МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
Аліна САВЧЕНКО
«_____» _____ 2022р.

# КВАЛІФІКАЦІЙНА РОБОТА

**(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)**
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ **«МАГІСТР»**
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

**Тема: «Програмний компонент системи управління інформаційною безпекою з використанням електронного сховища файлів підприємства»**

**Виконавець:** студентка групи УС-211М Хвостова Дар'я Вікторівна

**Керівник:** к.т.н., доцент кафедри Колісник Олена Василівна

**Нормоконтролер:** Ігор РАЙЧЕВ

**Київ — 2022**

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет *Кібербезпеки, комп'ютерної та програмної інженерії*

Кафедра *Комп'ютерних інформаційних* технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 «Інформаційні технології», 122 «Комп'ютерні науки», «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач випускової кафедри

Аліна САВЧЕНКО

«____»_____2022р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи студента**

Хвостової Дар'ї Вікторівни

(прізвище, ім'я, по батькові)

**1. Тема роботи:** «Програмний компонент системи управління інформаційною безпекою з використанням електронного сховища файлів підприємства» затверджена наказом ректора від «28» вересня 2022 р. за № 1774/ст.

**2. Термін виконання роботи:** з 26 вересня 2022 р. по 21 листопада 2022 р.

**3. Вихідні дані до роботи:** мова програмування ASP.NET, HTML, CSS; мережева архітектура клієнт-сервер; огляд, документація та практичне випробування програмного застосунку GALA (Group Access List Application).

**4. Зміст пояснювальної записки**: вступ, системи управління інформаційною безпекою, стандартизація, розгляд архітектури СУІБ, розгляд архітектури MVC,

розгляд архітектури власне розробленого програмного компонента СУІБ, випробування його на практиці.

**5. Перелік обов'язкового ілюстративного матеріалу**: слайди, презентація

**6. Календарний план-графік**

| № п/п | Завдання | Термін виконання | Підпис керівника |
|---|---|---|---|
| 1. | Отримання завдання на кваліфікаційну роботу, створення плану кваліфікаційної роботи та побудова плану-графіку виконання робіт. | 26.09.2022 - 28.09.2022 | |
| 2. | Огляд та аналіз наукової літератури по темі кваліфікаційної роботи та написання Розділу 1. | 29.09.2022 - 09.10.2022 | |
| 3. | Написання Розділу 2 кваліфікаційної роботи. | 10.10.2022 -20.10.2022 | |
| 4. | Написання Розділу 3 і Розділу 4 кваліфікаційної роботи. Завершення створення пояснювальної записки кваліфікаційної роботи. | 21.10.2022 - 31.10.2022 | |
| 5. | Оформлення та друк пояснювальної записки. | 01.11.2022 - 07.11.2022 | |
| 6. | Створення презентації, доповіді та підготовка до захисту кваліфікаційної роботи. | 08.11.2022 - 15.11.2022 | |
| 7. | Підготовка матеріалів кваліфікаційної роботи для передачі секретарю ДЕК (папка, конверт, диск із файлом диплому, рецензія, відгук). | 16.11.2022 - 18.11.2022 | |

**7. Дата видачі завдання:** «26» вересня 2022 р.

**Керівник кваліфікаційної роботи** _____ Олена КОЛІСНИК
(підпис керівника) (П.І.Б.)

**Завдання прийняв до виконання** _____ Дар'я ХВОСТОВА
(підпис випускника) (П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Програмний компонент системи управління інформаційною безпекою з використанням електронного сховища файлів підприємства» складається зі вступу, чотирьох розділів, висновку, списку бібліографічних посилань та двох додатків, і містить 133 сторінки та 11 рисунків. Список бібліографічних посилань складається з 50 найменувань.

**Ключові слова:** СУІБ, СХОВИЩЕ ФАЙЛІВ, ASP.NET, MVC, АРХІТЕКТУРА КЛІЄНТ-СЕРВЕР, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНА СИСТЕМА.

**Актуальність** теми полягає саме у розвитку цифрової трансформації різних процесів підприємства та супутніх задач і ризиків, які постають через це, а саме для цієї роботи — управління та забезпечення інформаційної безпеки цифрового інтелектуального надбання підприємства у вигляді файлів, що створюються та зберігаються на корпоративному мережевому файловому сховищі.

**Метою кваліфікаційної роботи** є дослідження, розробка та впровадження програмного компоненту системи управління інформаційною безпекою на прикладі електронного сховища файлів підприємства, а саме системи аудиту прав доступу до мережевих папок.

Для досягнення цієї мети необхідно розв’язати наступні **задачі**:

1. Проаналізувати основні поняття та завдання сучасних систем управління інформаційною безпекою;
2. Провести аналіз та порівняння існуючих систем управління інформаційною безпекою;
3. Реалізувати програмний компонент системи управління інформаційною безпекою.

**Об'єктом дослідження** — процеси проектування та впровадження програмного компонента систем управління інформаційною безпекою.

**Предметом дослідження** — системи управління інформаційною безпекою.

**Основним науковим результатом роботи** є розробка модульної архітектури та складання переліку вимог до програмного компоненту системи управління

інформаційною безпекою, що дозволяє виконувати аудит актуального доступу до мережевих папок, а також інші процедури аудиту доступу до інформації згідно зі стандартом ISO 27001.

**Практична цінність роботи** полягає в розробці та впровадженні на підприємстві веб-застосунку на базі модульної архітектури з використанням мови програмування ASP.NET, включаючи виконання аудиту актуальних прав доступу до мережевих папок згідно з вимогами стандарту ISO 27001.

Також проведено **практичну апробацію результатів роботи**. Розроблений застосунок введено в дослідну (тестову) експлуатацію на підприємстві. За результатами проведених випробувань отримано схвальні відгуки від учасників, а застосунок рекомендовано до впровадження.

**ЗМІСТ**

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

| | |
|---|---|
| **AD** | LDAP-сумісна реалізація інтелектуальної служби каталогів корпорації Microsoft для операційних систем родини Windows NT |
| **COBIT** | Контроль за інформаційними та пов'язаними технологіями |
| **ERP** | Планування ресурсів підприємства |
| **GUI** | Графічний інтерфейс користувача |
| **IaaS** | Інфраструктура як послуга |
| **IEC** | Міжнародна електротехнічна комісія |
| **ISO** | Міжнародна організація стандартизації |
| **ISO 27001** | Стандарт, що визначає вимоги щодо планування, впровадження, експлуатації та постійного моніторингу та вдосконалення СУІБ |
| **ITIL** | Бібліотека IT-інфраструктури |
| **LDAP** | Полегшений протокол доступу до директорій / каталогів |
| **MVC** | Контролер представлення моделі |
| **OLAP** | Аналітична обробка у реальному часі |
| **PKI** | Інфраструктура відкритих ключів |
| **SaaS** | Програма як послуга |
| **ЗОС** | Зареєстровані органи сертифікації |
| **ІТ** | Інформаційні технології |
| **СЕД** | Система електронного документообігу |
| **СУБД** | Система управління базами даних |
| **СУІБ** | Система управління інформаційною безпекою |

# ВСТУП

Інформаційна безпека є невід'ємним елементом фідуціарного обов'язку, який вважається частиною управління IT. Метою інформаційної безпеки є захист цінних ресурсів організації, таких як інформація. У відповідних стандартах і структурах, а також у літературі повідомлялося про постійне зростання залежності майже всіх організацій від відповідної безпечної обробки інформації. Було розроблено та створено стандарти управління інформаційною безпекою та збірники найкращих практик. Відповідно до найважливіших і найбільш широко прийнятих міжнародних ініціатив щодо розробки та експлуатації системи управління інформаційною безпекою (СУІБ) є ISO 270xx, ITIL і COBIT, також актуальні в таких аспектах, як управління інформацією та безпекою, а також хмарне управління.

Типові відділи, включаючи управління ризиками, юридичний відділ, аудит, відповідність, конфіденційність, безперервність бізнесу, контроль якості, обладнання, людські ресурси, IT-безпеку, інформаційну безпеку та фізичну безпеку, займаються діяльністю, яка має відношення до безпеки або пов'язана з нею. Інтеграція цих заходів у структуру процесу інформаційної безпеки, яка чітко визначає взаємозв'язки, забезпечить економічно ефективну безпеку, але їхню діяльність, як правило, розглядають як ізольовані. Протягом останніх кількох років дискусії про співвідношення витрат і прибутків вплинули на практику інформаційної безпеки. Цінність інформації повинна виправдовувати витрати на захист. Налаштування та економічна ефективність є ключовими елементами успішної СУІБ. Знання місії необхідні для узгодження процесів СУІБ з організацією та її місією. Беручи до уваги важливість узгодження бізнесу та економічної ефективності для успішної роботи СУІБ, дослідницькі внески повинні вирішувати обидві проблеми, дозволяючи чітко визначити необхідні та відповідні процеси СУІБ як основні елементи кожної СУІБ [2].

Проблема полягає в тому, що насправді такої структури процесу для управління безпекою не існує [2]. Це все ще є проблемою, оскільки управління інформаційною безпекою є складним питанням, а поточні дослідження зосереджені

на економіці та аналізі витрат і прибутків інвестицій у інформаційну безпеку щодо окремих заходів захисту інформації.

Цифрова трансформація та використання цифрових сервісів замість так званих «аналогових» стала дуже популярною. Кожен сегмент промисловості в усьому світі свідомо прямує до цифрових інновацій, щоб випередити своїх конкурентів. Іншими словами, кожен аспект ведення бізнесу наділений цифровими можливостями, щоб отримати всі переваги цифрової парадигми. Усі види бізнесу, що використовує цифрові технології, по всьому світі здатні досягати більших і кращих результатів. Їхні споживачі та клієнти отримують величезні переваги завдяки реальним ініціативам та впровадженням цифрової трансформації. Довгоочікувану трансформацію бізнесу можна легко й елегантно здійснити за допомогою дієвої та виграшної стратегії цифрової трансформації, плану та виконання.

Існує багато доступних технологій цифрової трансформації, які дозволяють спростити та прискорити процес необхідної трансформації. Ці технологічні інновації є достатньо компетентними та універсальними, щоб задовольнити різноманітні вимоги щодо створення та підтримки цифрових підприємств [32].

Сьогодні інформація стала важливою частиною сучасної економіки, і сучасні організації сьогодні залежать у своїй комерційній діяльності від інформації, щоб вижити на ринках. Це означає, що інформація є одним із найважливіших ресурсів, якими володіє організація, тому організаціям потрібна ця інформація бути точним, надійним, готовим у будь-який час за запитом користувача та в правильному форматі, оскільки інформація стала джерелом життя організації. Сучасне технологічне прискорення надає компаніям багато факторів, таких як економія часу, конкуренція, обслуговування клієнтів і відкриття нових інвестиційних можливостей для організацій. Ця технологія вимагає від компаній запровадити політику та процедури безпеки, щоб забезпечити свою безпеку та здатність надавати послуги населенню та завоювати їх довіру.

Технологічний розвиток відкрив нові ринки, і йти в ногу з розвитком стало необхідним для організацій, тому сьогодні інформаційна безпека стала однією з найбільших проблем, які загрожують організаціям, і це джерело занепокоєння для

організацій, оскільки економіка організацій залежить від Тому багато організацій розробляють політику та процедури, щоб зменшити ризики безпеки, з якими ви можете зіткнутися через страх економічного колапсу.

З організаціями, які покладаються на технології, капітал або лідерство більше не є єдиними важливими елементами успіху організації, але тут з'явився новий елемент, яким є інформаційна безпека. Компонент інформаційної безпеки організацій став пріоритетом для організацій, оскільки організації усвідомили, що цей елемент може мати значний негативний вплив на вартість і репутацію організації на ринках, і що стало необхідним захистити організації від ризики та загрози безпеці, які розвиваються щодня. Цей інтерес до компоненту безпеки, у свою чергу, призвів до зростання інтересу організацій до управління інформаційною безпекою.

Управління інформаційною безпекою — це планування, координація, реалізація та контроль діяльності з метою захисту інформації організації. СУІБ займається захистом інформації, незалежно від її форми, та інформаційних активів організації за допомогою впорядкованої та безперервної послідовності операцій, спрямованих на досягнення цілей організації [3]. Ці операції є всеохоплюючою основою для встановлення, функціонування, впровадження, моніторингу, перегляду, підтримки та покращення інформаційної безпеки. Це допомагає організації уникнути ризиків і штрафних санкцій, пом'якшити інциденти безпеки, подолати слабкі сторони та забезпечити безперервність бізнесу шляхом впровадження проактивних заходів безпеки для зменшення ризиків. Таким чином, багато організацій розробили багато стандартів, які відповідають потребам інформаційної безпеки організацій, незалежно від розміру, типу та діяльності організації, і будь-яка організація може адаптуватися до цих стандартів, а також існує багато структур безпеки, які були створені для задоволення потреб конкретної організації. Ці рамки можуть бути недостатньо здійсненними для організації іншого типу чи діяльності.

Використання мережевого файлового сховища, доступного у локальній мережі має кілька переваг. Однією з головних є централізація дорогих ресурсів, таких як диски. Замість того, щоб кожен процесор мав приватний диск, один чи кілька

комп’ютерів можуть надавати службу зберігання для всіх інших у мережі. Також не можна не сказати про те, що сьогодні, можливості файлового сховища все ще відіграють вирішальну роль у захисті файлів, призначених для колективної групи користувачів. Таке файлове сховище є центральним місцем накопичення важливих для бізнесу або мережі файлів. Таке сховище конфігурується відповідно до операційних потреб організації та надає можливість віддаленого доступу до мережевих файлів через підключення до Інтернету. Отже, не можна не відзначити важливість наявності системи управління інформаційною безпекою для захисту такого надбання підприємства від потенційних загроз (зовнішніх або внутрішніх).

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1. Системи управління інформаційною безпекою

### 1.1.1. Загальна інформація

Інформація є джерелом життя організацій, життєво важливим бізнес-активом у сучасному ІТ-світі. Доступ до високоякісної, повної, точної та актуальної інформації є життєво важливим для підтримки процесу прийняття управлінських рішень, який веде до обґрунтованих рішень. Таким чином, безпека ресурсів інформаційної системи є надзвичайно важливою для забезпечення належного захисту ресурсів. Регламенти та різні політики конфіденційності/захисту даних накладають на організації низку зобов'язань. Організаціям необхідно використовувати СУІБ для ефективного управління своїми інформаційними активами. СУІБ в основному складається з наборів політик, створених організацією для визначення, створення, розробки та підтримки безпеки свого комп'ютера на основі апаратних і програмних ресурсів, а також аудиту їх виконання [18].

Організації іноді витрачають значні кошти на брандмауер, проксі-сервер, антивірус, механізм виявлення вторгнень, цифрові підписи, спеціальні мережеві пристрої та протоколи тощо, припускаючи, що безпеку інформації можна якимось чином забезпечити шляхом придбання цих технологічних рішень на ринку [5]. Це неправильне уявлення, оскільки управління безпекою — це більше керування наскрізною системою, а не просто встановлення технічних рішень. Як і будь-яка інша повноцінна система, ця система має багато компонентів, включаючи людей, політику, процедури, процеси, стандарти та технології.

Розробити систему захисту інформації не так просто. Така система повинна базуватися виключно на цінностях і керуватися бізнесом. Необхідно провести

належний аналіз і дизайн із залученням усіх згаданих вище компонентів. Співробітники, починаючи від вищого керівництва і закінчуючи кінцевими користувачами, повинні взяти на себе відповідну роль у створенні та впровадженні системи інформаційної безпеки в організації. Процеси мають бути визначені з конкретними бізнес-цілями для захисту «інформаційних активів» [14]. Технологічні рішення необхідно впроваджувати належним чином для боротьби із загрозами та ризиками або для автоматизації певних процесів. Необхідно встановити політику та процедури, щоб визначити, хто робитиме що, коли та як, щоб запобігти загрозі, виявити її, коли вона виникла, та вжити коригувальних заходів для усунення збитків, якщо такі є. Всередині організації також має відбутися культурна зміна щодо роботи з інформацією та її безпекою в цілому. Наприклад, працівник не повинен передавати свій пароль іншим колегам по-дружньому, поки він/вона йде у відпустку. Натомість він/вона має передати привілеї призначеним особам належним чином і повернути статус після закінчення відпустки. Люди повинні бути морально підготовлені або якось мотивовані прийняти важливість безпеки та дотримуватися правил.

З перспективи бізнесу очевидно, що при створенні СУІБ повинні бути додаткові інвестиції в різні ресурси. Виникає питання, скільки інвестувати? Інвестиції залежатимуть від уразливості, факторів ризику, пов’язаних із бізнесом, а також від його типу, типу та розміру. Очевидно, що такі інвестиції не повинні перевищувати вартість інформації та активів, що захищаються. Нажаль, більшість організацій витрачають 10-13% свого загального ІТ-бюджету на захист інформації. Перевага від наявності відповідної СУІБ завжди виправдовує такі інвестиції.

Оскільки організації стають все більш залежними від інформаційних систем для стратегічної переваги та операцій, питання безпеки інформаційних систем також стає все більш важливим. У сучасному взаємопов’язаному електронному бізнес-середовищі питання безпеки мають першорядне значення [16].

Керівництво має інвестувати в безпеку ІБ, щоб запобігти зловживанням, які можуть призвести до невигідного конкурентного становища. Для кожної організації,

малої чи великої, виникає потреба мати СУІБ, щоб виявляти, керувати та захищати цінні ресурси, такі як обладнання, програмне забезпечення та кваліфіковані люди. Компоненти (апаратне забезпечення, програмне забезпечення, процеси, політики, люди) мають бути пов’язані в систему, яку слід ретельно впроваджувати для боротьби з існуючими та новими загрозами безпеці [8]. Така система називається системою управління інформаційною безпекою, результатом одного з найбільш стратегічних корпоративних рішень і основою інформаційної безпеки в організації.

Загроза інформації, що зберігається організаціями, стає все більш серйозною, і, з іншого боку, зростає залежність майже всіх організацій щодо відповідної безпечної обробки інформації. Стандарти управління інформацією та збірники найкращих практик добре встановлені та задокументовані. Найважливішими та найбільш прийнятими міжнародними стандартами для розробки та експлуатації СУІБ є ISO 27001-27006 [7].

ISO 27001 визначає вимоги щодо планування, впровадження, експлуатації та постійного моніторингу та вдосконалення СУІБ [8], орієнтованої на процес, але структура процесів не представлена в ISO 27001. Стандарти інформаційної безпеки зосереджуються на існуванні процесів, а не на їх контент і стандарти ISO/IEC серії 27000 зосереджені на вимогах, засобах контролю безпеки та орієнтації для впровадження СУІБ в організації. ISO 27001 безпосередньо посилається на цикл «Плануй-Виконуй-Перевіряй-Дій» із класичного управління якістю Демінга, який наголошує на необхідності орієнтації на процес, а також інтеграції планування операцій і постійної перевірки відповідності плануванню впровадження. Отже, основна проблема полягає в тому, що ISO 27001 не надає моделі процесу для процесів СУІБ [10]. Крім того, ISO 27001 містить засоби контролю інформаційної безпеки, які також частково призводять до впровадження процесів.

Але які процеси слід включити до СУІБ? Для більшості організацій немає альтернативи процесно-орієнтованому менеджменту. Управління процесами — це не просто спосіб вирішення конкретних проблем, це також платформа для капіталізації нових можливостей, що є навіть більш важливим у сучасному суспільстві обробки інформації. У літературі повідомлялося про дослідження та

досвід щодо 27001, але, знову ж таки, фактичної структури процесу для управління безпекою, яка б чітко розрізняла процеси СУІБ та заходів безпеки чи контролю, ініційованих процесами СУІБ, не існує [37]. Таким чином, важливо розробити детальну, але також загальну структуру основного процесу СУІБ, яка може бути легко прийнята та впроваджена всіма організаціями [11]. Як передумова для цього необхідно розробити критерії для ідентифікації основних процесів СУІБ, оскільки, наскільки відомо авторам, не існує набору критеріїв, ідентифікованих і перевірених, а також загально узгоджених для основних процесів СУІБ. Автори цього дослідження вирішили цю проблему, провівши першу частину дослідження з експертним опитувальником для визначення відповідних критеріїв. Щоб досягти головної мети — розробити детальну, але також загальну структуру основних процесів СУІБ, друга частина дослідження мала на меті визначити відповідні основні процеси СУІБ за допомогою опитувальника експертів.

СУІБ зазвичай стосується поведінки та процесів, що відносяться до працівників, а також даних і технологій. СУІБ може бути націлена на певний тип даних, як-от дані (файли) організації, або може бути реалізована комплексним способом, який стане частиною культури компанії [17].

СУІБ зазвичай має сенс для всіх компаній, незалежно від галузі та розміру компанії. Основна увага зосереджена на компаніях, керованих програмним забезпеченням, цифрових і SaaS. СУІБ забезпечує системний підхід до управління інформаційною безпекою організації. Інформаційна безпека охоплює певні широкі політики, які контролюють і керують рівнями ризику безпеки в організації. Оскільки інформаційна безпека відіграє дуже важливу роль у підтримці діяльності організації, нам потрібен стандарт або еталон, який регулює управління інформаційною безпекою. Існує кілька стандартів управління IT, які забезпечують інформаційну безпеку, наприклад PRINCE2, OPM3, CMMI, P-CMM, PMMM, ISO27001, BS7799, PCIDSS, COSO, SOA, ITIL та COBIT [16].

Системи управління інформаційною безпекою в компаніях орієнтовані на процес і завжди є відповідальністю керівництва, використовуючи підхід зверху вниз. Виконання, але не відповідальність, може бути делеговане. Залежно від

потреб, керівництво обирає процедури та методи, які слід застосувати або побудувати для забезпечення інформаційної безпеки в корпоративній діяльності. Керівництво має регулярно перевіряти масштаби, інтенсивність та прогрес заходів.

Метою СУІБ є не досягнення максимальної безпеки інформації. Швидше, це досягнення бажаного рівня інформаційної безпеки організації. Схильність до ризику є ключовою [4]. Корпорація повинна знати свою інформацію, ризики та фінансовий вплив матеріалізованого ризику. Базуючись на цих знаннях, керівництво має вирішити, до якої міри слід зменшити ризики за допомогою СУІБ.

Метою СУІБ є не обов'язкове максимізація інформаційної безпеки, а радше досягнення бажаного для організації рівня інформаційної безпеки. Залежно від конкретних потреб галузі ці рівні контролю можуть змінюватися. Наприклад, оскільки охорона здоров'я є строго регульованою сферою, організація охорони здоров'я може розробити систему для забезпечення повного захисту конфіденційних даних пацієнтів [19].

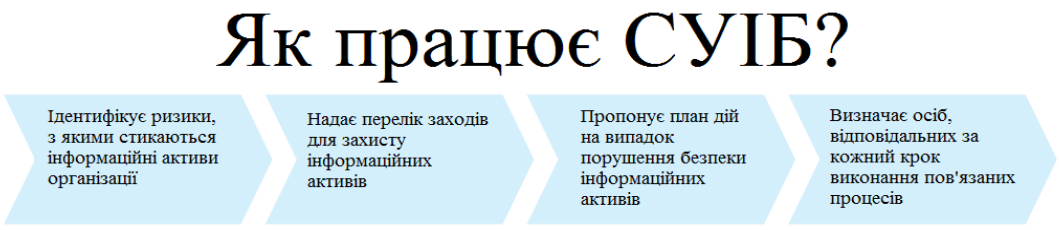


Рис. 1.1. Логіка роботи СУІБ

СУІБ забезпечує систематичний підхід до управління інформаційною безпекою організації та включає політику та процедури для управління її даними. Організаціям необхідно визначати та керувати багатьма видами діяльності, щоб функціонувати ефективно та результативно. Будь-якою діяльністю з використанням ресурсів необхідно керувати, щоб уможливити перетворення входів у результати за допомогою набору взаємопов'язаних або взаємодіючих дій — це також відомо як процес. Іншими словами, процес — це набір взаємопов'язаних або взаємодіючих дій, які перетворюють входи на результати — ціль процесу. Бізнес-процес

описується як процедура, важлива для додавання вартості організації. Процеси можуть бути частиною інших процесів, містити інші процеси або ініціювати інші процеси. На відміну від проектів процеси можна виконувати багаторазово [17]. Процеси часто виконуються між відділами та є частиною операційної структури організації. Процеси розрізняються на основні, управлінські та допоміжні. Основні процеси створюють очевидну та пряму цінність для споживачів і є похідними від компетенції організації.

Процеси управління визначають цілі організації, а також контролюють і контролюють досягнення цілей на рівні основних процесів і всієї організації. Вони містять проект, якість, управління безпекою та ризиками, а також стратегічне планування. Допоміжні процеси забезпечують і керують необхідними ресурсами, не забезпечуючи безпосереднього споживача. Вони підтримують основні процеси та процеси управління. Типовими допоміжними процесами є управління персоналом, управління фінансами та IT [12].

СУІБ забезпечує цілісний підхід до управління інформаційними системами в організації. Це забезпечує численні переваги, деякі з яких висвітлено нижче. СУІБ захищає всі типи пропрієтарних інформаційних активів, незалежно від того, є вони на папері, зберігаються в цифровому вигляді чи знаходяться в хмарі. Ці активи можуть включати персональні дані, інтелектуальну власність, фінансові дані, дані клієнтів і дані, довірені компаніям через треті сторони.

СУІБ допомагає організаціям виконувати всі нормативні та договірні вимоги, а також забезпечує краще розуміння законності навколо інформаційних систем. Оскільки порушення законодавчих норм супроводжується значними штрафами, наявність СУІБ може бути особливо корисною для суворо регульованих галузей із критично важливою інфраструктурою, як-от фінанси чи охорона здоров’я [50].

Коли організації інвестують у СУІБ, вони автоматично підвищують свій рівень захисту від загроз. Це зменшує кількість інцидентів безпеки, таких як кібератаки, що призводить до меншої кількості збоїв і простою, що є важливим фактором для підтримки безперервності бізнесу [50].

СУІБ пропонує ретельну оцінку ризиків усіх активів. Це дає змогу організаціям визначити пріоритетність активів із найвищим ризиком, щоб запобігти невибірковим витратам на непотрібні засоби захисту та забезпечити цілеспрямований підхід до їх захисту [50]. Цей структурований підхід, а також менший час простою через зменшення інцидентів безпеки значно скорочує загальні витрати організації.

СУІБ забезпечує всеохоплюючий підхід до безпеки та управління активами в усій організації, не обмежуючись ІТ-безпекою. Це заохочує всіх співробітників розуміти ризики, пов’язані з інформаційними активами, і застосовувати найкращі методи безпеки як частину своєї щоденної роботи. Загрози безпеці постійно розвиваються. СУІБ допомагає організаціям підготуватися та адаптуватися до нових загроз і постійно мінливих вимог середовища безпеки.

### 1.1.2. Побудова структури СУІБ

Прийнявши принципи стандартів ISO 17799, ISO/IEC 27001 і норми правил відповідності, структура СУІБ може бути розроблена чином, що описано нижче.

Політика інформаційної безпеки: розробіть документ із політикою інформаційної безпеки з чітким обсягом і межами, враховуючи тип бізнесу, його місцезнаходження, активи та технологію з належним обґрунтуванням того, що будь-яка сфера виключається зі сфери й меж. Цей документ містить огляд потреб безпеки та найвищого рівня схеми безпеки. Він визначає активи, що підлягають захисту, і ступінь їх захисту [18]. Це офіційно затверджений документ політики корпоративного управління та доказ того, що керівництво вжило відповідних заходів щодо встановлення інформаційної безпеки для захисту інформації організації від усіх можливих загроз [49].

Організація інформаційної безпеки визначає групу інформаційної безпеки з розподілом обов’язків і зобов’язань. Налаштуйте процес авторизації, конфіденційність, умови нерозголошення та належні процедури зв’язку в цій структурі безпеки. Встановіть умови безпеки для зовнішніх сторін (постачальників, партнерів, підрядників і постачальників).

Управління активами визначає інформаційні активи з відповідальними власниками. Необхідно визначити правила прийнятного використання цих активів з точки зору безпеки, класифікувати активи, використовуючи будь-який стандартний механізм класифікації, як-от «Конфіденційний», «Приватний» і «Загальнодоступний», а також процедури обробки, маркування та утилізації [24].

Безпека людських ресурсів встановлює процедуру перевірки безпеки в процесі найму (працівників, сумісників, підрядників), включаючи умови нерозголошення в трудовій угоді. Необхідно визначити ролі та обов'язки безпеки в описі посади та включити умови безпеки при переведенні, відпустці, звільненні, виході на пенсію тощо з відповідними пунктами дисциплінарних заходів у разі порушення [18].

Фізична безпека та безпека навколишнього середовища забезпечує фізичну безпеку в зонах інформаційних і комп'ютерних засобів, включаючи комп'ютерний центр, зону доставки, зону збору, пункти утилізації/видалення. Тут необхідно забезпечити охорону та технічне обслуговування меж, навколишнього середовища, протипожежного захисту, кондиціонування повітря, кабелів, електропостачання, замків та сигналізації, а також створити систему реєстрації користувачів, відвідувачів та обладнання, які входять або виходять із зон інформаційних засобів [18].

Управління зв'язком та операціями записує процедури та обов'язки для всіх пов'язаних операцій, включаючи ведення господарства, управління змінами/оновленнями, розподіл обов'язків, критерії прийняття та розгортання програмного забезпечення або послуг (внутрішні, аутсорсинг), захист мережі (провідної, бездротової, мобільної), електронна комерція, синхронізація годинника, резервне копіювання, відновлення, обмін або передача носіїв даних, обмін зв'язком, використання електронної пошти, факсу та обробка публічної інформації, щоб забезпечити безпеку та правильність обробки інформації. Механізми державного моніторингу, включаючи ведення аудитів і журналів [18].

Контроль доступу визначає процедури та обов'язки для всіх завдань, пов'язаних із доступом. Це включатиме створення/реєстрацію користувача для мережі (дротової, бездротової, мобільної та телефонної мережі), операційної

системи, програми та баз даних, розподіл прав і привілеїв, використання системних утиліт, критерії відкриття/закриття порту, моніторинг пароля та доступ до критично важливих систем тощо [24]. Варто контролювати доступ до інформації шляхом ведення аудитів і журналів [18].

Придбання, розробка та технічне обслуговування інформаційної системи (власне або зовнішнє) вказує формальні вимоги до засобів контролю безпеки під час розробки нової системи, оновлення чи модифікації будь-якої існуючої системи, тестування, впровадження, обробки, введення/виведення, перевірки повідомлень та зміни операційної системи. Тут необхідно вказати процедури для захисту джерел і об’єктів програми, визначити процедури використання шифрування, цифрових підписів, сертифікатів та інфраструктури відкритих ключів (PKI), де це необхідно та переконатися, що будь-яке програмне забезпечення сторонніх розробників не містить шкідливих програм [24].

Управління інцидентами інформаційної безпеки визначає обов’язки та процедури для вирішення всіх можливих інцидентів безпеки та недоліків. Необхідно створити плани на випадок непередбачених ситуацій для швидкого відновлення систем або послуг у разі збоїв, вказати механізм зв’язку, звітності, збору доказів, журналів, аудитів, аналізу, документування інциденту та рішення [18].

Управління безперервністю бізнесу визначає процеси безперервності бізнесу, визначивши та встановивши пріоритети для критичних бізнес-сфер. Це потребує аналізу загроз і наслідків для подій, які можуть призвести до переривання бізнесу, розробки узгодженої структури безперервності бізнесу з точки зору часового розриву між невдачею та безперервністю [18]. Варто також розробити комплексну політику збереження та резервного копіювання даних разом із процедурами відновлення, організувати профілактичне обслуговування всього критичного обладнання, мережі, програмного забезпечення, баз даних і програм. Також, варто розробити реплікації або резервного онлайн-сайту для критично важливої системи електронної комерції, навчити співробітників планам безперервності бізнесу разом із тестами в реальному часі, а також ефективно оновлювати плани безперервності відповідно до будь-яких змін у бізнесі чи політиці [24].

Відповідність вимогам безпеки вимагає дотримання усіх юридичних вимог, застосовних до вашого бізнесу згідно з місцевими чи міжнародними правилами, визначаючи потреби та визначаючи процедури впровадження з розподілом обов'язків. Варто дотримуватись відповідних прав інтелектуальної власності та авторських прав на програмне забезпечення, захищати записи організації та конфіденційність особистої інформації, виконувати перевірки відповідності безпеки в питаннях політики, а також у технологічних сферах, щоб переконатися, що відповідні процедури дотримуються для досягнення прийнятного рівня відповідності та проводити системні аудити відповідно до законодавчих актів урядом та іншими органами [18, 24].

### 1.1.3. Процес планування СУІБ

Процес планування СУІБ — це процес специфікації та проектування СУІБ від початку до створення планів впровадження. Процес контролю документації та записів — це процес ідентифікації, створення, оновлення та контролю інформації, визначеної як необхідна для ефективності СУІБ [48].

Ключем до досягнення цілей СУІБ є сучасне розуміння потреб і очікувань зацікавлених сторін, що стосуються інформаційної безпеки та СУІБ. Це реалізується в рамках процесу управління вимогами, який забезпечує визначені юридичні, законодавчі, нормативні та договірні вимоги до процесу оцінки ризиків, процесу внутрішнього аудиту та процесу контролю процесів, переданих аутсорсингу.

У процесі оцінки ризиків ризики ідентифікуються, аналізуються та оцінюються. Результатом цього процесу є задокументовані та оцінені ризики у списку пріоритетних ризиків, включаючи загрози, уразливості та власників ризиків, наслідки та вплив на бізнес, ймовірність і порівняння з критеріями ризику, а також оцінені ризики запропонованих змін, які є вхідними для комунікації процес і процес обробки ризиків інформаційної безпеки [48].

У процесі обробки ризиків інформаційної безпеки ідентифікуються та вибираються варіанти обробки ризиків, включаючи цілі контролю та засоби контролю [48]. Результатом цього процесу є список із вибраними засобами

контролю та цілями контролю, план обробки ризиків, включаючи прийняття залишкових ризиків, план впровадження контролю та запити на зміни в процесі управління змінами інформаційної безпеки, які використовуються як вхідні дані в різних процесах СУІБ.

Ресурси, необхідні для впровадження засобів контролю, а також для запуску процесів СУІБ, визначаються, розподіляються та контролюються в процесі управління ресурсами. Результатом процесу управління ресурсами є заплановані/задокументовані ресурси для впровадження та запуску вибраних засобів контролю, категоризація засобів контролю щодо того, хто фінансує контроль, заплановані та задокументовані ресурси для запуску основних процесів СУІБ, звіти щодо використання ресурсів основних процесів СУІБ та для процес управління взаємовідносинами з клієнтами інформаційної безпеки: звіти про використання ресурсів [46]. Впровадження засобів контролю завжди призводить до змін, якими можна керувати в рамках загального процесу управління змінами організації, що впроваджує, або — якщо зміна зосереджена на елементі СУІБ — в рамках процесу управління змінами інформаційної безпеки. Процес управління змінами в інформаційній безпеці — це процес контролю змін елементів СУІБ та перегляду наслідків ненавмисних змін. Цей процес зосереджений лише на управлінні змінами СУІБ [46]. Результатом цього процесу є необхідні зміни (для процесу контролю документації та записів), запропоновані та необхідні зміни, а також результати змін (для та від процесу оцінки ризику), ініціювання оцінки ризику, коли пропонуються або відбуваються значні зміни та результати зміни в процесі управління інцидентами інформаційної безпеки, оскільки вони були ініційовані цим процесом.

Процес управління інцидентами інформаційної безпеки призначений для виявлення інцидентів інформаційної безпеки, звітування, оцінювання, реагування на них, роботи з ними та навчання на них. Результатом цього процесу є ідентифіковані інциденти, які використовуються в різних процесах СУІБ, включаючи процес управління змінами інформаційної безпеки та процес забезпечення необхідної обізнаності.

У процесі інформування про інформаційну безпеку розробляється та впроваджується програма інформування про інформаційну безпеку, навчання та навчання, щоб гарантувати, що весь персонал отримує необхідну підготовку та/або освіту з питань безпеки [46]. Оскільки послуги передаються на аутсорсинг, ці послуги необхідно визначати та контролювати, що реалізується в рамках процесу контролю за послугами, переданими на аутсорсинг.

Процес оцінки ефективності включає моніторинг, вимірювання, аналіз та оцінку двох основних критеріїв. По-перше, продуктивність засобів контролю безпеки, а по-друге, продуктивність процесів СУІБ. Вимірювання ефективності відрізняється від аудиту ефективності (внутрішнього аудиту) щодо результативності та ефективності СУІБ та запроваджених засобів контролю, які виконуються незалежно в рамках процесу внутрішнього аудиту.

Результати процесу оцінки ефективності, процесу внутрішнього аудиту, а також результати аудиту постачальника послуг у процесі контролю послуг, переданих аутсорсингом, використовуються для підвищення ефективності, ефективності, придатності та адекватності СУІБ та засобів контролю. Це реалізується в рамках процесу підвищення інформаційної безпеки.

Результати майже всіх процесів СУІБ централізовано передаються в рамках процесу комунікації зацікавленим сторонам за межами СУІБ. Це включає повідомлення про ризики та звіти про управління інформаційною безпекою. Ці звіти, а також визначені вимоги є вхідними для процесу управління інформаційною безпекою, який забезпечує узгодження СУІБ з цілями та потребами керівних зацікавлених сторін.

Окрім процесу управління інформаційною безпекою, який формує взаємодію між СУІБ та її зацікавленими сторонами, необхідно реалізувати оперативне управління рівнем задоволеності клієнтів, а також постійну демонстрацію додаткової вартості інвестицій в інформаційну безпеку. Це робиться в рамках процесу управління взаємовідносинами з клієнтами інформаційної безпеки.

### 1.1.4. Компоненти та впровадження СУІБ

Як вже було зазначено вище, система управління безпекою інформації включає наступні компоненти [47]: 1) управління ризиками: на основі показників конфіденційності, цілісності та доступності; 2) повне управління якістю: на основі показників ефективності та результативності; 3) моделювання моніторингу та звітності: заснована на рівнях абстракції; 4) структурований підхід: включає людей, процес і технологію; 5) розширювана структура, за допомогою якої можна керувати відповідністю інформаційної безпеки.

Система управління інформаційною безпекою забезпечує вимоги до створення, впровадження, підтримки та вдосконалення системи управління інформаційною безпекою. Це прийняття є стратегічним рішенням для організації, на яке вплинули потреби та цілі організації, вимоги безпеки та масштабовано відповідно до потреб організації. Система управління інформаційною безпекою застосовує процес управління ризиками для захисту конфіденційності, цілісності та доступності інформації. СУІБ може використовуватися внутрішніми та зовнішніми сторонами та описується ISO/IEC 27000 [1]. Він надає каталог елементів керування, які можуть бути реалізовані для СУІБ.

Система управління інформаційною безпекою забезпечує вимоги до створення, впровадження, підтримки та вдосконалення системи управління інформаційною безпекою. Це прийняття є стратегічним рішенням для організації, на яке вплинули потреби та цілі організації, вимоги безпеки та масштабовано відповідно до потреб організації. Система управління інформаційною безпекою застосовує процес управління ризиками для захисту конфіденційності, цілісності та доступності інформації. СУІБ може використовуватися внутрішніми та зовнішніми сторонами та описується ISO/IEC 27000 [1]. Він надає каталог елементів керування, які можуть бути реалізовані для СУІБ. СУІБ включає наступні основні компоненти (див. рис. 1.2): принципи управління, ресурси, персонал, і процеси захисту інформації:

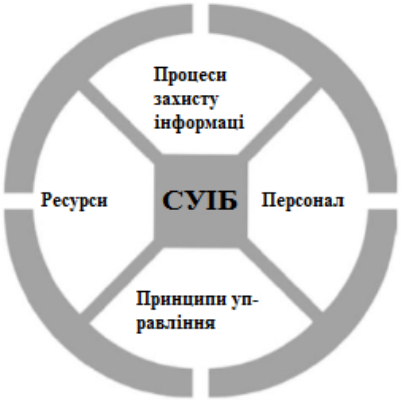


Рис. 1.2. Компоненти системи управління інформаційною безпекою

Система управління інформаційною безпекою є невід'ємною частиною системи управління та бізнес-культури організації. Ця система містить організаційні структури, планування, політику, процеси та ресурси. Далі наведено список кроків, які можна виконати під час впровадження системи управління інформаційною безпекою в організації разом із структурою СУІБ [9].

1) Формування технічної групи з питань інформаційної безпеки, секцію, департамент або відділ із призначенням голови, скажімо, керівника служби безпеки, якщо це не було зроблено раніше.

2) Визначення виконавчого комітету для проекту інформаційної безпеки, схвалений і до складу якого входять люди з Ради директорів, генерального директора, інформаційного директора, вищих менеджерів і менеджерів.

3) Визначення групи впровадження безпеки, яка складатиметься з обраних людей з технічної групи, виконавчого комітету та відповідних бізнес-сфер в організації.

4) Навчання команди впровадження найновішим стандартам, вимогам і найкращим практикам для такого роду проектів управління інформаційною безпекою. Дотримання стандартів на кожному кроці, щоб не пропустити жодної важливої сфери. При необхідності зверніться за допомогою до кваліфікованих консультантів.

5) Проведення всебічного аналізу ризиків і вразливостей, пов'язаних з різними «інформаційними активами», щодо їх класифікації безпеки, встановленої раніше, і їхньої ролі в бізнесі.

6) Визначення для кожного активу, який потрібно захистити, додаткових вимірювань безпеки, необхідних щодо існуючих, і можливих ресурси, необхідних для цієї мети. Створіть матричну діаграму як результат цієї вправи.

7) Створення проектної пропозиції з планом впровадження разом із вимогами до ресурсів (поточних і поточних), включаючи обладнання, програмне забезпечення, навчання, робочу силу та зобов'язання. Винесення проектної пропозиції на затвердження виконавчого комітету. Включення рекомендованих змін, якщо такі є, і створення остаточної затвердженої пропозиції.

8) Створення документації політик і процедур, які охоплюють усі відповідні сфери, проаналізовані раніше, і використовуючи структуру СУІБ.

Політика безпеки зазвичай окреслює вимоги високого рівня до елементів керування безпекою або правил, які мають бути виконані для певної області чи активу. Процедура безпеки — це зазвичай набір дуже чітко написаних покрокових вказівок для конкретної системи, яких настійно рекомендуємо дотримуватися для ефективного впровадження відповідної політики [9].

Наприклад, у політиці «Резервне копіювання та зовнішнє зберігання медіафайлів» може бути зазначено, що резервне копіювання критично важливих баз даних компанії має створюватися щодня о 1:00 ночі, а стрічкові носії мають передаватися у зовнішній сейф до 10:00 наступного дня. Процедура для наведеної вище політики містить деталі всіх критичних баз даних, включаючи тип резервного копіювання (холодне резервне копіювання, гаряче резервне копіювання), цикли резервного копіювання (повне, інкрементне), політику передачі та ротації (період зберігання) носія, маркування касет, збереження замка та ключа, ведення журналу чи реєстру тощо [9]. Кожна політика та процедура адресована відповідній аудиторії (користувачам, менеджерам, ІТ-персоналу, партнерам, постачальникам). Загалом будь-яка політика чи процедура матиме такі розділи, як сфера застосування, цілі, аудиторія, обов'язки, вимоги, положення про виконання, дата перегляду, визначення термінів та інші посилання, де це можливо. Іноді політики та процедури написані в одному документі. У такому випадку наведена вище політика та її процедура

можуть бути окремим розділом усього документа з назвою розділу «Резервне копіювання та зовнішнє зберігання медіафайлів».

Після того, як усі політики та процедури охоплено, рекомендовано підготувати зведену матрицю охоплення, яка показує проблеми політики та підполітики по рядках із заголовками стовпців як результати аналізу, номер політики, заява про політику, контрольний номер процедури, аудиторія, відповідальність , статус і коментарі.

9) Надання документів щодо політики та процедур (або єдиного комбінованого документу) виконавчому комітету для доопрацювання та остаточного затвердження керівництвом.

10) Впровадження проекту поетапно відповідно до пріоритетів. Фази повинні бути визначені раніше в плані впровадження. Будь-які закупівлі обладнання, програмного забезпечення тощо мають бути пов’язані з етапами впровадження.

11) Разом із іншими вимогами рекомендовано дотримуватися деяких найкращих практик, таких як стандарти ITIL, COBIT і ISF, як зазначено в списку відповідностей.

12) Проведення інформаційного тренінгу для працівників, щоб вони зрозуміли політику та процедури безпеки, які мають на них поширюватися. Переконайтеся, що політика та процедури надіслані всім підрозділам і департаментам організації та зовнішнім сторонам, якщо це можливо.

13) Впровадження процедури для періодичних тестів на вразливість і проникнення в чутливих областях.

14) Проведення аудиту системи інформаційної безпеки та її функціонування компетентним аудитором, визнаним уповноваженим органом, таким як Асоціація аудиту та контролю інформаційних систем (ISACA) [24].

Звичайно, кроки впровадження відрізнятимуться залежно від поточного стану інфраструктури безпеки. Вищезазначені кроки були сформульовані з припущенням, що проект розпочнеться з нуля та завершиться успішним впровадженням системи СУІБ із подальшим отриманням сертифікату безпеки ISO.

Управління інформаційною безпекою є безперервним процесом [11]. СУІБ разом із політикою, процедурами та відповідністю слід переглянути й оновити відповідно до останніх ринкових тенденцій і вимог. Цикл перегляду, аналізу недоліків і оновлення забезпечить довгострокову вигоду для організації шляхом захисту її ІТ-активів найефективнішим способом.

Розуміння важливості впровадження безпеки є дуже важливим. Якщо працівники не розуміють необхідності цього, вони можуть не брати участь у реалізації всім серцем, і це може призвести до провалу проекту або затримки досягнення результатів. Вище керівництво, будучи головним спонсором і мотиватором проекту, відіграє важливу роль у цьому питанні з самого початку.

Рішення безпеки має бути ретельно розроблено, щоб досягти економічної ефективності та повернення інвестицій (ROI), додаючи бізнес-цінності, окрім дотримання нормативних вимог, пам’ятаючи, що інвестиції в інформаційну безпеку є витратами на страхування, які захистять інформацію організації від втрати або знищення, уникаючи простоїв і таким чином підвищуючи продуктивність. Успіх системи управління інформаційною безпекою полягає в найкращому поєднанні людей, політик, процедур, відповідності, стандартів, процесів, продуктів і технологій.

## 1.2. Стандартизація

### 1.2.1. Загальна інформація

Інформація та інформаційні системи є важливою основою для компаній. Зокрема, дедалі більше внутрішньої та міжкорпораційної передачі даних і використання відкритих мереж збільшують ризики, яким піддаються інформація та інформаційні системи. Щоб зменшити ризики та уникнути збитків для компаній, необхідно подбати про забезпечення належної безпеки інформації. Для захисту інформації та інформаційних систем стандарт ISO 27001 забезпечує контрольні цілі, спеціальні засоби контролю, вимоги та вказівки, за допомогою яких компанія може досягти належної інформаційної безпеки [1]. Таким чином, ISO 27001 дає змогу компанії отримати сертифікат відповідно до стандарту, за допомогою якого

інформаційна безпека може бути задокументована як суворе застосування та управління відповідно до міжнародно визнаного організаційного стандарту. Завдяки сертифікації відповідно до ISO 27001 компанія перевіряє виконання загальновідомих і прийнятих стандартів безпеки і таким чином сприяє довірі клієнтів. Подібним чином перевірка відповідності міжнародному стандарту зменшує ризик штрафів або компенсаційних виплат у результаті судових суперечок, оскільки правові вимоги, такі як надання резервів відповідно до «сучасного рівня» та з «належною обачністю», можуть протистояти дотриманню стандартів [3].

Інформація та інформаційні системи все більше піддаються ризикам через зростання підтримки бізнес-процесів, що надається інформаційними технологіями, а також підвищення рівня мережевих зв'язків у компаніях і з зовнішніми сторонами. Ефективна СУІБ допомагає знизити ризики та запобігти порушенням безпеки [7]. Стандарт ISO 27001 є частиною структури для проектування та експлуатації СУІБ, заснованої на багаторічному досвіді розробки. Завдяки цьому компаніям пропонується можливість узгодити свої IT-процедури та методи для забезпечення належного рівня інформаційної безпеки з міжнародними стандартами. Сертифікація СУІБ відповідно до ISO 27001 також створює позитивний імідж через перевірку систематичного управління інформаційною безпекою. Цей стандарт також використовується в правових постановах як мірило та основа для оцінки щодо інформаційної безпеки — тут сертифікат згідно з ISO 27001 доводить «надання найсучасніших послуг» щодо інформаційної безпеки. Організації можуть продемонструвати, що вони «достатньо готові» для безпечного надання IT-послуг. Сертифікат може здійснювати перевірку відповідності щодо інформаційної безпеки. Стандарт ISO 27001 отримав широке поширення в Європі та Азії. Значення сертифікації сумісної інформаційної безпеки з рішеннями щодо закупівель IT-послуг зростатиме, тому також слід очікувати подальшого збільшення кількості сертифікацій відповідно до ISO 27001.

За наявності всіх стандартів, політик, процедур, аудиту тощо все ще мають місце помилки, крадіжки даних та особистих даних, відмивання грошей, шахрайство, скандали через зловживання інформацією та катастрофи [24]. Деяким

організаціям, які постраждали від таких збоїв і катастроф, довелося зіткнутися з серйозними наслідками, включаючи навіть припинення роботи. Вважаючи, що ці компанії не вжили достатніх заходів для захисту від цих загроз і для того, щоб звести до мінімуму випадки таких катастроф, уряди та компетентні органи змушують організації дотримуватись певних правил відповідності як законодавчих повноважень відповідно до типу та виду їхнього бізнесу . Ці відповідності та правила введено в дію, щоб доповнити стандарти безпеки ISO [21].

У недавньому минулому ці відповідності позиціонувалися як життєво важливі вимоги для забезпечення інформаційної безпеки [22]. Наприклад, Закон Сарбейнса-Окслі (SOX) є одним із найважливіших законів, що регулюють корпоративне управління та розкриття фінансової інформації для місцевих і глобальних організацій, які ведуть бізнес у США. Ключовим питанням впровадження SOX є вимірювання та планування прийнятних рівнів відповідності для IT-систем. Генеральні директори, фінансові директори та інформаційні директори (C-рівень) беруть на себе відповідальність і підписують фінансові звіти, підтверджуючи, що рівні контролю над процесами фінансової звітності та безпекою, точністю та надійністю пов’язаних інформаційних систем є достатніми та відповідають бізнес-нормам.

Швидке зростання використання інформаційних технологій у різних бізнесах і перехід конфіденційної інформації в цифрові записи призвели до створення багатьох таких нормативних документів, інструкцій, правил і регулюючих органів. Нижче наведено список деяких із них [21].

1) Закон Сарбейнса-Окслі (SOX) — в обов'язковому порядку застосовується до всіх публічних компаній;

2) Закон про перенесення та підзвітність медичного страхування (HIPAA) — стосується будь-якої організації, яка обробляє інформацію про здоров’я особи;

3) Закон Гремма-Ліча-Блайлі (GLBA) — застосовується до будь-якої фінансової установи та компаній, які надають послуги установі;

4) Закон Каліфорнії про повідомлення про порушення безпеки (раніше SB 1386) — вимагає від компаній, які зберігають дані про жителів Каліфорнії,

повідомляти осіб про будь-які порушення безпеки, пов’язані з їхньою особистою інформацією;

5) Європейський захист даних (European Safe Harbor Registration) — норми безпеки даних для всіх міжнародних фірм, які мають офіси як у США, так і в ЄС;

6) Президентські директиви внутрішньої безпеки (HSPD12) — директиви щодо загального стандарту ідентифікації для федеральних службовців і підрядників США;

7) Федеральна рада з перевірки фінансових установ (FFIEC) — рекомендації щодо розширеної багаторівневої процедури автентифікації для банківських установ;

8) Комітет спонсорської організації Торговельної комісії (COSO) — загальне визначення внутрішнього контролю, стандарти та критерії, за якими організації можуть оцінювати свої системи контролю;

9) Індустрія платіжних карток (PCI), стандарти безпеки даних (DSS) — регулюють стандарти безпеки для більшості платіжних галузей (Visa, MasterCard тощо);

10) Закони про свободу інформації 2000 року — урядове законодавство Великобританії, що визначає, яку інформацію організації державного сектору зобов’язані надавати на запит;

11) Федеральний закон про управління інформаційною безпекою (FISMA) — Федеральний закон США, як Закон про електронний уряд, накладає обов’язковий набір процесів, яких необхідно дотримуватися;

12) Цілі контролю для інформаційних і пов’язаних технологій (COBIT) — найкращі практики для кращого контролю, аудиту та вимірювання;

13) Бібліотека інфраструктури інформаційних технологій (ITIL) — найкращі практики для кращих IT-послуг;

14) Стандарт належної практики Форуму інформаційної безпеки (ISF) — посібник з управління бізнес-ризиками, пов’язаними з інформаційними системами організації;

15) Положення про стандарти аудиту (SAS) 70 — визначає стандарти аудиту для оцінки внутрішнього контролю за договором обслуговуючої організації.

Окрім демонстрації узгодженості з політикою та процедурами безпеки відповідно до стандартів ISO 17799/27001, організації повинні встановити вищезазначені відповідності, правила та стандарти аудиту відповідно до характеру свого бізнесу. Порушення цих правил може призвести до неприйнятних перевірок, штрафних санкцій, відповідальності, покарань для керівників С-рівня і навіть до повного закриття бізнесу.

### 1.2.2. Огляд деяких стандартів та документів з інформаційної безпеки

#### 1.2.2.1. ISO

Стандарти виникають через розробку детальних описів конкретних характеристик продукту чи послуги експертами компаній і наукових установ. Вони являють собою консенсус щодо таких характеристик, як якість, безпека та надійність, які повинні залишатися застосовними протягом тривалого періоду часу, тому їх документують і публікують. Метою розробки стандартів є підтримка як фізичних осіб, так і компаній при закупівлі продуктів і послуг [1]. Постачальники продуктів і послуг можуть підвищити свою репутацію, сертифікувавши свою відповідність стандартам. Організація ISO заснована 23 лютого 1947 р. Вона оприлюднює всесвітні приватні промислові та комерційні стандарти, має штаб-квартиру в Женеві, Швейцарія. До неї входять 163 національні члени із 203 країн світу. Стандарти ISO 27000 до ISO 27002 були розроблені у співпраці з «Міжнародною електротехнічною комісією» (IEC), яка є провідним світовим розробником міжнародних стандартів у галузі електроніки та електронних технологій.

#### 1.2.2.2. ITIL

Бібліотека IT-інфраструктури (ITIL) є найкращою практикою для управління IT-послугами. Управління IT-послугами — це управління всіма процесами, які взаємодіють для забезпечення якості живих IT-послуг, відповідно до рівнів обслуговування, погоджених із клієнтами [23]. Основна мета управління послугами — забезпечити відповідність IT-послуг потребам бізнесу та їх активну підтримку.

ITIL був розроблений Центральним обчислювальним і телекомунікаційним агентством — нині Управлінням урядової торгівлі — і сьогодні доступний у третій версії. ITIL містить п'ять книг:

1) Service Strategy — це керівництво для розробки та впровадження управління послугами як стратегічного активу. Стратегія надання послуг забезпечує управління витратами та ризиками портфеля послуг. Не лише зосереджуючись на операційній ефективності, він також забезпечує цілісні та стійкі послуги;

2) Service Design — надає інструкції щодо розробки та проектування послуг і процесів. Представлено принципи та методи проектування для перетворення стратегічних цілей на портфель послуг та активів послуг;

3) Service Transition — містить інформацію про розвиток і вдосконалення можливостей щодо впровадження нових або змінених послуг у виробництво;

4) Service Operation — зосереджується на роботі IT-служб щодо ефективності та результативності;

5) Continual Service Improvement — містить інструкції щодо постійного вдосконалення дизайну, реалізації та експлуатації IT-сервісів (процес постійного вдосконалення).

ISO/IEC 20000 є міжнародним стандартом для управління послугами, що містить вимоги до системи управління послугами, тоді як ITIL надає сукупність знань для досягнення цих вимог.

### 1.2.2.3. COBIT

Цілі контролю за інформаційними та пов’язаними технологіями (COBIT) є системою контролю, яка допомагає організації забезпечити узгодженість між використанням інформаційних технологій та її бізнес-цілями. COBIT базується на п’яти ключових принципах [23]: задоволення потреб зацікавлених сторін; наскрізне покриття підприємства; застосування єдиної інтегрованої основи; забезпечення цілісного підходу; відокремлення управління від управління.

COBIT також містить еталонну модель процесу, загальні атрибути можливостей процесу та модель оцінки процесу, яка описує, як виконати оцінку

можливостей ефективним і ефективним способом. COBIT буде проаналізовано з метою використання або адаптації еталонної моделі процесу для використання з основними процесами СУІБ. Крім того, надається професійний посібник COBIT 5 з інформаційної безпеки, який зосереджується на інформаційній безпеці та містить більш детальні та більш практичні вказівки. Доступні відображення та інтеграція між серіями COBIT, ITIL та ISO/IEC27000. Сімейство COBIT використовується для ідентифікації основних процесів СУІБ та для інтеграції рівнів зрілості в структуру основних процесів СУІБ.

### 1.2.3. Огляд сімейства стандартів ISO з інформаційної безпеки

Появу стандарту ISO 27001 можна віднести до 1993 року (рис. 1.3), коли британська професійна асоціація Національний обчислювальний центр (NCC) опублікувала документ під назвою «PD 0003 Кодекс практики управління інформаційною безпекою». Британський інститут стандартів (BSI) прийняв це і видав «BS 7799-1 IT — Методи безпеки — Кодекс практики управління інформаційною безпекою» як національний стандарт у 1995 році. Додаткова частина «BS 7799-2 Системи управління інформаційною безпекою — Специфікація з керівництвом до використання» дозволяє компаніям сертифікувати свої процеси.

ISO гармонізувала цей стандарт з іншими, такими як ISO 9001, і розробила ISO 27001 у жовтні 2005 року. Відтоді компанії можуть сертифікувати свої процеси відповідно до цього міжнародного стандарту [1, 21]. ISO 27001 став основою для сімейства стандартів ISO 27 K, які охоплюють різні стандарти інформаційної безпеки. У 2007 році старий стандарт ISO 17799 було віднесено до сімейства ISO 27 K як ISO 27002. У 2009 році ISO 27000 був виданий для надання огляду, вступу та пояснення термінології під назвою «IT — Методи безпеки — Системи управління інформаційною безпекою — Огляд і словник».

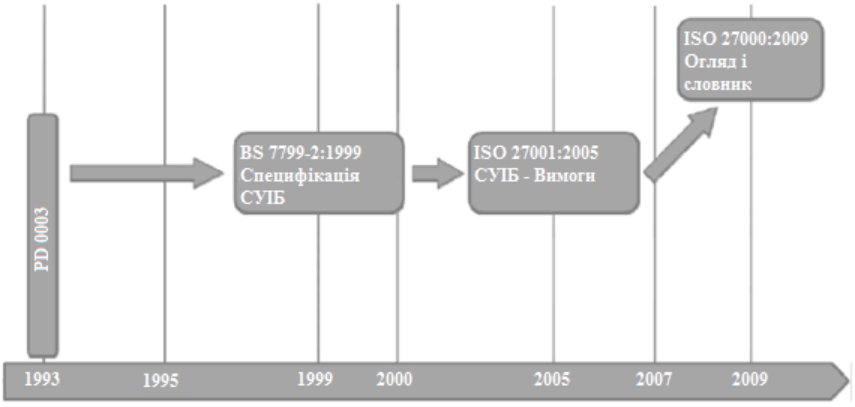


Рис. 1.3. Розвиток стандарту ISO/IEC 27001

Міжнародна організація стандартизації (далі — ISO) та Міжнародна електротехнічна комісія (далі — IEC) створили спільний технічний комітет — ISO/IEC JTC 1. Підкомітет SC 27 цього комітету має робочу групу. РГ 1, яка розробляє та підтримує міжнародні стандарти для СУІБ. ISO 27001 як міжнародний стандарт від ISO/IEC JTC 1 SC27 WG1 для систем управління інформаційною безпекою є стандартом безпеки на підприємствах. ISO 27001 містить вимоги до планування, впровадження, експлуатації та вдосконалення СУІБ. Вимоги сформульовані в загальному вигляді, щоб відповідати всім організаціям, незалежно від їх розміру, цілей, розташування бізнес-моделі тощо. У ISO 27001 не сформульовано абсолютно жодних вимог до будь-якої конкретної технології, але стандарт містить вимоги до основного процесу СУІБ. Таким чином, цей стандарт формує основу для визначення основних процесів СУІБ.

Серія ISO 27000 містить не лише ISO 27001. Іншим загальним стандартом для інформаційної безпеки серії ISO 27000 є ISO 27002, що містить засоби контролю, які повинні бути реалізовані в СУІБ [1, 21]. ISO 27002 пов’язаний із ISO 27001 із Додатком до ISO 27001, у якому перераховано засоби керування ISO 27002. Подальші стандарти серії ISO 27000:

1) ISO 27000 — СУІБ — огляд і словник;

2) ISO 27003 — Керівництво з впровадження СУІБ;

3) ISO 27004 — Управління інформаційною безпекою — Вимірювання;

4) ISO 27005 — Управління ризиками інформаційної безпеки;

5) ISO 27006 — Вимоги до органів, що здійснюють аудит та сертифікацію СУІБ;

6) ISO 27007 — Рекомендації щодо аудиту СУІБ;

7) ISO 27008 — Керівництво для аудиторів щодо контролю СУІБ;

8) ISO 27010 і наступні — секторальні стандарти;

9) ISO 27030 і наступні — стандарти для технічного контролю та вказівки щодо контролю за ISO 27002.

### 1.2.4. ISO/IEC 27001

ISO/IEC 27001 — це міжнародний стандарт інформаційної безпеки та створення СУІБ. Спільно опублікований Міжнародною організацією зі стандартизації та Міжнародною електротехнічною комісією, стандарт не вимагає конкретних дій, але містить пропозиції щодо документації, внутрішнього аудиту, постійного вдосконалення, а також коригувальних і запобіжних дій. Але він визначає вимоги до створення, впровадження, експлуатації, моніторингу, перегляду, підтримки та вдосконалення документованої СУІБ в організації [1].

Стандарт ISO 27001 був опублікований у 2005 році під назвою «Інформаційні технології — Методи безпеки — Системи управління інформаційною безпекою — Вимоги». На 42 сторінках описуються вимоги, яким має відповідати СУІБ для отримання сертифікації. Як основа, стандарт спрямований на компанії з усіх секторів і будь-якого розміру. Однак є деякі сумніви щодо придатності для СУІБ. Конкретні заходи щодо виконання вимог стандартом не обумовлюються, а мають бути розроблені та впроваджені на основі конкретної компанії. Сертифікаційні вимоги ISO 27001 пояснюються через розробку термінів і концепцій і доповнюються інструкціями щодо впровадження в рамках ISO 27002. Центральним пунктом ISO 27001 є вимога щодо планування, впровадження, експлуатації та постійного моніторингу та вдосконалення СУІБ, орієнтованої на процеси. . Підхід має узгоджуватися з циклом «Плануй-Виконуй-Перевіряй-Дій» (рис. 1.4). Охоплення та обсяг СУІБ повинні бути визначені для планування та впровадження.

Слід ідентифікувати та оцінити ризики, а також визначити цілі контролю для інформації та інформаційних систем.

З цього слід вивести відповідні заходи для захисту операцій. У додатку А стандарту загалом перераховано 39 цілей контролю та 134 заходи для управління безпекою, які чітко обумовлені. Цілі контролю перераховані в таблиці 2, розділені за доменами. Вони описані далі та детально в стандарті ISO 27002. Необхідно розробити відповідну підготовку для впровадження, щоб проштовхнути передбачені процедури та встановити їх, а також сформувати усвідомлення їх необхідності. Необхідно постійно контролювати дотримання процедур. Заходи повинні бути перевірені та вдосконалені в ході безперервного вдосконалення, а ризики безпеки повинні бути виявлені та оцінені з метою постійного підвищення ефективності та ефективності СУІБ. Вимоги, які мають застосовуватися до документації СУІБ, описані в стандарті шляхом визначення основного змісту, необхідних документів, а також специфікацій і структур моніторингу для управління документами, таких як: процеси змін і затвердження; контроль версій; правила для прав доступу та захисту доступу; специфікації систем файлів.

Вони охоплюють визначення та реалізацію політики безпеки, визначення ролей та обов’язків, набір та підготовку необхідного персоналу та матеріальних ресурсів, а також рішення щодо управління ризиками. Удосконалення та подальший розвиток СУІБ має здійснюватися безперервно на основі політики безпеки, реєстрації та оцінки операцій, результатів тестування, а також результатів заходів із покращення. Крім того, удосконалення та подальший розвиток слід просувати шляхом регулярних внутрішніх аудитів. Адекватне впровадження політики безпеки, а також її придатність і повнота мають бути забезпечені шляхом щорічного перегляду керівництвом. Щоб отримати сертифікат ISO 27001, організації потрібна СУІБ, яка ідентифікує активи організації та надає оцінку: ризиків, з якими стикаються інформаційні активи; заходів, вжиті для захисту інформаційних активів; планів дій на випадок порушення безпеки; і визначень осіб, відповідальних за кожен крок процесу забезпечення інформаційної безпеки [1, 12].

Стандарт ISO/IEC 27001 призначений для забезпечення вибору адекватних і пропорційних заходів безпеки для захисту інформаційних активів. Цей стандарт зазвичай застосовується до всіх типів організацій, як приватних, так і державних. Стандарт вводить циклічну модель, відому як модель «Плануй-Виконуй-Перевіряй-Дій», яка спрямована на встановлення, впровадження, моніторинг і підвищення ефективності СУІБ організації (див. рис. 1.4).

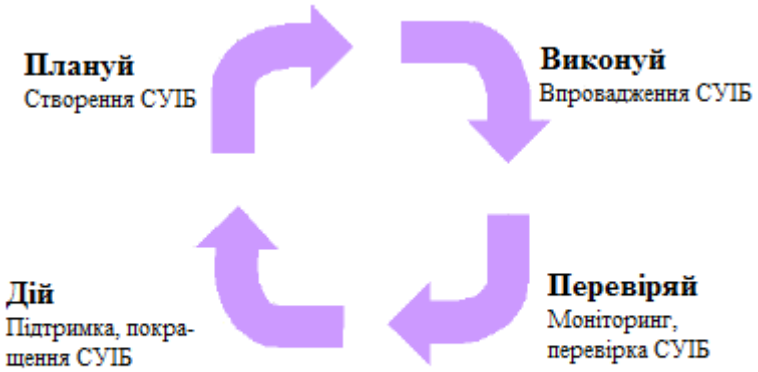


Рис. 1.4. Циклічна модель стандарту ISO/IEC 27001.

ISO27001 допомагає організаціям керувати безпекою активів. Проте ISO27001 є найвідомішим стандартом, який містить вимоги до системи управління інформаційною безпекою [3]: 1) підвищення ефективності бізнесу; 2) зменшення операційного ризику; 3) забезпечення раціонального застосування інформаційної безпеки; 4) гарантії діловим партнерам і клієнтам через сертифікацію який використовувався як маркетингова ініціатива; 5) поінформованісті про безпеку серед співробітників і менеджерів.

Стандарт ISO 27001 разом із стандартами ISO 27002 пропонує найкращі практичні вказівки щодо встановлення СУІБ. Нижче наведено перелік найкращих практик, які слід розглянути перед тим, як інвестувати в СУІБ:

1) *Розуміння потреб бізнесу*. Перед виконанням СУІБ організаціям важливо оглянути бізнес-операції, інструменти та системи управління, щоб зрозуміти вимоги

до бізнесу та безпеки. Це також допомагає вивчити, як структура ISO 27001 може допомогти із захистом даних і особами, які відповідатимуть за виконання СУІБ.

2) *Створення політики інформаційної безпеки.* Наявність політики інформаційної безпеки перед встановленням СУІБ корисно, оскільки це може допомогти організації виявити слабкі сторони політики. Політика безпеки зазвичай повинна надавати загальний огляд поточних засобів контролю безпеки в організації.

3) *Контроль доступу до даних.* Компанії повинні контролювати свою політику контролю доступу, щоб гарантувати, що лише авторизовані особи отримують доступ до конфіденційної інформації. Цей моніторинг має спостерігати, хто, коли і звідки отримує доступ до даних. Окрім моніторингу доступу до даних, компанії також повинні відстежувати входи та автентифікацію та зберігати їх для подальшого дослідження.

4) *Навчання з питань безпеки*. Усі співробітники повинні регулярно проходити навчання з питань безпеки. Навчання має познайомити користувачів із загрозами, що розвиваються, поширеними вразливими місцями в інформаційних системах, а також методами пом’якшення та запобігання, щоб захистити дані від злому.

5) *Захист усіх організаційних пристроїв* від фізичного пошкодження та втручання, вживаючи заходів безпеки, щоб запобігти спробам злому. Інструменти, зокрема Google Workspace і Office 365, мають бути встановлені на всіх пристроях, оскільки вони забезпечують вбудований захист пристрою.

6) *Шифрування даних* запобігає несанкціонованому доступу та є найкращим способом захисту від загроз безпеці. Усі організаційні дані слід зашифрувати перед налаштуванням СУІБ, оскільки це запобігатиме будь-яким несанкціонованим спробам саботувати важливі дані.

7) *Резервне копіювання даних* відіграє ключову роль у запобіганні втраті даних і має бути частиною політики безпеки компанії перед встановленням СУІБ. Окрім регулярних резервних копій, слід спланувати місце та частоту резервних копій. Організації також повинні розробити план безпеки резервних копій, який має застосовуватися як до локальних, так і до хмарних резервних копій.

8) *Внутрішній аудит безпеки*. Перед виконанням СУІБ слід провести внутрішній аудит безпеки. Внутрішні аудити є чудовим способом для організацій отримати видимість своїх систем безпеки, програмного забезпечення та пристроїв, оскільки вони можуть виявити та виправити лазівки в безпеці перед виконанням СУІБ.

### 1.2.5. Порядок отримання сертифікації ISO/IEC 27001

Щоб перевірити відповідність СУІБ стандарту ISO 27001, компанія має пройти процедуру сертифікації, керовану уповноваженою організацією сертифікації (зареєстровані органи сертифікації ЗОС), ISO надає список ЗОС. Компанія починає процедуру шляхом вибору ЗОС [20]. Під час попередньої експертизи за підтримки ЗОС можна визначити, наскільки вже існує відповідність стандарту та які дії ще існують для успішної сертифікації. Відповідно, заходи, необхідні для відповідності СУІБ, повинні бути здійснені в підготовчому проекті. Для цього необхідні відповідні знання та досвід із процесами сертифікації, а також спеціальний досвід у сфері інформаційної безпеки, який слід отримати, залучивши зовнішніх експертів, якщо це необхідно. У першу чергу експертиза для сертифікації (аудит) включає перевірку всіх документів (політика безпеки, описи процесів тощо) ЗОС, для цього документи повинні бути надіслані до сертифікуючої організації [20].

Перевірка документації служить підготовкою до основного аудиту, де представники сертифікаційної організації проводять детальний огляд під час виїзного візиту, який триває кілька днів. Це включатиме проведення інтерв'ю з усіма відповідальними особами, під час яких вони пояснюватимуть своє розуміння політики безпеки, описуватимуть процеси, представлятимуть деталі та особливості на випадковій основі, пояснюватимуть документацію процесу, а також обговорюватимуть відомі недоліки та започатковані заходи щодо покращення. Тоді сертифікаційна організація створить звіт, у якому пояснюються результати аудиту та перераховуються заходи щодо покращення, які необхідно впровадити обов'язково до наступного аудиту [13]. У разі позитивного загального результату компанія отримує офіційний сертифікат на підтвердження відповідності СУІБ

вимогам ISO 27001. Впровадження відповідної СУІБ може зайняти від кількох місяців до кількох років, значною мірою залежно від зрілості управління IT-безпекою в межах організації.

Коли вже встановлені процеси відповідно до таких структур, як COBIT, ISO 20000 або ITIL, час і витрати на впровадження будуть меншими. Процес сертифікації займе ще кілька місяців. Сертифікат діє 3 роки; після цього можна застосувати повторну сертифікацію, яка, як правило, вимагає менше зусиль, ніж початкова сертифікація. Постійне дотримання вимог стандарту ISO 27001 та постійне вдосконалення СУІБ забезпечується щорічними моніторинговими аудитами [15]. Ці аудити проводяться аудиторами з RCB, причому перший моніторинговий аудит має відбутися до того, як мине 12 місяців після видачі сертифікату. Якщо під час моніторингового аудиту будуть виявлені серйозні відхилення від вимог стандарту, RCB може призупинити або навіть відкликати сертифікат до усунення відхилень. Деякі національні альтернативи існують. Для німецьких компаній Федеральне відомство з інформаційної безпеки (BSI) пропонує з 1994 року процедурні рекомендації — так звані «IT-Grundschutz» — для підтримки органів влади та компаній щодо безпеки. У 2006 році ці специфікації були переглянуті на основі ISO 27001, і відповідність між «IT-Grundschutz» BSI та стандартом ISO 27001 була офіційно підтверджена. З 2006 року BSI присвоює цю «сертифікацію ISO 27001 на основі IT-Grundschutz», за допомогою якої сертифікується як відповідність ISO 27001, так і оцінка заходів безпеки IT за каталогами IT-Grundschutz.

### 1.2.6. Сучасне розповсюдження ISO/IEC 27001

На кінець 2010 року в усьому світі діють 15 625 сертифікатів відповідно до ISO 27001, більш свіжої та надійної інформації не існує. На рис. 1.5 показано розвиток з 2006 по 2010 роки та значне зростання розповсюдження. Зважаючи на велику кількість сертифікатів у 2006 році, слід зазначити, що організації, які мали сертифікати відповідно до попередніх стандартів, змогли конвертувати їх у ISO 27001 у спрощеному процесі [13]. Усі наведені цифри показують кількість

сертифікатів відповідно до ISO 27001, а не кількість сертифікованих організацій. Неможливо вказати кількість організацій, які мають сертифікати, оскільки деякі організації мають декілька сертифікатів, напр. для кількох сайтів або груп інші організації мають один сертифікат для кількох сайтів. Тільки в Японії було зареєстровано 6264 сертифікати через місцеве національне законодавство Японії, яке часто вимагає подання доказів або перевірки відповідності управління безпекою стандартам. Крім того, напрочуд високу кількість сертифікатів в Азії, окрім Японії, можна частково пояснити наступним чином: однією з цілей компаній у Європі та Північній Америці є зниження витрат за рахунок аутсорсингу IT-послуг [13].

IT-провайдери в Азії прагнуть досягти цієї мети насамперед за рахунок використання нижчих витрат на персонал. Однак ці провайдери здебільшого невідомі в Європі та Північній Америці та не мають ані іміджу, ані репутації. Менеджерам, які збираються передати частину своєї IT-діяльності на аутсорсинг, потрібна впевненість у надійності та професіоналізмі азійських IT-провайдерів. Зазвичай вони намагаються забезпечити це детальними та дорогими контрактами та угодами, перевірками, оцінками та оглядами. Незалежні атестації постачальників можуть бути підтримкою та підкріпленням [15]. Завдяки сертифікату відповідно до ISO 27001 IT-провайдери можуть підтвердити відповідність своїх процесів безпеки визнаному стандарту. Сертифікат служить перевіркою незалежного органу та забезпечує впевненість щодо відповідних заходів безпеки; він служить знаком якості, що підвищує конкурентоспроможність IT-провайдера.

Невелика кількість 329 сертифікатів, зареєстрованих у Північній Америці, підтверджує загальне припущення, що міжнародні IT-стандарти зараз не привертають там особливої уваги. У Європі ISO 27001 отримав широке розповсюдження. Велику кількість сертифікатів у Великобританії також можна пояснити тим, що британський стандарт був основою для міжнародного ISO 27001 стандарту, тому існує довша традиція сертифікації відповідно до стандартів безпеки [20].

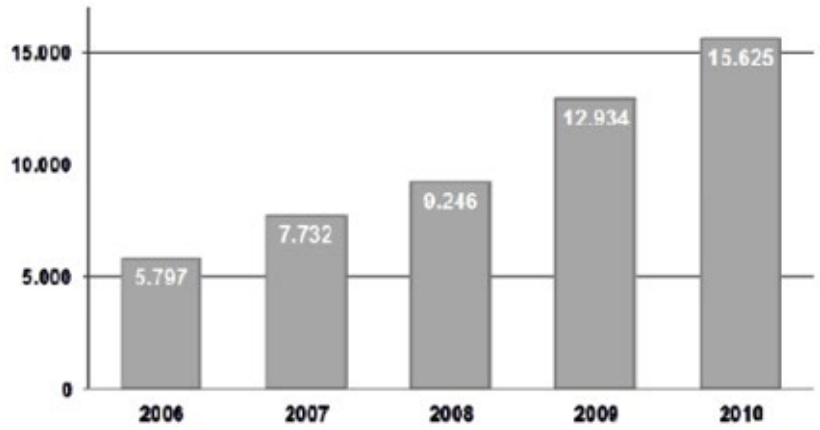


Рис. 1.5. Кількість сертифікатів ISO 27001

## 1.3. Шаблон MVC

### 1.3.1. Загальна інформація

Програмне забезпечення, звичайно, повинно з чимось взаємодіяти, щоб бути корисним. Іноді воно взаємодіє з іншими машинами; дуже часто це з людьми. І тому, звичайно, є інтерфейси. Дійсно, більше зусиль витрачається на інтерфейс, ніж на решту програми.

Розумно припустити, що будь-яка програма, імовірно, змінить свій інтерфейс з часом або справді матиме кілька інтерфейсів у будь-який момент часу. Проте основна програма цілком може бути досить постійною. Банківська програма, яка раніше працювала за символьними системами меню або інтерфейсами командного рядка, ймовірно, буде точно такою ж програмою, яка сьогодні, ймовірно, сидить за графічним інтерфейсом користувача (GUI). Вбудовування будь-якого конкретного інтерфейсу в програму зашкодить як самій програмі, роблячи її менш гнучкою та складнішою для міграції; і інтерфейс, що ускладнює використання для інших програм [25]. Тоді має сенс відокремлювати суть програми від будь-яких і всіх її інтерфейсів.

Контролер представлення моделі (MVC) — це архітектурний шаблон, який зазвичай використовується в веб-застосунках. Він забезпечує три основні шари; модель, подання та контролер. Багато розробників використовують MVC як

стандартний шаблон проектування. Це повна структура. MVC надає три типи класів [25]:

1) Класи моделі використовуються для реалізації логіки доменів даних. Ці класи використовуються для отримання, вставки або оновлення даних у базу даних, пов’язану з застосунком.

2) Перегляд — представлення використовуються для підготовки інтерфейсу застосунка. За допомогою цього інтерфейсу користувачі взаємодіють із застосунком.

3) Класи контролерів використовуються для відповіді на запити користувача. Класи контролерів виконують дії, які вимагає користувач. Ці класи працюють з класами моделі та вибирають відповідне подання, яке має відображатися користувачеві відповідно до запитів користувача.

Архітектура шаблону MVC в основному є трирівневою архітектурою. Він розділяє характеристики застосування. Його перший рівень пов'язаний з логікою введення користувача, другий рівень пов'язаний з бізнес-логікою, а третій рівень використовується для реалізації логіки інтерфейсу користувача. MVC забезпечує дуже слабкий зв'язок між ними. Шаблон MVC використовується для визначення розташування кожної логіки в застосунку.

Шаблони MVC забезпечують можливість паралельної розробки. Це означає, що кожен рівень програми незалежний один від одного, тобто три розробники можуть працювати над однією програмою одночасно. Один розробник працюватиме над логікою введення користувача (логікою контролера), інший розробник працюватиме над логікою інтерфейсу користувача (переглядом), а третій розробник одночасно працюватиме над бізнес-логікою (моделлю).

### 1.3.2. Застосування шаблону MVC

Архітектура шаблону MVC дає нам ідею поділу інтересів, вона допомагає нам реалізувати поділ інтересів між класами моделі, перегляду та контролера в програмах. Відокремлення інтересів полегшує тестування застосунка, оскільки зв’язок між різними компонентами програми є чіткішим і узгодженішим [26]. MVC допомагає нам реалізувати підхід розробки, керований тестуванням, у якому ми

впроваджуємо автоматизовані тестові випадки перед написанням коду. Ці модульні тести допомагають нам заздалегідь визначити та перевірити вимоги нового коду перед його написанням.

За розробки застосунка, яка має дуже високу продуктивність на стороні сервера та невелику комунікацію на стороні клієнта, тоді ми потрібно використовувати архітектуру шаблону MVC, натомість необхідно використовувати прості налаштування, такі як веб-модель форми. Нижче наведено деякі характеристики, які допоможуть нам використовувати архітектуру MVC у застосунку чи ні [27]: 1) застосунок потребує асинхронного зв'язку на сервері; 2) він має функцію, яка не дозволяє перезавантажувати повну сторінку, наприклад, коментувати дописи або нескінченне прокручування тощо; 3) маніпуляції з даними здебільшого відбуваються на стороні клієнта (браузера), а не на стороні сервера; 4) дані одного типу надаються різними способами на одній сторінці (навігація); 5) коли застосунок має багато незначних з'єднань, які використовуються для зміни даних (кнопки, перемикачі).

### 1.3.3. Архітектура MVC

Реалізація контролера model-view-controller у трьох різних динамічних частинах має багато прямих переваг. Наприклад, розробникам або членам команди легко делегувати роботу з розробки та розподілити загальні зусилля, а модель MVC гарантує, що зміни в одній веб-програмі не вплинуть на іншу веб-програму. Наприклад, веб-дизайнер і веб-розробник можуть працювати незалежно, або програміст, який працює над бізнес-логікою, може працювати незалежно від професійного керування потоком програм [26].

Завдяки реалізації архітектури MVC стає легше створювати прототип роботи, дотримуючись дуже простих кроків, таких як створення прототипу веб-застосунку, який отримує доступ до кількох програм на основі робочих станцій [27]. Вносячи невеликі зміни до конфігураційних файлів або перейменовуючи лише вміст сервера, можна реалізувати програми продуктивного рівня, які можуть працювати на різних платформах без необхідності переписувати вихідний код.

Міграція застарілих програм стала легшою, оскільки модель і контролер повністю розділені в MVC, що значно спрощує адаптацію категорії користувачів або платформи [27]. MVC також має великий внесок у спрощення проблеми масштабованості програмного забезпечення великого розміру застосунків і полегшення модифікації та обслуговування застосунків завдяки чіткому розподілу завдань.

Архітектура MVC має наступні переваги [26]: 1) архітектура MVC допомагає нам контролювати складність програми, розділяючи її на три компоненти, тобто модель, представлення та контролер; 2) MVC не використовує серверні форми, тому він ідеальний для тих розробників, які хочуть повний контроль над поведінкою своєї програми; 3) підхід розробки, орієнтований на тестування, підтримується архітектурою MVC; 4) MVC використовує передній шаблон контролера. Шаблон переднього контролера обробляє кілька вхідних запитів за допомогою єдиного інтерфейсу (контролера). Передній контролер забезпечує централізоване управління. Нам потрібно налаштувати лише один контролер на веб-сервері замість багатьох; 5) головний контролер забезпечує підтримку розширеної маршрутизації для розробки веб-застосунку.

### 1.3.4. Особливості фреймворків MVC

Оскільки розділяється логіка застосунку на три завдання (логіка введення, бізнес-логіка, логіка інтерфейсу), тестування цих компонентів стане дуже легким. Тестування є дуже швидким і гнучким, оскільки ми можемо використовувати будь-який фреймворк модульного тестування, сумісний із фреймворком MVC. Це фреймворк, який розширюється та підключається. Ми можемо розробити компоненти застосунку таким чином, щоб їх можна було легко замінити або змінити. Ми можемо підключити власний механізм перегляду, стратегію маршрутизації URL-адрес, серіалізацію обмежень методу дії [26]. Замість того, щоб створювати об’єкти залежно від класу, ми використовуємо ін’єкцію залежностей техніки (DI), яка дозволяє нам вводити об’єкт у класи. Інша техніка інверсії

керування (IOC) використовується, щоб показати залежність між об’єктами, вона визначає, який об’єкт потребує іншого об’єкта.

MVC надає компоненту відображення URL-адрес, який допомагає нам створювати за допомогою зрозумілих і доступних для пошуку URL-адрес. Замість використання розширень імен файлів MVC підтримує шаблони іменування URL-адрес, які дуже корисні для оптимізації пошукових систем (SEO) і адресації передачі репрезентативного стану (REST). Деякі фреймворки MVC, такі як фреймворк ASP.NET MVC, надають нам деякі вбудовані функції, такі як автентифікація форми, керування сеансами, транзакційна бізнес-логіка, безпека веб-застосунків, об’єктно-реляційне відображення, локалізація, членство та ролі та авторизація URL-адрес тощо. Найпопулярніші сьогодні доступні фреймворки backbone.js, ember.js; angular.js і knockout.js [27]. Кожен фреймворк має свої переваги та недоліки. Розробники можуть використовувати будь-які фреймворки відповідно до його вимог, які відповідають їхнім веб-застосункам.

### 1.3.5. Інструменти та технології, що використовуються з MVC

Існує багато інструментів і технологій, які можна використовувати для розробки веб-застосунків за допомогою архітектури MVC. Залежно від зацікавленості розробників, вони можуть використовувати будь-які інструменти та технології для розробки веб-застосунків. Нижче перелічені деякі інструменти та технології, які можна використовувати для розробки веб-застосунків з використанням архітектури MVC [26].

Visual Studio — це не просто інструмент, а повне середовище розробки, яке надає нам можливість створювати різні типи програм. Коли ми хочемо розробити програму за допомогою фреймворку ASP.NET MVC, тоді Visual Studio дуже допоможе. Отже, необхідні інструменти [27] — це 1) MYSQL Server — реляційний сервер управління базами даних для підтримки бази даних; 2) SQL Server — механізм бази даних для підтримки бази даних так само, як сервер MYSQL; 3) MYSQL Workbench — інструмент проектування бази даних; 4) Net Beans — IDE

(інтегроване середовище розробки) забезпечує повне середовище для розробки різних програм; 5) Сервер Glassfish: сервер застосунків Java EE.

Наступні технології використовуються з MVC [27]: 1) HTML, CSS, JQUERY, AJAX для проектування; 2) Сервлети та сторінки сервера Java (JSP), що використовуються з Net beans; 3) Технології EJB (Enterprise Java beans); 4) JSTL (стандартні бібліотеки тегів сторінок сервера Java); 5) JPA; 6) JDBC (підключення до бази даних Java); 7) ASP.NET MVC використовується з Visual Studio. Є багато інших інструментів і технологій, які можна використовувати з архітектурою MVC, але ми перерахували деякі з тих інструментів і технологій, які ми збираємося використовувати для створення веб-застосунку з використанням архітектури MVC.

# РОЗДІЛ 2. АРХІТЕКТУРА ЕЛЕКТРОННОГО СХОВИЩА ФАЙЛІВ ПІДПРИЄМСТВА

## 2.1. Огляд електронних сховищ файлів

Підприємства є ключовим елементом більшості економічних систем. Протягом усього часу бізнес часто недооцінювався на основі зроблених припущень, таких як наявність цінної інформації для всіх зацікавлених сторін (клієнтів, постачальників, співробітників, ділових партнерів тощо). На часі все більше погоджується, що інформація є найціннішим активом будь-якого підприємства. У поточній економічній кон'юнктурі, яка характеризується зростаючою невизначеністю, володіння цінною інформацією відіграє все більш важливу роль у процесах прийняття рішень, що здійснюються різними суб'єктами бізнесу [28].

Поняття підприємництва пов’язане з різними аспектами, пов’язаними з управлінням бізнесом, зокрема щодо аспекту створення нових та інноваційних підприємств. Оцінка підприємницької діяльності та рівня її успішності в досягненні поставлених цілей вимагає врахування багатьох аспектів впливу. Одним із них, безсумнівно, є те, як підприємець керує (збирає, організовує, зберігає, оновлює та поширює) бізнес-інформацію.

Використання оновленої інформації для підтримки прийняття рішень є однією з головних проблем кількох бізнес-процесів [28]. Обмін інформацією в промисловості здійснюється за допомогою багатьох видів технологічних баз, але найнадійнішою формою підтвердження бізнес-операції, внутрішньої чи зовнішньої в офісах, є документи. Документ — це все, що зберігається в доступному джерелі.

Інформаційні технології та мережі змінюють те, як професіонали стикаються з багатьма бізнес-процесами, а використання електронних документів і систем автоматизації офісу змушує нас думати, як оцінити проблеми управління корпоративним контентом. Системи електронного документообігу (СЕД) використовуються в кількох галузях промисловості, таких як банки, виробництво, фармацевтика, страхування, машинобудування та інші. СЕД можна використовувати від стратегічних рівнів до операційних. Управління будівельною документацією є важливою складовою загальної функції управління проектом. Більше того, збільшення обсягу виробництва, публікації та загальнокорпоративного розповсюдження документів через системи електронної пошти та файлові менеджери на робочих станціях загострило проблеми безпеки, контролю, відстеження та пошуку документів [45].

Управління проектуванням приділяється багато уваги через його сильний вплив на всю роботу [29]. Серед факторів, що обговорюються, часто присутній фрагментований потік комунікації та інформації. Погана комунікація, відсутність відповідної документації, недостатня або відсутня вхідна інформація, незбалансований розподіл ресурсів, відсутність координації між дисциплінами та нестійке прийняття рішень були відзначені як основні проблеми в управлінні дизайном. Проблеми, пов'язані з погано скоординованим документообігом, присутні не тільки на ранніх стадіях розробки проекту, але й у всьому проекті будівлі. Зростаюча складність і масштаб проектів призвели до збільшення проблем, пов’язаних із керуванням та пошуком документів вручну [45].

Системи електронного документообігу [33] — це набір технологій, пов’язаних для досягнення мети. Наступні технології, як правило, вбудовані в СЕД: Imaging, що стосується необхідності перетворення паперових документів у цифрові за допомогою сканерів; Full-Text-Retrieval, який отримує документи за допомогою пошуку слів у них; Робочий процес, який дозволяє інновувати продуктивні процеси шляхом їх реінжинірингу, що дозволяє контролювати маршрут документації всередині компанії; і Мультимедіа, остання розроблена технологія СЕД, яка дозволяє зберігати та отримувати кадри анімації, звуки тощо.

СЕД зазвичай базується на двох основних системах: управління документами, яка керує динамічними документами, такими як файли текстового процесора, і відображення документів, яка працює зі статичними документами, такими як зображення документів, отримані за допомогою сканера. Система також може включати бази даних, які допомагають упорядковувати паперові документи. СЕД — це система для зберігання та пошук інформації. Це можуть бути факси, відскановані зображення чи документи, креслення, текстові документи, електронні таблиці, звіти з бази даних, листи, специфікації та, фактично, будь-який документ.

Системи управління документами використовуються для контролю життєвого циклу документа. Системи часто є модульними, їх можна знайти як набір апаратного та програмного забезпечення, яке керує створенням, затвердженням, розповсюдженням та іншими етапами життєвого циклу документа. Електронне керування документами не означає використання лише електронних документів [33].

Документи простіше визначити, починаючи з прикладів, загальні документи можна проілюструвати доповідями, листуванням, повідомленнями електронної пошти, меморандумами, електронними таблицями, дослідженнями тощо. Документ визначається як усе, що було збережено в доступному джерелі. Люди зазвичай асоціюють документ із папером, але його можна знайти в багатьох формах. Виходячи з наведеного вище визначення, документами можуть бути тексти, записані звуки та зображення (відскановані папери чи відеокасети) [31]. Управління документами можна визначити як процес нагляду за офіційними діловими операціями підприємства, записами про прийняття рішень і важливими тимчасовими документами, які представлені у форматі документа [31].

Ця технологія використовується для керування завданнями затвердження та розподілу в життєвому циклі документа. Робочий процес базується на правилах, які передають в електронному вигляді завдання з документом або деякими документами на робочий стіл професіоналів. Зазвичай вони поділяють робочу діяльність на чітко визначені завдання, ролі, правила та процедури, які регулюють більшу частину роботи на виробництві та в офісі. Запровадження робочого процесу

може включати активацію, відстеження, моніторинг стану, обмін повідомленнями, передачу черги та маршрутизацію документів. Стає можливим моніторинг процесу управління, а також відстеження та маршрутизація пов’язаних документів [31].

Управління робочим процесом передбачає: моделювання процесу, що вимагає моделей робочого процесу та методів для фіксації та опису процесу; реінжиніринг процесів, що вимагає прийомів оптимізації процесу; і реалізація та автоматизація робочого процесу, що вимагає методологій і технологій для використання інформаційних систем і людей-виконавців для впровадження, планування, виконання та контролю завдань робочого процесу, як описано в специфікації робочого процесу.

### 2.2. Організація електронного сховища файлів

Оскільки ця робота мала на меті визначити не лише можливі архітектури, а й методології, які підтримують програму впровадження системи сховищ даних, важливо розрізняти ці два поняття. Архітектура ідентифікує частини (компоненти), їх характеристики та встановлює зв’язки між сторонами. Методологія визначається як ідентифікація набору заходів (процесу) та їх послідовності для того, щоб привести до кінцевої мети.

Щодо найпоширеніших архітектур, виділяють п’ять [29]: (1) Незалежні вітрини даних; (2) Data Mart Bus; (3) Hub and Spoke; (4) Централізована та (5) Федеративна. Виходячи з класифікації, ми можемо трохи краще зрозуміти, як вони організовують кожну з архітектур [29]:

• *Незалежні вітрини даних* призначені для роботи незалежно одна від одної. Кожна вітрина даних має своє визначення даних, а розміри та показники між кількома вітринами даних не нормалізуються, що ускладнює аналіз даних між ними. Надмірність даних може бути великою.

• *Data Mart Bus* базується на аналізі вимог до конкретного бізнес-процесу. Перша вітрина даних створена для єдиного бізнес-процесу з використанням нормалізованих розмірів і показників, які потім використовуються з іншими вітринами даних.

• *Hub and Spoke* базується на аналізі вимог для розширюваного корпоративного рівня, а також зосереджує увагу на побудові масштабованої інфраструктури та тривалому обслуговуванні. Він розробляється в ітераційний спосіб, предмет за предметом. Вітрини даних, створені з джерела даних сховища даних, можуть бути розроблені відділом, функціональною сферою або спеціальними цілями (наприклад, інтелектуальний аналіз даних) і можуть мати структури нормалізованих, денормалізованих або зведених/уточнених даних на основі потреб користувача.

• *Централізована* відрізняється від архітектури Hub and Spoke, оскільки не має вітрин даних. Цей централізований підхід надає користувачеві доступ до всіх даних у сховищі даних. Це також дозволяє зменшити обсяг даних для передачі або зміни, таким чином спрощуючи керування та адміністрування ними.

• *Федеративна* базується на підтримці структур підтримки прийняття рішень, і відповідно до бізнес-вимог доступ до даних здійснюється з цих джерел. Дані логічно та фізично інтегруються за допомогою спільних ключів, глобальних метаданих, розподілених запитів та інших методів.

Якою б не була обрана архітектура, компоненти, які завжди включені в реалізацію системи сховища даних [29]: 1) OLTP (Online Transaction Processing) — операційні системи реєстру, які фіксують щоденні транзакції компанії; 2) ETL (Extract, Transfer and Load) — процес, який є першим кроком завдання отримання даних із систем OLTP у середовище Data Warehouse; 3) DSA (Data Staging Area) — програма, яка здійснює зв’язок між системами OLTP та Data Warehouse.

Разом із компонентами, перерахованими вище, застосунки, орієнтовані на підтримку прийняття рішень, також є частинами реалізації системи Data Warehouse. Оскільки відповідна діяльність залежить від управління якістю даних і метаданими, система сховища даних повинна містити інструменти для цієї мети. Коли дані надходять до DSA, відбувається багато змін, як-от фільтрація даних (виправлення помилок друку, розв’язання конфліктів доменів тощо), інтеграція даних із кількох джерел, видалення дублікатів даних і призначення ключа [29].

Загалом, наступні послідовні дії в процесі впровадження системи сховища даних: аналіз вимог бізнесу; дизайн даних (модель даних і нормалізація); проектування архітектури; реалізація; технічне обслуговування. Однак зроблено висновок, що залежно від обраної архітектури дизайн даних буде проходити різними шляхами [29]. Пізніше цей підхід обговорюється та концептуалізується інший, який відповідає дійсності більш відповідним чином. Найбільш використовуваними платформами є ORACLE, MICROSOFT і IBM. Для цих постачальників MICROSOFT і IBM дотримуються архітектури Hub and Spoke, тоді як ORACLE дотримується архітектури Data Mart Bus.

Порівнюючи цих постачальників щодо їх основної компетенції, здається, що в категорії «Технології» основна компетенція лежить на рівні систем керування базами даних (СУБД); оскільки у випадку категорії «Інфраструктура» його компетенція зосереджена на рівні програмного забезпечення для бізнес-аналізу (OLAP, інтелектуальний аналіз даних, прогнозний аналіз тощо); а щодо постачальників у категорії консалтингу, їхні навички також зосереджені на рівні бізнес-аналітики, а в деяких випадках в основному присвячені управлінню ресурсами підприємства (ERP).

Кожну систему необхідно оцінити після впровадження та певного періоду використання [44]. Оцінку можна проводити з різних точок зору, залежно від цілей організації. Для оцінки продуктивності системи сховища даних використовуються такі показники: якість інформації та системи; індивідуальний та організаційний вплив; час і вартість розробки.

По відношенню до біноміальної вартості/часу та з аналізу літератури, здається, що п’ять архітектур розподілені. Під час впровадження СЕД можна безпосередньо побудувати блок-схему процесу, а потім точно визначити та виміряти діяльність, що не додає цінності [44]. Таким чином, перш ніж запровадити робочий процес, необхідно подумати про процес, відобразити потік інформації, що дозволить зменшити діяльність, яка не додає цінності.

Використовуючи систему робочого процесу, система надасть доступ до необхідної інформації на робочому столі працівника. Система надає додаткову

цінність із профілем метаданих, щоб гарантувати, що документ можна знайти знову. Можливості отримання та пошуку СЕД дозволяють працівнику виграти час, який інакше було б втрачено, намагаючись знайти паперові файли. СЕД також усуває дубліковану інформацію та скорочує час повторного створення [42, 43].

У процесі ручного керування деякі дії, які не мають доданої вартості, можуть бути, наприклад: зателефонувати комусь, щоб знайти документ, піти в кімнату з файлами, підготувати або дочекатися передавання факсу та інші. Якщо неможливо усунути діяльність, яка не створює доданої вартості, альтернативою є зробити її більш ефективною. У цьому відношенні СЕД є рішенням для розширення співпраці та кращої інтеграції для прийняття рішень.

Принцип зменшення варіативності може бути реалізований через чітке визначення процесу, включаючи види діяльності, які повинні бути виконані, їх залежність, ролі та відповідальність та інформацію про основний потік. За визначенням робочого процесу цей принцип застосовується в системі документообігу [42]. Час циклу розглядається як додавання (1) часу обробки, (2) часу перевірки, (3) часу очікування та (4) часу переміщення.

Одним із підходів до скорочення часу циклу є зменшення відстані між етапами процесу (час переміщення). Однією великою перевагою використання інформаційних технологій є скорочення відстані. Існує відстань, яку документи проходять під час паперового процесу. За допомогою електронних документів і використання робочого процесу ці відстані скорочуються або навіть усуваються. СЕД усуває географічні кордони в організаціях [41].

Використовуючи робочий процес, коли особа має рішення, що робити на основі документа, цей документ буде доступний на його/її робочому столі (час переміщення, час очікування). Працівнику пред'являється документ лише тоді, коли вимагається його внесення. Це приклад технології push, за допомогою якої інформація автоматично надсилається користувачеві без його активного втручання.

Ще одна перевага впровадження системи робочих процесів полягає в тому, що люди критикують процес, змінюючи порядок дій. Використовуючи електронні

документи, інформаційний потік може бути в паралельному порядку, навіть узгодження [41].

Спрощення тут можна розуміти як (1) зменшення кількості компонентів у продукті або (2) зменшення кількості кроків і зв'язків у інформаційному потоці. У цьому випадку спрощення може бути реалізовано шляхом усунення діяльності, що не додає цінності, із процесу документообігу за допомогою СЕД та інструментів робочого процесу.

Гнучкість виробництва можна згрупувати за чотирма основними типами: гнучкість асортименту (кількість різних вироблених продуктів), гнучкість нового продукту (швидкість впровадження продукту), гнучкість обсягу (можливість варіювати виробництво) і гнучкість часу доставки. У будівельній промисловості ці типи гнучкості створюють зміни в дизайні на різних етапах виробничого процесу (проектування та будівництво).

Одним із практичних підходів до підвищення гнучкості є налаштування процедур на пізнішій стадії процесу, за допомогою яких клієнти можуть подавати запити на зміни дизайну. СЕД дозволяє дизайнерам вносити більше змін в дизайн за рахунок стиснення за часом і підвищення прозорості. Однією з цілей є зробити процес прозорим і доступним для спостереження для полегшення контролю та вдосконалення. Графічний робочий процес дозволяє людям контролювати статус кожної дії. Можна точно визначити, де відбуваються збої в процесі, наприклад, де документ очікує на затвердження або хто починає чи завершує його/її завдання після запланованого часу.

Потоки документів показують, хто насправді використовує та виробляє інформацію, і, природно, хто не виконує жодної діяльності в організації. При необхідності легко перевірити звіти про будь-який документообіг. Кожен з аспектів потоку та перетворення має різний потенціал для покращення, чим вища складність процесу, тим вищий вплив покращення потоку. Процес проектування є складним і недостатньо структурованим, цей принцип спрямований на збалансований потік і покращення перетворення, і його можна досягти під час реалізації робочих процесів [29].

### 2.3. Ключові технології проектування електронного сховища файлів

Архітектура програми керування електронними файлами включає рівень презентації, рівень бізнес-застосунку, рівень підтримки плагінів служби, рівень інфраструктури та стандартний рівень підтримки. Рівень представлення — це бізнес-система, задіяна в застосунку, включаючи систему прийому та відправлення документів, систему документообігу, систему пошуку документів, уніфіковану автентифікацію, єдиний вхід та інші бізнес-системи; рівень бізнес-застосунків — це платформа, що підтримує різні бізнес-застосунки проекту. Програмне забезпечення, включаючи захоплення файлів, схему класифікації, обробку автентифікації, пошук файлів, зберігання даних, керування даними, керування правами, керування журналами, користувачами системи, статистичні звіти, керування інтерфейсом, керування системою тощо [29].

Рівень плагіна служби — це різноманітні інструменти. Програмні продукти — це в основному інструменти підтримки програмного забезпечення для забезпечення безпеки системи для прикладних систем. Включно з електронною офіційною печаткою, автентифікацією, підтвердженням макета, цифровим підписом, інкапсуляцією даних, електронною таблицею, захистом авторських прав, привілейованою роллю, підтримкою інструментів; рівень інфраструктури включає обладнання хмарних обчислень та сервіси IaaS; рівень стандартного носія містить список необхідних галузевих стандартних специфікацій

Найважливішим модулем електронного управління є обробка даних і зберігання електронних файлів. Електронні файли мають характеристики великого обсягу даних і малих файлів (менше 64 МБ). Дані електронного файлу складаються з даних сутності та метаданих. У конструкції сховища даних в основному використовується все хмарне сховище, окремо встановлюються бібліотека електронних файлів і тимчасова бібліотека електронних файлів, відповідно зберігаються стабільні електронні файли та електронні файли, які підлягають обробці [29]. Забезпечуючи безпечний і надійний центр зберігання даних у режимі хмарних обчислень, платформа керування даними виконує уніфікований розподіл

ресурсів для всіх ресурсів зберігання даних, балансування навантаження, розгортання програмного забезпечення та керування безпекою даних у них.

У той же час, з точки зору спільного використання даних, управління політикою здійснюється відповідно до різних прав безпеки, а безпека даних гарантується на рівні мережі. Коли дані зберігаються в терміналі, вони позначаються відповідною системною категорією програми та рівнем безпеки. Під час проходження через хмарну мережу рівень безпеки даних перевіряється мережевим рівнем, і якщо високий рівень безпеки, дані надсилаються на низький рівень безпеки. У зоні операція видалення виконується відповідно до політики безпеки, а обробка сигналізації виконується для забезпечення безпеки даних на рівні циркуляції.

Виходячи з важливості та критичності електронних документів, керування резервним копіюванням електронних файлів є дуже важливим. Сама система хмарних обчислень має потужний механізм резервного копіювання. Для коду служби електронних файлів і метаданих електронних файлів повне резервне копіювання виконується кожного разу, коли служба оновлюється або встановлюється нова служба. Для бібліотек електронних документів щоденні оновлення даних є великими, а повне резервне копіювання планується через місяць або тиждень. На цій основі інкрементні резервні копії створюються через короткий проміжок часу. Це також важлива частина керування носіями, для яких уже створено резервні копії.

У процесі впровадження системи необхідно максимально запобігти втраті даних, пошкодженню, підробці та витоку, підвищити безпеку, надійність і стабільність роботи системи, а також забезпечити безпечну роботу системи. Необхідно відстежувати та керувати системою, своєчасно отримувати інформацію про поточний стан та конфігурацію ресурсів мережевої інформаційної системи, відображати доступність і працездатність ресурсів інформаційної системи та створювати кероване IT-середовище для забезпечення різноманітних систем бізнес-застосунків інформаційна система користувача. Він може працювати надійно, ефективно, безперервно та безпечно.

### 2.4. Огляд Windows Server 2016 як компонента побудови СУІБ

Windows 2016 — одна з найновіших ітерацій сімейства програмного забезпечення Windows. Це надійна серверна операційна система, яка надає різноманітні функції. У цьому розділі буде розглянуто загальний огляд і функції Windows 2016.

Серверні версії Windows з'явилися та зникли з вересня 1994 року, коли було представлено сервер Windows NT 3.5. Серверні операційні системи відрізняються від операційних систем для настільних комп'ютерів тим, що вони обробляють програми клієнт/сервер, як-от повідомлення електронної пошти та безліч інших програм, які зазвичай не можна знайти на персональному комп'ютері. Починаючи з Windows NT 3.5, було створено багато різних ітерацій серверних продуктів Windows, які згодом перетворилися на поточний продукт Windows Server 2016, який базується на ядрі робочого столу Windows 10 [34]. Останні версії сервера Windows, термін експлуатації яких закінчився, це випуски Windows Server 2008 і 2012.

Windows 2016 має широкий набір нових і розширених функцій і можливостей порівняно з попередніми ітераціями серверних продуктів Windows. Windows 2016 постачається з новими можливостями контейнеризації або полегшеної віртуалізації. Платформа віртуалізації є, мабуть, найбільш вдосконаленою функцією Windows 2016 [34]. Вона містить дві ліцензії на Hyper-V, програму гіпервізора Microsoft, і підтримує необмежену кількість розміщених контейнерів, які працюють під керуванням полегшеної версії Windows Server 2016 під назвою Nano Server. Це дозволяє створювати віртуальні сервери та віртуальні машини в установці хост-сервера для роботи програмного забезпечення та служб як Windows, так і не на основі Windows. Крім того, Windows Server 2016 — це платформа корпоративного центру обробки даних, яка забезпечить надійне відновлення та балансування мережевого навантаження, а також швидке розгортання та масштабування програм і веб-сайтів [34].

Інші функції включають процеси розгортання віртуальної машини на основі UEFI. Швидше розгортання віртуальної машини, включаючи автоматичне

розгортання віртуальної машини, зменшує час, потрібний для налаштування цих машин. Групова політика реалізує параметр кешу політики, який дозволяє машинам, які приєдналися до домену Active Directory, зберігати копію налаштувань групової політики на локальній клієнтській машині. Таким чином, ці локальні параметри можуть використовуватися клієнтськими машинами під час запуску, і не потрібно запитувати їх через мережу з контролера Active Directory. Це зменшує перевантаження мережевого трафіку. Windows 2016 також використовує новішу версію Windows Defender як у серверному ядрі, так і в робочому столі, яка встановлена та ввімкнена за замовчуванням. Він також дозволяє використовувати служби розгортання Windows і команди через WDS з PowerShell.

Windows 2016 випускається в трьох випусках для споживачів: Essentials, Standard, та Datacenter. Усі ці версії мають спеціальні застосунки, на які вони більше орієнтовані. На відміну від попередніх версій Windows Server, Windows 2016 не має випуску Foundations. Вартість Essentials і Standard подібна, але набір функцій значно відрізняється тим, що стандарт дозволяє віртуалізацію, а Essential — ні. Крім того, деякі випуски накладають суворіші обмеження на використання, як-от максимальну кількість користувачів, підключень або продуктивність програмного/апаратного забезпечення, щоб забезпечити сегментацію ринку. Datacenter залишається найдорожчою версією операційної системи, але має доступ до всіх функцій і майже без обмежень щодо використання програмного забезпечення [34].

### 2.5. Active Directory

Довідник (Active Directory) — це особливим чином упорядкована сукупність інформації. Організаційні методи роблять сортування інформації легким і швидким, щоб ми могли знайти потрібні дані.

Служби довідників часто порівнюють із телефонною книгою. Телефонна книга — це набір даних, упорядкованих за прізвищем, іменем, номером телефону, містом і країною. Оскільки ця інформація організована особливим чином, ми можемо швидко знайти конкретну людину та отримати її номер телефону. Довідники не є чимось новим у технології телефонної книги, і до тих пір, поки

телефонна книга все ще існує, вона все ще використовує довідники, але з точки зору комп'ютерних мереж, довідники все ще є передовою частиною мережевих технологій.

Microsoft Active Directory (AD) — це служба каталогів, яка використовується для зберігання інформації про мережеві ресурси та забезпечує ієрархічний спосіб для всієї мережі пристроїв і програм. Служба Active Directory була представлена в Windows 2000 Server. AD — це центральна колекція користувачів, груп і комп'ютерів, яка забезпечує єдиний вхід (SSO) для пристроїв і програм, приєднаних до домену AD. Active Directory зберігає інформацію про мережевий ресурс, такий як користувачі, пароль користувача, групи, мережевий принтер і комп'ютер, і робить цю інформацію доступною для користувачів і адміністрації.

Active Directory не є першою службою каталогів, яка з'являється на ринку. Насправді служби каталогів існують уже досить давно. Однак випуск Windows 2000 і Active Directory від Microsoft і поява NDS від Novell підкріплюють ідею, що мережа повинна базуватися на каталозі.

Ще кілька років тому нетворкінг не був таким важливим, як сьогодні. Раніше, звичайно, були великі підприємства з великими мейнфреймами та великою кількістю даних. Але лише після того, як комп'ютери заволоділи обчислювальною технікою, це почало змінюватися, і мережі почали розвиватися із загрозливою швидкістю. У більшості основних мереж сьогодні кожен користувач має комп'ютер, загальнодоступні та приватні дані та різні типи обчислювальних потреб. Поточна мережа не може впоратися з усім цим, оскільки надто багато серверів, надто багато ресурсів, забагато безладу. Насправді пошук потрібної інформації в мережі може зайняти багато часу.

Метою служби довідників є наведення порядку у великих і малих мережах. Служби каталогів забезпечують ефективний підхід до мережі та пошуку ресурсів. За допомогою каталогу користувачі можуть виконувати пошукові запити та швидко та легко знаходити мережеву інформацію. Active Directory — це відповідь Microsoft на сучасні потреби служби каталогів мережі.

Active Directory дозволяє адміністраторам ефективно керувати інформацією в межах великого підприємства з центрального сховища, яке можна поширювати по всьому світу [35]. Щойно інформацію про користувачів і групи, комп'ютери та принтери, програми та служби буде додано до Active Directory, її можна буде зробити доступною для використання в усьому підприємстві будь-якій кількості людей, як ми забажаємо. Інформаційну структуру можна адаптувати до організаційної структури, і користувачі можуть запитувати Active Directory, щоб знайти місцезнаходження принтера або електронну адресу колеги.

Active Directory включає більшість операційних систем Windows Server як набір процесів і служб. Структура активного каталогу (AD) — це ієрархічна структура об'єктів. Об'єкти поділяються на три великі категорії: ресурс (принтер), служба (електронна пошта), користувач (облікові записи або користувач і група). Активний каталог містить інформацію про об'єкти, організовує об'єкти, контролює доступ і встановлює безпеку. Active Directory використовує полегшений протокол доступу до каталогу (LDAP) версії 2 і 3, версію Microsoft Kerberos і DNS. Active Directory дозволяє мережевим адміністраторам створювати та керувати доменами, користувачами та об'єктами в мережі. Наприклад, адміністратор може створити групу користувачів і надати їм певні права доступу до певних каталогів на сервері [35]. До Windows 2000 модель автентифікації та авторизації Microsoft вимагала розбиття мережі на домени, а потім зв'язування цих доменів за допомогою складної, а іноді й непередбачуваної системи одно- та двосторонніх довіри. Служба Active Directory була представлена в Windows 2000 як спосіб надання служб каталогів для великих і складних середовищ.

Дані, що зберігаються в Active Directory, представлені користувачеві в ієрархічній формі, подібній до того, як дані зберігаються у файловій системі. Кожен запис називається об'єктом. На структурному рівні розрізняють два типи об'єктів: контейнери та неконтейнери. Неконтейнери також відомі як листові вузли. Один або кілька контейнерів розгалужуються в ієрархічному порядку кореневого контейнера. Кожен контейнер може містити листкові вузли або інші контейнери. Як випливає з назви, листові вузли можуть взагалі не містити об'єктів.

Хоча дані в Active Directory представлені ієрархічно, насправді вони зберігаються в базі даних із рядками та стовпцями. Отже, Active Directory — це ієрархічна структура, яка зберігає інформацію про об'єкти в мережі. Active Directory має такі основні служби, як служба домену, служба сертифікатів, спрощений каталог, служби об'єднання, керування правами.

***1. Служба домену*** Active Directory зберігає централізовані дані та керує зв'язком між користувачем і доменом. Служби домену також включають автентифікацію під час входу та функцію пошуку. Доменні служби активного каталогу також використовуються як роль серверів, які дозволяють адміністратору керувати та зберігати інформацію про мережу ресурсів, а також допомагають керувати елементами мережі та записувати їх в ієрархію.

***2. Сертифікаційні служби*** активного каталогу використовуються для керування, генерації та спільного використання сертифікатів. Сервіс Microsoft Active Directory Certificate використовується для надання платформи для керування сертифікатом інфраструктури відкритого ключа. За словами самої Microsoft, AD CS — це «роль сервера, яка дозволяє створювати інфраструктуру відкритих ключів (PKI) і надавати криптографію з відкритими ключами, цифрові сертифікати та можливості цифрового підпису для вашої організації».

***3. Служба об'єднання каталогів*** Active Directory надають користувачу єдиний вхід для автентифікації користувача в кількох веб-застосунках за один сеанс. Служби федерації працюють на основі федеративної ідентифікації. Крім того, це функція ОС віконного сервера, яка розширює доступ користувача до програми та системи за межами корпоративного брандмауера за допомогою єдиного входу.

***4. Управління правами*** захищає захищену авторським правом інформацію, запобігаючи несанкціонованому використанню та розповсюдженню цифрового вмісту. Керування правами використовується як інструмент безпеки, який забезпечує постійний захист даних шляхом застосування політики доступу до даних [38]. Він використовує шифрування та форму вибіркової заборони функціональності для обмеження доступу до таких документів, як корпоративна електронна пошта,

документи Microsoft Word і веб-сторінок, а також операцій, які авторизовані користувачі можуть виконувати з ними.

Логічні структури Microsoft Active Directory включають основні: 1. Об'єкт 2. Ліс 3. Дерево 4. Домен 5. Розділ. Каркас, який утримує об'єкт, розглядається як декілька рівнів. На верхній структурі знаходиться ліс. Ліс — це сукупність усіх об'єктів, їх атрибутів і правил (синтаксису атрибутів) у Active Directory. Ліс містить одне або кілька транзитивних довірчих пов'язаних дерев.

Дерево містить один або кілька доменів і дерево доменів, знову пов'язаних у транзитивній ієрархії довіри. Домени ідентифікуються за структурою імен DNS, простором імен. Домен з одним ім'ям DNS. Об'єкт, що міститься в домені, можна згрупувати в організаційні одиниці. Організаційні одиниці надають домену ієрархію, полегшують його адміністрування та можуть створити напівбаланс структури компанії Active Directory в організаційному чи географічному плані.

Організаційні підрозділи можуть містити OU, домени є контейнерами в цьому сенсі та можуть містити кілька вкладених OU. Корпорація Майкрософт рекомендує якомога менше доменів у AD і покладається на організаційний підрозділ для створення структури та покращення впровадження політик і адміністрування.

Якщо є можливість зберігати мільйони об'єктів в Active Directory, кожен об'єкт має бути розміщений і однозначно ідентифікований. З цієї причини об'єкти мають глобальний унікальний ідентифікатор (GUID) (що складається зі 128-бітного числа), який призначає їм система під час створення. GUID залишатиметься в об'єкті, доки його не буде видалено, незалежно від того, чи буде він замінений чи переміщений у інформаційному дереві каталогу (DIT), представленні даних у структурі ієрархічного дерева, що складається з розрізнених імен (DN). GUID об'єкта також буде збережено, якщо ми перемістимо об'єкт між доменами в багатодоменному лісі.

Незважаючи на те, що GUID об'єкта є гнучким, його нелегко запам'ятати, а також він не базується на ієрархії каталогів. З цієї причини частіше використовується інший спосіб посилання на об'єкти, який називається розрізняльнім іменем (DN). Розрізняльні імена можна використовувати для

унікального посилання на об’єкт. Розрізняльні імена визначені в стандарті LDAP як засіб посилання на кожен об’єкт у каталозі.

LDAP базується на протоколі доступу до каталогу (DAP), який є реалізацією мережі X.500. X.500 — це дуже обширна служба каталогів, вбудована в ієрархічну структуру, схожу на DNS. Каталоги X.500 легко шукати, а DAP використовується в мережах X.500, щоб мати можливість запитувати бази даних для пошуку інформації каталогу. Проблема полягає в тому, що DAP накладає велике навантаження на клієнтські комп’ютери та отримує репутацію високого рівня накладних витрат. LDAP був розроблений на основі DAP (RFC 1777), але не мав великих накладних витрат на DAP і не вимагав впровадження мережі X.500. LDAP зберігає функціональність DAP без накладних витрат X.500.

З моменту розробки LDAP став стандартом Інтернету. Ми використовуємо його в пошукових системах і групах новин. LDAP працює добре, є стандартним і використовується в Active Directory для клієнтських запитів. Наприклад, користувач виконує пошук у каталозі, щоб знайти всі «лазерні принтери». LDAP використовує ключові слова для пошуку об’єктів і атрибутів для пошуку всіх лазерних принтерів. Весь доступ до об’єктів Active Directory здійснюється через LDAP і використовується, коли адміністратори змінюють об’єкти Active Directory.

Розрізняльні імена для об’єктів Active Directory зазвичай представлені за допомогою синтаксису та правил, визначених у стандарті LDAP. Логічна структура Active Directory представлена наступним чином:

Домени є основною логічною структурою Active Directory. Домени можуть зберігати мільйони об’єктів. Усі об’єкти в мережі існують у домені, і кожен домен зберігає інформацію лише про об’єкти, які він містить. Домен Active Directory складається з таких компонентів: ієрархічна структура контейнерів і об'єктів на основі X.500; ім’я домену DNS для унікальної ідентифікації; служби безпеки, які автентифікують і авторизують доступ до ресурсів через облікові записи в доменах або довіри з іншими доменами; політики, які визначають, як функціональність обмежена користувачами або машинами в домені.

Контролер домену (DC) (фізична структура, яка є сервером) може мати лише один домен. Неможливо мати кілька доменів на одному DC. Сам домен, незалежно від його вмісту, автоматично створюється як корінь ієрархічної структури, що називається деревом доменів. По суті, це ряд доменів, пов’язаних між собою ієрархічно, усі з використанням суміжної схеми іменування. Кожне дерево домену посилається за іменем, присвоєним кореню дерева. Навіть якщо буде лише один домен, це все одно буде дерево доменів, навіть лише з одним доменом.

Дерева спрощують керування ресурсами та доступ до них, оскільки всі домени в дереві доменів безперечно довіряють один одному за допомогою транзитивної довіри. У транзитивній довірі, якщо домен A довіряє домену B, а домен B довіряє домену C, це означає, що домен A також довіряє домену C.

Довірчі відносини не гарантують безпеки; вони просто створюють потенціал для доступу до ресурсів. Фактичні дозволи на доступ все ще мають бути надані адміністратором. Ось чому ми повинні уникати надання доступу до ресурсів людям або автентифікованим користувачам. Після встановлення довіри кожен у довіреному домені також зможе отримати доступ до ресурсів.

Якщо дерево доменів — це сукупність доменів, то ліс — це сукупність одного або кількох дерев доменів. Це дерево домену має спільний контейнер схеми та конфігурації, а дерева в цілому пов’язані між собою за допомогою транзитивної довіри. Як тільки ми створюємо домен, у нас є ліс. Якщо додати домен до початкового дерева доменів або нове дерево доменів, то залишиться один ліс. Лісова довіра дозволяє адміністраторам створювати перехідну односторонню або двосторонню довіру між двома кореневими доменами лісу. Ця довіра дозволяє всім доменам в одному лісі довіряти всім доменам в іншому лісі, і навпаки.

Якщо ми маємо незалежні бізнес-одиниці і, по суті, хочемо бути ізольованими одна від одної, то нам не обов’язково об’єднувати їх у ліс. Якщо ми просто надамо кожній бізнес-одиниці власний домен, ці бізнес-одиниці створять враження, що вони автономні та ізольовані одна від одної. Однак у Active Directory такого рівня автономії та ізоляції можна досягти лише за допомогою окремих лісів. Це також стосується випадків, коли нам потрібно виконати нормативні чи юридичні вимоги

ізоляції. Нарешті, деякі організації вирішують розгортати Microsoft Exchange в окремих лісах ресурсів для обробки окремих адміністративних структур і вимог.

Після обговорення широкомасштабного представлення (домени, дерева та ліси) Active Directory, тепер ми поговоримо про маломасштабне. Коли ми дивимося всередину домену Active Directory, ми бачимо ієрархічну структуру об'єктів. Ця ієрархія складається з об'єктів, які можуть діяти як контейнери, і об'єктів, які не можуть. Основний тип контейнера, у який ми будемо розміщувати об'єкти, називається організаційною одиницею (OU). Інший тип контейнера, який насправді називається контейнером, також може використовуватися для зберігання ієрархій об'єктів і контейнерів. Фізична структура Active Directory представлена наступним чином:

DC — це сервер, який керує основними службами та є контейнером для бази даних Active Directory. Оскільки домен може містити один або кілька контролерів домену, кожен контролер домену має повну копію доменної частини каталогу. Контролер домену може обслуговувати лише один домен. Контролер домену також автентифікує користувачів, які наразі ввійшли в систему, а також підтримує політику безпеки домену.

Кожен контролер домену зберігає повну копію всієї інформації Active Directory для цього домену, керує будь-якими змінами цієї інформації та реплікує будь-які зміни на інші контролери домену в домені. Усі контролери домену в домені автоматично копіюють інформацію всіх об'єктів цього домену один з одним.

Сайт в Active Directory — це фізична група комп'ютерів. Сайт, за визначенням, включає в себе конкретне географічне розташування, де всі комп'ютери знаходяться в одній або кількох підмережах з хорошим зв'язком. Сайт — це фізичне угруповання комп'ютерів на основі з'єднання TCP/IP, а домен — це логічне угруповання користувачів, комп'ютерів та інших об'єктів Active Directory на основі потреб адміністрування та безпеки. Дуже важливо підтримувати визначення та використання доменів і сайтів.

Кожен контролер домену в доменному лісі контролюється доменними службами Active Directory, включаючи розділи каталогу. Поділ каталогів також відомий як іменування контексту. Розрізняють такі типи поділу:

***Розділ схеми*** містить об'єкти classSchema та attributeSchema, які визначають типи об'єктів, які можуть існувати в лісі. Кожен контролер домену в лісі має репліку того самого розділу схеми.

***Розділ конфігурації*** містить топологію реплікації та інші конфігураційні дані, які мають бути відтворені в лісі. Кожен контролер домену в лісі має репліку тієї самої конфігурації розділу.

***Розділ домену*** містить об'єкти каталогу, такі як користувачі та комп'ютери, пов'язані з локальним доменом. Домен може мати кілька контролерів домену, а ліс може мати кілька доменів. Кожен контролер домену зберігає всі репліки розділу домену для локального домену, але не зберігає репліки розділу домену для інших доменів.

Глобальний каталог (GC) є дуже важливою частиною Active Directory, оскільки він використовується для виконання пошуку в усьому лісі. Як випливає з назви, глобальний каталог — це каталог усіх об'єктів у лісі, який містить підмножину атрибутів для кожного об'єкта. Доступ до GC можна отримати через LDAP на порту 3268 або LDAP/SSL через порт 3269. Глобальний каталог доступний лише для читання і не може бути оновлений безпосередньо.

У багатодоменному лісі зазвичай ми повинні спочатку запитати GC, щоб знайти об'єкт. Тоді ми можемо виконати більше запитів, спрямованих проти контролера домену для об'єкта домену, якщо ми хочемо отримати доступ до всіх атрибутів, доступних на об'єкті.

Атрибути, доступні в глобальному каталозі, є членами часткового набору атрибутів (PAS). Ми можемо додавати та видаляти атрибути до та з PAS за допомогою таких інструментів, як оснащення схеми Active Directory або змінюючи об'єкт attributeSchema для атрибутів безпосередньо в схемі. У Windows 2000 додавання атрибута для PAS змушує всі глобальні каталоги в лісі синхронізувати весь вміст GC. Це може призвести до великих наслідків реплікації та мережевого

трафіку. На щастя, ця проблема була вирішена з Windows Server 2003; Повторна синхронізація GC більше не відбувається після додавання PAS.

Гнучка єдина головна операція (FSMO) містить 5 основних ролей:

1. ***Майстер іменування домену***, який контролер домену керує додаванням і видаленням домену в лісі. Ліс може мати лише один майстер іменування домену, який можна перенести на інший контролер домену через домен активного каталогу та довіру.

2. ***Майстер схеми***, який контролює оновлення даних схеми домену. Це один майстер схеми у всьому лісі. Його можна перенести на інший контролер домену через закріплення майстра схеми Active Directory.

3. ***Головний емулятор PDC***, що у змішаному середовищі 2000 і вікна NT головний емулятор PDC підтримує BDC. Таким чином, він керує зміною облікового запису користувача та пароля та пересилає цю інформацію до BDC Window NT. У середовищі Windows 2000 у рідному режимі головний емулятор PDC отримав перевагу в реплікації пароля облікового запису користувача. Перед невдалим входом перевіряється наявність оновленої інформації. Цю головну роль можна передати іншому контролеру домену через закріплення користувача та комп'ютера Active Directory.

4. ***Майстер відносного ідентифікатора***, який є єдиним майстром відносного ідентифікатора в кожному домені дерева керує виділенням послідовного відносного ідентифікатора (RID) кожному контролеру домену. Це робить усі ідентифікатори безпеки (SID), створені в домені, відносно контролера домену. Цю головну роль можна передати іншому контролеру домену за допомогою активного каталогу та комп'ютера.

5. ***Майстер інфраструктури*** відповідає за керування групами та довідками користувачів. За винятком затримки зміни користувача, коли вони вносяться через домен. Оновлення до іншого домену здійснюється головним контролером домену інфраструктури за допомогою процесу, який називається головною реплікацією. Головну роль можна передати іншому контролеру домену за допомогою активного каталогу користувача та комп'ютера.

Робочу групу найкраще розуміти як однорангову мережу, тобто кожен комп’ютер є самостійним. Він має власний список користувачів, власний контроль доступу та власні ресурси. Щоб користувач міг отримати доступ до ресурсу в іншій робочій групі, комп’ютер цього користувача має бути налаштований на іншому комп’ютері. Робоча група пропонує мало безпеки, крім базового контролю доступу. «Дозвіл на спільний доступ» у вікні дуже простий і не пропонує жодної деталізації «Хто» може отримати доступ до «Що» тощо. Робоча група використовується для малого бізнесу та домашнього використання тощо.

Домен — це довірена група комп’ютерів, які спільно використовують безпеку, контроль доступу та дані, що передаються з централізованого сервера контролера домену. Контролер домену керує всіма аспектами надання користувачам дозволу на вхід. Вони воротар. Більшість доменів використовує активний каталог, який забезпечує більш централізовану точку для розповсюдження програмного забезпечення, керування користувачами та керування комп’ютером.

### 2.6. Файловий сервер

Файловий сервер — це центральний сервер у комп’ютерній мережі, який надає підключеним клієнтам файлові системи або принаймні частини файлової системи. Таким чином, файлові сервери пропонують користувачам центральне місце для зберігання файлів на внутрішніх носіях даних, яке доступне для всіх авторизованих клієнтів [36]. Тут адміністратор сервера визначає суворі правила щодо того, які користувачі мають які права доступу: наприклад, конфігурація або авторизація файлів відповідної файлової системи дозволяє адміністратору встановлювати, які файли можуть переглядати та відкривати певний користувач або група користувачів, і чи можна дані лише переглядати чи також додавати, редагувати чи видаляти.

З файловими серверами, підключеними до Інтернету та налаштованими відповідним чином, користувачі не можуть отримувати доступ до файлів лише через локальну мережу, але й отримують переваги від віддаленого доступу. Це дає змогу отримувати доступ до файлів і зберігати їх на файловому сервері, навіть коли

користувачі перебувають у дорозі. Усі сучасні операційні системи, такі як Windows, Linux або macOS, можна використовувати на файловому сервері, хоча пристрої, доступні в мережі, мають бути сумісні з операційною системою. Але файлові сервери використовуються не лише для зберігання та керування файлами. Вони також часто використовуються як репозиторій для програм, які мають бути доступні багатьом учасникам мережі, і як резервний сервер.

Правильне апаратне забезпечення є основою для надійного файлового сервера. Найважливіше, звичайно, це включає жорсткий диск, який повинен запропонувати достатньо місця для файлів і необхідних програм, а також відповідну операційну систему та програмне забезпечення для використання клієнтів. Сервер також потребує достатньої робочої пам’яті та обчислювальної потужності, щоб якомога швидше та бездоганно обробляти доступ до файлів і програм для різних користувачів. Чи може стандартний ПК задовольнити апаратні вимоги чи потрібна спеціальна настройка сервера, в першу чергу залежить від кількості користувачів.

Спеціальні мережеві протоколи відповідають за зв’язок між файловими серверами та клієнтами: хоча протокол SMB (Server Message Block), розроблений IBM, використовується в локальних мережах із пристроями Windows і macOS, комп’ютери з Unix-подібними системами, такими як дистрибутиви Linux, здебільшого працюють з протоколом NFS (Network File System). Щоб поєднати обидва типи протоколів в одній мережі, клієнти та файлові сервери на базі Unix/Linux повинні бути відповідним чином оснащені програмним забезпеченням, яке реалізує протокол SMB у цих системах — наприклад, набір безкоштовного програмного забезпечення Samba.

Доступ до файлового сервера через Інтернет зазвичай працює через FTP (протокол передачі файлів) або його зашифрований варіант SFTP (безпечний FTP). Крім того, також використовуються зашифровані протоколи SCP (Secure Copy) і WebDAV на основі HTTP. WebDAV використовує той самий порт (80), що й HTTP. Порівняно з такими альтернативами, як FTP або SCP, він має перевагу в тому, що порт зазвичай не потрібно відкривати в клієнті, оскільки він відкритий як стандарт, що дозволяє використовувати всесвітню мережу.

Деякі з протоколів файлового сервера: блок повідомлень сервера (SMB), файлова система мережі (NFS) і протокол передачі файлів (FTP). Блок повідомлень сервера (SMB): мережевий протокол обміну файлами, який дозволяє програмам виконувати деякі операції для запиту послуг для сервера, називається блоком повідомлень сервера (SMB). Такими операціями можуть бути читання або запис файлів у комп'ютерній мережі. Файлові сервери локальної мережі використовують цей протокол. Він підтримується для Windows і macOS.

Мережева файлова система (NFS): розподілена файлова система, яка функціонує для зберігання файлів у мережі, називається мережевою файловою системою (NFS). Ці операції можуть полягати в доступі до файлів (створення, видалення, читання, запис) і каталогів через мережу та дії, наче вони присутні локально.

Протокол передачі файлів (FTP): процес, який передбачає надсилання та отримання файлів між пристроями через мережу, називається протоколом передачі файлів (FTP). Це стандартний протокол зв'язку. Він побудований на архітектурі клієнт-серверної моделі, що означає, що клієнти можуть виконувати інформацію з віддаленої файлової системи безпосередньо. Файлові сервери можна класифікувати як:

***Виділений файловий сервер*** надає сервіси виключно іншим комп'ютерам. Це може бути в певній локальній мережі або мати належним чином авторизований запит на доступ, пов'язаний з комп'ютерною системою. Він призначений для однієї мети — бути файловим сервером. Спеціальний файловий сервер пропонує достатньо місця для зберігання веб-сайту. Крім того, це більш безпечно.

***Невиділений файловий сервер*** схожий на будь-яку іншу робочу станцію, яка дозволяє йому використовувати себе як робочу станцію. Ці файлові сервери можна використовувати одночасно як робочу станцію, а також для щоденних завдань. Невиділений файловий сервер пропонує менше місця для зберігання веб-сайту. Він менш безпечний і може бути скомпрометований шахраєм.

Як уже згадувалося, основні функції файлового сервера полягають у тому, щоб надати багатьом користувачам доступ до збережених файлів і звільнити простір

для зберігання файлів у сховищі [36]. З цієї причини ці сервери особливо популярні як центральне місце зберігання внутрішніх файлів компанії, які стосуються не лише окремих користувачів. У багатьох випадках компанії (особливо в секторі з відкритим вихідним кодом) також використовують файловий сервер як сервер завантаження, підключений до власної веб-пропозиції. Таким чином вони дозволяють своїм клієнтам або відвідувачам веб-сайтів легко завантажувати вибраний вміст, наприклад програми, драйвери, оновлення, зображення або відео.

Другим основним застосуванням файлових серверів є резервне копіювання даних. На відміну від збереження та спільного керування відповідними файлами, це зокрема стосується створення та підтримки звичайних резервних копій — системних чи файлів користувача (або обох) залежно від потреби. Зберігання цих резервних копій на файловому сервері є простою та недорогою альтернативою необхідності планувати та покривати необхідні додаткові вимоги до зберігання на кожному окремому клієнті.

Це працює подібно до того, як файлові сервери використовуються для розміщення програмного забезпечення та забезпечення доступу для всіх авторизованих користувачів: оскільки системи хост-терміналів використовувалися в минулому, обчислювальну потужність і ємність зберігання зручно передати аутсорсингу, тобто клієнтські пристрої мають виконувати лише введення даних і дисплей.

Файлові сервери не вносять жодних змін до існуючих файлів. Це пояснюється тим, що вони зберігають дані як купу двійкових даних і файлів у формі «блобів» (Binary Large Object). Тому вони не виконують жодної додаткової фільтрації чи обробки даних (виконуваних файлів, документів, фотографій і відео). Єдиний спосіб роботи з файловими серверами — зробити файлову систему доступною для клієнтів.

Особливості файлового сервера [36]: кілька користувачів можуть мати доступ до файлів одночасно; встановлюються протоколи авторизації; FTP (протокол передачі файлів) і SFTP (протокол безпечної передачі файлів) використовуються через Інтернет для доступу до файлів; блокування кількох користувачів від

редагування одного файлу одночасно називається блокуванням файлу; для зручності можна використовувати сервер завантаження.

Для багатьох компаній варто розглянути використання файлового сервера з ряду причин. По-перше, звичайно, є перевага центральності, яка гарантує кожному авторизованому учаснику мережі доступ до збережених файлів. Це робить можливим спільну роботу над цими файлами. Конфлікти між різними версіями документа можна практично виключити, оскільки певні дії, такі як редагування або видалення, блокуються для інших користувачів, щойно ви відкриваєте файл. Якщо користувачам доведеться натомість ділитися потрібними файлами у власній системі або передавати їх за допомогою знімних носіїв, це займе значно більше часу та громіздко — і, швидше за все, призведе до різних версій файлів.

Ще одна ключова перевага використання файлових серверів полягає в тому, що це зменшує навантаження на клієнтські ресурси. За винятком особистих документів, практично всі бізнес-файли та резервні копії можуть зберігатися на файловому сервері, залежно від того, як компанія бажає використовувати сховище файлів. А з правильною організацією (включаючи каталоги, папки тощо) користувачі автоматично матимуть набагато кращий огляд усієї інвентаризації файлів.

Якщо файловий сервер налаштовано для віддаленого доступу через Інтернет, файли також доступні в дорозі — подібно до служби онлайн-сховища. Але на відміну від хмарного рішення, компанія зберігає контроль над файлами та їх безпекою в будь-який час. Це є очевидною перевагою перед сторонніми рішеннями [39].

Перелік переваг файлового сервера: 1) легка організація всієї інвентаризації файлів; 2) високий ступінь чіткості; 3) зручний обмін файлами; 4) співпраця без конфліктів версій; 5) полегшення клієнтських комп’ютерів (можливе майже необмежене зберігання); 6) можливий віддалений доступ через WebDAV, (S)FTP або SCP; 7) захист даних і безпека.

Перелік недоліків файлового сервера: 1) дороге налаштування; 2) ризик від вірусів і шкідливих програм; 3) йому бракує незалежності; 4) потрібен час для постійного введення; 5) йому бракує надійності.

Перелічені переваги чітко показують, наскільки цінним може бути файловий сервер. Однак багато компаній роблять помилку, недооцінюючи роботу, пов’язану з налаштуванням і керуванням таким сервером. Компанії часто обходяться без попереднього планування. Як наслідок, через короткий проміжок часу апаратне забезпечення не тільки розвантажується до своїх меж, але й багато переваг файлового сервера не діють. Наприклад, якщо немає чіткого принципу призначення прав, ймовірно виникнуть ситуації, коли користувачі не зможуть виконати необхідні дії. Проблеми також можуть виникнути, коли структура каталогу та папки нечітка, або якщо структура папок не існує взагалі.

Тому, якщо необхідно використовувати файловий сервер, вам слід розглянути ці аспекти з самого початку, а також комплексну концепцію безпеки. У випадку останнього це важливо, якщо файловий сервер також доступний через Інтернет. Встановлення та конфігурація програмного забезпечення безпеки є такими ж критичними, як і навчання співробітників, які мають доступ до файлового сервера. Лише коли вони ознайомлені з такими темами, як кібербезпека та захист даних, налаштовані механізми захисту можуть працювати належним чином [36]. Це також вірно, коли мова йде про зберігання файлів: необхідно чітко встановити, де і як файли повинні зберігатися на файловому сервері, щоб запобігти виникненню хаотичної ситуації з даними.

# РОЗДІЛ 3. МОДУЛЬНА АРХІТЕКТУРА ПРОГРАМНОГО КОМПОНЕНТУ СУІБ

## 3.1. Загальна ідея практичної реалізації

Ця робота базується на ідеї, що кооперативне робоче середовище, що включає синергетичні програмні компоненти, може впоратися з проблемами, які важко вирішити за допомогою традиційного централізованого підходу. Натомість менші програмні об’єкти — програмні компоненти — зі спеціальними можливостями (автономними, реактивними, проактивними та соціальними) використовуються замість цього для динамічної взаємодії. Компоненти моделюють цілі та дії один одного; вони також можуть безпосередньо взаємодіяти (спілкуватися) [38].

Компоненти — це об’єкти програмного забезпечення, які мають дуже конкретне завдання і які самі вирішують, що їм потрібно робити, щоб задовольнити свої цілі. Вони сприймають навколишнє середовище за допомогою датчиків і діють на це середовище за допомогою внутрішніх або фізичних властивостей компонента. Нижче наведено деякі загальні характеристики:

1) ***Автономія***: компонент може діяти від імені іншого без особливих вказівок;

2) ***Комунікація***: компонент може спілкуватися з іншими компонентами на спільну тему дискурсу, обмінюючись послідовністю повідомлень мовою, заснованою на мовленні, яку інші розуміють. Область дискурсу описується його онтологією;

3) ***Мобільність***: компонент може переходити з однієї системи в іншу заздалегідь визначеним способом або на власний розсуд. Відповідно компоненти можуть бути статичними або мобільними;

4) ***Навчання***: компонент може отримувати нову інформацію про середовище, в якому він працює, і динамічно вдосконалювати свою поведінку;

5) ***Співпраця***: компонент може співпрацювати та співпрацювати з іншими агентами або своїм користувачем під час виконання, щоб мінімізувати надмірність і вирішити загальну проблему.

Отже, компонент — це програмний застосунок-частина СУІБ, який дозволить проводити внутрішній контроль та аудит прав доступу користувачів до мережевих папок, розташованих на файловому сервері. Права доступу визначаються групами Active Directory, які призначені на мережеву папку, та в які додані облікові записи користувачів. Архітектуру робочого середовища застосунка наведено на рис. 3.1 нижче. Задля комфорту та подальших планованих змін, рисунок згенеровано за допомогою мови UML (код наведено у додатку А).

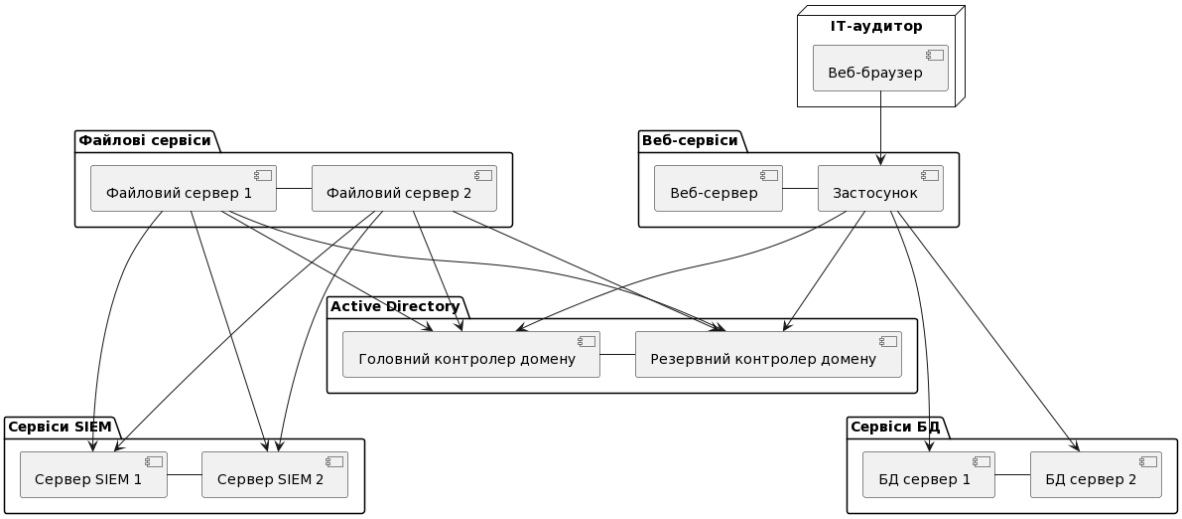


Рис. 3.1. Архітектура робочого середовища застосунку

Власне компонент — це веб-застосунок, написаний на ASP.NET, та впроваджений на веб-сервері підприємства, який дозволяє в режимі реального часу та в комфортному форматуванні переглядати листи доступу до мережевих папок. Застосунок виконує всі запити до Active Directory в фоновому режимі, та, задля безпеки, генерує текстову HTML-сторінку з таблицею груп доступу. На рис. 3.2 наведено архітектуру власне застосунка та класи, які планується використовувати,

також наведені залежності класів. Задля комфорту та подальших планованих змін, рисунку згенеровано за допомогою мови UML (код наведено у додатку Б).

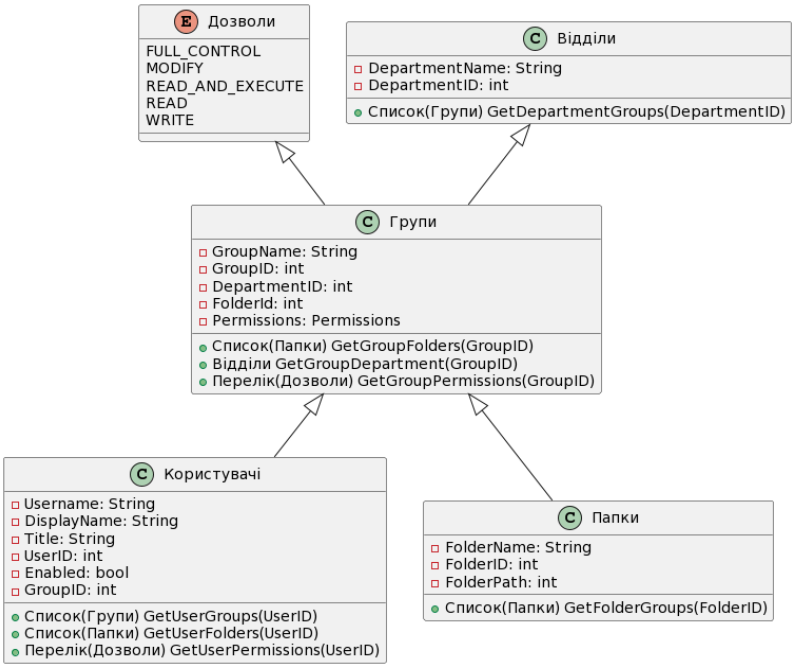


Рис. 3.2. Архітектура застосунка

Практична користь цього застосунку полягає в тому, що його можна використовувати як для поточних завдань IT-відділу по керуванню правами доступу — надання інформації про доступ для володарів інформації в папках, для самоперевірки, що доступ надано правильним обліковим записам і ці облікові записи активні. Також цей застосунок стане у нагоді під час проведення IT-аудиту, коли потрібно надати великий обсяг інформації у зручному форматі.

Застосунок розроблено з використанням шаблону MVC (Model — View — Controller, модель-перегляд-контролер). Шаблон проектування MVC вперше був розроблений Трюгве Реенскаугом у 1970-х роках у Xerox Parc [6]. Тут викладено, що головна мета MVC полягає в тому, щоб подолати розрив між ментальною моделлю людини та цифровою моделлю, яка існує в комп’ютері (рис. 3.3).

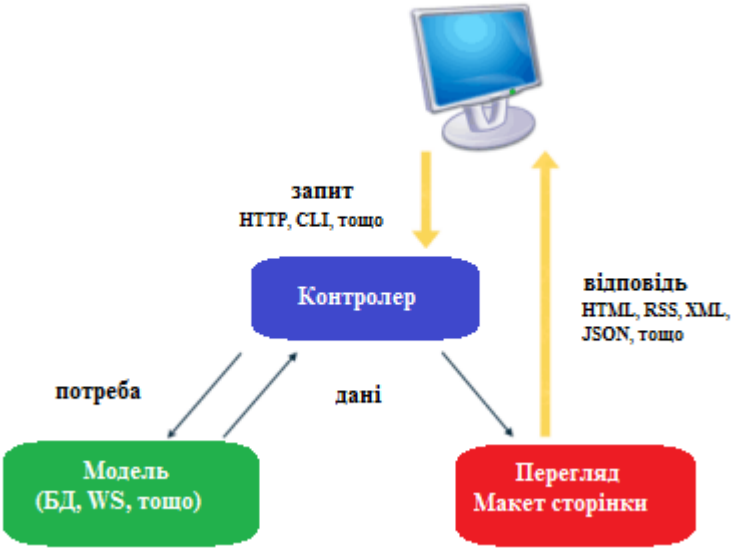


Рис. 3.3. Макет MVC

Максимальна ізоляція функціональних блоків один від одного полегшує розробнику застосунків розуміти та змінювати кожну окрему одиницю, не знаючи всього про інші одиниці. Програма поділяється на три основні категорії: модель основної області програми, представлення даних у цій моделі та взаємодія з користувачем. Шаблон MVC розподіляє обов'язки на три основні ролі, що забезпечує більш ефективну співпрацю. Ці основні ролі — розробка, проектування та інтеграція. Роль розробки беруть на себе досвідчені програмісти, які відповідають за логіку застосунку. Вони піклуються про запит даних, перевірку, обробку тощо.

Роль дизайну належить розробникам, які відповідають за зовнішній вигляд програми. Вони відображають дані, які надходять від розробників, які працюють над першою роллю. Роль інтеграції об'єднує розробників із відповідальністю за склеювання роботи двох попередніх ролей. Шаблон дизайну MVC так добре підходить для розробки веб-застосунків, оскільки вони поєднують кілька технологій, які зазвичай розбиваються на набір рівнів. Крім того, специфічна поведінка MVC може полягати у надсиланні певних представлень до різних типів агентів користувача.

Взаємодія користувача з програмою MVC відбувається за природним циклом: користувач виконує дію, і у відповідь програма змінює свою модель даних і надає

користувачеві оновлене представлення. А потім цикл повторюється. Це дуже зручний варіант для веб-застосунків, які постачаються у вигляді серії HTTP-запитів і відповідей.

### 3.2. Пояснення програмного компоненту та його методів

#### 3.2.1. Загальна інформація про програмний компонент

GALA (Group Access List Application) — це веб-застосунок, розроблений для того, щоб персонал ІКТ міг отримати перераховану інформацію. Застосунок надає таку інформацію: 1) список усіх користувачів із зазначенням їх ID та статусу договору; 2) статус облікового запису користувача (заблокований, прострочений, вимкнений або щойно створений); 3) список доступу для конкретного користувача; 4) список усіх загальних папок на загальних дисках; 5) список користувачів, які мають доступ до конкретних папок із зазначенням рівня доступу; 6) список усіх списків розсилки; 7) список користувачів, які мають доступ до списків розсилки; 8) список усіх поштових скриньок спільного доступу; 9) список користувачів, які мають доступ до спільних поштових скриньок із зазначенням рівня доступу; 10) можливість відправити попередньо згенерований електронний лист до ІТ зі стандартним запитом.

Передбачено наступні випадки використання: 1) зменшення кількості однотипних запитів щодо списків доступу, які надсилаються до підрозділу ІТ, що призведе до зменшення їх завантаження; 2) зменшення часу, необхідного для відправки електронного листа із запитом та отримання відповіді для пересилання запитувачу (у випадку списків доступу); 3) уніфікація формату доставки інформації кінцевим споживачам; 4) централізована заміна численних сценаріїв PowerShell, що виконують подібні дії; 5) централізована заміна сценаріїв PowerShell, які можна запускати на термінальному сервері лише з підвищеним доступом; 6) якщо GALA опублікувати на внутрішньому веб-сайті, застосунок буде доступним через будь-який пристрій, за умови використання двофакторної автентифікації, що покращить надання послуг за допомогою ІТ віддалено; 7) автоматизація окремих дій (отримання списку статусу облікового запису користувача); 8) можливість для

віддаленого персоналу IT отримати інформацію, пов'язану з доступом, яка недоступна для них без підвищеного доступу до термінального сервера; 9) власникам інформації та відповідальному персоналу може бути надано доступ до GALA після схвалення керівника IT для перегляду онлайн-доступу до їхніх відповідних елементів;

GALA має такі системні вимоги: веб-сервер з підтримкою ASP.NET; веб-сервер повинен мати доступ до домену Active Directory; веб-сервер повинен підтримувати автентифікацію Windows; веб-сервер повинен бути доступний для IT та іншого персоналу в усіх офісах, а також з VPN. GALA має такі вимоги до користувача: оновлений веб-браузер (Google Chrome або Internet Explorer); підключення до корпоративної мережі. GALA не потребує жодної бази даних; він миттєво отримує всі дані з Active Directory.

### 3.2.2. Опис методів, використовуваних в програмному компоненті

Отже, як було зазначено вище, GALA — це веб-застосунок, написаний на ASP.NET, що використовує архітектуру MVC. Нижче наведені методи, використовувані застосунком для виконання пошуку в Active Directory та відображення необхідних результатів пошуку.

Одним з ключових класів, що використовується в GALA — є DirectorySearcher. DirectorySearcher шукає та виконує запити щодо ієрархії доменних служб Active Directory за допомогою полегшеного протоколу доступу до каталогу (LDAP). LDAP — це єдиний системний постачальник інтерфейсів служби Active Directory (ADSI), який підтримує пошук у каталозі.

При створенні екземпляру DirectorySearcher, необхідно вказати корінь, який потрібно отримати, і додатковий список властивостей, які потрібно отримати. Властивість SearchRoot дає змогу встановити додаткові властивості для виконання таких завдань: 1) кешування результати пошуку на локальному комп’ютері. Оновлення вносяться до цього локального кешу та передаються в доменні служби Active Directory лише під час виклику методу DirectoryEntry.CommitChanges; 2) керування тривалістю пошуку (ServerTimeLimit); 3) можливість отримувати лише

назви атрибутів (PropertyNamesOnly); 4) виконання пошуку по сторінках (PageSize вказує максимальну кількість об’єктів, які повертаються під час пошуку); 5) керування максимальною кількістю записів для повернення (SizeLimit).

Змінна userDataGlobal містить результати відпрацьовування всіх методів, наведених нижче, та з неї Controller та View беруть дані для відтворення на веб-сторінці. Отже, повернемося до розглядання методів, використовуваних в застосунку в Model.

1) ***Метод getLockedUsers()*** отримує користувачів, чиї облікові записи заблоковано або ще не встановлено. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Результати перераховуються, потім передаються до userInfo і згодом повертаються через список userDataGlobal(). Метод getLockedUsers() повертає список об’єктів класу User. Код методу наведено нижче:

```
public static List<User> getLockedUsers()    {
List<User> userDataGlobal = new List<User>();
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
        ds = MvcApplication.userSearcher(de);
        ds.Filter =
"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)(
ObjectClass=user)(!UserAccountControl:1.2.840.113556.1.4.803:=2)(lastlogon=*)(|((lock
outTime>=1)(!pwdlastset=*)(pwdlastset=0))))"; // виконується фільтрація записів в
Active Directory за певними властивостями
        results = ds.FindAll();
        foreach (SearchResult sr in results)
        {
           var userInfo = new User();
           userInfo.employeeID = sr.getPropertyCount("employeeid");
           userInfo.userName = sr.getPropertyValue("samaccountname");
```

```
            userInfo.displayName = sr.getPropertyValue("displayname");
            userInfo.lastLogon = sr.getDateTimeValue("lastLogon");
            userInfo.lockOutTime = sr.getDateTimeValue("lockouttime");
            #region Отримання даних про електронну пошту та реєстрацію в MDM
            if (sr.Properties["mail"].Count > 0)
            {
               userInfo.Mail = sr.getPropertyValue("mail");
            }              else            {
               userInfo.Mail = "Mailbox pending";
            }
            if (sr.Properties["msexchmobilemailboxflags"].Count > 0)
            {
               userInfo.MDM = "Yes";
            }              else            {
               userInfo.MDM = "No";
            }
            #endregion
            #region Отримання інформації про пароль
            long pwd = (long)sr.Properties["pwdlastset"][0];
            if (pwd == 0)
            {                 userInfo.pwdLastSet = userInfo.pwdExpires = "*Expired";
            }              else            {
               userInfo.pwdExpires =
DateTime.FromFileTime(pwd).AddDays(90).ToString(MvcApplication.formatDateTime);
               userInfo.pwdLastSet =
DateTime.FromFileTime(pwd).ToString(MvcApplication.formatDateTime);
            }
            #endregion
            #region Надсилання запиту по електронній пошті до IT відділу
```

```
            if (sr.Properties["lastlogon"] != null && sr.Properties["lastlogon"].Count > 0)
            {
                userInfo.requestBody = "Шановні колеги, прошу розблокувати обліковий
запис користувача " + userInfo.userName + ". Дякую.";
                userInfo.requestSubject = "Запит на розблокування облікового запису";
            }               else            {
                userInfo.requestBody = "Шановні колеги, прошу надіслати дані для
першого входу для користувача " + userInfo.userName + ". Дякую.";
                userInfo.requestSubject = "Запит даних для входу";
            }
            userInfo.requestTo = "IT@comp.org";
            #endregion
            userDataGlobal.Add(userInfo);
        }
        de.Dispose();
        ds.Dispose();
        return userDataGlobal;
    }
```

2) ***Метод usersLists()*** отримує список усіх користувачів. Метод входить в AD (de) і шукає користувачів за вказаним фільтром (ds.Filter). Він заповнює userInfo — об'єкт класу User(). Метод перевіряє, чи ввімкнено обліковий запис користувача чи ні (ifEnabled). Наприкінці він заповнює список імен користувачів зібраними даними. Метод userLists() повертає список об’єктів класу User. Код методу наведено нижче:

```
public static List<User> usersLists()       {
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
        ds = new DirectorySearcher(de);
```

```
        ds.Filter =
@"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)
(ObjectClass=user))";
        results = ds.FindAll();
        List<User> userNames = new List<User>();
        foreach (SearchResult sr in results)
        {
           var userInfo = new User();
           userInfo.employeeID = sr.getPropertyCount("employeeid");
           userInfo.userName = sr.getPropertyValue("samaccountname");
           userInfo.displayName = sr.getPropertyValue("displayname");
           if (sr.getPropertyCount("msExchRecipientDisplayType") == 1073741824 &&
sr.getPropertyCount("msExchRecipientTypeDetails") == 1)
           {
              userInfo.enabled = sr.ifEnabled();
           }              else              {
              userInfo.enabled = true;
           }              userNames.Add(userInfo);
        }
        de.Dispose();
        ds.Dispose();
        return userNames.ToList();
     }
```

3) ***Метод getUserAccess()*** отримує членство певного користувача в групах безпеки, списках розсилки та спільних поштових скриньках. Вхідний параметр «userName» приймає бажане ім'я користувача для перевірки членства. Метод входить в AD (de) і шукає користувачів за вказаним фільтром (ds.Filter), використовуючи вказаний параметр userName.

Метод працює на наступних етапах: 1) він читає атрибут memberof для знайденого облікового запису AD і заповнює список userDataGlobal() знайденими даними; 2) він читає атрибут msexchdelegatelistbl для знайденого облікового запису AD, щоб перевірити, чи має вказаний користувач повний доступ до будь-якої спільної поштової скриньки; 3) якщо було знайдено повний доступ, далі метод перевіряє, чи має користувач доступ «Надіслати як» до спільної поштової скриньки, де він має повний доступ. Метод getUserAccess() повертає список об'єктів класу ADGroups. Код методу наведено нижче:

```
public static List<ADGroups> getUserAccess(string userName)        {
        List<ADGroups> userDataGlobal = new List<ADGroups>();
        if (userName == null)
        {
           return userDataGlobal;
        }           else           {
           SearchResultCollection results;
           DirectorySearcher ds = null;
           DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
           ds = MvcApplication.userSearcher(de);
           ds.Filter =
"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)(
ObjectClass=user)(samaccountname=" + userName + "))";
           results = ds.FindAll();
           foreach (SearchResult sr in results)
           {
           if (sr.Properties["memberof"] != null && sr.Properties["memberof"].Count > 0)
              {
                 var accessList = sr.Properties["memberof"].GetEnumerator();
                 while (accessList.MoveNext())
                 {
```

```
                    var userInfo = new ADGroups();
                    string[] stringsArray = accessList.Current.ToString().Split(',');
                    userInfo.objectName = stringsArray[0].Replace("CN=", "");
                    userInfo.objectType = stringsArray[1].Replace("OU=", "");
                    userInfo.container = stringsArray[2].Replace("OU=", "");
                    userInfo.location = stringsArray[3].Replace("OU=", "");
                    userInfo.comment = userName;
                    userDataGlobal.Add(userInfo);
                }
            }
            if (sr.Properties["msexchdelegatelistbl"] != null &&
sr.Properties["msexchdelegatelistbl"].Count > 0)
            {
                var access = sr.Properties["msexchdelegatelistbl"].GetEnumerator();
                while (access.MoveNext())
                {
                    var userInfo = new ADGroups();
                    string[] stringsArray = access.Current.ToString().Split(',');
                    userInfo.objectName = stringsArray[0].Replace("CN=", "") +
"@comp.org";
                    userInfo.objectType = stringsArray[1].Replace("OU=", "") + ": FULL";
                    #region shared mailboxes send as
                    SearchResultCollection results0;
                    DirectorySearcher ds0 = null;
                    DirectoryEntry de0 = new DirectoryEntry(MvcApplication.deUA);
                    ds0 = MvcApplication.userSearcher(de0);
                    ds0.Filter =
"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)(
ObjectClass=user)(distinguishedname=" + access.Current.ToString() + "))";
```

```
                    results0 = ds0.FindAll();
                    Guid guid_sendas = Guid.Parse("ab721a54-1e2f-11d0-9819-
00aa0040529b");
                    var user_sid_bytes = (byte[])sr.Properties["objectsid"][0];
                    var user_sid = new SecurityIdentifier(user_sid_bytes, 0);
                    foreach (SearchResult sr0 in results0)
                    {
                       DirectoryEntry sharedmailbox = sr0.GetDirectoryEntry();
                       ActiveDirectorySecurity adsec = sharedmailbox.ObjectSecurity;
                       AuthorizationRuleCollection aces = adsec.GetAccessRules(true, true,
typeof(SecurityIdentifier));
                       foreach (AuthorizationRule ace in aces)
                       {
                          ActiveDirectoryAccessRule adace =
(ActiveDirectoryAccessRule)ace;
                          if (adace.ObjectType == guid_sendas)
                          {
                             if (user_sid.ToString() == adace.IdentityReference.Value)
                             {
                            userInfo.objectType = userInfo.objectType + " AND SEND AS";
                             }
                          }
                       }
                    }
                    de0.Dispose();
                    ds0.Dispose();
                    #endregion
                    userInfo.container = stringsArray[2].Replace("OU=", "");
                    userInfo.location = stringsArray[3].Replace("OU=", "");
```

```
                    userInfo.comment = userName;
                    userDataGlobal.Add(userInfo);
                }
            }
        }
        de.Dispose();
        ds.Dispose();
        return userDataGlobal;
    }
}
```

4) ***Метод getSharedMailboxes()*** отримує список усіх спільних поштових скриньок. Метод входить в AD (de) і шукає спільні поштові скриньки за вказаним фільтром (ds.Filter). Передбачається, що для спільної поштової скриньки встановлено такі атрибути: msexchrecipientdisplaytype = 0 і msexchrecipienttypedetails = 4. Метод getSharedMailboxes() повертає список адрес електронної пошти спільних поштових скриньок у вигляді рядків. Код методу наведено нижче:

```
    public static List<string> getSharedMailboxes()        {
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompSM);
        ds = new DirectorySearcher(de);
        ds.Filter =
@"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)
(ObjectClass=user)(msexchrecipientdisplaytype=0)(msexchrecipienttypedetails=4))";
        results = ds.FindAll();
        List<String> mailboxes = new List<String>();
        foreach (SearchResult sr in results)
        {
```

```
            mailboxes.Add(sr.getPropertyValue("mail"));
        }
        de.Dispose();
        ds.Dispose();
        return mailboxes.ToList();
    }
```

5) ***Метод getMailboxUsers()*** отримує список тих, хто має повний доступ і/або доступ «Надіслати як» до вказаної спільної поштової скриньки. Вхідний параметр «mail» приймає адресу електронної пошти бажаної спільної поштової скриньки для перевірки членства. Метод входить в AD (de) і шукає спільну поштову скриньку за вказаним фільтром (ds.Filter), використовуючи вказаний параметр пошти. Передбачається, що спільна поштова скринька має такі атрибути: msexchrecipientdisplaytype = 0 і msexchrecipienttypedetails = 4. Метод працює за такими кроками: 1) він читає атрибут msexchdelegatelistlink і заповнює список знайденими користувачами. Цей атрибут показує користувачів із повним доступом до вказаної спільної поштової скриньки; 2) метод перевіряє, чи є доступ Send As до вказаної спільної поштової скриньки для будь-якого користувача, і заповнює список знайденими користувачами; 3) метод перевіряє, чи було знайдено те саме ім'я на двох попередніх кроках, і якщо знайдено, призначає значення «FULL AND SEND AS» знайденому імені користувача та заповнює список лише однією копією такого імені користувача. Метод getMailboxUsers() повертає список об'єктів класу User. Код методу наведено нижче:

```
public static List<User> getMailboxUsers(string mail)        {
        List<User> userNamesFull = new List<User>();
        List<User> userNamesSendAs = new List<User>();
        List<User> userNames = new List<User>();
        if (mail == null)
        {
```

```
            return userNames;
        }           else           {
            SearchResultCollection results;
            DirectorySearcher ds = null;
            DirectoryEntry de = new DirectoryEntry(MvcApplication.deUA);
            ds = MvcApplication.userSearcher(de);
            ds.Filter =
@"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)
(ObjectClass=user)(msexchrecipientdisplaytype=0)(msexchrecipienttypedetails=4)(mail="
+ mail + "))";
            results = ds.FindAll();
            Guid guid_sendas = Guid.Parse("ab721a54-1e2f-11d0-9819-00aa0040529b");
            foreach (SearchResult sr in results)
            {
               #region Виконання пошуку користувачів з повним доступом
               if (sr.Properties["msexchdelegatelistlink"] != null &&
sr.Properties["msexchdelegatelistlink"].Count > 0)
               {
                  var access = sr.Properties["msexchdelegatelistlink"].GetEnumerator();
                  while (access.MoveNext())
                  {
                     SearchResultCollection results0;
                     DirectorySearcher ds0 = null;
                     DirectoryEntry de0 = new
DirectoryEntry(MvcApplication.deUACompUsers);
                     ds0 = new DirectorySearcher(de0);
                     ds0.Filter =
@"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)
(ObjectClass=user)(distinguishedname=" + access.Current.ToString() + "))";
```

```
                    results0 = ds0.FindAll();
                    foreach (SearchResult sr0 in results0)
                    {
                       var userInfo = new User();
                       userInfo.userName = sr0.getPropertyValue("samaccountname");
                       userInfo.displayName = sr0.getPropertyValue("displayname");
                       userInfo.comment = mail;
                       userInfo.organizationUnit =
sr0.getPropertyValue("complocationcode");
                       userInfo.membership = "FULL";
                       if (sr0.Properties["comppositionname"] != null &&
sr0.Properties["comppositionname"].Count > 0)
                       {
                          userInfo.position =
sr0.Properties["comppositionname"][0].ToString();
                          userInfo.positionNumber =
sr0.getPropertyCount("comppositionsortorder");
                       }                    else                    {
                          if (sr0.getPropertyCount("msExchRecipientDisplayType") ==
1073741824 && sr0.getPropertyCount("msExchRecipientTypeDetails") == 1)
                          {
                             userInfo.position = "UNKNOWN";
                             userInfo.positionNumber = 100000;
                          }                          else                    {
                             userInfo.position = "SHARED MAILBOX";
                             userInfo.positionNumber = 100000;
                          }
                       }
                       if (sr0.getPropertyCount("msExchRecipientDisplayType") ==
```

```
1073741824 && sr0.getPropertyCount("msExchRecipientTypeDetails") == 1)
                    {
                        userInfo.enabled = sr0.ifEnabled();
                    }                           else                        {
                        userInfo.enabled = true;
                    }
                    userNamesFull.Add(userInfo);
                }
                de0.Dispose();
                ds0.Dispose();
            }
        }
        #endregion
        #region пошук всіх користувачів з доступом лише на відправку від імені
поштової скриньки
        DirectoryEntry sharedmailbox = sr.GetDirectoryEntry();
        ActiveDirectorySecurity adsec = sharedmailbox.ObjectSecurity;
        AuthorizationRuleCollection aces = adsec.GetAccessRules(true, true,
typeof(SecurityIdentifier));
        foreach (AuthorizationRule ace in aces)
        {
            ActiveDirectoryAccessRule adace = (ActiveDirectoryAccessRule)ace;
            if (adace.ObjectType == guid_sendas)
            {
                SearchResultCollection results1;
                DirectorySearcher ds1 = null;
                DirectoryEntry de1 = new
DirectoryEntry(MvcApplication.deUACompUsers);
                ds1 = new DirectorySearcher(de1);
```

```
                ds1.Filter =
@"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)
(ObjectClass=user)(objectsid=" + adace.IdentityReference.Value + "))";
                results1 = ds1.FindAll();
                foreach (SearchResult sr1 in results1)
                {
                   var userInfo = new User();
                   userInfo.userName = sr1.getPropertyValue("samaccountname");
                   userInfo.displayName = sr1.getPropertyValue("displayname");
                   userInfo.organizationUnit =
sr1.getPropertyValue("complocationcode");
                   userInfo.membership = "SEND AS";
                   if (sr1.Properties["comppositionname"] != null &&
sr1.Properties["comppositionname"].Count > 0)
                   {
                      userInfo.position =
sr1.Properties["comppositionname"][0].ToString();
                      userInfo.positionNumber =
sr1.getPropertyCount("comppositionsortorder");
                   }                    else                    {
                      if (sr1.getPropertyCount("msExchRecipientDisplayType") ==
1073741824 && sr1.getPropertyCount("msExchRecipientTypeDetails") == 1)
                      {
                         userInfo.position = "UNKNOWN";
                         userInfo.positionNumber = 100000;
                      }                          else                          {
                         userInfo.position = "SHARED MAILBOX";
                         userInfo.positionNumber = 100000;
                      }
```

```
                    }
                    if (sr1.getPropertyCount("msExchRecipientDisplayType") ==
1073741824 && sr1.getPropertyCount("msExchRecipientTypeDetails") == 1)
                    {
                        userInfo.enabled = sr1.ifEnabled();
                    }                           else                        {
                        userInfo.enabled = true;
                    }
                    userNamesSendAs.Add(userInfo);
                }
                de1.Dispose();
                ds1.Dispose();
            }
        }
        #endregion
        #region перевірка чи має користувач обидва види доступу
        foreach (User in userNamesFull)
        {
            bool coincidence = false;
            foreach (User user1 in userNamesSendAs)
            {
                if (user.displayName == user1.displayName)
                {
                    coincidence = true;
                    user.membership = "FULL and SEND AS";
                    userNames.Add(user);
                }
            }
            if (coincidence == false)
```

```
            {
                userNames.Add(user);
            }
        }
        foreach (User in userNamesSendAs)
        {
            bool coincidence = false;
            foreach (User user1 in userNamesFull)
            {
                if (user.displayName == user1.displayName)
                {
                    coincidence = true;
                }
            }
            if (coincidence == false)
            {
                userNames.Add(user);
            }
        }
        #endregion
    }
    de.Dispose();
    ds.Dispose();
    return userNames;
}       }
```

6) ***Метод getSeparatingUsers()*** отримує список користувачів, які звільнюються або звільнені з компанії відповідно до оновлень із запису HR SAP. Метод входить в AD (de) і шукає користувачів за вказаним фільтром (ds.Filter). Передбачається, що користувач із призначеною датою звільнення має атрибут accountExpires більше 0 і

менше 9223372036854775807 («ніколи»): метод читає атрибут accountExpires і заповнює список знайденими користувачами. Цей атрибут показує користувачів із зазначеною датою звільнення. Метод getSeparatingUsers() повертає список об’єктів класу User. Код методу наведено нижче:

```
public static List<User> getSeparatingUsers()       {
        List<User> userDataGlobal = new List<User>();
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
        ds = MvcApplication.userSearcher(de);
        ds.Filter =
@"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)
(ObjectClass=user)(!(accountExpires=0))(!(accountExpires=9223372036854775807)))";
        results = ds.FindAll();
        foreach (SearchResult sr in results)
        {
           var userInfo = new User();
           userInfo.employeeID = sr.getPropertyCount("employeeid");
           userInfo.userName = sr.getPropertyValue("samaccountname");
           userInfo.displayName = sr.getPropertyValue("displayname");
           long acc = (long)sr.Properties["accountExpires"][0];
           userInfo.accountExpires =
DateTime.FromFileTime(acc).ToString(MvcApplication.formatDateTime);
           userDataGlobal.Add(userInfo);
        }
        de.Dispose();
        ds.Dispose();
        return userDataGlobal;
     }
```

7) ***Метод getNewUsers()*** отримує список користувачів, які мають приєднатися до компанії відповідно до оновлених записів HR SAP. Метод входить в AD (de) і шукає користувачів за вказаним фільтром (ds.Filter). Передбачається, що новий користувач має лише створений обліковий запис AD, а атрибути lastLogon і pwdLastSet дорівнюють 0. Метод читає атрибути lastLogon і pwdLastSet і заповнює список знайденими користувачами. Метод getNewUsers() повертає список об'єктів класу User. Код методу наведено нижче:

```
public static List<User> getNewUsers()        {
        List<User> userDataGlobal = new List<User>();
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
        ds = MvcApplication.userSearcher(de);
        ds.Filter =
"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)(
ObjectClass=user)(!UserAccountControl:1.2.840.113556.1.4.803:=2)(!lastlogon=*)(|((loc
koutTime>=1)(!pwdlastset=*)(pwdlastset=0))))";
        results = ds.FindAll();
        foreach (SearchResult sr in results)
        {
           var userInfo = new User();
           userInfo.employeeID = sr.getPropertyCount("employeeid");
           userInfo.userName = sr.getPropertyValue("samaccountname");
           userInfo.displayName = sr.getPropertyValue("displayname");
           userInfo.lastLogon = sr.getDateTimeValue("lastLogon");
           #region Email
           if (sr.Properties["mail"].Count > 0)
           {
              userInfo.Mail = sr.getPropertyValue("mail");
```

```
            }               else            {
                userInfo.Mail = "Mailbox pending";
            }
            #endregion
            #region Password: last set
            long pwd = (long)sr.Properties["pwdlastset"][0];
            userInfo.pwdLastSet = userInfo.pwdExpires = "*Expired";
            #endregion
            #region Send Request
              if (sr.Properties["lastlogon"] != null && sr.Properties["lastlogon"].Count > 0)
             {
            userInfo.requestBody = "Шановні колеги, прошу надіслати дані для першого
входу користувача " + userInfo.userName + ". Дякую.";
            userInfo.requestSubject = "Запит на дані для першого входу";
            }
            userInfo.requestTo = "IT@comp.org";
            #endregion
            userDataGlobal.Add(userInfo);
        }
```

8) ***Метод getExpiringPasswords()*** отримує користувачів, чиї облікові записи заблоковано або ще не встановлено. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Результати перераховуються, потім передаються до userInfo і згодом повертаються через список userDataGlobal(). Метод getLockedUsers() повертає список об’єктів класу User. Код методу наведено нижче:

```
public static List<User> getExpiringPasswords()       {
        List<User> userDataGlobal = new List<User>();
        SearchResultCollection results;
        DirectorySearcher ds = null;
```

```
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
        ds = MvcApplication.userSearcher(de);
        ds.Filter =
"(&(ObjectCategory=CN=Person,CN=Schema,CN=Configuration,DC=comp,DC=intra)(
ObjectClass=user)(!UserAccountControl:1.2.840.113556.1.4.803:=2)(lastlogon=*)(!emplo
yeeid=0)(|((pwdlastset=*)(pwdlastset=0))))";
        results = ds.FindAll();
        foreach (SearchResult sr in results)
        {
           var userInfo = new User();
           userInfo.employeeID = sr.getPropertyCount("employeeid");
           userInfo.userName = sr.getPropertyValue("samaccountname");
           userInfo.displayName = sr.getPropertyValue("displayname");
           userInfo.lastLogon = sr.getDateTimeValue("lastLogon");
           #region Password: last set and expiration
           long pwd = (long)sr.Properties["pwdlastset"][0];
           if (pwd == 0)
           {                userInfo.pwdLastSet = userInfo.pwdExpires = "*Expired";
           }              else              {
             userInfo.pwdExpires =
DateTime.FromFileTime(pwd).AddDays(90).ToString(MvcApplication.formatDateTime);
               userInfo.pwdLastSet =
DateTime.FromFileTime(pwd).ToString(MvcApplication.formatDateTime);
           }              #endregion
           userDataGlobal.Add(userInfo);
        }          de.Dispose();
        ds.Dispose();
        return userDataGlobal;
     }
```

9) ***Метод getQuarantined()*** отримує користувачів, чиї пристрої ActiveSync заблоковано в MDM. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Результати перераховуються, потім передаються до userInfo і згодом повертаються через список userDataGlobal(). Метод getQuarantined() повертає список об'єктів класу User. Код методу наведено нижче:

```
public static List<User> getQuarantined()       {
        List<User> userDataGlobal = new List<User>();
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUACompUsers);
        ds = MvcApplication.userSearcher(de);
        ds.Filter = "(&(ObjectClass=*))";
        ds.PropertiesToLoad.Add("msExchActiveSyncDevice");
        ds.PropertiesToLoad.Add("msExchDeviceAccessState");
        //ds.PropertiesToLoad.Add("user");
        ds.PropertiesToLoad.Add("samaccountname");
        ds.PropertiesToLoad.Add("displayname");
        ds.PropertiesToLoad.Add("msExchDeviceType");
        ds.PropertiesToLoad.Add("msExchDeviceID");
        ds.PropertiesToLoad.Add("msExchDeviceIMEI");
        ds.PropertiesToLoad.Add("msExchDeviceModel");
        results = ds.FindAll();
        foreach (SearchResult sr in results)
        {
           var userInfo = new User();
           userInfo.employeeID = sr.getPropertyCount("employeeid");
           userInfo.userName = sr.getPropertyValue("samaccountname");
           userInfo.displayName = sr.getPropertyValue("displayname");
           userInfo.deviceType = sr.getPropertyValue("msExchDeviceType");
```

```
            userInfo.deviceModel = sr.getPropertyValue("msExchDeviceModel");
            userInfo.deviceID = sr.getPropertyCount("msExchDeviceID");
            userInfo.deviceIMEI = sr.getPropertyCount("msExchDeviceIMEI");
            userInfo.deviceAccessState =
sr.getPropertyValue("msExchDeviceAccessState");
           #region Отримання даних про електронну пошту та реєстрацію в MDM
            if (sr.Properties["msExchActiveSyncDevice"].Count == 0)
            {
               userInfo.airWatch = "Заблоковано";
            }               else            {
               userInfo.airWatch = "Не заблоковано";
            }
            #endregion */
            #region Надсилання запиту
            if (sr.Properties["lastlogon"] != null && sr.Properties["lastlogon"].Count > 0)
            {
               userInfo.requestBody = "Шановні колеги, прошу розблокувати
обліковий запис користувача " + userInfo.userName + ". Дякую.";
               userInfo.requestSubject = "Запит на розблокування облікового запису";
            }               else            {
               userInfo.requestBody = "Шановні колеги, прошу надіслати дані для
входу користувача " + userInfo.userName + ". Дякую.";
               userInfo.requestSubject = "Запит даних для входу";
            }
            userInfo.requestTo = "IT@comp.org";
            #endregion
            userDataGlobal.Add(userInfo);
         }
         de.Dispose();
```

```
        ds.Dispose();
        return userDataGlobal;
    }   }
```

10) ***Метод foldersList()*** отримує всі спільні папки на дисках Departments:\ та Projects:\. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Передбачається, що атрибут опису групи безпеки містить \\comp.intra\global\Departments\COMP\UA* або \\comp.intra\global\Projects\COMP\UA*. Метод foldersList() повертає список імен спільних папок у вигляді рядків. Код методу наведено нижче:

```
public static List<String> foldersList()        {
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUA);
        ds = new DirectorySearcher(de);
        ds.Filter =
@"(&(objectClass=group)(objectCategory=CN=Group,CN=Schema,CN=Configuration,D
C=comp,DC=intra)(|(description=" + MvcApplication.departments + "*)(description=" +
MvcApplication.projects + "*)))";
        results = ds.FindAll();
        List<string> folderName = new List<string>();
        foreach (SearchResult sr in results)
        {
           folderName.Add(sr.getPropertyValue("description"));
        }
        de.Dispose();
        ds.Dispose();
        return folderName.Distinct().ToList();
    }
```

11) ***Метод getADGroups()*** отримує список користувачів, які мають доступ до вказаної спільної папки. Вхідний параметр «folderPath» приймає потрібну назву спільної папки для перевірки списку учасників. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Метод працює на наступних етапах: 1) він знаходить усі групи безпеки на основі вказаного імені спільної папки за допомогою атрибута description; 2) він шукає кожну знайдену групу безпеки в AD і отримує її атрибут члена; 3) він шукає інформацію про кожного користувача, знайдену в атрибуті member; 4) Він перевіряє тип доступу до спільної папки для кожного знайденого користувача та відповідно встановлює значення членства для кожного типу. Метод getADGroups() повертає список об'єктів класу User. Код методу наведено нижче:

```
public static List<User> getADGroups(string folderPath)        {
        List<User> userDataGlobal = new List<User>();
        if (folderPath == null)
        {
           return userDataGlobal;
        }           else           {
           List<string> groups = new List<string>();
           SearchResultCollection results1;
           DirectorySearcher ds1 = null;
           DirectoryEntry de1 = new DirectoryEntry(MvcApplication.deUA);
           ds1 = new DirectorySearcher(de1);
           ds1.Filter = "(&(description=" + folderPath + "))";
           results1 = ds1.FindAll();
           foreach (SearchResult sr1 in results1)
           {
              groups.Add(sr1.getPropertyValue("CN"));
           }
           de1.Dispose();
```

```
            ds1.Dispose();
            foreach (string groupName in groups)
            {
               SearchResultCollection results;
               DirectorySearcher ds = null;
               DirectoryEntry de = new DirectoryEntry(MvcApplication.deUA);
               ds = MvcApplication.userSearcher(de);
               ds.Filter =
"(&(objectClass=group)(objectCategory=CN=Group,CN=Schema,CN=Configuration,DC
=comp,DC=intra)(CN=" + groupName + "))";
               results = ds.FindAll();
               foreach (SearchResult sr in results)
               {
                  //if members attribute exists and not empty
                  if (sr.Properties["member"] != null && sr.Properties["member"].Count > 0)
                  {
                     var users = sr.Properties["member"].GetEnumerator();
                     while (users.MoveNext())
                     {
                        SearchResultCollection results0;
                        DirectoryEntry de0 = new DirectoryEntry(MvcApplication.deRoot);
                        DirectorySearcher ds0 = null;
                        ds0 = MvcApplication.userSearcher(de0);
                        ds0.Filter = "(&(distinguishedName=" + users.Current.ToString() +
"))";
                        results0 = ds0.FindAll();
                        //get user-related data
                        foreach (SearchResult sr0 in results0)
                        {
```

```
                    var userInfo = new User();
                    userInfo.displayName = sr0.getPropertyValue("displayname");
                    userInfo.userName = sr0.getPropertyValue("samaccountname");
                    userInfo.organizationUnit =
sr0.getPropertyValue("complocationcode");
                    userInfo.comment =
folderPath.Replace(MvcApplication.departments,
@"Departments:\").Replace(MvcApplication.projects, @"Project:\");
                    userInfo.enabled = sr0.ifEnabled();
                    if (sr0.Properties["comppositionname"] != null &&
sr0.Properties["comppositionname"].Count > 0)
                    {
                       userInfo.position =
sr0.Properties["comppositionname"][0].ToString();
                       userInfo.positionNumber =
sr0.getPropertyCount("comppositionsortorder");
                    }                          else                          {
                       userInfo.position = "UNKNOWN ";
                       userInfo.positionNumber = 100000;
                    }
                    //verifying group type
                    if (groupName.EndsWith("_RO"))
                    {
                       userInfo.membership = "READ ONLY";
                    }                          else {
                       userInfo.membership = "READ and WRITE";
                    }
                    userDataGlobal.Add(userInfo);
                 }                       de0.Dispose();
```

```
                    ds0.Dispose();                    }
                }
            }                de.Dispose();
            ds.Dispose();            }
        return userDataGlobal;
    }
}
```

12) ***Метод distributionLists()*** отримує список усіх списків розсилки. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Припускається, що атрибут msexchrecipientdisplaytype = 1. Метод distributionLists() повертає список електронних адрес списків розсилки у вигляді рядків. Код методу наведено нижче:

```
public static List<String> distributionLists()      {
        SearchResultCollection results;
        DirectorySearcher ds = null;
        DirectoryEntry de = new DirectoryEntry(MvcApplication.deUA);
        ds = new DirectorySearcher(de);
        ds.Filter =
@"(&(objectClass=group)(objectCategory=CN=Group,CN=Schema,CN=Configuration,D
C=comp,DC=intra)(msexchrecipientdisplaytype=1))";
        results = ds.FindAll();
        List<string> listName = new List<string>();
        foreach (SearchResult sr in results)
        {
            string distinguishedName = sr.getPropertyValue("distinguishedname");
            if (distinguishedName != null)
            {
                List<string> splitDN = new List<string>(distinguishedName.Split(','));
```

```
            if (splitDN.Count > 6)
            {
               List<string> sliceDN = splitDN.GetRange(splitDN.Count-5, 5);
               string OU = string.Join(",", sliceDN.ToArray());
               if (MvcApplication.deUAComp.Contains(OU))
               {                    listName.Add(sr.getPropertyValue("mail"));
               }
            }          }
      }          de.Dispose();
      ds.Dispose();
      return listName.Distinct().ToList();
   }
```

13) ***Метод getDLMembers()*** отримує членів указаного списку розсилки. Вхідний параметр "mail" приймає бажану електронну адресу списку розсилки для перевірки членства. Цей метод входить в AD (de) і виконує пошук за фільтром, указаним у фільтрі ds.Filter. Метод працює на наступних етапах: 1) він отримує атрибут члена для знайденого списку розсилки; 2) він шукає інформацію про кожного користувача, знайдену в атрибуті member. Метод getDLMembers() повертає список об’єктів класу User. Код методу наведено нижче:

```
public static List<User> getDLMembers(string mail)       {
      List<User> userDataGlobal = new List<User>();
      if (mail == null)
      {
         return userDataGlobal;
      }          else          {
         SearchResultCollection results;
         DirectorySearcher ds = null;
         DirectoryEntry de = new DirectoryEntry(MvcApplication.deUA);
```

```
            ds = MvcApplication.userSearcher(de);
            ds.Filter =
"(&(objectClass=group)(objectCategory=CN=Group,CN=Schema,CN=Configuration,DC
=comp,DC=intra)(mail=" + mail + "))";
            results = ds.FindAll();
            foreach (SearchResult sr in results)
            {
               if (sr.Properties["member"] != null && sr.Properties["member"].Count > 0)
               {
                  var users = sr.Properties["member"].GetEnumerator();
                  while (users.MoveNext())
                  {
                     SearchResultCollection results0;
                     DirectoryEntry de0 = new DirectoryEntry(MvcApplication.deRoot);
                     DirectorySearcher ds0 = null;
                     ds0 = MvcApplication.userSearcher(de0);
                     ds0.Filter = "(&(distinguishedName=" + users.Current.ToString() + "))";
                     results0 = ds0.FindAll();
                     foreach (SearchResult sr0 in results0)
                     {
                        var userInfo = new User();
                        userInfo.displayName = sr0.getPropertyValue("displayname");
                        userInfo.userName = sr0.getPropertyValue("samaccountname");
                        userInfo.organizationUnit =
sr0.getPropertyValue("complocationcode");
                        userInfo.comment = mail;
                        if (sr0.getPropertyCount("msExchRecipientDisplayType") ==
1073741824 && sr0.getPropertyCount("msExchRecipientTypeDetails") == 1)
                        {
```

```
                    userInfo.enabled = sr0.ifEnabled();
                }                    else                    {
                    userInfo.enabled = true;
                }
                if (sr0.Properties["comppositionname"] != null &&
sr0.Properties["comppositionname"].Count > 0)
                {
                    userInfo.position =
sr0.Properties["comppositionname"][0].ToString();
                    userInfo.positionNumber =
sr0.getPropertyCount("comppositionsortorder");
                }                    else                    {
                    if (sr0.getPropertyCount("msExchRecipientDisplayType") ==
1073741824 && sr0.getPropertyCount("msExchRecipientTypeDetails") == 1)
                    {
                        userInfo.position = "UNKNOWN";
                        userInfo.positionNumber = 100000;
                    }                        else                    {
                        userInfo.position = "SHARED MAILBOX";
                        userInfo.positionNumber = 100000;
                    }
                }
                userDataGlobal.Add(userInfo);
            }                    de0.Dispose();
            ds0.Dispose();                }
        }                de.Dispose();
        ds.Dispose();            }
      return userDataGlobal;
    }
```

```
    }
}
```

Окрім методів, описаних в Controller, застосунок також містить базові засоби для аудиту та захисту доступу до інформації, розташовані в файлі Global (тобто застосовані до всього веб-застосунку).

А) ***Ведення журналу користувачів, що відкривали застосунок***. Код наведно нижче:

```
protected void Application_Start()        {
    log4net.Config.XmlConfigurator.Configure();
    Logger.InitLogger();
    AreaRegistration.RegisterAllAreas();
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    BundleConfig.RegisterBundles(BundleTable.Bundles);
}
```

Б) ***Система захисту від краулерів даних***. Код, наведений нижче, починає працювати в момент, коли користувач відкриває першу сторінку. Якщо ж користувач відкриває сторінки занадто часто та занадто багато, то веб-застосунок поверне йому помилку та запросить почекати 5 хвилин, щоб мати змогу знову відкривати сторінки та посилання. Код наведено нижче:

```
protected void Session_Start(object sender, EventArgs e)        {
    string u = User.Identity.Name;
    if (Application[u] == null)
    {
        var credit = new Models.Credit();
        credit.userName = u;
        credit.timestamp = DateTime.Now;
```

```
            credit.credit = 16;
            Application.Lock();
            Application[u] = credit;
            Application.UnLock();
        }
    }

    protected void Application_PreRequestHandlerExecute()
    {
        string u = User.Identity.Name;
        if (Application[u] == null)
        {
            var cred = new Models.Credit();
            cred.userName = u;
            cred.credit = 16;
            cred.timestamp = DateTime.Now;
            Application.Lock();
            Application[u] = cred;
            Application.UnLock();
        }
        var credupd = new Models.Credit();
        credupd = (Models.Credit)Application[u];
        DateTime ts = DateTime.Now;
        TimeSpan diff = ts.Subtract(credupd.timestamp);
        if (diff.TotalSeconds >= 60)
        {
            credupd.credit = 16;
        }
        if (credupd.credit <= 0)
```

```
        {
            Response.StatusCode = 403;
            Response.End();
        }
        credupd.credit--;
        credupd.timestamp = ts;
        Application.Lock();
        Application[u] = credupd;
        Application.UnLock();
    }
```

Далі розглянемо один із контролерів (ADGroupsListingController), який бере дані з моделі та передає їх до HTML-сторінки перегляду. Цей контролер відноситься до категорії Users і керує сторінками, що містять список загальнодоступних папок. Отже, він використовує заповнені змінні та методи, що використовувалися в моделі, сортує дані відповідним чином і розподіляє їх за відповідними сторінками перегляду. Також, контролер містить елементи контролю за тими, хто відкривав відповідні сторінки перегляду (Logger.Log.Info). Код контролеру наведено нижче:

```
public class ADGroupsListingController : Controller    {
    // GET: ADGroupsListing
    public ActionResult Groups_Listing(string folder)
    {
        List<User> empty = new List<Models.User>();
        if (folder == null)
        {
            Logger.Log.Info(User.Identity.Name + " " + " " + Request.Url.ToString());
            return View(empty);
        }           else          {
            List<User> listADGroups =
```

```
Models.ADGroups.getADGroups(folder.Replace(@"\", @"\5C"));
            List<User> sortedList = listADGroups.OrderBy(o =>
o.membership).ThenBy(o => o.organizationUnit).ThenBy(o =>
o.positionNumber).ThenBy(o => o.position).ThenBy(o => o.displayName).ToList();
            Logger.Log.Info(User.Identity.Name + " accessed " +
Request.Url.ToString());
            return View(sortedList);
        }
    }
    public ActionResult Index(string searchString, int page = 0)
    {
        List<String> folders = Models.ADGroups.foldersList();
        List<String> sortedFolders = folders.OrderBy(o => o).ToList();
        var count = sortedFolders.Count();
        var data = sortedFolders.Skip(page *
MvcApplication.pageSize).Take(MvcApplication.pageSize).ToList();
        ViewBag.MaxPage = (count / MvcApplication.pageSize) - (count %
MvcApplication.pageSize == 0 ? 1 : 0);
        ViewBag.Page = page;
        var strings = searchString;
        if (!String.IsNullOrEmpty(searchString))
        {
            foreach (var splitString in strings.Split(' '))
            {
                data = sortedFolders.Where(s => s.Contains(searchString) ||
s.ToLower().Contains(searchString)).ToList();
            }
        }
        ViewData["Search"] = searchString;
```

```
        Logger.Log.Info(User.Identity.Name + " accessed " + Request.Url.ToString());
        return View(data);
    }    }
```

Далі, контролер ADGroupsListingController передає дані на дві сторінки перегляду . Перша сторінка містить власне список загальнодоступних папок:

```
@model IEnumerable<GALA.Models.User>
@{ViewBag.Title = "List of Users";}
<body id="useraccess">
  <div>
    <table class="table table-striped table-bordered">
      <caption><h2>Users List</h2><br /></caption>
      <tr>
        <th>#</th>
        <th>Full name</th>
        <th>Username<sup>[IRMA ID]</sup></th>
      </tr>
      @{int count = (ViewBag.Page * MvcApplication.pageSize) + 1; }
      @foreach (var el in Model)
      {
        <tr>
          <td>@count</td>
          <td>
            <a href="~/UserAccess/Access_Listing?user=@el.userName">
              @{if (el.enabled)
                {                @el.displayName
                }                        else                    {
                  <del>@el.displayName</del>
                }
```

```
                }                    </a>
            </td>                <td>
                <a href="~/UserAccess/Access_Listing?user=@el.userName">
                    @{if (el.enabled)
                    {       @el.userName<sup>[@el.employeeID]</sup>
                    }                           else                        {
                        <del>@el.userName<sup>[@el.employeeID]</sup></del>
                    }                       }                   </a>
            </td>
        </tr>
                        count++;
                    }
</table>        <br />
@if (ViewBag.Page == 0)
    {
    <a href="@Url.Action("Index", new { page = ViewBag.Page - 1 })"
       class="btn btn-default" disabled>
        &laquo; Prev
    </a>
}
@if (ViewBag.Page > 0)
    {
    <a href="@Url.Action("Index", new { page = ViewBag.Page - 1 })"
       class="btn btn-default">
        &laquo; Prev
    </a>
}
@if (ViewBag.Page < ViewBag.MaxPage && ViewData["Search"] == null)
    {
```

```
        <a href="@Url.Action("Index", new { page = ViewBag.Page + 1 })"
           class="btn btn-default">
            Next &raquo;
        </a>
    }
    @if (ViewBag.Page == ViewBag.MaxPage || ViewData["Search"] != null)
        {
        <a href="@Url.Action("Index", new { page = ViewBag.Page + 1 })"
           class="btn btn-default" disabled>
            Next &raquo;
        </a>
    }
  </div>
  <footer>
    <p align="right">Total pages: @(ViewBag.MaxPage + 1).</p>
    <p align="right">If the name is strikeout, consider the account to be disabled.</p>
  </footer>
</body>
```

Далі, якщо ж вибрано потрібну папку, відкривається пов’язана сторінка, що містить список користувачів, що мають доступ до цієї папки та рівень їх доступу:

```
@model IEnumerable<GALA.Models.ADGroups>
@{ViewBag.Title = "User Access List"; }
<body id="useraccess">
  <div>
    <table class="table table-striped table-bordered">
      <caption>
        <h2>
          Username: @if (@Model.Count() > 0)
```

```
        {@Model.First().comment}
          </h2><br />
        </caption>
        <tr>
          <th>Object Name</th>
          <th>Object Type</th>
          <th>Container</th>
          <th>Location</th>
        </tr>
        @foreach (var el in Model)
        {
          <tr>
            @{
              switch (el.objectType)
              {
                    <td><a
href="~/ADGroupsListing/Groups_Listing?folder=@el">@el.objectName</a></td>
                    <td>@el.objectType</td>
                    <td>@el.container</td>
                    <td>@el.location</td>
                    }
                    }
</tr>
              }
      </table>
      <br />
  </div>
  <a href="@Url.Action("Index")"
    class="btn btn-default">
```

```
      &laquo; Back
   </a>
   <footer>
      <p align="right">Total elements: @Model.Count().</p>
   </footer>
</body>
```

Таким чином, в цьому застосунку використовуються всі переваги моделі MVC. Дані обробляються на стороні сервера (Model), передаються вже відібрані дані до контролеру (Controller), а потім View формує з них зручну веб-сторінку, де дані містяться в текстовому режимі. Отже, користувачевs передаються лише текстові дані, без службової інформації, яка залишається на стороні сервера, де і виконується.

### 3.3. Приклад застосування програмного компоненту

Цей програмний компонент пройшов тестову фазу на підприємстві. Нижче наведений огляд сторінок, доступних в застосунку та зображення його використання.

За замовчанням ***головною сторінкою*** є сторінка «Заблоковані користувачі». Однак сторінку можна налаштувати для відображення підсумкової інформації про користувачів, групи, спільні папки, спільні поштові скриньки, списки розсилки, ярлики для запитів тощо, якщо потрібно. Головна сторінка містить посилання на інші сторінки на верхній панелі інструментів. Ці сторінки містять списки з детальною інформацією. Усі облікові записи користувачів, які вимкнено через закінчення контракту або інцидент, викреслюються. Кожна сторінка містить лічильник відповідних відображених елементів і час EET, коли він був згенерований.

На ***сторінці «Заблоковані користувачі»*** відображається список користувачів, які зараз заблоковані. Там можна побачити таку інформацію про користувача: 1) ПІБ користувача; 2) SAP ID; 3) Доменне ім’я користувача; 4) Час встановлення

пароля; 5) Термін дії пароля; 6) Час останнього входу з корпоративного комп'ютера; 7) Час блокування; 8) Наявність зареєстрованих пристроїв MDM.

Для користувачів із простроченими обліковими даними, новостворених користувачів і користувачів із скинутим паролем усі параметри відображаються як прострочені. Існує також кнопка, яка дозволяє надсилати електронний лист до IT відділу через Outlook. Він містить попередньо згенерований запит, адаптований для конкретних користувачів і випадку. Сторінка автоматично оновлюється кожні 5 хвилин.

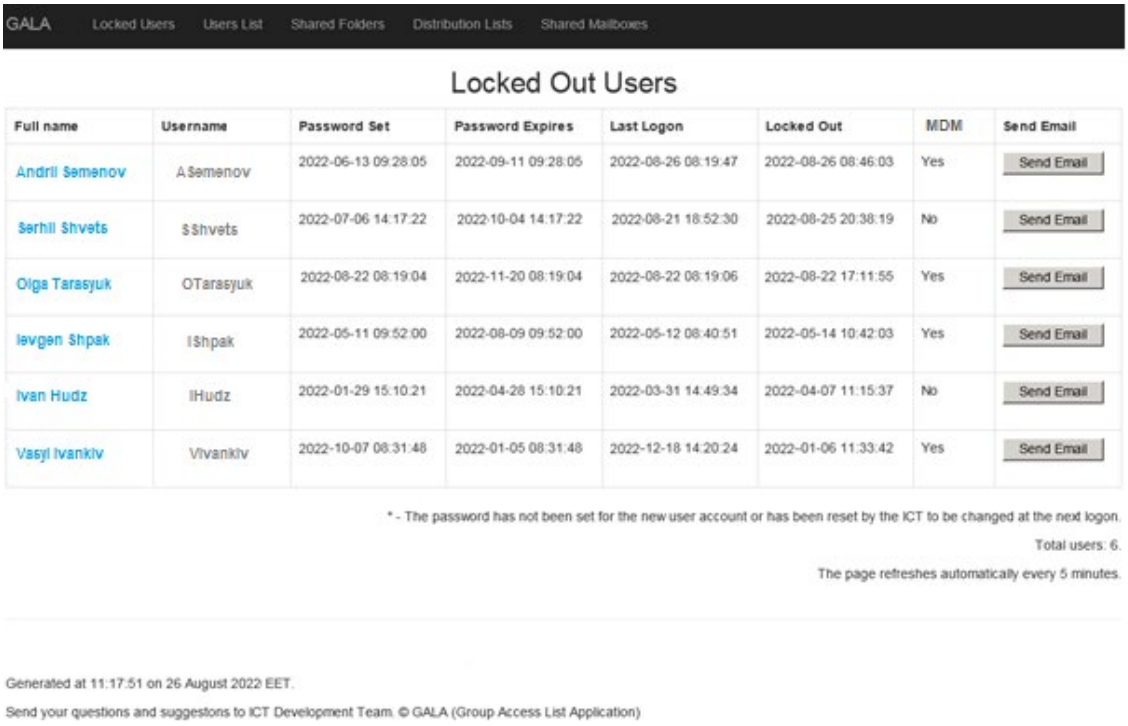


Рис. 3.4. Заблоковані користувачі

***Сторінка «Список користувачів»*** містить список усіх користувачів компанії, включаючи повні імена, імена користувачів та ідентифікатор SAP, які є гіперпосиланнями на відповідні списки доступу.

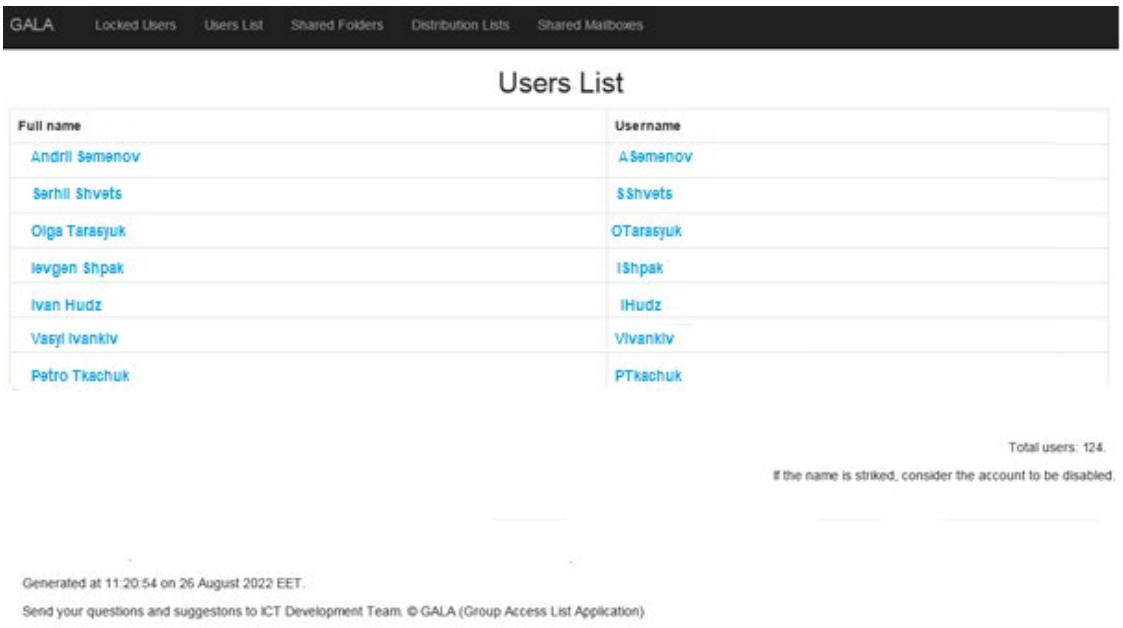


Рис. 3.5. Список користувачів

Користувач застосунку має знайти ім'я особи, чий список доступу потрібен, і натиснути на ім'я або ім'я користувача цієї особи. Потім відкриється сторінка з певним списком доступу з таблицею зі списками розсилки, спільних поштових скриньок і спільних папок із рівнем доступу. Інформація сортується за типами об’єктів, контейнерами AD і розташуванням.

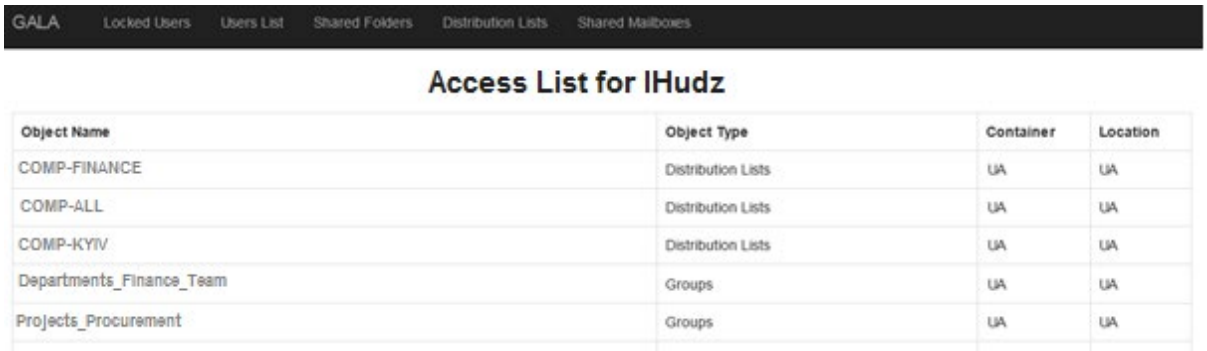


Рис. 3.6. Конкретний список доступу для користувача

***Сторінка «Спільні папки»*** містить список усіх спільних папок на дисках відділів (Departments:\) і проектів (Projects:\). На назву кожної папки є гіперпосилання. Користувач повинен знайти та клацнути на потрібному імені папки, і він(-а) перейде на сторінку з таблицею зі списком усіх користувачів (повні імена та

імена користувачів), які мають доступ до цієї папки, і їхній рівень доступу. Інформація відсортована за рівнем доступу, підрозділом, номером посади та назвою.

На ***сторінці «Списки розсилки»*** міститься список усіх списків розсилки. На назву кожного списку є гіперпосилання. Передбачається, що користувач знайде назву потрібного списку та клацне на ньому, і він перейде на сторінку з таблицею з переліком усіх користувачів (повні імена та імена користувачів), які мають доступ до цього списку. Інформація відсортована за підрозділом, номером позиції та назвою.

***Сторінка «Спільні поштові скриньки»*** містить список усіх спільних поштових скриньок. Ім'я кожної поштової скриньки є гіперпосиланням. Передбачається, що користувач знайде та клацне на потрібному імені спільної поштової скриньки, і він (вона) перейде на сторінку з таблицею з переліком усіх користувачів (повні імена та імена користувачів), які мають доступ до цієї поштової скриньки, і їхній рівень доступу. Інформація відсортована за рівнем доступу, підрозділом, номером посади та назвою.

# ВИСНОВКИ

Управління інформаційною безпекою є безперервним процесом. СУІБ разом із політикою, процедурами та відповідністю слід переглянути й оновити відповідно до останніх ринкових тенденцій і вимог. Цикл перегляду, аналізу недоліків і оновлення забезпечить довгострокову вигоду для організації шляхом захисту її ІТ-активів найефективнішим способом.

Розуміння важливості впровадження безпеки є дуже важливим. Якщо працівники не розуміють необхідності цього, вони можуть не брати участь у реалізації всім серцем, і це може призвести до провалу проекту або затримки досягнення результатів. Вище керівництво, будучи головним спонсором і мотиватором проекту, відіграє важливу роль у цьому питанні з самого початку.

Рішення безпеки має бути ретельно розроблено, щоб досягти економічної ефективності та повернення інвестицій (RoI), додаючи бізнес-цінності, окрім дотримання нормативних вимог, пам’ятаючи, що інвестиції в інформаційну безпеку є витратами на страхування, які захистять інформацію організації від втрати або знищення, уникаючи простоїв і таким чином підвищуючи продуктивність [40].

Файловий сервер є ключовим фактором для здійснення обміну даними, необхідного для розподілених систем. Файловий сервер є, мабуть, найбільш активно використовуваним ресурсом розподілених систем і як наслідок; його продуктивність є життєво важливою та критичною для перемоги системи. Стрімке зростання веб-вмісту та користувачів Інтернету призвело до збільшення уваги до двох основних проблем у цих системах: масштабованості та високої доступності мережевої файлової системи. Простий механізм розподілу навантаження для клієнта NFS для перемикання на сервер із невеликим навантаженням на основі кількості запитів RPC клієнта NFS протягом певного періоду часу, що робить ці системи більш ефективними та масштабованими. Методи реплікації використовуються для підвищення доступності розподілених систем.

Запропоноване в цій роботі рішення є більш безпечним і ефективним у тому сенсі, що користувач може переглядати певні параметри доступу до спільних ресурсів і характеристики облікових записів. Також вдосконалено механізм

балансування навантаження, де застосунок зберігає записи користувачів, підключених до сервера, а також не дає робити забагато запитів від одного користувача в певний момент часу.

Практичні надбання цієї роботи полягають в тому, що вона містить фактично працюючий застосунок, впроваджений в роботу в компанії та який приносить певну користь, як-от зниження навантаження на відділ підтримки користувачів і підвищення ефективності їх роботи. В майбутньому планується покращення цього застосунку шляхом збирання відгуків користувачів і надання йому нових потрібних функцій. Також, з огляду на те, що розглянутий застосунок є лише компонентом СУІБ, в планах також є повна його інтеграція в існуючу СУІБ підприємства.

# СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. ISO/IEC 27001:2013 [Електронне джерело] — Режим доступу: URL https://www.iso.org/isoiec-27001-information-security.html. Назва з екрану.

2. В.В. Цуркан. «Метод функціонального аналізування систем управління інформаційною безпекою», Кібербезпека: освіта, наука, техніка, №4(8), С. 192-201, 2020.

3. А.М. Гребенюк, Л.В. Рибальченко. Основи управління інформаційною безпекою. Дніпро: ДДУВС, 2020.

4. О.В. Коротун, Т.А. Вакалюк, В.В. Зубрицький, І.В. Гордієнко. «Теоретичні аспекти розробки системи управління навчанням», Таврійський науковий вісник, Серія: Технічні науки, №1, С. 36-46, 2020.

5. В.Д. Хох, Є.В. Мелешко, О.А. Смірнов. «Дослідження методів аудиту систем управління інформаційною безпекою», Системи управління, навігації та зв'язку, №1(41), С. 39-42, 2017.

6. T. Reenskaug. MVC XEROX PARC 1978-79 [Електронне джерело] — Режим доступу: URL http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html. Назва з екрану.

7. М.Ю. Комаров, С.Ф. Гончар. «Методика побудови системи управління інформаційною безпекою на об’єктах критичної інфраструктури», Моделювання та інформаційні технології, №81, С. 12-19, 2017.

8. О.О. Цвілій. «Безпека інформаційних технологій: сучасний стан стандартів ISO27k системи управління інформаційною безпекою», Телекомунікаційні та інформаційні технології, №2, С. 73-79, 2014.

9. М.Ю. Комаров, С.Ф. Гончар, А.В. Ониськова. «Нормативний аспект побудови та впровадження системи управління інформаційною безпекою на об’єктах критичної інфраструктури», Моделювання та інформаційні технології, №82, С. 40-48, 2018.

10. Ю.О. Русіна, В.Ю. Острякова. «Удосконалення системи управління інформаційною безпекою на підприємстві», Міжнародний науковий журнал Інтернаука, №14, С. 135-139, 2017.

11. В.В. Домарєв, Д.В. Домарєв, С.Б. Гордієнко. «Обгрунтування основних функцій системи управління інформаційною безпекою», Вісник Державного університету інформаційно-комунікаційних технологій, №10(2), С. 102-104, 2012.

12. В.В. Мохор, В.В Цуркан, О. Бакалинський, Я.Ю. Дорогий. «Метод концептуалізування системних досліджень систем управління інформаційною безпекою», Information Technology and Security, №8(1), С. 92-101б 2020.

13. М. Пацера. «Система управління інформаційною безпекою як важлива складова загальної системи управління банком», Вісник Національного банку України, №6, С. 48-49, 2015.

14. О.Є. Ананченко. «Питання формування організаційної структури системи управління інформаційною безпекою підприємства», Сучасний захист інформації, №1, С. 79-83, 2016.

15. А.В. Міщенко, В.В. Козловський. «Методологія наукового дослідження економічної складової системи управління інформаційною безпекою авіаційного транспортного комплексу», Системи управління, навігації та зв'язку, №1, С. 92-94, 2014.

16. В.В. Мохор, В.В. Цуркан, Я.Ю. Дорогий, Ю.М. Штифурак. «Структури архітектури систем управління інформаційною безпекою», Informatics & Mathematical Methods in Simulation, №9(4), 2019.

17. В.О. Темников. «Принципи побудови систем прийняття рішень в процесі управління інформаційною безпекою», Системи управління, навігації та зв’язку. Збірник наукових праць, №4(44), С. 119-121, 2017.

18. Е. Хемфрі. «Діяльність з кібер-безпеки. Рішення для бізнесу», Стандартизація. Сертифікація. Якість, №1, С. 16-18, 2013.

19. О.В. Файчук, М.А. Лещенко, Д.Е. Ткач. «Інформаційна безпека як індикатор стабільності страхового ринку України», Правове регулювання

фінансових послуг: національний, європейський, глобалізаційний виміри, №2022, С. 94, 2022.

20. Г.В. Хворост, Д.В. Стецюра. «Стандарт ISO/IEC 27001 та особливості впровадження вимог стандарту на вітчизняних підприємствах», Редакційна колегія збірника, №2014, С. 248, 2014.

21. О.А. Замула, В.І. Черниш. «Аналіз міжнародних стандартів в галузі оцінювання ризиків інформаційної безпеки», Системи обробки інформації, №2, С. 53-56, 2011.

22. Н.М. Волошина. «Поняття безпека інформації та інформаційна безпека в сучасному науковому просторі», Сучасні інформаційні технології у сфері безпеки та оборони, №2, С. 53-56, 2010.

23. Х. Засадна. «Стандарти управління інформаційною безпекою банку», Фінансовий простір, №3(3), С. 60-64, 2011.

24. Information Systems Audit and Control Association (ISACA). IT-Governance and Process Maturity [Електронне джерело] — Режим доступу: URL https://www.isaca.org/bookstore/it-governance-and-business-management/wgpm. Назва з екрану.

25. G.E. Krasner, S.T. Pope. «A cookbook for using the model-view controller user interface paradigm in Smalltalk-80», Journal of Object-Oriented Programming, №1(3), С. 26-49, 1998.

26. О.І. Пурський, Д.П. Мазоха. «Метод побудови мережі вітрин інтернет-магазинів на основі архітектури MVC», Бізнес Інформ, №10(477), С. 319-324, 2017.

27. І. Глабець. «Використання архітектурного шаблону MVC при проектуванні програмного забезпечення», Матеріали VI всеукраїнської студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання», №1, С. 60, 2013.

28. Г. Охріменко. «Основні принципи та проблеми впровадження електронного документообігу в організації», Наукові записки Національного

університету Острозька академія, серія: Культура і соціальні комунікації, №1, С. 300-307, 2009.

29. Г. Асєєв. «Архітектура корпоративного сховища даних», Вісник Книжкової палати, №10, С. 20-25, 2010.

30. Є.Б. Артамонов, О.О. Бєляков. «Електронні сховища даних із захищеним доступом», Наукоємні технології, №4, С. 402-405, 2013.

31. К.В. Лобузіна. «Системно-інтегрована технологія побудови сховища знань бібліотеки», Бібліотекознавство. Документознавство. Інформологія, №2, С. 51-57, 2013.

32. А. Стеценко. «Тенденції та напрями розвитку систем довгострокового зберігання інформації», Наукові праці Національної бібліотеки України ім. В.І. Вернадського, №28, С. 239-246, 2010.

33. С.В. Поперешняк, О.І. Недбайло. «Актуальна проблема електронного документообігу–нестача дискового простору», Вісник соціально-економічних досліджень, №1, С. 147-152, 2013.

34. О.В. Наумук, І. Наумук. «Функціональні особливості операційної системи Windows Server 2016», Technics and technology, №12, С. 66-68, 2018.

35. Р.В. Киричок, П.М. Складанний, В.Л. Бурячок, Г.М. Гулак, В.А. Козачок. «Проблеми забезпечення контролю захищеності корпоративних мереж та шляхи їх вирішення», Наукові записки Українського науково-дослідного інституту зв'язку, №3, С. 48-61, 2016.

36. В.М. Струков, В.В. Гуділін. «Захист від атак підвищення привілеїв в корпоративних інформаційних системах», Протидія кіберзлочинності та торгівлі людьми: збірник матеріалів, С. 79-82, 2021.

37. В. Ситніченко, Г. Кісельова, Є. Стоякін. «Формування інформаційної безпеки на основі стандарту ISO/IEC 27001: 2005», Стандартизація. Сертифікація. Якість, №2, С. 50-56б 2010.

38. Ю. Якименко, Т. Мужанова, С. Легомінова. «Системний аналіз технічних систем забезпечення інформаційної безпеки підприємств від компанії Fireeye», Кібербезпека: освіта, наука, техніка, №4(12), С. 36-50, 2021.

39. Р.В. Киричок, П.М. Складанний, В.Л. Бурячок, Г.М. Гулак, В.А. Козачок. «Проблеми забезпечення контролю захищеності корпоративних мереж та шляхи їх вирішення», Наукові записки Українського науково-дослідного інституту зв'язку, №3, С. 48-61, 2016.

40. С.О. Спасітєлєва, В.Л. Бурячок. «Комплексний захист гетерогенних корпоративних сховищ даних», Сучасний захист інформації, №1, С. 58-65, 2017.

41. Д.С. Замятін, О.С. Кебкало, Л.І. Лукашевська, А.Ю. Михайлюк, О.С. Михайлюк, А.В. Петрашенко, В.П. Тарасенко. «Організація програмного забезпечення корпоративного документо-орієнтованого сховища даних освітньої установи», Інформаційні технології та комп'ютерна інженерія, №3, С. 24-33, 2012.

42. М.Б. Вітер. «Технологія побудови оптимальної моделі сховища даних у державних органах», Науково-технічна інформація, №1, С. 35, 2014.

43. А.О. Лященко. «Використання систем дедуплікації даних», Технологічний аудит і резерви підприємства, №3(1(11)), С. 32-35, 2013.

44. С.С. Петровський. «Побудова інформаційного середовища ВНЗ», Вісник Хмельницького національного університету. Технічні науки, №2, С. 199-201, 2019.

45. Ю.В. Борсуковський, В.Ю. Борсуковська. «Рекомендації по категоріюванню інформації з обмеженим доступом», Сучасний захист інформації, №4, С. 9-17, 2017.

46. О.А. Стенін. «Розробка фізичних і логічних метрик для задачі багатокритеріальної оптимізації інформаційного навантаження при структуризації корпоративного центру даних», Адаптивні системи автоматичного управління, №2(15), С. 97-102, 2009.

47. О.П. Цвид-Гром, Т.І. Скляренко. «Упровадження ECM-системи як інтегрованої платформи під час роботи з корпоративним контентом», Соціальні комунікації: теорія і практика», №2016, С. 104, 2016.

48. К.Г. Сердюков. «Розроблення архітектури розподілу корпоративного контролю в інтегрованому акціонерному товаристві», Економічний вісник Запорізької державної інженерної академії, №5(2), С. 35-40, 2017.

49. І.А. Гораш, Т.В. Січко. «Аналіз популярних корпоративних інформаційних систем», Комп’ютерні технології обробки даних, №2020, С. 26-30, 2020.

50. О.В. Коваль. «Узагальнена архітектура аналітичної складової корпоративних інформаційно-аналітичних систем», Реєстрація, зберігання і обробка даних, №13(2), С. 53-73, 2011.

## ДОДАТОК А

Архітектура робочого середовища застосунка (виконано в PlantUML, онлайн доступ за посиланням https://bit.ly/3LlPeAn)

```
@startuml
package "Active Directory" {
  [Резервний контролер домену] - [Головний контролер домену]
} package "Файлові сервіси" {
  [Файловий сервер 1] - [Файловий сервер 2]
}
package "Сервіси SIEM" {
  [Сервер SIEM 1] - [Сервер SIEM 2]
}
package "Веб-сервіси" {
  [Веб-сервер] - [Застосунок]
}
package "Сервіси БД" {
  [БД сервер 1] - [БД сервер 2]
}
node "IT-аудитор" {
  [Веб-браузер]
}
[Файловий сервер 1] ---> [Головний контролер домену]
[Файловий сервер 2] ---> [Головний контролер домену]
[Файловий сервер 1] ---> [Резервний контролер домену]
[Файловий сервер 2] ---> [Резервний контролер домену]
[Файловий сервер 1] ----> [Сервер SIEM 1]
[Файловий сервер 1] ----> [Сервер SIEM 2]
[Файловий сервер 2] ----> [Сервер SIEM 1]
[Файловий сервер 2] ----> [Сервер SIEM 2]
```

```
[Застосунок] ---> [Головний контролер домену]
[Застосунок] ---> [Резервний контролер домену]
[Застосунок] ----> [БД сервер 1]
[Застосунок] ----> [БД сервер 2]
[Веб-браузер] --> [Застосунок]
@enduml
```

## ДОДАТОК Б

Архітектура застосунка (виконано в PlantUML, онлайн доступ за посиланням https://bit.ly/3BeCbw2)

```
@startuml
Групи <|-- Користувачі
Дозволи <|-- Групи
Відділи <|-- Групи
Групи <|-- Папки
class Користувачі{
  -Username: String
  -DisplayName: String
  -Title: String
  -UserID: int
  -Enabled: bool
  -GroupID: int
  +Список(Групи) GetUserGroups(UserID)
  +Список(Папки) GetUserFolders(UserID)
  +Перелік(Дозволи) GetUserPermissions(UserID)
}
class Відділи{
  -DepartmentName: String
  -DepartmentID: int
  +Список(Групи) GetDepartmentGroups(DepartmentID)
}
class Групи{
  -GroupName: String
  -GroupID: int
  -DepartmentID: int
  -FolderId: int
```

```
  -Permissions: Permissions
  +Список(Папки) GetGroupFolders(GroupID)
  +Відділи GetGroupDepartment(GroupID)
  +Перелік(Дозволи) GetGroupPermissions(GroupID)
}
class Папки{
  -FolderName: String
  -FolderID: int
  -FolderPath: int
  +Список(Папки) GetFolderGroups(FolderID)
}
enum Дозволи {
  FULL_CONTROL
  MODIFY
  READ_AND_EXECUTE
  READ
  WRITE
}
@enduml
```