МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

Литвиненко О.Є.
"_____"_____2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ
"МАГІСТР"

Тема: Програмний засіб системи цифрового документообігу засобами .NET CORE

Виконавець: Горалік Р. М.

Керівник: к.т.н. доцент Марченко Н. Б.

Нормоконтролер: Тупота Є. В.

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютеризованих систем управління
Спеціальність 123 «Комп'ютерна інженерія»
(шифр, найменування)

Освітньо-професійна програма «Системне програмування»
Форма навчання денна

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____
«_____»_____202 __ р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи (проєкту)
Гораліка Ростислава Миколайовича
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема кваліфікаційної роботи (проєкту) Програмний засіб системи цифрового документообігу засобами .NET CORE

затверджена наказом ректора від 16.09.2022 № 1530/ст

2. Термін виконання роботи (проєкту): з 05.09.2022 по 30.11.2022

3. Вихідні дані до роботи (проєкту): програмний продукт розробити за допомогою .NET CORE.

4. Зміст пояснювальної записки:

- Аналіз існуючих застосунків системи цифрового документообігу.
- Вимоги до програмного засобу системи цифрового документообігу.
- Структура веб-застосунку системи цифрового документообігу.
- Реалізація програмного засобу системи цифрового документообігу

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

- Функціональні можливості системи цифрового документообігу.
- Інтерфейс веб застосунку системи цифрового документообігу.
- Структура програмного засобу системи цифрового документообігу.
- Результати реалізації застосунку системи цифрового документообігу.
- Результати аналізу існуючих систем цифрового документообігу.

Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Провести аналіз літератури за темою дипломної роботи. Підготувати перший розділ пояснювальної записки.	05.09.2022 – 19.09.2022	
2	Зробити вибір технологій та компонентів програмної розробки.	20.09.2022 – 01.10.2022	
3	Розробити структуру програмної розробки.	02.10.2022 – 18.10.2022	
4	Розробити програмні засоби. Підготувати другий розділ пояснювальної записки.	19.10.2022 – 10.11.2022	
5	Провести налаштування та тестування програмної розробки.	11.11.2022- 14.11.2022	
6	Завершити оформлення пояснювальної записки.	14.11.2022 – 18.11.2022	
7	Розробити текст доповіді. Оформити графічний матеріал для презентації.	19.11.2022 – 21.11.2022	

8. Дата видачі завдання: “5" вересня 2022 р.

Керівник кваліфікаційної роботи: Марченко Н. Б.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Горалік Р. М.
(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Програмний засіб системи цифрового документообігу засобами .NET CORE»: 56 с., 29 рис., 12 інформаційних джерел.

ВЕБ-ЗАСТОСУНОК, ЦИФРОВИЙ ДОКУМЕНТООБІГ, ДОКУМЕНТ, ШИФРУВАННЯ, ВАЛІДАЦІЯ, ВЕРИФІКАЦІЯ, ПЕРЕГЛЯД, ЕЛЕКТРОННИЙ ПІДПИС.

Об’єкт розробки - веб-застосунок системи цифрового документообігу.

Мета роботи – розробити рішення системи цифрового документообігу засобами .NET CORE.

ABSTRACT

Explanatory note to the qualification work "Software application of the digital document management system using .NET CORE": 56 p., 29 fig., 12 information sources.

WEB APPLICATION, DOCUMENT MANAGEMENT, DOCUMENT, ENCRYPTION, VALIDATION, VERIFICATION, VIEW, DIGITAL SIGNATURE.

The object of development - web application of the digital document management system.

The purpose of the work - develop the digital document management system.

Зміст

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

MVC – model-view-controller

API – application programming interface

UI - user interface

ООП – об’єктно-орієнтоване програмування

ПК – персональний комп’ютер

ПЗ – програмне забезпечення

БД – база даних

СУБД – система управління базами даних

DMS - система цифрового документообігу (document management system)

ЕЦП – електронний цифровий підпис

ВСТУП

Система цифрового документообігу не є чимось новим чи революційним. Перші такі застосунки були створені відразу після винайдення людиною обчислювальної техніки. Проте щороку з'являються безліч аналогів, які реалізують новий функціонал чи удосконалюють вже існуючий. На перший погляд це може здатися дивним, адже за такий тривалий термін можливо створити ледве не ідеальну систему. Тоді чому так багато аналогів на ринку і їх обсяг збільшується щороку?

Насправді тут немає нічого дивного, ми живемо в епоху бурхливого розвитку і системи цифрового документообігу не стали виключенням. Від звичайних документів в державних установах до цифрових чеків в магазинах. Всюди необхідні системи які будуть керувати документами та даними, обробляти їх та надсилати іншим. Такі процеси зумовлені зміною вимог до цих систем та загальним розвитком техніки.

Людство постійно прагне до розвитку та покращення всіх можливих аспектів життя. Через це виникають нові потреби та вимоги до цифрового документообігу. І єдина можливість задовольнити їх - це пристосовуватися та розробляти нові рішення, використовуючи передові технології. Так ефективніші бази даних, швидші алгоритми та сучасні методики значною мірою покращують ефективність систем.

Також дуже важливою лишається проблема інтеграції та кооперації різних систем. Адже неможливо створити ідеальну систему, яка задовільнить потреби всіх можливих користувачів. Для кожної сфери життя необхідно враховувати свої вимоги та певні умовності. Також це створює надлишок функціоналу, що буде лише сповільнювати роботу системи. Але, завдяки використовуванню загальноприйнятих стандартів та спрощенню процесів інтеграції між системами, з'явиться можливість приєднувати одні системи до інших.

Такий підхід значно розширить можливості системи. Не потрібно вивантажувати документ з однієї системи та редагувати для того щоб вірно відправити його до іншої. Натомість автоматизовані процеси самі можуть конвертувати уніфіковані дані та обробити їх відповідно до вимог різних систем. Процеси автоматизації та уніфікації даних можуть зменшити витрати на розв'язання задач, не пов'язаних з обслуговуванням.

Також це один з перших можливих кроків для створення передової цифрової держави. Адже основа держави це в першу чергу дані які треба передати та обробити. Так кожна людина витрачає значну частину свого часу на черги в державні органи і структури, заповнюючи купу бюрократичних анкет та бланків. Натомість весь цей час можна витратити на саморозвиток чи інші важливі справи, просто залишити всі ці труднощі на комп’ютерні системи.

Інтеграція різних систем безумовно є дуже важливою складовою великих систем. Проте надійність такої системи також має бути на високому рівні. Адже від якості створеної системи можуть залежати цілі сфери життя людей. Наприклад збій в документообігу банківської системи, не лише наражає їх на фінансові збитки, а також в цілому піднімає рівень недовіри населення до банківської сфери. Тоді як збій в документообігу медичної системи взагалі загрожує життю людей.

Один з найважливіших моментів системи цифрового документообігу це захист даних користувача. Захист власного підпису та конфіденційності інформації має бути одним з головних пріоритетів при розробці. Так за даними поліції лише за 2019 рік було зареєстровано понад 1100 інцидентів пов’язаних з викрадення, привласнення, вимагання документів, штампів і печаток чи заволодіння ними шляхом шахрайства. Цього можна уникнути розробивши ефективні методи шифрування та верифікації даних в системі. Але також безпечне використання та передача даних в системі залежить від обізнаності

користувача. Тому варто звернути увагу на інформування та швидку і якісну підтримку клієнтів в екстрених ситуаціях.

Також треба згадати, що використання систем цифрового документообігу позитивно вплине на екологічну проблему планети. Вирубка лісів є однією з найпоширеніших проблем пов’язаних з довкіллям. Але саме це необхідно для виробництва паперу, який ми використовуємо в документах. Завдяки зниженню потреб на паперові носії вдасться значним чином зменшити негативний вплив людини на природу та довкілля. Так замінивши паперові носії в банківській сфері можна запобігти вирубці цілих тисяч гектарів лісів.

Тому мета цього проекту – дослідження та розробка програмного засобу системи цифрового документообігу засобами .NET CORE.

Актуальність даної теми обумовлена стрімким розширенням вимог та впровадженням нових технологій в системах цифрового документообігу.

Об’єкт проектування – система цифрового документообігу.

Предмет проектування - програмний засіб системи цифрового документообігу.

Задачі проектування:

1. Проаналізувати предметну область та існуючі аналоги систем цифрового документообігу.
2. Визначити вимоги до системи цифрового документообігу.
3. Спроектувати та реалізувати систему цифрового документообігу враховуючи визначені вимоги.

Практична цінність проекту полягає у розробці програмного засобу системи цифрового документообігу засобами .NET CORE.

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ ВЕБ-ЗАСТОСУНКІВ СИСТЕМИ ЦИФРОВОГО ДОКУМЕНТООБІГУ

1.1 Аналіз цифрового документообігу

Зазвичай кажучи «документ» людина має на увазі контейнер інформації (зазвичай на папері), що містить письмову або графічну інформацію для певної мети в структурованому вигляді.

І дійсно традиційно документ – це аркуш паперу або збірка документів, наприклад: записка, лист, заява, опис або рахунок-фактура. А головне в ідеї документа полягає в тому, що його можна легко передавати, зберігати та обробляти як єдине ціле. Саме це ми і розуміємо під словом документ - певну інформацію у структурованій формі, яка затверджена реквізитами та підкріплена цінністю.

В сучасному світі інформаційні технології здатні створювати новий електронний документ, який може вміщувати графіку, текст, CAD та мультимедійні об’єкти. І все це можна вважати цифровим документом.

Документи обробляються і зберігаються в електронному вигляді не як фізичні об’єкти, а як цифрові. Документ більше не є місцем, де слова розміщуються на сторінці, а скоріше сукупністю елементів або об’єктів, пов’язаних із певною темою, об’єднаних разом в чітку структуру. Тому в електронну епоху з’являється нове визначення документа.

Електронний документ — це інформаційний контейнер в електронній формі, який збирає разом інформацію з різних джерел, у кількох форматах, навколо певної теми для задоволення потреб конкретної особи.

Користувач може створити електронний документ на персональному комп’ютері без створення паперового документа. Електронний документ можна отримати, зберегти, ідентифікувати, та перевірити використовуючи технічні засоби. Один електронний документ може бути оброблений і

переданий іншим користувачам у мережі на одному робочому місці або навіть користувачами з усього світу через Інтернет.

Однією з переваг електронного документа є те, що кожному користувачеві не обов'язково мати однакові носії. Електронний документ може бути доставлений у будь-якому форматі, який відповідає потребам користувача.

Концепція документу має кілька значень. Насправді, основний момент полягає в тому, що документ містить інформацію у такому форматі, щоб нею можна було ділитися, розповсюджувати, зберігати та виконувати певні дії. Відповідно до цього, все, що було збережено в доступному джерелі, є документом, будь то джерело, ящик або база даних, є документом. Недоступні речі не є документами, як-от незадокументовані обговорення минулого тижня. З точки зору електронних послуг, усе, що зберігається в базі даних, як файл або об'єкт даних, у найширшому розумінні являє собою документ.

Традиційний документообіг був спрямований на виконання таких функцій:

- організаційне та документаційне забезпечення діяльності підприємства;
- організація єдиного порядку роботи з документами у підрозділах;
- обмін документами всередині та між структурними підрозділами організації;
- використання уніфікованих форм для опрацювання документів;
- реєстрація, облік, видання, розсилання та повернення, забезпечення зберігання вхідних і вихідних документів;
- забезпечення зберігання, обліку та використання документів;
- перевірка правильності та вчасності виконання документів;
- одержання звітів, у тому числі статистичних, на основі інформації про документи і стан їх виконання.

Всі ці функції стандартизовані та закріплені відповідними державними та галузевими стандартами і нормативними матеріалами, що створює основу їх автоматизації з використанням інформаційних комп’ютерних технологій опрацювання документів.

Основними параметрами для систем цифрового документообігу є:

- обсяг документообігу;
- швидкість руху документів;
- вартість виконання типових операцій над документами.

Застосування електронного документообігу значно скорочує витрати паперу, прискорює обмін інформацією на підприємстві та за його межами, знижує непродуктивні витрати робочого часу, підвищує рівень корпоративної культури, та зменшує витрати на розв'язання задач, не пов'язаних з обслуговуванням.

Одним з головних подій стало офіційне затвердження юридичної сили цифрового документа та електронного підпису. Одним з таких став закон України «Про електронний цифровий підпис», який був затверджений 2003 року. Прийняття цього закону стало дуже важливим кроком для діджиталізації багатьох процесів в Україні. А також це дало потужний поштовх для початку створення вітчизняних систем цифрового документообігу.

За визначенням даним в законі який закріплює правовий статус електронного документу подається визначення: електронний документ − документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа. Він може бути створений, переданий, збережений і перетворений засобами у візуальну форму (відображення даних, які він містить, електронними засобами або на папері у формі, придатній для приймання його змісту людиною).

Також було чітко визначено склад та порядок розміщення обов'язкових реквізитів електронних документів. Так відповідно до законодавства електронний документ може бути створений, переданий, збережений і

перетворений електронними засобами у візуальну форму. Візуальною формою подання електронного документа є відображення даних, які він містить, електронними засобами або на папері у формі, придатній для сприймання його змісту людиною.

Таким чином можна визначити основні функції електронного документа у системі управління:

- забезпечення ефективного управління за рахунок автоматичного контролю виконання, прозорості діяльності всієї організації на всіх рівнях;
- ефективний доступ всіх співробітників до інформації і знань;
- підтримка комунікацій всередині підприємства за рахунок засобів розвинутої маршрутизації електронного документа;
- горизонтальний і вертикальний обмін між органами державного управління, між органами державного управління і громадянами.

Відштовхуючись від функцій можливо визначити основні вимоги. Електронні документи характеризується такими ознаками:

- електронні документи є програмно-технічно залежними продуктами;
- електронні документи мають широкий спектр інформаційного відображення (текстові, графічні, електронні таблиці, бази даних,
- мультимедійні);
- форма електронних документів може бути відокремлена від вмісту, а зміст документів може бути фрагментованим (бази даних), тобто фізично документ може зберігатись в кількох різних файлах;
- електронні документи можуть мати посилання, які не контролюються авторами, наприклад, використовування Інтернет-файлів або файлів корпоративних баз даних з коротким діапазоном життєвого циклу;

- електронні документи зберігають на фізичних носіях інформації, що не можуть гарантувати довготривале збереження інформації.

Також слід виокремити певні твердження які дозволяють стандартизувати та уніфікувати основні положення при обробці електронних документів. Так електронний документ слід вважати одержаним адресатом лише з часу надходження авторові повідомлення в електронній формі від отримувача про одержання цього документа, якщо інше не передбачено законодавством або попередньою домовленістю між суб'єктами електронного документообігу.

Через необхідність перевірки цілісності документів суб'єкти такої системи повинні зберігати електронні документи на електронних носіях інформації у формі, що дає змогу перевірити їх цілісність на цих носіях. Строк зберігання електронних документів на електронних носіях інформації повинен бути більшим ніж термін, встановлений законодавством для відповідних документів на папері. Це необхідно для урівноваження юридичної цінності паперових та цифрових документів. При неможливості такого зберігання, суб'єкти електронного документообігу мають продублювати документи на кількох електронних носіях інформації. Якщо неможливо виконати ці вимоги, електронні документи повинні зберігатися у вигляді копії документа на папері. При копіюванні електронного документа з електронного носія інформації обов'язково здійснюється перевірка цілісності даних на цьому носії.

Зберігання цифрових документів виконується з виконанням таких вимог:

1) інформація, що міститься в електронних документах, повинна бути доступною для її подальшого використання;

2) має бути забезпечена можливість відновлення електронного документа у тому форматі, в якому він був створений, відправлений або одержаний;

3) у разі наявності повинна зберігатися інформація, яка дає змогу встановити походження та призначення електронного документа, а також дату і час його відправлення чи одержання.

Обов'язковим реквізитом електронного документа є електронний підпис, який використовується для ідентифікації автора електронного документа іншими суб'єктами електронного документообігу. Накладанням електронного підпису завершується створення електронного документа.

Оригіналом документа вважається примірник з обов'язковими реквізитами, у тому числі з електронним цифровим підписом автора. У разі надсилання електронного документа багатьом отримувачам або його зберігання на кількох електронних носіях, кожний з електронних примірників вважається оригіналом електронного документа. Лише так можливо забезпечити юридичну цінність цифрових документів. При цьому якщо автор створив ідентичні примірники електронного та паперового документів, кожен з них буде вважатися оригіналом і матиме однакову юридичну силу.

Юридична сила електронного документа не може бути заперечена виключно через те, що він має електронну форму. Допустимість електронного документа як доказу не може заперечуватися виключно на підставі того, що він має електронну форму. Юридична чинність і доказовість цифровому документу надається завдяки електронному підпису.

Суть електронного цифрового підпису полягає у тому, що він заснований на алгоритмах криптографічного захисту інформації. Такий підпис накладається за допомогою особистого ключа − спеціального коду, відомого тільки власнику. Якщо цей код повідомити програмі, то відповідно до криптографічного алгоритму вона сформує унікальне контрольне значення і додасть його до документа, тобто підпише електронний документ унікальним цифровим підписом. В дійсності цього підпису можна впевнитися за допомогою відкритого ключа − коду перевірки, доступного решті суб'єктів електронного документообігу. Цей код унеможливлює підробку, і при цьому

надає можливість перевірити його справжність. Цей код система звірить з отриманим в документі підписом автора і якщо ці контрольні значення зійдуться, підпис вважатиметься справжнім, а отриманий документ - цілісним. Окрім того відкритий ключ повинен бути затвердженим центром сертифікації ключів.

Електронний підпис − дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов'язані та призначені для ідентифікації підписанта.

Електронний цифровий підпис − вид електронного підпису, отриманого за результатом криптографічного перетворення набору даних.

Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Засіб електронного цифрового підпису − програмний засіб або апаратний пристрій, які призначені для генерації ключів, накладення і перевірки електронного цифрового підпису.

Особистий ключ − параметр криптографічного алгоритму формування електронного цифрового підпису, доступний тільки підписанту.

Відкритий ключ − параметр криптографічного алгоритму перевірки електронного цифрового підпису, доступний суб'єктам відносин у сфері використання електронного цифрового підпису. Використовується при перевірці цілісності документу.

На сьогоднішній день документи практично не виготовляються вручну, але багато документів все ще передаються шляхом їх друку та надсилання іншим сторонам поштою чи кур’єрами, часто використовуючи копіювальні компанії як посередників. Трохи складніший метод полягає в тому, що документи створюються в цифровому вигляді та передаються як вкладення електронної пошти. Це пришвидшує передачу документів, але з точки зору керування документами це навряд чи покращує ситуацію, оскільки знайти

документ на персональному комп’ютері іншої людини може бути навіть важче, ніж на полицях.

Одним з найскладніших методів доставки, який зараз використовується, є система управління документами. В ній документи зберігаються централізовано на сервері, а користувачі взаємодіють із цим центральним сховищем через інтерфейси, реалізовані за допомогою стандартних веб-браузерів.

DMS розробляються, щоб забезпечити бібліотеку та сховище, де можна створювати документи, керувати ними та зберігати для полегшення доступу відділів і користувачів на підприємстві.

Рано чи пізно кожна компанія відчуває потребу в якійсь системі електронного документообігу, щоб контролювати кількість різноманітних документів і креслень, що постійно зростає.

Компанії часто опираються «цьому прагненню» і їх відлякують витрати та складність впровадження DMS. Ефективне використання DMS вимагає суттєвих змін у робочій практиці, хоча більшість технічних аспектів вирішуються за допомогою використання недорогих баз даних і простішої інтеграції з середовищем Windows. Корисна DMS повинна не тільки контролювати документи, але й надавати доступ до них у всій компанії і навіть клієнтам або іншим учасникам проекту через Інтернет. DMS також має централізувати дані в легкодоступному середовищі, дозволяючи користувачам зберігати, отримувати доступ і змінювати інформацію легко та швидко.

Стандартні функції хорошої системи все ще повинні включати: можливість пошуку

- перегляд без використання оригінальної програми;
- функцію червоної лінії та розмітки;
- друк і малювання;
- робочі процеси та життєві цикли документів;

- перегляд і контроль версій;
- безпека документів;
- взаємозв'язки документів;
- звітування про стан;
- керування випуском і розповсюдженням;
- віддалений доступ.

Мета управління документами полягає в тому, щоб обмінюватися інформацією, роблячи документи безпечними, доступними, доступними для пошуку та взаємозамінними. Тому чудовим рішенням цієї проблеми є системи електронного документообігу.

1.2 Дослідження особливостей цифрового документообігу в Україні

Аналізуючи законодавство, що регламентує державну політику в сфері впровадження ефективних систем документообігу в Україні можна зробити висновок, що основними стратегічними завданнями такої політики є:

- організація та забезпечення функціонування механізмів систематичного наповнення, надійного захисту та збереження інформації в системах документообігу;
- забезпечення реалізації права кожного знайомитись з документами та інформаційними матеріалами, а також збирати та зберігати інформацію у всіх органах державної влади, якщо таке право передбачено законом;
- спрощення роботи працівників апарату держави;
- забезпечення дотримання системності та конструктивності у законотворчій, правозастосовній діяльності;
- розвиток системи “відкритих дверей”;
- розвиток інформаційних технологій, як інструменту забезпечення формування, захисту та використання національних інформаційних ресурсів;
- розвиток електронних інформаційних ресурсів.

Такі завдання на сьогоднішній день можна вирішити лише шляхом впровадження електронних систем з подальшою їх інтеграцією в Єдину глобальну національну систему інформаційних ресурсів. Таке впровадження можливе лише після вирішення цілого спектру проблем.

Електронний документообіг – це високотехнологічний і прогресивний підхід до суттєвого підвищення ефективності роботи органів державної влади і місцевого самоврядування, який забезпечується сукупністю технологій, які не тільки значно оптимізують, але й істотно змінюють роботу з документами.

Для створення систем документообігу необхідний кадровий склад, який володіє відповідними інформаційно-технічними, правовими та іншими знаннями, вміннями та навичками. Державний апарат не володіє персоналом, який воліє переходити на електронний документообіг. Така поведінка працівників обумовлена не тільки небажанням навчатися й перенавчатися, а також, можливо, низькою освіченістю. А отже для такого впровадження необхідно буде змінювати кадрову політику в сфері державної служби.

Невирішеним залишається питання трансформації існуючих чинних документів з паперової форми в електронну з метою створення електронного архіву. На наш погляд, саме такий архів забезпечить надійний захист інформаційного ресурсу, та зручність в користуванні архівами. Таку проблему, на думку деяких науковців можна вирішити шляхом сканування документів. Сканування документів, хоч і не є швидким засобом вирішення цієї проблеми, оскільки в межах країни це купа роботи, але наразі він є єдино можливим. Створення електронного архіву необхідно починати з єдиного порядку зберігання документів у каталогах, і зберігати їх не на локальних дисках комп’ютерів, а розміщувати у внутрішній мережі. Це суттєво облегшить подальшу міграцію документів у систему документообігу.

1.3 Дослідження переваг та недоліків систем цифрового документообігу

Багато компаній використовують DMS, щоб стандартизувати інформацію для будь-кого з відповідними повноваженнями для пошуку та доступу до потрібного документа. DMS допомагає користувачам легше виконувати роботу та забезпечує компанії безпеку, надійність даних і управління робочим процесом. Багато з цих функцій зрештою економлять час, спрощують роботу, захищають інвестиції, зроблені у створення цих документів, забезпечують дотримання стандартів якості, забезпечують аудиторський слід і забезпечують підзвітність.

DMS має такі переваги:

- Загалом ефективне розташування та доставка документації
- Можливість керувати документами та даними незалежно від вихідної системи чи формату
- Здатність інтегрувати комп’ютеризовані та паперові системи
- Контроль доступу, поширення та модифікації документів
- Надання інструментів редагування та розмітки документів

З іншого боку, є і недоліки. Проблеми з впровадженням нового і зміна робочої культури та практики, які потрібні спочатку, дуже часто відлякують користувачів.

Досягнення тих цілей, які необхідні в сьогоднішньому середовищі, вимагає серйозних змін в організації, включаючи практики, системи, методики та робочі процеси. Необхідно розробити правильні стратегії та плани впровадження, повідомити про них і втілити їх у життя. І оскільки це непросто, такі питання, визначення стратегії, вибір системи, розробка програми навчання, визначення операційних процедур, зміна організаційних структур, перегляд використання, розширення використання тощо, потребують ретельного розгляду і дослідженню.

Оцінити, вибрати та використовувати DMS нелегко, тому не слід ігнорувати можливості зберігання даних, розширені мережеві можливості, потужні настільні комп'ютери, підтримку програмного забезпечення та впровадження. Також слід розглянути бізнес та організаційні питання, такі як початкові витрати, максимізація та аналіз окупності, обґрунтування витрат та економія.

Вся інформація має бути в електронному форматі, який створюється в електронному вигляді або сканується з паперової версії. Це включає рукописні нотатки та ескізи, а також великі карти та складні малюнки. Багато зусиль витрачається на взаємодію з несумісними системами, особливо паперовими.

1.4 Аналіз існуючих рішень

На ринку систем цифрового документообігу існує безліч готових продуктів. Для аналізу обрано кілька відомих систем:

- Google Drive;
- OnlyOffice;
- DocuWare;
- LogicalDoc;
- M.E.Doc.

Почати аналіз варто з відомого рішення від компанії Google. Google Drive – це хмарна DMS, доступна кожному, хто має обліковий запис Google. Вона може легко зберігати, синхронізувати та обмінюватися файлами.

Цей інструмент найкраще підходить для команд, які спільно працюють над проектами в реальному часі. Функції спільного використання та редагування в режимі реального часу Google Drive виводять професійну співпрацю на новий рівень.

Також однією з переваг є доступність цього інструменту як онлайн через браузер так і завантаживши клієнт на комп'ютер.

Ключові риси:

- Забезпечує велике сховище для багатьох типів файлів.

- Дозволяє інтегрувати з іншими продуктами Google і сторонніми програмами та файлами.
- Пропонує функцію пошуку для пошуку документів за типом файлу або назвою.
- Дозволяє конвертувати файли, історію редагування файлів, резервне копіювання документів у хмару для легкої синхронізації між пристроями та легкого відновлення.
- Пропонує легкий обмін файлами із зовнішніми учасниками за допомогою спільного доступу та електронної пошти.
- Пропонує автономні можливості для вибраних файлів і облікових записів.

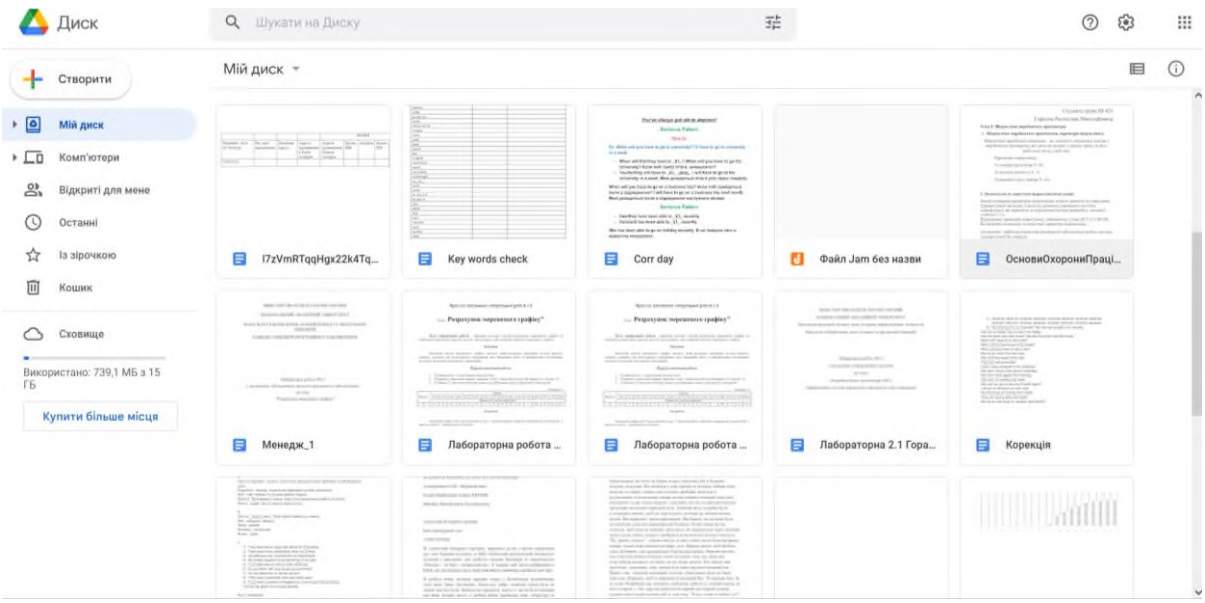

Рис. 1.4.1. Інтерфейс системи Google Drive

Переваги: завдяки своєму інтуїтивно зрозумілому інтерфейсу Google Диск перевершує його, коли йдеться про зручність використання. Його ефективна вбудована пошукова система є родзинкою, що дозволяє користувачам шукати файли за типом і власником.

Найвидатнішою особливістю Google Drive є простота використання. Дуже просто розібратися в усьому завдяки величезній кількості підтримки клієнтів. Однак одним із напрямків удосконалення є категоризація та сортування документів і папок.

Наступним було досліджено OnlyOffice. Це офісний пакет програмного забезпечення та DMS, розроблений Ascensio Systems SIA.

OnlyOffice надає платформу для керування документами, яка найкраще підходить для багатьох освітніх установ, малих і середніх компаній, підприємств і державних організацій.

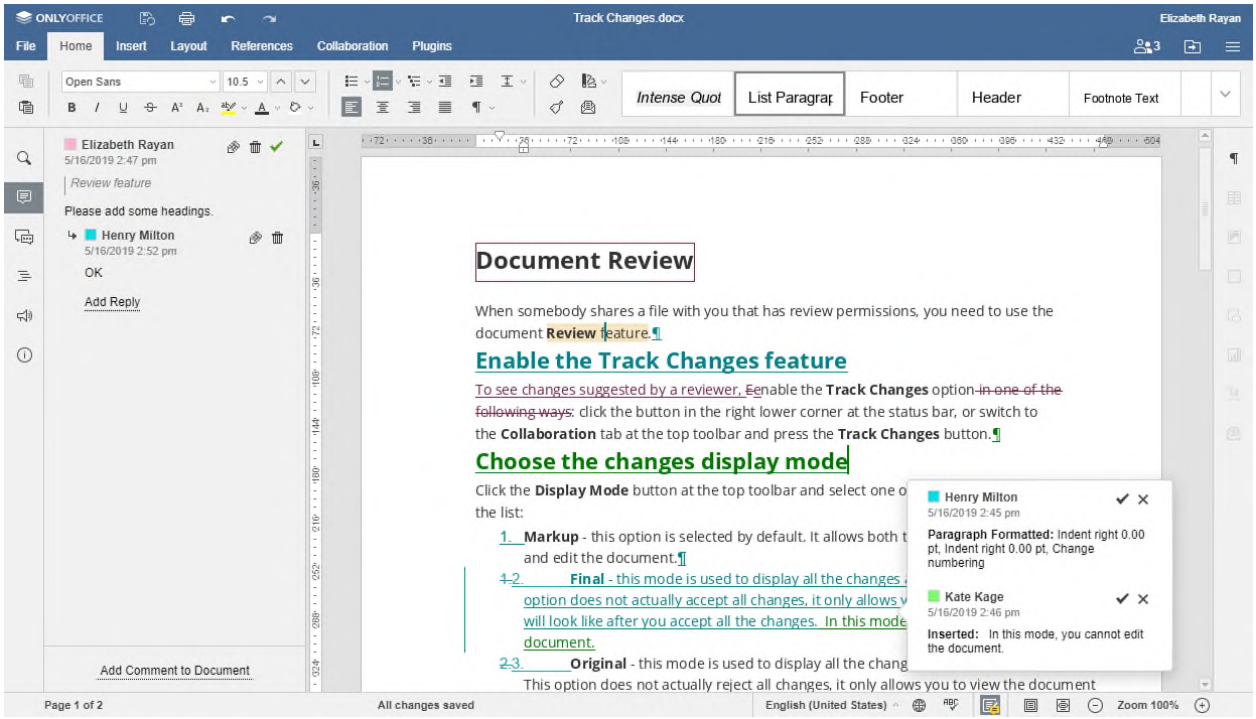

Рис. 1.4.2. Інтерфейс системи OnlyOffice

Ключові риси:

- Пропонує інструменти для співпраці, електронний підпис, архівування документів, контроль версій, відновлення файлів, перетворення типів файлів, оптичне розпізнавання символів, збірку документів.
- Надає інструменти для керування проектами, командні чати, обмін файлами та інструменти для презентацій.
- Працює у настільній версії для програм Windows, iOS та Android. До нього можна отримати доступ через веб-браузер.

Плюси: OnlyOffice має ідеальну інтеграцію з робочим столом. Він працює на смартфонах Linux, Mac OS і Windows і в хмарі, як жодне інше програмне забезпечення на ринку.

Ціна: наразі пропонується 180-денна безкоштовна пробна версія їхнього хмарного сервісу. Хмарне видання може коштувати лише 3 долари США на місяць за користувача, якщо його придбати за 3-річною підпискою. Також запропоновані версії Enterprise, Integration і Developer.

Проте в OnlyOffice відсутні відстеження відповідності, офлайн-доступ та індексація документів, що викликає певні труднощі в використанні.

Далі було досліджено DocuWare. Вона була заснована в 1988 році, і на даний момент є одним із провідних програм для керування документами та вмістом. Вона доступнп 16 мовами та використовується такими великими компаніями, як IKEA та Sony.

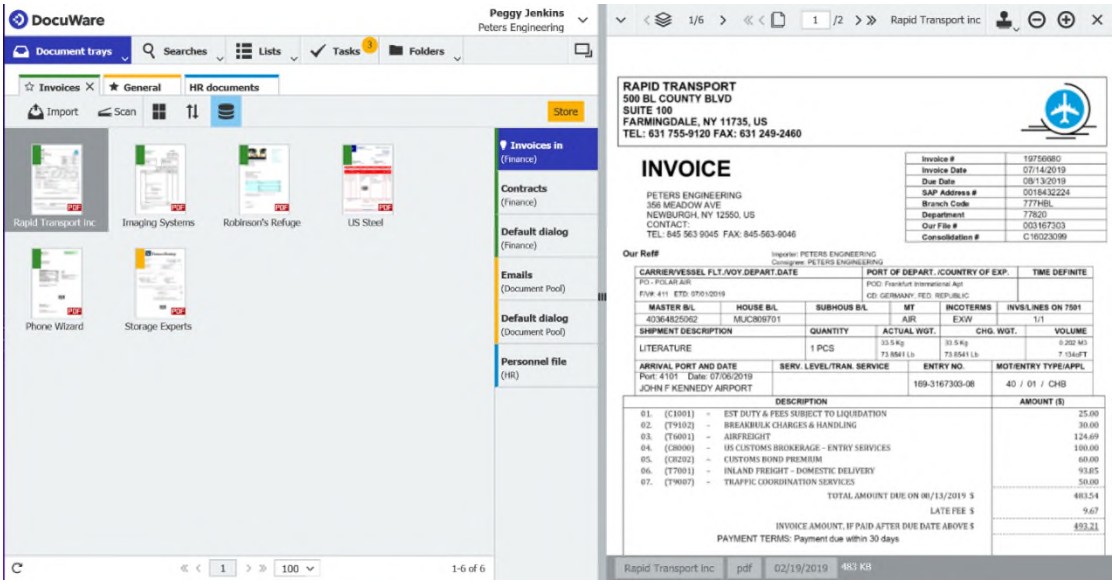

Рис. 1.4.3. Інтерфейс системи DocuWare

DocuWare забезпечує керування цифровими документами та автоматизовані рішення для робочих процесів. Ця система гарно підходить для організацій будь-якого розміру та в усіх основних галузях промисловості, від виробництва та роздрібної торгівлі до охорони здоров'я та державного управління.

Ключові риси:

- Працює на робочих столах Windows, веб-браузерах або мобільних програмах для iOS/Android.

- Пропонує особисті та дистанційні тренінги з підтримкою клієнтів у звичайний робочий час.
- Надає інструменти для співпраці, відстеження відповідності та електронний підпис.
- Пропонує функції для архівування документів, збирання, індексування та збереження.
- Пропонує оптичне розпізнавання символів (OCR), відновлення файлів, контроль версій і офлайн-доступ.

Забезпечує інтеграцію з більш ніж 500 різними програмами.

Плюси: інтерфейс користувача простий, інтуїтивно зрозумілий і зручний. Інструмент має ефективну функцію перетягування, і користувачі можуть виконувати кілька завдань у кількох документах, не залишаючи головної консолі.

Ціна: DocuWare пропонує безкоштовну пробну версію. Його базовий план надає чотирьом користувачам 20 ГБ пам'яті за 300 доларів на місяць. DocuWare має вбудований інструмент адміністрування, що дозволяє створювати кілька профілів доступу з правилами для перегляду та запису. Це дозволяє створити кілька папок для файлів і слугує для контролю доступу. Однак підтримка залишає бажати кращого. Службі підтримки DocuWare іноді потрібно багато часу, щоб відповісти на прості запитання.

Наступним було проаналізовано LogicalDoc. LogicalDOC має на меті допомогти організаціям отримати контроль над керуванням документами, зосередившись на швидкому пошуку документів і автоматизації бізнес-процесів.

Завдяки своїй гнучкості LogicalDOC може адаптуватися до різних потреб і найкраще підходить для малих, середніх і великих організацій, включаючи банки, заклади охорони здоров'я, машинобудівні підприємства та муніципалітети середнього розміру.

Ключові риси:

- Працює в веб-браузерах, iOS і Android App з багатомовним інтерфейсом.
- Забезпечує багатомовне повнотекстове індексування, контроль версій, захист документів паролем.
- Дозволяє пошук документів, функцію реєстрації/виписки, анотації, внутрішній обмін повідомленнями та електронну пошту.
- Дозволяє модифікувати певні функції.

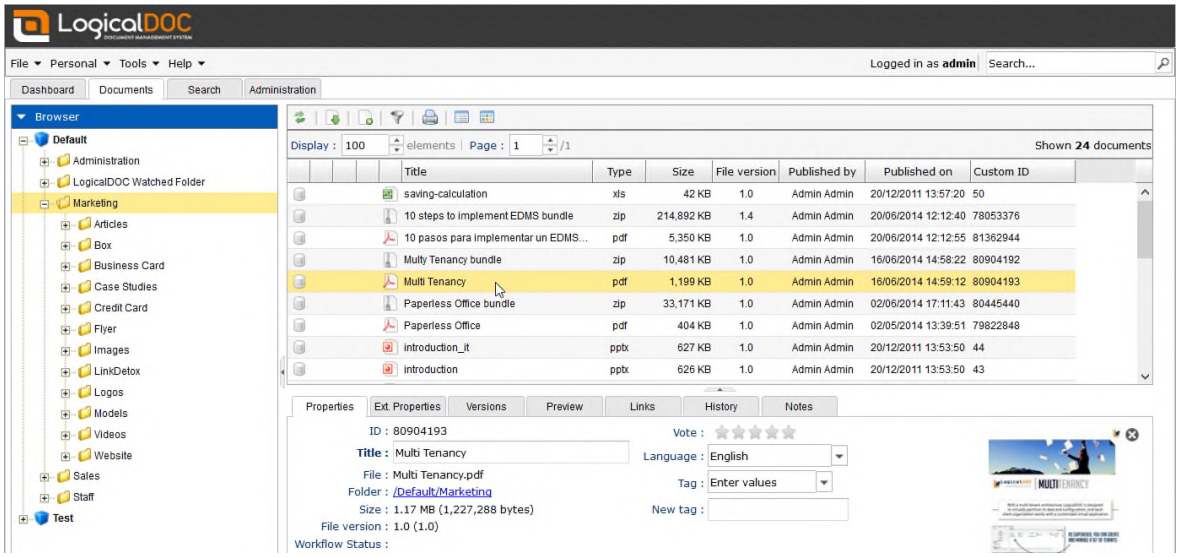

Рис. 1.4.4. Інтерфейс системи LogicalDoc

Плюси: LogicalDOC — це дуже інтуїтивно зрозумілий інструмент для планування, виконання та керування бізнес-проектами.

Ціна: LogicalDOC Community — це безкоштовне програмне забезпечення з відкритим кодом. За консультацією доступні більші пакети.

Безкоштовне програмне забезпечення з відкритим вихідним кодом, хоча й є економічно вигіднішим варіантом, не має деяких функцій безпеки та відновлення документів.

Останньою була проаналізована систему M.E.Doc. Це поширене українське програмне забезпечення для подання звітності до контролюючих органів та обміну юридично значущими первинними документами між контрагентами в електронному вигляді.

Розробка системи почалась з розвитком нових технологій і вимог ринку. Програмне забезпечення безперервно актуалізувалося і вдосконалювалося,

результатом чого в 2010 році стала поява на українському ринку розробки M.E.Doc. У 2019 році програма M.E.Doc отримала нагороду «Вибір року 2019» у номінації «Комерційна програма року для подачі звітності та електронного документообігу».

Програма M.E.Doc призначена для подачі звітності в усі контролюючі органи України (ДФСУ, ДССУ, ПФУ, ФСС з ТВП, ДКСУ, міністерства і відомства), для реєстрації податкових накладних та юридично значущого електронного документообігу. Крім того, програма надає модулі для нарахування зарплати, звітності великих компаній з розгалуженою структурою підрозділів і роботи банків. Також система підтримує роботу з ЕЦП найбільш використовуваних центрів сертифікації, а також із захищеними носіями ключів SecureToken.

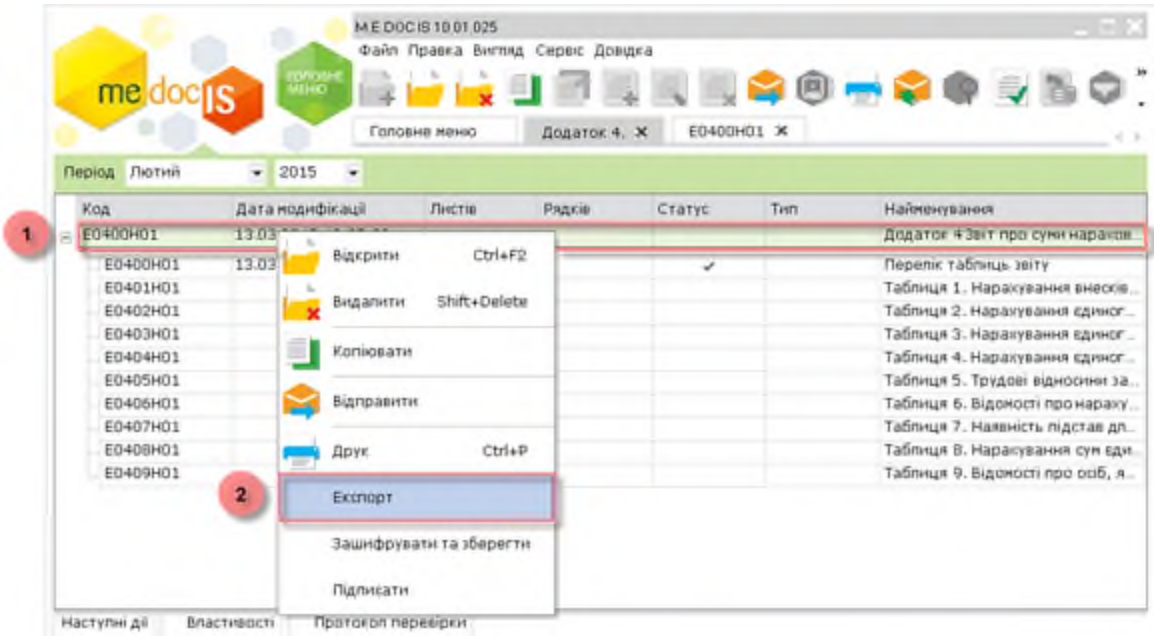

Рис. 1.4.5. Інтерфейс системи M.E.Doc

Програма має як готові рішення для основних потреб користувачів, так і додаткові модулі:

- Рішення M.E.Doc.Держава — подача всіх видів звітності в усі контролюючі органи, реєстрація податкових накладних в Єдиному реєстрі податкових накладних та обмін ними з контрагентами

- Рішення M.E.Doc.Бізнес — обмін первинними бухгалтерськими документами з контрагентами
- Модуль M.E.Doc.Акциз та ТТН — для роботи з системою електронного адміністрування реалізації палива та поводження з товарно-транспортними накладними
- Модуль M.E.Doc.Зарплата — розрахунок і нарахування заробітної плати співробітникам, облік і управління персоналом
- Модуль M.E.Doc.Корпорація — консолідація звітності підприємств з розгалуженою структурою

З переваг варто відмітити наявність цілодобової підтримки клієнтів. Та інтеграцію з багатьма іншими українськими системами.

Проте ця система працює лише з завантаженим клієнтом. Хоч і компанія повідомляє що веде розробку хмарного рішення, на даний момент ця можливість не доступна.

Висновок

Отже, під час дослідження процесів системи цифрового документообігу було досліджено основні вимоги та функції таких систем. Також виявлено основні проблеми, переваги і недоліки. Було виявлено основні тенденції в області систем цифрового документообігу та їх перспективи.

Також було окремо досліджено законодавчі вимоги до цифрових документів, підписів та систем управління. Виявлено ряд вимог яким необхідно чітко дотримуватися для юридичної сили системи цифрового документообігу.

Знайдено проблеми з впровадженням таких систем в Україні. І запропоновано методи їх вирішення.

Для розроблення власного застосунку який зможе забезпечити всі потреби користувачів та буде дотримано законодавчі вимоги було проаналізовано найпоширеніші аналоги: Google Drive, OnlyOffice, DocuWare, LogicalDoc та M.E.Doc. Було досліджено їхні переваги та недоліки. А також

визначено функції які необхідно реалізувати для забезпечення якісного сервісу системи. Для аналізу було вибрано дуже відомі та поширені системи з широким застосуванням і функціоналом. Цей аналіз дав змогу виявити слабкі місця сучасних аналогів, та дозволив звернути увагу на проблеми які варто вирішити у власному програмному застосунку.

РОЗДІЛ 2
ВИМОГИ ДО ПРОГРАМНОГО ЗАСОБУ СИСТЕМИ ЦИФРОВОГО ДОКУМЕНТООБІГУ

2.1 Аналіз вимог

Проаналізувавши існуючі аналоги в попередньому розділі було виявлено переваги та недоліки. І на основі цього аналізу тепер можливо створити вимоги до власної системи. Яка зможе реалізувати вдалі рішення та уникнути помилок аналогів.

Вірне визначення вимог для застосунку значно полегшить процес розробки та підтримки продукту. Створення вимог зазвичай можна розподілити на декілька етапів, а саме:

- Знаходження вимог (збір, визначення потреб зацікавлених осіб та систем).
- Аналіз вимог (перевірка цілісності та закінченості).
- Специфікація (документування вимог).
- Тестування вимог.

В процесі аналізу та визначення вимог існує багато труднощів як для розробників так і для замовників продукту. Це може бути погане формування вимог від самого замовника чи постійне додання нових побажань вже в процесі розробки. Через такі проблеми потрібно підбирати правильні методології управління проектом та ретельно ставитися до усіх процесів розробки. Переважна більшість створюваних програм закінчуються провалом саме через недоліки в процесі створення вимог. Нерідко через погано складені вимоги до продукту замовник отримуєте зовсім не те що він хотів.

Вимоги – це вихідні дані, на підставі яких проектуються й створюються автоматизовані інформаційні системи. Це можуть бути зовсім різні вимоги і стосуватися різних аспектів розробки, роботи чи впровадження системи. Існує безліч класифікацій вимог до програмного продукту. Для своєї системи я розподілив вимоги на функціональні та нефункціональні.

2.2 Функціональні вимоги

Функціональні вимоги відповідають за перелік функцій або сервісів, які повинна надавати система, а також обмежень на дані і поводження системи при їхньому виконанні.

Перелік функціональних вимог до програмний засобу системи цифрового документообігу:

- Зберігання документів;
- Надсилання документів;
- Перегляд та редагування документів;
- Створення персонального кабінету;
- Створення та управління організацією;
- Створення та управління груп користувачів;
- Контроль доступів в групах;
- Сповіщення користувачів;
- Створення документів за загальними шаблонами;
- Завантаження документів в різних форматах;
- Підписання документів за допомогою ЕЦП;
- Перевірка ЕЦП;
- Створення вимог для підпису документів;
- Відміна надсилання документу;
- Формування звітності по використаним документам;
- Поповнення балансу;
- Перевірка балансу;
- Купівля пакету;
- Визначення посад для клієнтів від організацій;
- Інтеграція з іншими системами;
- Зворотній зв'язок з підтримкою.

Для кращого розуміння було створено діаграму прецедентів (рис. 2.1). На рисунку 2.1. відображені всі основні функціональні вимоги до програмного

застосунку системи цифрового документообігу. А також залежності між ними та спільний функціонал.

Рис. 2.1. Діаграма прецедентів

На діаграмі зображено три актори:

- Користувач – це звичайний клієнт який вирішив купити пакет для користування як приватна особа.
- Адміністратор організації – це клієнт який виступає в ролі управляючого організацією, він створює організацію при першій реєстрації та налаштовує її. Його функціонал також включає

налаштування груп користувачів від організації, а також отримання звітності по цій організації. Також з його акаунту можлива оплата послуг системи.

- Співробітник організації – це клієнт з обмеженим функціоналом, він виступає в ролі робітника організації. І його акаунт додається в систему з належністю до організації. Він має доступ до груп в які він доданий та може виконувати функції, до який йому відкрили доступ.

Тобто система фактично матиме кілька типів можливих користувачів. Звичайні користувачі не мають доступу до функціоналу організацій та груп. Співробітники організацій, можуть виконувати лише ті дії, які їм дозволені. Тоді як адміністратори мають найширший функціонал, проте перед отриманням адміністраторського акаунту, організація повинна звернутись в підтримку та оплатити пакет.

2.3 Нефункціональні вимоги

Нефункціональні вимоги - це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути.

Це значно ширша категорія, куди можна внести всі побажання відповідно до певних груп вимог.

Вимоги до інтерфейсу користувача:

- кольорова гама користувацького інтерфейсу повинна мати 2-4 основних кольори, що добре контрастують між собою, та до чотирьох відтінків кожного із цих кольорів.
- усі функціонально схожі елементі управління (кнопки, списки, тощо) повинні бути виконані в одному дизайні та мати схожі ефекти анімацій (при наведенні миші, вибору елемента списку, тощо). Допускаються незначні відмінності для виділення більш

конкретного функціоналу (кнопки для навігації та кнопки для підтвердження дії, тощо);

- елементи управління не повинні робити інтерфейс занадто громіздким. Тому повинні бути розташовані не більше трьох різнотипних елементів в одній візуальній групі. Елементі розташовані поряд повинні мати схожі розміри;
- перша, за замовчуванням, сторінка веб-застосунку має бути сторінкою персональної папки;
- меню з вкладками навігації має знаходитися в лівій частині та завжди бути доступним;
- горизонтальне меню з вкладками персонального кабінету, налаштувань та підтримкою повинне бути закріпленим у верхній частині фрейму;
- уся користувацька частина застосунку повинна бути адаптована та оптимізована для коректного відображення як на мобільних пристроях так і персональних комп'ютерах.

Вимоги до загальної структури проекту:

- реалізація вкладок навігації, повинне утворювати кілька ієрархічних дерев;
- веб-застосунок повинен відповідати сучасним вимогам пошукових систем, для коректної індексації та відображення у результатах пошуку;
- оскільки веб-застосунок має зберігати дуже великий обсяг інформації, і інформація, що надається користувачам, пов'язана із фінансами та особистою інформацією, то весь обсяг даних веб-застосунку має зберігатися у надійному сховищі, що забезпечує цілісність даних.
- лише адміністратор повинен мати доступ до управління організацією.

- інтеграція з іншими системами повинна надаватися в вигляді, вводу токену або ключа, через який можливо ідентифікувати інтеграцію.
- для контролю документів, необхідно додати систему контролю версіями документів, а також систему логування змін.
- пошук отримувача може здійснюватися або за допомогою email, або іменем, або номером телефону.
- через законодавчі вимоги, кожен користувач повинен мати доступ до перевірки підпису всіх контрагентів в документі.
- електронний документ повинен бути представлено у вигляді набору загальної інформації, самих документів та файлів підписів.

Апаратні вимоги:

Мінімальні:

- Операційна система: Windows XP.
- Процесор: Pentium 2 ядра, від 1.5 ГГЦ.
- Оперативна пам’ять: 1 ГБ
- Доступ до мережі Інтернет
- Наявність браузера (Microsoft Edge, Mozilla Firefox, Google Chrome)
- Відеоадаптер та монітор: вбудований у процесор та монітор з роздільною здатністю 1280 на 720.
- Засоби вводу: Клавіатура та миш.

Рекомендовані:

- Операційна система: Windows 10.
- Процесор: Intel i3 2 ядра, від 2 ГГЦ.
- Оперативна пам’ять: 2 ГБ
- Доступ до мережі Інтернет
- Наявність браузера (Microsoft Edge, Mozilla Firefox, Google Chrome)

- Відеоадаптер та монітор: вбудований у процесор або зовнішня відеокарта та монітор з роздільною здатністю 1920 на 1080.
- Засоби вводу: Клавіатура та миш.

Формати для завантаження документів:

- PDF – цей формат відмінно підходить для сучасних текстових електронних документів. Цей формат розроблений для друку тому повністю описує як повинен виглядати документ – включаючи розмір паперу, види шрифтів тощо. Крім тексту PDF може містити векторну і растрову графіку а також метадані.
- XLSX – міжнародний стандарт формату файлів для електронних документів, як-от електронні таблиці, діаграми, презентації та текстові документи, що базується на XML, підготований для продукту Microsoft Excel.
- DOCX – міжнародний стандарт формату файлів для електронних документів, як-от електронні таблиці, діаграми, презентації та текстові документи, що базується на XML Microsoft Word.

Висновок

У даному розділі були визначені вимоги до проектування, розробки та кінцевого вигляду програмний засіб системи цифрового документообігу, який є метою цього проекту. Також було описано вимоги для форматів документів які повинна використовувати система. Ці вимоги достатньо описують архітектуру, функціонал та користувацький інтерфейс майбутнього веб-застосунку. Це дозволить спростити розробку веб-застосунку та допоможе точно обрати найбільш відповідні програмні інструменти, архітектурні підходи та алгоритми для задоволення потреб користувачів.

РОЗДІЛ 3
СТРУКТУРА ПРОГРАМНОГО ЗАСОБУ СИСТЕМИ ЦИФРОВОГО ДОКУМЕНТООБІГУ

3.1 Архітектура програмного засобу системи цифрового документообігу

Програмний засіб проектується за допомогою технології ASP.NET Core MVC. ASP.NET Core - це новий, надійний і багатофункціональний фреймворк, що надає функції для розробки швидких API для веб-додатків.

На відміну від ASP.NET MVC, новий фреймворк .NET Core надає вбудовані шаблони для двох найбільш популярних фреймворків JavaScript - Angular і React. JavaScriptServices в новому ASP.NET Core забезпечує інфраструктуру, яка необхідна розробникам для програмування клієнтських додатків з використанням вищезазначених фреймворків JavaScript.

JavaScriptServices в основному націлена на усунення що лежить в основі сполучної ланки і дозволяє розробникам швидше писати додатки, роблячи можливим створення багатофункціональних інтерфейсних веб-додатки.

ASP.NET Core - це крос-платформенний фреймворк, що означає, що додатки, побудовані з використанням цього фреймворка, можуть працювати в операційних системах Windows, Linux і Mac.

Ця технологія являється зручною багатофункціональною платформою для створення веб-застосунків за допомогою шаблону Model-View-Controller.

Цей шаблон розділяє роботу засобу на три окремі функціональні ролі (рис. 3.1):

- Модель (Model) - цей компонент відповідає за дані, а також визначає структуру програми;
- Представлення (View) - це компонент який відповідає за взаємодію з користувачем. Тобто код компонента view визначає зовнішній вигляд програми і способи його використання;

- Контролер (Controller) - цей компонент відповідає за зв'язок між моделями та представлення. Код компонента controller визначає, як сайт реагує на дії користувача. По суті, це мозок MVC-додатків.

Таким чином, зміни, що вносяться в один з компонентів, надають мінімально можливий вплив на інші компоненти.

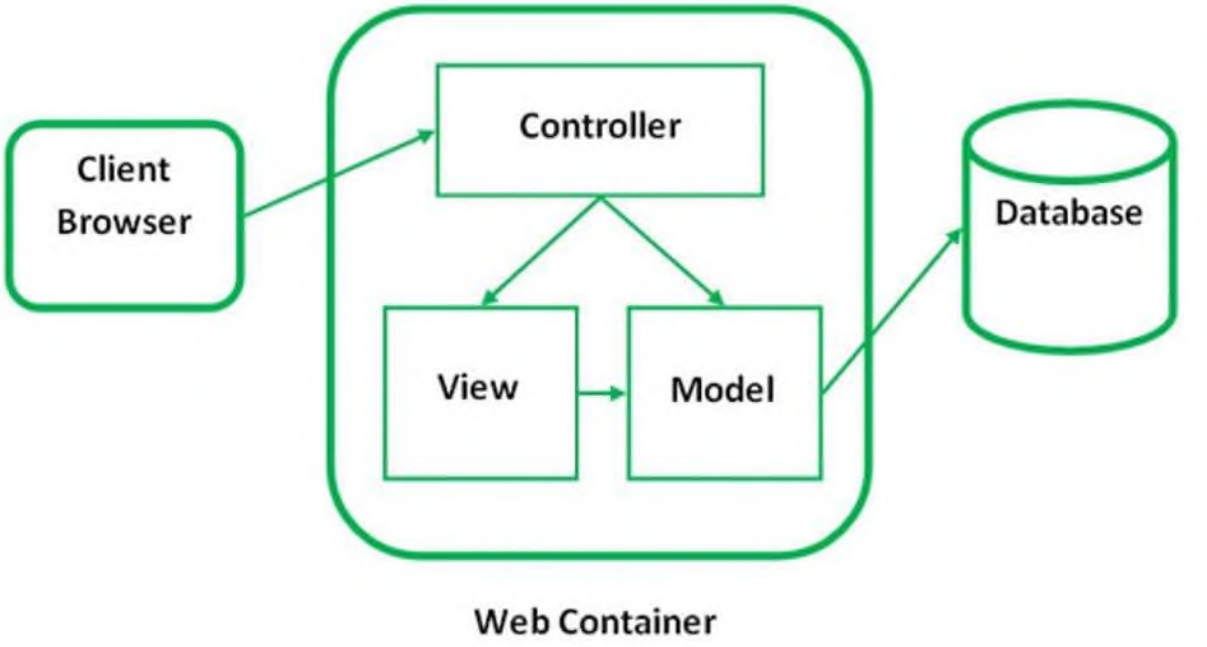

Рис. 3.1. Шаблон MVC

В даному шаблоні модель не залежить від подання або керуючої логіки, що робить можливим проектування моделі як незалежного компонента і, наприклад, створювати кілька подань для однієї моделі.

Такий поділ web-додатку на частини спрощує структуру програми за рахунок більш строго поділу його рівнів. Досягається повне відділення логіки роботи програми від представлення даних. Розробник отримує повний контроль над сформованим HTML-документом а також полегшується завдання виконання тестування програми.

Також враховуючи мультиплатформенність ASP.NET Core можна використовувати для створення програмного засобу у вигляді веб-застосунку, і Windows клієнта.

Для розробки буде використана система IT – Enterprise. Це система для розробки застосунків, яка використовує .NET, Vue JS і Python. В цій системі є багато бібліотек і функцій для прискорення процесу розробки та написання коду.

Для пришвидшення створення Windows клієнта застосовується візуальний редактор. Він значно спрощує процеси прототипування та реалізації інтерфейсів.

Vue JS використовується для веб-інтерфейсу. Це фреймворк для розробки веб-сторінок на основі JavaScript і TypeScript. Його особливість в тому що він легко інтегрується з іншими JS бібліотеками. Також для покращення візуальної складової використовується Bootstrap 4, який широко використовується в розробці адаптивних веб-сайтів.

3.2 Взаємодія з даними

Для роботи з даними буде використано Microsoft SQL Server. Це система управління базами даних, яка розробляється корпорацією Microsoft. До її переваг належать:

- СУБД масштабується, тому працювати з нею можна на портативних ПК або потужній мультипроцесорної техніці. Процесор може одночасно обробляти великий обсяг запитів.
- Розмір сторінок - до 8 кб, тому дані витягуються швидко, детальну і складну інформацію зберігати зручніше. Система дозволяє обробляти транзакції в інтерактивному режимі, є динамічне блокування.
- Рутинні адміністративні завдання автоматизовані: це управління блокуванням, пам'яттю, редагування розмірів файлів. У системи продумані настройки, можна створити профілі користувачів.
- Реалізовано пошук по фразах, тексту, словами, можна створювати ключові індекси.

- У SQL Server є реплікації через інтернет, передбачена синхронізація. Є повноцінний веб-асистент для форматування сторінок.
- В систему інтегрований сервер інтерактивного аналізу для прийняття рішень, створення корпоративних звітів. Є служби перетворення інформації.
- Запити можна формулювати англійською мовою, без програмування.
- СУБД підтримує роботу з іншими продуктами Microsoft: Access, MS Excel.

Тому ця система підходить для реалізації програмного засобу системи цифрового документообігу.

3.3 Діаграма розгортання

Діаграма розгортання призначена для візуалізації елементів і компонентів ПС, які існують лишена етапі її виконання. Ця діаграма допомагає раціональніше організувати компоненти, від чого залежить в числі іншого і продуктивність системи, а також вирішити допоміжні завдання, наприклад, пов'язані з безпекою.

Кожен вузол на діаграмі розгортанняявляє собою деякий тип обчислювального пристрою, у більшості випадків – частину апаратури.

Створена діаграма містить такі вузли:

- Вузол веб-сервер – це веб-сервер системи, клієнт взаємодіє з ним через браузер допомогою протоколу HTTP.
- Вузол сервер бази даних – це сервер який містить базу даних.
- Вузол робоча станція – це девайс за допомогою якого клієнт взаємодіє з рештою веб-застосунку.

У вузлу веб-сервер (рис. 3.2) розгорнуто компоненти за шаблоном MVC, де вони взаємодіють кожен між собою. Цей вузол містить логіку роботи з базою даних та сутностями проекту які представлені в компоненті Model.

Компонент Controller забезпечує зв'язок браузера і моделями. Вузол View забезпечує візуалізацію відповідей на запити користувача.

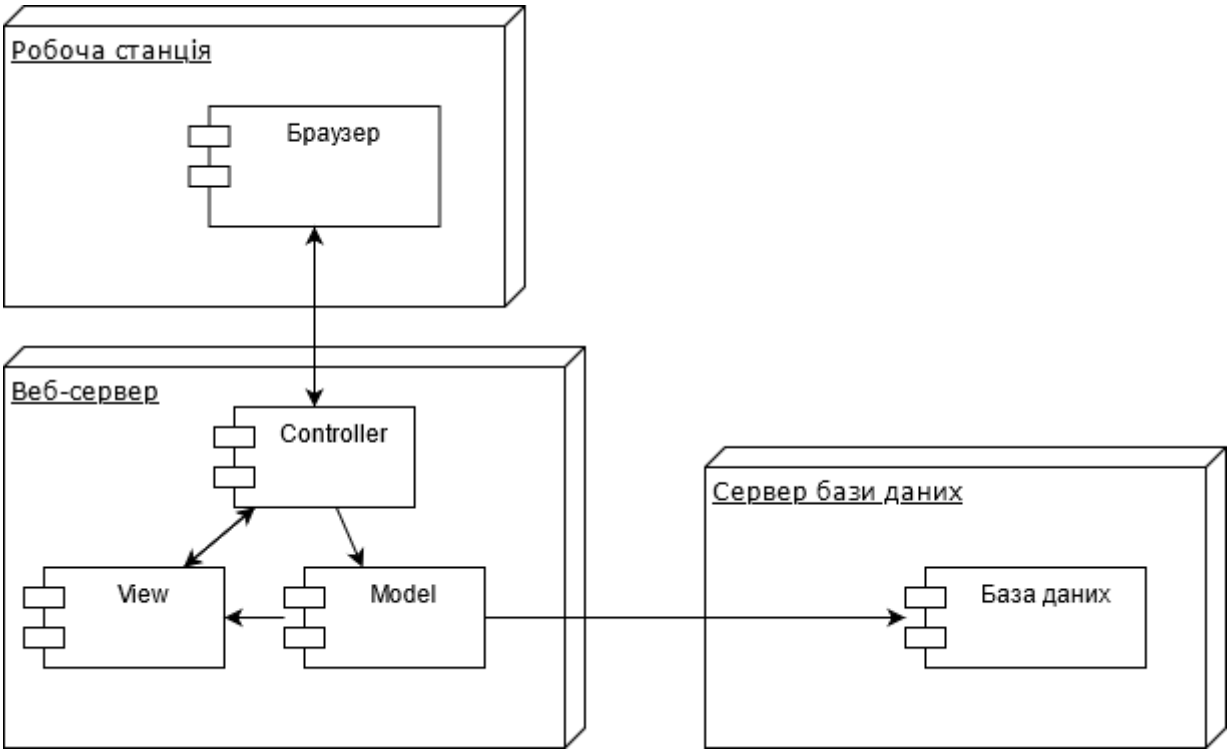

Рис. 3.3. Діаграма розгортання

3.4 Діаграма класів

Діаграма класів відображає класи та їх взаємовідносини, тим самим представляючи логічний аспект проекту. Діаграма класів визначає типи об'єктів системи й різного роду статичні зв'язки, які існують між ними.

Діаграма класів має три види зв’язків:

- Агрегація;
- Асоціація;
- Композиція.

Агрегація – це асоціація по типу «ціле-частина». Це зв’язок основного класу і з класами з яких складається основний клас.

Асоціація – це представлення відношення між екземплярами класів.

Композиція – це зв'язок де об’єкти не можуть існувати самі по собі, вони обов’язково потребують іншого класу-агрегатора.

Для полегшення розробки було розроблено діаграму класів (рис. 3.3). Вона дозволить легко зрозуміти зв’язки та сутності застосунку. На даній діаграмі було зображено основні класи та атрибути, що значно спростить написання коду.

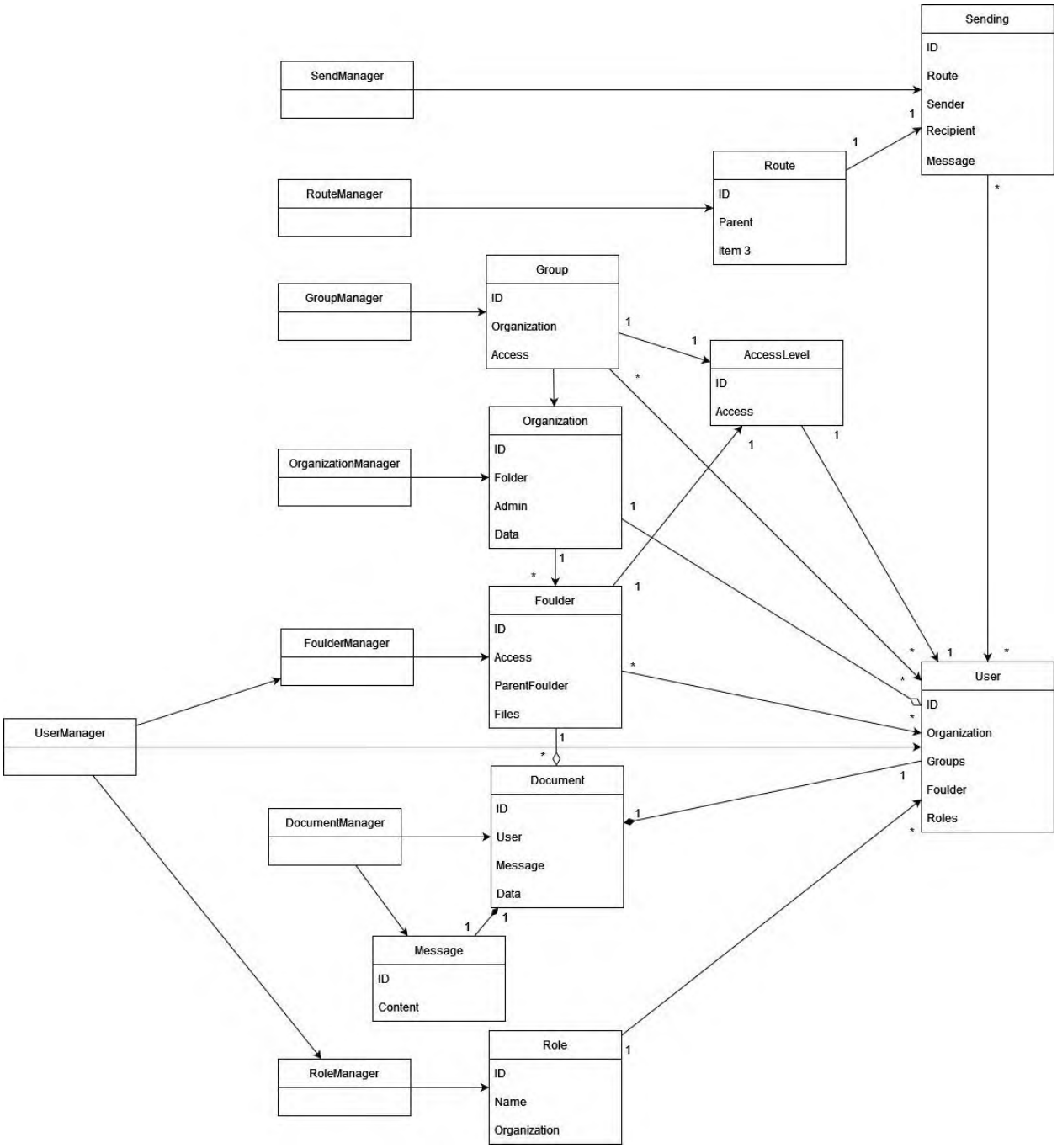

Рис. 3.4. Діаграма класів

Висновок

В даному розділі було розглянуто структуру веб-застосунку для подальшої реалізації. Адже правильно обрана структура проекту значно

полегшує процес розробки та підтримки ПЗ. Створення ПЗ без попередньої розробки структури майже завжди закінчується провалом. Особливо це стосується великих систем із значним розмаїттям функціоналу.

В ході аналізу структури було створено діаграму розгортання та діаграму класів. Ці дві діаграми відіграють важливу роль в процесі проектування ПЗ. Діаграма класів дозволяє уникнути плутанини в класах, їх зв’язках та структурі системи. А діаграма розгортання дає уявлення про організацію фізичних компонентів та допомагає вирішити ряд проблем в безпеці.

Для реалізації було обрано технологію ASP.NET Core MVC. Ця технологія гарно підходить для розробки веб-застосунків а також вона використовує шаблон MVC. Такий шаблон забезпечує поділ застосунку на частини, що спрощує структуру програми за рахунок більш строго поділу його рівнів.

Для роботи з даними було обрано Microsoft SQL Server. Він має багато переваг серед інших СУБД, і добре підходить для розробки веб-застосунків.

Розробивши структуру і обравши підходи до розробки можливо приступити до наступного етапу – реалізації веб застосунку.

РОЗДІЛ 4
РЕАЛІЗАЦІЯ ЗАСТОСУНКУ СИСТЕМИ ЦИФРОВОГО ДОКУМЕНТООБІГУ

4.1 Опис класів

В минулому розділі було обрано технологію та архітектуру для створення програмного засобу системи цифрового документообігу. Даний проект розробляється за допомогою технології ASP.NET Core MVC, яка дотримується шаблону MVC. Використовуючи цей шаблон було розроблено застосунок який поділено на моделі контролери та представлення. Класи моделей, та контролерів схожі за функціоналом та складом. Тому для опису було відібрано тільки найголовніші класи.

До моделей відносяться наступні класи:

- Document – документ;
- User – користувач;
- Route – маршрут;
- Group – група;
- Organization – організація;
- Role – роль;
- Message – тип повідомлення;
- Foulder – папка;
- AccessLevel – рівень доступу;
- Sending – відправка документу.

На рисунку 4.1.1 зображено код класу Document. В цьому класі є поле Id для ідентифікації документів, User, Message, та поле Data. Так поле Id є унікальним ідентифікатором для доступу до кожного документа. Поле User містить посилання на користувача, який створив цей документ. Поле Message відповідає за коротке повідомлення яке буде додано до листа документа. Поле Data містить в собі дані документа, при відправці. Також в класі Document

існують поля для інших властивостей: тип документу, дата відправки, дата створення, електронний підпис на ньому, та додатки які прикріплюються.

```
public class ObjectInfo
{
    /// <summary>
    /// Код объекта.
    /// </summary>
    [MapField("IDOBJ")]
    – references
    public Guid Id { get; set; }

    /// <summary>
    /// Код родительского объекта.
    /// </summary>
    [MapField(SqlName = "IDPOBJ", CanMissing = MissingType.Sometimes)]
    – references
    public Guid IdParent { get; set; }

    /// <summary>
    /// Папки в которых находиться документ, и они доступны пользователю.
    /// </summary>
    [MapField(CanMissing = MissingType.Always)]
    – references
    public List<ParentObject> ParentObjects { get; set; }

    /// <summary>
    /// Скрыт в корзине
    /// </summary>
    [MapField(CanMissing = MissingType.Always)]
```

Рис. 4.1.1. Клас Document

Наступний важливий клас це User (рис. 4.1.2). Цей клас використовується для відображення в системі користувачів. Основні поля це: Id, Name, Surname, Login, Type, Organization, Groups, Foulder, Roles. Id відповідає за ідентифікацію в системі.

```
internal class UserInfoWithEmailAndTel : ShortUserInfo
{
    /// <summary>
    /// Почта
    /// </summary>
    [MapField("EMAIL")]
    0 references
    public string Email { get; set; }

    /// <summary>
    /// Телефон пользователя
    /// </summary>
    [MapField(SqlName = "TEL")]
    2 references
    public string Tel { get; set; }

    /// <summary>
    /// Телефон пользователя
    /// </summary>
    [MapField(SqlName = "MOBILE")]
    0 references
    public string Mobile { get; set; }
```

Рис. 4.1.2. Клас User

Name і Surname слугують для відображення ім’я користувача. Login використовується для відправки повідомлень. Type – відображує тип користувача, тобто це може бути адміністратор чи користувач з обмеженим функціоналом. Organization – посилання на організацію де був створений користувач. Groups містить список груп в організації, до яких додано користувача. Foulder це посилання на особисту папку користувача. Roles відображає ролі в групах та організаціях користувача, представляється в вигляді словника. Також тут міститься інформація про email та телефон користувача.

Клас Route слугує для відображення маршруту в системі. (рис. 4.1.3).

```
public class Route
{
    /// <summary>
    /// Отображать статус текущего шага
    /// </summary>
    [MapField("SHCURRSTEP")]
    [JsonIgnore]
    private string _showCurrentStepState;

    /// <summary>
    /// Уникальный идентификатор маршрута
    /// </summary>
    [MapField("UNID")]
    [JsonProperty(PropertyName = "uid")]
    public Guid Uid { get; private set; }

    /// <summary>
    /// Код маршрута
    /// </summary>
    [MapField("ID")]
    [JsonProperty(PropertyName = "id")]
```

Рис. 4.1.3. Клас Route

Він також містить поля ідентифікації, та дані необхідні для коректного надсилання, та дотримання всіх вимог в цьому процесі.

Іншим важливим класом серед моделей є Role (рис. 4.1.4). Цей клас відповідає за ролі в групах та організаціях. Це необхідно для того щоб розмежувати обов’язки для користувачів, та запропонувати простий інтерфейс для відображення необхідної посади в підписі документа. Цей клас містить

ідентифікатор, назву ролі, організації, а також додаткові параметри для детальнішого налаштування, через цю властивість.

```
public class SignaturesRole : SignaturesRoleInfo
{
    /// <summary>
    /// Роль подписи
    /// </summary>
    [MapField("ROLE", CanMissing = MissingType.None)]
    [JsonProperty("roleId")]
    6 references
    internal string RoleId { get; set; }

    /// <summary>
    /// Название роли
    /// </summary>
    [MapField("NAMEROLE", CanMissing = MissingType.None)]
    [JsonProperty("roleName")]
    1 reference
    internal string RoleName { get; set; }

    /// <summary>
    /// Код контрагента
    /// </summary>
    [MapField("ORG", CanMissing = MissingType.None)]
    [JsonProperty("orgId")]

    internal int OrgId { get; set; }

    [MapField(CanMissing = MissingType.Always)]
    [JsonIgnore]

    internal string Signature { get; private set; }

    [JsonConstructor]
    public SignaturesRole()
```

Рис. 4.1.4. Клас Role

Клас Foulder зображує сутність папки. Для реалізації деревовидної структури містить Id та ParentId. Id використовується для ідентифікації об’єктів, а ParentId для зберігання посилання на батьківський об’єкт. Так якщо посилання на батьківський об’єкт пустий, то це корінна папка. Також клас Foulder зберігає поля Organization та User. Ці поля містять посилання на організацію та користувача відповідно. Якщо одне з цих полів а також поле батьківської папки пусті, то це означає що це корінна папка, або організації

або користувача. Також для зберігання та відображення даних клас містить поля з інформацією про назву, опис, тип, дату створення, тощо.

Наступним є клас Group (рис. 4.1.5). Він містить наступні поля: Id, Name, Organization, Access, Description. Ці основні поля забезпечують ідентифікацію, назву, опис, організація, та доступ. Також та присутні додаткові поля для відображення та деталізації даних. Для збереження та використання параметрів існує поле Params у вигляді словника.

```
– references
public class GroupParameters : UpdatableGroupParameters
{
    /// <summary>
    /// Код организации
    /// </summary>
    [JsonProperty("org")]
    73 references
    public int? OrganisationId { get; set; }

    /// <summary>
    /// Системная группа
    /// </summary>
    [JsonProperty("sys")]
    11 references
    public bool IsSystemGroup { get; set; }

    /// <summary>
    /// Внутреняя группа
    /// </summary>
    [Obsolete]
    [JsonIgnore]
    2 references
    public string IsInnerGroup { get; set; }

    /// <summary>
    /// Тип группы
    /// </summary>
    4 references
    private TypeGroup _typeGroup { get; set; }

    /// <summary>
```

Рис. 4.1.5. Клас Group

В описі Organization містяться дані про організації про її адміністратора, ролі, папки, назву, детальний опис, та параметри. Для ідентифікації використовується поле Id. Name і Description дозволяють описати організації та отримати детальні дані для відображення.

В клас Message є лише кілька основних полів, таких як Id та Content, проте також є поле Param для детальнішого налаштування та відображення повідомлення.

AccessLevel – відображує сутність рівня доступу. Він має лише кілька полів таких як Id, Name, Available. Available зображує доступність цього рівня доступу. Якщо ця дата буде меншою ніж дата серверу, то це означає що цей рівень більше не доступний. Інші данні зберігаються в вигляді словника.

Клас Sending відповідає за відправку документів. Його поля це: Id, Route, Sender, Recipient, Message, Attachments, Date, тощо. Вони зберігають інформацію про маршрут, відправника, отримувача, повідомлення, документи, дату відправки та інші дані в вигляді словника.

Контролери відповідають за обробку запитів та виклик необхідних представлень з передачею в них моделей.

До контролерів розробленого застосунку належать:

- UserManager;
- RoleManager;
- DocumentManager;
- FoulderManager;
- OrganizationManager;
- GroupManager;
- RouteManager;
- SendManager.

Контролер UserManager забезпечує роботу з користувачами (рис. 4.1.6). Він містить методи для роботи з користувачами, організаціями користувачів. Також цей контролер відповідає за створення нових редагування та видалення користувачів.

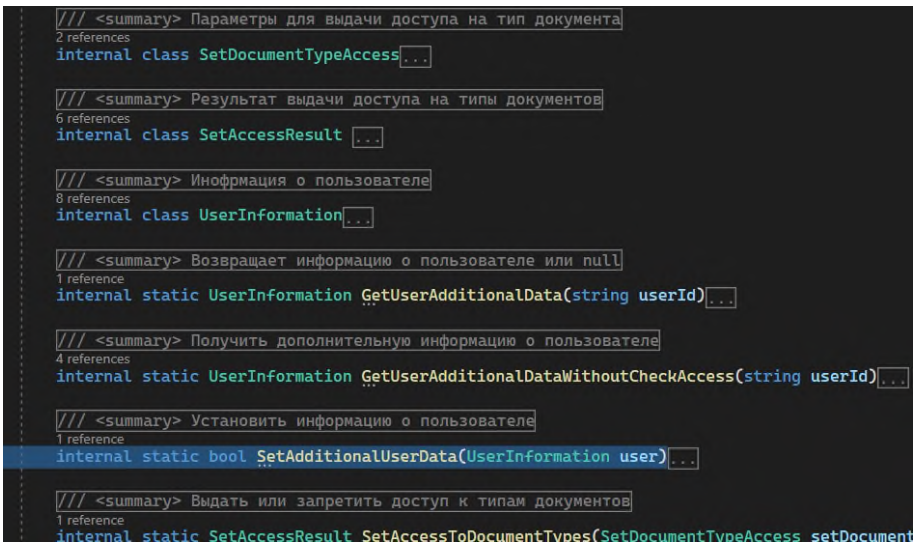

Рис. 4.1.6 Клас UserManager

Наступний контролер це DocumentManager (рис. 4.1.7). Даний контролер відповідає роботу з документами. До його функцій належать: створення, редагування, видалення, переміщення та прив’язка документів. Також він зв’язує форму з даними в БД.

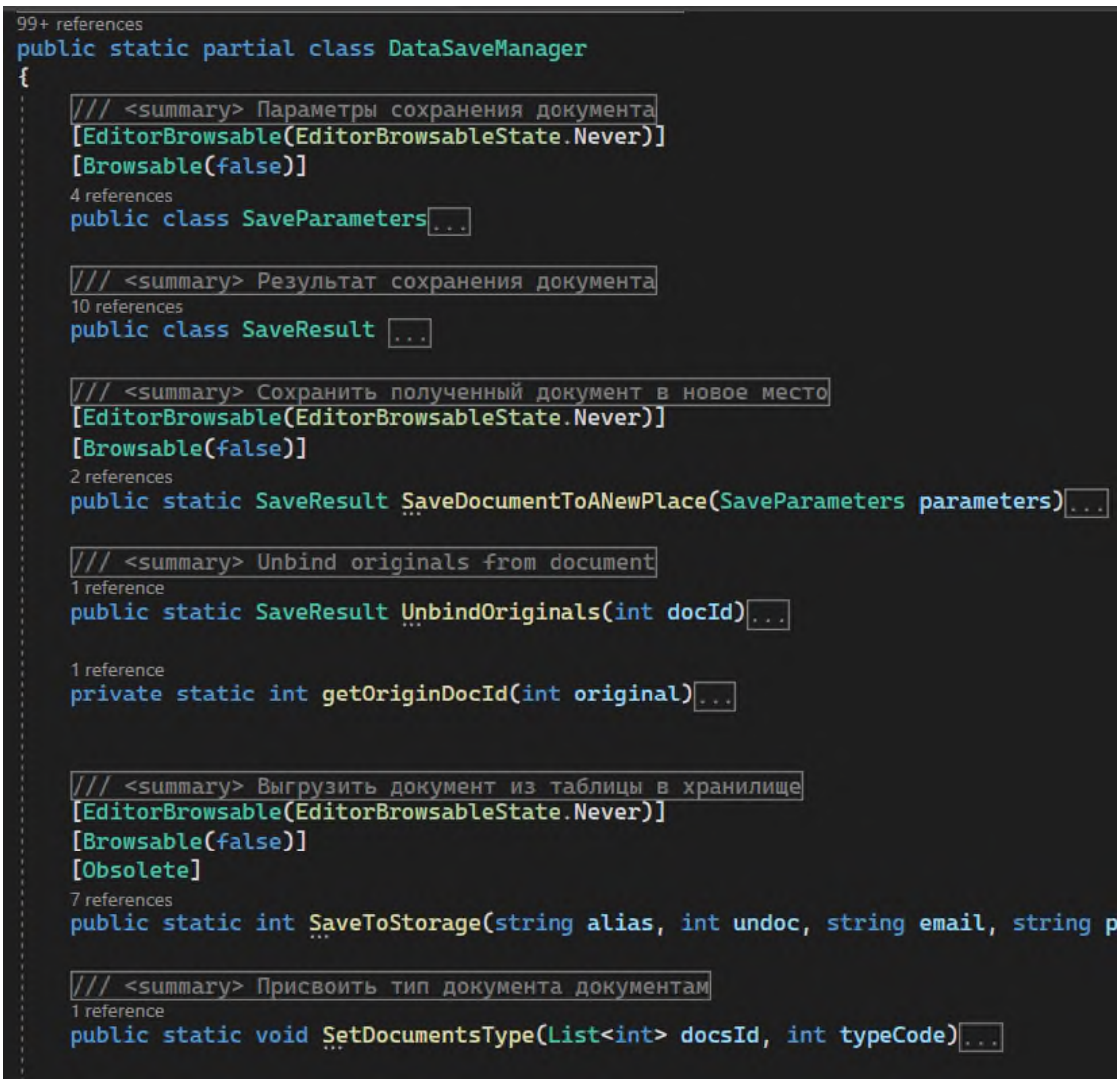

Рис. 4.1.7 Клас DocumentManager

SendManager – це контролер який відповідає за доступ до методів надсилання документів. Основний його функціонал це:

- Отримання документів;
- Збір в єдиний лист;
- Отримання вкладень та повідомлень;
- Визначення отримувача;
- Запуск маршрутів;
- Запуск процесів надсилання та скасування документів.

Наступним контролером є контролер RoleManager (рис. 4.1.8). Цей контролер забезпечує управління ролями. До нього відноситься надання, та

редагування ролей в групах і організаціях. А також він відповідає за зв'язок ролей та підписів.

```
public class SignaturesRolesManager
{
    private readonly string _userId;

    private const string _signaturesRolesTable = "FSSSROLE";
    private const string _rolesTable = "AGROLE";

    16 references
    public SignaturesRolesManager()...

    2 references
    internal SignaturesRolesManager(string userId)...

    1 reference
    internal static DocumentManager.SignType GetSignatureType(string keyNumber, string signature)...

    /// <summary> Получить изменения справочника ролей по дате
    1 reference
    internal List<SignaturesRole> GetChangedSignaturesRoles(DateTime dateLastExchange, int org)...

    /// <summary> Сохранить изменения ролей подписей
    1 reference
    internal bool SaveSignaturesRolesChanges(List<SignaturesRole> signaturesRoles)...

    /// <summary> Получить должности
    2 references
    internal static ICollection<RoleInfo> GetRoles(string userId = null, ICollection<string> roleCode

    /// <summary> Проверить доступность роли
    2 references
    internal static bool CheckRoleAvailibility(string roleCode, int? organizationCode = null)...

    /// <summary> Получить список последних использованных подписей
    1 reference
    internal List<SignaturesRoleInfo> GetLatestUsedSignatures(int orgId, int entriesCount = 3)...

    /// <summary> Добавить подпись в справочник организаций
```

Рис. 4.1.8. Клас RoleManager

Для створення та управління маршрутами створено контролер RouteManager. Він керує маршрутами, створює та підпорядковує їх. Функціонал ралізує багато методів та сценаріїв перевірок та проходження цих маршрутів.

Для обробки запитів до папок було створено контролер FoulderManager (рис. 4.1.9). Основна мета цього контролера це обробка даних та створення і редагування папок. Саме цей контролер відповідає за створення ієрархії в папках, шляхом побудови дерева папок, де в кожному дочірньому елементі буде знаходитися посилання на батьківський. Зокрема папка організації, групи чи користувача буде вважатися корінною, тобто її батьківський елемент буде пустим. Таким чином можна легко виявити де корінна папка користувача та

почати пошук необхідних файлів. Він відповідає за доступ, перевірку правил та контроль в папках.

```
internal class Address
{
    [MapField("ID")]
    [JsonProperty("id")]
    9 references
    internal Guid Id { get; set; }

    [MapField("ORG")]
    [JsonProperty("org")]
    6 references
    internal int OrgId { get; set; }

    [MapField("ADDRESS")]
    [JsonProperty("address")]
    9 references
    internal string AddressName { get; set; }

    [MapField("OKPOORINN", CanMissing = MissingType.Sometimes)]
    [JsonProperty("okpoOrInn")]
```

Рис. 4.1.9. Клас FoulderManager

Контролер OrganizationManager (рис. 4.1.10) забезпечує управління організаціями. Цей клас дозволяє створювати видаляти та керувати ними. Також він відповідає за пошук організацій, отримання даних відповідно до полів і налаштувань інтеграцій.

```
/// <summary> Получить список текущих интеграций организации
1 reference
internal static GetIntegrationResult GetIntegrations(string okpoOrInn, bool checkAcce

/// <summary> Удалить настройки интеграции
1 reference
internal static string DeleteIntegration(IntegrationParameters parameters)...

/// <summary> Параметры интеграции
16 references
internal class IntegrationParameters...

/// <summary> Добавить или установить токен интеграции
1 reference
internal static string SetOrUpdateIntegration(IntegrationParameters parameters)...

/// <summary> Создать организацию в таблице привязки организаций к объектам учет ...
3 references
private static bool createKobjOrganization(int organization)...

/// <summary> Получить список общих организаций для пользователей
1 reference
internal static List<int> GetUsersCommonOrganizations(List<string> users)...

/// <summary> Finds organizations by text (returns top 100 records).
5 references
internal static List<int> FindOrganizationByText(string searchText)...

/// <summary> Найти организации по окпо или инн или названию
1 reference
internal static List<int> FindOrganizationsByText(List<string> okpoOrInnsOrNames)...
```

Рис. 4.1.10. Клас OrganizationManager

Наступним контролером є GroupManager (рис. 4.1.11). Він дозволяє роботу з групами. В методах цього класу описано реалізовано керування групами. Також він дозволяє пошук груп, та отримання даних по групах. Цей клас тісно співпрацює з класом організацій, бо вони реалізують функціонал, який доповнює один іншого. Тож деякі функції дозволяють отримувати організації за групами та навпаки.

```
public class GroupManager
{
    /// <summary> Обновить наименование папки группы "Документы организации".
    1 reference
    public static void UpdateOrganizationGroupFolderName(int dayCount)...

    /// <summary> Имя таблицы групп
    private const string _tableName = "FSSGROUP";

    /// <summary> Информация о пользователе группы
    3 references
    public class GroupUserInfo ...

    /// <summary> Параметры которые можно изменять для группы
    16 references
    public class UpdateGroupParameters ...

    /// <summary> Обновляемые параметры группы
    2 references
    public class UpdatableGroupParameters...

    /// <summary> Параметры для создания или изменения группы
    90 references
    public class GroupParameters ...

    /// <summary> Удаление групп
    2 references
    private class GroupDeleter...

    /// <summary> Удалить группу
    2 references
    public static bool DeleteGroup(Guid groupId, out string errorMessage)...

    /// <summary> Параметры поиска группы
    10 references
    public class FindGroupParameters...
```

Рис. 4.1.11. Клас GroupManager

Окрім наведених вище класів було створено також допоміжний клас DBContext. Він разом з класами-репозиторіями реалізують основний

функціонал роботи з базою даних. Проте для налаштування зв’язку з базою даних існує також файл dbsettings.json.

Також в проекті існують кілька стандартних класів: Startup та Program. Виконання програми починається з класу Program, після чого клас Startup проводить налаштування та підключає корисні додаткові функції в застосунок.

За представлення веб-застосунку відповідають View-файли. Вони мають розширення .cshtml. В файлах з таким розширенням використовується технологія Vue JS. Ця технологія поєднує C#, JS, CSS та HTML, що дозволяє впроваджувати код в веб-сторінки.

Для кожного контролера створюється набір представлень які забезпечують відображення певної інформації. На рисунку 4.1.12 зображений перелік представлень.

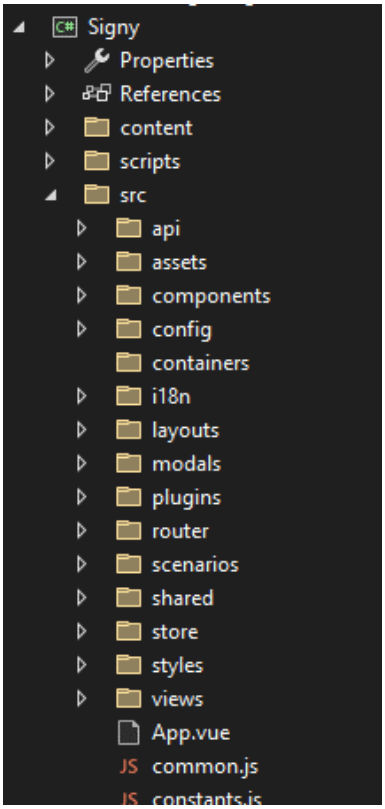

Рис. 4.1.12. Представлення

Також існують представлення які працюють як шаблони. В ці шаблони вбудовуються представлення від контролерів. Один з основних таких шаблонів це _Layout.cshtml (рис. 4.1.13). В цьому представленні написано код для відображення сторінки з папками та документами. Використовуючи Vue

JS і HTML вдається легко створювати контейнери для всіх елементів, щоб забезпечити найкраще відображення на веб сторінці.

DocRouteStepEditor.vue

```
<template>
  <div>
    <b-container fluid class="px-0">
      <b-row no-gutters class="list-row">
        <b-col>…</b-col>
      </b-row>
      <b-row class="align-items-center sender-row">…</b-row>
      <b-row class="align-items-start whom-row">…</b-row>
      <b-row class="attachment-row align-items-right">…</b-row>
      <b-row class="role-row align-items-center">…</b-row>
      <b-row no-gutters class="add-recipient-btn-wrapper">
        <b-col cols="6" class="text-right ml-auto">
          <div v-if="editable">
            <b-button
                id="add-position-btn-doc-rout-step-editor"
                v-show="!simpleRoute"
                variant="primary"
                @click="addPosition"
                :disabled="!canAddRecipient">
              <template v-if="addingPos">
                <b-spinner id="add-position-spinner-doc-rout-step-editor" small />
              </template>
              <template v-else>
                <span><i class="fal fa-user-plus" /></span>
              </template>
              <span class="ml-2">{{ $t('fss.docRoute.addRecipient') }}</span>
            </b-button>
            <b-button
                id="save-route-btn-doc-rout-step-editor"
                variant="primary"
                :disabled="saveDisabled"
                @click="saveClick">
              <span class="mr-2"><i :class="firstSend ? 'fal fa-share-square' : 'fal fa-save'" /></span>
              <span>{{ firstSend ? $t('fss.action.send') : $t('fss.docRoute.save') }}</span>
            </b-button>
          </div>
          <div v-else>
```

Рис. 4.1.13. Представлення _Layout.cshtml

4.2 Опис інтерфейсу

При запуску веб-застосунку відкривається головна сторінка де виводиться сторінка з особистими документами (рис. 4.2.1).

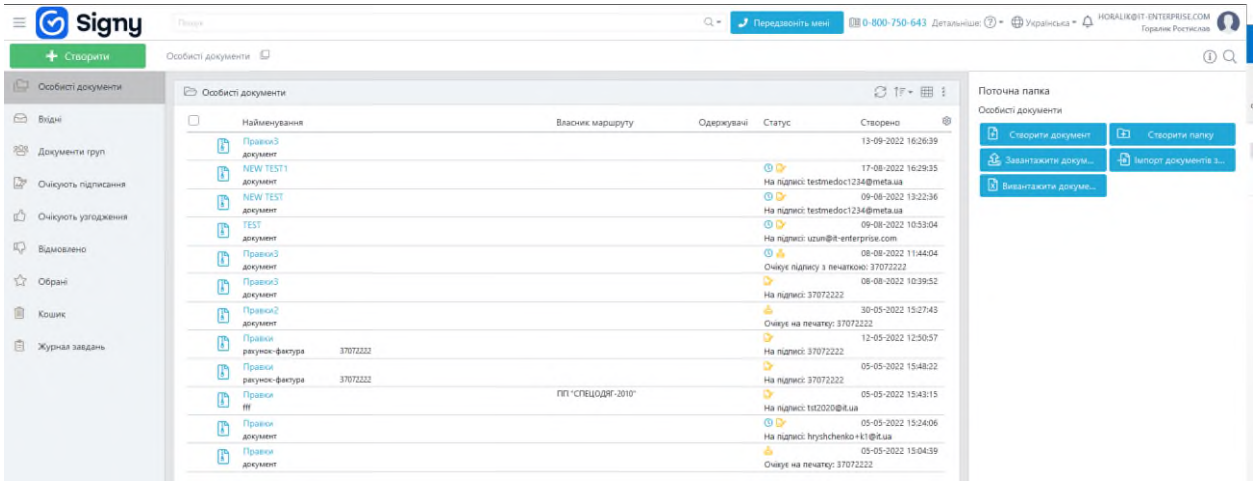

Рис. 4.2.1. Головний екран

На рисунку 4.2.2 зображено особистий кабінет. Тут містяться налаштування користувача, організацій, груп, документів і сповіщень.

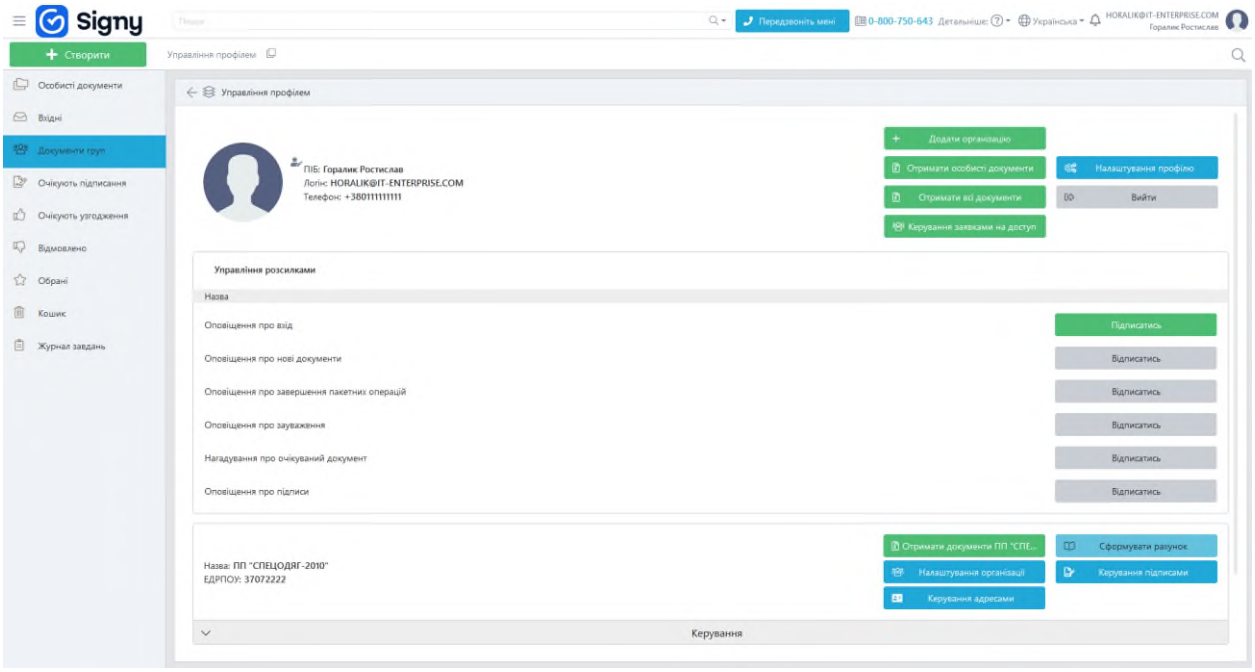

Рис. 4.2.3. Особистий кабінет

На рисунку 4.2.4 виведено папку організації та історію по одному з документів.

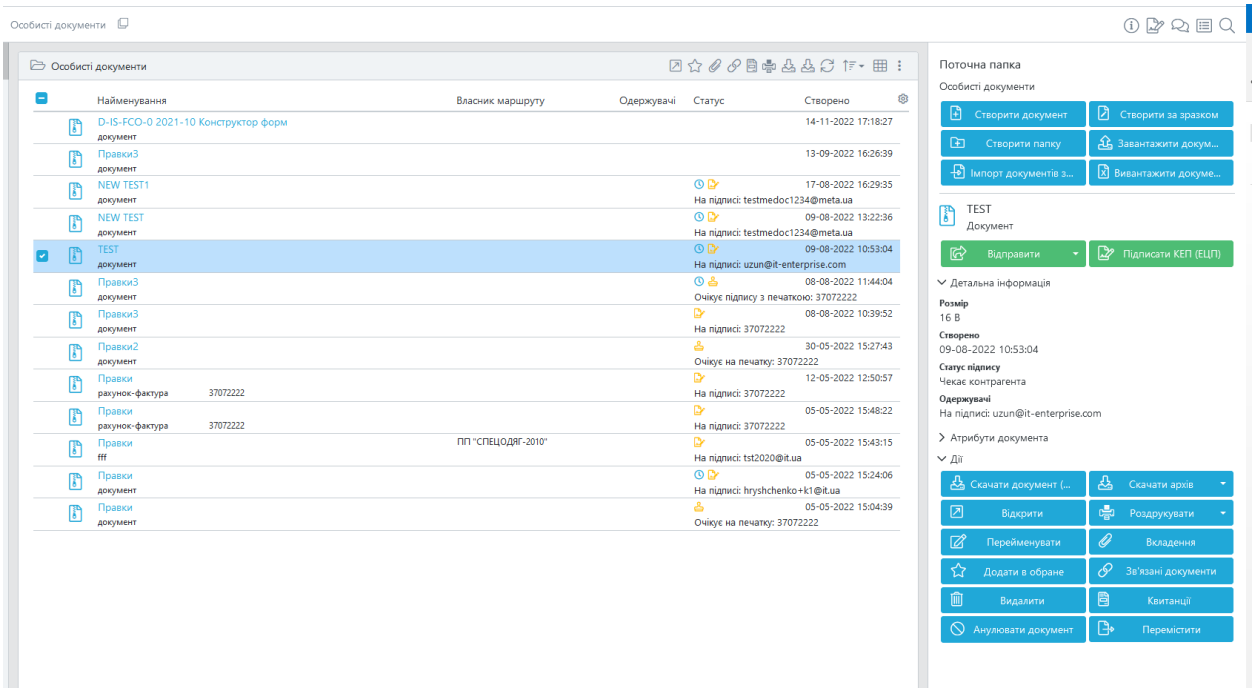

Рис. 4.2.4. Папка організації

Сторінка надсилання документа зображена на рисунку 4.2.5. Там знаходяться основні поля які необхідно заповнити для надсилання документу. Також можна відразу накласти електронний підпис, та обрати налаштування для надсилання документу.

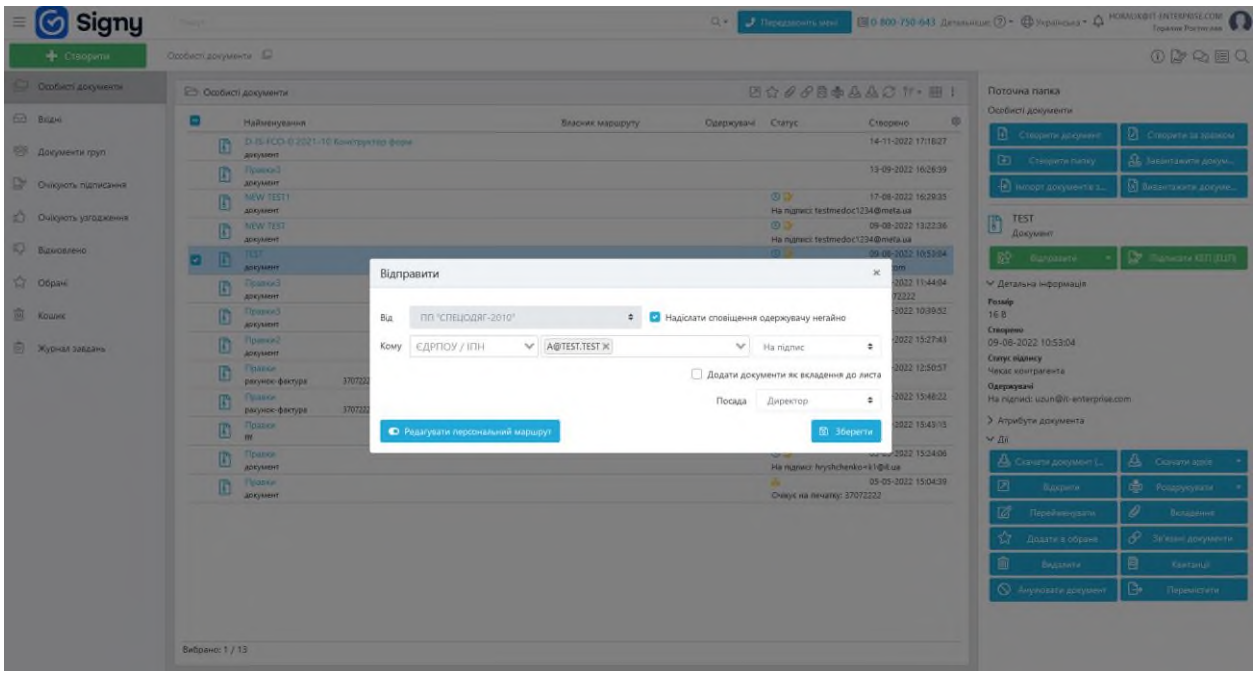

Рис. 4.2.5. Надсилання документу

Реалізацію функції перегляду та редагування документу зображено на рисунку 4.2.6. Тут міститься візуальний редактор вбудований в вікно системи. Та дані про цей документ зображено на правому фреймі.

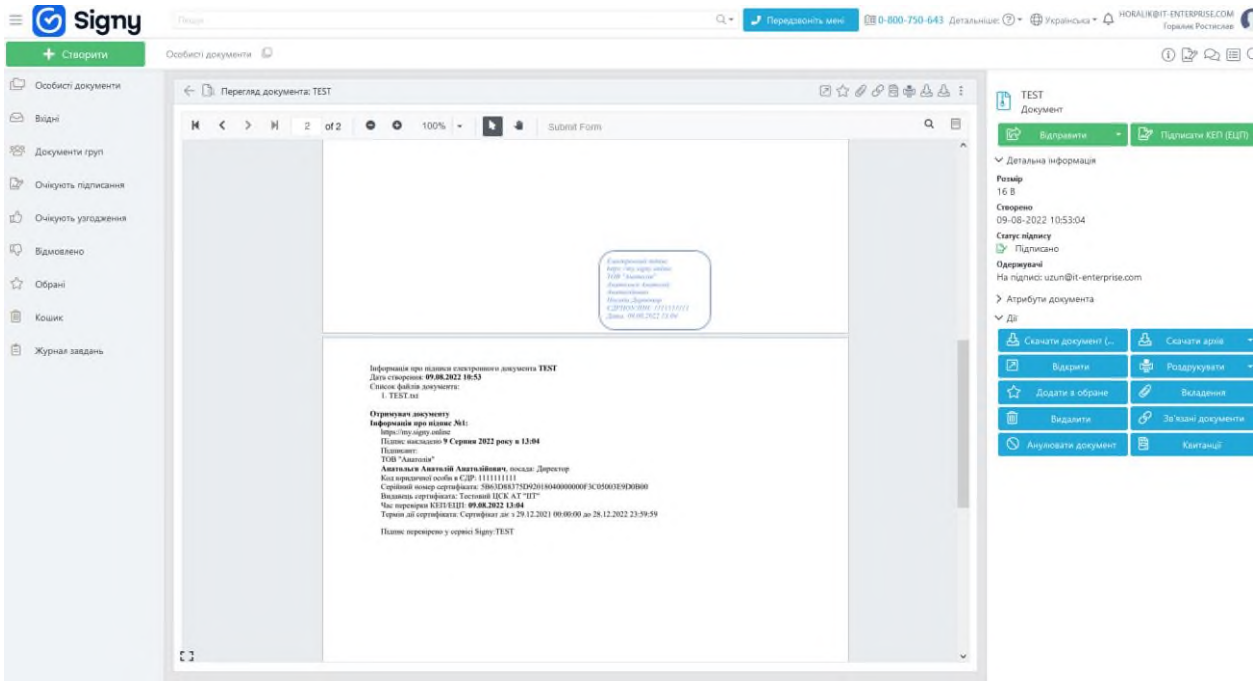

Рис. 4.2.6. Перегляд документу

Наступна сторінка це історія документу зображена на рисунку 4.2.7. Вона відображається на правому фреймі в вигляді списку подій та коротким

описом з датою цієї дії. Таким чином можна чітко виявити хто, коли та які дії виконував з документом.

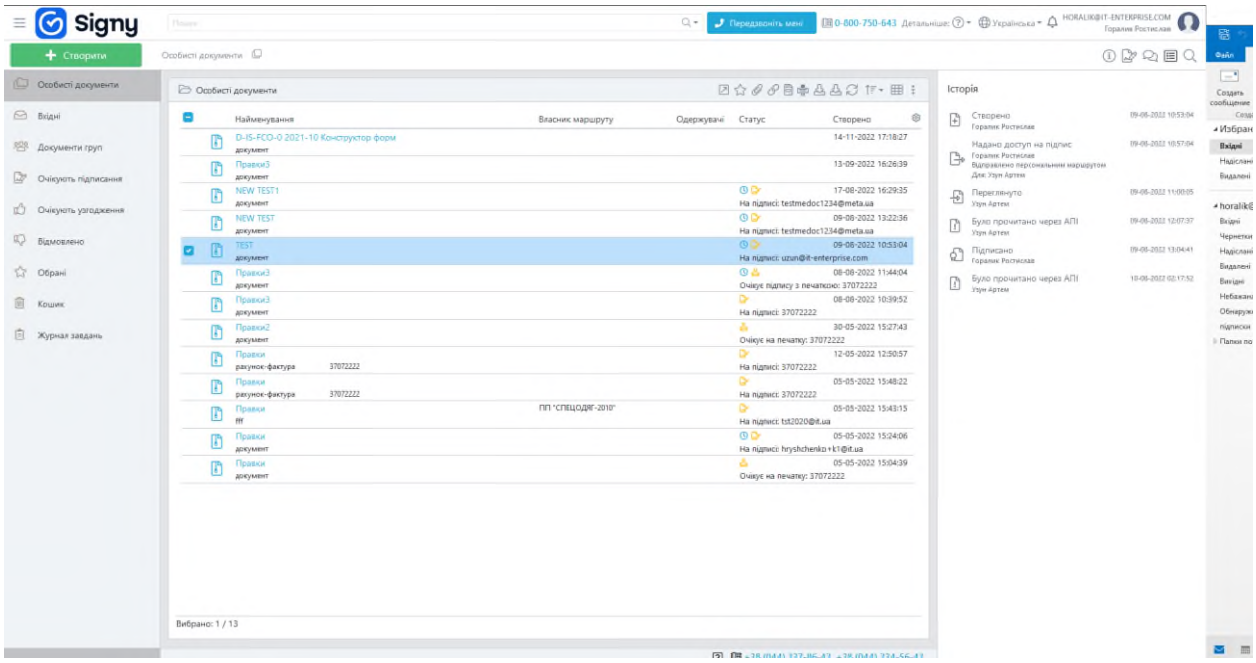

Рис. 4.2.7. Історія документу

Це основні сторінки створеної системи цифрового документообігу. Інтерфейс програмного застосунку був розроблений з використанням вимог та норм створення таких інтерфейсів, які були визначені раніше.

Висновок

В даному розділі було розглянуто результат розробки програмного застосунку система цифрового документообігу. На основі вимог та структурі застосунку описаних в минулих розділах було розроблено застосунок з використанням технології ASP.NET Core MVC. Було описано основні моделі та контролери розробленого застосунку. Для дослідження представлення було описано кілька файлів. А також було продемонстровано інтерфейс веб-застосунку.

ВИСНОВКИ

В даній дипломній роботі було проаналізовано область цифрового документообігу. Було розглянуто сучасний стан та проблеми, які заважають розвитку веб застосунків. Було виявлено певні проблеми в даній області, що сповільнюють розробку таких систем. Також було проаналізовано проблеми впровадження систем цифрового документообігу в Україні. Завдяки дослідженню аналогів цієї системи було виявлено недоліки та проблеми які варто розглянути перед створенням власного продукту.

Для визначення вимог та потреб на ринку цифрового документообігу були проаналізовані існуючі аналоги, які користуються найбільшим попитом в світі та Україні. Були визначені функціональні та нефункціональні вимоги, а також вимоги до типів документів з якими дозволяє працювати система. В результаті виявлено переваги та недоліки кожного аналогу, що дозволяє створити веб-застосунок, високої якості та задовольнити потреби ринку.

Аналіз предметної області дав змогу створити чіткі вимоги та очікування до створюваного веб-застосунку. Для кращого відображення функціональних вимог було побудовано діаграму прецедентів. На ній було відображено основний функціонал цієї системи.

Маючи чіткий список вимог вдалося скласти структуру для програмного застосунку системи цифрового документообігу. Було обрано одні з найкращих та найпопулярніших засобів і технологій, які дозволяють реалізувати всі вимоги. Задля полегшення процесу розробки та впровадження було створено діаграму класів та діаграму розгортання. На цій діаграмі було зображено лише основні вузли через великий об’єм цієї системи. Проте цього було достатньо щоб значною мірою спростити процес розробки застосунку цифрового документообігу.

В результаті було розроблено програмний засіб цифрового документообігу, який зможе задовільнити потреби користувачів. Було продемонстровано інтерфейс веб застосунку, та реалізацію основних класів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Overview of ASP.NET Core MVC: [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/enus/aspnet/core/mvc/overview?view=aspnetcore-5.0
2. Вимоги до сучасних інтерфейсів користувача. Демидов Дмитро Григорович. Журнал: Вісник Московського державного університету друку, 2016. [Електронний ресурс] – Режим доступу: https://cyberleninka.ru/article/n/trebovaniya-k-sovremennym-polzovatelskim-interfeysam/viewer
3. Architect Modern Web Applications with ASP.NET Core and Azure: [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/
4. Entity Framework Core Documentation And Tutorials | Learn Entity Framework Core **[**Електронний ресурс] – Режим доступу до ресурсу: https://www.learnentityframeworkcore.com/
5. Матвієнко О. В. Основи організації електронного документообігу: навч. посіб. / О. В. Матвієнко, М.Н. Цивін. – К. : ЦУЛ, 2008. – 112с.
6. Швецова Г. М. Документознавство : навч. посіб. / Г. М. Швецова. – К. : Знання, 2007. – 398 с.
7. Main concepts of the document management system required for its implementation in enterprises. **[**Електронний ресурс] – Режим доступу до ресурсу: https://journal.eu-jr.eu/sciencerise/article/view/1384/1284
8. Electronic Document Management System (EDMS) Implementation **[**Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/357259844_Electronic_Document_Management_System_EDMS_Implementation_Implications_for_the_Future_of_Digital_Transformation_in_Philippine_Healthcare

9. Integrated Management Systems and Workflow-Based Electronic Document Management. [Електронний ресурс] – Режим доступу до ресурсу: http://dx.doi.org/10.3926/jiem.846
10. Comparison Of Document Management Systems By Meta Modelling And Workforce Centric Tuning Measures [Електронний ресурс] – Режим доступу до ресурсу: https://arxiv.org/abs/1403.3131v1
11. Система електронного документообігу. Електронний цифровий підпис. [Електронний ресурс] – Режим доступу до ресурсу: http://www.znannya.org/?view=e-government-documents
12. Цимбалюк В.І. Впровадження електронних систем документообігу в Україні: проблеми та шляхи їх вирішення. [Електронний ресурс] – Режим доступу до ресурсу: http://nbuv.gov.ua/UJRN/Pinform_2014_4_4