

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти – бакалавр

за освітньо-професійною програмою

«Комп'ютерні науки»

зі спеціальності

122 – Комп'ютерні науки

тема роботи:

***«Використання технології blockchain для систем електронного
документообігу»***

Виконав студент гр. КН-20 _____ Шаблій Я. Е.

Керівник _____ Маринич І. А.

Нормоконтроль _____ Маринич І. А.

Завідувач кафедри _____ Рубан С. А.

Кривий Ріг – 2024

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Зав. кафедрою: к.т.н. Рубан С.А.

« 27 » березня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

студентові групи КН-20 Шаблію Ярославу Едуардовичу

1. Тема кваліфікаційної роботи: «Використання технології blockchain для систем електронного документообігу»

затверджено наказом по університету № 235с від 27.03.2024 р.

2. Термін здачі кваліфікаційної роботи: 05.06.2024 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка обсягом 78с., додатки, презентація у Microsoft PowerPoint (13 слайдів) в електронному та друкованому вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-2

доц. Маринич І. А.

Нормоконтроль

доц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>01.04.24</i>
2	<i>Розділ 1</i>	<i>05.04.24</i>
3	<i>Розділ 2</i>	<i>01.05.24</i>
4	<i>Висновки</i>	<i>25.05.24</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>28.05.24</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>20.05.24</i>
7	<i>Підготовка доповіді до захисту</i>	<i>05.06.24</i>

6. Дата видачі завдання: 29.01.2024р.

Керівник _____ / Маринич І. А./

7. Запевнення: Я, Шаблій Ярослав Едуардович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент _____ / Шаблій Я. Е./

АНОТАЦІЯ

Шаблій Я. Е. Використання технології blockchain для систем електронного документообігу

Кваліфікаційна робота на здобуття ступеня вищої освіти – бакалавр, за спеціальністю 122 Комп'ютерні науки. Криворізький національний університет, Кривий Ріг, 2024.

Робота складається зі вступу, двох розділів, висновків, переліку використаної літератури з 22 позицій. Загальний обсяг роботи становить 78 сторінок, з яких основний зміст роботи викладено на 75 сторінках, включає 21 таблицю і 33 рисунки.

Актуальність теми зумовлена зростаючою потребою в системах електронного документообігу серед вітчизняних підприємств та відсутності ефективних аналогів.

Метою роботи є розробка системи документообігу за допомогою технології блокчейн.

Об'єктом дослідження є система електронного документообігу.

Для досягнення поставленої мети розв'язуються такі завдання: дослідження і аналіз аналогів систем електронного документообігу; розробка системи електронного документообігу; розробка інтеграції технології блокчейн; розробка і тестування архітектури системи електронного документообігу; створення інтерфейсу користувача.

Робота повністю висвітлює необхідність таких систем, адже сфера блокчейн швидко розширюється, і потреба в системах електронного документообігу на підприємствах продовжує зростати.

Ключові слова:

БЛОКЧЕЙН, БЛОКЧЕЙН ТЕХНОЛОГІЯ, ЕЛЕКТРОННИЙ
ДОКУМЕНТООБІГ, ЕЦП, СИСТЕМА ЕЛЕКТРОННОГО
ДОКУМЕНТООБІГУ, СЕД

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ОПИС ОСНОВНИХ ПОНЯТЬ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	7
1.1 Основні поняття та визначення	7
1.1.1 Існуючі види документообігу	9
1.1.2 Методи організації документообігу.....	11
1.1.3 Призначення електронного документообігу	11
1.2 Структура та принципи електронного документообігу	12
1.3 Порівняльний аналіз типів документообігу.....	15
1.3.1 Недоліки електронного та традиційного документообігу	15
1.3.2 Переваги електронного документообігу над паперовим.....	17
1.3.3 Підходи для побудови СЕД.....	19
1.3.4 Підготовка документів.....	21
1.3.5 Склад СЕД.....	22
1.3.6 Структура СЕД.....	24
1.4 Огляд існуючих СЕД.....	25
1.4.1 Docs Fusion (розроблена компанією Hummingbird).....	25
1.4.2 Documentum.....	25
1.4.3 FossDoc.....	26
1.5 Властивості та вимоги до СЕД	26
1.6 Постановка завдання.....	28
Висновки до розділу.....	29
РОЗДІЛ 2. АНАЛІЗ МОЖЛИВОСТЕЙ ТЕХНОЛОГІЇ БЛОКЧЕЙН, РОБОТКА ТА РЕАЛІЗАЦІЯ СЕД	30
2.1 Застосування технології blockchain при побудові СЕД.....	30
2.2 Принцип роботи та архітектура технології blockchain	31
2.3 Функціональний опис системи	36
2.4 Вибір та обґрунтування компонент.....	39
2.5 Приклади варіантів використання.....	43

2.6 Розробка доменної моделі	47
2.7 Програмні компоненти системи.....	55
2.8 Структура розгортання системи.....	58
2.9 Розробка тест-кейсів для тестування системи.....	64
2.10 Розробка інструкції користувача	64
Висновки до розділу.....	73
ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	76

ВСТУП

Впровадження систем електронного документообігу виявляється вигідним для багатьох організацій в Україні, адже вони можуть скористатися перевагами сучасних технологій у документальній діяльності. Однак, навіть тим, хто вже активно використовує автоматизацію документообігу, можливо, доведеться змінити свій вибір і задуматися про підвищення ефективності управління документами. Це зумовлено змінами на ринку, зростанням кількості організацій, які потребують реструктуризації та впровадження інформаційно-комунікаційних технологій. З одного боку, це відкриває нові можливості для ведення бізнесу, з іншого - змушує не відставати від своїх конкурентів.

Кожна організація має свої власні причини для запровадження автоматизованого документообігу: для одних це сприяє підвищенню ефективності організаційно-розпорядчої роботи з документами, для інших — покращенню якості роботи працівників, які постійно працюють з документацією. Проте, рідко хто розглядає це питання комплексно. Різні методи в управлінні документацією обумовлені різними ролями та значеннями документів у роботі підприємства. Це залежить від стилю керівництва, розміру організації, сектору економіки та інших чинників. Так, на одному підприємстві документ може бути ключовим інструментом управління, тоді як на іншому - він слугує засобом комунікації чи навіть продуктом.

Кожна організація розпочинає свою діяльність з вибору підходящої системи з доступних на ринку варіантів. Однак створення автоматизованої системи як окремого робочого продукту не є раціональним рішенням. Тому доцільно розширити її функціональні та технологічні можливості, використання технології блокчейн або інтегрувавши систему управління контентом.

У процесі роботи будь-якої сучасної організації постійно виникає необхідність у створенні та зберіганні значної кількості даних, а також часто відбуваються підписання договорів та обмін повідомленнями між зацікавленими сторонами. Для створення додаткової довіри можна використовувати технологію блокчейн. Збереження інформації про дії сторін дозволить уникнути необхідності залучення третіх сторін, а унікальний цифровий відбиток договору дає можливість сторонам назавжди зберегти правильну версію договору без необхідності в паперових копіях чи посередниках. Важливою також є можливість підтвердження факту підпису документа тим чи іншим співробітником організації, що спрощує процеси узгодження документів всередині компанії.

Ця робота спрямована на розробку системи електронного документообігу з використанням блокчейн технології та аналіз проблем, що виникають при її автоматизації. Для досягнення цієї мети поставлено наступні завдання:

- Проаналізувати існуючі аналоги систем електронного документообігу;
- Розробити архітектуру спроектованої системи;
- Описати процес реалізації системи електронного документообігу;
- Реалізувати інтеграцію з блокчейн технологією;
- Презентувати результати тестування розробленої системи.

РОЗДІЛ 1

ОПИС ОСНОВНИХ ПОНЯТЬ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Щодня ми стикаємося з електронним обміном даними. Це відбувається за допомогою різних засобів, таких як інтернет, комп'ютери, смартфони, смарт-годинники та інші пристрої.

Останніми роками з'явилися нові інструменти, які підвищують ефективність управлінських процесів у компаніях. Одним із таких інструментів є програмне забезпечення для обробки документів. Поняття електронного документа (ЕД), електронного цифрового підпису (ЕЦП) та системи електронного документообігу (СЕД) стали широко вживаними.

1.1 Основні поняття та визначення

Документообіг - це рух документів у установі від моменту їх створення або отримання ззовні до передачі на зберігання в архів. Він охоплює: прийом документів, їх обробку та реєстрацію, контроль виконання, обробку та відправлення вихідної кореспонденції.

Електронний документообіг - означає набір процесів, які включають створення, обробку, редагування, відправлення, приймання, збереження, використання та знищення електронних документів. Ці процеси включають також перевірку цілісності документів та, за необхідності, підтвердження отримання таких документів.

Електронний документ - це документ, представлений у вигляді файлу відповідного формату, створеного за допомогою комп'ютерної обробки інформації та підписаного електронним цифровим підписом (ЕЦП). Такий документ можна зберігати на електронних носіях, копіювати та тиражувати.

Електронний цифровий підпис (ЕЦП) — це електронний аналог особистого підпису, створений за допомогою криптографічного перетворення набору даних. Він слугує засобом захисту інформації, підтверджуючи її цілісність та ідентифікуючи підписувача.

Документ є базовим елементом у системі документообігу. Такі системи забезпечують рух документів всередині підприємства, дозволяють відстежувати процеси, до яких належать ті чи інші документи, і забезпечують функції зберігання самих документів, так і інформацію про виконані зміни. У кожній організації, де використовується система документообігу, документ вважається основним інструментом управлінського процесу. Без використання документів наприклад неможливо видати доручення або ухвалити наказ. Під документообігом в організації розуміють систематичне переміщення документів, які були створені у процесі діяльності відповідними службовцями.

1.1.1 Існуючі види документообігу

Залежно від характеру операцій організацій, у яких відбувається рух (переміщення та облік) документів, виділяють наступні види документообігу.

Фінансовий документообіг включає всі операції з діловодством та дозволяє підприємству управляти різними нормативними документами, такими як договори, законодавчі акти, звіти, кореспонденцію та інше, що використовується в бізнес-процесах.

Управлінський документообіг включає процеси збору, аналізу та синтезу інформації для створення точних, достовірних і зрозумілих управлінських звітів. На основі цих звітів в організації приймаються рішення і контролюється їх вчасне виконання.

Технічний документообіг передбачає організацію координації інформаційних потоків для забезпечення правильної роботи всіх етапів життєвого циклу документації проекту.

Кадровий документообіг включає процеси збору та аналізу інформації, необхідної для обліку персоналу на підприємстві. Це охоплює дані про прийом на роботу та звільнення працівників, відсутності на робочому місці, зміни у посадах, планування відпусток, графіки робочих змін, посадові інструкції та інше.

Архівний документообіг забезпечує систематичну та своєчасну передачу документів для зберігання в архів, а також номенклатурний облік справ під час діловодства. У програмному забезпеченні існують три основні види документообігу, які вимагають автоматизації.

Офісний документообіг підтримує виконання повсякденних та рутинних офісних завдань, обмежених конкретним проектом, таких як підготовка завдань та інструкцій. Керівник приймає рішення про розподіл та виконання цих завдань, визначаючи відповідальних працівників. Електронне сповіщення, яке містить інструкції та інформацію про статус завдання, автоматично надсилається працівникам згідно з установленим порядком.

Спільний документообіг розроблений для ситуацій, коли нестандартні процеси вимагають взаємодії між різними відділами або навіть декількома підприємствами. Приклади спільного документообігу включають створення нової продукції, розробку концепту, проектування.

Адміністративний документообіг об'єднує процеси, які раніше використовували лише паперові документи. Цей документообіг призначений для ефективної обробки звітів та форм, пов'язаних з адміністративними витратами. Замість традиційного заповнення паперових форм, працівник заповнює електронну версію та відправляє її через електронну пошту безпосередньо отримувачу.

1.1.2 Методи організації документообігу

Розрізняють три методи обліку документів:

Електронний - всі бізнес-процеси реєструються в електронних документах без необхідності паперових аналогів. Для підтвердження отримання таких документів і надання їм юридичної сили застосовується ЕЦП, який захищає дані від змін та спотворень. Цей метод зменшує час на створення документа та дозволяє контролювати всі дії з ним.

Паперовий - усі операції на підприємстві документуються на паперових носіях із обов'язковими реквізитами та підписами. Цей метод є затратним як щодо матеріалів, так і часу, потребує ресурсів для зберігання, контролю та створення документів. В даному випадку є досить високий ризик втрати інформації.

Змішаний - створення, узгодження та затвердження документів відбувається на паперових носіях, але їх обмін між структурними підрозділами або працівниками здійснюється через автоматизовану систему. У цьому випадку юридичну силу має лише оригінальний документ на паперовому носії. Цей метод оптимізує структуру переміщення документації на підприємстві.

1.1.3 Призначення електронного документообігу

Ефективна робота працівників є ключовим фактором для якісного обслуговування громадськості. В сучасному світі традиційні методи обробки інформації вже не відповідають потребам часу, тому необхідно мінімізувати витрати часу та ресурсів на виконання завдань, активно впроваджуючи інформаційні технології. ЕД - це сучасний підхід до підвищення якості та швидкості роботи державних органів та підприємств.

Автоматизація процесів тиражування документів, відстеження історії їх обігу на підприємстві та контроль за конфіденційністю даних істотно

скорочує часові витрати працівників, а контроль виконання етапів роботи за допомогою автоматизованої системи підвищує якість роботи діловодів, дозволяє точніше прогнозувати терміни виконання завдань, і розширює можливості в управлінні персоналом.

Системи електронного документообігу часто інтегруються з багатофункціональними сховищами даних. Це дозволяє систематизувати та об'єднувати дані, роблячи створення та аналіз звітів легким і швидким. Крім того, з'являється можливість виявляти закономірності серед збереженої інформації, що сприяє прийняттю виважених і ефективних рішень за допомогою аналізу даних.

Усі ці можливості доступні лише в СЕД, які значно спрощують процеси управління даними порівняно з традиційним паперовим документообігом. Вони забезпечують автоматизацію і централізацію обміну даними, а також агрегацію інформації з відповідних джерел. Ці системи вносять вклад у покращення організаційної культури, роблячи роботу діловодів легшою, продуктивнішою та значущішою. Вони дозволяють ефективно вирішувати різноманітні проблеми спільними зусиллями та прискорюють перехід на вищий рівень обслуговування.

1.2 Структура та принципи електронного документообігу

Основними принципами електронного документообігу є:

- Єдина реєстрація документа — дозволяє однозначно ідентифікувати кожен документ.
- Паралельне виконання операцій — скорочує час, призначений на рух документів, підвищуючи оперативність їх виконання.

– Безперервність руху документів — дає можливість ідентифікувати відповідальну особу за виконання завдання, пов'язаного з документом, на будь-якому етапі його життєвого циклу.

– Виключення дублювання документів — здійснюється через використання єдиної (або узгодженої розподіленої) бази даних документної інформації.

– Ефективна організація пошуку електронних документів — дозволяє знаходити документи за мінімальною кількістю ключових слів.

– Прогресивна система побудови звітів — забезпечує створення звітів за різними статусами документів та атрибутами даних, підвищуючи контроль над рухом документів на різних етапах їх життєвого циклу і спрощуючи прийняття управлінських рішень на основі отриманих даних.

До недавнього часу було прийнятним розділення паперового та електронного документообігу. Зараз це змінилося з появою нового типу документообігу, який поєднує кращі аспекти традиційних та сучасних технологій.

У вітчизняному діловодстві використовується комбінований підхід до документообігу, який інтегрує кращі аспекти традиційних методів з сучасними технологіями створення, обробки та передачі документів. Хоча електронний документообіг має певну перевагу у великих містах та адміністративних центрах, паперовий документообіг все ще залишається розповсюдженим у специфічних областях, зокрема при обробці конфіденційних документів.

Отже, сучасний документообіг на підприємствах є переважно змішаним, з пріоритетом на використання електронних технологій роботи з документами. Це вказує на те, що створення документів здійснюється за допомогою технічних засобів, а їх рух, обробка та зберігання відбуваються в обох форматах: електронному та паперовому. Такий підхід забезпечує

гнучкість у виборі методів роботи з документами в залежності від потреб та вимог підприємства.

Сучасний документообіг підприємства сформований завдяки інтеграції трьох основних потоків інформації: внутрішніх документів, що циркулюють всередині організації, вхідних та вихідних документів. Внутрішній потік включає переміщення документів у межах організації, тоді як зовнішній потік охоплює рух вхідної та вихідної кореспонденції, забезпечуючи обмін інформацією зі сторонніми установами або партнерами.

Внутрішні документи - це ті, які створюються в рамках організації для виконання завдань та задоволення потреб її структурних підрозділів, відповідно до внутрішніх правил і політик, встановлених керівними особами. Ці документи є продуктом внутрішньої діяльності підприємства і обслуговують його оперативні та управлінські потреби.

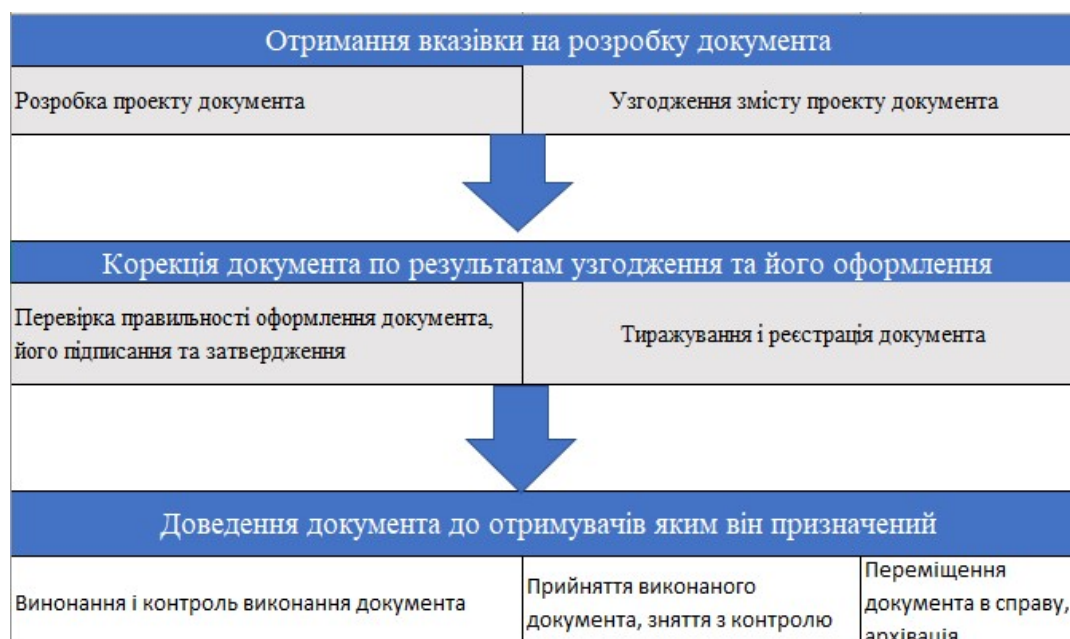


Рисунок 1.1 – Організація роботи з внутрішніми документами

Загальний процес організації роботи з такими документами може бути відображений у певній схемі (рис.1.1), яка демонструє взаємодію між різними

підрозділами та процеси обробки, зберігання і передачі документації в межах організації.

Вхідні документи - це документи, які надходять до організації ззовні через різні канали зв'язку. На рисунку 1.2 наведено приклад такої організації роботи на підприємстві.

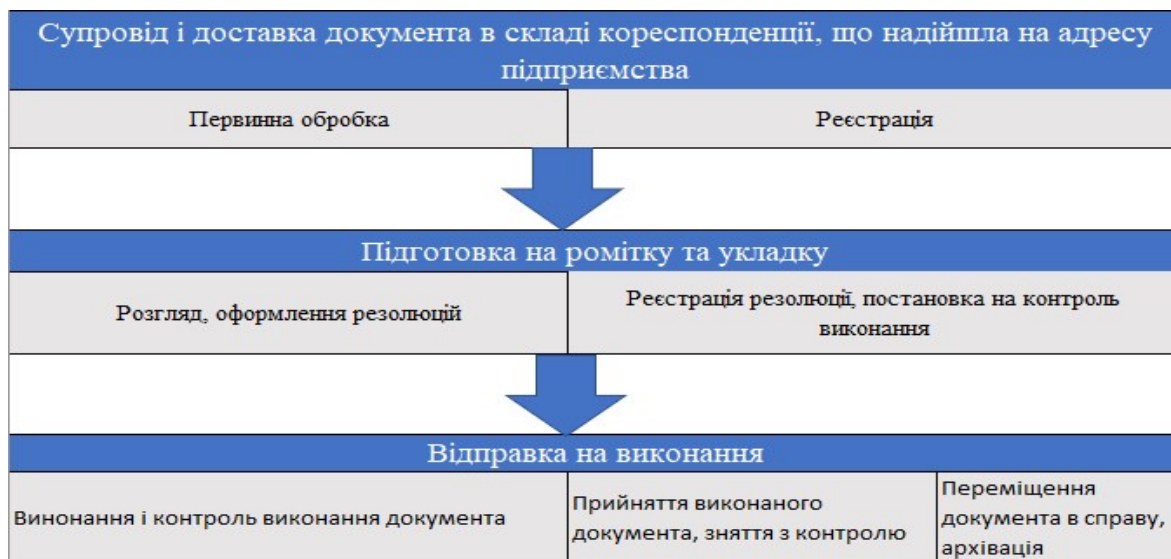


Рисунок 1.2 – Організація роботи з вхідними документами



Рисунок 1.3 – Організація роботи з вихідними документам

Вихідні документи - це документи, створені всередині організації та відправлені до нижчих органів управління з метою виконання наказів вищого рівня. На рисунку 1.3 наведено приклад такої організації роботи підприємства.

1.3 Порівняльний аналіз типів документообігу

1.3.1 Недоліки електронного та традиційного документообігу

Багато малих та середніх підприємств, так само як і великі підприємства, мають відділення, що є територіально розділеними. Управління документообігом у таких відділеннях зазвичай нерозвинене або застосовуються лише окремі ланки управлінської діяльності. Однак основою будь-якого управління є робота з документами, тому організованість цієї роботи є важливою і впливає на якість та швидкість виконання прийнятих рішень.

У сучасну епоху розвитку інформаційних технологій традиційна схема управління документообігом не здатна забезпечити виконання наявних обсягів роботи. Використовуючи класичний паперовий процес діловодства, різні організації стикаються з однаковими проблемами:

- Суперечливість інформації та зниження швидкості передачі документів керівним особам від яких залежить прийняття відповідних рішень.
- Витік конфіденційної інформації до конкурентів або третіх осіб.
- Надмірна витрата часу на кореспонденцію, обробку внутрішніх документів та ознайомлення з ними.
- Досить суттєві втрати часу створення та узгодження документів, що знижує швидкість обробки інформації та уповільнює реакцію на майбутні виклики.

- Зберігання великої кількості документів незрозумілого походження та призначення.
- Висока імовірність втрати важливих документів та інформації.
- Неможливість відтворити історію роботи над документами.
- Перевитрата часу на пошук необхідних документів, формування каталогів і добірок за темами.
- Перевитрата матеріалів для створення копій документів.

В результаті документообіг містить значну кількість зайвих документів та етапів розгляду, рішення часто дублюються, а іноді суперечать одне одному. Це негативно впливає на керівництво, яке не може відстежити попередню та поточну діяльність відділів і виконавців, а також етапи підготовки та затвердження важливих документів.

Для вирішення цих проблем починають впроваджувати нові технології, але часто ці рішення є простими і не відповідають вимогам компанії. Наприклад, спільний доступ до файлів на сервері як механізм спрощення роботи, або використання корпоративної електронної пошти для контролю за надсиланням документів. В дійсності ці підходи працюють лише на початкових етапах розвитку підприємства. Коли управлінські процеси ускладнюються і зростає кількість працівників та підрозділів, такі рішення стають недостатніми, і на підприємстві виникає хаос в управлінсько-інформаційній діяльності.

Рішенням цих проблем є впровадження автоматизованої системи документообігу підприємства. Однак, разом з перевагами впровадження новітніх технологій виникають інші проблеми, серед яких можна виділити наступні:

- Ризик несанкціонованого доступу до конфіденційної інформації.
- Можливість втрати даних через технічні неполадки, некваліфіковані дії персоналу, віруси та кібер-атаки.

- Проблеми нормативного характеру.
- Складність визнання юридичної сили ЕД державними органами.
- Відсутність стандарту роботи з ЕД, який би забезпечував їх юридичну силу.
- Відсутність практик оцінки якості ЕД та їх архівації.

І паперовий й так само електронний спосіб ведення документообігу мають свої переваги та недоліки. Паперові документи менш надійні для довгострокового зберігання інформації, але їх легше сприймати. ЕД мають переваги в спрощеній процедурі копіювання та швидкості передачі до адресата. У сфері управління ключовими вимогами є швидкість, надійність та оперативність документообігу. Тому для ефективності діяльності підприємства часто використовується гібридний підхід, який комбінує як паперові, так і електронні носії документів. Це дозволяє вибрати оптимальний спосіб обробки документів залежно від конкретної задачі та умов, забезпечуючи максимальну ефективність і гнучкість в управлінських процесах.

1.3.2 Переваги електронного документообігу над паперовим

Все більше підприємств визнають переваги електронного документообігу (який наведено на рис. 1.4) над паперовим (який наведено на рис. 1.4).

Використання електронного документообігу має ряд переваг, які можуть значно покращити ефективність роботи працівників порівняно з традиційним паперовим документообігом.

Ось декілька ключових аспектів:

- Швидкість обробки: Електронні системи дозволяють швидше обмінюватися документами між відділами чи навіть організаціями.

Документи можуть бути відправлені та отримані в миттєвий спосіб, що знижує час на їх обробку.



Рисунок 1.4 – Використання електронного документообігу

- Зменшення помилок: Автоматизація допомагає знизити людські помилки, наприклад, введення даних, оскільки інформація вводиться один раз і потім використовується системою автоматично.
- Легший доступ до документів: Електронні документи можуть бути доступні з будь-якого місця, де є доступ до інтернету, що дозволяє працівникам працювати віддалено або з різних офісів без потреби в пересиланні паперових копій.
- Ефективне зберігання: Електронний документообіг значно знижує потребу в фізичному зберіганні документів, що економить простір у офісі та знижує витрати на архівацію.
- Покращена безпека: Електронні документи можуть бути захищені кращими засобами безпеки, такими як шифрування та контроль доступу, що зменшує ризик несанкціонованого доступу або втрати інформації.
- Екологічність: Зменшення використання паперу не тільки економить ресурси, але й знижує вплив на навколишнє середовище, що є важливим аспектом корпоративної відповідальності.



Рисунок 1.5 – Використання паперового документообігу

Такі системи, як електронний документообіг, дозволяють організаціям бути більш гнучкими та адаптивними в сучасному швидкозмінному бізнес-середовищі.

Впровадження електронного документообігу дозволяє зменшити кількість служб, що займаються роботою з документами (кур'єри, канцелярські працівники тощо). На наведеному малюнку показано, як скорочується час окремих етапів обробки документів при переході від паперового до електронного процесу.

За умов електронного документообігу витрати на перебудову документообігу при зміні зовнішніх умов, наприклад, вимог до форми звітності, значно зменшуються.

1.3.3 Підходи для побудови СЕД

Перед інтеграцією системи електронного документообігу необхідно провести ґрунтовну оцінку готовності до покращення якості всіх процесів. Впровадження системи електронного документообігу вимагає наявності таких засобів у компанії:

- Забезпечення кожного працівника компанії персональним комп'ютером, якому необхідний доступ до системи.
- Обчислювальна техніка, яка є достатньо продуктивною для підтримки програмного забезпечення, запланованого до впровадження.
- Готовність керівництва до використання ЕЦП.
- Надійний доступ до мережі.
- Служба адміністрування технічних засобів і систем, а також можливість для відтворення копій паперових документів.

Якщо працівники не матимуть доступу до системи електронного документообігу або технічних засобів, вони не зможуть брати участь у роботі компанії. Такі працівники зможуть працювати лише з паперовими копіями ЕД, але не матимуть можливості аналізу даних без доступу до СЕД.

Підхід до часткового впровадження великих технічних систем є дійсно раціональним і може значно сприяти зменшенню ризиків і витрат. Розглянемо кілька ключових аспектів такого підходу:

- Мінімізація перерв у роботі: Часткове впровадження дозволяє організаціям поступово адаптуватися до нової системи без серйозних перерв у їх звичайній діяльності. Це особливо важливо для критичних операцій, де тривалі перерви можуть мати значні фінансові наслідки.
- Тестування та оптимізація: Починаючи впровадження з одного відділу чи процесу, компанії можуть виявити потенційні проблеми та вирішити їх перед масштабним розгортанням. Це допомагає оптимізувати процеси та технологічні рішення, забезпечуючи гладкіше та більш ефективне впровадження.
- Навчання та адаптація персоналу: Часткове впровадження дає можливість поступово навчати співробітників, що зменшує навантаження на персонал і допомагає засвоїти нові системи більш ефективно. Це також

сприяє зменшенню опору з боку співробітників, які можуть відчувати невпевненість або страх перед новими технологіями.

– Контроль витрат: Здійснення часткового впровадження дозволяє краще контролювати витрати, оскільки інвестиції розподіляються поетапно. Це також дозволяє аналізувати рентабельність інвестицій на кожному етапі перед додатковими вкладеннями.

– Флексибільність у змінах: Під час часткового впровадження, компанії можуть більш гнучко підходити до змін у бізнес-процесах, адаптуючи та налаштовуючи систему під потреби конкретного відділу або процесу, що може виявитись вкрай корисним перед ширшим впровадженням.

Застосування новітніх технологій сприяє швидкому переходу з традиційного паперового документообігу до роботи з документами в електронному форматі. Однак ключовими факторами є підготовка та готовність співробітників до змін у робочих процесах. При адаптації до нової системи документообігу, співробітники часто стикаються з проблемами, зокрема з підписанням електронних документів. Це вимагає від них розуміння та володіння технологією електронного цифрового підпису (ЕЦП).

На сьогоднішній день існує проблема "інформаційної нерівності", коли люди без доступу до інтернету відчують затримку у процесі інтеграції в інформаційне суспільство. Через це системи електронного документообігу мають включати функції, які дозволяють обробляти паперові документи і конвертувати їх у електронний формат. Відсутність таких функцій може бути витлумачена як порушення прав певних груп громадян.

1.3.4 Підготовка документів

У більшості випадків реальна робота з документами здійснюється в електронній формі. Такий тип роботи стає звичним для працівників, оскільки відмінності в автоматизованому документообігу незначні. Метою такого

рішення є збільшення використання сучасних засобів для покращення організації роботи.

Важливою перевагою ЕД є використання ЕЦП для ідентифікації достовірності інформації. Важливим критерієм у виборі системи автоматизованого документообігу є наявність спільного сховища для документів, яке об'єднує офіційний, неофіційний та архівний документообіг.

Усі ці можливості дозволяють підвищити продуктивність праці у підготовці та обробці документів, а також дають можливість використовувати напрацювання інших працівників, аналітичні матеріали, звіти та дослідження інформації.

1.3.5 Склад СЕД

Електронна система управління документами призначена для автоматизації та оптимізації процесів життєвого циклу документів, що сприяє підвищенню ефективності управління бізнес-процесами в організації та полегшує взаємодію між співробітниками. Головною частиною електронного документообігу є документ, зазвичай представлений як неструктурований текстовий файл. Системи електронного документообігу включають підсистему автоматизації ділових процесів, призначену для колективної обробки документів, функціональність для підтримки ділових процесів, а також архів ЕД.

Важливо розрізнити концепції діловодства та документообігу. Діловодство визначає набір правил для обігу документів, які можуть бути імплементовані у системі електронного документообігу. Розробники систем електронного документообігу зазвичай враховують кілька категорій:

Універсальні СЕД:

- Мають дещо обмежений функціонал.
- Не передбачають налаштування під специфічні потреби підприємства.
- Вимагають мінімальних витрат на впровадження.

- Мають невисоку вартість.
- Поставляються з ліцензією.

Індивідуально розроблені СЕД:

- Найбільше відповідають вимогам клієнта.
- Потребують значного часу для впровадження.
- Висока вартість розробки та впровадження.
- Необхідність навчання співробітників та додаткового технічного забезпечення.

Комплексні СЕД:

- Мають модульну структуру.
- Повністю відповідають потребам користувачів.
- Мають невеликі витрати на введення в експлуатацію.
- Надають повні права на продукт.
- Легкі у використанні.

СЕД за своїм функціоналом поділяються на наступні категорії:

Електронний архів - система документообігу, яка зосереджена на ефективному зберіганні та пошуку інформації. Такі системи часто мають повнотекстовий та розширений пошук серед доступних інструментів і використовують ефективні технології зберігання даних у базах даних. Вони підтримують інтеграції з різними системами.

Системи підтримки робочого процесу - використовуються для організації та спрощення переміщення документів за встановленими маршрутами або правилами. На кожному етапі процесу можна вносити зміни до документів. Ці системи дозволяють організувати роботу, де всі етапи відомі до початку процесу.

Комплексні системи документообігу - інтегрують функції електронного архівування та систем підтримки робочих процесів. Вони часто застосовуються в державному управлінні та великих компаніях, де необхідна гнучка система для регулювання правил обігу документів та їхньої структури.

Такі системи включають підсистеми для адміністрування, колективної роботи над документами, підтримують численні інтеграції, забезпечують ефективний пошук, легко масштабуються та можуть бути впроваджені поетапно. Головна мета цих систем — забезпечити ефективну спільну роботу персоналу з мінімізацією часових та зусильних витрат.

1.3.6 Структура СЕД

На рисунку 1.6 наведено структуру складових частин СЕД.

Сервер додатків - серверні компоненти, що виконують бізнес-логіку системи.

Сервер бази даних - компоненти для зберігання та доступу до даних.

Клієнтське робоче місце - клієнтський додаток системи, який забезпечує користувацький інтерфейс і можливості управління.

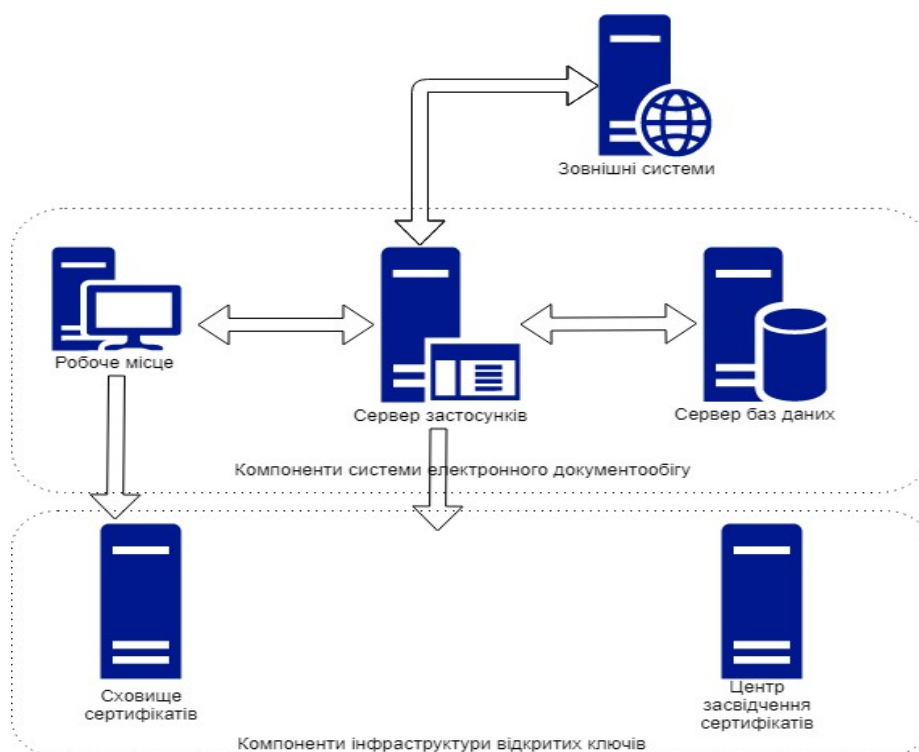


Рисунок 1.6 – Структура системи електронного документообігу

API забезпечує взаємодію інших систем з компонентами СЕД. Це підвищує гнучкість системи та надає можливість інтеграції зі сторонніми

системами, які можуть забезпечити доступ до функцій, раніше недоступних користувачам.

API реалізований у вигляді REST веб-сервісу та у якості протоколу передачі даних використовує HTTP.

Виділяють основні проблеми, властиві будь-яким організаціям під час впровадження і підтримки СЕД:

- Небажання перенавчатися та низький рівень освіченості персоналу, страх перед прозорістю діяльності для керівництва.
- Небажання використовувати комп'ютер і працювати з електронними документами.
- Часті зміни в структурі організації та зміни в бізнес-процесах.
- Необхідність взаємодії із зовнішніми паперовими документами.

1.4 Огляд існуючих СЕД

У сучасному українському ринку доступно багато СЕД, які відповідають вимогам та завданням підприємств, але розроблені з використанням різних програмних вимог. Нижче представлено огляд декількох найпоширеніших СЕД.

1.4.1 Docs Fusion (розроблена компанією Hummingbird).

Це одна з найпопулярніших систем у світі також давно представлена і в Україні. Docs Fusion підтримує колективну роботу і може використовуватися як на великих підприємствах із чисельністю більше тисячі працівників, так і в середньому бізнесі. Система призначена для організацій, які інтенсивно працюють з документами.

1.4.2 Documentum

СЕД для великих компаній, впровадження якої в Україні почалося відносно недавно. Вона пропонує платформу для створення розподілених архівів, підтримки стандартів діяльності, управління проектами в групах,

організації діловодства та контент-менеджменту корпоративних інтернет-порталів.

1.4.3 FossDoc

Дана СЕД є рішенням, розробленим на базі платформи FossLook, яке призначене для створення електронних архівів документів, автоматизації бізнес-процесів та організації корпоративного документообігу. Система надає можливість вирішувати різні завдання за допомогою спеціалізованих модулів. Вона підтримує налаштування з урахуванням потреб кожного підприємства. Архітектура системи FossDoc є розподіленою, з використанням багатьох модулів з можливістю окремого налаштування. На рисунку 1.7 наведено архітектуру даної СЕД на якій видно, що СУБД та сервер додатків розташовані на окремих фізичних носіях з зв'язком для обробки інформації.

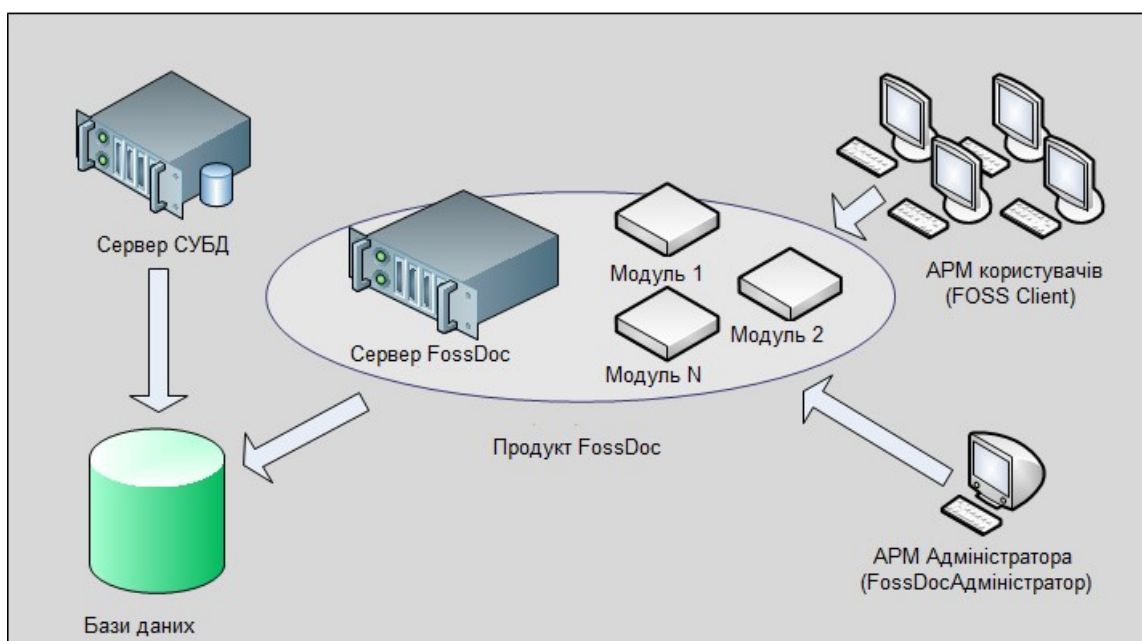


Рисунок 1.7 – Структура побудови СЕД FossDoc

1.5 Властивості та вимоги до СЕД

Щоб забезпечити ефективну діяльність сучасних підприємств, системи електронного документообігу повинні відповідати таким вимогам:

Масштабованість - система повинна підтримувати необхідну кількість користувачів та мати можливість збільшувати потужність завдяки апаратному забезпеченню.

Розподіленість - архітектура системи повинна дозволяти її розподіл між віддаленими підрозділами та забезпечувати різні комунікаційні канали.

Модульність - система повинна складатися з окремих модулів, об'єднаних між собою, для гнучкого впровадження.

Відкритість - система повинна мати відкриті програмні інтерфейси для інтеграції зі сторонніми сервісами.

Надійність - система має забезпечувати стабільну та безперебійну роботу, незважаючи на можливі збої.

Захищеність - система повинна мати можливість гнучко керувати доступом до різних документів, пошти та медіафайлів.

Доступність - система повинна бути доступною через веб-додатки, настільні додатки та інші клієнти.

СЕД повинна відповідати таким вимогам:

- Забезпечити надійне зберігання електронних документів.
- Надати можливості для роботи з документами, включаючи створення, редагування, публікацію, забезпечення конфіденційності та зберігання в архіві.
- Підтримувати обробку різних типів документів і пов'язаних з ними даних.
- Забезпечувати категоризацію для зручного пошуку документів.
- Виконувати частковий і повнотекстовий пошук документів.
- Забезпечувати розподіл доступу на основі ролей у системі згідно з організаційною структурою.
- Підтримувати контроль і відстеження подій у системі з можливістю адміністрування.
- Підтримувати віддалений доступ до системи.

Сучасні СЕД повинні забезпечувати:

- Використання кластерних баз даних для надійної роботи системи.
- Підтримку територіально розділених організацій.
- Застосування сучасних та надійних алгоритмів шифрування для передачі та зберігання інформації.
- Можливість застосування ЕЦП.

Вимоги до архітектури СЕД:

- Наявність сервера додатків.
- Підтримка доступу до системи через браузер.
- Масштабованість системи.

Інші додаткові вимоги до інтеграції:

- Інтеграція з онлайн інструментами обробки інформації.
- Інтеграція з поштовими сервісами.
- Наявність API.
- Адаптація користувацького інтерфейсу під конкретні завдання.
- Можливість розширення системи за допомогою спеціалізованих компонентів.

1.6 Постановка завдання

Потрібно створити багатофункціональну систему електронного обігу документів, яка спростить та пришвидшить роботу з ними. Система має відповідати таким критеріям:

- У системі слід впровадити розмежування прав доступу користувачів з двома ролями: адміністратор і користувач.
- Доступ до системи повинен надаватися лише користувачам, яких зареєстрував адміністратор.
- Необхідно забезпечити кросплатформенність додатку.

– Додаток не повинен бути прив'язаний до конкретної спеціалізації підприємства, повинна бути можливість налаштування системи під потреби будь-якої організації адміністратором.

– Адміністратор повинен мати змогу додавати, видаляти та редагувати користувачів, групи, каталоги, а також видаляти файли.

– Для кожного документу мають зберігатися метадані.

– Користувачі повинні мати можливість додавати та видаляти файли, встановлювати терміни зберігання файлів, додавати коментарі до файлів та відправляти оповіщення на електронну пошту іншим користувачам.

– Дизайн системи має бути ненав'язливим.

Система повинна бути зручною у використанні та мати інтуїтивно зрозумілий інтерфейс.

Висновки до розділу:

Вірний вибір СЕД є одним з ключових завдань для багатьох вітчизняних підприємств та організацій сьогодні. Особливо це є важливим саме для нових компаній, які ще не сформували свої управлінські правила і технології. Щоб забезпечити порядок руху документів в організації, необхідно гарантувати збереження та швидкий пошук інформації та відправлення до користувача у визначені терміни без втрат.

СЕД створені для виконання таких процесів, як створення та управління великими обсягами документів, їх поширення мережею, управління доступом до даних та контроль використання документів.

Сьогодні більшість підприємств прагнуть впроваджувати програмні комплекси та системи, які не тільки формують та контролюють документи, але й мають широкий спектр можливостей для управління поширенням інформації в рамках всієї організації чи мережі та підтримують багато бізнес-процесів.

РОЗДІЛ 2

АНАЛІЗ МОЖЛИВОСТЕЙ ТЕХНОЛОГІЇ БЛОКЧЕЙН, РОРОБКА ТА РЕАЛІЗАЦІЯ СЕД

2.1 Застосування технології blockchain при побудові СЕД

Недосконалість сучасних СЕД привертає все більше уваги до використання блокчейн технологій. Експерти зазначають, що потенційно блокчейн може бути адаптований для здійснення будь-яких операцій (фінансових, матеріальних і нематеріальних). А саме головне, що тип blockchain-сервісу, кількість учасників та їхнє географічне розташування не мають значення.

Особливо варто підкреслити, що загальносвітовою тенденцією щодо використання блокчейн технологій є ретельний аналіз усіх ризиків та можливих негативних наслідків. Великі компанії та організації в основному заявляють про дослідження, розробку або тестування цієї технології. Зокрема, у 2016 році у Швеції розпочалося масштабне дослідження, щоб з'ясувати можливості блокчейну для переведення земельного реєстру на цю технологію. Передбачається, що кожен об'єкт реєстрації отримає "блокчейн паспорт" з технічними характеристиками.

Впровадження цієї технології забезпечує надійну синхронізацію даних, що запобігає їх підміні через зовнішнє втручання, гарантує прозорість та дає можливість здійснювати контроль над системою.

Платформу blockchain можна розглянути як розподілену базу даних без централізованого управління чи нагляду, з можливістю спільного користування. Blockchain сприяє здійсненню різноманітних транзакцій, веденню обліку та зберіганню даних, і може застосовуватися в багатьох сферах життя людини, таких як нерухомість, страхування, логістика, державні реєстри, судові реєстри тощо.

Blockchain вперше був використаний у 2009 році для створення криптовалюти біткоїн. Сьогодні blockchain активно розвивається у різних сферах і країнах. У Китаї blockchain використовується в національному страховому фонді населення та для впровадження технології у систему "розумних міст", яка планується до реалізації. Також на основі blockchain вже існують стартапи в медицині, сфері захисту авторських прав, патентів, розробляються системи ідентифікації, браузері, системи зберігання даних, месенджери, соціальні мережі.

Смарт-контракти, які працюють на базі blockchain, все частіше стають темою обговорення, оскільки вони значно спрощують використання ЕЦП для договорів. Розумні контракти вперше з'явилися в мережі blockchain Ethereum. Blockchain із використанням смарт-контрактів дозволяє обходитися без посередництва третьої сторони, яка зазвичай гарантує умови укладеного договору. Код смарт-контракту самостійно визначає, які дії потрібно виконати для збереження конфіденційності інформації та виконання умов договору на основі заданих умов. Кожен учасник процесу може перевірити актуальність угоди у будь-який момент.

Технологію вперше випробували у 2016 році на платформі Wave, де була здійснена угода на 100 тисяч доларів між ірландським підприємством та сейшельською компанією щодо експорту молочних продуктів. Стандартно така угода займає мінімум тиждень, а за допомогою blockchain вона була завершена за близько чотири години.

2.2 Принцип роботи та архітектура технології blockchain

Блокчейн є розподіленою базою даних, що зберігає інформацію про кожну транзакцію, яка відбувається в системі. Дані в системі представлені у вигляді послідовності блоків (звідси і назва — блокчейн), які містять записи про транзакції. Ці дані не можна підробити, оскільки кожен новий запис взаємно підтверджує існуючі ланцюжки. Для підробки даних потрібно

змінювати інформацію у всіх інших блоках. У системі блокчейн актуальні дані про записи зберігаються у всіх її учасників і автоматично оновлюються при внесенні нових змін. Така структура забезпечує розгалуженість і прозорість транзакцій, що є ключовими перевагами blockchain.

Нові блоки у блокчейн-базі формуються постійно, кожен з них включає групу впорядкованих записів про транзакції за останній період, а також заголовок, що забезпечує їх інтеграцію в ланцюг. Приклад можна побачити на рисунку 2.1.



Рисунок 2.1 - Схематичне представлення блоку у blockchain

Транзакції в blockchain - це будь-які дії, які користувачі виконують у мережі, такі як відправка коштів, реєстрація прав власності, покупка товарів тощо. Коли користувач формує транзакцію, вона надсилається до пулу пам'яті, де очікує на додавання в один з блоків, щоб бути підтвердженою.

Коли блок формується, він автоматично перевіряється іншими учасниками мережі, і якщо всі погоджуються, записується в кінець ланцюжка блоків. Після цього внесення змін до блоку стає неможливим. Окрім нової

інформації, кожен блок зберігає (у зашифрованому вигляді) також дані про попередні блоки (див. рисунок 2.2).

База даних оновлюється на всіх комп'ютерах, підключених до системи, а майнери (валідатори) починають формування наступного блоку.



Рисунок 2.2 – Реалізація зв'язку між елементами блоку

Ключові характеристики технології блокчейн включають незмінність записів, високий рівень безпеки та захисту, відкритість і прозорість транзакцій, а також децентралізацію та розподіленість системи.

Будь-яка важлива інформація, що стосується різних аспектів життя, десь зберігається. Дані про покупку автомобіля, квартири, новий кредит чи реєстрацію шлюбу зберігаються у державних чи приватних реєстрах, які зазвичай мають централізовану систему. Це часто призводить до втрати інформації або небажаного використання даних, а також надає можливість зловмисникам змінювати інформацію. Блокчейн пропонує інший підхід до вирішення цих проблем. Його суть полягає в тому, що блокчейн використовує базу даних, розподілену фізично та логічно на тисячах комп'ютерів по всьому світу. Тому вірогідність виводу з ладу такої системи надзвичайно мала. Поки існує хоча б один вузол, блокчейн може функціонувати.

Як уже згадувалося, централізовану базу даних можна скомпрометувати та внести до неї зміни. У випадку з блокчейном це набагато складніше. Зламати один блок і змінити в ньому інформацію немає сенсу, оскільки доведеться змінювати всі блоки на всіх комп'ютерах у мережі, а для цього потрібні величезні обчислювальні потужності. Нові блоки включають зашифровану інформацію про попередні блоки, тому будь-яка спроба втручання негайно виявляється іншими учасниками мережі. Крім того, фальсифікації ускладнюються завдяки використанню потужних алгоритмів шифрування, що включають хеш-функції та цифрові підписи.

В концепції асиметричного шифрування – для підпису документу застосовують відкритий та закритий ключ. Відкритий ключ потрібен для перевірки підпису, закритий - для його створення і має залишатися секретним. Цифрові підписи забезпечують доступ користувачів до певної інформації. Хеш-функція може здатися просто послідовністю випадкових чисел та букв, але вона відіграє ключову роль у забезпеченні незмінності записів у блокчейні. Робота блокчейн ґрунтується на математичних обчисленнях, а не на умовній довірі користувачів.

Важливо також відзначити, що в нашому житті ми завжди стикаємося з посередниками - фінансові операції здійснюються через банки, платіжні системи та обмінники, а документи затверджуються у нотаріусів. Не рідкість, коли гроші можуть не дійти до адресата, оскільки банк може заблокувати транзакцію для перевірки. Підробка документів також залишається поширеною проблемою.

Таким чином, хоча ми часто не можемо повністю довіряти різним посередникам і вимушені користуватися їхніми послугами, що несе ризики, блокчейн пропонує альтернативу у вигляді безпосереднього обміну даними між сторонами. У системі блокчейн правильність транзакцій перевіряють безпосередньо її учасники або автоматизовані смарт-контракти, що передбачено системою.

Учасників мережі блокчейн поділяють на два типи: користувачі, майнери, валідатори. Звичайні користувачі створюють у мережі нові записи-транзакції. Інші транзакції ігноруються і не вважаються дійсними, доки не будуть додані до одного з наступних блоків. Скористатися певним записом у блокчейні може лише власник ключа, який відкриває до нього доступ. Щоб стати майнером, досить виділити обчислювальні потужності свого комп'ютера для створення нових блоків. Для підключення до мережі використовується спеціальне програмне забезпечення. В деяких системах замість традиційного майнінгу за алгоритмом Proof-of-Work (PoW) застосовуються альтернативні протоколи, такі як Proof-of-Stake. У цьому випадку для формування блоку необхідно володіти певною кількістю криптовалюти на рахунку.

На основі технології блокчейн можуть бути розроблені різноманітні типи систем. Існують публічні міжнародні блокчейн-системи, відкриті для участі будь-якої особи, яка бажає стати користувачем або майнером. Окрім цього, існують приватні блокчейн-мережі, які створюються і контролюються певними організаціями. Для доступу до таких мереж необхідно задовольнити спеціальні умови, встановлені їх засновниками.

Blockchain - універсальна технологія, яка може застосовуватися в різних країнах та аспектах життя громадян, що відкриває великі можливості для розвитку. Крім таких властивостей, як відкритість, безпека і захищеність, блокчейн також знижує витрати на роботу з транзакціями та скорочує час укладення угод між сторонами з кількох днів до кількох годин. Це дозволяє підприємствам і організаціям уникати зайвих витрат. Однак недоліком є проблема масштабованості системи. Сьогодні блокчейн не здатний забезпечити достатню кількість транзакцій за короткий час. Щодня збільшується і розмір бази даних, яка має бути збережена на комп'ютерах у мережі. Іншою важливою проблемою є навантаження на електричні мережі, особливо в системах, що працюють за алгоритмом PoW із-за навантаження обчисленням хеш-сум.

Наразі в Україні на блокчейн вже переведено державний земельний кадастр. На цій технології базується перевірка виписок з реєстру. Також планується переведення державного реєстру речових прав на нерухоме майно на блокчейн.

2.3 Функціональний опис системи

Користувацька модель описує функції, які система буде виконувати, і те, як буде відбуватись взаємодія з користувачем.

Згідно з технологією Microsoft Solution Framework, процес проектування стартує з ретельного аналізу потенційних користувачів. Цей аналіз дозволяє визначити різні типи користувачів системи та їхні робочі функції.

Потім створюється набір сценаріїв використання, кожен з яких розбивається на послідовність конкретних дій, що називаються прикладами використання.

Web-додаток «BlockDoc» передбачає наявність двох типів користувачів у системі електронного документообігу:

- 1) Користувач – кожен працівник компанії.
- 2) Адміністратор – визначена особа, що має спеціальні повноваження в системі.

Спільними функціями для обох типів користувачів є такі:

- Авторизація користувача: після введення правильного логіну та пароля, користувач отримує доступ до системи і може здійснювати певні дії відповідно до своїх прав. Як логін використовується електронна пошта, оскільки це зручно і не вводить користувача в оману.
- Перегляд інформації про файл: користувач може побачити такі дані про файл – його назву, дату створення, автора, розмір та термін зберігання.
- Видалення файлу: під час перегляду файлу користувач має можливість видалити його. При цьому разом з файлом видаляються і всі його попередні версії. Видаляти файл можуть лише його власник або адміністратор.

- Завантаження файлу: користувач має можливість завантажити файл, доступний для перегляду. Для цього необхідно обрати папку для його збереження.

- Перегляд доступу до файлу: користувачі і адміністратор з доступом до файлу можуть переглядати списки користувачів і груп, яким цей файл доступний.

- Перегляд завдань до файлу: обравши певний файл, користувач може бачити все, що з ним пов'язано (хто додав, виконавець, статус виконання завдання).

- Завантаження версії файлу: користувач може зберегти версію файлу, доступну для перегляду, обравши місце для збереження.

- Видалення версій файлу: адміністратор або будь-який користувач з правами доступу до файлу може видаляти версії файлу.

- Перегляд користувачів групи: користувач може бачити списки користувачів, які входять до поточної групи.

- Перегляд списку користувачів: доступний список всіх користувачів системи з вказанням ПІБ, посади та електронної адреси.

- Перегляд списку груп: користувач може бачити перелік груп, що існують в системі.

Для користувачів з правами адміністратора передбачені такі дії:

- Додавання користувача: адміністратор може додати нового користувача увівши всі відомості про нього (ПІБ, посада, email, логін та пароль).

- Видалення користувача: адміністратор може помітити користувача як видаленого, що прибирає його зі списків.

- Відновлення користувача: адміністратор має можливість відновити користувача, який раніше був зареєстрований у системі.

- Редагування інформації про користувача: адміністратор може змінювати такі дані користувача, як ПІБ, посада, електронна адреса, логін та пароль.
 - Перегляд груп користувача: адміністратор має можливість бачити групи, створені конкретним учасником.
 - Додавання завдання: адміністратор може створити нове завдання, яке користувачі зможуть призначати іншим користувачам для виконання.
 - Видалення завдання: якщо завдання втратило актуальність адміністратор може видалити його зі списку. У такому разі завдання буде зберігатися тільки для файлів, які вже є в системі.
 - Перегляд списку завдань: адміністратор може переглядати список завдань, доданих ним, які будуть використовуватися користувачами для створення нових завдань до файлу.
 - Створення групи: створення групи користувачів виконує адміністратор з обов'язковим вказуванням її назви.
 - Видалення групи: видалити будь-яку групу може адміністратор.
 - Додавання користувача до групи: адміністратор може додати до групи зареєстрованих у системі користувачів.
 - Видалення користувача з групи: адміністратор може виключити користувача з певної групи.
- Для користувачів, авторизованих як звичайні співробітники установи, сценарії використання, відмінні від адміністратора, включають такі:
- Додавання файлу: щоб додати файл, користувач повинен ввести його назву, коментар, термін зберігання та вибрати файл для завантаження. Його основні параметри створюються автоматично і файл доступний усім користувачам системи за замовчуванням.
 - Перегляд доданих файлів: переглядати завантажені файли може сам користувач у вигляді таблиці зі списком, що містить назву, дату додавання, розмір, коментарі та термін зберігання кожного файлу.

– Перегляд завдань користувача: показуються всі завдання до файлів, призначені для виконання конкретним користувачем. Для кожного завдання можна побачити такі дані: текст завдання, файл, до якого воно додано, хто додав завдання, та статус виконання.

– Перегляд доданих завдань: відображаються всі завдання, які користувач призначив іншим. Для кожного завдання можна побачити такі дані: текст завдання, файл, до якого воно додано, виконавець і статус виконання.

– Позначка про виконання завдання: користувач, який додав завдання, або виконавець можуть відзначити виконання завдання після завершення робіт. При позначці "завдання виконано" користувач, який призначив завдання, отримує повідомлення на електронну пошту про завершення, а у списку завдань з'являється позначка про виконання.

2.4 Вибір та обґрунтування компонент

При виборі технології для розробки системи було враховано такі вимоги:

Система повинна бути кросплатформенною, тому обрана платформа .Net Core з використанням мови С#. А її розробка у вигляді веб-додатку забезпечує такі переваги:

– модель клієнт-сервер дозволяє централізувати та консолідувати інформацію;

– веб-додатки не потребують ручної інсталяції, вони запускаються та встановлюються автоматично;

– для роботи користувачам потрібен лише браузер, що дозволяє працювати віддалено через мережу.

Тому було обрано технологію ASP.NET Core, а для проектування використано модель MVC (Model-View-Controller).

Функції, які описані в сервісах, реалізовані у вигляді SQL-запитів до бази даних та звернень до файлової системи. Вони також включають реалізацію необхідних алгоритмів.

Для реалізації представлень була використана технологія Razor, яка дозволяє поєднувати HTML-розмітку веб-сторінок з фрагментами коду на C#.

Коли клієнт робить запит до веб-сторінки, веб-сервер перенаправляє цей запит до відповідного контролера. Контролер викликає сервіси і потім генерує відповідь, яку відправляє клієнтові. Було розроблено CSHTML-сторінки, кожна з яких відповідає за певний функціонал у системі та доступна через браузер.

Для зручності користувачів інтерфейс був розроблений з використанням технології AJAX. AJAX – це підхід до створення інтерфейсів веб-додатків, який полягає у фоновому обміні даними між браузером і веб-сервером. Завдяки цьому веб-сторінка не перезавантажується (при оновленні даних), що робить веб-додатки швидшими та зручнішими. Для взаємодії з базою даних використовувалася технологія ODBC – програмний інтерфейс, який дозволяє системі управління базами даних підключатися до великої кількості SQL баз даних і отримувати доступ до інших джерел табличних даних, таких як електронні таблиці або неструктуровані файли.

При розробці системи було застосовано сучасні підходи до розробки програмних комплексів, що забезпечують масштабованість, простоту оновлення та впровадження функціоналу. Процес розробки супроводжувався створенням UML діаграм, таких як діаграми компонентів, ER-діаграми.

Відкрита реалізація платформи Ethereum – Nethereum обрана для взаємодії з блокчейном, а також мову програмування Solidity для створення смарт-контрактів.

Ethereum – платформа призначена для створення децентралізованих онлайн-сервісів на основі blockchain, які працюють за допомогою смарт-

контрактів. Вона функціонує як єдина децентралізована віртуальна машина. Ethereum дозволяє користувачам легко розробляти децентралізовані додатки за допомогою технології blockchain. Децентралізована програма виглядає для користувача як звичайна, проте має важливу особливість: незалежність від будь-якого централізованого посередника.

Blockchain Ethereum можна також охарактеризувати як blockchain з інтегрованою мовою програмування або як глобальну віртуальну машину, що працює на основі консенсусу.

Віртуальна машина підтримує два типи акаунтів, які можуть взаємодіяти між собою:

- Індивідуальний акаунт, захищений приватним ключем.
- Контракт, який має свій власний код і управляється цим кодом.

За замовчуванням середовище Ethereum неактивне, але кожен користувач може його активувати, шляхом надсилання транзакції зі свого облікового запису. Якщо транзакція адресована контракту, контракт автоматично активується і запускає свій код. Код може читати і записувати дані у власне внутрішнє сховище (база даних, що відображає 32-байтні ключі), зберігати та відправляти повідомлення послідовно іншим контрактам. Після завершення виконання коду контракту та повного завершення всіх підвикликів, середовище виконання зупиняється до наступної транзакції.

Зазвичай контракти виконують чотири основні функції:

- 1) Сховище невеликого обсягу даних, корисних для інших контрактів.
- 2) Обліковий засіб зі складною політикою доступу, відомий як контракт переадресації.
- 3) Управління поточним контрактом або зв'язками між кількома користувачами, наприклад, фінансовий договір з кількома посередниками.
- 4) Надання функцій іншим контрактам, фактично виступаючи бібліотекою для програмного забезпечення.

Контракти взаємодіють один з одним через дії, відомі як "виклик" або "відправлення повідомлення". Повідомлення є об'єктом, що містить певну кількість ефіру (внутрішньої віртуальної валюти), дані у вигляді масиву байт, а також адреси відправника і одержувача. Коли контракт отримує повідомлення, він може опціонально повернути деякі дані, які початковий відправник може відразу використовувати. Таким чином, "відправлення повідомлення" діє подібно до виклику функції.

Структура blockchain Ethereum наведена на рисунку 2.3

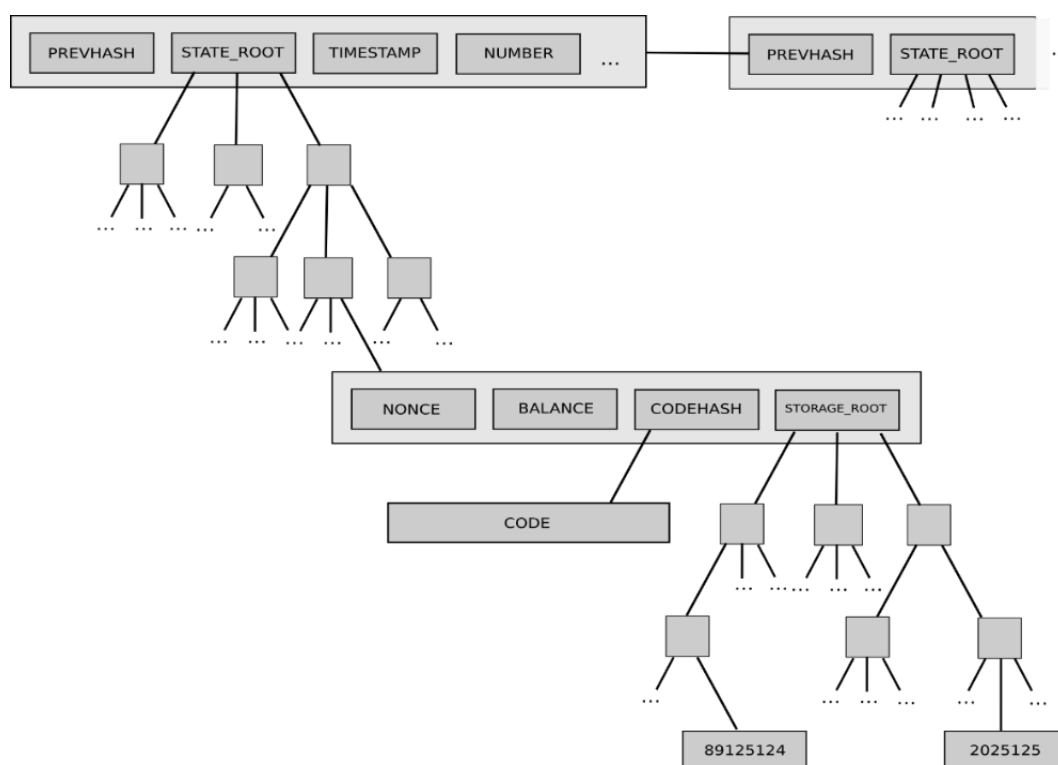


Рисунок 2.3 – Структура blockchain Ethereum

Blockchain Ethereum ("книга") – це децентралізована, широко реплікована база даних, що зберігає поточний стан усіх облікових записів. Blockchain використовує структуру даних, відому як дерево Patricia, це спеціалізований тип дерева Меркле, який служить загальним сховищем ключів. Дерево Patricia має "кореневий хеш" і його вміст не може бути змінений без зміни цього кореневого хеша. Для кожного облікового запису

дерево зберігає чотири компоненти: `account_nonce` (кількість транзакцій, надісланих з облікового запису), `ether_balance` (баланс рахунку), `code_hash` (хеш код), `storage_root` (корінь іншого дерева Patricia).

Кожну хвилину майнер створює новий блок (концепція майнінгу в Ethereum аналогічна Bitcoin). Цей блок містить список транзакцій, які відбулися з моменту створення останнього блоку. Майнер також отримує винагороду у вигляді ефіру за створення блоку.

Для створення смарт-контрактів була обрана мова Solidity – об'єктноорієнтована мова високого рівня. Смарт-контракти – це програми, для регулювання поведінки облікових записів у blockchain Ethereum. На розробку Solidity вплинули мови C++, Python та JavaScript. За її допомогою можна створювати смарт-контракти для різних цілей, таких як голосування, краудфандинг, сліпі аукціони та блокчейн-гаманці. При створенні смарт-контрактів рекомендується використовувати останню доступну версію Solidity через швидкий розвиток цієї мови.

Приклад контракту Solidity:

```
contract SimpleStorage
{
    unit storedData;
    function set(unit x)
    {
        storedData = x;
    }
    function get() constant returns (unit retVal)
    {
        return storedData;
    }
} unit storedData
```

2.5 Приклади варіантів використання

У загальному випадку система містить 21 варіант використання, але для опису особливостей поведінки СЕД на основі технології blockchain у роботі

описано 8 ключових варіантів використання, які є найбільш важливими для системи в розрізі Адміністратора та Користувача.

Нижче наведено детальні пояснення для кожного з варіантів використання:

Варіант використання 1. Вхід - Sign in (табл.2.1).

Таблиця 2.1 – Потік подій для варіанту 1

№	Дійова особа	Крок
1	Користувач/Адміністратор	Переходить на сторінку входу до системи
2	Користувач/Адміністратор	Вводить логін та пароль
3	Система	Перевіряє введені логін та пароль користувача
4	Система	Підтверджує вхід
5	Користувач/Адміністратор	Переходить на головну сторінку

Варіант використання 2. Додавання файлу - Add file (табл.2.2).

Таблиця 2.2 – Потік подій для варіанту 2

№	Дійова особа	Крок
1	Користувач	Переходить до сторінки завантаження файлу
2	Користувач	Натискає кнопку «Upload file»
3	Користувач	Вибирає файл у файловій системі та натискає відправити
4	Система	Завантажує файл та визначає його метадані
5	Система	Записує файл на диск для подальшого зберігання
6	Система	Додає запис про файл у блокчейн
7	Система	Додає запис про файл до бази даних
8	Система	Сповіщує користувача про успішне додання файлу

Варіант використання 3. Підписання документа - Sign Document (табл.2.3).

Таблиця 2.3 – Потік подій для варіанту 3

№	Дійова особа	Крок
1	Користувач	Переходить до сторінки перегляду документів
2	Користувач	Натискає кнопку «Sign file»
3	Система	Генерує унікальний підпис на основі даних про користувача системи та документа
4	Система	Записує інформацію про підпис до блокчейн
6	Система	Сповіщує користувача про успішність/помилку підпису за допомогою вікна інформації

Варіант використання 4. Перевірка документа - Verify Document (табл.2.4).

Таблиця 2.4 – Потік подій для варіанту 4

№	Дійова особа	Крок
1	Користувач	Переходить до сторінки перевірки файлу
2	Користувач	Натискає кнопку «Upload file»
3	Користувач	Вибирає файл у файловій системі та натискає відправити
4	Система	Завантажує файл та визначає його хеш
5	Система	Робить запит у блокчейн за хешем файлу
6	Система	Перевіряє чи співпадає наявний запис із збереженим у блокчейн
7	Система	Сповіщує користувача про успішність перевірки

Варіант використання 5. Створення завдання - Create Task (табл.2.5).

Таблиця 2.5 – Потік подій для варіанту 5

№	Дійова особа	Крок
1	Користувач	Переходить до сторінки перегляду документів
2	Користувач	Обирає потрібний документ
3	Користувач	Натискає створити завдання
4	Користувач	Обирає виконавця та вводить текст завдання
5	Система	Створює нове завдання, пов'язане з документом, з даними введеними користувачем
6	Система	Сповіщує користувача про успішність/помилку створення завдання за допомогою вікна інформації

Варіант використання 6. Зміна дозволу до документа - Change Document Permission (табл.2.6).

Таблиця 2.6 – Потік подій для варіанту 6

№	Дійова особа	Крок
1	Користувач	Переходить до сторінки перегляду документів
2	Користувач	Обирає потрібний документ
3	Користувач	Натискає кнопку «Change permission»
4	Користувач	Обирає користувачів яким доступний документ
5	Система	Зберігає дані про доступ у БД

Варіант використання 7. Створення категорії документів - Create Document Category (табл.2.7).

Таблиця 2.7 – Потік подій для варіанту 7

№	Дійова особа	Крок
1	Користувач	Переходить до сторінки створення категорії
2	Користувач	Вводить назву категорії
3	Користувач	Натискає кнопку «Add Category»
4	Система	Зберігає дані про нову категорію
5	Система	Сповіщує користувача про успішність операції

Варіант використання 8. Створення користувача - Create User (табл.2.8).

Таблиця 2.8 – Потік подій для варіанту 8

№	Дійова особа	Крок
1	Адміністратор	Переходить до сторінки створення користувача
2	Адміністратор	Вводить основну інформацію про користувача
3	Адміністратор	Натискає кнопку «Create»
4	Система	Перевіряє чи не був користувач з заданими даними зареєстрований в системі раніше та додає нового користувача до БД
5	Система	Сповіщує адміністратора про успішність створення користувача

2.6 Розробка доменної моделі

Розробка моделі домену є ключовою частиною процесу створення додатка. Ця фаза включає аналіз характеристик сфери застосування програми та формування моделі на основі об'єктів.

У системі функціонують дві ключові сутності: Користувач (User) та Документ (Document). Вивчення доменної моделі показало, що кожен користувач має можливість управляти категоріями документів, регулювати

доступ до них, взаємодіяти з групами користувачів та виконувати завдання, пов'язані з документами. Для реалізації цих функцій були створені наступні сутності: Категорія (Category), Завдання (Task), Завдання Документу (DocumentTask), Група (Group), Група Користувачів (UserGroup).

Сутність "Category" містить інформацію про категорію, до якої відносяться документи в системі. Користувачі мають можливість конфігурувати індивідуальні категорії для своїх документів.

Сутність "Task" містить інформацію про завдання, призначене для певного документа у системі, включаючи статус його виконання та користувача, який відповідає за це завдання. "DocumentTask" є допоміжною сутністю, яка створює зв'язок між завданнями та документами.

Сутність "Group" призначена для опису групи користувачів у системі, які можуть бути організовані за певними логічними критеріями для розділення ролей в організації, де використовується система. "UserGroup" є допоміжною сутністю, що забезпечує зв'язок між користувачами та групами.

Сутність "DocumentBase" створена для використання смарт-контрактів Ethereum, яка призначена для зберігання хешу документа та його ідентифікатора в мережі блокчейн.

Для оптимізації взаємодії з базою даних було обрано ORM технологію – Entity Framework, а також застосовано паттерн Repository. Основна ідея паттерна репозиторій – створення абстрактного шару між доступом до даних та бізнес-логікою. Такий підхід дозволяє ізолювати додаток від змін у базі даних і полегшує тестування компонентів, що взаємодіють з даними.

Entity Framework центрується на концепції сутностей, які відображають комплекси даних, асоційованих з конкретними об'єктами. Відповідно, робота здійснюється не з таблицями бази даних, а з об'єктами, що сприяє інтуїтивному проектуванню та розробці.

Кожна сутність, подібно до об'єктів у реальному світі, володіє різноманітними атрибутами. Наприклад, описуючи сутність, яка представляє людину, можна виокремити такі атрибути як ім'я, прізвище, зріст, вік та вага. Атрибути можуть бути не тільки простими даними, але й представляти більш складні структури. У кожній сутності також можуть бути одна чи декілька ключових властивостей, що дозволяють відрізнити її від інших сутностей та унікально ідентифікувати. Такі властивості називаються ключами.

Сутності можуть взаємодіяти між собою за допомогою асоціативних зв'язків таких типів: "один до одного", "багато до багатьох" та "один до багатьох", подібно до зв'язків, які встановлюються через зовнішні ключі в традиційних базах даних.

Однією з ключових особливостей Entity Framework є можливість використання запитів LINQ для маніпулювання даними. Завдяки LINQ можна не лише витягувати певні записи з бази даних, а й здійснювати складні операції з об'єктами, що зв'язані різними асоціативними відносинами.

Відношення між SQL Server, EF, DbContext та додатком наведено на рисунку 2.4.

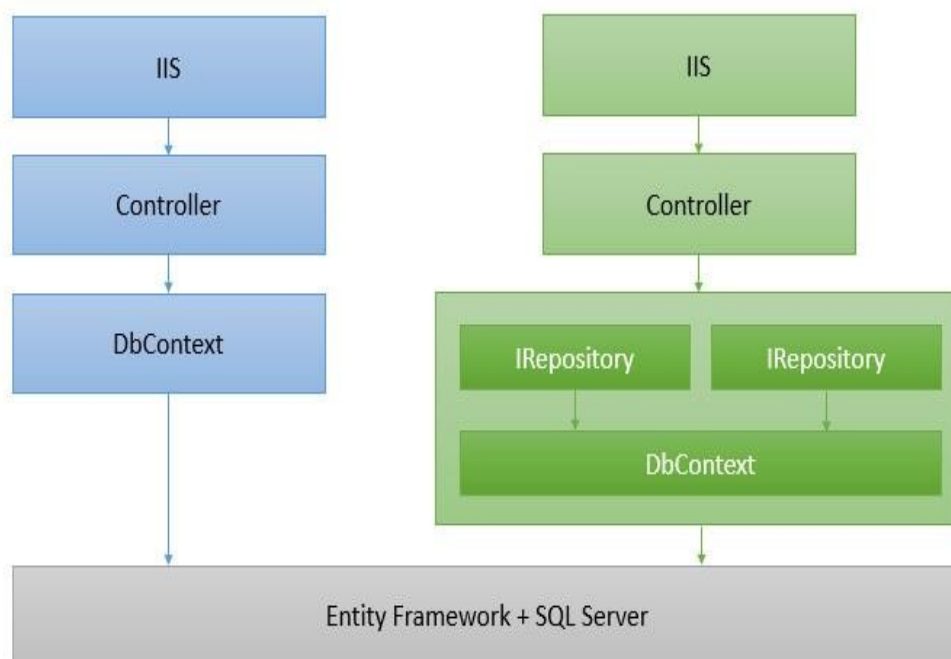


Рисунок 2.4 – Відношення між компонентами

Entity Framework також передбачає створення класів репозиторіїв для кожного згенерованого класу сутності, а також відповідних інтерфейсів. Ці інтерфейси служать для абстракції реалізації класів, що дозволяє контролерам взаємодіяти з різними джерелами даних через уніфікований інтерфейс, незалежно від того, чи використовується Entity Framework або інше сховище даних.

Далі приведено опис таблиць бази даних системи.

Таблиця 2.9 - Користувач - User

Назва	Тип даних	Опис даних
UserId	int	Ідентифікатор користувача
PasswordHash	Varchar	Пароль користувача в системі
Name	Varchar	Ім'я користувача
Surname	Varchar	Прізвище користувача
Email	Varchar	Адреса електронної пошти
Active	int	Статус користувача

Дана таблиця використовується для зберігання інформації про користувачів та їхній статус у системі.

Таблиця 2.10 – Роль користувача - Role

Назва	Тип даних	Опис даних
RoleId	int	Ідентифікатор ролі
Name	Varchar	Назва ролі
Active	int	Статус ролі

Дана таблиця призначена для встановлення зв'язків між користувачами та їх ролями в системі, що дозволяє керувати доступом користувачів до різних функцій системи.

Таблиця 2.11 – Дозвіл - Permission

Назва	Тип даних	Опис даних
PermissionId	int	Ідентифікатор дозволу
Name	Varchar	Назва дозволу

Дана таблиця використовується для визначення доступних дозволів у системі, дозволяючи керувати доступом до різних ресурсів та функціональностей.

Таблиця 2.12 - Дозвіл-Роль - RolePermission

Назва	Тип даних	Опис даних
PermissionId	int	Ідентифікатор дозволу
RoleId	int	Ідентифікатор ролі

Дана таблиця використовується для визначення відносин між дозволами та ролями у системі, забезпечуючи зв'язок, який вказує, які дозволи доступні конкретним ролям користувачів.

Таблиця 2.13 – Користувач-Група - UserGroup

Назва	Тип даних	Опис даних
UserId	int	Ідентифікатор користувача
GroupId	int	Ідентифікатор групи

Дана таблиця використовується для зв'язків між групами та користувачами в системі, дозволяючи асоціювати конкретних користувачів з певними групами документів або групами користувачів.

Таблиця 2.14 – Група - Group

Назва	Тип даних	Опис даних
GroupId	int	Ідентифікатор групи
Name	Varchar	Назва групи
Description	Varchar	Опис групи
DateCreated	DateTime	Дата створення

Дана таблиця призначена для зберігання інформації про групи, до яких належать користувачі системи, дозволяючи організувати користувачів за певними категоріями або командами.

Таблиця 2.15 - Документ - Document

Назва	Тип даних	Опис даних
DocumentId	int	Ідентифікатор документа
UserId	int	Ідентифікатор користувача власника документа
Comment	Varchar	Коментар до документа
DateCreated	DateTime	Дата створення
CategoryId	int	Категорія документа
FilePath	Varchar	Шлях до файлу документа у файловій системі
FileNane	Varchar	Ім'я документа
Locked	Bit	Статус документа
Hash	Varchar	Хеш документа

Дана таблиця призначена для зберігання основної інформації про документи, які користувачі завантажують у систему. Вона включає такі дані,

як час створення документа, коментарі до нього, категорія, до якої документ належить, назва файлу, а також шлях до його розташування на диску.

Таблиця 2.16 – Завдання - Task

Назва	Тип даних	Опис даних
TaskId	int	Ідентифікатор завдання
Description	Varchar	Опис завдання
AssignedUserId	int	Користувач відповідальний за виконання завдання
CreatedByUserId	int	Користувач який створив завдання
DateCompleted	Datetime	Дата завершення завдання
DateCreated	Datetime	Дата створення завдання

Дана таблиця використовується для зберігання інформації про завдання, які призначені користувачам системи. Кожне завдання має поля для ідентифікації виконавця та власника, які забезпечують зв'язок з таблицею "Користувач". Це дозволяє відстежувати відповідальності за виконання та власність завдань у системі.

Таблиця 2.17 - Документ-Завдання - DocumentTask

Назва	Тип даних	Опис даних
DocumentId	int	Ідентифікатор документа
TaskId	int	Ідентифікатор завдання

Таблиця 2.18 – Тег Tag

Назва	Тип даних	Опис даних
TagId	int	Ідентифікатор тегу
Name	Varchar	Назва тегу
Description	Varchar	Опис тегу

Дана таблиця призначена для зберігання переліку маркерів, які можуть бути прикріплені до документів у системі. Ці теги допомагають визначити належність документа до певного логічного типу або категорії, полегшуючи сортування та пошук документів за специфічними характеристиками.

Таблиця 2.19 – Документ-Тег - DocumentTag

Назва	Тип даних	Опис даних
DocumentId	int	Ідентифікатор документа
TagId	int	Ідентифікатор тегу

Таблиця 2.20 – Архів - DocumentArchive

Назва	Тип даних	Опис даних
DocumentId	int	Ідентифікатор документа
UserId	int	Ідентифікатор користувача власника документа
Comment	Varchar	Коментар до документа
DateCreated	DateTime	Дата створення
CategoryId	int	Категорія документа
FilePath	Varchar	Шлях до файлу документа у файловій системі
FileName	Varchar	Ім'я документа
Hash	Varchar	Хеш документа

Дана таблиця призначена для зберігання документів, які були відібрані для переміщення в архів. Це дозволяє системі відокремлювати поточні та активні документи від тих, що більше не використовуються на регулярній основі, але потребують збереження для історичної важливості чи відповідно до вимог зберігання даних.

Таблиця 2.21 – Категорія - Category

Назва	Тип даних	Опис даних
CategoryId	int	Ідентифікатор категорії
Description	Varchar	Опис категорії
Name	Varchar	Назва категорії
DateCreated	Datetime	Дата створення категорії

Дана таблиця використовується для визначення категорій, до яких можуть бути віднесені документи користувачів у системі. Кожен користувач має можливість створювати та керувати множиною категорій для організації своїх документів, що дозволяє ефективніше управляти і сортувати свою інформацію.

2.7 Програмні компоненти системи

В системі було розроблено такі компоненти, як:

"AuthService", який забезпечує аутентифікацію користувачів під час їх входу в систему. Цей компонент є універсальним і призначений для використання як звичайними користувачами, так і адміністраторами, забезпечуючи безпечний доступ до функціоналу системи.

Компонент "UserService" відповідає за управління даними, пов'язаними з користувачами та їхніми групами. Цей сервіс дозволяє виконувати такі дії, як перегляд груп, отримання інформації про користувачів, а також управління іншими аспектами, що стосуються користувацьких даних. Це включає зміни в профілях користувачів, редагування груп та керування ролями та дозволами користувачів у системі).

Компонент "CategoryService" в системі відповідає за управління категоріями файлів та документів. Цей сервіс забезпечує функції для створення, редагування, видалення категорій, а також можливість перегляду

існуючих категорій. Він дозволяє користувачам ефективно класифікувати документи та файли згідно з встановленими параметрами, що сприяє кращій організації і легшому доступу до необхідної інформації в системі.

Компонент "DocumentService" в системі забезпечує комплексне управління документами. Цей сервіс включає в себе функції створення, зміни та видалення документів, а також керування завданнями, що пов'язані з цими документами. Він дозволяє користувачам організувати документообіг в системі, забезпечуючи контроль над важливою документацією, її актуалізацію та адекватне видалення застарілих файлів.

Контролер "AuthController" створений для обробки даних, що стосуються сторінки аутентифікації в системі. Цей контролер керує входом користувачів, реєстрацією нових акаунтів, а також обробляє запити на відновлення паролів. Він відіграє ключову роль у забезпеченні безпеки системи, контролюючи доступ до ресурсів та перевіряючи ідентифікацію користувачів перед наданням доступу до важливих функцій.

Контролер "CategoryController" в системі відповідає за обробку даних, пов'язаних з категоріями файлів та документів.

Контролер "DocumentController" відповідає за обробку даних, пов'язаних з переглядом списку документів у системі. Основна функція цього контролера — забезпечення інтерфейсу для користувачів, які можуть переглядати, сортувати та вибирати документи згідно з різними параметрами. Цей контролер допомагає користувачам ефективно управляти доступом до документів, їх організацією та моніторингом змін.

"DocumentController" є ключовим компонентом системи, який сприяє високій організації та ефективності документообігу в організації.

Контролер "HomeController" відповідає за управління головною сторінкою системи. Цей контролер є точкою входу для користувачів після аутентифікації, забезпечуючи доступ до основних функцій та ресурсів, які система пропонує. Зазвичай "HomeController" містить методи, які керують

відображенням головної дашборди або панелі управління, де користувачі можуть отримати швидкий огляд активностей та актуальних даних.

"HomeController" є важливим компонентом для орієнтації користувачів у системі, допомагаючи їм легко навігувати до необхідних функцій або отримувати базову інформацію про ресурси, якими вони можуть скористатися.

Контролер "UserController" призначений для управління даними користувачів у системі. Цей контролер забезпечує функціональність для перегляду списку користувачів, створення нових користувачів та редагування інформації про існуючих користувачів. "UserController" є ключовим компонентом для управління доступом до системи і контролем за користувачами, гарантуючи, що правильні люди мають доступ до відповідних ресурсів.

У системі впроваджено сервіс для інтеграції з технологією блокчейн під назвою Blockchain Document Storage Manager. Цей сервіс використовується для взаємодії з блокчейн мережею, зокрема, для створення нових блоків та зчитування даних з існуючих блоків. Він також забезпечує функції перевірки версій документів та їх підписів, що дозволяє забезпечити надійність та автентичність документації в системі.

Основні можливості Blockchain Document Storage Manager:

- Створення нових блоків: Інтеграція з блокчейн дозволяє системі реєструвати транзакції та зміни в документах як окремі блоки в ланцюгу, забезпечуючи їх незмінність та прозорість.
- Зчитування даних з існуючих блоків: Це дозволяє отримати історію змін та перегляди документів, що є критично важливим для верифікації та аудиту.

– Перевірка версій документів: Система може перевіряти чи документ був змінений або модифікований після його підписання, використовуючи хеш-суми збережені в блокчейні.

– Автентифікація підписів: Валідація електронних підписів документів для перевірки їх легітимності та дійсності.

Код смарт-контракту для роботи з Ethereum:

```
contract DocumentStorage {
    struct Document {
        bytes documentHash;
        int32 documentId;
    }
    mapping(int32 => Document) documents;    int32[] documentsById;
    function set(Document memory document) public
    {
        if(documents[document.documentId].documentHash.length != 0)
        {
            return;
        }
        if(document.documentHash.length != 0)
        {
            documentsById.push(document.documentId);
        }
        documents[document.documentId].documentId =
        document.documentId;
        documents[document.documentId].documentHash =
        document.documentHash;
    }
    function get(int32 id) public view returns (Document memory SavedDocument) {
        SavedDocument = documents[id];
        return SavedDocument;
    }
    function getAllIds() public view returns (int32[]
memory)
    {
        return documentsById;
    }
}
```

2.8 Структура розгортання системи

Розглянемо приклад розгортання системи на локальному сервері. Першим кроком у цьому процесі є інсталяція Internet Information Services (IIS) на сервері. Для цього ефективно використовується Інсталятор Веб-платформи (Web Platform Installer, WPI), який спрощує процес встановлення, автоматично підбираючи та інсталюючи необхідне серверне програмне забезпечення. WPI забезпечує інсталяцію відповідно до рекомендованих налаштувань, включаючи всі критичні компоненти для IIS та Web Deploy. Якщо деякі

КОМПОНЕНТИ ВЖЕ встановлені, WPI додатково встановить лише ті компоненти, яких бракує.

На ілюстраціях 2.5 - 2.6 представлено перелік компонентів, які необхідно встановити для повноцінної роботи системи на сервері. Цей підхід забезпечує не тільки правильну підготовку серверного середовища, але й оптимізує подальші кроки розгортання і налаштування системи, гарантуючи її стабільну та ефективну роботу:



Рисунок 2.5 – Встановлення IIS

Щоб вибрати потрібні елементи, скористайтесь рядком пошуку та введіть назву компонента.

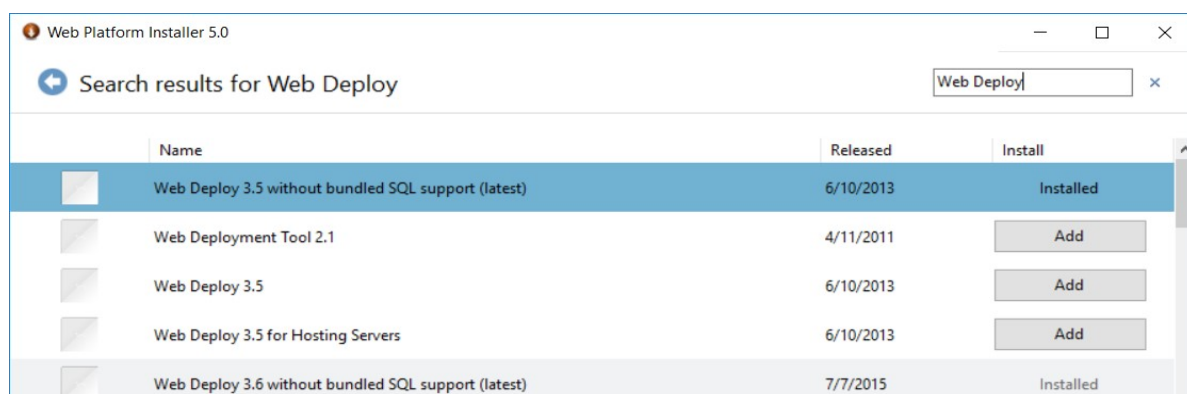


Рисунок 2.6 – Встановлення Web Deploy

На наступному етапі необхідно активувати `www.Services` та IIS Management Console за допомогою інструмента панелі управління – Windows Features. Налаштування представлено на рис. 2.7.

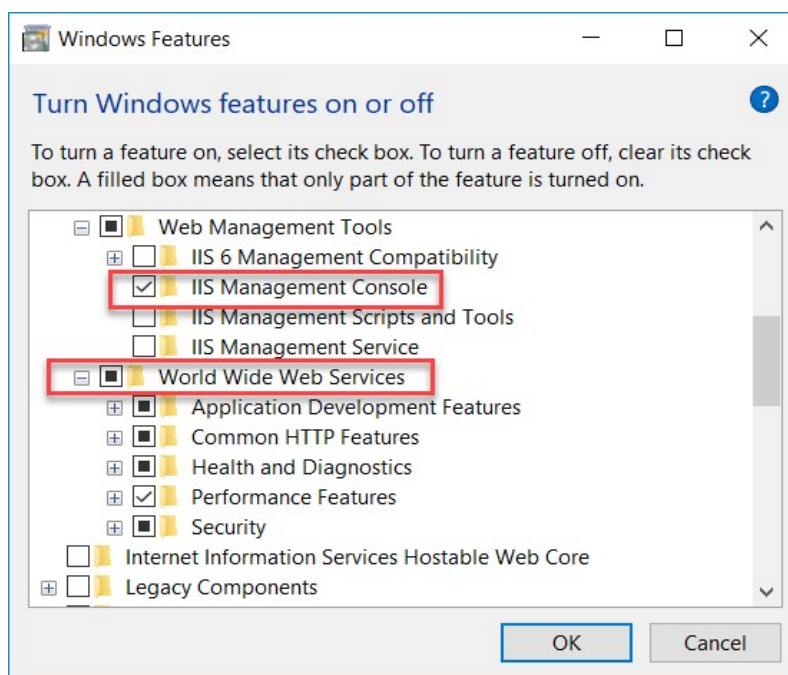


Рисунок 2.7 – Налаштування Windows Features

Потім необхідно створити пул додатків, використовуючи встановлений IIS Manager. Для цього слід відкрити менеджер веб-сервера та перейти до меню "Application Pools". За замовчуванням у списку пулів вже буде додано три нових пули, що належать до системних (рис.2.8). А на рис. 2.9 наведено вікно налаштувань пулу додатків

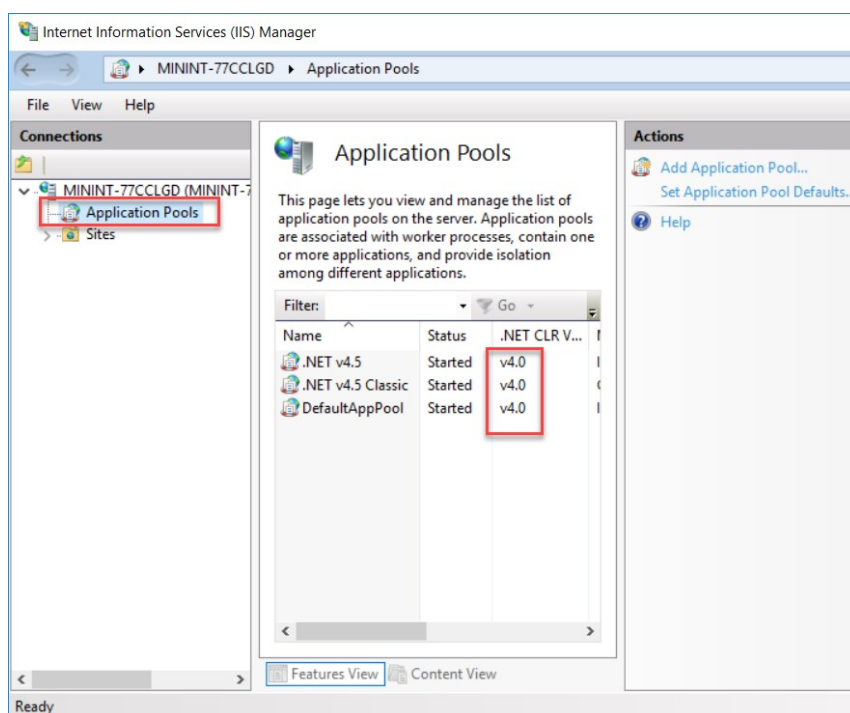


Рисунок 2.8 – Приклад налаштувань IIS

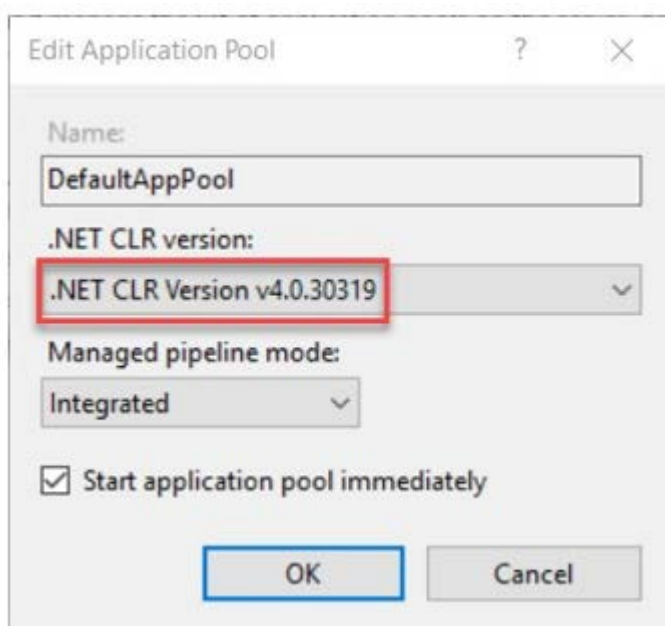


Рисунок 2.9 – Приклад налаштувань пулу додатків

Для створення бази даних необхідно використати SQL Management Studio і виконати команду "Create new SQL database" (рис. 2.10).

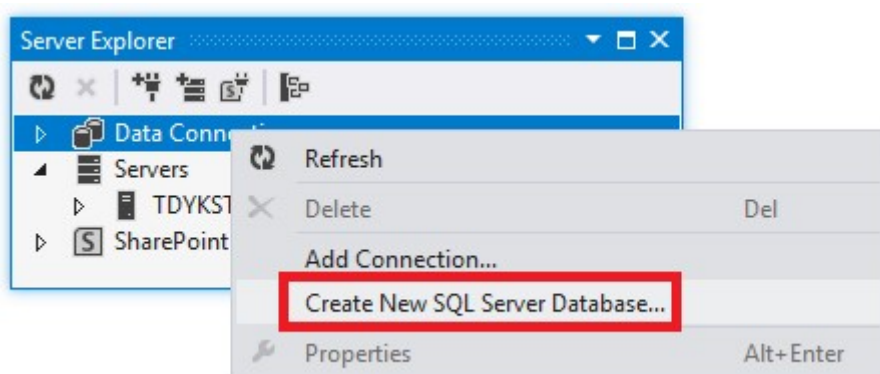


Рисунок 2.10 – Створення нової бази даних

За замовчуванням до бази даних стандартний пул додатків не має доступу. Для надання відповідних прав необхідно виконати наступну команду в створеній базі даних:

```

IF NOT EXISTS (SELECT name FROM sys.server_principals WHERE name = 'IIS
APPPool\DefaultAppPool')
BEGIN
    CREATE LOGIN [IIS APPPOOL\DefaultAppPool]
        FROM WINDOWS WITH DEFAULT_DATABASE=[master],
        DEFAULT_LANGUAGE=[us_english]
END
GO
CREATE USER [SystemUser]
    FOR LOGIN [IIS APPPOOL\DefaultAppPool]
GO
EXEC sp_addrolemember 'db_owner', 'SystemUser'
GO

```

Третім важливим етапом є розгортання блокчейн мережі. Для підвищення надійності системи необхідно розгорнути кілька вузлів мережі. Розглянемо приклад створення одного з таких вузлів. Основним інструментом для керування мережею блокчейн Ethereum є Geth.

Після запуску, Geth автоматично шукає інші вузли однорангової блокчейн мережі, використовуючи протокол виявлення. Цей протокол дозволяє вузлам обмінюватися інформацією про учасників мережі.

Щоб почати роботу, Geth використовує набір вузлів, зазначених у налаштуваннях. Ось приклад команди для налаштування вузла:

```

geth --bootnodes
"enode://pubkey1@ip1:port1
enode://pubkey2@ip2:port2
enode://pubkey3@ip3:port3"

```

Щоб перевірити кількість підключених вузлів, можна скористатися спеціальним модулем `net`, який доступний через інтерактивну консоль Geth. Цей модуль має кілька атрибутів, що дозволяють отримати інформацію про підключені вузли, такі як кількість вузлів і їхній статус:

```

> net.listening true
> net.peerCount
4|

```


Geth також підтримує статичні вузли, що забезпечує постійне підключення до певних вузлів системи. Це особливо корисно, якщо потрібно мати стабільне підключення до конкретних вузлів. Такі вузли можуть бути налаштовані у файлі налаштувань `static-nodes.json`:

```
[
  "enode://f4642fa65af50cfdea8fa7414a5def7bb7991478b768e296f5e4a54e8b995de102e0ceae2e826f293c481b5325f89be6d207b003382e18a8ecba66fbaf6416c0@33.4.2.1:30303",
  {"enode://pubkey@ip:port" }
]
```

Для запуску декількох вузлів у локальній мережі необхідно врахувати кілька важливих аспектів:

- Окрема директорія для кожного екземпляру: Кожен вузол повинен мати власну директорію для зберігання даних. Це налаштовується за допомогою параметра `--datadir`.

- Окремі порти для Ethereum та RPC: Кожен вузол має використовувати різні порти для взаємодії з мережею Ethereum та для RPC-з'єднань. Це налаштовується за допомогою параметрів `--port` та `--rpcport`.

- Знання про інші вузли в кластері: Якщо вузли об'єднані в кластер, вони повинні мати інформацію один про одного, щоб встановити з'єднання.

Для запуску першого вузла використовується команда:

```
geth --datadir="/tmp/eth/" -verbosity 6 --ipcdisable --port 30301 --rpcport 8101 console 2>> /tmp/eth.log.txt
```

Для інших вузлів можуть бути використані аналогічні команди. Розгорнута мережа Ethereum є приватною мережею, де вузли не підключені до вузлів глобальної блокчейн-мережі. Це означає, що мережа є ізольованою та захищеною від зовнішніх підключень, за умови дотримання при розгортанні серверів всіх правил безпеки.

2.9 Розробка тест-кейсів для тестування системи

Додаток тестувався локально, і тестами були покриті найбільш важливі компоненти системи, включаючи клієнтський додаток та код інтеграції з blockchain. Інтеграційне тестування дозволило перевірити коректність логіки контролерів, а також оцінити вірність повертаючихся даних користувачу через шар доступу до даних, а також роботу сервісів, відповідальних за бізнес-логіку.

Основна перевага інтеграційних тестів при локальному тестуванні полягає у відсутності мережових затримок, оскільки всі тести виконуються на одному локальному сервері. Це дає можливість швидко реалізовувати та проводити тести перед впровадженням функціоналу в продуктивне середовище.

Було проведено ручне тестування за допомогою тест-кейсів, зазначених нижче.

- Тест кейс 1. Вхід користувача до системи документообігу
- Тест кейс 2. Створення нового користувача адміністратором
- Тест кейс 3. Завантаження нового документа до системи
- Тест кейс 4. Підписання документа

2.10 Розробка інструкції користувача

Для початку роботи з системою, кожен користувач повинен пройти процедуру авторизації, заповнити відповідну форму (рис. 2.11).

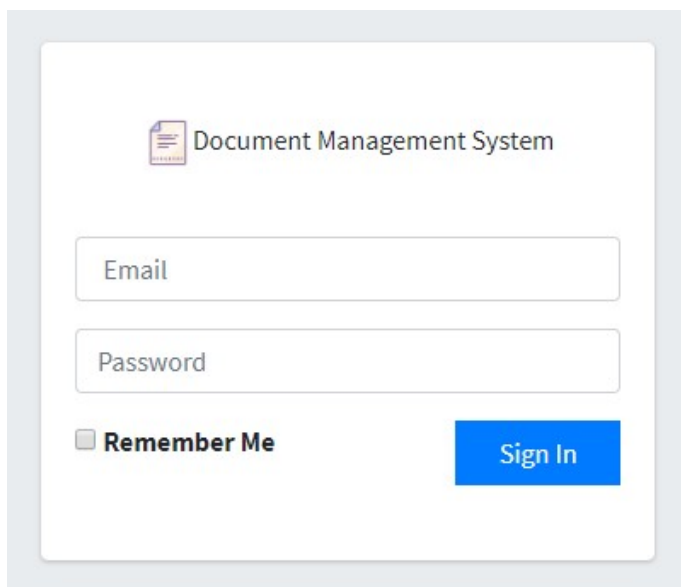
The image shows a login form for a 'Document Management System'. At the top, there is a document icon and the text 'Document Management System'. Below this are two input fields: 'Email' and 'Password'. Under the 'Email' field is a checkbox labeled 'Remember Me'. To the right of the 'Remember Me' checkbox is a blue button with the text 'Sign In'.

Рисунок 2.11 – Вікно авторизації

При помилці авторизації користувач отримає наступне повідомлення (рис. 2.12).

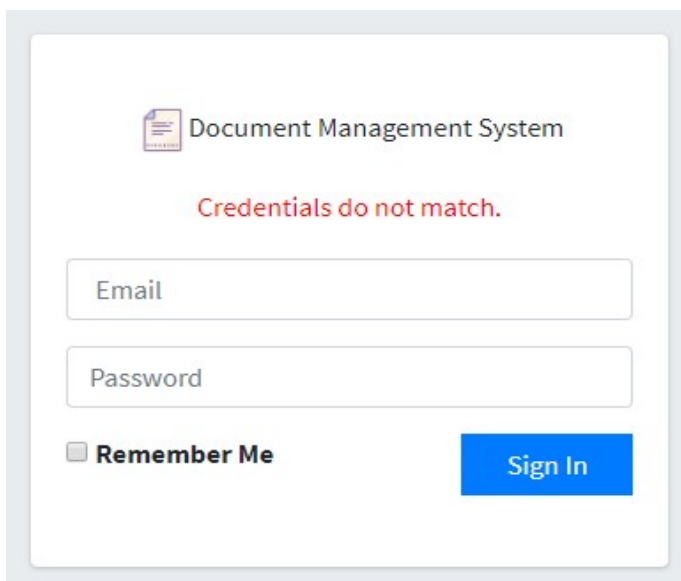
The image shows the same login form as in Figure 2.11, but with an error message. The error message 'Credentials do not match.' is displayed in red text above the 'Email' and 'Password' input fields. The 'Remember Me' checkbox and the 'Sign In' button are still visible at the bottom.

Рисунок 2.12 – Вікно помилки авторизації

У системі виділено дві ролі: користувач та адміністратор. Після успішного входу користувач потрапляє на головну сторінку – дашборд, де відображається інформація про кількість користувачів та інша основна інформація (рис. 2.13).

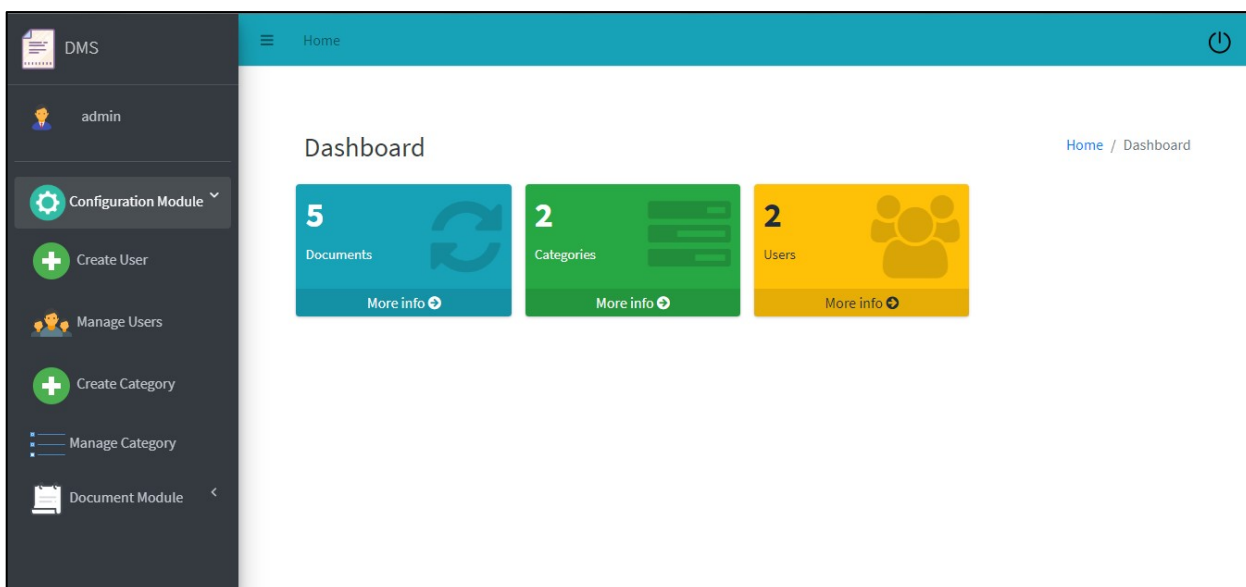


Рисунок 2.13 – Вікно початкової сторінки

Для зареєстрованих користувачів розроблено два основні модулі: на рисунку 2.14 наведено модуль конфігурації для адміністратора з його основними функціями та та модуль роботи з документами.

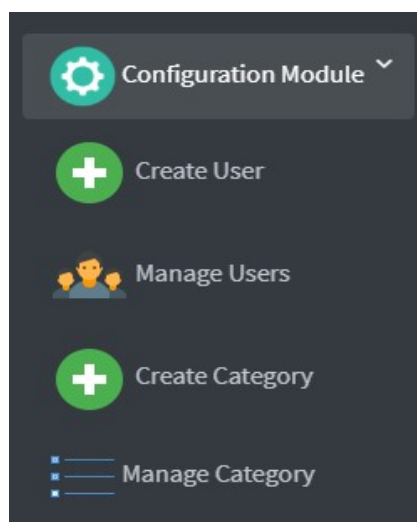
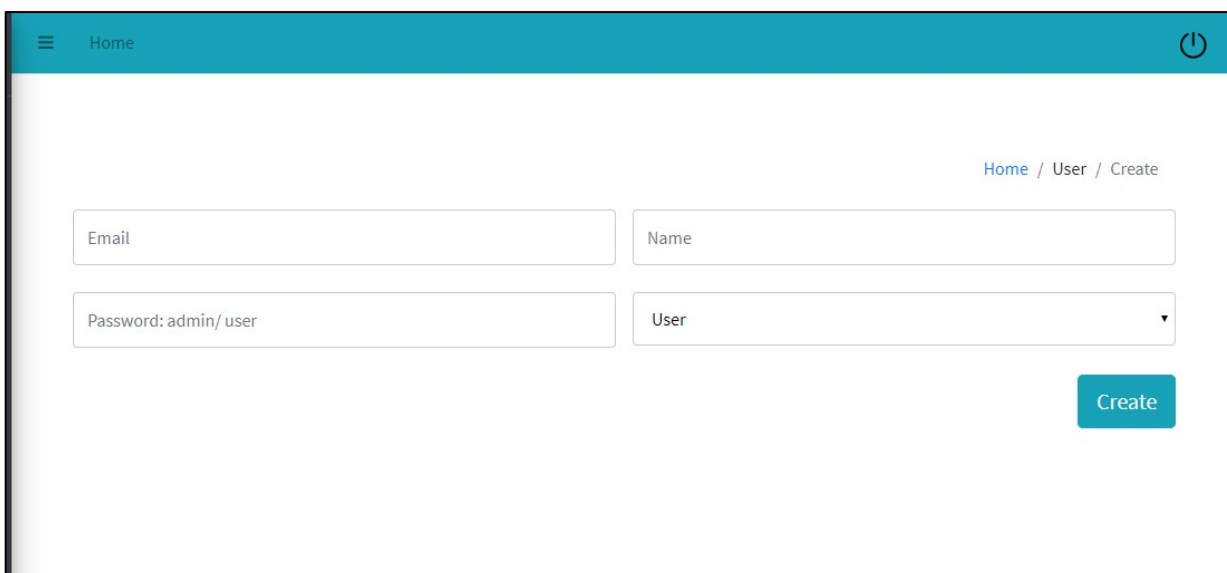


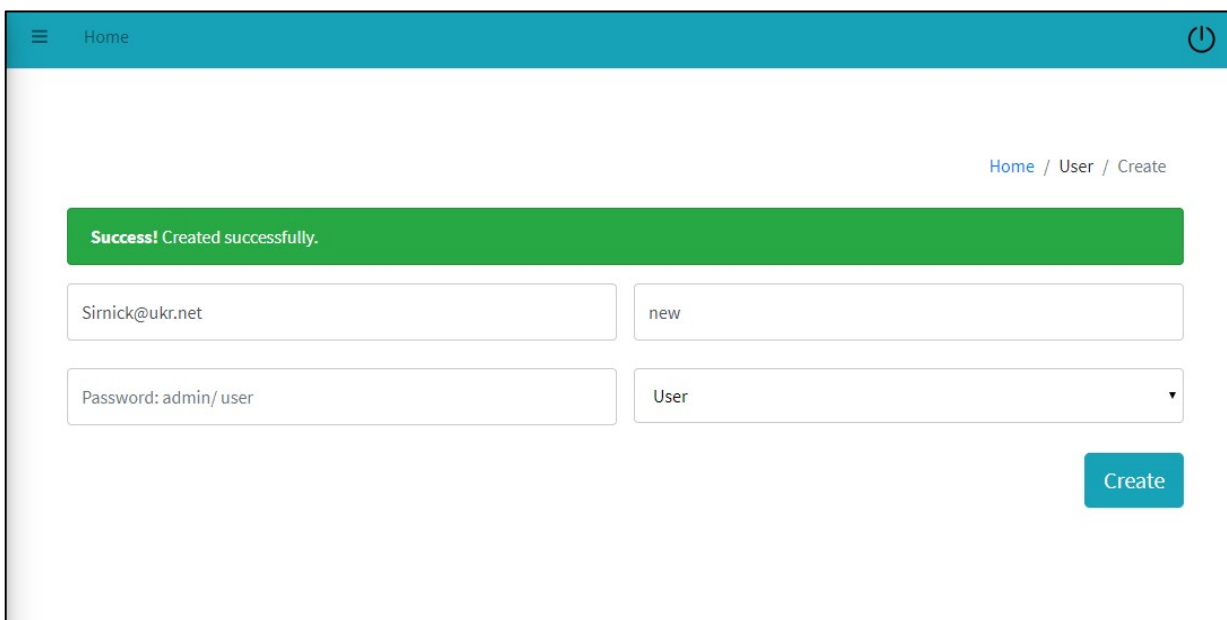
Рисунок 2.14 – Вікно з функціями модулю конфігурації

Тільки для адміністратора системи доступна функція створення нового користувача. На рисунку 2.15 наведено вікно реєстрації нового користувача з відповідними полями, які повинні бути заповненні, а на рисунку 2.16 показано приклад спливаючого повідомлення у випадку вдалої реєстрації.



The screenshot shows a web application interface with a teal header bar containing a menu icon and the text 'Home' on the left, and a power icon on the right. Below the header, a breadcrumb trail reads 'Home / User / Create'. The main content area contains a registration form with four input fields: 'Email', 'Name', 'Password: admin/ user', and a dropdown menu labeled 'User'. A teal 'Create' button is positioned at the bottom right of the form.

Рисунок 2.15 – Вікно реєстрації нового користувача



This screenshot shows the same registration form as Figure 2.15, but with a green success message banner at the top: 'Success! Created successfully.'. The form fields now contain the following data: 'Email' is 'Sirnick@ukr.net', 'Name' is 'new', 'Password' is 'admin/ user', and the 'User' dropdown is set to 'User'. The 'Create' button remains at the bottom right.

Рисунок 2.16 – Спливаюче повідомлення про вдалу реєстрацію

На рисунку 2.17 наведено вікно функції управління користувачами, яка є однією з важливих функцій модуля.

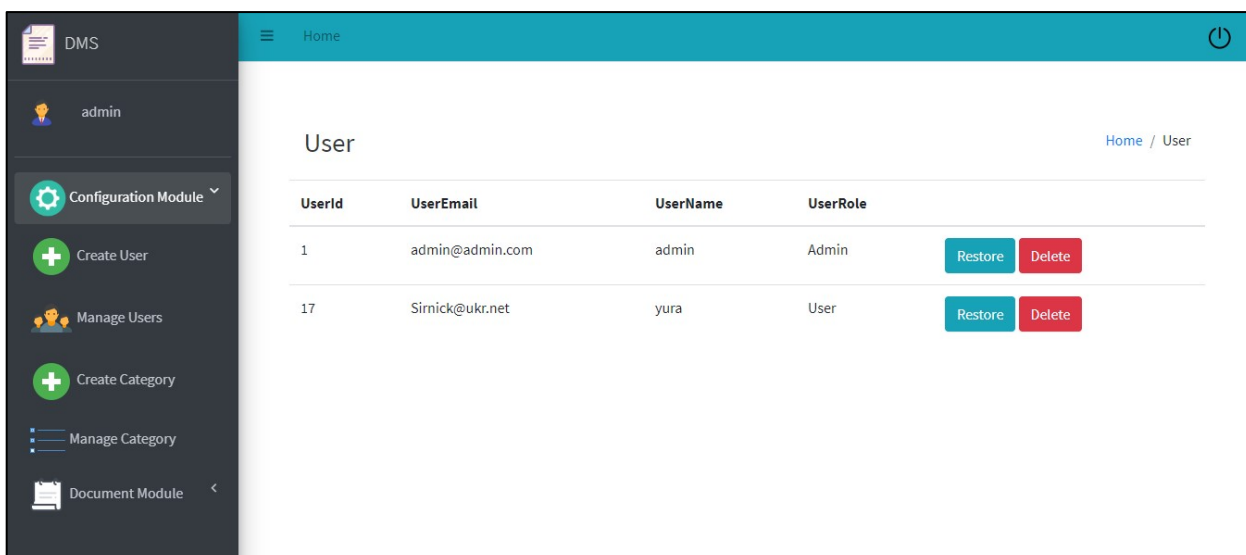


Рисунок 2.17 – Вікно функції управління користувачами

Функції перегляду, видалення та відновлення користувачів доступні тільки для адміністратора.

Користувачам та адміністраторам доступна функція "Створити категорію" для створення категорій документів (рис.2.18).

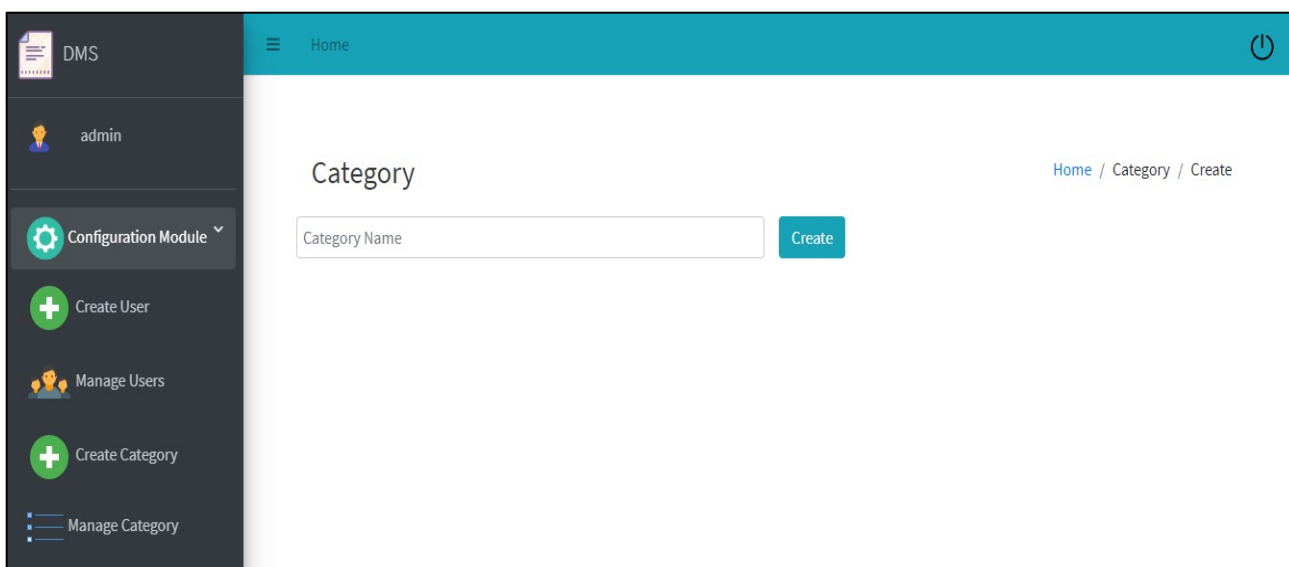


Рисунок 2.18 – Вікно функції Створення категорії

Увівши унікальну назву і натиснувши кнопку "Створити" у разі успіху побачимо спливаюче повідомлення (рис. 2.19).

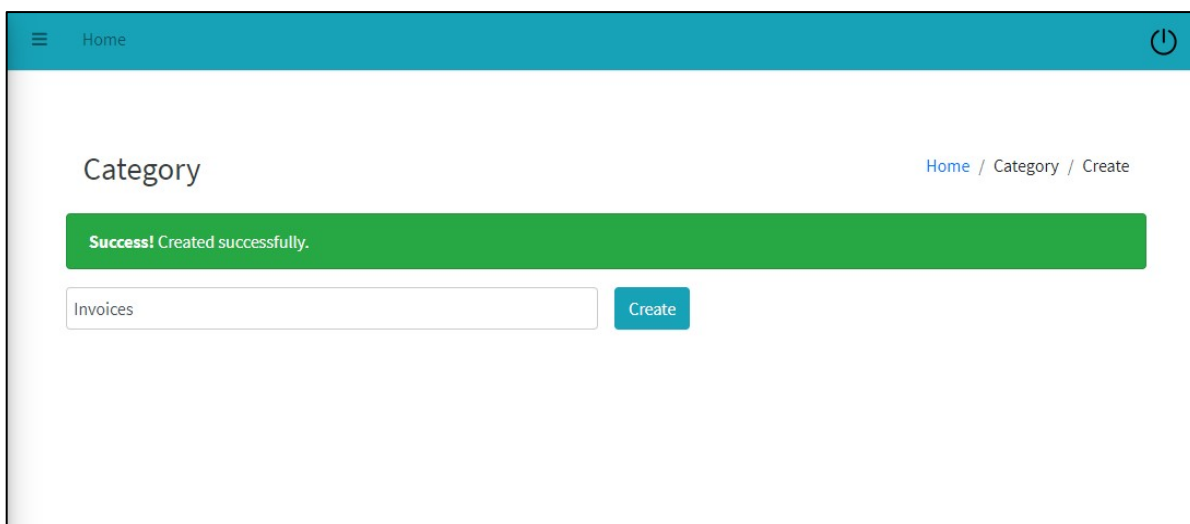


Рисунок 2.19 – Повідомлення успішного створення категорії

Користувач також може переглядати та видаляти категорії документів у меню "Керування категоріями"(рис. 2.20).

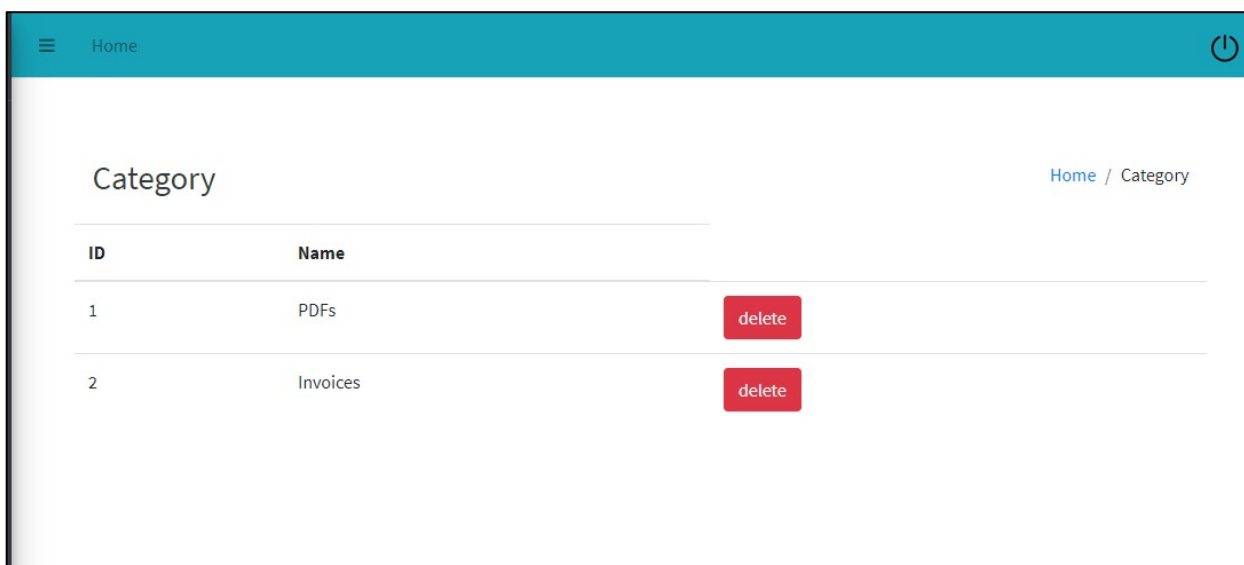


Рисунок 2.20 – Вікно керування та перегляду категорій

Меню керування документами "Модуль документів" доступне для кожного користувача системи. Щоб створити новий документ у системі, потрібно перейти до розділу завантаження документа з назвою "Завантажити документ" (рис. 2.21) та обрати потрібний файл. Після цього необхідно обрати категорію файла, яку необхідно створити раніше перед завантаженням.

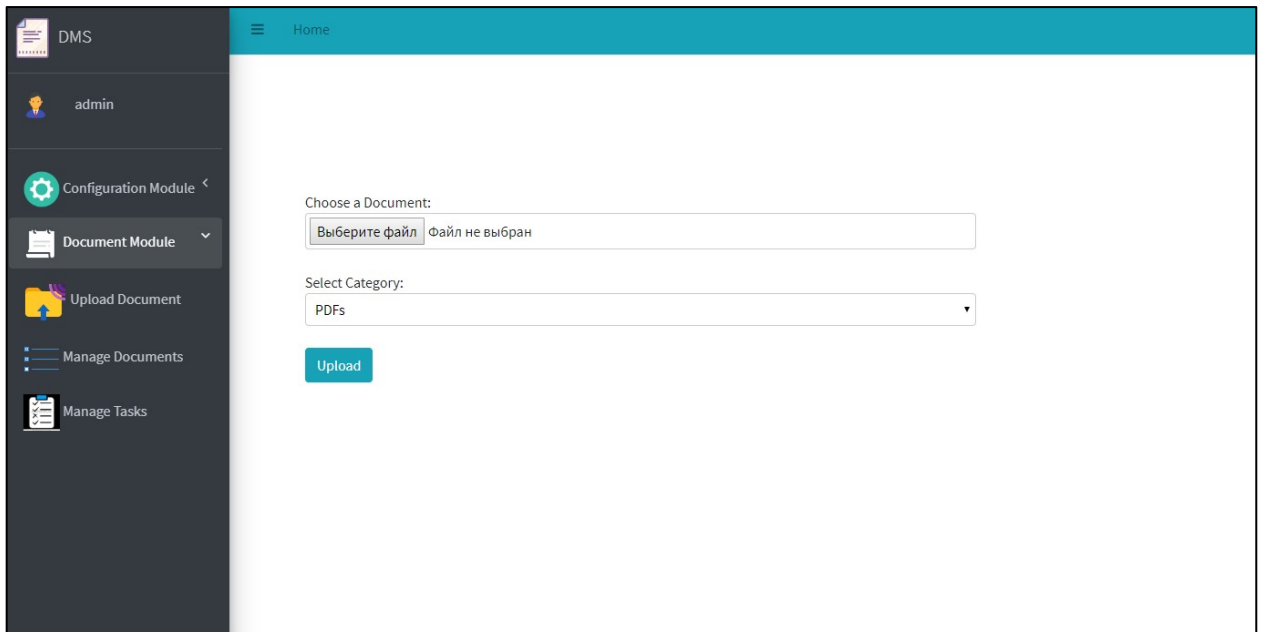


Рисунок 2.21 – Вікно модуля Завантаження документу

Для перегляду доступних документів користувача використовується розділ керування документами під назвою "Керування документами" (рис. 2.22). Саме тут користувач може завантажити документ, підписати його та перевірити статус документу.

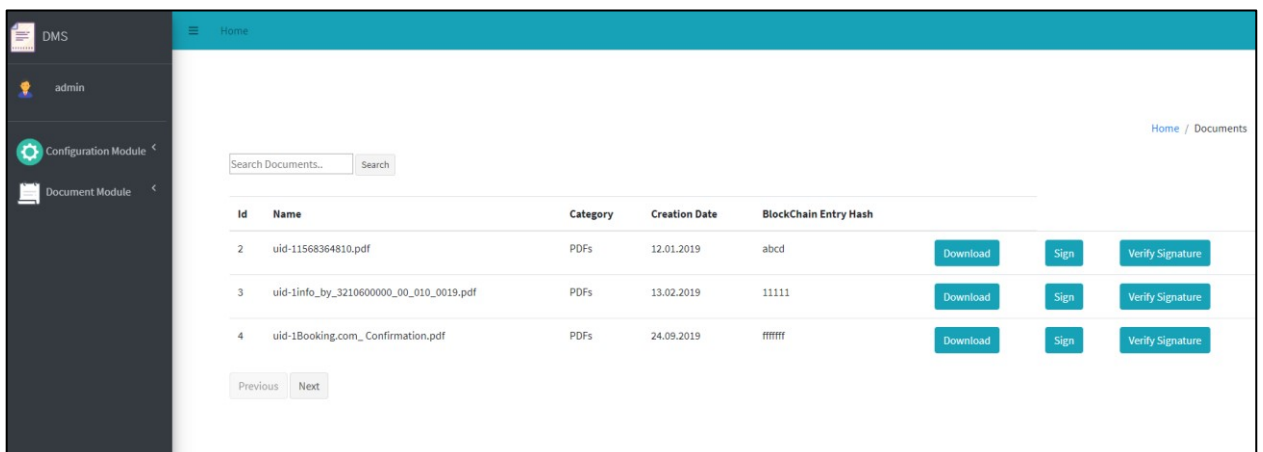


Рисунок 2.22 – Сторінка керування документами

Також користувач системи має можливість створювати завдання для інших користувачів на основі наявного документа (рис. 2.23) і у разі успішного створення отримує відповідне сповіщення (рис. 2.24).

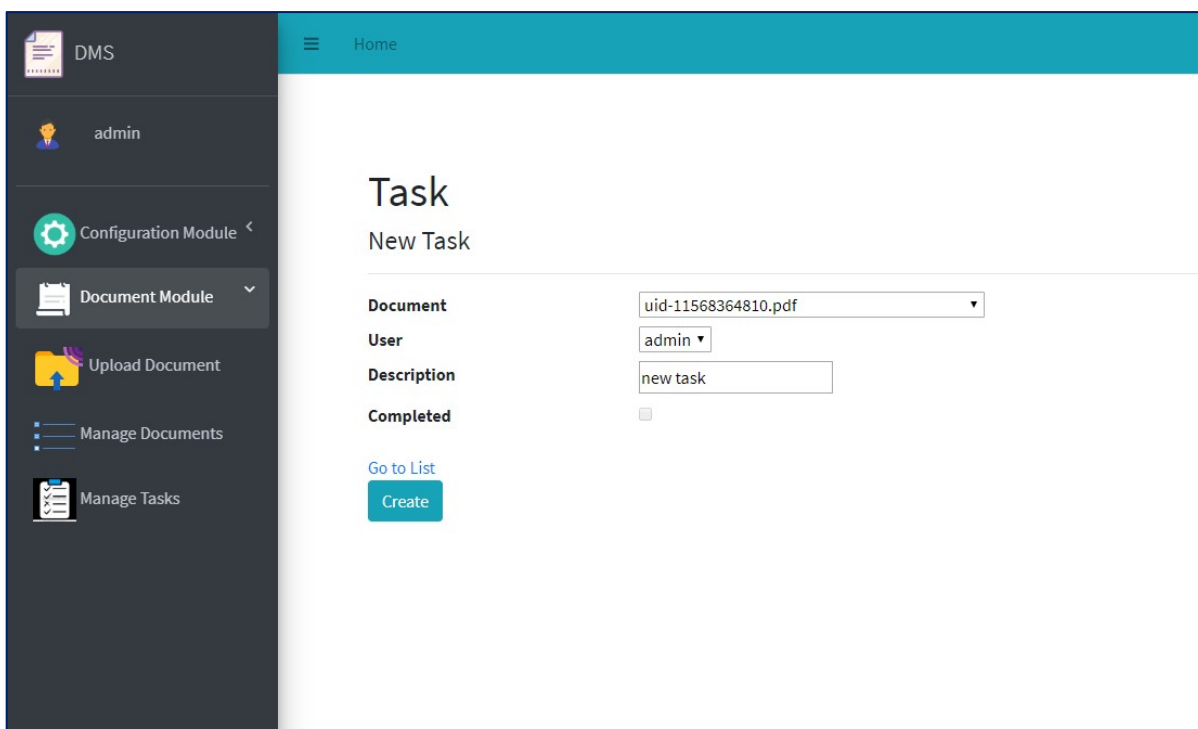


Рисунок 2.23 – Вікно створення завдання

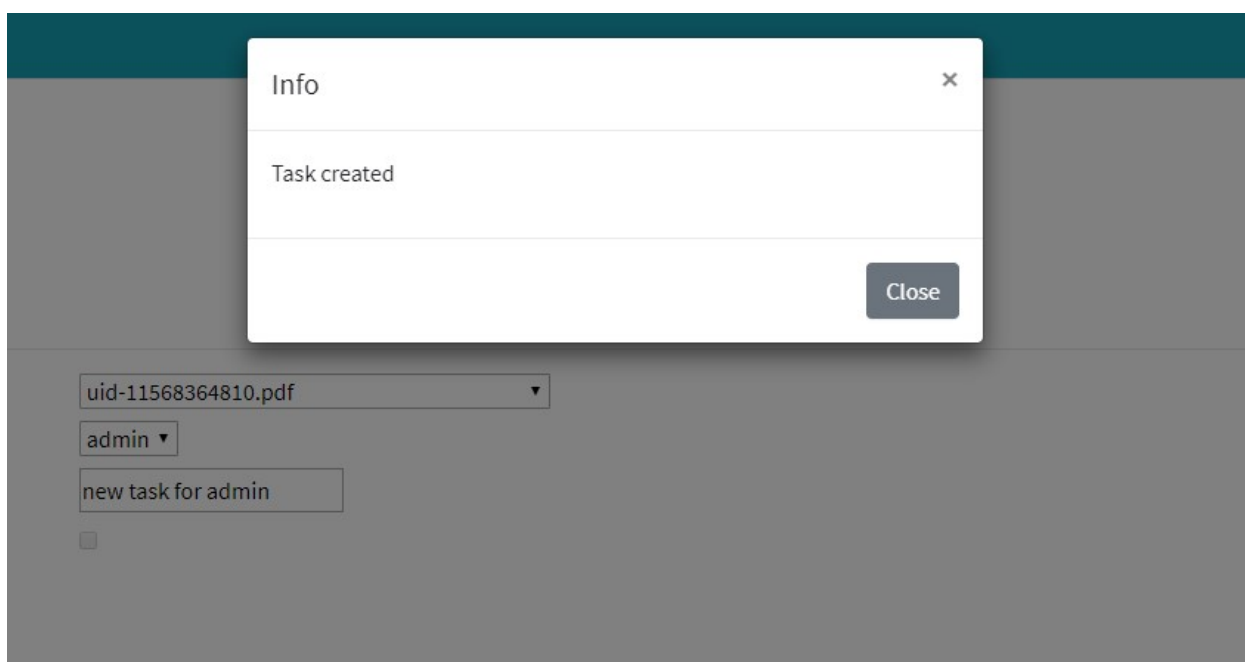


Рисунок 2.24 – Повідомлення про успішне створення

Для перегляду завдань, які створив користувач або які призначені йому, існує сторінка зі списком завдань (рис. 2.25).

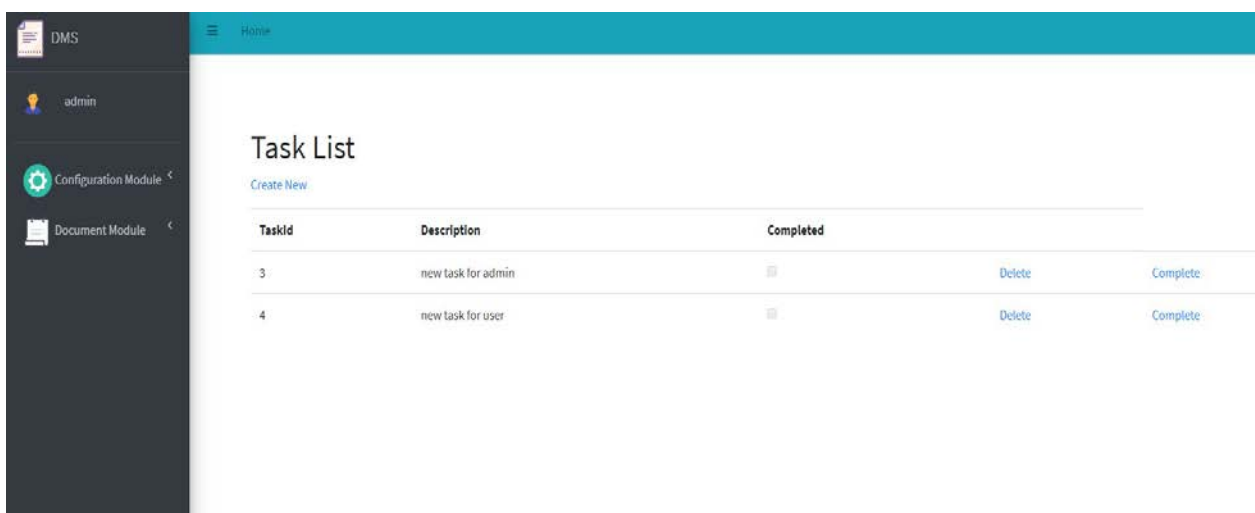


Рисунок 2.25 – Приклад вікна зі списком завдань

На сторінці кожного завдання відображається його унікальний номер у системі, опис завдання та мітка про його виконання. Для завершення або видалення завдання доступні функції "Видалити" та "Завершити". Після виконання цих дій користувач отримує сповіщення про успішне видалення чи завершення завдання.

Для навігації в системі використовуються кнопки "DMS" та "Головна", а для виходу - знак вимкнення у правому верхньому куті (рис. 2.26).

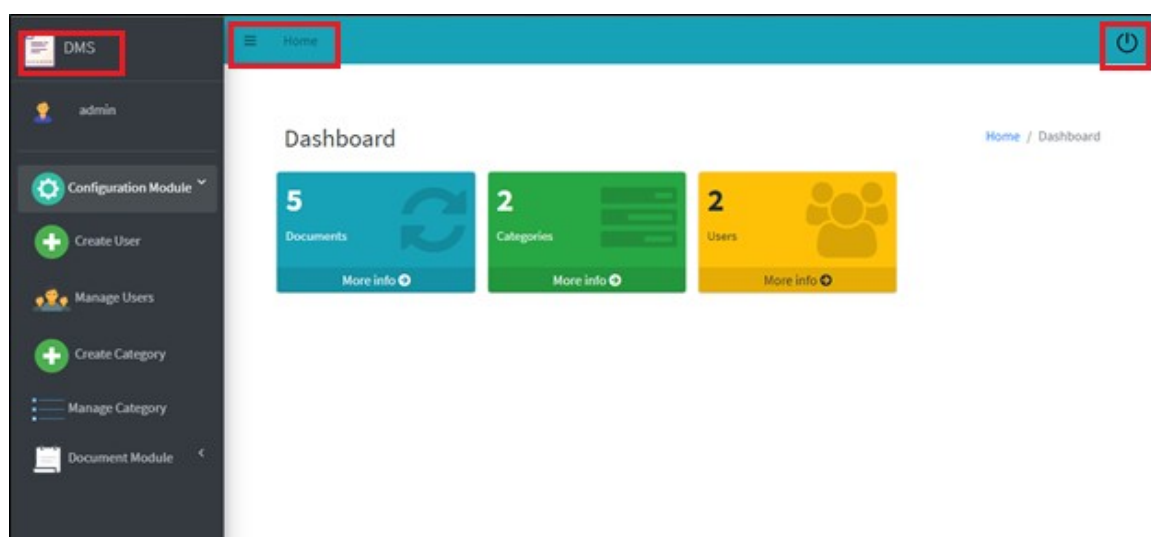


Рисунок 2.26 – Розташування кнопок навігації на головній сторінці

Висновки до розділу:

При виборі технології для розробки системи було враховано такі вимоги:

Система повинна бути кросплатформенною, тому обрана платформа .Net Core з використанням мови C#.

Для зручності користувачів інтерфейс був розроблений з використанням технології AJAX. AJAX – це підхід до створення інтерфейсів веб-додатків, який полягає у фоновому обміні даними між браузером і веб-сервером.

Відкрита реалізація платформи Ethereum – Nethereum обрана для взаємодії з блокчейном, а також мову програмування Solidity для створення смарт-контрактів.

У загальному випадку система містить 21 варіант використання, але для опису особливостей поведінки СЕД на основі технології blockchain у роботі описано 8 ключових варіантів використання, які є найбільш важливими для системи в розрізі Адміністратора та Користувача.

У системі впроваджено сервіс для інтеграції з технологією блокчейн під назвою Blockchain Document Storage Manager. Цей сервіс використовується для взаємодії з блокчейн мережею, зокрема, для створення нових блоків та зчитування даних з існуючих блоків. Він також забезпечує функції перевірки версій документів та їх підписів, що дозволяє забезпечити надійність та автентичність документації в системі.

ВИСНОВКИ

В роботі були вивчені процеси аналізу, розробки та імплементації системи електронного документообігу, що базується на блокчейн-технологіях.

Наразі в Україні на блокчейн вже переведено державний земельний кадастр. На цій технології базується перевірка виписок з реєстру. Також планується переведення державного реєстру речових прав на нерухоме майно на блокчейн. Суть блокчейн полягає в тому, що блокчейн використовує базу даних, розподілену фізично та логічно на тисячах комп'ютерів по всьому світу. Тому вірогідність виводу з ладу такої системи надзвичайно мала. Поки існує хоча б один вузол, блокчейн може функціонувати

Технологія блокчейн виступає перспективною основою для захисту, обміну та перевірки електронних документів у системах документообігу. При верифікації підпису документа, блокчейн може зберігати інформацію про емітента та отримувача кожного ключа разом із самим підписом у базі даних, копії якої ідентично розподілені між комп'ютерами приватної блокчейн-мережі.

Таким чином, документи, зареєстровані в системі блокчейн, мають значні переваги порівняно зі звичайними цифровими підписами. Основною характеристикою мережі блокчейн є неможливість зміни записаних даних, що виключає можливість фальсифікації документів або їхніх підписів.

Кожен підпис може бути перевірений при наявності доступу до мережі блокчейн.

Використання цієї технології також уникає необхідності залучення посередників, що підвищує рівень конфіденційності документів у системі.

Було проведено аналіз існуючих систем електронного документообігу, ідентифіковано ключові вимоги, проблеми та недоліки, які є в сучасних системах цього типу. Також були розглянуті основні принципи та складові елементи технології блокчейн. В результаті було підсумовано основні

переваги використання блокчейн-технологій у системах електронного документообігу.

Технологія блокчейн дозволяє створювати безпечні, надійні та стійкі системи електронного документообігу, здатні знижувати часові витрати користувачів на обробку документів, виконання завдань та архівацію даних.

Ця технологія вирішує основну проблему захисту документів від підроблення. Зростаюча популярність блокчейну в банківській сфері, серед фінансових інститутів та в інших галузях, де критично важлива незмінність даних у транзакціях, підтверджує ефективність цієї технології.

На основі проведеного дослідження можна зробити висновок, що технологія блокчейн представляє значний потенціал для створення надійних та безпечних систем електронного документообігу. Завдяки своїм властивостям незмінності даних, блокчейн забезпечує високий рівень захисту документів від підробок і зловживань. Швидке розповсюдження технології у фінансовій сфері та інших галузях підкреслює її значимість і актуальність. Все це підтверджує зростаючу необхідність імплементації систем електронного документообігу на основі блокчейну в різних сферах бізнесу та управління, що забезпечує оптимізацію процесів, зменшення витрат та підвищення ефективності роботи.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Барановський В.П. Документаційне забезпечення діяльності організації: Навчальний посібник. М.: Асоціація авторів і видавців «Тандем». Вид.-во ЕКМОС, 1999р.
2. Алексєєва Т., Потапенко М. Системи електронного документообігу: від усвідомлення потреби до оцінки та імплементації. ІТСпец. 2008. № 12. С. 40-44.
3. Матвієнко О.В., Цивін М.Н. Основи організації електронного документообігу. 2008. 112 с.
4. Брускіна Т., Дроздов Д. Спецобзор: програмні рішення в сфері електронного документообігу. ІТСпец. 2008. № 12. С. 28-35.
5. Система електронного документообігу FossDoc URL: <https://fossdoc.com/> (Дата звернення 20.04.2024)
6. Система електронного документообігу ДокПроф URL: <http://docprof.com.ua/> (Дата звернення 20.04.2024)
7. Кукарін О.Б. Електронний документообіг та захист інформації : навчальний посібник. К:НАДУ, 2015. 18 с
8. K. Wüst and A. Gervais. Do you need a Blockchain. IACR Cryptol. ePrint Arch., 2017. 375 p
9. W. Meng et al.: When Intrusion Detection Meets Blockchain Technology: A Review. IEEE Access, 2018. 5 p
10. Olnes, S., Ubacht, J., & Janssen, M. Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. // Government Information Quarterly, 34(3), 2017. p.355–364.
11. Arvind Narayanan. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. // Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder. Princeton University Press, 2016.
12. Architecting Modern Web Applications with ASP.NET Core and Microsoft Azure // Steve Smith, 2018.

13. Chris Dannen Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners // APRESS: New York, USA, 2017, DOI 10.1007/978-1-4842-2535-6.

14. Xiuping Lin. Semi-centralized Blockchain Smart Contracts: Centralized Verification and Smart Computing under Chains in the Ethereum Blockchain // Department of Information Engineering, National Taiwan University, Taiwan, R.O.C., 2017.

15. Kendall KE, Kendall JE. Systems Analysis and Design. Fifth Edition. // New Jersey: Prentice Hall. 2002.

16. Леонов И.В. Введення в методологію розробки програмного забезпечення. Ескейп, 2004. 301 с.

17. Про електронні документи та електронний документообіг: Закон України від 07.11.2018 № 36. База даних Законодавство України / ВР України. URL: <https://zakon.rada.gov.ua/laws/show/851-15> (дата звернення: 25.03.2024).

18. Про електронні довірчі послуги: Закон України від 13.02.2020 № 440-IX База даних Законодавство України / ВР України. URL: <https://zakon.rada.gov.ua/laws/show/2155-19> (дата звернення: 25.03.2024)

19. Чукут С.А., Буряченко К.О. Блокчейн чи система електронного документообігу: сучасні тенденції впровадження в органах виконавчої влади України: наукова стаття. К : КПІ ім. Ігоря Сікорського, 2018. 41 с

20. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).

21. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).

22. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація).