

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня
магістр
на тему: “Проектування та розроблення
веб-орієнтованої системи електронного документообігу організації”

Виконав: спеціальності	студент VI курсу, групи СПмд-61 121 Інженерія програмного забезпечення Стоєв О.О.	
	_____	_____
	(підпис)	(прізвище та ініціали)
Керівник	_____	_____
	(підпис)	Бревус В.М. (прізвище та ініціали)
Нормоконтроль	_____	_____
	(підпис)	(прізвище та ініціали)
Завідувач кафедри	_____	_____
	(підпис)	Петрик М.Р. (прізвище та ініціали)
Рецензент	_____	_____
	(підпис)	(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М. Р.

(підпис)

(прізвище та ініціали)

« »

2025 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Стоєву Олександр Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування та розроблення веб-орієнтованої системи
електронного
документообігу організації

Керівник роботи Бревус Віталій Миколайович, PhD, доц. каф. ПІ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20__ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи завдання на аналіз та розробку веб-орієнтованої системи
електронного документообігу організації з розробкою прототипу системи

4. Зміст роботи (перелік питань, які потрібно розробити)
Провести аналіз предметної області електронного документообігу та виявити основні проблеми.
Виконати огляд і порівняльний аналіз існуючих систем, визначити їх переваги та недоліки.
Сформулювати функціональні та нефункціональні вимоги до системи. Розробити та обґрунтувати
архітектуру системи. Створити UML-моделі системи. Розробити логічну модель даних і
структуру бази даних системи. Спроектувати модулі системи. Провести модульне, інтеграційне
та навантажувальне тестування програмної системи, оцінити продуктивність та стійкість до
навантажень. Обґрунтувати практичну цінність розробленого програмного забезпечення
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Ілюстративна частина кваліфікаційної роботи оформляється у вигляді слайдів, висвітлює

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці			
Безпека в надз. ситуаціях			
Нормоконтроль			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз предметної області, огляд існуючих систем електронного документообігу, формування проблематики	01.09 – 10.09.2025	
2.	Формування функціональних і нефункціональних вимог	10.09 – 20.09.2025	
3.	Розроблення діаграм: прецедентів, діяльності, станів	20.09 – 30.09.2025	
4.	Проектування архітектури системи, визначення підходу до масштабування; побудова архітектурної діаграми	01.10 – 08.10.2025	
5.	Проектування моделі даних і логічної структури бази даних	08.10 – 15.10.2025	
6.	Реалізація основних модулів системи: документи, маршрути, погодження, підписання	15.10 – 10.11.2025	
7.	Реалізація модулів пошуку, аудиту та адміністративної частини	10.11 – 22.11.2025	
8.	Інтеграція модулів, модульне та інтеграційне тестування	22.11 – 01.12.2025	
9.	Навантажувальне тестування, аналіз продуктивності	01.12 – 07.12.2025	
10.	Оформлення пояснювальної записки згідно методичних рекомендацій	07.12 – 12.12.2025	
11.	Підготовка графічних матеріалів та презентації	12.12 – 14.12.2025	
12.	Передзахист, роботи	15.12.2025	
13.	Внесення правок після передзахисту, остаточне доопрацювання	15.12 – 19.12.2025	
14.	Подання завершеної магістерської роботи на кафедру	До 20.12.2025	
15.	Захист роботи	23.12.2025	

Студент

(підпис)

Стоєв О. О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Бреус В. М.

(прізвище та ініціали)

АНОТАЦІЯ

Стоєв Олександр Олександрович. Проектування та розроблення веб-орієнтованої системи електронного документообігу організації. Кваліфікаційна робота освітнього ступеня «магістр». Тернопільський національний технічний університет імені Івана Пулюя, кафедра програмної інженерії, група СПмд-61, спеціальність 121 «Інженерія програмного забезпечення». Тернопіль, 2025. Сторінок 97, таблиць 4, рисунків 15, додатків 2, бібліографічних посилань 39.

Метою роботи є проектування та розроблення веб-орієнтованої системи електронного документообігу організації, яка забезпечує автоматизацію процесів створення, зберігання, оброблення, маршрутизації та контролю станів електронних документів.

Об'єктом дослідження є процеси електронного документообігу в організаціях.

Предметом дослідження є методи, моделі та програмні засоби проектування і розроблення веб-орієнтованих систем електронного документообігу.

У процесі виконання кваліфікаційної роботи проведено аналіз предметної області та існуючих програмних рішень у сфері електронного документообігу, сформовано функціональні та нефункціональні вимоги до системи, спроектовано архітектуру програмного забезпечення та структуру бази даних. Реалізовано програмний прототип веб-орієнтованої системи, що підтримує керування користувачами та ролями, створення й оброблення документів, маршрутизацію процесів погодження, контроль станів документів і збереження історії змін.

Отримані результати можуть бути використані для автоматизації внутрішнього документообігу організацій, а також як основа для подальшого розвитку та впровадження систем електронного документообігу.

Ключові слова: інженерія програмного забезпечення, електронний документообіг, веб-орієнтована система, інформаційна система, документообіг організації, база даних.

ABSTRACT

Stoiev Oleksandr Oleksandrovych. Design and Development of a Web-Oriented Electronic Document Management System for an Organization. Master's qualification thesis. Ternopil Ivan Puluj National Technical University, Department of Software Engineering, group SPmd-61, specialty 121 "Software Engineering". Ternopil, 2025. Pages 97, tables 4, figures 15, appendices 2, references 39.

The purpose of this thesis is to design and develop a web-oriented electronic document management system for an organization, which provides automation of document creation, storage, processing, routing, and control of electronic document states.

The object of the study is the processes of electronic document management in organizations.

The subject of the study is methods, models, and software tools for the design and development of web-oriented electronic document management systems.

During the execution of the qualification thesis, the subject area and existing software solutions in the field of electronic document management were analyzed, functional and non-functional requirements for the system were defined, and the software architecture and database structure were designed. A software prototype of a web-oriented system was developed, which supports user and role management, document creation and processing, routing of approval processes, control of document states, and preservation of change history.

The obtained results can be used to automate internal document management processes in organizations, as well as serve as a basis for further development and implementation of electronic document management systems.

Keywords: software engineering, electronic document management, web-oriented system, information system, organizational document management, database.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ.....	11
1.1 Особливості предметної області електронного документообігу.....	11
1.2 Нормативно-правове забезпечення електронного документообігу	12
1.3 Аналіз існуючих програмних рішень електронного документообігу.....	13
1.4 Базові вимоги до СЕД.....	17
1.5 Функціональні вимоги до системи	17
1.6 Нефункціональні вимоги	18
1.7 Обґрунтування доцільності розробки СЕД.....	19
1.8 Сучасні підходи до проєктування веб-орієнтованих інформаційних систем	20
2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ.....	23
2.1 Вибір та опис методології проєктування	23
2.2 Проєктування та моделювання вимог	26
2.3 Виявлення класів сутності.....	29
2.4 Визначення класів системи	31
2.5 Побудова діаграми діяльності бізнес-процесу	33
2.6 Побудова діаграми стану бізнес-процесу	35
2.7 Опис роботи системи	37
2.8 Архітектура системи електронного документообігу.....	40
2.9 Вибір мов програмування, технологій та середовища розробки	41
2.10 Реалізація серверної частини системи	42
3 ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ПІДТРИМКА СИСТЕМИ.....	52
3.1 Тестування в життєвому циклі системи.....	52
3.2 Цілі та завдання тестування системи	53
3.3 Види та класифікація тестування системи.....	55
3.4 Тестове середовище та умови проведення тестування.....	57
3.5 План та стратегія тестування системи.....	59
3.6 Тестування реалізації життєвого циклу електронного документа.....	62
3.7 Тестування основних бізнес-процесів системи.....	66
3.8 Аналіз результатів тестування системи	69
3.9 Впровадження системи в експлуатацію.....	71
3.10 Підтримка та супровід системи	74

4	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ...	78
4.1	Охорона праці	78
4.2	Характеристика робочого місця користувача	79
4.3	Вимоги до електробезпеки	80
4.4	Пожежна безпека	81
4.5	Ергономіка та організація робочого місця.....	82
	ВИСНОВКИ	84
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87
	ДОДАТОК А.....	92
	ДОДАТОК В	93

ВСТУП

Стрімка цифрова трансформація суспільства створює нові вимоги до організації управлінських процесів, підвищення їх прозорості, оперативності та ефективності. Одним із ключових інструментів модернізації діяльності організацій, зокрема державних і приватних установ, є впровадження систем електронного документообігу (далі – СЕД). Вони забезпечують автоматизацію роботи з документами, скорочують витрати часу на їх обробку, підвищують якість взаємодії між підрозділами та зменшують ризики, пов'язані з людським фактором.

Традиційні паперові процеси управління документацією характеризуються низькою швидкістю обробки, труднощами відстеження стану документів, високими затратами на зберігання та логістичний супровід. У свою чергу, перехід до електронного формату роботи дозволяє оптимізувати бізнес-процеси, забезпечити централізоване зберігання інформації, підвищити доступність даних, впровадити засоби аудиту та контролю виконання, а також дотриматися вимог законодавства у сфері інформаційної безпеки.

Правову основу для функціонування СЕД формують нормативні акти України: Закон України «Про електронні документи та електронний документообіг», Закон України «Про електронну ідентифікацію та електронні довірчі послуги», Закон України «Про захист інформації в інформаційно-комунікаційних системах», постанова Кабінету Міністрів України від 17 січня 2018 року № 55 «Деякі питання документування управлінської діяльності». Ці документи визначають статус електронного документа, правила накладання електронного підпису та вимоги до обробки інформації, що забезпечує юридичну значимість та безпечність електронного документообігу.

У межах проєкту особливу увагу приділено створенню веб-орієнтованої системи, здатної забезпечити повний життєвий цикл документа – від створення та погодження до підписання та архівування. Реалізація системи передбачає використання сучасних підходів до моделювання процесів, організації структури

даних, забезпечення пошуку документів, підтримки електронного підпису та реалізації механізмів безпеки.

Об'єкт дослідження – процес електронного документообігу в організації.
Предмет дослідження – методи, моделі та інструменти проєктування і розроблення веб-орієнтованої системи електронного документообігу.

Метою роботи є проєктування та розроблення веб-орієнтованої системи електронного документообігу, яка забезпечує автоматизацію роботи з документами, підтримує маршрути погодження, надає засоби електронного підписання, ведення журналів дій та захисту інформації відповідно до вимог сучасних технологічних та нормативних стандартів.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Проаналізувати предметну область і нормативно-правову базу електронного документообігу.
2. Дослідити існуючі СЕД і визначити їхні взаємні переваги та недоліки.
3. Сформулювати функціональні та нефункціональні вимоги до системи.
4. Виконати проєктування архітектури веб-орієнтованої СЕД.
5. Розробити інформаційну модель системи.
6. Провести тестування системи.
7. Оцінити ефективність реалізованої системи та можливості її подальшого розвитку.

Наукова новизна роботи полягає у розробленні комплексної моделі веб-орієнтованої СЕД, що поєднує:

- підтримку представлення структури документів, їхніх версій та маршрутів погодження;
- інтеграцію механізмів електронного підпису;
- реалізацію динамічних процесів погодження документів;
- застосування підходів повнотекстового індексування для ефективного пошуку документів;
- поєднання рішень у рамках клієнт-серверної архітектури з урахуванням вимог інформаційної безпеки.

Отже, робота пропонує підхід до побудови СЕД, який поєднує сучасні концепції організації документообігу з вимогами українського законодавства, що робить систему практично орієнтованою та адаптованою до потреб організацій різного масштабу.

Практичне значення полягає у створенні прикладного рішення, яке може бути впроваджене в організаціях для оптимізації внутрішнього документообігу, зменшення трудовитрат, підвищення прозорості управлінських процесів і забезпечення надійного захисту інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ

Документообіг є ключовим елементом управлінської діяльності будь-якої організації. Він охоплює створення, реєстрацію, розподіл, погодження, виконання, підписання, передачу, зберігання та архівування документів.

1.1 Особливості предметної області електронного документообігу

Ефективність документообігу безпосередньо впливає на швидкість прийняття рішень, прозорість бізнес-процесів, якість внутрішньої комунікації та рівень відповідальності працівників.

Традиційний паперовий документообіг характеризується рядом недоліків [29, 30]:

- тривалий час обробки та погодження документів;
- ризик втрати або пошкодження документів;
- дублювання інформації;
- відсутність можливості оперативного контролю виконання;
- значні матеріальні та трудові витрати;
- складнощі в організації доступу до документів з різних підрозділів та віддалених робочих місць.

Це приводить організації до необхідності впровадження електронного документообігу – автоматизованої технології обробки документів із застосуванням інформаційних систем.

Електронний документообіг забезпечує:

- централізоване та захищене зберігання даних;
- автоматизацію маршрутизації документів;
- підтримку електронного підпису;
- можливість повнотекстового пошуку;
- контроль усіх етапів життєвого циклу документа;

- аудит дій користувачів;
- скорочення строків опрацювання документів.

Отже, СЕД є не лише інструментом оптимізації внутрішніх процесів, а й засобом підвищення прозорості управління, якості діловодства та інформаційної безпеки.

1.2 Нормативно-правове забезпечення електронного документообігу

Створення та впровадження СЕД повинно відповідати чинному нормативно-правовому полю, яке встановлює вимоги до електронних документів, електронного підпису, захисту інформації, організації діловодства та використання інформаційних систем.

Основні нормативні документи, що регламентують електронний документообіг [1, 2, 3, 4]:

- Закон України «Про електронні документи та електронний документообіг», який визначає правовий статус електронного документа, його структуру, реквізити, правила створення, обігу, перевірки та зберігання. Електронний документ юридично прирівнюється до паперового у разі наявності кваліфікованого електронного підпису.
- Закон України «Про електронні довірчі послуги», який регламентує використання кваліфікованого електронного підпису (далі – КЕП).
- Закон України «Про захист інформації в інформаційно-комунікаційних системах», який визначає вимоги до побудови, функціонування та захисту інформаційних систем, у тому числі СЕД, обробки службової та конфіденційної інформації.
- Постанова Кабінету Міністрів України від 17 січня 2018 року № 55 «Деякі питання документування управлінської діяльності», яка встановлює правила документування управлінської діяльності, ведення діловодства в державних органах та визначає вимоги до електронного документообігу, а також якою затверджено [4]:

- Типову інструкцію з документування управлінської інформації в електронній формі та організації роботи з електронними документами в діловодстві, електронного міжвідомчого обміну;
- Типову інструкцію з діловодства в міністерствах, інших центральних та місцевих органах виконавчої влади;
- Регламент організації взаємодії органів виконавчої влади в електронній формі.

Відповідно до пункту 2 розділу I Типової інструкції з документування управлінської інформації в електронній формі та організації роботи з електронними документами в діловодстві, електронного міжвідомчого обміну основна форма провадження діловодства в установах є електронна. Документування управлінської інформації в установах здійснюється в електронній формі із застосуванням кваліфікованого електронного підпису, кваліфікованої електронної печатки та кваліфікованої електронної позначки часу, крім випадків наявності обґрунтованих підстав для документування управлінської інформації у паперовій формі, якими визнаються:

- документи, що містять інформацію з обмеженим доступом, вимога щодо захисту якої встановлена законом;
- електронні документи, що не можуть бути застосовані як оригінал згідно з вимогами закону;
- документи, вимога щодо опрацювання яких у паперовій формі встановлена актами Кабінету Міністрів України.

Врахування цих нормативно-правових актів є обов'язковою умовою під час проектування та розроблення СЕД, що гарантує юридичну значимість документів і належний рівень інформаційної безпеки.

1.3 Аналіз існуючих програмних рішень електронного документообігу

Ринок програмних продуктів для електронного документообігу представлений широким спектром рішень, що мають певні переваги та обмеження.

Аналіз найбільш поширених систем дозволяє виділити тенденції та недоліки, які потребують усунення в рамках створення власної СЕД.

У рамках дослідження доцільності розробки СЕД проведено аналіз найбільш розповсюджених в Україні програмних рішень: Megapolis.DocNet, АСКОД та М.Е.Дос Документообіг [6, 7, 8]. Ці системи використовуються як у державному, так і в приватному секторі, але мають різні функціональні призначення, технологічні особливості та рівень відповідності нормативним вимогам.

Megapolis.DocNet – корпоративна СЕД, створена на базі платформи Microsoft SharePoint. Вона орієнтована на великі державні установи й організації з великою кількістю користувачів.

Характеристики:

- працює у середовищі Microsoft Windows Server, використовує SQL Server або Oracle як систему керування базами даних (далі – СКБД);
- веб-інтерфейс (тонкий клієнт), доступ через браузер (Edge, Chrome);
- інтегрується з Active Directory, Microsoft Office 2019+, підтримує КЕП;
- підтримує складні маршрути погодження, архівацію, контроль виконання та журналювання дій;
- сертифікована відповідно до вимог Державної служби спеціального зв'язку та захисту інформації України.

Переваги:

- повна інтеграція з Microsoft-екосистемою;
- висока надійність і продуктивність;
- розвинена аналітика і звітність;
- відповідність державним стандартам захисту інформації.

Недоліки:

- висока вартість ліцензії, впровадження та супроводу;
- залежність від платформи Microsoft;
- складність кастомізації без участі розробника;
- високі вимоги до інфраструктури;

АСКОД – програмний комплекс електронного документообігу.

Характеристики:

- побудована на Java або .NET, працює на операційних системах Microsoft Windows або Linux.

- СКБД: Oracle, PostgreSQL або Microsoft SQL Server.
- доступна як через десктопний клієнт, так і через веб-інтерфейс.
- підтримує КЕП, інтеграцію з Дія ID, СЕВ ОБВ, електронною поштою.
- відповідає вимогам українського законодавства щодо документообігу.

Переваги:

- орієнтованість на державний сектор;
- повна відповідність нормативам України;
- гнучка архітектура маршрутизації документів;
- помірна вартість впровадження;
- наявність української підтримки та документації.

Недоліки:

- менш сучасний інтерфейс у порівнянні з Megapolis.DocNet;
- обмежена підтримка мобільних пристроїв;
- більш складна конфігурація прав доступу.

М.Е.Дос Документообіг – модуль документообігу в рамках бухгалтерського програмного комплексу М.Е.Дос. Основне призначення – обмін юридично значущими документами з контрагентами.

Характеристики:

- локальний Windows-додаток, який використовує Microsoft SQL Server;
- призначений для створення, підписання і надсилання документів (рахунки, акти, договори);
- інтегрується з програмним забезпеченням BAS, сервісами податкової звітності;
- підтримує КЕП і архівування документів.

Переваги:

- доступність і низька вартість;
- простота в освоєнні та впровадженні;
- висока швидкість обміну з зовнішніми контрагентами;
- інтеграція з бухгалтерськими системами.

Недоліки:

- обмежена маршрутизація і контроль виконання.
- відсутність веб-інтерфейсу і мобільної версії.
- не придатна для масштабного використання у великих організаціях.

Отже, проведене порівняння показує, що кожне із зазначених рішень має своє цільове призначення та сферу ефективного застосування:

Megapolis.DocNet – ідеальне рішення для великих організацій, які вже мають комплексну Microsoft-інфраструктуру, але воно є дорогим і менш гнучким у кастомізації.

АСКОД – найоптимальніше рішення для державних органів, яке відповідає законодавчим та технічним вимогам, має достатню гнучкість і підтримку українською мовою.

М.Е.Дос Документообіг – корисний інструмент для малого бізнесу та бухгалтерій, проте не придатний як основна система для внутрішнього документообігу ЦОВВ.

З огляду на це, розробка власної СЕД на базі сучасних веб-технологій виглядає технічно доцільною та стратегічно обґрунтованою альтернативою. Такий підхід дозволить створити рішення, яке буде:

- повністю відповідати функціональним і нефункціональним вимогам;
- інтегруватися з наявною інфраструктурою;
- адаптуватися до специфіки документообігу конкретної організації.

Це створює підстави для розроблення нової веб-орієнтованої СЕД, яка усуне виявлені недоліки та відповідатиме вимогам сучасної організації.

1.4 Базові вимоги до СЕД

СЕД повинна функціонувати як централізована система з використанням єдиної бази даних СЕД всіх підрозділів організації. Для кожного підрозділу встановлюється свій рівень доступу до документів та інформації, що зберігається в СЕД.

Підтримка процесів документообігу компонентами СЕД повинна реалізовуватися на всіх стадіях життєвого циклу документів – від моменту створення або надходження до відправлення адресату.

До об'єктів автоматизації, де передбачається використання СЕД, відносяться структурні підрозділи організації та її працівники.

Базове програмне забезпечення СЕД повинне забезпечувати можливість інформаційної взаємодії з системами електронного документообігу інших органів державної влади, органами місцевого самоврядування та іншими установами через систему електронної взаємодії органів виконавчої влади (далі – СЕВ ОВВ).

СЕД повинна забезпечувати зберігання та захист інформації, оперативний доступ до неї з урахуванням розмежування прав доступу для всіх користувачів СЕД.

СЕД повинна складатися з сукупності функціональних підсистем та модулів, які підтримують автоматизацію процесів роботи з документами на всіх етапах їхнього життєвого циклу [9, 10].

1.5 Функціональні вимоги до системи

Управління документами [9, 10]:

- створення, редагування, перегляд документів;
- завантаження файлів у форматах PDF, DOCX, XLSX;
- версіонування документів;
- ведення метаданих.

Маршрутизація та погодження:

- створення гнучких маршрутів руху документа;
- погодження на різних етапах;
- фіксація результатів погодження.

Підписання документів:

- накладання кваліфікованого електронного підпису;
- перевірка дійсності підпису;
- збереження інформації про підписантів.

Пошук:

- пошук за реквізитами документа;
- повнотекстовий пошук за витягнутим контентом;
- фільтрування та сортування.

Доступи та ролі:

- рольова модель доступу;
- обмеження прав (читання, редагування, погодження, підписання).

Аудит:

- фіксація усіх дій користувачів;
- формування журналів.

1.6 Нефункціональні вимоги

Безпека [11, 12]:

- автентифікація користувачів;
- шифрування даних;
- контроль доступу;
- захист від несанкціонованих дій.

Продуктивність:

- кількість одночасних користувачів до 200;
- кількість документів до 1000 документів на місяць;
- відкриття документу ≤ 3 с.;

- швидка індексація для пошуку.

Надійність:

- резервні копії,
- відновлення після збоїв.

Масштабованість;

- можливість розширення функцій;
- підтримка збільшення обсягу даних.

Сумісність:

- робота в сучасних браузерях;
- розгортання на Windows;
- відкритий стек технологій.

Юзабіліті:

- простий інтерфейс;
- логічна навігація;
- адаптивність.

1.7 Обґрунтування доцільності розробки СЕД

Актуальність розробки обумовлена загальнодержавною стратегією цифрової трансформації, зростаючою необхідністю забезпечення прозорості, оперативності та ефективності документообігу.

На підставі аналізу чинного законодавства та технічних можливостей встановлено, що розробка власної СЕД має низку переваг у порівнянні з використанням наявних програмних рішень. Зокрема, такі системи як Megapolis.DocNet, АСКОД чи М.Е.Дос мають певні обмеження у масштабованості, гнучкості, підтримці веб-інтерфейсів або надмірно високу вартість ліцензій. Це робить їх менш придатними для використання в невеликих організаціях без адаптації під специфіку установи [29, 30].

Розробка власного веб-орієнтованого рішення дозволяє врахувати конкретні потреби організації, а також зменшити витрати на підтримку, ліцензії та адаптацію.

Отже, в першому розділі, у результаті проведеного аналізу, визначено сучасний стан ринку електронного документообігу, його проблеми, нормативні вимоги та технологічні тенденції. Виявлено, що існуючі системи мають суттєві обмеження щодо гнучкості, сучасності інтерфейсу, адаптивності та вартості. На основі цього сформульовано вимоги до веб-орієнтованої СЕД, яка має забезпечувати автоматизацію життєвого циклу документа, підтримку маршрутизації, підписання, аудит, пошук та високий рівень безпеки. Отримані результати є основою для подальшого проєктування та розроблення системи.

1.8 Сучасні підходи до проєктування веб-орієнтованих інформаційних систем

Сучасні інформаційні системи, зокрема СЕД, характеризуються високим рівнем складності бізнес-логіки, наявністю регламентованих процесів і жорсткими вимогами до коректності обробки даних. У таких умовах традиційні підходи до розроблення програмного забезпечення, за яких специфікація використовується лише як описова документація, не завжди забезпечують належний рівень узгодженості між вимогами, реалізацією та тестуванням. Це зумовлює актуальність використання підходів, орієнтованих на формальну специфікацію поведінки системи.

Одним із таких підходів є Spec-Driven Development – методологія розроблення програмного забезпечення, за якої формальна специфікація системи виступає первинним артефактом і визначає допустиму поведінку програмного продукту. На відміну від традиційних процесів, у яких специфікація часто відокремлена від програмної реалізації, у Spec-Driven Development вона використовується як основа для проєктування, реалізації та подальшої перевірки коректності системи [35].

Ідеологія Spec-Driven Development полягає у тому, що програмна реалізація повинна строго відповідати заздалегідь визначеній специфікації. Така специфікація може бути подана у різних формах: формальних контрактів, описів станів, сценаріїв використання або моделей поведінки. У сучасних підходах до розроблення програмного забезпечення все частіше використовуються формалізовані моделі, які дозволяють не лише описати структуру системи, а й зафіксувати правила її динамічної поведінки.

У контексті СЕД застосування Spec-Driven Development є особливо доцільним, оскільки такі системи мають чітко визначений життєвий цикл документа. Документ у процесі обробки проходить низку станів – створення, реєстрацію, погодження, підписання, архівування – між якими допускаються лише строго визначені переходи. Порухення логіки цих переходів може призвести до втрати контролю над процесом документообігу або невідповідності нормативним вимогам.

У межах даної магістерської роботи принципи Spec-Driven Development реалізуються шляхом використання UML як мови формальної специфікації поведінки системи. UML-діаграми у роботі виконують не лише ілюстративну функцію, а й виступають засобом фіксації правил функціонування системи. Зокрема, діаграма станів електронного документа визначає повний перелік допустимих станів документа та умови переходу між ними. Ця діаграма фактично виконує роль поведінкової специфікації життєвого циклу документа.

Діаграми діяльності використовуються для формалізації бізнес-процесів електронного документообігу, зокрема процесів погодження та маршрутизації документів. Вони задають послідовність дій, умови виконання та альтернативні сценарії розвитку процесів. Отже, діаграми діяльності виконують роль процесної специфікації, яка визначає допустиму логіку виконання операцій у системі.

Діаграма прецедентів, у свою чергу, формалізує межі функціональності системи та визначає взаємодію користувачів із системою відповідно до їх ролей. Вона дозволяє зафіксувати перелік допустимих дій для кожного типу користувачів

і тим самим виступає специфікацією доступу до функціональних можливостей системи.

Використання UML-моделей у ролі специфікацій дозволяє забезпечити узгодженість між етапами проектування та реалізації. Програмна логіка серверної частини системи реалізується з урахуванням обмежень, визначених UML-діаграмами, зокрема щодо допустимих станів документа та переходів між ними. Це означає, що будь-яка операція над документом у системі може бути співвіднесена з відповідним елементом специфікації.

Окрему роль у Spec-Driven Development відіграє тестування, яке розглядається як засіб перевірки відповідності реалізації специфікації. У даній роботі тестові сценарії формуються на основі UML-діаграми станів і діаграм діяльності. Це дозволяє перевірити, що програмна система не допускає недопустимих переходів між станами документа та коректно реалізує визначені бізнес-процеси. Такий підхід забезпечує тісний зв'язок між специфікацією та процесом тестування.

Отже, у межах магістерської роботи Spec-Driven Development реалізується як наскрізний підхід, що охоплює аналіз предметної області, проектування, програмну реалізацію та тестування системи. Формалізовані UML-моделі виступають єдиним джерелом істини щодо поведінки системи, а програмна реалізація та тестування підтверджують дотримання цієї специфікації.

Застосування Spec-Driven Development у розробленні веб-орієнтованої СЕД дозволяє підвищити керованість процесу розроблення, зменшити кількість логічних помилок та забезпечити прозорість взаємозв'язку між вимогами, проєктними рішеннями і програмною реалізацією. Це створює методологічне підґрунтя для подальшого розвитку системи та її адаптації до змін бізнес-процесів і нормативних вимог.

2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

Методології розробки програмного забезпечення – це набір принципів і практик, які регулюють процес створення програмного забезпечення, починаючи від ідеї та закінчуючи випуском (впровадженням) такого програмного забезпечення та забезпечення його підтримки.

Існують різні методології до розроблення програмного забезпечення, кожна з яких має свої принципи, переваги та обмеження.

2.1 Вибір та опис методології проєктування

На сьогодні існує багато методологій розробки програмного забезпечення, з яких можна виділити наступні [13, 14]:

- лінійні або традиційні методології;
- гнучкі методології.

Лінійні методології розробки програмного забезпечення це підходи до управління розробкою, що передбачають сувору послідовність етапів і мінімальну гнучкість у зміні вимог під час реалізації.

Характеристики лінійних методологій:

- послідовність етапів: проєкт проходить чітко визначені фази (наприклад, аналіз, проєктування, реалізація, тестування, підтримка);
- фіксовані вимоги: усі вимоги до продукту визначаються на початку і залишаються незмінними;
- документованість: кожен етап супроводжується детальною документацією;
- жорсткий контроль: використовується планування й контроль часу, бюджету та ресурсів.

Найвідоміший приклад такої методології – модель водоспаду (Waterfall), що являє собою класичний послідовний підхід: завершення одного етапу передбачає перехід до наступного.

Основні фази моделі водоспаду:

- аналіз вимог;
- проєктування;
- розробка;
- тестування;
- впровадження;
- підтримка.

Гнучкі методології розробки програмного забезпечення (Agile) це підхід до управління проєктами, який зосереджений на адаптивному плануванні, постійній ітерації, командній співпраці та швидкому реагуванні на зміни.

Основні цінності Agile:

- люди та взаємодія важливіші за процеси та інструменти. Основна увага приділяється команді, її співпраці та взаємодії;
- робочий продукт важливіший за документацію. Програмний продукт має бути функціональним і відповідати потребам клієнта;
- співпраця з клієнтом важливіша за контрактні зобов'язання. Постійний зворотний зв'язок допомагає досягти кращого результату;
- реагування на зміни важливіше за дотримання початкового плану.

Гнучкість дозволяє швидко адаптуватися до змін у вимогах.

Серед популярних гнучких методологій можна виділити наступні:

- SCRUM;
- KANBAN;
- Extreme Programming (XP);
- Rational Unified Process (RUP);
- Dynamic Systems Development Method (DSDM);
- Rapid application development (RAD).

Для цілей цього проєкту обрано методологію Rational Unified Process (Раціональний уніфікований процес), яка 4 (чотири) фази проєктування.

Початок (Insertion), яка спрямована на визначення цілей, призначення та меж проєкту. Основні завдання включають:

- аналіз вимог;
- ідентифікацію ризиків;
- визначення доцільності проєкту.

Проектування (Elaboration), яка спрямована на створення чіткого розуміння вимог і архітектури. Основні завдання включають:

- деталізацію вимог;
- створення моделі архітектури.

Побудова (Construction), яка фокусується на побудові системи на основі визначеної архітектури та вимог. Основні завдання включають:

- програмування;
- тестування;
- інтеграцію компонентів.

Впровадження (Transition), яка передбачає підготовку системи до впровадження. Основні завдання включають:

- фінальне тестування;
- навчання користувачів;
- створення документації.

Система стає доступною кінцевим користувачам, і виправляються останні недоліки.

Після завершення проводиться аналіз результатів виконання всього проєкту.

Відповідно до цієї методології розробка проходить ітеративно, що дозволяє поступово вдосконалювати продукт протягом усього життєвого циклу програмного забезпечення.

2.2 Проєктування та моделювання вимог

Для цілей проєктування згідно з RUP застосуємо уніфіковану мову моделювання UML, яка використовується для графічного опису об'єктно-орієнтованих систем. Вона широко застосовується в галузі розробки програмного забезпечення, моделювання бізнес-процесів, системного проєктування, а також для візуалізації організаційних структур [16, 17].

UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, так званої UML-моделлю. UML не є мовою програмування, але на підставі моделей UML можлива генерація коду.

На основі технології UML будується єдина інформаційна модель. UML підтримує етапи життєвого циклу інформаційної системи і для цього надає графічні засоби – діаграми.

На етапі створення концептуальної моделі для опису діяльності використовуються діаграма прецедентів.

Діаграма прецедентів (діаграма варіантів використання) в UML – це діаграма, що відображає відносини між акторами та прецедентами і є складовою моделі прецедентів, що дозволяє описати систему на концептуальному рівні.

Керуючись вимогами до СЕД влади можна визначити наступний перелік акторів та варіанти використання

Акторами системи:

- Користувач – створює/переглядає документи.
- Погоджувач – розглядає та погоджує/відхиляє документи.
- Підписант – накладає електронний підпис.
- Архіваріус – виконує пошук і архівування документів.
- Адміністратор – керує користувачами, ролями та налаштуваннями системи.

Варіанти використання (прецеденти) системи:

- UC1 – Створити документ.

- UC2 – Зареєструвати документ.
- UC3 – Надіслати на погодження.
- UC4 – Погодити / Відхилити документ.
- UC5 – Накласти електронний підпис.
- UC6 – Переглянути документ.
- UC7 – Пошук документів.
- UC8 – Архівувати документ.
- UC9 – Управління користувачами та ролями.
- UC10 – Перегляд журналу подій.

Лістинг 2.1 – Програмний UML код діаграми варіантів використання (прецедентів):

```
@startuml
left to right direction
actor "Користувач" as User
actor "Погоджувач" as Approver
actor "Підписант" as Signatory
actor "Архіваріус" as Archivist
actor "Адміністратор" as Admin
rectangle "Система електронного документообігу" {
    usecase "UC1: Створити документ" as UC1
    usecase "UC2: Зареєструвати документ" as UC2
    usecase "UC3: Надіслати на погодження" as UC3
    usecase "UC4: Погодити/Відхилити документ" as UC4
    usecase "UC5: Накласти електронний підпис" as UC5
    usecase "UC6: Переглянути документ" as UC6
    usecase "UC7: Пошук документів" as UC7
    usecase "UC8: Архівувати документ" as UC8
    usecase "UC9: Управління користувачами та ролями" as UC9
    usecase "UC10: Перегляд журналу подій" as UC10
}
User --> UC1
User --> UC2
User --> UC3
User --> UC6
User --> UC7
Approver --> UC3
Approver --> UC4
Approver --> UC6

Signatory --> UC5
Signatory --> UC6
Archivist --> UC6
```

```

Archivist --> UC7
Archivist --> UC8
Admin --> UC9
Admin --> UC10
Admin --> UC7
UC3 ..> UC2 : «include»
UC5 ..> UC4 : «include»
@enduml

```

Відповідно до програмного UML коду діаграми варіантів використання (прецедентів) побудуємо діаграму прецедентів (діаграма прецедентів системи СЕД відображена на рисунку 2.1).

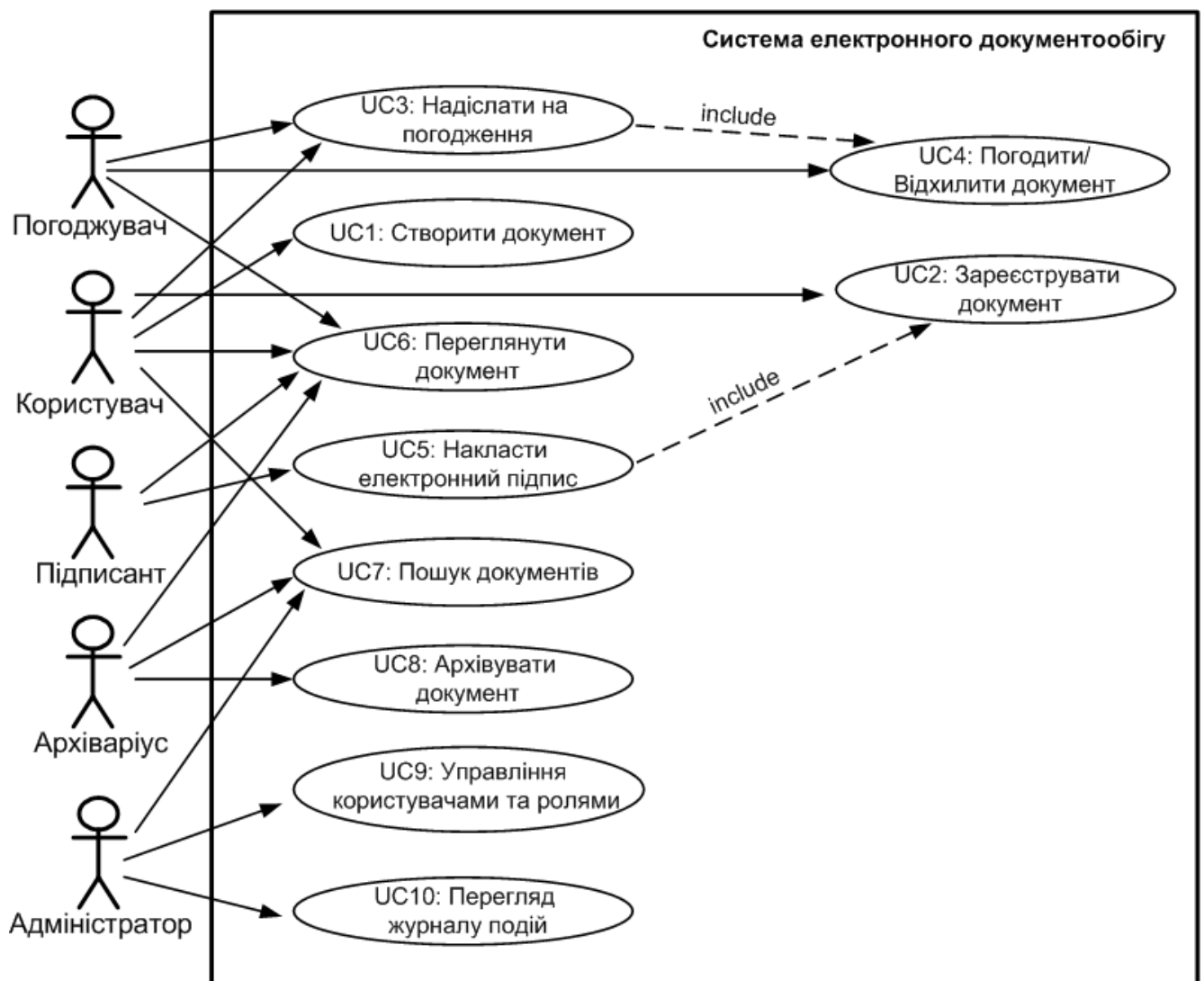


Рисунок 2.1 – Діаграма прецедентів системи СЕД

2.3 Виявлення класів сутності

У процесі проектування веб-орієнтованої СЕД одним із ключових етапів є виявлення класів сутності, які відображають основні об'єкти предметної області та формують основу інформаційної моделі системи. Виявлення таких класів здійснюється на основі аналізу функціональних вимог, варіантів використання (Use Case), бізнес-процесів документообігу та нормативних вимог до електронних документів.

Класи сутності представляють стабільні інформаційні об'єкти, що існують протягом усього життєвого циклу системи та зберігаються у базі даних. Вони використовуються для опису структури даних, взаємозв'язків між об'єктами та підтримки ключових процесів створення, погодження, підписання, зберігання та архівування документів [16, 17].

На основі аналізу предметної області та вимог до системи виявлено такі основні класи сутності.

Клас Document є центральною сутністю системи та призначений для представлення електронного документа, який проходить повний життєвий цикл у СЕД. Документ містить основні реквізити, статуси обробки, посилання на файл, інформацію про маршрут погодження та історію змін. Цей клас використовується у більшості варіантів використання системи, зокрема: створення документа, реєстрація, маршрутизація, погодження, підписання, пошук і архівування.

Клас DocumentVersion призначений для зберігання інформації про версії документа, що створюються під час редагування або накладання електронного підпису. Наявність окремої сутності для версій дозволяє забезпечити відстужеваність змін, аудит дій користувачів та відповідність вимогам нормативних документів. Кожна версія пов'язана з конкретним документом і містить посилання на файл відповідної версії, інформацію про автора змін та час створення.

Клас User представляє користувачів СЕД, які взаємодіють із документами відповідно до наданих ролей і прав доступу. Користувачі можуть виконувати дії створення, перегляду, погодження, підписання та адміністрування документів. Клас

використовується для реалізації механізмів автентифікації, авторизації, контролю доступу та фіксації дій у журналі аудиту.

Клас Role описує рольову модель доступу в системі та визначає набір дозволених дій для користувачів. Використання ролей дозволяє гнучко керувати правами доступу до функціональності системи та документів. Ролі використовуються під час маршрутизації документів, перевірки прав доступу та виконання бізнес-логіки погодження і підписання.

Клас Route призначений для представлення маршруту руху документа в системі. Маршрут визначає послідовність або паралельність етапів погодження, типи дій та ролі користувачів, відповідальних за виконання кожного етапу. Ця сутність дозволяє реалізувати гнучку модель погодження документів, адаптовану до потреб конкретної організації.

Клас RouteStep описує окремий етап маршруту погодження документа. Кожен етап містить інформацію про порядок виконання, необхідну роль користувача та тип дії (погодження, перегляд, підписання). Використання цього класу дозволяє формалізувати бізнес-процеси документообігу та забезпечити контроль проходження документа через усі етапи маршруту.

Клас Signature призначений для зберігання інформації про накладання кваліфікованого електронного підпису на документ. Сутність містить дані про підписанта, час підписання, серійний номер сертифіката та пов'язаний файл. Цей клас забезпечує юридичну значимість електронних документів і використовується у процесах підписання та перевірки підписів.

Клас File використовується для представлення файлів документів, що зберігаються у файловому сховищі системи. Він містить метадані файлу, такі як назва, тип, розмір, шлях зберігання та інформацію про користувача, який завантажив файл. Відокремлення файлів у самостійну сутність дозволяє ефективно керувати версіями документів і реалізувати масштабоване файлове сховище.

Клас AuditLog призначений для фіксації дій користувачів у системі, зокрема створення, редагування, погодження, підписання та архівування документів.

Журнал аудиту є важливим елементом забезпечення безпеки, прозорості та відповідності нормативним вимогам.

Клас Notification використовується для реалізації механізмів інформування користувачів про події в системі, такі як надходження документа на погодження, завершення етапу маршруту або успішне підписання документа.

2.4 Визначення класів системи

Після виявлення класів сутності наступним етапом проектування СЕД є визначення ролей та взаємодії основних класів у процесі реалізації прикладної логіки системи. У межах даної магістерської роботи під визначенням класів системи розуміється опис використання вже визначених доменних класів для підтримки бізнес-процесів створення, маршрутизації, погодження, підписання, пошуку та аудиту електронних документів. При цьому нові доменні класи не вводяться, а уточнюється їх функціональне призначення в межах системи [15, 16].

Визначення класів системи здійснювалося на основі аналізу функціональних вимог, варіантів використання (Use Case) та архітектурних рішень серверної частини системи. Такий підхід забезпечує узгодженість між доменною моделлю, UML-діаграмою класів та програмною реалізацією, описаною у наступних розділах роботи (діаграма класів системи відображена на рисунку 2.2, програмний код UML код діаграми класів відображено в лістингу В.1.).

Клас User представляє користувача СЕД та використовується для ідентифікації осіб, які взаємодіють із системою. Користувачі можуть виконувати дії створення, перегляду, погодження, підписання та адміністрування документів відповідно до наданих прав доступу. Інформація про користувачів використовується під час автентифікації, авторизації та фіксації дій у журналі аудиту.

Клас Role визначає роль користувача в системі та використовується для реалізації рольової моделі доступу. Ролі встановлюють перелік дозволених дій користувача на різних етапах життєвого циклу документа, зокрема під час

створення, погодження, підписання та архівування. Використання ролей забезпечує контроль доступу до функціональності системи та підвищує рівень безпеки.

Клас `Document` є центральним класом системи та представляє електронний документ, який проходить повний життєвий цикл у СЕД. Документ містить основні реквізити, поточний статус обробки, інформацію про маршрут документообігу та пов'язані версії і файли. Клас використовується у більшості бізнес-процесів системи.

Клас `DocumentVersion` призначений для зберігання інформації про версії документа, які створюються під час його редагування або підписання. Використання окремого класу для версій дозволяє забезпечити відстежуваність змін, збереження історії документа та відповідність нормативним вимогам до електронного документообігу.

Клас `FileAttachment` використовується для представлення файлів, прикріплених до електронних документів. Він містить метадані файлів і забезпечує зберігання та доступ до вмісту документів. Файли пов'язуються з конкретними версіями документів, що дозволяє реалізувати контроль змін та збереження попередніх станів.

Клас `Route` описує маршрут документообігу та використовується для визначення послідовності або паралельності етапів погодження документа. Маршрут задає загальну схему руху документа між користувачами та ролями в межах організації.

Клас `RouteStep` представляє окремий етап маршруту документообігу. Кожен етап визначає тип дії, порядок виконання та роль користувача, відповідального за обробку документа на відповідному кроці. Використання етапів маршруту дозволяє формалізувати бізнес-процеси погодження та підписання документів.

Клас `Signature` призначений для зберігання інформації про накладання електронного підпису на документ. Він містить дані про підписанта, час підписання та пов'язаний файл. Даний клас забезпечує юридичну значимість електронних документів і використовується у процесах підписання та перевірки підписів.

Клас AuditLog використовується для фіксації дій користувачів у системі, зокрема створення, редагування, погодження, підписання та архівування документів. Журнал аудиту є важливим елементом забезпечення прозорості, контролю та інформаційної безпеки системи.

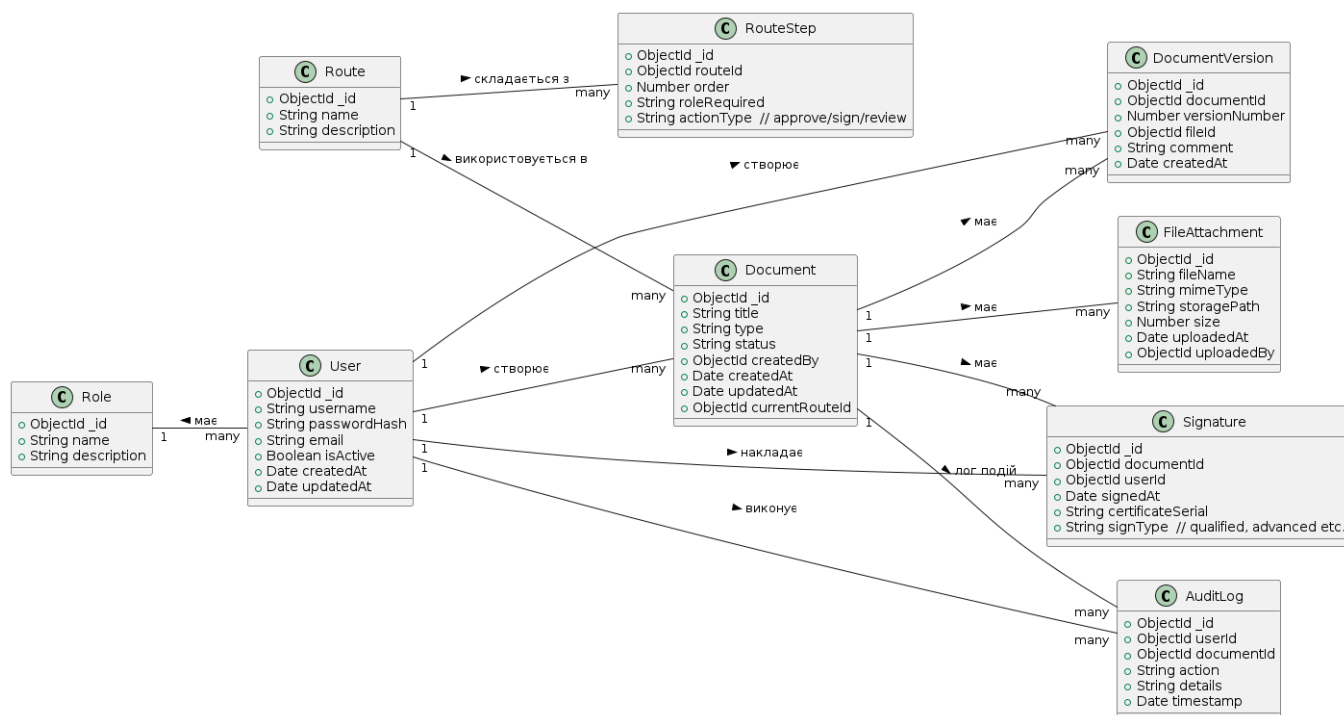


Рисунок 2.2 – Діаграма класів системи

2.5 Побудова діаграми діяльності бізнес-процесу

На базі основного бізнес-процес погодження електронного документа в СЕД розглянемо побудову діаграми діяльності бізнес-процесу. Процес охоплює етапи створення документа, вибору маршруту погодження, послідовного проходження етапів маршруту та завершення погодження з подальшою підготовкою документа до підписання [16].

Процес починається зі створення документа користувачем. Після створення документ реєструється в системі та зберігається як початковий запис. Далі користувач обирає маршрут погодження, який визначає послідовність етапів

перевірки документа. Після вибору маршруту системою призначається перший етап погодження, з якого стартує процес.

Основна частина бізнес-процесу реалізується у вигляді циклу, що повторюється до завершення маршруту погодження. На кожному етапі система надсилає сповіщення відповідальному користувачу про необхідність розгляду документа. Відповідальний переглядає документ та приймає рішення щодо його подальшої обробки.

У разі прийняття позитивного рішення документ отримує статус, що відображає успішне погодження на поточному етапі. Після цього система перевіряє наявність наступного етапу маршруту. Якщо такий етап існує, документ передається на наступний етап погодження, і цикл продовжується.

Якщо на будь-якому етапі погодження приймається рішення про відхилення документа, система змінює його статус на «Відхилено» та надсилає повідомлення ініціатору документа. Після цього бізнес-процес погодження завершується достроково без переходу до наступних етапів.

У випадку успішного проходження всіх етапів маршруту погодження цикл завершується. Документ вважається повністю погодженим, після чого його статус змінюється на «Готовий до підписання». На цьому етапі бізнес-процес погодження документа вважається завершеним (діаграма діяльності бізнес-процесу погодження документа відображена на рисунку 2.3).

Результатом виконання бізнес-процесу погодження може бути один із двох станів документа:

- документ повністю погоджено та підготовлено до підписання;
- документ відхилено на одному з етапів погодження з відповідним повідомленням ініціатора.

Отже, діаграма діяльності наочно відображає послідовність дій та логіку прийняття рішень у процесі погодження документа, а також демонструє циклічну природу проходження етапів маршруту погодження.

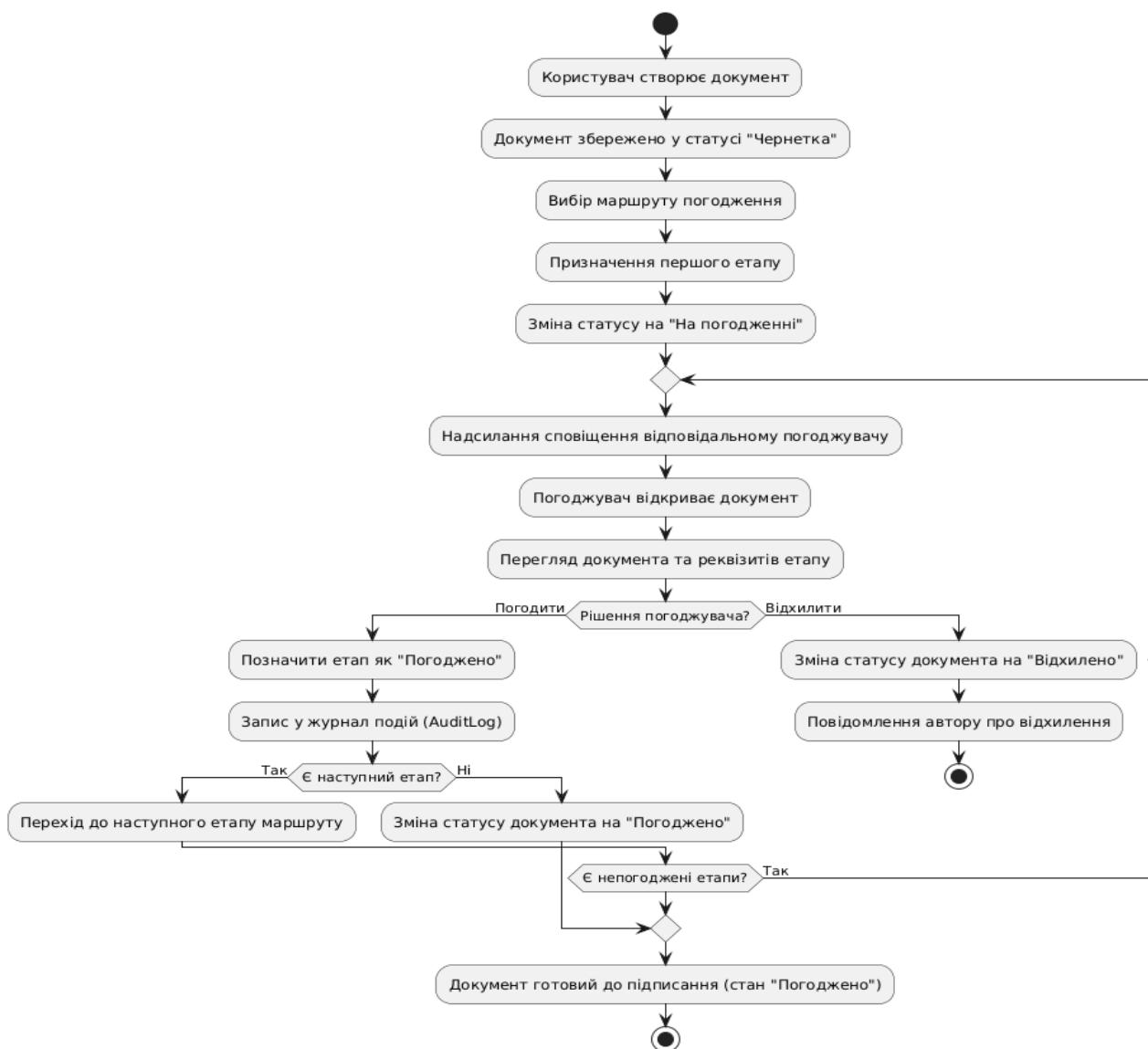


Рисунок 2.3 – Діаграма діяльності бізнес-процесу погодження документа

2.6 Побудова діаграми стану бізнес-процесу

На базі життєвого циклу електронного документа в СЕД розглянемо діаграму стану, яка відображає життєвий цикл електронного документа в СЕД та визначає множину допустимих станів документа, а також події, що ініціюють переходи між цими станами. Такий підхід дозволяє формалізувати поведінку документа протягом усього періоду його обробки – від створення до завершального архівування – та забезпечити контроль коректності виконання бізнес-процесів [9, 16].

Життєвий цикл документа починається зі створення документа, після чого він переходить у стан «Чернетка». У цьому стані документ доступний для редагування, заповнення реквізитів та додавання вкладень. Стан «Чернетка» використовується для підготовки документа до подальшої формалізації та запуску регламентованих процедур.

Після завершення підготовки виконується реєстрація документа, що передбачає присвоєння йому унікального реєстраційного номера. Внаслідок цієї дії документ переходить у стан «Зареєстровано». Реєстрація фіксує документ у системі як офіційну одиницю документообігу та забезпечує можливість його подальшого контролю і відстеження.

Із стану «Зареєстровано» документ переводиться у стан «На погодженні» шляхом запуску маршруту погодження (дія *«Надіслати на погодження»*). У цьому стані документ проходить послідовність етапів, визначених маршрутом документообігу. Кожен етап передбачає перегляд документа та прийняття рішення відповідальним користувачем.

Під час перебування у стані «На погодженні» можливі два альтернативні результати:

- якщо усі етапи погодження виконано успішно, документ переходить у стан «Погоджено», що означає завершення процесу узгодження;
- якщо документ відхилено на одному з етапів, він переходить у стан «Відхилено», і процес погодження завершується негативним рішенням.

Для документа у стані «Відхилено» передбачено можливість повернення на доопрацювання. У цьому випадку документ переводиться назад у стан «Чернетка», де може бути відредагований з урахуванням зауважень та повторно підготовлений до проходження життєвого циклу, включно з реєстрацією та повторним погодженням.

Після успішного погодження документ у стані «Погоджено» може бути переданий на накладання електронного підпису. У разі успішного підписання документ переходить у стан «Підписано», що підтверджує його юридичну значимість та автентичність.

Завершальним етапом життєвого циклу є архівування, після якого документ переходить у стан «В архіві». Архівування може виконуватися двома способами:

- після підписання – перехід зі стану «Підписано»;
- без накладання електронного підпису – безпосередньо зі стану «Погоджено», якщо це дозволено правилами обробки документа.

Документ у стані «В архіві» вважається завершеним у межах активного документообігу та не підлягає редагуванню.

Після переходу у стан «В архіві» життєвий цикл документа завершується. Подальші дії з таким документом обмежуються переглядом та довготривалим зберіганням відповідно до політик системи (діаграма стану бізнес-процесу відображена на рисунку 2.4).

2.7 Опис роботи системи

Робота веб-орієнтованої СЕД ґрунтується на взаємодії користувачів із серверною частиною через веб-інтерфейс та реалізації бізнес-процесів обробки електронних документів. Система забезпечує повний життєвий цикл документа – від створення та реєстрації до погодження, підписання, пошуку та архівування – із дотриманням вимог безпеки та нормативних актів. Функціонування системи організоване відповідно до ролей користувачів, маршрутизації документів та фіксації всіх дій у журналі аудиту, що забезпечує прозорість і контроль процесів документообігу [13, 14].

Користувачі взаємодіють із системою через веб-браузер, надсилаючи запити до серверної частини. Сервер обробляє запити, виконує перевірку прав доступу, реалізує бізнес-логіку та взаємодіє з базою даних і файловим сховищем. Результати виконання операцій повертаються користувачу у вигляді оновленого інтерфейсу або повідомлень про стан виконання дій.

Доступ до функціональності системи визначається роллю користувача, що дозволяє реалізувати розмежування прав доступу та запобігти несанкціонованим діям.

Робота з документом починається із завантаження документа до СЕД, яка формує початкову версію документа та зберігає її у файловому сховищі.

Після створення документа користувач ініціює процес маршрутизації, який передбачає призначення маршруту документообігу. Маршрут складається з послідовності етапів, кожен з яких визначає роль користувача, відповідального за виконання відповідної дії.

Документ передається між користувачами відповідно до визначеного маршруту.

На завершальному етапі погодження документ може бути переданий на підписання користувачу з відповідними повноваженнями. Підписання здійснюється з використанням електронного підпису, що забезпечує юридичну значимість документа.

Після успішного підписання створюється підписана версія документа, а інформація про підписанта, час підписання та параметри підпису зберігається в системі. Документ отримує статус «Підписано» та стає доступним для подальшого використання або архівування.

Файли документів зберігаються у файловому сховищі, а метадані – у базі даних. Такий підхід забезпечує масштабованість і ефективне управління даними.

Користувачі можуть переглядати документи відповідно до своїх прав доступу, що забезпечує конфіденційність та захист інформації.

Усі значущі дії користувачів фіксуються в журналі аудиту. Журнал містить інформацію про тип дії, користувача, документ та час виконання операції. Аудит дозволяє забезпечити контроль процесів, виявляти порушення та аналізувати роботу системи.

Після завершення життєвого циклу документ може бути переведений в архів. Архівні документи доступні лише для користувачів із відповідними правами та не підлягають редагуванню.

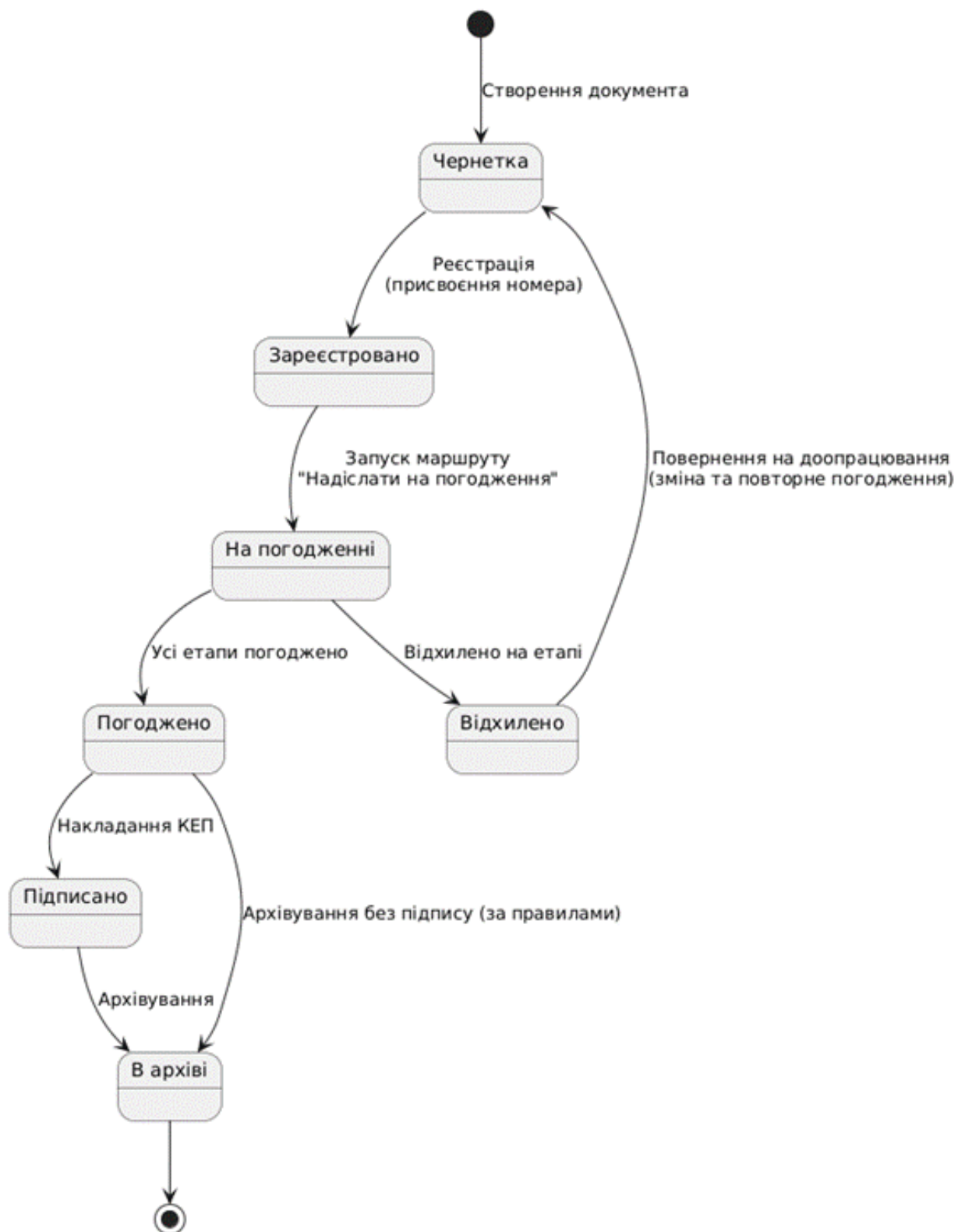


Рисунок 2.4 – Діаграма стану життєвого циклу електронного документа

2.8 Архітектура системи електронного документообігу

Архітектура веб-орієнтованої СЕД побудована за клієнт-серверним принципом і забезпечує розподіл функціональності між окремими рівнями системи. Такий підхід дозволяє підвищити масштабованість, безпеку та зручність супроводу програмного продукту.

На клієнтському рівні система представлена веб-інтерфейсом, з яким користувачі взаємодіють через браузер. Клієнтська частина відповідає за відображення інтерфейсу, введення даних та формування HTTP-запитів до серверної частини системи [15, 18].

Серверний рівень реалізує прикладну логіку системи та обробку запитів клієнта. Він включає контролери, які приймають запити, виконують базову перевірку даних та ініціюють виконання бізнес-логіки. Прикладна логіка відповідає за реалізацію процесів створення, маршрутизації, погодження, підписання та архівування електронних документів. Окрему роль відіграють проміжні компоненти, які забезпечують автентифікацію користувачів, контроль доступу та перевірку прав відповідно до ролей.

Рівень зберігання даних представлений базою даних, у якій зберігаються метадані документів, користувачів, маршрутів та журнал аудиту, а також файловим сховищем для збереження файлів документів і їх версій. Такий поділ дозволяє ефективно управляти великими обсягами даних та забезпечити цілісність інформації.

Для забезпечення юридичної значимості документів система взаємодіє з механізмами накладання кваліфікованого електронного підпису. Взаємодія між клієнтським і серверним рівнями здійснюється з використанням захищених мережових протоколів.

Запропонована архітектура забезпечує чітке розмежування відповідальностей між компонентами системи, узгоджується з UML-моделями, наведеними у попередніх підрозділах, та створює основу для подальшої програмної реалізації системи (діаграма архітектури системи відображена на рисунку 2.5).

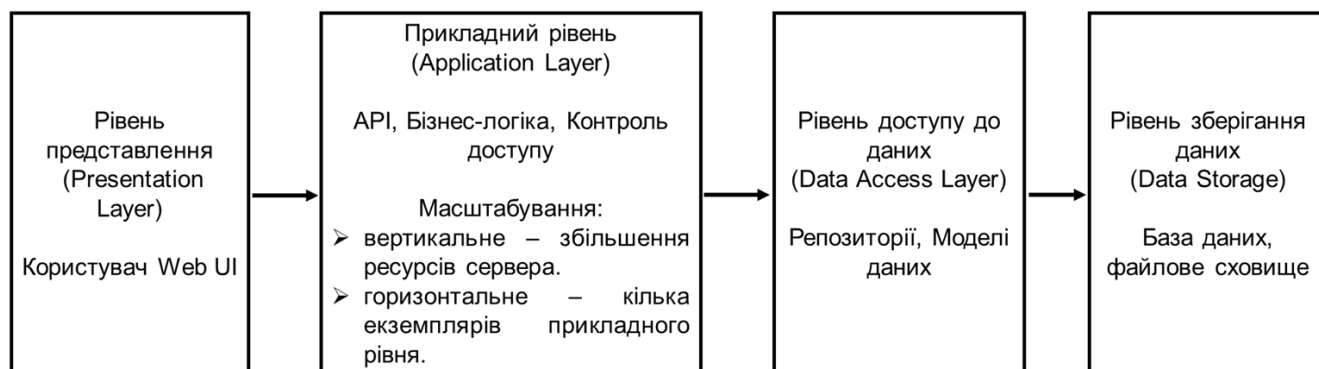


Рисунок 2.5 – Діаграма архітектури системи

2.9 Вибір мов програмування, технологій та середовища розробки

Для реалізації веб-орієнтованої СЕД обрано програмний стек, який забезпечує високу продуктивність, масштабованість, надійність та зручність подальшого супроводу системи. Вибір мов програмування, технологій та середовища розробки здійснювався з урахуванням архітектурних рішень, сформованих у попередніх підрозділах, а також вимог до реалізації повного життєвого циклу електронних документів.

Для реалізації серверної частини системи використовується мова програмування JavaScript, що працює на платформі Node.js. Застосування JavaScript дозволяє реалізувати асинхронну обробку запитів, ефективну роботу з мережею та гнучку взаємодію з базою даних [22, 23].

Для клієнтської частини системи застосовуються HTML5, CSS3 та JavaScript, які забезпечують формування користувацького інтерфейсу, обробку подій та взаємодію з серверною частиною через HTTP-запити. Для генерації HTML-сторінок на сервері використовується Embedded JavaScript Templates.

Як платформа виконання серверної частини обрано Node.js, що дозволяє створювати високопродуктивні веб-застосунки. Для побудови серверного застосунку та організації маршрутизації використовується фреймворк Express.js, який забезпечує зручну обробку HTTP-запитів та реалізацію REST-подібного API.

Embedded JavaScript Templates використовується для формування динамічних HTML-сторінок та забезпечує інтеграцію серверної логіки з користувацьким інтерфейсом [26].

Для зберігання даних системи використовується документоорієнтована система керування базами даних MongoDB. Вона забезпечує гнучку структуру зберігання даних, що є доцільним для роботи з електронними документами, їх версіями, маршрутами погодження та журналом аудиту [24, 25].

Як інтегроване середовище розробки використовується Visual Studio Code, що забезпечує підтримку мови JavaScript, налагодження коду та зручну роботу з проектом.

Розгортання серверної частини системи здійснюється на операційній системі Windows, що відповідає вимогам до інфраструктури організацій та забезпечує стабільну роботу веб-застосунку.

2.10 Реалізація серверної частини системи

Реалізація серверної частини системи виконана у вигляді модульного веб-застосунку. Точкою входу є файл `index.js`, у якому ініціалізується веб-сервер, підключаються контролери та налаштовується захищене HTTPS-з'єднання.

2.10.1 Реалізація контролерів та маршрутів

Контролери відповідають за обробку HTTP-запитів клієнтської частини та реалізують сценарії використання системи. Кожен контролер відповідає за окремий функціональний модуль документообігу.

Файли реалізації контролерів:

- `controllers/dasincoming.js` – вхідні документи;
- `controllers/dastransact.js` – операції над документами;
- `controllers/dasmonitor.js` – моніторинг станів документів;
- `controllers/dasfileoperation.js` – робота з файлами;

- controllers/dassearch.js – пошук документів;
- controllers/dassysadmin.js – адміністративні функції;
- controllers/login.js – автентифікація користувачів.

Маршрути визначають URL-шляхи та прив'язують їх до відповідних методів контролерів.

2.10.2 Реалізація бізнес-логіки документообігу

Бізнес-логіка системи реалізована безпосередньо у контролерах і допоміжних модулях. Вона охоплює створення документів, зміну статусів, управління маршрутами погодження, контроль переходів між етапами та ініціацію підписання.

Файли реалізації:

- controllers/dasincoming.js;
- controllers/dastransact.js;
- controllers/dasmonitor.js.

2.10.3 Реалізація механізмів автентифікації та авторизації

Механізми автентифікації та авторизації реалізовані з використанням хешування паролів та перевірки облікових даних користувачів [12, 27, 28].

Файли реалізації:

- controllers/login.js – логіка входу користувачів.

2.10.4 Реалізація роботи з документами, файлами та версіями

Функціональність роботи з документами включає створення документів, завантаження файлів, формування версій та їх збереження у файловому сховищі [18].

Файли реалізації:

- controllers/dasincoming.js – створення та реєстрація документів;

- controllers/dasfileoperation.js – завантаження та обробка файлів.

2.10.5 Реалізація процесів погодження та маршрутизації

Процеси погодження та маршрутизації документів реалізують основний бізнес-процес системи та забезпечують проходження документа через визначені етапи маршруту [16, 29, 30].

Файли реалізації:

- controllers/dastransact.js – виконання етапів маршруту;
- controllers/dasmonitor.js – контроль станів погодження.

2.10.6 Реалізація електронного підпису та аудиту дій

Підписання документів та фіксація дій користувачів реалізовані як завершальний етап документообігу [2, 11, 31].

Файли реалізації:

- controllers/dasfileoperation.js – накладання підпису на PDF;
- controllers/dasmonitor.js – реєстрація подій.

2.10.7 Реалізація інтерфейсу користувача

Інтерфейс користувача СЕД передбачає використання класичних ідентифікаторів логін та пароль користувача (інтерфейс користувача при вході в систему відображено на рисунку 2.6, HTML програмний код реалізації сторінки входу відображено в лістингу В.2, JavaScript програмний код реалізації сторінки входу відображено в лістингу В.3) [19].

Магістерська робота
TNTU

ІМ'Я КОРИСТУВАЧА

ПАРОЛЬ КОРИСТУВАЧА

☐ Запам'ятати мене

ВХІД ДО СИСТЕМИ

Рисунок 2.6 – Інтерфейс користувача при вході в систему

За результатами входу користувача з правами адміністратора СЕД надає можливість цьому користувачу здійснити додавання нових користувачів, налаштувати та ввести ролі користувачів, які потім надаються кожному користувачу в залежності від його повноважень (інтерфейс додавання нового користувача до системи та результат додавання відображено на рисунках 2.7, 2.8, 2.9)

Є-документ

ОБЛІКОВИЙ ЗАПИС

Додати користувача

Змінити дані користувача

НАЛАШТУВАННЯ

Список

Додатково

КОМУНІКАЦІЯ

Моніторинг

Ім'я користувача

Пароль користувача

Підтвердити пароль користувача

ПІБ користувача

Група/Підрозділ

Рівень доступу

Прив'язка шляху (для редагування документів)

Додати користувача

Рисунок 2.7 – Інтерфейс для введення параметрів нового користувача до системи

Рисунок 2.8 – Додавання нового користувача до системи

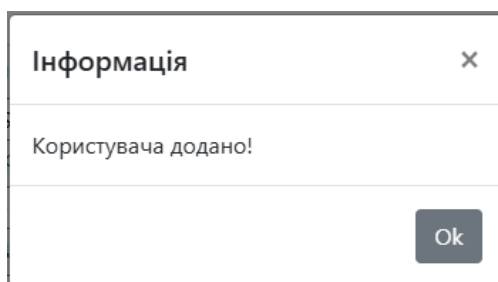


Рисунок 2.9 – Інтерфейс результату додавання нового користувача до системи

Водночас в СЕД реалізовано загальна матриця користувачів системи, яка для користувача з правами адміністратор системи, відображає перелік всіх користувачів системи, в якій відображаються всі параметри користувачів, їх приналежність до підрозділу організації, рівень доступу. Функціонал матриці користувачів надає також можливість здійснити корегування даних користувача та його видалення із системи (матриця користувачів системи відображена на рисунку 2.10).

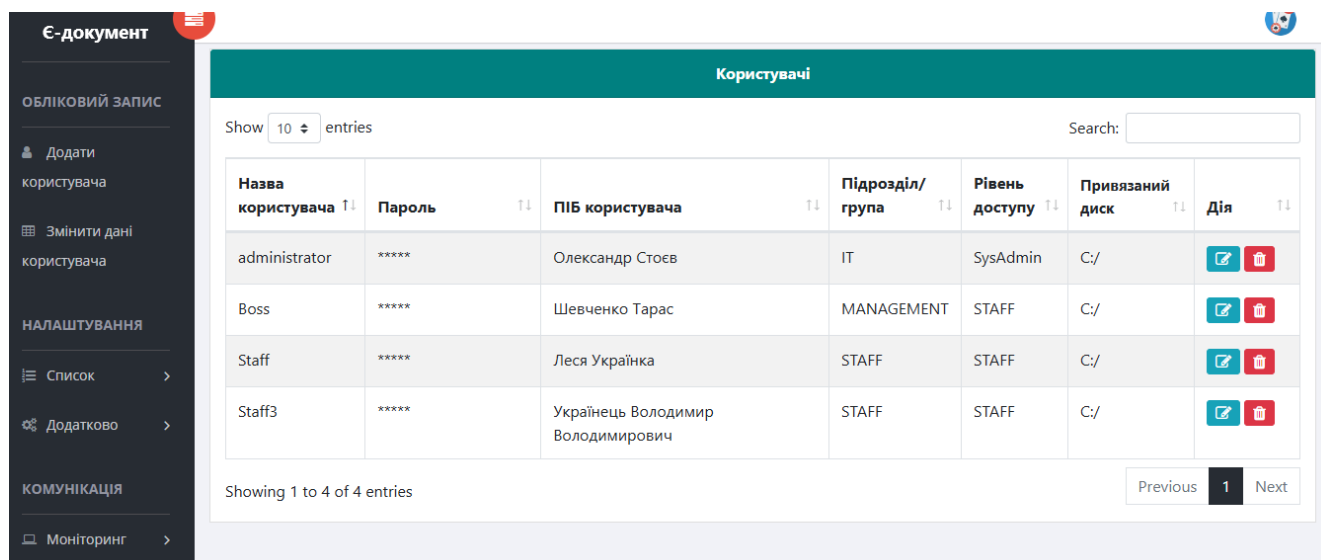


Рисунок 2.10 – Інтерфейс матриці користувачів системи

За результатом додавання користувача, цей користувач отримує можливість доступу до системи, в якій відповідно до його прав надається можливість завантаження файлу до системи (інтерфейс завантаження файлу до системи відображено на рисунку 2.11).

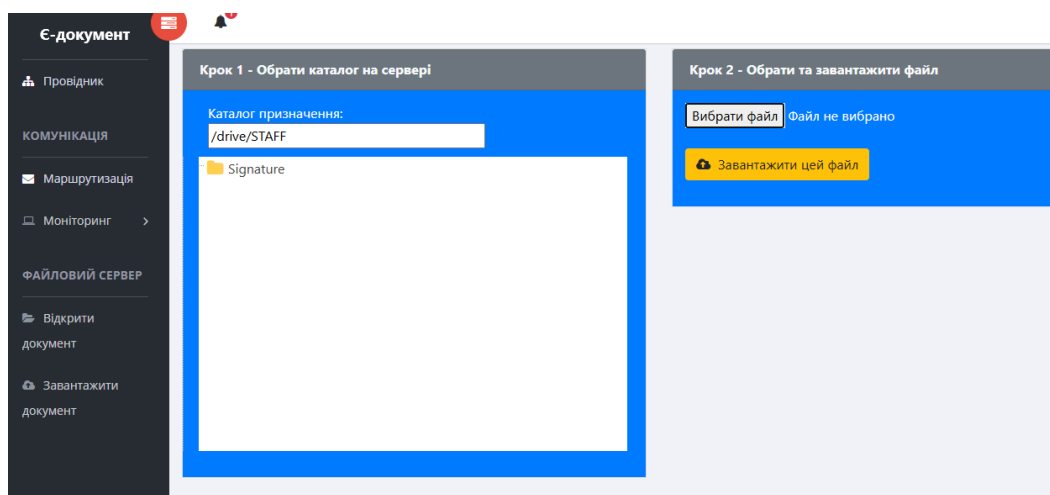


Рисунок 2.11 – Інтерфейс завантаження файлів до системи

Результатом завантаження файлу користувачем є візуальне відображення документа з можливістю його маршрутизації до підрозділу організації або до

конкретного працівника організації (інтерфейс результатів завантаження відображено на рисунку 2.12).

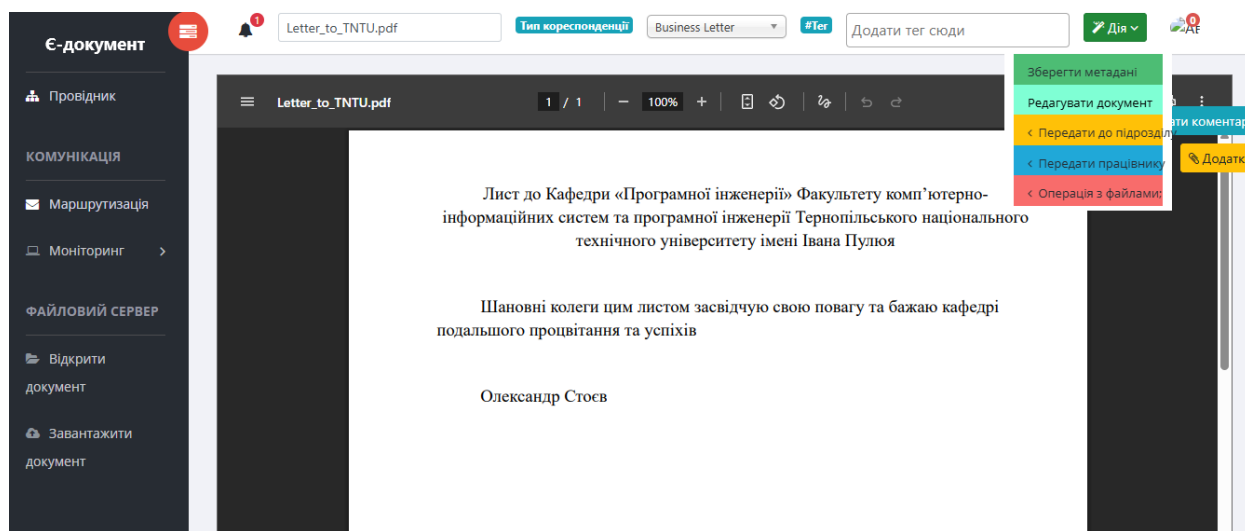


Рисунок 2.12 – Інтерфейс результату завантаження файлів до системи

За результатом завантаження до СЕД файлу користувач має можливість маршрутизації цього документа як на конкретного співробітника організації так і на підрозділ організації.

СЕД забезпечує сповіщення користувача про надходження нових документів, які можливо переглянути та здійснити їх подальшу обробку (інтерфейс отримання нових документів відображено на рисунку 2.13)

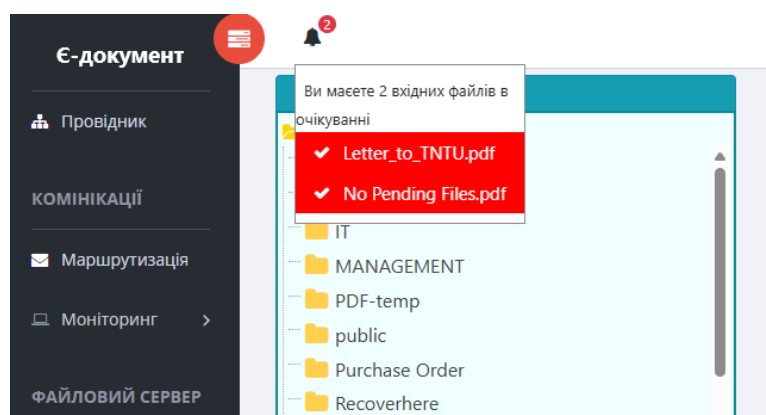
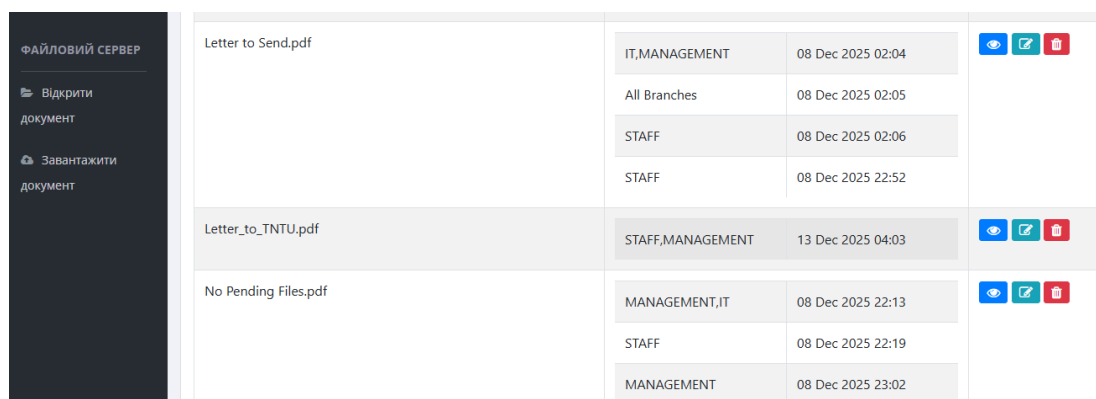


Рисунок 2.13 – Інтерфейс отримання файлу від користувача системи

За результатом маршрутизації документа та ознайомлення з документом здійснюється відображення в журналі подій надісланих документів користувачами документів (інтерфейс відображення історії маршрутизації відображено на рисунку 2.14).



Letter to Send.pdf	IT,MANAGEMENT	08 Dec 2025 02:04	  
	All Branches	08 Dec 2025 02:05	
	STAFF	08 Dec 2025 02:06	
	STAFF	08 Dec 2025 22:52	
Letter_to_TNTU.pdf	STAFF,MANAGEMENT	13 Dec 2025 04:03	  
No Pending Files.pdf	MANAGEMENT,IT	08 Dec 2025 22:13	  
	STAFF	08 Dec 2025 22:19	
	MANAGEMENT	08 Dec 2025 23:02	

Рисунок 2.14 – Інтерфейс відображення історії маршрутизації

За результатом ознайомлення із документом користувач має можливість підписати отриманий документ, який за результатом підписання направляється в архів системи (інтерфейс підпису файлу відображено на рисунку 2.15).

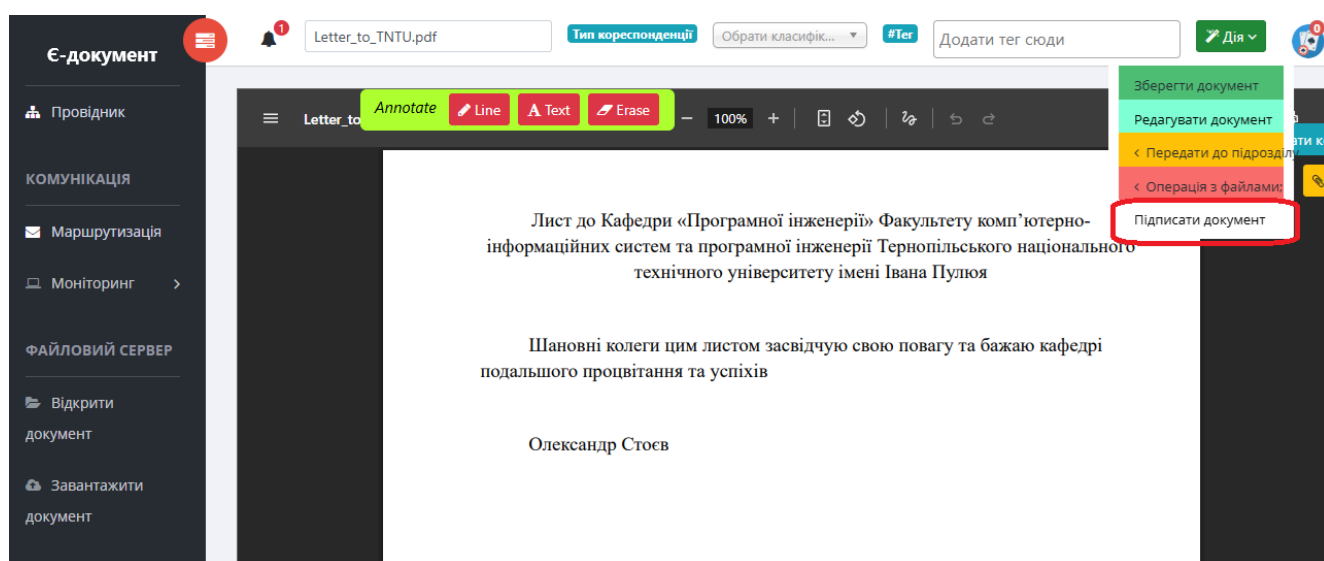


Рисунок 2.15 – Інтерфейс підписання документа

Отже, у другому розділі магістерської роботи виконано комплекс робіт з проєктування та розробки веб-орієнтованої СЕД. На основі обраної методології Rational Unified Process здійснено послідовний перехід від аналізу вимог і моделювання предметної області до практичної реалізації програмних компонентів системи.

У межах розділу сформовано UML-моделі, які забезпечують формалізований опис структури та поведінки системи. Діаграма прецедентів визначає основні ролі користувачів і варіанти використання системи, на основі яких виконано виявлення класів сутності та проєктування класів системи. Побудовані UML-діаграми дозволили узгодити вимоги предметної області з архітектурними рішеннями та подальшою програмною реалізацією.

Значну увагу приділено моделюванню динамічної поведінки системи. Діаграма діяльності відображає логіку виконання основного бізнес-процесу обробки документа, а діаграма стану формалізує життєвий цикл електронного документа та визначає допустимі переходи між його станами у процесі документообігу.

Важливим результатом розділу є те, що модель життєвого циклу електронного документа, представлена у вигляді діаграми станів, повністю реалізована у функціональності розробленої системи. Програмна реалізація забезпечує коректне проходження документа через усі етапи його обробки відповідно до визначеної логіки переходів між станами. Таким чином, функціональні можливості системи повністю відображають та підтримують життєвий цикл електронного документа, що підтверджує узгодженість проєктних рішень і програмної реалізації.

У розділі також розроблено архітектуру системи, обґрунтовано вибір мов програмування, технологій та середовища розробки, а також описано реалізацію серверної частини та користувацького інтерфейсу. Реалізовані механізми автентифікації, авторизації, управління документами, маршрутизації, електронного підпису та журналювання дій забезпечують повний функціональний цикл роботи СЕД.

Результати другого розділу підтверджують, що розроблена система є логічно узгодженою, функціонально повною та такою, що відповідає сформованим вимогам і UML-моделям, створюючи надійну основу для подальшого тестування, оцінки ефективності та впровадження системи.

3 ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ПІДТРИМКА СИСТЕМИ

Тестування є невід’ємним етапом життєвого циклу програмної системи та відіграє ключову роль у забезпеченні її якості, надійності й відповідності вимогам користувачів. Для СЕД, які оперують службовою інформацією, містять складні бізнес-процеси та використовуються у багатокористувацькому режимі, тестування має особливе значення, оскільки помилки в роботі такої системи можуть призводити до втрати даних, порушення регламентів або зниження ефективності управлінських процесів [13, 14].

3.1 Тестування в життєвому циклі системи

У межах даної магістерської роботи тестування розглядається не як одноразова перевірка працездатності програмного продукту, а як систематичний процес, що супроводжує розроблення програмної системи на всіх етапах – від реалізації окремих функціональних компонентів до підготовки системи до впровадження в експлуатацію. Такий підхід відповідає обраній методології розробки та дозволяє забезпечити узгодженість між проєктними рішеннями і програмною реалізацією.

Основною метою тестування у даній роботі є підтвердження коректності реалізації функціональних і нефункціональних вимог, сформульованих у попередніх розділах. Зокрема, тестування спрямоване на перевірку правильності реалізації бізнес-процесів електронного документообігу, визначених за допомогою UML-діаграм діяльності та станів, а також на підтвердження повної підтримки життєвого циклу електронного документа програмними засобами системи.

Особливу увагу у процесі тестування приділено перевірці динамічної поведінки системи. Для СЕД критично важливою є коректна зміна станів документа залежно від дій користувачів, ролей доступу та результатів виконання бізнес-операцій. У зв’язку з цим тестування охоплює перевірку переходів між

станами документа, коректність ініціації та завершення процесів погодження, а також дотримання обмежень, визначених логікою системи.

Крім функціональних аспектів, тестування у межах даної роботи спрямоване на оцінку надійності та стабільності програмної системи. Це включає перевірку роботи системи в умовах одночасної взаємодії декількох користувачів, коректності обробки помилкових або неповних даних, а також здатності системи відновлювати працездатність після типових збоїв. Такі перевірки дозволяють оцінити готовність системи до використання в реальному середовищі.

Важливою складовою тестування є також перевірка механізмів безпеки, зокрема автентифікації та авторизації користувачів, контролю доступу до функціональних можливостей системи та захисту конфіденційної інформації. Для СЕД ці аспекти мають принципове значення, оскільки забезпечують дотримання вимог інформаційної безпеки та нормативних обмежень.

Отже, тестування у межах даної магістерської роботи розглядається як комплексний процес, спрямований на підтвердження якості програмної реалізації, відповідності системи сформованим вимогам і готовності її до впровадження та подальшої експлуатації. Результати тестування створюють підґрунтя для оцінки ефективності системи та формують основу для її супроводу і розвитку.

3.2 Цілі та завдання тестування системи

Тестування СЕД у межах даної магістерської роботи проводиться з метою підтвердження коректності та повноти реалізації програмної системи, а також її відповідності функціональним і нефункціональним вимогам, сформульованим на етапі проєктування. Враховуючи специфіку СЕД, тестування спрямоване не лише на перевірку окремих функцій, а й на комплексну оцінку роботи системи в цілому.

Основною ціллю тестування є підтвердження того, що розроблена система забезпечує коректне виконання всіх основних бізнес-процесів електронного документообігу, зокрема створення, маршрутизації, погодження, підписання та архівування електронних документів. Досягнення цієї цілі передбачає перевірку

узгодженості між UML-моделями, наведеними у Розділі 2, та фактичною поведінкою системи під час її експлуатації [13, 14].

Для досягнення визначеної мети у процесі тестування були поставлені такі основні завдання:

- перевірка коректності реалізації варіантів використання системи відповідно до діаграми прецедентів;
- підтвердження правильності реалізації бізнес-процесів, змодельованих за допомогою UML-діаграм діяльності;
- перевірка відповідності переходів між станами документа логіці, визначеній діаграмою стану життєвого циклу електронного документа;
- оцінка коректності реалізації рольової моделі доступу та обмежень прав користувачів;
- перевірка механізмів автентифікації та авторизації користувачів;
- підтвердження правильності збереження та обробки даних у базі даних і файловому сховищі;
- перевірка надійності роботи системи в умовах одночасної взаємодії кількох користувачів;
- оцінка стабільності та передбачуваності поведінки системи у разі помилкових або некоректних дій користувачів.

Окремим завданням тестування є перевірка реалізації життєвого циклу електронного документа. Це передбачає контроль правильності ініціації процесів, коректності зміни станів документа, а також дотримання обмежень, які накладаються бізнес-правилами системи. Такий підхід дозволяє переконатися, що програмна реалізація не допускає некоректних переходів між станами та забезпечує цілісність процесу документообігу.

Важливим аспектом тестування є також оцінка готовності системи до впровадження. У цьому контексті тестування спрямоване на виявлення потенційних проблем, які можуть виникнути під час реальної експлуатації, зокрема пов'язаних із продуктивністю, стабільністю роботи та зручністю використання системи кінцевими користувачами.

Отже, цілі та завдання тестування визначають комплексний характер перевірки розробленої СЕД. Результати тестування слугують основою для підтвердження якості програмної реалізації, її відповідності проєктним рішенням і готовності системи до впровадження та подальшої підтримки.

3.3 Види та класифікація тестування системи

У процесі тестування СЕД застосовано комплексний підхід, що передбачає використання декількох видів тестування. Такий підхід зумовлений складністю програмної системи, наявністю багатокористувацького доступу, реалізацією бізнес-процесів документообігу та вимогами до безпеки і надійності. Класифікація видів тестування дозволяє систематизувати процес перевірки та забезпечити повноцінну оцінку якості програмної реалізації [13, 14].

3.3.1 Функціональне тестування

Функціональне тестування є базовим видом перевірки програмної системи та спрямоване на оцінку відповідності реалізованого функціоналу вимогам та UML-діаграмами. У межах даної роботи функціональне тестування охоплює перевірку основних сценаріїв використання системи, зокрема створення електронних документів, їх маршрутизації, погодження, підписання, архівування та пошуку.

Під час функціонального тестування особлива увага приділяється правильності обробки дій користувачів із різними ролями доступу. Перевіряється, що кожна роль має доступ лише до дозволених функцій, а спроби виконання несанкціонованих операцій коректно обробляються системою. Це дозволяє підтвердити правильність реалізації рольової моделі доступу.

3.3.2 Модульне тестування

Модульне тестування спрямоване на перевірку окремих програмних компонентів системи в ізоляції від інших частин. У СЕД такими компонентами є контролери, сервіси бізнес-логіки, моделі даних та допоміжні модулі.

Метою модульного тестування є виявлення помилок на ранніх етапах розроблення, а також перевірка коректності реалізації окремих функцій. Завдяки модульному тестуванню зменшується ймовірність поширення помилок на рівні всієї системи та спрощується процес їх локалізації і виправлення.

3.3.3 Інтеграційне тестування

Інтеграційне тестування використовується для перевірки коректності взаємодії між окремими компонентами системи. У межах даної роботи інтеграційне тестування спрямоване на перевірку взаємодії серверної частини з базою даних, файловим сховищем, механізмами автентифікації та користувацьким інтерфейсом.

Особливу увагу приділено перевірці узгодженості передачі даних між компонентами та коректності обробки помилок, що можуть виникати під час виконання складних бізнес-операцій. Результати інтеграційного тестування дозволяють підтвердити цілісність і узгодженість програмної архітектури.

3.3.4 Тестування безпеки

Тестування безпеки є одним із найважливіших видів тестування для СЕД. Воно спрямоване на перевірку механізмів захисту інформації, контролю доступу та запобігання несанкціонованим діям користувачів.

У межах тестування безпеки перевіряється коректність реалізації автентифікації та авторизації, захист облікових даних користувачів, а також дотримання обмежень доступу до функціоналу системи. Додатково оцінюється

стійкість системи до типових помилкових або некоректних дій користувачів, що можуть призвести до порушення цілісності даних.

3.3.5 Навантажувальне тестування

Навантажувальне тестування застосовується для оцінки стабільності та продуктивності системи в умовах підвищеного навантаження. Для СЕД це особливо актуально, оскільки вони можуть використовуватися великою кількістю користувачів одночасно.

Під час навантажувального тестування перевіряється здатність системи обробляти значну кількість одночасних запитів без втрати працездатності та суттєвого зниження швидкодії. Результати такого тестування дозволяють оцінити готовність системи до експлуатації в реальних умовах та виявити потенційні вузькі місця.

3.3.6 Регресійне тестування

Регресійне тестування проводиться після внесення змін або доопрацювань до системи з метою перевірки того, що нові зміни не призвели до порушення раніше реалізованого функціоналу. Цей вид тестування є важливим етапом супроводу системи та забезпечує стабільність її розвитку.

3.4 Тестове середовище та умови проведення тестування

Ефективність тестування програмної системи значною мірою залежить від правильно організованого тестового середовища та умов його проведення. Тестове середовище повинно максимально відповідати умовам реальної експлуатації системи, що дозволяє об'єктивно оцінити коректність реалізації функціональних можливостей і стабільність роботи програмного забезпечення [22, 23, 24].

У межах даної магістерської роботи тестування СЕД проводилося у контрольованому середовищі, яке відтворює типову конфігурацію серверної та клієнтської частин системи. Таке середовище дозволило виконати комплексну перевірку роботи системи в умовах, наближених до практичного використання.

3.4.1 Апаратно-програмне середовище

Тестування серверної частини системи здійснювалося в середовищі з типовими характеристиками, достатніми для розгортання та експлуатації веб-орієнтованих інформаційних систем. Серверна частина функціонувала на базі сучасної операційної системи з підтримкою мережевих сервісів, що забезпечує виконання бізнес-логіки системи та обробку запитів користувачів.

Для зберігання та обробки даних використовувалася система керування базами даних, яка підтримує роботу з документно-орієнтованими структурами. Такий підхід є доцільним для СЕД, оскільки дозволяє зберігати як структуровані дані, так і пов'язані з ними метадані та вкладення.

Клієнтська частина системи тестувалася у середовищі веб-браузера, що дозволило перевірити коректність роботи користувацького інтерфейсу, обробку дій користувачів та відображення інформації у процесі роботи з документами.

3.4.2 Організація тестових даних

Для проведення тестування підготовлено набір тестових даних, що включає облікові записи користувачів із різними ролями доступу, тестові електронні документи, а також сценарії їх обробки. Наявність користувачів із різними правами доступу дозволила перевірити коректність реалізації рольової моделі та обмежень функціоналу системи.

Тестові документи формувалися з урахуванням різних варіантів обробки, що дозволило перевірити поведінку системи у типових та граничних ситуаціях. Такий

підхід забезпечив перевірку основних бізнес-процесів електронного документообігу.

3.4.3 Умови та порядок проведення тестування

Тестування системи проводилося поетапно, починаючи з перевірки базових функцій та завершуючи комплексною оцінкою роботи системи в цілому. Перед кожним етапом тестування система приводилася у початковий стан, що забезпечувало об'єктивність результатів перевірки.

У процесі тестування фіксувалися виявлені зауваження та відхилення від очікуваної поведінки системи. Після усунення виявлених недоліків виконувалася повторна перевірка відповідного функціоналу. Такий підхід дозволив підвищити стабільність роботи системи та забезпечити її готовність до подальшої експлуатації.

3.5 План та стратегія тестування системи

Планування та визначення стратегії тестування є важливим етапом забезпечення якості програмної системи, оскільки дозволяє систематизувати процес перевірки, визначити послідовність дій та забезпечити повноту охоплення функціональних можливостей СЕД. У межах даної магістерської роботи стратегія тестування сформована з урахуванням особливостей архітектури системи, реалізованих бізнес-процесів та вимог до життєвого циклу електронного документа [13, 14].

Основною ідеєю обраної стратегії є поетапне тестування, яке передбачає поступовий перехід від перевірки окремих функціональних елементів до комплексної оцінки роботи системи в цілому. Такий підхід дозволяє своєчасно виявляти помилки, локалізувати їх причини та мінімізувати вплив на інші компоненти системи.

3.5.1 Підхід до планування тестування

План тестування формувався на основі варіантів використання системи, визначених у Розділі 2, а також з урахуванням UML-діаграм діяльності та станів. Кожен варіант використання розглядався як окремий сценарій тестування, що дозволило забезпечити повну перевірку реалізованих функціональних можливостей.

При плануванні тестування було визначено такі основні принципи:

- відповідність тестових сценаріїв функціональним вимогам;
- перевірка коректності переходів між етапами бізнес-процесів;
- урахування ролей користувачів та обмежень доступу;
- повторюваність і відтворюваність результатів тестування.

Такий підхід забезпечив системність тестування та дозволив охопити всі ключові аспекти роботи СЕД.

3.5.2 Етапи виконання тестування

Тестування системи здійснювалося у декілька послідовних етапів, кожен з яких мав чітко визначену мету та набір перевірок:

- підготовчий етап, на якому виконувалося налаштування тестового середовища, створення облікових записів користувачів з різними ролями доступу та підготовка тестових даних;
- перевірка базових функцій, що включала тестування входу до системи, навігації інтерфейсу та доступності основних функцій для різних ролей користувачів;
- тестування бізнес-процесів, під час якого перевірялася коректність виконання основних сценаріїв електронного документообігу, зокрема створення документів, їх обробка та зміна станів;
- комплексне тестування, спрямоване на оцінку узгодженої роботи всіх компонентів системи в рамках повного життєвого циклу електронного документа;

- повторне тестування, що виконувалося після усунення виявлених зауважень для підтвердження коректності внесених змін;

Послідовне виконання зазначених етапів дозволило забезпечити цілісність процесу тестування та поступове підвищення якості програмної реалізації.

3.5.3 Формування тестових сценаріїв

Тестові сценарії формувалися з урахуванням реальних умов використання системи та відображали типові дії користувачів. Кожен сценарій включав початкові умови, послідовність дій, очікуваний результат та фактичний результат виконання.

Особливу увагу приділено сценаріям, які охоплюють повний життєвий цикл електронного документа. Це дозволило перевірити не лише окремі операції, а й логіку їх виконання у межах одного наскрізного процесу. Такий підхід забезпечив відповідність тестування UML-діаграмам, побудованим у попередньому розділі.

3.5.4 Критерії успішності тестування

Для оцінки результатів тестування визначено критерії успішності, до яких належать:

- відповідність фактичної поведінки системи очікуваним результатам;
- відсутність критичних помилок, що унеможливлюють використання системи;
- коректна реалізація бізнес-правил та обмежень доступу;
- стабільна робота системи у межах визначених сценаріїв використання.

Досягнення зазначених критеріїв свідчить про готовність системи до подальшого впровадження та експлуатації.

3.6 Тестування реалізації життєвого циклу електронного документа

Життєвий цикл електронного документа є центральним елементом функціонування СЕД, оскільки саме він визначає допустимі стани документа та правила переходів між ними у процесі обробки. У межах даної магістерської роботи тестування реалізації життєвого циклу документа спрямоване на перевірку відповідності фактичної поведінки системи логіці, визначеній UML-діаграмою станів, сформованою на етапі проєктування [16].

Основною метою тестування цього аспекту є підтвердження того, що програмна реалізація забезпечує коректну зміну станів документа відповідно до дій користувачів, ролей доступу та бізнес-правил системи. Для систем електронного документообігу критично важливо, щоб жоден документ не міг перейти у некоректний стан або обійти обов'язкові етапи обробки, оскільки це може призвести до порушення регламентів і втрати контролю над процесом документообігу.

3.6.1 Підхід до тестування життєвого циклу документа

Тестування життєвого циклу електронного документа виконувалося на основі наскрізних сценаріїв, які охоплюють повний процес обробки документа – від його створення до завершення обробки. Кожен сценарій формувався таким чином, щоб ініціювати зміну стану документа шляхом виконання конкретних дій користувачів, передбачених бізнес-логікою системи.

У процесі тестування перевірялося:

- коректне встановлення початкового стану документа після його створення;
- можливість переходу документа лише у допустимі стани;
- блокування некоректних або несанкціонованих переходів;
- відповідність зміни стану документа виконаним діям користувачів;
- збереження актуального стану документа після завершення операцій.

Такий підхід дозволив перевірити не лише окремі переходи між станами, а й логічну цілісність життєвого циклу документа в цілому.

3.6.2 Перевірка коректності переходів між станами

Одним із ключових завдань тестування є перевірка коректності переходів між станами документа. Для цього виконувалися сценарії, що ініціюють зміну стану в результаті типових дій користувачів, таких як передача документа на обробку, виконання погодження або повернення на доопрацювання.

Під час тестування перевірялося, що кожна зміна стану документа супроводжується виконанням відповідної бізнес-операції та не може бути здійснена без дотримання визначених умов. Наприклад, перехід документа до наступного етапу обробки можливий лише після завершення попереднього етапу, а повернення документа на доопрацювання блокує подальші переходи до усунення зауважень.

Особлива увага приділялася перевірці того, що система не допускає пропуску етапів життєвого циклу або виконання дій, не передбачених логікою системи. Це підтверджує правильність реалізації бізнес-правил і забезпечує контроль за процесом документообігу.

3.6.3 Тестування впливу ролей користувачів на зміну станів

Зміна станів електронного документа у системі безпосередньо залежить від ролей користувачів, які виконують відповідні дії. У межах тестування перевірялася коректність реалізації рольової моделі та її вплив на життєвий цикл документа.

Для цього виконувалися сценарії з використанням облікових записів користувачів з різними ролями доступу. Перевірялося, що користувачі можуть ініціювати зміну стану документа лише у межах своїх повноважень, а спроби виконання заборонених дій коректно обробляються системою. Такий підхід дозволив підтвердити, що рольова модель доступу є узгодженою з логікою життєвого циклу документа.

3.6.4 Перевірка збереження стану та цілісності даних

Важливим аспектом тестування є перевірка збереження поточного стану документа та пов'язаних із ним даних після виконання операцій. У процесі тестування перевірялося, що після кожної зміни стану документа відповідна інформація коректно зберігається у базі даних і доступна для подальшої обробки.

Крім того, оцінювалася поведінка системи у разі повторного відкриття документа, оновлення сторінки або повторного входу користувача до системи. Це дозволило переконатися, що стан документа не втрачається та не змінюється некоректно у процесі експлуатації.

3.6.5 Узгодженість з UML-діаграмою станів

Результати тестування підтвердили узгодженість між UML-діаграмою станів, розробленою у Розділі 2, та фактичною програмною реалізацією системи. Усі перевірені сценарії продемонстрували, що програмна система підтримує визначену модель життєвого циклу електронного документа та забезпечує коректне виконання переходів між станами.

Отже, тестування реалізації життєвого циклу електронного документа підтвердило, що програмна реалізація повністю відповідає проєктним рішенням і забезпечує цілісність та керованість процесів електронного документообігу.

3.6.6 Результати тестування життєвого циклу електронного документа

Для формалізації результатів тестування реалізації життєвого циклу електронного документа використано табличне подання тестових сценаріїв. Таблиці дозволяють наочно відобразити відповідність між діями користувачів, очікуваною поведінкою системи та фактичними результатами виконання тестів.

Тестові сценарії сформовано на основі UML-діаграми станів електронного документа, розробленої у Розділі 2, та охоплюють основні переходи між станами,

що реалізуються у процесі електронного документообігу. Перевірка виконувалася для типових сценаріїв використання системи з урахуванням ролей користувачів та бізнес-правил (результати тестування відображено в таблицях 3.1, 3.2, 3.3, 3.4).

Таблиця 3.1 – Результати тестування ініціалізації та початкової обробки документа

№ з/п	Сценарій тестування	Початкові умови	Дії користувача	Очікуваний результат	Фактичний результат
1.	Створення нового документа	Користувач автентифікований	Створення нового документа з обов'язковим і атрибутами	Документ створено, ініціалізовано початковий стан	Відповідає
2.	Збереження документа	Документ створений	Збереження документа без помилок	Дані збережені, стан не змінено	Відповідає
3.	Повторне відкриття документа	Документ існує в системі	Відкриття документа	Поточний стан відображається коректно	Відповідає

Таблиця 3.2 – Результати тестування переходів документа в процесі обробки

№ з/п	Сценарій тестування	Початкові умови	Дії користувача	Очікуваний результат	Фактичний результат
1.	Передача документа на обробку	Документ ініціалізований	Запуск процесу обробки	Документ переходить у наступний стан	Відповідає
2.	Завершення етапу обробки	Активний етап обробки	Підтвердження завершення етапу	Стан документа оновлюється	Відповідає
3.	Спроба пропуску етапу	Активний етап	Спроба виконати недопустимий перехід	Перехід заблоковано системою	Відповідає

Таблиця 3.3 – Результати тестування обробки відхилень та повернення документа

№ з/п	Сценарій тестування	Початкові умови	Дії користувача	Очікуваний результат	Фактичний результат
1.	Відхилення документа	Документ на етапі обробки	Виконання дії відхилення	Документ переходить у відповідний стан	Відповідає
2.	Повернення на доопрацювання	Документ відхилений	Ініціація доопрацювання	Доступне редагування документа	Відповідає
3.	Повторна обробка	Документ доопрацьований	Повторний запуск процесу	Коректний перехід до наступного етапу	Відповідає

Таблиця 3.4 – Результати тестування завершення життєвого циклу документа

№ з/п	Сценарій тестування	Початкові умови	Дії користувача	Очікуваний результат	Фактичний результат
1.	Завершення обробки документа	Усі етапи виконані	Завершення процесу	Документ переходить у завершальний стан	Відповідає
2.	Обмеження змін	Документ завершений	Спроба редагування	Редагування заборонено	Відповідає
3.	Архівування документа	Документ завершений	Архівування документа	Документ доступний лише для перегляду	Відповідає

3.7 Тестування основних бізнес-процесів системи

Після перевірки коректності реалізації життєвого циклу електронного документа, що було детально розглянуто у підрозділі 3.6, наступним етапом тестування є оцінка функціонування основних бізнес-процесів СЕД. На відміну від тестування окремих переходів між станами документа, тестування бізнес-процесів спрямоване на перевірку узгодженої роботи всіх компонентів системи в межах наскрізних сценаріїв використання [29, 30].

Тестування бізнес-процесів проводилося з урахуванням варіантів використання, визначених у Розділі 2, та охоплювало повний цикл роботи з електронними документами. Особлива увага приділялася взаємодії користувачів із різними ролями доступу, коректності виконання бізнес-операцій та відповідності поведінки системи визначеним правилам документообігу.

3.7.1 Тестування бізнес-процесу створення та реєстрації документа

Першим бізнес-процесом, що підлягав тестуванню, є створення та реєстрація електронного документа. У процесі тестування перевірялося, що користувач із відповідними повноваженнями може створити новий документ, заповнити необхідні атрибути та зберегти його у системі.

Також перевірялася коректність первинної реєстрації документа, його відображення у списках активних документів та доступність для подальших етапів обробки. Результати тестування підтвердили, що бізнес-процес створення та реєстрації документа реалізований коректно та відповідає визначеним вимогам.

3.7.2 Тестування бізнес-процесу маршрутизації документа

Наступним етапом тестування є перевірка бізнес-процесу маршрутизації документа. У межах цього процесу перевірялося, що документ може бути переданий на обробку відповідно до визначеного маршруту, а система коректно визначає поточний етап виконання.

Особлива увага приділялася перевірці послідовності проходження етапів маршруту та неможливості пропуску окремих етапів. Тестування підтвердило, що система забезпечує контроль за виконанням маршруту та коректно відображає стан обробки документа для відповідних користувачів.

3.7.3 Тестування бізнес-процесу погодження документа

Бізнес-процес погодження документа є одним із ключових процесів СЕД. У процесі тестування перевірялося, що користувачі з відповідними ролями можуть виконувати дії з погодження або відхилення документа у межах своїх повноважень.

Також оцінювалася коректність впливу результатів погодження на подальший перебіг процесу обробки документа. У разі успішного погодження система забезпечує перехід до наступного етапу, а у разі відхилення – відповідну зміну стану документа та припинення поточного маршруту. Результати тестування підтвердили правильність реалізації бізнес-логіки погодження.

3.7.4 Тестування бізнес-процесу підписання документа

Після завершення погодження документ може бути переданий на підписання. У межах тестування перевірялося, що можливість підписання доступна лише за умови виконання всіх попередніх етапів обробки, а також що дії з підписання виконуються лише користувачами з відповідними повноваженнями.

Результати тестування підтвердили, що факт підписання документа коректно фіксується у системі та впливає на подальший перебіг життєвого циклу документа відповідно до визначеної логіки.

3.7.5 Тестування бізнес-процесу архівування документа

Завершальним бізнес-процесом є архівування електронного документа. У процесі тестування перевірялося, що архівування можливе лише після завершення основних етапів обробки документа, а доступ до архівних документів обмежується відповідно до ролей користувачів.

Тестування підтвердило, що архівні документи не можуть бути змінені або повторно включені до активного документообігу без виконання відповідних процедур, що забезпечує цілісність та завершеність процесу документообігу.

3.7.6 Узагальнення результатів тестування бізнес-процесів

Результати тестування основних бізнес-процесів підтвердили їх коректну реалізацію та узгодженість із проєктними рішеннями, визначеними у UML-моделях. Перевірені сценарії продемонстрували стабільну та передбачувану поведінку системи в умовах типової експлуатації.

Отже, тестування бізнес-процесів доповнює результати тестування життєвого циклу електронного документа, наведені у підрозділі 3.6, та підтверджує готовність системи до переходу на етап впровадження та подальшої підтримки.

3.8 Аналіз результатів тестування системи

Аналіз результатів тестування є завершальним етапом оцінки якості програмної реалізації СЕД та спрямований на узагальнення отриманих результатів, виявлення сильних сторін реалізації та підтвердження готовності системи до подальшого впровадження. На цьому етапі результати тестування розглядаються комплексно, з урахуванням як описових перевірок, так і формалізованих результатів, наведених у табличному вигляді [13, 14].

У межах даної магістерської роботи аналіз результатів тестування здійснювався на основі результатів, отриманих у підрозділах 3.6 та 3.7. Особливу роль при цьому відіграють результати табличного тестування життєвого циклу електронного документа, наведені у підрозділі 3.6.6, які дозволяють формально підтвердити відповідність між UML-діаграмою станів та фактичною поведінкою програмної системи.

3.8.1 Узагальнення результатів тестування життєвого циклу документа

Результати тестування реалізації життєвого циклу електронного документа, представлені у вигляді таблиць, підтвердили коректність ініціалізації документа, правильність переходів між етапами обробки та дотримання визначених бізнес-

правил. Усі перевірені сценарії виконалися відповідно до очікуваної логіки, а система не допускає некоректних або несанкціонованих переходів між станами.

Табличні результати тестування наочно продемонстрували, що кожна зміна стану документа ініціюється відповідними діями користувачів і супроводжується коректним оновленням даних у системі. Це свідчить про повну узгодженість між проектною моделлю життєвого циклу документа та її програмною реалізацією.

3.8.2 Аналіз результатів тестування бізнес-процесів

Аналіз результатів тестування основних бізнес-процесів СЕД показав, що реалізовані сценарії створення, маршрутизації, погодження, підписання та архівування документів виконуються коректно та послідовно. Проведені перевірки підтвердили правильність взаємодії користувачів із різними ролями доступу та дотримання обмежень, визначених бізнес-логікою системи.

Узгоджена робота бізнес-процесів у межах наскрізних сценаріїв підтверджує цілісність архітектури системи та стабільність її поведінки під час виконання типових операцій з електронними документами.

3.8.3 Аналіз виявлених зауважень та їх усунення

У процесі тестування були виявлені окремі зауваження, що стосувалися уточнення логіки виконання окремих операцій та підвищення зручності користування системою. Виявлені недоліки не мали критичного характеру та не впливали на коректність реалізації життєвого циклу документа або основних бізнес-процесів.

Після усунення виявлених зауважень виконано повторне тестування відповідних сценаріїв, результати якого підтвердили коректність внесених змін. Такий підхід дозволив підвищити стабільність роботи системи та забезпечити відповідність реалізації визначеним вимогам.

3.8.4 Оцінка стабільності та передбачуваності роботи системи

Результати тестування засвідчили стабільну та передбачувану поведінку системи у межах перевірених сценаріїв використання. Система коректно обробляє дії користувачів, забезпечує збереження станів документів і не допускає виникнення неконтрольованих ситуацій.

Також підтверджено, що система коректно реагує на типові помилкові дії користувачів, зберігаючи цілісність даних та забезпечуючи інформування про некоректні операції. Це є важливим показником готовності системи до використання у реальних умовах експлуатації.

3.8.5 Загальна оцінка результатів тестування

Узагальнюючи результати проведеного тестування, можна зробити висновок, що розроблена СЕД відповідає сформульованим вимогам, є функціонально повною та стабільною у роботі. Табличні результати тестування життєвого циклу документа разом із описовими результатами тестування бізнес-процесів підтверджують коректність програмної реалізації та її узгодженість із проєктними рішеннями.

Отримані результати свідчать про готовність системи до переходу на етап впровадження в експлуатацію та організації подальшої підтримки, що розглядається у наступних підрозділах роботи.

3.9 Впровадження системи в експлуатацію

Впровадження програмної системи в експлуатацію є завершальним етапом її розроблення та тестування і полягає у підготовці системи до практичного використання в реальних умовах. На цьому етапі результати проєктування та тестування трансформуються у готовий до використання програмний продукт, який може бути інтегрований у діяльність організації.

У межах даної магістерської роботи впровадження СЕД розглядається як сукупність організаційних і технічних заходів, спрямованих на забезпечення стабільної роботи системи, готовності інфраструктури та підготовки користувачів до роботи з програмним забезпеченням.

Важливо на цьому етапі відзначити, що постановою Кабінету Міністрів України від 21 лютого 2025 року № 205 «Деякі питання створення, адміністрування та забезпечення функціонування засобу інформатизації» затверджені, зокрема, Обов'язкові вимоги до створення (модернізації, модифікації, розвитку), адміністрування та забезпечення функціонування засобу інформатизації, відповідно до яких існує дослідна та промислова експлуатація засобу інформатизації [5].

3.9.1 Підготовка системи до впровадження

Підготовка системи до впровадження передбачає перевірку готовності програмного забезпечення до експлуатації, а також відповідність апаратно-програмного середовища вимогам системи. На цьому етапі виконуються остаточні налаштування серверної частини, бази даних та файлового сховища.

Особлива увага приділяється перевірці коректності конфігураційних параметрів, що впливають на безпеку та стабільність роботи системи. Це дозволяє мінімізувати ризики виникнення помилок на початковому етапі експлуатації.

3.9.2 Розгортання серверної та клієнтської частин

Розгортання системи передбачає встановлення та запуск серверної частини програмного забезпечення, а також забезпечення доступу користувачів до клієнтської частини системи через веб-інтерфейс. Серверна частина системи розгортається у середовищі, що забезпечує постійний доступ до мережі та необхідний рівень захисту інформації.

Клієнтська частина системи не потребує окремого встановлення, оскільки доступ до неї здійснюється за допомогою веб-браузера. Такий підхід спрощує процес впровадження та знижує вимоги до робочих місць користувачів.

3.9.3 Початкове налаштування та ініціалізація даних

Після розгортання системи виконується початкове налаштування, яке включає створення облікових записів користувачів, визначення ролей доступу та налаштування базових параметрів системи. Також на цьому етапі можуть бути створені типові маршрути обробки документів, які використовуватимуться у процесі експлуатації.

Ініціалізація даних дозволяє підготувати систему до роботи з реальними документами та забезпечує узгодженість початкового стану системи з визначеними бізнес-правилами.

3.9.4 Дослідна експлуатація системи

Перед повноцінним введенням системи в експлуатацію доцільним є проведення етапу дослідної експлуатації. На цьому етапі система використовується обмеженим колом користувачів для виконання реальних або наближених до реальних задач документообігу.

Дослідна експлуатація дозволяє виявити можливі недоліки, які не були зафіксовані під час тестування, а також оцінити зручність роботи системи з точки зору кінцевих користувачів. За результатами дослідної експлуатації можуть бути внесені незначні корективи до налаштувань або логіки роботи системи.

3.9.5 Введення системи в промислову експлуатацію

Після завершення дослідної експлуатації та усунення виявлених зауважень система може бути введена в промислову експлуатацію. На цьому етапі система

стає основним інструментом для роботи з електронними документами в межах організації.

Введення в промислову експлуатацію передбачає переведення користувачів на роботу з новою системою, а також організацію процесів супроводу та підтримки, що забезпечують безперебійну роботу програмного забезпечення.

3.10 Підтримка та супровід системи

Після впровадження СЕД в промислову експлуатацію важливого значення набуває організація її підтримки та супроводу. Підтримка системи спрямована на забезпечення стабільної роботи програмного забезпечення, оперативне усунення можливих несправностей, а також підтримання актуальності функціоналу відповідно до потреб користувачів і змін у нормативних вимогах [15, 18].

У межах даної магістерської роботи супровід системи розглядається як сукупність технічних і організаційних заходів, що забезпечують безперервність роботи СЕД протягом усього періоду її експлуатації.

3.10.1 Технічна підтримка та моніторинг роботи системи

Технічна підтримка системи передбачає постійний контроль її працездатності та своєчасне реагування на виникнення збоїв або відхилень у роботі. До основних завдань технічної підтримки належать контроль доступності системи, перевірка коректності виконання основних операцій та аналіз журналів подій.

Моніторинг роботи системи дозволяє виявляти потенційні проблеми на ранніх етапах, що сприяє підвищенню стабільності та надійності програмного забезпечення. Особлива увага приділяється контролю операцій, пов'язаних із обробкою електронних документів, оскільки порушення у цих процесах можуть впливати на діяльність користувачів.

3.10.2 Оновлення та розвиток функціональності системи

У процесі експлуатації системи може виникати потреба у розширенні або модифікації її функціональних можливостей. Підтримка системи передбачає можливість внесення змін до програмного забезпечення з урахуванням нових вимог, змін у бізнес-процесах або нормативно-правових актах.

Оновлення системи виконуються з дотриманням принципів зворотної сумісності, що дозволяє уникнути порушення роботи вже реалізованого функціоналу. Перед впровадженням оновлень доцільним є виконання повторного тестування відповідних компонентів системи для підтвердження їх коректної роботи.

3.10.3 Забезпечення безпеки та збереження даних

Важливою складовою супроводу системи є забезпечення безпеки даних та захисту інформації. Це включає контроль доступу до системи, перевірку коректності роботи механізмів автентифікації та авторизації, а також регулярний аналіз журналів подій з метою виявлення підозрілих дій.

Окрему увагу приділяється забезпеченню збереження даних електронних документів. Для цього у процесі супроводу системи доцільно застосовувати регулярне резервне копіювання даних, що дозволяє відновити працездатність системи у разі виникнення збоїв або втрати інформації.

3.10.4 Підтримка користувачів та експлуатаційна документація

Підтримка користувачів є важливим аспектом супроводу СЕД. Вона включає надання консультацій щодо роботи з системою, допомогу у вирішенні типових проблем, а також інформування про нові можливості або зміни у функціонуванні системи.

Для забезпечення ефективної підтримки доцільним є використання експлуатаційної документації, яка містить опис основних функцій системи, рекомендації щодо роботи з електронними документами та правила використання системи. Наявність такої документації сприяє швидкій адаптації користувачів та зниженню кількості помилок під час роботи з системою.

3.10.5 Перспективи подальшого супроводу та розвитку системи

Організація підтримки та супроводу створює основу для подальшого розвитку СЕД. У перспективі система може бути розширена шляхом інтеграції з іншими інформаційними системами, впровадження додаткових сервісів або оптимізації наявних бізнес-процесів.

Отже, у третьому розділі магістерської роботи виконано комплексне дослідження процесів тестування, впровадження та підтримки розробленої СЕД. Основну увагу було зосереджено на перевірці коректності програмної реалізації, оцінці стабільності роботи системи та підтвердженні її готовності до практичного використання.

У ході тестування визначено цілі та завдання перевірки, обґрунтовано вибір підходів до тестування та описано умови його проведення. Проведені перевірки охопили реалізацію життєвого циклу електронного документа та основні бізнес-процеси системи. Результати тестування, зокрема табличне подання перевірки життєвого циклу документа, підтвердили відповідність фактичної поведінки системи проєктним рішенням, визначеним UML-діаграмами.

Аналіз отриманих результатів показав, що розроблена система забезпечує стабільну та передбачувану роботу в межах перевірених сценаріїв використання, коректно обробляє дії користувачів і не допускає порушення визначених бізнес-правил. Виявлені зауваження мали некритичний характер і були усунені на етапі доопрацювання, що позитивно вплинуло на якість програмної реалізації.

У розділі також розглянуто процес впровадження системи в експлуатацію та основні підходи до організації її підтримки і супроводу. Визначено, що обрана

архітектура та реалізований функціонал створюють умови для стабільної експлуатації системи, її подальшого розвитку та адаптації до змін вимог користувачів.

Отже, результати третього розділу підтверджують, що СЕД є готовою до використання, відповідає поставленим вимогам та може бути впроваджена у практичну діяльність. Отримані результати створюють основу для формування загальних висновків магістерської роботи.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Охорона праці є системою правових, соціально-економічних, організаційних, технічних та лікувально-профілактичних заходів і засобів, спрямованих на збереження життя, здоров'я та працездатності людини в процесі трудової діяльності. Забезпечення належних умов праці є обов'язковою складовою організації робочого процесу та регламентується законодавством України [34].

4.1 Охорона праці

У процесі розроблення та експлуатації СЕД значну частину робочого часу користувачі проводять за персональними комп'ютерами, використовуючи програмно-технічні засоби, мережеве обладнання та серверну інфраструктуру. Така діяльність пов'язана з впливом ряду виробничих факторів, які можуть негативно впливати на стан здоров'я працівників за умови недотримання вимог охорони праці.

Відповідно до Закону України «Про охорону праці», роботодавець зобов'язаний створити безпечні та нешкідливі умови праці, забезпечити дотримання нормативних вимог щодо організації робочих місць, режиму праці та відпочинку, а також провести відповідні інструктажі з охорони праці. Для робіт, пов'язаних із використанням персональних комп'ютерів, додатково застосовуються санітарні норми та правила, що регламентують параметри мікроклімату, освітлення та ергономіки робочого місця.

Особливістю праці користувачів інформаційних систем є поєднання розумового навантаження з тривалим статичним положенням тіла та зоровою напругою. За відсутності належної організації робочого місця це може призводити до підвищеної втомлюваності, зниження працездатності та розвитку професійних захворювань. У зв'язку з цим важливим є дотримання вимог щодо раціональної організації праці, правильного розміщення обладнання та оптимізації режиму роботи.

У даному розділі розглядаються основні вимоги охорони праці, характерні для умов експлуатації СЕД, аналізуються потенційні небезпечні та шкідливі виробничі фактори, а також визначаються заходи щодо забезпечення безпечних та комфортних умов праці користувачів.

4.2 Характеристика робочого місця користувача

Робоче місце користувача СЕД розташоване в офісному приміщенні та призначене для виконання робіт, пов'язаних із обробкою інформації, використанням програмних засобів і тривалою роботою за персональним комп'ютером. Організація робочого місця повинна відповідати чинним нормативним вимогам з охорони праці та забезпечувати безпечні й комфортні умови праці [32, 34].

Відповідно до санітарних норм і правил, приміщення для роботи з персональними комп'ютерами має відповідати вимогам щодо площі, об'єму та висоти. Площа приміщення на одне робоче місце повинна становити не менше 6 м², об'єм – не менше 20 м³, а висота приміщення – не менше 3 м. Дотримання цих вимог забезпечує належний повітрообмін та комфортні умови перебування працівників.

Робоче місце обладнане персональним комп'ютером із сучасним монітором, системним блоком, клавіатурою та маніпулятором типу «миша». Також використовуються офісний стіл та регульоване крісло, які забезпечують правильне положення тіла користувача під час роботи. Конструкція меблів повинна відповідати антропометричним характеристикам працівника та дозволяти змінювати положення тіла у процесі роботи.

Монітор розміщується на відстані 60-70 см від очей користувача, а його верхній край повинен знаходитися на рівні очей або дещо нижче. Клавіатура та маніпулятор розташовуються на робочій поверхні столу таким чином, щоб забезпечити зручне положення рук та зменшити навантаження на м'язи плечового поясу.

Робоче місце забезпечується системою загального та місцевого освітлення, що гарантує достатній рівень освітленості робочої поверхні. Для підтримання оптимального мікроклімату приміщення обладнується системами вентиляції або кондиціонування повітря.

Електроживлення персонального комп'ютера та периферійних пристроїв здійснюється через справні електричні мережі із заземленням, що відповідають вимогам електробезпеки. Кабелі та електричні з'єднання розміщуються таким чином, щоб унеможливити їх механічне пошкодження та створення небезпечних ситуацій.

4.3 Вимоги до електробезпеки

Електробезпека є важливою складовою системи охорони праці та спрямована на запобігання ураженню електричним струмом під час експлуатації електрообладнання. У процесі роботи користувачів СЕД використовуються персональні комп'ютери, периферійні пристрої, мережеве обладнання та інші електротехнічні засоби, що потребує дотримання вимог електробезпеки.

Відповідно до чинних нормативних документів, електрообладнання, яке використовується на робочих місцях, повинно відповідати вимогам безпеки та мати справну ізоляцію струмоведучих частин. Персональні комп'ютери та офісне обладнання належать до електроустановок класу II, що передбачає використання подвійної або посиленої ізоляції та зменшує ризик ураження електричним струмом [34].

Для забезпечення електробезпеки робочі місця повинні бути обладнані електричними розетками із заземлювальними контактами. Забороняється використання пошкоджених кабелів, несправних подовжувачів та саморобних електричних з'єднань. Усі електричні з'єднання мають бути надійно закріплені та захищені від механічних пошкоджень.

Важливим заходом електробезпеки є застосування стабілізаторів напруги або джерел безперебійного живлення, що дозволяє захистити обладнання від перепадів

напруги та запобігти аварійним ситуаціям. Регулярний технічний огляд електрообладнання та своєчасне усунення виявлених несправностей є обов'язковими умовами безпечної експлуатації.

Під час роботи з електрообладнанням працівникам забороняється виконувати будь-які ремонтні роботи без відповідної підготовки та дозволу. У разі виявлення несправностей необхідно негайно вимкнути обладнання з мережі та повідомити відповідальних осіб.

4.4 Пожежна безпека

Пожежна безпека є важливою складовою системи охорони праці та спрямована на запобігання виникненню пожеж, обмеження їх поширення та зменшення можливих негативних наслідків для людей і матеріальних цінностей. У приміщеннях, де експлуатується комп'ютерна техніка та інше електричне обладнання, необхідно забезпечити дотримання вимог пожежної безпеки.

Основними причинами виникнення пожеж у офісних та виробничих приміщеннях є несправність електрообладнання, перевантаження електромереж, короткі замикання, а також порушення правил експлуатації електроприладів. Тому особлива увага приділяється справності електричних мереж, стану ізоляції проводів і правильному використанню електричного обладнання [33].

Приміщення, у яких розташовані робочі місця користувачів СЕД, повинні бути обладнані первинними засобами пожежогасіння, зокрема вогнегасниками, доступними для швидкого використання у разі виникнення пожежі. Кількість і тип вогнегасників визначаються відповідно до нормативних вимог та характеристик приміщення.

Важливим елементом забезпечення пожежної безпеки є наявність систем пожежної сигналізації та оповіщення, які дозволяють своєчасно виявити загрозу та організувати евакуацію працівників. Шляхи евакуації повинні бути вільними, чітко позначеними та освітленими.

Забороняється використання відкритого вогню, зберігання легкозаймистих матеріалів у робочих приміщеннях, а також самовільне підключення електрообладнання до мережі. Працівники повинні бути ознайомлені з правилами пожежної безпеки та порядком дій у разі виникнення пожежі.

4.5 Ергономіка та організація робочого місця

Ергономіка робочого місця є важливим чинником забезпечення безпечних та комфортних умов праці користувачів персональних комп'ютерів. Раціональна організація робочого місця дозволяє зменшити фізичне та психофізіологічне навантаження, запобігти виникненню професійних захворювань і підвищити ефективність роботи.

Робоче місце користувача СЕД повинно бути організоване з урахуванням антропометричних, фізіологічних і психофізіологічних особливостей людини. Основні елементи робочого місця – стіл, крісло, монітор, клавіатура та маніпулятори – повинні розміщуватися таким чином, щоб забезпечити зручне положення тіла та мінімізувати статичні навантаження [20, 21].

Висота робочого столу повинна забезпечувати розміщення передпліч на його поверхні під кутом, близьким до прямого. Крісло має бути регульованим по висоті та нахилу спинки, що дозволяє підтримувати природне положення хребта. Ноги користувача повинні повністю спиратися на підлогу або підставку для ніг.

Монітор розташовується на відстані 60-70 см від очей користувача, при цьому верхній край екрана має знаходитися на рівні очей або трохи нижче. Це зменшує навантаження на м'язи шиї та органи зору. Клавіатура і маніпулятор повинні бути розташовані так, щоб забезпечити природне положення кистей рук та уникнути перенапруження.

Важливою складовою ергономічної організації праці є дотримання раціонального режиму роботи та відпочинку. Під час тривалої роботи за комп'ютером рекомендується робити регулярні перерви для відпочинку очей і

зміни положення тіла. Чергування видів діяльності сприяє зниженню втомлюваності та підвищенню працездатності.

Отже, у четвертому розділі магістерської роботи розглянуто основні питання охорони праці та безпеки в надзвичайних ситуаціях, пов'язані з розробленням і експлуатацією СЕД. Аналіз виконано з урахуванням специфіки праці користувачів, діяльність яких пов'язана з тривалою роботою за персональними комп'ютерами в офісному середовищі.

У межах розділу охарактеризовано умови праці користувачів системи, визначено вимоги до організації робочого місця, електробезпеки, пожежної безпеки та ергономіки. Розглянуті заходи спрямовані на зниження впливу небезпечних і шкідливих виробничих факторів, забезпечення безпечних і комфортних умов праці, а також запобігання виникненню надзвичайних ситуацій.

Особливу увагу приділено раціональній організації робочого місця, дотриманню вимог електробезпеки та пожежної безпеки, що є ключовими чинниками збереження здоров'я працівників та безпечної експлуатації комп'ютерної техніки. Урахування ергономічних вимог і режиму праці та відпочинку сприяє зниженню втомлюваності та підвищенню ефективності роботи користувачів системи.

Отже, результати розділу підтверджують, що дотримання вимог охорони праці та безпеки в надзвичайних ситуаціях є необхідною умовою безпечної та ефективної експлуатації СЕД, а запропоновані заходи відповідають чинним нормативним вимогам і сприяють створенню належних умов праці.

ВИСНОВКИ

У магістерській роботі розв'язано науково-технічну задачу проєктування та розроблення веб-орієнтованої СЕД, яка забезпечує автоматизацію процесів створення, маршрутизації, погодження, підписання та архівування електронних документів з урахуванням визначеного життєвого циклу документа та ролей користувачів.

У ході виконання роботи проведено аналіз предметної області електронного документообігу, нормативно-правового забезпечення та існуючих програмних рішень. На основі результатів аналізу сформульовано функціональні та нефункціональні вимоги до системи, що стали підґрунтям для подальшого проєктування та програмної реалізації.

Теоретичне значення роботи полягає у формалізації процесів електронного документообігу з використанням UML-моделей, зокрема діаграм прецедентів, класів, діяльності та станів. Особливу увагу приділено моделюванню життєвого циклу електронного документа, що дозволило чітко визначити допустимі стани документа та правила переходів між ними. Отримані теоретичні результати можуть бути використані під час проєктування аналогічних інформаційних систем документообігу.

Практичне значення роботи полягає у створенні працездатної веб-орієнтованої СЕД, реалізованої з використанням сучасного програмного стеку. Розроблена система підтримує повний життєвий цикл електронного документа, забезпечує багатокористувацьку роботу, контроль доступу за ролями, ведення журналу дій та збереження історії змін документів. Реалізований функціонал може бути безпосередньо використаний в організаціях для оптимізації внутрішніх управлінських процесів.

Аналіз наукових публікацій підтверджує актуальність тематики веб-орієнтованих систем документообігу та доцільність застосування формалізованих підходів до їх проєктування.

Наукова новизна отриманих результатів полягає у поєднанні формалізованої UML-моделі життєвого циклу електронного документа з її повною програмною реалізацією у веб-орієнтованій архітектурі. Запропонований підхід забезпечує узгодженість між етапами проєктування, реалізації та тестування системи, що підвищує керованість та прозорість процесів документообігу.

У роботі продемонстровано можливість практичного застосування підходу Spec-Driven Development для розроблення веб-орієнтованих СЕД. Використання UML-моделей як формальних специфікацій дозволило забезпечити узгодженість між вимогами, архітектурними рішеннями, програмною реалізацією та тестуванням. Такий підхід зменшує ризик логічних помилок у реалізації життєвого циклу документа та створює основу для подальшого масштабування і розвитку системи.

Якісні показники отриманих результатів підтверджуються повнотою реалізації визначених вимог, стабільною та передбачуваною поведінкою системи під час виконання основних бізнес-процесів, а також відповідністю програмної реалізації UML-моделям. Кількісні показники відображені у результатах тестування, зокрема у перевірці всіх визначених станів життєвого циклу документа та основних сценаріїв використання системи.

Достовірність отриманих результатів обґрунтована використанням загальноприйнятих методів аналізу, проєктування та програмної реалізації, застосуванням UML як стандартної мови моделювання, а також результатами тестування, що підтвердили коректність реалізації бізнес-логіки та життєвого циклу електронного документа.

Результати магістерської роботи рекомендується використовувати для впровадження систем електронного документообігу в організаціях, а також як основу для подальшого розвитку функціоналу, зокрема інтеграції з зовнішніми інформаційними системами, розширення механізмів електронного підпису та масштабування системи для використання в розподілених середовищах.

Практична цінність отриманих результатів підтверджується апробацією розробленої СЕД в умовах діяльності державної установи «Відкриті публічні

фінанси». У ході тестової експлуатації система використовувалась для підтримки окремих процесів внутрішнього документообігу, зокрема створення, реєстрації, погодження та маршрутизації електронних документів.

За результатами апробації встановлено, що запропоновані архітектурні рішення, організація бізнес-процесів та технічна реалізація є обґрунтованими і доцільними для використання в діяльності установи. Розроблений програмний продукт продемонстрував можливість подальшого функціонального розширення та може слугувати основою для впровадження повноцінної СЕД в державній установі після відповідної адаптації до внутрішніх регламентів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про електронні документи та електронний документообіг» [Електронний ресурс]. <https://zakon.rada.gov.ua/laws/show/851-15#Text>
2. Закон України «Про електронну ідентифікацію та електронні довірчі послуги» [Електронний ресурс]. <https://zakon.rada.gov.ua/laws/show/2155-19#Text>
3. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [Електронний ресурс]. <https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text>
4. Постанова Кабінету Міністрів України від 17 січня 2018 року № 55 «Деякі питання документування управлінської діяльності» [Електронний ресурс]. <https://zakon.rada.gov.ua/laws/show/55-2018-%D0%BF#Text>
5. Постанова Кабінету Міністрів України від 21 лютого 2025 року № 205 «Деякі питання створення, адміністрування та забезпечення функціонування засобу інформатизації» [Електронний ресурс]. <https://zakon.rada.gov.ua/laws/show/205-2025-%D0%BF#Text>
6. Документація на програмний продукт АСКОД [Електронний ресурс]. <https://askod.ua/home>
7. Документація на програмний продукт Medoc [Електронний ресурс]. https://medoc.ua/elektronniy_dokumentoobig
8. Документація на програмний продукт Megapolis.DocNet [Електронний ресурс]. <https://softline.company.ua/solutions.html>
9. ДСТУ 4156-2003 «Інформаційні технології. Електронний документообіг. Основні положення»
10. ДСТУ 4163:2020 «Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлення документів»
11. ДСТУ 4145-2002 «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис»

12. ДСТУ ISO/IEC 27001:2015 «Системи управління інформаційною безпекою»
13. Sommerville I. Software Engineering. 10th ed. Pearson Education, 2016
14. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. McGraw-Hill, 2019
15. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Addison-Wesley, 2021
16. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, 2004
17. Larman C. Applying UML and Patterns. 3rd ed. Prentice Hall, 2004
18. Kleppmann M. Designing Data-Intensive Applications. O'Reilly Media, 2017
19. Nielsen J. Usability Engineering. Academic Press, 2012
20. ISO 9241-110:2020 Ergonomics of human-system interaction — Dialogue principles
21. ISO 9241-5:1998 Ergonomic requirements for office work with visual display terminals
22. Node.js Official Documentation [Електронний ресурс]. <https://nodejs.org>
23. Express.js Official Documentation [Електронний ресурс]. <https://expressjs.com>
24. MongoDB Official Documentation [Електронний ресурс]. <https://www.mongodb.com/docs>
25. Mongoose ODM Documentation [Електронний ресурс]. <https://mongoosejs.com>
26. EJS Template Engine Documentation [Електронний ресурс]. <https://ejs.co>
27. Dierks T., Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446
28. Krawczyk H., Bellare M., Canetti R. HMAC: Keyed-Hashing for Message Authentication. RFC 2104

29. Гуляєв В. М., Кравченко О. В. Інформаційні системи та технології в документальному забезпеченні управління. КНЕУ, 2018
30. Краус Н. М., Марущак А. І. Системи електронного документообігу: методологія проектування та впровадження. Львів, 2020
31. Іванченко В. В. Електронний цифровий підпис: технічні та правові аспекти. Харків, 2019
32. ДСанПіН 3.3.2-007-98 «Гігієнічні вимоги до відеодисплейних терміналів та організації роботи»
33. ДБН В.2.5-28:2006 «Природне і штучне освітлення»
34. НПАОП 0.00-1.31-99 «Правила охорони праці під час роботи з електронно-обчислювальними машинами»
35. Microsoft Spec-Driven Development. [Електронний ресурс]. <https://developer.microsoft.com/blog/spec-driven-development-spec-kit>
36. Aládé S. M. Design and Implementation of a Web-Based Document Management System. International Journal of Information Technology and Computer Science, 2023. URL: https://www.researchgate.net/publication/370534584_Design_and_Implementation_of_a_Web-based_Document_Management_System (дата звернення: 01.12.2025)
37. Nagrama N. D., De Leon J. P., Dela Cruz R. M. Web-Based Document Management System for Organizational Efficiency. International Journal of Science and Advanced Information Technology, 2024. URL: <https://www.warse.org/IJSAIT/static/pdf/file/ijisait031332024.pdf> (дата звернення: 01.12.2025)
38. Han J., Wang C., Miao J., Lu M., Wang Y., Shi J. Research on Electronic Document Management System Based on Cloud Computing. Computers, Materials & Continua, 2021. URL: https://www.researchgate.net/publication/348204161_Research_on_Electronic_Document_Management_System_Based_on_Cloud_Computing (дата звернення: 01.12.2025)
39. Ilagan J. V. A Study of the Electronic Document Management Systems (EDMS) of One Higher Education Institution: Benefits, Challenges and Implementation Strategies.

TWIST Journal, 2024. URL: <https://twistjournal.net/twist/article/view/337> (дата звернення: 01.12.2025)

ДОДАТКИ

ДОДАТОК А



МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНА УСТАНОВА «ВІДКРИТІ ПУБЛІЧНІ ФІНАНСИ»
ДУ «ВПФ»

вул. Межигірська, 11, м. Київ, 04071, для листування: а/с 135, м. Київ, 04071, тел.: (044) 206-69-99,
E-mail: inbox@publicfinance.gov.ua, код ЄДРПОУ 41242590

05.12.2025 р. № 37

ДОВІДКА

Державна установа «Відкриті публічні фінанси» повідомляє (далі – ДУ «ВПФ», установа), що в рамках виконання магістерської кваліфікаційної роботи студентом *Стоєвим Олександром Олександровичем* за спеціальністю 121 «Інженерія програмного забезпечення» розроблено програмний продукт – систему електронного документообігу, яка пройшла апробацію в умовах діяльності установи.

Розроблена система електронного документообігу використана в тестовому режимі для забезпечення окремих процесів внутрішнього документообігу установи, зокрема процесів створення, реєстрації, погодження та маршрутизації електронних документів.

За результатами проведеної апробації встановлено, що запропоновані у роботі підходи до організації електронного документообігу, архітектурні рішення та технічна реалізація системи є обґрунтованими та доцільними. Реалізований програмний продукт демонструє логічну структуру, чітку організацію бізнес-процесів документообігу та можливість подальшого розширення функціональності.

Особливо слід відзначити, що розроблена система враховує основні вимоги до безпеки інформації, ролей користувачів та контролю станів електронних документів, що є важливими аспектами для впровадження систем електронного документообігу в діяльність установ.

За результатами апробації можна зробити висновок, що розроблена в рамках магістерської кваліфікаційної роботи система електронного документообігу може слугувати суттєвим підґрунтям для впровадження повноцінної СЕД у ДУ «ВПФ» після відповідного доопрацювання та адаптації до внутрішніх регламентів установи.

Результати магістерської роботи мають практичну цінність та можуть бути використані при подальшому розвитку інформаційних систем документообігу установи.

В.о. директора

Каріна Гарбузенко, +380506541486



Ольга ПЛЮЩИК

ДОДАТОК В

Лістинг В.1 – Програмний UML код діаграми класів:

```
@startuml
class User
{
    +ObjectId _id
    +String username
    +String passwordHash
    +String email
    +Boolean isActive
    +Date createdAt
    +Date updatedAt
}
class Role {
    +ObjectId _id
    +String name
    +String description
}
class Document {
    +ObjectId _id
    +String title
    +String type
    +String status
    +ObjectId createdBy
    +Date createdAt
    +Date updatedAt
    +ObjectId currentRouteId
}
class DocumentVersion {
    +ObjectId _id
    +ObjectId documentId
    +Number versionNumber
    +ObjectId fileId
    +String comment
    +Date createdAt
}
class Route {
    +ObjectId _id
    +String name
    +String description
}
class RouteStep {
    +ObjectId _id
    +ObjectId routeId
    +Number order
    +String roleRequired
    +String actionType // approve/sign/review
}
class Signature {
    +ObjectId _id
```

```

+ObjectId documentId
+ObjectId userId
+Date signedAt
+String certificateSerial
+String signType // qualified, advanced etc.
}
class AuditLog {
+ObjectId _id
+ObjectId userId
+ObjectId documentId
+String action
+String details
+Date timestamp
}
class FileAttachment {
+ObjectId _id
+String fileName
+String mimeType
+String storagePath
+Number size
+Date uploadedAt
+ObjectId uploadedBy
}
User "1" -- "many" Document : створює >
User "1" -- "many" DocumentVersion : створює >
User "1" -- "many" Signature : накладає >
User "1" -- "many" AuditLog : виконує >
Role "1" -- "many" User : < має
Document "1" -- "many" DocumentVersion : має >
Document "1" -- "many" Signature : має >
Document "1" -- "many" FileAttachment : має >
Document "1" -- "many" AuditLog : лог подій >
Route "1" -- "many" RouteStep : складається з >
Route "1" -- "many" Document : використовується в >
@enduml

```

Лістинг В.2 – HTML програмний код реалізації сторінки входу:

```

<div class="sufee-login d-flex align-content-center flex-wrap">
  <div class="container">
    <div class="login-content">
      <div class="login-logo">
        <a href="index.html">
          
        </a>
      </div>
      <div class="login-form">

```

```
<form name="formlogin" id="formlogin" action="/"
method="post">
<div class="form-group">
    <label>Ім'я користувача</label>
    <input type="text" id="username" name="username"
class="form-control" placeholder="Ім'я
користувача">
</div>
<div class="form-group">
    <label>Пароль користувача</label>
    <input type="password" id="password"
name="password" class="form-control"
placeholder="Пароль користувача">
</div>
<div class="checkbox">
    <label>
        <input type="checkbox"> Запам'ятати мене
    </label>
</div>
    <button type="submit" class="btn btn-success btn-
flat m-b-30 m-t-30">Вхід до системи</button>
</form>
</div>
<br>
<% if (typeof error !== "undefined" && error ==
"wrongPass") { %>
<div id="alert" class="alert alert-danger"
style="display: block; text-align:center;"
role="alert"> Невірний користувач або пароль!
<% } else if (typeof error !== "undefined" &&
error == "wrongUser") { %>
<div id="alert" class="alert alert-danger"
style="display: block; text-align:center;"
role="alert"> Невірний користувач або пароль!
<% } else { %>
<div id="alert" class="alert alert-danger"
```

```

        style="display: none; text-align:center;"
        role="alert"> Успех!
    <% } %>
</div>
</div>
</div>
</div>

```

Лістинг В.3 – JavaScript програмний код реалізації сторінки входу:

```

passport.use('login', new localStrategy({
  username: 'username',
  password: 'password'
}, function (username, password, done){
  try {
    //Знайти користувача
    userModel.findOne({ userN:username }, function(err, user){
      if (err) { console.log(err); return done(err); }
      if (!user) {
        console.log('User not found');
        return done(null, false, "wrongUser");
      }
      //перевіти пароль та переконатись, що він збігається з
      відповідним хешем в базі даних
      if(!user || !user.validatePassword(password)) {
        console.log('invalid password')
        return done(null, false, "wrongPass"); //відхилити
        вхід
      } else {
        //забезпечення наявності папок на файловому сервері
        if (!fs.existsSync(drive+user.group))
          fs.mkdirSync(drive+user.group);
        if (!fs.existsSync(drive+user.group + '/Signature'))
          fs.mkdirSync(drive+user.group + '/Signature');
        //створити хеш-пароля
        qrcode.toFile(drive+user.group+'/Signature/' +

```

```
username + '.login.qr.png', password, { color: {dark:
'#00F', light: '#0000'}, width: 300, height: 300 },
async function (err) {
  clearIncoming(username);
    return done(null, user, "Valid"); //дозволити
    вхід
  });
}
});
} catch (error) {
  return done(error);
}
}));
```