МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____Олександр ЛИТВИНЕНКО

«___» _____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль моніторингу доступу до закритих корпоративних систем

Виконавець: Роман КУБРАК

Керівник: Олена НЕЧИПОРУК

Нормоконтролер: Євгеній ТУПОТА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Олександр ЛИТВИНЕНКО

«_____»_____2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Кубрак Роман Сергійович

1.Тема роботи: «Програмний модуль моніторингу доступу до закритих корпоративних систем»
затверджена наказом ректора від «28» серпня 2023 року № 1494 /ст.

2. Термін виконання роботи: з 02.10.2023 до 31.12.2023

3. Вихідні дані до роботи: 1) вимоги до змісту програмного модуля;

2) функції моніторингу системи контролю управління доступом в банку.

4.Зміст пояснювальної записки (перелік питань, що підлягають розробці):
1) аналіз принципів роботи систем контролю і управління доступом;
2) принципи роботи систем відеоспостереження;
3) описання програмного модуля моніторингу відеоспостереження в системі контролю управління доступом.

5.Перелік обов'язкового графічного матеріалу:
1) діаграма класів збереження об'єктів в програмному модулі;
2) діаграма класів команд програмного модуля;
3) основні елементи пакету системи зв'язку з сервером;
4) схема алгоритму збереження відеофайлів з камер спостереження.

6. Календарний план-графік

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1	Провести аналіз літератури за темою дипломного проєкту та аналіз існуючих систем	02.10.2023-12.10.2023	
2	Зробити вибір компонентів системи	13.10.2023-29.10.2023	
3	Розробити структуру програмних засобів системи цифрового репозитарію	30.10.2023-06.11.2023	
4	Розробити програмні засоби цифрового репозитарію	07.10.2023-14.11.2023	
5	Провести налаштування програмних засобів на сервері	15.10.2023-26.11.2023	
6	Написати пояснювальну записку	27.10.2023-11.12.2023	
7	Підготувати презентацію	12.12.2023-18.12.2023	
8	Оформити супрводжувальну документацію	19.12.2023-23.12.2023	

7. Дата видачі завдання «02» жовтня 2023 р.

Керівник дипломного проєкту_____ Олена НЕЧИПОРУК
(підпис)

Завдання прийняв до виконання_____ Роман КУБРАК
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи“ Програмний модуль моніторингу доступу до закритих корпоративних систем”: 81 с., 28 рис., 28 літературних джерел, 1 додаток.

ВІДЕОСПОСТЕРЕЖЕННЯ, НЕСАНКЦІОНОВАНИЙ ДОСТУП, ОХОРОННА СИСТЕМА, КОМПЛЕКСНА БЕЗПЕКА, КЛАСИ ЗАХИСТУ ОБ’ЄКТІВ, ОБСЛУГОВУВАННЯ ОХОРОННИХ СИСТЕМ, КОМПОНЕНТИ ОХОРОННИХ СИСТЕМ

Об’єкт дослідження – автоматизація процесу моніторингу роботи системи контролю управління доступом в банку.

Предмет дослідження – програмний модуль моніторингу роботи системи контролю управління доступом в банку.

Метою роботи є розробка програмного модуля для забезпечення моніторингу роботи системи контролю управління доступом в банку.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ 7
ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ ПРИНЦИПІВ РОБОТИ СИСТЕМ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ 10
1.1. Проблеми інформаційної безпеки підприємств та організацій 10
1.2. Сучасні технології організації безпечного доступу на об’єкт 12
1.3. Огляд існуючих систем контролю доступу на об’єкт 16
1.4. Принцип роботи мережевої системи контролю доступу 20
1.5. Висновки до розділу 24
РОЗДІЛ 2 ПРИНЦИПИ РОБОТИ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ 26
2.1. Структура і основні елементи систем відеоспостереження 26
2.2. Області застосування і огляд програмного забезпечення систем відеоспостереження 29
2.3. Інтегровані системи безпеки 31
2.4. Можливості використання відеокамер в сучасних біометричних методах ідентифікації 34
2.5. Висновки до розділу 35
РОЗДІЛ 3 ОПИСАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОДУЛЯ 38
3.1. Описання існуючої СКУД 38
3.2. Архітектура програмного забезпечення інтеграції з СКУД 48
3.3. Збереження об’єктів 51
3.4. Параметризоване створення об’єктів 55
3.5. Взаємодія додатків 58
3.6. Висновки до розділу 73
ВИСНОВКИ 75
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ 79
ДОДАТОК А 80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

BS – британський стандарт

EN – європейська норма

FMR – коефіцієнт помилкової відповідності

FRR – відсоток помилкових відмов

ID – ідентифікація

PAS – загальнодоступна специфікація

REX – запит на вихід

RFID – радіочастотна ідентифікація

SLA – угода про рівень обслуговування

T&A – час і відвідуваність

UID – унікальна ідентифікація

ППП – пакет прикладних програм

СКУД – система керування управлінням доступом

ВСТУП

Сучасний світ характеризується стрімким та безперервним розвитком інформаційних технологій та комп'ютерних систем. Велика кількість інформації, яка зберігається в електронному вигляді, а також зростаюча кількість користувачів, що мають доступ до цієї інформації, роблять проблему забезпечення безпеки та моніторингу доступу до корпоративних систем надзвичайно актуальною. Необхідність забезпечення конфіденційності, цілісності та доступності інформації стала ключовою у сфері інформаційної безпеки.

Проблема безпеки доступу до корпоративних систем є надзвичайно актуальною в сучасному бізнес-середовищі. Зростаюча кількість інцидентів з порушеннями безпеки, які призводять до витоку конфіденційної інформації, втрати фінансових ресурсів та пошкодження репутації компаній, свідчить про важливість цієї проблеми. Власники корпорацій та керівники підприємств прагнуть забезпечити безпеку своєї інформації та обмежити доступ до неї лише авторизованим користувачам. Актуальність проблеми полягає в постійній зміні загроз для інформаційної безпеки та необхідності розвивати нові методи та підходи до моніторингу та управління доступом.

Ця магістерська робота є дослідженням у галузі розробки систем моніторингу доступу до закритих корпоративних систем. Вона базується на наукових підходах та методах, спрямованих на розробку і вдосконалення інструментів та технологій, призначених для забезпечення безпеки та контролю доступу до корпоративних ресурсів. Дослідження охоплює важливі аспекти, такі як аутентифікація, авторизація, аудит та моніторинг подій, що стосуються доступу користувачів до інформаційних ресурсів компанії.

Об’єкт дослідження – автоматизація процесу моніторингу роботи системи контролю управління доступом в банку.

Предмет дослідження – програмний модуль моніторингу роботи системи контролю управління доступом в банку.

Метою роботи є розробка програмного модуля для забезпечення моніторингу роботи системи контролю управління доступом в банку.

Завданнями дослідження є:

1. Аналіз існуючих підходів до моніторингу та управління доступом. Розгляд існуючих рішень та їхніх переваг та недоліків.

2. Розробка нових методів та підходів до моніторингу доступу. Розробка інноваційних технологій та методів для підвищення безпеки та контролю доступу до корпоративних ресурсів.

3. Апробація розроблених методів на практиці. Проведення експериментальних досліджень та тестування розроблених інструментів у реальних умовах.

4. Оцінка результатів та розробка рекомендацій. Аналіз результатів експериментів та розробка рекомендацій щодо впровадження нових методів та інструментів у практику корпоративного сектора.

Основні задачі, які ставляться в дипломному дослідженні

У ході виконання магістерської роботи передбачається розв'язання таких основних задач:

1. Проведення літературного огляду. Аналіз наукових робіт та публікацій у сфері безпеки доступу до інформаційних систем.

2. Розробка концепції системи моніторингу. Визначення архітектури системи, вибір технологій та інструментів для реалізації.

3. Розробка методів аутентифікації та авторизації. Створення механізмів перевірки ідентифікації користувачів та надання їм відповідних прав доступу.

4. Розробка системи аудиту та моніторингу. Реалізація засобів для фіксації подій та відстеження дій користувачів.

5. Проведення експериментів та оцінка результатів. Тестування розроблених методів та інструментів у реальних умовах та аналіз отриманих даних.

6. Формулювання рекомендацій для практики. Висновки та рекомендації щодо впровадження розроблених рішень у практику корпоративного сектора.

РОЗДІЛ 1
АНАЛІЗ ПРИНЦИПІВ РОБОТИ СИСТЕМ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ

1.1. Проблеми інформаційної безпеки підприємств та організацій

Проблема інформаційної безпеки стала вкрай актуальною для підприємств та організацій у сучасному цифровому світі. Зростаюча кількість даних, які зберігаються електронно, а також збільшення обсягу інтернет-з'єднань відкривають нові можливості для зловмисників та загроз для конфіденційності, цілісності та доступності інформації. У цьому підрозділі ми розглянемо ключові аспекти проблеми інформаційної безпеки, що стосуються підприємств та організацій.

Загрози інформаційної безпеки різноманітні та постійно зростають у розмірі та складності. Основні види загроз включають:

1. Кібератаки. Це атаки на інформаційні системи та мережі з метою незаконного доступу, розповсюдження шкідливого програмного забезпечення або крадіжки конфіденційних даних.

2. Витік інформації. Втрата чутливої інформації може бути результатом внутрішніх порушень безпеки або недостатньої захищеності зовнішніх загроз.

3. Соціальна інженерія. Атаки, які використовують маніпулювання людьми з метою отримання доступу до систем або інформації.

4. Деніал-сервіс атаки. Атаки, спрямовані на перекриття доступу до ресурсів, що можуть включати в себе переповнення мережевих каналів або відмову у послузі.

5. Зловживання прав доступу. Несанкціоноване використання привілеїв користувача або адміністратора для незаконних дій.

Наслідки порушень інформаційної безпеки

Порушення інформаційної безпеки можуть призвести до серйозних

наслідків для підприємств та організацій. Деякі з можливих наслідків включають:

1. Фінансові втрати. Витрати на відновлення систем та послуг, відшкодування втрат даних або фінансових санкцій.

2. Втрата репутації. Порушення інформаційної безпеки може негативно вплинути на репутацію компанії та втрату довіри клієнтів.

3. Законодавчі санкції. Порушення норм інформаційної безпеки може призвести до юридичних наслідків, таких як штрафи та судові позови.

4. Втрата конкурентної переваги. Втрата конфіденційної інформації або технічних розробок може призвести до втрати конкурентних переваг на ринку.

Важливість систем контролю та управління доступом

Для запобігання порушенням інформаційної безпеки підприємства та організації повинні розвивати та впроваджувати ефективні системи контролю та управління доступом. Ці системи дозволяють:

1. Ідентифікувати користувачів. Визначення осіб, які мають доступ до інформаційних ресурсів.

2. Авторизувати користувачів. Надання прав користувачам відповідно до їхніх ролей та відповідності.

3. Моніторити та реєструвати дії користувачів. Фіксація подій та відстеження незвичайних або підозрілих дій.

4. Забороняти несанкціонований доступ. Захист від недопущення несанкціонованого доступу до систем та даних.

5. Вживати заходи безпеки. Застосування шифрування, аутентифікації та інших методів для забезпечення безпеки.

Сучасні тенденції в інформаційній безпеці

Інформаційна безпека постійно розвивається відповідно до сучасних тенденцій та викликів. Деякі з ключових тенденцій включають:

1. Мобільність. Зростання кількості мобільних пристроїв та роботи на відстані вимагають нових підходів до контролю доступу.

2. Хмарні технології. Використання хмарних рішень для зберігання та обробки даних вимагає спеціалізованих заходів безпеки.

3. Інтернет речей. Зростання кількості підключених пристроїв створює нові вектори атак.

4. Штучний інтелект та машинне навчання. Використання AI для виявлення загроз та попередження атак.

Аналіз проблем інформаційної безпеки для підприємств та організацій підкреслює важливість розвитку та впровадження систем контролю та управління доступом. Загрози інформаційної безпеки постійно зростають, і підприємства повинні бути готові до їх виявлення та реагування. Сучасні тенденції в інформаційній безпеці вимагають постійного оновлення підходів та технологій, що використовуються для захисту інформації. У цьому контексті, подальше вивчення систем контролю та управління доступом є надзвичайно важливим для забезпечення інформаційної безпеки в сучасному світі.

1.2. Сучасні технології організації безпечного доступу на об’єкт

Сучасний світ обумовлює необхідність захисту об’єктів та ресурсів від несанкціонованого доступу. Забезпечення безпеки на об'єктах стало однією з ключових проблем сучасного суспільства. У цьому підрозділі ми детально розглянемо сучасні технології, що використовуються для організації безпечного доступу на об'єкт.

Технології біометричної аутентифікації

Однією з найбільш перспективних сучасних технологій в області організації безпечного доступу є біометрична аутентифікація. Вона базується на використанні унікальних біологічних або фізіологічних характеристик особи для її ідентифікації. Деякі з типів біометричних технологій включають:

1. Відбиток пальця. Використовується аналіз унікальних рис на поверхні пальця для ідентифікації особи.

2. Розпізнавання обличчя. Технологія, яка аналізує структуру та особливості обличчя для ідентифікації користувача.

3. Сканування радужки. Використовується для аналізу унікальних рис

радужки ока.

4. Голосове впізнавання. Визначає особливості голосу для ідентифікації особи.

5. Сканування вен. Аналізується унікальна структура вен на руці чи іншій частині тіла.

Біометрична аутентифікація дозволяє забезпечити високий рівень безпеки, оскільки унікальні біологічні характеристики складно підробити або підмінити. Вона широко використовується в сучасних системах доступу до об'єктів, таких як корпоративні офіси, лабораторії та державні установи.

Системи двофакторної аутентифікації

Системи двофакторної аутентифікації (2FA) є ще однією важливою складовою сучасних технологій безпечного доступу. 2FA вимагає від користувача подання двох різних видів ідентифікаційної інформації для отримання доступу. Зазвичай це включає комбінацію чогось, що користувач знає (наприклад, пароль) та чогось, що він має (наприклад, мобільний пристрій).

Системи 2FA забезпечують додатковий рівень захисту, оскільки навіть у випадку витоку паролю, зловмисники не матимуть доступу до другого фактора ідентифікації. Популярні методи 2FA включають в себе використання смс-повідомлень, мобільних додатків для генерації одноразових кодів або використання фізичних апаратних пристроїв (наприклад, ключів безпеки).

Віртуальні приватні мережі (VPN)

Віртуальні приватні мережі (VPN) є іншою важливою технологією для організації безпечного доступу на об'єкт. VPN створюють безпечне з'єднання між користувачем і мережею через шифрування даних, що передаються через відкритий інтернет.

Основні переваги VPN включають в себе анонімність та конфіденційність даних, захист від перехоплення інформації з боку третіх осіб та можливість з'єднання з об'єктом з будь-якого місця. VPN широко використовуються для забезпечення безпечного доступу до корпоративних мереж, особливо при роботі з віддаленими працівниками та віддаленими об'єктами.

Системи одноразових паролів (OTP)

Системи одноразових паролів (OTP) використовуються для забезпечення додаткового рівня безпеки в системах контролю і управління доступом. OTP генеруються та використовуються лише один раз для ідентифікації користувача.

Одноразові паролі можуть бути надіслані на мобільний пристрій користувача, відображені на фізичному токені або згенеровані за допомогою спеціального додатку. Після використання OTP вони стають недійсними, що робить надмірно складним їх перехоплення та використання незаконними особами.

Сучасні технології організації безпечного доступу на об'єкт дозволяють забезпечити високий рівень захисту для об'єктів та ресурсів. Використання біометричної аутентифікації, систем двофакторної аутентифікації, VPN, систем одноразових паролів та інших сучасних методів дозволяє ефективно захищати інформацію та об'єкти від несанкціонованого доступу. Забезпечення безпечного доступу стає все важливішим завданням у світі, де кількість загроз постійно зростає, і розробка та впровадження сучасних технологій є важливим етапом у забезпеченні інформаційної безпеки.

Сучасні технології організації безпечного доступу на об'єкт є ключовим елементом систем контролю доступу і включають в себе різні методи та рішення для забезпечення надійного та безпечного входу на об'єкт. Вони враховують сучасні вимоги до безпеки та забезпечують захист від різних загроз. Нижче розглянемо детально деякі сучасні технології організації безпечного доступу на об'єкт:

1. Біометрична ідентифікація:

– Відбитки пальців: Цей метод використовує сканери відбитків пальців для ідентифікації особи за унікальними характеристиками відбитка пальця. Він є одним із найпоширеніших і надійних методів біометричної ідентифікації.

– Розпізнавання обличчя: Системи розпізнавання обличчя використовують аналіз особливостей обличчя для ідентифікації особи. Цей метод може бути використаний в системах відеоспостереження та доступу.

– Сканування радужки та сетківки: Ці методи базуються на скануванні радужки або сетківки ока. Вони вважаються найбільш надійними, оскільки унікальність біометричних ознак ока малоймовірно підробити.

2. Системи контролю доступу на основі смарт-карт та брелоків:

– Смарт-карти: Смарт-карти містять електронну інформацію, яка може бути використана для ідентифікації особи. Вони можуть бути чітко пов'язані з конкретним користувачем і легко скасовані або замінені у випадку втрати.

– Брелоки: Брелоки – це невеликі пристрої, які можуть бути прикріплені до ключів або одягу і використовуються для безконтактного доступу на об'єкт.

3. Системи на основі смартфонів:

– Мобільні додатки: Смартфони можуть бути використані як ідентифікатори за допомогою спеціальних додатків. Вони можуть використовувати різні технології, такі як Bluetooth, NFC або QR-коди, для забезпечення доступу.

4. Інтернет-зв'язок та хмарні технології:

– Хмарні системи контролю доступу: Ці системи дозволяють віддалено керувати та моніторити доступ на об'єкті через Інтернет. Інформація зберігається у хмарному сховищі, що робить її доступною з будь-якого місця з Інтернет-з'єднанням.

5. Використання штучного інтелекту (ШІ):

– ШІ для аналізу поведінки: Системи контролю доступу можуть використовувати ШІ для аналізу поведінки користувачів. Наприклад, ШІ може виявити незвичайну або підозрілу активність та вжити заходів для запобігання несанкціонованому доступу.

6. Системи відеоспостереження і відеоаналітика:

– Відеокамери з розпізнаванням обличчя: Деякі системи відеоспостереження використовують відеокамери з функцією розпізнавання обличчя для ідентифікації осіб, які намагаються отримати доступ.

– Відеоаналітика: Системи відеоаналітики можуть автоматично виявляти підозрілу або незвичайну активність на об'єкті і сповіщати операторів про можливі загрози.

7. Системи геолокації та GPS:

– Системи геолокації: Вони можуть використовувати GPS або інші методи для визначення місцезнаходження користувача і надавати доступ лише в певних зонах.

8. Технології розширеної реальності (AR):

– AR для ідентифікації: Технології розширеної реальності можуть використовуватися для ідентифікації осіб на основі їхніх фізичних параметрів.

Сучасні технології організації безпечного доступу на об'єкт надають широкий спектр можливостей для забезпечення безпеки та контролю доступу. Вибір конкретної технології залежить від потреб об'єкта, бюджету та рівня безпеки, який необхідно забезпечити.

1.3. Огляд існуючих систем контролю доступу на об’єкт

Огляд існуючих систем контролю доступу на об'єкт є важливою частиною будь-якого дослідження в галузі безпеки та систем безпеки. Цей огляд допомагає встановити те, які технології та підходи вже використовуються для забезпечення безпеки об'єктів, їх переваги та недоліки, і створити основу для подальшого дослідження та розробки нових систем контролю доступу.

1. Системи доступу на основі ключів та карток: Один із найстаріших методів контролю доступу - використання ключів або карток. Кожен співробітник отримує ключ або картку, яку потрібно пред'являти для доступу на об'єкт. Цей метод має деякі переваги, такі як простота і дешевизна, але він також має серйозні недоліки, такі як можливість втрати ключа, його крадіжки або дублювання.

2. Системи на основі PIN-кодів: Деякі системи контролю доступу використовують PIN-коди, що вводяться співробітниками для входу на об'єкт. Цей метод може бути більш надійним, оскільки не вимагає фізичного носія, але він також має вразливості, такі як можливість витоку коду.

3. Системи на основі біометричних даних: Сучасні системи контролю доступу використовують біометричні дані, такі як відбитки пальців, розпізнавання

обличчя, сканування радужки або голосове впізнавання. Ці системи надзвичайно надійні, оскільки вони ідентифікують особу на основі унікальних фізичних характеристик. Проте вони можуть бути вартісними та вимагати складніше обладнання.

4. Системи "світло-карта": Ці системи використовують комбінацію світлодіодів і карток для ідентифікації особи. Особа має піднести картку до считувача, а потім зіставити світлодіоди на картці зі світлодіодами на панелі. Цей метод може бути швидким та вартісним.

5. Системи "розумний дім": Сучасні системи контролю доступу можуть бути інтегровані з системами "розумного дому". Це дозволяє власникам об'єкта віддалено контролювати доступ та використовувати додатки на смартфонах для цього.

6. Інтегровані системи безпеки: Більші об'єкти та комплекси використовують інтегровані системи безпеки, які поєднують в собі різні аспекти безпеки, включаючи контроль доступу, відеоспостереження, сигналізацію та інші функції. Ці системи надзвичайно потужні і можуть бути настроєні під конкретні потреби об'єкта.

7. Системи відстеження та аналізу поведінки: Деякі сучасні системи використовують аналіз поведінки для визначення, чи є певна особа підозрілою або несправедливою. Це може включати в себе відстеження рухів, шаблонів поведінки та інші аспекти.

Цей огляд існуючих систем контролю доступу на об'єкт показує, що існує безліч різних підходів та технологій для забезпечення безпеки. Кожна з цих систем має свої переваги та недоліки, і вибір конкретної системи залежить від потреб об'єкта, бюджету та інших факторів.

У сучасному світі забезпечення безпеки об'єктів та ресурсів стає все більш актуальним завданням. Існуючі системи контролю і управління доступом грають ключову роль у забезпеченні безпеки об'єктів та інфраструктури. У цьому підрозділі ми розглянемо різні існуючі системи контролю доступу, їхні принципи роботи та особливості.

Системи контролю доступу (ACS) складаються з різних компонентів та підсистем, які спільно забезпечують безпеку об'єктів. Основні компоненти ACS включають:

1. Контролери доступу. Це електронні пристрої, які управляють доступом до об'єкта. Вони приймають рішення щодо надання чи відмови у доступі.

2. Сервери управління. Сервери забезпечують централізований контроль і управління системою, зберігають базу даних користувачів та журнали подій.

3. Системи ідентифікації. Для ідентифікації користувачів використовуються різні методи, такі як біометричні дані, картки доступу, PIN-коди тощо.

4. Системи контролю. Вони включають в себе різні пристрої, такі як кард-рідери, електромагнітні замки, відеокамери тощо.

5. Системи моніторингу та журналювання. Вони фіксують події та дії користувачів, забезпечуючи можливість аналізу та реагування на події.

Системи контролю доступу базуються на декількох основних принципах роботи:

1. Ідентифікація користувача. Користувач повинен ідентифікувати себе перед отриманням доступу. Це може бути зроблено за допомогою біометричних даних, карточки доступу, пароля чи іншого ідентифікаційного методу.

2. Аутентифікація користувача. Після ідентифікації користувач повинен аутентифікувати свою ідентичність, зазвичай, представивши відомий пароль чи інший фактор аутентифікації.

3. Авторизація. Після успішної аутентифікації система визначає права доступу користувача та визначає, які ресурси він може використовувати.

4. Моніторинг та журналювання. Всі дії користувачів фіксуються системою для подальшого аналізу та аудиту.

5. Керування доступом. Система контролю доступу визначає, чи надавати користувачу доступ до об'єкта, і в разі потреби блокує або обмежує доступ.

Існує кілька різновидів систем контролю доступу, включаючи:

1. Фізичні системи контролю доступу. Вони використовуються для контролю доступу до фізичних об'єктів, таких як приміщення, промислові об'єкти,

аеропорти тощо. Ці системи можуть включати в себе кард-рідери, бар'єри, електронні замки.

2. Логічні системи контролю доступу. Вони використовуються для обмеження доступу до комп'ютерних та інформаційних ресурсів, таких як файли, бази даних, програми тощо. Ці системи можуть базуватися на аутентифікації через паролі, біометричні дані, ключі безпеки.

3. Відеоспостереження та системи відстеження. Вони використовуються для відображення та відстеження подій на об'єкті, забезпечуючи можливість визначення осіб та виявлення незвичайних подій.

4. Мережеві системи контролю доступу. Вони забезпечують можливість дистанційного контролю доступу до об'єкта через мережу Інтернет або локальну мережу.

На ринку існує багато популярних систем контролю доступу, які використовуються в різних галузях. Деякі з них включають:

1. HID Global. Відомий світовий виробник систем контролю доступу, який пропонує широкий спектр рішень для фізичного та логічного контролю доступу.

2. Tyco Security Products. Компанія, яка спеціалізується на системах безпеки та контролю доступу для корпоративних клієнтів.

3. Bosch Security Systems. Виробник інноваційних рішень для безпеки та контролю доступу.

4. LenelS2. Компанія, яка пропонує інтегровані рішення для безпеки та контролю доступу.

Огляд існуючих систем контролю доступу показує, що ця область технологій постійно розвивається та розширюється. Сучасні системи контролю доступу надають багато можливостей для забезпечення безпеки об'єктів та ресурсів у різних галузях. За допомогою компонентів, таких як контролери доступу, сервери управління, системи ідентифікації, системи контролю та інші, системи контролю доступу забезпечують ідентифікацію, аутентифікацію, авторизацію та моніторинг користувачів, що робить їх ефективними інструментами для забезпечення безпеки об'єктів у сучасному світі.

1.4. Принцип роботи мережевої системи контролю доступу

Мережеві системи контролю доступу (Network Access Control, NAC) в сучасному світі стали невід'ємною частиною забезпечення безпеки мереж та об'єктів. Ці системи відповідають за здійснення контролю доступу до мережі та ресурсів, що підключені до неї. У цьому підрозділі ми детально розглянемо принципи роботи мережевих систем контролю доступу, їхні особливості та переваги.

Перед тим як розглядати принципи роботи NAC, давайте з'ясуємо, які завдання перед нею стоять:

1. Ідентифікація і аутентифікація користувачів і пристроїв. Однією з основних задач NAC є визначення, хто і що підключається до мережі. Це важливо для подальшої авторизації та керування доступом.

2. Авторизація та призначення прав доступу. Після ідентифікації користувача або пристрою NAC визначає, які ресурси та служби доступні цьому суб'єкту. Вона може встановлювати правила та обмеження щодо доступу.

3. Моніторинг і аналіз активності. NAC здатна відслідковувати активність користувачів і пристроїв у мережі, а також реагувати на незвичайні події та загрози.

4. Відокремлення незабезпечених пристроїв. Якщо NAC виявляє пристрій або користувача, який становить загрозу для мережі, вона може відокремити цей пристрій або надати йому обмежений доступ.

5. Забезпечення відповідності політикам безпеки. NAC дозволяє забезпечити відповідність мережевих ресурсів і користувачів політикам безпеки та стандартам компанії.

Основними принципами роботи мережевої системи контролю доступу є:

1. Ідентифікація користувачів і пристроїв. При підключенні користувача чи пристрою до мережі NAC виконує ідентифікацію за допомогою різних методів, таких як аутентифікація за допомогою логіну та паролю, біометричні дані,

сертифікати, MAC-адреси, IP-адреси тощо.

2. Аутентифікація. Після ідентифікації NAC перевіряє ідентифікаційні дані користувача або пристрою для визначення їхньої автентичності. Це може включати в себе перевірку паролю, використання біометричних даних або інших методів.

3. Авторизація. Після успішної аутентифікації NAC визначає, які ресурси та служби доступні користувачу або пристрою. Вона може надавати права доступу відповідно до політик безпеки та правил конфігурації.

4. Моніторинг та аналіз. NAC надає можливість моніторингу активності користувачів і пристроїв у мережі. Вона фіксує події, виявляє незвичайну активність та вживає заходів для реагування на загрози.

5. Відокремлення незабезпечених пристроїв. Якщо NAC виявляє пристрій чи користувача, який не відповідає політикам безпеки, вона може відокремити цей об'єкт від мережі або обмежити його доступ до обмежених ресурсів.

6. Забезпечення відповідності політикам безпеки. NAC дозволяє забезпечити відповідність користувачів і пристроїв політикам безпеки компанії чи організації. Вона перевіряє, чи відповідає конфігурація пристроїв стандартам безпеки.

Для більшої розбірливості принципів роботи мережевої системи контролю доступу, давайте розглянемо деякі її особливості:

1. Централізований контроль. NAC надає можливість централізованого управління доступом до мережі. Вся інформація про користувачів, пристрої та їхні права зберігається на центральному сервері, що дозволяє здійснювати контроль і управління з одного пункту.

2. Динамічний контроль. NAC може змінювати права доступу користувачів та пристроїв в реальному часі в залежності від їхньої активності та стану мережі. Це дозволяє реагувати на зміни в середовищі мережі.

3. Інтеграція з іншими системами безпеки. NAC може інтегруватися з іншими системами безпеки, такими як системи виявлення вторгнень (IDS), системи запобігання вторгненням (IPS) та антивірусні програми для забезпечення комплексного захисту мережі.

4. Керування віртуальними приватними мережами (VPN). NAC може керувати доступом до VPN, забезпечуючи безпеку підключення з віддалених місць.

5. Захист від несанкціонованого доступу. NAC допомагає запобігти несанкціонованому доступу до мережі шляхом виявлення та відокремлення небезпечних пристроїв.

Мережеві системи контролю доступу є важливою складовою систем безпеки в сучасних мережах. Вони забезпечують ідентифікацію, аутентифікацію, авторизацію, моніторинг та управління доступом користувачів і пристроїв до мережевих ресурсів. Принципи роботи NAC включають ідентифікацію, аутентифікацію, авторизацію, моніторинг, відокремлення незабезпечених пристроїв та забезпечення відповідності політикам безпеки. Основні переваги NAC полягають у централізованому управлінні, динамічному контролі, інтеграції з іншими системами безпеки та захисті від несанкціонованого доступу. Використання мережевих систем контролю доступу стає все важливішим у світі, де інтернет-з'єднання стають все доступнішими, а загрози для мереж безпеки зростають.

Розглянемо детально принципи роботи такої системи:

1. Ідентифікація та аутентифікація користувачів:

– Користувачі подають ідентифікатори, такі як смарт-карти, біометричні дані (відбитки пальців, обличчя, ірис), PIN-коди або інші електронні ключі для отримання доступу.

– Система перевіряє ідентифікаційні дані та аутентифікує користувача, перевіряючи, чи вони відповідають допускам та правилам доступу, збереженим у базі даних.

2. Управління правами доступу:

– Адміністратори системи налаштовують права доступу для різних користувачів та груп користувачів. Це включає в себе визначення, до яких зон або об'єктів можуть мати доступ користувачі, і в які часи.

– Правила доступу можуть бути індивідуальними (для окремих

користувачів) або груповими (для категорій користувачів), і вони регулюються на основі поточного статусу об'єкта та часу.

3. Контроль доступу та реєстрація подій:

– Кожен спроба доступу, яка включає в себе ідентифікацію та аутентифікацію, реєструється в системі.

– Система контролює, чи відповідає запит на доступ поточним правилам і може надавати або відмовляти доступу залежно від результату перевірки.

4. Інтеграція з іншими системами безпеки:

– Мережева СКУД може бути інтегрована з іншими системами безпеки, такими як системи відеоспостереження, системи охоронної та пожежної сигналізації, що дозволяє забезпечувати комплексний моніторинг та керування безпекою на об'єкті.

5. Віддалене керування та моніторинг:

– Сучасні мережеві СКУД можуть бути керовані та моніторені віддалено через Інтернет або мережу Інтранет. Це дозволяє адміністраторам відстежувати та керувати доступом навіть з віддалених місць.

6. Автоматичне сповіщення та реагування на події:

– Система може бути налаштована на автоматичне сповіщення адміністраторів або служби безпеки про підозрілу або нестандартну активність.

– Також може бути реалізована автоматична реакція, така як блокування доступу або включення сирени, у разі виявлення загрози.

7. Захист від несанкціонованого доступу:

– Система повинна забезпечувати високий рівень захисту від спроб несанкціонованого доступу, злому чи підбору ідентифікаційних даних.

8. Система архівування та аналізу подій:

– Всі події та записи повинні бути збережені в архіві для подальшого аналізу, розслідування та використання для статистичного аналізу.

9. Забезпечення роботи в аварійних ситуаціях:

– Система повинна мати механізми для забезпечення роботи навіть в умовах відключення живлення або інших аварійних ситуаціях.

10. Інтеграція з іншими системами управління об'єктом:

– Мережева СКУД може інтегруватися з системами управління освітленням, системами кондиціонування повітря та іншими системами, що дозволяє забезпечувати ефективне управління об'єктом та ресурсами.

Принцип роботи мережевої системи контролю доступу полягає в поєднанні та координації різних компонентів та процесів для забезпечення безпечного та ефективного контролю доступу до об'єкта. Така система дозволяє забезпечити високий рівень безпеки, комфорту та зручності для користувачів та адміністраторів об'єкта.

1.5. Висновки до розділу

При проведенні аналізу розглянутих вище систем контролю доступу, основаних на класичних методах, було виявлено ряд недоліків, які потребують уваги та вдосконалення:

1. Проблеми зі забороною подвійного проходу: В більшості систем контролю доступу, які контролюють як вхід, так і вихід особи, використовується функція заборони подвійного проходу. Ця функція призначена для запобігання передачі ключа-ідентифікатора іншій особі. Проте в житті виникає ситуації, коли особа, маючи ключ, не входить в приміщення або зволікає з входом. У таких випадках система відмовляє співробітникові у вході, що може призвести до незручностей та затримок.

2. Проблеми з втратою ідентифікатора: Іншою проблемою є втрата, забування або залишення ідентифікатора. Якщо подібна ситуація відбулася поза об’єктом, який охороняється, то виправлення може бути досить простим - особа може звернутися до охоронця або оператора системи контролю доступу. Однак, якщо ідентифікатор був втрачений на території підприємства і особа залишається в об’єкті, це може призвести до складнощів і незручностей.

3. Загроза викрадення або дублювання ідентифікатора: Іншою важливою проблемою є можливість викрадення або дублювання ідентифікатора, що може

вразити безпеку підприємства. Класичні системи контролю доступу не завжди можуть забезпечити високий рівень захисту об’єктів з підвищеними вимогами до безпеки.

Один із варіантів вирішення цих проблем - використання біометричних засобів ідентифікації. У біометричних системах ідентифікується не предмет (брелок, магнітна карта і т. д.), а сама особа на основі її біометричних параметрів, таких як відбиток пальця, розпізнавання обличчя, структура руки і т. д. Це забезпечує вищий рівень безпеки, оскільки ідентифікація здійснюється на основі унікальних фізичних характеристик особи.

Використання біометричної ідентифікації може бути важливим кроком у покращенні систем контролю доступу, забезпечуючи більшу надійність і безпеку для підприємств та організацій. Такий підхід дозволяє уникнути багатьох проблем, пов'язаних із традиційними методами контролю доступу.

РОЗДІЛ 2
ПРИНЦИПИ РОБОТИ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ

2.1. Структура і основні елементи систем відеоспостереження

Системи відеоспостереження є важливою складовою інфраструктури безпеки в сучасному світі. Вони використовуються для захисту об'єктів, моніторингу подій, забезпечення безпеки громадських місць, корпоративних приміщень, транспортних мереж та інших об'єктів. У цьому підрозділі ми детально розглянемо структуру і основні елементи систем відеоспостереження.

Проте як і раніше поширений «класичний» варіант реалізації на базі багатоканальних відеореєстраторів (рис. 2.1). Відеокамери розрізняються і за основними параметрами, і за конструктивними особливостями, і за принципом обробки сигналу.

Рис. 2.1. «Класичний» варіант системи відеоспостереження

2.1.2. Складові системи відеоспостереження

Система відеоспостереження складається з різних компонентів та елементів, які спільно забезпечують її функціональність та ефективність. Основними складовими системи відеоспостереження є:

1. Відеокамери. Відеокамери або камери відеоспостереження є ключовими пристроями в системі. Вони використовуються для захоплення зображень та відео в реальному часі. Відеокамери можуть бути різних типів, таких як аналогові, IP-камери, PTZ-камери (з панорамою, нахилом і зумом) тощо.

2. Об'єктиви. Об'єктиви встановлюються на відеокамери для керування фокусною відстанню, кутом огляду та зумом. Вони впливають на якість та характеристики відеозображення.

3. Відеореєстратори. Відеореєстратори (DVR - Digital Video Recorder) або NVR (Network Video Recorder) використовуються для запису та зберігання відеоматеріалів. Вони можуть мати велику ємність для зберігання відео, а також функції архівування та резервного копіювання.

4. Монітори. Монітори використовуються для відображення відеозображення в режимі реального часу або для перегляду записаних матеріалів. Розмір та кількість моніторів може варіюватися в залежності від потреб користувача.

5. Системи зберігання даних. Для зберігання великої кількості відеоданих потрібні надійні системи зберігання, такі як NAS (Network Attached Storage) або SAN (Storage Area Network).

6. Системи передачі даних. Для передачі відеоданих від камер до відеореєстраторів чи інших об'єктів використовуються мережеві протоколи та технології. Для IP-камер і NVR використовуються мережеві комутатори та маршрутизатори.

7. Системи аналізу відео. Деякі системи відеоспостереження включають аналітичні інструменти, такі як виявлення руху, розпізнавання номерних знаків, виявлення обличчя, аналіз поведінки тощо.

8. Комп'ютери та сервери. Комп'ютери та сервери використовуються для управління системою відеоспостереження, а також для встановлення та конфігурації програмного забезпечення.

Основні функції систем відеоспостереження

Системи відеоспостереження виконують різні функції, спрямовані на забезпечення безпеки та моніторингу подій. Основні функції включають в себе:

1. Спостереження в реальному часі. Системи відеоспостереження дозволяють персоналу спостерігати за об'єктом в режимі реального часу, що дозволяє вчасно реагувати на події.

2. Запис відеоматеріалів. Відеоспостереження може записувати відео на відеореєстратори для подальшого аналізу, слідкування за подіями чи розслідування інцидентів.

3. Деталізований огляд об'єкта. Завдяки різним типам камер і об'єктивів, системи відеоспостереження можуть надавати деталізований огляд об'єкта та його оточення.

4. Виявлення і відстеження руху. Деякі системи відеоспостереження обладнані функціями виявлення та відстеження руху об'єктів, що дозволяє автоматично реагувати на них.

5. Аналітика та розпізнавання. Високорозвинені системи відеоспостереження можуть використовувати аналітичні інструменти для розпізнавання обличчя, номерних знаків, предметів тощо.

Сучасні технології та тренди в сфері відеоспостереження

Сфера відеоспостереження постійно розвивається, і існують деякі сучасні технології та тренди, які впливають на її розвиток:

1. Висока роздільна здатність (HD та 4K). Сучасні відеокамери пропонують високу роздільну здатність зображення (HD і 4K), що дозволяє отримувати більш якісні та деталізовані відеозаписи.

2. Інтернет речей (IoT). Відеоспостереження поєднується з IoT для створення "розумних" систем, які автоматично реагують на різні події та умови.

3. Штучний інтелект (AI) та машинне навчання. Використання AI і машинного навчання дозволяє автоматизувати виявлення об'єктів, розпізнавання обличчя, аналіз поведінки та інші функції.

4. Хмарні рішення. Відеозапис може зберігатися в хмарних сховищах, що забезпечує віддалений доступ і забезпечує безпеку даних.

Системи відеоспостереження є важливою складовою для забезпечення безпеки та моніторингу подій у різних сферах життя. Вони складаються з різних компонентів, включаючи відеокамери, об'єктиви, відеореєстратори, монітори та інші елементи.

2.2. Області застосування і огляд програмного забезпечення систем відеоспостереження

Системи відеоспостереження знаходять широке застосування в сучасному світі, де забезпечення безпеки, відслідковування подій та контроль є важливими завданнями у різних галузях. У цьому підрозділі ми розглянемо основні області застосування систем відеоспостереження та проведемо огляд програмного забезпечення, що використовується в таких системах.

Системи відеоспостереження знаходять застосування в різних галузях і областях, забезпечуючи безпеку та контроль. Основні області застосування включають:

1. Охорона об'єктів. Системи відеоспостереження використовуються для забезпечення безпеки на об'єктах, таких як банки, магазини, ресторани, готелі, аеропорти, станції метро та інші публічні місця. Вони допомагають відслідковувати незвичайну активність, виявляти потенційні загрози та реагувати на них.

2. Моніторинг транспорту. Системи відеоспостереження встановлюються на дорогах, вулицях та громадському транспорті для контролю руху, виявлення порушень правил дорожнього руху, а також для відслідковування вандалізму та аварій.

3. Відеонагляд у промисловості. У промисловості системи відеоспостереження використовуються для моніторингу виробничих процесів, контролю за обладнанням та безпеки працівників. Вони допомагають у виявленні несправностей та аварійних ситуацій.

4. Системи безпеки в приватних будинках. Системи відеоспостереження встановлюються в приватних будинках та квартирах для захисту від зламів, відслідковування подій та надання спокою мешканцям.

5. Охорона природних резерватів і природних ресурсів. Відеоспостереження використовується для контролю за природними резерватами, заповідниками та природними ресурсами, щоб запобігти незаконному видобутку та нищівництву.

6. Медичні установи. В лікарнях і клініках системи відеоспостереження використовуються для контролю за доступом до медичних приміщень, а також для моніторингу пацієнтів і забезпечення їх безпеки.

Програмне забезпечення є важливою складовою систем відеоспостереження і відіграє рішучу роль у зборі, обробці і аналізі відеоданих. Основні функції програмного забезпечення включають:

1. Відображення відеопотоків. Програмне забезпечення дозволяє операторам переглядати відеопотоки з різних камер на одному екрані, змінювати режими відображення і масштабувати зображення.

2. Запис відеоданих. Програмне забезпечення здатне записувати відеопотоки на сервер або мережевий накопичувач, забезпечуючи зберігання даних для подальшого аналізу і використання в доказовій базі.

3. Аналіз відеоданих. Деяке програмне забезпечення має вбудовані аналітичні інструменти, які дозволяють автоматично виявляти рух, об'єкти, обличчя, автомобільні номери тощо.

4. Інтеграція з іншими системами. Програмне забезпечення може бути інтегроване з іншими системами безпеки, такими як системи контролю доступу, системи виявлення вторгнень, пожежні системи і т.д.

5. Дистанційне керування. Оператори можуть керувати системою відеоспостереження з віддаленого місця через інтерфейс веб-браузера або спеціальні додатки.

6. Забезпечення безпеки даних. Програмне забезпечення забезпечує захист відеоданих від несанкціонованого доступу та може використовувати різні методи шифрування та аутентифікації.

Системи відеоспостереження мають широке застосування в різних галузях та областях, забезпечуючи безпеку, контроль і відслідковування подій. Основні області застосування включають охорону об'єктів, моніторинг транспорту, відеонагляд у промисловості, системи безпеки в приватних будинках, охорону природних резерватів та медичні установи. Програмне забезпечення для систем відеоспостереження виконує важливі функції, включаючи відображення відеопотоків, запис відеоданих, аналіз даних, інтеграцію з іншими системами та забезпечення безпеки даних. Використання систем відеоспостереження допомагає підвищити рівень безпеки і ефективності в різних сферах діяльності.

2.3. Інтегровані системи безпеки

Інтегровані системи безпеки (ІСБ) є сучасними технологічними рішеннями, спрямованими на забезпечення комплексного контролю та захисту об'єктів, будівель, інфраструктури та інших цінних ресурсів. Цей підрозділ присвячений розгляду інтегрованих систем безпеки та їхніх основних принципів роботи.

Інтегрована система безпеки (ІСБ) є комплексною технологічною системою, яка об'єднує різні засоби і засоби безпеки для забезпечення цілісного підходу до захисту об'єкта або території. Основними характеристиками ІСБ є:

1. Мультимодальність. ІСБ може об'єднувати різні види систем безпеки, такі як системи відеоспостереження, системи контролю доступу, системи виявлення вторгнень, пожежні системи та інші.

2. Централізоване управління. ІСБ надає можливість централізованого контролю і управління всіма компонентами системи з одного пункту.

3. Інтеграція з іншими системами. ІСБ може бути інтегрованою з іншими інформаційними системами, такими як системи управління будівлями (BMS), системи автоматизації та управління, системи внутрішньої комунікації і т.д.

4. Аналітика та інформаційна обробка. ІСБ забезпечує можливість аналізу і обробки даних, що надходять від різних компонентів системи, для виявлення аномалій та подій.

5. Звітність і архівування. ІСБ може зберігати дані про події і відеозаписи для подальшого аналізу та створення звітів.

Основними принципами роботи ІСБ є:

1. Ідентифікація і аутентифікація. ІСБ визначає ідентифікацію користувачів та об'єктів, що входять на об'єкт або взаємодіють з ним, і аутентифікує їх для забезпечення доступу лише авторизованим особам і об'єктам.

2. Моніторинг і виявлення. ІСБ постійно моніторить стан об'єкта та виявляє аномалії або підозрілу активність, що може вказувати на загрози.

3. Реакція і управління. У разі виявлення загроз або подій ІСБ може активувати різні заходи безпеки, такі як сигналізація, виклик правоохоронних органів, блокування доступу та інші.

4. Аналіз і звітність. ІСБ забезпечує можливість аналізу подій, створення звітів і архівування даних для подальшого використання.

Інтегровані системи безпеки мають широкий спектр застосувань у різних галузях та областях. Деякі з основних областей застосування включають:

1. Охорона об'єктів. ІСБ використовуються для забезпечення безпеки на об'єктах, таких як бізнес-центри, заводи, банки, аеропорти, торгові центри і інші.

2. Міська інфраструктура. ІСБ встановлюються для контролю за міською інфраструктурою, такою як дороги, міста, метро, а також для забезпечення безпеки громадських місць.

3. Енергетика та інфраструктура. ІСБ використовуються в енергетичних підприємствах, станціях з виробництва та розподілу електроенергії, водозабірних спорудах і інших інфраструктурних об'єктах.

4. Транспортна інфраструктура. ІСБ встановлюються на залізничних станціях, автовокзалах, портах, парковках та інших транспортних об'єктах.

5. Охорона природних резерватів і ресурсів. ІСБ використовуються для контролю за природними резерватами, заповідниками та ресурсами для запобігання незаконному видобутку і нищівництву.

6. Системи безпеки в приватних будинках і комплексах. ІСБ встановлюються в приватних будинках, апартаментах та житлових комплексах для захисту житлових приміщень та індивідуальної безпеки мешканців.

Для функціонування ІСБ необхідне спеціалізоване програмне забезпечення та обладнання. Основні компоненти ІСБ включають:

1. Сервери та обладнання для зберігання даних. Сервери використовуються для обробки і зберігання даних, включаючи відеозаписи та інші дані системи безпеки.

2. Контролери і зчитувачі. Контролери і зчитувачі використовуються для систем контролю доступу і ідентифікації користувачів.

3. Відеокамери і аудіозаписувачі. Для відеоспостереження використовуються відеокамери різних типів і аудіозаписувачі для зафіксовування відео- та аудіосигналів.

4. Сенсори та датчики. Для виявлення руху, вбудованого освітлення, диму і газів використовуються сенсори та датчики.

5. Програмне забезпечення управління. Програмне забезпечення для централізованого управління системою безпеки та аналізу даних.

6. Мережеве обладнання. Для підключення всіх компонентів системи та передачі даних використовуються мережеві комутатори, маршрутизатори та інші пристрої.

Інтегровані системи безпеки є потужними інструментами для забезпечення безпеки та контролю в різних галузях і областях. Вони дозволяють об'єднати різні види систем безпеки в одну цілісну інфраструктуру, яка забезпечує моніторинг, виявлення і реагування на загрози. Інтегровані системи безпеки використовуються в охороні об'єктів, управлінні міською інфраструктурою, енергетиці, транспорті, охороні природних резерватів і багатьох інших областях. Вони вимагають спеціалізованого програмного забезпечення і обладнання для ефективної роботи та забезпечення безпеки об'єктів і ресурсів.

2.4. Можливості використання відеокамер в сучасних біометричних методах ідентифікації

Сучасні технології відеоспостереження відкривають широкі можливості для застосування біометричних методів ідентифікації. Відеокамери можуть служити не лише для відеоспостереження та реєстрації подій, але й для збору біометричних даних, таких як відбитки обличчя, розпізнавання голосу, і сканування радужки. Цей підрозділ присвячений розгляду можливостей використання відеокамер в сучасних біометричних методах ідентифікації.

2.4.2. Біометрична ідентифікація: Визначення та принципи

Біометрична ідентифікація - це метод ідентифікації особи на основі її фізіологічних або поведінкових характеристик. Основними принципами біометричної ідентифікації є:

1. Унікальність: Кожна людина має унікальні біометричні характеристики, такі як відбитки пальців, структура обличчя, голос і інші, що можуть бути використані для ідентифікації.

2. Стабільність: Біометричні характеристики залишаються стабільними протягом тривалого часу і не змінюються залежно від зовнішніх чинників.

3. Вимірюваність: Біометричні характеристики можуть бути виміряні та перетворені на цифровий формат для подальшого аналізу.

4. Необхідність активності користувача: Для отримання біометричних даних зазвичай не потрібна активна участь користувача, що робить цей метод зручним для ідентифікації в реальному часі.

2.4.3. Можливості використання відеокамер в біометричних методах ідентифікації

Відеокамери відкривають широкі можливості для застосування біометричних методів ідентифікації завдяки своїм характеристикам та функціоналу:

1. Розпізнавання обличчя (Face Recognition): Відеокамери можуть використовуватися для захоплення зображень обличчя і подальшого аналізу з

використанням алгоритмів розпізнавання обличчя. Цей метод дозволяє ідентифікувати особу на основі унікальних рис обличчя, таких як форма носа, очей, рота тощо.

2. Відбитки пальців (Fingerprint Recognition): Відеокамери можуть бути використані для захоплення відбитків пальців через спеціальні пристрої, такі як сканери відбитків пальців. Цей метод є одним із найбільш поширених і ефективних методів біометричної ідентифікації.

3. Розпізнавання голосу (Voice Recognition): Відеокамери можуть зафіксувати звуковий сигнал, алгоритми розпізнавання голосу можуть аналізувати особливості голосу, такі як тембр, інтонація та швидкість говоріння, для ідентифікації користувача.

4. Сканування радужки (Iris Recognition): Відеокамери можуть бути використані для сканування радужки очей, що є однією з найбільш надійних біометричних характеристик для ідентифікації.

Всі біометричні системи працюють практично за однаковою схемою. По-перше, система запам’ятовує зразок біометричної характеристики (це і називається процесом запису). Під час запису деякі біометричні системи можуть попросити зробити декілька зразків для того, щоб скласти найбільш точне зображення біометричної характеристики. Потім отримана інформація обробляється і перетворюється в математичний код [9].

2.5. Висновки до розділу

Системи контролю і управління доступом (СКУД) вже давно зайняли важливе місце серед технічних систем безпеки на ринку. Їх популярність постійно зростає, і це не випадково. СКУД призначені для автоматичного контролю доступу, що робить їх незамінними для організацій та підприємств. Такі системи дозволяють автоматично впускати на територію об’єкта тих, кому це дозволено, і забороняти доступ тим, кому це заборонено. Це забезпечує надійний контроль над

переміщенням співробітників і відвідувачів на підприємстві або в будь-якому іншому об’єкті.

Цікаво, що темпи зростання продажів устаткування СКУД становлять величезні 15%, що вдвічі перевищує темпи росту інших систем охорони, які складають лише 7%. Це свідчить про те, наскільки важливою є проблема контролю доступу в сучасному світі.

СКУД можна розділити на два основних типи: програмні і контролерні. Основна відмінність полягає в тому, як інформація про доступ і карти обробляються. В контролерних системах інформація спочатку зберігається на контролері і потім переноситься на комп’ютер. У програмних СКУД обладнання безпосередньо зв’язане з комп’ютером, що дозволяє отримувати та обробляти інформацію в реальному часі.

Разом із системами СКУД іншим важливим компонентом інтегрованих систем безпеки є системи відеоспостереження. Основне завдання систем відеоспостереження - отримання, запис та відтворення візуальної інформації про поточні події на об’єкті. Сучасне обладнання для систем відеоспостереження дозволяє створювати надійні та зручні в експлуатації системи з високими технічними характеристиками.

Контроль за периметром об’єкта чи стеження за потоком відвідувачів - це лише невелика частина можливостей IP-відеоспостереження. Гнучкість налаштувань та простота інтеграції роблять ці системи вельми універсальними та ефективними інструментами для рішення найрізноманітніших завдань.

Інтегрована система безпеки (ІСБ) є необхідною умовою для забезпечення повноцінного контролю і захисту об’єкта. Вона об'єднує різні технічні засоби, такі як системи охоронної та пожежної сигналізації, оповіщення, управління протипожежною автоматикою, контроль доступу та системи відеоспостереження в єдиний комплекс, який легко інтегрується та управляється.

У нашому підприємстві використовується обладнання PERCo для організації системи контролю доступу. Електронні прохідні компанії PERCo є готовими рішеннями, які спрощують налаштування систем контролю доступу. Також,

система відеоспостереження BEWARD використовується для отримання візуальної інформації про поточні події на об’єкті. Вибір цих систем обгрунтований їхньою надійністю та здатністю інтеграції.

Сучасні інтегровані системи безпеки відповідають сучасним вимогам та тенденціям розвитку технологій і безпеки. Проте важливо пам'ятати, що це лише початок, і перспективи розвитку інтегрованих систем безпеки є нескінченними.

РОЗДІЛ 3
ОПИСАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОДУЛЯ

3.1. Описання існуючої СКУД

Система контролю управління доступом (СКУД) є комплексним рішенням, спрямованим на обмеження фізичного чи логічного доступу до об'єкта, зони або ресурсу тільки авторизованим користувачам. Основні складові СКУД включають:

1. Читачі та контролери: Читачі можуть бути різних типів, таких як карткові читачі, біометричні считувачі відбитків пальців, RFID-считувачі тощо. Контролери виконують роль центрального обчислювального пристрою, який обробляє інформацію від читачів і приймає рішення про допуск або відмову у доступі.

2. Ключові медіа: Це можуть бути фізичні ключі, магнітні картки, смарт-карти, браслети або інші носії, які ідентифікують користувача.

3. База даних користувачів: СКУД зберігає інформацію про користувачів, їхні права доступу, біометричні дані, історію входів і виходів.

4. Програмне забезпечення: Це основна частина СКУД, яка включає в себе програмне забезпечення для налаштування прав доступу, моніторингу подій, генерації звітів та інші функції.

Існуюча система контролю управління доступом має ряд функціональних можливостей:

1. Аутентифікація користувачів: СКУД перевіряє ідентифікаційні дані користувача (наприклад, картку або біометричні дані) для визначення його особи.

2. Налаштування прав доступу: Адміністратор системи може налаштовувати права доступу для різних користувачів, визначаючи, до яких зон або ресурсів вони мають доступ.

3. Моніторинг подій: СКУД веде журнал подій, фіксуючи кожен вхід і вихід користувача, а також всі інші події, пов'язані з системою.

4. Звітність: СКУД може генерувати звіти про активність користувачів, історію подій, часові проміжки доступу і іншу інформацію.

5. Адміністрування: Адміністратор системи може керувати всіма параметрами і налаштуваннями СКУД, додавати або видаляти користувачів, налаштовувати читачі та інше обладнання.

Існуюча СКУД може бути інтегрована з іншими системами безпеки та управління, такими як системи відеоспостереження, системи виявлення вторгнень, системи пожежної безпеки і інші. Інтеграція дозволяє створити цілісну систему безпеки, яка може ефективно виявляти, реагувати і реагувати на загрози.

Хоча існуючі системи контролю управління доступом мають багато переваг, вони також мають свої недоліки. Деякі з них включають обмежену можливість використання біометричних методів ідентифікації, несумісність зі стандартами безпеки інформації, складність налаштування та обслуговування.

Опис існуючої системи контролю управління доступом є важливим етапом у розробці нового програмного модуля. Розуміння функціональності та недоліків існуючої системи дозволяє розробникам створити більш ефективне та прогресивне рішення. Інтеграція біометричних методів ідентифікації в СКУД може покращити рівень безпеки і зручності для користувачів, а також підвищити ефективність контролю управління доступом.

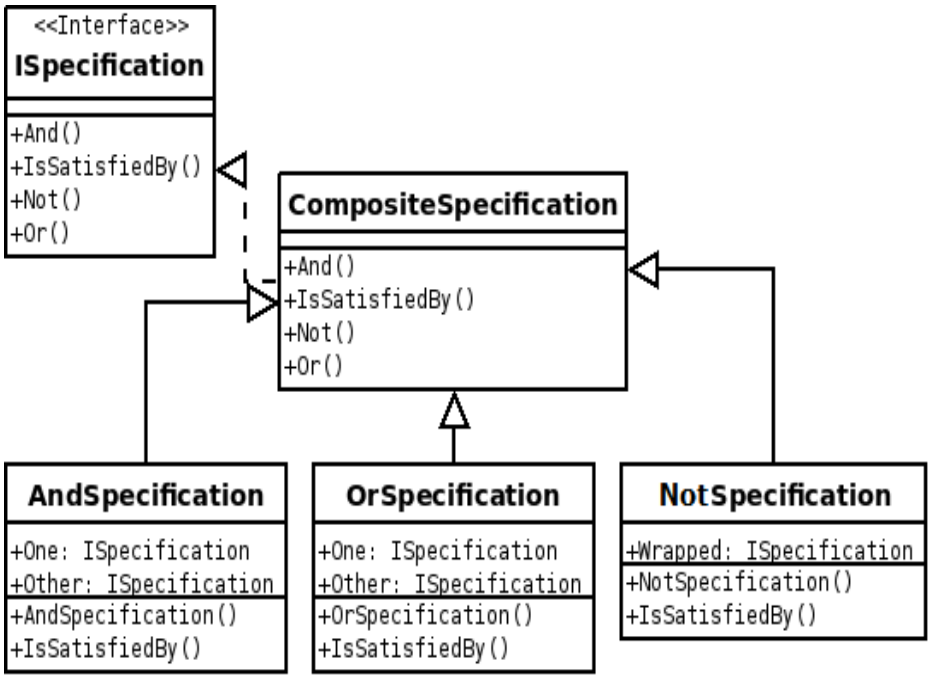

Рис. 3.1. Діаграма патерну *Specification*

Задачі розробки програмного коду можуть бути складними і вимагати структурованого та ефективного підходу до роботи з даними та бізнес-правилами. Шаблон специфікації - це інструмент, який здійснює структурування та абстрагування різних аспектів обробки даних, що дозволяє використовувати його в різних сценаріях в рамках програми. Нижче розглядаються більш детальні сценарії використання шаблону специфікації:

1. Під час застосування критеріїв фільтрації/пошуку: У великих наборах даних часто потрібно виконувати фільтрацію або пошук за певними критеріями. Шаблон специфікації дозволяє визначити ці критерії у вигляді об'єкта специфікації, що спрощує код і зроблює його більш зрозумілим та гнучким.

2. Вилучення бізнес-правил з коду: Бізнес-правила, які визначають, які дані допустимі і які - ні, можуть бути виокремлені в об'єкти специфікації. Це полегшує редагування цих правил без необхідності втручання у вихідний код програми.

3. Обробка журналів помилок: Шаблон специфікації може бути використаний для визначення правил обробки помилок. Він дозволяє зробити обробку помилок більш структурованою та контрольованою.

4. Проведення модульного тестування: Під час модульного тестування важливо перевірити правильність роботи окремих компонентів програми. Шаблон специфікації робить тести більш чіткими та легшими для створення, оскільки ви можете визначити вхідні критерії для модульних функцій.

5. Вибір компонента/конкретного об'єкта: Коли вам потрібно вибрати певний компонент або об'єкт зі списку на підставі певних критеріїв, специфікація може визначити ці критерії і спростити вибір, зробивши його більш структурованим.

6. Створення складної логіки аналізу: Під час складного аналізу даних або виконання операцій із збільшеною складністю, шаблон специфікації дозволяє виокремити цю логіку та зробити її більш структурованою та керованою.

Прикладом може служити поле ідентифікатора, що спільне для всіх об'єктів доменного рівня. Специфікація може визначити правила вибору об'єктів за

ідентифікатором та використовувати їх у всіх сценаріях, де потрібно взаємодіяти з об'єктами за ідентифікатором.

Загалом, шаблон специфікації сприяє структурованій та ефективній роботі з даними та бізнес-логікою в програмі, полегшує її підтримку та підвищує зрозумілість коду.

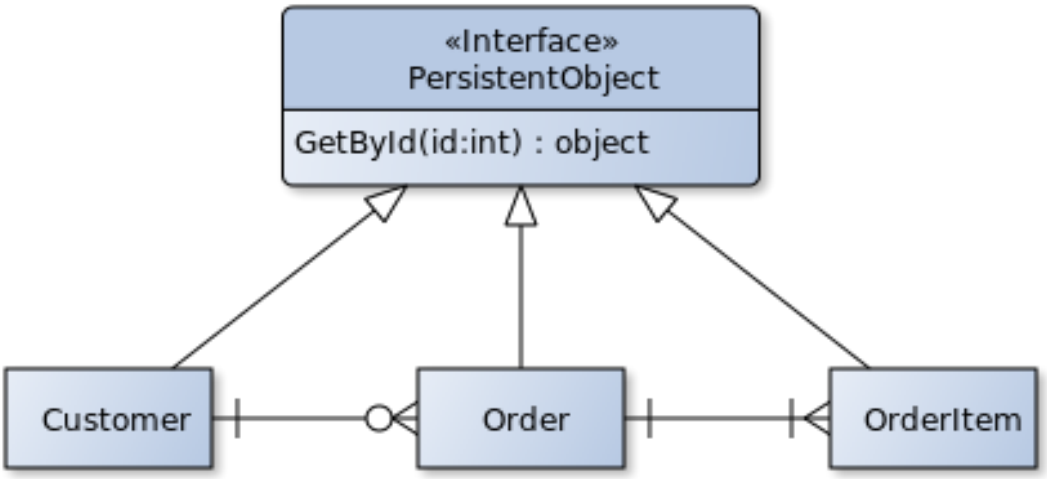

Рис. 3.2. Діаграма патерну *Layer Supertype*

Паттерн *Identity Map* зберігає записи про всі об'єкти, які були зчитані з БД під час виконання однієї дії. Коли відбувається звернення до об'єкта, перевіряється карта відповідності, щоб дізнатися, чи завантажено об'єкт.

Загальновизнаним терміном в об’єктно-орієнтованому програмуванні є *Factory*. Фабрика – це об’єкт, який несе повну відповідальність за створення інших об’єктів (рис. 3.3).

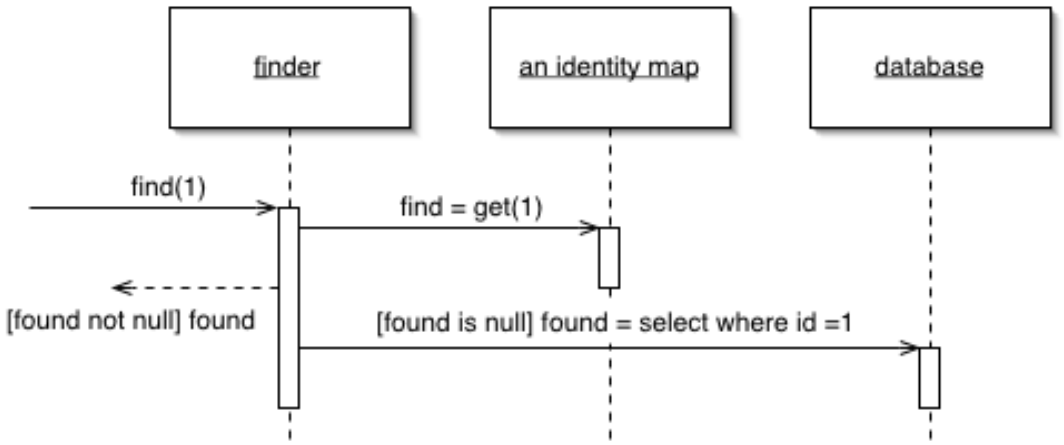

Рис. 3.3. Діаграма патерну *Identity Map*

Фабрики, безумовно, не є унікальними для доменно-орієнтованого дизайну, але вони відіграють важливу роль у доменно-орієнтованому проекті.

Фабрики часто використовуються для створення *Aggregate*. *Factory* може бути корисним під час створення нового *Aggregate,* оскільки він інкапсулює знання, необхідні для створення *Aggregate* у узгодженому стані та з усіма інваріантами.

Єдина мова *DDD* проекту не стосується безпосередньо процесу створення об'єктів. Проте розробники відповідають за технічне рішення для роботи з об'єктами домену програм. Це означає, що єдина мова проекту безпосередньо не посилається на фабрики, вони все одно утворюють важливу частину доменного рівня програми.

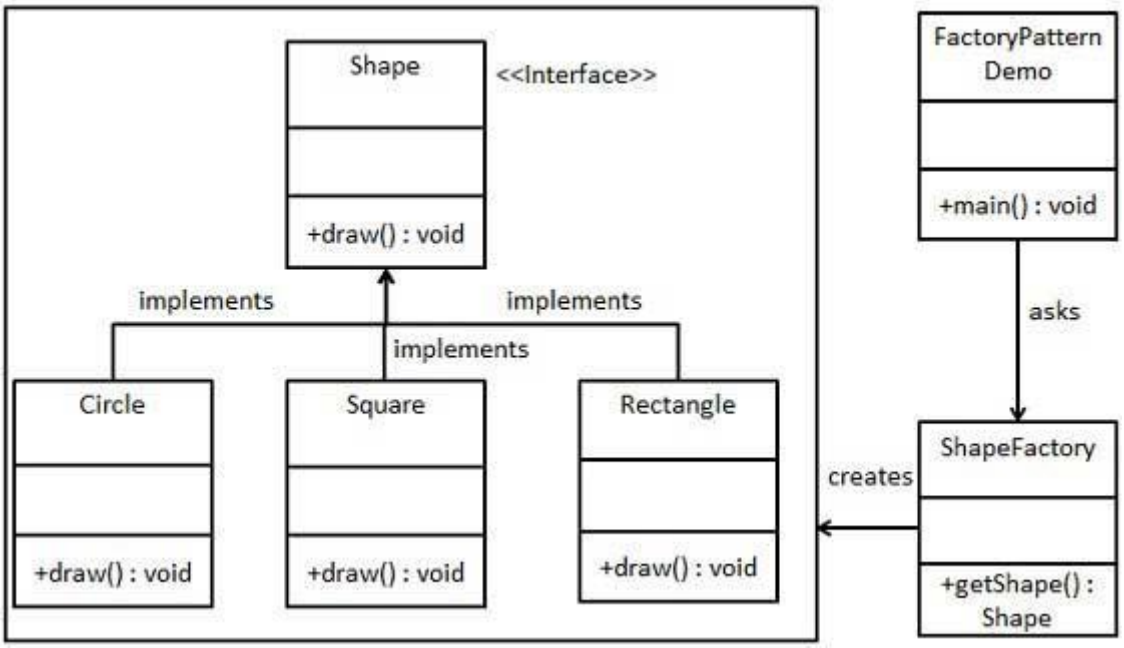

Рис. 3.4. Діаграма патерну *Factory*

Цей принцип покращує повторне використання коду та обмежує ефект хвилі, якщо потрібно змінити класи нижчого рівня. Але навіть якщо реалізація ідеальна, то все одно буде збережена залежність від класу нижчого рівня. Інтерфейс лише відокремлює використання класу нижчого рівня, але не його створення. У певному місці коду потрібно створювати екземпляр реалізації інтерфейсу. Це запобігає заміні реалізації інтерфейсу іншим.

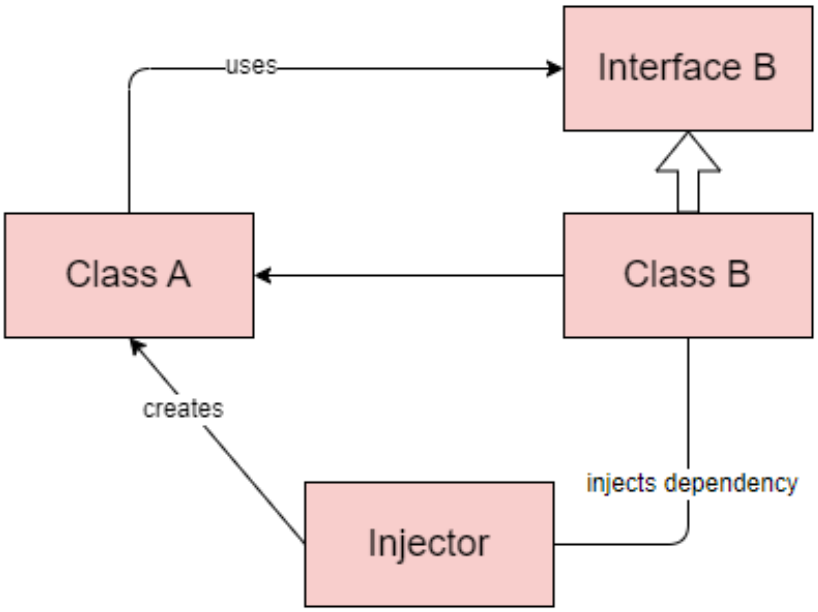

Рис. 3.5. Діаграма техніки *Dependency injection*

Оскільки код, згенерований *ORM,* добре протестований, не потрібно витрачати багато часу на тестування коду доступу до даних. Замість цього можна зосередитися на тестуванні бізнес-логіки та коду.

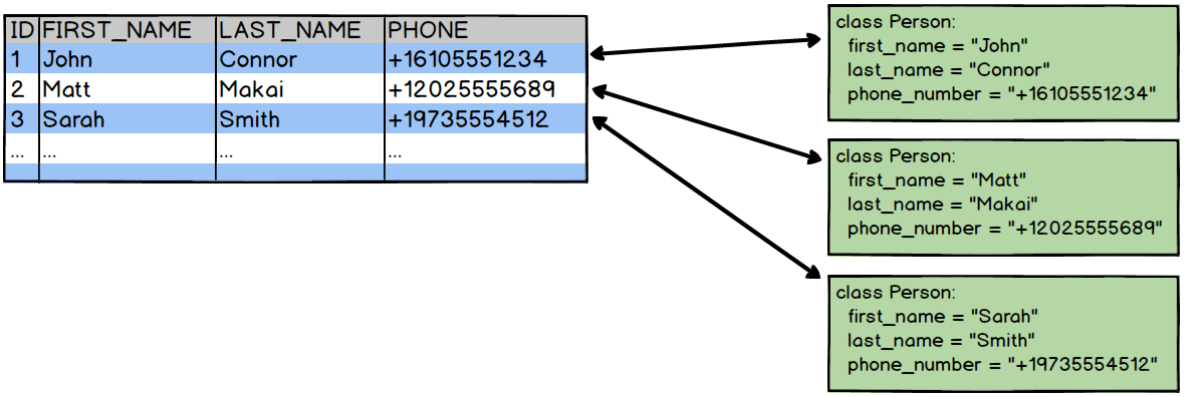

Рис. 3.6. Приклад використання *ORM*

Балансування навантаження може бути апаратним або програмним. Для роботи у відмовостійкому режимі потрібно використовувати балансир трафіку *HTTP/HTTPS,* який підтримує протокол *WebSocket*. *Creatio* протестовано на програмних балансувальниках навантаження *HAproxy* і *MS ARR* (*Microsoft Advanced Request Routing*). Відомі випадки успішного впровадження інших балансувальників, таких як *Citrix, Cisco, NginX, FortiGate*.

Модель або схема бази даних (рис. 3.7)– це відображення відношень між сутностями у певний момент часу, тому дуже важливо визначити як самі сутності, так і зв’язки між ними.

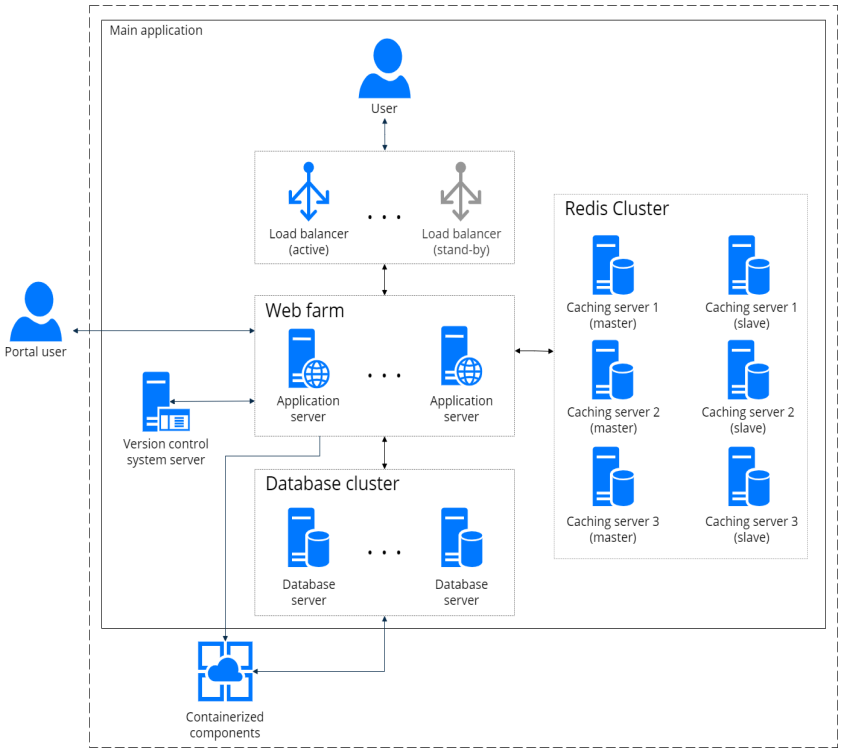

Рис. 3.7. Інфраструктура головного *CRM*-додатку

На рисунку 3.8 зображена діаграма використання програмної інтеграції разом з *Creatio*.

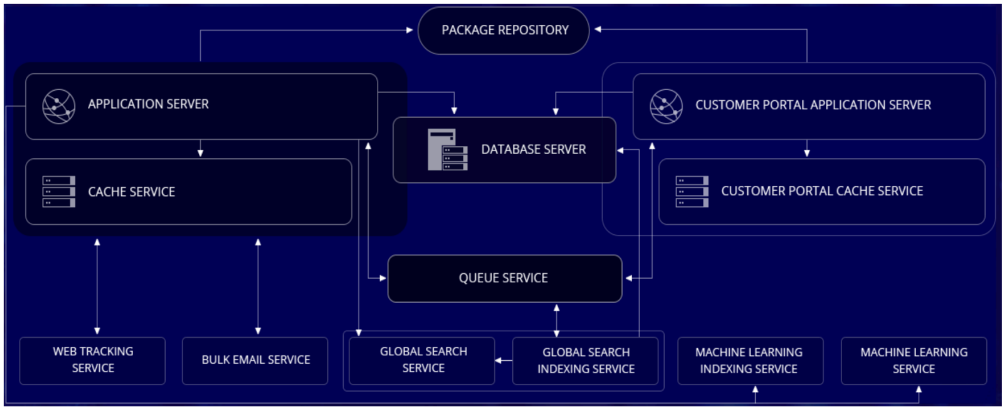

Рис. 3.8. Інфраструктура *Creatio*

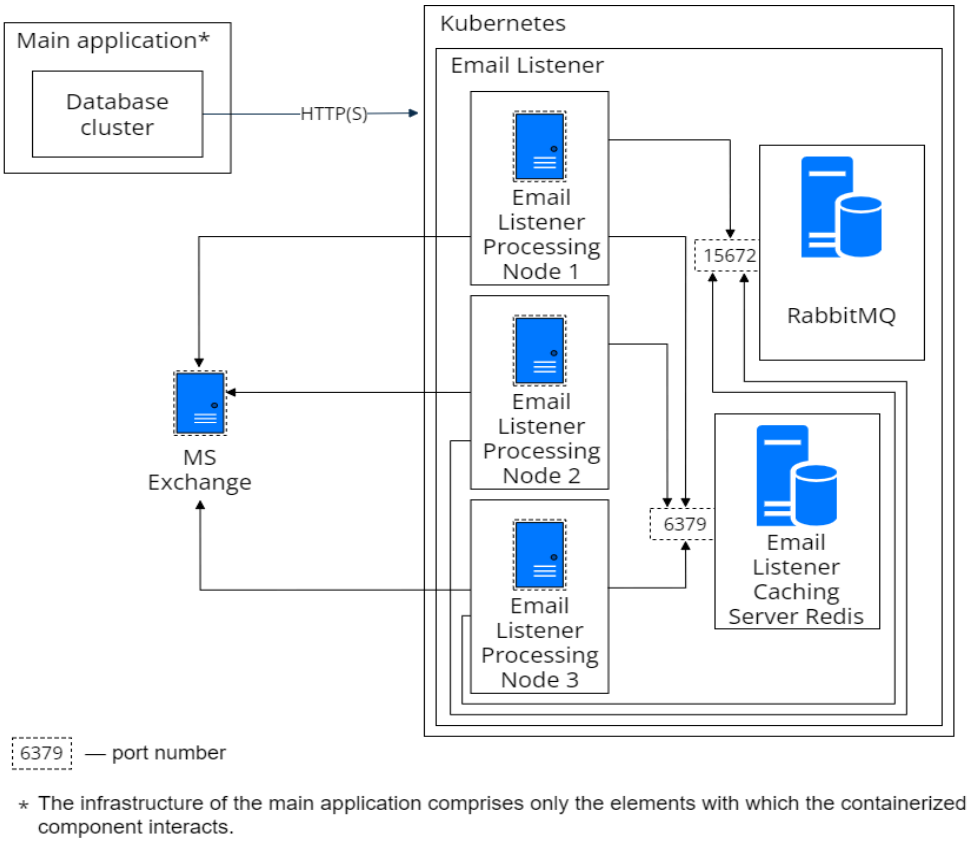

Рис. 3.9. Інфраструктура сервісу *Microsoft Exchange*

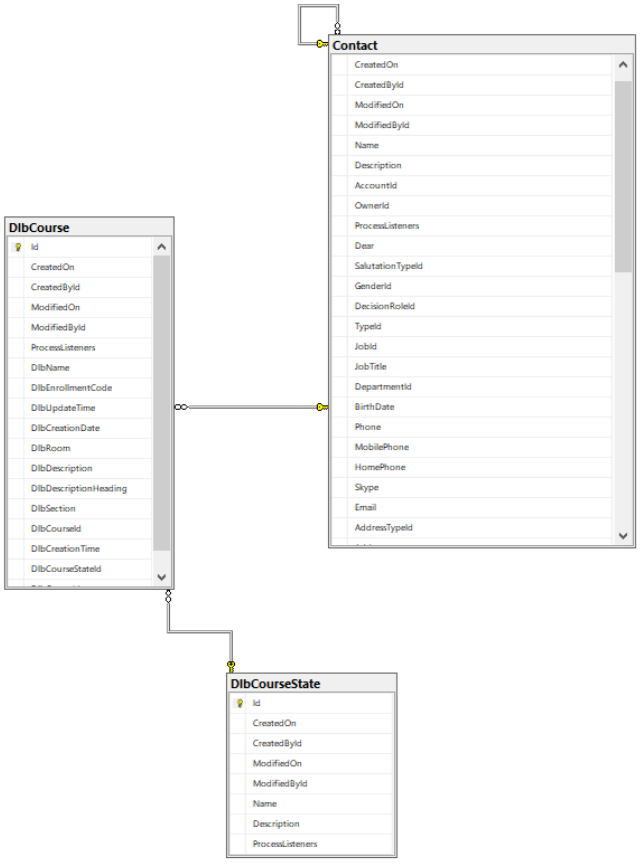

Рис. 3.10. Модель бази даних

Для розробки діаграми класів було використано *Class Designer* з *Visual Studio* 2022.

ї *PERCo*, яка

пеки власної

електронні

системи

Co проходять

ктробезпеку,

технологій

дозволяють *PERCo* розробляти, випускати і успішно представляти на ринку широкий асортимент товарів, що відповідають світовим стандартам в галузі обладнання і систем безпеки.

Електронні прохідні – це готові рішення, які дозволяють найпростішим і зручним способом організувати систему контролю доступу в установах і на підприємствах. Електронні прохідні виглядають як турнікети, але включають в себе всю електроніку систем контролю доступу.

Електронна прохідна *PERCo – KT* 02.3 (далі – ЕП) призначена для організації однієї двосторонньої точки проходу на територію підприємства.

Контроль доступу через ЕП здійснюється оператором за допомогою пульта дистанційного керування, що входить в комплект поставки або, після додаткової настройки з використанням ПЗ *PERCo–S*-20, за безконтактними картками доступу.

Склад:

– турнікет;

– вбудований контроллер СКУД;

– два вбудованих зчитувача безконтактних карт доступу (*HID* / *EM – Marin*);

– пульт дистанційного керування;

– програмне забезпечення *PERCo – SL* 01 «Локальне ПЗ».

Залежно від завдань підприємства вона може працювати як:

– самостійна система контролю доступу на 1 точку проходу;

– частина системи контролю доступу, яка обслуговує декілька точок проходу;

– автономний турнікет, керований оператором від пульта дистанційного керування.

Електронна прохідна КТ02.3 має можливість прямого підключення до комп'ютера або до локальної обчислювальної мережі підприємства (мережі *Ethernet*) для введення даних і отримання звітів.

У стандартному комплекті Електронної прохідної поставляється безкоштовне програмне забезпечення *PERCo – SL* 01 для організації набору функцій системи контролю доступу (рис. 3.12).

Так само існує *PERCo – SL* 02 «Локальне ПЗ з відеоідентифікації», яке дозволяє організувати захист від передачі перепустки іншій особі. На моніторі охоронця відображається фото власника карти, пред'явленої зчитувача. Охоронець має можливість порівняти фото з бази даних системи контролю доступу та особу пред'явника безконтактної карти або його зображення, якщо встановлена відеокамера.

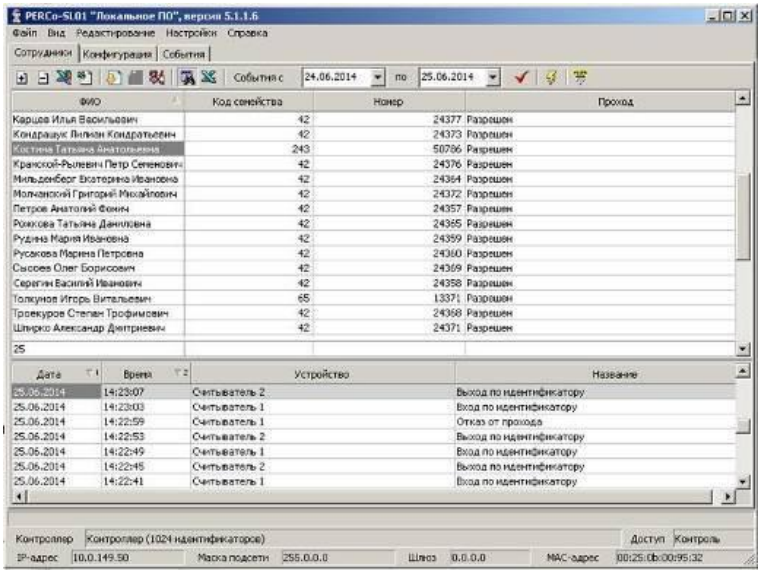

Рис. 3.12. Безкоштовне програмне забезпечення *PERCo – SL* 01

Установка мережевого ПЗ комплексної системи безпеки *PERCo – S* -20 дозволяє на базі електронної прохідний вирішувати завдання безпеки і

підвищення ефективності роботи підприємства. Крім контролю доступу можна організувати контроль порушень трудової дисципліни, автоматизація обліку робочого часу і розрахунку заробітної плати.

КТ02.3 може застосовуватися спільно з картоприймачем безконтактних карт доступу, дозволяючи організувати вилучення карт відвідувачів.

У моделі *КТ* 02.3 до вбудованого в стійку турнікета контролера можна підключити до 8-ми контролерів СКУД *PERCo – CL* 201 з вбудованим зчитувачем, що забезпечує економічне устаткування 8-ми приміщень системою контролю доступу. Для формування зони проходу *КТ* 02.3 може бути доповнена секціями огорожі. Вбудований замок механічного розблокування дозволяє відкрити турнікет за допомогою ключа, забезпечивши вільне обертання планок в обох напрямках.

3.2. Архітектура програмного забезпечення інтеграції з СКУД

Важливим аспектом інтеграції є забезпечення спільної роботи різних компонентів системи, ефективне обмін даними та збереження цілісності і безпеки інформації.

Метою інтеграції є створення програмного модуля, який дозволить розширити функціональність існуючої СКУД за допомогою біометричних методів ідентифікації. Головні завдання інтеграції включають:

1. Захоплення біометричних даних: Забезпечення можливості збору біометричних даних користувачів, таких як відбитки пальців, обличчя, голос тощо.

2. Аутентифікація користувачів: Забезпечення можливості використовувати біометричні дані для аутентифікації користувачів в СКУД.

3. Інтеграція з базою даних: Забезпечення спільної роботи програмного модуля з базою даних СКУД для збереження біометричних даних та інформації про користувачів.

4. Обробка та збереження даних: Забезпечення можливості обробки та збереження біометричних даних в безпечному форматі.

5. Забезпечення безпеки даних: Захист біометричних даних від несанкціонованого доступу і забезпечення їхньої конфіденційності.

Архітектура програмного модуля інтеграції з СКУД складається з декількох ключових компонентів:

1. Користувацький інтерфейс: Ця складова взаємодіє з операторами та адміністраторами СКУД і надає можливість налаштовувати параметри інтеграції, а також виконувати моніторинг та управління процесом.

2. Модуль збору біометричних даних: Цей модуль відповідає за захоплення біометричних даних, таких як відбитки пальців, обличчя, голос і інші. Він взаємодіє зі спеціальним обладнанням (считувачі, камери, мікрофони) для отримання цих даних.

3. Модуль обробки біометричних даних: Ця складова виконує обробку та аналіз біометричних даних для визначення ідентичності користувача. Вона може використовувати алгоритми розпізнавання обличчя, голосу, відбитків пальців і інші.

4. Модуль інтеграції з СКУД: Цей модуль відповідає за взаємодію з існуючою СКУД. Він передає інформацію про користувачів та результати біометричної ідентифікації в СКУД і забезпечує синхронізацію даних.

5. База даних біометричних даних: Ця база даних зберігає біометричні дані користувачів, які можуть бути використані для подальших порівнянь та ідентифікації.

Інтеграція з СКУД базується на кількох ключових принципах:

1. Стандартизація даних: Забезпечення використання стандартизованих форматів та протоколів для обміну інформацією між програмним модулем і СКУД.

2. Захист даних: Забезпечення безпеки біометричних даних під час їх передачі та зберігання в системі СКУД.

3. Інтеграція з базою даних: Можливість синхронізації бази даних біометричних даних з базою даних СКУД для одноразового управління користувачами та їхніми правами доступу.

4. Логування та моніторинг: Ведення журналів подій для відстеження процесу інтеграції та моніторингу її результатів.

3.2.5. Переваги інтеграції

Інтеграція біометричних методів ідентифікації з СКУД має численні переваги:

1. Підвищена безпека: Використання біометричних даних ускладнює можливість несанкціонованого доступу, оскільки біометричні характеристики унікальні для кожної особи.

2. Зручність користувачів: Біометричні методи ідентифікації забезпечують швидкий та зручний доступ для користувачів без потреби в картках чи інших носіях.

3. Висока точність ідентифікації: Біометричні методи зазвичай володіють високою точністю ідентифікації, що робить їх ефективними для застосування в системах контролю доступу.

4. Можливість використання для двофакторної аутентифікації: Біометричні дані можуть бути використані разом з іншими методами аутентифікації, такими як паролі, для підвищення рівня безпеки.

Архітектура програмного модуля інтеграції з СКУД включає в себе ряд компонентів та принципів, які дозволяють успішно об'єднати біометричні методи ідентифікації з існуючою системою контролю управління доступом. Інтеграція біометричних методів підвищує рівень безпеки та зручності для користувачів і може бути важливим кроком у розвитку системи безпеки організації.

3.3. Збереження об’єктів

Збереження об’єкта в файл. Деякі об’єкти відео системи вимагають збереження на диск і подальшого відновлення. Для цього застосовується механізм

серіалізациі (перетворення об’єкта в послідовну форму і назад). Принцип роботи цього механізму наступний:

– при виконанні функції збереження в файл (*Save*) об’єкт насамперед записує рядок з ім’ям свого класу. Після цього відбувається виклик функції сериализации (*Serialize*) з параметром «зберегти»;

– у цій функції об’єкт здійснює запис на диск всіх своїх атрибутів. Після цього об’єкт вважається збереженим;

– при виконанні функції проштовхування об’єктів (*Load*) з файлу зчитується ім’я класу завантажується об’єкта. Цей рядок передається механізму параметризовано створених об’єктів, який здійснює створення відповідного об’єкта. Після чого викликається функція *Serialize* з параметром «завантажити».

У функції *Serialize* відбувається зчитування з файлу атрибутів об’єкту, що зберігається. Таким чином, ми можемо зберігати в одному файлі довільне число об’єктів. Однак, щоб отримати доступ до будь-якого об’єкта, нам доведеться послідовно завантажити всі попередні об’єкти. Тому серіалізацію зручно використовувати лише для збереження стану одиночних об’єктів, або груп об’єктів, які використовуються одночасно. На рис. 3.13 представлена діаграма класів, що відображає структуру даного механізму.

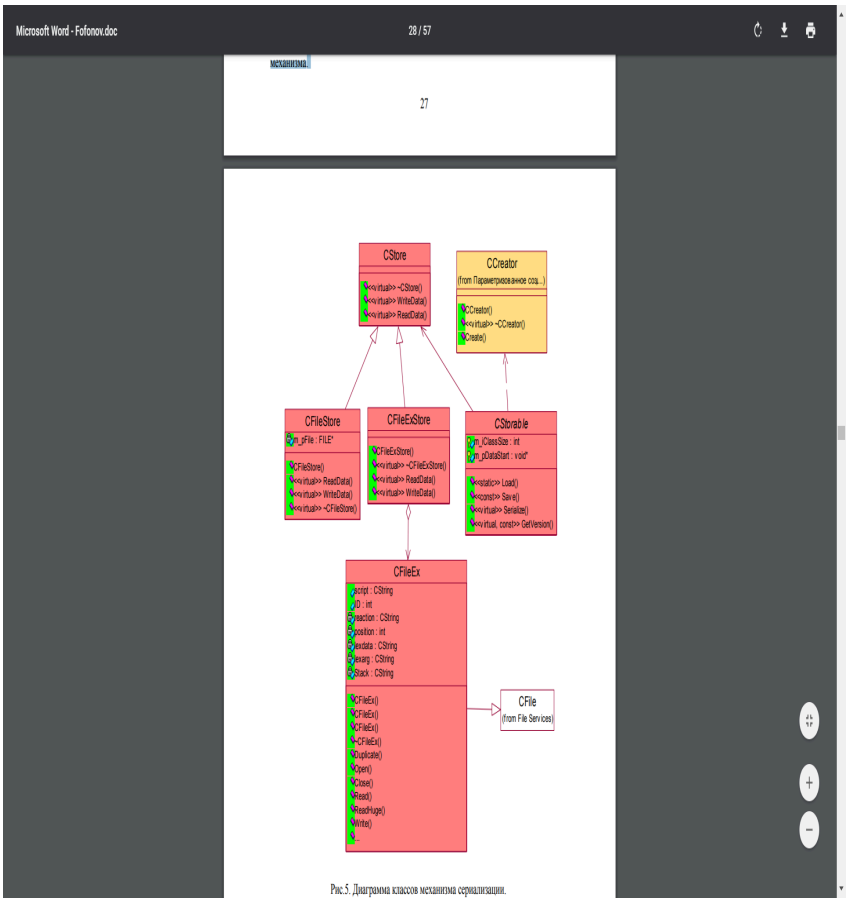

Рис. 3.13. Діаграма класів механізму серіалізації

Для використання цього механізму всі класи, які бажають, щоб їх об’єкти зберігалися, повинні наслідувати від класу, що задає інтерфейс для серіалізації (*CStorable,* модуль *Storable*. *Cpp* (*h*)). Клас *CStore* задає інтерфейс для використання різних типів файлових сховищ. Наприклад, клас *CFileStore* використовує для роботи з файлами функції *API,* а клас *CFileExStore* – функції класу *CFileEx* (реалізація в модулі З *Store*. *H*). Крім них можуть бути використані будь-які класи-обгортки, що забезпечують роботу з файлами.

Цей механізм є повністю об’єктно-орієнтованим, так як тільки сам об’єкт знає порядок завантаження / збереження своїх атрибутів. Серіалізация реалізована і використовується в стандартній бібліотеці класів *MFC*.

Однак *MFC* -реалізація цього механізму може бути неприйнятна в випадках, коли застосовується множинне успадкування або передбачається забезпечити кросплатформеність додатків. В [11] описана альтернативна реалізація цього підходу, без залучення *MFC*. Вона вимагає використання механізму параметризрвані створення об’єктів, який буде описаний окремо. Крім цього,

потрібно, щоб компілятор умів визначати інформацію про клас об'єкту під час виконання. Щоб прискорити процес запису об'єкта в файл, було прийнято наступне рішення. Замість того щоб записувати на диск по-черзі всі свої атрибути, об'єкт записує себе цілком шляхом копіювання області пам'яті.

Цей спосіб дозволяє заощадити на часі записи, але при цьому копіюються не тільки атрибути, а й таблиці віртуальних функцій об'єкта і всіх його базових класів. Однак таблиці віртуальних функцій об'єкта змінюються лише від компіляції до компіляції. Даний підхід може застосовуватися лише у випадках, коли необхідно проводити постійні збереження об'єктів.

Збереження в файл не дозволяє нам організувати швидкий пошук та вилучення об'єктів. Для вирішення цього завдання був розроблений спеціалізований клас *CStock,* що є контейнером одно- або двуключевих записів і реалізує можливості збереження і пошуку об'єктів (модуль С *Stock*. *Cpp* (*h*)).

Склад може зберігати тільки об'єкти одного типу. Для цього він ініціалізується ім'ям класу збережених об'єктів, положенням і довжиною ключів в запису, що характеризує об'єкт. Для розміщення об'єктів на склад використовується механізм, схожий на серіалізацію. Кожен об'єкт, який бажає зберігатися на складі, повинен мати функції, що задаються інтерфейсом *CStockItem* (модуль *CStockItem*. *Cpp* (*h*)).

При додаванні об'єкта на склад, викликається функція перетворення (*Transform*), яка за аналогією з функцією *Serialize* здійснює запис атрибутів об'єкта та їх вилучення. Однак в якості сховища атрибутів тут виступає не файл, а рядок (область пам'яті). Отриманий рядок додається в таблицю елементів і включається в індекс відповідно до свого ключем (ключами).

Клас *CStock* має два індекси (по одному на кожен ключ), які використовуються для швидкого пошуку елементів. Пошук елементів по індексам здійснюється за допомогою дихотомії, з використанням лексикографічного порівняння, що дає можливість використання в якості ключів будь-яких типів, в тому числі і рядків.

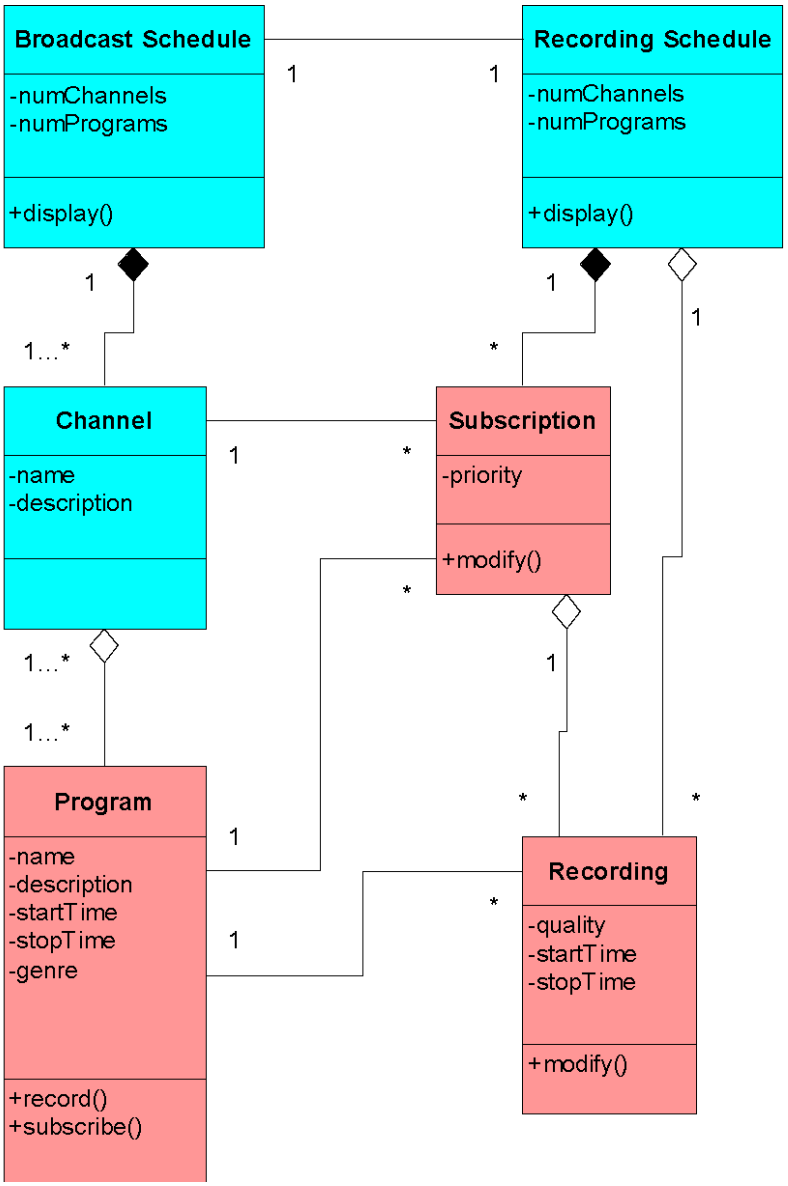

Рис. 3.14. Діаграма класів механізму складування об’єктів

Реалізовано можливості вилучення елемента зі складу з блокуванням записи і без блокування. Для створення об’єктів видобутих елементів використовується механізм параметризації конструювання об’єктів.

Загальна послідовність дій виглядає наступним чином: – створюється новий об’єкт класу CFactory, параметризовані класом об’екта продукту. При цьому автоматично здійснюється створення нового елемента CFactoryListItem і його додавання до існуючого списку фабрик; – викликається функція Create класу CCreator, якій, як параметр, передається рядок, що містить ім’я класу створюваного об’єкта. Клас CCreator здійснює перегляд списку фабрик і викликає функцію створення продукту для кожної фабрики; – фабрика перевіряє, чи

співпадає переданий параметр з ім’ям класу її продукції, і в залежності від цього створює чи ні об’єкт. Для створення об’єкта використовується конструктор за замовчуванням; – як тільки чергова фабрика повертає непорожнє значення, клас CCreator перериває перегляд списку фабрик і, в свою чергу, повертає створений об’єкт. Якщо жодна фабрика не змогла створити запитуваний об’єкт, то повертається порожній покажчик. Для коректної роботи з продуктами, які використовують множинне спадкування, необхідно здійснити динамічне перетворення створеного об’єкта до його типу. Об’єкти-фабрики можна створювати один раз при старті програми і знищувати при її завершенні. З іншого боку, якщо параметризрвані створення об’єктів проводиться рідко, то фабрики можна створювати і видаляти по мірі необхідності.

3.4. Параметризоване створення об’єктів

Для параметризації створюваного об’єкта ім’ям класу використовувався механізм, описаний в шаблоні проектування «Абстрактна фабрика». Застосовуваний підхід також описаний в [11].

Клас *CAbstractFactory* задає загальний інтерфейс для створення продукту (об’єкта) різними фабриками об’єктів. Клас *CFactory* ініціалізується створюваним продуктом і реалізує інтерфейс абстрактної фабрики. Клас *CFactoryListItem* є елементом списку фабрик. Клас *CCreator* здійснює перебір списку фабрик для створення конкретного продукту, визначеного переданим йому параметром. Діаграма класів представлена на рис. 3.15.

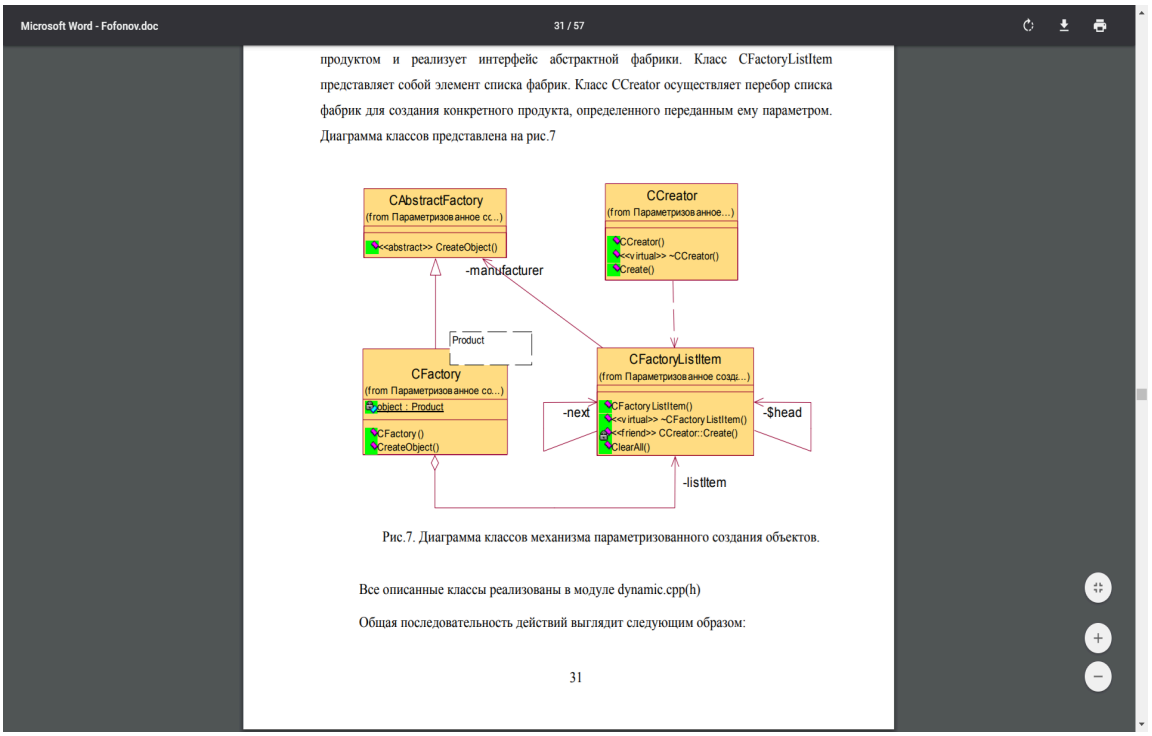

Рис. 3.15. Діаграма класів механізму параметризрвані створення об’єктів

На рисунку представлена діаграма класів, яка ілюструє реалізацію шаблону проектування "Абстрактна фабрика" для параметризованого створення об'єктів. Цей шаблон дозволяє генерувати сімейство пов'язаних або залежних об'єктів без конкретизації їхніх класів. Ось детальний опис основних компонентів цієї системи:

1. CAbstractFactory:

– Це абстрактний клас, який оголошує інтерфейс CreateObject(), визначаючи метод для створення об'єктів. Він діє як загальний шаблон для фабрик, які вироблятимуть продукти (об'єкти).

2. CFactory:

– Це конкретна реалізація CAbstractFactory, що ініціалізується об'єктом Product. Вона перевизначає метод CreateObject() для створення екземплярів продукту. Об'єкт, який вона створює, визначається типом продукту, асоційованим з цією фабрикою.

3. CFactoryListItem:

– Цей клас представляє елемент у зв'язаному списку фабрик, де кожен елемент містить посилання на наступний елемент зі списку (-next). Він також має

методи, такі як конструктор (CFactoryListItem()), деструктор (~CFactoryListItem()) і ClearAll(), які дозволяють управляти елементами списку та очищати список.

4. CCreator:

– Цей клас виконує роль директора, керуючи створенням об'єктів. Він використовує метод Create(), щоб перебирати елементи CFactoryListItem у списку і створювати продукт з використанням конкретної фабрики, визначеної переданим параметром.

На діаграмі зображено взаємозв'язки між цими класами. CFactory наслідується від CAbstractFactory і реалізує його метод CreateObject(). CFactoryListItem включає в себе об'єкти CFactory, і він з'єднаний із CCreator, який використовує ці елементи для створення продуктів.

Ця система дозволяє створювати об'єкти, використовуючи ім'я класу як параметр, що робить її гнучкою та здатною адаптуватися до змін у вимогах до типів продуктів, які мають бути створені, без зміни існуючого коду клієнтів.

Всі описані класи реалізовані в модулі *dynamic*. *cpp* (*h*) Загальна послідовність дій виглядає наступним чином:

– створюється новий об’єкт класу *CFactory,* параметризовані класом об’єкта продукту. При цьому автоматично здійснюється створення нового елемента *CFactoryListItem* і його додавання до існуючого списку фабрик;

– викликається функція *Create* класу *CCreator,* якій, як параметр, передається рядок, що містить ім’я класу створюваного об’єкта. Клас *CCreator* здійснює перегляд списку фабрик і викликає функцію створення продукту для кожної фабрики;

– фабрика перевіряє, чи співпадає переданий параметр з ім’ям класу її продукції, і в залежності від цього створює чи ні об’єкт. Для створення об’єкта використовується конструктор за замовчуванням;

– як тільки чергова фабрика повертає непорожнє значення, клас *CCreator* перериває перегляд списку фабрик і, в свою чергу, повертає створений об’єкт. Якщо жодна фабрика не змогла створити запитуваний об’єкт, то повертається порожній покажчик.

Для коректної роботи з продуктами, які використовують множинне спадкування, необхідно здійснити динамічне перетворення створеного об’єкта до його типу. Об’єкти-фабрики можна створювати один раз при старті програми і знищувати при її завершенні. З іншого боку, якщо параметризрвані створення об’єктів проводиться рідко, то фабрики можна створювати і видаляти по мірі необхідності.

3.5. Взаємодія додатків

В архітектурі "клієнт-сервер" додатки-клієнти взаємодіють з сервером шляхом надсилання запитів, а сервер відповідає на ці запити, виконуючи відповідні дії. Ця архітектура використовується в розподілених системах, де різні компоненти програми можуть працювати на різних серверах або пристроях.

Часто до сервера висуваються вимоги організації протоколювання цих запитів. Організація та контроль цього процесу важливі для ефективного функціонування системи. Саме для цього був розроблений та використаний шаблон проектування "Команда".

Шаблон "Команда" інкапсулює запит як об'єкт. Це означає, що запити можуть бути представлені як об'єкти зі своєю власною логікою та параметрами. Такий підхід дозволяє:

1. Задавати параметри клієнтів: Клієнти можуть налаштовувати параметри запитів, встановлювати аргументи та інші деталі, які впливають на обробку запиту на сервері.

2. Ставити запити в чергу або протоколювати їх: Запити можуть бути створені, а потім відправлені на виконання у зручний час, або ж їхнє виконання може бути зареєстроване та протокольоване для подальшого аналізу.

3. Підтримувати скасування операцій: В деяких випадках клієнти можуть бажати скасувати виконання запиту. Шаблон "Команда" дозволяє реалізувати цю можливість шляхом створення спеціальних команд для скасування виконання запитів.

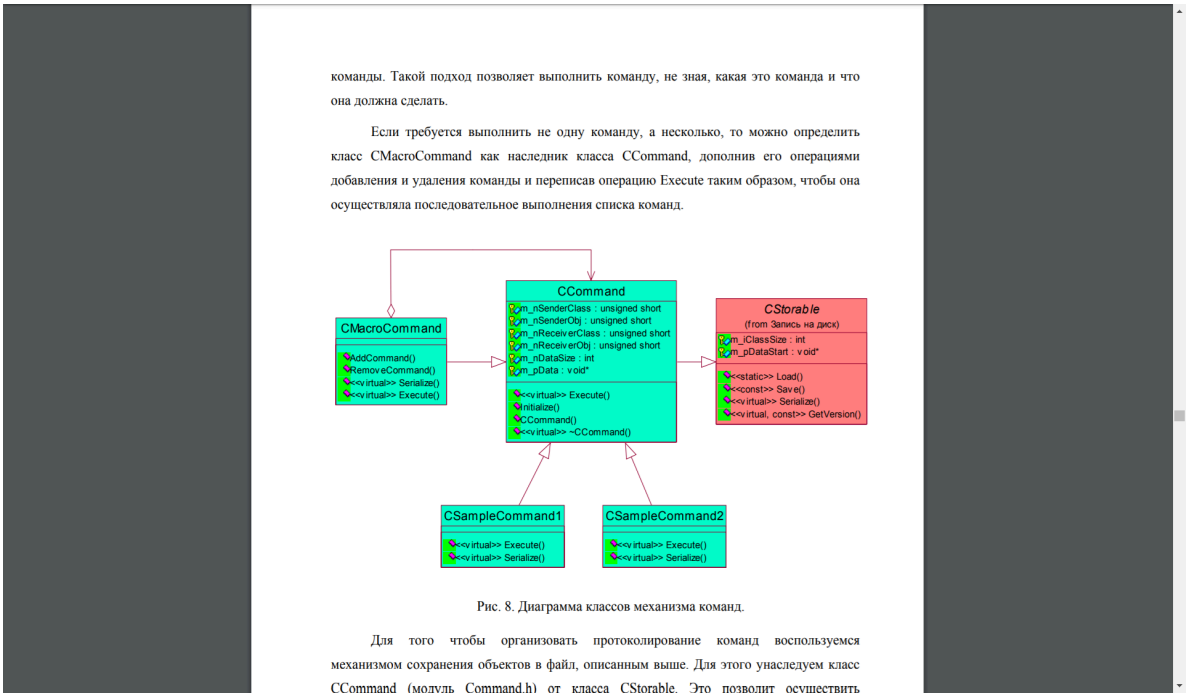

Рис. 3.16. Діаграма класів механізму команд

Такий підхід до взаємодії клієнтів та серверів дозволяє створити більш гнучкі та розширювані системи. Клієнти можуть взаємодіяти з сервером без прямого виклику конкретних методів, що дозволяє розділити функціональність між клієнтом і сервером та забезпечує легше втручання в процес взаємодії.

Застосування шаблону "Команда" у контексті архітектури "клієнт-сервер" допомагає створити більш зручну, керовану та ефективну систему обробки запитів, що є важливим аспектом у розробці розподілених програмних систем.

Схема візуалізує структуру класів та їхні відносини в рамках програмного модуля. Ось опис кожного з класів і їх функцій, згідно з тим, що видно на діаграмі:

1. CCommand:

– Це базовий клас для команд, що містить метадані, такі як m_nSenderClass, m_nSenderObj, m_nReceiverClass, m_nReceiverObj, m_nDataBase і m_pData, які, ймовірно, використовуються для ідентифікації відправника та одержувача, а також інформації для виконання команди.

– Функції: Initialize() та Execute(), що виконують ініціалізацію та виконання команд відповідно.

2. CMacroCommand:

– Це клас, що, імовірно, дозволяє комбінувати декілька CCommand в макро-команду.

– Функції: AddCommand() та RemoveCommand() для управління підкомандами та Execute() для їх виконання.

3. CStorable:

– Цей клас, судячи з назви, використовується для збереження даних.

– Функції: Load() та Save() для завантаження та збереження даних відповідно, Serialize() для перетворення даних в формат, що можна зберігати, та GetVersion(), що, можливо, повертає версію об'єкту або структури даних.

4. CSampleCommand1 і CSampleCommand2:

– Це конкретні реалізації CCommand, які надають власні версії Execute() та Serialize().

На основі цих класів, програмний модуль моніторингу доступу до закритих корпоративних систем може включати наступні можливості:

– Відправлення та прийом команд між різними компонентами системи.

– Виконання послідовностей команд, що дозволяє автоматизувати процеси моніторингу.

– Збереження стану моніторингу для подальшого аналізу або аудиту.

– Можливість розширення команд для специфічних потреб моніторингу через наслідування базового класу CCommand.

Ця діаграма є ключовою для розуміння того, як модуль буде структурований та яким чином він буде обробляти команди та зберігати дані.

3.5.1. Пакет «Система зв’язку сервера»

Розглянемо склад і принципи роботи системи зв’язку додатку з сервером в СКУД (рис. 3.17).

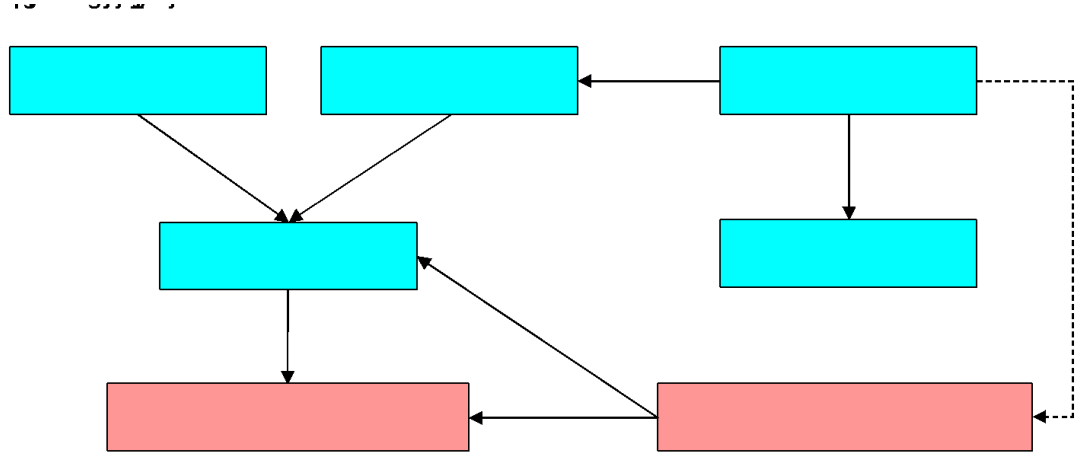

Рис. 3.17. Пакет «Система зв’язку сервера»

На діаграмі представлено частину системи зв'язку сервера, яка є складовою системи контролю та управління доступом (СКУД). Описані класи й їх зв'язки допомагають зрозуміти, як додаток взаємодіє з сервером. Ось докладний опис елементів діаграми та їх функцій:

1. CSPDevice:

– Це клас, що представляє пристрій зв'язку в системі, який може бути відповідальним за взаємодію з фізичними пристроями або інтерфейсами.

2. CAppSocket:

– Клас сокета додатку, який забезпечує мережеву точку з'єднання для відправлення та прийому даних до та від сервера.

3. CListeningSocket:

– Сокет для прослуховування, який очікує на вхідні з'єднання від клієнтів або інших компонентів системи.

4. CCommObject:

– Цей клас є центральним елементом системи зв'язку, який координує взаємодію між пристроями та сокетами.

5. CCommManager:

– Керує всіма аспектами комунікації в системі, забезпечуючи ефективне управління ресурсами зв'язку та розподіл з'єднань.

6. CIntegrationObject:

– Клас, який, можливо, відповідає за інтеграцію зовнішніх систем або модулів з основною системою зв'язку.

7. CApp:

– Представляє основний додаток або логічний модуль системи, який використовує засоби зв'язку для обробки даних та взаємодії з сервером.

Принципи роботи системи можуть бути такими:

– CApp та CSPDevice взаємодіють з CCommObject для здійснення зв'язку. CApp використовує CAppSocket для створення з'єднання з сервером, в той час як CSPDevice може забезпечувати з'єднання з фізичними пристроями.

– CListeningSocket працює як слухач для вхідних з'єднань, переказуючи їх до CCommManager.

– CCommManager управляє всіма вхідними та вихідними з'єднаннями, розподіляючи ресурси та забезпечуючи ефективне використання мережі.

– CIntegrationObject забезпечує інтеграцію з іншими системами або модулями, які потрібно включити в систему зв'язку СКУД.

Ця структура дозволяє системі зв'язку сервера СКУД ефективно управляти даними, що передаються між клієнтами (наприклад, системами контролю доступу) та сервером, забезпечуючи безпеку та надійність зв'язку.

3.5.2. Структурні об’єкти

Всі структурні елементи системи контролю доступу мають спеціальних агентів, які представляють інтереси даних елементів всередині програми-сервера. Взаємодія між агентами та іншими об’єктами сервера відбувається за допомогою диспетчера (інтерфейс *CInteractionObject*). Для забезпечення надійності системи структурні об’єкти зберігаються в файл (інтерфейс *CStorable*). У разі необхідності можна передати об’єкт з усіма його даними іншому додатку у вигляді рядка (інтерфейс *CTransformable*).

Рис. 3.18. Пакет «Структурні об’єкти»

На діаграмі відображено структуру структурних об'єктів в системі контролю доступу (СКУД). Кожен клас і його зв'язки відіграють роль в управлінні та представленні різних компонентів системи в програмі-сервері. Ось детальний опис елементів діаграми:

1. CSystem:

– Це може бути основний клас, який представляє загальну систему контролю доступу.

2. CStructObject:

– Це клас, що представляє структурний елемент системи, наприклад, вузол або компонент СКУД, який має власного агента в програмі-сервері.

3. CApp:

– Цей клас представляє програму або додаток, що взаємодіє зі структурними об'єктами.

4. CResource, CZone, CTurnstile, CDirection:

– Ці класи представляють ресурси системи, зони контролю доступу, турнікети та напрямки руху через турнікети відповідно. Вони є конкретними реалізаціями структурних об'єктів.

5. CController:

– Цей клас, ймовірно, відповідає за логіку управління доступом, інтегруючи різні структурні компоненти та управляючи їх взаємодією.

6. CInteractionObject:

– Інтерфейс для взаємодії структурних об'єктів з іншими компонентами системи. Це може включати комунікацію з диспетчером, який керує взаємодією між агентами.

7. CStorable:

– Інтерфейс, що визначає функціональність для збереження структурних об'єктів у файл, що забезпечує надійність системи через можливість відновлення стану після збоїв або перезапусків.

8. CTransformable:

– Інтерфейс, який може вказувати на можливість трансформації структурного об'єкта, можливо, для адаптації до змін у системі або змінних умов контролю доступу.

Кожен структурний об'єкт має агента, який діє всередині програми-сервера, представляючи інтереси цього об'єкта. Вони спілкуються з іншими об'єктами сервера через CInteractionObject, що дозволяє централізовано управляти різними взаємодіями. Збереження структурних об'єктів через CStorable забезпечує, що система може підтримувати свою становище та інформацію про стан в умовах, коли можуть виникати непередбачені збої.

Ця структура допомагає в організації ефективної та надійної системи контролю доступу, яка може масштабуватися та адаптуватися до змінних вимог безпеки.

Конфігурація процесів – якщо ми визначилися, що генерацію слід вести по трейсах, то й конфігуратор процесу нам теж варто зробити в просторі трейсів. Як домовилися вище, нотація BPMN нам не підходить через її важкість. Тому ми вирішили розробити власний легковажний формат нотації, який би оптимально підходив для наших цілей.

Цей формат має бути інтуїтивно зрозумілим та легким для інтеграції з існуючими інструментами візуалізації та аналізу трейсів. Він повинен підтримувати визначення залежностей між процесами, умовну логіку та паралельне виконання, але при цьому залишатися достатньо гнучким, щоб

вміщувати непередбачені варіанти використання, які можуть виникнути в динамічних системах.

Окрім цього, конфігуратор процесу має включати можливість визначення тригерів, що запускають процеси, і визначення кінцевих точок, де процеси мають завершуватися. Важливим є також забезпечення зворотного зв'язку про стан процесу – для цього ми можемо використовувати систему подій, що відстежуються, і візуалізувати їх через дашборди.

Конфігурація процесу в просторі трейсів також має враховувати потребу в аудиті та звітності. Це означає, що кожна зміна у конфігурації повинна бути записана, а історія трейсів – збережена для подальшого аналізу.

У кінцевому рахунку, ми прагнемо до створення модульної системи, де кожен компонент може бути легко замінений або оновлений без втрати функціональності або необхідності повного переписування конфігурацій. Ми також плануємо впровадити шар абстракції, який дозволить користувачам визначати вищі рівні бізнес-логіки, не заглиблюючись у технічні деталі реалізації окремих процесів.

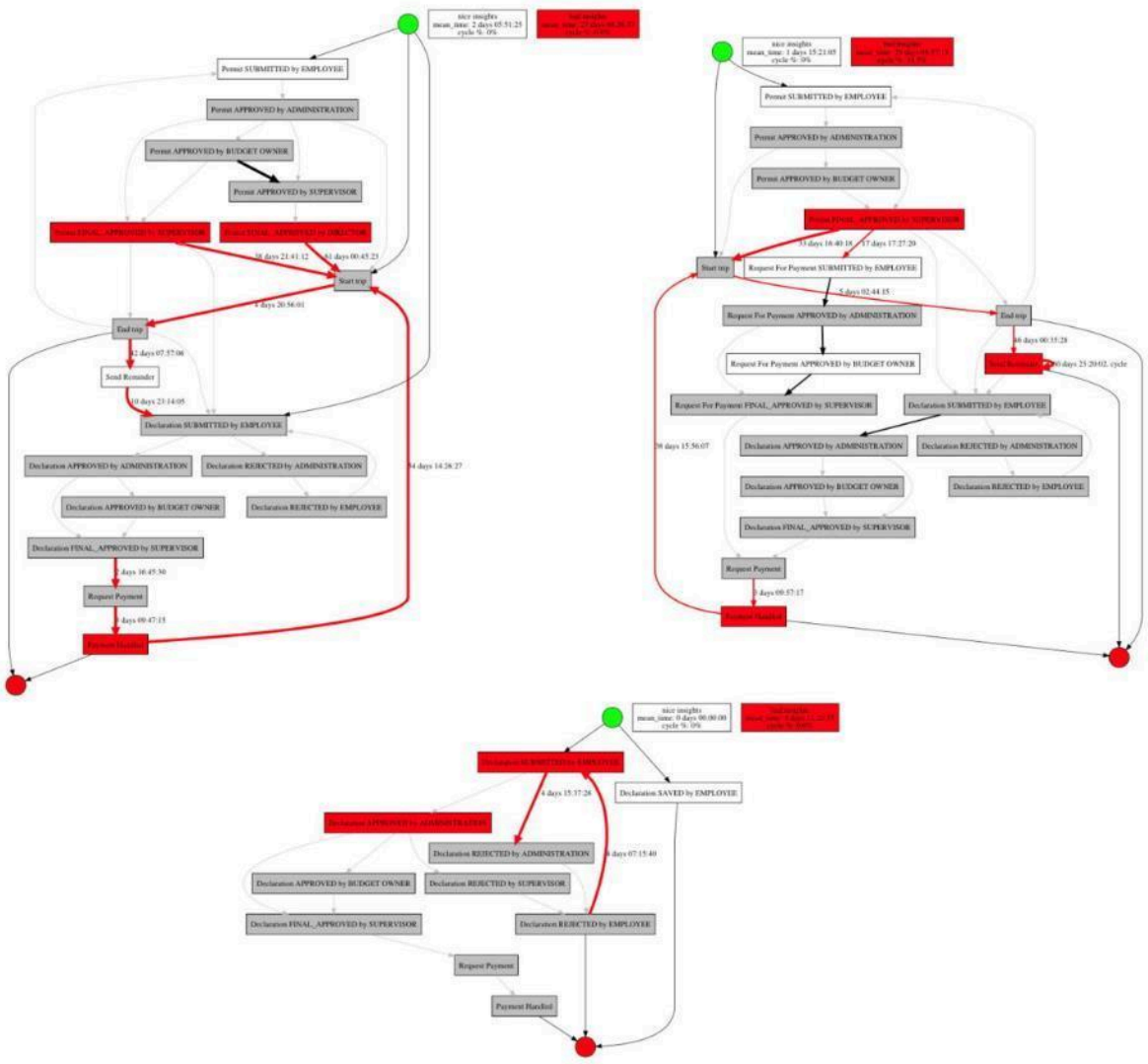

Рис. 3.19. Реалізовані схеми *DataHolder*

На зображенні представлена схема, що ілюструє різні алгоритми для обробки та візуалізації лог-файлів в бібліотеці, яка займається аналізом бізнес-процесів. Ключовим елементом в цій схемі є DataHolder, який є основним контейнером для зберігання лог-даних, та який використовується більшістю алгоритмів бібліотеки.

Лог-файл, який зберігається в DataHolder, забезпечує вичерпну інформацію про хід виконання бізнес-процесу, що дозволяє аналітикам відтворити реальний (AS-IS) процес, а не тільки його теоретичне планування.

Для відтворення графа реального процесу бібліотека надає декілька алгоритмів, які називаються майнерами:

– SimpleMiner створює граф, відображаючи всі взаємозв'язки, знайдені в лог-файлі.

– CausalMiner фокусується на прямих зв'язках, ігноруючи можливі побічні або випадкові кореляції.

– HeuMiner використовує поріг (threshold) для визначення та видалення рідкісних зв'язків, тим самим спрощуючи граф.

– AlphaMiner розробляє граф у формі мережі Петрі, враховуючи прямі, паралельні та незалежні зв'язки між активностями.

– AlphaPlusMiner є розширеною версією AlphaMiner, здатною обробляти ланцюги з одиночними циклами.

Кожен з наведених майнерів реалізує власні методи для аналізу та обробки даних логів та відтворення послідовності процесів. Вибір конкретного алгоритму аналізу логів залежить від визначеної мети аналізу та унікальних характеристик конкретного лог-файлу. Ця індивідуальність дозволяє аналітикам здійснювати гнучкий підхід до візуалізації процесів та оптимізації бізнес-процесів.

На графічній схемі, наведеній нижче, червоними лініями показано можливі шляхи переходу від одного кроку до іншого у відтворенні процесів. Зелені круги вказують на стартові точки різних процесів або алгоритмів, вказуючи на початок виконання певних завдань. У той час, червоні круги сигналізують про завершення виконання цих процесів або алгоритмів. Така схема дозволяє наглядно відобразити послідовність подій та логічні відносини між ними, допомагаючи аналітикам отримувати цінний інсайт та приймати стратегічні рішення на основі аналізу лог-файлів.

Кожен із вказаних майнерів може мати свою унікальну систему візуалізації та аналізу даних, що робить їх інструментами, допомагаючими в розумінні та вдосконаленні бізнес-процесів на різних етапах аналізу та оптимізації.

```python
from sberpm.miners import HeuMiner
from sberpm.visual import GraphvizPainter

# Майнер
heu_miner = HeuMiner(data_holder, threshold=0.8)
heu_miner.apply()
graph = heu_miner.graph

# Отрисовка
painter = GraphvizPainter()
painter.apply(graph)
painter.show()
```

Рис. 3.20. Код реалізації *Graphiz*

Можна також зберегти (імпорт) або завантажити (експорт) граф (рис. 3.21) у форматі *BPMN* (*Business Process Model Notation*).

```python
from sberpm.bpmn import BpmnExporter

bpmn_exporter = BpmnExporter()
bpmn_exporter.apply_petri(alpha_miner.graph)
bpmn_exporter.write('exported.bpmn')
```

Рис. 3.21. Код імпорту *Graphiz*

Візуальна схема дозволяє отримати повне уявлення про ланцюжок подій, а й досліджувати актуальний стан процесу на рівні деталізації. Як приклад розглянемо графи, побудовані різними майнерами, для одного й того самого синтетичного процесу. Кожен граф відображає унікальну інтерпретацію лог-даних, і, порівнюючи їх, можна виявити не тільки явні послідовності дій, але й потенційні узагальнення або аномалії в поведінці процесу.

SimpleMiner може надати нам базову картину, показуючи всі можливі переходи між подіями. Це дає змогу побачити повний обсяг активностей, але може бути перевантаженим та складним для аналізу при наявності великої кількості даних.

CausalMiner прибирає частину шуму, зосереджуючись на прямих причинно-наслідкових зв'язках. Це може допомогти у визначенні критичних шляхів і ключових точок взаємодії.

HeuMiner вдосконалює це подальше, усуваючи незначні зв'язки і залишаючи тільки ті, що перевищують визначений поріг частоти. Це дозволяє нам сконцентруватися на найбільш значущих аспектах процесу.

AlphaMiner і AlphaPlusMiner пропонують ще більшу деталізацію, представляючи процеси в контексті мережі Петрі, яка може включати паралельні

та циклічні події, а також незалежні взаємодії. Такий підхід дозволяє виявити складні залежності та потенційні точки оптимізації.

Аналізуючи різні графи, можна порівняти, як кожен майнер реагує на однакові дані, та визначити, яка візуалізація найкраще відповідає потребам користувача. Для більшої інформативності можливо впровадити інтерактивні елементи у візуалізації, дозволяючи користувачам занурюватися в специфічні ділянки графа, щоб детальніше розглянути окремі події або послідовності дій.

Усе це надає можливість не тільки зрозуміти поточний стан та поведінку процесу, але й спрогнозувати його майбутнє функціонування, виявити потенційні точки затору, і, в кінцевому підсумку, оптимізувати процес для підвищення ефективності та зниження витрат.

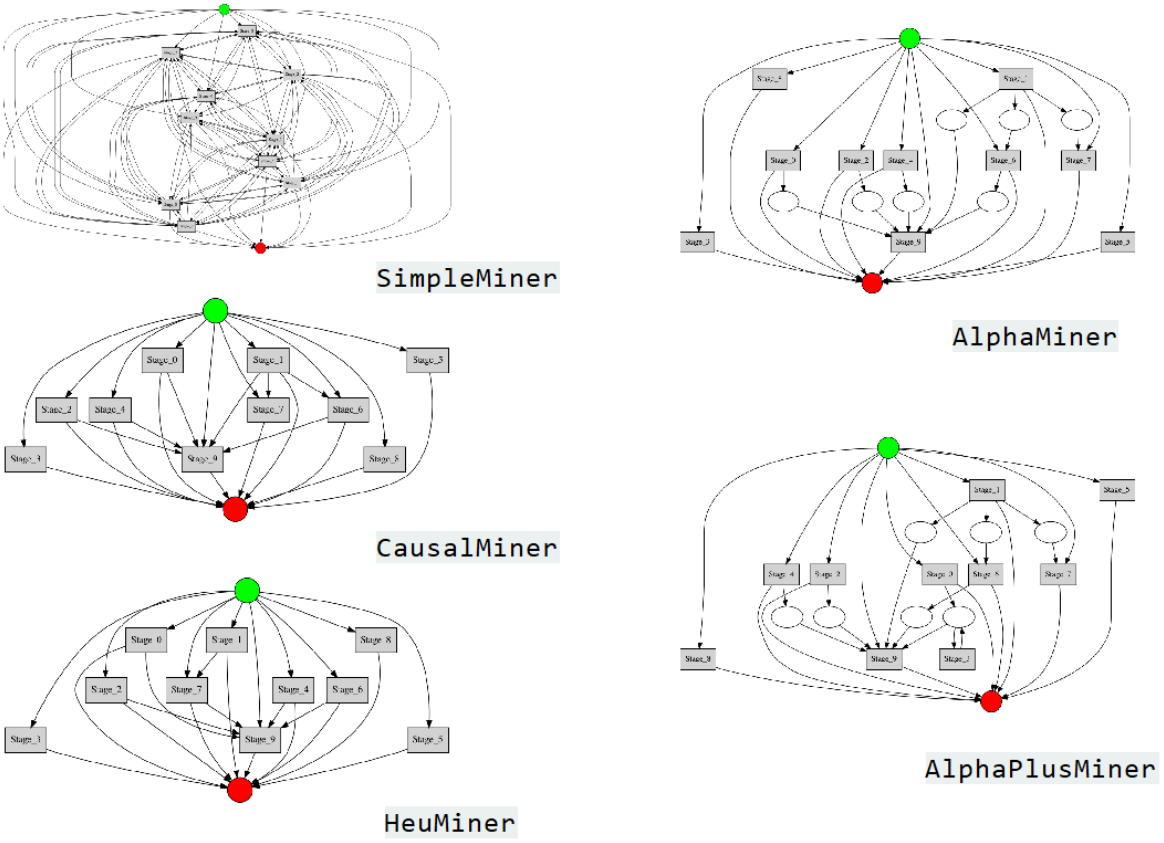

Рис. 3.22. Приклад побудови графів різними бібліотеками

За це в бібліотеці відповідає модуль метрик, в якому на даний момент реалізовано такі види статистики:

– *ActivityMetric*– метрики з унікальних активностей;

– *TransitionMetric*– метрики за унікальними переходами;

– *IdMetric*– метрики з *ID*;

– *TraceMetric*– метрики за унікальними ланцюжками активностей;

– *UserMetric*– метрики за унікальними користувачами;

– *TokenReplay*– *fitnes*, який показує, наскільки добре граф описує бізнес-процес.

У перших п'яти випадках для об'єкта угруповання розраховуються кількість появ, число унікальних *ID*/активностей/користувачів, відсоток зациклювань, часові характеристики (середня, медіанна, максимальна та інші види тривалості) тощо (рис. 3.23).

```
from sberpm.metrics import UserMetric

# Создание объекта UserMetric
user_metric = UserMetric(data_holder, time_unit='d')

# Расчет всех метрик
user_metric.apply().head()
```

	users	unique_activities	unique_activities_num	activities_count	unique_ids_num	workload_in_percent	total_duration	min_duration	max_duration	mean_d
0	Amelia	{Stage_3}	1	184	184	4.12	872.0	1.0	9.0	5
1	Barbara	{Stage_9}	1	136	136	3.05	96.0	1.0	9.0	5
2	Bethany	{Stage_4}	1	56	56	1.26	245.0	1.0	9.0	4
3	Callum	{Stage_7}	1	167	167	3.74	593.0	1.0	9.0	5
4	Charlie	{Stage_1}	1	67	67	1.50	299.0	1.0	9.0	4

Рис. 3.23. Приклад роботи класу *UserMetric*

Безперечною перевагою даного модуля є швидкість розрахунків. Припустимо, перед аналітиком стоїть завдання визначити середню тривалість найчастіших ланцюжків подій процесу. Рішення методами *pandas* займе 5 хвилин і більше 10 рядків коду, у той час як рішення методами – 1 хвилину та 3 рядки коду (рис. 3.24).

```
from sberpm.metrics import TraceMetric

trace_metric = TraceMetric(data_holder, time_unit='d')
res = trace_metric.apply().sort_values('total_count', ascending=False)
res.mean_duration.head()
```

```
0    20.583333
1    16.333333
2    11.727273
3    25.200000
4    16.100000
```

Рис. 3.24. Приклад роботи класу

В результаті на графі можна, наприклад, змінити ширину ребер і колір вузлів в залежності від значень метрик і тим самим відстежити найчастіші шляхи та довгі етапи процесу. Це візуалізує потоки даних або робочі процеси, що відбуваються з великою частотою, а також ті, які вимагають надмірно багато часу для виконання. Ширина ребер може відображати кількість подій або транзакцій, які пройшли через певний крок процесу, тоді як колір вузлів може індикувати середній час виконання або інші важливі показники.

Таким чином, аналізуючи модель реконструйованого процесу разом із даними про тривалість та особливості його виконання, можна виявити затримки за часом реалізації окремих дій. Можливо виявити часті перехрестя в процесі, де відбуваються затримки через взаємодію різних користувачів або відділів. Це також дозволяє ідентифікувати зацикленості, коли процес повторюється без досягнення результату, або виявляти неефективних виконавців, що можуть спричиняти збої або затримки у процесах.

Крім того, такий глибокий аналіз може виявити приховані недоліки, які не були очевидні при поверхневому огляді. Наприклад, можна визначити, що певні процедури або політики компанії призводять до непотрібних затримок або викликають зайві кроки в робочих процесах. Розуміння цих аспектів дає можливість для оптимізації і покращення загальної продуктивності, що, в свою чергу, може призвести до зростання ефективності роботи всієї організації.

Отже, застосування деталізованого візуального аналізу реальних даних про процеси дозволяє менеджменту не тільки виявляти й усувати конкретні проблеми, але й проводити профілактичні заходи для уникнення подібних ситуацій у майбутньому, а також розробляти стратегії для підвищення загальної продуктивності та ефективності (рис. 3.25).

Таким чином, аналізуючи модель реконструйованого процесу разом із даними про тривалість та особливості його виконання, можна виявити затримки за часом реалізації окремих дій, взаємозв'язку між користувачами, зацикленості в процесі, неефективних виконавців, а також приховані недоліки та проблеми у процесах, через які може значно знижуватися продуктивність цілої організації.

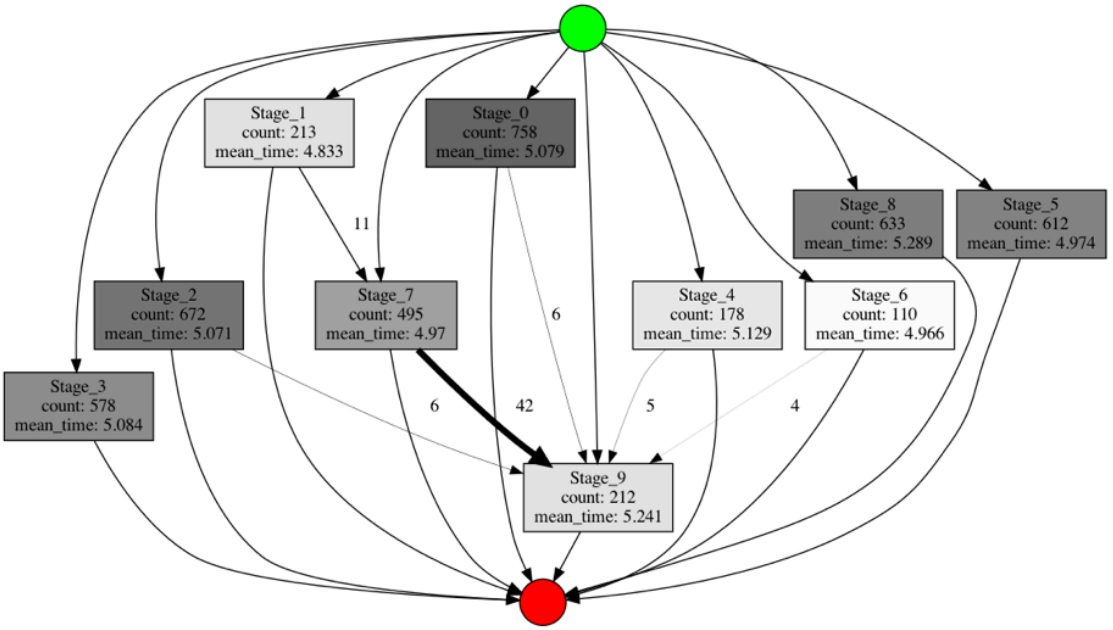

Рис. 3.25. Реалізована візуалізація бізнес-процесу

Модуль *ML* – окрім класичних інструментів *Process Mining,* пропонує функціонал методів машинного навчання. На даний момент користувачам доступні векторизація та кластеризація процесів, а також модуль автопошуку інсайтів. Розкажемо докладніше, навіщо це потрібно і як цим скористатися.

Нижче наведено приклад роботи з модулем та результат візуалізації інсайтів на графі (рис. 3.26).

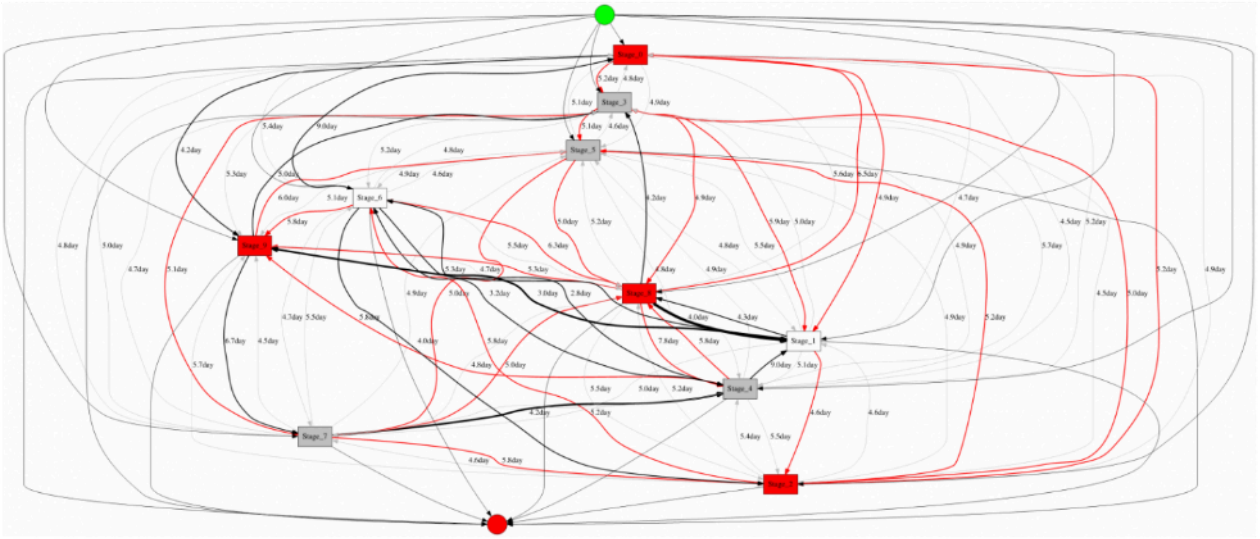

Рис. 3.26. Візкалізація інсайтів

«Погані» переходи й активності, потребують оптимізації, виділені червоним, «хороші», тобто. не потребують оптимізації – чорним, нейтральні – сірим. Товщина ребер на графі змінюється в залежності від оптимальності переходу.

Додатково для всіх активностей і переходів можна вивести детальнішу таблицю, де для кожного елемента відзначається, чи є він інсайтом і, якщо так, то за якою саме метрикою. Так, "1" у графі *insights* означає, що об'єкт є "хорошим" інсайтом, "-1" – "поганим" інсайтом, "0" – не є інсайтом зовсім.

```
auto_i.describe_nodes().head()
```

	activities	count	mean_time	cycle	nunique_users	insights
0	Stage_0	1	0	0	0	1
1	Stage_1	0	-1	0	0	-1
2	Stage_2	1	0	0	0	1
3	Stage_3	0	0	0	0	0
4	Stage_4	0	0	0	0	0

Рис. 3.27. Результат запуску

Взагалі *Process Mining* має сенс усюди, де існують бізнес-процеси, а вони є в будь-якій організації, навіть якщо вона некомерційна. Звичайно, від компанії до компанії процеси будуть різними, але ви знайдете їх і у великих холдингах, і маленьких фірмах. Наприклад, загальна схема бізнес-процесів виробничої організації може виглядати приблизно так:

3.6. Висновки до розділу

Розроблена система інтеграції в систему безпеки відповідає сучасним тенденціям, але не варто забувати і про перспективи розвитку інтегрованих систем безпеки. Основні напрями визначаються наступними вимогами:

– зниження ролі людини в процесі забезпечення безпеки за рахунок підвищення інтелектуальності систем;

– зниження рівня помилкових спрацьовувань за рахунок більш тісної використання підсистем;

– вимога відкритості. Розробники ІСБ повинні забезпечити замовнику за допомогою відкритих протоколів можливості підключення систем і устаткування інших виробників і гнучкого настроювання ІСБ під свої потреби.

Реалізація зазначених вимог з одного боку дозволить збільшити ефективність систем безпеки, зменшить вплив людського фактору, з іншого - зробить побудова інтегрованих систем більш прозорим.

ВИСНОВКИ

У кваліфікаційній роботі "Програмний модуль моніторингу доступу до закритих корпоративних систем" було розроблено і реалізовано комплексний підхід до контролю та аналізу доступу до корпоративних ресурсів. Під час дослідження основною метою було створення програмного рішення, яке дозволяло б контролювати доступ до чутливої інформації та забезпечувати відповідність корпоративним політикам безпеки.

Програмний модуль був розроблений з урахуванням кращих практик інформаційної безпеки та включав функції аутентифікації, авторизації, аудиту та реєстрації подій. Це дозволило відслідковувати і записувати всі спроби доступу до системи, включаючи успішні та невдалі логіни, а також спроби доступу до конкретних файлів або сервісів.

Для аналізу зібраних даних було реалізовано інтегрований інструмент з можливістю візуалізації даних моніторингу. Він надавав зрозумілі графіки та звіти, які допомагали ідентифікувати аномальну поведінку та потенційні внутрішні загрози. Крім того, введення системи оповіщень дозволило швидко реагувати на інциденти безпеки.

Робота також включала розробку модуля політик доступу, який забезпечував динамічне управління правилами доступу на основі ролей користувачів, контексту доступу та інших параметрів. Цей компонент модулю дозволив адміністраторам системи гнучко налаштовувати права доступу згідно з постійно змінними бізнес-вимогами та загрозами безпеки.

Завершальною частиною роботи стало тестування та валідація розробленого програмного модуля. Процес тестування включав перевірку функціональності, перевірку на вразливості та оцінку відповідності до корпоративних стандартів безпеки. Результати тестування показали, що програмний модуль ефективно ідентифікує та реагує на спроби несанкціонованого доступу, а також забезпечує адміністраторам потужний інструмент для моніторингу та аналізу безпеки корпоративних систем.

Була розроблена схема для представлення архітектури додатку-сервера СКУД:

– «Система зв’язку сервера» – пакет, який відповідає за зв’язок клієнтів і серверу, а також серверу і устаткування СКУД, такого як турнікети, датчики і т.д. Він організовує прийом вхідних повідомлень і відправку вихідних. Вхідні повідомлення інтерпретуються як події, їх публікація здійснюється від імені структурних об’єктів;

– пакет «Структурні об’єкти» містить об’єкти, які є агентами зовнішніх елементів системи: контролерів, програм-клієнтів та інших. Агенти використовують систему зв’язку для спілкування з зовнішнім світом;

– пакет «Об’єкти-дані» здійснює угруповання ряду пакетів, що представляють такі елементи предметної області, як карти, рахунки, групи, ресурси і персони;

– пакет «Виконавча підсистема» відповідає за виконання запитів від додатків і обробку подій контролерів. Він являє собою адаптовану до системи реалізацію механізму команд. Команди працюють як з об’єктами-даними, так і зі структурними об’єктами;

– пакет «Збереження об’єктів» показує, яким чином сервер використовує механізми збереження об’єктів. Цей пакет використовується командами, структурними об’єктами і об’єктами-даними.

З контролем доступу можна інтегрувати різноманітні системи, що забезпечують такі переваги, як:

– контроль підйому – обмежити доступ на несанкціоновані поверхи (наприклад, поверхи готелів);

– відправка ліфтів– ефективне спрямування користувачів до підйомників;

– *CCTV*– зв’язати події доступу з відеоспостереженням (наприклад, примусова перевірка дверей або вторинна перевірка);

– вогонь– контроль дверей та інтеграція плану сайту;

– сигналізація про злам– запобігання помилкових тривог (шляхом випадкового входу або встановлення зайнятої зони);

– система управління будівлею (*BMS*)– для екологічного контролю з використанням інформації про заповненість;

– *HR*– інтеграція даних персоналу з профілями доступу;

– системи управління школою та студентами– інтеграція студентських даних з профілями доступу;

– безготівкова торгівля– загальна картка для торгівлі та доступу;

– бібліотечні системи – загальна картка для видачі та контролю доступу;

– системи керування відвідувачами;

– системи паркування – загальні картки та спільні обмеження;

– управління активами – контроль руху активів і зв’язок з користувачем;

– аудіо відеодомофон– комбіноване використання читачів, віддалений доступ відвідувачів тощо;

– охоронна екскурсія– використовувати зчитувачі контролю доступу для керування поведінкою охоронців.

В кваліфікаційній роботі були поставлені такі виклики і знайдені для них рішення:

1. Захист чутливих даних. Рішення: Розробка рішення з використанням шифрування для зберігання та передачі даних, а також реалізація багаторівневої системи аутентифікації і авторизації для забезпечення того, щоб доступ до інформації мали лише уповноважені особи.

2. Виявлення несанкціонованого доступу. Рішення: Імплементація комплексної системи моніторингу з можливістю виявлення аномалій у поведінці користувачів і негайного сповіщення системних адміністраторів про будь-які підозрілі дії.

3. Швидкість обробки великих обсягів даних. Рішення: Використання ефективних алгоритмів та оптимізація баз даних для забезпечення швидкої обробки та відповіді в режимі реального часу, використання розподіленої обробки даних для зменшення навантаження на центральну систему.

4. Інтеграція з існуючими корпоративними системами. Рішення: Розробка модульної архітектури, що дозволяє легко інтегрувати новий моніторинговий модуль з існуючою IT-інфраструктурою, використання API або протоколів обміну даними стандартів для забезпечення сумісності.

5. Забезпечення масштабованості системи. Рішення: Використання хмарних технологій та мікросервісної архітектури для того, щоб система могла бути масштабована в залежності від змінюваних потреб організації.

6. Дотримання нормативних вимог. Рішення: Забезпечення відповідності програмного рішення нормативним і законодавчим вимогам шляхом регулярного аудиту, тестування на вразливості та оновлення політик безпеки.

7. Гнучкість налаштувань політик доступу. Рішення: Розробка інтерфейсу управління політиками доступу, який дозволяє адміністраторам легко оновлювати та налаштовувати правила безпеки відповідно до змінних умов безпеки та бізнес-процесів.

Ці рішення забезпечили створення надійної та ефективної системи моніторингу доступу, яка може бути адаптована до різних корпоративних середовищ і потреб.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проєкти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
2. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – 39 с.
3. E. B. Fernandez and R. B. France, ``Formal specification of real-time dependable systems ", Proc. of First IEEE Int. Conf. on Eng. of Complex Comp. Systems, Fort Lauderdale, FL, November 6-10, 1995, 342-348.
4. ACM Workshop on Role-Based Access Control, November 1997, 121-125. http://www.cse.fau.edu/~ed/RBAC.pdf
5. Procs. 10th Intl. Workshop on Database and Expert Systems Applications, 1999, 837-841. http://www.cse.fau.edu/~ed/Coordinationsecurity4.pdf
6. E.B. Fernandez and X. Yuan, “Semantic analysis patterns”, Procs. of 19th Int. Conf. on Conceptual Modeling, ER2000, 183-195. Also available from: http://www.cse.fau.edu/~ed/SAPpaper2.pdf
7. E.B.Fernandez and J.C.Sinibaldi, "More patterns for operating systems access control", Procs. EuroPLoP 2003, http://hillside.net/europlop
8. E. Gamma, R. Helm,R. Johnson, and J. Vlissides, Design patterns –Elements of reusable object-oriented software, Addison-Wesley 1995.
9. M. Howard and D. LeBlanc, Writing secure code, (2nd Ed.), Microsoft Press, 2003.
10. V. Hays, M. Loutrel, and E.B.Fernandez, “The Object Filter and Access Control framework”, Procs. Pattern Languages of Programs (PLoP2000) Conference, http://jerry.cs.uiuc.edu/~plop/plop2k
11. P.G.Neumann, “On hierarchical design of computer systems for critical applications”, IEEE Trans. on Software Eng., vol. SE-12, No 9, September 1986, 905-920.
12. P.G.Neumann, “The role of software engineering”, Comm. of the ACM, Vol. 36, No 5, May 1993, 114.

13. J.H.Saltzer and M.D.Schroeder, “The protection of information in computer systems”, Procs. of the IEEE, Vol. 63, No 9, 1975, 1278-1308. A web version is in: http://web.mit.edu/Saltzer/www/publications/protection/index.html

14. M. Schumacher, E.B.Fernandez, D. Hybertson, and F. Buschmann, Security Patterns, to be published by J. Wiley & Sons, 2004.

15. Ammann, P. E., Black, P. E., Majurski, W. Using Model Checking to Generate Tests from Specifications. Proceedings Second International Conference on For- mal Engineering Methods (Cat.No.98EX241), Brisbane, Queensland, Australia, 1998, 46-54. https://doi. org/10.1109/ICFEM.1998.730569

16. Ammar, B., Abdallah, K. Towards the Formal Specifi- cation and Verification of multi-Agent Based Systems. IJCSI, 2011, 8(4), 200-210.

17. Arts, T., Castro, L. M., Hughes, J. Testing Erlang Data Types with Quviq QuickCheck. 7th ACM SIGPLAN Workshop on ERLANG (Erlang’08), ACM, Victoria, BC, Canada, September 20-28, 2008, 1–8.

18. Bae, K., Meseguer, J. The Linear Temporal Logic of Re- writing Maude Model Checker. WRLA, Springer, 2010, 208-225.

19. Ballarini, P., Djafri, H., Duflot, M., Haddad, S., Pekergin,

20. N. Petri Nets Compositional Modeling and Verifica- tion of Flexible Manufacturing Systems. 2011 IEEE Conference on Automation Science and Engineer- ing (CASE), 2011, 588-593. https://doi.org/10.1109/ CASE.2011.6042488

21. Benantar, M. Access Control Systems: Security, Identi- ty Management and Trust Models. Springer Science & Business Media, 2006.

22. Bernot, G., Gaudel, M.C., Marre, B. Software Testing Based on Formal Specifications: A Theory and a Tool. Software Engineering Journal, 1991, 6(6), 387-405. https://doi.org/10.1049/sej.1991.0040

23. Burton, S., York, H. Automated Testing from Z Speci- fications. Technical Report, Department of Computer Science, University of York, 2000.

24. Castro, L. M. Advanced Management of Data Integrity: Property-Based Testing for Business Rules. Journal of Intelligent Information Systems, 2015, 44(3), 355-380. https://doi.org/10.1007/s10844-014-0335-2

25. Castro, L. M., Arts, T. Testing Data Consistency of Da- ta-Intensive Applications Using QuickCheck. 10th Spanish Conference on Programming and Languages (PROLE'10), Valencia, Spain, September 8- 10, 2010. Revised Selected Papers, Electronic Notes in Theoret- ical Computer Science, Elsevier Science Publishers, Amsterdam, the Netherlands, 2011, 271, 41–62. https:// doi.org/10.1016/j.entcs.2011.02.010

26. Castro, L. M., Francisco, M. A., Gulías, V. M. Testing Integration of Applications with QuickCheck. Interna- tional Conference on Computer Aided Systems Theory, 2009.

27. Chow, T. S. Testing Software Design Modeled by Fi- nite-State Machines. IEEE Transactions on Soft- ware Engineering, 1978, SE-4(3), 178-187. https://doi. org/10.1109/TSE.1978.231496

28. Clarke, E. M., Emerson, E. A. Design and Synthe- sis of Synchronization Skeletons Using Branching Time Temporal Logic. Springer, 1982. https://doi. org/10.1007/BFb0025774

29. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martı-Oliet, N., Meseguer, J., Quesada, J. A Maude Tutorial. Com- puter Science Laboratory, SRI International, 2000.

30. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martı-Oliet, N., Meseguer, J., Quesada, J. F. Maude: Specification and Programming in Rewriting Logic. Theoretical Computer Science, 2002, 285(2), 187-24. https://doi. org/10.1016/S0304-3975(01)00359-0

31. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C. All About Maude – A High-Per- formance Logical Framework: How to Specify, Program and Verify Systems in Rewriting Logic. Springer-Ver- lag, 2007.

32. David, R., Alla, H. On Hybrid Petri Nets. Discrete Event Dynamic Systems, 2001, 11(1-2), 9-40. https://doi. org/10.1023/A:1008330914786

33. Deng, Y., Lu, S., Evangelist, M. A Formal Approach for Architectural Modeling and Prototyping of Distributed Real-Time Systems. System Sciences, 1997, 1, 481-490. https://doi.org/10.1109/HICSS.1997.667304

34. Eker, S., Meseguer, J., Sridharanarayanan, A. The Maude LTL Model Checker. Electronic Notes in Theo- retical Computer Science, 2004, 71, 162-187. https://doi. org/10.1016/S1571-0661(05)82534-4

35. Fink, G., Bishop, M. Property-Based Testing: A New Ap- proach to Testing for Assurance. ACM SIGSOFT Soft- ware Engineering Notes, 1997, 22(4), 74-80. https://doi. org/10.1145/263244.263267

36. Fink, G., Levitt, K. Property-Based Testing of Privileged Programs. Proceedings 10th Annual Computer Securi- ty Applications Conference, 1994, 154-163. https://doi. org/10.1109/CSAC.1994.367311

37. Gargantini, A., Heitmeyer, C. Using Model Check- ing to Generate Tests from Requirements Speci- fications. ACM SIGSOFT Software Engineering Notes, Springer-Verlag, 1999, 24, 146-162. https://doi. org/10.1145/318774.318939

38. Goguen, J., Kirchner, C., Kirchner, H., Mégrelis, A., Meseguer, J., Winkler, T. An Introduction to OBJ 3. In- ternational Workshop on Conditional Term Rewriting Systems, Springer, 1987, 258-263.

39. Grumberg, O., Veith, H. 25 Years of Model Checking: History, Achievements, Perspectives. Springer, 2008, 5000. https://doi.org/10.1007/978-3-540-69850-0

40. Дозоркінц В.М. Світовий ринок СКУД // Автоматизація в промисловості. 2021. № 2. – С. 47-50

41. Євгенев Г.Б. Інтеграція прикладних систем на основі баз знань // Програмні продукти і системи. Додаток до Міжнародного журналу "Проблеми теорії і практики управління". 2005. № 3.

Додаток А

Лістинг основного програмного модуля