НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ІНФОРМАЦІЙНА СИСТЕМА КОРПОРАТИВНОГО ЗБЕРІГАННЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ»

Виконав: здобувач 2 курсу
Рівень магістр
Галузь знань 12 «Інформаційні технології»,
спеціальність 122 «Комп’ютерні науки»
Торколат Володимир Андрійович

Керівник: к.т.н., доцент
Чикунов Павло Олександрович

Рецензент: к.пед.н., доцент
Лобода Юлія Геннадіївна

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет **кібербезпеки та інформаційних технологій**
Кафедра **інформаційних технологій**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **122 Комп'ютерні науки**
Назва освітньої програми: **Комп'ютерні науки**

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних технологій
доцент Н.І. Логінова
_____ «____ »_____ 2025 року

З А В Д А Н Н Я
на кваліфікаційну (магістерську) роботу
здобувачу Торколат Володимиру Андрійовичу

1. Тема роботи: «Інформаційна система корпоративного зберігання документів і файлів»
Керівник роботи: к.т.н, доцент Чикунов Павло Олександрович
затверджені наказом НУ ОЮА від «10» жовтня 2025 року № 3182-06
2. Строк подання здобувачем роботи «25» листопада 2025 рік
3. Дата видачі завдання «02» червня 2025 рік

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Огляд предметної області та існуючих рішень.	10.09.2025	
2	Формування вимог до інформаційної системи.	01.10.2025	
3	Проєктування концептуальної моделі системи.	15.10.2025	
4	Моделювання структури даних.	01.11.2025	
5	UML-моделювання функціональних і поведінкових аспектів.	05.11.2025	
6	Специфікація та верифікація моделі системи.	10.11.2025	
7	Оформлення пояснювальної записки.	17.11.2025	
8	Подача електронного варіанта роботи для перевірки на плагіат.	20.11.2025	

Здобувач _____ Торколат В.А._____
(підпис) (прізвище та ініціали)

Керівник роботи _____Чикунов П.О._____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інформаційна система корпоративного зберігання електронних документів», виконана на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 122 Комп'ютерні науки, освітньою програмою «Комп'ютерні науки», містить 18 рисунків, 15 таблиць, 2 додатки, 35 літературних джерел за переліком посилань. Робота викладена на 63 сторінках загального тексту, з них 49 сторінок основного змісту.

Метою роботи є розроблення концептуальної моделі інформаційної системи (ІС) корпоративного зберігання електронних документів, яка забезпечує впорядковане керування документами, структуризацію даних, контроль версій та інформаційної безпеки на основі сучасних методів аналізу й моделювання.

Об'єктом дослідження є процеси організації, зберігання та управління електронними документами в корпоративному середовищі.

Предметом дослідження є методи, моделі та засоби концептуального й логічного проєктування інформаційних систем корпоративного зберігання документів, орієнтованих на забезпечення доступності, цілісності та конфіденційності даних.

Методи дослідження ґрунтуються на використанні системного аналізу, UML-моделювання, структурного та логічного моделювання даних, аналізу вимог, трасування функціональних залежностей та оцінювання узгодженості архітектурних рішень.

У роботі вирішено науково-практичне завдання щодо розроблення структурно-функціональної, архітектурної та даних моделей корпоративної інформаційної системи, а також виконано верифікацію моделі відповідно до вимог безпеки, доступності та функціональності, що забезпечує її готовність до програмної реалізації.

Ключові слова: комп'ютерні науки, корпоративні системи, зберігання електронних документів, інформаційна система, моделювання, UML, логічна модель даних, доступність, автентифікація, контроль версій, аудит.

ABSTRACT

The qualification thesis entitled "Corporate Electronic Document Storage Information System", submitted for the second (master's) level of higher education in the field of study 122 Computer Science under the educational program Computer Science, contains 18 figures, 15 tables, 2 appendices, and 35 bibliographic sources. The thesis comprises 63 pages of total volume, including 49 pages of the main content.

The aim of the work is to develop a conceptual model of an information system for corporate electronic document storage that ensures structured document management, data organization, version control, and information security based on modern methods of system analysis and modeling.

The object of the study is the processes of organization, storage, and management of electronic documents in a corporate environment.

The subject of the study is the methods, models, and tools for conceptual and logical design of information systems for corporate document storage, focused on ensuring data availability, integrity, and confidentiality.

The research methodology is based on the application of system analysis, UML modeling, structural and logical data modeling, requirements analysis, functional dependency tracing, and architectural consistency assessment.

The thesis presents the scientific and practical solution of developing structural-functional, architectural, and data models of a corporate information system, as well as performing a model verification in accordance with security, availability and functional requirements, ensuring its readiness for further software implementation.

Keywords: Computer Science, corporate systems, electronic document storage, information system, modeling, UML, logical data model, availability, authentication, version control, audit.

ЗМІСТ

ВСТУП

Ефективне управління електронними інформаційними ресурсами є ключовим чинником стабільного функціонування та розвитку сучасних організацій. Постійне зростання обсягів електронних документів, вимоги до захисту конфіденційної інформації, необхідність збереження історії змін і забезпечення контролю доступу зумовлюють потребу у впровадженні корпоративних систем зберігання електронних документів. Такі системи спрямовані на оптимізацію документообігу, підвищення продуктивності співробітників, прозорість управлінських процесів та відповідність чинним нормативним вимогам.

Попри широке розповсюдження відомих систем корпоративного зберігання документів, зокрема Microsoft SharePoint, Alfresco, OpenText та Nuxeo, їхнє впровадження часто ускладнюється складністю адаптації до специфіки організаційних процесів, інтеграції з існуючими IT-інфраструктурами та обмеженнями у налаштуванні політик доступу відповідно до вимог інформаційної безпеки. Це зумовлює потребу у здійсненні системного аналізу предметної області та побудові моделі корпоративної інформаційної системи, здатної забезпечити масштабованість, цілісність, безперервний контроль доступу та зручність керування електронними документами.

Метою кваліфікаційної роботи є розроблення концептуальної моделі інформаційної системи корпоративного зберігання електронних документів, яка забезпечує впорядковане зберігання документів, структуризацію метаданих, підтримку версійності та високий рівень інформаційної безпеки на основі методів системного аналізу та сучасних підходів до проєктування інформаційних систем.

Об’єктом дослідження є процеси організації, зберігання та управління електронними документами в корпоративному середовищі.

Предметом дослідження є методи, моделі та засоби концептуального й логічного моделювання інформаційних систем корпоративного зберігання документів, орієнтованих на забезпечення доступності, конфіденційності та

цілісності даних.

Методи дослідження ґрунтуються на застосуванні системного аналізу, UML-моделювання, логічного та структурного моделювання даних, аналізу архітектурних підходів та механізмів забезпечення інформаційної безпеки на всіх етапах життєвого циклу даних.

Наукова новизна роботи полягає у формуванні цілісної структурно-функціональної моделі інформаційної системи корпоративного зберігання електронних документів, що поєднує модульний підхід до побудови архітектури, розширювану рольову модель доступу (RBAC) та засоби забезпечення аудиту й контролю версій документів.

Практичне значення результатів полягає у можливості використання розробленої моделі як концептуальної основи для розроблення або адаптації корпоративних систем електронного документообігу в державному та приватному секторах.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз предметної області корпоративного документообігу та ідентифікувати властиві їй проблеми;

– визначити функціональні й нефункціональні вимоги до інформаційної системи;

– спроєктувати логічну модель даних та рольову модель доступу користувачів;

– розробити UML-діаграми, що описують архітектуру та ключові сценарії взаємодії користувача із системою;

– специфікацію та виконати верифікацію моделі на відповідність вимогам доступності, безпеки та структурної узгодженості.

Апробація роботи виконана під участі у роботі у IX міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни», (м. Харків, 26 листопада 2025 р.) з доповіддю на тему «Проєктування інформаційної системи зберігання електронних документів».

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд наявних рішень щодо інформаційної системи

Сфера корпоративного зберігання даних розвивається швидкими темпами, і на сучасному ринку пропонується велика кількість рішень від провідних світових компаній. Хмарні сервіси є найпопулярнішими завдяки простоті використання, можливості швидкого розгортання та мінімальним витратам на інфраструктуру [1]. Водночас у процесі вибору постачальника виникає проблема: жодне з рішень не може повністю задовольнити всі вимоги корпоративного користувача.

Для аналізу були розглянуті найбільш відомі платформи: Google Drive [2], Dropbox [3], OneDrive [4]. У табл. 1.1 наведено їх порівняння за ключовими критеріями.

Таблиця 1.1 – Аналіз популярних хмарних сховищ даних

Критерій	**Google Drive**	**Dropbox**	**OneDrive**
Організована файлова система	Так	Так	Так
Відсутність зломів баз даних користувачів	Ні	Ні	Окремі інциденти зі зломами акаунтів
Забезпечення безперебійної роботи серверів	Часткові обмеження (санкції, геополітичні фактори)	Обмеження в окремих країнах та регіонах	Обмеження в окремих країнах та регіонах
Рольова файлова система (гнучкі права доступу)	Так	Ні	Ні
Використання незалежно від екосистеми постачальника	Ні	Ні	Ні
Доступність для людей з інвалідністю	Так	Ні	Так

Як показує аналіз, усі платформи забезпечують базову організацію файлової системи та високий рівень доступності для широкого кола користувачів. Однак

вони мають і суттєві обмеження. Серед головних проблем – відсутність повноцінної підтримки сторонніх API, складність інтеграції зі сторонніми сервісами та недостатньо розвинуті механізми рольового доступу.

Ще одним обмежувальним чинником для корпоративного середовища є прив’язка до екосистеми постачальника. Більшість рішень передбачає роботу лише у зв’язці з іншими сервісами компанії, що зменшує гнучкість використання.

На основі проведеного огляду можна зробити висновок, що існуючі хмарні рішення забезпечують зручність і швидкість доступу до даних, проте не задовольняють усіх специфічних вимог корпоративних користувачів [5]. Це стосується безпеки, можливостей кастомізації та масштабованості під потреби організації. Доцільною є розробка адаптивної інформаційної системи (ІС) корпоративного зберігання документів і файлів.

Корпоративні системи зберігання документів (Enterprise Document Management Systems, EDMS) призначені для централізованого зберігання, управління та контролю за документами та файлами у межах організації. Вони дозволяють забезпечити швидкий доступ до інформації, контроль версій документів, аудиторський облік і захист від несанкціонованого доступу [9].

Серед найбільш відомих рішень варто виділити Microsoft SharePoint, Box, Alfresco, OpenText, Nuxeo та SAP Document Management. SharePoint інтегрується з іншими продуктами Microsoft, що забезпечує зручність роботи з документами, спільну редакцію та автоматизацію бізнес-процесів [10]. Box пропонує хмарну платформу з акцентом на безпеку та масштабованість, підтримує інтеграцію зі сторонніми сервісами та забезпечує рольову модель доступу [11]. Alfresco є відкритим корпоративним рішенням з потужними інструментами для керування контентом і автоматизації процесів, що дозволяє організації налаштовувати систему під власні потреби [12]. OpenText надає комплексні можливості для управління контентом та великими обсягами корпоративних даних, включаючи аналітику та інтеграцію з ERP і CRM-системами [13]. Nuxeo є платформою для управління контентом та цифровими активами, яка підтримує масштабовану хмарну інфраструктуру, гнучке налаштування робочих процесів та інтеграцію зі

сторонніми додатками [14]. SAP Document Management дозволяє централізовано організувати документи в контексті ERP-системи SAP, забезпечує контроль версій, безпеку та аудит змін [15]. Ці два продукти виконують схожі функції, але SAP більше орієнтований на інтеграцію з ERP, тоді як Nuxeo є універсальною платформою для управління різними видами контенту незалежно від ERP.

Рисунок 1.1 – Найбільш відомі корпоративні системи зберігання документів

Таблиця 1.2 – Порівняння корпоративних систем зберігання документів

Характеристика	SharePoint	Box	Alfresco	OpenText	Nuxeo	SAP DM
Тип системи	Інтегро-вана платформа	Хмарне сховище	Відкрите корпора-тивне рішення	Корпоративна платформа	Платформа управління контентом	ERP-орієнто-ване DMS
Хмарна підтримка	Часткова	Повна	Часткова	Часткова	Повна	Часткова
Автоматизація бізнес-процесів	Так, Power Automate	Частково	Так, BPM модулі	Так	Так, робочі процеси	Так, SAP ERP
Призначення	Управ-ління докумен-тами, спільна робота, бізнес-процеси	Хмарне зберігання, безпека, інтеграції	Управління контентом, автоматизація	Комплексне управління корпора-тивним контентом	Управління цифровими активами та контентом	Управ-ління докумен-тами в ERP-контексті
Безпека	Висока, Microsoft Security	Висока, багато-рівнева	Висока, модулі контролю доступу	Висока, корпоративний рівень	Висока, шифрування та контроль доступу	Висока, інтеграція з SAP Security

Аналіз наявних рішень показує, що вони мають спільні переваги: централізоване зберігання документів, контроль версій, розмежування доступу та інтеграцію з корпоративними інструментами. Водночас у більшості систем спостерігаються обмеження щодо кастомізації та інтеграції з нестандартними

додатками. Крім того, хмарні рішення часто прив’язані до конкретної платформи, що може обмежувати автономність організації у виборі інфраструктури.

На основі проведеного огляду можна зробити висновок, що сучасні корпоративні системи зберігання документів ефективно вирішують базові завдання організації даних і управління документами. Однак для повної відповідності специфічним вимогам підприємства до безпеки, масштабованості та інтеграції доцільним є розробка власної системи, адаптованої під конкретні бізнес-процеси.

Наступним важливим етапом є вибір моделі бази даних, яка визначатиме способи організації, доступу та обробки інформації. Тому доцільним є огляд основних типів баз даних та їх характеристик. Сучасний ринок пропонує кілька основних підходів до зберігання даних: реляційні SQL-бази, нереляційні NoSQL-рішення та файлові сховища. Кожен із цих варіантів має власні особливості, які впливають на архітектуру та ефективність майбутньої системи.

SQL-бази даних (реляційні) ґрунтуються на табличній моделі та мові запитів SQL. Вони забезпечують високу структурованість, підтримку транзакцій, цілісність даних та відповідність принципам ACID [6]. Завдяки цим властивостям SQL-бази, такі як Microsoft SQL Server, MySQL або PostgreSQL, традиційно використовуються у корпоративних середовищах, де важлива точність і логічні зв’язки між сутностями. Недоліком реляційних систем є складність масштабування та нижча продуктивність при роботі з великими обсягами неструктурованих даних.

NoSQL-бази даних виникли як альтернатива реляційним системам для роботи з великими та динамічними обсягами інформації [7]. Вони не обмежуються табличною структурою та підтримують різні моделі: документоорієнтовану (MongoDB), колонкову (Cassandra), графову (Neo4j) або ключ-значення (Redis). Основні переваги NoSQL – гнучкість структури даних, висока швидкість обробки запитів та горизонтальна масштабованість. Водночас вони часто поступаються SQL-системам у питанні транзакційності та цілісності даних.

Файлові сховища є найпростішим способом організації даних, коли інформація зберігається у вигляді файлів у файловій системі [8]. Такий підхід може бути ефективним у невеликих системах або у випадках, коли не потрібні складні

запити чи транзакції. Проте зі зростанням обсягів даних файлові сховища втрачають ефективність, адже пошук та обробка інформації стають повільними, а забезпечення безпеки й багатокористувацького доступу ускладнюється.

Як видно з табл. 1.3, SQL-бази забезпечують високу структурованість даних, підтримку транзакцій та надійність, але менш гнучкі та складніше масштабуються. NoSQL-системи характеризуються високою гнучкістю, горизонтальною масштабованістю та ефективністю при роботі з великими обсягами неструктурованих даних. Файлові сховища прості у використанні та дозволяють зберігати документи та мультимедіа, проте їхня продуктивність та безпека обмежені, а масштабованість залежить від файлової системи.

Таблиця 1.3 – Порівняння типів баз даних

Характеристика	**SQL**	**NoSQL**	**Файлові сховища**
Структура даних	Таблична, чітко визначені схеми	Гнучка, документоорієнтована, колонкова, графова, ключ-значення	Файли та каталоги без жорсткої схеми
Транзакції	Підтримка ACID, гарантія цілісності даних	Обмежена підтримка транзакцій, часткова консистентність	Відсутня, транзакційність обмежена
Масштабованість	Вертикальна (додавання ресурсів сервера)	Горизонтальна (додавання вузлів у кластер)	Обмежена масштабованість, залежить від файлової системи
Продуктивність	Висока для структурованих запитів, аналітики	Висока для великих обсягів неструктурованих даних	Обмежена при великих обсягах і складних запитах
Гнучкість схеми	Низька, зміни структури складні	Висока, дозволяє працювати з різнорідними даними	Дуже висока, але без контролю та структури
Застосування	Бізнес-процеси, фінанси, CRM, ERP	Великі обсяги даних, аналітика, IoT, соціальні мережі	Зберігання документів, мультимедіа, архіви
Безпека та контроль доступу	Високий рівень, підтримка ролей та прав доступу	Частково реалізовано, залежить від конкретної системи	Мінімальний контроль, часто реалізується на рівні ОС

Вибір типу бази даних залежить від специфіки завдань корпоративної системи. Для критично важливих бізнес-процесів із чіткою структурою найдоцільнішими залишаються SQL-рішення. Якщо ж ідеться про великі обсяги неструктурованої інформації, варто розглядати NoSQL-технології. Файлові сховища, у свою чергу, можуть виконувати допоміжну роль, наприклад, для зберігання документів, мультимедійних файлів чи архівів.

1.2 Огляд технологій розробки інформаційної системи

Розробка корпоративних ІС потребує застосування ефективних архітектурних та програмних підходів, які забезпечують масштабованість, гнучкість, безпеку та простоту підтримки. До таких технологій належать архітектурні патерни, патерни доступу до даних, REST API та веб-сервісні підходи.

Архітектурні патерни визначають загальну організацію програмного забезпечення і взаємодію між його компонентами. Патерн MVC (Model-View-Controller) розділяє застосунок на три логічні рівні: модель, що відповідає за бізнес-логіку і дані; представлення (View), яке відповідає за інтерфейс користувача; та контролер, що обробляє запити користувача та керує взаємодією між моделлю та представленням [17]. Такий підхід забезпечує чітку інкапсуляцію відповідальності між компонентами, підвищує підтримуваність коду, дозволяє тестувати бізнес-логіку окремо від UI і спрощує масштабування застосунку. Наприклад, у .NET MAUI MVC дозволяє організувати мультиплатформену логіку для Windows, Android та iOS, зберігаючи єдину модель даних.

Інший сучасний архітектурний підхід – Microservices (Мікросервіси). Цей патерн передбачає розподілення системи на набір незалежних сервісів, кожен з яких відповідає за конкретну бізнес-функцію [18]. Кожен сервіс може бути розгорнутий автономно, масштабований окремо та оновлюватися без простою всієї системи. Технічно це реалізується через ізольовані контейнери Docker або Kubernetes, які забезпечують відокремлене середовище виконання. Мікросервіси дозволяють використовувати різні технології для окремих сервісів (наприклад, C#

для бізнес-логіки, Node.js для API-шарів, MS SQL Server для сховища даних). Недоліком є необхідність організації комунікації між сервісами, часто через REST API або gRPC, та забезпечення консистентності даних у розподіленому середовищі.

Рисунок 1.2 – Патерн MVC з позначеними напрямками взаємодії користувачів

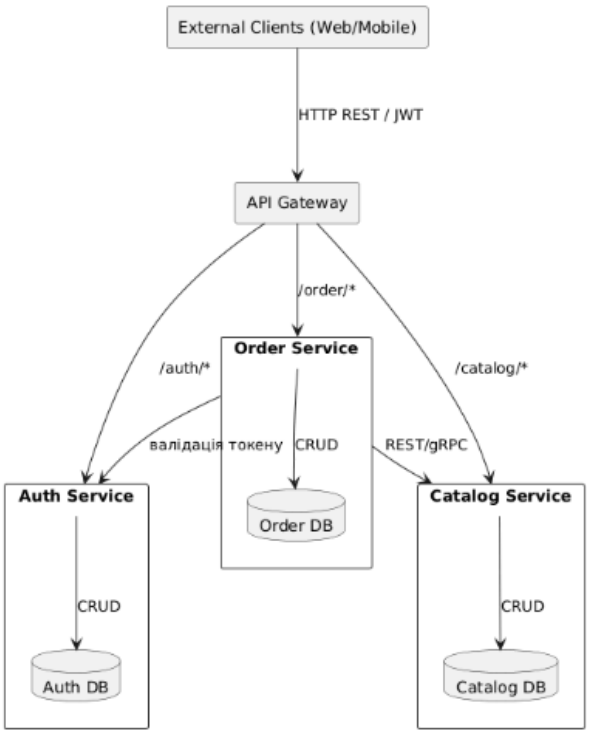

Рисунок 1.3 – Діаграма мікросервісів з базами даних чи контейнерами

Патерни доступу до даних визначають способи взаємодії бізнес-логіки з базами даних. Патерн Repository створює абстракцію над джерелом даних, ізолюючи бізнес-логіку від деталей роботи з базою та забезпечуючи єдиний інтерфейс для запитів і збереження даних [16]. Це особливо корисно при роботі з MS SQL Server, де Repository реалізує CRUD-операції, транзакції та оптимізацію запитів через LINQ. Патерн Observer дозволяє об’єктам підписуватися на зміни стану інших об’єктів, що корисно для реактивного оновлення UI та синхронізації станів у реальному часі, наприклад, при спільній роботі користувачів у корпоративному сховищі [19].

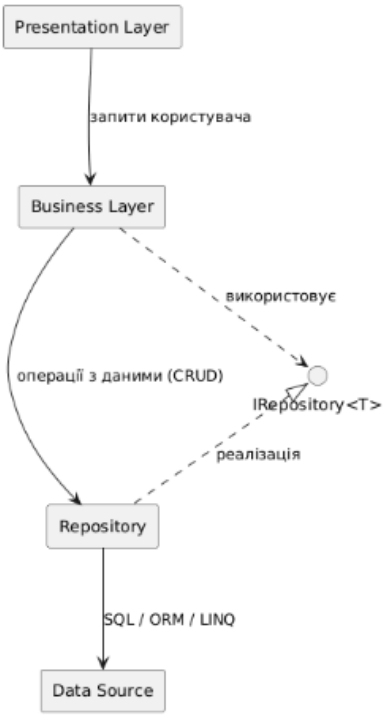

Рисунок 1.4 – Схема патерну Repository з шаром бізнес-логіки

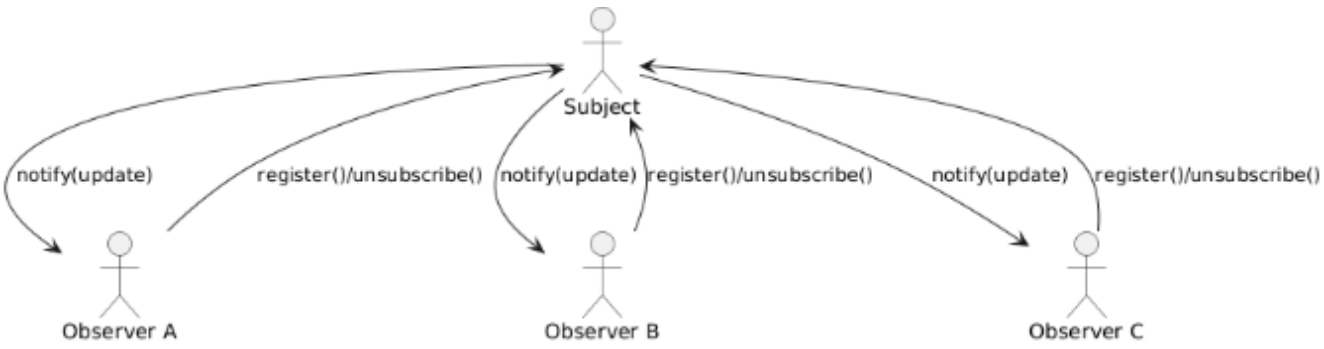

Рисунок 1.5 – Патерн Observer – об’єкт-постачальник та кілька підписників, які отримують оновлення

Сучасні корпоративні системи активно використовують REST API та веб-сервіси для інтеграції та обміну даними між компонентами та зовнішніми системами. REST (Representational State Transfer) визначає стандартизовані методи доступу до ресурсів через HTTP (GET, POST, PUT, DELETE), що забезпечує простоту, масштабованість та підтримку різних клієнтських платформ [20]. Технічно REST API дозволяє створювати маршрути для доступу до документів, користувачів та прав доступу, підтримує JSON та XML як формати обміну даними, а також інтегрується з системами автентифікації через OAuth 2.0 або JWT. Веб-сервіси SOAP використовують XML для структурованого обміну повідомленнями та забезпечують стандартизовану взаємодію з корпоративними системами, включаючи контроль доступу та шифрування даних на рівні повідомлень [21].

Застосування зазначених технологій у комплексі дозволяє створювати сучасні корпоративні інформаційні системи з чіткою структурою, високою масштабованістю, гнучким доступом до даних та можливістю інтеграції з іншими системами та сервісами. Використання патернів MVC і Repository забезпечує підтримуваність коду, Microservices та REST API – гнучкість і масштабованість, а Observer – інтерактивність та реактивність системи.

1.3 Огляд способів аутентифікації користувача

Логін/пароль залишається базовим методом аутентифікації, який використовується практично у всіх інформаційних системах. Суть полягає у введенні користувачем унікального ідентифікатора та секретного пароля, які перевіряються сервером. Перевагою такого підходу є простота реалізації та універсальність. Проте він має значні недоліки: паролі можуть бути легко підібрані або вкрадені, а користувачі часто застосовують прості або повторювані комбінації. Для підвищення безпеки застосовуються хешування паролів із використанням алгоритмів SHA-256 або bcrypt, додатково застосовують запобігання атак методом попередньо підготовлених таблиць. У корпоративних системах часто впроваджують політику складності паролів, обмеження на спроби входу та

регулярну зміну паролів [22].

OAuth 2.0 – це протокол авторизації, який дозволяє стороннім додаткам отримати доступ до ресурсів користувача без необхідності передавати пароль. Його основна перевага – можливість обмежити права доступу до конкретних ресурсів або дій, що забезпечує додатковий рівень безпеки. OAuth 2.0 використовує токени доступу (Access Token) та токени оновлення (Refresh Token) для управління сесіями. Токени можуть мати обмежений термін дії та права доступу (scopes), що дозволяє контролювати дії клієнтів на сервері. Поширені сценарії використання включають інтеграцію з хмарними сховищами, API сторонніх сервісів та мобільними додатками. Для реалізації протоколу необхідний Authorization Server та механізми захисту токенів, такі як HTTPS та JWT (JSON Web Token) [23-24].

Single Sign-On (SSO) дозволяє користувачеві здійснити один вхід і отримати доступ до багатьох незалежних систем без повторного введення облікових даних. Це підвищує зручність для користувачів та спрощує адміністрування доступу у великих організаціях. Технології SSO реалізуються через протоколи SAML або OpenID Connect, які забезпечують безпечну передачу атрибутів користувача між Identity Provider (IdP) та сервісами, що надають доступ. SSO широко застосовується у корпоративних середовищах із централізованим каталогом користувачів, наприклад, Active Directory або LDAP. Технічно важливо організувати надійне шифрування даних, контроль терміну дії сесій та аудит доступу [25-26].

Двофакторна аутентифікація (2FA) додає другий рівень перевірки користувача, поєднуючи щось, що він знає (пароль), з тим, що він має (смартфон, апаратний токен). Цей підхід значно підвищує безпеку, оскільки навіть при компрометації пароля зловмисник не зможе отримати доступ без другого фактора. Двофакторна аутентифікація застосовується через одноразові паролі (OTP), які генеруються у додатках типу Google Authenticator, через SMS або апаратні токени. Стандарти TOTP та HOTP дозволяють синхронізувати генерацію OTP між сервером та клієнтом, забезпечуючи захист від повторного використання пароля. Реалізація 2FA в корпоративних системах передбачає інтеграцію з базовими методами аутентифікації, аудит спроб входу та відновлення доступу [27].

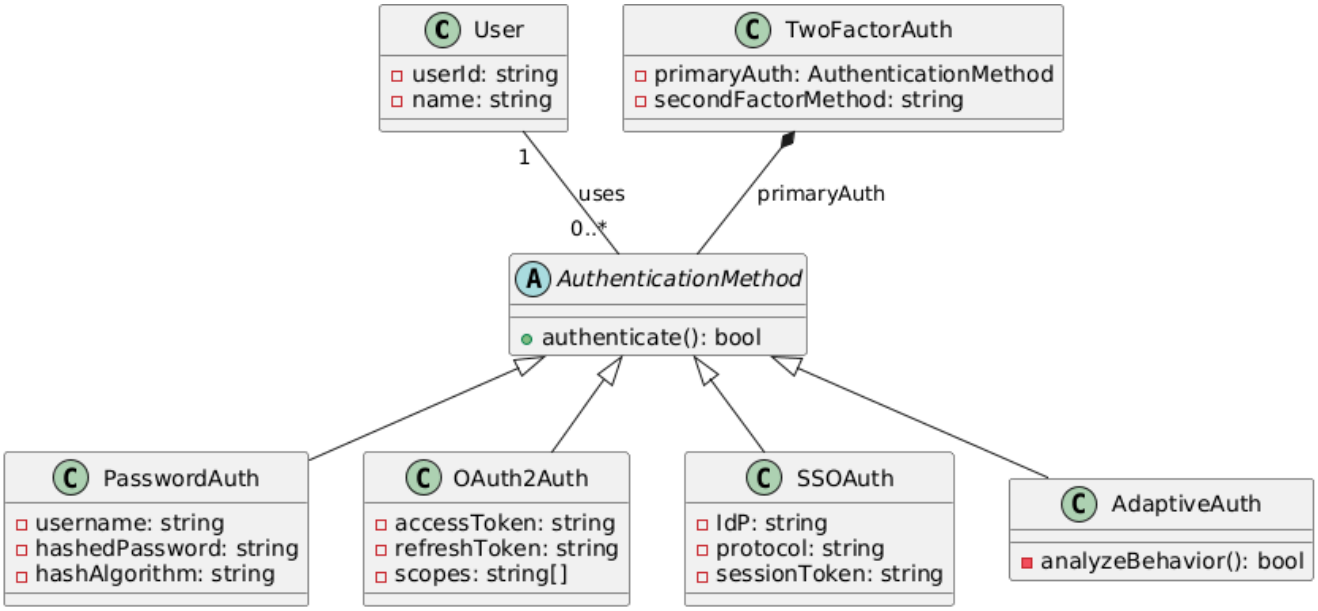

Рисунок 1.6 – UML-діаграма ієрархії класів для аутентифікації користувача

Застосування цих методів у комплексі дозволяє створювати багаторівневу систему безпеки корпоративних інформаційних систем. Наприклад, у системах зберігання документів користувач спочатку вводить логін і пароль, після чого система запитує другий фактор або перенаправляє на авторизацію через OAuth 2.0 або SSO. Це забезпечує баланс між зручністю використання та високим рівнем захисту даних. Крім того, сучасні системи забезпечують підтримку адаптивної аутентифікації, яка аналізує поведінку користувача та контекст (IP-адреса, пристрій, час входу) для підвищення безпеки та зниження ризику несанкціонованого доступу.

1.4 Огляд способів забезпечення доступності даних

Забезпечення високої доступності корпоративних інформаційних систем є ключовим чинником стабільної роботи сховищ документів та файлів. Доступність системи визначається здатністю обробляти запити користувачів у будь-який момент часу та гарантувати збереження даних навіть у разі відмови апаратних або програмних компонентів. У сучасних системах доступність розглядається як комплексний процес, який включає резервне копіювання, реплікацію,

балансування навантаження та відновлення після збоїв.

Резервне копіювання та реплікація є основою збереження даних. Резервне копіювання дозволяє створювати копії даних для відновлення у разі їх втрати або пошкодження. Реплікація забезпечує постійне дублювання даних на різних серверах або географічно віддалених локаціях, що дає змогу зберігати цілісність інформації навіть у разі виходу з ладу одного із ресурсів. Наприклад, корпоративні сховища часто використовують синхронну реплікацію для критичних даних, де будь-яка зміна на основному сервері одразу дублюється на резервний, а асинхронну реплікацію застосовують для менш критичних об’єктів, що дозволяє оптимізувати продуктивність [28].

Балансування навантаження дозволяє рівномірно розподілити запити користувачів між декількома серверами або вузлами, що не лише запобігає перевантаженню окремих компонентів, але й забезпечує масштабованість системи. У корпоративних середовищах програмне балансування на основі Nginx або HAProxy дозволяє автоматично перенаправляти трафік на доступні ресурси та зменшувати час відповіді системи [29]. Таке рішення особливо важливе у періоди пікового навантаження або при проведенні планових технічних робіт, коли частина серверів може бути тимчасово недоступною.

Відновлення після збоїв, або Disaster Recovery (DR), забезпечує повернення системи до нормальної роботи після серйозних аварій або катастроф. План DR передбачає визначення критичних компонентів, організацію резервного копіювання, автоматичне перемикання на резервні ресурси та періодичне тестування процедур відновлення. Рішення на кшталт CloudEndure або IBM Global Mirror дозволяють реалізувати відновлення як синхронне, так і асинхронне, що гарантує мінімальний час простою та цілісність даних [30].

Інтеграція систем з хмарними сервісами, такими як Microsoft Azure або AWS, забезпечує додаткові можливості для підвищення доступності. Використання хмарних рішень дозволяє розміщувати дані у різних географічних локаціях, автоматично масштабувати ресурси під навантаження та оптимізувати витрати на інфраструктуру. Важливим аспектом є визначення параметрів RPO (Recovery Point

Objective) та RTO (Recovery Time Objective), які дозволяють планувати максимально допустимий час втрати даних та час відновлення системи після відмови [31].

У комплексі ці методи дозволяють досягти високої доступності корпоративного сховища документів, забезпечити безперервність бізнес-процесів та захист даних від випадкових втрат, технічних збоїв та катастрофічних подій. Сучасні корпоративні рішення об’єднують резервне копіювання, реплікацію, балансування навантаження та інтеграцію з хмарними платформами, що забезпечує надійний та безпечний доступ користувачів до даних у будь-який момент часу [28].

1.5 Огляд видів кібератак та вразливостей

Сучасні корпоративні інформаційні системи, зокрема сховища документів і файлів, постійно піддаються різноманітним кібератакам. Загрози можуть походити як із зовнішнього середовища, так і зсередини організації, і їхнє використання зловмисниками здатне призвести до втрати даних, порушення доступності системи або компрометації конфіденційної інформації. Розуміння типів атак і вразливостей є фундаментальним для розробки надійних систем безпеки та планування заходів захисту.

Однією з найпоширеніших загроз є атаки через аутентифікацію. Зловмисники можуть отримати доступ до системи, використовуючи викрадені облікові дані, підбираючи паролі або використовуючи уразливості в механізмах авторизації. Для корпоративних сховищ документів особливо важливо застосовувати багатофакторну аутентифікацію (MFA), яка передбачає комбінацію логіну, пароля та додаткових факторів, таких як SMS-код або токен. Крім того, інтеграція протоколів OAuth 2.0 та SSO (Single Sign-On) дозволяє підвищити безпеку доступу, забезпечуючи централізоване управління обліковими записами та обмеження прав доступу [32].

Шкідливе програмне забезпечення (malware) є ще однією значною загрозою для корпоративних сховищ. До нього належать віруси, трояни, програми-вимагачі

(ransomware) та руткіти, здатні шифрувати, видаляти або викрадати дані. Наслідки таких атак можуть бути катастрофічними: блокування доступу до критично важливої інформації, порушення бізнес-процесів та фінансові збитки. Для захисту систем використовують комплекс антивірусних рішень, постійний моніторинг трафіку та поведінки користувачів, а також регулярні оновлення програмного забезпечення. Важливо зазначити, що ефективна стратегія безпеки передбачає резервне копіювання та реплікацію даних, що дозволяє швидко відновити систему після атаки [33].

Атаки на веб-додатки та API є особливо небезпечними для корпоративних сховищ, які забезпечують доступ до даних через веб-інтерфейси або REST API. Найпоширенішими є SQL-ін’єкції, Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF). Вразливості у коді веб-додатків дозволяють зловмисникам виконувати несанкціоновані операції, отримувати конфіденційну інформацію або порушувати цілісність даних. Захист передбачає регулярне тестування на проникнення, впровадження механізмів валідації та санітизації даних, а також обмеження прав доступу для користувачів і сервісів [34].

Соціальна інженерія є методом атак, спрямованих на вплив на користувачів з метою отримання конфіденційної інформації. Фішинг, телефонні шахрайства, підроблені електронні листи та шкідливі посилання – усі ці методи експлуатують психологічні слабкості користувачів. Навчання персоналу, регулярне тестування обізнаності та впровадження політик безпеки значно знижують ризики таких атак.

Вразливості корпоративних систем можуть бути обумовлені як технічними недоліками, так і організаційними помилками. Застаріле програмне забезпечення, неправильні налаштування прав доступу та слабкі політики безпеки створюють передумови для успішних атак. Регулярні оновлення систем, аудит безпеки та тестування на проникнення дозволяють своєчасно виявляти слабкі місця та мінімізувати ризики.

Комплексний підхід до безпеки включає інтеграцію технічних та організаційних заходів: багатофакторна аутентифікація, контроль доступу, моніторинг активності, резервне копіювання, шифрування даних та навчання

персоналу. Тільки застосування всіх цих заходів у комплексі забезпечує надійний захист корпоративного сховища документів від втрати даних, порушення доступності та компрометації інформації [34].

Таким чином, системний підхід до виявлення, оцінки та нейтралізації кіберзагроз є критично важливим для збереження цілісності, конфіденційності та доступності даних у сучасних корпоративних інформаційних системах.

1.6 SWOT-аналіз корпоративних систем зберігання

Для вибору оптимальної концепції побудови корпоративного сховища електронних документів необхідно враховувати сучасний стан і тенденції розвитку систем керування корпоративним контентом. Незважаючи на їх широке використання, кожна з них має власні переваги й обмеження, що впливають на можливість впровадження в конкретному організаційному середовищі. З цією метою був проведений SWOT-аналіз зазначених систем, який дозволяє оцінити сильні й слабкі сторони кожного рішення, а також потенційні можливості та загрози їх застосування в умовах корпоративного документообігу.

Таблиця 1.4 – Результати SWOT-аналізу корпоративних систем зберігання

Система	**S – Strengths (сильні сторони)**	**W – Weaknesses (слабкі сторони)**	**O – Opportunities (можливості)**	**T – Threats (загрози)**
1	**2**	**3**	**4**	**5**
Microsoft SharePoint	Тісна інтеграція з Microsoft 365; потужні засоби спільної роботи; масштабованість	Вартість ліцензій; складність налаштування; залежність від Microsoft	Розширення функцій через Power Platform; корпоративні інтеграції Azure	Залежність від інфраструктури MS; ризики збоїв у хмарі
Box	Орієнтація на корпоративний сектор; безпекові механізми; підтримка workflow	Дорожчий у порівнянні з іншими; обмежені локальні розгортання	Зростання попиту на хмарні ECM; інтеграції з AI-аналізом документів	Конкуренція зі SharePoint та Google Workspace
Alfresco	Відкрита платформа; гнучкість кастомізації; можливість локального розгортання	Вимагає високої кваліфікації персоналу; дорожчі версії Enterprise	Розширення екосистеми open-source; державний сектор	Розвиток комерційних конкурентів; складність оновлень

Продовження табл. 1.4

1	2	3	4	5
OpenText	Потужний ECM-функціонал; підтримка складних корпоративних процесів; сильна аналітика	Дуже висока вартість; складність імплементації	Великий корпоративний ринок; інтеграція з SAP	Ризики vendor-lock; потреба в IT-інфраструктурі високого рівня
Nuxeo	Гнучкість, модульність; сильний контент-менеджмент; API-орієнтованість	Менша популярність і підтримка; потребує значних налаштувань	Розвиток у напрямку headless-ECM; інтеграції через мікросервіси	Конкуренція з більш поширеними ECM-системами
SAP Document Management	Сильна інтеграція з SAP ERP; безпека та контроль бізнес-процесів; стабільність	Вартість і складність впровадження; залежність від SAP-екосистеми	Розширення у великих корпораціях; перехід до SAP Cloud	Ризики міграції при зміні ERP; висока ціна обслуговування

Системи корпоративного зберігання документів демонструють різний баланс між функціональною повнотою, вартістю володіння та гнучкістю впровадження. Крупні платформи (OpenText, SAP DM, SharePoint) ефективні для великих підприємств, але потребують значних ресурсів. Alfresco та Nuxeo лідирують у кастомізації та інтеграціях, проте мають вищі вимоги до технічного персоналу. Box виграє у напрямку хмарної безпеки, але менш придатний для глибокої внутрішньої інтеграції.

Це обґрунтовує доцільність розробки власної адаптивної моделі системи, що враховує специфіку корпоративного середовища та вимоги інформаційної безпеки.

1.7 Висновки за розділом

У першому розділі роботи проведено детальний огляд теоретичних основ корпоративних інформаційних систем зберігання документів і файлів, що включає аналіз наявних рішень, технологій розробки, методів аутентифікації користувачів, забезпечення доступності та виявлення кіберзагроз. Було встановлено, що сучасні сховища документів повинні поєднувати високий рівень безпеки, доступності та масштабованості, а також підтримувати інтеграцію з корпоративними сервісами.

Огляд наявних рішень показав, що на ринку представлені різноманітні корпоративні системи зберігання, такі як Microsoft SharePoint, Box, Alfresco, OpenText, Nuxeo та SAP Document Management, які відрізняються функціоналом, можливістю інтеграції, масштабованістю та рівнем безпеки. При виборі рішення необхідно враховувати можливість централізованого управління доступом, підтримку ролей користувачів, гнучкість налаштувань та відповідність вимогам.

Аналіз типів баз даних (SQL, NoSQL та файлові сховища) дозволив визначити, що вибір конкретної технології залежить від структури даних, вимог до швидкості обробки запитів, обсягу інформації та потреби у масштабуванні. Реляційні бази даних забезпечують надійну цілісність даних та гнучкі механізми запитів, тоді як NoSQL-системи краще підходять для роботи з великими обсягами неструктурованих даних та високою частотою змін.

Розглянуто архітектурні патерни розробки і принципи роботи REST API та веб-сервісів. Використання цих підходів дозволяє будувати гнучкі, масштабовані та підтримувані корпоративні системи, що здатні ефективно обробляти великі обсяги інформації та інтегруватися з іншими сервісами.

Огляд способів аутентифікації користувачів показав, що для забезпечення безпеки доступу до сховищ документів необхідно поєднувати кілька методів, включаючи логін/пароль, OAuth 2.0, SSO та двофакторну аутентифікацію. Такий підхід мінімізує ризики несанкціонованого доступу та підвищує рівень довіри.

Аналіз методів резервного копіювання, реплікації, балансування навантаження та відновлення після збоїв показав, що комплексне застосування цих заходів дозволяє гарантувати безперервність бізнес-процесів та захист даних від втрати або пошкодження.

Огляд видів кібератак та вразливостей дав змогу виділити ключові ризики для корпоративних сховищ документів, включаючи атаки через аутентифікацію, шкідливе програмне забезпечення, атаки на веб-додатки та API.

Таким чином, проведений огляд і аналіз теоретичних основ показав, що ефективне корпоративне сховище документів і файлів повинно поєднувати надійну безпеку, високу доступність, гнучку архітектуру та інтеграційні можливості.

2 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Постановка вимог до інформаційної системи

Проєктування інформаційної системи корпоративного зберігання електронних документів ґрунтується на результатах аналізу теоретичних засад та існуючих технологічних рішень. На цьому етапі визначаються функціональні особливості майбутньої системи, формуються вимоги до її архітектури, безпеки та користувацької взаємодії. Процес проєктування включає структуризацію бізнес-процесів документообігу, визначення ролей користувачів, розроблення логічної моделі даних, а також моделювання поведінки та взаємодії компонентів.

Важливою складовою є створення UML-діаграм, які формалізують функціональні сценарії та структуру системи за допомогою графічних моделей. Вони забезпечують уніфіковане відображення архітектурних рішень і спрощують подальшу їх верифікацію та удосконалення. Моделювання дозволяє заздалегідь оцінити відповідність проєктних рішень вимогам користувачів, визначити можливі ризики та шляхи інтеграції у корпоративне середовище. Корпоративне сховище є центральним елементом інформаційної інфраструктури, яке забезпечує цілісність, збереження й доступність даних для всіх бізнес-процесів організації. Визначення вимог до сховища дозволяє сформувати уніфікований підхід до побудови системи та гарантувати її ефективність у довгостроковій перспективі.

Насамперед до корпоративного сховища висувається вимога надійності. Дані становлять критично важливий актив, тому сховище повинно забезпечувати їх захист від втрати, пошкодження чи несанкціонованого доступу. Для цього передбачаються механізми резервного копіювання, дублювання інформації та відновлення після збоїв.

Не менш значущою є вимога масштабованості. Система повинна витримувати зростання обсягів даних без втрати продуктивності та дозволяти розширення ресурсів у міру зростання потреб організації. Це включає підтримку нових обчислювальних вузлів, дискових масивів і розподілених механізмів

зберігання.

Ще однією ключовою характеристикою виступає доступність. Користувачі та прикладні системи повинні отримувати безперервний доступ до сховища. Збої або затримки можуть негативно позначитися на роботі підприємства, тому важливим є забезпечення високої готовності системи та мінімізації часу простою.

Вимога продуктивності передбачає високу швидкість обробки запитів та виконання аналітичних операцій. Сховище повинно ефективно працювати з великими масивами даних, забезпечувати оптимізацію пошуку та інтеграцію з бізнес-аналітичними інструментами.

Окрему увагу слід приділити безпеці. Система має підтримувати багаторівневий контроль доступу, автентифікацію користувачів та шифрування даних. Захист інформації від зовнішніх і внутрішніх загроз є обов’язковою умовою функціонування корпоративного сховища.

Важливою є також вимога централізованого управління. Це означає єдину політику роботи з метаданими, уніфіковане керування правами доступу, а також контроль за використанням ресурсів. Завдяки цьому організація отримує прозорий механізм адміністрування та аналітики.

Підсумовуючи, можна визначити основні вимоги до інформаційної системи:

– надійність зберігання даних;

– масштабованість для роботи з великими обсягами інформації;

– висока доступність і безперервність функціонування;

– продуктивність у виконанні запитів і аналітики;

– захист і безпека інформації;

– централізоване управління ресурсами та політиками доступу.

2.2 Ідентифікація користувача та модель доступу

Ідентифікація користувача – це перший рівень захисту інформаційної системи, що дозволяє однозначно визначити, хто саме здійснює доступ до системних ресурсів. Вона реалізується шляхом надання облікових даних, які

зіставляються з даними в системі.

Найбільш поширеним є підхід з використанням пари логін/пароль, однак через загрози, пов’язані з фішингом, атаками словником та підбором паролів, усе ширше впроваджуються сучасні механізми ідентифікації.

Серед таких механізмів – єдина автентифікація (SSO), яка дозволяє користувачеві авторизуватися один раз і отримати доступ до всіх інтегрованих систем без повторного введення облікових даних. Технологічно це реалізується за допомогою протоколів SAML 2.0 або OpenID Connect, які забезпечують обмін токенами автентифікації між системами. Для ще більшої безпеки часто застосовується багатофакторна автентифікація (MFA), що вимагає підтвердження особи додатковими факторами: одноразовим кодом, відбитком пальця або апаратним ключем.

Після ідентифікації система переходить до авторизації – визначення, які дії може виконати користувач. Для цього застосовуються рольові моделі доступу. Найбільш поширеною є модель RBAC (Role-Based Access Control), яка забезпечує управління правами через призначення користувачам ролей. Наприклад, у корпоративній системі можна виділити ролі Адміністратор, Менеджер, Аналітик та Користувач, кожна з яких має власний набір дозволів: адміністратор може створювати та видаляти облікові записи, менеджер – редагувати бізнес-дані, аналітик – переглядати статистику, а звичайний користувач – лише працювати зі своїми даними.

У корпоративному сховищі документів і файлів доцільно передбачити кілька рівнів ролей, які відповідають різним функціональним обов’язкам користувачів:

Адміністратор – має повний контроль над системою, створює й видаляє облікові записи, керує ролями, налаштовує резервне копіювання, доступи та політики безпеки. Він відповідає за підтримку працездатності всієї системи.

Менеджер – організовує документообіг, створює та структурує сховища, надає доступи до окремих каталогів і документів, затверджує документи, контролює робочі процеси.

Редактор (або користувач з правом зміни) – завантажує файли, редагує та

оновлює їх в системі, може коментувати та вести спільну роботу над документами.

Переглядач (читач) – має право лише на перегляд документів, без можливості редагування або видалення. Зазвичай це співробітники, яким потрібен лише доступ до інформації.

Аналітик (за потреби) – користувач, що має доступ до статистики, логів та звітів щодо використання системи, але без можливості втручання у вміст документів.

У розширених корпоративних рішеннях можна реалізувати також гнучкі ролі з атрибутами доступу (наприклад, «Користувач відділу кадрів», «Користувач відділу фінансів»). У такому випадку доступ до документів буде залежати від організаційної структури підприємства та контексту (місце доступу, пристрій, час).

Реалізація рольової моделі передбачає створення в базі даних таких сутностей, як Users, Roles і Permissions, що формують відношення «багато-до-багатьох». Це дозволяє одному користувачу мати декілька ролей, а одній ролі – містити набір дозволів.

2.3 Моделювання структури даних інформаційної системи

Проєктування БД є ключовим етапом у створенні корпоративного сховища документів і файлів, оскільки правильна структура даних визначає ефективність зберігання, швидкість доступу, можливості масштабування та безпеку системи.

Для реалізації обрано реляційну базу даних MS SQL Server, яка забезпечує надійність транзакцій, цілісність даних та можливість виконання складних запитів. Також її вибір обумовлений сумісністю з технологіями .NET Core та MAUI, що дозволяє легко інтегрувати серверну та клієнтську частини системи.

Для підвищення продуктивності передбачено використання індексів за ключовими полями (назва документа, автор, дата створення) та оптимізацію запитів для роботи з великими обсягами даних. Розширюваність архітектури дозволяє додавати нові сутності та атрибути без порушення роботи системи.

Таблиця 2.1 – Опис сутностей бази даних

Сутність	Атрибути	Опис
Users	id (PK, int), username (string, unique), passwordHash (string), email (string), isActive (bool), createdAt (datetime)	Зберігає інформацію про облікові записи користувачів, логін, хеш пароля, email та статус активності.
Roles	id (PK, int), name (string, unique), description (string)	Визначає ролі користувачів у системі (Адміністратор, Менеджер, Редактор, Переглядач, Аналітик) та їх призначення.
UserRoles	id (PK, int), userId (FK, int), roleId (FK, int)	Проміжна таблиця для реалізації зв’язку «багато-до-багатьох» між користувачами та ролями.
Documents	id (PK, int), title (string), type (string), authorId (FK, int), folderId (FK, int), createdAt (datetime), updatedAt (datetime)	Містить метадані документів, включаючи автора, тип документа та дати створення й останньої модифікації.
DocumentVersions	id (PK, int), documentId (FK, int), versionNumber (int), fileId (int), changeNote (text), createdAt (datetime)	Контролює версії документів, забезпечує збереження історії змін та відновлення попередніх редакцій.
Files	id (PK, int), path (string), size (int), checksum (string), mimeType (string)	Зберігає інформацію про фізичне розташування файлів, їх розмір і контрольну суму для перевірки цілісності.
AccessRights	id (PK, int), userId (FK, int), documentId (FK, int), permissionLevel (enum: 'Read', 'Write', 'Delete')	Визначає права доступу користувачів до конкретних документів або каталогів (читання, редагування, видалення).
AuditLog	id (PK, int), userId (FK, int), documentId (FK, int), action (string), ipAddress (string), timestamp (datetime)	Фіксує дії користувачів у системі для аудиту та відстеження спроб несанкціонованого доступу.

На рис. 2.1 показана ER-діаграма БД інформаційної системи корпоративного зберігання електронних документів з контролем доступу та журналом аудиту. Показано основні сутності та зв’язки між ними, які забезпечуючи множинні відносини між користувачами, ролями та документами.

Запропонована модель БД забезпечує ефективне управління документами, контроль версій, гнучке призначення ролей, безпечне зберігання та масштабованість, що відповідає сучасним вимогам до корпоративних інформаційних систем.

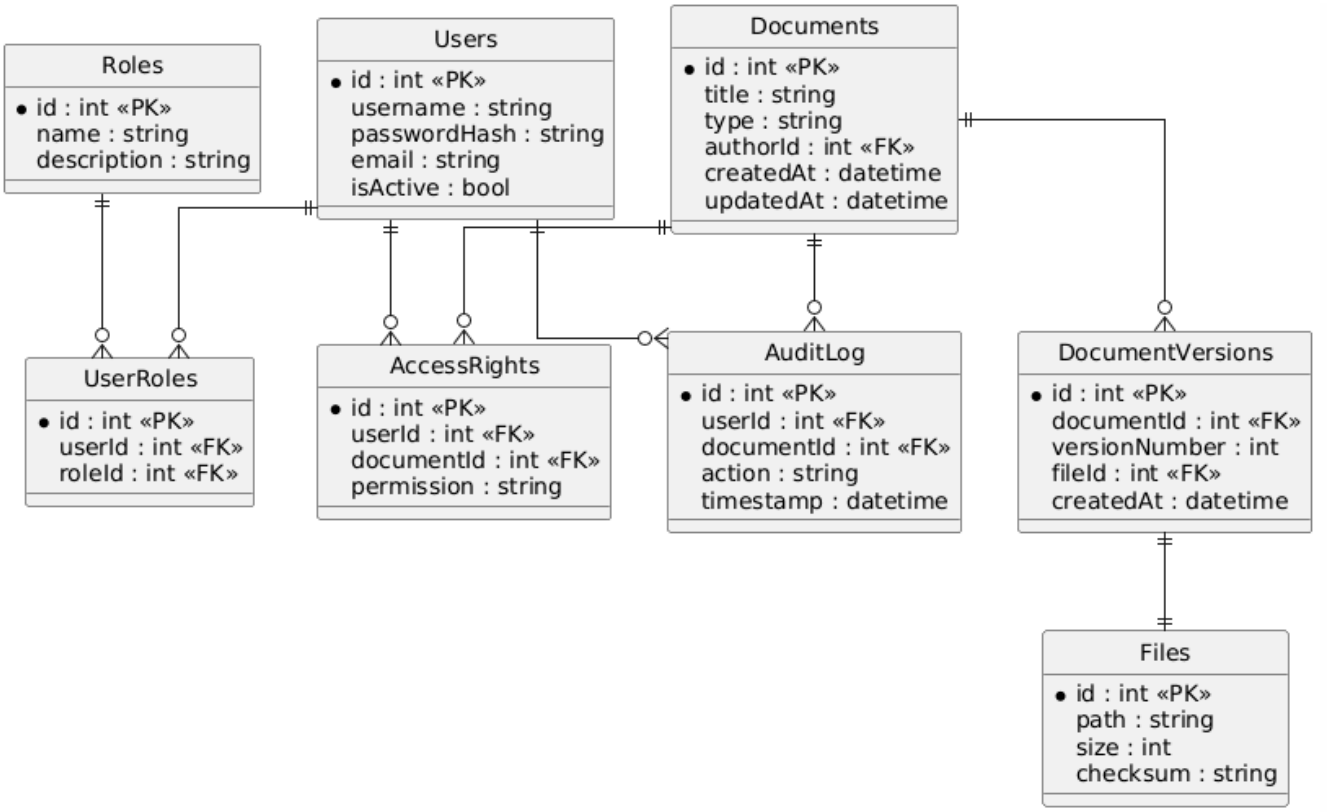

Рисунок 2.1 – ER-діаграма БД інформаційної системи

Особливу увагу приділено безпеці: паролі користувачів зберігаються у хешованому вигляді, файли супроводжуються контрольними сумами, а журнал подій забезпечує можливість аудиту та відстеження спроб несанкціонованого доступу. Логічні зв’язки між сутностями формують основу структури БД. Наприклад, один користувач може мати кілька ролей, а один документ – кілька версій. Взаємозв’язки багато-до-багатьох реалізовані через проміжні таблиці.

2.4 UML-моделі елементів інформаційної системи

Для визначення логічної структури програмних компонентів ІС розроблена UML-діаграма класів (рис. 2.2), яка відображає основні сутності предметної області, їх атрибути, методи та взаємозв'язки. Діаграма ілюструє, як ці класи взаємодіють для реалізації функціоналу системи, такого завантаження нових версій документів, перевірки прав доступу та ведення історії дій.

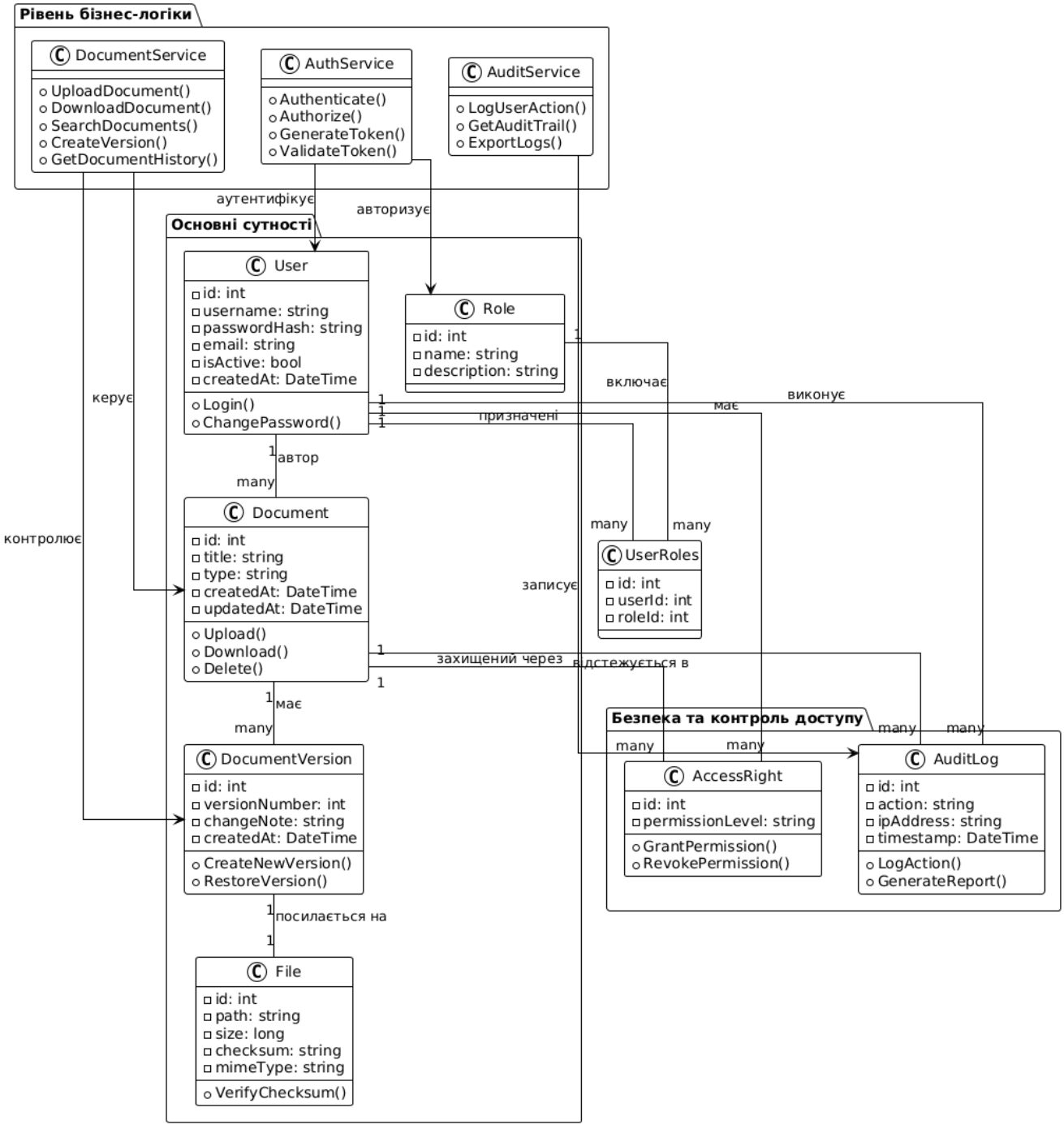

Рисунок 2.2 – Uml-діаграма ієрархії класів

Основними класами системи є:

– User та Role: відповідають за управління користувачами та авторизацію. Зв'язок між ними – багато-до-багатьох, реалізований через список ролей у користувача;

– Document: відповідає за метадані документа (назва, тип, автор). Містить посилання на поточну версію та список пов'язаних версій;

– DocumentVersion: зберігає дані конкретної версії документа, включаючи посилання на фізичний файл (File) та коментар до змін;

– File: інкапсулює інформацію про фізичний файл у сховищі (шлях, розмір, контрольна сума);

– AccessRight: визначає правило доступу, що зв'язує конкретного користувача з конкретним документом і рівнем дозволу;

– Сервісні класи: для інкапсуляції бізнес-логіки передбачено класи DocumentService (відповідає за операції з документами), AuthService (автентифікація та авторизація) та AuditService (логування подій).

2.5 Модель забезпечення безпеки даних та контролю доступу

Безпека даних є критично важливим аспектом корпоративної системи зберігання документів, оскільки система оперує конфіденційною інформацією, втрата або витік якої може призвести до серйозних наслідків для організації. Реалізація механізмів безпеки здійснюється на всіх рівнях архітектури системи з використанням сучасного технологічного стеку, що забезпечує цілісність, конфіденційність та доступність даних протягом усього їхнього життєвого циклу.

Для досягнення мети дослідження запропоновано багаторівневу архітектуру системи безпеки, яка використовує сучасний технологічний стек для захисту даних на всіх рівнях системи. Візуальне представлення архітектури безпеки показано на рис. 2.3, де продемонстровано взаємодію між чотирма основними рівнями захисту: рівнем застосунку, рівнем даних, рівнем інфраструктури та рівнем моніторингу.

На рівні автентифікації використовується інтеграція з IdentityServer 4, що забезпечує підтримку протоколу OAuth 2.0 та OpenID Connect для реалізації єдиного входу (SSO). Для зберігання облікових даних застосовується алгоритм хешування Argon2id розміром 16 байт та хешем 32 байт. Для підвищення рівня безпеки реалізовано багатофакторну автентифікацію через інтеграцію з Google Authenticator API з використанням TOTP токенів тривалістю 30 секунд. Система авторизації базується на ASP.NET Core Identity з розширеними політиками доступу, де модель RBAC реалізована через Claims-based авторизацію з підтримкою динамічних політик.

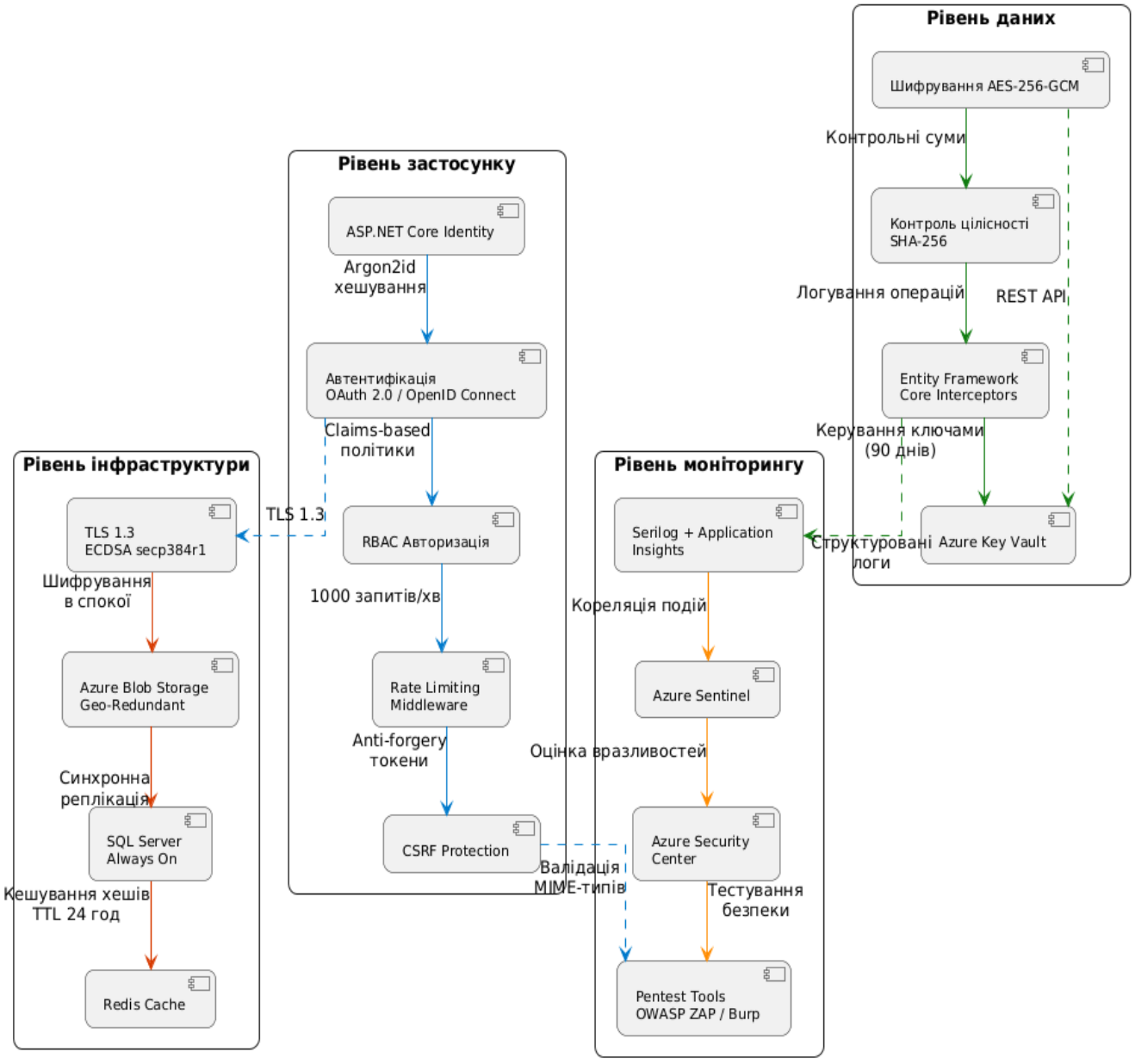

Рисунок 2.3 – Архітектура системи безпеки корпоративного сховища документів

Важливим компонентом захисту є багаторівневе шифрування даних. Для захисту комунікацій використовується TLS 1.3 з сертифікатом на алгоритмі ECDSA secp384r1. Шифрування файлів документів виконується за допомогою AES-256-GCM з ключами, що генеруються через RNGCryptoServiceProvider. Керування ключами здійснюється через Azure Key Vault REST API з автоматичним обертанням кожні 90 днів та версіонуванням ключів.

Для забезпечення цілісності зберіганих документів реалізовано систему контролю на основі криптографічних хеш-функцій через клас System.Security.Cryptography.SHA256. Контрольні суми обчислюються при кожній операції читання/запису, а для оптимізації продуктивності використовується кешування хеш-сум в Redis з TTL 24 години.

Система аудиту та моніторингу забезпечує повну прозорість усіх операцій з документами. Базується на Entity Framework Core Interceptors, що автоматично записують всі операції з базою даних. Логування подій здійснюється через Serilog з інтеграцією в Application Insights для моніторингу в реальному часі. Аналіз підозрілої активності виконується через Azure Sentinel з кастомними правилами кореляції подій, що дозволяє виявляти масове завантаження документів або несанкціоновані спроби доступу.

Архітектура системи включає комплексний захист від веб-загроз через ASP.NET Core Middleware для захисту від CSRF з використанням антифоргері-токенів. Валідація вхідних даних реалізована через FluentValidation з кастомними валідаторами для MIME-типів файлів. Захист від DoS-атак забезпечується через Rate Limiting Middleware з обмеженням 1000 запитів за хвилину на користувача, а параметризовані запити запобігають SQL-ін'єкціям.

Надійність системи підтримується через багаторівневу стратегію резервного копіювання та відновлення. Реалізовано SQL Server Always On Availability Groups з синхронною реплікацією в реальному часі. Для файлового сховища використовується Azure Blob Storage з гео-надмірним зберіганням (GRS) та версіонуванням об'єктів. Архівування логів здійснюється в Azure Data Lake Storage Gen2 з автоматичним переміщенням до холодного сховища після 30 днів.

Архітектура безпеки розроблена з урахуванням вимог міжнародних стандартів. Система відповідає вимогам GDPR через реалізацію Data Subject Requests API для експорту та видалення персональних даних. Для відповідності PCI DSS використовується токенізація платіжних даних через Stripe Elements. Моніторинг безпеки здійснюється через Azure Security Center з оцінкою вразливостей, а всі компоненти системи проходять регулярне пенетраційне тестування з використанням OWASP ZAP та Burp Suite.

Інтеграція з корпоративними системами безпеки забезпечується через REST API для обміну подіями безпеки в форматі CEF. Синхронізація з Active Directory виконується через LDAP протокол з використанням TLS для захисту каналу зв'язку. Така комплексна архітектура забезпечує ефективний захист від сучасних загроз інформаційній безпеці та дотримання вимог регуляторних органів.

2.6 Вибір технологічних рішень для реалізації концепції системи

Для формування концепції інформаційної системи корпоративного зберігання електронних документів здійснено аналіз сучасних технологічних рішень, за результатами якого визначено оптимальний набір програмних засобів, що забезпечують належний рівень продуктивності, масштабованості, інформаційної безпеки та зручності подальшого розвитку системи.

Серверна частина реалізується засобами мови програмування C# та платформи .NET Core. Цей вибір зумовлений тим, що C# є однією з найпоширеніших корпоративних мов програмування, яка поєднує високу продуктивність із простотою підтримки, а .NET Core забезпечує кросплатформеність та сучасні можливості побудови веб-сервісів. Крім того, використання .NET Core дозволяє реалізувати REST API для взаємодії з клієнтськими додатками та іншими сервісами, що є важливою вимогою корпоративних систем.

Клієнтська частина створюється з використанням технології .NET MAUI, яка надає можливість побудови кросплатформених застосунків із єдиною кодовою

базою. Це рішення обране завдяки його здатності забезпечити підтримку Windows, Android, iOS та macOS, що є актуальним для систем, орієнтованих на різні категорії користувачів. Використання MAUI також дозволяє спроєктувати адаптивний інтерфейс, доступний для людей із особливими потребами, що відповідає сучасним вимогам до корпоративних інформаційних систем.

Для зберігання даних застосовується Microsoft SQL Server. Ця система керування базами даних характеризується високою надійністю, стабільністю та сумісністю з платформою .NET. SQL Server забезпечує підтримку транзакцій, що є критично важливим у випадках одночасної роботи великої кількості користувачів, а також має розвинені засоби резервного копіювання та відновлення. Використання ORM-технології Entity Framework Core спрощує роботу з даними та гарантує ефективну інтеграцію з серверною частиною.

Забезпечення якості та стабільності системи здійснюється завдяки комплексному підходу до тестування. Для модульного тестування використовується бібліотека xUnit, що дозволяє перевіряти коректність реалізації бізнес-логіки та взаємодії з базою даних. Для інтеграційного тестування API застосовується інструмент Postman, який забезпечує перевірку коректності запитів і відповідей, а також дозволяє оцінити рівень захищеності системи від некоректних чи шкідливих дій з боку користувача.

Розробка та супровід системи здійснюється у середовищі Visual Studio. Цей вибір зумовлений повною підтримкою технологій C#, .NET Core і MAUI, інтегрованими інструментами для налагодження та відлагодження, зручною роботою з системою контролю версій Git, а також можливістю інтеграції з бібліотеками тестування. Visual Studio надає розробнику комплексний набір засобів, які дозволяють вести розробку у відповідності до сучасних інженерних стандартів.

Таким чином, комбінація C#, .NET Core, .NET MAUI, Microsoft SQL Server, xUnit, Postman і Visual Studio забезпечує побудову надійної корпоративної системи, яка поєднує продуктивність серверної частини, зручність клієнтського інтерфейсу, надійність зберігання даних і сучасні засоби тестування та підтримки. Обрані

технології не лише відповідають сучасним вимогам до інформаційних систем корпоративного рівня, а й забезпечують основу для подальшого розвитку та масштабування розробленого рішення.

2.7 Висновки за розділом

У розділі сформовано модель інформаційної системи корпоративного зберігання електронних документів. На основі аналізу предметної області визначено функціональні та нефункціональні вимоги до системи, серед яких ключову роль відіграють доступність, масштабованість, продуктивність, надійність зберігання даних, а також високий рівень інформаційної безпеки.

Сформовано модель користувачів та їх доступу, що передбачає використання ролей із чітким розмежуванням повноважень. Така система доступу дозволяє забезпечити контроль над документопотоками та зменшити ризики несанкціонованої модифікації інформації.

Розроблено логічну структуру даних, яка включає визначення сутностей, їх атрибутів і зв’язків у реляційній моделі, що забезпечує ефективне зберігання метаданих електронних документів, підтримку контролю версій та ведення аудиту користувацьких операцій.

Для формалізації архітектури й поведінки системи створено набір UML-моделей, які відображають ключові сценарії використання, взаємодію компонентів і структуру елементів інформаційної системи. Застосовані засоби моделювання забезпечують уніфіковане представлення проєктних рішень і полегшують подальшу їх перевірку на відповідність вимогам.

Модель безпеки, сформована в межах розділу, описує основні механізми захисту даних, зокрема автентифікацію, авторизацію, розмежування прав доступу й аудит користувачів, що забезпечує відповідність вимогам до захисту інформації.

Таким чином, результати проєктування та моделювання створюють цілісну основу для подальшої специфікації й верифікації моделі системи та можуть бути використані як методологічна база для її практичної реалізації.

3 СПЕЦИФІКАЦІЯ ТА ВЕРИФІКАЦІЯ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Специфікація архітектури інформаційної системи

Серверна частина має бути реалізована як REST-сервіс на платформі ASP.NET Core. Вона відповідає за автентифікацію/авторизацію користувачів, валідацію запитів, бізнес-логіку, доступ до бази даних (MS SQL Server), логування, кешування, обробку помилок і видачу API-відповідей у формі JSON. Архітектура побудована по принципах чистої архітектури / «поміркованих» мікросервісів: розділення на шари Presentation (Controllers), Application (Services, DTOs), Domain (Entities, Business rules), Infrastructure (EF Core, Repositories, External services).

Архітектурна схема серверної частини має такі компоненти:

– API layer (Controllers) – HTTP-ендпойнти, маршрути, аутентифікація/авторизація, валідація вхідних DTO;

– Application / Service layer – реалізація use-case'ів (завантаження файлу, створення документа, надання прав), транзакції, оркестрація викликів репозиторіїв;

– Domain layer – сутності (User, Document, File, DocumentVersion, AccessRight), бізнес-правила;

– Infrastructure layer – EF Core DbContext, репозиторії (Repository pattern), інтеграції з файловим сховищем (локальна FS або blob storage), поштові/логічні сервіси, конфігурація JWT/Azure AD тощо;

– Cross-cutting – логування (Serilog), обробка винятків (middleware), аудит (AuditLog), кеш (MemoryCache/Redis), rate limiting, health checks.

Для програмної реалізації серверної частини необхідно виконати такі дії:

1. створити проєкт MyCorp.Storage.Api (ASP.NET Core Web API, рис. 3.1);

2. впровадити шари Controllers, Services, Repositories/DbContext;

3. налаштувати EF Core + MS SQL Server, створити міграції і базову схему;

4. реалізувати аутентифікацію JWT (або інтеграцію з SSO), парольний хешинг;

5. розробити REST-ендпойнти (табл. 3.1), які мають повертати стандартизовану структуру відповіді (status, message, data, errors) та використовувати HTTP-статуси (200, 201, 204, 400, 401, 403, 404, 500);

6. додати global exception middleware, Serilog, health checks, rate limiting (якщо потрібно);

7. реалізувати тестування: unit (xUnit), інтеграція, Postman-колекції;

8. підготувати CI/CD-pipeline для застосування міграцій та розгортання ІС.

```
/src
  /MyCorp.Storage.Api               // ASP.NET Core Web API (Controllers, DTOs)
  /MyCorp.Storage.Application       // Services, interfaces, DTOs
  /MyCorp.Storage.Domain            // Entities, enums, domain services
  /MyCorp.Storage.Infrastructure    // EF Core DbContext, Repositories, Migrations
/tests
  /MyCorp.Storage.UnitTests         // xUnit tests for services/repos
```

Рисунок 3.1 – Орієнтовна файлова структура проєкту у Visual Studio

Таблиця 3.1 – Ресурси та ендпойнти ІС для керування документами, користувачами, правами доступу та версіями файлів

Ресурс	Метод	Ендпойнт	Опис	Приклад відповіді
1	2	3	4	5
Документи	GET	/api/documents	Отримати список усіх документів користувача або за роллю	[{ "id": 1, "title": "Наказ №12", "owner": "Іваненко", "createdAt": "2025-10-05" }]
	GET	/api/documents/{id}	Отримати метадані конкретного документа	{ "id": 1, "title": "Наказ №12", "type": "PDF", "version": "v3.0" }
	POST	/api/documents	Завантажити новий документ	{ "message": "Документ успішно додано" }
	PUT	/api/documents/{id}	Оновити метадані або версію документа	{ "message": "Документ оновлено" }
	DELETE	/api/documents/{id}	Видалити документ (для адміністраторів)	{ "message": "Документ видалено" }
Логи	GET	/api/logs	Отримати журнал дій користувачів	[{ "user": "Admin", "action": "Deleted document", "time": "2025-10-07T10:23" }]

Продовження табл. 3.1

1	2	3	4	5
Версії документів	GET	/api/documents/{id}/versions	Отримати історію версій конкретного документа	[{ "version": "v1.0", "date": "2025-09-01" }, { "version": "v2.0", "date": "2025-09-15" }]
	POST	/api/documents/{id}/versions	Додати нову версію документа	{ "message": "Нова версія створена" }
Користувачі	GET	/api/users	Отримати список користувачів (для адміністратора)	[{ "id": 1, "username": "admin", "role": "Administrator" }]
	GET	/api/users/{id}	Отримати профіль користувача	{ "id": 2, "username": "manager1", "role": "Manager" }
	POST	/api/users/register	Зареєструвати нового користувача	{ "message": "Користувача створено" }
	POST	/api/users/login	Авторизація користувача, повертає токен JWT	{ "token": "eyJhbGci..." }
Ролі	GET	/api/roles	Отримати список ролей у системі	["Administrator", "Editor", "Viewer"]
	PUT	/api/users/{id}/role	Змінити роль користувача	{ "message": "Роль користувача оновлено" }
Доступи	GET	/api/permissions	Переглянути політики доступу до документів	[{ "documentId": 3, "user": "Petrenko", "access": "read" }]
	POST	/api/permissions	Додати або змінити права доступу	{ "message": "Права доступу оновлено" }

Контролери повинні бути тонкими – приймати DTO, викликати Application/Service layer, обробляти помилки та повертати response DTO. Валідацію полів робимо через DataAnnotations (ModelState).

Бізнес-логіка ІС реалізує контроль доступу через перевірку авторизації та прав користувача безпосередньо у сервісному шарі, при цьому контролери додатково захищені атрибутами [Authorize]. Критичні операції, такі як створення документа, завантаження файлу та формування версії, виконуються в межах транзакцій бази даних, що забезпечує узгодженість даних. Для деяких REST-ендпойнтів реалізовано ідемпотентність, тобто повторне виконання запиту не

створює дублікатів завдяки використанню токенів операцій. Усі значущі дії користувачів фіксуються в журналі аудиту (AuditLog) за допомогою сервісу аудиту.

У системі передбачено валідацію вхідних даних за допомогою бібліотеки FluentValidation; у разі помилки сервер повертає код 400 із детальним описом. Для централізованої обробки винятків застосовується глобальний middleware (ExceptionHandlingMiddleware), який перетворює необроблені помилки у структуровані відповіді з кодом і трасуванням, що записується до логів, тоді як користувачеві надсилається узагальнене повідомлення.

Логування та моніторинг виконуються за допомогою Serilog із виведенням у файл, консоль або системи збору журналів на кшталт ElasticSearch чи Seq. Для трасування запитів використовується correlationId. Перевірка стану сервера реалізована через endpoint, що може використовуватись балансувальником навантаження.

Для підвищення продуктивності впроваджене кешування часто використовуваних метаданих за допомогою MemoryCache або Redis. Дані мають обмежений час життя (TTL) і оновлюються при зміні документа, що забезпечує актуальність кешу.

Для роботи з MS SQL Server використано EF Core (Entity Framework Core) як ORM. Архітектура використовує Repository + Unit of Work (або безпосередньо DbContext у сервісах при простоті). Приклад налаштування DbContext наведено у додатках.

Міграції бази даних у ІС виконуються за допомогою механізму EF Core, що забезпечує контроль над змінами структури даних. Усі міграції зберігаються в репозиторії, а під час розгортання системи передбачено автоматичне застосування міграцій, однак перед цим обов’язково проводиться тестування, щоб уникнути порушення цілісності даних.

Складні операції, що змінюють кілька таблиць одночасно, виконуються в межах транзакції, яка відкривається за допомогою патерну Unit of Work. Залежно від вимог бізнес-логіки можуть налаштовуватись рівні ізоляції транзакцій – від

ReadCommitted до RepeatableRead, що дозволяє балансувати між узгодженістю даних та конкурентністю доступу.

Оптимізація запитів до бази даних здійснюється шляхом створення індексів на полях, що часто використовуються у фільтрації чи сортуванні. Для запитів, які не передбачають змінювання об’єктів, застосовується режим AsNoTracking(), що знижує навантаження на контекст EF Core. Великі двійкові об’єкти, зокрема файли документів, не зберігаються безпосередньо в базі – у ній фіксуються лише метадані (шлях до файлу, контрольна сума, розмір), тоді як самі дані розміщуються у файловому сховищі, що дозволяє оптимізувати роботу з великими обсягами інформації.

Безпека з’єднання з БД забезпечується шифруванням каналу передачі, особливо у випадку її віддаленого або хмарного розміщення. Крім того, обліковий запис, під яким здійснюється підключення, має мінімально необхідні привілеї відповідно до принципу least privilege, що знижує ризики несанкціонованого доступу та підвищує загальний рівень безпеки системи. Приклад конфігурації автентифікації наведено у додатках.

Тестування серверної частини ІС корпоративного зберігання документів виконується для перевірки її надійності, коректності роботи бізнес-логіки, безпеки та продуктивності. Основною метою цього етапу є виявлення можливих помилок ще до розгортання системи у промисловому середовищі, а також підтвердження відповідності реалізації вимогам, визначеним на етапі проєктування.

Серверна частина побудована на основі .NET Core, що дозволяє використовувати вбудовані інструменти тестування, зокрема бібліотеки xUnit, Moq та FluentAssertions.

Модульне тестування зосереджене на перевірці окремих методів бізнес-логіки. Наприклад, для сервісу керування документами перевіряється коректність створення, оновлення, видалення, пошуку та фільтрації файлів за різними критеріями. У тестах застосовується імітація залежностей за допомогою бібліотеки Moq, що дозволяє відокремити тестований компонент від зовнішніх систем, таких як база даних або файлове сховище (рис. 3.2).

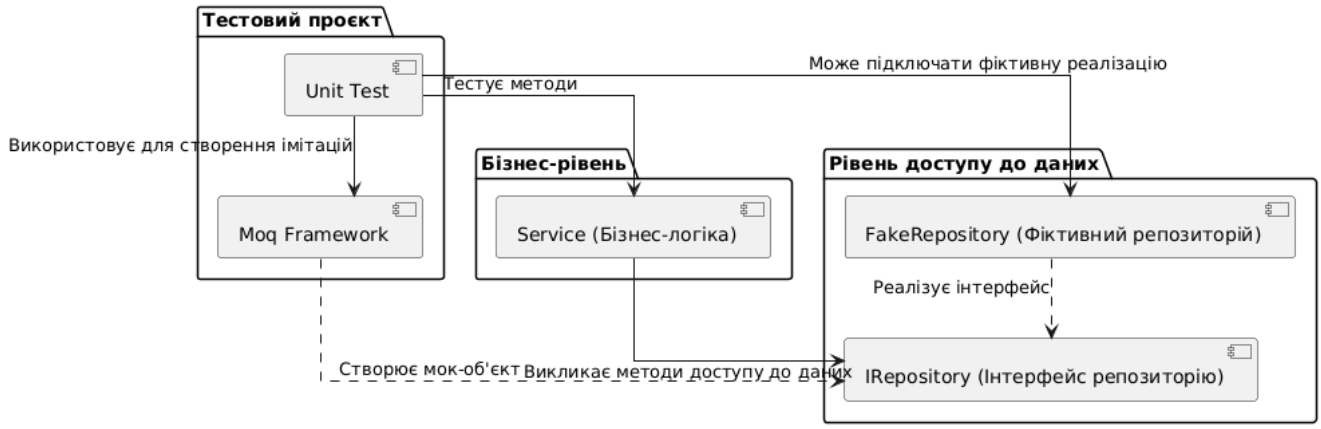

Рисунок 3.2 – Процес модульного тестування серверної частини інформаційної системи

Інтеграційне тестування спрямоване на перевірку взаємодії між компонентами – контролерами, сервісами та сховищем даних. Для цього використовується InMemoryDatabase або тестовий екземпляр MS SQL Server, який запускається у середовищі TestContainer або Docker. Запити виконуються через HTTP-запити до тестового серверу, що дозволяє перевірити повний цикл роботи REST API – від прийому запиту до повернення відповіді (рис. 3.3). Тести охоплюють як стандартні сценарії (отримання документів, автентифікація, оновлення метаданих), так і негативні випадки (відсутність прав, некоректні параметри, неіснуючі ресурси).

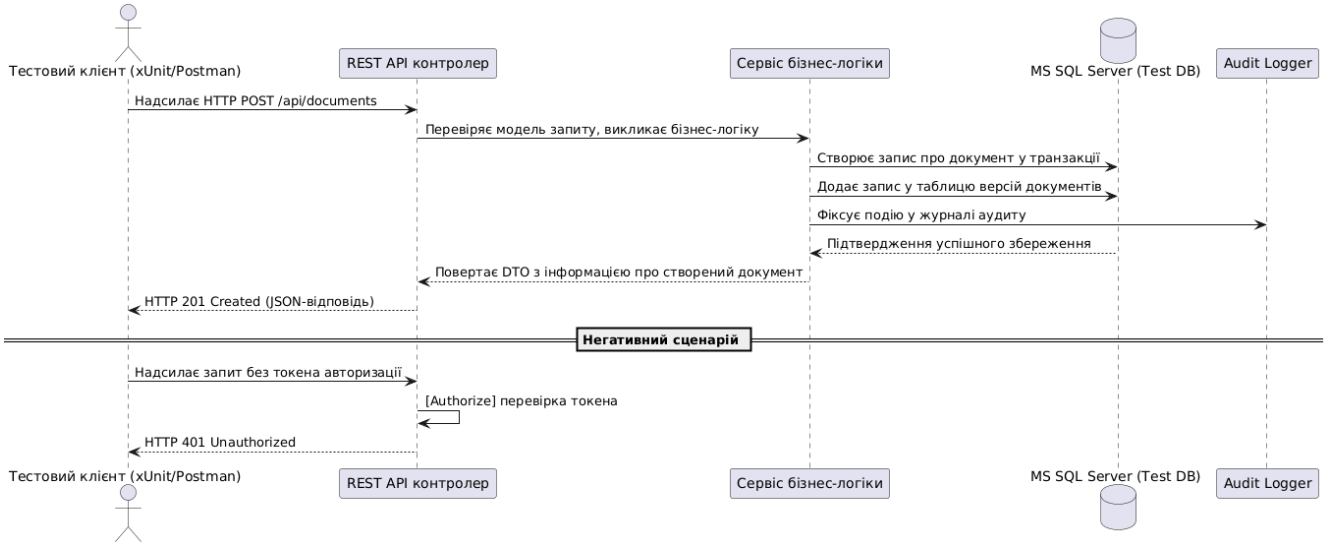

Рисунок 3.3 – Діаграми послідовності процесу інтеграційного тестування серверної частини інформаційної системи

Перевірка безпеки проводиться з метою переконатися, що механізми автентифікації та авторизації реалізовані належним чином. Тести включають спроби доступу без токена, з простроченим токеном або з недостатніми правами. Також перевіряється, чи правильно працюють політики доступу для різних ролей (адміністратор, користувач, гість).

Для перевірки продуктивності серверної частини застосовуються інструменти Postman Collection Runner або Apache JMeter, які дозволяють симулювати навантаження. Тестування проводиться з метою визначення часу відгуку REST API, стабільності при збільшенні кількості запитів і поведінки системи при пікових навантаженнях.

Результати тестування автоматично збираються за допомогою GitHub Actions, де тести виконуються при кожному коміті в репозиторій. У разі виявлення помилки процес збірки блокується до її виправлення, що гарантує стабільність головної гілки коду.

Загалом, тестування серверної частини забезпечує не лише виявлення технічних недоліків, а й формує основу для впевненого масштабування, подальшої розробки клієнтських компонентів і гарантує безпечну роботу системи у корпоративному середовищі.

3.2 Ключові сценарії використання моделі інформаційної системи

Процес специфікації моделі інформаційної системи корпоративного зберігання електронних документів передбачає формалізований опис поведінки користувачів у типових ситуаціях взаємодії із системою. Такі сценарії визначають очікувані функції, уточнюють вимоги до бізнес-логіки та забезпечують основу для верифікації узгодженості проєктних рішень.

У межах розробленої моделі виділено ролі користувачів: адміністратор, редактор, звичайний користувач, переглядач та аналітик. Кожна з ролей характеризується певним набором дій, спрямованих на керування документами, доступом і супровідними операціями. Основні сценарії використання

охоплюють такі процеси:

1. Аутентифікація та авторизація користувача. Користувач здійснює вхід у систему за допомогою логіна і пароля або через зовнішній сервіс OAuth 2.0. Після перевірки облікових даних система надає доступ до функцій відповідно до ролі користувача. Адміністратор може створювати та блокувати облікові записи.

2. Завантаження нового документа. Авторизований користувач вибирає файл для завантаження, вводить його опис і метадані (назву, категорію, теги). Система перевіряє унікальність і цілісність файлу, створює запис у базі даних і зберігає документ у файловому сховищі. Після цього документ стає доступним для пошуку та перегляду відповідно до встановлених прав доступу.

3. Перегляд і редагування документа. Користувач отримує перелік доступних документів, може переглядати метадані, історію змін і попередні версії. У разі наявності відповідних прав він може завантажити оновлену версію документа, яка фіксується в системі як нова ревізія з зазначенням часу, користувача та коментаря.

4. Управління правами доступу. Адміністратор визначає, які користувачі або групи мають право переглядати, редагувати чи видаляти певні документи або каталоги. Зміни прав доступу автоматично фіксуються в журналі аудиту, що забезпечує контроль дій користувачів і прозорість операцій.

5. Пошук та фільтрація документів. Користувач може виконувати пошук за ключовими словами, датою створення, автором або тегами. Система реалізує повнотекстовий пошук у метаданих і підтримує сортування результатів. Для зручності використовується кешування найчастіше виконуваних запитів.

6. Резервне копіювання та відновлення документів. Система автоматично створює резервні копії бази даних і файлів відповідно до політики збереження. У разі збоїв адміністратор може ініціювати відновлення даних до певного моменту часу, забезпечуючи безперервність роботи.

Для візуалізації ключових сценаріїв розроблена UML-діаграма варіантів використання, яка відображає взаємодію акторів із системою (рис. 3.4).

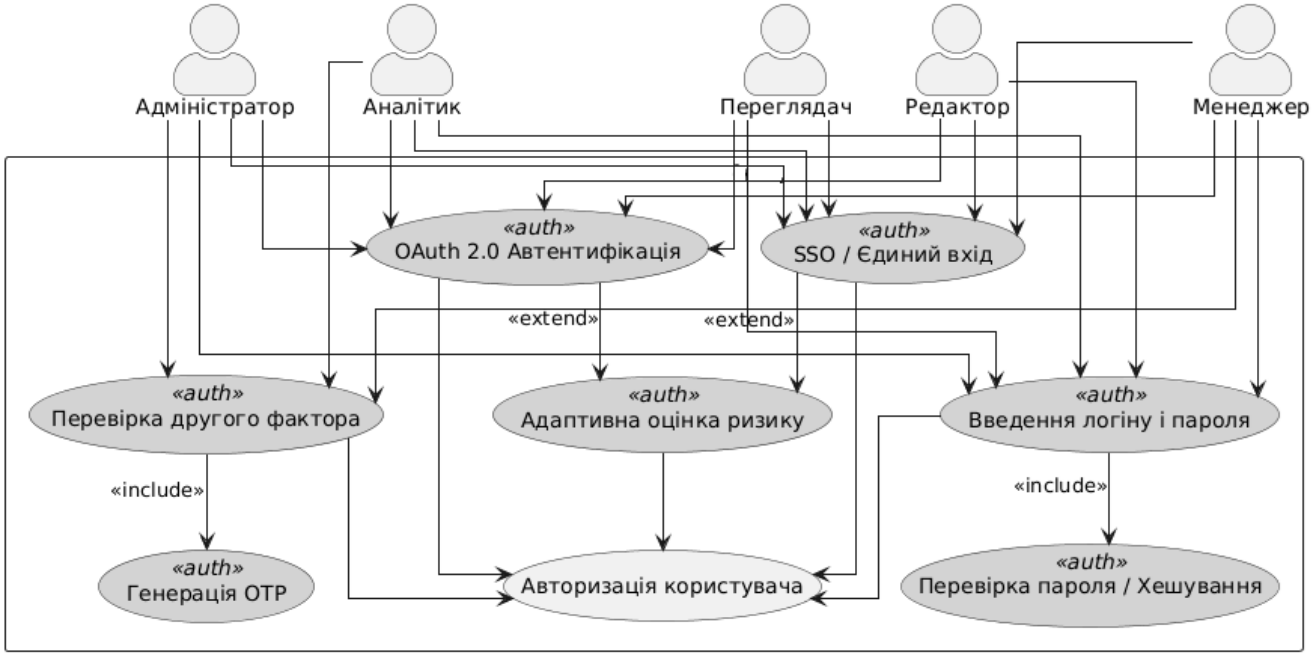

Рисунок 3.4 – UML-діаграма варіантів використання інформаційної системи

3.3 Специфікація моделі даних інформаційної системи

Модель даних є ключовим елементом інформаційної системи, оскільки визначає структуру збереження інформації, її зв’язність та правила забезпечення цілісності. Розроблена логічна модель спирається на результати проєктування, подані у розділі 2, та формалізує представлення основних сутностей предметної області у вигляді реляційної структури.

Метою специфікації моделі даних є деталізація атрибутного складу сутностей, визначення типів зв’язків між ними, а також встановлення обмежень, що забезпечують коректність і захищеність даних на всіх етапах їх обробки. Логічна модель підтримує контроль версій документів, аудит користувацьких дій та рольове керування доступом, що є критично важливими складовими корпоративної безпеки.

Атрибутивний склад сутностей ідентифіковано на етапі проєктування (табл. 2.1), тому в межах специфікації уточнено такі правила:

– кожна сутність має унікальний первинний ключ;

– зв’язки реалізуються через зовнішні ключі із забезпеченням цілісності;

– атрибути автентифікації (username, email) є унікальними, що запобігає

дублюванню облікових записів;

– атрибути аудиту (createdAt, updatedAt, timestamp) забезпечують відстеження всіх змін;

– атрибут versionNumber у сутності DocumentVersions забезпечує впорядкованість ревізій;

– в AccessRights використано обмежений домен значень permissionLevel ('Read', 'Write', 'Delete') для зниження ризиків несанкціонованого доступу.

Таблиця 3.1 – Комплекс обмежень, спрямованих на забезпечення надійності даних

Категорія обмеження	**Реалізація у моделі**	**Призначення**
Унікальність	Users.username, Roles.name	Запобігання дублюванню
Референційна цілісність	FK userId, roleId, documentId	Узгодженість інформації між сутностями
Доменні обмеження	permissionLevel ∈ {'Read','Write','Delete'}	Гарантія безпечного доступу
Історизація даних	Зберігання всіх версій у DocumentVersions	Контроль змін, аудит
Аудит операцій	AuditLog	Відстеження дій користувачів

Структура сутностей та зв’язків представлена на ER-діаграмі логічної моделі даних (рис. 2.1), яка є основою для формальної перевірки відповідності вимогам.

Проведена специфікація довела, що розроблена модель даних відповідає сформульованим у розділі 2 функціональним і нефункціональним вимогам. Вона повністю підтримує механізми розмежування прав доступу користувачів, забезпечуючи правильне відображення визначеної рольової моделі та можливість налаштування дозволів на рівні окремих документів. Структура зберігання інформації дозволяє фіксувати всі редакції електронних документів, що гарантує доступність історії змін та можливість відновлення попередніх версій.

Модель забезпечує збереження та обробку як метаданих, так і фізичного

вмісту файлів, що дозволяє системі функціонувати як повноцінне корпоративне сховище електронних документів. Наявність окремого механізму аудиту забезпечує прозоре відстеження всіх дій користувачів, створюючи передумови для виявлення несанкціонованих маніпуляцій та розслідування інцидентів інформаційної безпеки.

Завдяки специфікації можна стверджувати, що розроблена логічна модель даних є цілісною, повною і внутрішньо узгодженою. Вона адекватно відображає вимоги предметної області та формує надійну основу для подальшої реалізації інформаційної системи корпоративного зберігання електронних документів.

3.4 Специфікація механізмів безпеки інформаційної системи

Безпека даних є одним із ключових аспектів проєктування інформаційної системи корпоративного зберігання електронних документів. Розроблена модель безпеки охоплює механізми автентифікації користувачів, авторизації доступу, аудитування операцій та забезпечення цілісності інформації. Метою специфікації є формалізація визначених у розділі 2 правил безпечної взаємодії користувачів із ресурсами системи.

Запропонована модель доступу базується на концепції RBAC, де кожен користувач системи отримує певну роль, яка визначає доступні йому дії. Ролі адмініструються централізовано, що спрощує керування правами та знижує ризик ненавмисного надання надлишкових привілеїв.

Механізм доступу доповнюється окремим компонентом призначення дозволів на рівні окремих документів, що забезпечує деталізацію політики безпеки залежно від конфіденційності даних. Це дозволяє адміністраторам формувати індивідуальні профілі доступу і чітко контролювати обсяг дозволених операцій для кожного користувача.

Усі зміни ролей та дозволів фіксуються у внутрішніх журналах, що підвищує прозорість адміністрування й забезпечує контроль над операціями, пов’язаними із керуванням доступом.

Автентифікація забезпечує ідентифікацію користувача перед початком роботи із системою. Для цього застосовуються захищені облікові дані, включаючи хешовані паролі, а також механізми перевірки актуальності облікового запису. Авторизація проводиться після автентифікації і визначає рівень доступу користувача відповідно до його ролі та встановлених прав на документи.

Усі взаємодії користувачів із системою супроводжуються аудитом. Журнал аудиту містить інформацію про виконані дії, час проведення операції, ідентифікатор користувача та документ, до якого було здійснено доступ. Такий підхід дозволяє здійснювати як ретроспективний аналіз інцидентів безпеки, так і превентивний моніторинг спроб несанкціонованого доступу.

Модель безпеки передбачає використання криптографічних методів для запобігання спотворенню чи розкриттю захищеної інформації. Контрольні суми файлів дозволяють виявити будь-які зміни у фізичному вмісті документа, забезпечуючи валідацію цілісності під час зберігання та передачі.

Конфіденційність даних реалізується шляхом обмеження доступу лише для авторизованих користувачів і контролю за тим, щоб дозволи були мінімально необхідними для виконання службових обов’язків. Механізми захисту доповнюють перевірки актуальності користувацьких сесій і відстеження підозрілої поведінки, що дозволяє оперативно реагувати на потенційні кіберзагрози.

Узгодження зазначених методів забезпечує суцільне захищене середовище для збереження документів на всіх етапах їх життєвого.

Розроблена матриця трасування (табл. 3.2), яка дозволяє оцінити повноту реалізації вимог безпеки у моделі та підтверджує, що всі сформульовані механізми мають пряме відображення у структурі й поведінці системи.

На основі проведеного трасування встановлено, що модель безпеки інформаційної системи повністю забезпечує виконання вимог, визначених на етапі проєктування. Усі механізми захисту даних інтегровані в логічну структуру моделі та підтверджені верифікацією архітектурних і поведінкових компонентів. Це свідчить про цілісність і достатність запропонованого підходу до захисту інформації.

Таблиця 3.2 – Трасування вимог безпеки до реалізованих механізмів захисту

Вимога безпеки	Реалізація у моделі	Верифікація відповідності
Розмежування прав доступу до документів	RBAC-модель ролей і AccessRights	Аналіз відповідності сценаріям доступу
Захист від несанкціонованих змін	Контроль версій DocumentVersions	Перевірка узгодженості змін
Автентифікація користувачів	Зберігання автентифікаційних даних у Users	Верифікація коректності авторизації
Забезпечення цілісності документів	Контрольні суми файлів у Files	Перевірка збереження даних при операціях
Прозорість і контроль дій користувачів	AuditLog – журнал усіх операцій	Аналіз слідовності подій
Запобігання підвищенню привілеїв	Централізоване адміністрування ролей	Перевірка відповідності дозволів
Обмеження доступу до неактивних обліковок	Ознака Users.isActive	Перевірка виконання політики блокування

3.5 Верифікація моделі інформаційної системи

Метою верифікації моделі інформаційної системи корпоративного зберігання електронних документів є оцінка її відповідності функціональним і нефункціональним вимогам, а також перевірка логічної узгодженості архітектурних, поведінкових і структурних компонентів. Верифікація забезпечує підтвердження того, що побудована модель може бути використана як основа для подальшої технологічної реалізації без необхідності суттєвих коригувань.

Порівняльний аналіз показав, що всі ключові вимоги, визначені в розділі 2, повністю реалізовані в логічній моделі. Архітектурні рішення відповідають політиці централізованого керування доступом, а застосовані механізми контролю версій та аудит дій забезпечують функціональну завершеність системи. Модель здатна підтримувати всі сценарії взаємодії користувачів із документами, включаючи створення, зберігання, перегляд, редагування й резервування.

Під час аналізу на рівні UML-діаграм, ER-моделі та визначених бізнес-процесів встановлено, що модель не містить конфліктів між поведінковими та

інформаційними компонентами. Сценарії використання однозначно відображаються у структурі даних, а всі сутності предметної області мають чітке призначення й достатній набір атрибутів для опису функціоналу системи. Це забезпечує цілісність і контрольованість життєвого циклу електронних документів.

Для забезпечення системності оцінки розроблено матрицю трасування вимог до елементів моделі системи (табл. 3.3), яка дозволила встановити прямі відповідності між вимогами користувачів, вимогами безпеки, вимогами до доступності та відповідними елементами моделі: архітектурними компонентами, сутностями бази даних, логічними зв’язками, механізмами контролю доступу та сценаріями взаємодії.

Таблиця 3.3 – Матриця трасування вимог до елементів моделі системи

Сформульована вимога	**Джерело вимоги**	**Реалізація в моделі**	**Перевірка відповідності**
Автентифікація та авторизація користувачів	2.2	Users, Roles, AccessRights	3.4
Керування ролями і правами доступу	2.2, 2.5	RBAC-модель	3.4, 3.5
Збереження документів у корпоративному середовищі	2.3	Documents, Files	3.3
Контроль версій документів	2.3	DocumentVersions	3.3
Пошук документів за метаданими	2.4	Атрибути Documents	3.2, 3.5
Ведення журналу реєстрації операцій	2.5	AuditLog	3.4
Масштабованість системи	2.1	Багаторівнева архітектура	3.1, 3.5
Захист цілісності та конфіденційності	2.5	Контрольні суми, контроль доступу	3.4
Структурованість даних у каталогах	2.3	Ієрархічна модель	3.3

Проведено аналіз моделі за показниками безпеки, цілісності даних, керованості, масштабованості, продуктивності, надійності та зручності використання (табл. 3.4). Результати аналізу підтвердили, що модель забезпечує належний рівень вимог до корпоративних систем зберігання документів та відповідає очікуваному функціональному наповненню.

Таблиця 3.4 – Верифікація відповідності моделі функціональним і нефункціональним вимогам системи

Формулювання вимоги	**Реалізація в моделі**	**Верифікація відповідності**
Система має забезпечувати автентифікацію та рольову авторизацію	Модель доступу, ролі користувачів: адмін, редактор, переглядач, аналітик	Перевірка відповідності сценаріям доступу (п. 3.4, 3.5)
Забезпечення захищеного зберігання документів	Логічна модель даних з контролем версій (п. 2.3), аудит подій	Аналіз покриття ризиків та механізмів перевірки цілісності (п. 3.4)
Підтримка пошуку і фільтрації документів	Проєктування процесу пошуку в метаданих (п. 2.4)	Верифікація відповідності функціональним сценаріям (п. 3.2, 3.5)
Управління правами доступу адміністраторами системи	RBAC-модель, атрибути доступу до ресурсів (п. 2.2, 2.5)	Сценарії управління доступом і аудит дій (п. 3.2, 3.4)
Відновлення даних після збоїв або втрати	Збереження метаданих і моделей резервування (п. 2.3)	Верифікація логічної стійкості моделі даних (п. 3.3, 3.5)
Масштабованість системи під зростаючі обсяги документів	Архітектура на основі клієнт-серверної моделі (п. 2.4)	Узгодження архітектури з нефункціональними вимогами (п. 3.1, 3.5)
Забезпечення аудиту операцій користувача	Поля аудиту та відповідні сутності у БД (п. 2.3)	Верифікація покриття сценаріїв безпеки (п. 3.4)

3.6 Висновки за розділом

У розділі виконано специфікацію та верифікацію концептуальної моделі інформаційної системи корпоративного зберігання електронних документів. Побудовано узгоджену архітектуру системи, що включає модуль автентифікації та авторизації користувачів, модуль управління документами, підсистему контролю доступу, модуль аудиту та компонент інтеграції з базою даних. Проведено аналіз відповідності моделі вимогам за допомогою матриці трасування, що підтверджує повноту реалізації функціональних та нефункціональних характеристик.

ВИСНОВКИ

У кваліфікаційній роботі проведено комплексне дослідження, спрямоване на створення концептуальної моделі інформаційної системи корпоративного зберігання електронних документів. На основі аналізу предметної області визначено ключові проблеми сучасного корпоративного документообігу, зокрема недостатню керованість електронними документами, обмеженість механізмів контролю доступу, складність масштабування та інтеграції із внутрішніми бізнес-процесами організацій. Обґрунтовано актуальність розроблення системи, що забезпечувала б безпечне, доступне та структуроване зберігання документів.

Проведено огляд сучасних технологій, архітектурних підходів і моделей зберігання даних. На основі отриманих результатів сформовані вимоги до інформаційної системи, які враховують функціональні можливості, вимоги інформаційної безпеки, доступності та масштабованості. Визначено ролі користувачів, їх повноваження та побудовано модель доступу RBAC, що гарантує дотримання принципу мінімальних привілеїв і прозорість операцій.

Розроблено логічну модель бази даних, яка включає опис сутностей, їх атрибутів та зв’язків, що забезпечує підтримку повного життєвого циклу документа, контроль версій і можливість відновлення інформації. За допомогою UML-діаграм змодельовано структурні та поведінкові аспекти системи: архітектуру компонентів, сценарії взаємодії користувачів із системою, механізми аудиту та транзакційну обробку чутливих операцій. Це дозволило сформувати цілісне уявлення про функціонування системи та узгодити всі компоненти між собою.

Проведена верифікація моделі шляхом побудови матриці трасування вимог підтвердила відповідність розробленої системи поставленим вимогам, а також здатність запропонованої архітектури забезпечити захищене зберігання та ефективний доступ до електронних документів у корпоративному середовищі. Додатково виконано аналіз ризиків інформаційної безпеки та нормативної відповідності, що підтверджує стійкість системи до актуальних загроз та

придатність її використання в організаціях різного профілю.

Таким чином, у роботі досягнуто поставленої мети – розроблено концептуальну модель інформаційної системи корпоративного зберігання електронних документів, що забезпечує структурованість даних, керованість доступом, надійну фіксацію змін і підтримку контролю версій.

Результати дослідження можуть бути використані як методологічна та проєктна основа для подальшої програмної реалізації системи, її інтеграції з наявною IT-інфраструктурою підприємств, а також удосконалення механізмів електронного документообігу в державному й корпоративному секторах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Gartner. *Magic Quadrant for Distributed File Systems and Object Storage*. – Gartner, 2024.

2. Google. *Google Drive – Cloud Storage for Business and Personal Use*. URL: https://www.google.com/drive/

3. Dropbox. *Dropbox for Business – Secure File Sharing and Storage*. URL: https://www.dropbox.com/business

4. Microsoft. *OneDrive – Cloud Storage for Business*. URL: https://www.microsoft.com/onedrive

5. IDC. *Worldwide Cloud Storage Forecast, 2024–2028*. – IDC, 2024.

6. Codd E. F. *A Relational Model of Data for Large Shared Data Banks*. – Communications of the ACM, 1970.

7. MongoDB Inc. *What is NoSQL?*. URL: https://www.mongodb.com/nosql-explained

8. Silberschatz A., Korth H., Sudarshan S. *Database System Concepts*. – McGraw-Hill, 2020.

9. Тюрменко І.І., Грищенко М. Г. Організація архівних процесів у корпоративному середовищі. *Інформація та соціум*, 2024, 67-70.

10. Microsoft. *SharePoint*. URL: https://www.microsoft.com/sharepoint

11. Box. *Box for Business*. URL: https://www.box.com/business

12. Alfresco. *Alfresco Digital Business Platform*. URL: https://www.alfresco.com

13. OpenText. *OpenText Content Suite*. URL: https://www.opentext.com

14. Nuxeo. *Nuxeo Content Services Platform*. URL: https://www.nuxeo.com

15. SAP. *SAP Document Management*. URL: https://www.sap.com/products/document-management.html

16. Fowler M. *Patterns of Enterprise Application Architecture*. – Addison-Wesley, 2003.

17. Microsoft. *ASP.NET Core MVC*. URL: https://docs.microsoft.com/aspnet/core/mvc

18. Richardson C. *Microservices Patterns*. – Manning, 2018.

19. Gamma E., Helm R., Johnson R., Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. – Addison-Wesley, 1994.

20. Fielding R. *Architectural Styles and the Design of Network-based Software Architectures*. – University of California, 2000.

21. Pautasso C., Zimmermann O., Leymann F. *RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision*. – WWW Conference, 2008.

22. GoIT. *Ідентифікація, аутентифікація, авторизація – що означає та у чому різниця?*. URL: https://goit.global/ua/articles/identyfikatsiia-autentyfikatsiia-avtoryzatsiia-shcho-oznachaie-ta-u-chomu-riznytsia/

23. Stfalcon. *Базове розуміння OAuth 2.0*. URL: https://stfalcon.com/uk/blog/post/oauth-2-0

24. OAuth.net. *Офіційна документація OAuth 2.0*. URL: https://oauth.net/2/

25. Fwdays. *Безпечна аутентифікація SSO*. URL: https://fwdays.com/uk/event/dotnet-fwdays-2024/review/securing-sso-authentication-strategies-to-eliminate-vulnerabilities

26. LPNU. *Дисертація Балатської В.С., інтеграція SSO в корпоративні системи*. URL: https://lpnu.ua/sites/default/files/2025/radaphd/32586/disertaciyabalatskavaleriiasergiivna.pdf

27. Mezha Media. *Двофакторна автенфікація – що це та як встановити?*. URL: https://mezha.media/articles/dvofaktorna-avtentyfikatsiia-iak-vstanovyty/

28. IBM. *Що таке резервне копіювання та відновлення після збоїв*. URL: https://www.ibm.com/think/topics/backup-disaster-recovery

29. Microsoft. *Вступ до резервування, реплікації та резервного копіювання*. URL: https://learn.microsoft.com/en-us/azure/reliability/concept-redundancy-replication-backup

30. CloudEndure. *Офіційний сайт*. URL: https://www.cloudendure.com/

31. Veeam. *Veeam Backup & Replication*. URL: https://www.veeam.com/vm-backup-recovery-replication-software.html

32. ENISA. *Cybersecurity for Information Systems*. URL: https://www.enisa.europa.eu/topics/csirt-cert-services

33. Symantec. *Threat Intelligence Reports*. URL: https://www.broadcom.com/company/newsroom/press-releases

34. OWASP. *Open Web Application Security Project*. URL: https://owasp.org/

35. Microsoft. *Security best practices for cloud services*. URL: https://learn.microsoft.com/en-us/azure/security/fundamentals/overview

ДОДАТОК А. ТЕХНІЧНА СПЕЦИФІКАЦІЯ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

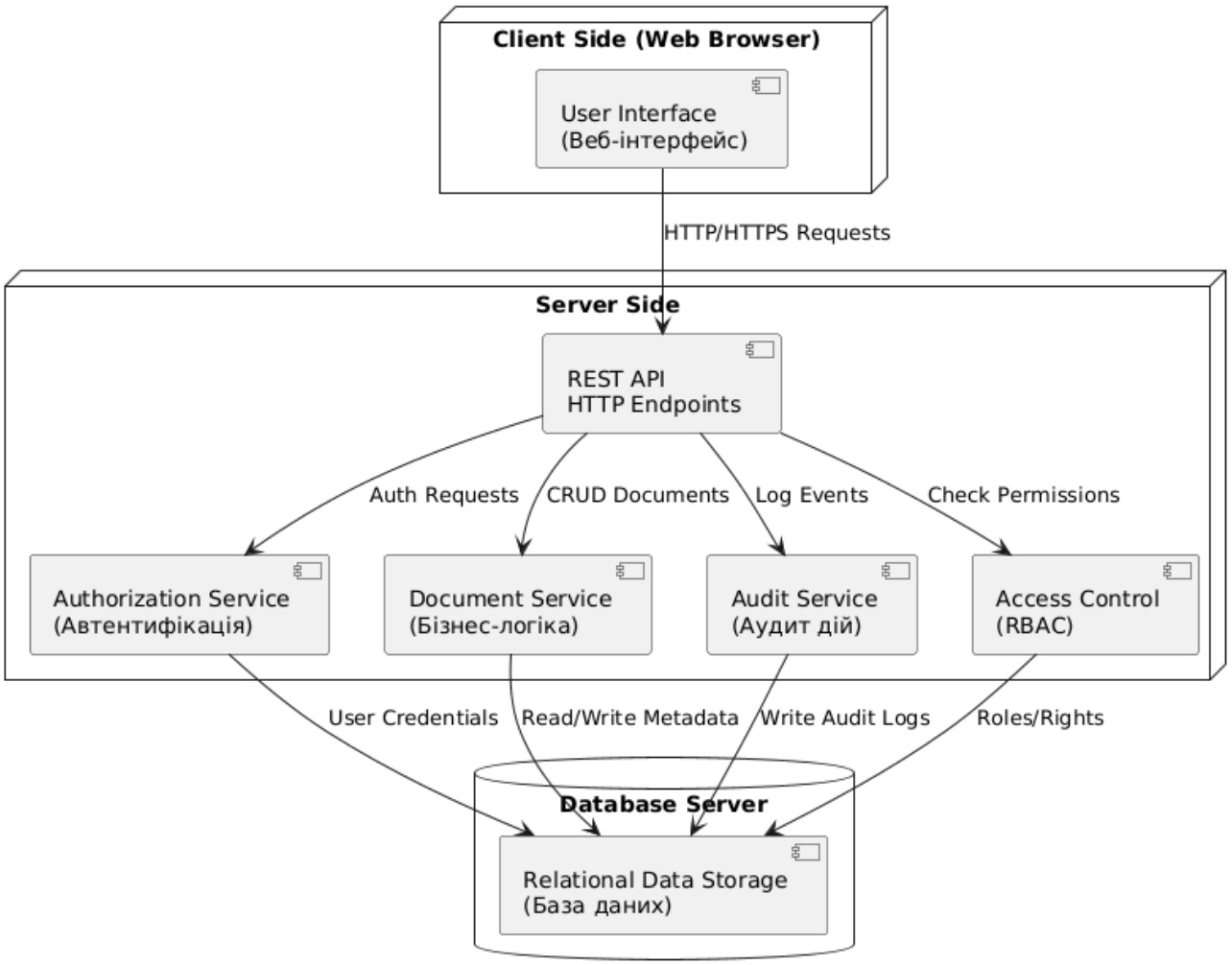

Рисунок А.1 – UML-діаграма компонентів архітектури ІС

Таблиця А.1 – Модель ролей і доступних дозволів користувачів

Роль	Перегляд документів	Додавання документів	Редагування	Видалення	Перегляд аудиту	Аналітика
Адміністратор	+	+	+	+	+	+
Менеджер	+	+	+	–	+	+
Редактор	+	+	+	–	–	–
Переглядач	+	–	–	–	–	–
Аналітик	+	–	–	–	+	+

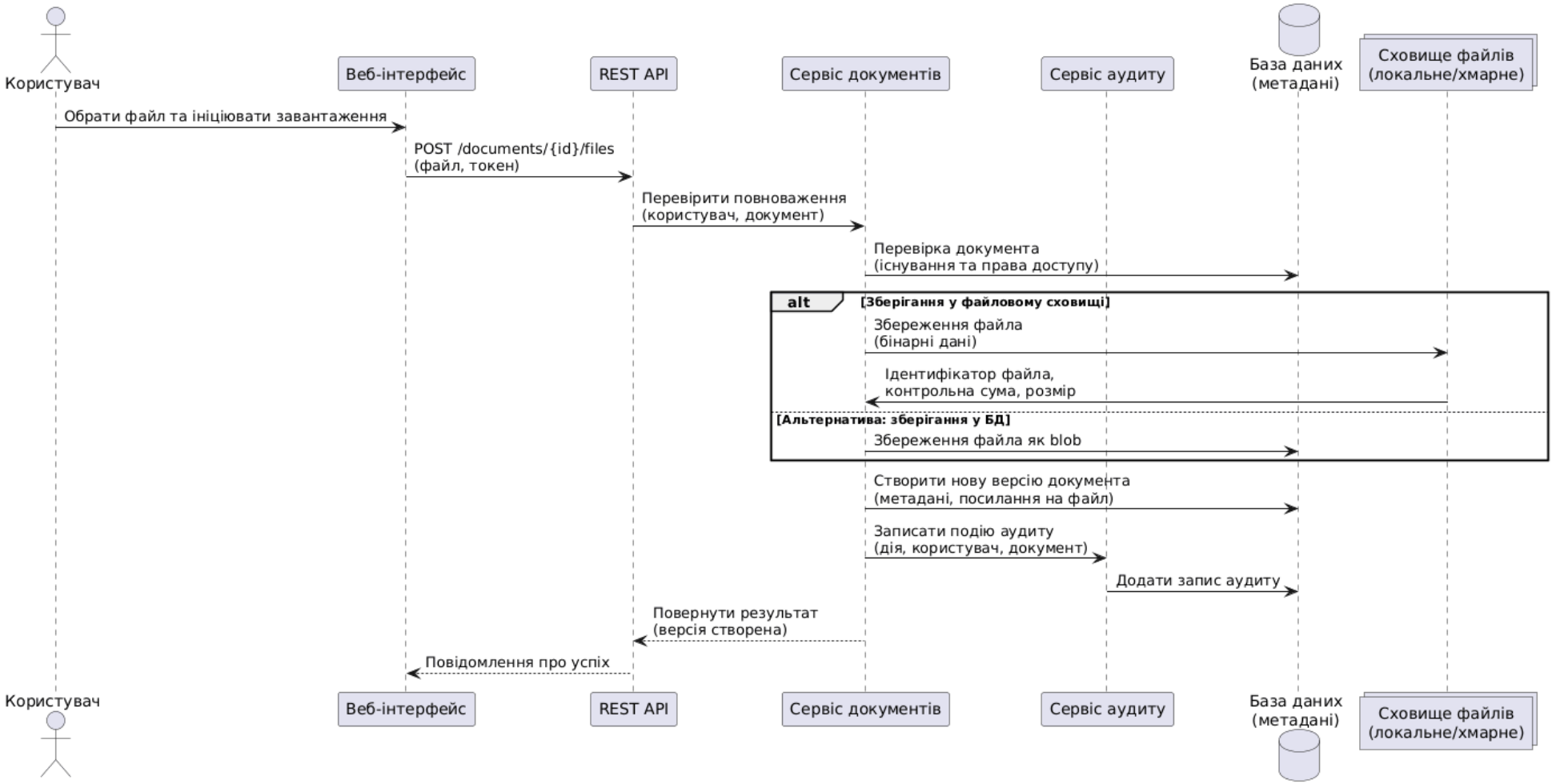

Рисунок А.2 – Діаграма послідовності: завантаження нової версії документа

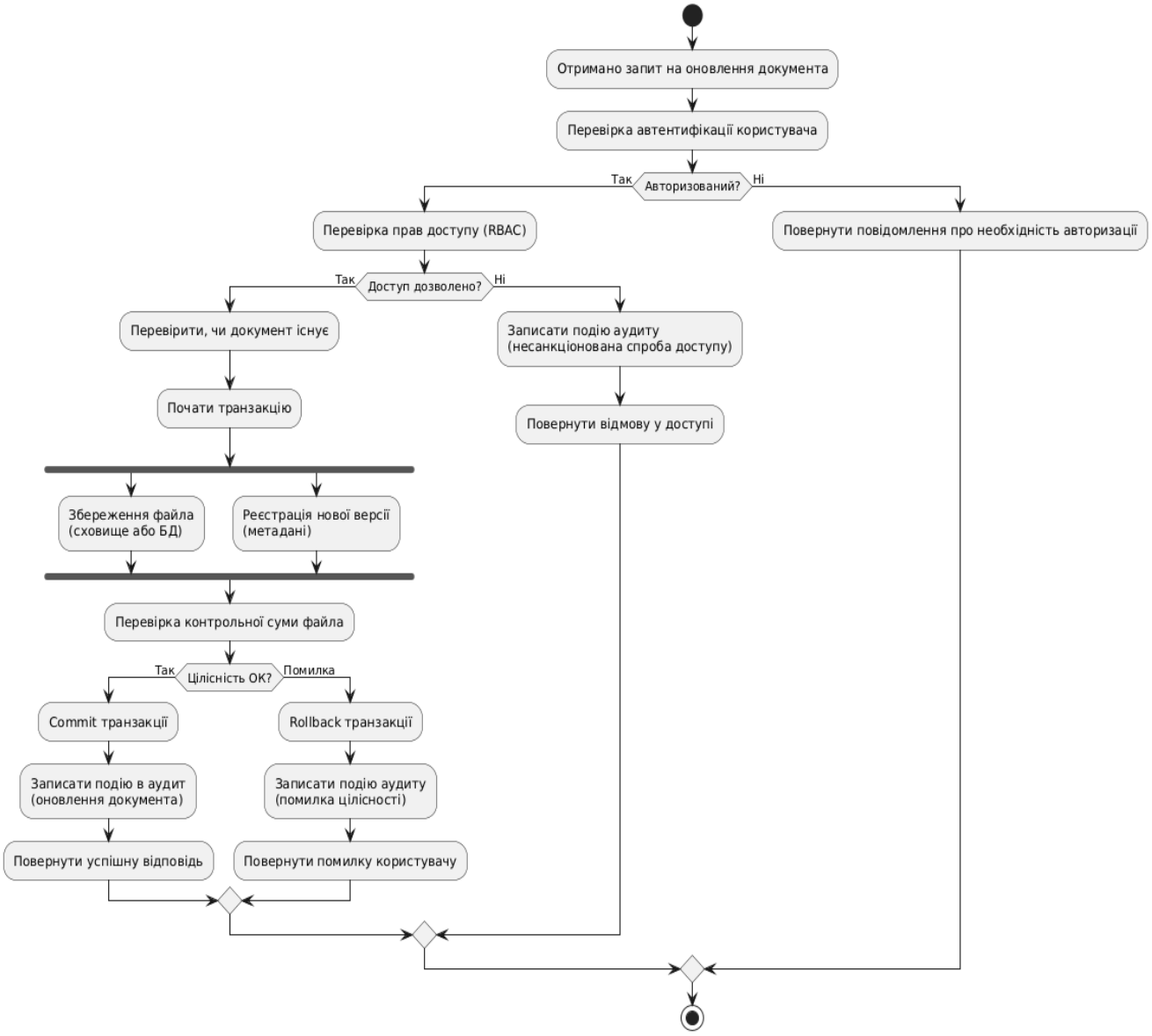

Рисунок А.3 – Діаграма діяльності транзакційної обробки під час оновлення документа

Таблиця А.2 – Таблиця ризиків та заходів захисту інформаційної системи

Ризик	**Наслідки**	**Контрзаходи**	**Реалізація в моделі**
Несанкціонований доступ	Витік документів	RBAC, аудит	Табл. А.1, рис. А.2
Пошкодження/втрата документа	Порушення бізнес-процесів	Транзакції + версії	рис. А.3
Втручання користувача	Порушення цілісності	Контроль цілісності	рис. А.3
Перевантаження системи	Недоступність	Масштабована архітектура	рис. А.1

Таблиця А.3 – Таблиця нормативної відповідності

Вимога	Стандарт	Обґрунтування виконання
Контроль доступу	ISO/IEC 27001 (A.9)	RBAC, аудит
Цілісність документів	ISO/IEC 27001 (A.12.2)	Контроль суми, транзакції
Доступність	ITIL / COBIT	Архітектурна модель
Аудит подій	GDPR Art. 30	Audit Log

Рисунок А.4 – Діаграма розгортання інформаційної системи

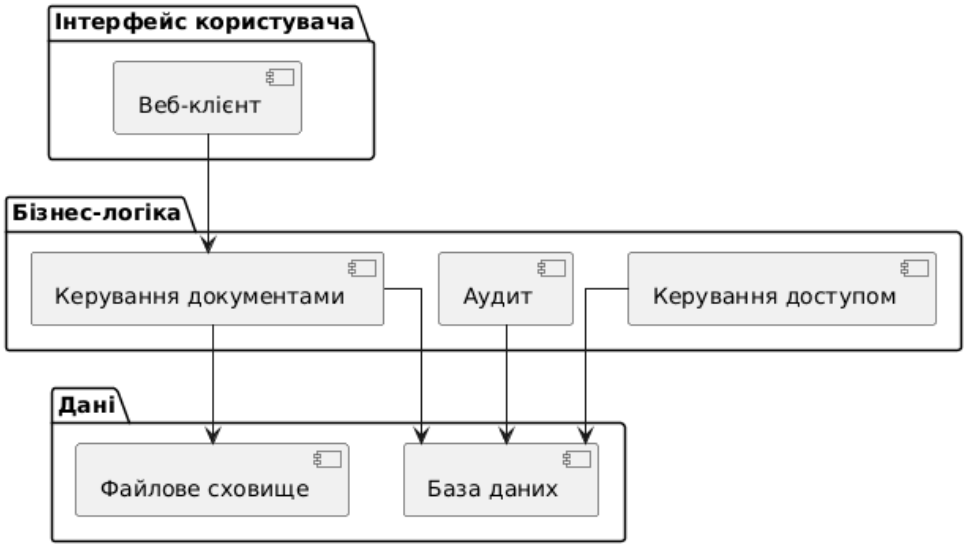

Рисунок А.5 – Діаграма пакетів інформаційної системи

ДОДАТОК Б. КОПІЇ ДОКУМЕНТІВ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ РОБОТИ

Технологічна секція

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗБЕРІГАННЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ

Автор: Торколат Володимир, магістр
Науковий керівник: Чикунов П.О., к.т.н., доц.
Національний університет «Одеська юридична академія»

Ефективне зберігання електронних документів у корпоративному середовищі вимагає забезпечення доступності, безпеки, керованості доступом і контролю версій. У зв'язку зі збільшенням обсягів даних традиційні методи організації файлового простору втрачають свою результативність. Корпоративні системи зберігання документів дозволяють централізувати управління електронними ресурсами, мінімізувати ризики втрати або несанкціонованого доступу та підвищити продуктивність співробітників.

Метою роботи є розроблення концептуальної моделі інформаційної системи корпоративного зберігання електронних документів, спрямованої на упорядкування документообігу, захист даних і підтримку інтеграційних процесів в інфраструктурі підприємства.

Завдання дослідження передбачали:

1. аналіз предметної області та існуючих систем;
2. визначення вимог до функціональності, безпеки та масштабованості;
3. проєктування моделі даних і архітектури системи;
4. побудову UML-діаграм (прецедентів, класів, компонентів, послідовностей);
5. формування ролей користувачів та політик контролю доступу.

У ході роботи було сформовано цілісну модель системи, що включає логічну архітектуру взаємодії компонентів, структуру корпоративного сховища, модель безпеки і засоби верифікації системних рішень.

Для програмної реалізації розробленої моделі запропоновано використання сучасних інструментів розробки та зберігання даних, зокрема:

– C# та .NET як основна платформа побудови серверної частини;
– REST API для організації взаємодії клієнтських додатків із сервером;
– MS SQL Server як система керування базами даних, орієнтована на цілісність і транзакційність;
– механізми автентифікації та авторизації на основі рольової моделі доступу.

Розроблена модель інформаційної системи корпоративного зберігання електронних документів ґрунтується на багаторівневій архітектурі, що включає рівень даних, бізнес-логіки та доступу користувачів. У центрі моделі знаходиться корпоративне сховище документів, представлене структурованою базою даних із підтримкою версіонування та атрибутного пошуку. Business-рівень забезпечує виконання операцій із документами відповідно до визначених бізнес-правил, а механізм рольового доступу гарантує диференційований доступ користувачів до функцій системи. Взаємодія між компонентами описується за

допомогою REST-інтерфейсу, що забезпечує можливість інтеграції з клієнтськими застосунками та зовнішніми сервісами.

Запропонована інформаційна система дозволить:

– забезпечити централізоване, надійне й безпечне зберігання корпоративних документів;

– оптимізувати бізнес-процеси шляхом автоматизації операцій документообігу;

– створити єдине середовище контролю версій документів і прав доступу;

– підвищити прозорість діяльності завдяки аудиту дій користувачів;

– забезпечити можливість масштабування рішення та його інтеграції з іншими сервісами.

Для досягнення мети дослідження запропоновано багаторівневу архітектуру системи безпеки, яка використовує технологічний стек для захисту даних на всіх рівнях системи. Візуальне представлення архітектури показано на рис. 1, де продемонстровано взаємодію між чотирма основними рівнями захисту: рівнем застосунку, рівнем даних, рівнем інфраструктури та рівнем моніторингу.

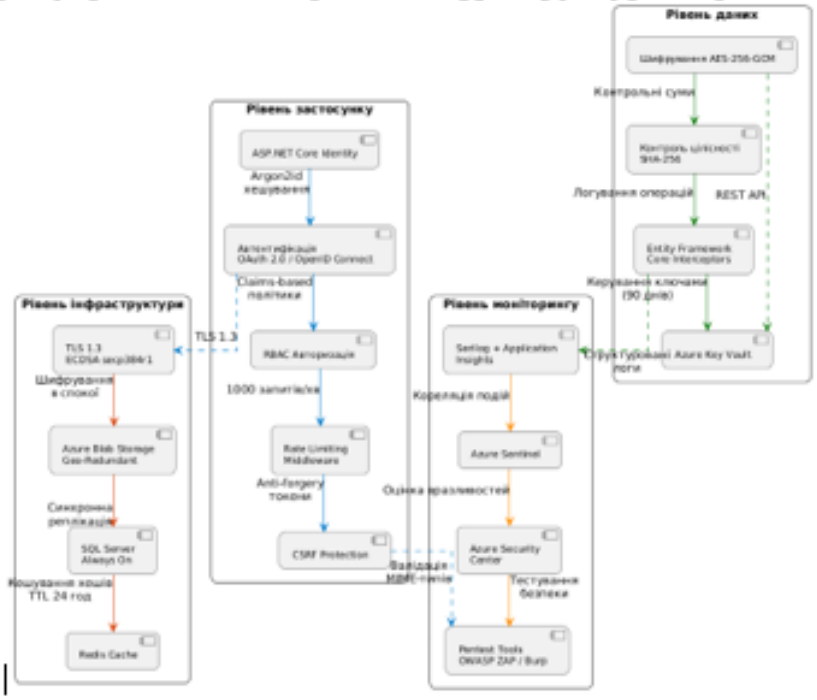

Рисунок 1 – Архітектура системи безпеки

Практична цінність роботи полягає у можливості використання розробленої моделі як основи для побудови повнофункціональної корпоративної системи електронного документообігу, що відповідає вимогам сучасного цифрового середовища організацій.

Список використаних джерел

1. Gartner. Magic Quadrant for Distributed File Systems and Object Storage. – Gartner, 2024.

2. IDC. Worldwide Cloud Storage Forecast, 2024-2028. IDC, 2024.

3. Тюрменко І.І., Грищенко М. Г. Організація архівних процесів у корпоративному середовищі. Інформація та соціум, 2024, 67-70.