

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи інтелектуального обміну
даними у електронному документообігу банку”**

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Лозінський О.С.
« ____ » _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Коваленко А.С.
« ____ » _____ 2026 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Лозінському Олександру Сергійовичу

(прізвище, ім’я, по батькові)

1. Тема роботи Програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку

2. Керівник роботи Коваленко Анна Степанівна, канд. техн. наук, доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи інтелектуального обміну даними у електронному документообігу банку

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 6 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 6 » лютого 2026 р.

Підпис керівника

Коваленко А.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » лютого 2026 р.

Підпис здобувача

Лозінський О.С.
(прізвище та ініціали)

АНОТАЦІЯ

Лозінський О.С. Програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального обміну даними у електронному документообігу банку.

Метою розробки є програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку.

Результат роботи – програмна реалізація системи інтелектуального обміну даними у електронному документообігу банку.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Visual C++.

Ключові слова: комп'ютерні науки, інтелектуальний обмін даними, електронний документообіг

ABSTRACT

Lozinskyi O.S. Software for the intelligent data exchange system in the electronic document flow of a bank. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the intelligent data exchange system in the electronic document flow of a bank.

The purpose of the development is the software for the intelligent data exchange system in the electronic document flow of a bank.

The result of the work is the software implementation of the intelligent data exchange system in the electronic document flow of a bank.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with OS Windows 10/11.

The program was developed in the Visual C++ environment.

Keywords: computer science, intelligent data exchange, electronic document management

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	4
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	15
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми	27
3.4 Розробка діаграми процесів.....	32
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	34
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	34
4.2 Захист розробленого програмного забезпечення.....	50
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	53
6 ОСНОВНІ ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57

					ВКРБ-122.26.0021.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку	Літ.	Аркуш	Аркушів
Розроб.		Лозінський О.С.				Б		
Перев.		Коваленко А.С.					1	63
Н.контр.		Коваленко А.С.				ЦНТУ КН-22		
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕА	—	електронний архів
ЕДО	—	електронний документообіг
САДД	—	система автоматизації електронного документообігу й діловодства
СЕА	—	системи електронних архівів
СДЕ	—	системи діловодства електронні
СУЕД	—	система управління електронним документообігом банку
ТЕ	—	типові елементи
ФФО	—	формат файлових об'єктів
EDMS	—	Electronic Document Management Systems

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Фундамент банківської справи майбутнього не може спиратися на вчорашню архітектуру. Застарілі системи та фрагментовані дані уповільнюють інновації та послаблюють захист. Хоча штучний інтелект посилює безпеку, він також розширює можливості зловмисників, що підживлює складні фінансові злочини. Статичного контролю більше недостатньо; банкам потрібні динамічні, прогнозні та інтегровані системи управління ризиками. 2026 рік – це не про поступові зміни, а про відповідальну трансформацію. Завдяки масштабному впровадженню штучного інтелекту та зміцненню систем управління ризиками, впровадженню сучасних систем електронного документообігу банки можуть забезпечити зростання, довіру та стійкість в рівній мірі.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем інтелектуального обміну даними у електронному документообігу банку.
- Дослідження системи інтелектуального обміну даними у електронному документообігу банку.
- Програмна реалізація системи інтелектуального обміну даними у електронному документообігу банку.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуального обміну даними у електронному документообігу банку.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Для того, щоб виконати постановку задачі, необхідно проаналізувати предметну область дослідження. Об'єктом дослідження в бакалаврській роботі є документообіг канцелярії банку АТ «Ощадбанк». Будь-якому банку в ході своєї діяльності доводиться мати справу з документацією. Документообіг є технологічним процесом в обслуговуванні діяльності підприємства. У банку прийнята змішана форма організації роботи з документами. У кожному структурному підрозділі виконуються операції по створенню спеціалізованих документів, їх систематизації та обробці. Частина операцій документообігу (прийом і обробка вступників та відправляються документів, їх реєстрація, контроль за термінами виконання, архівне зберігання) зосереджується в канцелярії.

Всі функції, права, обов'язки і відповідальність співробітників канцелярії в певній формі закріплені в посадовій інструкції, яка розробляється відділом кадрів банку [19].

1.2 Область застосування

Опис архітектури системи електронного документообігу:

1. Концептуальний рівень. Відповідають на питання: чому? (визначається масштаб системи управління та її інтеграція в загальну систему роботи банку) і хто? (встановлюються взаємозв'язки між причинами впровадження системи управління знаннями та її користувачами).

2. Логічний рівень. Визначаються процеси, які виконуються людьми і машинами. Вирішуються питання: що буде робити система і коли повинен запускатися кожен з процесів?

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

3. Предметний рівень. Втілюється фізична конструкція, що забезпечує функціонування системи. Приймаються рішення, як і де буде впроваджуватися система?

Основними цілями архітектури системи інтелектуального обміну даними у електронному документообігу банку на базі ІТ є:

- чітке формулювання припущень і напрямків політики щодо управління ІТ та документообігом;
- визначення простору управління ІТ та документообігом;
- встановлення найважливіших меж простору управління ІТ та документообігом банку;
- мінімізація перешкод в управлінні ІТ і документообігом (там, де повноваження і відповідальність за документи визначені нечітко);
- визначення бажаних і небажаних варіантів поведінки в процесі управління документообігом;
- розробка стратегії заохочення бажаної поведінки в процесі управління документообігом (ефективна "налаштування" етапів життєвого циклу інформації).

До допоміжних цілей архітектури системи інтелектуального обміну даними у електронному документообігу банку відносяться:

- виключення зайвих джерел і адрес відправки документів, в яких немає необхідності;
- введення в дію нових визначень для термінів, які допускають подвійне тлумачення (файл, запис, оригінал, публікування і ін);
- визначення стану об'єктів управління документообігом і атрибутів системної інформації з точки зору точності, своєчасності, релевантності, цілісності, доступності та гнучкості;
- представлення моделей спільного управління документообігом.

Результатом побудови архітектури системи інтелектуального обміну даними у електронному документообігу банку є:

- чіткі визначення термінів: репозитарій, каталог, публікація, документ, запис, база даних, оригінал, володіння та інше;

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

- збільшення відповідальності автора за створений ним документ протягом усього життєвого циклу управління документацією;
- зниження числа повторюваних джерел інформації і підвищення узгодженості колекції документів (сховищ).

Ієрархічна концепція архітектури системи управління документообігом банку включає в себе наступні страти: репозитарій, приміщення, шафа, ящик, папка, документ.

Система управління електронним документообігом банку реалізує два види процесів:

- 1) основні, орієнтовані на репозитарій, папки з файлами документів;
- 2) допоміжні, орієнтовані на дані, дизайн, управління текстами, текстовий запит, адміністрування, інтеграцію системи, управління списками контролю доступу, забезпечення безпеки протоколів роботи, управління системою робочого процесу, управління всередині – і міждокументаційними посиланнями.

Процеси, орієнтовані на репозитарій, що включають в себе створення репозитарію, модифікацію, і його знищення.

Процеси, орієнтовані на папку, що містять файл або підшивки: складання графіка, відкриття, закриття, встановлення і зняття обмежень на доступ, відгук, роздрук, розмітку, пошук, копіювання, перенесення, відновлення, знищення.

До процесів, орієнтованим на документи, що відносяться додавання, додавання ззовні, одночасне завантаження, копіювання, установка і зняття обмежень на доступ, перегляд, пересилання електронною поштою, архівування, відновлення, роздрук, вилучення, видалення.

Процеси, орієнтовані на дані, що включають в себе додавання, редагування, видалення, обмеження, індексацію, переіндексацію, пошук і складання звіту про системної інформації.

Процеси, орієнтовані на дизайн, що відповідають за різні варіанти подання системної інформації про документ користувачеві: додавання, редагування, видалення складання звіту про варіантах дизайну.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Процеси, орієнтовані на управління текстом, застосовуються для додавання, редагування, видалення, оцінки, індексації та переіндексації тексту.

Процеси текстового запиту використовуються при здійсненні пошуку і складанні звіту про текст.

Процеси адміністрування системи виконують такі адміністративні функції програми: інсталяція та деінсталяція системи клієнтського ПЗ, інсталяція та деінсталяція системи серверного ПЗ, запуск і перезапуск системи, створення резервних копій файлів системи, перенесення системи на інший носій, перевірка цілісності системи, зупинка системи, інсталяція виправлень і вдосконалених версій системи, підтримка системи даних.

Процеси інтеграції системи здійснюють функції впровадження системи програм: інтеграцію та дезінтеграцію середовища додатків, інтеграцію та дезінтеграцію середовища мультисерверних програм, встановлення та виключення взаємодії між репозитаріями, які працюють в різній середовищі, балансування завантаження сервера.

Процеси управління списками контролю доступу виконують функції по забезпеченню безпеки роботи додатків і включають в себе забезпечення безпеки сховища, папок з файлами, документа, версії документа, системної інформації, індивідуальних параметрів пошуку, варіанти дизайну, на рівні груп, на рівні користувача, а також складання звіту про безпеку об'єктів.

Процеси забезпечення безпеки протоколів роботи відповідають за протоколювання подій, що відбуваються в користувацької або серверному середовищі, і містять забезпечення безпеки протоколу роботи користувача і сервера, складання звіту про безпеку протоколу роботи користувача і сервера.

Управління системою робочого процесу передбачає збереження шляхи маршрутизації документа або папки з файлами до користувача машини і реалізує наступні функції: розподіл ролей і встановлення взаємозв'язків, підтримка цілісності робочого процесу.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Управління внутрішньо документальними посиланнями призначений для визначення необхідних посилань, що об'єднують окремі частини складного документа в єдине ціле.

Управління міждокументальними посиланнями використовується для зв'язку окремих об'єктів документів між собою.

Підтримка різних форматів даних. Сучасні системи управління знаннями здатні зберігати і відображати понад 200 різних текстових і графічних форматів даних, поширених в світі. Вони підтримують конвертацію цих форматів в універсальні формати даних, такі як PDF або HTML. Таке безліч форматів дозволяє здійснювати подання документа у вихідному форматі й універсальних форматів, використовуваних для зберігання електронних документів як оригіналів або чергових копій.

Керування версіями. Системи управління документообігом не можуть бути впроваджені в масштабах банку без підтримки ними всіх можливих версій цього документа. Це обумовлено тим, що кожна версія крім системи внутрішньої нумерації і цілісності посилань повинна містити необхідні коментарі, за якими можна відрізнити дану версію від інших (збережених у системі), а також мати можливість створювати окремі атрибути на кожную версію, що дозволяє в будь-який момент часу простежити всю історію обробки документів користувачами, створюють різні версії.

Атрибутування документів. При атрибутування документа користувач заповнює певну електронну форму, яка повинна мати можливість гнучко проектуватися за допомогою стандартних графічних засобів без програмування і дозволяти вносити зміни до складу атрибутів в ході експлуатації системи без ризику порушити бізнес-логіку обробки документів. При цьому атрибути документа повинні зберігатися окремо від його вмісту, що дозволяє забезпечити взаємооднозначну відповідність між фізичним місцезнаходженням вмісту документа і його електронної атрибутивної формою, за яким здійснюється логічний пошук документа. Такий підхід дозволяє побудувати територіально

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

розподілені сховища електронних документів, керованих централізовано, а також зняти будь-які обмеження на обсяг зберігання електронних документів в системі.

Забезпечення можливості створення складових і пов'язаних документів. Складові документи являють собою структури, кожна частина яких, у свою чергу, є самостійним документом зі своїми правилами доступу до нього і своїми процесами колективної розробки та етапами життєвого циклу. Така функціональність дозволяє формувати електронні підшивки, засновані не на фізичній, а на логічній добірці документів за похідними ознаками. В результаті один і той же реальний первинний документ може відображатися в різних віртуальних добірках документів, присвяченим різним тематикам. Це значно підвищує прозорість доступу до документів та дозволяє розділити доступ і повноваження різних підрозділів банку, що беруть участь у комплектації підшивки. Крім того, віртуальний документ сам по собі може брати участь в різних бізнес-процесах і мати власні права доступу та етапи життєвого циклу.

Повнотекстовий пошук та аналіз документів. Сучасні системи управління обов'язково повинні включати в себе підтримку механізмів пошуку за змістом документа (альтернативного атрибутивному пошуку і повнотекстовим аналізу (часто вбудовуються всередину стандартних пошукових систем і об'єднують слова і зв'язують їх логічні оператори)). Є спеціалізовані професійні системи лінгвістичного аналізу, що дозволяють працювати з сховищами документів нарівні з електронною поштою, файлової системи, СУБД, Інтернет, а також реалізовувати більш тонкі механізми вилучення неявних знань, коли сам предмет пошуку не є до кінця формалізованим. В основі таких систем лежить обробка запитів на природній мові, автоматичне анотування текстів, автоматична класифікація на непересічні помітні за змістом області без попереднього завдання критеріїв відмінності.

Управління етапами життєвого циклу документа. Під життєвим циклом розуміється час існування електронного документа від моменту його створення або імпорту до експорту або знищення. Перехід документа з одного етапу на інший

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

здійснюється при виконанні заданої умови зміни якого-небудь властивості документа, наприклад формату, прав доступу, виду атрибутивної картки. Переклад етапів життєвого циклу має здійснюватися як вручну, так і автоматично.

Управління бізнес-процесами банку. Автоматизація бізнес-процесів дозволяє добитися значного зниження непродуктивних втрат часу, пов'язаних з передачею результатів виконання функціональних завдань на інші робочі місця або в різні підрозділи, підтримати прозорість управлінських регламентів та їх відтворюваність, що відповідає стандартам якості ISO 9000 на рівні управління підприємством.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуального обміну даними у електронному документообігу банку, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

У рамках виконання бакалаврської роботи, дослідимо основні сучасні СУЕД банку, що мають широке комерційне використання: Hummingbird Enterprise, Documentum, Microsoft SharePoint Portal Server, Optima Workflow, на їхню відповідність типовим вимогам, сформульованим у роботі.

Сучасний бізнес припускає усе більш складні взаємодії. Необхідно налагодити бездоганну взаємодію й обмін інформацією між людьми, бізнес-системами підприємства з метою забезпечення оптимальної продуктивності, зміцнення ділових зв'язків і збільшення швидкості прийняття рішень. Однак компанії будь-якого розміру, що працюють у всіх галузях, стикаються із труднощами при спробах побудувати автоматизовані системи управління корпоративною інформацією, набудовуються легко у відповідності з мінливими умовами й ефективно використовують доступні й необхідні ресурси.

Hummingbird Enterprise

Hummingbird Enterprise являє собою всеосяжний програмний інструментарій для управління інформаційними потоками в масштабах банку.

Основні переваги.

Для кінцевих користувачів:

- Запуск процесів з популярних додатків, таких як Microsoft Outlook, Word і інших програм Office.

- Повідомлення виконавців про завдання через звичний інтерфейс – по електронній пошті, через браузер, систему миттєвого обміну повідомленнями, мобільний пристрій або модуль Webtop.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– Попередження про прострочені, незакінчені або розширені завдання, делегування одним клацанням миші (при наявності відповідних повноважень у користувача в цьому процесі).

Для адміністраторів:

– Наочний графічний інтерфейс, для простого завдання бізнес-потоків, використовуючи стандартну процедуру «drag and drop», завдання логіки умовних розгалужень, правил ескалації й бізнес-правил.

– Швидке моделювання методом проб і помилок на стадії проектування.

– Ефективний моніторинг статусу за допомогою графічних «моментальних знімків» всіх процесів безпосередньо з Web-браузера.

– Залучення партнерів, постачальників і інших зовнішніх учасників без необхідності додаткового навчання.

– Просте доналаштування й розширення бізнес-процесів при необхідності.

Для компаній:

– Швидке й погоджене просування документації на етапах створення, схвалення й публікації.

– Зберігання всієї інформації, документів і метаданих, пов'язаних із процесом, у захищеному репозитарії.

– Швидка реакція у відповідь на нові запити споживачів і мінливі умови ринку.

Основні компоненти Hummingbird Enterprise:

– Hummingbird DM – інтегрована система управління документами в масштабі підприємства, що дозволяє перетворювати електронну інформацію з документів у знання й яка є основою для рішень в області управління вмістом і колективною роботою. Надає великі можливості банку, пошуку й поширення за допомогою інтуїтивно-зрозумілого середовища, що забезпечує простий і безпечний доступ до будь-яких сховищ корпоративних даних.

– Hummingbird RM – сумісна з основними стандартами система управління записами для створення захищеного, організованого робітничого

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

середовища для контролю над життєвим циклом корпоративних даних від створення до знищення.

– Hummingbird WorkFlow – являє собою всеосяжний програмний інструментарій для управління інформаційними потоками в масштабах банку. Основні характеристики:

– просте графічне проектування процесів документообігу без допомоги ІТ-фахівців;

– контроль, управління й моніторинг всіх поточних процесів;

– залучення зовнішніх учасників при збереженні контролю над процесом;

– автоматичний запуск процесів, з огляду на різні етапи життєвого циклу інформації, таких як зворотне копіювання в сховище, публікація й знищення;

– розширення процесів Hummingbird Enterprise шляхом інтеграції зі сторонніми системами;

– призначення, моніторинг і завершення завдань і пов'язаної з ними документації через обраний користувачем інтерфейс – електронну пошту, браузер, ІМ або мобільний пристрій.

– Hummingbird Collaboration – надійно захищене мережне середовище колективної роботи й взаємодії для розподілених робочих груп для банку спільної роботи як внутрішніх (співробітників компанії), так і зовнішніх (замовників, партнерів) учасників на основі проектів / документів.

– Hummingbird KM – сполучення потужних технологій, що забезпечують пошук, категоризацію й добування інформації для найбільш швидкого й коректного доступу до корпоративних даних і накопиченого досвіду.

– Hummingbird Portal – робітниче середовище на основі Web, що погоджує воедино додатки й інформацію, надаючи повний огляд накопичених корпоративних даних.

– Hummingbird BI – комплексний пакет аналізу, що дозволяє задавати питання й одержувати відповіді для наступного використання при побудові актуальних і точних звітів. Це інтегроване й масштабоване рішення дозволяє

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

одержувати й аналізувати дані з CRM і ERP систем, сховищ даних, успадкованих додатків і нестандартних архівів даних.

– Hummingbird ETL – потужний засіб інтеграції даних, для виконання добування, перетворення й завантаження (ETL) даних між додатками, сховищами й архівами даних. Це рішення дозволяє здійснювати оперативний і коректний обмін даними таким чином, що критично важливі структуровані дані будуть доступні для інтелектуальних працівників банку для побудови запитів і аналізу.

Documentum

Платформа Documentum 6 є основою широкого спектра продуктів EMC Documentum, що включає крім платформної інфраструктури програмне забезпечення, згруповане у відповідні набори продуктів для кожної з областей керування змістом.

– Knowledge Worker Suite призначений для підтримки процесів створення вмісту й процесів взаємодії співробітників у ході його узгодження й обговорення. Найкраще застосування Knowledge Worker Suite знаходить у колективних процесах створення вмісту – таких, як керування технічною документацією, розробка нового продукту, розробка договорів і комерційних пропозицій й т.і.

– Transactional Content Management вирішує завдання автоматизації високопродуктивних операційних процесів. Основу Transactional Content Management становить EMC Documentum Business Process Suite, що забезпечує керування повним життєвим циклом бізнес-процесів – від проектування й моделювання, до аналізу й оптимізації.

– Interactive Content Management забезпечує процеси створення, керування, публікації й доставки різних видів цифрового медіа-змісту по різних каналах. Рішення поєднує технології Digital Asset Management і Web Content Management.

– Compliance & Archiving – це набір продуктів, що забезпечують можливості збору й архівування змісту для довгострокового зберігання

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

відповідно до корпоративних регламентів і вимогами регулювання. На основі продуктів Compliance & Archiving будуються рішення для довгострокового зберігання електронних документів, що відповідають таким стандартам керування записами, як Do 5015.2, DOMEA, MoReq, ISO 15489.

Гнучка, з найширшим діапазоном можливостей, платформа Documentum 6, є основою для додатків по керуванню змістом і архівуванню, надаючи єдині інструменти й сервіси для керування змістом, процесами й сховищами. Ключові інновації Documentum 6 включають:

- Спрощення розробки й інтеграції додатків завдяки сервісно-орієнтованому підходу до керування змістом.
- Прискорення розробки додатків завдяки новим засобам розробки, у яких застосовується технологія Eclipse.
- Розширення можливостей налаштування, що дозволяють прискорити розробку й багаторазово використовувати наявні результати.
- Нові засоби масштабування, істотно підвищувальна продуктивність у розподілених конфігураціях.

На сьогоднішній день EMC Documentum 6 є єдиною на ринку уніфікованою ЕСМ-платформою корпоративного класу. Documentum 6 надає кращу інфраструктуру для будь-яких додатків по керуванню змістом у корпоративному масштабі й надає самий широкий набір сервісів по керуванню змістом, що дозволяє замовникам працювати з максимальною продуктивністю за рахунок оптимального використання інформаційних активів.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації програми мною була використана мова програмування Visual C++. У зв'язку з тим, що сьогодні рівень складності програмного забезпечення дуже високий, розробка додатків Windows з використанням тільки

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

якої-небудь мови програмування значно утрудняється. Програміст повинен затратити масу часу на рішення стандартних завдань по створенню багатовіконного інтерфейсу. Реалізація технології зв'язування й вбудовування об'єктів – OLE – зажадає від програміста ще більш складної роботи. Щоб полегшити роботу програміста практично всі сучасні компілятори з мови C++ містять спеціальні бібліотеки класів. Такі бібліотеки містять у собі практично весь програмний інтерфейс Windows і дозволяють користуватися при програмуванні засобами більш високого рівня, чим звичайні виклики функцій. За рахунок цього значно спрощується розробка додатків, що мають складний інтерфейс користувача, полегшується підтримка технології OLE і взаємодія з базами даних. Сучасні інтегровані засоби розробки додатків Windows дозволяють автоматизувати процес створення додатка. Для цього використовуються генератори додатків. Програміст відповідає на питання генератора додатків і визначає властивості додатка – чи підтримує воно багатовіконний режим, технологію OLE, тривимірні органи керування, довідкову систему. Генератор додатків, створить додаток, що відповідає вимогам, і надасть вихідні тексти. Користуючись їм як шаблоном, програміст зможе швидко розробляти свої додатки. Подібні засоби автоматизованого створення додатків включені в компілятор Microsoft Visual C++ і називаються MFC AppWizard. Заповнивши кілька діалогових панелей, можна вказати характеристики додатка й одержати його тексти, постачені великими коментарями. MFC AppWizard дозволяє створювати одновіконні й багатовіконні додатки, а також додатки, що не мають головного вікна, – замість нього використовується діалогова панель. Можна також включити підтримку технології OLE, баз даних, довідкової системи. Звичайно, MFC AppWizard не всесильний. Прикладну частину додатка програмістові прийдеться розробляти самостійно. Вихідний текст додатка, створений MFC AppWizard, стане тільки основою, до якої потрібно підключити інше. Але працюючий шаблон додатка – це вже половина всієї роботи. Вихідні тексти додатків, автоматично отриманих від MFC AppWizard, можуть становити

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

сотні рядків тексту. Набір його вручну був би дуже стомлюючий. Потрібно відзначити, що MFC AppWizard створює тексти додатків тільки з використанням бібліотеки класів MFC (Microsoft Foundation Class library). Тому тільки вивчивши мову C++ і бібліотеку MFC, можна користуватися засобами автоматизованої розробки й створювати свої додатки в найкоротший термін. Як уже згадувався, MFC – це базовий набір (бібліотека) класів, написаних мовою C++ і призначених для спрощення й прискорення процесу програмування для Windows. Бібліотека містить багаторівневу ієрархію класів, що нараховує близько 200 членів. Вони дають можливість створювати Windows-додатки на базі об'єктно-орієнтованого підходу. З погляду програміста, MFC являє собою каркас, на основі якого можна писати програми для Windows. Бібліотека MFC розроблялася для спрощення завдань, що стоять перед програмістом. Як відомо, традиційний метод програмування під Windows вимагає написання досить довгих і складних програм, що мають ряд специфічних особливостей. Зокрема, для створення тільки каркаса програми таким методом знадобиться близько 75 рядків коду. У міру ж збільшення складності програми її код може досягати воістину неймовірних розмірів. Однак та ж сама програма, написана з використанням MFC, буде приблизно в три рази менше, оскільки більшість приватних деталей приховано від програміста.

Одною з основних переваг роботи з MFC є можливість багаторазового використання того самого коду. В зв'язку з тим, що бібліотека містить багато елементів, загальних для всіх Windows-додатків, немає необхідності щораз писати їх заново. Замість цього їх можна просто успадковувати (говорячи мовою об'єктно-орієнтованого програмування). Крім того, інтерфейс, забезпечуваний бібліотекою, практично незалежний від конкретних деталей, його що реалізують. Тому програми, написані на основі MFC, можуть бути легко адаптовані до нових версій Windows (на відміну від більшості програм, написаних звичайними методами). Ще однією істотною перевагою MFC є спрощення взаємодії із прикладним програмним інтерфейсом (API)

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Windows. Будь-який додаток взаємодіє з Windows через API, що містить кілька сотень функцій. Значний розмір API утрудняє спроби зрозуміти й вивчити його цілком. Найчастіше навіть складно простежити, як окремі частини API зв'язані один з одним! Але оскільки бібліотека MFC поєднує (шляхом інкапсуляції) функції API у логічно організовану безліч класів, інтерфейсом стає значно легше управляти.

Оскільки MFC являє собою набір класів, написаних мовою C++, тому програми, написані з використанням MFC, повинна бути в той же час програмами на C++. Для цього необхідно володіти відповідними знаннями. Для початку необхідно вміти створювати власні класи, розуміти принципи спадкування й уміти перевизначати віртуальні функції. Хоча програми, що використовують бібліотеку MFC, звичайно не містять занадто специфічних елементів з арсеналу C++, для їхнього написання проте потрібні солідні знання в даній області.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуального обміну даними у електронному документообігу банку.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ-2026

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Проаналізуємо, яким чином організована діяльність щодо документообігу в банку, для цього використовуємо методи функціонального моделювання. Для цього побудуємо функціональну модель предметної області.

В даний час в діяльності канцелярії банку беруть участь як співробітники канцелярії. Виконання завдань відділу починається з отримання вхідних документів.

На основі інструкції з діловодства та порядку документообігу виконується підготовка вихідних документів, а також ведення реєстрів вхідних і вихідних документів і підготовка звітності.

Розглянемо більш докладно діяльність по канцелярії.

В організації прийнята журнальна система обліку документів. Вона вважається застарілою, і має безліч недоліків. При реєстрації документів всі необхідні дані заносяться до відповідного журналу. При виникненні необхідності пошуку потрібного документа доводиться послідовно переглядати записи в журналі реєстрації. З огляду на, що обсяги документообігу канцелярії чималі, процес пошуку необхідного документа затягується на тривалий час.

У відділі немає чіткої системи відстеження за термінами виконання документів, поставлених на контроль, тому неможливо, вчасно знайти прострочений документ і вчасно вжити заходів до виконавця, який прострочив цей документ.

Обробка вхідних документів починається з їх попереднього розгляду діловом канцелярії. Він розподіляє документи:

- керівнику банку;
- в структурні підрозділи, які не потребують резолюції керівника.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Усі вхідні документи поділяються на:

– документи, які реєструються, (документи, адресовані керівнику організації і документи, що містять інформацію з найбільш важливих питань діяльності банку, такі як листи).

– документи, які не реєструються:

– вітальні листи та телеграми;

– кореспонденція з позначкою «особисто»;

– документи, що надходять на адресу громадських організацій;

– документи первинного бухгалтерського обліку (банківські рахунки, рахунки – фактури, кредитні довідки, накладні, вимоги, платіжні доручення, що приходять ордера);

– документи первинного статистичного обліку та звітності;

– документи з кадрових питань та підвищення кваліфікації;

– документи з питань технічної інформації (періодичні видання, реклама);

– договору, порядку – запрошення правоохоронних органів.

Неzareєстровані документи розкладаються по структурним підрозділам для доставки за призначенням, решта документація реєструється в «журналі реєстрації вхідних документів» і передається відповідно до попереднім розподілом для внесення резолюції.

При розгляді документа керівником пишеться резолюція, яка містить рішення по суті поставленого в документі питання. У резолюції визначається ПІБ виконавця, чіткі і конкретні вказівки по виконанню, термін виконання, підпис, дата. [15].

Розглянуті керівником документи повертаються в канцелярію на наступний день, де інформація про документи, ухвалені на контроль заноситься в журнал «контроль виконання», потім заповнюється реєстр отримання вхідної кореспонденції та документ передається виконавцю під розпис. Протягом терміну діловод здійснює регулярні перевірки ходу виконання. Виконавець повертає виконаний документ не пізніше дня виконання для внесення інформації в журнал і зняття документа з контролю. Виконані документи підшиваються в справу.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Робота з вихідними документами починається з підготовки відповідальним виконавцем роботи тексту документа, який потім узгоджується із зацікавленими посадовими особами. Остаточний варіант документа передається на підпис керівнику. Діловод може повернути документи виконавцю на доопрацювання, якщо вони оформлені неправильно. Дані про вихідному документі заносяться в «Журнал реєстрації вихідних документів». Листи пишуться в трьох примірниках. Перший примірник після реєстрації пересилається адресатові, другий примірник залишається у виконавця і третій підшивається до справи [15].

Внутрішні документи (накази) надходять до канцелярії для реєстрації в «журналі реєстрації наказів». При реєстрації їм присвоюється номер і дата. Після реєстрації наказ розмножується в канцелярії і копії віддаються особам, яких необхідно з ним ознайомити, а на оригіналі (на зворотному боці наказу) ставиться підпис всіх обізнаних осіб. Після виконання документ знімається з контролю і підшивається в справу. [15].

Щомісяця діловод показують керівнику зведення за результатами виконання документів за минулий період. Ці звіти узагальнюють інформацію по різних станам, рухам і виконанням документів.

Канцелярія становить наступні види звітів:

- Звіт про вхідні документи.
- Звіт про документи, що значаться за виконавцями, тобто знаходяться на контролі.
- Звіт про виконані документи.
- Звіт про передані на виконання документи.
- Звіт про невиконані вхідні документи.
- Звіт про внутрішні документи.

Аналіз звітів допомагає своєчасно вжити заходів, що сприяють своєчасному виконанню документів та підвищенню виконавської дисципліни. [15]

Система автоматизованого документообігу впроваджується, щоб вирішувати більш ефективно завдання, що стоять перед всією організацією:

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

- автоматична реєстрація документа. Занесення в реєстраційну контрольну картку відомостей про документ;
- автоматичне формування журналів реєстрації документів на основі заповненої реєстраційної картки;
- можливість внесення змін або видалення документа;
- можливість швидкого пошуку, як звичайного документа, так і такого, що знаходиться на контролі за різними реквізитами;
- організація автоматичного контролю за термінами виконання документів;
- автоматичне формування типових звітів;
- ведення та використання різних довідників.

Як вже було сказано, при впровадженні системи на банку, передбачається досягти наступних результатів:

1. Скорочення часу на обробку інформації.
2. Збільшення обсягів обробки інформації.
3. Скорочення часу пошуку документів і підготовки звітів.
4. Зменшення кількості помилок і підвищення якості обробки документації.
5. Підвищення технічного рівня і коефіцієнта використання обчислювальних засобів.
6. Підвищення якості контролю над документообігом, виключення втрати окремих документів і забезпечення контролю виконання документів на кожному рівні виконання;
7. Підвищення інформованості керівництва за підвищення повноти та достовірності звітів.
8. Зменшення вартості документаційного забезпечення управління підприємством за рахунок переходу від паперового діловодства до електронного, зниження вартості копіювання та передачі паперових документів.

Впровадження автоматизованої системи документообігу на банку повинно мати певну економічну ефективність. Саме аспект отримання найбільшого економічного ефекту представляється найважливішим при ухваленні рішення про реалізацію роботи впровадження автоматизованого документообігу.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3.2 Розробка структурної схеми

Відповідно, виходячи з вищесказаного, система управління електронного документообігу забезпечує:

1. Розмежування прав доступу до розробленої документації. Права доступу встановлюються для кожного об'єкта (залежно від ієрархії на дереві параметричної моделі), для кожного виробу (на рівні випуску певного типу документації).

2. Виключення дублювання й множинного введення тих самих характеристик у документах, що випускаються. При створенні нових документів виробляється автоматичний пошук наявності документа в дереві документів, при наявності документа в дереві формується запит на редагування наявного документа або заміни його новим документом. Будь-які дані мають у сховище даних своє унікальне місце розташування й дерево зв'язків цих даних з іншими об'єктами. Ці дані вводяться в сховище користувачем, що має на це відповідні права. При цьому активізуються зв'язки з іншими об'єктами й ним передаються посилання на ці дані із вказівкою, що об'єкт повинен зробити із цими даними (використовувати в якості вихідних даних, погодити документ і т.і.).

3. Автоматичну реєстрацію розроблених документів. При створенні нового документа виробляється автоматичне внесення документа в дерево документів з наданням прав на даний документ, після чого він здобуває статус робочого документа.

4. Відображення інформації про документ на різних картках електронної картотеки. Картка електронної картотеки являє собою дерево параметрів документів, що забезпечує швидкий пошук необхідної інформації й визначення стадії створення документа.

5. Надання списку всіх документів, що перебувають у роботі в когонебудь із проєктантів виробу. Це важливо, наприклад, для головних конструкторів і керівників відділів, які одержать можливість:

– у реальному масштабі часу, у дереві об'єктної моделі, визначити в якого об'єкта й у якому стані перебуває будь-який документ, що цікавить;

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

- одержати оперативну інформацію про маршрут проходження документа, що цікавить, і часу його знаходження на кожному етапі маршруту;
- визначити строки проходження етапів маршруту документа, що цікавить;
- у випадку порушення цих строків визначити зв'язку, по яких автоматично згенеруються повідомлення про порушення строків проходження етапів у маршруті документа;
- оцінити вплив порушення строків проходження документа на розробку пов'язаних з ним документів.

6. Забезпечення легкого пошуку документів і пов'язаної з ними інформації з можливістю збереження параметрів для подальшого використання. За допомогою електронної картотеки можна швидко знайти необхідний документ, або за допомогою запиту з певними параметрами знайти документи, схожі по певних атрибутах. Крім цього, можна використовувати повнотекстний пошук документації. При цьому запит може бути збережений для можливого його повторення в списку найбільше часто повторюваних запитів.

7. Автоматичну архівацію документів.

8. Маршрутизацію документів залежно від типу ділової процедури.

9. Роботу із загальнорозпорядничою документацією. За допомогою редактора загальнорозпорядничої документації можна створювати накази, рішення, розпорядження; погоджувати їх, ставити на контроль, відслідковувати строки виконання й одержувати звіти.

10. Супровід процесу розробки документації з паралельним звертанням до електронних документів.

При реалізації інформаційного забезпечення на основі параметричної моделі об'єкта схема взаємодії параметричних об'єктів проєктованих виробів має більш структурований вид, що різко зменшує кількість зв'язків, виключає дублювання потоків інформації, систематизує адміністрування, забезпечує контроль проходження документації й веде до істотних вигід, що дозволяє перевести процес автоматизації проєктування на принципово новий рівень.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

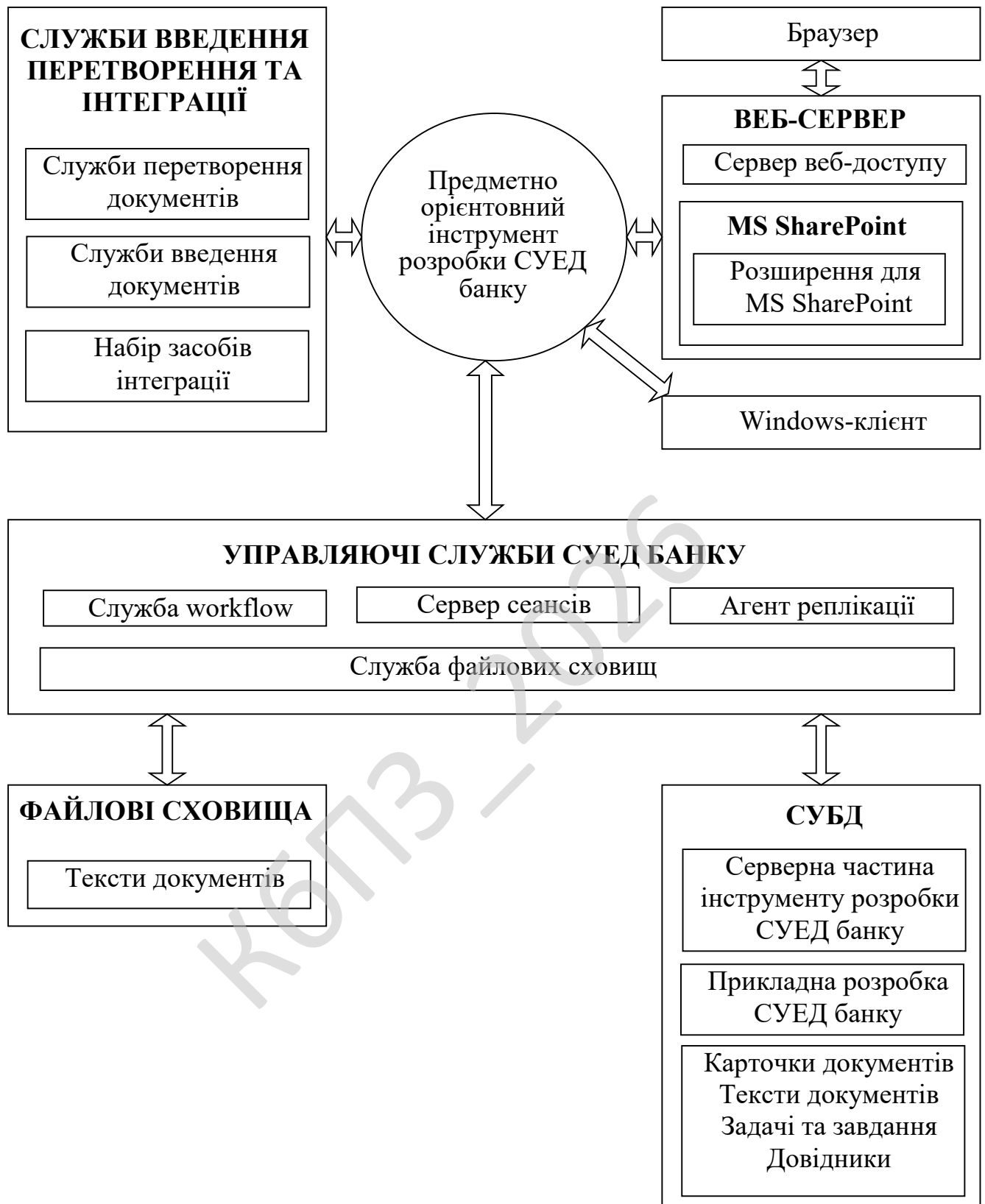


Рисунок 3.1 – Структурна схема системи

На базі системи СУЕД банку розробляється широкий набір бізнес-рішень, націлених на побудову ефективної системи керування бізнесом.

Бізнес-рішення – це програмно-консалтинговий продукт, орієнтований на досягнення заданого ефекту при рішенні певного бізнесу-завдання. Відмінні риси бізнес-рішення:

- Фокус на бізнес-завданнях. Метою бізнес-рішення є задоволення потреб конкретних бізнес-замовників і рішення конкретного бізнесу-завдання.

- Вимірюваний бізнес-ефект. Для кожного бізнес-рішення визначається перелік показників ефективності, які дозволяють виміряти ефект від його впровадження в банку.

- Методики й бізнес-консалтинг. Бізнес-рішення крім технічного рішення включає також методичні матеріали, технологію впровадження, послуги бізнесу-консалтингу.

Бізнес-рішення дають замовникам нові можливості в реалізації проектів, пов'язаних з електронним документообігом банку, керуванням бізнес-процесами й взаємодією. Завдяки пропрацьованості бізнес-рішень і орієнтації їх на конкретні бізнес-завдання, впровадження проходить із мінімальними ризиками.

Проекти реалізуються в цілому швидше й більш результативно. При цьому різні бізнес-рішення побудовані на одній основі – системи СУЕД банку. Це дозволяє забезпечити їхнє послідовне впровадження в банку й будувати з окремих бізнес-рішень цілісну інформаційну систему.

3.3 Розробка функціональної схеми

На основі структурної схеми моделі середовища системи обміну даними була розроблена функціональна схема моделі середовища відкритої СУЕД банку, при цьому визначені наступні об'єкти стандартизації:

- служби засобів проектування;
- служби засобів управління змістом;

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

- служби засобів моделювання процесів;
- служби засобів моделювання даних;
- формати файлових об'єктів;
- служби засобів забезпечення безпеки;
- інтерфейс користувача;
- служби засобів управління даними;
- служби засобів управління системою;
- служби засобів управління сховищем;
- служба комунікацій.

Отримані дані дозволили синтезувати функціональну схему еталонної моделі середовища відкритої СУЕД банку (рисунки 3.2).

Дана модель використана при розробці функціонального стандарту середовища відкритої СУЕД банку. Призначенням даного стандарту є:

- Забезпечення мобільності (переміщуваності) СУЕД банку.
- Забезпечення інтероперабельності СУЕД банку.
- Забезпечення масштабованості СУЕД банку.
- Забезпечення адаптуємості системи.

Визначено, що область застосування функціонального стандарту СУЕД банку установлює загальні положення по створенню й експлуатації відкритої системи управління документообігом банку на основі окремих модулів, що входять у єдину інформаційну систему банку.

Положення функціонального стандарту підлягають застосуванню при рішенні завдань:

- створення, модернізації СУЕД банку;
- банку доступу користувачів до ресурсів СУЕД банку;
- інтеграції обчислювальних і інформаційних ресурсів із СУЕД банку.

Для заповнення профілю в роботі була розроблена методика вибору стандартів на основі експерименту.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

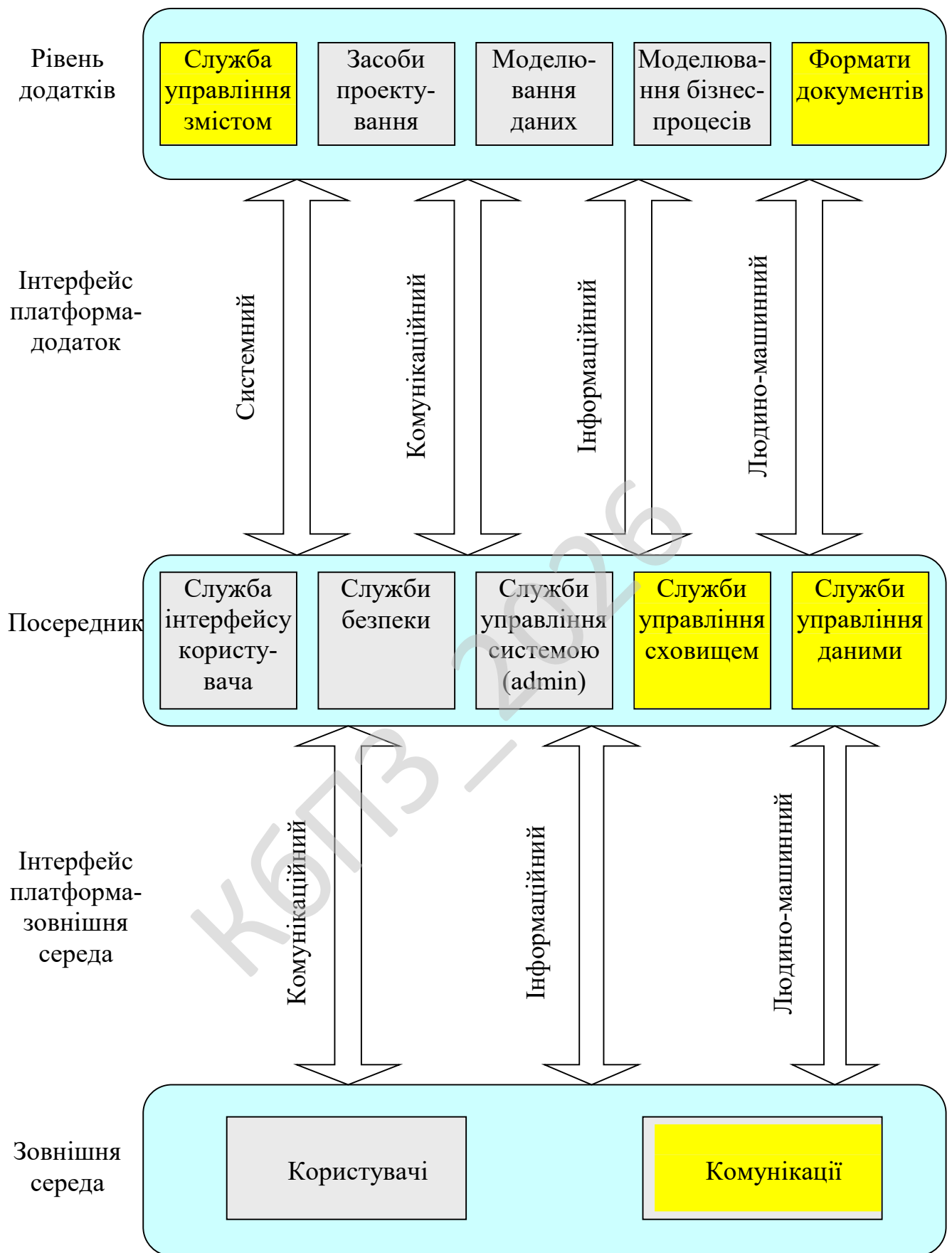


Рисунок 3.2 – Функціональна схема системи

Результатом розробки функціонального стандарту є набір вимог до елементів відкритої СУЕД банку, розроблений відповідності з ДСТУ 10000-1-99 «Інформаційна технологія. Основи й таксономія міжнародних функціональних стандартів. Частина 1. Загальні положення й основи документування» у вигляді функціонального стандарту – Профілю середовища відкритої СУЕД банку.

Проведемо дослідження особливості функціонування елементів системи управління електронним документообігом банку і експериментальним дослідженням з визначення ефективного формату файлових об'єктів СУЕД банку, як засобу підвищення техніко-економічних характеристик СУЕД банку.

Виходячи з вимог до функціонала систем управління й аналізу існуючих СУЕД банку, у системах управління виділені типові елементи (ТЕ):

- ТЕ «обробки інформації».
- ТЕ «передачі інформації».
- ТЕ «сполучна лінія».
- ТЕ «масив зберігання інформації».
- ТЕ «точка діалогу».

Дані типові елементи служать для опису інформаційної системи й інформаційних потоків у будь-якій системі, що неможливо зробити через описані вище служби еталонної моделі середовища відкритої СУЕД банку.

Таким чином показано, що через подібний базис типових елементів може бути представлена кожна, як завгодно складна інформаційна система.

Для проведення функціональної стандартизації ТЕ здійснемо перехід від ТЕ до служб СУЕД банку.

Для перевірки експериментальним шляхом залежності ефективності роботи типових елементів СУЕД банку від форматів використовуваних файлових об'єктів і вибору стандартів для відповідного розділу профілю розроблені:

- вимірювальна установка для виміру часу мережної затримки;
- програма статистичного аналізу файлових об'єктів СУЕД банку;

– методики проведення вимірів часу мережної затримки й розмірів файлів у сховище СУЕД банку.

Визначено співвідношення розмірів файлів залежно від типу інформації у файлах і їхніх розмірах. Визначено формати, у яких розмір файлу виходить мінімальним (таблиця 3.1). Визначено, як саме впливає формат файлових об'єктів (ФФО) на роботу служб СУЕД банку. На рисунку 3.3, на моделі СУЕД банку показані служби, на роботу яких впливає формат файлових об'єктів. Ці служби виділені жовтим кольором.

Результати досліджень дозволили визначити залежність розмірів файлових об'єктів від типу їхнього вмісту й залежність часу мережної затримки, що підтвердило, що швидкодія системи залежить від розміру файлових об'єктів СУЕД банку.

Таблиця 3.1 – Співвідношення типу інформації у файлах і їхніх розмірах

	Тест	Таблиці	Графіка	Формати, у яких розмір файлу виходить мінімальним
1.	100%	0	0	PDF, DOC
2.	80%	0	20%	PDF, DOC
3.	80%	20%	0	PDF, XML
4.	60%	0%	40%	PDF, XML
5.	60%	40%	0%	PDF, XML
6.	40%	0%	60%	XML, PDF
7.	40%	60%	0%	XML, PDF
8.	20%	0%	80%	XML, PDF
9.	20%	80%	0%	XML, PDF
10.	0%	0%	100%	XML, PDF
11.	0%	100%	0%	PDF, XML

Також дослідження дозволило підтвердити залежність ефективності роботи СУЕД банку від ФФО й визначити ефективний формат файлових об'єктів СУЕД банку, що дозволяє мінімізувати необхідні для роботи СУЕД банку комунікаційні й обчислювальні ресурси, що дозволило гармонізувати стандарти файлових об'єктів у функціональному стандарті (профілі) середовища відкритої системи управління електронним документообігом банку і підвищити ефективність роботи ТЕ.

Таким чином показано, що залежно від структури документів є ефективні можливості по поліпшенню роботи СУЕД банку залежно від сфери її застосування.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3.

З рисунку видно, що на початку роботи програми запускається процес початку/кінця роботи програми.

Він взаємодіє з наступними двома процесами:

- Процес введення документів.
- Процес управління базою даних.

Процес введення документів взаємодіє з процесом перетворення документів.

Процес перетворення документів у свою чергу взаємодіє з процесом інтеграції.

Процес інтеграції взаємодіє з процесом створення СУЕД банку.

Останній процес, у свою чергу, взаємодіє з процесом управління базою даних.

Процес управління базою даних взаємодіє з наступними процесами:

- Процесом реалізації служби файлових сховищ.
- Процесом реалізації агенту реплікації.

- Процесом реалізації серверу сеансів.
- Процесом реалізації служби workflow. Workflow – це повна або часткова автоматизація бізнес-процесу, при якій документи, інформація або завдання передаються від одного учасника (бізнес-процесу) до іншого для виконання дій відповідно до набору керівних правил.
- Процесом початку/кінця роботи програми.

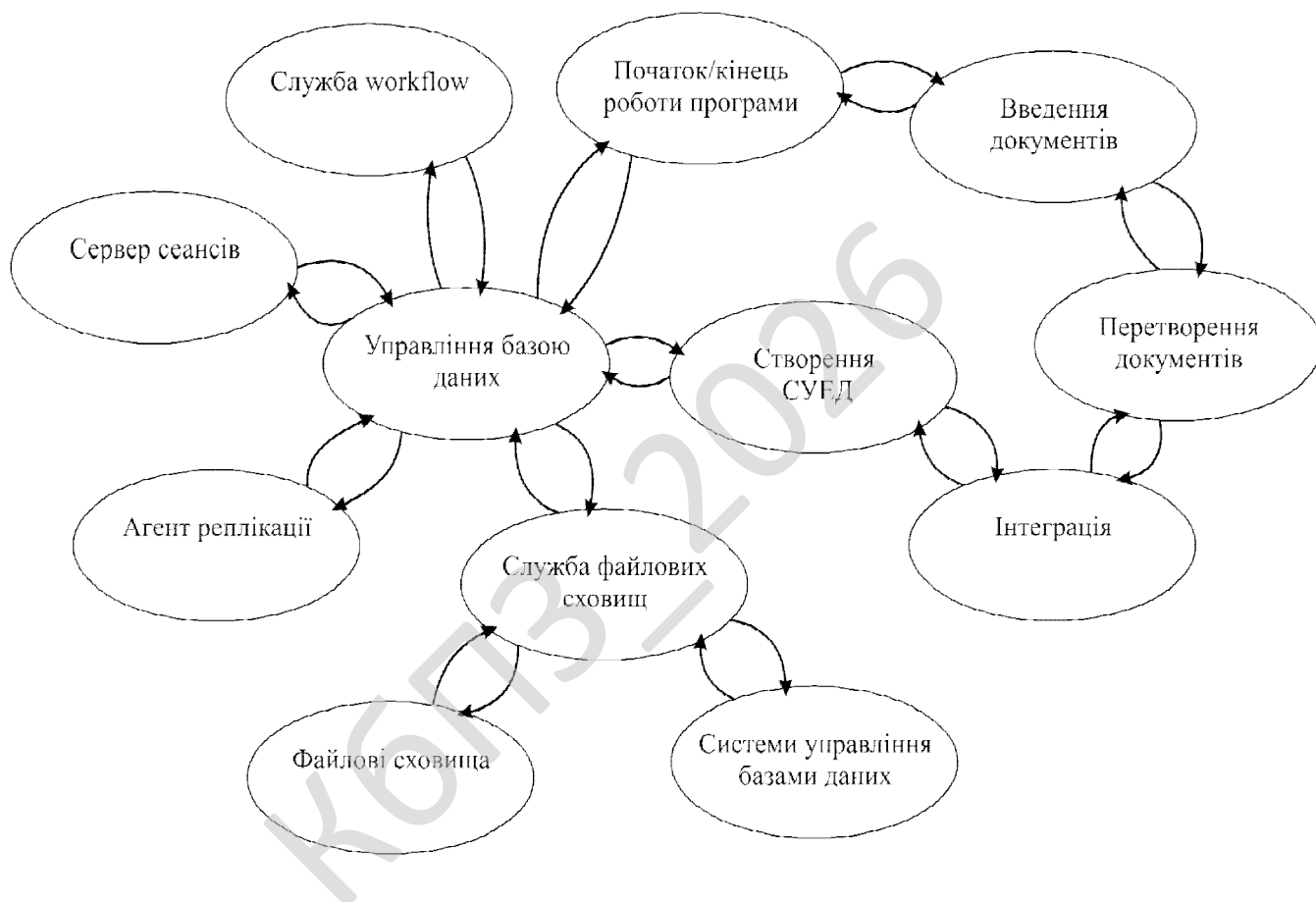


Рисунок 3.3 – Діаграма процесів системи

Процес реалізації служби файлових сховищ взаємодіє з наступними процесами:

- Процесом реалізації файлових сховищ.
- Процесом реалізації системи управління базами даних.

Процес управління базою даних є кінцевим процесом, тому саме він взаємодіє з процесом початку/кінця роботи програми.

Розглянувши у цьому розділі структурну схему розробленої відкритої системи обміну даними, структурну схему системи управління електронної документації, функціональну схему, діаграму взаємодії процесів, перейдемо до опису блок-схеми алгоритмів розробленого, у результаті виконання бакалаврської роботи, програмного забезпечення системи управління електронним документообігом банку.

К6ПЗ_2026

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схема алгоритму роботи основної програми зображена на рисунку 4.1. Після запуску програми на екран виводиться вікно авторизації користувача програми.

Користувач вводить логін та пароль у виведеному для цього вікні авторизації.

Відбувається перевірка введеного логіну та паролю.

Якщо обліковий запис не існує, то виводиться повідомлення про помилку, й система переходить у режим чекання введення логіну та паролю, який дозволить увійти у систему.

Якщо ж логін та пароль є легітимними, то відбувається перевірка, під якими правами зайшов користувач. При цьому існують наступні права користувача:

- Права адміністратора.
- Права користувача.

Якщо користувач отримав права адміністратора, то відбувається наступна послідовність дій:

- Виводиться вікно адміністратора.
- Створюються, редагуються та видаляються документи.
- Формуються звіти.
- Відбувається перегляд історії надходження документів.
- Відбувається перегляд історії виконань документів.

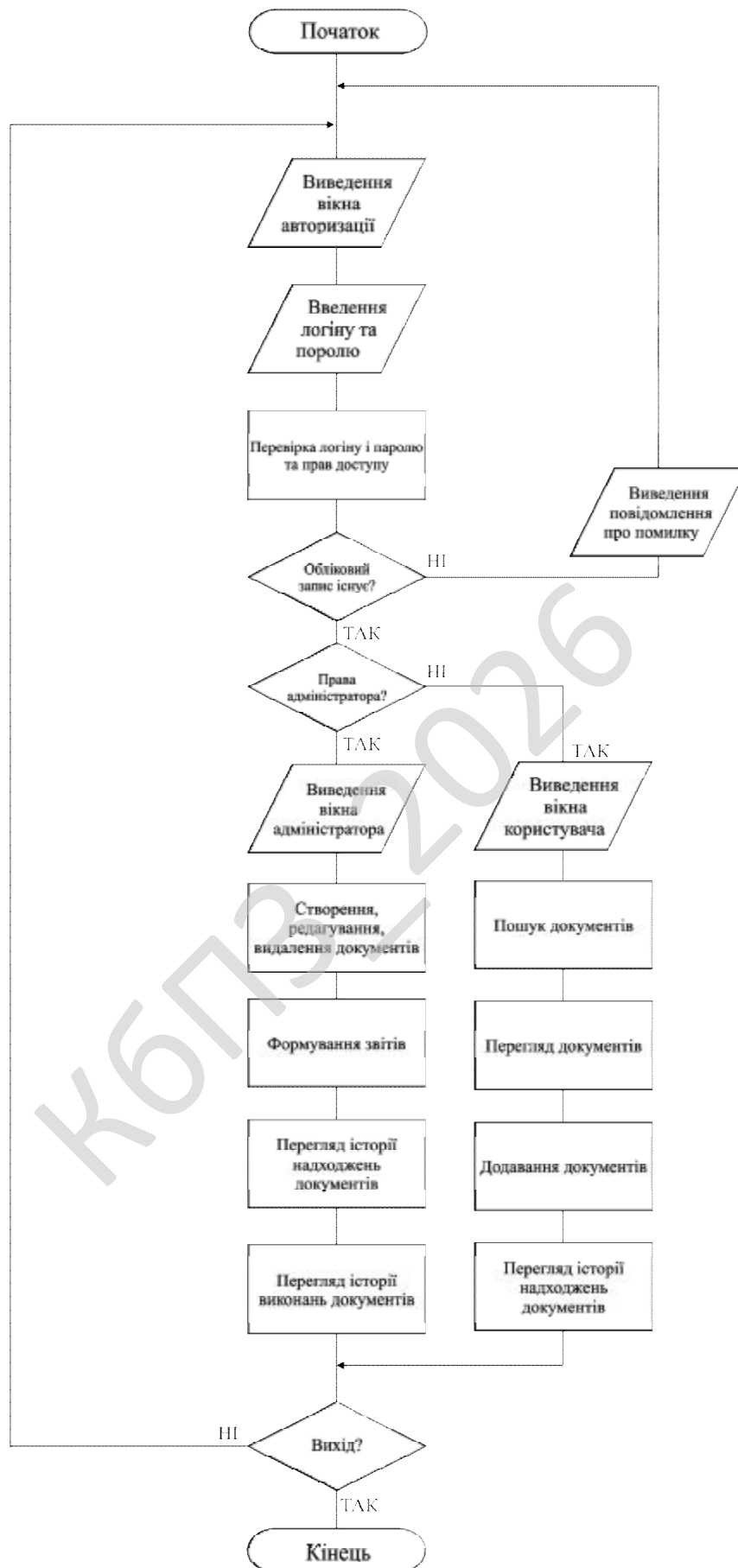


Рисунок 4.1 – Блок-схема алгоритму роботи основної програми

Якщо користувач отримав не права адміністратора, то відбувається наступна послідовність дій:

- Виводиться вікно користувача.
- Відбувається пошук документів.
- Відбувається перегляд документів.
- Відбувається додавання документів.
- Відбувається перегляд історії надходження документів.

Після цього користувач обирає працювати йому далі з системою, або ні.

Якщо він обирає, що працювати, тоді відбувається перехід у початок програми, до завантаження вікна авторизації.

У іншому випадку програма закінчує свою роботу.

Блок-схема алгоритму роботи підпрограми пошуку документів зображена на рисунку 4.2.

З неї ми бачимо, що спершу відбувається виведення вкладки пошуку документів.

Після цього відбувається підключення бази даних.

Наступним кроком є введення користувачем ключової фрази для пошуку документів.

При цьому вибирається категорія, у якій потрібно здійснити пошук.

Коли користувачем введена ключова фраза, та вибрана категорія у якій потрібно вести пошук, запускається процес пошуку.

Для цього спершу відбувається перехід до вказаного розділу.

Після цього, у цьому розділі, відбувається пошук документу за вказаною фразою.

Якщо пошук не знайшов документів, які відповідають заданим параметрам пошуку, то виводиться вікно про те, що такі документи з заданими параметрами не знайдені.

Після цього пропонується повторити пошук.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

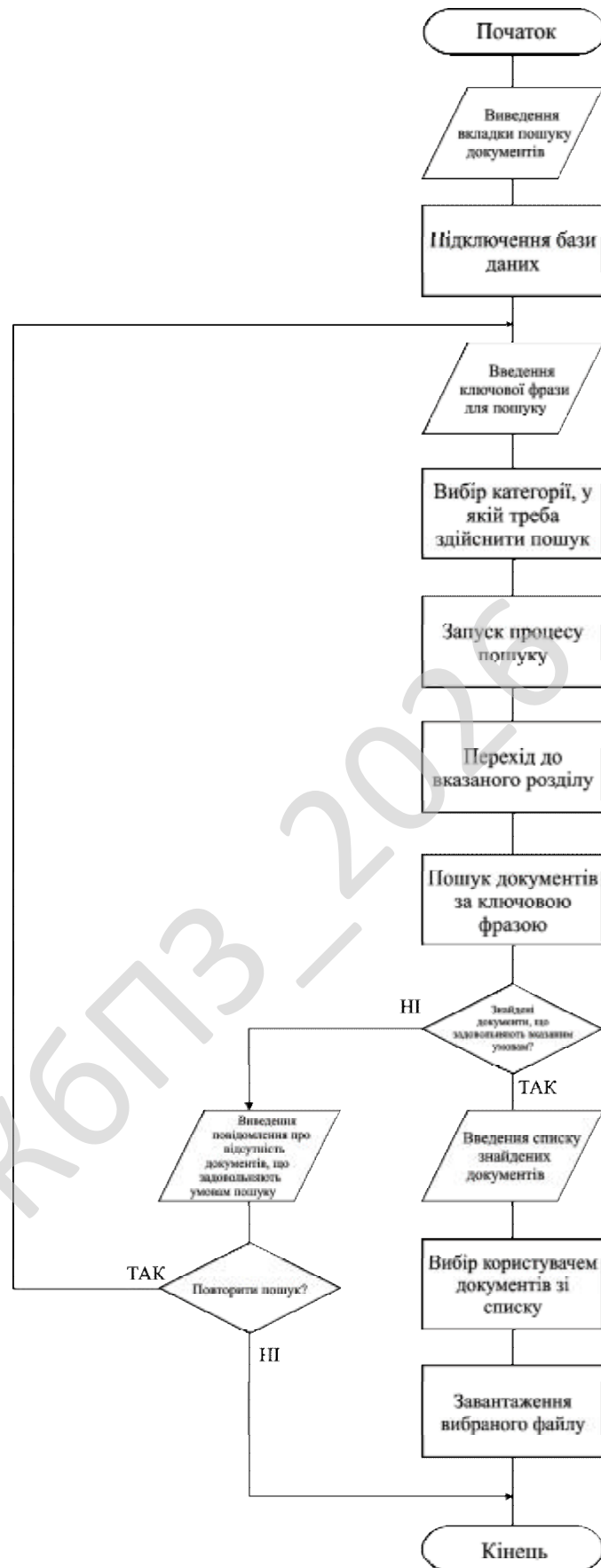


Рисунок 4.2 – Блок-схема алгоритму роботи підпрограми пошуку документів

Якщо користувач обирає вибір повтору пошуку, то переходить до виведення вікна пошуку, та його параметрів.

Якщо пошук знайшов документи, які відповідають заданим параметрам пошуку, то виводиться вікно, у якому відображено список знайдених документів.

Після цього користувач обирає документи з цього списку.

Кінцевим етапом підпрограми пошуку є завантаження вибраного файлу.

Приведемо частину програмного коду, яка реалізує функції головного вікна.

```

/*****/
// NAME:      CWorker
// DESCRIPTION:  Конструктор класу
// INPUT:      N/D
/*****/
CWorker::CWorker(void)
{
    listDoc = gcnew ArrayList();
    logInput = gcnew CLogger();
    logOutput = gcnew CLogger();
}
/*****/
// NAME:      GetIndexByISBNHash
// DESCRIPTION:  Функція одержання індексу документа в загальному списку по
                вхідному номеру документа
// INPUT:      ddHashValue - значення хеша вхідного номера документа
/*****/
Int32 CWorker::GetIndexByISBNHash(Int32 HashValue)
{
    // Цикл пошуку документа із заданим вхідним номером документа
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Хеш-значення вхідного номера документа заданого документа збігається з
        // хеш- значенням вхідного номера документа знайденого документа?
        if(HashValue == ((CDoc^)listDoc[i])->GetISBNHash())
        {
            // Документ знайдений, повернути індекс
            return i;
        }
    }
    // Документ не знайдений, повернути -1
}
```

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

```

return -1;
}
/*****/
// NAME:                AddDoc
// DESCRIPTION:          Функція додавання нового документа
// INPUT:                SourceDoc - об'єкт "документ" для включення в список
// OUTPUT:               TRUE - документ успішно доданий
//                      FALSE - такий документ уже існує в системі
/*****/
Boolean CWorker::AddDoc(CDoc^ SourceDoc)
{
    // Документи з таким вхідним номером документа не існує?
    if(this->GetIndexByISBNHash(SourceDoc->GetISBNHash()) == -1)
    {
        // Додати заданій документ у загальний список
        listDoc->Add(SourceDoc);
        // Записати подія в лог
        logInput->WriteEvent(
            "Новий документ '" + ((CDoc^)SourceDoc)->GetName() +
            "', Автор '" + ((CDoc^)SourceDoc)->GetAuthor() +
            "', Вхідний номер документа '" + ((CDoc^)SourceDoc)->GetISBN());
        // Функція відновила успішно
        return true;
    }
    else
    {
        // Документ із таким же вхідним номером документа існує, повернути помилку
        return false;
    }
}
/*****/
// NAME:                RemoveDoc
// DESCRIPTION:          Функція видалення документів із системи
// INPUT:                ddISBNHash - значення хеша вхідного номера документа
// OUTPUT:               0 - видалення зроблене
//                      1 - видалення неможливо, не всі екземпляри перебувають у
//                      системі
//                      2 - документ із заданим вхідним номером документа не
//                      знайдений
/*****/
Int32 CWorker::RemoveDoc(Int32 ddISBNHash)
{

```

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

```

// Знайти документ по вхідному номеру документа
Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
// Документ із таким вхідним номером документа існує?
if(ddIndex != -1)
{
    // Перевірити, що жодного документа немає в користувача
    if(((CDoc^)listDoc[ddIndex])->GetTotalNumber() ==
        ((CDoc^)listDoc[ddIndex])->GetFreeNumber())
    {
        // Записати подія в лог
        logOutput->WriteEvent(
            "Виконаний документ '" + ((CDoc^)listDoc[ddIndex])->GetName() +
            "', Автор '" + ((CDoc^)listDoc[ddIndex])->GetAuthor() +
            "', Вхідний номер документа '" + ((CDoc^)listDoc[ddIndex])->
            >GetISBN());
        // Видалити знайдений документ зі списку
        listDoc->RemoveAt(ddIndex);
        // Функція відновила успішно
        return 0;
    }
    else
    {
        // Повернути помилку, не всі документи перебувають у системі
        return 1;
    }
}
else
{
    // Повернути помилку, документ із таким вхідним номером документа не був
    знайдений
    return 2;
}
}

/*****/
// NAME: GiveDoc
// DESCRIPTION: Функція видачі документів користувачеві
// INPUT: listDoc - список документів користувача (для виконання
додавання)
// ddISBNHash - хеш вхідного номера документа, що потрібно
одержати
// OUTPUT: TRUE - операція пройшла успішно

```

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

```
//                                     FALSE - у наявності немає жодного екземпляра заданого
                                     документа
/*****/
Boolean CWorker::GiveDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
    // Зменшити кількість вільних документів
    if((ddIndex != -1) && ((CDoc^)listDoc[ddIndex])->DecFreeNumber())
    {
        // Додати документ у список користувача
        listReader->Add(listDoc[ddIndex]);
        // Функція відновила успішно
        return true;
    }
    else
    {
        // Такого документа не є в наявності
        return false;
    }
}
/*****/
// NAME:                            TakeDoc
// DESCRIPTION:                      Функція прийняття документів від користувача назад у
                                     систему
// INPUT:                            // listDoc - список документів користувача (для виконання
                                     видалення)
//                                   ddISBNHash - хеш вхідного номера документа, що потрібно
                                     повернути
/*****/
void CWorker::TakeDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа (документ існує, тому що його
    вхідний номер документа був повідомлений користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
    // Видалити документ зі списку користувача
    // listReader->RemoveAt(ddIndex);
    // Збільшити кількість вільних документів
    ((CDoc^)listDoc[ddIndex])->IncFreeNumber();
}
/*****/
// NAME:                            LoadDocList
```

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

```

// DESCRIPTION:          Функція завантаження документів системи з файлу
// INPUT:                N/D
/*****/

void CWorker::LoadDocList()
{
    // Перевірити існування файлу
    if(!File::Exists(DOCS_FILE))
    {
        // Файл не знайдений, закінчити виконання функції
        return;
    }
    // Відкрити файл
    StreamReader^ srDoc = gcnew StreamReader(DOCS_FILE);
    // Продовжувати, поки не буде знайдений кінець файлу
    while(srDoc->EndOfStream == false)
    {
        // Створити новий об'єкт "документ"
        CDoc^ NewDoc = gcnew CDoc();
        // Завантажити з файлу й установити параметри документа
        NewDoc->SetName(srDoc->ReadLine());
        NewDoc->SetAuthor(srDoc->ReadLine());
        NewDoc->SetISBN(srDoc->ReadLine());
        NewDoc->SetTheme(srDoc->ReadLine());
        NewDoc->SetPages(Convert::ToInt32(srDoc->ReadLine()));
        NewDoc->SetTotalNumber(Convert::ToInt32(srDoc->ReadLine()));
        NewDoc->SetFreeNumber(Convert::ToInt32(srDoc->ReadLine()));
        // Включити документ у загальний список документів
        listDoc->Add(NewDoc);
    }
    // Закрити файл
    srDoc->Close();
}
/*****/

// NAME:                SaveDocList
// DESCRIPTION:          Функція збереження документів системи у файл
// INPUT:                N/D
/*****/

void CWorker::SaveDocList()
{
    // Перевірити існування файлу
    if(File::Exists(DOCS_FILE))
    {

```

```

        // Видалити файл
        File::Delete(DOCS_FILE);
    }
    // Створити й відкрити файл
    StreamWriter^ swDoc = gcnew StreamWriter(DOCS_FILE);
    // Цикл запису по одному документу
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Одержати параметри документа й записати їх у файл
        swDoc->WriteLine(((CDoc^)listDoc[i])->GetName());
        swDoc->WriteLine(((CDoc^)listDoc[i])->GetAuthor());
        swDoc->WriteLine(((CDoc^)listDoc[i])->GetISBN());
        swDoc->WriteLine(((CDoc^)listDoc[i])->GetTheme());
        swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetPages()));
        swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetTotalNumber()));
        swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetFreeNumber()));
    }
    // Закрити файл
    swDoc->Close();
}
/*****/
// NAME:                FindDoc
// DESCRIPTION:          Функція пошуку документа по заданих параметрах
// INPUT:                strFindDoc - рядок для пошуку
/*****/
ArrayList^ CWorker::FindDoc(String^ strFindValue)
{
    // Список для результату
    ArrayList^ listResult = gcnew ArrayList();
    // Шукане значення без обліку регістра
    String^ strValue = strFindValue->ToLower();
    // Задана порожній рядок?
    if(String::IsNullOrEmpty(strValue->Trim()))
    {
        //Вивести весь список
        for(Int32 i = 0; i < listDoc->Count; i++)
        {
            // Додати документ у результуючий список
            listResult->Add(listDoc[i]);
        }
    }
    else

```

```

{
    // Вибрати з умовою
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Пошук рядка в кожному полі без обліку регістра
        if((((CDoc^)listDoc[i])->GetName())->ToLower())->IndexOf(strValue) != -
1 ||
            (((CDoc^)listDoc[i])->GetAuthor())->ToLower())->IndexOf(strValue) !=
-1 ||
            (((CDoc^)listDoc[i])->GetISBN())->ToLower())->IndexOf(strValue) != -
1 ||
            (((CDoc^)listDoc[i])->GetTheme())->ToLower())->IndexOf(strValue) !=
-1 ||
            Convert::ToString((((CDoc^)listDoc[i])->GetPages()))-
>IndexOf(strValue) != -1)
        {
            // Додати документ у результуючий список
            listResult->Add(listDoc[i]);
        }
    }
}
// Повернути список збірів
return listResult;
}

/*****/
// NAME: ViewDoc
// DESCRIPTION: Функція перегляду інформації про заданий документ
// INPUT: ddISBNHash - хеш вхідного номера документа, інформацію
           про яку треба переглянути
/*****/
CDoc^ CWorker::ViewDoc(Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа (документ існує, тому що його
    вхідний номер документа був повідомлений користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
    // Повернути елемент "документ" із загального списку
    return (CDoc^)listDoc[ddIndex];
}

/*****/
// NAME: ReadLogs
// DESCRIPTION: Функція зчитування історії з файлу
// INPUT: N/D

```

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

```

/*****/
void CWorker::ReadLogs()
{
    // Файл із історією надходжень існує?
    if(File::Exists(LOG_INPUT))
    {
        // Зчитування історії надходжень
        logInput->strLog = File::ReadAllText(LOG_INPUT);
    }
    else
    {
        logInput->strLog = "";
    }
    // Файл із історією списань існує?
    if(File::Exists(LOG_OUTPUT))
    {
        // Зчитування історії списань
        logOutput->strLog = File::ReadAllText(LOG_OUTPUT);
    }
    else
    {
        logOutput->strLog = "";
    }
}

/*****/
// NAME: WriteLogs
// DESCRIPTION: Функція збереження історії у файл
// INPUT: N/D
/*****/
void CWorker::WriteLogs()
{
    // Запис історії
    File::WriteAllText(LOG_INPUT, logInput->strLog);
    File::WriteAllText(LOG_OUTPUT, logOutput->strLog);
}

/*****/
// NAME: ClearLogs
// DESCRIPTION: Функція очищення історії
// INPUT: N/D
/*****/
void CWorker::ClearLogs()
{

```

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

```

logInput->strLog = "";
logOutput->strLog = "";
}
/*****/
// NAME:          ViewInputLog
// DESCRIPTION:    Функція перегляду історії надходжень
// INPUT:          N/D
/*****/
String^ CWorker::ViewInputLog()
{
    return logInput->strLog;
}
/*****/
// NAME:          ViewOutputLog
// DESCRIPTION:    Функція перегляду історії списань
// INPUT:          N/D
/*****/
String^ CWorker::ViewOutputLog()
{
    return logOutput->strLog;
}
/*****/

```

Взаємозв'язок класів які використані при реалізації програми, наведено на рисунку 4.3.

Проведемо оцінку якості роботи системи управління електронним документообігом банку.

В [2] показано основні переваги застосування теорії нечітких множин для рішення завдання оцінки ефективності роботи СУЕД банку у порівнянні із традиційними підходами теорії автоматичного управління.

Розроблено критерії оцінки якості роботи системи управління електронним документообігом банку. Показано архітектуру нечіткої моделі управління якістю роботи СУЕД банку. Основним модулем системи оцінки якості роботи СУЕД банку є блок прийняття рішень, хоча інші блоки не менш важливі для нормального функціонування моделі.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

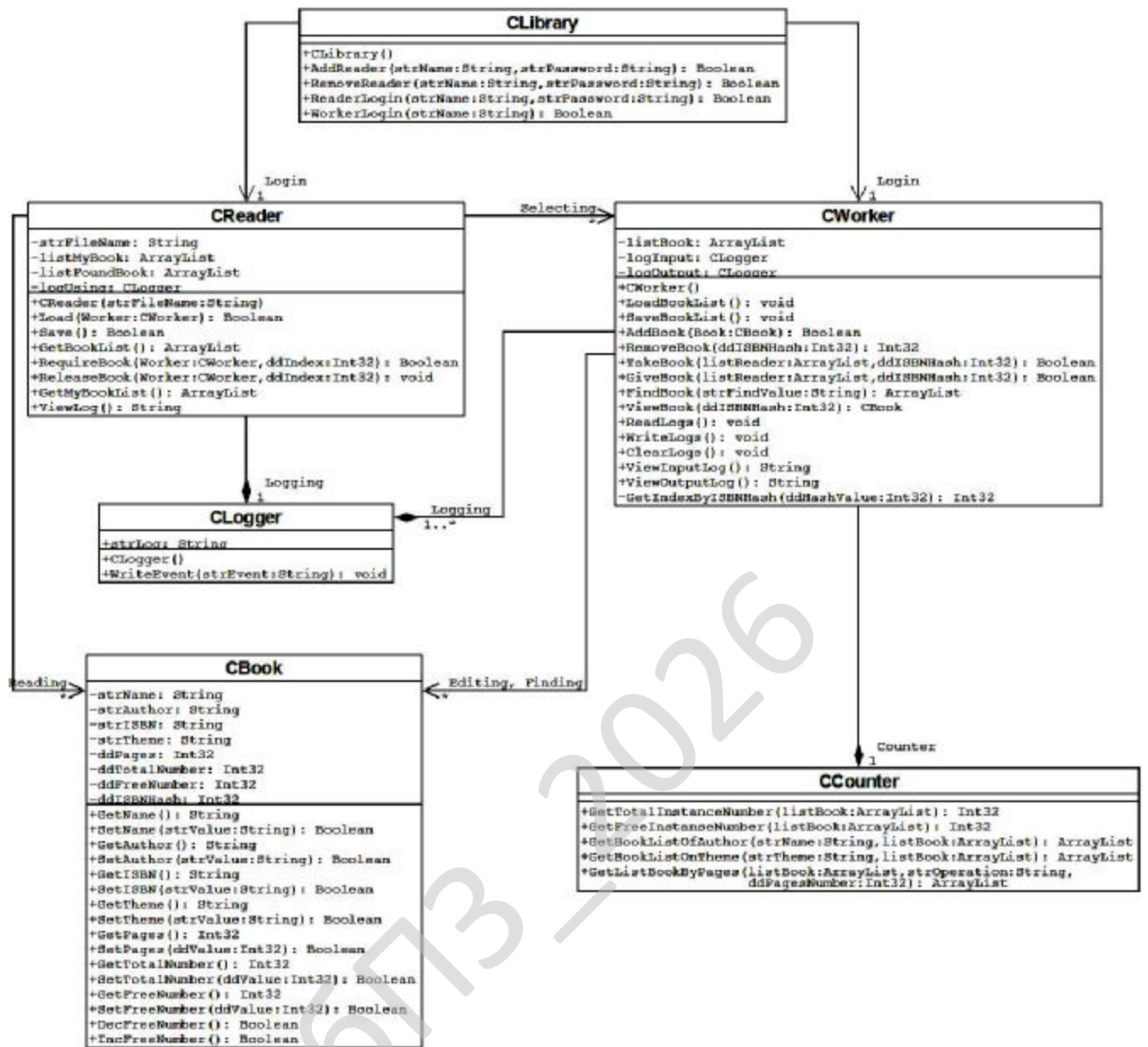


Рисунок 4.3 – Class_Diagramm

Блок оцінки станів, на основі вступник на його вхід інформації, буде формалізований опис ситуації, що виникла при роботі СУЕД банку.

У блоці видачі керуючих впливів здійснюється перехід від внутрішньої форми завдання керуючих рішень до зовнішньої форми.

Для оцінки якості роботи СУЕД банку скористаємося критерієм, вираженим безліччю М, елементи якого є інтервалами якісної (нечіткої) шкали:

$M=\{\text{«дуже добре»}, \text{«добре»}, \text{«задовільно»}, \text{«незадовільно»}\}$, для цього методом експертних оцінок були визначені конкретні значення процентного вмісту тексту, графіки й таблиць у файлах певного формату.

Виходячи з аналізу експериментальних даних, був отриманий масив $\{W_N(X)\}$, що містить інтервали значень фізичної величини, що характеризує ефективність роботи елемента, при яких забезпечується ефективна робота елементів пристрою (системи) і масив $\{WW_N(X)\}$, що містить інтервали значень фізичної величини, що характеризує функціонування елемента, і значення функції приналежності стану елемента системи до одного з можливих станів. Розроблено вирішальні правила виходячи зі значень функції приналежності стану елемента одному з можливих станів для кожного з інтервалів базових значень.

Для перевірки запропонованих правил була зроблена побудова моделі оцінки якості роботи системи управління електронним документообігом банку у середовищі MatLab для СУЕД банку фінансової банку. Модель заснована на методі Мамдани, що має у своїй основі базу знань у вигляді сукупності нечітких предикатних правил виду:

П1: якщо $x \in A1$, тоді $z \in B1$,

П2: якщо $x \in A2$, тоді $z \in B2$,

.....

Пn: якщо $x \in An$, тоді $z \in Bn$,

З метою перевірки даних про те, що використовувані критерії можуть застосовуватися для діючих систем проведений ряд експериментів–інструментальних досліджень на діючій системі, для чого якого були обмірювані й визначені величини:

- кількість переданих у СУЕД банку файлів;
- формати переданих у СУЕД банку файлів.

Отримано підтвердження того, що використовувані для оцінки ефективності роботи СУЕД банку критерії можуть застосовуватися для діючих

систем. Розроблено методику оцінки ефективності роботи елементів СУЕД банку. Зроблено оцінку ефективності й повернення інвестицій від впровадження відкритої системи управління електронним документообігом банку.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI_91. Механізм алгоритму LOKI_91 подібний DES (рисунок 4.4).

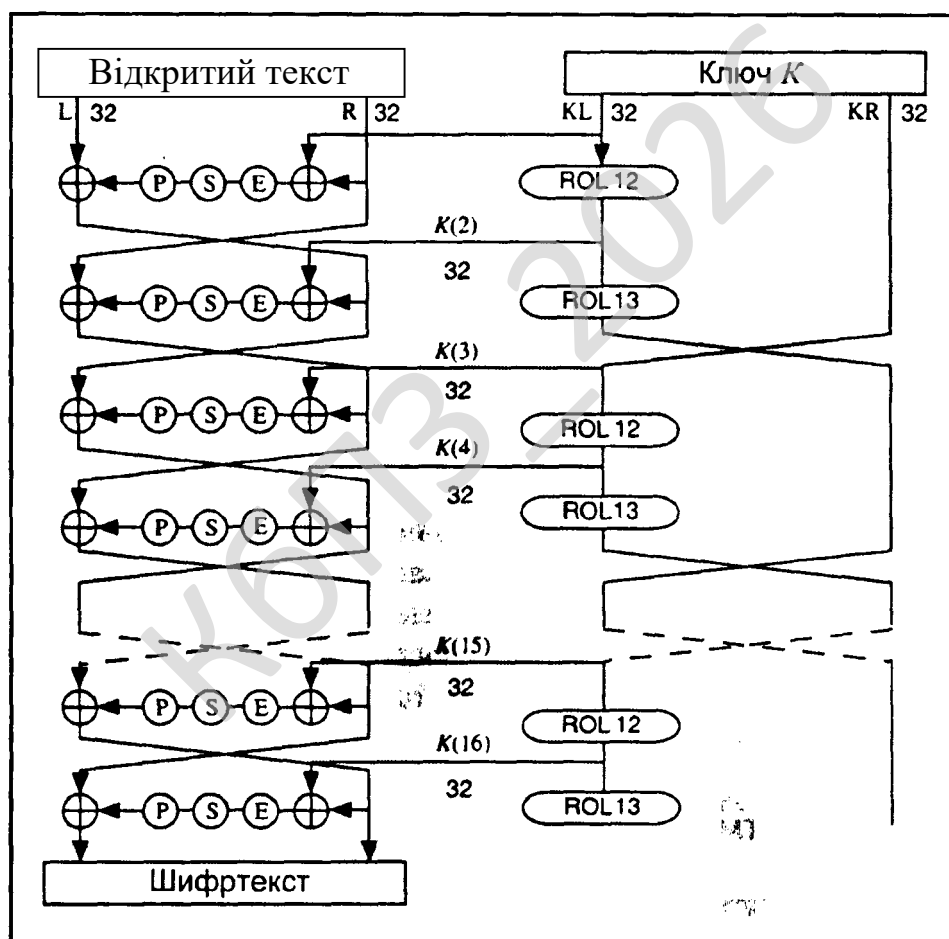


Рисунок 4.4 – Алгоритм LOKI91

Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку

піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера r , а вісім внутрішніх біт утворюють номер c . Вихід S-блоку, O , має наступне значення: $O(r,c) = (c + ((r*17) \oplus 0\text{xff})^{31} \bmod P_r$.

Таблиця 4.2 – Значення P_r

r	1	2	3	4	5	6	7	8
P_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
P_r	433	445	451	463	471	477	487	499

Після цього чотири 8-бітових результати знову поєднуються, утворюючи 32-бітове число, що піддається операції перестановки, описаній в таблиці 3. Нарешті, для одержання нової лівої половини виконується операція XOR правої половини з колишньою лівою половиною, а ліва половина стає новою правою половиною. Після 16 раундів для одержання остаточного шифртексту знову виконується операція XOR над блоком і ключем.

Таблиця 4.3 – Перестановка за допомогою Р-блоку

32	24	16	8	31	23	15	7	30	22	14	6	29	21	13	5
2	20	12	4	27	19	11	3	26	18	10	2	25	17	9	1

Підключи генеруються із ключа досить прямолінійно. 64-бітовий ключ розбивається на ліву й праву половини. На кожному раунді підключем служить ліва половина. Далі вона циклічно зрушується вліво на 12 або 13 біт, потім після кожних двох раундів ліва й права половини міняються місцями. Як і в DES, для зашифрування й розшифрування використовується один й той самий алгоритм із деякими змінами у використанні підключів.

КБПЗ_2026

					ВКРБ-122.26.0021.00.00.ПЗ		Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			52

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Головне вікно (рис. 1) програмного забезпечення системи інтелектуального обміну даними у електронному документообігу банку є центральним елементом взаємодії користувача із системою.

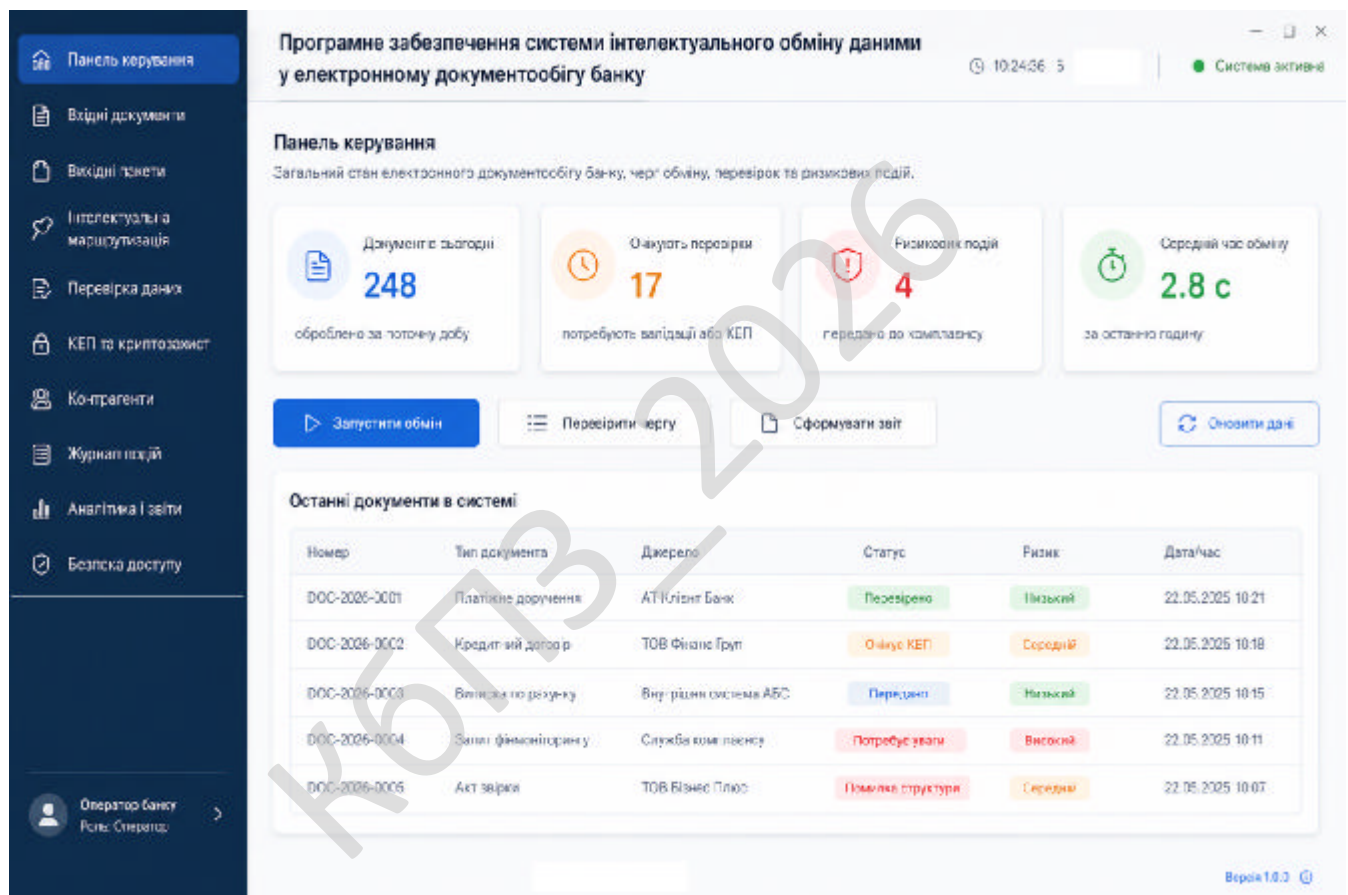


Рисунок 5.1 – Головне вікно

Через це вікно оператор отримує доступ до основних функціональних розділів, переглядає поточний стан документообігу, контролює вхідні та вихідні документи, відстежує результати перевірки даних, роботу електронного підпису, події безпеки та службову аналітику.

У головному вікні передбачено такі основні розділи: Панель керування; Вхідні документи; Вихідні пакети; Інтелектуальна маршрутизація; Перевірка даних; КЕП та криптозахист; Контрагенти; Журнал подій; Аналітика і звіти; Безпека доступу.

Також у головному вікні відображаються верхня інформаційна панель із назвою системи, поточним часом і статусом роботи, короткі інформаційні картки зі станом документообігу, область швидких дій, таблиця останніх документів та інформація про поточного користувача.

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

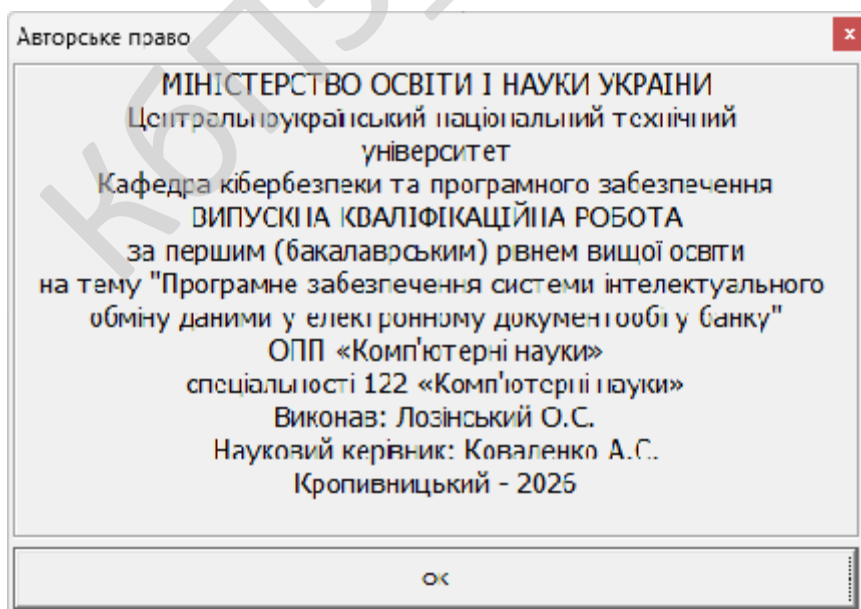


Рисунок 5.2 – Авторське право

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи інтелектуального обміну даними у електронному документообігу банку.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем інтелектуального обміну даними у електронному документообігу банку.
- Досліджена система інтелектуального обміну даними у електронному документообігу банку.
- На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуального обміну даними у електронному документообігу банку.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуального обміну даними у електронному документообігу банку.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Visual C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи інтелектуального обміну даними у електронному документообігу банку.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм LOKI_91.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
2. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
3. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
4. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
5. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
6. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
7. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
8. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
9. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
10. Вінтенко Б.Ю., Миронець І.В., Смірнов О.А. «Модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *IV Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2025)»* м.Черкаси 25 листопада 2025 року – Черкаси: ЧДТУ.– 2025. – С.101-103.

11. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

12. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427, 2025.

13. Смірнова, Т.В. «Дослідження методів, моделей та сучасних ІТ-рішень для підтримки технологічних процесів у критичній інфраструктурі держави». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С.195-208, 2025.

14. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

15. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

16. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуї А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки*. 2025. Вип. 11(42), ч. II. С.52-62.

17. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В. «Методи забезпечення відмовостійкості інтелектуальних систем підтримки

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

оператора». *VIII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”*, м. Кропивницький. 24-25 квітня 2025 р. – Кропивницький: ЦНТУ. – 2025. – С. 44-46.

18. Смірнов, О.А., Константинова, Л.В., Коноплицька-Слободенюк, О.К., Козірова, Н.В, Якименко, Н.М., Доренський, О.П., Буравченко, К.О. «Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних». *Кібербезпека: освіта, наука, техніка*. 2025. №3(27), С. 429–448.)

19. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings Volume 3664*, 2024, Pages 11-23.

20. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Коноплицька-Слободенюк, О., Смірнова, Т., Константинова, Л. «Дослідження застосування систем підтримки оперативного персоналу об’єкту критичної інфраструктури при керуванні енергоблоком АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2024. № 2(26), С. 6-26.

21. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

22. Kuznetsov O., Ilchenko O., Kryvinska N., Buravchenko K., Smirnov O., Savchenko Iu. «An Empirical Assessment of Leading Blockchain Financial Services». *2023 IEEE 1st Ukrainian Distributed Ledger Technology Forum (UADLTF)*, Kyiv, Ukraine, 2023, pp. 1-6,

23. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

24. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

25. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

26. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

27. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

28. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів ІЕС60880 та ІЕС62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

29. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

30. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

					ВКРБ-122.26.0021.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

31. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.

32. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція “Інформаційна безпека та комп'ютерні технології”, м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

33. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

34. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

35. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

36. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

37. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

38. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

39. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

40. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

41. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

42. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

43. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

44. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

45. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

46. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

47. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobayev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

48. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

49. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

50. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.