

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації
(повна назва)

Кафедра Радіотехнологій інформаційно-комунікаційних систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Проектування інформаційної системи відеоспостереження
(тема)

Виконав:

студент II курсу, групи АПСм-22-1

Адаменко В.О.

(прізвище, ініціали)

Спеціальність 126 Інформаційні системи та технології

(код і повна назва спеціальності)

Освітня програма Архітектурне проектування інформаційних систем

(повна назва освітньої програми)

Керівник доцент Зарудний О.А.

(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри

(підпис)

Зарудний О.А.

(прізвище, ініціали)

2024 р

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 126 Інформаційні системи та технології
(код і повна назва)

Освітня програма Архітектурне проектування інформаційних систем
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Адаменку Владиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування інформаційної системи відеоспостереження

затверджена наказом по університету від **3 жовтня 2023 р. № 1295 Ст**

2. Термін подання студентом роботи до екзаменаційної комісії **10 січня 2024 р.**

3. Вихідні дані до роботи створити проектне рішення з реалізацією прототипу системи відеоспостереження.

4. Перелік питань, що потрібно опрацювати в роботі

1) Огляд аналогічних проектних рішень

2) Проектування за методом АТАМ

3) Опис проектних рішень за методом 4+1

4) Реалізація прототипу на базі проектних рішень

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____
Ком'ютерна презентація _____

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доцент Зарудний О.А.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ	06.10.2023	
2	Огляд аналогічних рішень	26.10.2023	
3	Проектування за методами АТАМ та 4+1	05.12.2023	
4	Реалізація системи	30.12.2023	
5	Задача на перевірку та підпис кваліфікаційної роботи керівнику	10.01.2024	

Дата видачі завдання **4 жовтня 2023 р.**

Студент _____
(підпис)

Адаменко В.О. _____
(прізвище та ініціали)

Керівник роботи _____
(підпис)

доцент каф. РТІКС Зарудний О.А.
(посада, прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 79 с. , 28 рисунків, джерел 28 джерел посилення та 3 додатки.

ВІДЕОСПОСТЕРЕЖЕННЯ, ПРОЕКТУВАННЯ, ПРОТОТИП, КАМЕРА, ДОДАТОК, ЗАХИСТ.

Об'єктом розробки є проектування системи відеоспостереження визначеними методами 4+1 та АТАМ.

Предмет розробки – системи відеоспостереження та її програмне забезпечення.

Мета роботи – розроблення прототипу системи відеоспостереження з проектною частиною та який забезпечує захист об'єкту захисту.

У процесі виконання кваліфікаційної роботи було спроектовано інформаційну систему відеоспостереження та реалізовано програмне забезпечення для різних видів використання відеокамери.

Система спроектована та реалізована таким чином, що будь-який користувач може застосовувати її у різних додатках. Якщо користувачу достатньо однієї камери він має можливість приєднати її до телеграм боту та отримувати результат звітти. Якщо користувач має задачу з охорони якогось великого об'єкту де кількість камер більша, він може використати розроблене програмне забезпечення та додавати групу камер.

Система самостійно може реагувати на рух у полі зору датчика руху та камери та записувати події з надсиланням та збереженням відеоданих.

ABSTRACT

Explanatory note: 79 p. , 28 figures, 28 source references, and 3 appendices.

VIDEO SURVEILLANCE, DESIGN, PROTOTYPE, CAMERA, APPLICATION, PROTECTION.

The object of development is a video surveillance design system using prominent 4+1 and ATAM methods.

The subject of development is video surveillance systems and their software.

The purpose of the work is to develop a prototype of a video surveillance system with a design part that provides protection of the object of protection.

As part of the qualification work, a video surveillance information system was designed and software for various types of video camera use was implemented.

The system is designed and implemented in such a way that any user can use it in various applications. If one camera is enough for the user, he has the option to attach it to the bot's telegram and get the result from there. If the user has the task of guarding a large object with more cameras, he can use the developed software and add a group of cameras.

The system can independently respond to movement in the image field of the motion sensor and camera and record events by sending and saving video data.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	7
ВСТУП	8
1 ОГЛЯД АНАЛОГІЧНИХ ПРОЕКТНИХ РІШЕНЬ	10
1.1 Централізована архітектура в системах відеоспостереження	10
1.2 Розподілена архітектура системи відеоспостереження.....	11
1.3 Хмарна архітектура в системах відеоспостереження.....	13
1.4 Архітектура розподілених вузлів у системах відеоспостереження	15
1.5 Мережева архітектура в системах відеоспостереження	17
1.6 Вибір оптимальної архітектури для системи відеоспостереження	18
2 ПРОЕКТУВАННЯ ЗА МЕТОДОМ АТАМ	20
2.1 Метод компромісного аналізу архітектури АТАМ	20
2.2 Функціональні вимоги системи.....	21
2.3 Стейкхолдери системи їх цілі та зацікавленість у системі.....	22
2.4 Обмеження системи	23
2.5 Атрибути якості системи.....	24
3 ОПИС ПРОЕКТНИХ РІШЕНЬ ЗА МЕТОДОМ 4+1	26
3.1 Загальні відомості стосовно методу 4+1	26
3.2 Логічний вид.....	27
3.3 Вид розробки	28
3.4 Процесний вид.....	30
3.5 Фізичний вид	31
3.6 Сценарії використання	34
4 РЕАЛІЗАЦІЯ ПРОТОТИПУ НА БАЗІ ПРОЕКТНИХ РІШЕНЬ.....	36
4.1 Розробка та реалізація бази даних.....	36
4.2 Реалізація програмного забезпечення відеокамери.....	38

4.3	Реалізація програмного забезпечення системи відеоспостереження	42
	ВИСНОВКИ.....	47
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	48
	ДОДАТОК А.....	Error! Bookmark not defined.
	ДОДАТОК Б	Error! Bookmark not defined.
	ДОДАТОК В.....	Error! Bookmark not defined.

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БД – База даних;
СУБД - система управління базами даних;
API – Application programming interface;
ARM – Advanced RISC Machine;
АТАМ - Architecture tradeoff analysis method;
HDMI – High-Definition Multimedia Interface;
IP – Internet Protocol;
LAN - local area network;
NVR – мережевий відеореєстратор;
OTG – On The Go;
PIR – Passive Infrared Sensor;
RPI – Raspberry Pi;
SQL - Structured query language;
USB - Universal Serial Bus;
WiFi – wireless network protocols;
.NET – dot net;

ВСТУП

Проектування системи відеоспостереження - це комплексний процес де потрібно визначати усі можливі фактори застосування рішення та забезпечення безпеки визначеного об'єкту. Цей процес передбачає створення інфраструктури для зафіксування, зберігання, обробки та аналізу відеоданих для забезпечення неперервної безпеки та контролю об'єкту.

Для поточного завдання потрібно спроектувати інформаційну систему відеоспостереження. У цій темі поєднується дві основні частини системи. Перша частина складається з системи, яка буде відображувати поточний стан відеокамер та зберігати їх запис у хмарі задля збереження доказів. Ця хмара може бути як і локальна задля зменшення витрат по створенню цієї системи, так і інтегрована з будь якого хмарного сервісу, якщо є кошти. Друга частина складається з самого рішення відеокамери. Вона може бути різного типу з різним типом приєднання до загальної системи. Також вибір відеокамери залежить від її застосування на об'єкті який потрібно захистити.

Для проектування та реалізації наявного завдання буде використовуватись два методи архітектурного аналізу та документування.

Першим методом було обрано АТАМ, що у розшифруванні звучить як метод компромісного аналізу. Цей метод аналізу та оцінки архітектурних рішень для програмного забезпечення. Його ціль - допомогти командам розробників і архітекторам приймати обґрунтовані рішення щодо архітектури, враховуючи різні архітектурні альтернативи. АТАМ був розроблений в Середній лабораторії Інституту технічних проблем програмного забезпечення США та став популярним методом в галузі архітектурної оцінки.

Другим методом є 4+1. Метод "4+1" - це модель проектування програмних систем, яка поєднує п'ять різних видів документації для опису архітектури програмної системи. Цей підхід був розроблений за часів Рациональної корпорації (Rational Corporation) і став популярним в області

розробки програмного забезпечення для опису архітектурних аспектів системи. Подання використовується для опису системи з кількох точок зору.

Звіт розробленої кваліфікаційної роботи, буде містити

У першому розділі будуть описуватись огляд аналогічних архітектурно проектувальних рішень на базі яких буде проводитись аналіз та розуміння майбутньої системи.

У другому розділі буде відбуватись проектування за методом АТАМ з визначенням функціональних вимог до системи відеоспостереження та визначення атрибутів якості системи з обмеженнями.

Третій розділ буде наступним етапом проектування за допомогою методу 4+1 буде виведено:

- логічний вид;
- вид розробки;
- вид процесів;
- фізичний вид.

На базі різних видів архітектурний та проектних рішень можна буде реалізовувати систему відеоспостереження, саме це буде відбуватись у останньому розділі кваліфікаційної роботи.

1 ОГЛЯД АНАЛОГІЧНИХ ПРОЕКТНИХ РІШЕНЬ

1.1 Централізована архітектура в системах відеоспостереження

Централізована архітектура в системах відеоспостереження передбачає те, що всі відеокамери підключаються до центрального сервера або мережевого відеореєстратора (NVR). Цей сервер відповідає за обробку, збереження і управління відеоданими, які надходять з камер. Цій архітектурі притаманні такі характеристики та переваги:

- централізований контроль де оператори можуть моніторити всі камери спостереження та керувати ними з центрального місця;

- відеодані з камер зазвичай зберігаються в центральному сервері або NVR, що дозволяє легко здійснювати резервне копіювання та керувати збереженням даних;

- в централізованій архітектурі відеодані передаються через мережу лише один раз – від камер до центрального сервера що дозволяє зменшити навантаження на мережу, оскільки не потрібно передавати відеопотік з кожної камери окремо;

- обробка відеоданих, така як розпізнавання облич, аналіз поведінки або інші алгоритми, може виконуватися на центральному сервері, що дозволяє використовувати потужну обчислювальну техніку для складних завдань;

- в централізованій системі відеоспостереження потрібно менше обладнання для обробки та збереження відеоданих, що може призвести до зниження загальних витрат.

На рисунку 1.1 зображено схематичний приклад централізованої архітектури відеоспостереження.

В перевагах можна одразу побачити мінуси такого архітектурного рішення де централізована архітектура має обмеження. Де наприклад у разі відмови центрального сервера уся система може перестати працювати. Тому для великих систем та систем з високими вимогами доцільніше буде

використання інших архітектурних рішень таких як розподілена або хмарна архітектура [1].

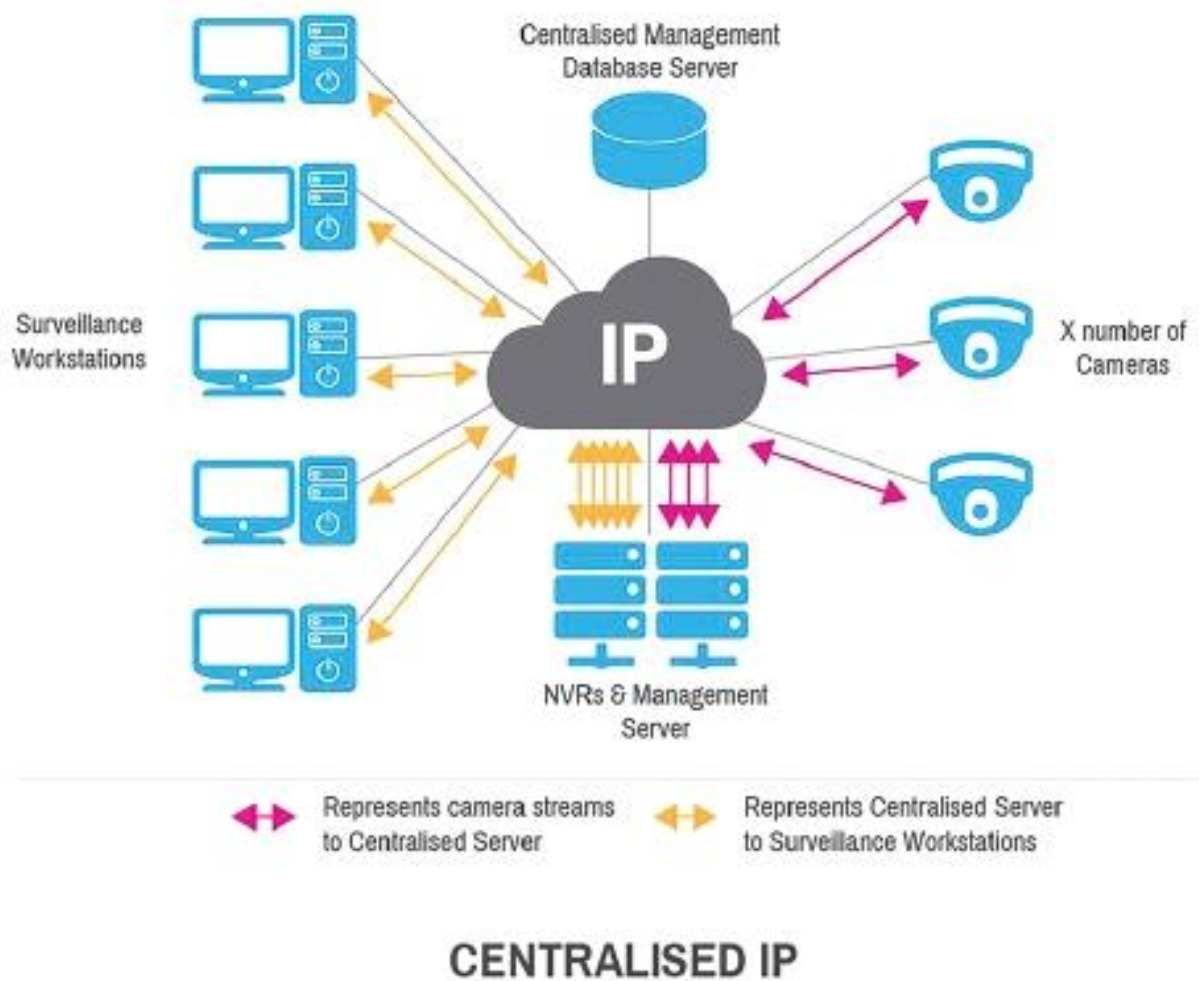


Рисунок 1.1 – Централізована система відеоспостереження

1.2 Розподілена архітектура системи відеоспостереження

Розподілена архітектура в системах відеоспостереження передбачає, що обробка відеоданих, збереження і управління системою розподіляються між різними серверами або вузлами, розташованими на об'єкті відеоспостереження. Цей підхід дозволяє досягти більшої гнучкості і масштабованості у порівнянні з централізованою архітектурою.

Цій архітектурі притаманні такі характеристики та переваги:

- в розподіленій архітектурі обробка відеоданих може виконувати на різних серверах, які розташовані біля камер або на інших стратегічних

пунктах, що приводить до зменшення навантаження на центральний сервер та дозволяє обробляти дані на місці їхнього виникнення;

-розподілена архітектура може бути більш надійною, оскільки відмова одного сервера не призводить до припинення всієї системи. Також резервне копіювання та розміщення дублікатів даних можуть забезпечити високій рівень доступності;

-архітектура розподіленої системи відеоспостереження дозволяє легко розширювати систему шляхом додавання нових серверів або вузлів без значних змін у структуру системи це особливо корисно для великих систем відеоспостереження;

-розподілена архітектура зазвичай зменшує обсяг трафіку в мережі, оскільки відеодані передаються та обробляються на рівні локальної мережі, а не через інтернет.

-розподілена архітектура забезпечує зменшення ризику несанкціонованого доступу до даних через те що відеодані можуть залишатись на об'єкті відеоспостереження;

Не зважаючи на переваги, розподілена архітектура також може бути складнішою у реалізації та управлінні порівняно з централізованою архітектурою. Також розподілена архітектура вимагає докладного планування та конфігурації для досягнення оптимального результату проектування [2].

На рисунку 1.2 зображена розподілена архітектура системи відеоспостереження.

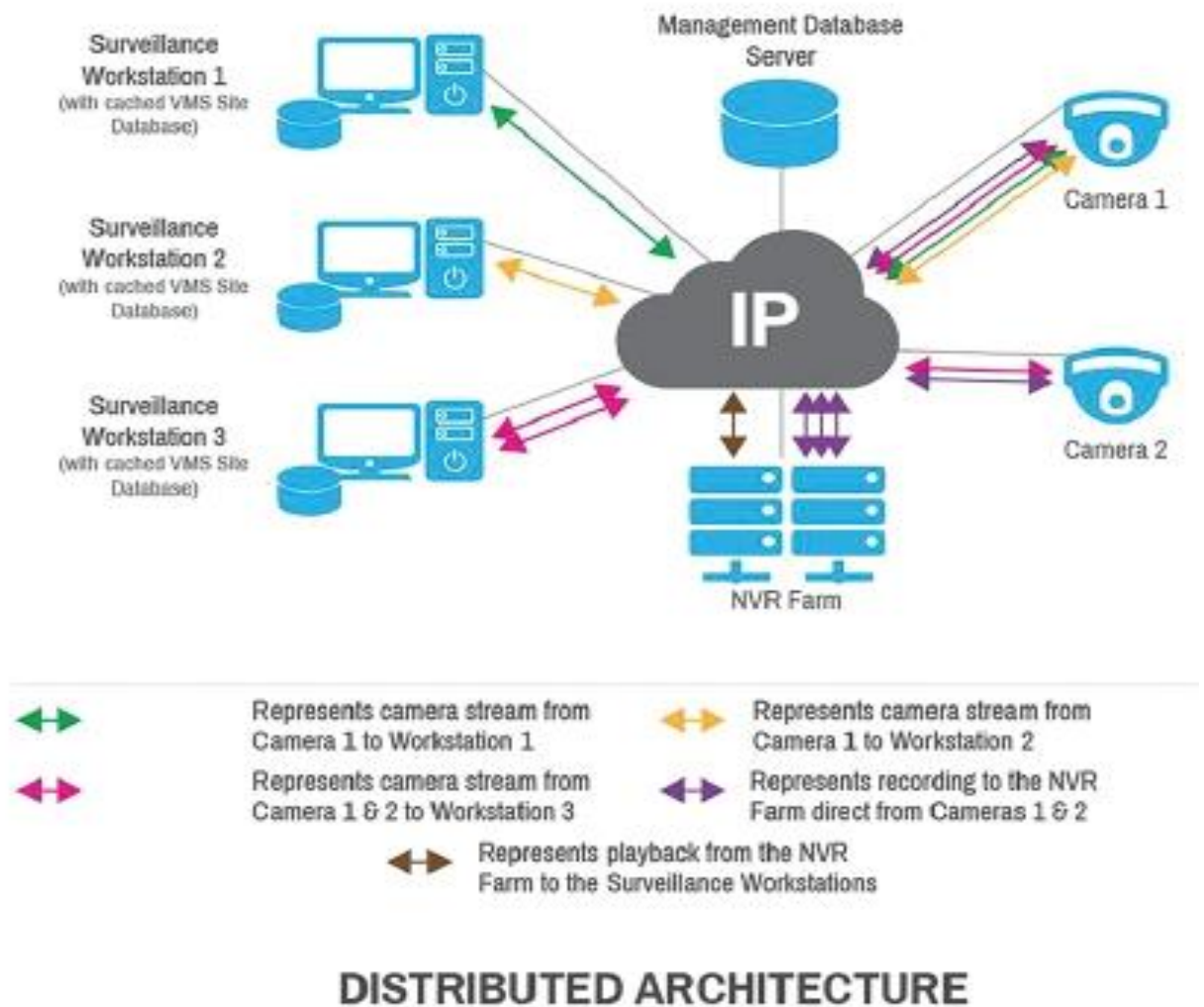


Рисунок 1.2 – Розподілена система відеоспостереження

1.3 Хмарна архітектура в системах відеоспостереження

Хмарна архітектура в системах відеоспостереження передбачає збереження, обробку та керування відеоданими на віддалених серверах, які розташовані в хмарних дата-центрах. Цей підхід надає численні переваги, особливо для користувачів, які шукають високу доступність, гнучкість та простоту використання.

Цій архітектурі притаманні такі характеристики та переваги:

-хмарні системи відеоспостереження можуть легко масштабуватись, дозволяючи додавати нові камери та вузли без значних зусиль це особливо

важливо для компаній та організацій зі зростаючими потребами відеоспостереження;

-хмарні рішення зазвичай гарантують високий рівень доступності завдяки резервному копіюванню та забезпеченню високої доступності на рівні дата-центрів;

-хмарні системи дозволяють здійснювати віддалений доступ до відеоданих та живих відеопотоків з будь-якого пристрою з інтернет-з'єднанням це особливо корисно для віддаленого моніторингу та керування;

-власники систем можуть уникнути великих витрат на покупку та обслуговування серверів та обладнання, оскільки вони можуть використовувати інфраструктуру хмарного постачальника;

-хмарні постачальники зазвичай надають автоматичні оновлення та технічну підтримку, що може спрощувати управління системою та забезпечує безпеку та актуальність програмного забезпечення;

-відеодані зберігаються в захищених хмарних сховищах, що зменшує ризик втрати даних внаслідок обриву жорсткого диска або інших технічних факторів, які можуть вплинути на відеодані;

-хмарні системи зазвичай підтримують стандартні протоколи і API для інтеграції з іншими системами безпеки та управління;

-збереження та обробка даних у хмарі може зменшити витрати на експлуатацію системи в порівнянні з традиційними системами [3].

За всіма перевагами хмарної архітектури в системах відеоспостереження важливо враховувати аспекти безпеки, конфіденційності та швидкості інтернет-з'єднання, оскільки відеодані передаються через мережу до хмари це досить велике навантаження на загальну систему передачу даних та може досить сильно знизити пропускну спроможність [4]. Також слід звернути увагу на обслуговування та ціну хмарного рішення. Воно може варюватись від масштабів об'єкту який підлягає захисту за допомогою системи відеоспостереження.

На рисунку 1.3 зображена схема хмарної системи відеоспостереження.

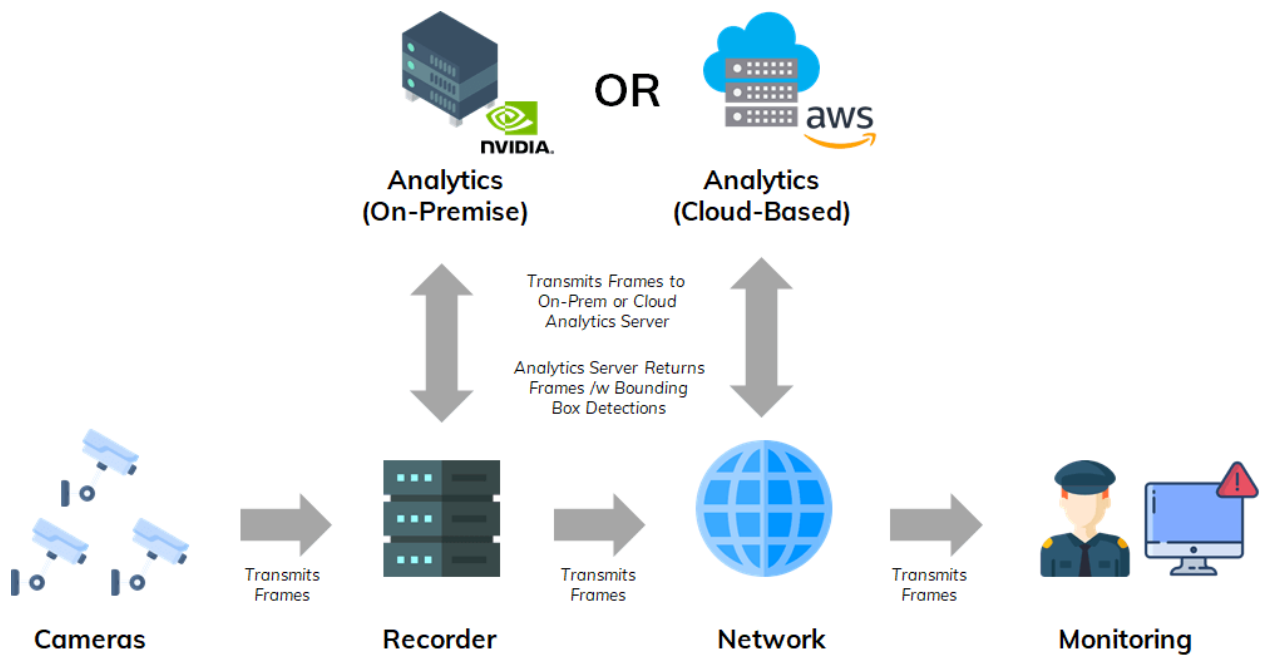


Рисунок 1.3 – Хмарна система відеоспостереження

1.4 Архітектура розподілених вузлів у системах відеоспостереження

Архітектура розподілених вузлів, передбачає обробку та збереження відеоданих на самій камері або на розподілених вузлах, які розташовані недалеко від камер. Цей підхід дозволяє виконувати обробку відеоданих на місці їхнього виникнення та зменшує навантаження на центральний сервер або хмару. Цій архітектурі притаманні такі характеристики та переваги:

- одні з головних переваг розподілених вузлів є можливість виконувати обробку відеоданих та аналіз прямо на камерах або на вузлах, що знаходяться близько від камер це дозволяє виявляти події, такі як втогрення або рух або визначення обличчя на місці розташування відеокамери;

- архітектура розподілених вузлів зменшує затримки у відеопотоку, оскільки відеодані обробляються на місці їхнього виникнення, а не передаються на центральний сервер або хмару для обробки;

- архітектура розподілених вузлів також забезпечує автономність роботи без постійного з'єднання з центральним сервером або хмарними ресурсами, що робить їх стійкими до перерв у мережевому з'єднанні;

-відеодані можуть зберігатися локально на вузлах, що забезпечує доступ до даних навіть в разі втрати зв'язку з центральним сервером;

-через те що відеокамера, може спрацьовувати від якоїсь події це може знизити витрати на мережевий трафік та зменшити загальне навантаження на мережу. Також локальна обробка може допомогти зменшити обсяг даних, що передаються через мережу, оскільки можуть передавати тільки важливі події або агреговані відомості;

-архітектура розподілених вузлів також гарантує зниження витрат на обладнання, оскільки частина обробки відеоданих виконується на самій камері [5];

На рисунку 1.4 зображено приклади застосування архітектури розподілених вузлів у системі відеоспостереження.

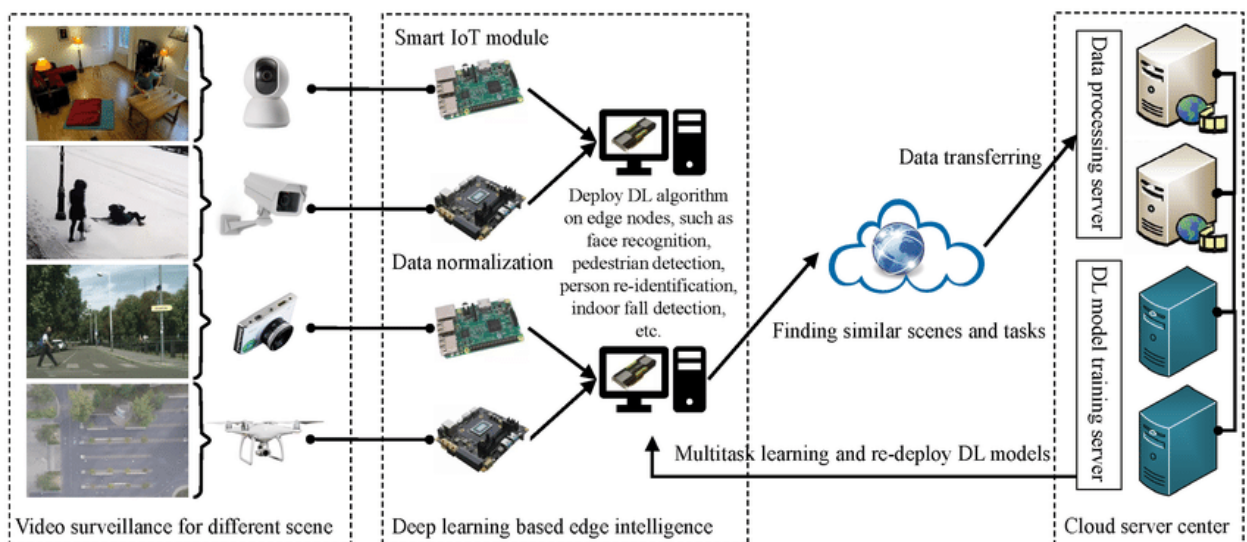


Рисунок 1.4 – Архітектура розподілених вузлів у системі відеоспостереження

Однак архітектура розподілених вузлів може бути складнішою у плануванні та управлінні, оскільки потребує розміщення вузлів біля кожної камери та налагодження їх взаємодії. Крім того вона може бути менш ефективною для систем з великою кількістю камер, якщо необхідно здійснювати обширну обробку даних на кожній з них.

1.5 Мережева архітектура в системах відеоспостереження

Мережева архітектура в системах відеоспостереження передбачає використання мережевих технологій для підключення та управління камерами та іншими складовими системами відеоспостереження. Цей підхід особливо корисний для великих систем відеоспостереження та систем розташованих на великій території. Цій архітектурі притаманні такі характеристики та переваги:

- мережева архітектура дозволяє централізовано керувати всіма камерами та іншими пристроями відеоспостереження це спрощує процес управління та моніторингу системи;

- мережеві системи можуть легко масштабуватися шляхом додавання нових камер або вузлів до мережі без значних змін у структурі системи;

- мережеві камери можуть надавати високу якість відео завдяки підтримці високорозширених форматів, таких як Full HD або 4K;

- мережева архітектура дозволяє здійснювати віддалений моніторинг та управління системою відеоспостереження з будь-якого місця, де є доступ до мережі;

- мережеві системи можуть легко інтегрувати з іншими системами безпеки та управління завдяки підтримці стандартних протоколів та API;

Для цієї архітектури важливо забезпечити безпеку мережі, оскільки відеодані передаються через мережу. Заходи безпеки можуть включати в себе шифрування, обмеження доступу та захист від несанкціонованого доступу.

Також потрібно зазначити що мережева архітектура також може стати вразливою до відмови мережі та в разі відмови зв'язку може призвести до втрати доступу до відеоданих. Тому важливо ретельно планувати резервні мережеві з'єднання та розглядати можливості автономної роботи в разі відмови мережі [6].

Загальна схема мережевої архітектури системи відеоспостереження зображено на рисунку 1.5.

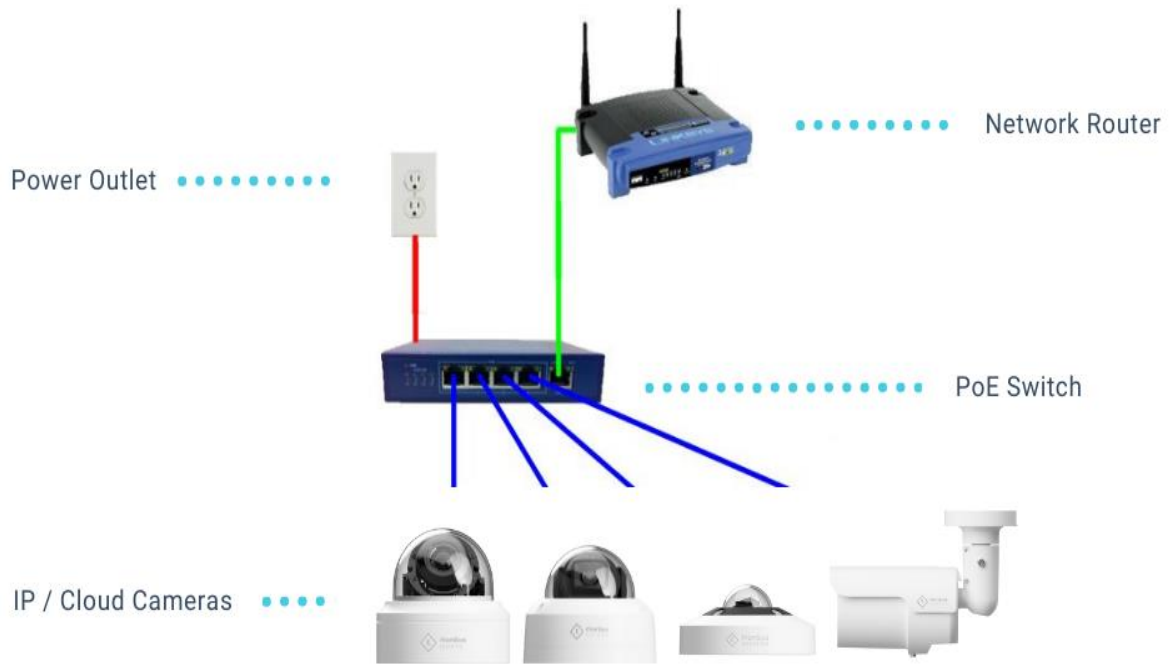


Рисунок 1.5 – Мережева архітектура системи відеоспостереження

1.6 Вибір оптимальної архітектури для системи відеоспостереження

Для реалізації поточної кваліфікаційної роботи для інформаційної системи та її майбутньої архітектури потрібно обрати архітектуру системи відеоспостереження. Кінцевий результат повинен містити рішення, яке зможе працювати без світла та інтернету через ситуацію в Україні. З цього витікає що камера повинна реагувати на рух у кадрі та включати запис на якийсь час, щоб зменшити енерговитрати. Після запису камера повинна передавати дані до загального серверу для збереження запису. Через те що система повинна працювати без інтернету, то відеодані повинні передаватись по локальній мережі до основного серверу. Який в свою чергу після увімкнення світла або відновлення мережі, повинен передавати відеодані до хмари, яка в свою чергу буде передавати дані до інформаційної системи відеоспостереження. Безумовно є варіант з інформаційною системою локального плану, але результат відеоспостереження повинен працювати через мережу. Дивлячись на цей опис, виходить так що архітектура для системи відеоспостереження буде гібридною. Ця архітектура поєднує різні підходи, наприклад,

централізовану та розподілену систему. Воона може включати в себе як центральний сервер, так і розподілені сервери для різних завдань відеоспостереження.

Для проектування інформаційної системи відеоспостереження, кваліфікаційна робота буде будуватись на декількох архітектурах. Для забезпечення доступності до відеоданих буде використовуватись хмарна архітектура. Для забезпечення довшої роботи відеокамер буде використовуватись архітектура розподілених вузлів. Це допоможе зберігати відеодані на самих відеокамерах у разі знеструмлення мережі та центрального сервера. З останнього опису витікає що до описаних архітектур додається частинка централізованої архітектури. Де центральний сервер передає дані на хмарний. Це досить загальний опис вибору архітектури системи відеоспостереження. В процесі проектування інформаційної системи загальна архітектура системи відеоспостереження може видозмінюватись.

2 ПРОЕКТУВАННЯ ЗА МЕТОДОМ АТАМ

2.1 Метод компромісного аналізу архітектури АТАМ

Метод компромісного аналізу архітектури (АТАМ) - це процес оцінки архітектурного рішення для інформаційних систем або програмних проєктів. АТАМ допомагає інженерам здійснювати баланс між різними архітектурними цілями, такими як продуктивність, масштабованість, безпека та інші, і приймати обґрунтоване рішення щодо архітектури системи.

Основні кроки методу АТАМ включають:

- метод презентується зацікавленим сторонам та описується;
- керівник проєкту характеризує бізнес-цілі, які спонукають до розробки;
- створення дерева атрибутів якості;
- аналіз архітектурних підходів на основі високопріоритетних факторів, визначених на дереві атрибутів якості, вході цього етапу отримуються архітектурні ризики та точки чутливості та точки компромісу;
- проведення мозкового штурму та отримання пріоритетних сценаріїв де на основі типових сценаріїв будуть взяті за основу;
- після останнього кроку потрібно повторно проаналізувати архітектурні підходи завдяки тестовим сценаріям можна виявити додаткові архітектурні підходи та ризики. Також точки чутливості та компроміси, які потім ретельно документуються.
- презентування результатів на основі інформації зібраної за допомогою АТАМ та представлення результату зацікавленим сторонам[7];

В загальному метод АТАМ, чудово підходить для проєктування кваліфікаційної роботи за темою інформаційної системи відеоспостереження. Наступні підрозділи цього розділу будуть стосуватись саме АТАМ методики.

На рисунку 2.1 зображена загальна схема дії методу АТАМ.

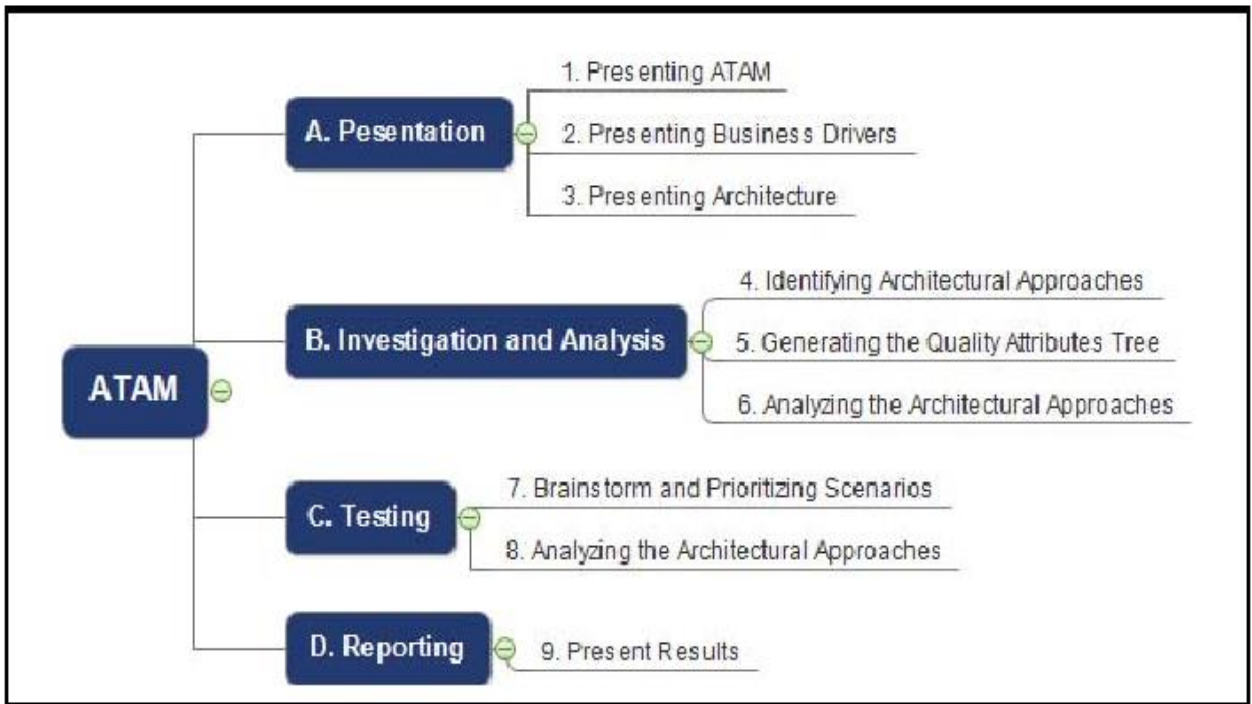


Рисунок 2.1 - Загальна схема методу АТАМ

2.2 Функціональні вимоги системи

Функціональні вимоги – це частина вимог до програмного продукту або системи, яка описує, як система повинна функціонувати, які функції повинні бути доступні користувачам або іншим системам, та як вони мають працювати.

Функціональні вимоги до системи краще розділити на інформаційну систему та систему відеоспостереження. Для інформаційної системи функціональні вимоги складаються з:

- збереження відеоданих;
- збереження метаданих коли відео було зроблено та коли зафіксовано рух;
- можливість відображення того чи іншого відео;
- пошук потрібного відео за датою;
- регістрація користувача;
- вхід користувача;

- конфігурувати відеокамеру через WiFi мережу;
- надання доступу іншим користувачам системи;
- відображення усіх пристроїв у локальній мережі;
- відображення заряду батареї відеокамер.

Функціональні вимоги для системи відеоспостереження складаються з:

- режим сну для відеокамери;
- визначення об'єкту який викликав рух у кадрі;
- збереження даних на камері;
- передача на центральний сервер відеоданих;
- передача даних на хмару для збереження;
- робота з нічною зйомкою у високій якості;
- робота від батареї.

На цих функціональних вимогах буде будуватись майбутня інформаційна система відеоспостереження. Завдяки ним можна приблизно побачити кінцевий функціонал системи.

2.3 Стейкхолдери системи їх цілі та зацікавленість у системі

Стейкхолдери - це особи, групи або організації, які мають інтерес або стейк в конкретному проекті, продукті або організації. Це люди або групи, які можуть бути безпосередньо або опосередковано залучені до цього проекту або організації, і вони мають вплив на його успіх або результат. Зацікавлені сторони можуть мати прямий або опосередкований вплив на діяльність або проекти організації. Їх підтримка часто потрібна для успіху бізнесу та проектів [8].

Стейкхолдерами системи є:

- компанія розробник системи;
- замовник системи;
- впроваджувач системи, тобто компанія, яка буде встановлювати систему відеоспостереження;

-постачальник, який забезпечує систему усіма потрібними елементами для системи відеоспостереження;

-кінцевий користувач, хто буде взаємодіяти з системою відеоспостереження .

На рисунку 2.2 зображена схема усіх наявних стейкхолдерів системи.



Рисунок 2.2 – Стейхолдери системи

2.4 Обмеження системи

Обмеження системи залежить від кількості камер, які будуть використовуватись на об'єкті для захищення. Система відеоспостереження яка проектується буде мати до 5 відеокамер спостереження, але так як система також буде будуватись на архітектурі розподілених вузлів додавання нових точок для спостереження може бути досить незручним. Тому було прийняте рішення додати елемент централізованого серверу, який буде шиною між хмарним сервером та центральним.

Також в обмеження системи входить дистанція локальної мережі через яку буде відбуватись передача даних між камерами та основним сервером. Це обмеження поєднується з наступним обмеженням, а саме ціна проектного та реалізованого рішення.

2.5 Атрибути якості системи

Атрибути якості - це вимірювані характеристики, які визначають якість програмного продукту або системи. Вони допомагають оцінювати, наскільки система відповідає вимогам та очікуванням користувачів і задовольняє стандарти та критерії ефективності та надійності.

Атрибути якості допомагають визначити сильні та слабкі сторони програмного продукту і визначити області для покращення.

Першим атрибутом якості системи є запобігання несанкціонованого доступу до системи відеоспостереження. Наприклад у системі будуть профілі, які зможуть отримувати дані стосовно картини та стану відеоспостереження. Ці профілі повинні бути захищені від несанкціонованого доступу.

Другим атрибутом якості системи є надійність. В цьому атрибуті якості мається на увазі стійкість до відмов та гарантування безперервну роботу системи відеоспостереження під різними зовнішніми чинниками. Наприклад система відеоспостереження повинна працювати в момент аварійного вимкнення струму.

Третім атрибутом якості системи є швидкість обробки зображення в різних розширення. Наприклад швидкість обробки може бути різною для VGA розширення та Full HD.

Система відеоспостереження повинна масштабуватись до зростання обсягів даних з відеокамер. Це масштабування стосується також збільшення точок де стоїть захист і кількість інформації буде збільшуватись за рахунок створення нових точок де стоять відеокамери.

Наступним атрибутом системи відеоспостереження є сумісність з іншими системами та додатками. Відеокамери повинні спокійно приєднуватись через IP до інших систем та налаштовуватись через них.

Частково тотожним атрибутом якості системи є портативність системи. Тобто майбутній додаток, який буде взаємодіяти з серверної частиною повинен портуватись під різні операційні системи такі як Android чи Windows.

Наступним атрибутом системи є вартість загального рішення для кінцевого користувача. Система повинна бути в першу чергу ефективною з економічної точки зору.

Останнім атрибутом якості системи є споживання ресурсів. Як було описано раніше система повинна масштабуватись під збільшення відеоданих які приходять з відеокамер, але також важливо забезпечити ефективність використання внутрішніх ресурсів, щоб зменшити межу масштабування та знизити вартість кінцевої системи.

На рисунку 2.3 зображений перелік у якому скорочено описані усі атрибути якості системи.

Атрибути якості	Сценарій
Безпека	Запобігання несанкціонованому доступу
Надійність	Стійкість до відмов та гарантування безперервну роботу системи відеоспостереження
Ефективність	Швидкість обробки зображення в тому чи іншому розширенні
Масштабованість	Змога масштабування до зростання обсягів даних
Сумісність	Можливість приєднювати систему відеоспостереження до інших додатків або систем
Портативність	Робота застосунку який підтримує: Android, Windows.
Вартість	Система відеоспостереження повинна бути ефективною з економічної точки зору
Споживання ресурсів	Ефективність використання ресурсів такі як енергія та пам'ять де буде зберігатись відеодані

Рисунок 2.3 – Атрибути якості системи

Цим остаточним переліком аналіз майбутньої системи за допомогою методу АТАМ закінчено та можна переходити до документування за методом 4+1.

3 ОПИС ПРОЕКТНИХ РІШЕНЬ ЗА МЕТОДОМ 4+1

3.1 Загальні відомості стосовно методу 4+1

Метод 4+1 (англ. 4+1 architectural view model) є архітектурним підходом до розробки програмного забезпечення, який дозволяє організувати та представити архітектурну інформацію про систему. Цей метод дозволяє розглядати систему з різних точок зору, використовуючи чотири архітектурні види (4) та один сценарій (1). Він розроблений для полегшення розуміння та спілкування між членами розробницького команду та іншими зацікавленими сторонами.

Основні положення методу 4+1:

- логічний вид відображує системи на рівні класів або об'єктів. Вказує, як система розбивається на компоненти, як вони взаємодіють та як вони відповідають за реалізацію бізнес-логіки;

- процесний вид описує взаємодію різних процесів у системі це важливо для розуміння та оцінки розподіленої архітектури системи;

- вид розробки описує компоненти системи з представленням щодо взаємодії між компонентами;

- фізичний вид опису як компоненти системи розташовані на апаратному обладнанні;

- сценарії описують, як користувачі взаємодіють із системою через різні сценарії використання [9].

На рисунку 3.1 зображено усі положення методу 4+1 через схематичне представлення самого методу.

На базі цього методу буде вестись подальше проектне документування системи відеоспостереження починаючи з логічного виду.

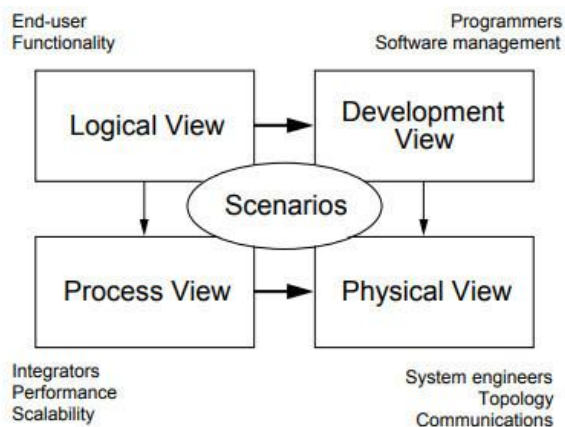


Рисунок 3.1 – Представлення методу 4+1

3.2 Логічний вид

Логічний вид системи відеоспостереження, частково був описаний у першому розділі.

Для збереження даних та їх доступності буде використовуватись хмара. Для забезпечення енергоефективності та доступності відеокамер буде використовуватись архітектура розподілених вузлів. Для збільшення стійкості системи буде також використовуватись центральний сервер, який буде взаємодіяти з відеокамерами через LAN мережу та передавати на хмару усі відеодані з відеокамер. На цьому концепті логічних вид був побудований та зображений на рисунку 3.2.

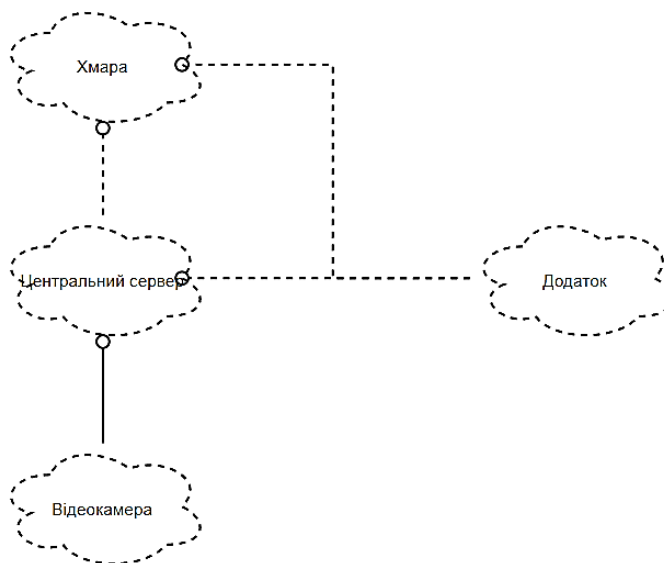


Рисунок 3.2 – Логічний вид системи відеоспостереження

Схема будувалась за допомогою логічного виду який представлений у документі 4+1. На рисунку 3.3 зображено типи з'єднувальних елементів у логічному виді.

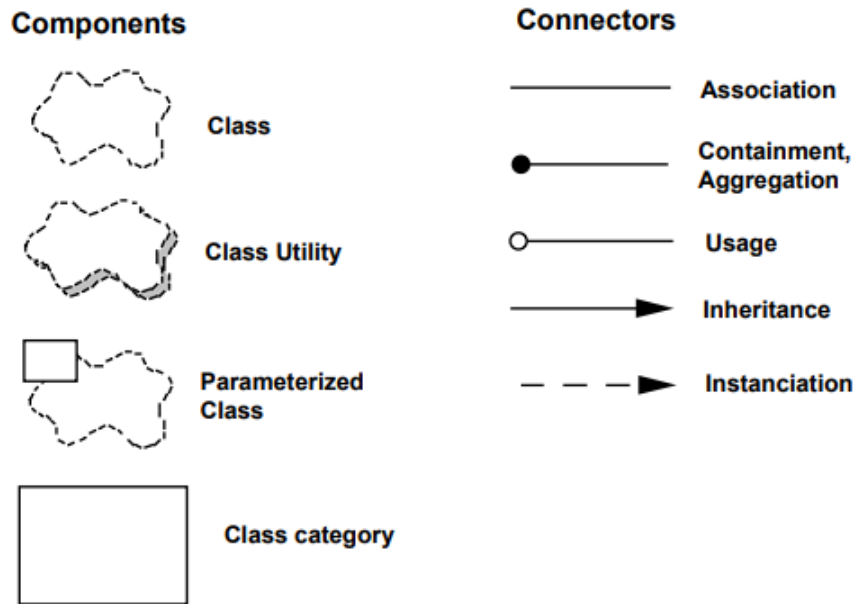


Рисунок 3.3 – Компоненти для побудови логічного виду системи

3.3 Вид розробки

Вид розробки буде представлений у вигляді діаграми компонентів додатку системи відеоспостереження та діаграми компонентів самої відеокамери.

Діаграма компонентів відеокамери зображено на рисунку 3.4

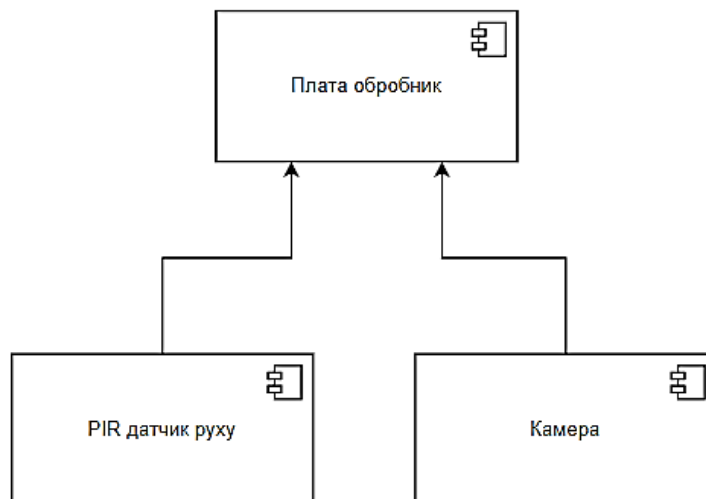


Рисунок 3.4 – Діаграма компонентів відеокамери

Відеокамера буде будуватись на платі обробника або мікроконтролера, який взмозі обробляти відеодані, які надходять з другого компоненту, а саме самої камери. Також для забезпечення енергоефективності кінцевого рішення було прийнято рішення додати до системи PIR датчик, який буде сигналізувати рух у об'єктиві та починати запис того що відбувається. Якщо руху не було, то відеокамера знижує своє споживання, та переходить у режим сну.

Діаграма компонентів додатку для системи відеоспостереження зображено на рисунку 3.5

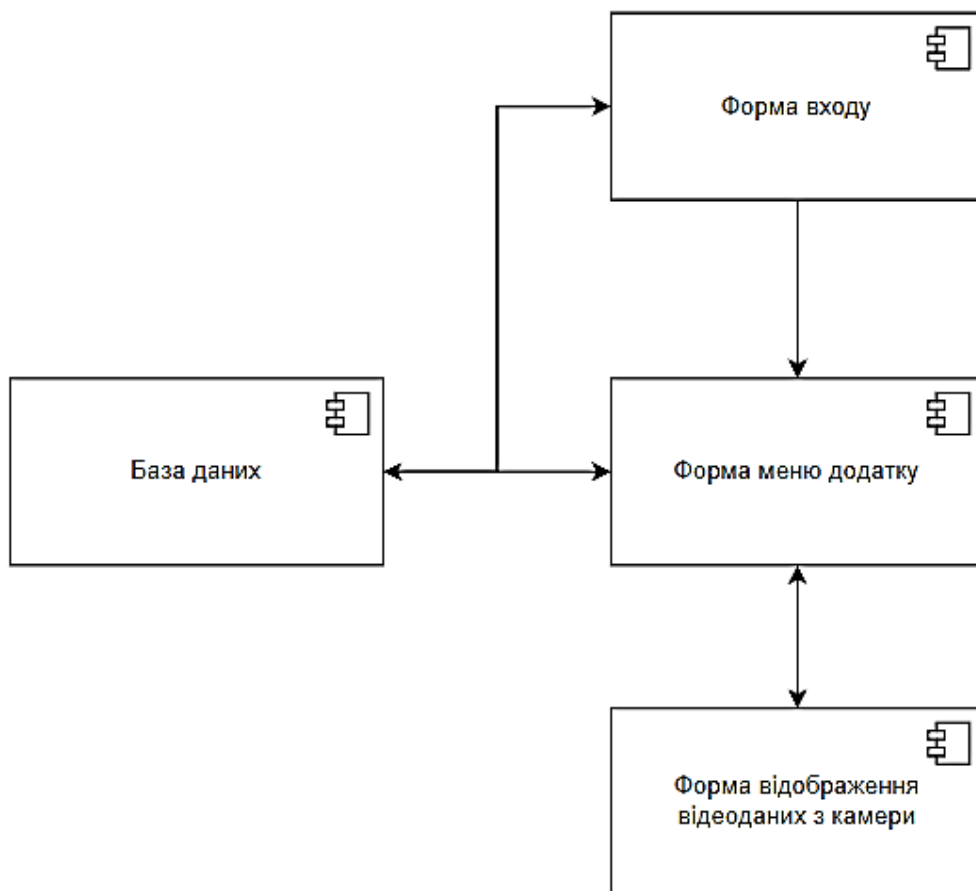


Рисунок 3.5 – Діаграма компонентів додатку

Діаграма компонентів додатку для системи відеоспостереження включає у себе три основних форми, а саме:

-форма входу де користувач повинен ввести свої дані, щоб зайти у систему відеоспостереження;

- форма меню додатку де користувач буде зображені усі дані системи;
- форма відображення відеоданих з камери це форма прямої трансляції з камери.

Дві форми будуть тісно взаємодіяти з базою даних, яка буде перебувати на хмарі та зберігати важливі дані системи відеоспостереження.

3.4 Процесний вид

Процесний вид буде складатись з усіх процесів, які входять до компонентів системи. На рисунку 3.6 зображена схема процесів системи відеоспостереження.

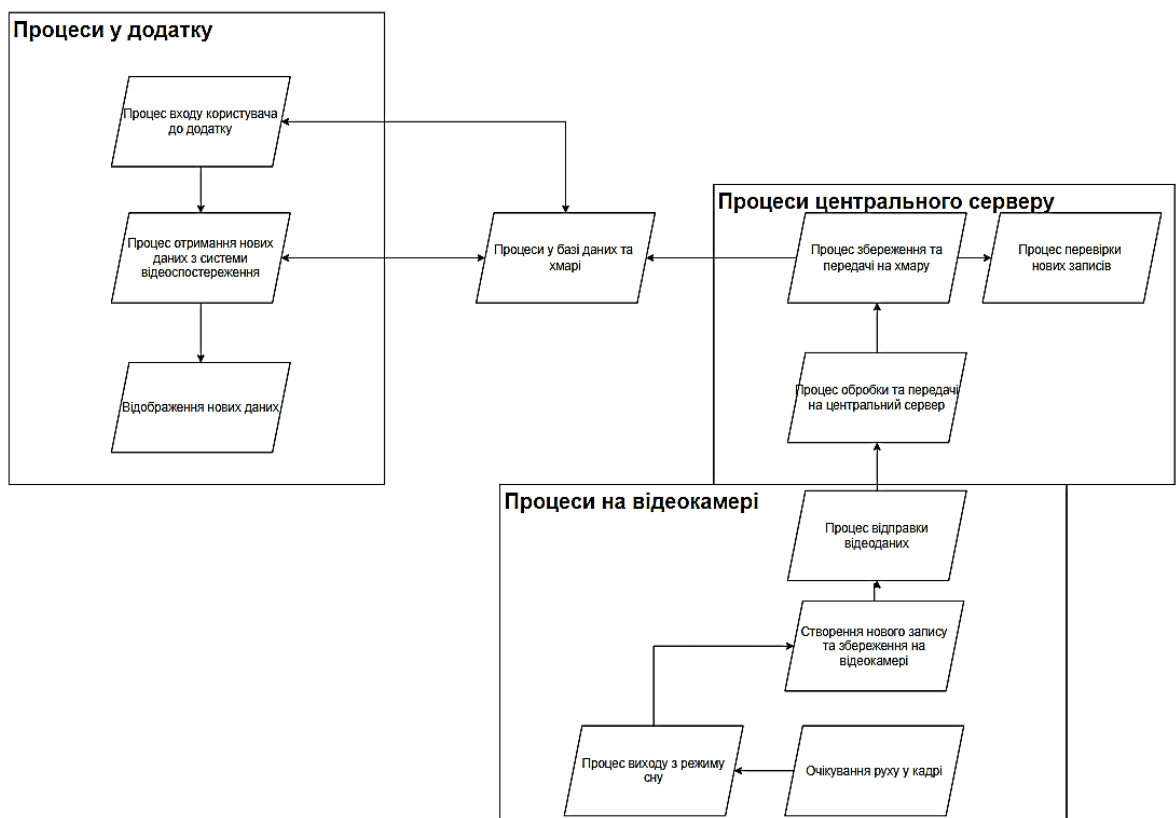


Рисунок 3.6 – Процеси системи відеоспостереження

На схемі процесів системи відеоспостереження є розділ на три основних компоненти, які поєднуються процесами у базі даних та хмарі. Основними процесами додатку є процес входу користувача та отримання та

відображення нових даних. Основні процеси центрального процесору є збереження та передачі даних на хмару. Також серверна частина повинна перевіряти наявність нових записів у відеокамер і у разі якщо щось є зберігати. Процеси відеокамери складаються з очікування руху у кадрі та процес виходу з режиму сну. Після виходу зі сну камера створює новий запис та зберігає його локально на самій відеокамері. Після збереження відеокамера передає запис до центрального серверу.

3.5 Фізичний вид

Фізична вид буде складати з вибору компонентів відеокамери та схематичне зображення кінцевого пристрою, який буде використовуватись у реалізації прототипу на базі проєктних рішень..

Першим та головним компонентом системи є сама камера. Ця камера повинна забезпечувати якісний запис локації та знімати в будь-який час включаючи нічний. За такими задачами було обрано камеру нічного бачення з інфрачервоним підсвічуванням RPI Camera Waveshare.

В основі цієї камери лежить сенсор OV5647 5МП. Роздільна здатність на 1080p що повністю забезпечує гідну якість [10]. Також камера має змогу фокусуватись на 3.6мм з різним налаштуванням. RPI Camera Waveshare зображена на рисунку 3.7



Рисунок 3.7 – RPI Camera Waveshare

Ця камера чудово підходить під Raspberry Pi будь-якої моделі, тому було прийняте рішення шукати потрібний мікроконтролер серед цієї лінійки. Є багато моделей Raspberry та всі вони є досить дорогими порівнюючи їх з іншими мікроконтролерами, так як Raspberry в основі своїх рахується більше ні як мікроконтролер, а як одноплатний комп'ютер. Під клас малих мікроконтролерів можна віднести Raspberry Pi Pico [11] та Raspberry Pi Zero. Raspberry Pi Pico дешевий варіант Raspberry Pi Zero, але його проблема полягає у тому, що немає підтримки відеокамер. Тобто якщо будувати рішення на цьому мікроконтролері прийдеться використовувати перехідники на саму камеру нічого бачення, що тим самим збільшить загальний об'єм рішення та можливо ціну за рахунок збільшення часу [12]. Тому кращим варіантом є Raspberry Pi Zero W. В якій одразу вбудований WiFi модуль та має двоядерний процесор на базі архітектури Arm.

Архітектура ARM [13] є сімейством архітектур для проектування мікропроцесорів та мікроконтролерів. ARM став популярним завдяки своїй простоті, ефективності та низькому енергоспоживанню. Технічна специфікація Raspberry Pi Zero W зображено на рисунку 3.8.

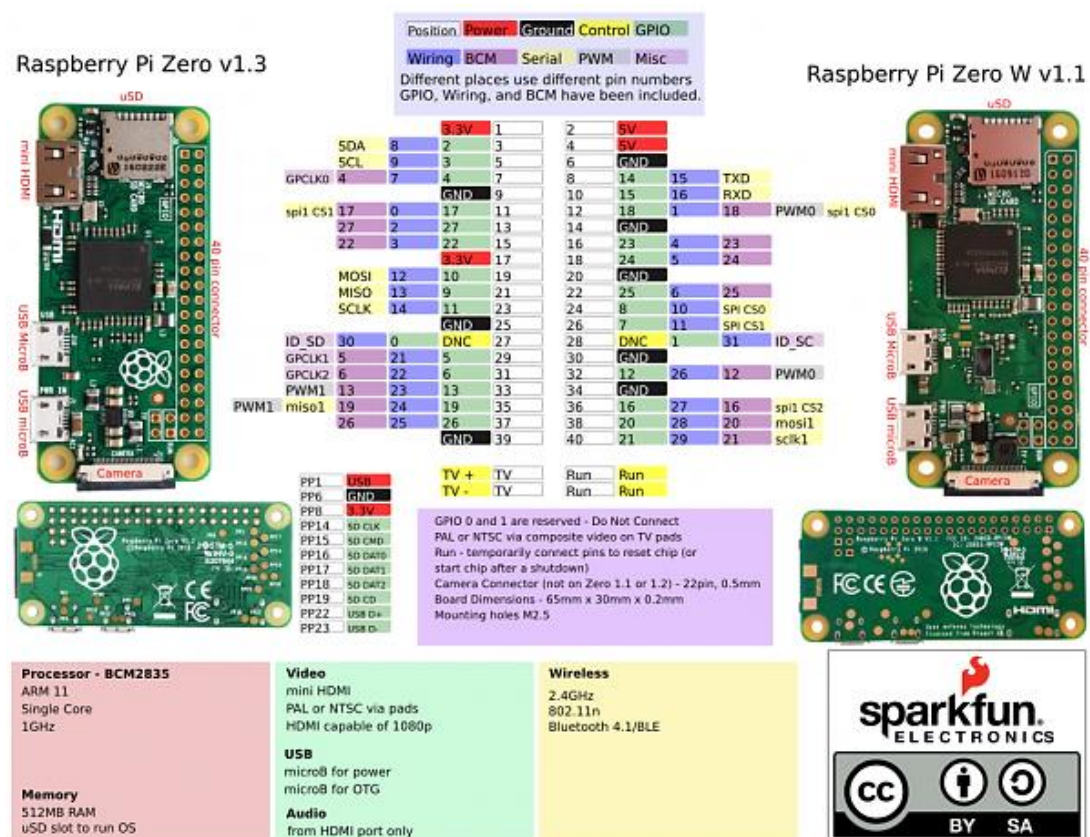


Рисунок 3.8 – Технічна специфікація Raspberry Pi Zero W

Мікроконтролер має такі характеристики:

- процесор Broadcom Arm11 з тактовою частотою 1ГГц;
- оперативна пам'ять 512Мб;
- підтримка USB, OTG, HDMI, Bluetooth, WiFi;
- має 40 портів для підключення різноманітної периферії.

Цього мікроконтролера буде більш ніж достатньо для обробки та передачі відеоданих на серверну частину.

Для забезпечення ефективного режиму відеокамери та забезпечення зменшення об'ємів даних було прийнято рішення з встановленням на камеру PIR датчику, який буде відслідковувати рух у області запису та включати запис лише в момент, коли в області запису був зафіксований рух.

PIR (Passive Infrared Sensor) — це електронний пристрій, який виявляє інфрачервоне випромінювання, що випромінюється об'єктами, що знаходяться в його полі зору [14].

Було обрано інфрачервоний датчик руху SR501. Він працює від напруги живлення 5v та його ефективний кут 120 градусів, чого достатньо для визначення руху у камері [15]. Зображення PIR датчику на рисунку 3.9.



Рисунок 3.9 – PIR-датчик

На цьому елементі усі компоненти було обрано, тому можна переходити до кінцевої схеми пристрою. На рисунку 3.10 зображена загальна схема фізичного виду

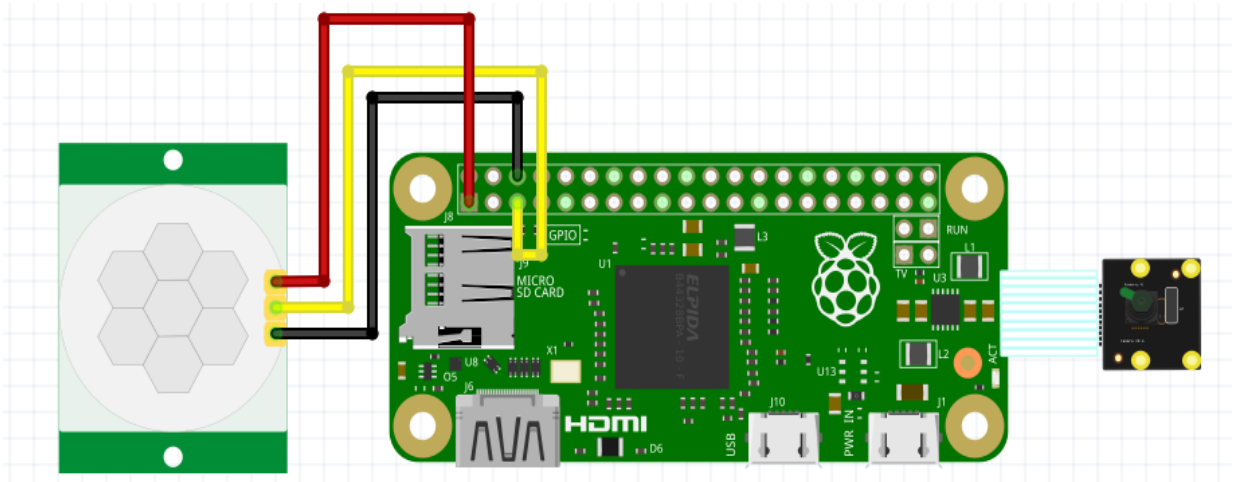


Рисунок 3.10 – Загальна схема фізичного виду

На цьому кроці проєктна частина закінчена, можна переходити до реалізації на базі проєктних рішень.

3.6 Сценарії використання

Сценарії для користувача починається з запуску додатку для керування системи відеоспостереження. Далі користувач повинен обрати, реєстрацію нового користувача чи вхід в існуючий акаунт. При реєстрації нового користувача відбуваються операції в базі даних. Після входу в акаунт користувач повинен обрати вікно де він буде надалі працювати. Першим варіантом є перегляд існуючих відеофайлів з камер та вибір потрібного відео за датою чи іншою характеристикою. Після вибору медіаплеєр автоматично завантажує відео та користувач його може дивитись. Другим варіантом це вхід у вікно режиму трансляції камери. Де користувач повинен додати камеру та вписати IP-камери для перегляду стану. Після цього користувач буде отримувати стабільний потік кадрів на визначену позицію. На рисунку

4.13 зображена діаграма варіантів використання у додатку системи відеоспостереження.

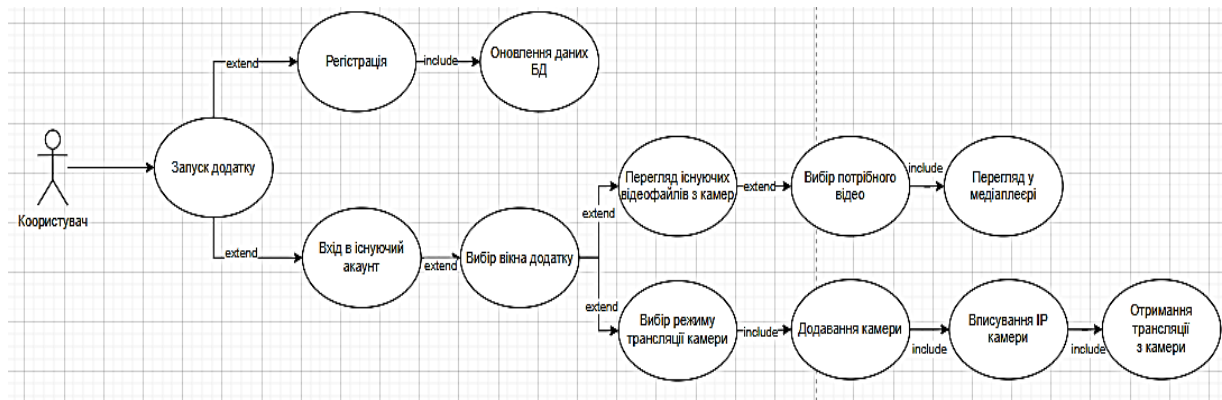


Рисунок 3.11 – Діаграма варіантів використання для користувача

На цьому проектування системи відеоспостереження закінчено, можна переходити до реалізації системи відеоспостереження з розробкою програмного забезпечення.

4 РЕАЛІЗАЦІЯ ПРОТОТИПУ НА БАЗІ ПРОЕКТНИХ РІШЕНЬ

4.1 Розробка та реалізація бази даних

Розробка концептуальної моделі бази даних — це етап розробки, на якому визначається структура та взаємозв'язки між сутностями, що будуть використовуватися для організації та зберігання даних [16].

Для розробки бази даних потрібно на базі функціональних вимог визначити сутності бази даних та їх внутрішні атрибути з типами даних.

В реляційній моделі баз даних існують різні типи зв'язків між сутностями. Зв'язки визначають, як сутності взаємодіють та пов'язані одна з одною в контексті баз даних. Є такі типи зв'язків між сутностями:

- один до одного де кожен запис однієї сутності пов'язаний з одним записом в іншій сутності.

- один до багатьох де кожен запис однієї сутності може мати багато відповідних записів в іншій сутності, але запис в іншій сутності пов'язаний лише з одним записом;

- багато до одного де кожен запис в одній сутності пов'язаний лише з одним записом в іншій сутності, але запис в одній сутності може мати багато відповідних записів в іншій сутності;

- багато до багатьох є не бажаним зв'язком у реляційній моделі бази даних, але все ж таки він існує де кожен запис в одній сутності може мати багато відповідних записів в іншій сутності та навпаки.

Ці типи зв'язків визначають структуру бази даних та допомагають визначити, як дані пов'язані одне з одним. На рисунку 4.1 зображена концептуальна модель бази даних системи відеоспостереження.

На концептуальній моделі бази даних присутні такі сутності як:

- сутність адміна для авторизації охоронців чи людей, які будуть спостерігати за роботою системи;

-сутність камери відеоспостереження, яка включає у себе IP камери та унікальний ідентифікатор камери;

-сутність відео складається з усіх характеристик відео включно з айді файлом, який буде зберігатись на хмарі;

-сутність користувача включає доступ до камери та відео, які були записані з камери користувача;

-асоціативна сутність групи камер, коли у користувача більше ніж одна камера.

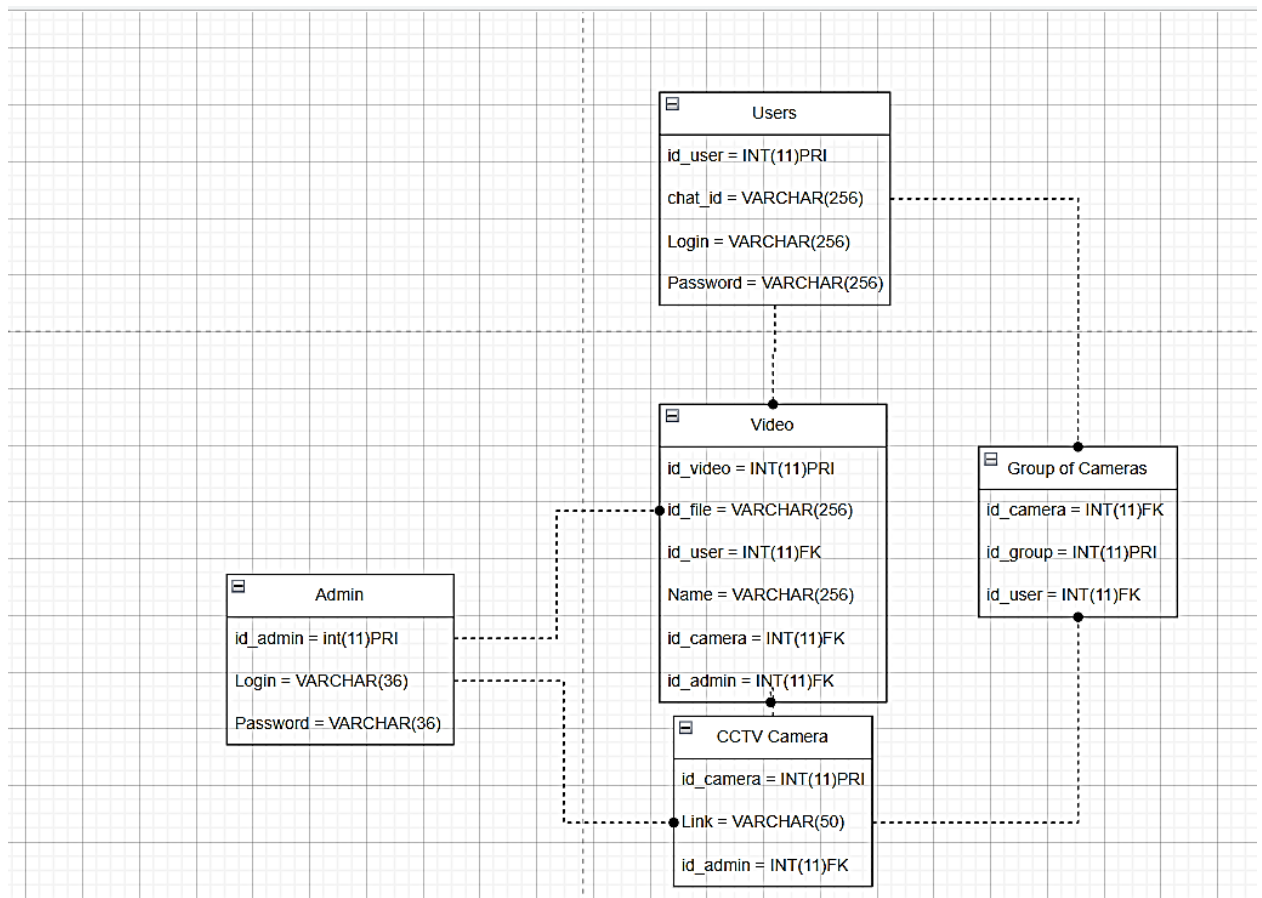


Рисунок 4.1 – Концептуальна модель бази даних

Зв'язки між сутностями бази даних складаються:

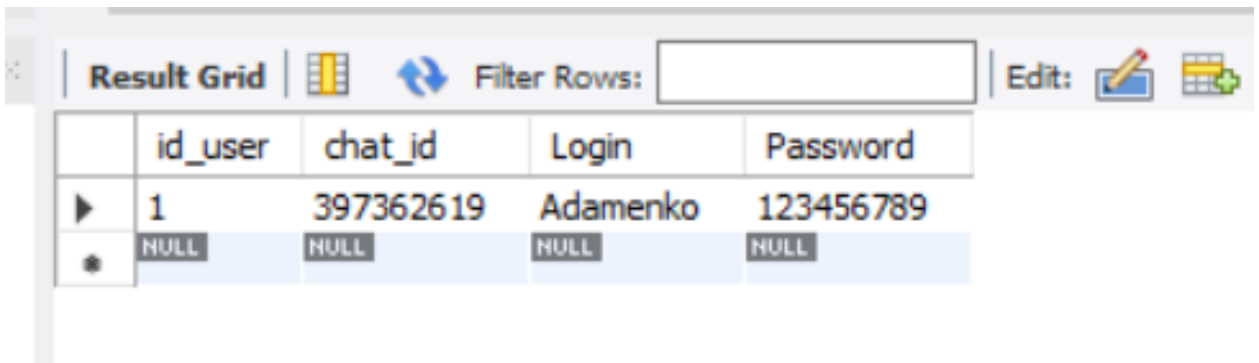
- адмін до камери – один до багатьох;
- адмін до відео – один до багатьох;
- камера до відео – один до багатьох;
- камера до групи камер – одна до багатьох;
- користувач до відео – один до багатьох;

-користувач до групи камер – один до багатьох;

Реалізація бази даних буде вестись за допомогою MySQL. MySQL - це система управління базами даних (СУБД), яка є однією з найпопулярніших відкритих реляційних баз даних. Вона розроблена компанією Oracle Corporation і є частиною великої сім'ї продуктів, пов'язаних з управлінням базами даних [17].

MySQL використовує стандартний SQL для взаємодії з базою даних. Це дозволяє виконувати операції вставки, вибірки, оновлення та видалення даних, а також виконувати різноманітні операції з базовими структурами даних.

На базі концептуальної моделі було створено таблиці та їх поля. Також було записано користувача для взаємодії з програмним забезпеченням, який буде розроблятися пізніше. Результат запису зображений на рисунку 4.2.



	id_user	chat_id	Login	Password
▶	1	397362619	Adamenko	123456789
•	NULL	NULL	NULL	NULL

Рисунок 4.2 – Результат реєстрації користувача.

На базі цієї бази даних буде вестись розробка додатку для відеоспостереження.

4.2 Реалізація програмного забезпечення відеокамери

Реалізація програмного забезпечення відеокамери включає у себе взаємодію з кінцевим користувачем через Telegram API.

Telegram API - це інтерфейс програмування застосунків (API), який надається Telegram для взаємодії з їхньою платформою. Це набір правил та

протоколів, які дозволяють розробникам створювати власні програми та сервіси, які можуть обмінюватися даними з Telegram [18].

За допомогою Telegram API, якщо користувач відеокамери має лише одну відеокамеру, то йому буде досить зручно користуватись Telegram додатком для моніторингу стану об'єкту захисту.

Для реалізації програмного забезпечення відеокамери на Raspberry Pi Zero W буде використовуватись мова програмування Python.

Python - це високорівнева мова програмування загального використання та є однією з найпопулярніших та широко використовуваних мов програмування. Ця мова програмування підходить через велику кількість різноманітних бібліотек, які підтримуються та розробляються. Для реалізації буде використовуватись декілька бібліотек, а саме:

- psutil(процеси та системні утиліти) міжплатформна бібліотека для отримання інформації про запущені процеси та використання системи різних ресурсів з моніторингом стану ЦП, пам'яті, мережі та інших елементів системи [20];

- gpiozero простий інтерфейс для взаємодії з GPIO пристроями у Raspberry Pi [21];

- pytz ця бібліотека дозволяє роботи точні обчислення часових поясів між платформами [22];

- picamera забезпечує чистий інтерфейс для модуля камери Raspberry Pi [23];

- python telegram bot забезпечує зв'язок між камерою та Telegram API та надає функціонал для взаємодії з користувачем [24].

На цих бібліотеках реалізовано програмне забезпечення відеокамери, яке можна побачити у додатку А.

Для взаємодії з камерою, користувач повинен створити телеграм бот та приєднати його до камери відеоспостереження. Після чого встановити на бажану позицію. На рисунку 4.3 зображений результат реагування на команди, які надсилаються відеокамері.

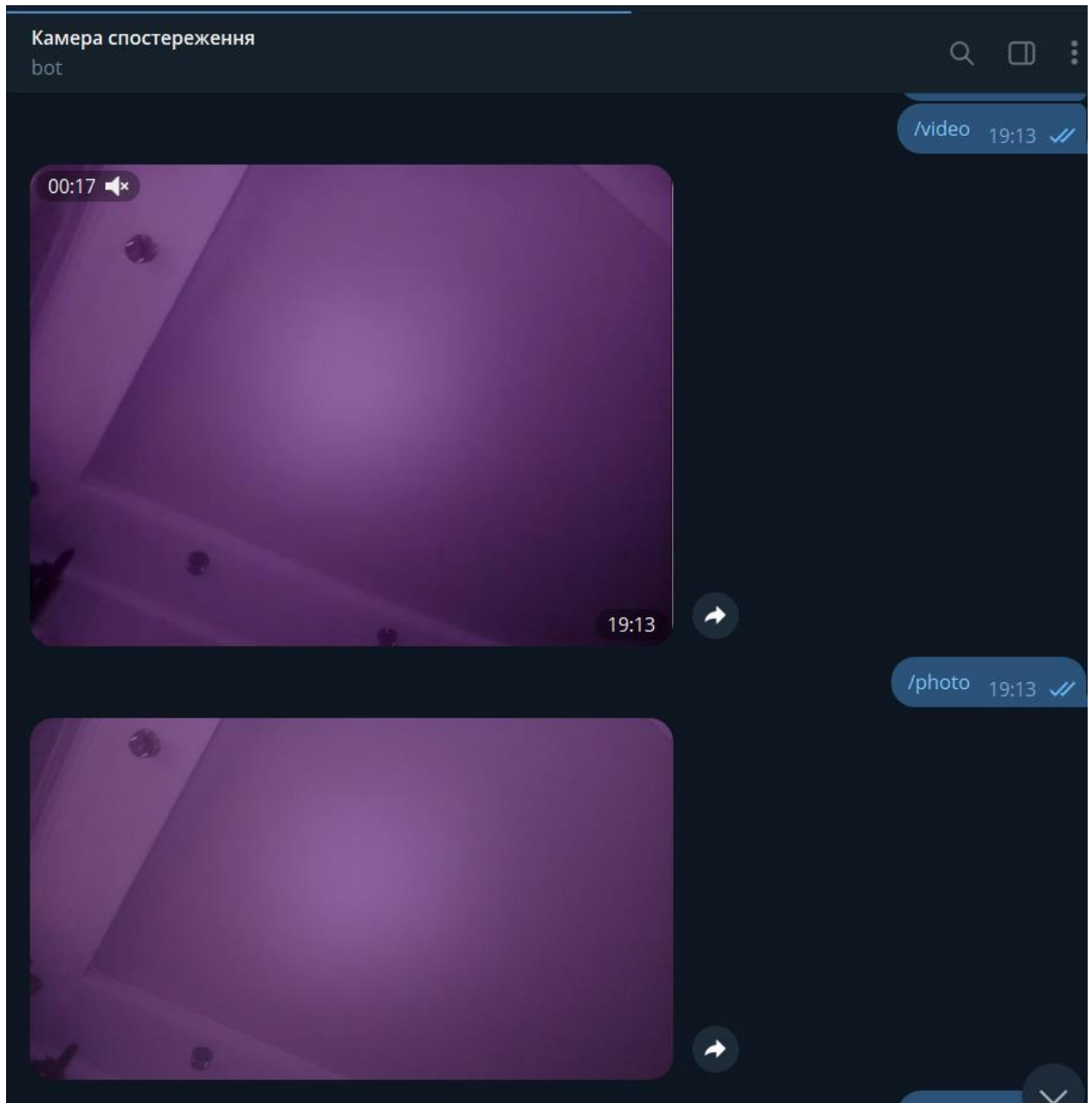


Рисунок 4.3 – Виконання команд камерою

Коли користувач надсилає команду створення відео. Камера виходить з сплячого режиму та починає запис 30 секунд. Після чого надсилає відео до користувача. Також для більш швидкого отримання стану об'єкту захисту, користувач може обрати команду фотографії, де результат надходить майже одразу. Також камера реагує на рух за рахунок датчику руху SR501 та надсилає відео після зафіксування руху. Також надсилається час запису відео за часовим поясом Києву. На рисунку 4.4 зображено результат визначення руху та надсилання відео.

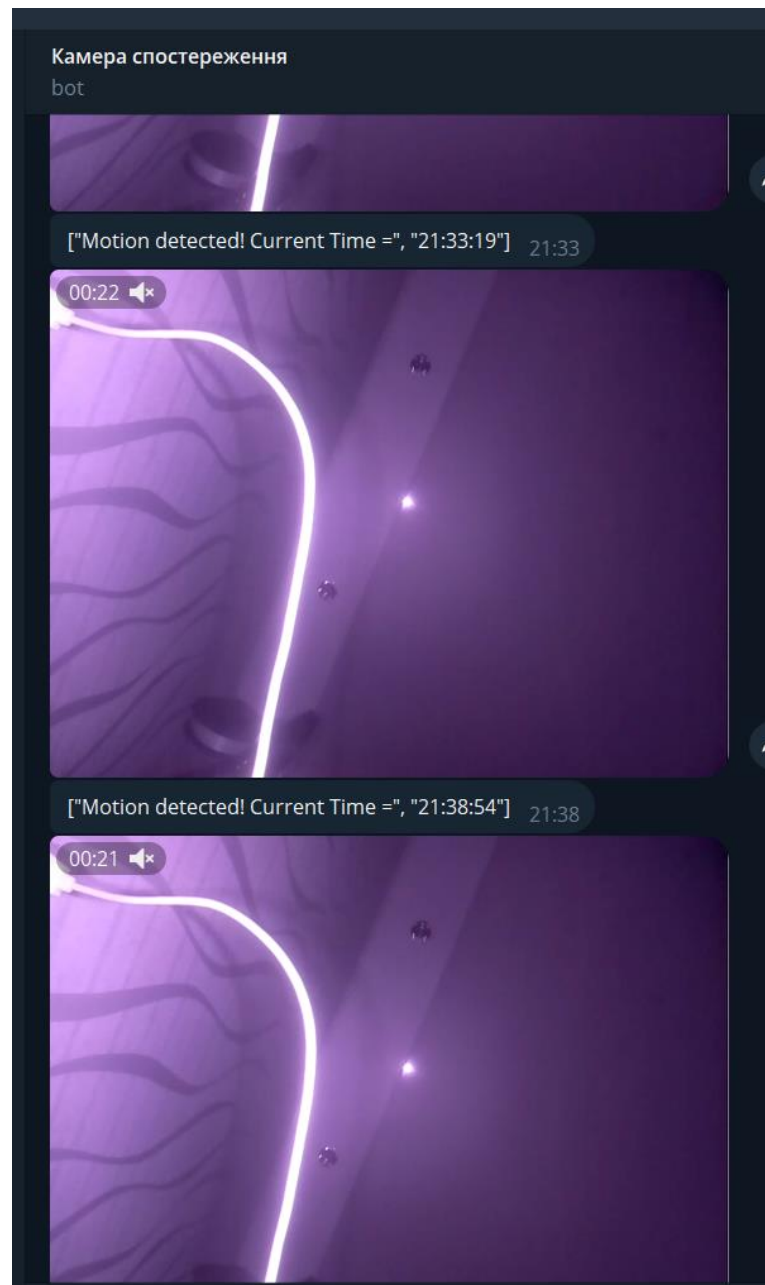


Рисунок 4.4 – Реагування камери на рух

Також відеокамера може працювати у режимі трансляції через IP. Для цього буде створено додаток для додавання групи камер для захисту об'єкту відеоспостереження. Реалізація програмного забезпечення буде описано у наступному підрозділі. На цьому моменті розробка для одного користувача рахується закінченою.

Кінцевий вигляд усієї відеокамери з датчиком SR501 зображено на рисунку 4.5.



Рисунок 4.5 – Кінцевий вигляд відеокамери на базі Raspberry Pi Zero W

4.3 Реалізація програмного забезпечення системи відеоспостереження

Реалізація програмного забезпечення буде відбуватись за допомогою двох ресурсів. Першим ресурсом є мова програмування C# це мова програмування, розроблена компанією Microsoft. Вона була представлена в

2000 році і стала частиною ініціативи Microsoft [26] для розробки програмного забезпечення для платформи .NET. .NET є другим ресурсом розробки, так як він надає широкий спектр можливостей для реалізації додатку. .NET є програмною платформою для створення Windows додатків та інших видів додатків [27].

Першим кроком реалізації програмного забезпечення системи відеоспостереження є створення форми де буде описуватись вхід користувача до системи та реєстрація користувача у системі. У розробці буде також використовуватись бібліотека для покращеного інтерфейсу Windows form Krypton. Ця бібліотека має більшу частину усіх звичайних елементів Windows form, але має змогу глибшої зміни в плані дизайну та вигляду при наведенні на кнопку чи на якийсь елемент [28]. На рисунку 4.6 зображено результат реалізації вхідної та реєстраційної форми.

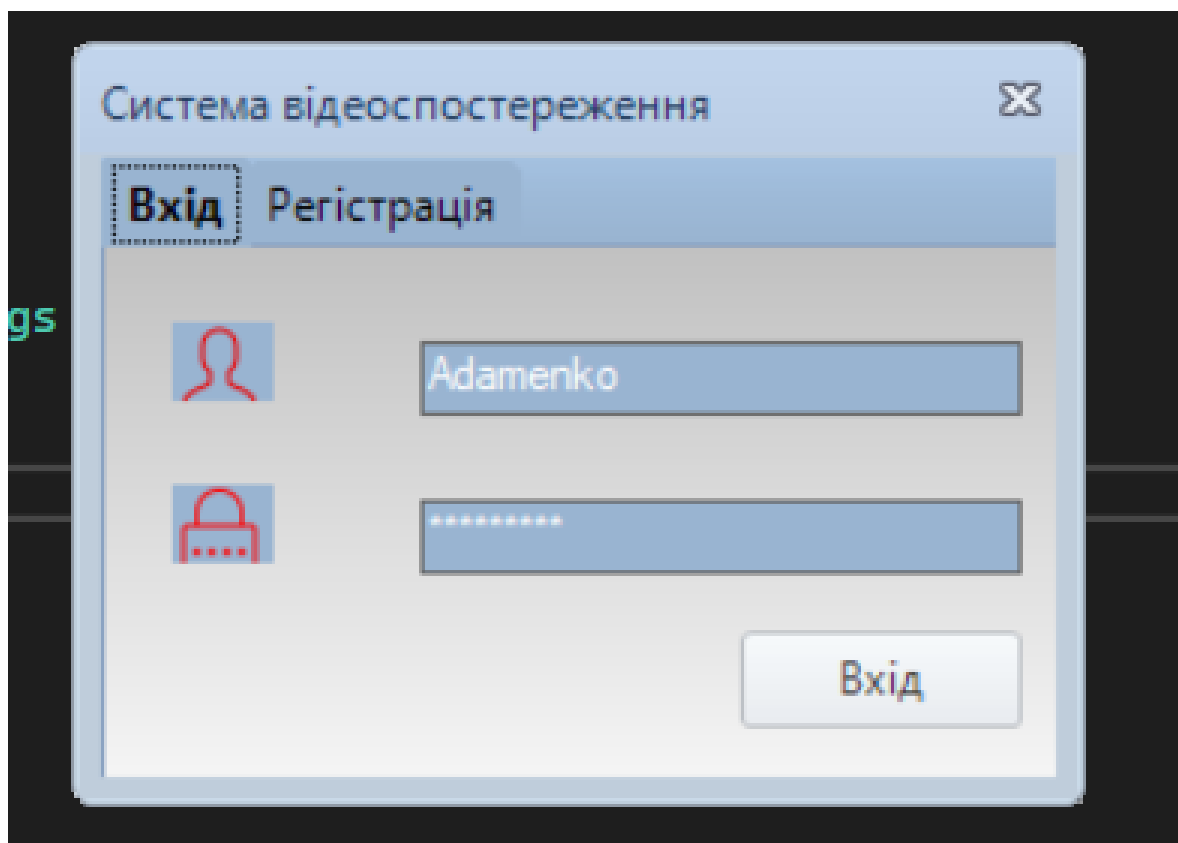


Рисунок 4.6 – Форма реєстрації та входу у системі відеоспостереження

Можна побачити, що поле при введенні паролю замінює символи на зірочки. Після введення даних користувач повинен натиснути кнопку входу.

Пароль та логін порівнюються з даними, які знаходяться у базі даних. Якщо пароль або логін був введений не вірно, то система не пускає користувача до системи. Відхилення користувача до входу у систему буде зображено на рисунку 4.7.

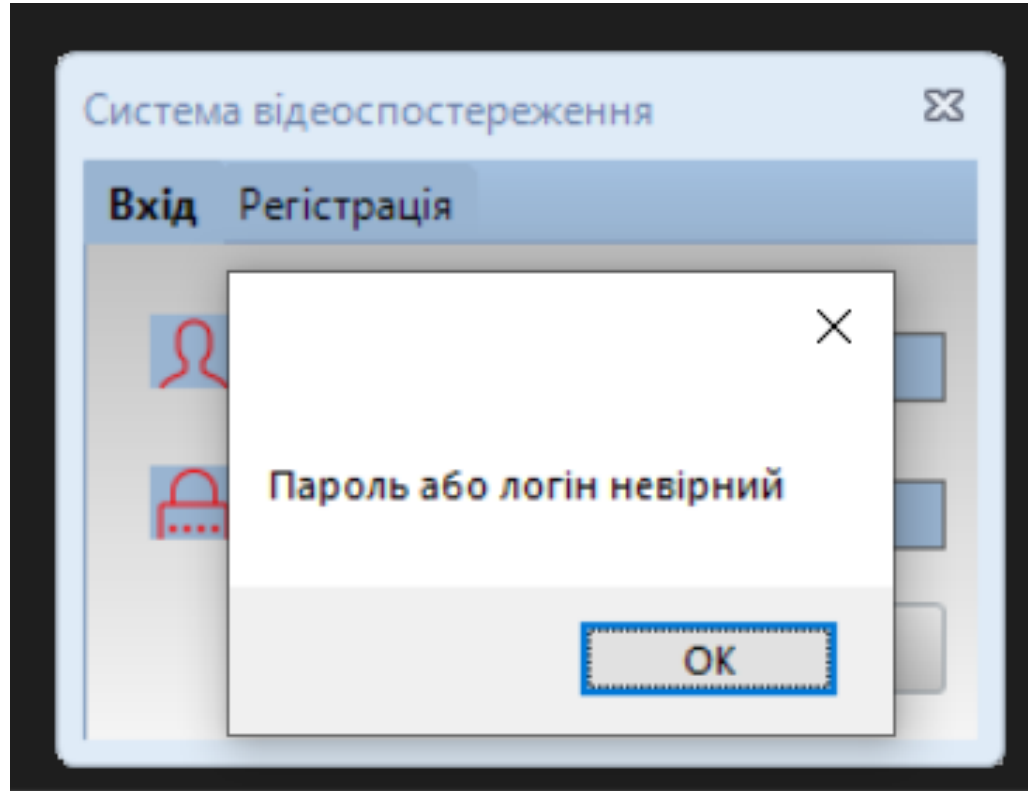


Рисунок 4.7 – Відхилення користувача до входу у систему

Якщо пароль та логін вірний форма входу замінюється формою головного меню, яка розділяється на дві частини, а саме:

- перша частина дає можливість приєднатись до камери на пряму через IP. Максимальна кількість підключених одночасно камер становиться 30 штук;

- друга частина складається з завантажування усіх відеофайлів, які були зроблені до системи, та можливість їхнього перегляду через програвач, який було імпортовано.

На рисунку 4.8 буде зображена система приєднання до камери через IP та зображено результат підключення за рахунок виводу кожного кадру у зону для відео.

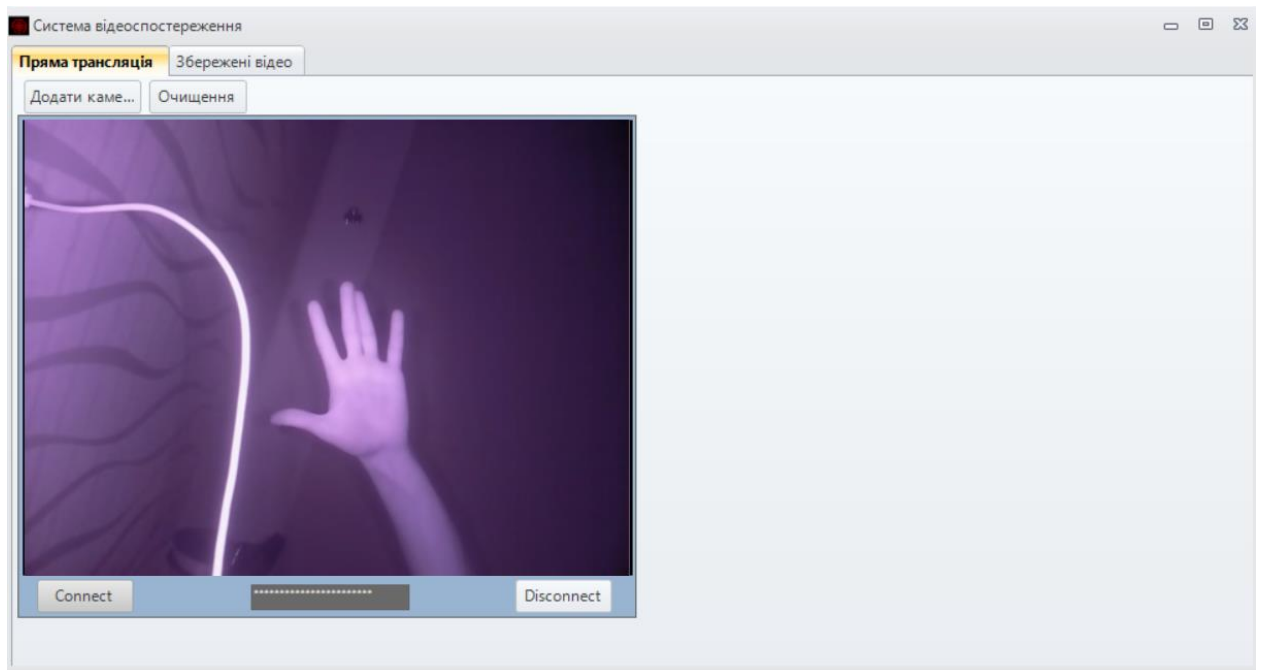


Рисунок 4.8 – Підключення до прямої трансляції відеокамери

Адмін має змогу прибрати усі камери та провести відключення від камери. Також слід зазначити, що якщо камера працює у режимі трансляції усі її інші функції недоступні.

Процес вибору відео є досить простим на момент розробки працює вибір відео та його запуск у медіа програвачі, але у майбутньому цей розділ потребує доробки. По перше не вистачає пошуку за камерою, якщо камер багато. Також потрібен пошук за часом запису та днем запису або місяцем. На даний момент тека з відео заповнюється з відео які були збережені у базі даних та на хмарному сервері. Для зменшення кількості коштів було використано Telegram. При відправці будь-якого відео у чат з користувачем надіслане відео отримує файл ID у чаті з користувачем. При вході до програмного забезпечення усі відео надходять з порівнянням чату ID та користувача та обираються відео, які відносяться до користувача. На рисунку 4.9 зображено результат реалізації відкриття різних відеофайлів, які відносяться до камери користувача.

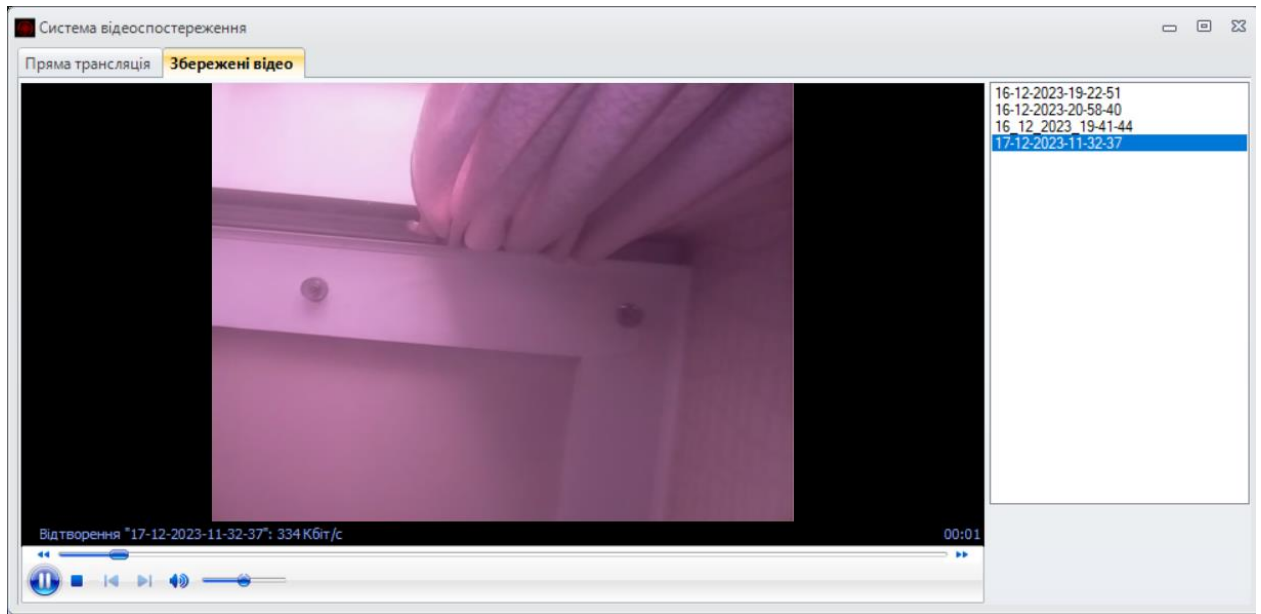


Рисунок 4.9 – Збережені відео відеокамери

На даному етапі можна рахувати систему реалізованою та є фундаментом до майбутніх покращень та інтеграцій у різні об'єкти захисту.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було спроектовано інформаційну систему відеоспостереження та реалізовано програмне забезпечення для різних видів використання відеокамери.

Система спроектована та реалізована таким чином, що будь-який користувач може застосовувати її у різних додатках. Якщо користувачу достатньо однієї камери він має можливість приєднати її до телеграм боту та отримувати результат звідти. Якщо користувач має задачу з охорони якогось великого об'єкту де кількість камер більша, він може використати розроблене програмне забезпечення та додавати групу камер.

Система самостійно може реагувати на рух у полі зору датчика руху та камери та записувати події з надсиланням та збереженням відеоданих.

В результаті розробленої кваліфікаційної роботи був проведений:

- огляд аналогічних проектних рішень з описом різних архітектур, які використовуються у створенні відеоспостережних систем;

- проектування системи та визначення вимог за методологією АТАМ та отримання розуміння загальної структури майбутнього рішення;

- проектування більш деталізовано з використанням різностороннього методу 4+1;

- реалізація прототипу програмного забезпечення для камери та для додатку, через який відбувається керування та інші функціональні вимоги до системи.

Підсумовуючи вище сказане кінцевий проєкт має усе для подальшої розробки та інтеграції до різних компаній, які потребують захисту своїх об'єктів та також підходить до звичайних користувачів. Збереження даних для звичайних користувачів є повністю безкоштовним, так як було інтегровано розумну систему збереження відеофайлів за рахунок Telegram.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Централізоване система URL: <https://www.camerasecuritynow.com/features/centralized-management> (дата звернення: 01.09.2023).
2. Розподілена система URL: https://www.researchgate.net/figure/Distributed-Video-Surveillance-Network-Architecture_fig1_220166473 (дата звернення: 02.09.2023).
3. Хмарна система: URL: https://www.researchgate.net/publication/320039476_A_CLOUD-BASED_ARCHITECTURE_FOR_SMART_VIDEO_SURVEILLANCE (дата звернення: 22.09.2023).
4. AWS cloud: URL: <https://actuate.ai/security-resources/smart-video-surveillance> (дата звернення: 22.09.2023).
5. Розподілені вузли: URL: https://link.springer.com/chapter/10.1007/978-3-030-37277-4_47 (дата звернення: 02.10.2023).
6. Мережева система: URL: https://www.researchgate.net/figure/Network-video-surveillance-system_fig1_257559144 (дата звернення: 10.10.2023).
7. ATAM URL: <https://insights.sei.cmu.edu/library/architecture-tradeoff-analysis-method-collection/> (дата звернення: 12.10.2023).
8. StakeHolder URL: <https://www.projectmanager.com/blog/what-is-a-stakeholder> (дата звернення: 15.10.2023).
9. 4+1 URL: <https://medium.com/javarevisited/4-1-architectural-view-model-in-software-ec407bf27258> (дата звернення: 18.10.2023).
10. RPI Camera Waveshare URL: <https://www.waveshare.com/product/raspberry-pi/cameras.html> (дата звернення: 21.10.2023).
11. Raspberry pi Pico URL: <https://www.raspberrypi.com/products/raspberry-pi-pico/> (дата звернення: 22.10.2023).
12. Raspberry pi Zero W URL: <https://www.raspberrypi.com/products/raspberry-pi-zero-w/> (дата звернення: 22.10.2023).
13. ARM URL: <https://www.arm.com/glossary/risc> (дата звернення: 23.10.2023).

14. PIR URL: <https://www.sciencedirect.com/topics/computer-science/passive-infrared-sensor> (дата звернення: 25.10.2023).
15. SR501 URL: <https://www.mini-tech.com.ua/datchik-dvizheniya-infrakrasniy-pir-sensor-hc-sr501> (дата звернення: 28.10.2023).
16. Концептуальна модель БД URL: <https://www.sciencedirect.com/topics/computer-science/conceptual-data-model#:~:text=The%20conceptual%20data%20model%20is,processing%20flow%20or%20physical%20characteristics.> (дата звернення: 01.11.2023).
17. MySQL URL: <https://dev.mysql.com/doc/> (дата звернення: 10.11.2023).
18. Telegram API URL: <https://core.telegram.org/> (дата звернення: 12.11.2023).
19. Python URL: <https://www.python.org/doc/> (дата звернення: 20.11.2023).
20. psutil URL: <https://pypi.org/project/psutil/> (дата звернення: 22.11.2023).
21. gpiozero URL: <https://pypi.org/project/gpiozero/> (дата звернення: 23.11.2023).
22. pytz URL: <https://pypi.org/project/pytz/> (дата звернення: 24.11.2023).
23. picamera URL: <https://pypi.org/project/picamera/> (дата звернення: 12.10.2023).
24. Python telegram bot URL: <https://pypi.org/project/python-telegram-bot/> (дата звернення: 30.11.2023).
25. C# URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 05.12.2023).
26. Microsoft URL: <https://www.microsoft.com/uk-ua/> (дата звернення: 10.12.2023).
27. .NET URL: <https://dotnet.microsoft.com/en-us/> (дата звернення: 12.12.2023).

28. Krypton URL: <https://krypton-suite.github.io/Standard-Toolkit-Online-Help/Source/Help/Output/articles/topic38.html> (дата звернення: 16.12.2023).