему відеоспостереження за технологією VSaaS застосовуємо практично за прямим призначенням, як систему відеоспостереження в житлових будівлях. Вона виконує такі функціональні задачі:

– Безперервне візуальне оцінювання ситуації в житловому секторі методом телевізійного слідування за під'їздами й навколишніми територіями, і також всередині будинків за ліфтами й сходовими майданчиками.

– Своєчасне інформування служб охорони правопорядку та інших екстрених служб міста про виникнення або підозру на виникнення надзвичайних ситуацій, які можуть нести загрозу життю та здоров'ю людей, збереженню особистого й державного майна.

– Доставка службам охорони правопорядку й іншим службам критичної інфраструктури архівних даних телевізійного спостереження для встановлення розвитку подій, допомоги у проведенні оперативно-слідчої роботи і т.д.

– Інтеграція систем відеоспостереження з локально-обчислювальними мережами, а також доступ до системи відеоспостереження з кожної точки світу, застосовуючи підключення до Інтернету.

– Зоровий контроль освітлення під'їздів та громадських територій у нічний час.

– Зоровий контроль ліфтів, на факти виявлення порушень правопорядку та пошкодження громадського майна.

– Контроль утримання територій в порядку: вчасне прибирання, вивіз сміття тощо.

КБПЗ – 2024

					VKPM-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

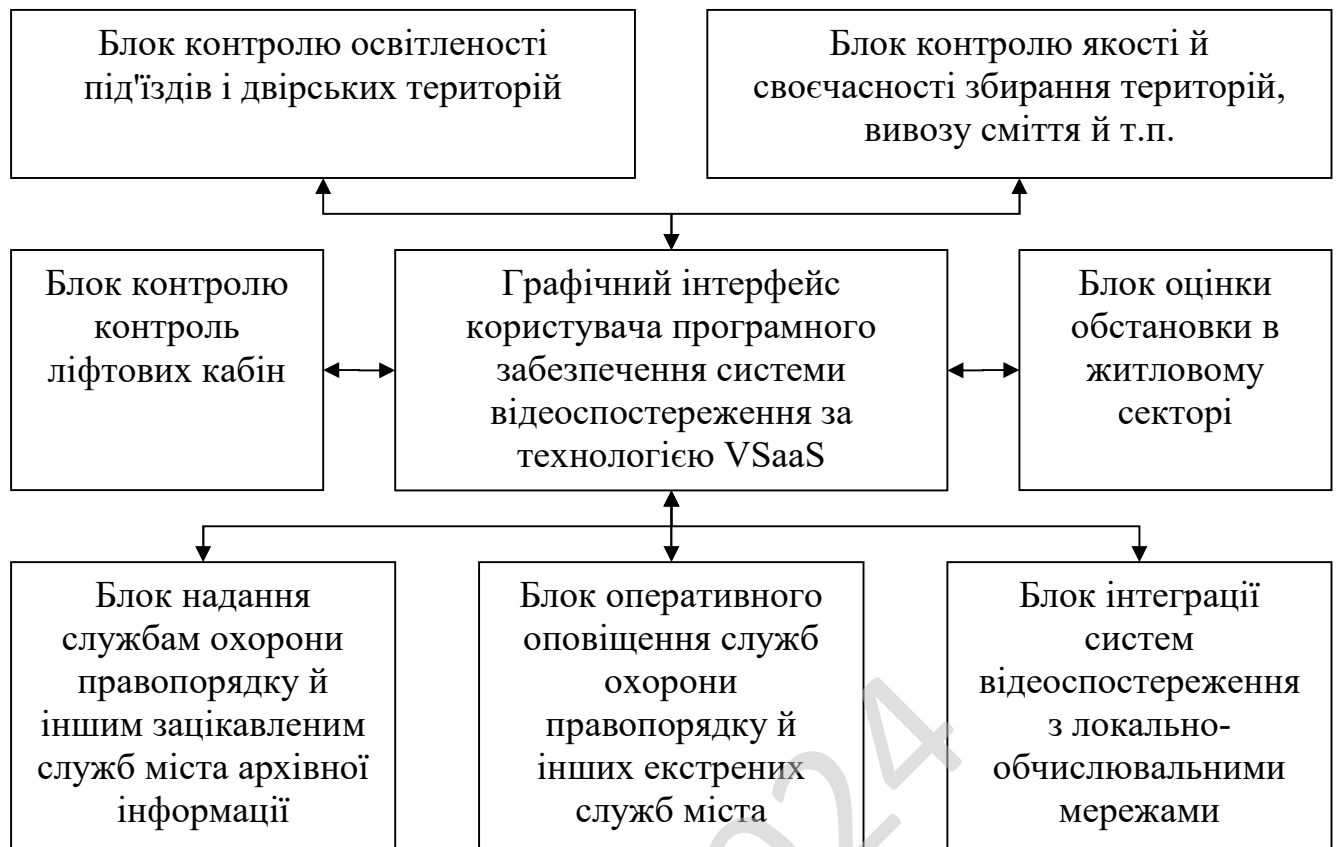


Рисунок 3.2 – Функціональна схема системи

Визначаються головні точки відеоспостереження житлової будівлі з позиції людей, які в ньому живуть:

- відеоспостереження у під'їзді (вхід);
- відеоспостереження стоянки автомобілів;
- відеоспостереження ігрового майданчика.

Зараз детально розберемо кожну зону в залежності від задач системи спостереження будинку.

Відеоспостереження ігрового майданчика

Одна із основних ліній побудови системи відеоспостереження простору житлової території.

Є варіант обмежитись використанням лиш однієї камери, але за рахунок її монтажу настільки високо, щоб був огляд на весь майданчик. Дозвіл 700 ТВЛ effію для аналогової камери, а у випадку IP відеоспостереження мегапіксельна

оснащенням для передавання відеоматеріалу на дальні відстані (до 1000м.), зручним інтерфейсом, що вражає характеристиками. Виготовлений він в Україні та має адекватний сервіс, чого не можна сказати про більшість низькопробних реєстраторів невідомих виробників.

– Коннектори для відеоспостереження, теж варто застосовувати лише відповідної якості. Якісні коннектори – як мінімум, виготовлені з міді й мідного сплаву, точка з'єднання коннекторів захищає від впливів зовнішніх факторів, монтажна коробка, бажано, як мінімум, місце контакту помістити в термоусадку.

– Без використання кабелю теж можна спорудити систему ір відеоспостереження досить просто, навіть застосуванням камер без wi-fi модуля, методом використання додаткового мережного бездротового устаткування (wi-fi точка доступу для відеонагляду, wi-fi роутер).

– Блок живлення для системи відеоспостереження, один чи кілька, розраховується відповідно до потрібних потужностей камер, але з певним резервом на затрату живлення в проводах або максимальні перегони при включенні ПЧ підсвічування, наприклад, якщо камери використовують 2,8 А, то, обираючи між блоком живлення 3А та 5 А потрібно взяти 5А.

– Для системи відеоспостереження багатоквартирної будівлі необхідно підбирати реєстратор із частотою 25 кадрів в d1 25к/с, як мінімум, на каналах ігрового майданчика та входу до під'їзду, оскільки, прохідність тут максимальна.

– Для ІР відеоспостереження треба серйозно поставитися до вибору ПЗ й модулів, які підключаються, наприклад, модуль для встановлення номерів автомобілів, чи модуль встановлення осіб – ідентифікація відвідувачів.

Також потрібно налаштувати можливість віддаленого доступу для жителів: дистанційний перегляд шляхом використання хмарного сервісу.

Дуже важливо при монтування відеоспостереження будинку одразу закрити питання про сервісне обслуговування всієї системи. Це може бути індивідуальний візит спеціаліста за потребою чи, значно вигідніший варіант:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

укладення договору про довгострокове сервісне обслуговування. У другому варіанті оплата проводиться щомісяця чи щокварталу, і, крім індивідуальних виїздів, профілактичні огляди, налаштування за необхідністю та пріоритети в черзі на обслуговування.

Відеоспостереження стоянки автомобілів

Тотальний нагляд за стоянкою, цілодобове спостереження за рухом транспортних засобів на її території.

Тут потрібна камера вуличного виконання зі стопроцентним режимом день/ніч чи ІЧ підсвічуванням. Дуже корисною тут виявиться камера з функцією антивідблиск, для перешкоджання засвіченню відеокамери автомобільними фарами. Один із варіантів: камера на самому в'їзді для визначення автомобільних номерів, особливо зручним це виявляється, якщо в'їзд на стоянку обмежується шлагбаумом чи використовується спеціальне програмне забезпечення для відеоспостереження. В такому випадку дуже просто вести облік автомобілів, які заїжджають та виїжджають з території. Взагалі, на стоянці має бути розміщено, як мінімум, дві камери: одна для спостереження за загальною територією стоянки, друга фіксує державні номери автомобілів.

Переваги житлового простору з системою безпеки та без неї перераховувати, звичайно ж, немає потреби. Людське життя – найвища цінність.

Після розбору блоків функціональної схеми логічно буде перейти до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів застосовується для візуалізації маневрів обробки інформації (структурне проектування). Для розробника є цілком звичним починати з креслення діаграми взаємодії процесів даних ступеня контексту, завдяки цьому, демонструється взаємодія системи.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Надалі діаграма уточнюється методом деталізації процесів та струменів інформації, маючи на меті продемонструвати систему, що розробляється. Діаграма процесів розробленої системи відбита на рисунку 3.3. При її докладному розборі можемо побачити, як саме здійснюється взаємодія у розробленій системі. Застосовується модель проектування, графічне представлення «потоків» сигналів в системі інформації.

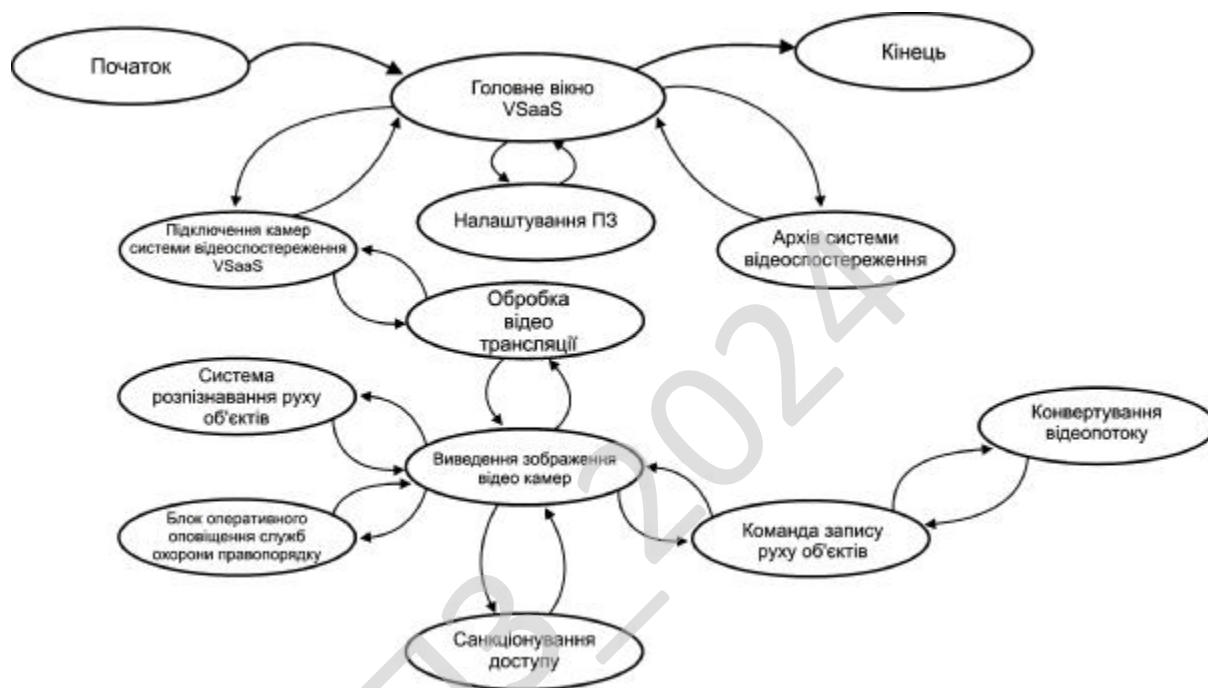


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних мають чотири види елементів:

- Процеси, які представляють собою трансформацію інформації в межах системи, що описується.
- Сховища даних (репозиторії).
- Зовнішні, стосовно системи, сутності.
- Потоки сигналів між сегментами трьох попередніх видів.

Отож, після детального розбору опису системи, структурної, функціональної схеми системи, та діаграми взаємодії процесів, маємо перейти до опису блок-схем центральної програми та підпрограм, які беруться для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Відповідно, від коректності та деталізації пропрацювання блок-схеми залежить результативність програми в цілому.

Під час визначення вихідної точки відліку при будівництві схем бралося до уваги, що, виходячи з обрання мови програмування та інших технічних засобів, програма вийде об'єктно-орієнтована, що передбачає оптимізацію, а також те, що при проектуванні програми необхідно виділити головну роль модулю відеоспостереження за технологією VSaaS.

Функціональні блоки на схемі позначаються прямокутниками, в центрі яких пишуться їх найменування відносно до виконуваних функцій. Відношення між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілочками, які показують напрямок впливів.

Функціональні блоки можуть реалізовуватися в збільшеному і розгорнутому вигляді. У першому випадку на схемі зображуються основні блоки системи та взаємозв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 відображена робота підпрограми.

По рисунках зрозуміло, що робота основної програми утворюється з початкових етапів ініціалізації ПЗ, перевірки присутності ресурсів системи, блоку початку головного циклу з очікуванням запиту від користувача, в якому робиться виклик підпрограми та крайньої стадії – перевірки наявного стану та поверненням на старт схеми чи з закінченням роботи розробленого ПЗ.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

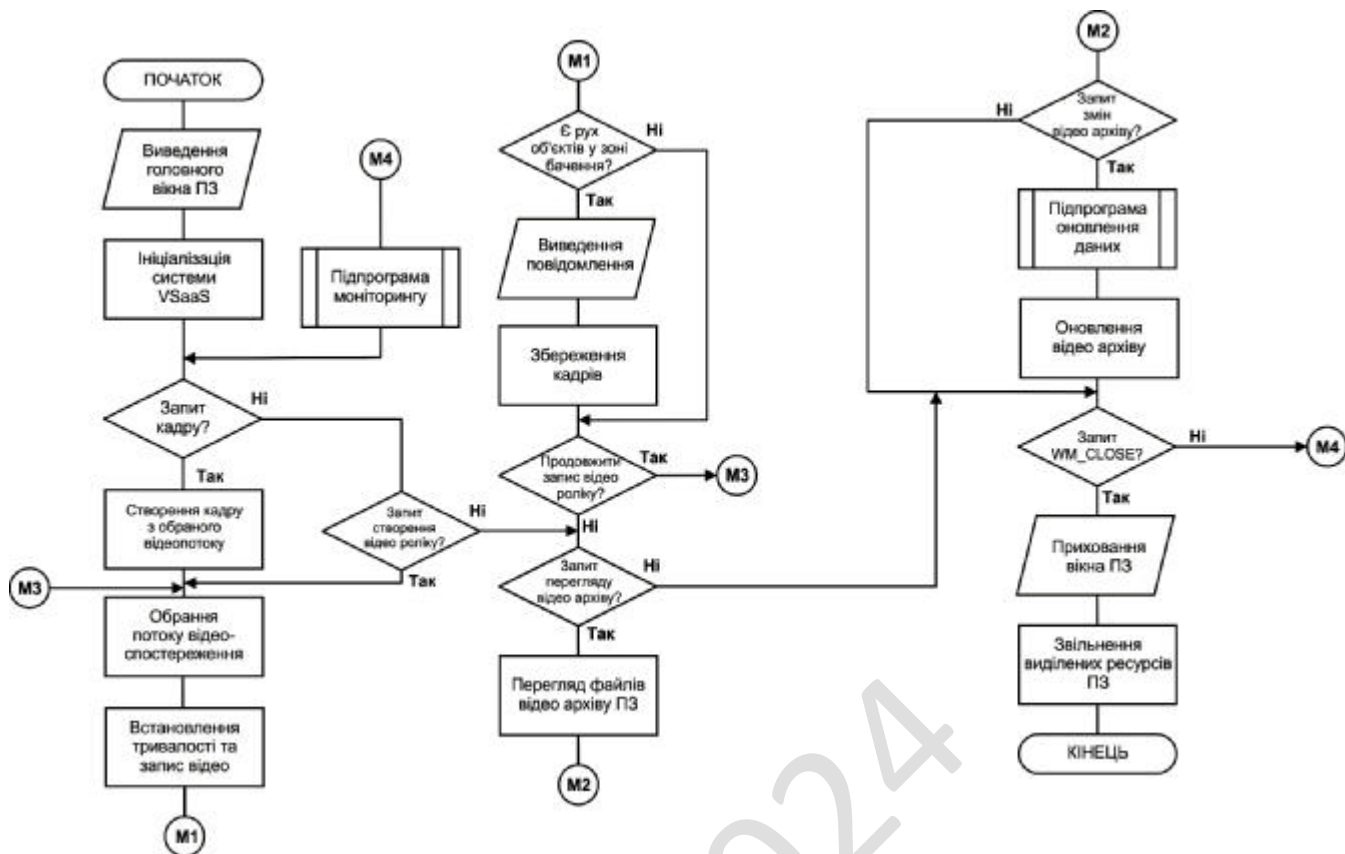


Рисунок 4.1 – Блок-схема основної програми

Під час роботи підпрограми здійснюється основний функціонал системи з повторюваними послідовностями, перевіркою теперішнього стану та вертання в головну програму прапорів стану виконання.

Були застосовані прийоми із застосуванням UML, це стандартизована мова моделювання, використовується у сукупності об'єктно-орієнтованого програмування. Є необхідною ланкою шаблонного процесу розроблення програмного забезпечення. UML є мовою обширного профілю, це досяжний стандарт, що застосовує графічні позначки для формування схематичної моделі системи, яку названо UML-моделлю. UML був розроблений для визначення, ілюстрування, планування та документування в основному програмних систем. UML не є мова програмування, але в методах виконання UML-моделей як витлумаченого коду можлива кодогенерація.

Проаналізуємо обрані технології та їх ключові компоненти, які підтверджують вірність застосованих проектних рішень.

При розроблянні ПЗ були взяті підходи ризик-менеджменту – це режим управління ризиками, яка включає в себе стратегію та спосіб управління, спрямовані на досягнення центральних цілей. Дієвий ризик-менеджмент включає:

- систему управління;
- систему ідентифікації та відмірювання;
- систему супроводу (спостереження та контролю).

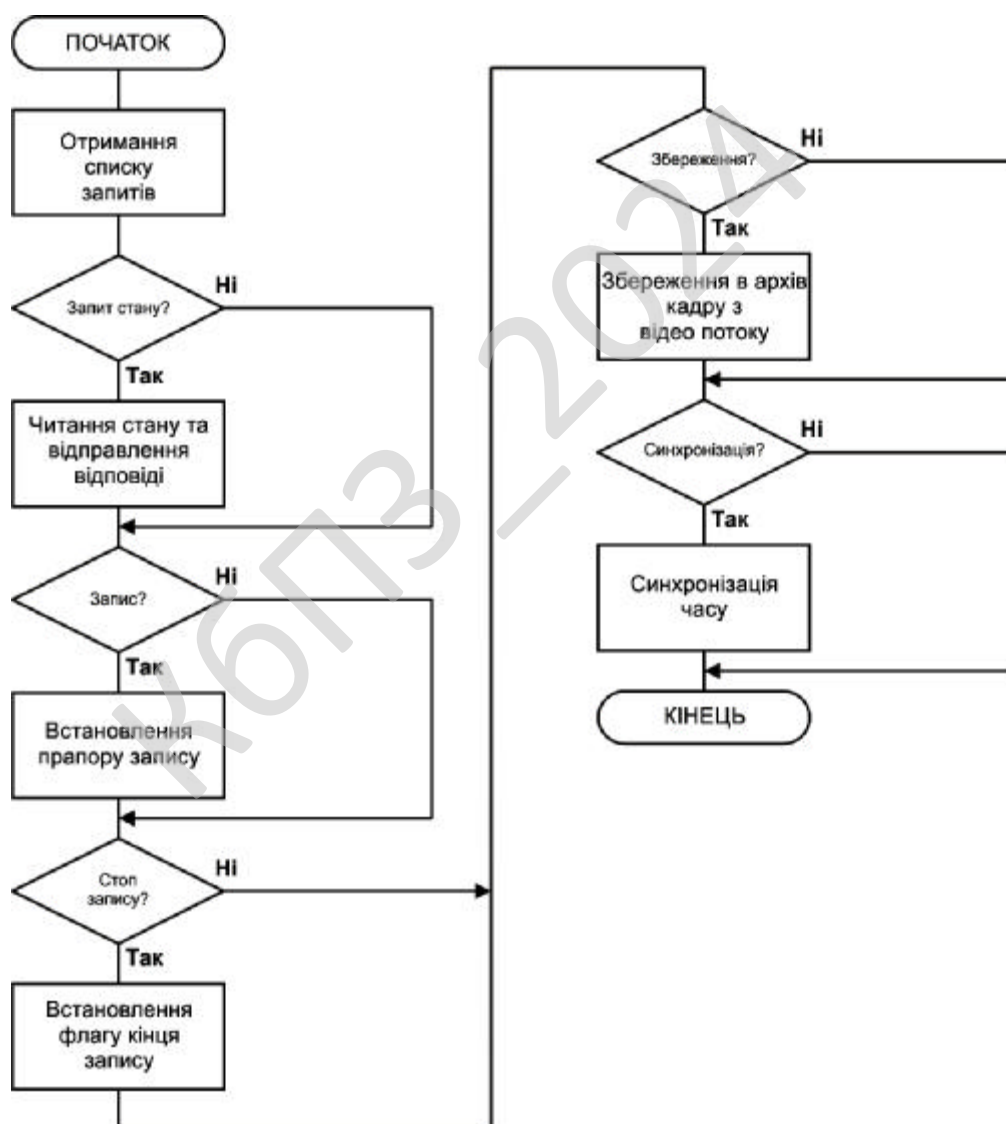


Рисунок 4.2 – Блок-схема роботи підпрограми

Сьогочасна наука бачить ризик як достовірну подію, в результаті приходу якої можуть статися благодотворні, нейтральні або несприятливі зміни. Якщо ризик допускає і плюсові, і мінусові результати, то його відносять до спекулятивних ризиків. Якщо ж результати лише мінусові чи їх немає взагалі, то подібний ризик називають чистим.

Ціль ризик-менеджменту – максимальна конкурентоспроможність індивідів господарювання способом захищення від реалізації прямих небезпек.

Теорія ризик-менеджменту заснована на кількох провідних поняттях: рентабельність, регресія, диверсифікація.

У 1738 відомий математик Даніель Бернуллі (Швеція) збагатив теорію ймовірності методом вигідності чи привабливості тих чи інших наслідків подій. Ідея Бернуллі полягає в тому, що в процесі приймання рішення люди в основному звертають максимально увагу на розмір наслідків різноманітних результатів, ніж їх можливість.

Наприкінці ХІХ століття дослідник Ф. Гальтон (Англія) пропонував числити регресію чи вертання до середнього значення сумарною статистичною закономірністю. Сутність регресії пояснювалася ним як вороття явищ до звичного стану з часом. Пізніше було доказано, що аксіома регресії працює в різних ситуаціях, стартуючи з гемблінгу та розрахування ймовірності створення непередбачуваних ситуацій, і фінішуючи передбаченням змін економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично пояснив стратегію диверсифікації інвестиційного портфеля, приміром, він роз'яснив, як способом проаналізованого розподілення вкладень зменшити відхилення доходу від прогнозованого показника. У 1990 Г. Марковіц отримав Нобелівську премію за створення теорії і практики оптимізації портфеля фондових активів.

Етапи ризик-менеджменту

У ризик-менеджменті виділяють кілька головних етапів:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

- на першому етапі з'ясовується ризик з супровідною оцінкою ймовірністю його реалізації і меж поширення результатів;
- на наступному етапі розробляється ризик-стратегія з метою зменшення можливості здійснення ризику і зменшення прогнозованих негативних наслідків;
- потім обираються методи та інструментарій керування визначеним ризиком;
- четвертий етап – безпосереднє керування ризиком;
- на кінцевому етапі відбувається оцінка отриманих результатів та корекція ризик-стратегії.

Визначальним ступенем ризик-менеджменту є стадія вибору методів та інструментарію керування ризиком.

Методи та інструментарій ризик-менеджменту

Основні методи ризик-менеджменту: відступання від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій досить широкий. Тут є: політичні, координаційні, правові, економічні, суспільні інструменти, до того ж ризик-менеджмент як система дозволяє синхронно активувати кілька методів та інструментів ризик-управління.

Найбільш використовуване знаряддя ризик-менеджменту – страхування. Воно допускає перекладання відповідальності за компенсацію можливої шкоди сторонній організації (страховій компанії).

Зразки других інструментів: відхилення від непомірно ризикових дій (метод відмови), попередження чи диверсифікація (метод зниження), аутсорсинг збиткових ризикових функцій (метод передачі), розробляння ресурсів чи запасів (метод ухвалення).

Виконано реалізацію стиску зображень за допомогою популярного та досить поширеного алгоритму JPEG. Розберемо вихідний код.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

unit JP_IO;
// Модуль стиску зображень
interface
// Інтерфейсна частина
{$WARN SYMBOL_PLATFORM OFF}
// компіляторні налаштування
{$WARNINGS OFF}
// Бібліотеки які використовуються
uses Windows, SysUtils, Classes, Graphics, IJL;

type
// створення типу виключення
    EIJLException = class(Exception);

Type
// створення типу стиску
    TIJLScale=(ijlFull, ijlHalf, ijlQuarter, ijlEighth);
...
// Процедура потокової конвертації до формату Jpeg
procedure SaveBmpToJpegStream(Bitmap:TBitmap;Stream: TMemoryStream; const
    Quality: integer);
...
implementation
// Секція реалізації

procedure IJLCheck(Code:Integer);
// процедура роботи з помилками
begin
    if Code<>IJL_OK then
        raise EIJLException.Create(ijlErrorStr(Code));
end;
// Процедура потокової конвертації до формату Jpeg
procedure SaveBmpToJpegStream(Bitmap:TBitmap;Stream: TMemoryStream; const
Quality:integer);
var
// Об'ява змінних
    jcprops:TJPEG_CORE_PROPERTIES;
    DIB: TDIBSection;
Begin
// Початок процедури
    FillChar(jcprops, SizeOf (jcprops), 0);
// заповнення

```

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55


```

        IJLCheck(ijlWrite(@jprops, IJL_JBUFF_WRITEWHOLEIMAGE));
        Stream.Size:=jprops.JPGSizeBytes;
// Розмір потоку
    finally
        IJLCheck(ijlFree(@jprops));
    end;
end;
...
end.

```

Алгоритми машинної графіки можливо розмежувати на два ступеня: верхній та нижній Категорію алгоритмів нижнього ступеню рекомендована для запровадження графічних примітивів (математичні розрахунки, лінії, кола, заповнення і т.п.).

Ці алгоритми чи схожі до них відображені в графічних сховищах мов найвищого ступеня або втілені апаратно в графічних процесорах робочих станцій.

Між алгоритмами нижнього рівня реально виокремити такі групи.

1. Найелементарніші в значенні математичних методів та абсолютно прості в реалізації. Загалом, ці алгоритми не найкращі за обсягом здійснюваних обчислень чи потрібними ресурсами пам'яті.

2. Алгоритми з більш складними математичними передумовами, що різняться більшою продуктивністю.

3. До третьої групи можна зарахувати алгоритми, реалізація яких допустима без значних складнощів (допускаючи розпаралелювання, рекурсивні, реалізовані в простих вказівках). До цієї групи цікавито можна додати і алгоритми перших двох груп.

4. І на кінець, четверта група, до якої входять алгоритми із спеціальними завданнями (наприклад, для прибирання сходового ефекту, мерехтіння картинки і багато інших).

До алгоритмів вищого рівня входять головним чином алгоритми знищення непомітних ліній та поверхонь.

Завдання ліквідації непомітних ліній і поверхонь, як і було, лишається провідним в машинній графіці.

До завдання знищення невидимих ліній й поверхонь додається завдання побудови (замальовування) півтонових (реалістичних) картинок, себто обліку явищ, поєднаних з числом і видом потоків світла, обрахування властивостей поверхні тіла (проглядність, заломлення, відбивання світла).

Та, слід не забувати, що вивід предметів в алгоритмах вищих рівнів гарантується примітивами, які готують алгоритми нижніх рівнів, то ж не рекомендується нехтувати питанням вибору та розробки дієвих алгоритмів низького рівня.

Для найрізноманітніших сфер використання машинної графіки на передній план можуть виступати різні характеристики алгоритмів.

Для наукової графіки вирішальну роль відіграє універсальність алгоритму, а швидкість дії при цьому може бути вторинною потребою. При цьому, для моделюючих систем, які займаються відтворенням об'єктів, що рухаються, швидкодія постає найголовнішим критерієм, бо першочерговою є потреба генерувати картинку практично в поточному масштабі часу.

Властивості растрової графіки поєднані з тим, що рядові зображення, які попадаються в основній діяльності (малюнки, графіки, мапи, художні картини і т.п.), здійснюються на площі, що складається з безкінечного набору точок.

Екран растрового монітору представляється матрицею дискретних частинок, які мають визначені фізичні розміри.

За цих умов їхня кількість вельми обмежена. Тому фізично неможливо протягти точну лінію з однієї точки в другу, а можливо здійснити лише апроксимацію цієї лінії з відбиттям її на дискретній матриці (площині).

Ця площина ще носить назву цільна чисельними ґратами, растрова площина або растр. Ці ґрати являються квадратною сіткою з кроком 1. Відбиття якогось об'єкту на цільна чисельні ґрати зветься розкладанням його в растр чи растровим уявленням.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Архітектура ПЗ, що використовувалась

Архітектура клієнт-сервер – архітектурний шаблон програмного забезпечення, йде панівною концепцією у формуванні розділених мережних програм і визначає взаємодію та чейндж інформацією між ними. Виділяє наступні головні складові:

- комплект серверів, які надають відомості чи інші різноманітні послуги програмам, які апелюють до них;
- комплект користувачів, які застосовують послуги, надані серверами;
- мережа, яка формує взаємодію між користувачами та серверами.

Сервери не залежать один від одного. Користувачі теж працюють хоч і одночасно, але абсолютно самостійно й не залежачи ні від кого. До серверів клієнти не прив'язані. Цілком стандартним є одночасна обробка одним сервером інформації від цілого потоку клієнтів; та також, кожен клієнт має можливість використовувати в роботі різні сервери. Клієнти зазвичай знають про наявні доступні сервери, але цілком не обов'язково та не потрібно їм знати про наявність інших користувачів.

Тут також потрібно розуміти, хто чи що є тим самим «клієнтом». Можемо мати на увазі клієнтський комп'ютер, з якого проводиться звертання до інших комп'ютерів. Йдеться мова про клієнтське й серверне програмне забезпечення. Ну і в кінці кінців, це, звичайно ж, люди, які за допомогою певного програмного й апаратного забезпечення отримують доступ до необхідної інформації.

Загальнозрозумілим є положення, що клієнти й сервери в першу чергу - програмні модулі. Зазвичай вони розташовані на абсолютно різних комп'ютерах, та все ж трапляються випадки, коли і та і друга програми – і клієнтська, і серверна, фактично містяться в одному місці; в цьому випадку сервер носить назву локальний.

Модель клієнт-серверних взаємин визначається першим чином розподіленням обов'язків між клієнтом та сервером. Абсолютно резонно виділяються три рівні операцій:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

– рівень представлення даних, який представляє собою панель управління користувача і несе відповідальність за донесення сигналів користувачу і подання від нього управляючих команд;

– прикладний рівень, він виконує головну логіку ПЗ і на ньому виконується потрібна обробка даних;

– рівень керування даними, цей гарантує зберігання інформації та доступ до неї.

Дворівнева клієнт-серверна архітектура визначає співдію обох програмних модулів – клієнтського й серверного. Залежно від того, яким чином між ними розділяються подані раніше функції, виділяють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та керування даними сконцентрована на сервері. Клієнтська програма здійснює лише дії рівня представлення;

– модель товстого клієнта, коли сервер просто управляє сигналами, а упорядкування даних та панель управління користувача сконцентровані на боці клієнта. Товстими клієнтами частенько ще визначають пристрої з лімітованою потужністю: кишенькові пристрої, мобільні телефони і т.п.

Стандартним зразком клієнт-серверної взаємодії є WWW. В дійсності є велика чисельність веб-серверів, на яких розташовуються ті чи інші дані. У максимально простому варіанті ця інформація представляється набором веб-сторінок, які мають можливість зберігатися на сервері у формі файлів, позначених способом мови розмітки HTML. Але найчастіше, все значно складніше: величезна частка веб-ресурсів на теперішньому етапі є динамічними, тобто не зберігаються в завчасно підготовленому вигляді, а формуються конкретно в процесі опрацювання запиту від клієнта.

Для того, щоб людина, яка працює в мережі Інтернет, мала доступ до тієї чи іншої сторінки, на її комп'ютері повинно бути встановлено потрібне програмне забезпечення. Програми для доступу до веб-сторінок називаються браузерями.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Проте, окрім браузерів, до серверами можуть користуватися й інші клієнти, а конкретно – автономні програми. Вони можуть окреслювати взаємодію з людиною, та можуть функціонувати в абсолютно самостійному режимі. Стандартним класом цих програм є роботи, рекомендовані для автоматичного перегляду веб-ресурсів. В тому числі, роботи є значущим ресурсом пошукових систем і застосовуються ними для перегляду сторінок і збирання даних про них.

Щоб виконати запит до веб-сервера клієнтська програма зобов'язана надати місцезнаходження комп'ютера, на якому міститься серверна програма, найменування необхідного документа та, ймовірно, інші дані, що специфікують запит. Мережа гарантує визначення сервера та передавання йому клієнтського запиту. Серверні програми опрацьовують цей запит, результат надсилається мережею клієнту.

Трирівнева клієнт-серверна архітектура, що стартувала в середині 90-х років, окреслює відділення прикладного рівня від керування даними. Відділяється окремий програмний рівень, на якому зосереджена прикладна логіка ПЗ. Програми перехідного етапу можуть працювати під скеровуванням спеціальних серверів ПЗ, але стартування цих програм може реалізовуватися і під керуванням стандартного веб-сервера. Врешті, керування даними забезпечується сервером даних.

Для діяльності з системою користувач застосовує шаблонне програмне забезпечення – звичайний браузер. Це прибирає потребу завантажувати та інсталювати спеціальні програми (але все ж часом ця необхідність виникає).

Користувачу потрібно дати у використання інтерфейс, який надавав би можливість працювати з системою та зручно формувати запити. Форми, що наповнюють цей інтерфейс, розташовуються на веб-сторінках і завантажуються разом із ними.

Веб-оглядач створює запит та переправляє його на сервер, який опрацьовує його. За потребою сервер застосовує серверні програмні модулі, які обробляють запит і, якщо виникає така необхідність, апелюють до сервера даних.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Сервер даних виконує операції з інформацією, що знаходиться в системі і складає її інформаційну основу. Загалом, він здатний досконало виконати відбір потрібних даних з наявної інформаційної бази та надати її модулю проміжного рівня для наступного опрацювання. Інформація, з якою працює сервер даних, як правило, організована як реляційна база даних.

В основному, веб-сервер та серверні модулі проміжного рівня розташовані на одному комп'ютері, хоча і представляють собою абсолютно самостійні та логічно незалежні програмні модулі.

На сьогоднішній день для програмування модулів перехідного рівня застосовують мову серверних сценаріїв PHP, а для керування даними – СУБД MySQL. Відповідно, нитку PHP-MySQL слід розглядати як звичний інструмент для проектування порівняно нескладних інтерактивних веб-сайтів та систем електронної комерції; біля 90% комерційних систем у сучасності створюється саме на цій базі. Одночасно як способи керування даними, так і middleware-засоби можуть бути абсолютно різними. Тобто, для виготовлення серверних програм, крім PHP, активно використовуються Java, Perl, Python, Delphi.

Загалом, технології винайдення розподілених, зокрема веб-програм, швидко розвиваються. Не варто забувати про технології EJB (Enterprise Java Beans), CORBA, і ще про .NET – відносно новітню ініціативу компанії Microsoft. Щоб зберегти дані та передати їх за призначенням, активно застосовується так названа розширювана мова розмітки XML (Extensible Markup Language).

Використана методологія розробки

В реалізації програми реалізовано підходи прискорення розробки на основі методологій Agile.

Змінна розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розроблення програмного забезпечення, яке базується на ітеративній розробці, де потреби та розв'язки змінюються через співпрацю між самоорганізованими багатофункціональними командами.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Мінлива розробка – відмінний засіб для максималізації ефективності розробників програмного забезпечення.

Більшість змінних методологій спрямовані на зменшення ризиків, методом зведення розробки до лінії коротеньких циклів, які носять назву ітерацій, які в основному продовжуються один-два тижні. Кожна ітерація в принципі має вигляд програмного проекту в мініатюрі, включає всі задачі, потрібні для видавання найменшого приросту за функціональністю: планування, аналізування вимог, проектування, кодування, тестування, документування. Окрема ітерації, зазвичай, замало для виходу нової версії продукту, тобто, змінний програмний проєкт готовий до випуску під самий кінець кожної ітерації. Після закінчення кожної ітерації, команда розробляє ревізію пріоритетів розробки.

Agile наголошує увагу на невимушеному спілкуванні «віч-на-віч». Більша частина agile команд розміщені в єдиному офісі, його часто називають bullpen. Вона має в складі і «замовників» (замовники, які встановлюють продукт, до того ж це можуть бути менеджери продукту, бізнес-аналітики чи клієнти). Офіс інколи включає тестувальників, дизайнерів панелі користувача, технічних авторів, менеджерів.

Керуючою метрикою agile методів є робоча продукція. Віддаючи першість невимушеному спілкуванню, agile-методи мінімізують об'єм письмової документації порівняно з іншими методами. Саме тому ці методи піддаються критиці, як недисципліновані.

Agile – джерело процесів розробки, але не один можливий підхід в розробці програмного забезпечення, й визначається Agile Manifesto. Agile не містить практик, а встановлює цінності та стандарти, якими керуються блискучі команди.

Agile Manifesto розроблений та прийнятий 17-ма програмістами 11-13 лютого 2001 року на лижному курорті The Lodge at Snowbird в горах Юти. Маніфест підписали представники цих методологій Extreme programming, Scrum, DSDM, Adaptive software development, Crystal Clear, Feature driven development,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

у клієнта). Це продукує капітальне пониження якості кінцевого продукту і нагромадження дефектів.

Методології. Існують методології, які дотримуються цінностей та принципів заявлених в Agile Manifesto, кілька з них:

1. Agile Modeling – комплекс понять, принципів та прийомів (практик), які надають можливість стрімко та просто здійснювати моделювання й записування в проектах розробки програмного забезпечення. Тут не включено деталізовану інструкцію з проектування, не включено ніяких описів, як робити діаграми на UML.

Центральна ціль – результативне моделювання та записування; але не охоплює програмування та тестування, не включає питання керування проектом, розширення і супроводу системи. Проте, містить в собі перевірку моделі кодом.

2. Agile Unified Process (AUP) – спрощена версія IBM Rational Unified Process (RUP), спроектована Скоттом Амблером, вона змальовує елементарне й повністю зрозуміле приближення (модель) для формування програмного забезпечення для бізнес-додатків.

3. Agile Data Method – група ітеративних методів розробляння програмного забезпечення, тут вимоги та рішення здобуваються в межах взаємозв'язку різних крос-функціональних вказівок.

4. DSDM сформований на принципі швидкої розробки додатків (Rapid Application Development, RAD). Представляє ітеративний та інкрементний підхід, який надає виокремленого значення довготривалій участі у процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка.

Використане в FDD поняття функції чи властивості (feature) системи досить поряд

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

до поняття прецеденту використання, використовуваному в RUP, значна відмінність – це додатковий ліміт: «кожна функція має надавати реалізацію не довше, як за два тижні». Якщо сценарій застосування вельми невеликий, то його цілком можна визнавати функцією. Якщо все ж великий, то його можна розділити на кілька порівняно самостійних функцій.

8. Getting Real – ітераційний підхід без функціональних специфікацій, який застосовується для веб-додатків. У цьому методі спочатку проектується сам інтерфейс програми, а вже пізніше її функціональна сторона.

9. OpenUP – це ітераційно-інкрементний спосіб розроблення програмного забезпечення. Кваліфікується, як найпростіший і змінний варіант RUP. OpenUP розділяє функціональний цикл проекту на наступні фази: початкова фаза, фаза уточнення, фаза конструювання, фаза передачі. Функціональний цикл проекту гарантує подачу допитливим особам та учасникам групи точок ознайомлення та прийняття рішень протягом всього проекту. Це дає можливість ефективно інспектувати обстановку і своєчасно вирішувати виникаючі проблеми для найкращих кінцевих результатів. План проекту визначає функціональний цикл, а фінішним результатом є остаточний додаток.

10. Scrum визначає правила управління процесом розробки та дозволяє застосовувати вже існуючі практики кодування, при цьому також корегує запити або вносить тактичні зміни. Застосування цієї методології надає перспективу знаходити та ліквідувати будь-які відхилення від очікуваного результату на самих ранніх етапах розроблення програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Застосовує підходи з концепції бережливого виробництва.

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо способом використання алгоритму захисту інформації RSA. Для початку потрібно обчислити пару ключів

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

(секретний ключ і відкритий ключ). Для цього відправник електронних документів обчислює два більших простих числа P і Q , після чого вираховує їх добуток $N = P * Q$ та значення функції $\varphi(N) = (P-1)(Q-1)$. Потім відправник вираховує число E з умов $E < \varphi(N)$, НЗД $(E, \varphi(N)) = 1$ та число D з умов $D < N$, $E * D \equiv 1 \pmod{\varphi(N)}$.

Пари чисел (E, N) являються відкритим ключем. Цю пару чисел автор надає партнерам по переписці для звіряння цифрових підписів. Число D утримується автором як секретний ключ для підписування.

Припустимо, що відправник планує підписати повідомлення M перед його відправкою. На початку повідомлення M (блок інформації, файл, таблиця) стискають за допомогою хеш-функції $h(-)$ у ціле число m : $m = h(M)$.

Далі обчислюють цифровий підпис S під електронним документом M , застосовуючи хеш-значення m і секретний ключ D : $S = m \pmod{N}$.

Пари (M, S) передаються партнеру-отримувачу як електронний документ M , підписаний цифровим підписом S , причому підпис S сформований власником секретного ключа D .

Після прийняття пари (M, S) отримувач обчислює хеш-значення повідомлення M двома різними методами. В першу чергу, він поновлює хеш-значення m' , використовуючи криптографічне перетворення підпису S з застосуванням відкритого ключа E : $m' = S^E \pmod{N}$.

Окрім цього, він знаходить результат хешування отриманого повідомлення M з допомогою такої ж хеш-функції $h(-)$: $m = h(M)$.

За умови дотримання рівності обчислених значень, тобто $S^E \pmod{N} = h(M)$, отримувач визнає пару (M, S) справжньою. Доказано, що лише власник секретного ключа D може створити цифровий підпис S по документу M , а визначити секретне число D по відкритому числу E не легше, ніж розкласти модуль N на множники. Окрім цього, можна чисто математично аргументувати, що результат перевірки цифрового підпису S буде позитивним лише у випадку, коли при вирахування S був застосований секретний ключ D , що відповідає

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

відкритому ключу *E*. Тому відкритий ключ *E* іноді називають "ідентифікатором" того, хто підписав.

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розберемо розроблене ПЗ відеоспостереження за технологією VSaaS, зображене на рисунку 5.1. З рисунку можна роздивитися, що інтерфейс головного вікна розподілено на слідуючі функціональні розділи:

- Навігаційне меню: Архівні дані; Налаштування; Довідка.
- Функції, представлені у графічному вигляді.
- Розділ обрання відеопотоку.
- Розділ виведення результату роботи системи.
- Навігаційне меню, яке можна вивести на екран натисканням правої клавіші маніпулятора миші.
- Функціональні кнопки ПЗ.

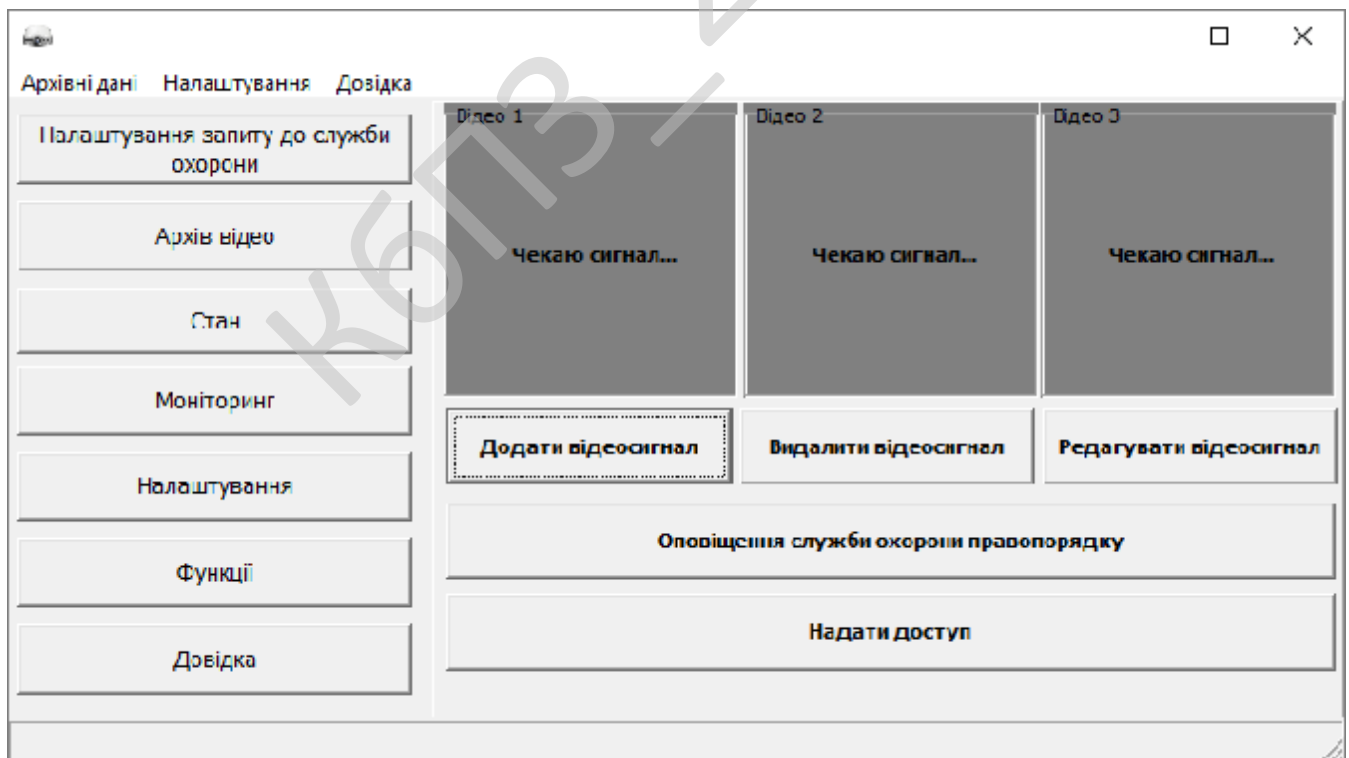


Рисунок 5.1 – Головне вікно ПЗ

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Розроблена програма має надзвичайно простий та інтуїтивно зрозумілий інтерфейс користувача. Кожен, хто хоча б на елементарному рівні володіє операційним середовищем Windows, без особливих труднощів опанує і цю програму.

Якщо програма працює та не видає жодних помилок – впевнено та спокійно працюємо. Якщо ж ні, то виконуємо інструкції й виправляємо проблеми.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

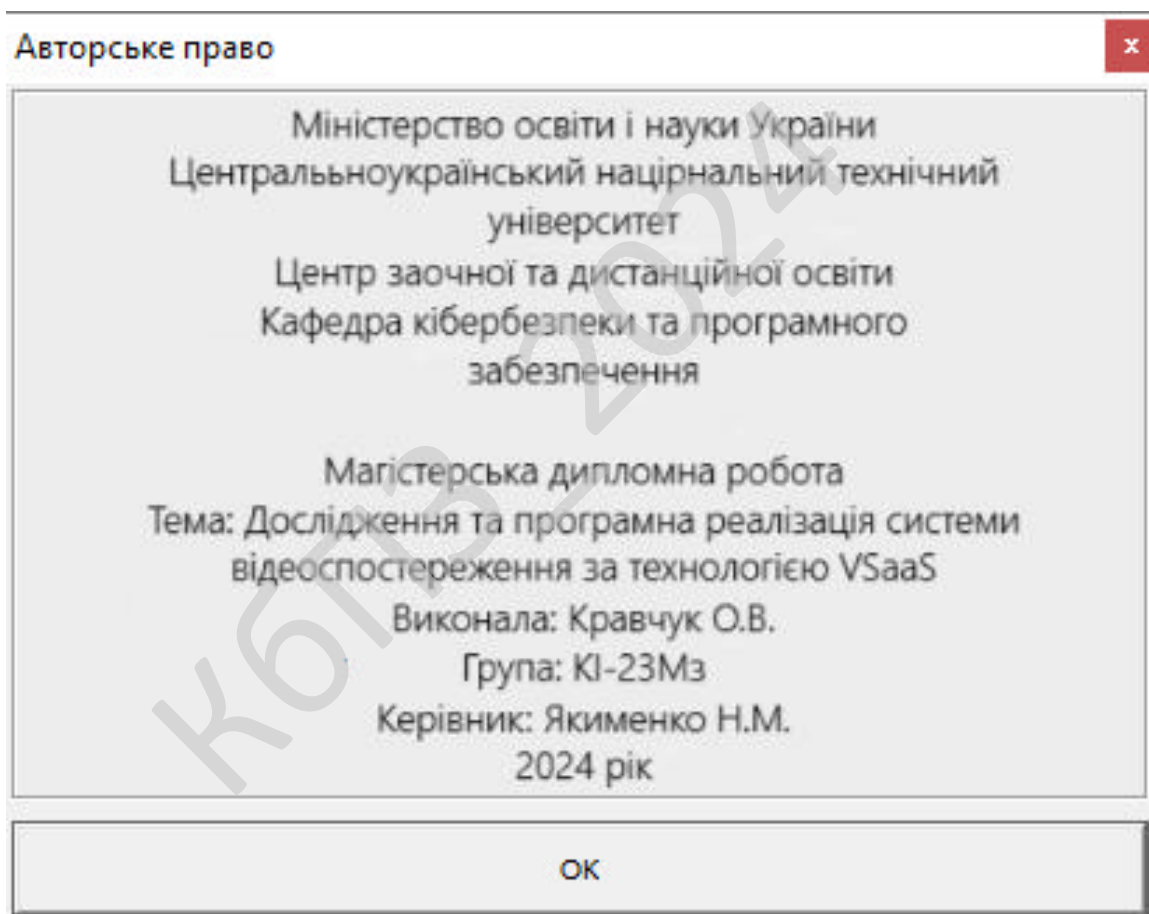


Рисунок 5.2 – Авторське право

Роздивимось процес встановлення програмного забезпечення, це процес підлаштування програмного забезпечення до відповідних умов застосування, а також інструктажу споживачів по роботі з програмним продуктом. Впровадження/встановлення програмного забезпечення це вся діяльність, яка

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

робить спроектовану програмну систему придатною до експлуатації. Цей процес є шматком життєвого циклу програмного забезпечення.

Взагалі, період розгортання створюється з кількох пов'язаних між собою дій із допустимими переходами між ними. Ця ініціативність може проявлятися як зі сторони виробника так і зі сторони споживача. Взагалі, всяка програмна система є винятковою, тому усі перебіги дій та маніпуляції під час встановлення передбачити неможливо. З цих причин, "розгортання" можна пояснювати як узагальнений процес стосовно відповідних вимог та характеристик. Розгортання може виконуватись спеціалістом одночасно з процесом розробки програмного забезпечення.

Діяльності, пов'язані з розгортанням програмного забезпечення це:

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

Під час встановлення програмного забезпечення необхідно брати до уваги наступні дії:

– Виокремлення критичних, в сенсі загального позитивного результату, маніпуляцій в роботі організації. Коли комплекс цих маніпуляцій виокремлений, потрібно насамперед застосовувати ІТ рішення для доведення операцій до автоматизму всередині самих же процедур. Таким способом, спроектоване ІТ рішення фактично і природньо стає кардинально важливим і потрібним для організації, і додатком буде гарантована публічність механізму запровадження;

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

– Збільшення нормативної бази альянсу методом включення до неї інструкцій, які окреслюють упорядкованість процедур автоматизованих процесів. Інакше є небезпека утворення розладу між автоматизованими операціями та іншими механізмами організації.

– Виконання робіт з загальної стандартизації наявної діяльності організації, коли виокремлюються найвідмінніші практики здійснення процедур і вмикаються в ІТ рішення за принципом максимальної користі для більшості співників. Процент схожих операцій відносно загального об'єму автоматизації може бути зовсім малий, але це доставляє процесу формування рішення вагомість в організації за рахунок збільшення його потрібності.

В процесі праці над програмою було виконано тестування програмного забезпечення, себто технічний аналіз, призначений для знаходження даних про якість продукту стосовно контексту, в якому планується його застосування.

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

Виконана оцінка:

- відповідність поставленим вимогам;
- вірна відповідь для всіх ймовірних вхідних даних;
- виконання функцій за відповідний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

З огляду на те, що чисельність ймовірних тестів для компонентів програми практично безкінечна, стратегія перевірки полягає в тому, щоб провести всі ймовірні проби із врахуванням поточного часу й ресурсів.

Як результат ПЗ перевірялось звичайним виконанням програми з ціллю знаходження збоїв чи інших дефектів.

Підібрано умови поширення – commercial software.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Програмне забезпечення, розроблене комерційною спілкою з ціллю здобуття доходу від його експлуатації іншими, наприклад, способом продажу копій.

Найбільш важливою якістю комерційних програмних продуктів є допомога крупним компаніям, яскраво зацікавлених у розповсюдженні програм. Більшість компаній пропонують винятково оплачувану підтримку своєї продукції. Для продукції, що реалізовується на комерційному ґрунті, як правило, діють безоплатні сервісні служби з метою збільшення рівня довіри у клієнтів і можливих споживачів.

Не у всіх, але в більшості, терміни критично важливих перемін в комерційній продукції набагато менші, ніж у некомерційних проєктів. Це зумовлено тим, що розробкою платних продуктів займається досить велика, спеціально підібрана група спеціалістів, це є їх прямим, достойно оплачуваним заняттям. Початківці ж, зазвичай, шукають додаткові варіанти заробітку, що збільшує час, затрачений на доповнення та зміни програм. Оскільки основною метою проєктування комерційного ПЗ є отримання доходу, то комерційні програмні продукти в числі перших заповнюють всі доступні ніші та рекомендують способи вирішення задач одразу.

Виокремлена категорія комерційних програм, розробка яких фінансується напряму замовником. Ці програми, як правило, позбавлені козирів комерційних продуктів через те, що обмежені визначеним бюджетом, але вони більше пристосовані до потреб замовника, аніж аналоги.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

6 НАУКОВА НОВИЗНА

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

Метою розробки є дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS.

Об'єктом дослідження є процес відеоспостереження за технологією VSaaS.

Предметом дослідження є методи відеоспостереження за технологією VSaaS

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

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

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи відеоспостереження за технологією VSaaS (Video Surveillance as a Service) можуть бути цікавими для аудиторії, представленої на рисунку 7.1.

Бізнес-сектор	Малі та середні підприємства (МСП) – шукають ефективні та економічні рішення для забезпечення безпеки своїх об'єктів.
	Великі корпорації – для централізованого управління системою відеоспостереження на кількох локаціях.
Установи, що займаються безпекою	Служби охорони – для моніторингу об'єктів клієнтів.
	Охоронні компанії – які надають послуги з моніторингу за моделлю "безпека як послуга".
Громадські та державні організації	Муніципальні органи – для управління безпекою місцевих об'єктів (камери вулиць, парків, зупинок).
	Правоохоронні органи – для розширення функціоналу моніторингу та аналізу відеоданих.
Житлово-комунальний сектор	ОСББ та житлові комплекси – для забезпечення безпеки мешканців та спільної території.
	Управляючі компанії – для оптимізації моніторингу інфраструктури.
IT-компанії	Розробники програмного забезпечення – для інтеграції нових рішень із хмарними платформами або IoT.
	Стартапи – які працюють над інноваційними підходами до безпеки та аналітики.
Технологічні партнери та інтегратори	Постачальники обладнання для відеоспостереження (камери, сенсори, мережеве обладнання).
	Партнери з розробки інфраструктурних рішень (наприклад, Amazon AWS, Microsoft Azure).
Індустрія роздрібної торгівлі та логістики	Торгові мережі – для моніторингу магазинів та аналітики поведінки клієнтів.
	Склади та логістичні центри – для забезпечення безпеки вантажів і операцій.
Освітні та медичні заклади	Школи та університети – для забезпечення безпеки студентів та викладачів.
	Медичні установи – для контролю за доступом до території та моніторингу загального порядку.
Приватні особи	Власники приватних будинків або квартир, зацікавлені у підключенні доступних хмарних систем відеоспостереження для особистих потреб.
Інвестори та аналітики	Зацікавлені в інвестиціях у VSaaS-технології, які мають значний потенціал зростання на ринку.

Рисунок 7.1 – Цільова аудиторія проекту

Це рішення може також викликати інтерес у стартапів, які шукають інноваційні бізнес-моделі, та організацій, які прагнуть цифрової трансформації у сфері безпеки.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Метод експертних оцінок дозволяє залучити кваліфікованих фахівців для аналізу ключових аспектів проєкту. Поетапно оцінка привабливості проєкту за цим методом виконується наступним чином.

1. Формування цілей оцінки. Ціль: визначити перспективність і конкурентоспроможність програмної реалізації системи VSaaS з погляду інвестиційної привабливості, ринкового потенціалу та технічної реалізації.

2. Визначення критеріїв оцінки. Експерти оцінюють проєкт за наступними критеріями (рисунок 7.2):



Рисунок 7.2 – Критерії експертної оцінки

3. Залучення експертів. До оцінки залучаються 5 експертів із таких сфер: розробка ПЗ; хмарні технології; ринок безпеки; інвестори; управління проєктами.

4. Ранжування критеріїв за важливістю. Експерти надають ваги кожному критерію за шкалою від 0 до 1, де сума всіх ваг дорівнює 1: $K_1 = 0.2$, $K_2 = 0.15$, $K_3 = 0.25$, $K_4 = 0.2$, $K_5 = 0.1$, $K_6 = 0.1$.

5. Оцінка проєкту за кожним критерієм. Кожен експерт оцінює проєкт за 10-бальною шкалою для кожного критерію.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Експерт 4	Експерт 5	Середня оцінка
K1	8	9	7	8	9	8.2
K2	6	7	8	6	7	6.8
K3	9	8	8	9	8	8.4
K4	8	8	7	8	7	7.6
K5	7	7	8	6	7	7.0
K6	6	5	7	6	5	5.8

6. Підрахунок інтегральної оцінки. Інтегральна оцінка проєкту розраховується як зважена сума середніх оцінок за формулою:

$I = \sum_{i=1}^n K_i * W_i$, де K_i – середня оцінка за критерієм, W_i – вага критерію.

Розрахунок:

$$I = (8.2 * 0.2) + (6.8 * 0.15) + (8.4 * 0.25) + (7.6 * 0.2) + (7.0 * 0.1) + (5.8 * 0.1) = 1.64 + 1.02 + 2.1 + 1.52 + 0.7 + 0.58 = 7.56$$

7. Інтерпретація результатів. Отримана інтегральна оцінка 7.56 (з 10 можливих) вказує на високу привабливість проєкту. Найсильніші сторони:

ринковий потенціал (8.4), інноваційність (8.2). Слабке місце: висока конкуренція (5.8). Для покращення варто посилити унікальність пропозиції для зниження конкуренції; залучити додаткові інвестиції для масштабування, акцентувати увагу на високий ринковий потенціал у комунікації з партнерами та інвесторами. Таким чином, метод експертних оцінок дозволяє комплексно оцінити привабливість проєкту й визначити ключові аспекти для його вдосконалення.

7.3 Вибір методу оцінки вартості ПЗ

Оцінка вартості програмної реалізації системи відеоспостереження за технологією VSaaS (Video Surveillance as a Service) може бути проведена за допомогою кількох методів. Найбільш підходящий метод залежить від цілей оцінки, стадії розробки проєкту та доступності даних.

Якщо проєкт на ранній стадії, доцільно використати метод аналогій для швидкої оцінки.

Якщо потрібно швидке рішення, але немає точних даних, залучіть експертів.

Метод аналогій (Comparative or Analogous Estimation). Цей метод базується на порівнянні із вартістю подібних проєктів. Застосовувати на початкових етапах проєкту, коли детальні дані про ресурсні потреби ще недоступні та для швидкої оцінки вартості.

Його ключові етапи: порівняти із витратами на розробку аналогічних систем відеоспостереження або інших хмарних сервісів та врахувати відмінності, як-от масштаб, складність функціоналу, використані технології.

Переваги: швидкість і простота.

Недоліки: залежність від доступності даних про аналогічні проєкти.

Метод експертних оцінок (Expert Judgment). Експерти оцінюють вартість проєкту на основі свого досвіду. Застосовується для невеликих проєктів або коли є обмежені дані.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Його ключові етапи: залучити експертів із програмної розробки, хмарних технологій та бізнес-аналітики та обговорити кожен аспект проєкту та узгодити оцінку. Переваги: швидкість та адаптивність. Недоліки: суб'єктивність оцінок.

Комбінація методів дозволить отримати більш точну та обґрунтовану оцінку вартості.

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

Оцінка економічної ефективності впровадження системи відеоспостереження за технологією VSaaS може враховувати такі аспекти, як зниження витрат, підвищення продуктивності, зменшення ризиків та покращення якості обслуговування. Зведені вхідні та розрахункові дані внесено в таблицю 7.1.

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Таблиця 7.2 – Основні показники впровадження проєкту

Вхідні дані для оцінки ефективності	
<p>Поточні витрати на систему відеоспостереження (традиційне обладнання):</p> <p>Закупівля обладнання: 500 000 грн. Встановлення: 100 000 грн. Обслуговування за рік: 50 000 грн. Енергоспоживання: 10 000 грн/рік.</p> <p>Вартість впровадження VSaaS:</p> <p>Підключення до платформи: 50 000 грн. Місячна абонплата: 10 000 грн/міс (120 000 грн/рік). Відсутність витрат на обслуговування та енергоспоживання.</p>	<p>Додаткові вигоди від VSaaS:</p> <p>Зменшення витрат на охорону (скорочення персоналу): 200 000 грн/рік. Покращення безпеки – скорочення втрат від крадіжок на 100 000 грн/рік. Період аналізу: 3 роки.</p>
Оцінка витрат після впровадження нової системи	
<p>Після впровадження нової системи на базі флеш-накопичувачів витрати змінилися:</p> <p>Витрати на обладнання та флеш-накопичувачі: \$5 000 на рік. Витрати на підтримку програмного забезпечення: \$7 000 на рік.</p>	<p>Час медичного персоналу на обробку даних зменшився на 50% завдяки автоматизації — \$10 000 на рік (економія \$10 000).</p> <p>Загальні витрати після впровадження: \$22 000 на рік.</p>
Розрахунок витрат	
<p>1. Традиційна система (без VSaaS):</p> <p>Сума витрат за 3 роки:</p> <p>Витрати на закупівлю та встановлення: 500 000 грн + 100 000 грн = 600 000 грн. Обслуговування та енергоспоживання: 50 000 грн/рік + 10 000 грн/рік × 3 роки = 180 000 грн. Загальні витрати: 600 000 грн + 180 000 грн = 780 000 грн.</p>	<p>2. Система VSaaS:</p> <p>Сума витрат за 3 роки:</p> <p>Витрати на підключення: 50 000 грн. Місячна абонплата: 120 000 грн/рік × 3 роки = 360 000 грн. Загальні витрати: 50 000 грн + 360 000 грн = 410 000 грн.</p>
Розрахунок економічного ефекту	
<p>Прямі вигоди:</p> <p>Економія на витратах охорони: 200 000 грн/рік × 3 роки = 600 000 грн. Скорочення втрат від крадіжок: 100 000 грн/рік × 3 роки = 300 000 грн.</p>	<p>Сумарна економія:</p> <p>Традиційна система: 780 000 грн – (600 000 грн + 300 000 грн) = -120 000 грн (збитки). VSaaS: 410 000 грн – (600 000 грн + 300 000 грн) = -490 000 грн (економія).</p>

Продовження таблиці 7.2 – Основні показники впровадження проєкту

	Чиста економія від впровадження VSaaS: 490 000 грн – 120 000 грн = 610 000 грн за 3 роки.
--	--

Впровадження системи відеоспостереження на базі VSaaS дозволяє: скоротити витрати на обладнання, обслуговування та охорону, забезпечити пряму економію у розмірі 610 000 грн за 3 роки, покращити контроль безпеки без необхідності в значних початкових інвестиціях. Це робить VSaaS економічно доцільним вибором, особливо для компаній із обмеженим бюджетом або потребою в гнучких масштабованих рішеннях.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Алгоритм просування проєкту програмної реалізації системи відеоспостереження за технологією VSaaS представимо схематично на рисунку 7.3.



Рисунок 7.3 – Алгоритм просування проєкту

2. Співпраця з ІТ-провайдерами:

- інтеграція системи у пропозиції хмарних сервісів ІТ-компаній.

3. Рітейл-мережі:

- розміщення пропозиції VSaaS у мережах, що продають системи безпеки та камери.

Оптимізація збуту й шляхів реалізації передбачає впровадження багатоканального підходу, активну роботу з партнерами та індивідуальний підхід до клієнтів. Використання сучасних технологій і акцент на автоматизацію процесів дозволить швидко масштабувати проєкт та забезпечити його економічну ефективність.

7.7 Визначення ключових факторів успіху конкретного проєкту

Успішна реалізація проєкту VSaaS (Video Surveillance as a Service) залежить від багатьох чинників. Ось основні ключові фактори, які забезпечують ефективність, привабливість і конкурентоспроможність такого проєкту:



Рисунок 7.4 – Ключові фактори успіху проєкту

Ключовими факторами успіху є забезпечення технологічної досконалості, клієнтоорієнтованості та ефективної бізнес-моделі. Проєкт, який пропонує

інноваційність, надійність, економічну вигідність та високий рівень обслуговування, матиме перевагу над конкурентами на ринку.

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Масштабне використання комп'ютерних технологій – яскрава ознака прогресу майже у всіх сферах діяльності людства. Сьогоднішній день, а тим паче близьке і далеке майбутнє, неможливо уявити без мобільного телефону, комп'ютера чи якихось інших технічних засобів. Дякуючи їм, ми маємо можливість швидко отримувати та передавати різні види інформації, обробляти великі обсяги інформації, виконувати точні розрахунки, збереження значних обсягів даних в електронному вигляді та чимало іншого.

Активне залучання комп'ютерів в сферу управління виробництвом, в банківську систему, бізнес, систему освіти, транспорт, сферу обслуговування призвело до стабільної, майже цілодобової взаємодії людей з комп'ютером, а часом, навіть до залежності. Постає цілком логічне запитання: чи безпечна ця взаємодія? Адже існує аксіома: всяка відносини людини й засобів праці є двохсторонніми.

Людина постійно розвивають засоби праці, а засоби праці в свою чергу безперервно впливають на людину. Тобто, абсолютно всі сучасні технології й техніка, до яких, звичайно, відносяться й комп'ютерні технології та ЕОМ, додають в наше життя певні можливі небезпеки. Звідси виникає потреба в адекватній оцінці умов та характеру праці, щоб розробити ефективні заходи та засоби для збереження здоров'я та забезпечення комфортних умов для цього.

Законодавством України чітко встановлено норми та вимоги щодо правил експлуатації комп'ютерної техніки на підприємствах, відповідно і охорона праці на підприємствах при роботі з комп'ютером, це «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», затверджені наказом Мінсоцполітики від 14.02.2018 № 207 [1], «Державні санітарні правила і

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98. [2].

Загальні вимоги пожежної безпеки під час використання комп'ютерної техніки встановлюють «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417) [3], комп'ютерних класів — пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.08.2016 № 974). [4] та інші державні стандарти, які регламентують експлуатування комп'ютерної техніки як радіоелектронного обладнання.

8.2 Шкідливі та небезпечні фактори при роботі з комп'ютером

Можна визначити кілька факторів, які впливають на стан здоров'я людей, працюючих за комп'ютером:

- сидяче положення протягом довгого відрізка часу;
- вплив електромагнітного випромінювання монітора;
- навантаження на зір, що призводить до втоми очей та провокує псування зору;
- перевантаження суглобів кистей;
- психо-емоційні перевантаження: стрес при втраті інформації, при виникненні чи допусканні критичних помилок.

У кожному випадку міра ризику прямо пропорційна часу, проведеному за монітором комп'ютера чи недалеко від нього. В сучасних умовах взаємодія людини стала складнішою, що вимагає переосмислення, нових підходів: людина, технічні засоби, середовище роботи мають розглядатися як єдина система. І це система «людина–комп'ютер–середовище».

Середовище виробництва має надзвичайно важливий вплив на працездатність та здоров'я користувачів комп'ютерів. Це середовище у виробничих приміщеннях (офісах), як правило, визначається мікрокліматом,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

За даними, які подано у табл. 8.1, можемо зробити висновок, що дані показники: площа та об'єм приміщення у розрахунку на одно робоче місце користувача ПК відповідає чинним нормам і вимогам.

Стосовно мікроклімату, то відповідно до ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [5] роботу з ПК можна віднести до категорії легка – 1а. Джерела тепла в такому приміщенні: люди, електрообладнання, освітлювальні механізми в темні години доби та опалювальна система в холодні пори року. Оператором виділяється до 120 ккал теплової енергії за годину. Допустимі та фактичні значення параметрів мікроклімату наведені в таблиці 8.2.

Таблиця 8.2 – Значення параметрів мікроклімату

Період року	Параметр	Оптимальний*	Фактичний
Теплий	Температура	23 – 25 ⁰ С	25 ⁰ С
	Вологість	40 – 60%	45%
	Швидкість повітря	< 0,1м/с	
Холодний	Температура	22 – 24 ⁰ С	22 ⁰ С
	Вологість	40 – 60%	50%
	Швидкість повітря	< 0,1м/с	

*ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

За даними показниками параметрів мікроклімату можемо зробити висновок, що норми дотримані.

Щодо освітлення, то відповідно з ДБН В.2.5-28:2006 «Природне і штучне освітлення» [6] ця робота відноситься до Va розряду зорових робіт. Використовується природне, штучне і змішане освітлення.

Природне освітлення забезпечується через вікно, площа якого складає $S' =$

2 м. х 1,6 м. = 3,2 м² і є бічним освітленням. У світильниках місцевого і загального освітлення вмонтовано світлодіодні лампи потужністю 20 Вт із світловим потоком однієї лампи 900 лм. Відповідно до вимірів рівень освітлення в нашому приміщенні та на робочому місці сягає меж 350 -500 лк, що цілком відповідає встановленій нормі.

Основним джерелом шуму в офісі є комп'ютер. Вентилятори (кулери) системного блоку, процесори, відеокарти і блоки живлення є сучасними і дають незначний рівень шуму. Відповідно до технічної документації шум, зумовлений кулером в блоці живлення, дає 25 дБ, кулером процесора – 30 дБ, загальний – 34 дБ. Беручи до уваги низький рівень шуму від ПК і низький рівень фонового шуму від іншого обладнання, можна констатувати, що загальний рівень шумового навантаження приміщення не вищий за встановлену норму та складає не більше 50 дБА, що відповідає рівню шуму для офісів з комп'ютерною технікою згідно Державних санітарних правил і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

В офісі немає джерел інфрачервоного, ультрафіолетового і електромагнітного випромінювання, бо монітор ПК створений на основі рідкокристалічної матриці, і підсвітка забезпечується неоновією лампою, яка не мають потужного електромагнітного випромінювання та є сертифікованою на території України.

Блок живлення – екранований та не виробляє вищезазначених видів випромінювання.

8.4 Розробка заходів з умов поліпшення охорони праці

Перечислимо заходи по забезпеченню умов праці на робочому місці користувача ПК.

Відносно забезпечення електробезпеки можна виокремити такі заходи: встановлення розподільних щитів зі спеціальними розетками з заземлюючими

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

контактами; організація заземлення для наявного обладнання; регулярна перевірка всіх приладів і пристроїв; проведення іспитів з охорони праці раз на рік. Відносно забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості: облаштування природної вентиляції, використанням дефлектора, щоб забезпечити повітряний обмін в приміщенні; монтаж системи центрального опалювання, для підтримки оптимальної температури в будь-яку пору року; організація штучного відповідного освітлення, щоб забезпечити комфортні та не шкідливі умови для зору, відповідні стандартам, оформлення паспорта на приміщення вузла, з фіксацією в ньому вимірювань світла та рівня звуку, проведених відділом охорони праці.

Окрім рекомендацій стосовно конкретного приміщення, де проводилось встановлення умов праці, є певні загальні вимоги, регламентовані відповідними нормативними документами.

Правильне облаштування робочих місць перешкоджає передчасній втомлюваності користувача та сприяє збереженню здоров'я. Облаштування робочого місця передбачає:

- вірне розташування робочого місця у виробничому офісі;
- вибір ергономічного обґрунтованого робочого положення, виробничих меблів з врахуванням анатомічних особливостей та індивідуальних характеристик людини;
- раціональне комплектування обладнання на робочих позиціях;
- врахування характеру та особливостей трудової діяльності. Відносно робочих місць працівника ВДТ, організація робочого повинна організовуватися відповідно до ДСанПіН 3.3.2-007-98. Щоб попередити перевтомлення очей, необхідно періодично виконувати комплекс спеціальних вправ для очей і, обов'язково, притримуватися правил розпорядку роботи та відпочинку. Для забезпечення цього моменту, на робочому місці був встановлений такий режим відпочинку: кожні дві години роботи – перерва для відпочинку, виконання вправ для очей.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

8.5 Протипожежний захист

Пожежі в офісах з оргтехнікою, що цілком логічно, становлять величезну небезпеку, тому що можуть нести за собою колосальні матеріальні збитки. Пожежа ймовірна при співдії горючих речовин та джерел запалювання. Горючі речовини це будівельні легкозаймісті матеріали, пластмасові корпуси техніки, проводи та шнури тощо. Можуть бути розетки, переноски, електронні схеми обладнання, пристроїв електроживлення, де можливе перегрівання елементів, та утворення електричних іскор, здатних викликати займання та стати причиною пожежі.

Під час виконання сервісних та профілактичних робіт часто застосовуються різні легкозаймісті речовини, прокладаються тимчасові електропровідники, виконуються спаювання. Створюються додаткові, сприятливі для виникнення пожежі, умови. Тому необхідно попередити це, застосувавши засоби протипожежної безпеки.

Є потреба створення спеціальних куточків, які мають містити найпростіші засоби для гасіння невеликих пожеж: вогнегасники, сухий пісок, азбестові ковдри. В офісах з комп'ютерами немає потреби облаштування систем автоматичного пожежогасіння, але переносні вуглекислотні вогнегасники мають бути в наявності та у вільному, швидкому доступі обов'язково, з розрахунку 2 шт. на кожні 20 м² приміщення.

Звукопоглинаюче облицювання стін, стель офісів потрібно робити з негорючих та важкозаймістих матеріалів.

Для своєчасного виявлення на перших стадіях джерел займання та запобігання розповсюдженню, рекомендовано застосовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають Правила пожежної безпеки. Також, в обов'язковому порядку, на випадок пожежі, у доступних місцях (двері, стенди) мають бути розміщені схеми евакуації з приміщення у випадку пожежі.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

8.6 Розрахункова частина

В офісі (де відсутні джерела виділення шкідливих речовин) працює дві людини. Робота пов'язана з використанням ПЕОМ. Розміри приміщення: $A = 5$ м, $B = 4.5$ м, $H = 2.8$ м, устаткування займає 15% об'єму. Визначимо найменшу необхідну кількість повітря для вентиляції.

Для офісів, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря ($\text{м}^3/\text{год.}$), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

$$L = L' \cdot N,$$

де L' – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення, $\text{м}^3/(\text{год.}\cdot\text{люд.})$;

N – кількість працюючих.

Питомий об'єм приміщення $V_{\text{п}}$, ($\text{м}^3/\text{люд.}$), визначається за формулою:

$$V_{\text{п}} = V/N,$$

де V – об'єм приміщення, м^3 .

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0,85 = 5 \cdot 4,5 \cdot 2,8 \cdot 0,85 = 53,55 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 53,55 / 2 = 33,2 \text{ м}^3/\text{люд.} > 26,77 \text{ м}^3/\text{люд.}$$

Нормована кількість повітря на одну людину при $V' > 20 \text{ м}^3/\text{люд.}$ становить $30 \text{ м}^3/(\text{год.}\cdot\text{люд.})$.

8.7 Висновки до розділу

У цьому розділі магістерської роботи виконано аналіз умов робітника, який працює з комп'ютером та оргтехнікою. Зроблено аналіз головних

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

санітарно – гігієнічних показників у певному приміщенні, в якому особа стабільно працює за комп'ютером. Організовані умови мають забезпечувати комфортну роботу. На основі розглянутої літератури з цієї проблеми, були визначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано мінімально потрібний об'єм повітря для вентиляції.

Виконання умов, які визначають оптимальну організацію робочих місць робітників, надасть можливість забезпечити високу працездатність протягом всього робочого часу.

КБПЗ – 2024

					VKPM-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

9 ОСНОВНІ ВИСНОВКИ

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

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 системи відеоспостереження за технологією VSaaS.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем системи відеоспостереження за технологією VSaaS.
- Досліджена система системи відеоспостереження за технологією VSaaS.
- На основі отриманих результатів досліджень створена програмна реалізація системи системи відеоспостереження за технологією VSaaS.

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

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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

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

Програма реалізована на мові високого рівня Delphi XE8. 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 RSA.

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.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – *Режим доступу до ресурсу:* <https://zakon.rada.gov.ua/rada/show/v0007282-98>.

3. Наказ Міністерства внутрішніх справ України 30.12.2014 №1417 «Про затвердження Правил пожежної безпеки України» – *Режим доступу до ресурсу* <https://zakon.rada.gov.ua/laws/show/z0252-15#Text>

4. Наказ Міністерства освіти і науки України 15.08.2016 № 974 «Про затвердження Правил пожежної безпеки для навчальних закладів та установ системи освіти України» – *Режим доступу до ресурсу* <https://zakon.rada.gov.ua/laws/show/z1229-16#Text>

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

6. Державні будівельні норми України: ДБН В.2.5-28:2018. – *Режим доступу до ресурсу:* <https://goo.su/9AkQ>

7. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. — 19 с. [Електронний ресурс]. – *Режим доступу*

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

<http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> .

8. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press. 2022 1677 p.

9. Will Grant. 101 UX Principles. Packt Publishing. 2022. 432 p.

10. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.

11. Henry Lloyd. Interactive Computer Graphics. States Academic Press. 2022. 247 p.

12. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB®With Applications to Pattern Recognition. CRC Press. 2021. 406 p.

13. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.

14. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009

15. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.

16. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.

17. 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.

18. 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. ун-т. – Д.: НГУ, 2016. – 187 с.

19. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.

20. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

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

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

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

53. 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.

54. 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.

55. 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

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

57. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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-123.24.0004.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Кравчук О.В.				Літ.	Аркуш	Аркушів
Перевірів	Якименко Н.М.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23Мз		
Затв.	Смірнов О.А.						
Дослідження та програмна реалізація системи відеоспостереження за технологією VSaaS					М	1	6

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи відеоспостереження за технологією VSaaS.

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 відеоспостереження за технологією VSaaS.

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-123.24.0004.00.00.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 відеоспостереження за технологією VSaaS.

- 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-123.24.0004.00.00.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 XP/Vista/7/8/10 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 XP/Vista/7/8/10.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi XE8

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

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

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинно бути розглянуто стандарти умов робітника, який працює з комп'ютером та оргтехнікою; зроблено аналіз головних санітарно – гігієнічних показників у приміщенні, в якому особа стабільно працює за комп'ютером; визначено оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також мінімально потрібний об'єм повітря для вентиляції.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 103 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 07.12.2024 р.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Якименко Н.М.

*Дослідження та програмна реалізація
системи відеоспостереження за технологією VSaaS*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 37

Літера: РП

Кропивницький – 2024 року

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - налаштування SET_DATA.pas

```

// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit SET_DATA;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, ExtCtrls, StdCtrls, DSPack, DSUtil, DirectShow9;

type // Об'ява власних типів та структур

  TForm2 = class(TForm) // Оголошення класу
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Button1: TButton;
    VideoSourceFilter: TFilter;
    DataSourceFilter: TFilter;
    DataFormats: TListBox;
    Label4: TLabel;
    VideoFormats: TListBox;
    Label3: TLabel;
    DataCapFilters: TListBox;
    Label2: TLabel;
    VideoCapFilters: TListBox;
    Label1: TLabel;
    InputLines: TComboBox;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure VideoCapFiltersClick(Sender: TObject);
    procedure DataCapFiltersClick(Sender: TObject);
    procedure VideoFormatsClick(Sender: TObject);
    procedure DataFormatsClick(Sender: TObject);
  private
    { Private секція }
  public
    { Public секція }
  end;

var
  Form2: TForm2;

Implementation // Реалізація

uses Unit1, LOG; // Які бібліотеки використовуємо у модулі

{$R *.dfm} // Підключення файлу ресурсів

procedure TForm2.Button1Click(Sender: TObject);
begin
  Form1.show;
  Form2.hide;
end;

procedure TForm2.FormShow(Sender: TObject);

begin
  Form1.ShowCAPdevise;

end;

procedure TForm2.VideoCapFiltersClick(Sender: TObject);
var
  PinList: TPinList;

```

```

i: integer;

begin

  CapEnum.SelectGUIDCategory(CLSID_VideoInputDeviceCategory);
  if VideoCapFilters.ItemIndex <> -1 then
  begin

    VideoSourceFilter.BaseFilter.Moniker :=
    CapEnum.GetMoniker(VideoCapFilters.ItemIndex);

    VideoSourceFilter.FilterGraph := Form1.CaptureGraph;
    Form1.CaptureGraph.Active := true;

    PinList := TPinList.Create(VideoSourceFilter as IBaseFilter);
    VideoFormats.Clear;

    VideoMediaTypes.Assign(PinList.First);
    for i := 0 to VideoMediaTypes.Count - 1 do
    VideoFormats.Items.Add(VideoMediaTypes.MediaDescription[i]);

    Form1.CaptureGraph.Active := false;
    PinList.Free;

    Form1.StartButton.Enabled := true;
    Form3.Memo1.Lines.Add('Налаштовано'+DataCapFilters.Items.Strings[VideoCapFi
    lters.ItemIndex]);
  end;
end;

procedure TForm2.DataCapFiltersClick(Sender: TObject);
var
  PinList: TPinList;
  g,i, LineIndex: integer;
  ABool: LongBool;

begin

  CapEnum.SelectGUIDCategory(CLSID_DataInputDeviceCategory);
  if DataCapFilters.ItemIndex <> -1 then

  begin
    DataSourceFilter.BaseFilter.Moniker :=
    CapEnum.GetMoniker(DataCapFilters.ItemIndex);

    DataSourceFilter.FilterGraph := Form1.CaptureGraph;
    Form1.CaptureGraph.Active := true;

  PinList := TPinList.Create(DataSourceFilter as IBaseFilter);
  DataFormats.Clear;
  i := 0;
  while i < PinList.Count do
    if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
    begin
      DataMediaTypes.Assign(PinList.Items[i]);
      PinList.Delete(i);
    end else inc(i);

    for i := 0 to DataMediaTypes.Count - 1 do
    begin
      DataFormats.Items.Add(DataMediaTypes.MediaDescription[i]);
    end;

    Form1.CaptureGraph.Active := false;
    InputLines.Clear;
    LineIndex := -1;
    for i := 0 to PinList.Count - 1 do
    begin
      InputLines.Items.Add(PinList.PinInfo[i].achName);

```

```
        with (PinList.Items[i] as IAMDataInputMixer) do get_Enable(ABool);
        if ABool then LineIndex := i;
    end;

    InputLines.ItemIndex := LineIndex;
    PinList.Free;
    Form1.StartButton.Enabled := true;
    g:=DataCapFilters.ItemIndex;

    Form3.Memo1.lines.add(' Налаштовано '+DataCapFilters.Items.Strings[g]);

    end;
end;

procedure TForm2.VideoFormatsClick(Sender: TObject);
begin
    if VideoCapFilters.ItemIndex <> -1 then
        begin
            Form3.Memo1.lines.add('Відео'+VideoCapFilters.Items.Strings[VideoCapFilters.Item
                Index]);
        end;
    end;

procedure TForm2.DataFormatsClick(Sender: TObject);
begin
    if DataFormats.ItemIndex <> -1 then
        begin
            Form3.Memo1.lines.add('Відео'+DataFormats.Items.Strings[DataFormats.ItemIndex]);
        end;
    end;

end.
```

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - зберігання відео VIDEO_GRAPH.pas

```
// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit VIDEO_GRAPH;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type // Об'ява власних типів та структур
  TForm4 = class(TForm) // Оголошення класу
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Button1: TButton;
  private
    { Private секція }
  public
    { Public секція }
  end;

var // Об'ява змінних
  Form4: TForm4;

Implementation // Реалізація

{$R *.dfm} // Підключення файлу ресурсів

end.
```

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - головна форма програми Unit.pas

```
// Copyrights © Тараканов Артем Олександрович KI-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації unit Unit1;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, ExtCtrls, StdCtrls, DSPack, DSUtil, DirectShow9,
  jpeg, Buttons;

type // Об'ява власних типів та структур
  TForm1 = class(TForm) // Оголошення класу
    // Компоненти
    Panell1: TPanel;
    StatusBar1: TStatusBar;
    MainMenu1: TMainMenu;
    N1231: TMenuItem;
    Panel2: TPanel;
    Panel3: TPanel;
    VideoWindow: TVideoWindow;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Button7: TButton;
    Button8: TButton;
    Button9: TButton;
    Button10: TButton;
    CaptureGraph: TFilterGraph;
    SaveDialog: TSaveDialog;
    StartButton: TBitBtn;
    Image1: TImage;
    Timer1: TTimer;
    StopButton: TBitBtn;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N14: TMenuItem;
    N15: TMenuItem;
    N16: TMenuItem;
    N17: TMenuItem;
    N18: TMenuItem;
    N19: TMenuItem;
    N20: TMenuItem;
    N21: TMenuItem;
    N22: TMenuItem;
    N23: TMenuItem;
    N24: TMenuItem;
    N25: TMenuItem;
    N26: TMenuItem;
    N27: TMenuItem;
    N28: TMenuItem;
    Timer2: TTimer;
```

```

Button11: TButton;
Label1: TLabel;
SampleGrabber: TSampleGrabber;
Label2: TLabel;
SaveDialog1: TSaveDialog;
BitBtn1: TBitBtn;
Image: TImage;
// Реалізація подій
procedure Button9Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure StartButtonClick(Sender: TObject);
procedure StopButtonClick(Sender: TObject);
procedure N8Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure N16Click(Sender: TObject);
procedure N18Click(Sender: TObject);
procedure N20Click(Sender: TObject);
procedure N22Click(Sender: TObject);
procedure N24Click(Sender: TObject);
procedure N26Click(Sender: TObject);
procedure N28Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SampleGrabberBuffer(sender: TObject; SampleTime: Double;
    pBuffer: Pointer; BufferLen: Integer);
procedure Button11Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
private
    { Private секція }
public //зона бачення змінних та типів даних
    procedure ShowCAPdevice;
    procedure SETKAS;
end;

var // Об'ява змінних
    Form1: TForm1;
    SETKA: integer;
    CapEnum: TSysDevEnum;
    VideoMediaTypes, DataMediaTypes: TEnumMediaType;
    Ok_normal: boolean;
    AR: array of array of tcolor;
    GO: boolean;

Implementation // Реалізація

uses About, LOG, SET_DATA; // Які бібліотеки використовуємо у модулі

{$R *.dfm} // Підключення файлу ресурсів

// Підпрограма виклику форми аворського права
procedure TForm1.Button9Click(Sender: TObject);
begin
    Form1.hide;
    Form3.Memo1.Lines.Add('Вызов просмотра о программе');
    AboutBox.show;
end;

procedure TForm1.Button1Click(Sender: TObject);

```

```

begin
    MessageDlg('Beta version', mtInformation, [mbOK], 0);
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
    Form1.hide;
    Form3.show;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
    S:string;
begin
    InputQuery('Внимание', 'Введите путь для сохранения видео', S);
    CapFile:=S;
    Form1.hide;
    Form2.show;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    Form1.hide;
    Form2.show;
end;

procedure TForm1.Button10Click(Sender: TObject);
begin
    Form1.Close;
end;

procedure TForm1.ShowCAPdevise;
var
    i:integer;
begin
    Form2.VideoCapFilters.Items.Clear;
    Form2.DataCapFilters.Items.Clear;
    CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
        Form2.VideoCapFilters.Items.Add(CapEnum.Filters[i].FriendlyName);
    CapEnum.SelectGUIDCategory(CLSID_DataInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
        Form2.DataCapFilters.Items.Add(CapEnum.Filters[i].FriendlyName);
    VideoMediaTypes := TEnumMediaType.Create;
    DataMediaTypes := TEnumMediaType.Create;
end;
// Підпрограма виклику інтервального спрацювання
procedure TForm1.Timer1Timer(Sender: TObject);
var
    position: int64;
    Hour, Min, Sec, MSec: Word;
Const // Константи
    MiliSecInOneDay = 86400000;
begin
    if CaptureGraph.Active then
        begin
            with CaptureGraph as IMediaSeeking do
                GetCurrentPosition(position);
                DecodeTime(position div 10000 / MiliSecInOneDay, Hour, Min, Sec, MSec);
                StatusBar1.panels[1].Text := Format('%d:%d:%d:%d', [Hour, Min, Sec, MSec]);
            end;
        end;
end;

procedure TForm1.StartButtonClick(Sender: TObject);
var
    multiplexer: IBaseFilter;
    Writer: IFileSinkFilter;
    PinList: TPinList;
    i: integer;

```

```

begin

    // Активувати фільтр
    CaptureGraph.Active := true;

    // налаштувати дані
    if Form2.DataSourceFilter.FilterGraph <> nil then
    begin
        PinList := TPinList.Create(Form2.DataSourceFilter as IBaseFilter);
        i := 0;
        while i < PinList.Count do
            if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
            begin
                if Form2.DataFormats.ItemIndex <> -1 then
                    with (PinList.Items[i] as IAMStreamConfig) do

SetFormat(DataMediaTypes.Items[Form2.DataFormats.ItemIndex].AMMediaType^);
PinList.Delete(i);
                end else inc(i);
                if Form2.InputLines.ItemIndex <> -1 then
                    with (PinList.Items[Form2.InputLines.ItemIndex] as IAMDataInputMixer) do
                        put_Enable(true);
                PinList.Free;
            end;

            // налаштування відео
            if Form2.VideoSourceFilter.FilterGraph <> nil then
            begin
                PinList := TPinList.Create(Form2.VideoSourceFilter as IBaseFilter);
                if Form2.VideoFormats.ItemIndex <> -1 then
                    with (PinList.First as IAMStreamConfig) do
                        SetFormat(VideoMediaTypes.Items[Form2.VideoFormats.ItemIndex].AMMediaType^);
                PinList.Free;
            end;
            // початок роботи
            with CaptureGraph as IcaptureGraphBuilder2 do
            begin
                // встановлення вихідного і'мя файлу
                SetOutputFileName(MEDIASUBTYPE_Avi, PWideChar(CapFile), multiplexer, Writer);
                // підключення
                if Form2.VideoSourceFilter.BaseFilter.DataLength > 0 then
                    RenderStream(@PIN_CATEGORY_PREVIEW, nil, Form2.VideoSourceFilter as IBaseFilter,
                        nil, VideoWindow as IBaseFilter);
                    if Form2.VideoSourceFilter.FilterGraph <> nil then
                        RenderStream(@PIN_CATEGORY_CAPTURE, nil, Form2.VideoSourceFilter as IBaseFilter,
                            nil, multiplexer as IBaseFilter);

                    if Form2.DataSourceFilter.FilterGraph <> nil then
                        begin
                            RenderStream(nil, nil, Form2.DataSourceFilter as IBaseFilter,
                                nil, multiplexer as IBaseFilter);
                        end;
                end;

            end;

            // вмикання захвату відеопотоку
            CaptureGraph.Play;
            StopButton.Enabled := true;
            StartButton.Enabled := false;
            Form2.DataFormats.Enabled := false;
            Form2.DataCapFilters.Enabled := false;
            Form2.VideoFormats.Enabled := false;
            Form2.VideoCapFilters.Enabled := false;

            Go:=false;
            Timer1.Enabled := true;
            VideoWindow.Visible:=false;
        end;
    end;

```

```
// Підпрограма відключення захвату відеопотоку
procedure TForm1.StopButtonClick(Sender: TObject);
begin
  Timer1.Enabled := false;
  StopButton.Enabled := false;
  StartButton.Enabled := true;
  CaptureGraph.Stop;
  CaptureGraph.Active := False;
  Form2.DataFormats.Enabled := true;
  Form2.DataCapFilters.Enabled := true;
  Form2.VideoFormats.Enabled := true;
  Form2.VideoCapFilters.Enabled := true;
end;

procedure TForm1.N8Click(Sender: TObject);
begin
  Form1.Close;
end;

procedure TForm1.N4Click(Sender: TObject);
begin
  Form1.hide;
  AboutBox.show;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
  MessageDlg('Betta version', mtInformation, [mbOK], 0);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  Form1.hide; // Приховати форму 1
  Form2.show; // Показати форму 2
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
  if Button7.Caption='B' then
  begin
    Go:=false;
    Timer2.Enabled:=true;
  end
  else
  begin
    Timer2.Enabled:=false;
  end;
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
  Form1.hide; // Приховати форму 1
  Form2.show; // Показати форму 2
end;

procedure TForm1.N16Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;
end;
```

```

procedure TForm1.N18Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;

end;

procedure TForm1.N20Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;

end;

procedure TForm1.N22Click(Sender: TObject);
begin
  Form1.hide;
  Form3.show;
end;

procedure TForm1.N24Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;
end;

procedure TForm1.N26Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;
end;

procedure TForm1.N28Click(Sender: TObject);
begin
  Form1.hide; // Приховати форму 1
  Form2.show; // Показати форму 2
end;

procedure TForm1.N9Click(Sender: TObject);
begin
Application.Minimize; // Приховання вікна ПЗ
end;

procedure TForm1.N10Click(Sender: TObject);
begin
if Ok_normal then
  begin
    MessageDlg('Ok', mtInformation, [mbOK], 0);
    end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
  X,Y:integer;
begin
  Ok_normal:=true;
  SETKA:=50;
  VideoWindow.Canvas.Pen.Color:=clLime;
  VideoWindow.Canvas.Pen.Width:=4;
  BL:=1;

  for x :=1 to 320 do
    begin
      if (x mod SETKA)=0 then
        begin
          inc (BL);
        end;
    end;
end;

```

```

        for y :=1 to 240 do
            begin
                if (y mod SETKA)=0 then
                    begin
                        inc(BL);
                    end;
                end;
            end;
        SetLength(Ar, 320, 240);
        end;

    procedure TForm1.SampleGrabberBuffer(sender: TObject; SampleTime: Double;
                                          pBuffer: Pointer; BufferLen: Integer);
    begin
        Image.Canvas.Lock;
        try
            SampleGrabber.GetBitmap(Image.Picture.Bitmap, pBuffer, BufferLen);
            SETKAS;
        finally
            Image.Canvas.Unlock;
        end;
        SampleGrabber.GetBitmap(Image2.Picture.Bitmap, pBuffer, BufferLen);
        Image2.Canvas.Rectangle(1,1,400,400);
    end;

    // Підпрограма захвату зображення з камери
    procedure TForm1.Button11Click(Sender: TObject);
    begin
        if SaveDialog1.Execute then
            begin
                if SaveDialog1.FileName<>' ' then
                    begin
                        SampleGrabber.GetBitmap(Image.Picture.Bitmap);
                        Image.Picture.SaveToFile(SaveDialog1.FileName);
                    end;
            end;
        end;

    // Підпрограма аналізу зображення накладенням сітки
    procedure TForm1.SETKAS;
    var
        x,y:integer;
        G:tcolor;
    begin
        //Опис як обробляються координати
        //X VideoWindow.ClientWidth
        //Y VideoWindow.ClientHeight
        VideoWindow.Canvas.Pixels[]
        Labell.Caption:=ColorToString(Image.Canvas.Pixels[20,20]);
        //-----
        if go and ((Image.Picture.Width<>0) and (Image.Picture.Height<>0))then
            begin
                for x :=1 to Image.Picture.Width do
                    begin
                        for y :=1 to Image.Picture.Height do
                            begin
                                if ((y mod SETKA)=0) and ((x mod SETKA)=0) then
                                    begin
                                        Image.Canvas.Rectangle(x, y, x+10, y+10);
                                        Labell.Caption:=ColorToString(VideoWindow.Canvas.Pixels[x, y]);
                                        //*****
                                        if Ar[X, Y]<>Image.Canvas.Pixels[x, y] then
                                            begin
                                                Ar[X, Y]:=Image.Canvas.Pixels[x, y];
                                                Form3.Memol.lines.add('');
                                            end;
                                        //*****
                                    end;
                                end;
                            end;
                        end;
                    end;
                if ((y mod SETKA)=0) and (x = 1) then

```

```

begin
  Image.Canvas.Rectangle(x,y,x+4,y+4);
  //*****
  if Ar[X,Y]<>Image.Canvas.Pixels[x,y] then
  begin
    Ar[X,Y]:=Image.Canvas.Pixels[x,y];
    Form3.Memo1.lines.add('');
  end;
  //*****
end;
if (y = 1) and ((x mod SETKA)=0) then
begin
  Image.Canvas.Rectangle(x,y,x+4,y+4);
  //*****
  if Ar[X,Y]<>Image.Canvas.Pixels[x,y] then
  begin
    Ar[X,Y]:=Image.Canvas.Pixels[x,y];
    Form3.Memo1.lines.add('');
  end;
  //*****
end;
if (y = 1) and (x = 1) then
begin
  Image.Canvas.Rectangle(x,y,x+4,y+4);
  //*****
  if Ar[X,Y]<>Image.Canvas.Pixels[x,y] then
  begin
    Ar[X,Y]:=Image.Canvas.Pixels[x,y];
    Form3.Memo1.lines.add('');
  end;
  //*****
end;
end;
end
end;
-----
//-----
if (not go)and ((Image.Picture.Width<>0) and (Image.Picture.Height<>0)) then
begin
  for x :=1 to Image.Picture.Width do
  begin
    for y :=1 to Image.Picture.Height do
    begin
      if ((y mod SETKA)=0) and ((x mod SETKA)=0) then
      begin
        Image.Canvas.Rectangle(x,y,x+10,y+10);
        Label1.Caption:=ColorToString(VideoWindow.Canvas.Pixels[x,y]);
        Ar[X,Y]:=Image.Canvas.Pixels[x,y];
      end;

      if ((y mod SETKA)=0) and (x = 1) then
      begin
        Image.Canvas.Rectangle(x,y,x+4,y+4);
        Ar[X,Y]:=Image.Canvas.Pixels[x,y];
      end;

      if (y = 1) and ((x mod SETKA)=0) then
      begin
        Image.Canvas.Rectangle(x,y,x+4,y+4);
        Ar[X,Y]:=Image.Canvas.Pixels[x,y];
      end;

      if (y = 1) and (x = 1) then
      begin
        Image.Canvas.Rectangle(x,y,x+4,y+4);
        Ar[X,Y]:=Image.Canvas.Pixels[x,y];
      end;
    end;
  end;
end;
end;

```

```
        end
    end;
end;
//-----
go:=true;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    Label1.Caption:=inttostr(Image.Picture.Width); // Діагональ зображення
    Label2.Caption:=inttostr(Image.Picture.Height); // горизонталь зображення
end;

end.
```

КБПЗ_2024

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - підключення плагінів PLUG.pas

```
// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit PLUG;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type // Об'ява власних типів та структур

  TForm_9 = class(TForm) // Оголошення класу

    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;

    DoT_A:TpluginsAddStrings;

    DoT_B:TpluginsAddFiles;

    DoT_C:TpluginsAddData;

  private
    { Private секція }

  public//зона бачення змінних та типів даних
    { Public секція }

  end;

var
  Form9: TForm9;

Implementation // Реалізація

{$R *.dfm} // Підключення файлу ресурсів

end.
```

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - бібліотека відео потоку DSPack.pas

```

// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit DSPack; // Бібліотека відео

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Classes, SysUtils, Messages, Graphics, Forms, Controls, ActiveX,
  DirectShow9, DirectDraw, DSUtil, ComCtrls, MMSystem, Math, Consts, ExtCtrls,
  MultiMon, Dialogs, Registry, SyncObjs, Direct3D9, WMF9;

Const // Константи
  WM_GRAPHNOTIFY = WM_APP + 1;

  WM_CAPTURE_BITMAP = WM_APP + 2;
type
// Об'ява типів
  TVideoMode = ( // Тип режиму
    vmNormal,
    vmVMR
  );

  TGraphMode = ( // Якості
    gmNormal,
    gmCapture,
    gmCAM
  );

{$IFDEF VER140}
  TVMRenderDevice = (
    rdOverlay = 1,
    rdVidMem = 2,
    rdSysMem = 4
  );
{$ELSE}
  TVMRenderDevice = Integer;
const
  rdOverlay = 1;
  rdVidMem = 2;
  rdSysMem = 4;
type
{$ENDIF}

  (@exclude)
  TGraphState = ( // Поточне дія
    gsUninitialized,
    gsStopped,
    gsPaused,
    gsPlaying
  );

  { Структура відео потоку }

// Абсолютна позиція у потоці
TSeekingCap = (
  CanSeekAbsolute,
  CanSeekForwards,
  CanSeekBackwards,
  CanGetCurrentPos,
  CanGetStopPos,
  CanGetDuration,
  CanPlayBackwards,
  CanDoSegments,
  IMediaSeeking.SetPositions).Source);

```

```

TSeekingCaps = set of TSeekingCap;

TVMRPreference = (
    vpForceOffscreen,
    vpForceOverlays,
    vpForceMixer,
    vpDoNotRenderColorKeyAndBorder,
    vpRestrictToInitialMonitor,
    vpPreferAGPMemWhenMixing
);

PVMRPreferences = ^TVMRPreferences;
TVMRPreferences = set of TVMRPreference;

TONDSEvent=procedure(sender: TComponent; Event, Param1, Param2: Integer) of
    object;
TONGraphBufferingData=procedure(sender: TObject; Buffering: boolean) of object ;
TONGraphComplete=procedure(sender: TObject; Result: HRESULT; Renderer:
IBaseFilter) of object;
    TONGraphDeviceLost          = procedure(sender: TObject; Device: IUnknown;
        Removed: Boolean) of object ;
    TONGraphEndOfSegment        = procedure(sender: TObject; StreamTime:
TReferenceTime; NumSegment: Cardinal) of object ;
    TONDSResult                 = procedure(sender: TObject; Result: HRESULT) of
object ;
    TONGraphFullscreenLost      = procedure(sender: TObject; Renderer:
IBaseFilter) of object ;
    TONGraphOLEEvent            = procedure(sender: TObject; String1, String2:
WideString) of object ;
    TONGraphOpeningFile         = procedure(sender: TObject; opening: boolean) of
object ;
    TONGraphSNDDDevError        = procedure(sender: TObject; OccurWhen:
TSndDevErr; ErrorCode: LongWord) of object ;
    TONGraphStreamControl       = procedure(sender: TObject; PinSender: IPin;
        Cookie: LongWord) of object ;
    TONGraphStreamError         = procedure(sender: TObject; Operation: HRESULT;
        Value: LongWord) of object ;
    TONGraphVideoSizeChanged    = procedure(sender: TObject; Width, height: word)
of object ;
    TONGraphTimeCodeAvailable= procedure(sender: TObject; From: IBaseFilter;
        DeviceID: LongWord) of object ;
    TONGraphEXTDeviceModeChange = procedure(sender: TObject; NewMode, DeviceID:
        LongWord) of object ;
    TONGraphVMRRenderDevice= procedure(sender: TObject; RenderDevice:
TVMRRenderDevice) of object;

TONCAMDataStreamChange         = procedure(sender: TObject; stream, lcid: Integer;
        Lang: string) of object;
    TONCAMCurrentTime           = procedure(sender: TObject; Hours,
        minutes,seconds,frames,frate : Integer) of object;
    TONCAMTitleChange           = procedure(sender: TObject; title: Integer) of object;
TONCAMChapterStart=procedure(sender: TObject; chapter: Integer) of object;
TONCAMValidUOPSChange = procedure(sender: TObject; UOPS: Integer) of object;
TONCAMChange = procedure(sender: TObject; total,current: Integer) of object;
TONCAMStillOn = procedure(sender: TObject; NoButtonAvailable: boolean; seconds:
        Integer) of object;
TONCAMSubpictureStreamChange = procedure(sender: TObject; SubNum, lcid: Integer;
        Lang: string) of object;
TONCAMPlaybackRateChange = procedure(sender: TObject; rate: single) of object;
TONCAMParentalLevelChange= procedure(sender: TObject; level: Integer) of object;
TONCAMAnglesAvailable=procedure(sender: TObject; available: boolean) of object;
TONCAMButtonAutoActivated=procedure(sender: TObject; Button: Cardinal) of
object;
TONCAMCMD=procedure(sender: TObject; CmdID: Cardinal) of object;
TONCAMCurrentHMSFTime = procedure(sender: TObject; HMSFTimeCode:
        TCAMHMSFTimeCode; TimeCode: TCAMTimeCode) of object;
TONCAMKaraokeMode=procedure(sender: TObject; Played: boolean) of object;
TONBuffer = procedure(sender: TObject; SampleTime: Double; pBuffer: Pointer;
        BufferLen: longint) of object ;

```

```

// *****
// Структура TFilterOperation
// *****

TFilterOperation = (
    foAdding,
    foAdded,
    foRemoving,
    foRemoved,
    foRefresh
);

IFilter = interface // Об'ява інтерфейсу
['{887F94DA-29E9-44C6-B48E-1FBF0FB59878}']
    function GetFilter: IBaseFilter;
    function GetName: string;
    procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
end;

TControlEvent = (
    cePlay,
    cePause,
    ceStop,
    ceFileRendering,
    ceFileRendered,
    ceCAMRendering,
    ceCAMRendered,
    ceActive
);

IEvent = interface // Інтерфейс
['{6C0DCD7B-1A98-44EF-A6D5-E23CBC24E620}']
    procedure GraphEvent(Event, Param1, Param2: integer);
    procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
end;

// *****
// Структура TFilterGraph
// *****

TFilterGraph = class(TComponent) // Оголошення класу
private
    FActive      : boolean;
    FAutoCreate  : boolean;
    FHandle      : THandle; // основний вказівник
    FMode        : TGraphMode;

    FFilters: TInterfaceList;
    FGraphEvents: TInterfaceList;

    // налаштування
    FFilterGraph : IGraphBuilder;
    FCaptureGraph : ICaptureGraphBuilder2;
    FCAMGraph     : ICamGraphBuilder;
    FMediaEventEx : IMediaEventEx;
    FGraphEdit    : boolean;
    FGraphEditID  : Integer;

    // Файл Log
    FLogFileName: String;
    FLogFile: TFileStream;
    FOnActivate: TNotifyEvent;
    // All Events Code
    FOnDSEvent : TOnDSEvent;
    // Generic Graph Events
    FOnGraphBufferingData : TOnGraphBufferingData;
    FOnGraphClockChanged  : TNotifyEvent;
    FOnGraphComplete      : TOnGraphComplete;

```

```

FOnGraphDeviceLost           : TOnGraphDeviceLost;
FOnGraphEndOfSegment        : TOnGraphEndOfSegment;
FOnGraphErrorStillPlaying   : TOnDSResult;
FOnGraphErrorAbort         : TOnDSResult;
FOnGraphFullscreenLost     : TOnGraphFullscreenLost;
FOnGraphChanged            : TNotifyEvent;
FOnGraphOleEvent           : TOnGraphOleEvent;
FOnGraphOpeningFile        : TOnGraphOpeningFile;
FOnGraphPaletteChanged     : TNotifyEvent;
FOnGraphPaused             : TOnDSResult;
FOnGraphQualityChange      : TNotifyEvent;
FOnGraphSNDDDevInError     : TOnGraphSNDDDevError;
FOnGraphSNDDDevOutError    : TOnGraphSNDDDevError;
FOnGraphStepComplete       : TNotifyEvent;
FOnGraphStreamControlStarted : TOnGraphStreamControl;
FOnGraphStreamControlStopped : TOnGraphStreamControl;
FOnGraphStreamErrorStillPlaying : TOnGraphStreamError;
FOnGraphStreamErrorStopped : TOnGraphStreamError;
FOnGraphUserAbort         : TNotifyEvent;
FOnGraphVideoSizeChanged   : TOnGraphVideoSizeChanged;
FOnGraphTimeCodeAvailable  : TOnGraphTimeCodeAvailable;
FOnGraphEXTDeviceModeChange : TOnGraphEXTDeviceModeChange;
FOnGraphClockUnset        : TNotifyEvent;
FOnGraphVMRRenderDevice    : TOnGraphVMRRenderDevice;

FOnCAMDataStreamChange     : TOnCAMDataStreamChange;
FOnCAMCurrentTime         : TOnCAMCurrentTime;
FOnCAMTitleChange         : TOnCAMTitleChange;
FOnCAMChapterStart        : TOnCAMChapterStart;
FOnCAMAngleChange         : TOnCAMChange;
FOnCAMValidUOPSChange     : TOnCAMValidUOPSChange;
FOnCAMButtonChange        : TOnCAMChange;
FOnCAMChapterAutoStop     : TNotifyEvent;
FOnCAMStillOn             : TOnCAMStillOn;
FOnCAMStilloff           : TNotifyEvent;
FOnCAMSubpictureStreamChange : TOnCAMSubpictureStreamChange;
FOnCAMNoFP_PGC            : TNotifyEvent;
FOnCAMPlaybackRateChange  : TOnCAMPlaybackRateChange;
FOnCAMParentalLevelChange : TOnCAMParentalLevelChange;
FOnCAMPlaybackStopped     : TNotifyEvent;
FOnCAMAnglesAvailable     : TOnCAMAnglesAvailable;
FOnCAMPlayPeriodAutoStop  : TNotifyEvent;
FOnCAMButtonAutoActivated : TOnCAMButtonAutoActivated;
FOnCAMCMDStart            : TOnCAMCMD;
FOnCAMCMDEnd              : TOnCAMCMD;
FOnCAMDiscEjected         : TNotifyEvent;
FOnCAMDiscInserted       : TNotifyEvent;
FOnCAMCurrentHMSFTime     : TOnCAMCurrentHMSFTime;
FOnCAMKaraokeMode         : TOnCAMKaraokeMode;
FOnCAMWarningInvalidCAM1_0Disc : TNotifyEvent;//=1,
FOnCAMWarningFormatNotSupported : TNotifyEvent;//=2,
FOnCAMWarningIllegalNavCommand : TNotifyEvent;//=3
FOnCAMWarningOpen        : TNotifyEvent;//=4
FOnCAMWarningSeek        : TNotifyEvent;//=5
FOnCAMWarningRead        : TNotifyEvent;//=6
FOnCAMDomainFirstPlay    : TNotifyEvent;
FOnCAMDomainVideoManagerMenu : TNotifyEvent;
FOnCAMDomainVideoTitleSetMenu : TNotifyEvent;
FOnCAMDomainTitle        : TNotifyEvent;
FOnCAMDomainStop         : TNotifyEvent;
FOnCAMErrorUnexpected    : TNotifyEvent;
FOnCAMErrorCopyProtectFail : TNotifyEvent;
FOnCAMErrorInvalidCAM1_0Disc : TNotifyEvent;
FOnCAMErrorInvalidDiscRegion : TNotifyEvent;
FOnCAMErrorLowParentalLevel : TNotifyEvent;
FOnCAMErrorMacrovisionFail : TNotifyEvent;
FOnCAMErrorIncompatibleSystemAndDecoderRegions : TNotifyEvent;
FOnCAMErrorIncompatibleDiscAndDecoderRegions : TNotifyEvent;
procedure HandleEvents;

```

```

procedure WndProc(var Msg: TMessage);
procedure SetActive(Activate: boolean);
procedure SetGraphMode(Mode: TGraphMode);
procedure SetGraphEdit(enable: boolean);
procedure ClearOwnFilters;
procedure AddOwnFilters;
procedure GraphEvents(Event, Param1, Param2: integer);
procedure ControlEvents(Event: TControlEvent; Param: integer = 0);
procedure SetLogFile(FileName: String);
function GetState: TGraphState;
function GetVolume: integer;
procedure SetVolume(Volume: Integer);
function GetBalance: integer;
procedure SetBalance(Balance: integer);
function GetSeekCaps: TSeekingCaps;
procedure SetRate(Rate: double);
function GetRate: double;
function GetDuration: integer;
protected
  procedure DoEvent(Event, Param1, Param2: Integer); virtual;
  procedure InsertFilter(AFilter: IFilter);
  procedure RemoveFilter(AFilter: IFilter);
  procedure InsertEventNotifier(AEvent: IEvent);
  procedure RemoveEventNotifier(AEvent: IEvent);
public //зона бачення змінних та типів даних
  property Duration: Integer read GetDuration; // завдання властивості типу
  property Rate: Double read GetRate write SetRate;
  property SeekCapabilities: TSeekingCaps read GetSeekCaps;
  property Balance: integer read GetBalance write SetBalance;
  property Volume: integer read GetVolume write SetVolume;
  property State: TGraphState read GetState;
  constructor Create(AOwner: TComponent); override; //Конструктор
  destructor Destroy; override; //Деструктор
  procedure Loaded; override;
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
  function Play: boolean;
  function Pause: boolean;
  function Stop: boolean;
  procedure DisconnectFilters;
  procedure ClearGraph;
  function RenderFile(FileName: WideString): HRESULT;
  function RenderFileEx(FileName: WideString): HRESULT;
  function RenderCAM(out status: TAMCAMRenderStatus;
  FileName: WideString = ''; Mode: Integer = AM_CAM_HWDEC_PREFER): HRESULT;

  procedure CAMSaveBookmark(BookMarkFile: WideString);
  procedure CAMRestoreBookmark(BookMarkFile: WideString);
published //зона бачення змінних
  property LogFile: String read FLogFileName write SetLogFile;
  property Active: boolean read FActive write SetActive default False;
  property AutoCreate: boolean read FAutoCreate write FAutoCreate default
False;
  property Mode: TGraphMode read FMode write SetGraphMode default gmNormal;
  property GraphEdit: boolean read FGraphEdit write SetGraphEdit;

  // -----
  // Події
  // -----

  property OnActivate: TNotifyEvent read FOnActivate write FOnActivate;
  property OnDSEvent: TOnDSEvent read FOnDSEvent write FOnDSEvent;

  property OnGraphBufferingData: TOnGraphBufferingData read
    FOnGraphBufferingData write FOnGraphBufferingData;
  property OnGraphClockChanged: TNotifyEvent read FOnGraphClockChanged write
FOnGraphClockChanged;
  property OnGraphComplete: TOnGraphComplete read FOnGraphComplete write
FOnGraphComplete;

```

```

property OnGraphDeviceLost: TOnGraphDeviceLost read FOnGraphDeviceLost write
FOnGraphDeviceLost;

property OnGraphEndOfSegment: TOnGraphEndOfSegment read FOnGraphEndOfSegment
write FOnGraphEndOfSegment;
property OnGraphErrorStillPlaying: TOnDSResult read
FOnGraphErrorStillPlaying write FOnGraphErrorStillPlaying;

property OnGraphErrorAbort: TOnDSResult read FOnGraphErrorAbort write
FOnGraphErrorAbort;

property OnGraphFullscreenLost: TOnGraphFullscreenLost read
FOnGraphFullscreenLost write FOnGraphFullscreenLost;

property OnGraphChanged: TNotifyEvent read FOnGraphChanged write
FOnGraphChanged;

property OnGraphOleEvent: TOnGraphOleEvent read FOnGraphOleEvent write
FOnGraphOleEvent;

property OnGraphOpeningFile: TOnGraphOpeningFile read FOnGraphOpeningFile
write FOnGraphOpeningFile;

property OnGraphPaletteChanged: TNotifyEvent read FOnGraphPaletteChanged
write FOnGraphPaletteChanged;

property OnGraphPaused: TOnDSResult read FOnGraphPaused write
FOnGraphPaused;

property OnGraphQualityChange: TNotifyEvent read FOnGraphQualityChange write
FOnGraphQualityChange;
property OnGraphSNDDevInError: TOnGraphSNDDevError read
FOnGraphSNDDevInError write FOnGraphSNDDevInError;
property OnGraphSNDDevOutError: TOnGraphSNDDevError read
FOnGraphSNDDevOutError write FOnGraphSNDDevOutError;
property OnGraphStepComplete: TNotifyEvent read FOnGraphStepComplete write
FOnGraphStepComplete;
property OnGraphStreamControlStarted: TOnGraphStreamControl read
FOnGraphStreamControlStarted write FOnGraphStreamControlStarted;
property OnGraphStreamControlStopped: TOnGraphStreamControl read
FOnGraphStreamControlStopped write FOnGraphStreamControlStopped;
property OnGraphStreamErrorStillPlaying : TOnGraphStreamError read
FOnGraphStreamErrorStillPlaying write FOnGraphStreamErrorStillPlaying;

property OnGraphStreamErrorStopped: TOnGraphStreamError read
FOnGraphStreamErrorStopped write FOnGraphStreamErrorStopped;
property OnGraphUserAbort: TNotifyEvent read FOnGraphUserAbort write
FOnGraphUserAbort;
property OnGraphVideoSizeChanged: TOnGraphVideoSizeChanged read
FOnGraphVideoSizeChanged write FOnGraphVideoSizeChanged;
property OnGraphTimeCodeAvailable: TOnGraphTimeCodeAvailable read
FOnGraphTimeCodeAvailable write FOnGraphTimeCodeAvailable;
property OnGraphEXTDeviceModeChange: TOnGraphEXTDeviceModeChange read
FOnGraphEXTDeviceModeChange write FOnGraphEXTDeviceModeChange;
property OnGraphClockUnset: TNotifyEvent read FOnGraphClockUnset write
FOnGraphClockUnset;
property OnGraphVMRRenderDevice: TOnGraphVMRRenderDevice read
FOnGraphVMRRenderDevice write FOnGraphVMRRenderDevice;
property OnCAMDataStreamChange: TOnCAMDataStreamChange read
FOnCAMDataStreamChange write FOnCAMDataStreamChange;
property OnCAMCurrentTime: TOnCAMCurrentTime read FOnCAMCurrentTime write
FOnCAMCurrentTime;
property OnCAMTitleChange: TOnCAMTitleChange read FOnCAMTitleChange write
FOnCAMTitleChange;
property OnCAMChapterStart: TOnCAMChapterStart read FOnCAMChapterStart write
FOnCAMChapterStart;
property OnCAMAngleChange: TOnCAMChange read FOnCAMAngleChange write
FOnCAMAngleChange;

```

```

    property OnCAMValidUOPSChange: TOnCAMValidUOPSChange read
FOnCAMValidUOPSChange write FOnCAMValidUOPSChange;
    property OnCAMButtonChange: TOnCAMChange read FOnCAMButtonChange write
FOnCAMButtonChange;
    property OnCAMChapterAutoStop: TNotifyEvent read FOnCAMChapterAutoStop write
FOnCAMChapterAutoStop;
    property OnCAMStillOn: TOnCAMStillOn read FOnCAMStillOn write FOnCAMStillOn;
    property OnCAMStilloff: TNotifyEvent read FOnCAMStilloff write
FOnCAMStilloff;
    property OnCAMSubpictureStreamChange: TOnCAMSubpictureStreamChange read
FOnCAMSubpictureStreamChange write FOnCAMSubpictureStreamChange;
    property OnCAMNoFP_PGC: TNotifyEvent read FOnCAMNoFP_PGC write
FOnCAMNoFP_PGC;
    property OnCAMPlaybackRateChange: TOnCAMPlaybackRateChange read
FOnCAMPlaybackRateChange write FOnCAMPlaybackRateChange;
    property OnCAMParentalLevelChange: TOnCAMParentalLevelChange read
FOnCAMParentalLevelChange write FOnCAMParentalLevelChange;
    property OnCAMPlaybackStopped: TNotifyEvent read FOnCAMPlaybackStopped write
FOnCAMPlaybackStopped;
    property OnCAMAnglesAvailable: TOnCAMAnglesAvailable read
FOnCAMAnglesAvailable write FOnCAMAnglesAvailable;
    property OnCAMPlayPeriodAutoStop: TNotifyEvent read FOnCAMPlayPeriodAutoStop
write FOnCAMPlayPeriodAutoStop;
    property OnCAMButtonAutoActivated: TOnCAMButtonAutoActivated read
FOnCAMButtonAutoActivated write FOnCAMButtonAutoActivated;
    property OnCAMCMDStart: TOnCAMCMD read FOnCAMCMDStart Write FOnCAMCMDStart;
    property OnCAMCMDEnd: TOnCAMCMD read FOnCAMCMDEnd Write FOnCAMCMDEnd;
    property OnCAMDiscEjected: TNotifyEvent read FOnCAMDiscEjected Write
FOnCAMDiscEjected;
    property OnCAMDiscInserted: TNotifyEvent read FOnCAMDiscInserted write
FOnCAMDiscInserted;
    property OnCAMCurrentHMSFTime: TOnCAMCurrentHMSFTime read
FOnCAMCurrentHMSFTime write FOnCAMCurrentHMSFTime;
    property OnCAMKaraokeMode: TOnCAMKaraokeMode read FOnCAMKaraokeMode write
FOnCAMKaraokeMode;
    property OnCAMDomainFirstPlay: TNotifyEvent read FOnCAMDomainFirstPlay write
FOnCAMDomainFirstPlay;
    property OnCAMDomainVideoManagerMenu: TNotifyEvent read
FOnCAMDomainVideoManagerMenu write FOnCAMDomainVideoManagerMenu;
    property OnCAMDomainVideoTitleSetMenu: TNotifyEvent read
FOnCAMDomainVideoTitleSetMenu write FOnCAMDomainVideoTitleSetMenu;
    property OnCAMDomainTitle: TNotifyEvent read FOnCAMDomainTitle write
FOnCAMDomainTitle;
    property OnCAMDomainStop: TNotifyEvent read FOnCAMDomainStop write
FOnCAMDomainStop;
    property OnCAMErrorUnexpected: TNotifyEvent read FOnCAMErrorUnexpected write
FOnCAMErrorUnexpected;
    property OnCAMErrorInvalidCAM1_0Disc: TNotifyEvent read
FOnCAMErrorInvalidCAM1_0Disc write FOnCAMErrorInvalidCAM1_0Disc;

    property OnCAMErrorInvalidDiscRegion: TNotifyEvent read
FOnCAMErrorInvalidDiscRegion write FOnCAMErrorInvalidDiscRegion;
    property OnCAMErrorLowParentalLevel: TNotifyEvent read
FOnCAMErrorLowParentalLevel write FOnCAMErrorLowParentalLevel;

    property OnCAMErrorMacrovisionFail: TNotifyEvent read
FOnCAMErrorMacrovisionFail write FOnCAMErrorMacrovisionFail;
    property OnCAMErrorIncompatibleSystemAndDecoderRegions: TNotifyEvent read
FOnCAMErrorIncompatibleSystemAndDecoderRegions write
FOnCAMErrorIncompatibleSystemAndDecoderRegions;
    property OnCAMErrorIncompatibleDiscAndDecoderRegions: TNotifyEvent read
FOnCAMErrorIncompatibleDiscAndDecoderRegions write
FOnCAMErrorIncompatibleDiscAndDecoderRegions;
    property OnCAMWarningInvalidCAM1_0Disc: TNotifyEvent read
FOnCAMWarningInvalidCAM1_0Disc write FOnCAMWarningInvalidCAM1_0Disc;
    property OnCAMWarningFormatNotSupported : TNotifyEvent read
FOnCAMWarningFormatNotSupported write FOnCAMWarningFormatNotSupported;
    property OnCAMWarningIllegalNavCommand : TNotifyEvent read
FOnCAMWarningIllegalNavCommand write FOnCAMWarningIllegalNavCommand;

```

```

    property OnCAMWarningOpen: TNotifyEvent read FOnCAMWarningOpen write
FOnCAMWarningOpen;
    property OnCAMWarningSeek: TNotifyEvent read FOnCAMWarningSeek write
FOnCAMWarningSeek;
    property OnCAMWarningRead: TNotifyEvent read FOnCAMWarningRead write
FOnCAMWarningRead;
end;

// *****
//  Структура TVMROptions
//  *****

TVideoWindow = class; // Оголошення класу

TVMRVideoMode = (
    vmrWindowed,
    vmrWindowless,
    vmrRenderless
);

TVMROptions = class(TPersistent) // Оголошення класу
private
    FOwner: TVideoWindow;
    FStreams: cardinal;
    FPreferences: TVMRPreferences;
    FMode: TVMRVideoMode;
    FKeepAspectRatio: boolean;
    procedure SetStreams(Streams: cardinal);
    procedure SetPreferences(Preferences: TVMRPreferences);
    procedure SetMode(AMode: TVMRVideoMode);
    procedure SetKeepAspectRatio(Keep: boolean);
public //зона бачення змінних та типів даних
    constructor Create(AOwner: TVideoWindow); //Конструктор
published //зона бачення змінних та типів даних
    property Mode: TVMRVideoMode read FMode write SetMode;
    // завдання властивості типу
    property Streams: Cardinal read FStreams write SetStreams default 4;
    property Preferences: TVMRPreferences read FPreferences write SetPreferences
default [vpForceMixer];
    property KeepAspectRatio: boolean read FKeepAspectRatio write
SetKeepAspectRatio default True;
end;

// *****
//  структура TVideoWindow
//  *****

TAbstractAllocator = class(TInterfacedObject) // Оголошення класу
    constructor Create(out hr: HRESULT; wnd: THandle; d3d: IDirect3D9 = nil;
d3dd: IDirect3DDevice9 = nil); virtual; abstract;
end;
TAbstractAllocatorClass = class of TAbstractAllocator; // Оголошення класу
TVideoWindow = class(TCustomControl, IFilter, IEvent) // Оголошення класу
private
    FMode          : TVMRVideoMode;
    FVMOOptions    : TVMROptions;
    FBaseFilter    : IBaseFilter;
    FVideoWindow   : IVideoWindow;
    FWindowLess    : IVMRWindowlessControl9;

    FFullScreen    : boolean;
    FFilterGraph   : TFilterGraph;
    FWindowState   : LongWord;
    FWindowStateEx : LongWord;
    FTopMost       : boolean;
    FIsFullScreen  : boolean;
    FOnPaint       : TNotifyEvent;
    FKeepAspectRatio: boolean;

```

```

FAllocatorClass: TAbstractAllocatorClass;
FCurrentAllocator: TAbstractAllocator;
FRenderLessUserID: Cardinal;
procedure SetVideoMode(AMode: TVideoMode);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
procedure SetFullScreen(Value: boolean);
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure GraphEvent(Event, Param1, Param2: integer);
function GetName: string;
function GetVideoHandle: THandle;
procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
procedure SetTopMost(TopMost: boolean);
function GetVisible: boolean;
procedure SetVisible(Vis: boolean);
protected
  {@exclude}
  procedure Loaded; override;
  {@exclude}
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
  {@exclude}
  procedure Resize; override;
  {@exclude}
  procedure ConstrainedResize(var MinWidth, MinHeight, MaxWidth, MaxHeight:
Integer); override;
  {@exclude}
  function GetFilter: IBaseFilter;
  {@exclude}
  procedure WndProc(var Message: TMessage); override;
  {@exclude}
  procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y:
Integer); override;
  {@exclude}
  procedure MouseMove(Shift: TShiftState; X, Y: Integer); override;
  {@exclude}
  procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
  {@exclude}
  procedure Paint; override;
public //зона бачення змінних та типів даних
  {@exclude}
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
  constructor Create(AOwner: TComponent);override; //Конструктор
  destructor Destroy; override; //Деструктор
  class function CheckVMR: boolean;
  function VMRGetBitmap(Stream: TStream): boolean;
  function CheckInputPinsConnected: boolean;
  procedure SetAllocator(Allocator: TAbstractAllocatorClass; UserID:
Cardinal);
  published //зона бачення змінних
  property OnPaint: TNotifyEvent read FOnPaint write FOnPaint;
  // завдання властивості типу
  property FullScreenTopMost: boolean read FTopMost write SetTopMost
    default false;
  property Mode: TVideoMode read FMode write SetVideoMode default vmNormal;
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
  property VideoHandle: THandle read GetVideoHandle;
  property VMROptions: TVMROptions read FVMROptions write FVMROptions;
  property FullScreen: boolean read FFullScreen write SetFullScreen
    default false;

  property Color;
  property Visible: boolean read GetVisible write SetVisible default True;
  property ShowHint;
  property Anchors;
  property Canvas;
  property PopupMenu;
  property Align;
  property TabStop default True;

```

```

property OnEnter;
property OnExit;
property OnKeyDown;
property OnKeyPress;
property OnKeyUp;
property OnCanResize;
property OnClick;
property OnConstrainedResize;
property OnDblClick;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp;
property OnMouseWheel;
property OnMouseWheelDown;
property OnMouseWheelUp;
property OnResize;
end;

TSampleGrabber = class; // Оголошення класу

TSampleGrabber = class(TComponent, IFilter,
                        ISampleGrabberCB) // Оголошення класу
private
    FOnBuffer: TOnBuffer;
    FBaseFilter: IBaseFilter;
    FFilterGraph : TFilterGraph;
    FMediaType: TMediaType;
    BMPInfo : PBitmapInfo;
    FCriticalSection: TCriticalSection;
    function GetFilter: IBaseFilter;
    function GetName: string;
    procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
    procedure SetFilterGraph(AFilterGraph: TFilterGraph);
    function SampleCB(SampleTime: Double; pSample: IMediaSample): HRESULT;
        stdcall;
    function BufferCB(SampleTime: Double; pBuffer: PByte; BufferLen: longint):
        HRESULT; stdcall;
protected
    procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public //зона бачення змінних та типів даних
    SampleGrabber: ISampleGrabber;
    InPutPin : IPin;
    OutPutPin : IPin;
    constructor Create(AOwner: TComponent); override; //Конструктор
    destructor Destroy; override; //Деструктор
    procedure UpdateMediaType;
    function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;

    procedure SetBMPCompatible(Source: PAMMediaType; SetDefault: cardinal);
    function GetBitmap(Bitmap: TBitmap; Buffer: Pointer; BufferLen: Integer):
boolean; overload;
    function GetBitmap(Bitmap: TBitmap): boolean; overload;
    class function CheckFilter: boolean;
published //зона бачення змінних
    property OnBuffer: TOnBuffer read FOnBuffer write FOnBuffer;
    property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
    // завдання властивості типу
    property MediaType: TMediaType read FMediaType write FMediaType;
end;

// *****
// структура TFilter
// *****
TFilter = class(TComponent, IFilter) // Оголошення класу
private
    FFilterGraph : TFilterGraph;

```

```

FBaseFilter: TBaseFilter;
FFilter: IBaseFilter;
function GetFilter: IBaseFilter;
function GetName: string;
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
protected
  { @exclude }
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public //зона бачення змінних та типів даних
  { Конструктор }
  constructor Create(AOwner: TComponent); override;
  { Деструктор method. }
  destructor Destroy; override; //Деструктор
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
published //зона бачення змінних та типів даних
  property BaseFilter: TBaseFilter read FBaseFilter write FBaseFilter;
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
end;

// *****
// структура TASFWriter
// *****

TASFWriter = class(TComponent, IFilter) // Оголошення класу
private
  FFilterGraph : TFilterGraph;
  FFilter      : IBaseFilter;
  FPort        : Cardinal;
  FMaxUsers    : Cardinal;
  FProfile     : TWMPofiles8;
  FFileName    : WideString;
  FAutoIndex   : boolean;
  FMultiPass   : boolean;
  FDontCompress: boolean;
  function GetProfile: TWMPofiles8;
  procedure SetProfile(profile: TWMPofiles8);
  function GetFileName: String;
  procedure SetFileName(FileName: String);
  function GetFilter: IBaseFilter;
  function GetName: string;
  procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
  procedure SetFilterGraph(AFilterGraph: TFilterGraph);
protected
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public //зона бачення змінних та типів даних
  WriterAdvanced2 : IWMWriterAdvanced2;
  { NetWork streaming configuration. }
  WriterNetworkSink : IWMWriterNetworkSink;
  { Вхідні дані }
  DataInput          : IPin;
  VideoInput         : IPin;
  DataStreamConfig   : IAMStreamConfig;
  VideoStreamConfig  : IAMStreamConfig;
  constructor Create(AOwner: TComponent); override; //Конструктор
  destructor Destroy; override; //Деструктор
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
published //зона бачення змінних
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
  property Profile: TWMPofiles8 read GetProfile write SetProfile;
  property FileName: String read GetFileName write SetFileName;
  // завдання властивості типу
  property Port: DWORD read FPort write FPort;
  property MaxUsers: DWORD read FMaxUsers write FMaxUsers;

```

```

    property AutoIndex      : boolean read FAutoIndex write FAutoIndex default
True;
    property MultiPass      : boolean read FMultiPass write FMultiPass default
False;
    property DontCompress: boolean read FDontCompress write FDontCompress
default False;

end;

// *****
//  структура TDSTackBar
//  *****

TTimerEvent = procedure(sender: TObject; CurrentPos, StopPos: Cardinal) of
                object ;

TDSTackBar = class(TTrackBar, IEvent) // Оголошення класу
private
    FFilterGraph: TFilterGraph;
    FMediaSeeking: IMediaSeeking;
    FWindowHandle: HWND;
    FInterval: Cardinal;
    FOnTimer: TTimerEvent;
    FEnabled: Boolean;
    FMouseDown: boolean;
    procedure UpdateTimer;
    procedure SetTimerEnabled(Value: Boolean);
    procedure SetInterval(Value: Cardinal);
    procedure SetOnTimer(Value: TTimerEvent);
    procedure SetFilterGraph(AFilterGraph: TFilterGraph);
    procedure GraphEvent(Event, Param1, Param2: integer);
    procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
    procedure TimerWndProc(var Msg: TMessage);
    property TimerEnabled: Boolean read FEnabled write SetTimerEnabled;
protected
    { @exclude }
    procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
    { @exclude }
    procedure MouseUp(Button: TMouseButton; Shift: TShiftState;
        X, Y: Integer); override;
    { @exclude }
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState;
        X, Y: Integer); override;
    { @exclude }
    procedure Timer; dynamic;
public //зона бачення змінних та типів даних
    { Конструктор }
    constructor Create(AOwner: TComponent); override;
    { Деструктор }
    destructor Destroy; override; //Деструктор
published //зона бачення змінних
    property FilterGraph: TFilterGraph read FFilterGraph Write SetFilterGraph;
    property TimerInterval: Cardinal read FInterval write SetInterval default
1000;
    property OnTimer: TTimerEvent read FOnTimer write SetOnTimer;
end;

{ @exclude }
TDSVideoWindowEx2 = class; // Оголошення класу

// *****
//  структура TColorControl
//  *****

TColorControl = class(TPersistent) // Оголошення класу
private
    FBrightness : Integer;
    FContrast   : Integer;

```

```

    FHue          : Integer;
    FSaturation   : Integer;
    FSharpness    : Integer;
    FGamma        : Integer;
    FUtilColor    : Boolean;
    FDefault      : TDDColorControl;
protected
{ Protected секція }
{ @exclude }
FOwner : TDSVideoWindowEx2;
{ @exclude }
Procedure SetBrightness(Value : Integer);
{ @exclude }
Procedure SetContrast(Value : Integer);
{ @exclude }
procedure SetHue(Value : Integer);
{ @exclude }
procedure SetSaturation(Value : Integer);
{ @exclude }
procedure SetSharpness(Value : Integer);
{ @exclude }
procedure SetGamma(Value : Integer);
{ @exclude }
procedure SetUtilColor(Value : Boolean);
{ @exclude }
function GetBrightness : Integer;
function GetContrast : Integer;
function GetHue : Integer;
function GetSaturation : Integer;
function GetSharpness : Integer;
function GetGamma : Integer;
{ @exclude }
function GetUtilColor : Boolean;
{ @exclude }
Procedure ReadDefault;
{ @exclude }
procedure UpdateColorControls;
{ @exclude }
procedure GetColorControls;
public //зона бачення змінних та типів даних
{ Public секція }
constructor Create(AOwner: TDSVideoWindowEx2); virtual; //Конструктор
procedure RestoreDefault;
Published //зона бачення змінних
property Brightness : Integer read GetBrightness write SetBrightness;
// завдання властивості типу
property Contrast : Integer read GetContrast write SetContrast;
property Hue : Integer read GetHue write SetHue;

property Saturation : Integer read GetSaturation write SetSaturation;

property Sharpness : Integer read GetSharpness write SetSharpness;

property Gamma : Integer read GetGamma write SetGamma;

property ColorEnable : Boolean read GetUtilColor write SetUtilColor;
end;

// *****
// структура TDSVideoWindowEx2Caps
// *****
TDSVideoWindowEx2Caps = class(TPersistent) // Оголошення класу
protected
{ Protected секція }
Owner : TDSVideoWindowEx2;
function GetCanOverlay : Boolean;
function GetCanControlBrigtness : Boolean;
function GetCanControlContrast : Boolean;
function GetCanControlHue : Boolean;

```

```

function GetCanControlSaturation : Boolean;
function GetCanControlSharpness : Boolean;
function GetCanControlGamma : Boolean;
function GetCanControlUtilizedColor : Boolean;
public
{ Public секція }
{ @exclude }
constructor Create(AOwner: TDSVideoWindowEx2); virtual; //Конструктор
published //зона бачення змінних
Property CanOverlayGraphic : Boolean read GetCanOverlay;
Property CanControlBrightness : Boolean read GetCanControlBrightness;
Property CanControlContrast : Boolean read GetCanControlContrast;
Property CanControlHue : Boolean read GetCanControlHue;

Property CanControlSaturation : Boolean read GetCanControlSaturation;
Property CanControlSharpness : Boolean read GetCanControlSharpness;
Property CanControlGamma : Boolean read GetCanControlGamma;
Property CanControlColorEnabled : Boolean read GetCanControlUtilizedColor;
end;

// *****
// структура TOverlayCallback
// *****

TOverlayCallback = class(TInterfacedObject, IDDrawExclModeVideoCallBack)
    AOwner : TObject; // Оголошення класу
    constructor Create(Owner : TObject); virtual; //Конструктор
    function OnUpdateOverlay(bBefore: BOOL; dwFlags: DWORD; bOldVisible: BOOL;
        var prcOldSrc, prcOldDest: TRECT; bNewVisible: BOOL; var prcNewSrc,
        prcNewDest: TRECT): HRESULT; stdcall;
    function OnUpdateColorKey(var pKey: TCOLORKEY; dwColor: DWORD): HRESULT;
        stdcall;
    function OnUpdateSize(dwWidth, dwHeight, dwARWidth, dwARHeight: DWORD):
        HRESULT; stdcall;
end;

// *****
// структура TDSVideoWindowEx2
// *****

TRatioModes = (rmStretched, rmLetterBox, rmCrop);

TOverlayVisibleEvent = procedure (Sender: TObject; Visible : Boolean) of
object;

{ @exclude }
TCursorVisibleEvent = procedure (Sender: TObject; Visible : Boolean) of
object;
TDSVideoWindowEx2 = class(TCustomControl, IFilter, IEvent) // Оголошення класу
private
    FVideoWindow          : IVideoWindow;
    FFilterGraph          : TFilterGraph;
    FBaseFilter           : IBaseFilter;
    FOverlayMixer         : IBaseFilter;
    FVideoRenderer       : IBaseFilter;
    FDDXM                 : IDDrawExclModeVideo;
    FFullScreen           : Boolean;
    FTopMost              : Boolean;
    FColorKey             : TColor;
    FWindowState         : LongWord;
    FWindowStateEx       : LongWord;
    FVideoRect           : TRect;
    FOnPaint              : TNotifyEvent;
    FOnColorKey          : TNotifyEvent;
    FOnCursorVisible     : TCursorVisibleEvent;
    FOnOverlay           : TOverlayVisibleEvent;
    FColorControl        : TColorControl;
    FCaps                 : TDSVideoWindowEx2Caps;
    FZoom                : Integer;

```

```

FAspectMode      : TRatioModes;
FNoScreenSaver   : Boolean;
FIdleCursor      : Integer;
FMonitor         : TMonitor;
FFullscreenControl : TForm;
GraphWasUpdatet  : Boolean;
FOldParent       : TWinControl;
OverlayCallback  : TOverlayCallback;
GraphBuildOK     : Boolean;
FVideoWindowHandle : HWND;
LMousePos        : TPoint;
LCursorMov       : DWord;
RememberCursor   : TCursor;
IsHidden         : Bool;
FOverlayVisible  : Boolean;
OldDesktopColor  : Longint;
OldDesktopPic    : String;
FDesktopPlay     : Boolean;
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure GraphEvent(Event, Param1, Param2: integer);
function GetName: string;
procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
procedure SetTopMost(TopMost: boolean);
procedure SetZoom(Value : Integer);
function UpdateGraph : HRESULT;
function GetVideoInfo : HRESULT;
procedure SetAspectMode(Value : TRatioModes);
procedure FullScreenCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure SetVideoZOrder;
protected
  {@exclude}
  function GetFilter: IBaseFilter;
  {@exclude}
  procedure resize; override;
  {@exclude}
  procedure Loaded; override;
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
  {@exclude}
  procedure WndProc(var Message: TMessage); override;
  procedure Paint; override;
  procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y:
Integer); override;
  procedure MouseMove(Shift: TShiftState; X, Y: Integer); override;
  procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
  procedure MyIdleHandler(Sender: TObject; var Done: Boolean);
  procedure RefreshVideoWindow;
public //зона бачення змінних та типів даних
  constructor Create(AOwner: TComponent); override; //Конструктор
  destructor Destroy; override; //Деструктор

  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
  procedure ClearBack;
  procedure StartDesktopPlayback; overload;
  procedure StartDesktopPlayBack(OnMonitor : TMonitor); overload;
  procedure NormalPlayback;
  procedure StartFullScreen; overload;
  procedure StartFullScreen(OnMonitor : TMonitor); overload;
  property FullScreen: boolean read FFullScreen;
  property DesktopPlayback : Boolean Read FDesktopPlay;
  property Canvas;
  property ColorKey : TColor read FColorKey;
  property Capabilities : TDSVideoWindowEx2Caps read FCaps;
  property OverlayVisible : Boolean read FOverlayVisible;
published //зона бачення змінних
  property AspectRatio : TRatioModes read FAspectMode write SetAspectMode;

```

```

property AutoHideCursor : Integer read FIdleCursor write FIdleCursor;
// завдання властивості типу
property DigitalZoom : Integer read FZoom write SetZoom;
property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
property FullScreenTopMost: boolean read FTopMost write SetTopMost default
false;
property OnColorKeyChanged: TNotifyEvent read FOnColorKey write FOnColorKey;
property ColorControl: TColorControl read FColorControl write FColorControl;

property NoScreenSaver: Boolean read FNoScreenSaver write FNoScreenSaver;

property OnOverlayVisible: TOverlayVisibleEvent read FOnOverlay write
FOnOverlay;

property OnPaint : TNotifyevent read FOnPaint Write FOnPaint;
property OnCursorShowHide : TCursorVisibleEvent read FOnCursorVisible write
FOnCursorVisible;

property Color;
property Visible;
property ShowHint;
property Anchors;
property PopupMenu;
property Align;
property TabStop default True;
property OnEnter;
property OnExit;
property OnKeyDown;
property OnKeyPress;
property OnKeyUp;
property OnCanResize;
property OnClick;
property OnConstrainedResize;
property OnDbClick;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp;
property OnMouseWheel;
property OnMouseWheelDown;
property OnMouseWheelUp;
property OnResize;
end;

////////////////////////////////////
// Оголошення класу TVMRBitmap
////////////////////////////////////
type // Об'ява власних типів та структур

TVMRBitmapOption=(
  vmrbDisable,
  vmrbSrcColorKey,
  vmrbSrcRect);

TVMRBitmapOptions = set of TVMRBitmapOption;

TVMRBitmap = class // Класи
private
  FVideoWindow: TVideoWindow;
  FCanvas: TCanvas;
  FVMR9ALPHABITMAP: TVMR9ALPHABITMAP;
  FOptions: TVMRBitmapOptions;
  FBMPold: HBITMAP;
  procedure SetOptions(Options: TVMRBitmapOptions);
  procedure ResetBitmap;
  procedure SetAlpha(const Value: Single);
  procedure SetColorKey(const Value: COLORREF);
  procedure SetDest(const Value: TVMR9NormalizedRect);
  procedure SetDestBottom(const Value: Single);
  procedure SetDestLeft(const Value: Single);
  procedure SetDestRight(const Value: Single);

```

```

procedure SetDestTop(const Value: Single);
procedure SetSource(const Value: TRect);
function GetAlpha: Single;
function GetColorKey: COLORREF;
function GetDest: TVMR9NormalizedRect;
function GetDestBottom: Single;
function GetDestLeft: Single;
function GetDestRight: Single;
function GetDestTop: Single;
function GetSource: TRect;
public //зона бачення змінних та типів даних
constructor Create(VideoWindow: TVideoWindow); //Конструктор
destructor Destroy; override; //Деструктор
procedure LoadBitmap(Bitmap: TBitmap);
procedure LoadEmptyBitmap(Width, Height: Integer; PixelFormat: TPixelFormat;
    Color: TColor);
procedure Draw;
procedure DrawTo(Left, Top, Right, Bottom, Alpha: Single; doUpdate: boolean
    = false);
procedure Update;
property Canvas: TCanvas read FCanvas write FCanvas;
property Alpha: Single read GetAlpha write SetAlpha;
property Source: TRect read GetSource write SetSource;
property DestLeft : Single read GetDestLeft write SetDestLeft;
property DestTop : Single read GetDestTop write SetDestTop;
property DestRight : Single read GetDestRight write SetDestRight;
property DestBottom : Single read GetDestBottom write SetDestBottom;
property Dest: TVMR9NormalizedRect read GetDest write SetDest;
// завдання властивості типу COLORREF
property ColorKey: COLORREF read GetColorKey write SetColorKey;
property Options: TVMRBitmapOptions read FOptions write SetOptions;
end;
End.

```

Під час написання програмного продукту були використанні бібліотеки DirectShow:

comlite.h, errors.h, dv.h, strmif.h, mmstream.h, amstream.h,
ddstream.h, astream.h, mpconfig.h, control.h, qnetwork.h,
playlist.h, il21dec.h, amvideo.h, amData.h, vptype.h
sbe.idl, wmdxva.h, vmr9.idl

КБПЗ_2024

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування Video_Stream.dpr

```
// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
program VideoStream; // Назва програми

uses // Які бібліотеки використовуємо у модулі

    Forms,
    SysUtils,
    Unit1 in 'Unit1.pas' {Form1},
    LOG in 'LOG.pas' {Form3},
    VIDEO_GRAPH in 'VIDEO_GRAPH.pas' {Form4},
    SET_DATA in 'SET_DATA.pas' {Form2},
    SPLASH_WINDOW in 'SPLASH_WINDOW.pas' {Form7},
    About in 'About.pas' {AboutBox};

{$R *.res} /// Підключення ресурсного файлу проекту

Begin
    Form7:=TForm7.Create(Application); // Створення форми заставки
    Form7.Show; // Виведення форми заставки
    Form7.Update; // Оновлення
try
    Application.Initialize; // Ініціалізація ПЗ
    Application.CreateForm(TForm1, Form1);
    // Створення форм
        Application.CreateForm(TForm3, Form3);
        Application.CreateForm(TForm4, Form4);
        Application.CreateForm(TForm2, Form2);
        Application.CreateForm(TAboutBox, AboutBox);
    finally
        Form7.Free; // Видалення форми заставки
    end;
Application.Run; // Початок роботи розробленого ПЗ

end.
```

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - модуль статистики роботи LOG.pas

```

// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit LOG;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type // Об'ява власних типів та структур
  TForm3 = class(TForm) // Оголошення класу
    Panel1: TPanel;
    Panel2: TPanel;
    Button1: TButton;
    Memo1: TMemo;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private секція }
  public //зона бачення змінних та типів даних
    { Public секція }
  end;

var
  Form3: TForm3;

implementation // Реалізація

uses Unit1; // Які бібліотеки використовуємо у модулі
{$R *.dfm} // Підключення файлу ресурсів

procedure TForm3.Button1Click(Sender: TObject);
begin
  Form1.show;
  Form3.hide;
end;

procedure TForm3.FormCreate(Sender: TObject);
begin
  Form3.Memo1.lines.add('Ok'+DateToStr(now));
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
  Form3.Memo1.lines.Clear;
end;

end.

```

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування – вікно заставки SPLASH_WINDOW.pas

```
// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit SPLASH_WINDOW;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls;

type // Об'ява власних типів та структур

  TForm7 = class(TForm) // Оголошення класу
    Image1: TImage;
    ScrollBox1: TScrollBox;
  private
    { Private секція }
  public
    { Public секція }
  end;

var
  Form7: TForm7;

Implementation // Реалізація

{$R *.dfm} // Підключення файлу ресурсів

end.
```

Файл програмного забезпечення, розробленого у ході виконання дипломного проектування - авторське право About.pas

```
// Copyrights © Тараканов Артем Олександрович КІ-15-2СК 2017 ЦНТУ, кафедра
програмування та захисту інформації
unit About;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
Buttons, ExtCtrls;

type // Об'ява власних типів та структур
  TAboutBox = class(TForm) // Оголошення класу

    Panell: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    OKButton: TButton;
    procedure OKButtonClick(Sender: TObject);
  private
    { Private секція }
  public
    { Public секція }
  end;

var
  AboutBox: TAboutBox;

Implementation // Реалізація

uses Unit1; // Які бібліотеки використовуємо у модулі
{$R *.dfm} // Підключення файлу ресурсів

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
  Form1.show;
  AboutBox.hide;
end;

end.
```