НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК 004.4_____

«До захисту допущено»
Завідувач кафедри
_____ Наталія АУШЕВА
“___”_____2023 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему Засоби інтеграції систем електронного документообігу на прикладі сервісу Signy

Виконав: студент 2 курсу, групи ТР-21мп
Узун Артем Сергійович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник професор, доцент, д.т.н. Шушура О.М. _____
(посада, вчене звання, науковий ступінь, ім’я ПРІЗВИЩЕ) (підпис)

Рецензент_____ _____
(посада, вчене звання, науковий ступінь, ім’я ПРІЗВИЩЕ) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2023

Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти другий (магістерський)

За освітньою програмою ""“Цифрові технології в енергетиці”

Спеціальності 122 Комп’ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Наталія АУШЕВА
(підпис)

«____» _____ 2023 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Узуну Артему Сергійовичу
(прізвище, ім’я, по батькові)

1. Тема дисертації Засоби інтеграції систем електронного документообігу на прикладі сервісу Signy

Науковий керівник Шушура О.М., д.т.н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “06” листопада 2023 року №5152-с

2. Строк подання студентом дисертації 18 грудня 2023р

3. Вихідні дані до роботи інформаційна технологія для інтеграції сервісу Signy з іншими системами електронного документообігу

4. Перелік питань, які потрібно розробити аналіз популярних систем електронного документообігу в Україні та можливостей інтеграцій з ними, дослідження існуючих типів та методів інтеграцій комп’ютенх систем, проектування та розробка власного механізму інтеграції, проведення тестування створеного програмного забезпечення

5. Орієнтований перелік ілюстративного матеріалу схеми архітектури системи, схема бази даних, діаграми класів, зразки розробленого інтерфейсу

6. Орієнтований перелік публікацій _XX міжнародна науково-практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», 25-28 квітня, 2023, Київ, Україна._____
_____
_____
_____
_____

7. Консультанти розділів дисертації_____

8. Дата видачі завдання « 24 » жовтня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Аналіз підходів, методів та програмних засобів	24.10.2022 – 10.01.2023	
2	Розробка архітектури бази даних	11.01.2023 – 01.03.2023	
3	Розробка архітектури програмного забезпечення	02.03.2023 – 15.06.2023	
4	Створення програмного забезпечення	16.05.2023 – 23.09.2023	
5	Проведення тестування створеного програмного забезпечення	24.09.2023 – 01.11.2023	
6	Оформлення магістерської дисертації	02.11.2023–18.12.2023	

Студент _____ **Артем УЗУН**
(підпис) (ім'я ПРІЗВИЩЕ)

Науковий керівник _____ **Олексій ШУШУРА**
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Актуальність теми. Збільшення кількості систем електронного документообігу створює виклик, оскільки контрагенти можуть використовувати різні платформи, що призводить до ускладнень у процесі обробки електронних документів та подовжує час виконання рутинних завдань. Заради подолання цієї проблеми необхідно розглядати можливість створення інтеграцій між різними системами. Це дозволить автоматизувати та полегшити обмін даними між платформами, зменшити час обробки електронних документів та сприяти більш ефективній роботі всього електронного документообігу. Інтеграція систем забезпечить єдиною точкою з'єднання для різних платформ, що спростить взаємодію між ними та підвищить загальну продуктивність управління документами.

Метою дослідження є автоматизація процесу обміну документами між користувачами різних систем електронного документообігу.

Завдання дослідження:

- аналіз популярних систем електронного документообігу в Україні та можливостей інтеграцій з ними;
- дослідження існуючих типів та методів інтеграцій комп’ютерних систем;
- на основі аналізу та дослідження спроектувати та розробити власний механізм інтеграції;
- проведення тестування створеного програмного забезпечення.

Об’єкт дослідження – інтеграція систем електронного документообігу.

Предмет дослідження – програмне забезпечення для інтеграції систем електронного документообігу на прикладі сервісу Signy.

Практична цінність отриманих в роботі результатів полягає в наданні можливостей обміну документами між різними системами. Спроектоване програмне забезпечення дає користувачам сервісу Signy змогу працювати з клієнтами з сервісів Вчасно, Document Online, Deals.

Апробація результатів дисертації. Основні положення даної роботи доповідались та обговорювались на:

XX міжнародна науково-практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», 25-28 квітня, 2023, Київ, Україна.

Дисертація складається з вступу, п’яти розділів та висновків. Повний обсяг дисертації складає 97 сторінок, в тому числі 85 сторінок основного тексту, 17 таблиць, 23 рисунків, 3 сторінки списку використаних джерел у кількості 30 найменувань.

Ключові слова: системи електронного документообігу, електронний документ, електронний підпис, інтеграція, мова програмування C#, користувацький інтерфейс, Signy.

ABSTRACT

Relevance of the Topic. The increase in the number of electronic document management systems poses a challenge, as counterparties may use different platforms, complicating the processing of electronic documents and extending the time required for routine tasks. To address this issue, exploring the possibility of creating integrations between various systems is essential. This would automate and streamline data exchange between platforms, reduce processing time for electronic documents, and contribute to the overall efficiency of electronic document management. System integration would serve as a single point of connection for diverse platforms, simplifying their interaction and enhancing the overall productivity of document management.

The aim of the research is to automate the document exchange process among users of different electronic document management systems.

Research Objectives:

- Analyze popular electronic document management systems in Ukraine and integration possibilities;
- Investigate existing types and methods of computer system integration;
- Design and develop a proprietary integration mechanism based on analysis and research;
- Conduct testing of the developed software.

Research Object – Integration of the Signy service with other electronic document management systems.

Research Subject – Software for integration.

Practical Value of the Obtained Results lies in providing document exchange capabilities between different systems. The designed software enables Signy service users to collaborate with clients from services like Vchasno, Document Online, and Deals.

Approval of Dissertation Results. The main findings of this work were presented and discussed at :

XX International scientific and practical conference of young scientists and students "Modern problems of scientific support of power engineering", April 25-28, 2023, Kyiv, Ukraine.

The dissertation consists of an introduction, five chapters, and conclusions. The total volume of the dissertation is 97 pages, including 85 pages of the main text, 17 tables, 23 figures, and 3 pages of the list of references with 30 entries.

Keywords: electronic document management systems, electronic document, electronic signature, integration, C# programming language, user interface, Signy.

ЗМІСТ

ВСТУП

У сучасному світі розвиток технологій дає нам змогу використовувати їх у всіх сферах нашого життя. Особливо це стосується сфери бізнесу та виробництва, де використання сучасних технологій є обов’язковим, бо надає переваги конкурентоспроможності підприємства перед іншими. Гарним прикладом цього є системи електронного документообігу - це відносно нова технологія, якою користуються дуже багато підприємств та приватних осіб. Використання електронного документообігу дозволяє значно скоротити час підписання документів (відповідно і час прийняття рішень) та зменшує зусилля, необхідні для цього.

Особливістю електронного документообігу в Україні є велика кількість систем, які надають відповідні послуги. Для клієнта це дає можливість обрати систему, яка найбільше підходить у його конкретній ситуації, однак це також створює проблему обміну документів, якщо клієнти вирішили використовувати різні системи документообігу. Зазвичай, для вирішення цієї проблеми, один з клієнтів має перейти до використання системи іншого клієнта. Однак це призводить до відмови від використання власної системи, або, в більшості випадків, до використання декількох систем паралельно. Це суперечить основній меті систем електронного документообігу - зручному і швидкому використанню елетронних документів.

Актуальність даної роботи це вирішення вищеописаної проблеми та надання можливості користувачам системи Signy [1] можливості до зручного обміну електронними документами без прикладання великих зусиль.

1 АНАЛІЗ ПРОБЛЕМИ ІНТЕГРАЦІЇ В СИСТЕМАХ ЕДО ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Проблема, яка виникає при роботі з ЕДО

Ще з давніх часів документи володіли важливим значенням у різних аспектах життя, використовувались у законодавстві, торгівлі та взаємовідносинах. З тих пір документообіг виявився в стадії значного розвитку та суворого законодавчого регулювання. Сучасні документи мають класифікацію за призначенням:

- організаційні документи (посадова інструкція, статут, штатний розпис, структура та штатна чисельність, становище);
- розпорядчі документи (наказ, витяг із наказу, рішення, розпорядження, вказівка, постанова);
- інформаційно-довідкові документи (акт, довідка, протокол, доповідна записка, пояснювальна записка, лист, службова записка);
- обліково-розрахункові документи (платіжне доручення, рахунок-фактура, акт, накладна тощо);
- нормативні документи (стандарти, правила, норми, склепіння правил, регламенти тощо);
- інші.

Існують і інші типи класифікацій:

- за часом створення: первинні та вторинні (анотація, реферат, огляд та інші);
- за містом змісту: іконічні (графічні), текстові, ідеографічні (схеми, карти, ноти), мультимедійні, аудіальні;
- за місцем видання: внутрішні, зовнішні;
- у напрямку відправлення: вхідні, вихідні;
- за поширенням: опубліковані, неопубліковані, проміжні;
- за необхідністю технічних засобів: людиночитані, машиночитані;
- за рівнем секретності: несекретні, конфіденційні, секретні, з різним рівнем секретності та інші.

Це дозволяє впорядковано класифікувати та керувати різноманітністю документів в сучасному світі, де вони залишаються ключовим елементом взаємодії та ділової активності.

Сьогодні діяльність будь-якого підприємства пов’язана з документами тому проблеми оптимізації процесу обміну документів стоять дуже гостро, так як від швидкості обробки документів залежить швидкість роботи підприємства. В Україні процес документообігу регламентується інструкцією з діловодства, яка була затверджена Кабінетом Міністрів України №1242 від 30.11.2011. Дана інструкція є обов’язковою для виконання [2].

Розглянемо список проблем які можуть виникати на підприємствах при використанні різних систем документообігу:

1. дублювання даних – використання різних систем може призводити до дублювання даних, оскільки інформацію слід вручну вводити в кожну систему окремо. Це може викликати неузгодженість інформації та збільшувати ймовірність помилок;
2. затримки в обробці документів – ручний обмін даними між системами може призводити до затримок у часі, оскільки необхідно чекати на завершення обміну даними або на їх введення вручну. Це може уповільнювати бізнес-процеси та знижувати продуктивність;
3. ускладнення аналізу та звітності – розділеність систем може ускладнювати аналіз та формування звітності, оскільки необхідно об'єднувати дані з різних джерел. Інтеграція дозволяє автоматизувати цей процес та спрощує виведення аналітичної інформації;
4. брак єдності в управлінні документами – відсутність єдиної точки управління документами може призводити до втрати документів, складнощів у відстеженні їхнього обігу та недостатньої контрольованості над всім документообігом.

Створення інтеграцій між різними системами документообігу є важливою стратегічною необхідністю для підприємств, оскільки це сприяє спланованому та ефективному обміну інформацією між різними підприємствами. Цей процес вносить значні переваги для підвищення продуктивності та спрощення рутинних операцій.

По-перше, інтеграції між системами документообігу забезпечують єдність інформаційного простору, дозволяючи різним компаніям взаємодіяти та обмінюватися даними без зайвих труднощів. Це сприяє уникненню дублювання даних та розбіжностей в інформації.

По-друге, інтеграції дозволяють автоматизувати перенесення даних між системами, що призводить до значного зменшення часу, витраченого на ручне введення та перевірку інформації. Це дозволяє прискорити обробку документів та підвищити ефективність робочих процесів.

По-третє, інтеграції забезпечують єдність стандартів та форматів обміну даними, що полегшує взаємодію між різними системами. Це робить можливим безперервний та гармонійний обмін інформацією без необхідності внутрішньої адаптації.

По-четверте, інтеграції сприяють створенню єдиної точки управління, де можна відстежувати та контролювати потік документів між системами. Це робить процес управління документами більш прозорим та ефективним.

Важливість створення інтеграцій між різними системами документообігу полягає в покращенні співпраці, ефективності та узгодженості в обміні інформацією, що, в свою чергу, сприяє успішному функціонуванню підприємств.

1.2 Електронний документообіг

Електронний документообіг (ЕДО) – це система, яка використовується для автоматизації процесів створення, обміну, підписання, зберігання та обробки документів в електронній формі у великих організаціях та підприємствах. Головна мета систем електронного документообігу полягає в підвищенні продуктивності праці в організаціях, де існує велика кількість нормативних або первинних документів, а також в прискоренні взаємодії між контрагентами та забезпеченні високого рівня захисту даних. Наприклад, до впровадження системи юридично значущого електронного документообігу та його юридичної бази, процес укладання договору між двома організаціями займав значно більше часу. Причиною цього є те, що для доставки документів використовувалася пошта, і отримання документів

отримувачем могло займати декілька днів. У контрасті до цього, завдяки доставці електронних документів через мережу Інтернет, цей процес займає всього кілька секунд. Крім того, використання електронних документів є значно економічно вигіднішим і швидшим порівняно з їхніми паперовими аналогами [3].

Системи електронного документообігу спрямовані на вирішення ряду проблем, пов'язаних із традиційним паперовим обігом документів. Давайте розглянемо детально ці проблеми:

Часові затрати: Електронний документообіг дозволяє миттєво обмінюватися та обробляти документи в онлайн-режимі, скорочуючи час на їхню обробку та пересилання.

Витрати на матеріали: Використання електронних документів зменшує потребу в фізичних матеріалах, таких як папір, а також в енергозатратах на друк та зберігання.

Ризик помилок: Електронний документообіг автоматизує багато операцій, що зменшує ймовірність виникнення помилок та підвищує точність обробки.

Безпека і конфіденційність: Електронні системи забезпечують високий рівень захисту конфіденційної інформації через шифрування, автентифікацію та інші заходи безпеки.

Оптимізація зберігання та пошуку: Електронні системи забезпечують ефективне зберігання та швидкий пошук документів за допомогою індексації, фільтрації та інших технологій.

Оптимізація контролю: Електронні системи надають інструменти для відстеження статусу та маршруту документів, що полегшує контроль за їхнім обігом.

Усі ці аспекти вказують на те, що електронний документообіг допомагає ефективно вирішувати проблеми, пов'язані із традиційним паперовим обігом документів, сприяючи покращенню продуктивності та ефективності бізнес-процесів. Електронний документообіг дозволяє підприємствам суттєво покращити свою документальну діяльність, зробити її більш швидкою, ефективною та економічно вигідною. Схему роботи системи електронного документообігу представлено на рисунку 1.3.

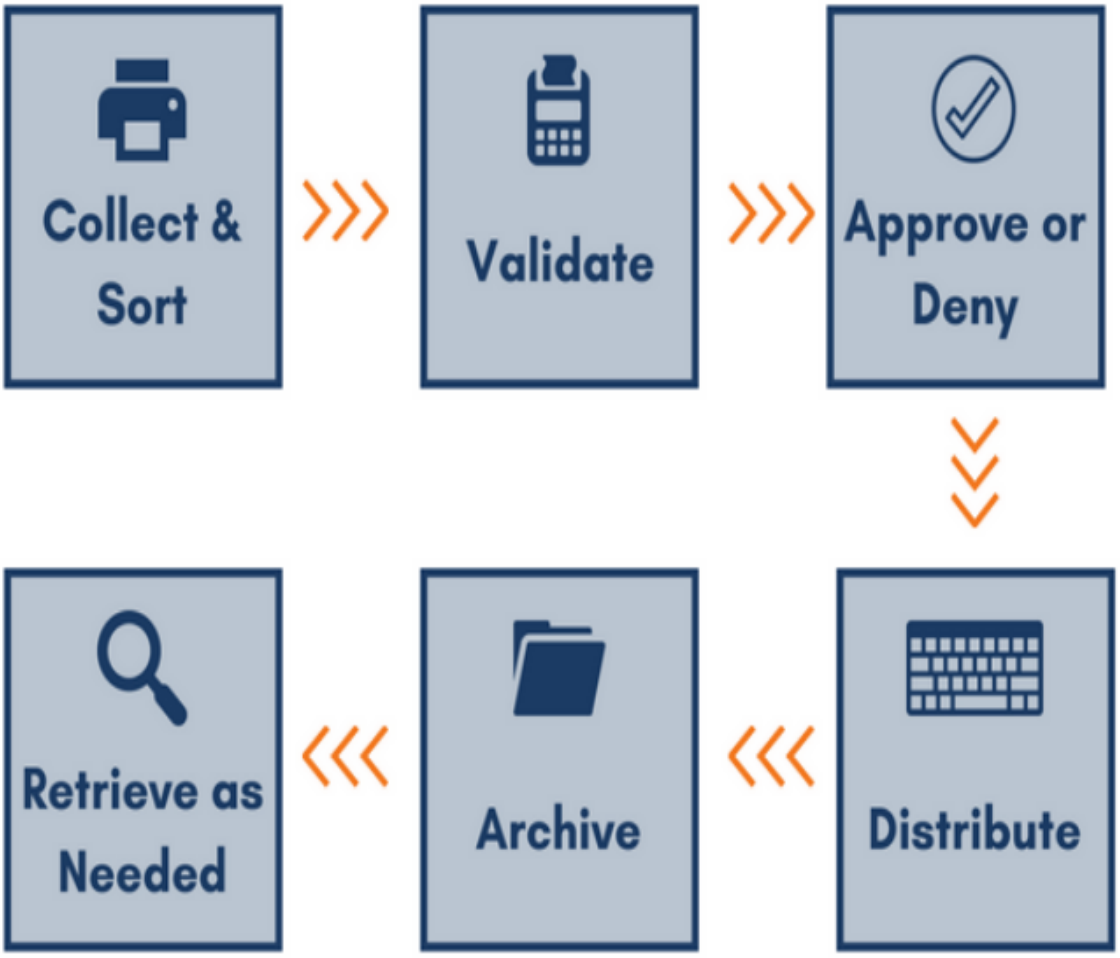

Рисунок 1.3 – Схема роботи електронного документообігу

1.3 Електронний підпис

ЕЦП є електронним еквівалентом звичайного підпису в паперовому світі, але забезпечує вищий рівень безпеки та невідємність.

ЕЦП також вирішує проблему недостатньої безпеки при електронних транзакціях, запобігаючи підробці або зміні документів та забезпечуючи надійний механізм визначення авторства. Це стає ключовим елементом довіри в електронному середовищі, сприяючи розвитку та узгодженому використанню електронних технологій у різних галузях.

Ключі використовуються для створення унікального "цифрового відбитку" документа, який в подальшому може бути перевірений та визнаний як відповідний підпису конкретної особи чи організації. Цифровий підпис складається з криптографічних ключів та алгоритмів, які гарантують безпеку, цілісність та автентифікацію електронних документів чи повідомлень. Основні компоненти електронного підпису включають:

Приватний ключ: Це секретний ключ, який використовується для створення цифрового підпису. Цей ключ зберігається в безпечному місці та відомий лише власнику ЕЦП.

Публічний ключ: Це відкритий ключ, який може бути розповсюджений. Він використовується для перевірки цифрового підпису, створеного відповідним приватним ключем.

Цифровий підпис: Це криптографічний об'єкт, який включає в себе результат застосування приватного ключа до електронного документа. Він служить електронним відбитком, що підтверджує автентичність та цілісність документа.

Будь-яка зміна вмісту документу робить підпис до документу недійсним, адже хеш код документа буде змінено. Це захищає документи від підробки. Наявність відкритого ключа дає можливість ідентифікувати користувача, який наклав підпис, та не дає користувачу можливості відмовитися від своїх зобов’язань.

Фізичні носії електронного підпису часто представлені у вигляді USB-ключів або смарт-карт. USB-ключі зазвичай містять вбудований чип, де зберігається приватний ключ, і можуть бути вставлені безпосередньо в USB-порт комп'ютера. Смарт-карти, у свою чергу, є пластиковими картками з вбудованим чипом, які можна використовувати для аутентифікації через карт-рідер або спеціальний пристрій. Існують також програмні носії електронного підпису, які представляють собою зашифровані файли або програми, що забезпечують безпечне зберігання та використання ключів. Такі програмні рішення можуть працювати на комп'ютерах, смартфонах або інших електронних пристроях.

Основний принцип роботи носіїв електронного підпису полягає в забезпеченні безпечного зберігання приватного ключа та можливості його використання для створення цифрового підпису. Фізичні носії зазвичай мають механізми

аутентифікації, які перевіряють правомірність власника ключа перед наданням доступу до нього. У випадку програмних носіїв, доступ може контролюватися паролем або іншими методами автентифікації.

Сучасний правовий статус безпаперового документообігу із застосуванням електронного цифрового підпису визначений двома законами України, прийнятими у 2003 році: закон «Про електронні документи та електронний документообіг» (№ 851-IV) і закон «Про електронний цифровий підпис» (№ 852-IV).

1.4 Системи електронного документообігу

Система електронного документообігу (СЕД) – це комплекс програмних та апаратних засобів, спрямованих на автоматизацію процесів створення, обігу, контролю та зберігання документів в електронній формі в організації. Основна мета СЕД полягає в поліпшенні ефективності роботи, зниженні часових та фінансових витрат, а також в забезпеченні безпеки та точності обігу документів.

Ці системи дозволяють автоматизувати процеси створення, розгляду, підписання та зберігання електронних документів. Вони забезпечують ефективний потік інформації в організації, дозволяючи швидко реагувати на зміни, впроваджувати покращення та зменшувати час, необхідний для прийняття рішень.

Основні функції СЕД включають в себе автоматизацію процесів обігу документів, створення електронного відображення традиційних робочих потоків, забезпечення зручності управління доступом до документів та їх контроль. Крім того, СЕД дозволяють легко впроваджувати електронні підписи для забезпечення відповідності юридичним вимогам.

Серед основних операцій роботи над електронним документом які надають СЕД можна виокремити наступні операції:

1. створення – процес наповнення електронного документу необхідною інформацією;
2. оброблення – процес додавання додаткової інформації до документа, накладання на документ підписів і печаток;

3. відправлення – процес, під час якого електронний документ відправляється адресату;
4. одержання – процес одержання адресатом електронного документа;
5. зберігання – збереження інформації електронного документу, його підписів і печаток;
6. використання – використання електронних документів як і звичайних паперових документи;
7. знищення – знищення документу, його підписів і печаток.

Відповідно до чинного законодавства України, всі операції над електронними документами мають виконуватись із застосуванням перевірок цілісності та, у разі необхідності, з підтвердженням факту одержання документів. Також, системи електронного документообігу мають функцію перегляду документа – відображення даних, які він містить, електронними засобами у формі, придатній для сприйняття змісту людиною.

Застосування СЕД в сучасному бізнесі дозволяє значно прискорити обіг документів, зменшити ймовірність помилок та витрат, пов'язаних з паперовою документацією. Такі системи стають невід'ємною частиною стратегії цифрової трансформації підприємств, сприяючи підвищенню продуктивності та конкурентоспроможності.

1.5 Системи електронного документообігу в Україні

Поява систем електронного документообігу в Україні пов'язана з розвитком інформаційних технологій та стрімким переходом до цифрової економіки. Одним із ключових етапів цього процесу була адаптація сучасних технологій в управлінських та бізнес-процесах організацій.

З початком 2000-х років в Україні почалося активне впровадження інформаційних систем для автоматизації обігу документів. Організації, незалежно від їхнього масштабу та галузі, почали визнавати переваги електронного документообігу в порівнянні з традиційним паперовим.

Заклади влади, підприємства, та організації почали використовувати системи електронного документообігу для оптимізації робочих процесів, зменшення паперової документації, прискорення прийняття рішень та поліпшення комунікації між підрозділами.

Однією з важливих віх в історії впровадження електронного документообігу в Україні стала реформа державного управління та впровадження електронного урядування. Державні органи та підприємства стали активно використовувати електронні технології для спрощення взаємодії з громадянами та підприємствами.

Зараз системи електронного документообігу в Україні використовуються в різних сферах, таких як бізнес, органи влади, медицина, освіта, тощо. Заходи з цифрової трансформації та підтримка інновацій забезпечують постійне вдосконалення та розширення функціональності систем електронного документообігу в Україні.

1.5.1 Огляд системи «Вчасно»

Вчасно, започаткована у 2017 році, на сьогодні є визнаним лідером на українському ринку електронного документообігу, надаючи широкий спектр послуг для ефективного обміну електронними документами [4]. Однією з ключових особливостей цієї платформи є відсутність необхідності встановлювати додаткове програмне забезпечення – достатньо лише відвідати веб-сайт та розпочати обмін документами з контрагентами, що значно спрощує процес впровадження та користування.

Однією з важливих переваг Вчасно є готовність до інтеграції з основними обліковими системами, що функціонують в Україні, а також наявність REST API для налаштування інтеграції з будь-якими системами, що мають доступ до мережі Інтернет.

У своїй суті, Вчасно представляє собою не просто систему, а повноцінну платформу, яка надає користувачам зручний веб-інтерфейс для управління та обміну різноманітними документами. Перевагою є можливість завантажувати як

стандартизовані, так і документи у різноманітних форматах, що дозволяє користувачам працювати з різноманітними видами інформації.

Незважаючи на це, важливо відзначити, що серед недоліків даного рішення слід відзначити відсутність можливості відправки документів у інші системи та відсутність шифрування відправці документів. Це питання безпеки може стати об'єктом уваги для користувачів, які акцентують на захисті конфіденційності своєї інформації.

1.5.2 Огляд системи «Document Online»

Систему документообігу під назвою "Document Online" є ефективним інструментом для обміну та обробки електронних документів, спрямованих на полегшення бізнес-процесів у сфері обліку, звітності та взаємодії з контрагентами [5].

"Document Online" надає стандартний для систем документообігу спектр функцій, серед яких:

Електронний Документообіг: Платформа дозволяє легко створювати, обмінюватись та обробляти різноманітні документи в електронному форматі.

Інтеграція з Обліковими Системами: "Document Online" надає можливість інтеграції з 1С та надає клієнт для інтеграції з файловою системою, що значно спрощує обробку фінансової інформації. Також має відкрите API.

Безпека та Захист Даних: Платформа забезпечує високий рівень безпеки для збереження конфіденційності та цілісності електронних документів зберігаючи документи на серверах Microsoft Azure.

Незважаючи на відкрите API, що дозволяє іншим системам інтегруватись з Document Online, у самої системи відсутня можливість відправки документів в інші системи.

1.5.3 Огляд сервісу «Paperless»

Сервіс Paperless є українським сервісом електронного документообігу, який надає свої послуги безкоштовно з метою розвитку електронного документообміну в Україні [6].

На відміну від класичних сервісів документообігу, сервіс Paperless надає обмежений функціонал для роботи з електронними документами. Відсутні такі функції як створення маршрутів документів, узгодження та відхилення документів. Також відсутні можливості по створенню та управлінню організаціями, зате як альтернатива даній функції є можливість додавання агентів (максимум 5). Користувачу доступні функції підписання, перегляду та видачі доступу для документу.

Всі документі зберігаються у потрійному екземплярі на незалежних зашифрованих серверах сервісу. Також сервіс має відкрите API, що надає можливість інтеграції з ним.

Також сервіс надає послуги створення власного електронного підпису через Приват24.

Незважаючи на обмежений в порівнянні з іншими сервісами функціонал, система має перевагу у вигляді безкоштовності та простоти використання для користувачів, які не використовують документообіг на рівні підприємств та великих компаній.

1.5.3 Огляд системи «Signy»

Signy – сервіс електронного документообігу, створений компанією SmartTender – одним з найбільших торговельних майданчиків проведення державних тендерів та комерційних торгів в Україні. Сервіс Signy має зручний та зрозумілий інтерфейс, подібний до інтерфейсів типових email-сервісів. Для роботи з ним користувачам не потрібно встановлювати додаткове програмне забезпечення. Необхідно лише пройти просту реєстрацію на сайті та мати пароль або КЕП (кваліфікований електронний

підпис, отриманий в центрі видачі електронних ключів) для входу в свій особистий кабінет. Сервіс дозволяє здійснювати наступні дії:

- обмінюватися будь-якими документами зі своїми колегами для підписання (договір, акт приймання-передачі, акт наданих послуг тощо) чи фіксації факту отримання та ознайомлення (лист, рахунок тощо) набагато швидше та безпечніше, ніж з використанням звичайних email-сервісів;
- погоджувати документи в режимі онлайн;
- надавати документам юридичної сили за допомогою ЕЦП/КЕП.

Окрім таких функцій, як підписання, зберігання та відправка юридично значущих документів, даний сервіс підтримує наступні можливості:

- створення документів за допомогою попередньо налаштованих загальноприйнятих або індивідуальних (згідно з вашим технічним завданням) шаблонів;
- коментування та редагування документів;
- налаштування індивідуального маршруту підписання та узгодження документа;
- відстежування статусу документа та історії дій по ньому;
- завантаження XML-файлів з формуванням візуального відображення документа;
- створення груп користувачів для внутрішнього обговорення документа;
- можливість роботи з декількома юридичними особами з одного особистого кабінету;
- підтримка еТТН (електронно товарно-транспортна накладна);
- інтеграція з іншими системами ЕДО;
- імпорт списку контрагентів у базу Signy;
- розумний пошук та доступ до документів 24/7.

Сервіс також надає додаткова можливісті для користувачів майданчика SmartTender.biz.

Signy має відкрите API і крім цього надає можливості своїм користувачам для інтеграції з іншими системами засобами сервісу.

1.6 Постановка задачі інтеграції сервісу Signy з іншими системами ЕДО

На підставі виконаного аналізу систем електронного документообігу зроблено висновок про необхідність створення програмного забезпечення для інтеграції сервісу Signy з іншими системами. Це програмне забезпечення задовольнить потреби користувачів та надасть сервісу перевагу над конкурентами. Користувачами створеного ПЗ будуть користувачі системи Signy.

Для досягнення мети треба виконати задачі:

- аналіз популярних систем електронного документообігу в Україні та можливостей інтеграцій з ними;
- дослідження існуючих типів та методів інтеграцій комп'ютерних систем;
- на основі аналізу та дослідження спроектувати та розробити власний механізм інтеграції;
- проведення тестування створеного програмного забезпечення.

ПЗ має виконувати наступні функції:

- відправка документів з системи Signy в інші системи;
- відправка підписів по документам в інші системи;
- отримання документів з інших систем;
- отримання підписів з інших систем;
- отримання поточного статусу документа (очікування підпису, підписано, відхилено, видалено).

Додатково для зручності може бути реалізовано обмін коментарями по документу між системою Signy та іншою системою.

Висновки до розділу 1

У даному розділі розглянуто проблеми які виникають при використанні підприємствами різних систем документообігу. Наведено визначення понять

електронного документу та електронного підпису. Описано основні принципи роботи електронного документообігу та наведено схему роботи.

Було описано особливості електронного документообігу в Україні та оглянуто основні системи, які надають подібні послуги, а саме: Signy, Вчасно, Document Online та Paperless.

2 ОГЛЯД ПІДХОДІВ ТА МЕТОДІВ ІНТЕГРАЦІЙ КОМП'ЮТЕРНИХ СИСТЕМ

2.1 Типи інтеграцій

Інтеграція комп'ютерних систем - це процес об'єднання різних програмних та апаратних компонентів в єдину функціональну систему [7]. Мета інтеграції полягає в тому, щоб різні системи працювали як єдина і взаємодійшли для виконання спільних завдань.

Цей процес може включати в себе:

- обмін даними, який полягає в здатності систем обмінюватись інформацією між собою. Це може включати передачу даних в реальному часі або використання спільної бази даних;
- спільний доступ до ресурсів, який полягає в можливості інтегрованих систем мати спільний доступ до ресурсів, таких як принтери, сховища даних, обчислювальні потужності тощо;
- спільні бізнес-процеси – можливість налаштовувати системи для автоматизації та співпраці в області спільних бізнес-процесів, забезпечуючи злагоджену роботу між відділами та функціональними блоками підприємства;
- єдину точка управління, це можливість інтегрованих систем мати єдиний інтерфейс управління, що спрощує керування та моніторинг компонентів системи;
- забезпечення безпеки і цілісності даних, це включення засобі захисту даних та забезпечення їх цілісності під час обміну та обробки в інтеграції.

Розглянемо основні типи інтеграцій:

2.1.1 Базова Інтеграція Даних

Базова інтеграція даних є комплексним підходом, спрямованим на об'єднання та оптимізацію управління різноманітними джерелами інформації в організації.

Головною метою цього процесу є створення централізованої точки доступу до даних, що сприяє консолідації та підвищенню якості інформації.

На перший етап здійснюється збір даних із різних джерел, включаючи бази даних, файли та веб-сервіси. Ці дані потім агрегуються, створюючи централізовану базу даних. Процес трансформації забезпечує стандартизацію та адаптацію даних до єдиного формату для забезпечення консистентності.

Однією з ключових функцій базової інтеграції даних є забезпечення єдиної версії даних. Це означає, що кожен користувач отримує доступ до однієї таємниці, що уникне суперечностей і нев'язок в інформації.

Автоматизація процесів, таких як збір, трансформація та завантаження (ETL), є ключовою складовою базової інтеграції даних. Це спрощує та прискорює робочі процеси, роблячи їх менш затратними та більш ефективними.

Переваги використання базової інтеграції даних включають підвищення ефективності бізнес-процесів, зменшення витрат на обслуговування, поліпшення якості даних та забезпечення гнучкості та розширюваності для майбутніх потреб організації.

2.1.2 Інтеграція Додатків

Інтеграція додатків – це процес об'єднання різних програмних застосунків чи систем для сприяння взаємодії та обміну даними між ними. Це може включати в себе поєднання функціональності різних додатків для забезпечення більш ефективного та комплексного рішення для користувача.

Однією з ключових мет інтеграції додатків є покращення зручності використання та оптимізація робочих процесів. При взаємодії різних додатків можна досягти синергії, де цілісний результат перевищує суму окремих компонентів.

Інтеграція може бути реалізована різними способами, включаючи використання API (інтерфейсів програмування додатків), зокрема RESTful або SOAP, які надають стандартизовані засоби обміну даними між додатками.

Процес інтеграції додатків передбачає здатність програм зчитувати, розуміти та взаємодіяти з даними інших додатків. Це може включати автоматичну

синхронізацію даних, обмін повідомленнями або навіть виклик функцій іншого додатка.

До переваг інтеграції додатків відносять підвищення ефективності та продуктивності, зменшення ризику помилок при ручному введенні даних, а також збільшення гнучкості і адаптивності організації до змін у бізнес-середовищі. Однак інтеграція додатків також може стикатися з викликами, такими як забезпечення сумісності між різними версіями програм, вирішення питань безпеки та захисту даних, а також визначення оптимального рівня інтеграції для конкретної бізнес-потреби.

2.1.3 Інтеграція Бізнес-Процесів

Інтеграція бізнес-процесів – це стратегічний підхід, спрямований на створення єдиної та оптимізованої системи взаємодії між різними елементами підприємства. Головною метою цього процесу є поліпшення ефективності та результативності робочих процесів, зниження зайвих витрат та максимізація використання ресурсів.

Інтеграція бізнес-процесів передбачає спрощення та оптимізацію потоків роботи в організації. Цей процес може включати автоматизацію рутинних завдань, забезпечення спільного доступу до даних та ресурсів для різних відділів, а також впровадження єдиної системи моніторингу та звітності.

Інтеграція бізнес-процесів може бути реалізована за допомогою різних інструментів, таких як інтеграційні платформи, API (інтерфейси програмування додатків), та спеціальні програмні рішення. Важливим аспектом є здатність різних систем і додатків взаємодіяти між собою та обмінюватися необхідною інформацією.

Переваги інтеграції бізнес-процесів включають підвищення ефективності роботи, зниження часу на виконання завдань, покращення якості прийняття рішень через доступ до актуальних даних, а також підвищення гнучкості та адаптивності підприємства до змін у бізнес-середовищі.

Однак інтеграція бізнес-процесів може також вимагати значних зусиль на етапі планування та впровадження, враховуючи необхідність аналізу та адаптації існуючих процесів, а також вирішення питань безпеки та конфіденційності даних.

2.1.4 Інтеграція Хмарних Сервісів

Інтеграція хмарних сервісів – це процес об'єднання та взаємодії різних хмарних платформ та сервісів з метою створення єдиної та оптимізованої інфраструктури для зберігання даних, обчислень та обміну інформацією. Це дозволяє підприємствам використовувати різні хмарні рішення та сервіси, об'єднуючи їх в єдиний функціональний блок.

Однією з ключових переваг інтеграції хмарних сервісів є можливість забезпечення гнучкості та масштабованості. Підприємства можуть використовувати різні хмарні сервіси для різних аспектів своєї діяльності, таких як зберігання даних, обробка великих обсягів інформації, робота зі штучним інтелектом тощо.

Інтеграція хмарних сервісів також дозволяє ефективно використовувати можливості кожного конкретного хмарного сервісу. Наприклад, одна платформа може спеціалізуватися на аналітиці даних, інша - на забезпеченні безпеки, а третя - на обчисленнях в реальному часі. Інтеграція забезпечує взаємодію цих сервісів для отримання повноцінного функціоналу.

Однак важливо враховувати питання безпеки та конфіденційності даних при інтеграції хмарних сервісів. Це може включати в себе застосування шифрування, правильне керування доступом та забезпечення відповідності стандартам безпеки.

Інтеграція хмарних сервісів стає ключовим компонентом стратегії цифрової трансформації, дозволяючи підприємствам бути більш гнучкими та реагувати на зміни в бізнес-середовищі.

2.1.5 Інтеграція IoT (Internet of Things)

Інтеграція Інтернету речей (IoT) – це складний та стратегічний підхід до об'єднання різноманітних пристроїв та сенсорів у єдину систему для збору, обробки та використання даних в реальному часі. Цей процес створює зв'язане середовище, де пристрої можуть співпрацювати та обмінюватися інформацією для покращення ефективності та приносить нові можливості в різних сферах життя та бізнесу.

Інтеграція IoT включає в себе роботу з великим обсягом даних, зібраних з різних пристроїв, що може бути розподілено по всьому глобальному масштабу. Одним із ключових аспектів є розробка ефективних механізмів для збору, передачі та обробки цих даних в реальному часі.

Інтеграція IoT також передбачає розробку програмного забезпечення та платформ, які забезпечують взаємодію між пристроями та додатками. Це може включати в себе створення застосунків для відображення та аналізу даних, а також розробку інтерфейсів для взаємодії з IoT-пристроями.

Безпека є важливим аспектом інтеграції IoT, оскільки збільшується кількість точок доступу до мережі. Шифрування даних, автентифікація пристроїв та застосунків, а також моніторинг безпеки стають ключовими складовими для захисту від потенційних загроз.

Інтеграція IoT відкриває нові перспективи для бізнесу та споживачів, включаючи покращене управління ресурсами, зменшення витрат та створення інноваційних продуктів та послуг.

2.1.6 Мікросервісна Інтеграція

Мікросервісна інтеграція є стратегічним підходом розробки програмного забезпечення. Він базується на використанні невеликих та незалежних компонентів, відомих як мікросервіси. Ці мікросервіси розглядаються як окремі функціональні одиниці, що виконують конкретні завдання та можуть взаємодіяти один з одним через стандартизовані інтерфейси.

Основними принципами мікросервісної інтеграції є декомпозиція монолітних застосунків на невеликі та автономні мікросервіси, що спрощує розробку, тестування та розгортання. Кожен мікросервіс може виконувати свої функції та взаємодіяти з іншими мікросервісами за допомогою мережевих викликів або інших механізмів комунікації.

Мікросервісна архітектура дозволяє незалежно розгортати, масштабувати та оновлювати кожен мікросервіс окремо, зменшуючи вплив змін на інші частини

системи. Це також сприяє гнучкості та швидкості розробки, оскільки окремі команди можуть працювати над різними мікросервісами паралельно.

Важливою частиною мікросервісної інтеграції є використання стандартизованих протоколів комунікації, таких як Message Queue або HTTP, для обміну даними між мікросервісами. Також важливо забезпечити моніторинг та логування для ефективного виявлення проблем та усунення їх.

Завдяки мікросервісній інтеграції підприємства можуть досягати більшої гнучкості, швидкості вдосконалення та розширення, впроваджуючи нові функції та зміни без значних перебоїв у роботі системи в цілому.

2.2 Методи інтеграцій

Існує кілька методів інтеграції між комп'ютерними системами, які використовуються для забезпечення спільної роботи та обміну даними між різними програмними продуктами [8].

2.2.1 API (Application Programming Interface) Інтеграція

API інтеграція – це спосіб забезпечення взаємодії між різними програмами та системами. Цей метод базується на використанні програмних інтерфейсів, які визначають правила обміну даними та функціональністю між різними програмами.

Основні технології для API інтеграції включають RESTful API та SOAP. RESTful API використовує протокол HTTP для передачі даних та є легким та швидким. SOAP використовує XML для структурованої інформації та є більш стандартизованим, але може бути менш легким.

Основними елементами API є ендпойнти, які представляють URL-адреси, за якими програми можуть викликати функції чи отримувати дані з інших систем. Автентифікація та авторизація забезпечують перевірку ідентифікації та контроль доступу. Дані, що передаються через API, можуть бути у форматах JSON або XML.

Процес включає реєстрацію та отримання ключа API, аутентифікацію через цей ключ, взаємодію через ендпойнти, обробку отриманих відповідей та моніторинг з логуванням дій.

API інтеграція є гнучким та ефективним методом для взаємодії різних програм та систем, незалежно від їхньої технічної архітектури чи мови програмування. Документація API відіграє ключову роль у спрощенні розробки та використання цього методу інтеграції.

2.2.2 Middleware

Middleware – це програмне забезпечення, яке виступає посередником між різними програмами або компонентами системи для полегшення їх взаємодії та інтеграції. Основною метою middleware є створення єдиного інтерфейсу для обміну даними та повідомленнями між різними частинами системи, що дозволяє їм працювати разом як єдина, злагоджена система.

Middleware може включати в себе різноманітні технології та сервіси, такі як Message-Oriented Middleware (MOM), Enterprise Service Bus (ESB), та інші. Воно дозволяє взаємодію різних програм, компонентів або систем, навіть якщо вони розгорнуті на різних фізичних серверах або мають різні технічні стеки.

Middleware допомагає у створенні єдиного середовища для обміну даними, використовуючи стандартизовані інтерфейси та протоколи. Це полегшує розробку, тестування та підтримку розподілених систем, оскільки різні частини можуть взаємодіяти через middleware, не залежно від їхньої конкретної реалізації чи технологічного стеку.

Middleware також забезпечує такі функціональності, як маршрутизація повідомлень, управління транзакціями, безпека та моніторинг, що робить його важливим інструментом для побудови надійних та масштабованих розподілених систем.

Розглянемо основні аспекти роботи Middleware.

Стандартизований Інтерфейс – Middleware визначає стандартизований інтерфейс для обміну даними. Це може включати у себе визначення форматів даних,

протоколів комунікації та інших стандартів, які дозволяють різним компонентам взаємодіяти безпроблемно.

Маршрутизація Повідомлень – Middleware відповідає за ефективну маршрутизацію повідомлень між різними частинами системи. Це забезпечує доставку повідомлень від відправника до отримувача, навіть якщо вони знаходяться на різних серверах чи в різних мережах.

Управління Транзакціями – Middleware може підтримувати управління транзакціями, забезпечуючи консистентність даних у випадку виникнення помилок або відмов.

Безпека та Аутентифікація – Middleware може включати в себе механізми безпеки для захисту обмінюваних даних. Це включає в себе аутентифікацію, авторизацію та шифрування інформації.

Моніторинг та Логування – Middleware може забезпечувати інструменти для моніторингу та логування обміну даними. Це допомагає відслідковувати роботу системи, виявляти проблеми та вдосконалювати її продуктивність.

Скальпельність та Масштабованість – Middleware розробляється таким чином, щоб бути масштабованим та забезпечувати ефективність при збільшенні обсягу обміну даними та кількості систем.

Спрощення Розробки та Підтримки – використання Middleware спрощує розробку, тестування та підтримку розподілених систем, оскільки він надає абстракцію від деталей взаємодії між компонентами.

Принцип роботи Middleware полягає в створенні невидимого шару, який дозволяє різним частинам системи спілкуватися одна з одною, незалежно від їхнього фізичного місцезнаходження чи технічних особливостей. Це сприяє побудові гнучких, масштабованих та легко змінюваних систем.

2.2.3 База даних та реплікація

Метод інтеграції "База даних та реплікація" спрямований на забезпечення синхронізації та обміну даними між різними базами даних. Замість того, щоб кожна

система працювала з власною копією даних, реплікація дозволяє створювати копії даних та оновлювати їх автоматично в інших системах.

Принцип роботи полягає в тому, що зміни, внесені в одній базі даних, автоматично відображаються в інших базах даних, які беруть участь у реплікації. Це забезпечує консистентність даних між різними частинами системи та дозволяє їм працювати з актуальною інформацією.

Реплікація може бути налаштована для виконання в режимі реального часу або періодично, залежно від вимог системи та обсягу даних. Цей метод інтеграції особливо ефективний для сценаріїв, де різні частини системи мають власні локальні бази даних, але потребують доступу до спільної інформації.

Важливою частиною цього методу є механізми визначення та вирішення конфліктів, які можуть виникати, коли дані змінюються в обох кінцях реплікаційного процесу. Реплікація може бути організована в різних топологіях, таких як "один до одного", "багато до багатьох" або "багато до одного", що дозволяє адаптувати систему до конкретних потреб та обмежень.

Застосування методу "База даних та реплікація" дозволяє створювати розподілені системи з високою доступністю та швидкодією, забезпечуючи надійний обмін даними між різними компонентами архітектури.

2.2.3 Методи віддаленого виклику

Метод інтеграції "Remote Procedure Call" (RPC) є парадигмою програмування, яка дозволяє викликати функції або процедури в інших програмах чи на віддалених вузлах мережі. Його принцип роботи базується на тому, що клієнт може викликати функції чи процедури на віддаленому сервері, подібно до того, як це відбувається локально.

У контексті RPC клієнт і сервер можуть бути розташовані на різних машинах та спілкуватися через мережу. Процедура, яку клієнт хоче викликати, повинна бути визначена на віддаленому сервері, і клієнт має можливість передавати параметри цій процедурі.

Принцип роботи методу інтеграції "Remote Procedure Call" (RPC) полягає в можливості викликати функції або процедури, які виконуються на віддаленому сервері, так, ніби вони викликаються локально. Основна ідея полягає в тому, що клієнт може ініціювати виклик віддаленої процедури на сервері, надсилаючи необхідні параметри, а сервер виконує цю процедуру та повертає результат клієнту.

Процес взаємодії між клієнтом та сервером через RPC може бути розкладений на кілька етапів. По-перше, клієнт ініціює виклик віддаленої процедури, передаючи параметри цій процедурі. Далі відбувається передача даних через мережу до сервера. На стороні сервера відбувається виклик віддаленої процедури з переданими параметрами, обробка цієї процедури та генерація результату. Затим результат повертається через мережу клієнту, який його отримує і може продовжити виконання локальних операцій.

Однією з ключових переваг RPC є абстрагування від деталей мережевої комунікації, тобто розробникам не потрібно вручну вирішувати питання передачі даних через мережу. Крім того, цей метод інтеграції дозволяє зручно взаємодіяти з віддаленими сервісами, або навіть з віддаленими серверами, які можуть бути реалізовані на різних мовах програмування чи працювати на різних платформах.

Однак існують виклики стосовно безпеки та надійності у випадку використання RPC, включаючи необхідність ефективної обробки помилок, управління транзакціями та забезпечення захисту від вторгнень при обміні даними через мережу.

2.2.4 Webhooks

Метод інтеграції "Webhooks" є механізмом взаємодії між двома або більше програмними системами, який дозволяє автоматично сповіщати одну систему про події чи зміни в іншій системі. Принцип його роботи базується на асинхронній комунікації та реалізується через HTTP-протокол.

Щоб використовувати Webhooks, користувач (власник системи) реєструє URL-адресу в системі, яка підтримує цей метод інтеграції. Цей URL-адрес функціонує як точка призначення для сповіщень. Коли в системі виникає певна подія

або зміна, вона автоматично генерує HTTP-запит і відправляє його на зазначений URL-адрес.

При отриманні запиту по вказаному URL-адресу система-одержувач (клієнт) обробляє цей запит. В тілі запиту може міститися інформація про саму подію, параметри чи будь-які інші дані, які необхідно передати. Система-одержувач виконує відповідні дії в залежності від отриманої події, такі як оновлення даних, сповіщення користувачів або виклик інших сервісів.

Один з основних аспектів Webhooks – це асинхронність. Система, що відправляє повідомлення, не очікує негайної відповіді від системи-одержувача. Це дозволяє обом системам працювати асинхронно та надає гнучкість в організації обміну інформацією.

Важливою частиною використання Webhooks є можливість управління підписками. Користувач може змінювати чи відміняти підписку на певні події, а також змінювати URL-адресу для сповіщень. Це надає гнучкість та контроль над інтеграцією.

2.3 Формати обміну даними

Формати обміну даними при інтеграції є стандартами, які визначають, як дані повинні бути структуровані та представлені під час їх передачі між різними системами. Ці формати грають ключову роль у забезпеченні взаєморозуміння між різнорідними програмами та дозволяють їм ефективно обмінюватися інформацією. Декілька популярних форматів обміну даними включають:

- JSON (JavaScript Object Notation);
- XML (eXtensible Markup Language);
- CSV (Comma-Separated Values);
- YAML (YAML Ain't Markup Language).

JSON є легким та зрозумілим форматом обміну даними, який базується на синтаксисі JavaScript. Він використовує пари "ключ-значення" та підтримує різноманітні типи даних. JSON є популярним у веб-розробці та використовується для передачі структурованих даних між веб-клієнтами та серверами[9];

XML є розширюваним мовою розмітки, яка використовується для створення людино-читабельних та машино-читабельних текстових документів. Він дозволяє створювати власні теги та визначати структуру даних. XML широко використовується у різних галузях, включаючи обмін інформацією між веб-серверами та базами даних;

CSV використовується для представлення табличних даних у вигляді текстового файлу, де кожен рядок представляє запис, а роздільники (зазвичай коми) визначають різні стовпці. Цей формат легко читається іншими програмами та часто використовується для обміну даними між електронними таблицями та базами даних;

YAML є людино-читабельним форматом, що використовується для конфігурації та обміну даними. Він використовує простий синтаксис, базований на відступах, і відповідає структурам даних вигляду "ключ: значення". YAML знаходить застосування у конфігураційних файлах та API для зручності читання та написання людьми.

Ці формати обміну даними надають інтерфейс для стандартизованого обміну інформацією між різними системами, що дозволяє їм ефективно співпрацювати та виконувати інтеграцію.

Висновки до розділу 2

В даному розділі наведено визначення інтеграції комп’ютерних систем та описані можливі типи таких інтеграцій, а саме:

- базова інтеграція даних;
- інтеграція додатків;
- інтеграція бізнес-процесів;
- інтеграція хмарних сервісів;
- інтеграція IOT;
- мікросервісна інтеграція.

Також описані основні методи інтеграцій та описано основні формати обміну даних при використанні інтеграцій.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ

3.1 Засоби програмної реалізації

3.1.1 Середовище Microsoft Visual Studio

Середовище Microsoft Visual Studio – це середовище розробки програмного забезпечення від фірми Microsoft [10].

Це середовище надає можливості розробки програм з використанням різних мов програмування, наприклад: Python, C, C++, C# та інші. Також існує можливість розробки додатків не тільки під Windows, а й під інші платформи, такі як: Android, iOS. Також є можливість створення веб додатків.

Visual Studio Community – це безкоштовна версія програми середовища розробки, яка доступна для студентів, учнів та розробників. Інтерфейс програми середовища розробки представлено на рисунку 3.1.

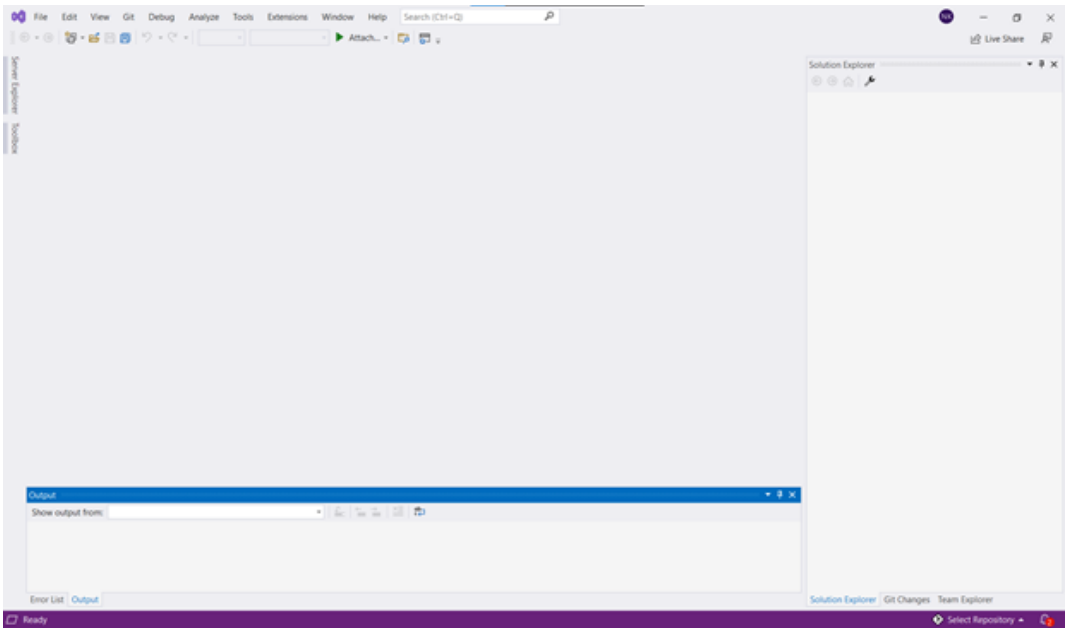

Рисунок 3.1 – Інтерфейс середовища розробки Microsoft Visual Studio

Основними елементами середовища розробки є вікно коду та Solution Explorer. Також, для командної розробки присутні інструменти Git Changes та Team Explorer.

3.1.2 Мова розмітки HTML

Мова розмітки HTML, або HyperText Markup Language, є стандартом для створення і відображення веб-сторінок у веб-браузерах [12]. HTML визначає структуру документа та визначає, як його вміст повинен бути представлений. Браузер інтерпретує HTML і відформатований текст відображається на моніторі комп'ютера або на екрані мобільного пристрою. У мережі Інтернет сторінки HTML зазвичай передаються з сервера в браузер через HTTP або HTTPS, використовуючи простий текст або шифрування.

В HTML можна вбудовувати код мови програмування JavaScript задля контролю поведінки та вмісту веб-сторінок. Також, CSS, який міститься в HTML, визначає зовнішній вигляд і макет сторінки. Всі HTML-документи є наборами елементів, кожний з яких має власний початковий та кінцевий тег. Елемент може бути порожнім, тобто не містити тексту чи інших даних; в цьому випадку кінцевий тег, як правило, не вказується (наприклад, тег розриву рядка
 є одиночним і не потребує закриття). Кожен елемент може також мати атрибути, які визначають його властивості, такі як атрибут href = "" у посиланні. Атрибути вказуються у початковому тезі.

Крім елементів, в HTML-документах існують англійські сутності, відомі як "спеціальні символи". Сутність починається з символу &, і вона може мати форму &name; або &#NNNN;, де NNNN - це код символу Unicode у десятковому вираженні.

Кожен HTML-документ, який відповідає конкретній версії специфікації HTML, має містити вказаний тип документу HTML (<! DOCTYPE …>). Потім початок і кінець документа позначаються тегами <html> та </html> відповідно. Всередині цих тегів розташовані теги заголовка (<head> </head>) та теги тіла (<body> </body>) документа.

3.1.3 Мова стилю сторінок

CSS(Cascading Style Sheets), що означає "каскадні таблиці стилів", представляє собою важливий інструмент для веб-розробників, призначений для оформлення та

форматування веб-документів, які створені за допомогою мов розмітки, таких як HTML або XML [13]. У веб-екосистемі взаємодії між користувачем і контентом, CSS використовується для надання стилізованого та привабливого зовнішнього вигляду веб-сторінок, що грає ключову роль поряд із HTML та JavaScript.

Завдяки CSS розробники можуть визначати такі аспекти, як кольори, шрифти, стилі і макет окремих елементів. Основною метою використання CSS є відокремлення логічної структури веб-сторінки (створеної мовами розмітки) від її зовнішнього вигляду, який тепер здійснюється за допомогою цієї формальної мови. Це не тільки полегшує розуміння і модифікацію коду, але і робить веб-сторінки більш доступними та адаптованими до різних пристроїв та платформ.

Крім зазначених можливостей, CSS дозволяє відображати один і той же контент у різних стилях або за допомогою різних методів виводу, таких як екранна візуалізація, друк, читання голосом або використання шрифтів Брайля. Це забезпечує гнучкість інтерфейсу та зручність взаємодії користувачів з вмістом в Інтернеті.

3.1.4 ASP.NET MVC

ASP.NET є технологією, розробленою компанією Майкрософт у 2009 році, для створення веб-застосунків та веб-сервісів. Ця технологія входить до складу платформи Microsoft.NET і ідеально підходить для нашого проєкту, який використовує мову програмування C# та .NET Framework - складову цієї ж платформи. Важливою перевагою ASP.NET є його висока швидкість порівняно з іншими технологіями, що базуються на скриптових мовах програмування. Розроблений Майкрософт, цей набір включає розширювальні елементи управління, бібліотеки та класи, що значно полегшують розробку та впровадження веб-застосунків (рисунок 3.2).

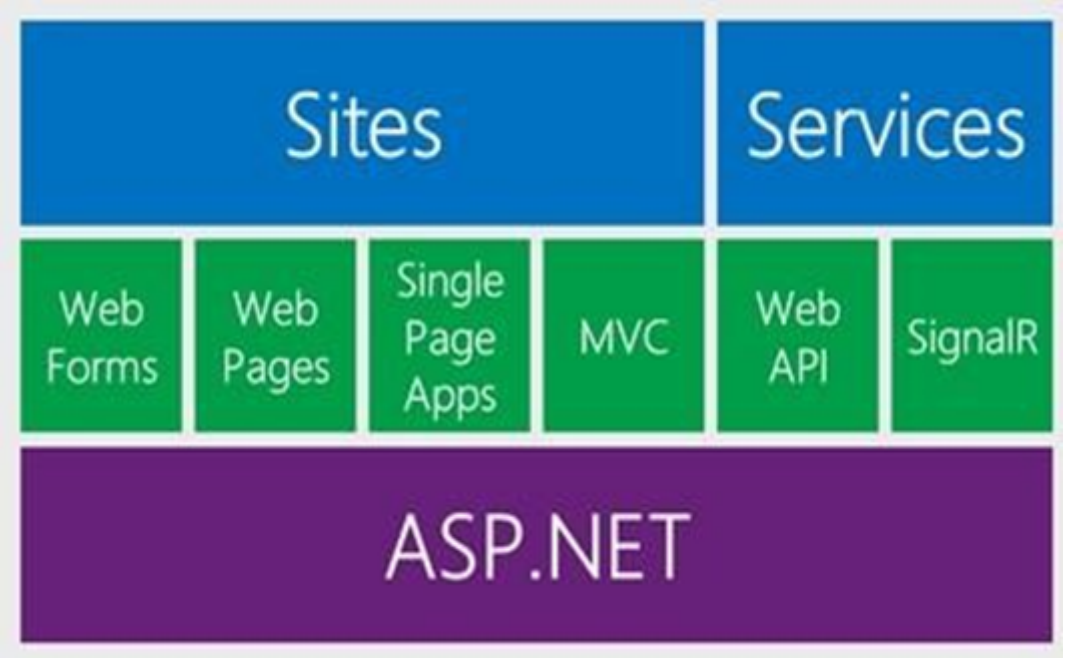

Рисунок 3.2 – Компоненти платформи ASP.NET

ASP.NET опирається на багатомовні можливості .NET, які дозволяють писати код з використанням різних мов програмування, таких як: C#, VB, C/C++ та інших. Також використання ASP.NET має такі плюси, як:

– поділ візуальної частини та бізнес-логіки рішення;

– розширювана модель обробки запитів, яка дає змогу розробнику встановлювати свої обробники в конвеєр обробки запиту.

Шаблон архітектури Model-View-Controller (MVC) розбиває додаток на такі основні компоненти: модель, представлення і контролер. ASP.NET MVC виступає як альтернатива схемі веб-форм ASP.NET при створенні веб-додатків. Ця легковагова платформа MVC надає широкі можливості тестування і, так само, як і додаток на основі веб-форм, інтегрується з існуючими функціями ASP.NET.

MVC представляє собою стандартний шаблон розробки, який відомий багатьом фахівцям і розробникам. Деякі типи веб-додатків мають переваги при використанні платформи MVC, тоді як іншим може бути доцільно використовувати традиційну схему додатків з використанням зворотньої передачі даних ASP.NET. Використання одного з цих підходів не обмежує програміста в застосуванні іншого.

Платформа MVC включає такі компоненти (рисунок 3.3):

Моделі: це частини програми, які втілюють логіку збереження даних у додатку.

Представлення: призначені для відображення інтерфейсу користувача додатку (інтерфейс зазвичай створюється на основі даних моделі).

Контролери: взаємодіють з користувачем, працюють з моделлю та вибирають представлення, яке відображає призначений для користувача інтерфейс.

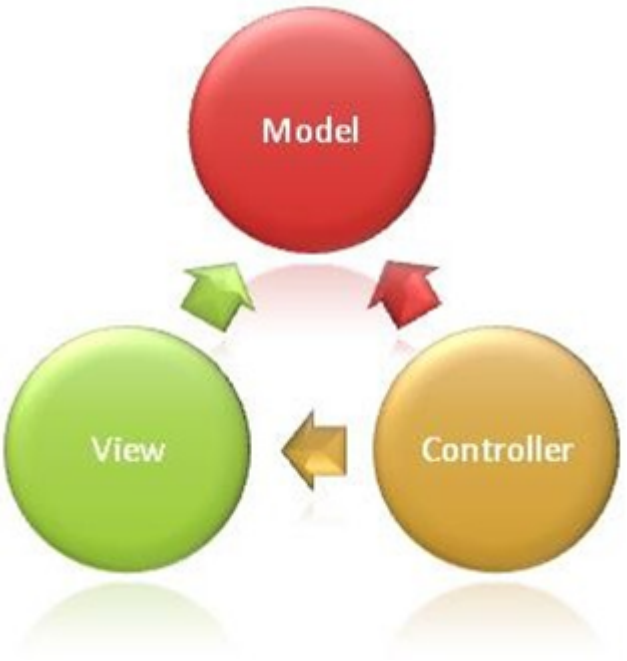

Рисунок 3.3 – Компоненти платформи ASP.NET

У веб-додатку MVC роль представлення обмежена лише відображенням даних, водночас контролер обробляє введені дані та реагує на дії користувача. Модель відповідає за збереження оброблених даних. Наприклад, контролер може обробляти значення запиту та змінювати отримані дані, передаючи їх у модель. Модель, у свою чергу, має можливість використовувати ці значення для створення запиту до бази даних, зберігання або модифікації інформації. Шаблон MVC дозволяє створювати додатки, де різні аспекти, такі як логіка введення, логіка інтерфейсу і бізнес-логіка розділені, але мають тісну між собою. Зв'язок основних компонентів програми MVC також спрощує паралельну розробку додатку. Наприклад, один з розробників може створювати представлення, інший створювати логіку контролера, а третій – бізнес-логіку моделі. Це розширює можливості залучення різних команд розробників задля більш швидкого впровадження рішення.

3.1.5 Фреймворк Vue.js

Vue.js – прогресивний JavaScript фреймворк який використовується для створення користувацьких інтерфейсів [16]. Він призначений для розробки веб-застосунків та односторінкових додатків. Vue.js є легким, гнучким та ефективним інструментом, який широко використовується для створення сучасних та інтуїтивно зрозумілих веб-інтерфейсів. Цей фреймворк має декларативний підхід до програмування і базується на шаблонах, які забезпечують простоту розробки та зрозумілість коду. Vue.js дозволяє легко реагувати на зміни даних та автоматично оновлювати відображення безпосередньо на сторінці.

Однією з ключових особливостей Vue.js є власний інтегрований сервіс Vue CLI, який надає можливості для швидкої налаштування проектів та автоматичного виведення оптимальної конфігурації. Vue.js також підтримує компонентний підхід до розробки, де інтерфейс розділяється на невеликі, перевикористовувані компоненти, що полегшує управління та розширення проектів. Бібліотека компонентів та декларативна система даних дозволяють створювати ефективні та легко розширювані додатки.

Більше того, Vue.js володіє ефективною системою управління станом додатку. Використовуючи концепцію реактивних об'єктів та директив, таких як v-model, розробники можуть легко синхронізувати дані між моделлю та представленням. Фреймворк підтримує велику кількість розширень та плагінів, що робить його гнучким і відкритим для використання в різноманітних проектах. Vue.js також можна легко інтегрувати з іншими бібліотеками та фреймворками, що надає розробникам вибір у стеку технологій для їхніх проектів. Однією з основних переваг Vue.js є його висока продуктивність. Фреймворк працює ефективно як для великих інтернет-застосунків, так і для невеликих проектів, забезпечуючи швидку реакцію на дії користувача та оптимальне використання ресурсів.

Усі ці особливості роблять Vue.js потужним інструментом для розробки веб-додатків, де надається акцент на простоту використання, продуктивність та розширюваність. Крім того, Vue.js активно підтримується та має велике співтовариство розробників, що сприяє постійному вдосконаленню та розвитку цього фреймворку.

3.1.4 Мова програмування C#

Мова програмування C# є однією з найпопулярніших мов програмування. Була розроблена Microsoft в 2000 році і стала ключовою мовою розробки програм на платформах Microsoft .NET Framework [17].

C# це об'єктно-орієнтована мова програмування, що дозволяє розробникам створювати класи, об'єкти і використовувати їх для створення програм. Синтаксис C# подібний до синтаксису мов програмування C і C++. Це полегшує перехід для тих розробників, які вже мають досвід роботи з цими мовами.

Мова C# використовується платформами Microsoft .NET Framework і .NET Core (.NET 5 і вище). Це надає можливості створення різноманітних додатків, таких як веб-додатки, десктоп-додатки, мобільні додатки, ігри та інші. З виходом .NET 5 і .NET 6 (і наступних версій), розробники мають можливість використовувати мову C# для створення мультиплатформених додатків, включаючи ті, які працюють на Windows, macOS, Linux та інших операційних системах.

3.1.5 Бібліотека Newtonsoft.Json

Newtonsoft.Json – це бібліотека для мови програмування C#, яка надає зручні і потужні інструменти для роботи з форматом даних JSON. Основні можливості бібліотеки включають в себе серіалізацію і десеріалізацію об'єктів у формат JSON та навпаки.

Бібліотека підтримує різні функції, які роблять роботу з JSON простою та ефективною. Деякі з ключових можливостей включають:

- серіалізація та десеріалізація: Newtonsoft.Json дозволяє конвертувати об'єкти .NET в JSON та обернено. Це особливо корисно при роботі з веб-службами, базами даних та іншими місцями, де використовується обмін даними у форматі JSON;
- гнучкі налаштування: бібліотека надає ряд параметрів та налаштувань для контролю над процесами серіалізації та десеріалізації. Це включає в себе

можливість задавати атрибути, налаштовувати формат виведення, обробляти дефолтні значення та багато іншого;

- робота з LINQ: Newtonsoft.Json підтримує LINQ-запити для отримання доступу до елементів JSON об'єкта. Це дозволяє виконувати різні операції фільтрації, сортування та перетворення даних;
- обробка JSON масивів та об'єктів: бібліотека легко впорається з роботою з масивами та вкладеними об'єктами в форматі JSON.

Newtonsoft.Json є широко використовуваною бібліотекою в галузі розробки C# завдяки своїй простоті використання та високій ефективності. Вона дозволяє розробникам зручно працювати з JSON-даними в своїх додатках та взаємодіяти з різними сервісами та інтерфейсами, які використовують цей формат обміну даними.

3.1.6 Мова запитів SQL

SQL, (Structured Query Language) – це мова запитів, яка використовується для отримання, видалення чи коригування інформації в реляційних базах даних.

Вона надає можливості використання різних ключових слів таких як:

- отримання даних полів таблиць – “select”;
- додавання рядків в таблиці – “insert”;
- оновлення даних в рядку – “update”;
- створення нових таблиць в базі даних – “create”;
- коригування полів таблиць – “alter”;
- видалення таблиць – “drop”.

У запитах можна використовувати ключові слова, наприклад “where”, для фільтрування та “order by” для сортування полів таблиць.

SQL використовує функції “sum”, “avg”, “count”, “max” та “min” для виконання обчислень та агрегацій даних.

Ця мова запитів підтримує можливість поєднання даних з різних таблиць за допомогою ключового слова “join” з опціональним модифікатором “left” або “right” і групування даних по полям за допомогою ключового слова “group by”.

Мова SQL підтримується більшістю систем керування базами даних. Основними СКБД є Microsoft SQL Server, MySQL, PostgreSQL, Oracle Database та SQLite, та інші [19].

3.1.7 Протокол HTTP

HTTP (Hypertext Transfer Protocol) є стандартом для передачі гіпертекстових даних через мережу. Він визначає, як клієнт і сервер взаємодіють під час обміну інформацією на веб-сайтах. HTTP є протоколом без стану, що означає, що кожен запит вважається незалежним від попередніх.

Основними методами HTTP є GET і POST. Метод GET використовується для запиту ресурсу, тоді як POST використовується для відправки даних на сервер для обробки. HTTP також підтримує інші методи, такі як PUT, DELETE, HEAD і OPTIONS, які розширюють можливості взаємодії.

Протокол HTTP базується на моделі клієнт-сервер, в якій клієнт робить запит, а сервер надає відповідь. Він використовує TCP (Transmission Control Protocol) як транспортний рівень для надійної доставки даних.

HTTP підтримує ряд заголовків, які дозволяють клієнту та серверу передавати додаткову інформацію про запит чи відповідь. Заголовки можуть містити інформацію про тип вмісту, кешування, кодування та інші параметри.

Важливою частиною HTTP є статус-коди, які повертаються сервером для позначення результату запиту. Наприклад, код 200 означає успішний запит, а 404 вказує на те, що ресурс не знайдено.

HTTP може бути розширений за допомогою розширень, таких як HTTPS, яке використовує шифрування за допомогою SSL або TLS для безпечної передачі даних. HTTP/2 та HTTP/3 представляють нові версії протоколу з покращеною продуктивністю та ефективністю передачі даних. Також існують версії протоколу, такі як HTTP/1.1 і HTTP/2, які вдосконалюють ефективність та можливості комунікації між клієнтом і сервером.

3.2 Архітектура програмного забезпечення

3.2.1 Сервісно-орієнтована архітектура

Для реалізації даного програмного забезпечення було обрано платформу IT-Enterprise, яка побудована з використанням сервісно-орієнтованої архітектури. Сервісно-орієнтована архітектура (СОА) – це підхід до розробки програмних систем, в якому функціональність програми розглядається як набір послуг, або "сервісів". В основі СОА лежить ідея розбиття програмного застосунку на невеликі, самостійні сервіси, які можуть виконувати конкретні функції та взаємодіяти один з одним через стандартизовані інтерфейси.

Ключовими характеристиками СОА є:

- самостійність сервісів – кожен сервіс функціонує незалежно від інших, що дозволяє їм бути переносимими та перевикористовуваними. Кожен сервіс може бути реалізований, випущений та оновлюваний незалежно від інших;
- стандартизовані інтерфейси – сервіси спілкуються між собою за допомогою чітко визначених стандартизованих інтерфейсів. Це спрощує інтеграцію та взаємодію між сервісами;
- інкапсуляція – кожен сервіс приховує свою внутрішню реалізацію від інших сервісів. Це надає гнучкість у внесенні змін в сервіс без впливу на інші частини системи;
- легка масштабованість – сервіси можуть бути масштабовані незалежно один від одного. Це дозволяє підвищувати продуктивність лише тих сервісів, які вимагають більше ресурсів;
- гнучкість та зручність розширення – додавання нового функціоналу в систему зазвичай вимагає лише створення нового сервісу або модифікацію існуючого, без необхідності перебудови всієї системи;
- ієрархія послуг – система може бути побудована у вигляді ієрархії послуг, де більші абстрактні послуги можуть використовувати менш абстрактні для досягнення своїх цілей.

Принци сервісно-орієнтованої архітектури ви можете побачити на рисунку 3.4.

Рисунок 3.4 – Сервісно-орієнтована архітектура.

Інформація про модулі системи представлена у такому вигляді, що використання цих модулів не потребує розуміння внутрішніх рішень та технологій, що є в них використані. Це сприяє втіленню принципу інкапсуляції даних в системі. Забезпечення обробки даних здійснюється через низку серверів баз даних, які взаємодіють з платформою IT-Enterprise. У якості серверів баз даних використовується Microsoft SQL Server 2014/2016/2017 та Oracle Database 11/12. У простому випадку може бути використаний один сервер баз даних, проте система має можливість масштабування кількості серверів в разі необхідності. Кількість використовуваних серверів баз даних не має жорстких обмежень, що надає можливості для подальшого розвитку системи при зростанні навантаження.

На серверах додатків виконується бізнес-логіка обробки даних. Сервери додатків формують запити до серверу бази даних, виконують розрахунки на основі отриманих даних і пересилають готові результати. Кількість серверів додатків необмежена, що також надає можливості для подальшого розвитку системи, у разі збільшення навантаження.

Для web-доступу і для роботи мобільних додатків чи інших систем, які потребують віддаленого інтернет доступу до серверів, застосовуються web-сервери. Такі сервери служать для взаємодії з клієнтом і серверами додатку.

Кількість web-серверів необмежена, що надає можливості для збільшення кількості серверів в разі збільшення навантаження.

3.2.2 Робота системи

Для забезпечення успішної інтеграції з будь-якою системою документообігу, важливо визначити основні функції, які необхідно підтримувати. Ось деякі з цих функцій:

- відправка документів;
- відправка підписів;
- відправка коментарів;
- отримання нових та змінених документів.

Функція відправки документів дозволяє користувачам надсилати документи контрагентам з інших систем для обробки, підпису або зберігання. Функція відправки підписів дозволяє підписувати документи електронним підписом та забезпечує їх легальну силу. Функція відправки коментарів дає користувачам можливість додавання коментарів до документів для обговорення їх з іншими користувачами. Функція отримання нових та змінених документів дозволяє системі автоматично отримувати оновлені документи або нові документи з інших систем

Всі інтеграції з іншими системами були об’єднані в один інтерфейс і реалізують необхідні функції по-своєму, індивідуально для кожної системи. Для всіх документів, що передаються або отримуються з інших систем, були визначені основні атрибути, такі як унікальний ідентифікатор, власник документа, одержувач, дата створення, тип та інші. Ці атрибути допомагають стандартизувати інформацію та забезпечують її консистентність в усій системі. В базі даних була спеціально створена окрема таблиця для зберігання інформації про всі вхідні та вихідні документи. Ця таблиця включає в себе важливі атрибути кожного документа, такі як ідентифікатор, власник, одержувач, дата створення та інші. Крім того, вона містить

дані про статус документа та можливі помилки, що можуть виникнути під час інтеграції. Були впроваджені додаткові таблиці для коментарів та підписів для забезпечення можливості відстеження додаткової інформації, пов'язаної з кожним конкретним документом. Це дозволяє зберігати контекст та надає більше можливостей для аналізу та моніторингу документообігу в системі.

Для реалізації основного завдання створено додаток Signy Connector, далі – Коннектор. Коннектор встановлюється на сервері Signy і працює без участі користувача. Функції отримання, відправлення та завантаження документів виконуються автоматично через певні проміжки часу. Всі три функції працюють незалежно одна від одної.

Коли користувач Signy надсилає документ через інтеграцію з іншою системою, дані документа зберігаються в таблиці Connector. При відправці документів такий документ буде відправлено в іншу систему. Функція отримання документів отримує та зберігає інформацію про документи з інших систем у таблицях Connector. Функція завантаження документів завантажує документ у Signy відповідно до даних у таблиці. Регулярне виконання цих функцій забезпечує постійний обмін документами між клієнтами Signy та клієнтами інших систем.

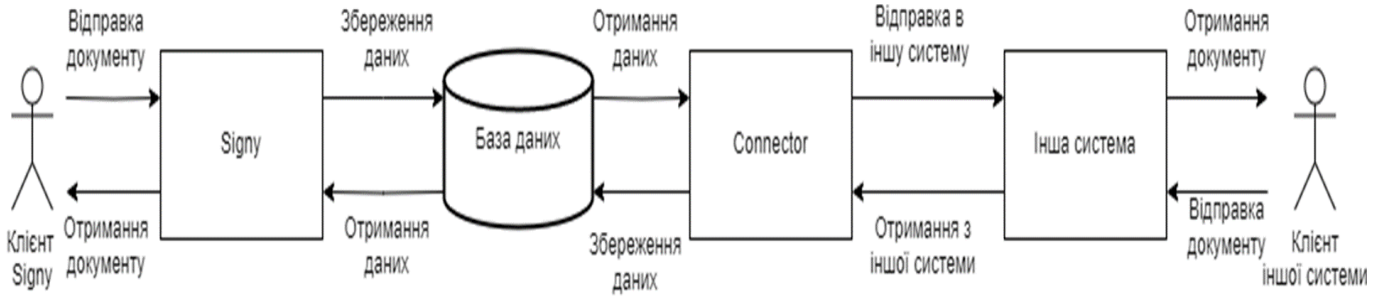

Рисунок 3.5 – Схема роботи конектора

Клієнт Signy відправляє документ контрагенту – клієнту іншого сервісу. Інформація про документ зберігається в таблицях Connector і надсилається в іншу систему за допомогою функції надсилання документа. Контрагент підписує документ і надсилає свій документ клієнту Signy. Інформація про вжиті дії отримується функцією пошуку документів і зберігається в таблицях Connector.

Функція завантаження документів завантажує нові документи та підписи в Signy, де клієнт має до них доступ.

3.2.3 Архітектура коннектора

Для зручної роботи та для забезпечення легкості додавання нових інтеграцій до вже існуючих, всі інтеграції були об’єднані одним уніфікованим інтерфейсом. Створення нової інтеграції з новою системою вимагає реалізації існуючого інтерфейсу конкретними функціями інтеграцій з конкретною системою. Інтерфейс складається з наступних функцій:

– функція отримання загальної інформації по нових та змінених документах GetDocuments. Так як всі системи містять інформацію про дату створення документа і більшість систем має інформацію по даті зміни документа, ця функція містить параметр дати, щоб отримувати нові документи від дати останнього отримання;

– функція отримання підписів по документах з інших систем GetSignatures;

– функція отримання інформації по коментарям по певним документам з інших систем GetComments;

– функція відправки документів в інші системи SendDocuments;

– функція відправки підписів по документах в інші системи SendSignatures;

– функція відправки коментарів по документах в інші системи SendComments.

Діаграма класів представлена на рисунку 3.6.

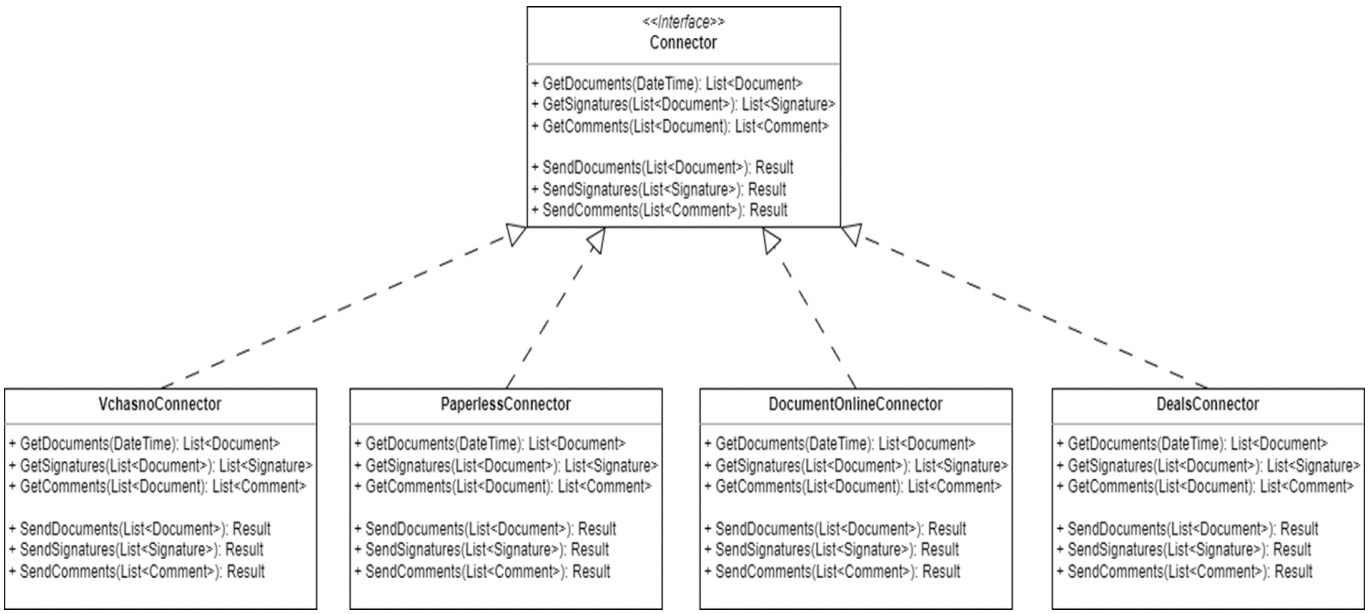

Рисунок 3.6 – Діаграма класів

Платформа IT-Enterprise надає можливість створення та налаштування планувальників – програмного коду з певною періодичністю виконання. Для роботи конектора було створено планувальники надсилання даних та планувальники отримання даних.

Планувальник надсилання даних отримує з бази даних інформацію про документи, які мають бути відправлені, інформацію про налаштування користувача, який відправляє документ, та інформацію про систему, в яку має бути відправлено документ. Далі планувальник викликає відповідні реалізації функцій інтерфейсу SendDocuments, SendSignatures та SendComments для системи з якою відбувається інтеграція. Інформація про успішність або помилку при відправці даних записується у відповідні документу поля бази даних у таблиці INTDOCS. Схема роботи представлена на рисунку 3.7.

Рисунок 3.7 – Схема роботи планувальника надсилання даних

Планувальник отримання даних отримує інформацію про налаштування користувача з бази даних та викликає відповідні реалізації функцій інтерфейсу GetDocuments, GetSignatures, GetComments. Інформація про отримані дані зберагіється у відповідних таблицях. Так як підписи та коментарі отримуються для кожного документу, а також отримання документу при наявності змін повертає повні дані по документу, то виникає ситуація отримання одних даних декілька разів. Для коректної обробки подібних ситуацій таблиці документів, підписів та коментарів містять унікальні ідентифікатори даних з інших систем – при отриманні однакових даних в таблиці вже буде запис з таким самим унікальним ідентифікатором, що дозволяє коректно обробляти подібні випадки (оновлювати інформацію чи ігнорувати за відсутності змін). Схема роботи представлена на рисунку 3.8.

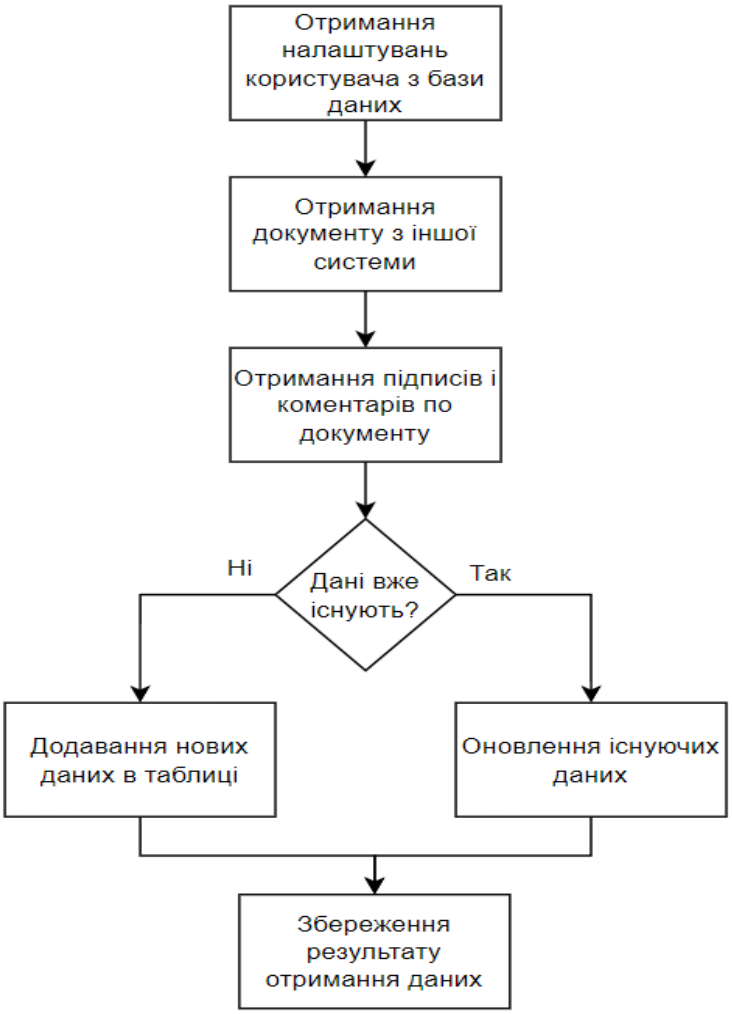

Рисунок 3.8 – Схема роботи планувальника отримання даних

Основною таблицею бази даних конектора є таблиця INTDOCS. Ця таблиця, як було зазначено вище, зберігає в собі основну інформацію про документ та складається з наступних полів:

– ID – унікальний номер рядка в таблиці;

– NAME – назва файлу;

– TYPE – тип документу (вхідний/вихідний);

– SYSTEM – ознака системи у яку має бути відправлений документ або з якої документ був отриманий;

– SENDED – ознака того, що документ вже було відправлено в іншу систему;

– USERID – ідентифікатор користувача власника документу;

– IDEXT – унікальний ідентифікатор документу в іншій системі;

– SENDER – назва організації відправника документу;

– ORGSENDER – ОКПО/ІНН організації відправника документу;

– DOCTYPE – тип документу (акт/рахунок/накладна чи інше);

– LOADED – ознака того, що документ було успішно завантажено на Signy;

– ERRMSG – повідомлення про можливу помилку при роботі з документом;

– LINK – посилання на документ в іншій системі;

– ORGOWNER – ОКПО/ІНН організації власника документу;

– DATEADD – дата запису інформації в базу даних.

Таблиця INTFILES складається з наступних полів:

– DOCID – id документу в таблиці INTDOCS;

– FILEPATH – інформація про місце зберігання самого файлу;

– FILENAME – ім’я файлу (іноді ім’я окремих файлів не співпадають з іменем документу);

– DATEADD – дата запису інформації в базу даних.

Таблиці коментарів та підписів містять інформацію про сам коментар/підпис, унікальний ідентифікатор та ID з таблиці INTDOCS для однозначного визначення документу, до якого вони відносяться.

3.3 Структура бази даних конектора

Сервери баз даних в системі IT-Enterprise розташовані в безпечній мережевій зоні, до якої має доступ лише адміністратор під час налаштування системи. Ці сервери баз даних доступні лише для серверів додатків і web-серверів. Зазвичай, з метою максимального забезпечення безпеки, сервери баз даних навіть не мають фізичного зв'язку з зовнішнім світом.

Обмін даними між серверами додатків і базою даних відбувається за протоколом TCP/IP. Сервер додатків встановлює з'єднання з серверами баз даних, відправляє SQL-запити і отримує відповіді з даними, які повернув SQL-сервер.

Дані на різних серверах баз даних можуть бути сегментовані (розділені на частини на різних серверах), зеркальовані (забезпечення паралельного доступу до дзеркальних копій), розподілені серед різних служб (підтримка єдиної бази даних для різних комплектів тощо). Всі ці технології взаємодії дозволяють необмежене масштабування на рівні баз даних. Схема бази даних представлена на рисунку 3.9.

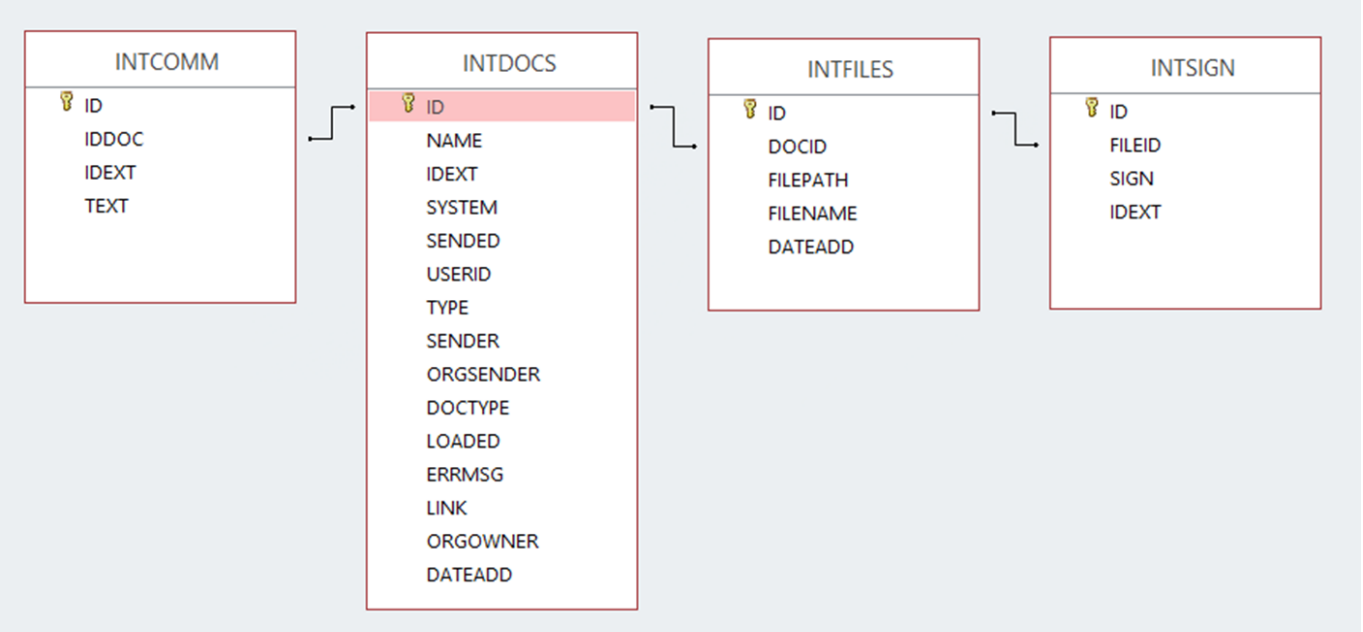

Рисунок 3.9 – Схема бази даних

Основною таблицею бази даних конектора є таблиця INTDOCS (Integration documents). Ця таблиця, як було зазначено вище, зберігає в собі основну інформацію про документ та його поточний стан. Однак, оскільки іноді один документ може включати в себе кілька файлів, таких як xml-структура та pdf-відображення, було важливо створити додаткову таблицю INTFILES. Ця таблиця розширює можливості зберігання інформації, яка стосується кожного конкретного файлу в межах окремого документа, включаючи його поточний стан. Крім того, для ефективного зберігання підписів була введена окрема таблиця INTSIGN. У цій таблиці зберігаються коди підписів та унікальні ідентифікатори, що дозволяє чітко відстежувати та ідентифікувати різні підписи, пов'язані з документами.

Щоб надати можливість додаткового контексту та обгрунтування, використано окрему таблицю для коментарів, а саме INTCOMM. Ця таблиця вміщує в собі текст

коментарів та унікальні ідентифікатори, що допомагають ефективно відслідковувати та керувати коментарями, що стосуються конкретних документів.

Невід'ємною частиною цього дизайну бази даних є забезпечення зв'язку між основною таблицею та усіма допоміжними таблицями за допомогою поля ID таблиці INTDOCS, що є ідентифікатором документа в основній таблиці. Це забезпечує здійснення зв'язку та координацію інформації між різними таблицями бази даних.

3.4 Тестування

Тестування – це процес перевірки програмного забезпечення з метою забезпечення його правильності, ефективності та відповідності вимогам. Цей процес включає в себе виконання програм або систем з метою виявлення потенційних помилок та вдосконалення їх якості. Тестування допомагає гарантувати, що програмне забезпечення працює так, як очікується, та взаємодіє із користувачем чи іншими системами належним чином. Розглянемо існуючі види тестування:

Модульне тестування – це процес, який охоплює перевірку окремих компонентів або модулів програмного забезпечення. Головною метою цього виду тестування є визначення правильності та коректності функцій, які виконуються цими модулями. В процесі модульного тестування інженери спрямовують свої зусилля на те, щоб впевнитися в тому, що кожен модуль програми працює самостійно та відповідає специфікації. Під час модульного тестування використовуються різноманітні техніки, такі як юніт-тести, які перевіряють окремі функції чи методи. Цей підхід дозволяє виявити можливі помилки та недоліки в роботі конкретних частин програми на ранніх етапах розробки, що сприяє подальшому поліпшенню якості коду.Модульне тестування забезпечує ізоляцію окремих компонентів для перевірки їх функціональності та ефективності. Кожен модуль тестується незалежно, що робить його досить ефективним методом виявлення та виправлення помилок, особливо на ранніх етапах розробки програмного забезпечення.

Інтеграційне тестування – це етап в процесі тестування програмного забезпечення, який фокусується на перевірці взаємодії між різними модулями або

компонентами системи. Основна мета цього виду тестування полягає в перевірці того, як різні частини програми виконують взаємодію між собою та чи інтегруються правильно для досягнення бажаного функціоналу. Інтеграційне тестування важливе для виявлення проблем, які можуть виникнути при об'єднанні окремих модулів у єдину систему. Під час цього процесу тестери переконуються, що дані правильно передаються між компонентами, і що система в цілому працює так, як очікується.Завдання інтеграційного тестування – це перевірка взаємодії різних частин програми, виявлення та виправлення конфліктів, які можуть виникнути при їх спільній роботі. Це допомагає забезпечити стабільну та ефективну роботу системи в цілому.

Функціональне тестування – це тестування програмного забезпечення, спрямоване на перевірку того, чи відповідає функціонал програми визначеним вимогам та очікуванням користувачів. Під час цього виду тестування перевіряється, як елементи системи взаємодіють між собою для забезпечення правильної роботи програмного продукту.Основна мета функціонального тестування – визначення того, чи виконує програма ті функції, для яких вона була створена, та чи відповідає очікуванням користувачів. Тестувальники переконуються, що всі функції працюють правильно, виконуючи тестові сценарії, які покривають різні аспекти функціональності програми. Під час функціонального тестування перевіряються такі аспекти, як правильність введення та обробки даних, коректність виведення результатів, відповідність програми визначеним стандартам та вимогам. Також перевіряється взаємодія програми з іншими системами та елементами оточення. Завдання функціонального тестування – гарантувати, що програмний продукт виконує ті функції, які передбачені в специфікації, та що користувач отримує очікуваний результат при використанні програми.

Відмовостійке тестування – це процес оцінки стійкості та надійності програмного забезпечення під час його роботи в умовах реального використання. Основна мета цього виду тестування полягає в виявленні можливих відмов або ситуацій, коли програма не здатна коректно виконувати свої функції. Під час відмовостійкого тестування перевіряються реакція програми на непередбачені ситуації та навантаження, що перевищують нормальні умови роботи. Важливим

аспектом є також перевірка часу відновлення програми після виникнення відмови та її здатність до автоматичного відновлення роботи. Цей вид тестування допомагає виявити слабкі місця в програмному продукті, які можуть призвести до його відмови або некоректної роботи в реальних умовах експлуатації. Також відмовостійке тестування дозволяє здійснити оцінку продуктивності та надійності системи під час тривалого періоду функціонування. Завдання відмовостійкого тестування – забезпечити високу робочу стабільність програмного забезпечення та зменшити ймовірність виникнення відмов при його використанні в реальних умовах.

Тестування продуктивності є процесом, спрямованим на визначення та оцінку ефективності та швидкодії програмного забезпечення під час виконання різних завдань та умов навантаження. Його основна мета полягає в тому, щоб визначити, як швидко та ефективно система відповідає на різні сценарії використання та обсяги даних. У процесі тестування продуктивності оцінюються такі параметри, як швидкість відгуку системи, пропускна здатність, ефективність роботи при великому обсязі даних або користувачів, а також ресурсоємність програми. Також важливим є виявлення можливих проблем та неефективностей у роботі системи під високими навантаженнями. Тестування продуктивності дозволяє визначити, чи може програма витримати роботу в реальних умовах, забезпечуючи необхідну продуктивність та відповідаючи очікуванням користувачів. Воно також допомагає виявити та усунути можливі проблеми, пов'язані з низькою продуктивністю або надто великим споживанням ресурсів. Такий вид тестування важливий для забезпечення оптимальної роботи системи під час її експлуатації в умовах реального використання.

Регресійне тестування – це тестування програмного забезпечення, спрямоване на перевірку того, чи не виникли нові помилки або проблеми в програмі після внесення змін, оновлень чи виправлень в існуючий код. Основна ідея цього виду тестування полягає в тому, щоб переконатися, що зміни не вплинули негативно на функціональність та стабільність системи. Під час регресійного тестування проводять повторні тести для перевірки тих самих функцій, які раніше працювали коректно, а також тестують нові частини коду, які були внесені. Такий підхід допомагає виявити можливі взаємодії між різними частинами програми та уникнути

появи помилок через внесення змін. Регресійне тестування є важливою частиною життєвого циклу розробки програмного забезпечення, оскільки нові функції, виправлення помилок чи оновлення можуть впливати на вже існуючий функціонал. Забезпечуючи стабільність та надійність програми, регресійне тестування грає ключову роль у забезпеченні якості програмного продукту.

Для тестування роботи системи використовуються всі вищеописані види тестування. Також використовується ручне тестування.

Висновки до розділу 3

В цьому розділі описані засоби програмної реалізації та технології які використовувались для створення необхідного програмного забезпечення. Також розділ містить детальний опис архітектура програмного забезпечення та опис структури основних таблиць бази даних.

Були детально описані типи тестування які використовуються для перевірки правильності роботи створеного програмного забезпечення.

4 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

4.1 Веб-інтерфейс

Для зручного налаштування інтеграцій системи Signy з іншими системами ЕДО було доопрацьовано меню “Управління профілем” (рис. 4.1).

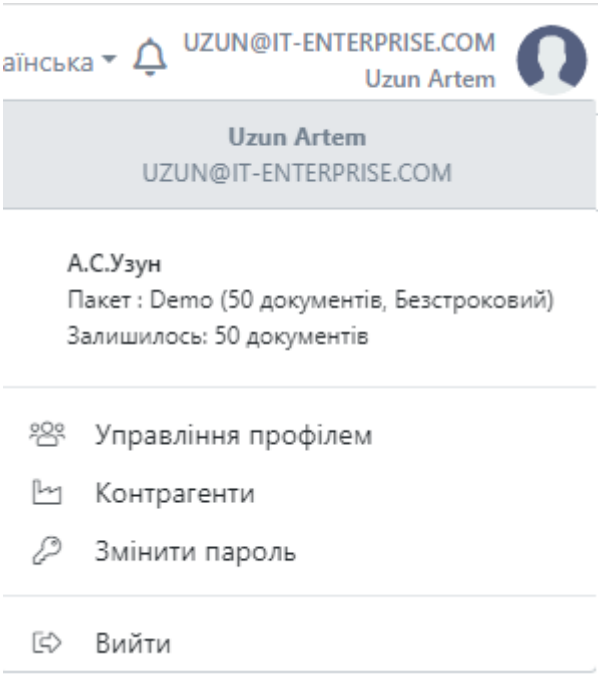

Рисунок 4.1 – Меню користувача

На вкладці “Налаштування організації” було додано пункт меню “Налаштування інтеграції” (рис 4.2). Як видно на рисунку 4.2, система має зручний, «user friendly»-інтерфейс, на якому розміщені прапорці для ввімкнення інтеграцій з різними системами та набір налаштувань, необхідних для інтеграції з відповідною системою (зазвичай токен або пароль). Використовуючи це меню можна вмикати та вимикати інтеграції для своєї організації.

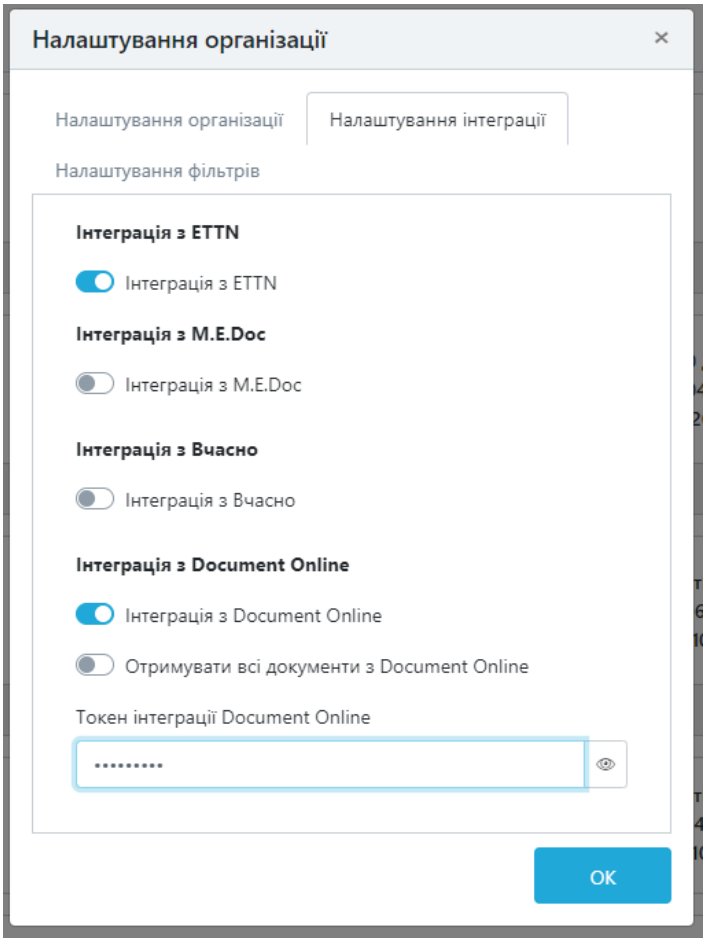

Рисунок 4.2 – Меню інтеграцій

Для налаштування контрагентів для отримання документів через інтеграції було створено довідник контрагентів рисунок 4.3.

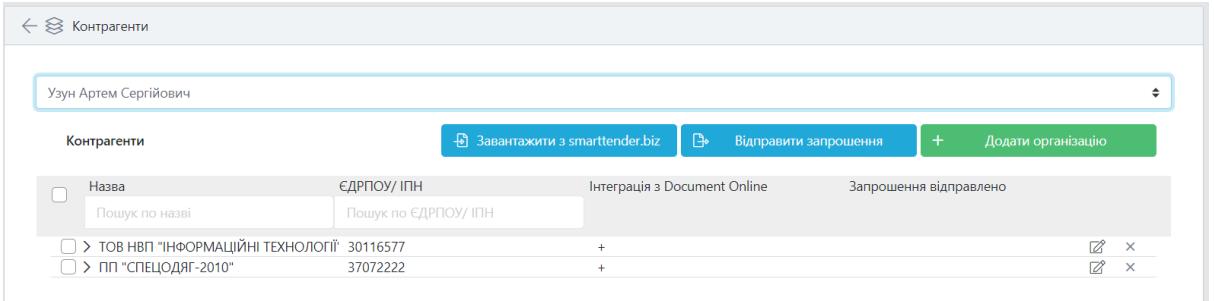

Рисунок 4.3 – Довідник контрагентів

Інтерфейс довідника контрагентів представляє собою таблицю в якій користувачу відображаються дані про контрагента, та про параметри, які вказав користувач при додаванні чи редагуванні контрагента. Користувач має можливість додавання нових контрагентів, змінення доданих та видалення вже не актуальних даних. При додаванні чи зміні даних користувач має можливість вказання систем документообігу, які користувач бажає використовувати для відправки документа конкретному отримувачу. Вікно додавання та коригування контрагента представлено на рисунку 4.4. Складається з таких елементів:

– поле для введення ЄДРПОУ чи ІПН контрагента;
– поле для введення назви підприємства контрагента. Поле автоматично заповнюється на основі публічних даних реєстру при введенні ЄДРПОУ чи ІПН;
– ознака інтеграції з Document Online – обравши дану ознаку користувач буде отримувати документи від вказаного контрагента з сервісу Document Online. Також відправлені цьому контрагенту документи, будуть відправлені через сервісі Document Online.

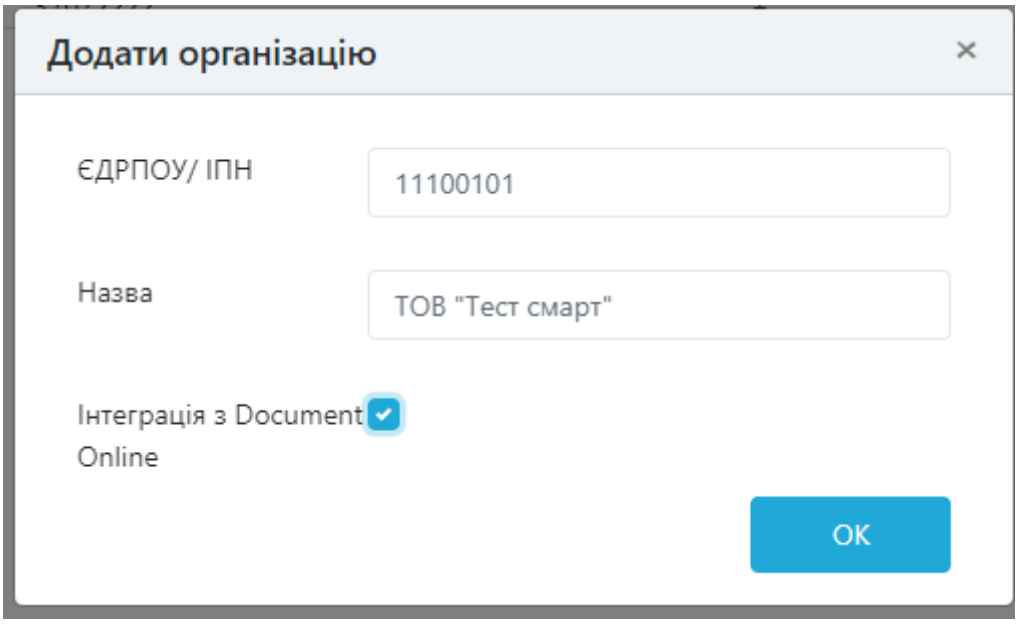

Рисунок 4.4 – Додавання контрагенту

Розглянемо вікно налаштування маршрутів(рис. 4.5), яке дає змогу вказати параметри маршруту, такі як:

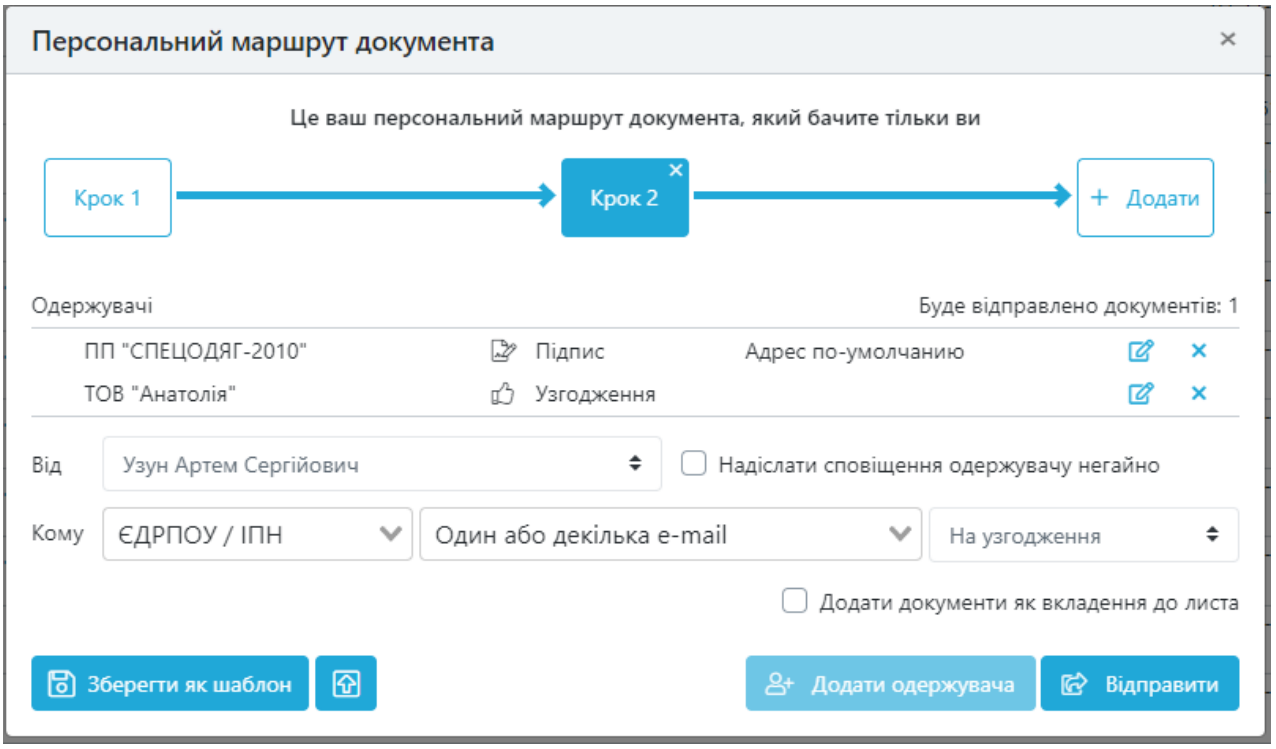

Рисунок 4.5 – Вікно налаштування маршруту.

1. Кроки маршруту. Користувач може створювати довільну кількість кроків маршруту для документа. В кожному кроці можна створювати декілька одержувачів, наприклад, різних отримувачів документу. Після створення маршруту кроки будуть виконуватись почергово;
2. Одержувачі. Для кожного кроку треба вказати одержувачів документу на цьому кроці. Це може бути як одна компанія з різними діями по документу, так і різні компанії;
3. Реквізити одержувачів. Користувач може налаштувати параметри одержувача, такі як:
   3.1. Код організації одержувача. Код ЄДРПОУ або ІНН отримувача документа. Може вказуватись разом з email або самостійно;
   3.2. Email. Адреса електронної пошти отримувача документа або декілька email при відправці на конкретного користувача, а не на

організацію в цілому. Може вказуватись разом з кодом організації або без нього;

3.3. Тип відправки документа. Цей реквізит означає тип дії який очікується від отримувача документа. Може приймати такі значення:

3.3.1. На підпис;

3.3.2. На узгодження;

3.3.3. Для перегляду;

3.3.4. Редагування;

3.3.5. Редагування і узгодження;

3.3.6. Підпис і печатка;

3.3.7. На печатку;

4. Кнопка зберегти як шаблон. Дає користувачу можливість збереження налаштованого маршруту як шаблону для відправки інших документів;
5. Кнопка завантаження маршруту з шаблону. Дозволяє користувачу завантажити раніше збережений шаблон маршруту;
6. Кнопка відправити. Зберігає маршрут документа, та відправляє документ.

4.2 Сценарій роботи користувача з системою

Для тестування обміну документів з іншою системою було обрано систему Document.Online. Для цього був створений аккаунт в даній системі та налаштовано інтеграцію між даною системою та Signy. Для цього було отримано токен інтеграції з системою Document.Online на сторінці налаштувань користувача (рис. 4.6).

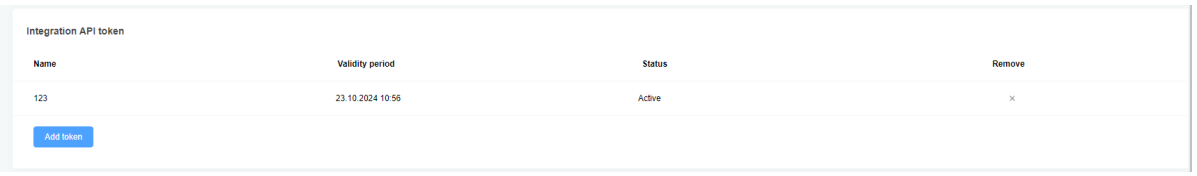

Рисунок 4.6 – Токен в налаштуваннях Document Online

В налаштуваннях профілю був доданий токен інтеграції (рис. 4.7)

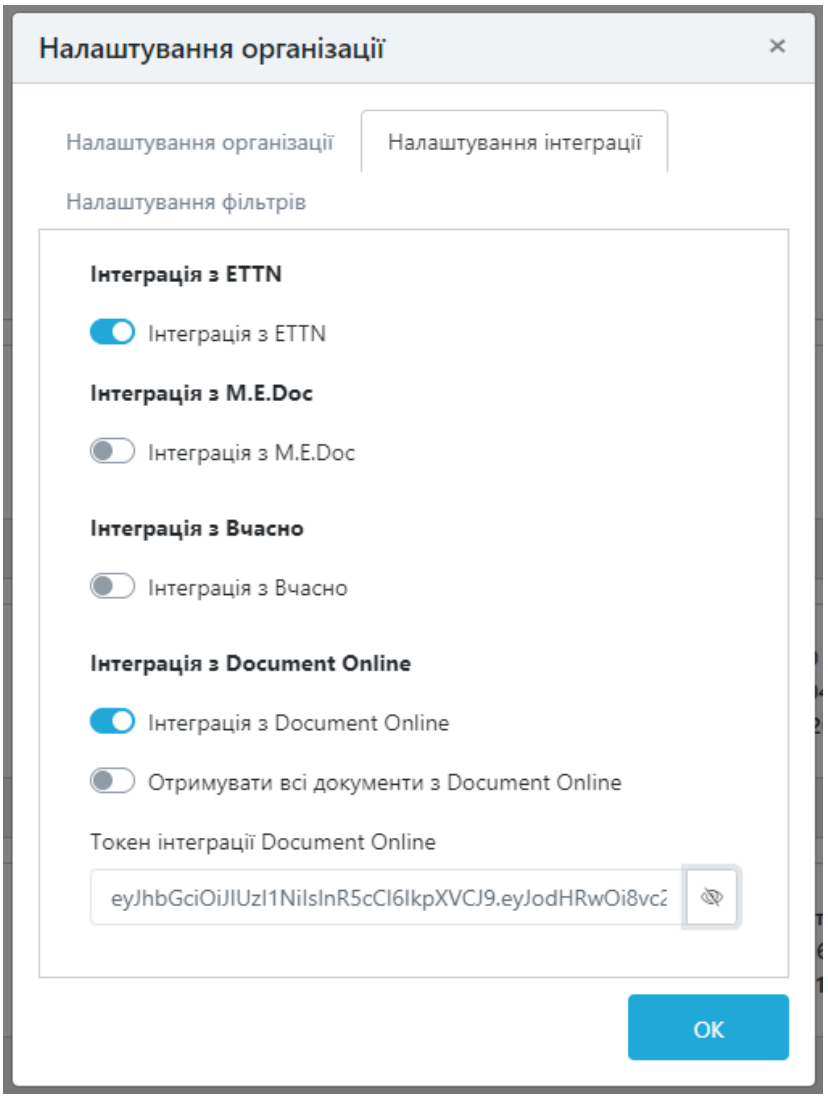

Рисунок 4.7 – Токен в налаштуваннях профілю

Після збереження налаштувань можна побачити повідомлення про успішне збереження даних (рис.4.8)

Рисунок 4.8 – Повідомлення про успішне збереження налаштувань

На сторінці контрагентів було налаштовано інтеграцію з Document Online для тестової організації. Тепер при відправці документа на цю організацію, буде здійснюватись відправка в систему Document Online (рис. 4.9)

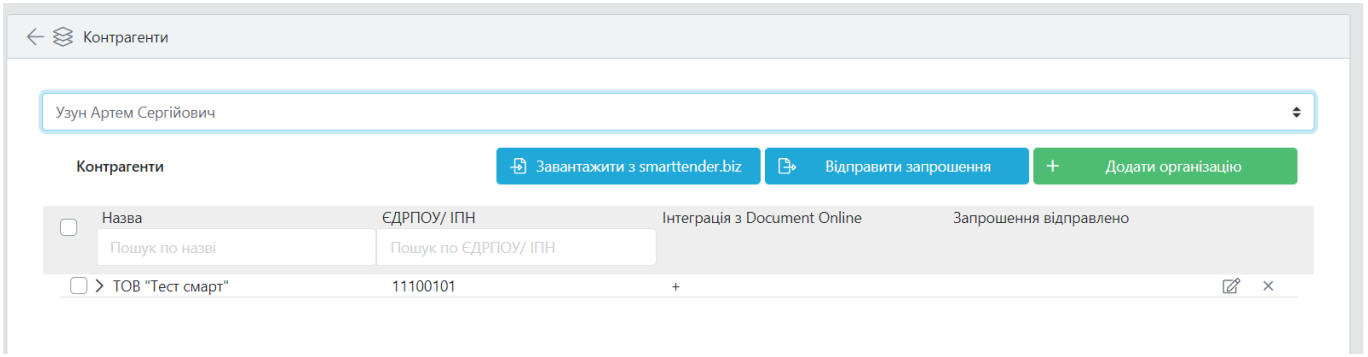

Рисунок 4.9 – Налаштування на сторінці контрагентів

В системі Signy було створено документ з назвою “test” (рис. 4.10).

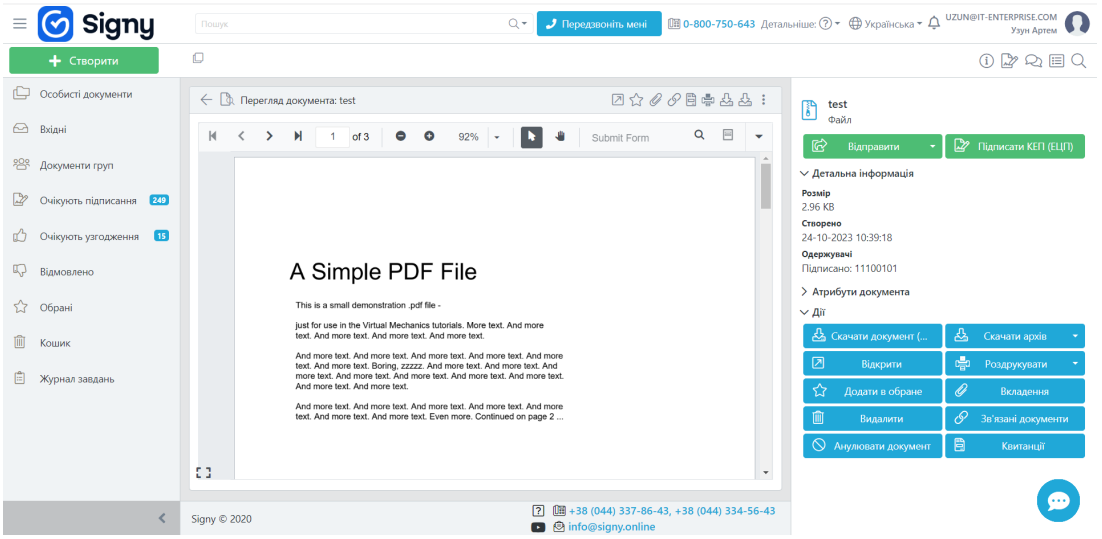

Рисунок 4.10 – Документ в системі Signy

Цей документ було відправлено на контрагента в системі Document.Online. Документ було отримано та підписано контрагентом. Вигляд цього документу в системі Document Online на рисунку 4.11.

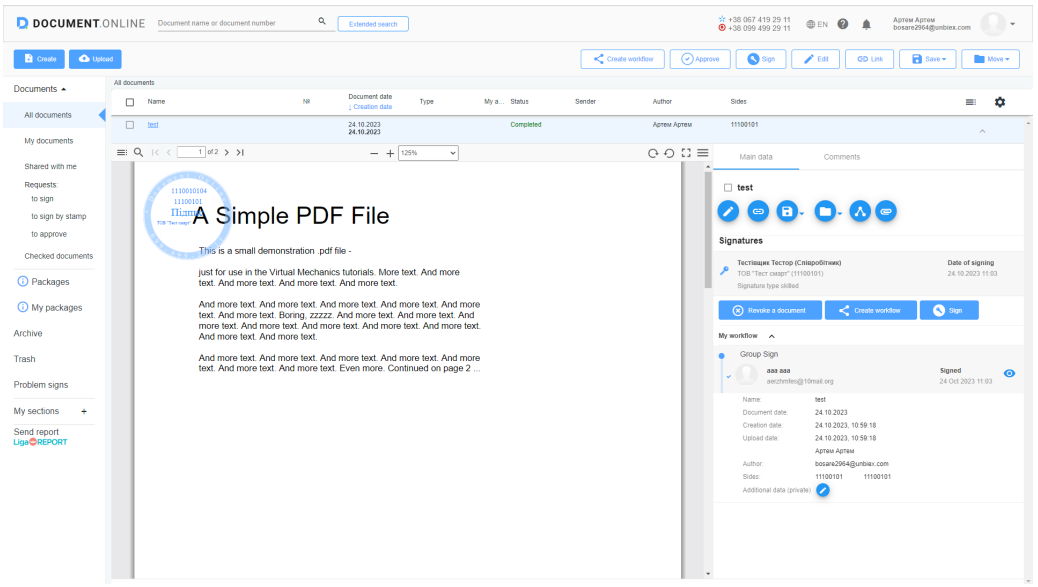

Рисунок 4.11 – Документ в системі Document.Online

Підпис було отримано користувачем у системі Signy (рис. 4.12).

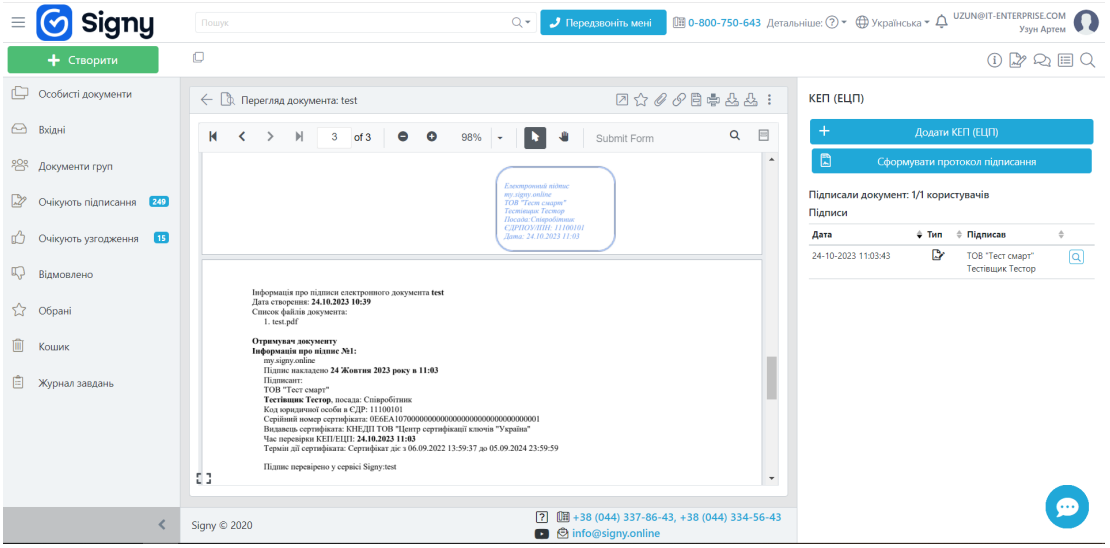

Рисунок 4.12 – Документ в системі Signy з підписом

4.3 Висновки до розділу 4

В даному розділі детально описано інтерфейс користувача сервісу Signy, а саме: меню користувача, меню управління інтеграціями та сторінку довідника

контрагентів. Також було детально описано інтерфейс створення персонального маршруту документа, описано необхідні для створення маршруту налаштування. Наведено сценарій використання програмного забезпечення для відправки документа в іншу систему користувачем. Під час роботи над розділом було виявлено, що програмне забезпечення працює стабільно, інтерфейс є зрозумілим та зручним у використанні.

Так як продукт є частиною веб сервісу Signy, користувачу не треба виконувати ніяких додаткових дій для інсталяції програмного забезпечення.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Даний розділ має на меті проведення маркетингового аналізу стартап проекту для визначення можливостей його ринкового впровадження та визначення можливих напрямків реалізації цього впровадження.

5.1 Опис ідеї системи

Прогрес технологій, який переживає наш сучасний світ, суттєво полегшує повсякденне життя людей у всіх сферах. Проте реальність вказує на те, що із покращенням ситуації виникає постійне зростання вимог, пов'язаних з інтеграцією технологій у різні сфери. Різноманітні компанії змушені впроваджувати новаторські підходи до автоматизації роботи, щоб залишатися конкурентоспроможними та уникати витрат часу працівників на монотонну рутинну роботу. Однією з таких сфер, яка пройшла автоматизацію, є сфера документообігу. З великим попитом на системи документообігу з'явилася багато цих систем, і багато компаній використовують різні платформи. Це створює додаткові завдання для працівників, оскільки вони змушені одночасно працювати з різними системами. Оскільки цей процес є досить монотонним і в основному пов'язаним з переміщенням документів з однієї системи в іншу, це значно збільшує ймовірність помилок через одноманітність роботи та великий обсяг документів. З метою автоматизації процесу було проведено дане дослідження та було розроблено програмний продукт.

Таблиця 5.1 – Опис ідеї стартап проекту

Зміст ідеї (що пропонується)	Напрямки застосування	Вигоди для користувача
Розробка програмного забезпечення, що дозволяє клієнтам сервісу Signy працювати з контрагентами з інших систем ЕДО	1. Надсилання та отримання документів, підписів та коментарів контрагентам, що використовують системи електронного документообігу відмінні від системи користувача 2. Збереження всіх документів підприємства в одній системи документообігу	1. Всі документи підприємства зберігаються в одному місці. 2. Підприємство не залежить від конкретної системи електронного документообігу і може обмінюватись документами з контрагентами з будь якої системи.

Отже, користувачеві надається новий автоматизований спосіб обміну документів при використанні контрагентом системи електронного документообігу відмінної від системи користувача.

Розглянемо техніко-економічні переваги такого проекту відносно ручного обміну документів в інші системи:

– визначимо перелік техніко-економічних властивостей та характеристик ідеї;

– проведемо збір інформації щодо значень техніко-економічних показників ідеї власного проекту та ручної роботи;

− проведемо порівняльний аналіз показників: для власної ідеї визначимо показники, що мають а) гірші значення (W); б) аналогічні (N) значення; в) кращі значення (S) (таблиця. 5.2).

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ з/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W	N	S
		Власний проект	Ручна праця			
1.	Економічність	Вимагає купівлі пакету інтеграція на системі Signy та в системі з якою планується документообіг	Вимагає купівлі пакету на Signy та пакету в іншій системі		Так	
2.	Швидкодія	Відправка та отримання змін в межах 20 хвилин	Відправка та отримання змін відбувається миттєво, однак потрібен час на переніс даних з однієї системи в іншу	Так		
3.	Впорядкованість	Всі документи зберігаються на одному аккаунті	Документи зберігаються в декількох аккаунтах			Так
4.	Простота використання	Користувачу треба навчитися та звикнути працювати в одній системі документообігу	Користувач має працювати в різних системах			Так

Після аналізу слабких, нейтральних і сильних сторін ідеї товару було зроблено висновок, що дана система може вважатися конкурентоспроможною згідно умов поставленого завдання.

5.2 Технологічний аудит ідеї проекту

У цьому розділі представлено аудит технологій, за допомогою яких можна виконати реалізацію ідеї проекту. Технологічна здійсненність ідеї проекту представлена в таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ з/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1.	Розширювана база даних	SQL, MS SQL Server	Наявна	Доступна
2.	Спільний інтерфейс для реалізації інтеграцій	Мова програмування C#, Microsoft Visual Studio	Наявна	Доступна
3.	Доступ до API інших систем документообігу	HTTP запити	Наявна	Доступна
4.	Інтерфейс для користувача	HTML/CSS, JavaScript	Наявна	Доступна
5.	Планувальники отримання та надсилання даних	Платформою IT-Enterprise	Наявна	Доступна

З даних, поданих у таблиці було зроблено висновок, що з технологічної точки зору, реалізація програмного забезпечення з запланованим функціоналом здійсненна.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначимо можливості на ринку, які можна використовувати під час впровадження проекту, а також виявимо загрози на ринку, які можуть ускладнити його реалізацію. Аналіз ринку виявляється ключовою частиною процесу розробки стартапу, оскільки вплив ринкової динаміки визначає майбутній успіх створеного продукту.

Якщо створюється продукт, який не має аналогів, важливо включити в стратегію пояснення користувачеві процесів та суті продукту. Стартап-проекти такого роду визначаються як найважчі для створення і рідко досягають успіху, навіть порівняно з іншими стартапами, де відсоток успіху в цілому низький.

Проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку та структуруємо дані в таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ з/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	1
2.	Загальний обсяг продаж, грн/ум.од	4800
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу	Немає

Кінець таблиці 5.4

5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі (або по ринку), %	300%

Наступним кроком буде визначення потенційних груп клієнтів та їх характеристик та формування орієнтовних вимог до товару для кожної з груп (таблиця 5.5).

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ з/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Сторення інтеграцій до різних систем документообігу	Компанії користувачі сервісу Signy	Різні компанії можуть мати необхідність інтеграцій з різними системами або з декількома системами одночасно	Швидкий обмін даних з різними системами; доступна ціна; зрозумілий інтерфейс налаштування

Стартап має одну можливостість для монетизації – продаж програмного забезпечення клієнтам користувачам системи Signy.

Проаналізовані фактори потенційних загроз та можливостей для майбутнього стартап проекту. Результати наведені у таблицях 5.6 і 5.7.

Таблиця 5.6 – Фактори загроз

№ з/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Невідомість	Клієнти надають перевагу старій ручній праці	Розширення реклами та маркетингу
2.	Консерватизм	Небажання клієнтів переходити на автоматизовану технологію	Розширення реклами та маркетингу

Таблиця 5.7 – Фактори можливостей

№ з/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Слабка конкуренція	Інші системи електронного документообігу надають можливості для інтеграції з ними, однак не надають власних механізмів інтеграції з іншими системами	Створення інтеграцій з іншими системами

Кінець таблиці 5.7

2.	Партнерство з великими компаніями користувачами	Можливість дізнатись потреби користувачів, які більше всього будуть користуватись продуктом	Реалізація потреб користувачів продукту

Після проведення аналізу можливостей було проведено ступеневий аналіз конкуренції на ринку (таблиця 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії щоб бути конкурентоспроможними)
1. Вказати тип конкуренції монополія/олігополія/ монополістична/чиста	Монополістична конкуренція, так як великі компанії можуть створювати власні інтеграції	Постійне вдосконалення продукту
2. За рівнем конкурентної боротьби: локальний/національний	Українські конкуренти, так як основні гравці на ринку – українські компанії	Розвиток в українській ІТ сфері та вихід на міжнародний ринок
3. За галузевою ознакою – міжгалузева/ внутрішньогалузева	Внутрішньогалузева, адже діяльність систем зосереджена на електронному документообігу	Розробка ПЗ, що є вузьконаправленим

Кінець таблиці 5.8

4. Конкуренція за видами товарів: — товарно-родова — товарно-видова — між бажаннями	Товарно-видова, через конкуренцію між програмними продуктами одного виду	Розробити програмний продукт, враховуючи побажання клієнтів
5. За характером конкурентних переваг: цінова / нецінова	Цінова, адже сервіс є платним для використання користувачів	Підтримання високої конкуренції за рахунок унікальних переваг для користувача
6. За інтенсивністю: марочна/не марочна	Не марочна конкуренція	Розробка надійного та ефективного програмного продукту

Наступним кроком є більш детальний аналіз умов конкуренції за моделлю М. Портера (таблиця 5.9).

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	-	Document Online, Paperless	Доступ до системи постачається за рахунок мережі Інтернет	Користувачі сервісу Signy	Власні продукти великих компаній

Кінець таблиці 5.9

Висновки:	Прямі конкуренти відсутні	Наявні потенційні конкуренти	На розробку даного продукту постачальники не мають впливу, так як розробка не потребує платних засобів розробки	Клієнти вказують вимоги згідно з власних потреб	Абсолютні товари замінники відсутні, присутні лише власні рішення окремих компаній рішення.

Враховуючи результати ступеневого аналізу та аналізу за М. Портером було зроблено висновок, що ймовірність успіху при виході на ринок є високою.

На основі аналізу конкуренції з урахуванням характеристик ідеї проекту, вимог споживачів до продукту та факторів маркетингового середовища було визначено перелік факторів конкурентоспроможності (таблиця 5.10).

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ з/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Потреби споживачів	Необхідність створення продукту визначається вимогами споживачів. Якісний аналіз дозволяє розробити чітке та повне ТЗ.

Кінець таблиці 5.10

2.	Ціна та собівартість	Використання безкоштовних засобів розробки дозволяє поставити низьку ринкову ціну
3.	Пропозиція функцій, відсутніх у конкурентів	Послуги які пропонуються не надаються основними конкурентами сервісу Signy

Порівняльний аналіз слабких та сильних сторін системи детально описаний в таблиці 5.11:

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін системи

№ з/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів конкурентів у порівняні з даним продуктом						
			-3	-2	-1	0	1	2	3
1.	Потреби споживачів	15				+			
2.	Ціна та собівартість	10			+				
3.	Пропозиція функцій, відсутніх у конкурентів	20		+					

Фінальним етапом ринкового аналізу можливостей впровадження даного проекту є складання SWOT-аналізу (матриці аналізу сильних та слабких сторін, загроз та можливостей), на основі виділених ринкових загроз та можливостей, сильних і слабких сторін. (таблиця 5.12)

Таблиця 5.12 – SWOT- аналіз стартап-проект

Сильні сторони: Інтеграція з різними системами	Слабкі сторони: Витрати на розробку нового функціоналу
Можливості: Відсутність повноцінних альтернатив Вихід на нові ринки	Загрози: Недопрацювання на початкових етапах може відштовхнути клієнтів.

Альтернативи ринкового впровадження описано в таблиці 5.13

Таблиця 5.13 – Альтернативи ринкового впровадження

№ з/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Створення програмного забезпечення, що буде функціонувати лише в сервісі «Signy».	95%	2 місяці
2.	Створення універсального програмного забезпечення, що буде працювати для будь якої системи.	5%	6 місяців

Після аналізу альтернатив було обрано першу, так як вона є найбільш оптимальною за часом та має високу ймовірність отримання ресурсів

5.4 Розроблення ринкової стратегії проекту

У розробці ринкової стратегії першим кроком передбачається визначення стратегії охоплення ринку – опис цільових груп потенційних споживачів (таблиця 5.14).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ з/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1.	Користувачі сервісу Signy	Готові	середній	Мала конкуренція	Легкий

Основною цільовою групою є користувачі сервісу Signy, так як саме для цієї групи споживачів наявна можливість повного задоволення вимог до програмного продукту. У даної групи є середній попит та майже відсутня конкуренція, простота входу у сегмент – легка.

Для проведення роботи в обраному сегменті ринку необхідне формування базової стратегії розвитку, що описана в таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ з/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Створення програмного забезпечення для використання в сервісі Signy	Стратегія диференціації	Низька ціна; відповідність потребам користувачів	Стратегія спеціалізації

Далі необхідне визначення стратегії конкурентної поведінки(таблиця 5.16).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ з/п	Чи є проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Так	Компанія буде шукати нових споживачів.	Ні	Стратегія заняття конкретної ніші

Визначення стратегії позиціонування стартап-проекту з описом вимогами до готового продукту від цільової аудиторії детально описано в таблиці 5.17:

Таблиця 5.17 – Визначення стратегії позиціонування

№ з/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Низька вартість, зручність у використанні, надійність	Стратегія спеціалізації	Низька ціна; відповідність потребам користувачів	Стабільність, надійність, зручність.

Цей розділ породжує конкретну систему рішень, що визначає, як компанія буде діяти на ринку та у якому напрямку буде розвиватися.

5.5 Розроблення маркетингової програми стартап-проекту

В процесі розроблення ефективної маркетингової програми важливим першим кроком стає ретельне визначення концепції товару, яку планується представити на ринок та який повиннен здобути схвалення та інтерес споживачів.(таблиця 5.18). Для цього проаналізуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ з/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Потреба обміну документами з контрагентами, що використовують системи електронного документообігу відміну від Signy	Автоматичний обмін документами між користувачами системи Signy і їх контрагентами користувачами інших систем	Практична відсутність конкурентів

Далі йде розробка трирівневої маркетингової моделі товару: уточнення ідеї продукту, його складових, особливостей його надання (таблиця 5.19).

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмне забезпечення для інтеграції сервісу Signy з іншими системами електронного документообігу		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Інтерфейс користувача	Нм	Тх
	Зручність використання	Нм	Тх

Кінець таблиці 5.19

II. Товар у реальному виконанні	Надійність використання	Нм	Тх
	Якість: стандарти, нормативи, параметри тестування		
	Марка: «Інтеграції Signy»		
III. Товар із підкріпленням	До продажу: презентація програмного продукту		
	Після продажу: підтримка клієнтів, розширення функціоналу		

Наступним етапом є встановлення меж ціноутворення, які слід враховувати при визначенні ціни на продукт. Це включає аналіз цінової політики конкурентів та фінансових можливостей цільової аудиторії що виконано в таблиці 5.20, таблиці 5.21 та таблиці 5.22.

Таблиця 5.21 – Формування системи збуту

№ з/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Клієнт отримує доступ до інтеграції з конкретною системою	Надання доступу до інтеграції, підтримка користувачів	Виробник безпосередньо продає товар клієнту	Через офіційний сайт сервісу Signy

Таблиця 5.22 – Концепція маркетингових комунікацій

№ з/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Прямий маркетинг	Інтернет, соціальні мережі	Зручність, надійність	Виявлення потенційних клієнтів	Пропозиція

Таблиця 5.23 – Визначення меж встановлення ціни

№ з/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів ціль групи споживачів	Верхня та нижня встановлення ціни на товар/послугу
1.	-	-	Високий	1500-2000 грн.

5.6 Висновки до розділу 5

Після проведення маркетингового аналізу стартап-проекту були виявлені загальні напрями використання потенційного програмного продукту та його унікальні характеристики порівняно з конкурентами. Отримано такі результати щодо показників стартап-проекту:

– реалізація та імплементація продукту є обґрунтованою та залишає можливості для подальшого вдосконалення;

– конкуренція на ринку є низькою, в порівнянні з попитом, що вказує на високі перспективи успішного впровадження продукту.

ВИСНОВКИ

В даній роботі було створено програмне забезпечення для інтеграції сервісу Signy з іншими системами електронного документообігу. При виконанні даної роботи була визначена необхідність створення інтеграцій між різними системами електронного документообігу та поставлено задачу розробки. Були досліджені різні типи та методи інтеграцій комп’ютерних систем, їх особливості, переваги і недоліки. Проведено аналіз існуючих систем електронного документообіг, особливостей їх роботи і способів інтеграції з ними.

Для програмної реалізації було обрано мову програмування C# та SQL для реалізації бек енд частини та Vue.js для створення зручного користувацького інтерфейсу. Для роботи з API інших систем було використано протокол HTTP та бібліотеку Newtonsoft.Json для обміну даними у необхідному форматі. Було спроектовано архітектуру бази даних та програмного інтерфейсу для роботи з інтеграціями.

Розроблена система є веб-застосунком, і не потребує ручного встановлення користувачем. Для запуску системи та роботи з нею користувачеві необхідно мати веб-браузер,та акаунт у сервісі Signy.

Розроблений продукт відповідає поставленим вимогам, є гнучким до розширення функціоналу, а також може стати основою для інтеграції з різними системами електронного документообігу для користувачів сервісу Signy.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сервіс електронного документообігу Signy: веб-сайт. URL: https://signy.online pdm (дата звернення: 07.06.2023).

2. Закон України “Про електронні документи та електронний документообіг”: веб-сайт. URL: https://zakon.rada.gov.ua/laws/show/851-15#Text (дата звернення: 07.06.2023).

3. Основні принципи та переваги впровадження електронного документообігу: веб-сайт. URL: https://sites.google.com/site/elektrdokumentoobig/osnovni-principi-ta-perevagi-vprovadzenna-elektronnogo-dokumentoobigu (дата звернення: 08.06.2023).

4. Сервіс електронного документообігу Vchasno: веб-сайт. URL: https://vchasno.ua (дата звернення: 08.06.2023).

5. Сервіс електронного документообігу Document Online: веб-сайт. URL: https://document.online/?lang=ua (дата звернення: 08.06.2023).

6. Сервіс електронного документообігу Paperless: веб-сайт. URL: https://paperless.com.ua/uk/ (дата звернення: 08.06.2023).

7. Системна інтеграція: веб-сайт. URL: https://en.wikipedia.org/wiki/System_integration (дата звернення: 10.06.2023).

8. Основні поняття комп'ютерно-інтегрованих технологій: веб-сайт. URL: https://www.uzhnu.edu.ua/uk/infocentre/get/63700 (дата звернення: 10.06.2023).

9. The limitations of JSON: веб-сайт. URL: https://web.archive.org/web/20071012014221/http://blogs.sun.com/bblfish/entry/the_limitations_of_json (дата звернення: 07.10.2023).

10. Середовище розробки Microsoft Visual Studio: веб-сайт. URL: https://informatics.in.ua/programming_csharp/part_01.php (дата звернення: 10.06.2023).

11. Visual Studio Installer: веб-сайт. URL: https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/2kt85ked(v%3dvs.100) (дата звернення: 10.06.2023).

12. Документація HTML: веб-сайт. URL: https://developer.mozilla.org/ru/docs/Web/HTML (дата звернення: 10.06.2023).

13. Документація CSS: веб-сайт. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics (дата звернення: 10.06.2023).

14. Brown T. B. CSS Master. SitePoint, 2018. – 354 p.

15. Документація Vue js: веб-сайт. URL :https://ru.vuejs.org/index.html (дата звернення: 10.06.2023).

16. Mayer G. E. T., Awesomeness J. JavaScript: JavaScript Awesomeness Book. CreateSpace Independent Publishing Platform/ 2017 – 68 p.

17. Microsoft Visual C# Step by Step (9th Edition) (Developer Reference), 2017 – 723 p.

18. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming 2nd Edition Stephen Cleary/2019, 2019 – 373 p.

19. Моделі баз даних: веб-сайт. URL: https://sites.google.com/site/raznyeurokipoinformatiki/home/bazy-dannyh/teoria-po-bazam-dannyh/varianty-hranenia-dannyh/osnovnye-modeli-baz-dannyh (дата звернення: 10.06.2023).

20. Connect to projects in Team Explorer: веб-сайт. URL: https://docs.microsoft.com/en-us/visualstudio/ide/connect-team-project?view=vs-2019 (дата звернення: 10.06.2023).

21. Розробка програмного забезпечення: веб-сайт. URL: https://uk.theastrologypage.com/software-development (дата звернення: 11.06.2023).

22. Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / КПІ ім. Ігоря Сікорського ; уклад. О. А. Гавриш, С. О. Солнцев,В. В. Дергачова, О. В. Зозульов [та ін.]: веб-сайт. URL: https://ela.kpi.ua/handle/123456789/35763 (дата звернення: 10.08.2023).

23. Узун, А. С. Резервне копіювання в інформаційних системах зовнішнього документообігу на прикладі сервісу Signy : дипломна робота ... бакалавра : 122 Комп’ютерні науки / Узун Артем Сергійович. – Київ, 2022. – 73 с. URL: https://ela.kpi.ua/handle/123456789/50490

24. Герасимчук, Д. М. Засоби оптимізації документообігу на підприємстві : магістерська дис. : 123 Комп’ютерна інженерія / Герасимчук Данило Михайлович. – Київ, 2021. – 98 с. URL: https://ela.kpi.ua/bitstream/123456789/45924/3/Gerasymchuk_magistr.pdf

25. Martin R. Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert Martin. – New York: Prentice Hall, 2000. – 468 p

26. Scott Chacon, Ben Straub, Pro Git: Apress; 2nd ed. edition 2014. 440 p.

27. Kolawa A., Huizinga, D. Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press, 2007 – 75 p.

28. C# in Depth, Fourth Edition , 2008 – 1178 p.

29. Freeman E. Head First Design Patterns: A Brain-Friendly Guide / E. Freeman, B. Bates. – San-Francisco: Headfirst, 2002. – 586 p.

30. Hyde J. Apache Calcite: A Foundational Framework for Optimized Query Processing Over Heterogeneous Data Sources / J. Hyde, D. Lemire, E. Begoli., 2018. — 154 p.

ДОДАТОК А

Засоби інтеграції систем електронного документообігу на прикладі сервісу Signy

Апробація роботи

УКР.НТУУ“КПІ ім. Ігоря Сікорського”.ТР-2106мп_23М

Аркушів 5

2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XX Міжнародної
науково-практичної конференції
молодих вчених і студентів
(присвячена 125-річчю КПІ ім. Ігоря Сікорського та 90-річчю НН ІАТЕ (ТЕФ))
м. Київ, 25–28 квітня 2023 року

ТОМ 2

Київ- 2023

УДК 620.9(062)+621.311(062)
С91

Сучасні проблеми наукового забезпечення енергетики. У 2-х т. :Матеріали XX Міжнар. наук.-практ. конф. молод. вчених і студ. *(присвячена 125-річчю КПІ ім. Ігоря Сікорського та 90-річчю НН ІАТЕ (ТЕФ))*, м. Київ,25–28 квіт.2023 р. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2023. – Т. 2. – 269 с.

ISBN 978-966-990-025-8(Заг.)
ISBN 978-966-990-027-2(Т. 2)

Подано тези доповідей XX Міжнародної науково-практичної конференції молодих вчених і студентів «Сучасні проблеми наукового забезпечення енергетики» за напрямами: автоматизація теплоенергетичних процесів, моделювання та аналіз теплоенергетичних процесів, програмне забезпечення інформаційних систем та мережних комплексів, комп'ютерний еколого-економічний моніторинг та геометричне моделювання процесів і систем, інформаційні технології та комп'ютерне моделювання.

Друкується в авторській редакції за рішенням Вченої ради Навчально-наукового інституту атомної та теплової енергетики Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (протокол №9від 27 березня 2023 р.)

ISBN 978-966-990-025-8(Заг.)
ISBN 978-966-990-027-2(Т. 2)

СЕКЦІЯ №7

Автоматизація теплоенергетичних процесів

УДК 004.60

Master 1st year, gr. ТР-21мп Uzun A.S.
Prof., doc.phys.-math.sc. Shushura O.M.

MEANS OF INTEGRATION OF ELECTRONIC DOCUMENT MANAGEMENT SYSTEMS USING THE EXAMPLE OF THE SIGNY SERVICE.

Software for organizing electronic document management using modern technologies is a tool that enables the automation of processes related to the exchange of legally significant documents, quickly apply signatures to necessary documents, and provide participants and all interested parties with secure and reliable means of working with documents using computer capabilities.

Representatives of large and medium-sized businesses, as well as government authorities and enterprises, exchange a large number of documents, contracts, and other legal papers during their work. Automating this process will speed up the work of enterprises and make accounting more transparent.

System integration is the process of setting up 'connections' between different information systems to obtain a unified information space and simplify working with business processes that are at the intersection of the systems' operations.

There are four main integration approaches (or styles):

–**File exchange.** Historically, this is an early integration approach that is relatively simple. Its essence is as follows: one program creates a file, and another program reads this file. Programs that are integrated using this method must agree on the approach to naming files, their location, format, and deletion procedure;

–**Shared database.** In this approach, several information systems use a common logical data structure;

–**Remote procedure call.** One program provides access to its functionality through a remote procedure call;

–**Asynchronous message exchange.** This is probably the only approach among those listed that was specifically created for integrating information systems. One program sends messages to a specialized message broker for this purpose, and another program receives messages intended for it through the broker. The idea is conceptually similar to how email works. Interaction between applications is performed in an asynchronous mode, which means that the sending application does not have to wait for the message to reach the recipient, wait for it to be processed, a response to be formed, and so on.[1].

The goal of this work is to create a mechanism for integrating the Signy Service[2] with other electronic document management systems. This will allow clients from different systems to cooperate and exchange documents.

The mechanism should provide the following tasks:

– sending documents, signatures, and comments from the Signy service to clients in other systems;

– receiving documents, signatures, and comments from clients in other systems to the Signy service;

– logging errors;

– ensuring continuous operation.

To enable integration with any document management system, the main necessary functions were identified: sending documents, sending signatures, sending comments, receiving new and modified documents. All integrations with other systems will be consolidated into one interface and will implement the necessary functions in their own, individual way for each system. For all documents that are sent to or received from other systems, basic properties were identified, such as a unique identifier, document owner, recipient, creation date, type, and others.

Thus, documents for integrations of all systems are stored in a separate table in the database. This table contains additional information about the current status of the document, any errors (if any), which system the document was created for integration with, etc. Separate tables for comments and signatures were also created, which are linked to the document table.

To implement the main task, the Signy Connector application, hereinafter referred to as the Connector, was created. The Connector is installed on the Signy server and works without the user's involvement. The functions of receiving, sending, and uploading documents are automatically performed at certain intervals. All three functions work independently of each other.

When a Signy user sends a document through integration with another system, the document data is stored in the Connector table. When sending documents, such a document will be sent to the other system. The document receiving function receives and stores information about documents from other systems in the Connector tables. The document uploading function uploads the document to Signy according to the data in the table. Regular execution of these functions ensures continuous exchange of documents between Signy clients and clients of other systems.

The technologies used to create the software product were NET Framework 4.7.2, .NET Framework 4.0.3 [3], and MSSQL [4]. Integrations with other systems were implemented through their open REST API.Scheme of the Connector's workis shown on fig. 1.

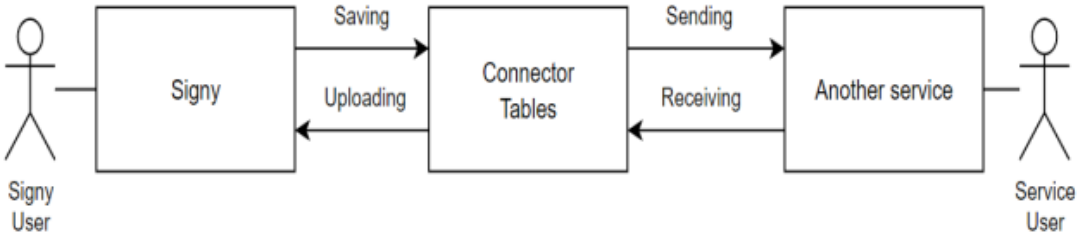

Figure 1 – Scheme of the Connector's work

The Signy client sends a document to a counterparty - a client of another service. Information about the document is stored in the Connector tables and sent to the other system with the document sending function. The counterparty signs the document and sends their document to the Signy client. Information about the actions taken is received by the document retrieval function and stored in the Connector tables. The document loading function loads new documents and signatures onto Signy, where the client has access to them.

The developed product meets all requirements, is flexible for expanding functionality, and can become the basis for more narrowly targeted products for remote work with the Signy service.

References:

1. What is integration and why is it needed? [Electronic resource] – Resource access mode: https://wearecommunity.io/communities/integration/articles/314.

2. Signy - Electronic document circulation for companies in Ukraine. [Electronic resource] – Resource access mode: https://signy.online/.

3. .NET Framework versions and dependencies [Electronic resource] – Resource access mode: https://docs.microsoft.com/uk-ua/dotnet/framework/migration-guide/versions-and-dependencies.

4. Microsoft SQL Server [Electronic resource] – Resource access mode: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server.