ЗМІСТ

# ВСТУП

Швидкий розвиток технології впливає на всі сфери сучасного світу, трансформуючи його до невпізнання. Не стала винятком і людська оселя, яка із кожним днем все більше комп'ютеризується для автоматизації рутинних задач і створення комфортних умов життя. Але розумні будинки — це не тільки комфорт і енергоефективність, а й покращені системи безпеки [1]. Спостереження за навколишнім середовищем дозволяє зберегти трату грошей і вберегти власність від злодіїв. Нажаль, рівень злочинності залишається високим [2], тому потреба в кращих системах спостереження тільки росте, особливо це стосується людських домів.

На ринку представлено багато систем відеоспостереження, таких як камери замкнутого типу (CCTV), яким для роботи потрібен центр контролю для моніторингу активності за допомогою камер, або радіочастотне розпізнавання (RFID), яке використовує радіо хвилі для автоматичної ідентифікації людей або об'єктів засобами RFID транспондерів та зчитувачів. Але часто, інформації від однієї системи спостереження може бути недостатньо для автоматичного розпізнавання вторгнення в середовище, за яким ведеться спостереження. Об'єднання декількох джерел інформації, таких як бездротові сенсори та камери, може покращити ефективність системи спостереження. Дійсно, сенсори присутності (такі як RFID, PIR і т.д.) часто генерують хибно позитивні сигнали, вплив яких може бути зменшено використовуючи камери відеоспостереження.

Метою цієї роботи є підвищення безпеки розумного дому за рахунок створення багаторівневої інтегрованої системи контролю станів та об'єктів навколишнього середовища та вироблення адекватних аналізованому стану сповіщень. В пропонованій моделі системи відеоспостереження різні технології, вже встановлені в домі, інтегруються і координуються центральним компонентом щоб забезпечити високий рівень захищеності. В ролі камери відеоспостереження виступатиме мобільна платформа, що дозволить збільшити загальну швидкість реакції системи. Така система дозволить моніто-

ти стан середовища і помічати факт порушення її цілісності. В роботі описуються різні технології і програмні рішення, інтеграція яких в єдину систему дозволить реалізувати описаний сценарій роботи системи відеоспостереження.

# РОЗДІЛ 1. АНАЛІЗ СТАНУ РОЗВИТКУ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ В КОНЦЕПЦІЇ СТВОРЕННЯ РОЗУМНОГО ДОМУ

## 1.1. Концепція та технології створення інтелектуальних будівель

Створення «Інтелектуальних будівель» або «Розумних домів» продовжує набирати популярність у зв'язку з появою все нових доступних і простих в установці модулів для їх побудови. Спочатку, цей термін застосовувався для складних інженерних систем автоматизації, заснованих кабельно-дротовому з'єднанні. Але з появою концепції Інтернету речей (Internet of Things, IoT) «розумні будинки» стали використовувати бездротові інтерфейси зв'язку між компонентами, а також підключення до мережі Інтернет для дистанційного керування. Для того щоб спроектувати компонент систем даного типу, здатний до моніторингу та віддаленого контролю, необхідно зрозуміти основні принципи концепції «Інтернету речей» і особливості архітектури «Інтелектуальних будівель».

Термін вперше був сформульований в 1999 році Кевіном Ештоном(Англ. Kevin Ashton), працівником дослідницької групи в компаніїProcter & Gamble. Він запропонував запровадити радіочастотну ідентифікацію – RFID-мітки або маркери (Radio Frequency IDentification), для відстеження переміщення товарів компанії.

Зміст концепції Інтернету речей можна сформулювати наступним чином: для збільшення комфорту життя людей і надання складних комплексних послуг необхідно створення глобальної інфраструктури, що складається з безлічі речей (віртуальних і фізичних), які з'єднані між собою за рахунок існуючих і таких, що тільки розвиваються, функціонально сумісних технологій інформаційної комунікації.

Фізичні речі в даній концепції - це речі реального фізичного світу (датчики і різні пристрої), а віртуальні – речі інформаційного світу (наприклад, віртуальні гроші, і все, що матеріальну ціну, але не має фізичного носія). Кожна з таких речей може бути ідентифікована або інтегрована в мережу.

Якщо подивитися з практичної точки зору, то концепція Інтернету речей спрямована на автоматизацію діяльності в різних сферах діяльності, виключення з них людини, і, як наслідок, підвищення ефективності економічних і суспільних процесів.

Цікавою є модель Інтернету речей у вигляді «чотиришарового пирога».

Перший шар включає в себе ідентифікацію об'єктів, наприклад, за допомогою датчиків або RFID-маркерів. На цьому етапі кожна Інтернет-річ отримує засіб зв'язку з навколишнім світом і унікальні дані.

Другий шар - обслуговування споживача або сервіс. тут об'єкти об'єднуються в мережі для виконання певної функції в рамках поставленого завдання. Найпоширенішими прикладами є системи моніторингу навколишнього середовища за допомогою бездротових сенсорних мереж і, звичайно, «Розумні будинки».

Третій шар заснований на тенденції урбанізації міського життя, так звані «Розумні міста», які мають на увазі дослідження і збір інформації на конкретній території (кварталі, районі тощо) і надання всієї необхідної інформації її мешканцям.

Четвертий і найвищий рівень - це сенсорна планета, коли всі існуючі мережі об'єднуються в глобальну інформаційну інфраструктуру.

Іншими словами, Інтернет речей - це мережа мереж.

## 1.2. Концепція архітектури “Розумних будівель”

Автоматизація будівель почала впроваджуватися ще в 60-70-х роках минулого століття, тоді і було сформульовано поняття «розумний будинок». Спочатку воно формулювалося як «будинок для ефективного використання робочого простору», але на сьогоднішній день його сенс набагато ширше.

«Розумний будинок» для сучасної людини - не просто система раціонального використання робочого простору, це інтелектуальна система, яка об'єднує в собі як інженерні комунікації та системи безпеки, так і інформаційні системи будівлі. Такі об'єднані рішення покликані підвищити комфортність

приміщень і забезпечити їх безпеку. У багатьох випадках приводом для установки систем розумного будинку є бажання підвищити ступінь комфорту за рахунок автоматизації рутинних дій.

Реалізація розумного будинку ділиться на дві частини: апаратну і програмну. На рис. 1.1. представлена одна з можливих і найбільш часто використовуваних схем апаратного забезпечення розумного будинку. Апаратна частина, як правило, складається з контролера, модулів розширення і кінцевого обладнання.

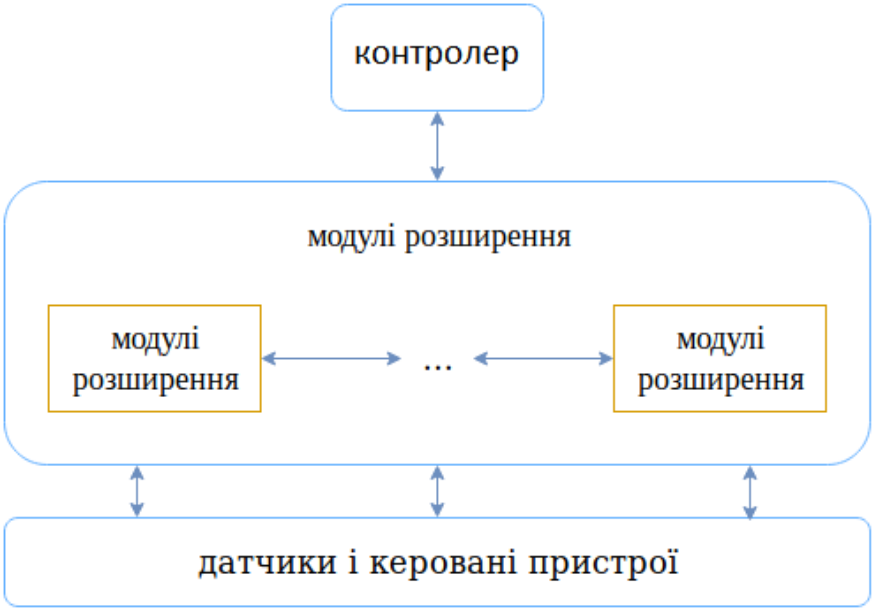


Рисунок 1.1 Типова схема апаратного забезпечення розумного дому

Контролером може виступати ПК, планшет, смартфон, на які встановлюється програмне забезпечення для управління системами розумного будинку всередині домашньої (робочої) мережі або через мережу Інтернет.

Модулями або платами розширення називають спеціальні пристрої з підключеними датчиками різного типу і керованими частинами системи.

В категорію кінцевого обладнання входять датчики для відстеження різних параметрів середовища і стану пристроїв, які необхідно регулювати.

Програмне забезпечення може бути реалізовано багатьма способами: від звичайного пульта управління до складного синхронізованого комплексу

програмного забезпечення, що встановлюється на будь-яку кількість гаджетів і ПК власника інтелектуальної будівлі і повністю автоматичних систем, що включають в себе елементи інтелектуалізації.

### 1.2.1. Дистанційне керування розумним домом

Розвиток бездротових інтерфейсів зв'язку та розширення концепції Інтернету речей призвело до виходу систем розумних будинків за межі приміщень і будівель, в яких вони встановлені. Взаємодія автоматизованих комплексів з мережею Інтернет дала можливість управляти ними в режимі віддаленого доступу.

Дистанційне керування має ряд переваг перед управлінням виключно автоматичним і через системи, що працюють в межах встановленого радіусу. Перш за все, це підвищення рівня безпеки і комфорту.

Основна перевага — підвищення комфорту використання простору будинку або офісу, що і є основною ідеєю для створення розумного будинку. При наявності функції віддаленого управління користувач може включити, вимкнути або налаштувати потрібні йому пристрої (освітлення, побутові прилади та інші системи), де б він не перебував. Наприклад, підігріти чай перед приходом додому з роботи або заздалегідь включити опалення, якщо за показаннями датчиків в приміщенні некомфортна температура.

Але головною перевагою варто вважати підвищення рівня безпеки. При відсутності людей в приміщенні можуть відбутися ситуації, що загрожують схоронності майна і самої будівлі. Для запобігання подібних інцидентів можливе підключення камер для відеоспостереження за обстановкою в приміщенні або віддалений моніторинг з допомогою аналізу інформації, що надходить від різних датчиків, які використовуються в системах безпеки (датчики вогню, датчики відкриття / закриття дверей і т. д.). В тому числі, автоматичне відключення електроприладів та світла допоможе не тільки заощадити споживання електроенергії, а й зменшити ризик самозаймання електропроводки в порожньому приміщенні;

Система дистанційного керування в більшості випадків проста: користувач, з використанням гаджетів або ПК, відсилає команди системі через веб-додаток або зі сторінки веб-сайту, система аналізує отриману команду і за допомогою контролера виконує вказану дію.

Так як дистанційна робота з датчиками, камерами і іншим обладнанням передбачає зберігання, обробку і аналіз великої кількості інформації, і зручний доступ до результатів декількома користувачами, доцільне застосування хмарних технологій. При такому підході в системі розумного будинку з'являється хмарний сервіс, який дозволяє позбавити користувача турбот щодо обслуговування серверної частини системи, за допомогою якої відбувається управління всією системою.

Для підключення хмари існує два варіанти. Перший - використання хмари в якості контролера. Всі пристрої підключаються безпосередньо до хмари і управляються з неї безпосередньо. Керуюча частина системи може бути повністю винесена за межі будівлі. Другий варіант - збір пристроїв на контролерах і підключення їх до хмари. В цьому випадку хмара буде керувати діями контролерів, передаючи інформацію між кількома модулями. контролер розміщується всередині будівлі, але все програмне забезпечення винесено на сервіс.

Обидва варіанти передбачають винесення аналізуючої частини системи в хмару, що дозволяє знизити вимоги до контролера, головна відмінність в тому, що в першому випадку управління пристроями ведеться автономно один від одного, а в другому - через загальний контролер.

Багато сучасних модулів працюють по власних протоколах передачі даних і, взаємодіють з Інтернет-сервісами тільки через свої API, що створює складності обміну інформацією між пристроями безпосередньо, а також заважає розширенню системи розумного будинку.

## 1.3. Системи безпеки розумних будинків

Потреба забезпечення безпеки змушує людей шукати все нових способів і технологій для захисту своєї власності. Багато систем пропонує контроль над внутрішнім та/або зовнішнім середовищем дому. Об'єднання різних елементів безпеки будівлі укупі з іншими технічним обладнанням в єдину автоматизовану мережу дає, мабуть, самий відчутний ефект від вкладених в «розумний дім» коштів. Системи безпеки розумного дому наведено на рис. 1.2.

Мова йде не тільки про захист від непрошених гостей, але і про запобігання аварійних ситуацій. Система безпеки покликана забезпечити особисту безпеку мешканців приватного будинку, співробітників підприємства, злагоджене функціонування всіх служб житлового, офісного, виробничого або торгового будівлі.

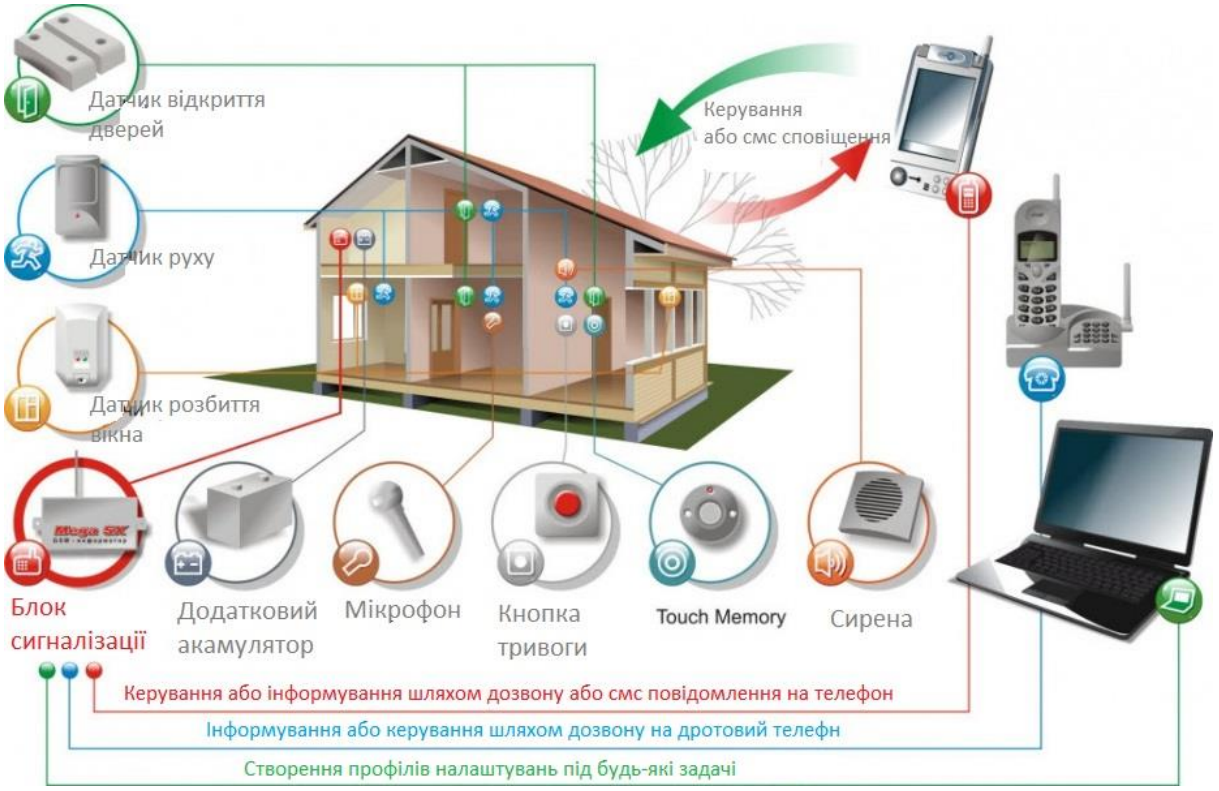


Рисунок 1.2 Системи безпеки розумних будинків

Охоронна сигналізація. Всі можливі шляхи попадання в дім (вікна, двері, периметрце навколишньої ділянки) оснащуються сенсорам проникнення. Всередині кімнат дому розміщують пристрої, що реагують на переміщення людей. При цьому домашні тварини можуть переміщатися по будинку вільно.

На випадок, якщо зловмисники все ж проникли в будинок, обдуривши господарів, встановлюються приховані тривожні кнопки, їх можна носити і в кишені.

Контроль здійснюється зонально, реакція на зміну ситуації може бути різною: повідомлення на пульт охорони поліції або іншого відомства, SMS господареві або його довіреним особам. Наприклад, про несанкціоноване проникнення в будинок повідомлення буде отримувати служба охорони, а про відкриття сейфа або навіть про те, що хтось надмірно цікавий входив в робочий кабінет - тільки господар будинку.

Охоронна сигналізація може працювати в декількох режимах. Наприклад, в нічний час під охорону може здаватися тільки периметр території та приміщення першого поверху. Які системи безпеки можуть входити до складу «розумного будинку»?

Контроль доступу. Всі двері, ворота, шлагбауми можуть бути оснащені пристроями доступу, як кодовими, так і зчитують інформацію з карт доступу, радіочіпи, або розраховані на отримання біометричних даних. Персоналізація засобів доступу дозволить мати точну інформацію про пересування по будівлі домочадців або працівників. Не всім співробітникам, відвідувачам або домашнім працівникам буде довірено право входу в деякі приміщення. До систем контролю доступу можна віднести і захист вікон: ролети і механічні віконниці. Їх закриттям і відкриттям можна управляти за допомогою заданого алгоритму або вручну.

Відеоспостереження. Домофон; камери, розташовані поза і всередині будівлі, через контролер підключаються до телекомунікаційних пристроїв. Вони дають можливість спостерігати за тим, що відбувається всередині поза будівлею з будь-якого телевізора або комп'ютера, встановленого в будинку. Або дистанційно, з пульта охорони, комп'ютера або екрану смартфона.

Камери можуть бути як видимими, так і зовсім непомітними. Події реєструються на відео, записи належний час зберігаються на сервері.

Пожежна і аварійна сигналізація, безпека інженерних систем. Аварійні інженерні мережі припускають наявність датчиків вогню, диму, витоку газу і води. У разі небезпеки тривожне сповіщення надходить в ДСНС або диспетчерську житлових або відомчих служб. При відповідному сигналі буде припинено надходження в будівлю газу, перекриті водопровідні вентилі. Включиться система автоматичного пожежогасіння, якщо вона є в будівлі.

Енергетична безпека забезпечується системою енергозабезпечення. У разі припинення подачі електроенергії протягом декількох секунд вводиться в дію автоматичний резерв, включається автономна генераторна установка, побудована на базі двигуна внутрішнього згоряння або газового генератора. Можуть використовуватися і акумуляторні батареї, якщо їх потужності і часу роботи досить.

Для забезпечення першочергових потреб виділяються окремі «аварійні» енергетичні ланцюги. Життєво важлива чітка безперебійна робота енергетичної системи в медичних установах, транспортних вузлах і місцях скупчення людей, стратегічних об'єктах.

Локальне забезпечення електроенергією електронних компонентів беруть на себе джерела безперебійної енергії. Система охорони для розумного дому, напрямлена на його автоматизацію, повинна виконувати багато різних функції. Як показано на рис. 1.2., вона помічає пожежу на ранніх стадіях завдяки сенсору температури, повідомляє про можливе затоплення або витік газу, а також обмежує можливість доступу до приміщень дому.

### 1.3.1. Системи відеоспостереження розумних будинків

Системи відеоспостереження як засіб охорони служать людському суспільству вже понад 70 років [3]. Вони працюють практично у кожному місті, магазині, банку. Навіть у менш розвинутих країнах ринок засобів відеоспостереження швидко зростає. Для збільшення якості відеоспостереження, уже багато років ведуться розробки у сферах покращення якості відео і зйомки, кодування зображень, стрімінгу відео і збереження та стиснення даних. Ці

розробки збільшують ефективність відео нагляду, дозволяючи зберегти більше деталей використовуючи менше місця. Сучасні рішення стають все більш вигідними, компактними, водночас забезпечуючи більшу якість зйомки.

Система відеоспостереження є однією з найважливіших компонент розумного дому. Зараз на ринку представлено багато рішень, є варіанти, що прив'язані до центральної системи дому, так і працюючі окремо від неї, і користувач може обрати систему собі до смаку: просту мережу IP камер, як з розумною обробкою відео так і без неї, систему прихованих камер, дротову або бездротову.

На ринку представлені такі варіанти систем відеоспостереження:

ADT's Home Security Starter Kit. Система охорони розумного дому, що працює на основі платформи Samsung's SmartThings. Постачається як набір компонентів, які користувач повинен самостійно розмістити в своєму домі. Складається з:

– ADT Security Hub, семи дюймового кольорового тачскріну, який слугує контролюючий пристроєм системи.

– Двох бездротових сенсорів стану вікна та дверей, сенсору руху та камери.

– Додаткової батареї, для підтримки безперебійного живлення при відсутності електропостачання.

Ціна початкового комплекту – $400. Камери відеоспостереження можна докуповувати окремо за ціною $70.

Vivint. Комплект складається з камери-вхідного дзвінка, двох відеокамер, розумних замків, термостату і системи контролю дверей гаража. Система здатна записувати відео 24/7, яке можна переглядати в мобільному додатку і зберігати дані усіх датчиків за довгий період часу.

Система потребує професійного встановлення, контроль усіх елементів (замків, дверей гаража, температури) здійснюється за допомогою мобільного додатку Vivint. Також через нього можна отримувати сповіщення про викори-

стання дверного дзвінка, відкриття дверей, переміщення всередині дому. Підтримуються голосові асистенти від Google та Amazon.

Ціна комплекту - $500, річна підписка обійдеться ще в $300, додаткові компоненти, як то камери, сенсори диму і газу, можна придбати окремо.

Xfinity. Система, що потребує професійного встановлення і пропонує 24/7 моніторинг, екстрену батарею, що гарантує роботу системи при перебоях в електропостачанні та панель керування, з якої можна віддавати команди окремими елементами дому.

Xfinity включає в себе сенсори відкриття дверей та вікон, сенсор руху та планшет, який слугує головним терміналом системи. Компанія пропонує доповнення системи: камери внутрішнього та зовнішнього відеоспостереження, термостат, розумні крани та сенсори диму.

Система сумісна з багатьма приладами інших компаній, що пропонують рішення для розумного дому, наприклад з розумними замками August, контролерами дверей гаража Chamberlain, термостатами Nest та лампами Phillips.

Система постачається за умовою підписання 2 річного договору, який обійдеться в $720.

Проаналізувавши вищеописані системи, можемо виокремити їх основні недоліки:

– Закриті протоколи обміну даними, які роблять неможливим розширення і налаштовування системи компонентами інших виробників.

– Висока вартість продуктів і додаткових компонентів, система місячної підписки.

– Необхідність професійного встановлення системи, яка може бути недоступною в багатьох країнах.

– Обмежена можливість керувати відеоспостереженням, віддавати команди і реагувати на зміни середовища.

### 1.3.2. Системи відеоспостереження із використанням безпілотних літаючих апаратів

Дрони, також відомі як безпілотні літаючі апарати (БЛА) це машини, оснащені сенсорами, такими як камери, акселерометри, сенсори нахилу і магнітного поля, вимірювачами інерції. Дрони можуть контролюватися людиною оператором, так і працювати повністю або частково автоматично. Дрони це нова і швидко еволюціонуюча технологія, основним застосуванням якої донедавна була військова справа [4]. Та останнім часом сфера використання дронів росте надзвичайно швидко, БЛА проникають у все нові професійні та промислові середовища. Їх використовують рятувальні служби, для пошуку вижившних під час рятувальних операцій, науковці, для проведення досліджень в сурових кліматичних умовах. Вони знайшли своє місце і в сфері безпеки, поліція та приватні охороні компанії використовують дронів для відеоспостереження за небезпечними ситуаціями без ризику для життя операторів.

Згідно дослідження ринку дронів, в США станом на 2016 рік кількість цивільних дронів перевищить 1млн. загальною вартістю $200млн. а загальний прибуток галузі – від $200 до $400млн.[5] Більшість варіантів використання дронів як засобу відеоспостереження в домашніх умовах зараз, це самостійні рішення, які не інтегруються в систему керування розумним домом. Вони пропонують окремий пульт, з якого можна керувати дроном і дивитися відео з його камери, що обмежує застосування БЛА в контексті системи розумного дому. Зараз на ринку не представлено жодної інтегрованої системи охорони середовища дому, яка б використовувала дрон в якості камери відеоспостереження, але є компанії, які обіцяють представити такі розробки вже найближчим часом.

Sunflower Home Awareness System. Компанія Sunflower розробляє новітню систему безпеки, що об’єднує розумні наземні сенсори з літаючою камерою. При появі нештатної ситуації користувач отримуватиме сповіщення на телефон і зможе відправити дрон-камеру, відео з якої транслюватиметься в додаток. Система оснащена здатністю до навчання, з часом вона знатиме зви-

чайні маршрути жильців, зможе розпізнавати чужаків і відлякувати їх звуковими та світловими сигналами.

Рисунок 1.3 Реклама Sunflower Home Awareness System

Основним компонентом є сенсор Sunflower Smart Light, який суміщає в собі 360° сенсор руху, мікрофон, сенсор вібрацій і сонячну панель для зарядки батареї. Дрон оснащений GPS навігатором, автопілотом, камерою з високою роздільною здатністю і камерою для стабілізації зображення. На рис 1.3. зображено як виглядатиме система в дії.

В комплекті постачається 4 сенсори, літаюча камера та мобільний додаток, з якого можна керувати системою. Оголошена ціна - $159 за один сенсор і $799 за камеру, випуск системи для тестування обіцяють в кінці 2018 року.

Були і невдалі спроби розробки систем із дронами-охоронцями: компанії Hawkeye та Aire пропонували покупцям профінансувати їх рішення за системою краудфандинг, але, не дивлячись на успішний збір коштів, проекти закрили, повернувши кошти клієнтам.

Варто зазначити, що дрон не є повноцінною заміною мережі камер, в обох рішень є свої переваги і недоліки, використовуючи їх разом можна забезпечити найвищий рівень безпеки розумного дому.

## 1.4. Постановка задачі

Отже, для забезпечення відео нагляду за навколишнім середовищем розумного дому, необхідно розробити ПЗ системи відеоспостереження, яке інтегрується в існуючу систему автоматизації розумного будинку і об’єднує центральний контролер, сенсори та дрон в єдину систему моніторингу, що здатна реагувати на нештатні ситуації і повідомляти про них користувача. Для успішного проектування необхідно виконати такі задачі:

– Поставити вимоги до розроблюваного ПЗ і на їх основі – вимоги до використовуваних технологій.

– Провести аналіз можливих технологій розробки і потенціальних середовищ розгортання системи, вибрати такі, що задовольняють вимогам.

– Розробити алгоритми роботи та архітектуру системи відеоспостереження розумного дому.

– Розробити програмне забезпечення модулів системи відеоспостереження.

## Висновки до розділу

В цьому розділі була розглянута концепція Інтернету речей, її архітектура, компоненти та недоліки. На основі цього виведена концепція розумного будинку, пояснена типова організація його компонентів і пояснена можливість дистанційного керування таким об’єктом. Наступним кроком стало визначення системи безпеки таких будинків і опис її складових. Детальніше розглянута одна з таких систем – система відеоспостереження, функції якої виконуватиме розроблюване програмне забезпечення. Наступним кроком став аналіз існуючих на ринку систем відеоспостереження і визначення їх недоліків. В результаті проведеного в цьому розділі дослідження були поставлені задачі цієї роботи.

# РОЗДІЛ 2. ТЕХНОЛОГІЇ СТВОРЕННЯ СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ СТАНУ ТА ОБ'ЄКТІВ РОЗУМНОГО ДОМУ

## 2.1. Специфікація вимог до системи відеоспостереження

Основою успішної розробки системи програмного є її специфікація. Чітко сформульована специфікація допомагає уникнути багатьох проблем в процесі розробки і цим зробити його більш простим.

На найвищому рівні абстракції система, що розробляється в даній роботі є набором програмних компонентів, що інтегруються в готове середовище керування розумним домом. Вони забезпечують зв'язок сенсорів, центрального серверу та мобільної платформи, пов'язуючи їх у систему відеоспостереження.

Таке програмне забезпечення повинне реалізовувати наступні функції:

– підтримка постійного зв'язку із сенсорами, реєстрація показників та збереження їх у сховищі даних.

– отримання команд від користувача, інтерпретація і виконання

– передача інструкцій дрону залежно від показників сенсорів або команд користувача

– отримання відео потоку від дрона, розпізнавання об'єктів на відео і виконання дій залежо від його результатів

– надання можливості отримання даних системи клієнтським частинам

## 2.2. Вибір архітектури системи відеоспостереження

Одним із найістотніших етапів розробки будь-якої технічної системи є створення її архітектури, вона слугує фундаментом всього наступного процесу побудови. Якісна архітектура забезпечить легке масштабування і гнучкість всього проекту, зменшить поріг входження для розробників та користувачів і навпаки, погано продумана архітектура зробить розробку складнішою: чим більшим ставатиме проект тим відчутнішими стануть перешкоди перед роз-

робниками, добавлення нового функціонал буде більш незграбним і каверзним. Як результат, на якомусь з етапів розробки ми отримаємо програмний продукт, масштабування і доповнення якого стануть настільки ресурсозатратними, що буде вигідніше почати процес розробки з початку. Сформуємо критерії хорошої архітектури і вимоги, які повинна задовольняти розроблювана система:

– Ефективність

Першочергово, програма повинна вирішувати поставлені задачі і добре виконувати свої функції, тобто швидко обробляти дані і передавати повідомлення.

– Гнучкість

Внесення змін до існуючого функціоналу повинно бути простою задачею, чим менше помилок і проблем це викликає – тим більш гнучкою є система. Компоненти повинні бути достатньо незалежними один від одного, щоб зміна одного з них не призводила до помилок в інших частинах системи.

– Розширюваність

Розширюваною можна назвати систему, в яку можна додати нові сутності і функціонал, не змінюючи і не переписуючи при при цьому існуючі компоненти.

– Можливість перевикористання

Систему варто проектувати так, щоб її компоненти можна було повторно задіяти при розробці інших додатків вносячи мінімальні зміни.

Розглянемо підходи до проектування архітектури, які дозволять задовольнити поставлені вимоги.

### 2.2.1. Вибір підходу до проектування архітектури системи відеоспостереження

«Мікросервіси» - ще один новий термін в мінливій сфері розробки ПЗ. Будь-яка нова технологія – це завжди невідома територія до якої варто ставитися досить насторожено, необхідно провести дослідження і оцінити ризики.

Конкретно цей термін описує стиль розробки ПО, який світова спільнота знаходить все більш і більш привабливим. За останні кілька років було створено безліч проектів, що використовують цей стиль, і результати досі були досить позитивними. Настільки, що для багатьох компаній та команд по всьому світу цей стає основним стилем розробки ПО.

Немає єдиного формального визначення мікросервісної архітектури, існують певні характеристики, які допомагають нам зрозуміти цей стиль [6]. По суті, архітектура мікросервісу є методом розробки програмних додатків як набору незалежних для розгортання невеликих, модульних сервісів, в якій кожен сервіс запускається як унікальний процес і спілкується через чітко визначений, легкий механізм, що служить для досягнення бізнес-цілей.

Способи взаємодії сервісів між собою залежать від вимог вашої програми, але багато розробників використовують HTTP/REST за допомогою JSON або Protobuf. Професіонали DevOps, звичайно, можуть вільно вибирати будь-який комунікаційний протокол, який вони вважають підходящим, але в більшості випадків REST (Representational State Transfer) є корисним методом інтеграції через його простоту і універсальність.

Щоб краще осягнути ідею мікросервісів, варто згадати підхід, який був домінуючим в сфері розробки програмного забезпечення довгі роки – моноліт. Основна ідея монолітної архітектури в тому, що додаток будується як єдине ціле, всі частини якого запускаються одним процесом. Це прямолінійний і зрозумілий підхід до побудови, основними перевагами якого є простота розробки, розгортання і тестування, які можуть виконуватися розробником на власній машині, легкий процес масштабування додатку. На рис. 2.1. зображено приклади монолітної і мікросервісної систем. Додаток-моноліт може благополучно виконувати свої функції, але з розповсюдженням хмарних технологій розміщення додатків, все більше користувачів відмовляється від цього підходу через вагомі недоліки такої архітектури, а саме:

– для внесення навіть найменших модифікацій в систему потрібно перезбирати весь проект і викладати його на сервер, що дорого і займає багато часу.

– В процесі розробки, задача збереження чистої модульної структури стає все важчою, модифікація логіки одного компоненту може призвести до зміни поведінки інших,

– Якщо один з компонентів додатку потребує масштабування, доведеться дублювати всю систему.

– Застарілість технологій. Багато компаній підтримують проекти, створені з використанням неактуальних зараз технічних рішень.

– Високий поріг входу для нових розробників.

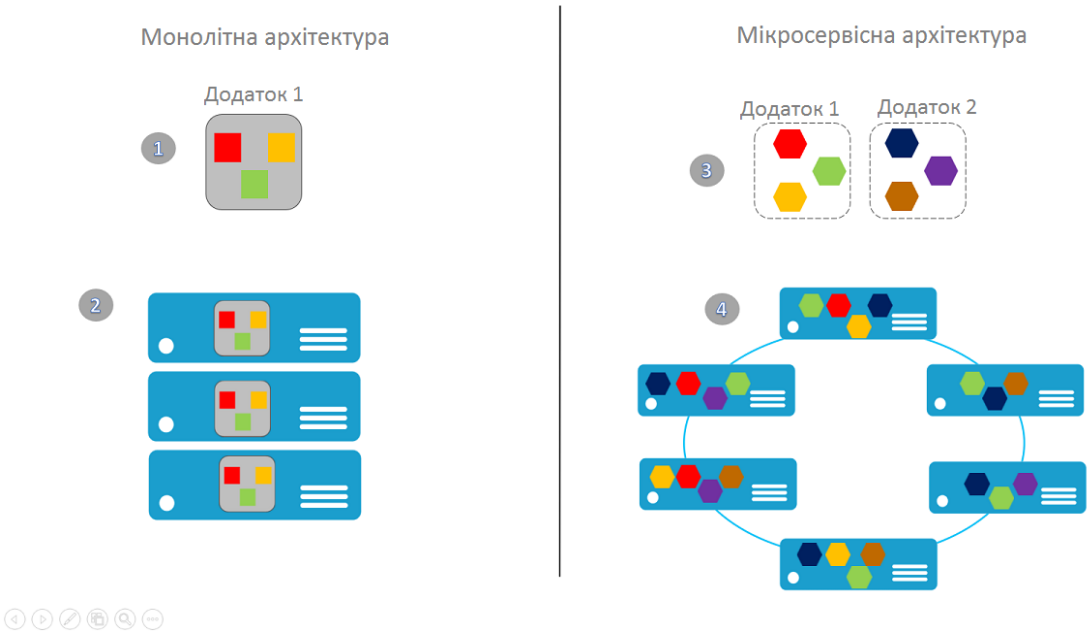


Рисунок 2.1 Порівняння монолітної та мікросервісної архітектур

Мікросервісна архітектура запропонована як ідея, націлена на вирішення недоліків моноліту. Спробуємо окреслити загальні характеристики мікросервісів:

– Розбиття на сервіси

Головна причина використання сервісів, а не пакетів, це можливість викладати на сервери окремо кожен із компонентів. Це робить частини незале-

жними одна від одної, модифікація однієї із них не призводить до змін логіки інших (хоча й може знадобитися переробити договорености між сервісами).

– Організація навколо вимог бізнесу

При розробці моноліту, звичайною практикою є розподіл команд розробки по використовуваних технологіях, наприклад команда розробки інтерфейсів користувача, команда бекенду, команда баз даних. Це призводить до проблем з інтеграцією їх розробок навіть при невеликих змінах, через необхідність взаємодії між різними командами. Мікросервісний стиль заохочує створення команд, які працюють над вирішенням єдиної бізнес задачі, що спричинює формування команд з професіоналів в різних областях.

– Децентралізоване керування даними

Модель предметної області може відрізнятися у різних команд розробки одного продукту, одне поняття може розглядатися з декількох позицій, що призводить до різниці в його представленнях. Монолітні додатки зазвичай використовують єдину базу даних, збереження в ній різних моделей об’єктів може привести до надлишковості Мікросервісна архітектура дає змогу кожному сервісу керувати своєю базою даних, яка зберігає саме те представлення, яке необхідне для реалізації бізнес потреби.

Не дивлячись на усі переваги мікросервісів, вони не стали універсальним рішенням усіх проблем, що постають перед архітекторами систем. Цей підхід змушує розробників вирішувати ряд нових проблем.

– Комплексність. Мікросервісне рішення складається з більшої кількості компонентів, ніж монолітний аналог. Кожен окремий сервіс простіший, ніж у моноліті, але система в цілому стає складнішою.

– Складність операційної підтримки. Потрібні вмілі DevOps інженери, неперервна розгортка і автоматичний моніторинг.

– Цілісність даних. Так як кожен сервіс відповідає за збереження даних, забезпечення цілісності може бути проблемою.

– Затримки мережі. Використання багатьох невеликих сервісів призводить до інтенсивнішого числа взаємодій між ними. Зі зростанням кількості взаємодій у системі мережеві затримки можуть стати проблемою.

Проаналізувавши особливості мікросервісної і монолітної архітектур, можемо прийти висновку, що для розробки системи відеоспостереження середовища розумного дому краще обрати мікросервісну архітектуру. Гнучкість, яку вона пропонує варта потенціальної комплексності. Можливість зробити додаток по-справжньому розширюваним і високоефективним відкриває великі можливості по впровадженню системи не тільки в середовищі розумного дому.

## 2.3. Специфікація вимог до технологій розробки системи відеоспостереження

Визначившись із архітектурою ми можемо поставити вимоги до технологій, які будуть використані для її створення.

Вимога ефективності системи змушує шукати найбільш продуктивні рішення, такі, що дозволять відеоспостереженню і розпізнаванню працювати в режимі реального часу. Затримки при обробці неприпустимі, так як вони потенційно можуть поставити під небезпеку користувача.

Для виконання вимоги розширюваності, вибрані технології повинні підтримувати багато апаратних засобів, не залежачи від конкретного виробника або реалізації. Також потрібно, щоб система дозволяла збільшити кількість зовнішніх пристроїв – необхідна підтримка обробки багатьох потоків даних. Сприяють досягненню розширюваності технології з відкритим кодом, відкриті протоколи дають можливість зв’язувати нові компоненти з існуючими.

### 2.3.1. Універсальний інтерфейс пристроїв розумного дому

Програмне забезпечення автоматизації розумного дому- це додатки, якеі полегшують контроль над типовими компонентами розумного дому, такими як освітлення, контроль клмату, контроль доступу, розприскувачі і т.д.

Ці системи, як правило, підтримують декілька інтерфейсів для зв'язку із зовнішніми пристроями, такі як XMPP, Zigbee, Z-Wave, X10 тощо.

Зараз серед систем автоматизації з відкритим кодом найбільш розвинутими є openHAB та Home Assistant. Обидві системи мають багато спільного – велику спільноту та позиціонують себе як найбільш гнучку платформу для контролю приладів розумного дому.

openHAB написаний на Java, створений раніше, але, на даний момент, розвивається більш повільно. Новий функціонал, як то підтримка WebCoRE, залишається на розробку спільноті. Home Assistant побудований із використанням Python та підтримує двох тижневий цикл розробки, що дозволяє частіше випускати нові версії і підтримувати більше новинок.

Процес встановлення схожий для обох рішень, хоча Home Assistant підтримує більшу кількість платформ, а openHAB обмежений ОС на основі ядра Linux. Зараз в openHAB більш детальна документація, але спільнота Home Assistant активно працює над покращенням своїх мануалів.

Обидва продукти підтримують велику кількість пристроїв, які можна приєднати до системи і може бути складно визначити, в кого більше, але, роблячи висновки по темпу розробки, Home Assistant може інтегрувати більше девайсів. Також процес під'єднання більш простий, здатність автоматичного пошуку та підключення дуже зручна, коли в openHAB для кожного пристрою потрібно вручну змінювати файл налаштувань. Також обидва рішення пропонують просте і гнучке API для керування приладами.

Можемо зробити висновок, що Home Assistant краще підходить для поставленої задачі, так як він підтримує більше пристроїв і робить їх інтеграцію простим завданням.

### 2.3.2. Мова програмування

Основними вимогами до мови програмування, на якій буде написана система відеоспостереження є швидкодія, кросплатформеність та простота переносу на різні платформи, підтримка багато потокової обробки даних, під-

тримка багатьох протоколів зв’язку в мережі. Зараз розробнику доступний великий вибір мов для реалізації серверних додатків, як то Python, Java, C#, Ruby, Golang.

Python та Ruby – популярні інтерпритовні мови із багатими вбудованими бібліотеки та неосяжною кількістю модулів, написаних сторонніми розробниками. Вони роблять процес написання коду простим та ефективним для розробника. Недоліком цих мов є їх інтерпритовність, яка заповільнює процес виконання програми та змушує встановлювати інтерпретатор в середовище розгортання системи.

Java та C# – компільовані мови, які найчастіше використовуються для розробки додатків бізнес рівня. За роки існування, вони забезпечили собі ім’я як стабільні, високо рівневі мови – надійний вибір для проектування і розробки нового програмного продукту. Однак для запуску вони потребують додаткових середовищ, а їх сконцентрованість на об’єктно-орієнтованій парадигмі обмежує процес розробки. Також швидкість компіляції є проблемою, з ростом проекту все більше часу займатиме очікування збирання проекту.

Golang або Go — статично компільована і типобезпечна мова з багатою стандартною бібліотекою [7]. Вона була створена згідно потреби Google мати ефективний інструмент для роботи із серверами та розподіленими системами, а також для більш повного використання потенціалу багато ядерних комп’ютерів. Go програми компілюються в об'єктний код цільової платформи і можуть працювати безпосередньо, без додаткових компонентів.

Підтримка паралельного виконання в Go – одна із найважливіших її можливостей. Ключове слово go дозволяє запустити вибрану частину програми в окремій горутині (корутині) – абстракції над ідеєю потоків ОС, таким чином не прив’язуючи розробника до ручного керування. Рантайм контролює розподіл горутин по потоках самостійно, оптимізуючи процес.

З огляду на означені особливості Go можемо зробити висновок, що вона підходить для створення системи, мультиплатформенність забезпечує підтримку різних центральних контролерів, а швидкість виконання і підтримка

багато потоковості забезпечать обробку даних від сенсорів і передачу команд іншим пристроям в режимі реального часу.

### 2.3.3. Протоколи зв’язку

gRPC (gRPC Remote Procedure Calls) це система виклику віддалених процедур (RPC) з відкрити кодом, розроблена компанією Google. Вона використовує HTTP/2 в якості транспорту, Protocol Buffers як інтерфейс мови опису взаємодії і надає такий функціонал, як аутентифікацію, двонапрямлений стрімінг і контроль потоку, блокуючі та не блокуючі виклики, таймаути. Є вбудовані інструменти для генерації крос-платформених клієнтських та серверних частин для багатьох мов програмування.

Основні переваги, які зумовлюють вибір gRPC як фреймворку зв’язу між компонентами, є:

- формат Protocol Buffer – мультимовний, багатоплатформений, розширюваний механізм для серіалізації структурованих даних. Він виконує функції XML: або JSON, але, при цьому, повідомлення займають менше місця, передаються швидше, опис структур даних спрощується.
- Асинхроні виклики дозволяють не блокувати мережу для отримання нового повідомлення, це дозволяє одночасно опрацьовувати декілька запитів і відповідати на них.
- Двонапрямлений стрімінг RPC – клієнт і сервер обмінюються метаданими, описами очікуваних структур повідомлень і відкривають дуплексне з’єднання. Вони можуть писати незалежно один від одного і обмінювати потоками повідомлень, змінюючи поведінку залежно від отриманих результатів, це робить gRPC ідеальним вибором для систем, які повинні підтримувати постійний зв’язок компоненті і реагувати на різні події в режимі реального часу.

Для зв’язку із зовнішніми системами використовується формат передачі даних JSON, а для реалізації потокової передачі відео протокол RTSP

### 2.3.4. Система розпізнавання об’єктів на відео

Однією із вимог, поставлених до системи, яка розробляється в цій роботі є здатність розпізнавати об’єкти на відео. Задача аналізу зображень і, відповідно, відео, - комплексна і глибока, підкріплена вражаючим математичним апаратом і технологічними інноваціями. Обробка відео і пов’язані з розпізнаванням алгоритми виходять за межі цієї роботи, тому необхідно знайти готове рішення, яке можна інтегрувати в систему. Так як очікується, що сервер розумного дому не буде потужним, варто подумати про рішення, які не потребують обчислювальних ресурсів користувача, наприклад, хмарні сервіси. Це сервіси з дистанційним керуванням, які надають користувачу обчислювальні потужності та засоби зберігання даних із оплатою яка залежить від інтенсивності. Ви використовуєте потужності для роботи вашого ПЗ, а сховища - для зберігання даних, з якими працює це ПЗ.

Надання послуг засобами хмарних сервісів є однією з найбільш швидко ростучих моделей розповсюдження ПЗ сьогодні. Згідно з прогнозом, він виросте в період 2015-2021 років більш ніж в 3 рази, з $31.4 мільйонів до $117.1 [8]. І це не дивно, адже вони дозволяють делегувати рішення багатьох технічних задач, як то покупка і адміністрування заліза, встановлення та оновлення програмного середовища, імплементація сервісів для доступу і алгоритмів обробки на провайдера. Користувачу залишається тільки використовувати потрібні йому функції.

Найпопулярнішими сервісами по розпізнаванню відео є Google Cloud Intelligence та Amazon Rekognition, розроблені компаніями Google та Amazon відповідно. Вони пропонують широкий спектр широкий спектр рішень, деякі з яких можна порівняти в термінах функціональності, якості, ефективності та ціни.

Таблиця 2.1

Особливості хмарних платформ розпізнавання відео

| Функція | Google Cloud Intelligence | Amazon Rekognition |
|---|---|---|
| Розпізнавання об’єктів на відео (класифікація) | ✓ | ✓ |
| Розпізнавання об’єктів на відео потоці в режимі реального часу (класифікація) | | ✓ |
| Розпізнавання лиць | ✓ | ✓ |
| Розпізнавання емоцій | ✓ | ✓ |
| OCR | ✓ | |
| Порівняння лиць | | ✓ |
| Розпізнавання об'єктів на відео | | ✓ |
| Розпізнавання об'єктів на відео (місцезнаходження) | | |
| Пошук зображень | | |
| Ціна | Перші 1000 хвилин в місяць безплатні, 1001-10000 - $0.10 | Перші 1000 хвилин в місяць безплатні, 1001-10000 - $0.10 |

Бачимо, що Google та Amazon пропонують майже однаковий набір функцій, але для розроблюваної системи важливими є можливість розпізнавати об’єкти в режимі реального часу та здатність порівнювати обличчя (що дозволить відрізняти людей, яким дозволено перебувати на спостережуваній території). Для реалізації потокової обробки в реальному часі, відео потрібно постійно завантажувати на Amazon Kinesis Video Streams, сервіс Amazon, що забезпечує безпечну потокову передачу відео з підключених систем а також також надійно зберігає, шифрує, індексує дані відеопотоків і дозволяє отримувати доступ до цих даних за допомогою простих у використанні API. Kinesis Video Streams дозволяє швидко створювати програми комп'ютерного зору та машинного навчання за допомогою інтеграції з Amazon Recognition Video. Для початку роботи достатньо встановити SDK Kinesis Video Streams, зареєструвати пристрій в консолі керування AWS (Amazon Web Services, «домівка» хмарних сервісів Amazon), з якого йтиме відео потік і почати потокову передачу відео в AWS для аналізу, обробки і зберігання. Вартість використ-

тання (за гігабайт даних): передача даних в KVS- $0,00850, отримання даних з KVS $0,00850, зберігання даних в KVS $0,02300.

Провівши аналіз можна стверджувати, що хмарні сервіси Amazon задовольняють вимоги до технологічних засобів для розробки системи і дозволяють виконати поставлену перед ПЗ вимогу щодо розпізнавання відео. Вони недорогі, ефективні, агностичні щодо джерел даних і надають зручний у користуванні інтерфейс.

### 2.3.5. Набір засобів розробки для контролю літаючого дрону

Основним компонентом, контролем якого повинна займатись розроблювана система відеоспостереження є безпілотний літаючий засіб – дрон. Вибір конкретної моделі виходить за рамки цього дослідження. Для виконання вимоги до ПЗ про контроль переміщення дрону за допомогою розроблюваної системи, відправку дрону команд та отримання відео потоку необхідно вибрати набір засобів розробки (SDK), які дозволять взаємодіяти з дроном через API. Для забезпечення гнучкості системи потрібно, аби SDK підтримувала дронів від різних виробників.

З ростом популярності дронів, як мобільної платформи відеоспостереження, розробники почали випускати свої SDK для ентузіастів, зацікавлених в розробці додатків які використовують БЛА. Головним недоліком більшості SDK є прив’язка до дронів конкретного виробника, що суперечить ідеї гнучкості розроблюваної системи. Прикладами таких платформ розробки є DJI SDK компанії DJI та Parrot Developer SDK від Parrot. Не дивлячись на їх вражаючий функціонал, що дозволяє керувати рухом апарату та його камерою, передавати відео потік в режимі реального часу та створювати сценарії роботи, вони нам не підходять. Також є інший тип середовищ, які потребують окремий контролер для встановлення, на якому вони розвертають власні інструменти взаємодії з дроном. Це, наприклад, FlytOS та OttoFly API. Вони також нам не підходять через вимогу, що розроблювана система повинна інтегруватися в існуючий та працюючий контролер розумного дому. В процесі

пошуку рішення, яке б задовольняло вимогам, була знайдена платформа Gobot – набір засобів розробки для робототехніки, фізичних обчислень та Інтернету речей (IoT), написаних на мові програмування Go.

Gobot забезпечує набір драйверів та адаптерів для управління різноманітними фізичними пристроями від низькорівневих Arduino та Raspberry Pi, до дронів, іграшок та інших пристроїв, що мають власне API.

"Робот" – основна абстракція програмного забезпечення, що полегшує створення цікавих функцій високого рівня для підтримуваних пристроїв. Роботи (програми Go), написані за допомогою Gobot, можуть запускатися на хост-машині, що з'єднується з підключеними пристроями, або безпосередньо на одноплатному комп'ютері з Linux або в будь-якому місці.

Gobot також забезпечує зовнішнє API, яке дозволяє іншим програмам керувати як окремими пристроями так і цілими групами роботів по спільній мережі, що реалізується за допомогою JSON поверх HTTP API. Підхід Gobot до стандартизації та абстракції полегшує написання програми, які працюють на багатьох апаратних платформах майже не потребуючи модифікацій.

Платформа Gobot задовольняє вимоги до ПЗ, вона підтримує велику кількість дронів, надає багаті можливості по їх контролю через зручне API, та легко інтегрується з вибраними засобами розробки. Gobot написаний мовою Go, що дозволить використовувати все багатство функцій, описаних в пункті 2.2.3, для побудови високоефективної системи взаємодії.

## 2.4. Інструментальні засоби розробки ПЗ

Важко переоціни значення інструментів, якими користується розробник в процесі створення програмних систем. Вміло налаштоване середовище засобів забезпечує високу ефективність, збільшує швидкість розробки і спрощує процеси відлагодження продукту.

### 2.4.1. Середовище розробки Goland

Goland – нова комерційна IDE від компанії JetBrains, задача якої - надати той же рівень зручності при програмуванні на Go, який PyCharm забезпечує для Python, а IntelliJ IDEA - для Java. Ось деякі із її можливостей:

– Гнучка системи збору проекту на основі технології gobuild, компіляції в один натиск під всі підтримувані архітектури та операційні системи.

– Інтеграція технології профілювання Pprof, що дозволяє знаходити вузькі місця проекту і оптимізовувати їх.

– Шаблони коду для допомоги в створені типових додатків.

– Підтримка популярних систем контролю версій, таких як Git, SVN, Mercurial.

– Підтримка великої кількості доповнень від IntelliJ та сторонніх розробників.

– Вбудований помічник для Google Cloud Platform, який полегшує інтеграцію Cloud Video Intelligence та App Engine.

Враховуючи вище описані особливості, а також ергономічний дизайн, статичний аналіз коду і розумне доповнення, можемо зробити висновок, що GoLand – оптимальний вибір для створення систем мовою Go.

## Висновки до розділу

В даному розділі ми визначили вимоги до розроблюваного системи відеоспостереження. На їх основі був складений список необхідних технологічних компонентів системи і поставлені вимоги до них. Шляхом порівняння існуючих рішень на відповідність критеріям, були обрані:

Наведемо типову будову середовища, за яким буде проводитися відеоспостереження

– контролер

– середовище інтеграції

- мова програмування
- протоколи взаємодії
- система розпізнавання об’єктів на відео
- SDK дрону

Далі ми визначили вимоги до інструментальних засобів та підібрали відповідні їм середовище розробки та систему контролю версій. Після формування вимог до архітектури було розглянуто мікросервісний та монолітний архітектурні підходи. Мікросервісний підхід краще задовольняє вимоги до архітектури та ПЗ, тому для проектування він був вибраний для проектування архітектури системи.

# РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ РОЗУМНОГО ДОМУ

## 3.1. Розробка архітектури програмного забезпечення системи відеоспостереження

Тепер, коли ми визначили функціонал ПЗ, обрали стиль організації додатку та знайшли задовольняючі вимоги технологічні засоби, можна приступити до проектування архітектури системи відеоспостереження.

Для цього скористаємось принципами побудови мікросервісної архітектури описаної в пункті 2.1., а саме «один сервіс – одна зона відповідальності» і «побудова сервісу навколо вимоги».

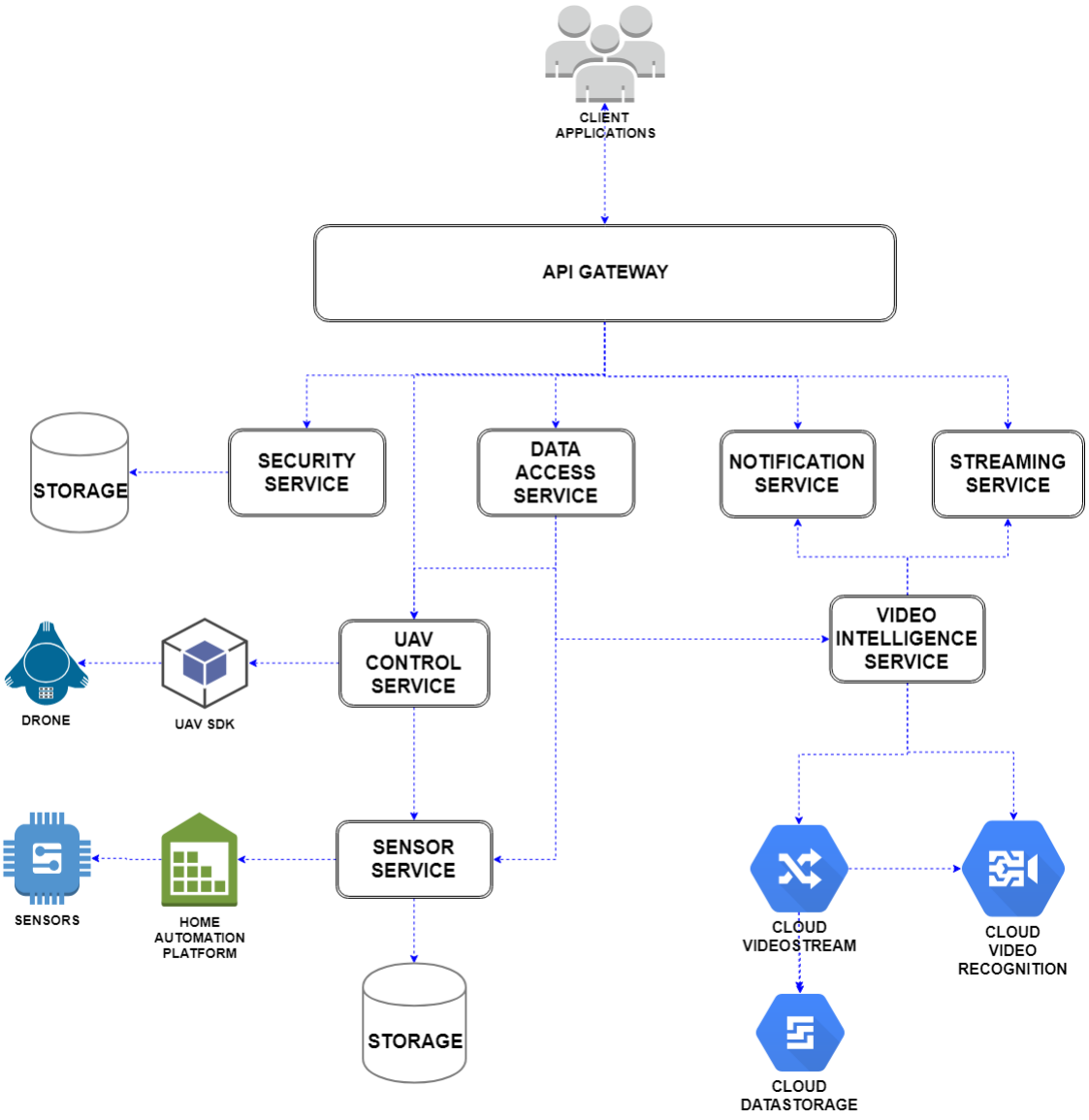


Рисунок 3.1 Структурна схема системи відеоспостереження

Структурна схема схеми відображає зв'язки між створюваними сервісами програмного забезпечення, платформою інтеграції компонентів розумного дому, сенсорами, середовищем керування БЛА, дроном, хмарними сервісами для роботи з відео і клієнтськими додатками.

Були розроблені такі сервіси:

– Сенсорний сервіс (SENSOR SERVICE)

Відповідає за розпізнавання відео потоків та їх збереження шляхом взаємодії із хмарними сервісами.

– Сервіс обробки відео (VIDEO INTELLIGENCE SERVICE)

Відповідає за розпізнавання відео потоків та їх збереження шляхом взаємодії із хмарними сервісами.

– Сервіс контролю БЛА (UAV CONTROL SERVICE)

Відповідає за взаємодію з дроном, його переміщення та навігацію в навколишньому середовищі, виконання користувацьких сценаріїв роботи.

– Сервіс потокової передачі відео (STREAMING SERVICE)

Реалізує потокову передачу відео на клієнтські додатки.

– Сервіс безпеки (SECURITY SERVICE)

Відповідає за реєстрацію та автентифікацію користувачі API на основі JSON Web Token.

– Сервіс сповіщень (NOTIFICATION SERVICE)

Відповідає за створення та відправку повідомлень щодо роботи сервісу клієнтам.

– Клієнтський шлюз (API GATEWAY)

Слугує єдиною точкою взаємодії системи і клієнтських додатків, формує запити до сервісів системи залежно від отриманих повідомлень.

– Сервіс доступу до даних (DATA ACCESS SERVICE)

Забезпечує доступ до даних пристроїв та знятих відео, що зберігаються в хмарі.

## 3.2. Опис роботи системи відеоспостереження

Система використовує декілька протоколів та форматів даних для реалізації всього різноманіття функціоналу: gRPC та Proto Buff для комунікації між власними сервісами, HTTP(S) та JSON для звернень до зовнішніх сервісів та клієнтів і RSTP для підтримки потокової передачі відео на хмарні сервіси та на клієнтські додатки.

Після встановлення програмного забезпечення на контролер, розгортання сервісів, встановлення зв’язків із пристроями дому та дроном, SENSOR SERVICE починає періодично опитувати сервіси щодо зміни їх стану. При отриманні сигналу від одного з них, SENSOR SERVICE формує повідомлення відповідного типу, зберігає його в сховищі і відправляє в чергу звернень до UAV CONTROL SERVICE, який відповідає за переміщення дрону до місця розташування пристрою, що дав сигнал. Звідти сервіс ініціалізує зйомку середовища і потоком відправляє відео до сервісу розпізнавання.

VIDEO INTELLIGENCE SERVICE реєструє новий відео потік на Amazon Kinesis Video Streams і одночасно ініціалізує його розпізнавання в Amazon Video Rekognition. На основі отриманих в якості відповіді описів відео, формується сповіщення для користувача, яке передається в NOTIFICATION SERVICE і ретранслюється потік відео до STREAMING SERVICE, який віддає його користувацькому додатку. Користувач може віддавати команди дрону і дому керуючись отриманими даними. Також у користувача є можливість створювати місії, послідовності дій і команд дрону, команди на виконання яких можна передавати системі відеоспостереження через API.

Блок схему алгоритму роботи системи відображено у додатку 1.

## 3.3. Сервіс роботи з пристроями розумного дому

openHAB надає зручне і просте API для взаємодії із сенсорами, підключеними до мережі, його основні функції:

Таблиця 3.1

Home Assistant REST API

| | |
|---|---|
| /rest/devices/ | Отримати список зі всіма зареєстрованими приладами |
| /rest/type/{type_id}devices/{device_id}/status | Отримати стан приладу |
| /rest/type/{type_id}devices/{device_id}/location | Отримати місцезнаходження приладу |

openHAB підтримує технологію Long Polling, яка дозволяє підтримувати з'єднання з пристроєм і отримувати повідомлення тільки тоді, коли стан пристрою змінився. При першому запуску системи сервіс отримує список усіх пристроїв і створює горутини для кожного з них, щоб обробляти з'єднання паралельно. Коли приходить сигнал типу []byte про зміну стану, він записується у спільний для всіх горутин канал з якого читає функція котра визначає тип сенсору що прислав оновлення стану і приводить масив байт до повідомлення відповідного типу, що відправляється до сервісу контролю дрону. Сервіси з'єднанні по gRPC і між ними налаштоване з'єднання для постійного стрімінгу повідомлень. Після цього повідомлення записується у БД, по композитному ключу з ідентифікатору девайсу і часу, коли прийшло оновлення. В БД також зберігається інформація про підключені сенсори

Пакет sensorDB реалізує методи CRUD для роботи з базою даних і надає інтерфейс для сервісу по збору даних. Для збереження данних використовується BoltDB – просте сховище типу ключ-значення, агностичне до типу хранимих даних. Основні методи *func (b *Bucket) Get(key []byte) []byte* та *func (b *Bucket) Put(key []byte, value []byte) error* використовують тип даних []byte, що дозволяє зберегти будь-що.

### 3.4. Сервіс роботи з пристроями розумного дому

При першому запуску, сервіс робить запит до сервісу роботи із пристроями дому для отримання масиву інформації про всі підключені сенсори і зберігає в пам’яті його ідентифікатор і, якщо сенсор обладнаний GPS модулем, інформацію про його розташування. Для всіх сенсорів, GPS позиція яких невідома, користувачу необхідно ввести її самостійно, що можна зробити просто відіславши з мобільного клієнтського додатку своє місцезнаходження перебуваючи біля сенсору на ендпоінт /surv/api/device/%device_id%, який опрацьовує сервіс сенсорів і відправляє запит на оновлення даних сервісу дрону. Також потрібно зареєструвати місце «проживання» дрону, точку, в яку він повертатиметься після успішного виконання завдань (зазвичай – хаб для зарядки на даху дому).

Планування шляху дрону

SDK керування дроном дає доволі прості засоби керування переміщенням.

Таблиця 3.2

Методи переміщення дрону в Gobot

| Сигнатура методу | Опис |
|---|---|
| func (a *Driver) Backward(speed int) | Рух назад із заданою швидкістю |
| func (a *Driver) Clockwise(speed int) | Обертання по часовій стрілці із заданою швидкістю |
| func (a *Driver) CClockwise(speed int) | Обертання проти часовій стрілці із заданою швидкістю |
| func (a *Driver) Down(speed int) | Рух вниз із заданою швидкістю |
| func (a *Driver) Forward(speed int) | Рух вперед із заданою швидкістю |
| func (a *Driver) Land() | Посадка |
| func (a *Driver) Left(speed int) | Рух вліво із заданою швидкістю |

Продовження табл.3.2

| Сигнатура методу | Опис |
|---|---|
| func (a *Driver) Right(speed int) | Рух вправо із заданою швидкістю |
| func (a *Driver) TakeOff() | Взліт |
| func (a *Driver) Up(speed int) | Рух вверх із заданою швидкістю |
| func (a *Driver) GPSPos() *gps.Position | Повертає поточне положення дрону по GPS |

Для реалізації навігації дрону, ці методи обгортаються логікою вищого рівня. Основним методом пакету *GPSNavigate* є *CreatePathPoints(from, to *gps.Position) []*Marks* який отримує на вхід 2 GPS місцезнаходження і повертає масив точок на кінцях відрізків, якими рухатиметься дрон. Інший метод *ExecutePath(*Marks) chan TrajectoryPoint* приймає цей масив і на його основі формує і виконує послідовності команд, необхідних для переміщення дрону по вказаних точках, повертає канал, куди відсилає точки траєкторії (позиція і швидкість) траєкторії дрону.

Користувач може створювати і викликати сценарії поведінки – послідовності точок для відвідування дроном і дії в цих точках (обертання, зйомка відео)

Таблиця 3.3

Методи зйомки відео дроном в Gobot

| Сигнатура методу | Опис |
|---|---|
| func (a *Driver) Video() chan []byte | Повертає канал, по якому йтиме потік відео |
| func (a *Driver) VideoEnable(enable bool) error | Повідомляє дрону почати/закінчити потокову зйомку відео |
| func (a *Driver) VideoStreamMode(mode int8) error | Вказує дрону, який режим використовувати для зйомки відео |
| func (a *Driver) StartRecording() error | Почати запис відео в пам’ять |

Продовження табл. 3.3

| Сигнатура методу | Опис |
|---|---|
| func (a *Driver) StopRecording() error | Закінчити запис відео в пам’ять |

Далі сервіс очікує повідомлень від сервісу сенсорів. Коли воно приходить, із пам’яті зчитуються данні про тип і розташування сенсору і починається процес польоту до отриманої позиції. По прильоті дрон починає знімати стан середовища і потоком передавати його у сервіс, звідки він відсилається далі, обгортаючись даними про тип сенсору, що передав сигнал, до сервісу розпізнавання поки дрон знаходитисься над джерелом виклику. Методи пов’язані зі зйомкою відео вказані в табл.3.3. Через деякий час або по настанні часу повернення дрон отримує команду від користувача і або переміщується на нове місце і продовжує знімати, або повертається у хаб.

## 3.5. Сервіс обробки відео

При запуску сервісу клієнт вводить свої данні для авторизації використання API Amazon Video Rekognition, Kinesis Stream та AWS Cloud.

Сервіс перебуває в режимі очікування до отримання стріму повідомлень від сервісу дрону.

Відео з дрону обробляється наступним чином: коли приходить перший фрейм потоку ми опрацьовуємо тип сенсору, щоб надалі формувати конкретні сповіщення для користувача. З отримануваного потоку нарізається по 2 кадри з секунди і формується масив, який передаватиметься в Amazon Rekognition, щоб зменшити об’єм даних і збільшити швидкість обробки.

Викликається ендпоінт хмарного сервісу для створення відеопотоків /kinesis/stream/create, якому вказується в якості джерела даних посилання на новий масив і у відповідь приходить підтвердження прийому і ідентифікатор нового стріму. Стрім записується на диск у файл, після закінчення обробки файл зберігається в хмарному сервісі і файл видаляється Після цього ми ро-

бимо запит до хмарного сервісу розпізнавання, передаючи йому в якості джерела даних ідентифікатор відеопотоку в Kinesis Stream.

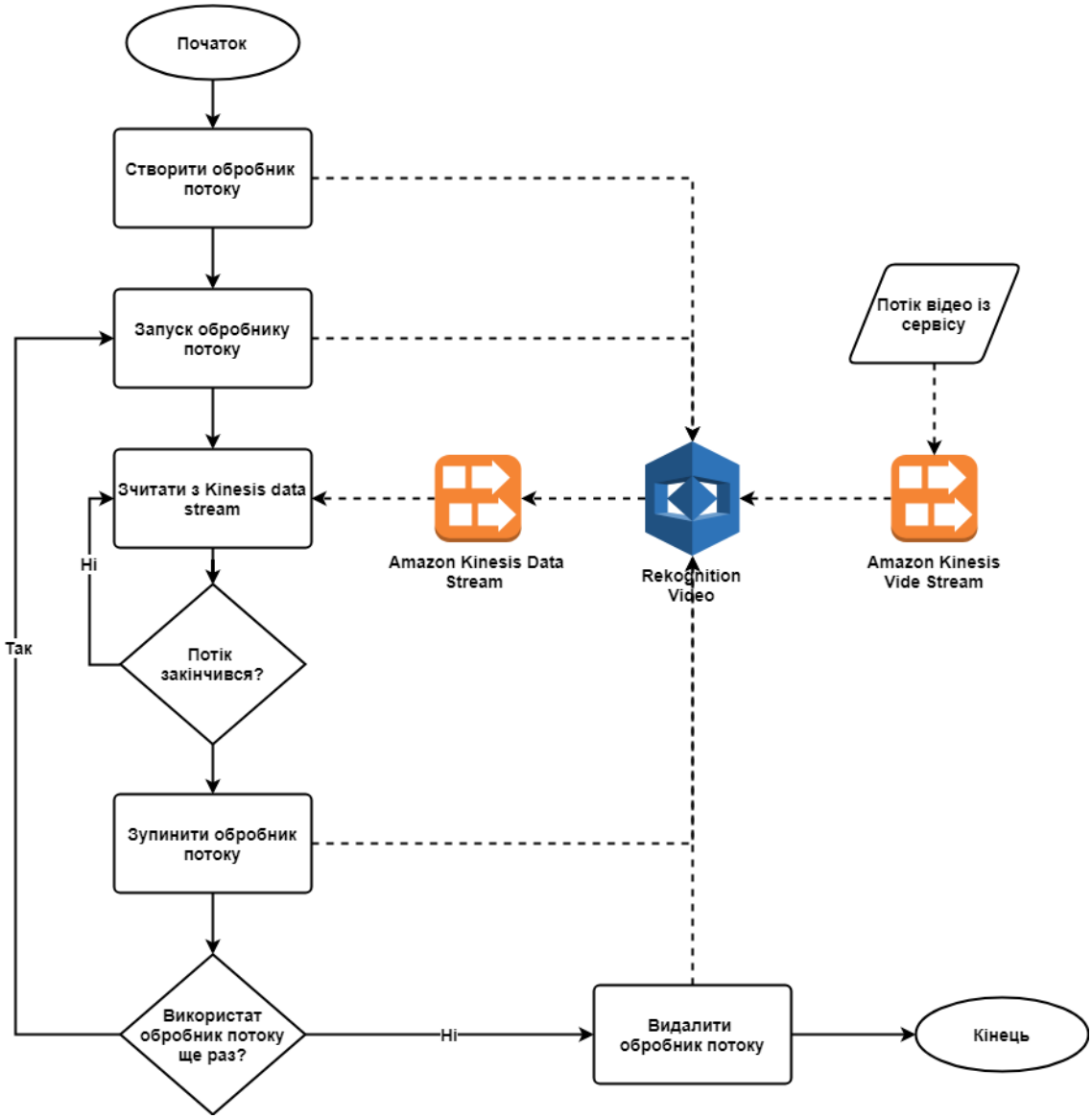


Рисунок 3.2 Алгоритм обробки відео потоку хмарними сервісами

Викликається ендпоінт хмарного сервісу для створення відеопотоків /kinesis/stream/create, якому вказується в якості джерела даних посилання на новий масив і у відповідь приходить підтвердження прийому і ідентифікатор нового стріму. Стрім записується на диск у файл, після закінчення обробки файл зберігається в хмарному сервісі і файл видаляється Після цього ми робимо запит до хмарного сервісу розпізнавання, передаючи йому в якості джерела даних ідентифікатор відеопотоку в Kinesis Stream. Сервіс розпізнавання починає передавати відповіді в форматі JSON, які містять списки категорій предметів. Метод *processVideoTags([]byte) json.RecognizedMessage* опрацьовує

відповідь від сервісу розпізнавання і формує повідомлення, залежно від об’єктів, які були знайдені у відео потоці. Повідомлення передаються у сервіси нотифікацій і стрімінгу на клієнтські додатку.

У табл.3.4. наведено запити, використовуючи які можна взаємодіяти з сервісом Amazon для збереження і отримання файлів. Кожен із запитів повинен містити HTTP заголовок Authorization: *authorization string* який отримується шляхом авторизації в сервісі. Він складається із алгоритму створення підпису, ключу доступу клієнта, дату, регіон та сервіс, який використовувався для створення підпису, списку заголовків, які використовувалися для обчислення підпису, і самого підпису.

Таблиця 3.4

Запити для взаємодії з хмарним сховищем S3

| Запит | Опис |
|---|---|
| GET / HTTP/1.1<br>Host: s3.amazonaws.com | Повертає JSON зі списком усіх сховищ створених користувачем |
| POST / HTTP/1.1<br>Host: *destinationBucket*.s3.amazonaws.com | Добавити новий об’єкт в сховище |
| GET /?list-type=2 HTTP/1.1<br>Host: *BucketName*.s3.amazonaws.com | Отримати список всіх об’єктів сховища |
| GET /*ObjectName*?torrent HTTP/1.1<br>Host: *BucketName*.s3.amazonaws.com | Отримати торент для завантаження об’єкта |
| GET /*ObjectName* HTTP/1.1<br>Host: *BucketName*.s3.amazonaws.com | Отримати поточну версію об’єкта |

## 3.6. Сервіс стрімінгу відео

Сервіс стрімінгу відео розділяється на дві логічні частини: веб сервер та медіа сервер. Спочатку, від сервісу пуш повідомлень на клієнтський додаток приходить сповіщення, що доступний стрім. Додаток відсилає HTTP запит на отримання відео потоку, він опрацьовується на стороні API GATEWAY. Першим кроком обробки є приведення отриманого запиту приводиться до формату запитів RTSP. З'єднання клієнта модифікується до

постійного, йому повертається ключ сесії, який буде відсилатися із кожним наступним запитом по контролю відеопотоку.

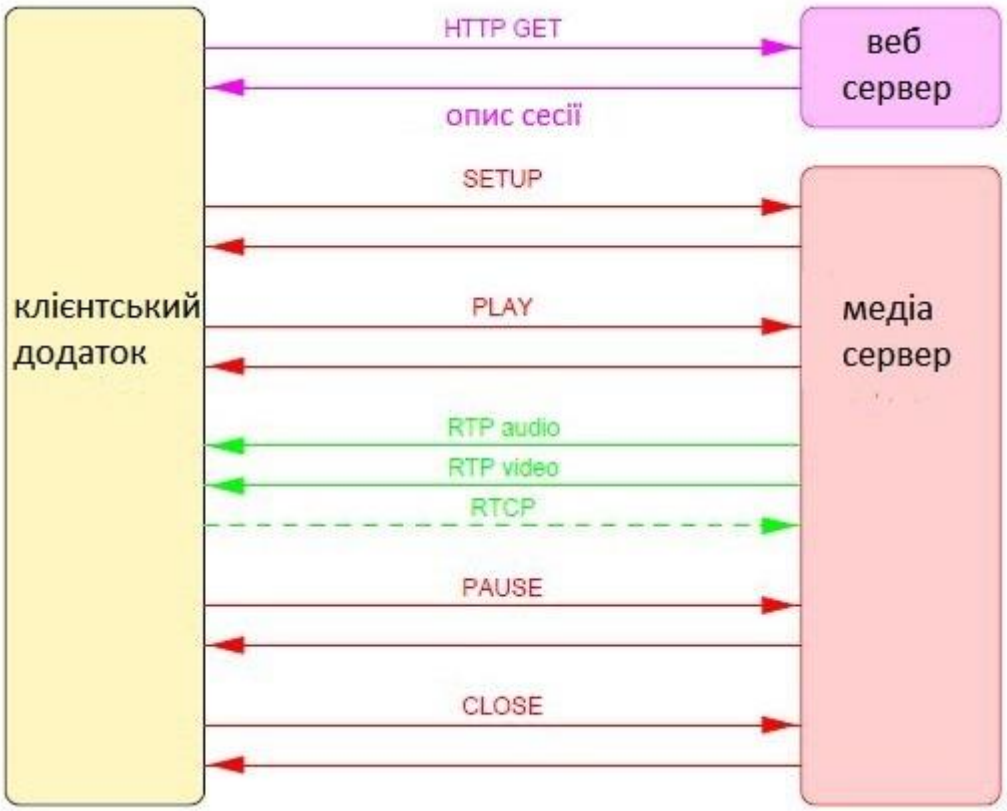


Рисунок 3.3 Процес взаємодії із RTSP сервером

Підключені клієнти, їх HTTP з'єднання та створені сесії зберігаються в пам'яті серверу.

Таблиця 3.5

Основні методи сервісу стрімінгу відео

| Сигнатура | Опис |
|---|---|
| func (s *RTSPServer) setupOurSocket(portNum int) (*net.TCPListener, error) | Запускає сервер на вказаному порті |
| func (s *RTSPServer) SetupTunnelingOverHTTP (httpPort int) bool | Створює обгортку для HTTP серве-ру, перенаправляючи запити |
| func newRTSPClientSession(connection *RTSPClientConnection, sessionID string) *RTSPClientSession | Створює сесію для нового клієнту |

Продовження табл. 3.5

| Сигнатура | Опис |
|---|---|
| func (s *RTSPClientSession) handleCommandPlay (subsession livemedia.IServerMediaSubsession, fullRequestStr string) | Опрацювання RTSP команди PLAY, запуск відео, яке отримує конкретна сесія |
| func (c *RTSPClientConnection) incomingRequestHandler() | Опрацювання нового запиту |

## 3.7. Сервіс сповіщень

Введемо позначення, пуш сервіс – сервіс, що реалізовується різними платформами для підтримки Web Push Protocol, сервіс сповіщень – один із сервісів системи відеоспостереження.

Для того, щоб клієнт почав отримувати пуш повідомлення, спочатку його треба «підписати» на наш сервіс. Підписка потребує двох речей: дозволу від користувача на отримання повідомлень і об’єкту PushSubscription який ми отримуємо від браузеру або пуш сервісу тієї платформи, на якій запущений клієнтський додаток. Варто зазначити, що деталі реалізації відрізняються залежно від типу платформи, будь то Android або iOS, але в описі сконцентруємось на загальному алгоритмі. PushSubscription містить усю інформацію необхідну для відправки повідомлень, ми зберігатимемо його в базі даних сервісу. Сервісу сповіщень необхідні унікальні VAPID ключі, щоб пуш сервіс браузеру/додатку міг встановити, хто відсилає сповіщення.

Щоб відправити сповіщення, потрібно робити API запити до пуш сервісу. Запит містить дані для відправки, ідентифікатор отримувача і будь-які критерії того, як відправляти повідомлення. Сервіс отримує запит, валідує його і доставляє повідомлення до клієнстького додатку. Кожен браузер і платформа пропонують свої сервіси для надислання сповіщень, але всі вони реалізують Web Push Protocol, тому приймають запити в одному форматуі, що дозволяє нам не хвилюватися щодо деталей реалізації кожного з них, головне – забезпечити коректність запиту. Отриманий раніше об’єкт PushSubscription

містить унікальну адресу, яку сервіс генерує для конкретної пари сервер-клієнт.

Контент повідомлення потрібно зашифрувати, щоб прибрати можливість його прослуховування пуш сервісами, для цього використовується алгоритм AES із довжиною блоку 128 біт. Для повідомлення можна вказати час життя, після якого воно вважається неактуальним і видаляється пуш сервісом, важливість повідомленні і тему. Коли повідомлення відправлено в пуш сервіс, воно стає в чергу і доходить до користувача коли у його додатку з'являється доступ до мережі.

Таблиця 3.6

Основні методи сервісу сповіщень

| Сигнатура | Опис |
|---|---|
| func GenerateVAPIDKeys() (privateKey, publicKey string, err error) | Генерація унікальних для сервісу ключів |
| func SendNotification(message []byte, s *Subscription, options *Options) (*http.Response, error) | Відправка повідомлення |
| func Encrypt(sub *Subscription, message string, encoding ContentEncoding) (*EncryptionResult, error) | Шифрування змісту повідомлення |
| func SubscriptionFromJSON(b []byte) (*Subscription, error) | Створення підписки із запиту користувача |

## 3.8. Сервіс доступу до даних

Метою створення сервісу є необхідність розділення логіки отримання користувачем даних, які не необхідні для функціонування користувацького додатку і логіки роботи системи, що робить отримання незалежним від деталей реалізації окремих компонентів системи і їх сховищ. Він є єдиною точкою системи, яка обробляє запити до даних і отримує їх із сервісів, які зберігають ці дані у своїх сховищах. Для кожного сервісу описується інтерфейс зв'зяків і типи повідомлень у форматі Proto Buff, які передаватимуться по gRPC

Таблиця 3.7

Основні методи сервісу доступу до даних

| Сигнатура | Опис |
|---|---|
| getSensorDataByType(sensorType string, dateFrom int) | Отримати показники всіх сенсорів певного типу з деякого часу. |
| getDroneFootage(date int) | Отримати всі відео зняті дроном в певний день |
| getNotificationByTopic(topic string) | Отримати всі сповіщення певної теми |
| getNotificationByUrgency(topic string) | Отримати всі сповіщення по типу важливості |

## 3.9. Сервіс безпеки

В монолітному додатку звичайною практикою є використання станів системи, тому використовується авторизація на основі сесій. Та у випадку з мікросервісами такий підхід не працює, бо нам потрібно перенаправляти запити до багатьох незалежних сервісів. Для забезпечення відсутності стану системи варто використати автентифікацію на основі токенів. Вимоги користувача запаковуються в jwt. JSON Web Token (jwt) це відкритий стандартний RFC 7519 метод для безпечного представлення вимог між сторонами спілкування.

Перший користувач (власник системи) отримує акаунт відразу і змінює стандартний пароль на унікальний. Запити інших користувачів на реєстрацію в системі (це можуть бути правоохоронні органи, служба пожежної безпеки, тощо) повинен погодити власник системи.

Коли користувач успішно логіниться до системи, йому повертається jwt. На кожен наступний запит користувач повинен включати цей токен до в HTTP заголовку. Це необхідно, щоб сервіс авторизації міг встановити ідентичність користувача та отримати його вимоги користувача з токену.

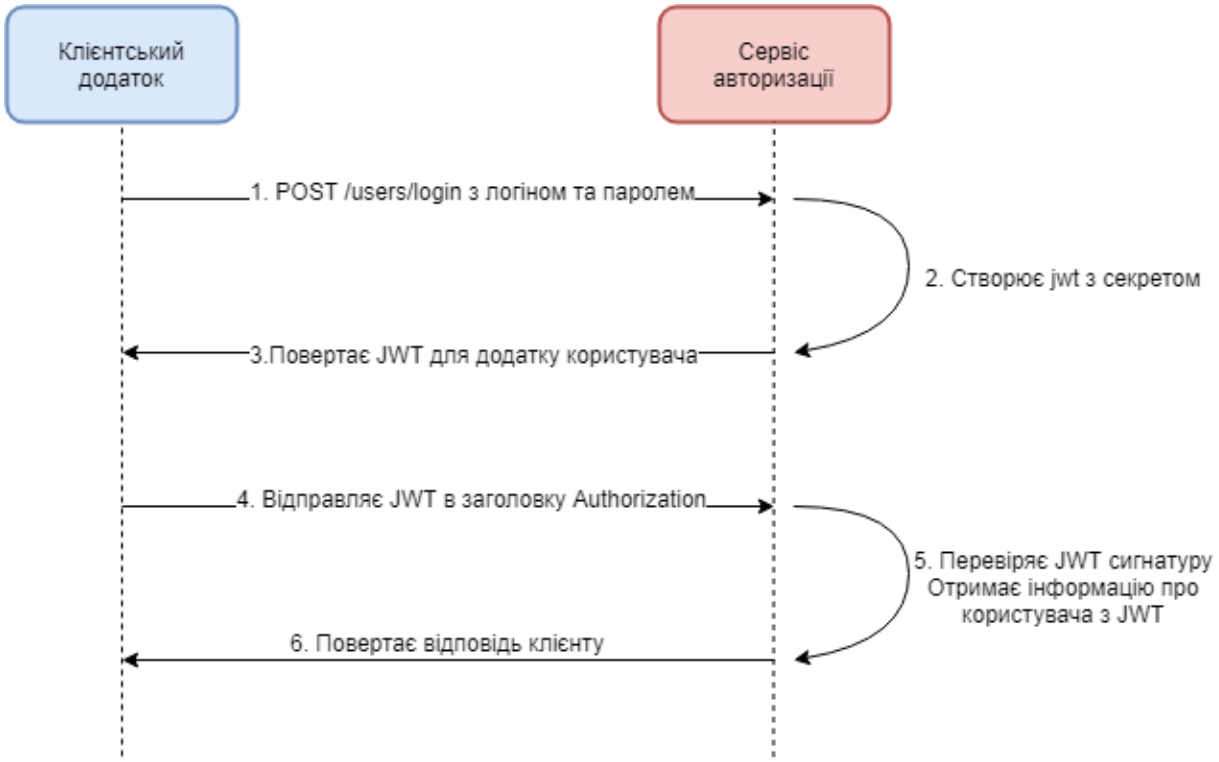


Рисунок .3.5 Процес авторизації

Авторизація в системі базується на дозволах, закодованих в jwt. Дозволи використовуються для обмеження доступу до API. В якості БД сервісу авторизації використовуються SQLite, на рисунку 3.6. зображена її ER діаграма. Коли користувач робить запит до API GATEWAY на виконання будь-якої функції сервісу, запит проходить до сервісу безпеки, де із нього отримується jwt (з кукі або заголовку), перевіряється його коректність і отримуються вимоги на доступ. Після цього, викликається метод Authorize пакету авторизації, якому передаються дозволи, url та http слово викликуваного ендпоінту сервісу. Він повертає true якщо якийсь із дозволів користувача має доступ до ендпоінту.

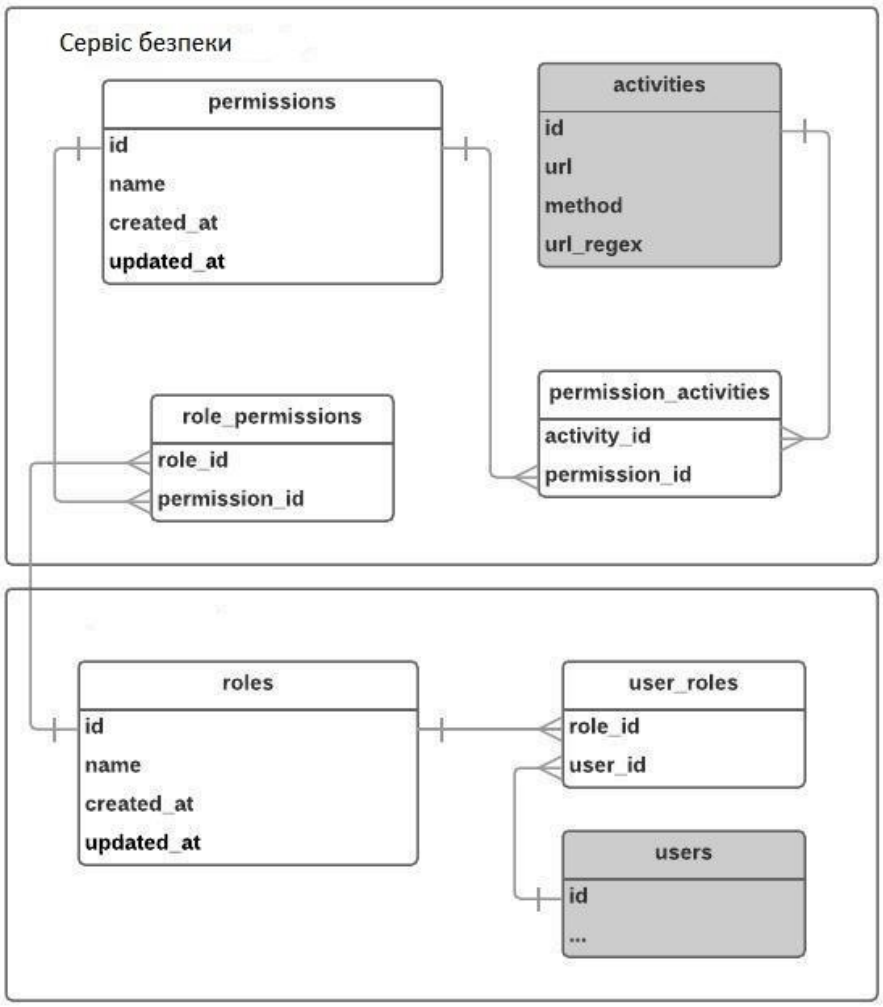


Рисунок 3.6 ER діаграма сервісу безпеки

Висновки до розділу

В даному розділі була створена структурна схема системи відеоспостереження навколишнього середовища розумного дому. Був даний загальний опис алгоритму роботи програмного забезпечення і розглянуті особливості реалізації кожного сервісу.

# РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності [9].

Дипломний проект на тему «Мобільна система відеоспостереження стану та об’єктів навколишнього середовища розумного дому» пов’язаний з комп’ютерним моделюванням та розробкою програмного забезпечення.

В даному розділі описано робоче місце розробника, що розроблятиме програмне забезпечення системи відеоспостереження з урахуванням необхідних показників і встановлених норм. Працюючи за комп'ютером, людина потрапляє під вплив різноманітних факторів: електромагнітних полів (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоного та іонізуючого випромінювання, шуму і вібрацій, статичної електрики. Розділ включає аналіз мікроклімату, освітлення, опис інструкції пожежної безпеки приміщення, в якому буде відбуватися робота адміністратора.

## 4.1. Характеристика робочого місця

Робоче місце розробника знаходиться в одній із комп’ютерних лабораторій, яка обладнана для роботи одного працівника. Лінійні розміри становлять 3,5 м × 2,5 м, висота стелі 2,6 м. (табл. 4.1). Площа та об’єм приміщення зазначені в таблиці 4.2.

Таблиця 4.1

Розміри приміщення

| | |
|---|---|
| Довжина, м (L) | 3,5 |
| Ширина, м (D) | 2,5 |
| Висота, м (H) | 2,6 |

Таблиця 4.2

Площа та об’єм приміщення

| Геометрична характеристика | Одиниця виміру | Нормативне значення | Фактичне значення |
|---|---|---|---|
| Площа, S | $м^2$ | не менш 6.0 | 8,75 |
| Об’єм, V | $м^3$ | не менш 20 | 22,75 |

За даними, що наведені вище у таблиці 4.2, можна зробити висновок, що геометричні розміри приміщення відповідають правилам [10].

З меблів в лабораторії знаходиться один кутовий стіл, одне крісло, дві шафи з документами та прямокутний стіл для розташування іншої техніки. З техніки наявні один персональний комп’ютер, принтер та телефон.

План приміщення зображено на рисунку 4.1. На ньому цифрами позначено наявні елементи: 1 – кутовий стіл; 2 – прямокутний стіл; 3 – крісло; 4,5 – шафи; 6 – монітор; 7 – телефон; 8 – принтер.

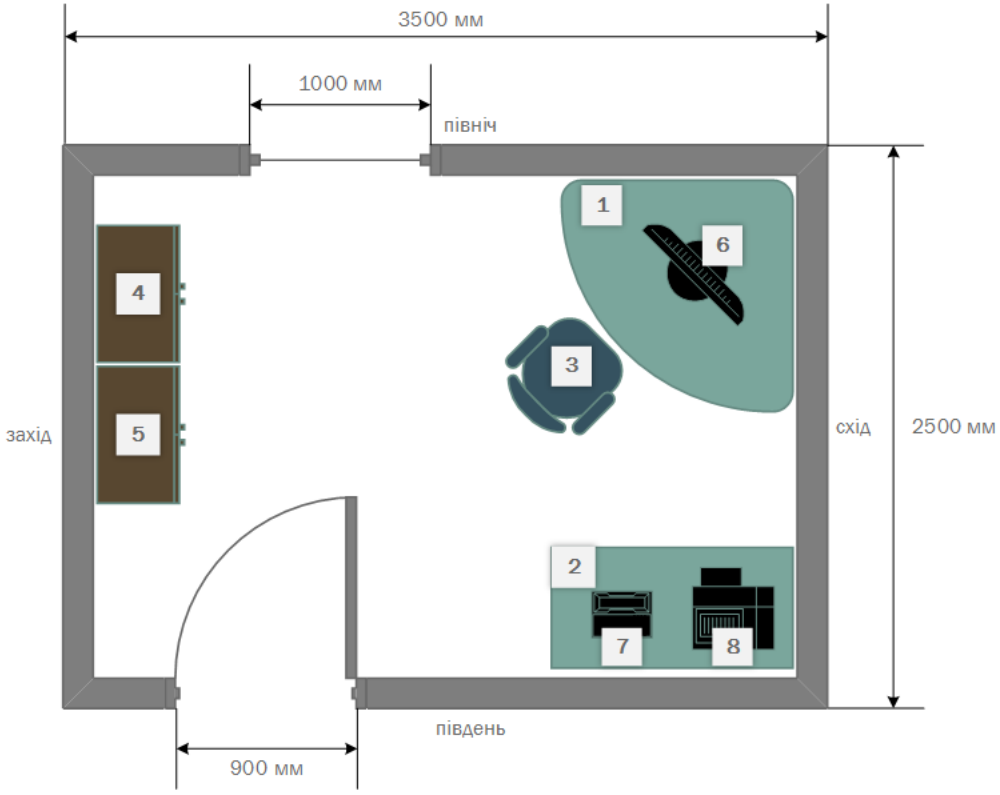


Рисунок 4.1 Спрощений план приміщення

## 4.2. Аналіз і оцінка шкідливих виробничих факторів

Розглянемо робоче місце користувача ПК з точки зору оцінки впливу шкідливих виробничих факторів відповідно до гігієнічної класифікації праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу [11]. Відповідно до цього документу, на працівника, який працює з комп'ютером діють такі шкідливі виробничі чинники: мікроклімат робочої зони; шум; випромінювання; недостатність штучного освітлення та пожежонебезпека.

### 4.2.1. Мікроклімат робочої зони

Оскільки робота виконується сидячи та без фізичного навантаження, то її категорія оцінюється як «легка-1а». Відповідно до встановлених вимог [12] роботодавець зобов'язаний забезпечити в приміщеннях для даного типу роботи оптимальні параметри виробничого середовища (таблиця 4.3).

Причиною підвищеної температури робочої зони можуть бути освітлювальні пристрої, величина тепловиділення яких становить 35-60 Вт/м$^2$, а також комп'ютер, середня величина тепловиділення якого становить 310 Вт/м$^2$.

Таблиця 4.3

Норми мікроклімату для приміщень з ЕОМ

| **Пора року** | **Параметр мікроклімату** | **Оптимальне значення** | **Фактичне значення** |
|---|---|---|---|
| Холодна | Температура повітря | 22-24°С | 23°С |
| | Відносна вологість повітря | 60 - 40% | 31% |
| | Швидкість руху повітря | 0,1м/с | 0,1м/с |
| Тепла | Температура повітря | 23 - 25°С | 24,5°С |
| | Відносна вологість повітря | 60 - 40% | 58% |
| | Швидкість руху повітря | 0,1 м/с | 0,1м/с |

4.2.2. Шум

На комп'ютеризованих робочих місцях основними джерелами шуму є вентилятори системного блоку та принтери. Сильний шум викликає труднощі з розпізнаванням колірних сигналів, знижує швидкість сприйняття кольорів, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5-12% продуктивність праці.

На даному робочому місці основними джерелами шуму є вентилятори системи охолодження системного блоку комп'ютера, а також принтери та телефон. Також варто врахувати шум, що надходить ззовні, і який ліквідується використанням акустичних поглиначів звуку, а також вікон, що щільно закриваються.

Для покращення робочої обстановки необхідне технічне вдосконалення та періодичне обслуговування системних систем охолодження комп'ютерів. А принтери перемістити за межі лабораторії, або помістити в звукоізоляційну коробку.

Розрахуємо рівень шуму в приміщені. Величина рівня шуму типових джерел надано в таблиці 4.4.

Таблиця 4.4

Значення рівня шуму для типових джерел

| **Джерело шуму** | **Рівень шуму, дБА** |
|---|---|
| Жорсткий диск | 30 |
| Вентилятор | 45 |
| Монітор | 15 |
| Клавіатура | 8 |
| Принтер | 40 |
| Телефон | 45 |

Максимальний час роботи принтера за один день – 1,5 години.

Робочий день Т = 8 годин. Розрахуємо фактичний рівень шуму за наступною формулою.

$$L_{\Sigma} = 10\lg(10^3 + 10^{4,5} + 10^{1,7} + 10^{0,8} + 10^4 + 10^4) = 48{,}7 \text{ дБА}$$

Під час роботи за комп'ютером рівень шуму відповідно до норм [13] не повинен перевищувати 50 дБА, а фактичний становить 48,7 дБА. Отже, наше приміщення відповідає діючим санітарним нормам.

### 4.2.3. Випромінювання

Дисплей ПК є джерелом сильного електромагнітного випромінювання. Крім електромагнітних полів та випромінювання безпосередньо від монітора, на користувача додатково впливають так звані фонові поля – поля від сторонніх джерел, які знаходяться у приміщенні або поблизу від нього. Такими джерелами є мережі живлення і освітлення, побутові прилади (кондиціонер, обігрівач), мобільні телефони, бездротова мережа тощо.

Відповідно до [10], припустима інтенсивність потоку енергії 10 Вт/м$^2$, а напруженість електричного поля в електричній складовій на відстані 0,5 м. від екрану – 10 В/м. В сучасних LCD дисплеїв інтенсивність потоку складає не більше 1 Вт/м$^2$, а значить умови відповідають встановленим нормам.

### 4.2.4. Освітлення

Для запобігання прямого відблиску світла дисплеї розміщуються боком до вікна. Під час роботи в приміщенні, близько 75% всього часу погляд працівника спрямований в напрямку робочої поверхні, тобто в напрямку дисплея. Розряд робіт – високої точності (через відносно невеликі розміри знаків на дисплеї), а фон – світлий.

В приміщенні використовується бокова система природного освітлення та загальна система штучного освітлення. Через це нормоване значення освітлення повинно бути 300 лк, а КПО становить 1.5. Коефіцієнт пульсації не перевищує 5%, що задовольняє вимогам [14].

Фактична площа вікон 1,5 м$^2$, що не відповідає чинним вимогам, тому в приміщенні застосовується освітлення з допомогою ламп денного світла.

Для освітлення приміщення використаний світильник з дзеркальними параболічними решітками, укомплектований електронним пускорегулюючим апаратом (ЕПРА). Світильник з ЕПРА має дві випереджальні та дві відстаючі гілки. Світильник розташовано над робочим місцем адміністратора (рисунок 4.2), а його характеристики такого світильника наведено в таблиці 4.5.

Таблиця 4.5

Характеристики світильника

| **Назва** | **Потуж-ність, Вт** | **Струм, А** | **Напруга, В** | **Габаритні розміри, мм** | **Світловий потік, лм** | **Термін служби, годин** |
|---|---|---|---|---|---|---|
| Світиль-ник LM160 | 60 | ,73 | 20 | 38×1514,2 ×1500 | 6 000 | 1 0000 |

## 4.3. Електробезпека

Трансформатор від якого живиться приміщення розташований поза ним, напруга на його первинній обмотці: 6.3 кВ, на вторинній: 380 В. Режим нейтралі відносно землі – заземлений, режим електричної мережі – однофазний.

Електрична проводка виконана: мідним проводом в поліхлорвініловій ізоляції, площа поперечного розрізу 4 мм$^2$.

Живлення електрообладнання здійснюється напругою 220В.

Підвищена електрична небезпека наявна в приміщенні, якщо воно відповідає таким критеріям:

- відносна вологість повітря більше 75%;
- наявність струмопровідного пилу;

- наявність агресивного хімічного чи біологічного середовища, яке може стати причиною руйнації ізоляції;
- температура повітря більше 35°С;
- наявна можливість торкання до заземлених металевих конструкцій і струмопровідних частин водночас;
- наявність струмопровідної підлоги.

Оскільки, дане приміщення не відповідає наведеним вище критеріям, то воно не відноситься до групи із підвищеною електричною небезпекою.

Електробезпека приміщення забезпечується технічними способами і засобами захисту, а також організаційними заходами.

Всі елементи електроприладів й устаткування виконані відповідно до умов техніки електробезпеки, мають необхідне ізоляційне покриття (подвійна ізоляція) і властивості, що виключає можливість ураження електричним струмом при підключенні й експлуатації устаткування. Розетки змонтовані на негорючих пластинах і мають сучасну триконтактну конструкцію, захист виконаний у вигляді бічних контактів, які взаємодіють першими, при включені вилки в розетку й відключаються останніми при витягуванні вилки з розетки, що відповідає умовам [11].

Зважаючи на все вищевказане, за класом небезпечності приміщення відноситься до приміщень без підвищеної небезпечності.

### 4.4. Пожежна безпека

В приміщенні знаходяться пожежонебезпечні матеріали: папір (документація) та дерево (столи). Вибухонебезпечні матеріали у приміщенні відсутні.

Виходячи з властивостей та кількості пожежо- та вибухонебезпечних матеріалів за нормативною документацією [15] приміщення має категорію В.

Можливі причини виникнення пожежі – короткі замикання у комп’ютерах, підвищення температури в приміщенні або поява розжарених матеріалів.

Біля дверей розміщено план евакуації з даного приміщення на вулицю.

4.4.1. Засоби пожежогасіння

Приміщення обладнане двома вогнегасниками: ВВ-3,5 (кожен містить 3,5 кг вогнегасної речовини).

Відстань між місцями розташування вогнегасників не повинна перевищувати 15 м. В приміщенні знаходяться два вогнегасники (для зручності позначені індексами на схемі). Вогнегасники знаходяться на відстані один від одного, що відповідає нормам (відстань між першим і другим вогнегасником становить 3,22 метра).

4.4.2. Пожежна сигналізація

Згідно з додатком К до норм [16], в приміщенні встановленні димові датчики автоматичного знаходження пожежі ІПК-2. Датчики під’єднанні до центрального пульту охорони та безпеки.

4.5. Інструкція з техніки безпеки

Під час роботи, пересвідчившись у справності обладнання, увімкнути електроживлення ПК, розпочати роботу, дотримуючись умов інструкції з його експлуатації.

Забороняється:

- замінювати змінні елементи або вузли та проводити перемонтаж при ввімкненому ПК;
- з'єднувати і роз'єднувати вилки та розетки первинних мереж електроживлення, які знаходяться під напругою;
- знімати кришки, які закривають доступ до струмопровідних частин мережі первинного електроживлення при ввімкненому обладнанні;
- користуватися паяльником з незаземленим корпусом;
- замінювати запобіжники під напругою;
- залишати ПК у ввімкненому стані без нагляду.

Після закінчення роботи на ПЕОМ працівник повинен дотримуватись такої послідовності вимикання обладнання:

- здійснити закриття всіх активних завдань;
- вимкнути живлення системного блоку;
- вимкнути живлення всіх периферійних пристроїв;
- штепсельні вилки витягнути з розеток;
- накрити клавіатуру кришкою.
- про всі недоліки, що виявились у процесі роботи повідомити керівника робіт.

Висновки до розділу

У результаті проведеного аналізу умов безпеки праці на робочому місці працівника були виявлені шкідливі і небезпечні фактори, а також визначені та запропоновані варіанти вирішення його недоліків.

Було проведено розрахунок рівня шуму в приміщені та було встановлено, що він задовольняє нормам.

Окрім цього були розглянуті інструкції з охорони праці, питання пожежної безпеки та визначено необхідні умови для її забезпечення.

В результаті недотримання умов безпечної праці на робочу місці, у працівників спостерігається незадоволеність роботою, головний біль, роздратування, порушення сну, втома і больові відчуття в очах, попереку, у ділянці шиї та рук.

# ВИСНОВКИ

В ході виконання даного проекту проаналізовано поточний стан технологій створення розумних будівель та проблеми забезпечення відеоспостереження їх навколишнього середовища. Розглянуті технології створення системи відеоспостереження стану та об’єктів навколишнього середовища розумного дому і проблема інтеграції мобільної платформи в існуючу систему автоматизації будівлі. Розроблено архітектуру програмного забезпечення та описано її складові сервіси.

# ПЕРЕЛІК ПОСИЛАНЬ


1. Ohtmar K. How To Smart Home / Ohtmar K., Jacobson B. – Key Concept Press, 2017. – с. 22

2. Злочинність в Україні [Електронний ресурс] : інтернет-ресурс Слово і діло. – Режим доступу до ресурсу: https://ru.slovoidilo.ua/2018/02/16/infografika/obshhestvo/prestupnost-ukraine-statistika-proshlyj-god

3. Home Alarm Industry Statistics [Електронний ресурс] : інтернет-ресурс ProtectAmerica. – Режим доступу до ресурсу: https://www.protectamerica.com/home-security-blog/just-for-fun/home-alarm-industry-statistics-just-how-big-is-the-industry_16263

4. Dornberg W. Surveillance in history / Dornberg W. – Balantine Books, 2016 – с. 16

5. Enshani R. The rise of small UAVs in precision agriculture. / Enshani R., Maja JM. – Engineering and Technology for Sustainable World, 2018 – с.18-19

6. Microservices [Електронний ресурс] : інтернет-ресурс marting-fowler. – Режим доступу до ресурсу: https://www.martinfowler.com/articles/microservices.html

7. Go at Google: Language Design in the Service of Software Engineering [Електронний ресурс] : інтернет-ресурс talks.golang. – Режим доступу до ресурс – https://talks.golang.org/2012/splash.article

8. Public cloud application services/software as a service (SaaS) market revenues worldwide from 2015 to 2021 [Електронний ресурс] : інтернет-ресурс Statista. – Режим доступу до ресурсу – https://www.statista.com/statistics/505243/worldwide-software-as-a-service-revenue

9. Закон України про охорону праці : за станом на 09 липня 2010 року. – Офіц. вид. – К. : Парламентське вид-во, 2010. - 28 с. – (Закони України).

10. НПАОП 0.00-1.28-10. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин – Київ, 1999. – 18с.

11. ДНАОП 0.00.-1.31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин – Київ, 1999. – 30с.

12. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. – Київ, 2000

13. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. -Київ, 2000

14. ДБН В.2.5-28-2006. Державні будівельні норми України на природне і штучне освітлення. – Київ, 2006

15. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. – Київ, 2007

16. ДБН В.2.5-13-98 Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд. – Київ, 2006