

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Програмний модуль «Договори» системи діловодства суб'єкту господарювання»

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

Миклашевського
Дениса Сергійовича

підпис студента

Науковий керівник
асистент кафедри інженерії
програмного забезпечення та
кібербезпеки

Хорольська Карина
Вікторівна

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Рзаєва Світлана
Леонідівна

підпис гаранта

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Миклашевському Денису Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Програмний модуль
«Договори» системи діловодства суб'єкту господарювання»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту розробити програмний модуль для документообігу

Об'єкт дослідження програмний модуль для документообігу

Предмет дослідження розробка та тестування програмного модулю для
системи документообігу

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ ТА ХАРАКТЕРИСТИК

1.1 Аналіз предметної області

1.2 Відстеження договорів та налаштування сповіщень

1.2.1 Відстеження

1.2.2 Сповіщення

1.2.3 Механізми ескалації

1.3 Управління документами

1.3.1 Завантаження та зберігання файлів

1.3.2 Контроль версій

1.3.3 Метадані та теги

1.3.4 Одночасне використання та обмін документами

1.4 Безпека програмного модуля

1.4.1 Аутентифікація користувачів

1.4.2 Контроль доступу

1.4.3 Шифрування

1.4.4 Конфіденційність даних та управління згодою

1.5 Технічне завдання

Висновок до розділу 1

РОЗДІЛ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

2.1 Вибір програмних засобів для розробки програмного модуля

2.2 Архітектура програмного забезпечення

2.3 Опис бази даних проекту

2.4 Логічна модель бази даних

2.5 Фізична модель бази даних

2.6 Макет програмного забезпечення

Висновок до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЮ «DMS»

3.1 Розробка програмного модуля

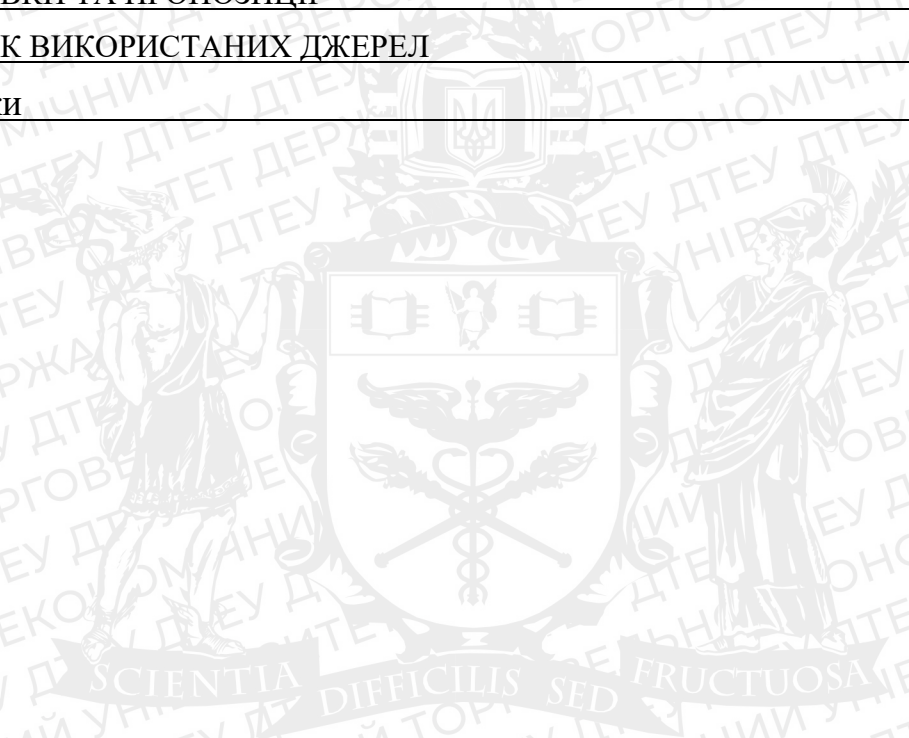
3.2 Інтерфейс програмного модуля

Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Додатки



6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Аналіз предметної області продажу вантажних шин</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Вибір програмних засобів та проєктування веб-додатку</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Реалізація веб-додатку</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>		
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>		
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>		
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту _____

Хорольська К.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент _____

Миклашевський Д.С.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту

_____ (підпис, дата)

Відмітка про попередній захист _____

_____ (ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента _____ Миклашевський Д.С
(прізвище, ініціали)

може бути допущений до захисту екзаменаційній комісії.

Гарант освітньої програми _____ Рзаєва С.Л.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« _____ » 20 _____ р.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці програмного модуля для документообігу.

У результаті порівняльного аналізу аналогічних рішень визначено, що на ринку України існує досить мало подібних програмних додатків.

Розробка серверної частини виконана за допомогою .NET Core 9.0. Готовий програмний комплекс DMS було успішно розроблено та протестовано.

Ключові слова: «DMS», програмний модуль, C#, БД, документообіг, файли, контракти.

ABSTRACT

In accordance with the purpose of the study, the work is devoted to the development of a software module for document management.

As a result of a comparative analysis of similar solutions, it was determined that there are few similar software applications on the Ukrainian market.

The development of the server side was performed using .NET Core 9.0. The finished DMS software package was successfully developed and tested.

Keywords: «DMS», program module, C#, database, document management, files, contracts.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОС – Операційна система

ПЗ – програмне забезпечення

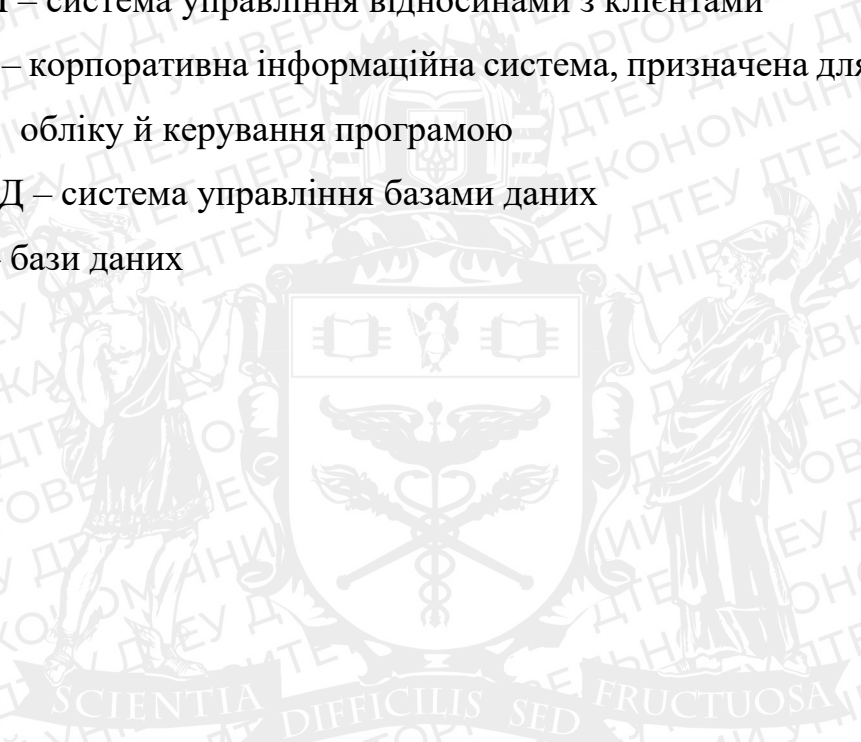
Visual Studio – Microsoft Visual Studio

CRM – система управління відносинами з клієнтами

ERP – корпоративна інформаційна система, призначена для автоматизації обліку й керування програмою

СУБД – система управління базами даних

БД – бази даних



<i>ДТЕУ 121 07-19.БР</i>							
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		14.04.23			
Керівник		Хорольська К.В.		14.04.23			
Гарант		Рзаєва С.Л.		14.04.23			
Розробив		Миклашевський Д.С.		14.04.23			
<i>Перелік умовних скорочень</i>							
Програмний модуль «Договори» системи діловодства суб'єкту господарювання					<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
					<i>ПС</i>	<i>1</i>	<i>31</i>
<i>Перелік умовних скорочень</i>					<i>Факультет інформаційних технологій</i>		
					<i>4 курс, 7 група</i>		

ЗМІСТ	
ВСТУП	5
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ ТА ХАРАКТЕРИСТИК	7
1.1. Аналіз предметної області.....	7
1.2 Відстеження договорів та налаштування сповіщень	7
1.2.1 Відстеження	8
1.2.2 Сповіщення.....	8
1.2.3 Механізми ескалації.....	8
1.3 Управління документами	8
1.3.1 Завантаження та зберігання файлів	9
1.3.2 Контроль версій	9
1.3.3 Метадані та теги	9
1.3.4 Одночасне використання та обмін документами.....	9
1.4 Безпека програмного модуля	10
1.4.1 Аутентифікація користувачів	10
1.4.2 Контроль доступу	10
1.4.3 Шифрування.....	10
1.4.4 Конфіденційність даних та управління згодою	10
1.5. Технічне завдання	11
1.6. Висновок до розділу 1	13
РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ	15
2.1. Вибір програмних засобів для розробки програмного модуля.....	15
2.2 Архітектура програмного забезпечення.....	17
2.3 Опис бази даних	18
2.4 Логічна модель бази даних	20
2.5 Фізична модель бази даних.....	21
2.6 Макет програмного забезпечення	22
Висновок до розділу 2	24
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ «DMS»	25
3.1 Розробка програмного модуля	25

					<i>ДТЕУ 121 07-19.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		23.12.22	Програмний модуль «Договори» системи діловодства суб'єкту господарювання	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Хорольська К.В		23.12.22		3	2	31
Гарант		Рзаєва С.Л.		23.12.22		<i>Факультет інформаційних технологій 4 курс, 7 група</i>		
Розробив		Миклашевський Д.С		23.12.22				

3.2 Інтерфейс програмного модуля	26
Висновок до розділу 3	30
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32
Додатки.....	32



					<i>ДТЕУ 121 07-19.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		3

ВСТУП

Електронні документи є невід'ємною частиною сучасного бізнесу та особистого життя. Контракти, фінансові документи, паспорти, сертифікати – це лише мала частина зростаючого обсягу важливих документів. Разом з цим обсягом зростає потреба у зручному та безпечному способі збереження та організації електронних документів.

Метою дипломної роботи є дослідження та аналіз різноманітних програмних засобів для зберігання документів, їх переваг та недоліків, а також розробка власного програмного забезпечення для зберігання та організації документів. В рамках дослідження буде проаналізовано такі питання, як безпека даних, швидкість та легкість використання, можливості синхронізації з іншими пристроями та доступ до документів з будь-якого місця.

Тема роботи є важливою для бізнес-лідерів, юристів, адміністраторів, фінансистів та звичайних користувачів, які мають потребу в організації та збереженні документів.

Мета дослідження: розробити програмний модуль для документообігу

Об'єкт дослідження: програмний модуль для документообігу

Предмет дослідження: розробка та тестування програмного модулю для системи документообігу.

У відповідності з метою дослідження поставлені наступні завдання:

- Розробити схему бази даних проєкту
- Розробити серверну частину проєкту

					<i>ДТЕУ 121 07-19.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		23.12.22	Програмний модуль «Договори» системи діловодства суб'єкту господарювання вантажних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Хорольська К.В.		23.12.22		В	4	31
Гарант		Рзаєва С.Л.		23.12.22		<i>Факультет інформаційних технологій 4 курс, 7 група</i>		
Розробив		Миклашевський Л.С.		23.12.22				
					<i>Вступ</i>			

- Розробити клієнтську частину прпоекту з інтуєтивно-зрозумілим інтерфейсом корсистувача
- Протестувати програмний модуль.

Методи дослідження: Провевши літературий аналіз джерел, було виявлено ряд наукових досліджень, що освітлюють проблему електронного документообігу. В роботі Дж. Сміта досліджується важливість та переваги використання модулів, які автоматизують процес зберігання та організації файлів. Автор підкреслює, що подібні програмні модулі сприяють зниженню ризиків та покращенню ефективності на підприємстві.[1]

Наступне дослідження, проведене Барбарою Ромзек та Джоселін М Джонстон, засореджено на аналізі автоматизації у системі діловодства великих корпорацій. Автори висвітлюють, як програмний модуль забезпечує стандартизацію та автоматизацію процесів документообігу, що дозволяє зменшити час та ресурси, необхідні для управління документами всередні підприємства.[2]

Практичне значення дослідження: розробка та тестування програмного модуля.

					ДТЕУ 121 07-19.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ ТА ХАРАКТЕРИСТИК

1.1. Аналіз предметної області

Предметною областю даного дипломного проекту є програмний модуль під назвою «DMS», який призначений для зберігання та організації документів. Ефективне зберігання, безпека, організація та пошук документів – все це буде розглянуто в цій роботі.

Крім того, в рамках предметної області дипломного проекту розглянемо різноманітні типи документів, такі як текстові документи, електронні таблиці, презентації, файли зображень, відео- та аудіоматеріали тощо.

Важливим аспектом є безпека документів, зберіганих у програмному модулі. В цьому контексті розглянемо заходи забезпечення безпеки даних, такі як шифрування, резервне копіювання, авторизація користувачів та інші.

Узагальнюючи, предметна область включає в себе програмний модуль, який забезпечує зберігання, організацію та пошук різних типів документів, а також заходи забезпечення безпеки.

1.2 Відстеження договорів та налаштування сповіщень

Функція своєчасних сповіщень у нашому програмному модулі відіграє важливу роль для своєчасного управління документами. Вона дозволяє централізовано переглядати всі контракти та пов'язані з ними етапи, залишаючи користувачів у курсі важливих дат та зобов'язань.

					<i>ДТЕУ 121 07-19.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		27.01.23	Програмний модуль «Договори» системи діловодства суб'єкту господарювання вантажних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Хорольська К.В.		27.01.23		<i>РІ</i>	<i>6</i>	<i>31</i>
Гарант		Рзаєва С.Л.		27.01.23	<i>Аналіз предмета дослідження і опис його основних параметрів та характеристик</i>	<i>Факультет інформаційних технологій 4 курс, 7 група</i>		
Розробив		Миклашевський Л.С.		27.01.23				

1.2.1 Відстеження

За допомогою програмного модулю «DMS» користувачі можуть відстежувати прогрес роботи з документами, переглядати останні оновлення та визначати ключові етапи в рамках документа – дата початку, час перегляду та інше. Можливість поєднати ці етапи з функцією сповіщень, робить «DMS» надійним та зручним у використанні.

1.2.2 Сповіщення

Можливість отримувати сповіщення на електронну пошту або навіть в телефон, є можливою завдяки інтеграціям з комунікаційними платформами, такими як Slack або Microsoft Teams. Ця функція дозволяє користувачам своєчасно реагувати на сповіщення та попередження, навіть якщо він знаходиться не за своїм робочим комп'ютером.

1.2.3 Механізми ескалації

Правила користування програмним модулем «DMS» є важливим аспектом його використання. Користувачі не можуть використовувати модуль для зберігання даних, які заборонені законодавчим правом.

Для запобігання порушення правил користувача, програмний модуль має функцію ескалації. Ця функція гарантує, що адміністратори будуть своєчасно повідомлені і зможуть вжити коригувальні заходи, щоб уникнути потенційних порушень.

1.3 Управління документами

Управління документами – одна з багатьох важливих функцій, що робить користування програмним модулем зручним та простим. Організація документів відбувається за різними типами файлів, дати їх завантаження в модуль, часу останнього оновлення та інше. Це гарантує, що всі необхідні файли легкодоступні та не будуть загублені в купі інших файлів.

					ДТЕУ 121 07-19.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

1.3.1 Завантаження та зберігання файлів

Основною функцією програмного модулю є завантаження та подальше зберігання файлів. Ці файли можуть включати різноманітні документи, такі як юридичні угоди, листи або будь-які інші файли, що мають відношення до документів. Програмний модуль забезпечує безпечне зберігання цих файлів, а також легкий доступ до них.

1.3.2 Контроль версій

Для ведення чіткого обліку змін в файлах, програмний модуль включає функцію контролю версій. Ця функція дозволяє відстежувати редагування документів і переглядати попередні версії цих файлів. Це гарантує, що актуальна версія буде завжди доступна для користування.

1.3.3 Метадані та теги

Для швидкого пошуку потрібних файлів, в модулі є можливість тегування. Ця функція дозволяє користувачу присвоювати файлам необхідні мітки для зручної класифікації та доступу до документів, на основі різноманітних тегів.

1.3.4 Одночасне використання та обмін документами

Робота в команді – важливий аспект швидкого виконання різноманітних задач. Щоб дати можливість командам прискорити роботу з документами, в програмний модуль була додана функція одночасного використання. Вона включає такі функції, як створення та видалення файлів, коментування, редагування в режимі реального часу – все це підвищує ефективність спільної роботи.

Крім того, програмний модуль дозволяє надавати контрольний доступ, щоб лише обраний користувач мав доступ до файлів та їх редагування.

					ДТЕУ 121 07-19.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

1.4 Безпека програмного модуля

Збереження цілісності файлів та конфіденційності інформації має велике значення в роботі програмного модулю. Гарантом безпеки даних, їх захисту від несанкціонованого доступу є такі функції як шифрування, аутентифікація користувачів та контроль доступу.

1.4.1 Аутентифікація користувачів

Механізм аутентифікації користувачів, гарантує, що тільки авторизовані акаунти можуть отримати доступ до файлів. Це комбінації імені користувача та пароля. Ця функція є основною в запобіганні несанкціонованому доступу.

1.4.2 Контроль доступу

Адміністратори мають можливість визначати ролі та дозволи для користувачів. Різні типи дозволів можуть наприклад бути призначені на основі посад в компанії. Це гарантує, що користувачі зможуть переглядати, редагувати або видаляти лише ті файли, що відповідають їхньому рівню допуску.

1.4.3 Шифрування

За захист цілісності та конфіденційності файлів в модулі використовується шифрування. Шифрування є гарантом, що навіть у разі несанкціонованого доступу до даних, вони залишаться нечитабельними та непридатними для використання.

1.4.4 Конфіденційність даних та управління згодою

Модуль відповідає законам і нормативним актам про захист даних, таким як Загальний регламент про захист даних (GDPR)[1]. Це дозволяє керувати згодою користувачів та політикою збереження даних. Це забезпечує дотримання правил конфіденційності та сприяє прозорості обробки даних.

						Аркуш
					ДТЕУ 121 07-19.БР	9
Зм.	Аркуш	№ докум	Підпис	Дата		

1.5. Технічне завдання

Для розробки програмного модуля було вирішено створити технічне завдання, що надасть чітке розуміння потреб у розробці.

Загальні відомості:

1.1. Найменування системи

1.1.1. Повне найменування системи: «Система документообігу «Договори»»

1.1.2. Скорочене найменування системи: «DMS»

1.2. Планові терміни початку та закінчення робіт: 09.09.2022-05.05.2023

2. Мета та призначення створення системи

2.1. Призначення системи: Забезпечення ефективного обігу документів, їх збереження та організації.

2.2. Мета створення системи: Автоматизація процесів створення, зберігання, обробки та контролю документів з метою підвищення продуктивності на підприємстві.

3. Вимоги до системи

3.1. Вимоги до системи в цілому

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем:

- Система повинна мати модульну структуру, що дозволяє розширювати та модифікувати функціональні можливості окремих компонентів.
- Підсистеми системи повинні взаємодіяти між собою, забезпечуючи потрібну передачу та обробку даних.

					Аркуш
					10
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 07-19.БР

- Система повинна підтримувати різні режими функціонування, включаючи режими створення, редагування, збереження та пошуку документів.

3.1.2. Показники призначення

- Система повинна ефективно виконувати процеси створення, зберігання, редагування та контролю документів.
- Система повинна забезпечувати точність, повноту та надійність обробки даних.
- Система повинна мати інтуїтивно зрозумілий та зручний інтерфейс користувача для зручного взаємодії з нею.

3.2. Вимоги до функціональності системи

3.2.1. Збереження документів

- Система повинна забезпечувати можливість збереження договорів у цифровому форматі.
- Система повинна забезпечувати можливість категоризації та класифікації договорів за певними характеристиками.

3.2.2. Редагування та контроль документів

- Система повинна дозволяти користувачам редагувати договори, вносити зміни та зберігати історію змін.
- Система повинна забезпечувати контроль версій документів, дозволяючи відстежувати зміни та відновлювати попередні версії.

3.2.3. Пошук та перегляд документів

- Система повинна мати функціонал для пошуку та фільтрації договорів за різними критеріями.
- Система повинна дозволяти переглядати детальну інформацію про файли, включаючи метадані та прикріплені файли.

						Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-19.БР	

3.2.4. Керування правами доступу

- Система повинна мати механізм керування правами доступу до документів, що забезпечує рівні доступу для різних користувачів.
- Система повинна забезпечувати можливість обмеження доступу до конфіденційної інформації в договорах.

3.2.5. Звітність

- Система повинна надавати можливість генерувати звіти та статистику про договори за різними параметрами (тип договору, сторони, терміни тощо).
- Система повинна забезпечувати можливість експорту звітів в різні формати (наприклад, PDF, Excel).

4. Вимоги до інтерфейсу користувача:

- Система повинна мати інтуїтивно зрозумілий та простий інтерфейс користувача.
- Інтерфейс повинен бути зручним для навігації та взаємодії з системою.
- Інтерфейс повинен мати можливість налаштування для відображення необхідної інформації та функціоналу.

1.6. Висновок до розділу 1

Отже, основні функції програмного модуля «DMS» можна узагальнити наступним чином:

Зберігання та організація: Модуль забезпечує безпечне зберігання файлів в електронному вигляді та дозволяє ефективно організувати їх на основі різних параметрів.

					Аркуш
					12
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 07-19.БР

Сповіщення: Модуль має функцію сповіщень, покращуючи управління та дотримання термінів у роботі з документами.

Управління файлами: Модуль включає функції для додавання супровідних документів, забезпечуючи централізацію та легкий доступ до всіх необхідних файлів.

Одночасне використання: Модуль інтегрується з іншими додатками, оптимізуючи процеси та забезпечуючи зручну співпрацю між кількома користувачами.

Безпека: Аутентифікація користувачів, контроль доступу, шифрування – все це є захисником конфіденційності інформації та забезпечує дотримання законодавчих і нормативних вимог.

Охоплюючи ці функції можна сказати що, програмний модуль «DMS» покращує загальний процес управління документами, підвищує ефективність і допомагає компаніям зменшити ризики, пов'язані з електронним документообігом.

					Аркуш
					13
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 07-19.БР

РОЗДІЛ 2

ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

2.1. Вибір програмних засобів для розробки програмного модуля

Для розробки програмного забезпечення «DMS» було вибрано мову програмування C# та середовище розробки Microsoft Visual Studio (далі Visual Studio). Цей вибір базується на наступних пунктах:

- Мова C# є об'єктно-орієнтованою мовою програмування, що дозволяє створювати програми зі складною логікою та великими обсягами коду. Крім того, мова C# має сильну підтримку Microsoft, що дозволяє отримати швидкий та якісний технічний супровід та підтримку.[3]
- Середовище розробки Visual Studio має велику кількість інструментів для створення програмного забезпечення, що дозволяє розробникам створювати ефективний та якісний код, а також прискорює процес його розробки.[4]
- Windows Forms є технологією, яка дозволяє створювати інтерфейс користувача для десктопних додатків. Це дозволяє створювати зручний та простий інтерфейс для користувачів, що забезпечує їм комфортну роботу з програмним забезпеченням.[5]
- Бази даних, зроблені через SQL, є надійними та ефективними для зберігання великої кількості даних. SQL є стандартом для роботи з базами даних, що забезпечує високу сумісність з іншими програмними засобами та інфраструктурою.[6]

Таким чином, вибір мови програмування C#, середовища розробки Microsoft Visual Studio, технології Windows Form та баз даних через SQL є

					<i>ДТЕУ 121 07-19.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Вибір програмних засобів та проектування програмного модуля</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		03.03.23		P2	14	31
Керівник		Хорольська К.В.		03.03.23		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		03.03.23				

виправданим з точки зору швидкості розробки, ефективності та надійності роботи програми.

Зокрема, мова C# має дуже чистий та логічний синтаксис, що забезпечує зручну та швидку розробку ПЗ. Вона є частиною платформи .NET, що дозволяє використовувати багато готових бібліотек та фреймворків для створення програм.[3]

Середовище розробки Visual Studio має вбудовані інструменти для роботи з C#. Це дозволяє легко працювати з ПЗ. Також воно містить велику кількість плагінів та інших розширень, що дозволяють налаштувати середовище розробки на свій смак та прискорити всі процеси.[4]

Технологія Windows Form надає розробникам можливість створювати зручний та легкий інтерфейс, який підтримується на більшості десктопних операційних систем. Це забезпечує зручність та швидкість роботи з програмою, а також підвищує комфорт користувачів.[5]

База даних на SQL дозволяє зберігати великі обсяги документів та забезпечує швидкий та ефективний доступ до них. SQL є стандартом для роботи з реляційними БД, що забезпечує сумісність та легкість обміну даними з іншими програмами та системами.[6]

Крім того, БД на SQL дозволяє зберігати дані у структурованому та організованому форматі, що дозволяє швидко знаходити та отримувати необхідні документи. Також SQL підтримує транзакції, що забезпечує надійність та цілісність даних.[6]

Обрані технології та інструменти є більш ефективними серед інших аналогів. Вони забезпечують швидку розробку, зручну роботу та надійність даних. Ці інструменти мають велику спільноту розробників та підтримуються компанією Microsoft, що забезпечує доступ до великої кількості інформації та ресурсів.

						ДТЕУ 121 07-19.БР	Аркуш
							15
Зм.	Аркуш	№ докум	Підпис	Дата			

2.2 Архітектура програмного забезпечення

Еталонна модель програмного модуля «DMS» показує, як користувач працює з програмою. На рисунку 2.1 можна побачити, що зайшовши в додаток, він має доступ до програмного інтерфейсу, який дає змогу користувачу проводити різні операції.

До інтерфейсу «DMS» входять:

Текстовий інтерфейс – він забезпечує можливість взаємодіяти з програмою за допомогою текстових команд, наприклад «створити новий файл», «відкрити документ» тощо.

Графічний інтерфейс – інтерфейс, який використовується для взаємодії користувача з графічними елементами, такими як кнопки, поля введення, меню, вікна та інші візуальні компоненти.

Логіка інтерфейсу – взаємодія з графічними елементами, як вони зв'язані між собою. Вона орієнтована на забезпечення зручності та інтуїтивно зрозумілого способу навігації для користувача.

Без логічного змісту будь-який програмний додаток не буде виконувати свою роботу, в тому числі й «DMS». Завантаження та збереження файлів, їх подальша організація та оформлення можливе за допомогою програмного коду, з якого створений наш програмний додаток.

Основні складові еталонної моделі програмного додатку «DMS» включають наступне:

1. Зберігання документів в електронному вигляді: це база даних з різноманітними таблицями на основі SQL. Дає можливість зберігати дані для їх подальшого використання.
2. Організація документів: категоризація документів на основі тем, проєктів та дат. Це допоможе користувачам знайти необхідні документи серед багатьох інших.

						ДТЕУ 121 07-19.БР	Аркуш
							16
Зм.	Аркуш	№ докум	Підпис	Дата			

3. Захист даних: забезпечення безпеки. Включає в себе необхідність створення облікових засобів для нових користувачів й потребу введення паролю й логіну перед входом в програмний модуль.

4. Пошук та вибір документів: можливість швидкого пошуку та вибору документів за необхідними критеріями.

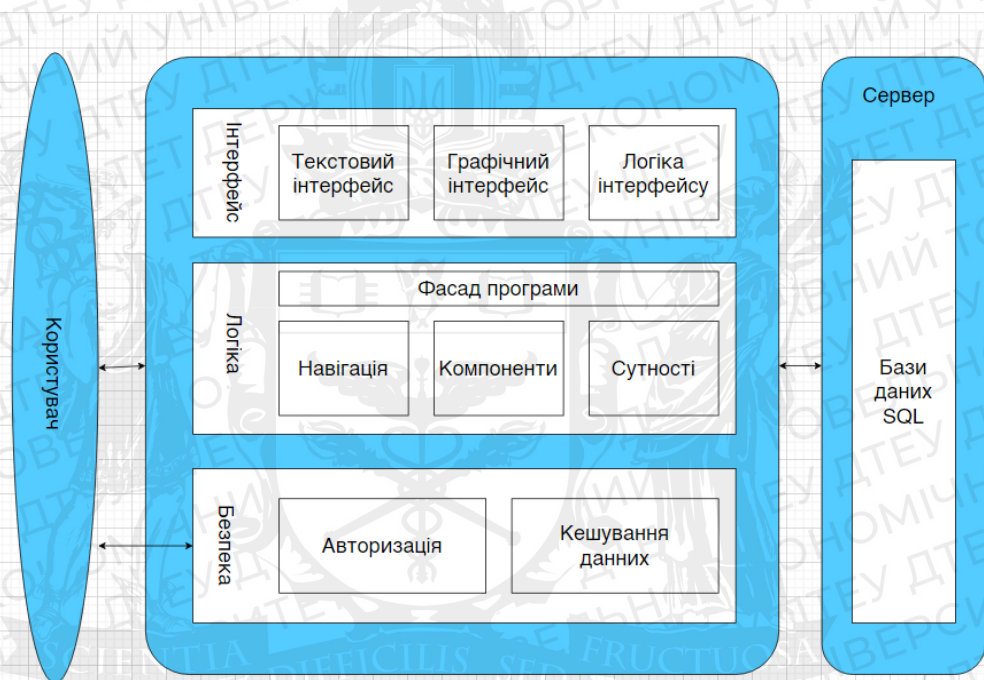


Рисунок 2.1. Еталонна модель програмного забезпечення.

Джерело: побудовано автором за допомогою веб-сайту

<https://app.diagrams.net>

2.3 Опис бази даних

База даних для програмного модуля «DMS» призначена для ефективного зберігання та управління різними аспектами системи. Вона складається з декількох таблиць, кожна з яких слугує в організації та пошуку даних. Ключовими таблицями в цій базі даних є Users, Admin, Files, LoginTime та LogoutTime.

						ДТЕУ 121 07-19.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			17

1. Таблиця Users:

Таблиця Users зберігає інформацію про користувачів. Вона включає такі поля, як UserID, Ім'я, Email та Пароль. Ця таблиця необхідна для аутентифікації користувачів, контролю доступу та виконання операцій у програмі, призначеній для конкретного користувача.

2. Таблиця Admin:

Таблиця Admin відповідає за управління користувачами з правами адміністратора. Вона містить такі поля, як AdminID, ім'я, адреса електронної пошти, пароль. Ця таблиця надає спеціальні адміністративні привілеї та полегшує контроль над функціональністю програми.

3. Таблиця Files:

Таблиця Files слугує центральним сховищем для зберігання файлів. Вона містить такі поля, як FileID, FileName, FileSize, FileType, FileDescription. Ця таблиця дозволяє ефективно зберігати, знаходити та організовувати документи в програмі.

4. Таблиця LoginTime:

Таблиця LoginTime зберігає записи про вхід користувача в систему. Вона містить такі поля, як UserID, LoginTime, LoginData. Ця таблиця дозволяє відстежувати та аналізувати шаблони входу користувачів та керувати сесіями.

5. Таблиця LogoutTime:

Таблиця LogoutTime відстежує дії користувачів щодо виходу з системи в програмі. Вона містить такі поля, як UserID, LogoutTime, LogoutData. Ця таблиця надає інформацію про сесії користувачів і допомагає в управлінні та аналізі сесій. Зв'язки між таблицями встановлюються за допомогою зовнішніх ключів, таких як UserID у таблицях LoginTime і LogoutTime, що посилаються на таблицю Users.

						ДТЕУ 121 07-19.БР	Аркуш
							18
Зм.	Аркуш	№ докум	Підпис	Дата			

2.4 Логічна модель бази даних

Логічна модель бази даних, яка розташована на рисунку 2.2, дає змогу візуально побачити, як влаштовані таблиці Users, Admin, Files, LoginTime та LogoutTime. Можна побачити, що було використано оптимальні змінні, наприклад змінна int для поля ID. Це допомагає оптимізувати велику кількість даних та зменшити час відклику між програмою та базою даних.

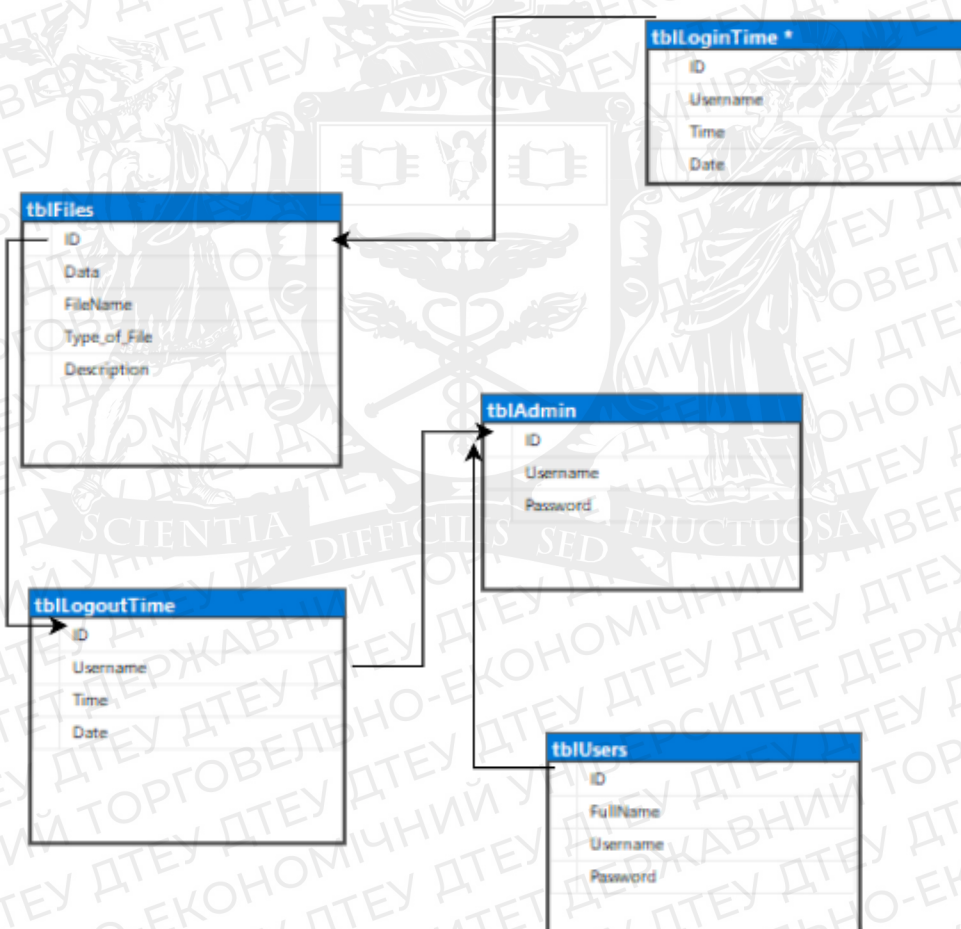


Рисунок 2.2. Логічна модель бази даних.

Джерело: побудовано автором.

2.5 Фізична модель бази даних

Фізична модель бази даних нашого програмного додатку показує нам таблиці 2.1 – 2.5, які використовуються базою даних для коректної роботи програми. Майже кожна таблиця має поле UserId, що дає можливість поєднати таблиці між собою.

Таблиця 2.1

Users				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
UserId	Primary key	Так	int	Не допустимі
FullName			varchar(50)	
Username			varchar(20)	
Password			varchar(20)	

Джерело: побудовано автором.

Таблиця 2.2

Admin				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
UserId	Primary key	Так	int	Не допустимі
Username			nvarchar(25)	
Password			nvarchar(25)	

Джерело: побудовано автором.

Таблиця 2.3

Files				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
FileId	Primary key	Так	int	Не допустимі
Data			varbinary(MAX)	
FileName			varchar(MAX)	
Type_of_File			char(14)	
Description			varchar(MAX)	

Джерело: побудовано автором.

						Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-19.БР	

Таблиця 2.4

LoginTime				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
UserId	Primary key	Так	int	Не допустимі
Username			varchar(15)	
Time			varchar(15)	
Date			date	

Джерело: побудовано автором.

Таблиця 2.5

LogoutTime				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
UserId	Primary key	Так	int	Не допустимі
Username			varchar(15)	
Time			varchar(15)	
Date			date	

Джерело: побудовано автором.

2.6 Макет програмного забезпечення

Інтерфейс нашого програмного додатку - це справжній шедевр технологічного дизайну. Він є зручним та дуже простим у використанні. Завдяки інтуїтивно зрозумілому інтерфейсу, користувач зможе швидко зрозуміти, як користуватися програмою, навіть якщо він ніколи раніше не користувався подібними продуктами.

Наша програма має чистий та сучасний дизайн, який не заважатиме роботі. Користувач зможе швидко знайти всі необхідні опції та функції завдяки зручному меню, яке розташоване у зручному місці зліва. Інтерфейс

						Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-19.БР	

максимально мінімалістичний та підійде будь-якій людині, яка цінує швидкість та ефективність.

Зовнішній вигляд «DMS» - зручний інструмент, який дозволяє ефективно та швидко виконувати роботу, переконатись в цьому можна, подивившись на рисунок 2.3.

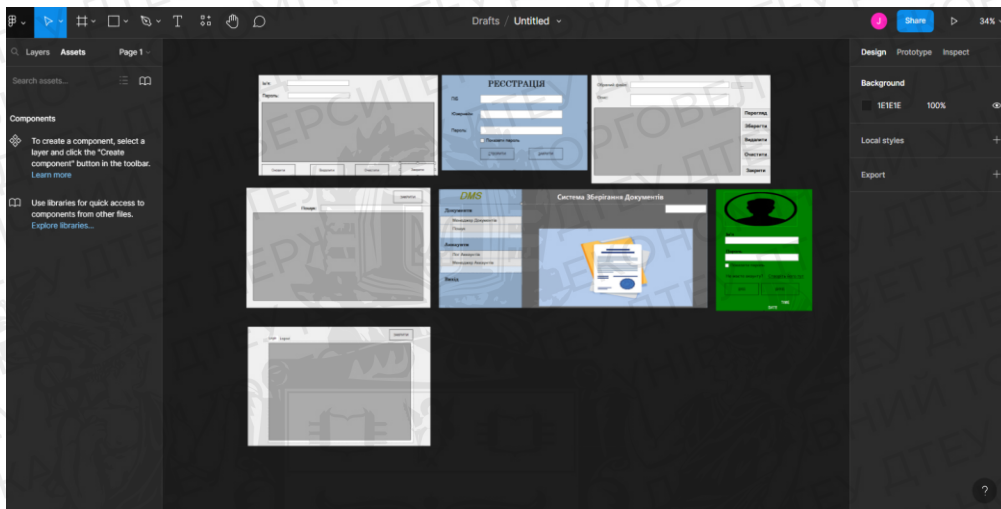


Рисунок 2.3. Фрагменти зовнішнього вигляду програмного додатку «DMS»

Джерело: побудовано автором у середовищі Figma

Дизайн програми було створено в Windows Form. Він включає наступні елементи:

- Головне вікно програми: вікно, яке містить назву програми, меню, кнопки та іншу інформацію.
- Вікно входу та вікно реєстрації: дозволяє користувачам увійти в систему, якщо вони зареєстровані, або зареєструватися, якщо вони нові користувачі.
- Головне вікно для взаємодії з документами: вікно, яке дозволяє користувачам виконувати всі наявні дії з файлами.

						ДТЕУ 121 07-19.БР	Аркуш
							22
Зм.	Аркуш	№ докум	Підпис	Дата			

- Вікно пошуку документів: дозволяє користувачам швидко та ефективно знайти потрібний документ серед багатьох інших документів, що зберігаються в системі.
- Адмін вікна програми: вікна, які доступні лише адміністратору програмного додатку, та дозволяють йому переглядати файли, користувачів та їх час сесії.

Висновок до розділу 2

На основі аналізу, проведеного в цьому розділі, можна зробити висновок, що C# є дуже гнучкою та ефективною мовою програмування, яка добре підходить для розробки програмних модулів. Середовище розробки Visual Studio - це потужна та багатофункціональна платформа, яка дозволяє швидко розробляти програмне забезпечення. Windows Forms - популярна технологія створення користувацьких інтерфейсів, що забезпечує простий та інтуїтивно зрозумілий інтерфейс для користувачів. Бази даних SQL - це галузевий стандарт для зберігання та пошуку великих обсягів даних, що забезпечує надійне та ефективне зберігання та пошук даних. Загалом, поєднання C#, Visual Studio, Windows Forms та баз даних SQL забезпечує комплексний набір інструментів для розробки програмних модулів зі складною логікою та вимогами до обробки даних.

						ДТЕУ 121 07-19.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			23

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ «DMS»

3.1 Розробка програмного модуля

Код програмного модуля "DMS" написаний мовою C#. Він включає два класи: "cc" та "DataClasses1DataContext".

Клас "cc" є класом з статичним методом "ExecSP". Цей метод використовується для виконання збережених процедур у базі даних SQL Server. Він приймає ім'я збереженої процедури і список параметрів SQL як вхідні дані, а повертає таблицю даних, що містить результати виконання збереженої процедури.

Клас "DataClasses1DataContext" пов'язаний з LINQ to SQL, технологією, яка дозволяє працювати з реляційними базами даних, використовуючи об'єктно-орієнтовані концепції. Клас успадковується від класу "DataContext", який є основною точкою входу для використання LINQ to SQL. Він включає декілька методів, які відповідають збереженим процедурам у базі даних, таким як "CreateUsers", "RemoveUser", "ViewUsers" та інші. Ці методи є функцією, яка зіставляє їх з відповідними збереженими процедурами в базі даних.

Крім того, є кілька часткових класів, таких як "ViewUsersResult", "ViewFilesResult", "SeachFilesResult", "ViewOutResult" і "ViewInResult". Ці класи представляють набори результатів, що повертаються вищезгаданими збереженими процедурами. Кожен клас містить властивості, які відображаються на стовпці у наборах результатів.

<i>ДТЕУ 121 07-19.БР</i>					
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	
Зав. каф.		Криворучко О.В.		14.04.23	
Керівник		Хорольська К.В.		14.04.23	
Гарант		Рзаєва С.Л.		14.04.23	
Розробив		Миклашевський Д.С.		14.04.23	
Програмний модуль «Договори» системи діловодства суб'єкту господарювання					
<i>Реалізація програмного модуля «DMS»</i>					
			<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
			<i>РЗ</i>	<i>24</i>	<i>31</i>
Факультет інформаційних технологій 4 курс, 7 група					

3.2 Інтерфейс програмного модуля

Коли користувач заходить в програмний модуль, він потрапляє в перше вікно (рисунок 3.1), в якому має ввести свої дані (логін, пароль) та натиснути на кнопку «Вхід», якщо він вже є зареєстрованим, або натиснути на «Створити новий обліковий запис», якщо користувач користується програмною перший раз.

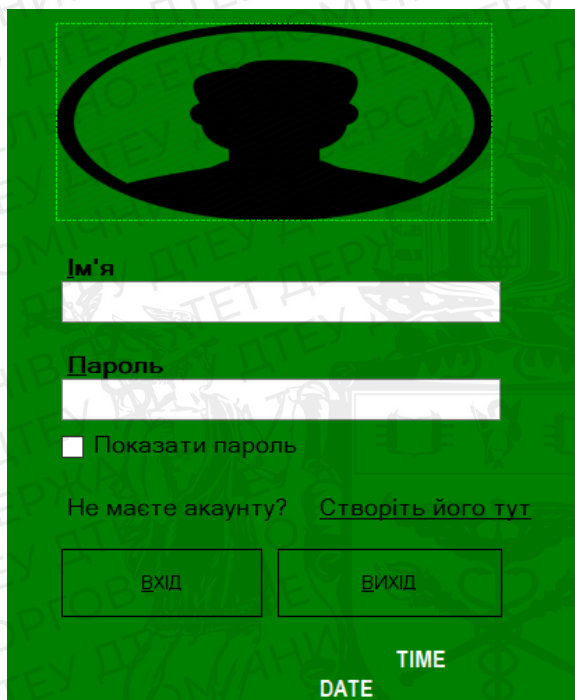


Рисунок 3.1. Вікно входу до програми.

Джерело: побудовано автором

Вікно реєстрації (Рисунок 3.2) дає змогу новому користувачеві створити свій особистий кабінет. Для цього йому доведеться ввести прізвище і ім'я по-батькові, для подальших юридичних послуг, свій юзернейм та пароль. Якщо юзернейм вже використовується в іншому обліковому записі, користувач отримає повідомлення, де йому запропонують обрати щось інше.

Закінчивши з заповненням необхідних полів, користувач має натиснути на кнопку «СТВОРИТИ» і, якщо всі дані були введені коректно, обліковий запис з'явиться в базі даних та буде придатний до подальшого використання.

						ДТЕУ 121 07-19.БР	Аркуш
							25
Зм.	Аркуш	№ докум	Підпис	Дата			

РЕЄСТРАЦІЯ

ПІБ

Юзернейм

Пароль:

Показати пароль

Рисунок 3.2. Вікно реєстрації нових користувачів.

Джерело: побудовано автором

Після входу в першому вікні, відкривається основне вікно програмного модуля (Рисунок 3.3). Тут ми можемо побачити категорію «Документи», в якій можна перейти в «Менеджер Документів» або в режим пошуку. Друга категорія «Аккаунти» недоступна для звичайних користувачів. Вона є допоміжним інструментом для адміністраторів програмного модуля та дає їм змогу управляти персональними кабінетами через кнопку «Менеджер Аккаунтів» або подивитись історію входу та виходу в програму через кнопку «Лог Аккаунтів».

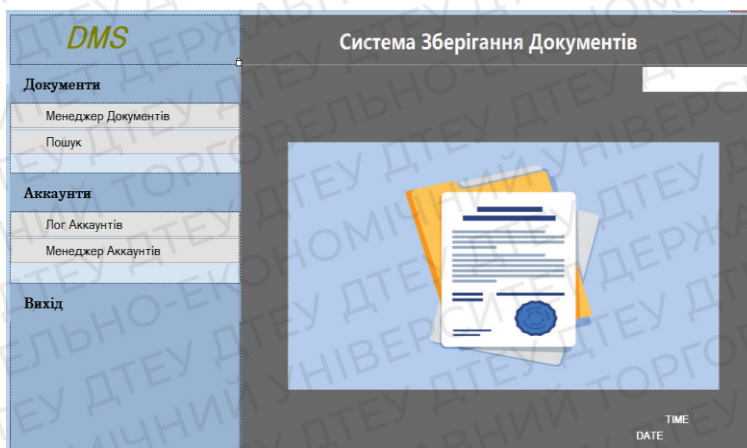


Рисунок 3.3. Головне вікно програми.

Джерело: побудовано автором

					ДТЕУ 121 07-19.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

Користувач, натиснувши на «Менеджер Документів», переходить в вікно для роботи з файлами (Рисунок 3.4). Тут він може завантажити новий файл, натиснувши на «...» та обравши потрібний документ, змінити його назву та додати опис.

Якщо потреби додати новий файл немає, користувач може переглянути інші файли, які знаходяться в програмному модулі, видалити або повністю його очистити поле з файлами.



Рисунок 3.4. Вікно для роботи з документами.

Джерело: побудовано автором

Для більш детального пошуку необхідних файлів, користувач може скористатися вікном пошуку (Рисунок 3.5), натиснувши необхідну кнопку в основному вікні програми. Тут можна вписати потрібні критерії та переглянути документи, які під них потрапили.

						Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-19.БР	

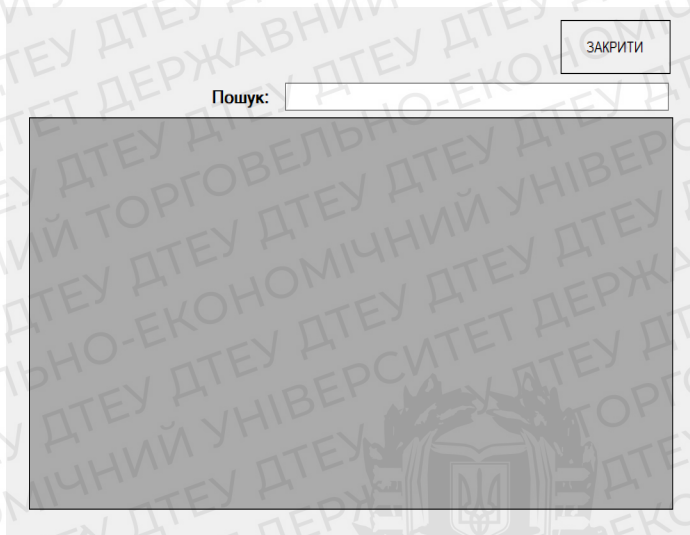


Рисунок 3.5. Вікно для пошуку документів.

Джерело: побудовано автором.

Адміністратор, зайшовши в вікно управління користувачами (Рисунок 3.6) , бачить всі облікові засоби, які були зареєстровані в програмному модулі. Для проведення потрібних маніпуляцій з ними, використовуються кнопки «Оновити» - щоб задати новий логін та пароль необхідному користувачу, «Видалити» - для повного видалення акаунту з програми та кнопка «Очистити» - для очистки вікна.

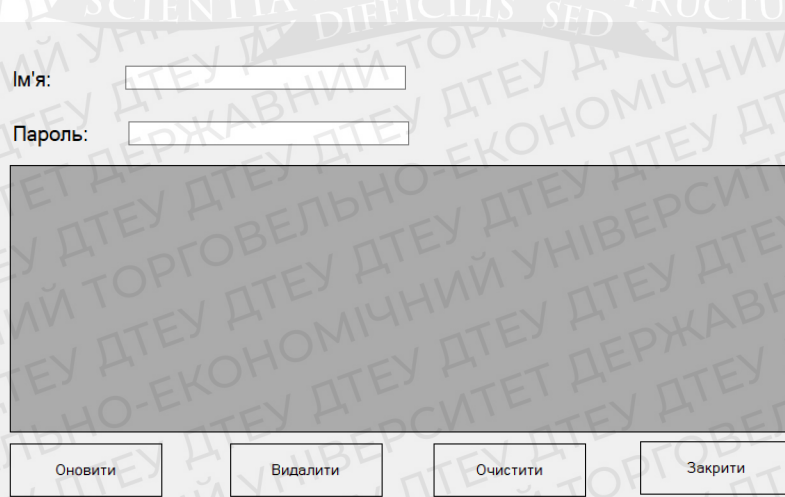


Рисунок 3.6. Адмін-вікно для управління користувачами.

Джерело: побудовано автором.

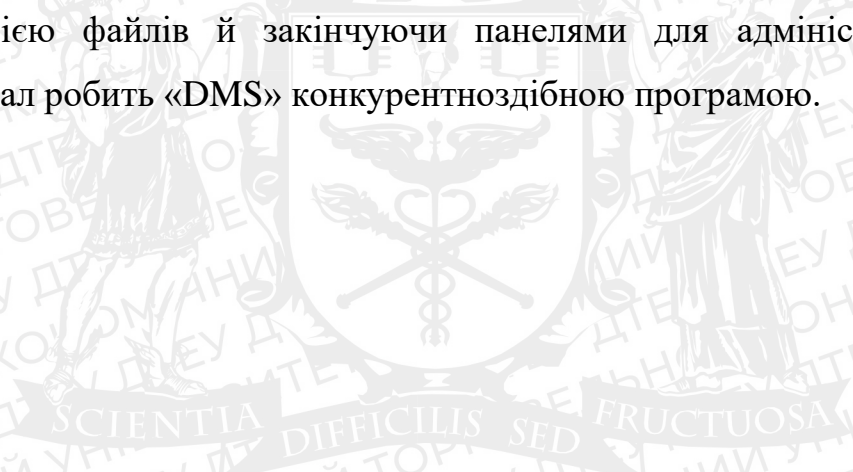
						Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-19.БР	

Висновок до розділу 3

Отже, код програмного модуля поєднує в собі ADO.NET для виконання збережених процедур і LINQ to SQL для роботи з базою даних з використанням об'єктно-орієнтованих концепцій. Це поєднання є зручним в роботі та написанні програми, а також допомагає оптимізувати код.

Інтерфейс програмного модуля є зручним у використанні. Прості кольори, інтуїтивне розміщення вікон – все це дозволяє користуватися «DMS» навіть людині, котра ніколи до цього не працювала в подібних програмах.

Програмний модуль має багато функцій, починаючи з простою організацією файлів й закінчуючи панелями для адміністраторів. Весь функціонал робить «DMS» конкурентноздібною програмою.



					Аркуш
					29
Зм.	Аркуш	№ докум	Підпис	Дата	

ДТЕУ 121 07-19.БР

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Електронний документообіг став невід’ємною частиною бізнесу, освіти та особистого життя. Зберігання та організація документів стала однією з головних проблем, з якою зіштовхуються різноманітні компанії.

Розробка програмного забезпечення для зберігання документів є важливим кроком для вирішення цієї проблеми. Такі продукти дозволяють зручно та ефективно зберігати та організовувати електронні документи.

Розвиток технологій та збільшення обсягу електронних документів створює необхідність в дослідженні та розробці нових програмних засобів, які зможуть задовільнити потреби різних груп користувачів. Велика зацікавленість та фінансування роблять цю сферу прибутковою та престижною.

Отже, розробка програмного забезпечення для зберігання електронних документів має великий потенціал для полегшення життя та роботи, а також забезпечення безпеки та конфіденційності даних.

					<i>ДТЕУ 121 07-19.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		28.04.23	Програмний модуль «Договори» системи діловодства суб'єкту господарювання вантажних шин	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Хорольська К.В.		28.04.23		<i>ВП</i>	30	31
Гарант		Рзаєва С.Л.		28.04.23	<i>Факультет інформаційних технологій 4 курс, 7 група</i>			
Розробив		Миклашевський Д.С		28.04.23				
					<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Smith, J. (2021) «Effective Contract Management Through Software Solution»
2. Barbara Romzek, Jocelyn M Johnston. (2002) «Effective Contract Implementation and Management: A Preliminary Model»
3. A tour of the C# language. URL:
<https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (Дата звернення 20.04.2023)
4. Documentation for Visual Studio Code. URL:
<https://code.visualstudio.com/docs> (Дата звернення 20.04.2023)
5. Desktop Guide (Windows Forms .NET). URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-7.0> (Дата звернення: 20.04.2023)
6. SQL Server 2022. URL: <https://www.microsoft.com/en-us/sql-server/sql-server-2022> (Дата звернення 21.04.2023)

<i>ДТЕУ 121 07-19.БР</i>					
Зм.	Аркуш	№ докум.	Підпис	Дата	
Зав. каф.		Криворучко О.В.		23.12.22	Програмний модуль «Договори» системи діловодства суб'єкту господарювання
Керівник		Хорольська К.В.		23.12.22	
Гарант		Рзаєва С.Л.		23.12.22	
Розробив		Миклашевський Д.С.		23.12.22	
					Факультет інформаційних технологій 4 курс, 7 група
<i>Список використаних джерел</i>					

Додатки

Додаток А

Програмний код cc.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Data;

namespace Document_Management_System
{
    class cc
    {
        public static DataTable ExecSP(string spName, List<SqlParameter> sqlParams =
        null)
        {
            string strConnect = "Server=DESKTOP-
            5QGHCUV\\SQLEXPRESS;Database=DMS;Trusted_Connection=True;";
            SqlConnection loginconn = new SqlConnection();
            DataTable dt = new DataTable();

            try
            {
                loginconn = new SqlConnection(strConnect);
                loginconn.Open();

                SqlCommand logincmd = new SqlCommand(spName, loginconn);
                logincmd.CommandType = CommandType.StoredProcedure;
                logincmd.Parameters.AddRange(sqlParams.ToArray());

                SqlCommand command = loginconn.CreateCommand();
                SqlDataReader dr = logincmd.ExecuteReader();

                dt.Load(dr);
            }
            catch (Exception ex)
            {
                throw ex;
            }
            finally
            {
                loginconn.Close();
            }
            return dt;
        }
    }
}
```

Програмний код **DataClasses1.designer.cs**

```

using System.Data.Linq;
using System.Data.Linq.Mapping;
using System.Data;
using System.Collections.Generic;
using System.Reflection;
using System.Linq;
using System.Linq.Expressions;
using System.ComponentModel;
using System;

[global::System.Data.Linq.Mapping.DatabaseAttribute(Name="DMS")]
public partial class DataClasses1DataContext : System.Data.Linq.DataContext
{
    private static System.Data.Linq.Mapping.MappingSource mappingSource =
new AttributeMappingSource();

    #region Extensibility Method Definitions
    partial void OnCreated();
    #endregion

    public DataClasses1DataContext() :
        base("Server=DESKTOP-
5QGHCUV\\SQLEXPRESS;Database=DMS;Trusted_Connection=True;", mappingSource)
    {
        OnCreated();
    }

    public DataClasses1DataContext(string connection) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public DataClasses1DataContext(System.Data.IDbConnection connection) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public DataClasses1DataContext(string connection,
System.Data.Linq.Mapping.MappingSource mappingSource) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public DataClasses1DataContext(System.Data.IDbConnection connection,
System.Data.Linq.Mapping.MappingSource mappingSource) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    [global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.CreateUsers")]
    public int
CreateUsers([global::System.Data.Linq.Mapping.ParameterAttribute(Name="FullName",
DbType="VarChar(50)")] string fullName,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Username",

```



```

DbType="VarChar(20)"] string username,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Password",
DbType="VarChar(20)")] string password)
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), fullName, username, password);
        return ((int)(result.ReturnValue));
    }

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.RemoveUser")]
public int
RemoveUser([global::System.Data.Linq.Mapping.ParameterAttribute(Name="ID",
DbType="Int")] System.Nullable<int> id)
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), id);
        return ((int)(result.ReturnValue));
    }

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.ViewUsers")]
public ISingleResult<ViewUsersResult> ViewUsers()
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)));
        return ((ISingleResult<ViewUsersResult>)(result.ReturnValue));
    }

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.LoginTime")]
public int
LoginTime([global::System.Data.Linq.Mapping.ParameterAttribute(Name="Username",
DbType="VarChar(15)")] string username,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Time",
DbType="VarChar(10)")] string time,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Date", DbType="Date")]
System.Nullable<System.DateTime> date)
    {
        // IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), username, time, date);
        // return ((int)(result.ReturnValue));
        return 1;
    }

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.ViewFiles")]
public ISingleResult<ViewFilesResult> ViewFiles()
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)));
        return ((ISingleResult<ViewFilesResult>)(result.ReturnValue));
    }

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.SeachFiles")]
public ISingleResult<SeachFilesResult>
SeachFiles([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="VarChar(MAX)
")] string key)
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), key);
        return ((ISingleResult<SeachFilesResult>)(result.ReturnValue));
    }

```

```

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.ViewOut")]
public ISingleResult<ViewOutResult> ViewOut()
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)));
    return ((ISingleResult<ViewOutResult>)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.ViewIn")]
public ISingleResult<ViewInResult> ViewIn()
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)));
    return ((ISingleResult<ViewInResult>)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.LogoutTime")]
public int
LogoutTime([global::System.Data.Linq.Mapping.ParameterAttribute(Name="Username",
DbType="VarChar(15)")] string username,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Time",
DbType="VarChar(10)")] string time,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Date", DbType="Date")]
System.Nullable<System.DateTime> date)
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), username, time, date);
    return ((int)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.UpdateUsers")]
public int
UpdateUsers([global::System.Data.Linq.Mapping.ParameterAttribute(Name="ID",
DbType="Int")] System.Nullable<int> id,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Username",
DbType="VarChar(20)")] string username,
[global::System.Data.Linq.Mapping.ParameterAttribute(Name="Password",
DbType="VarChar(20)")] string password)
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), id, username, password);
    return ((int)(result.ReturnValue));
}

public partial class ViewUsersResult
{
    private int _ID;
    private string _FullName;
    private string _Username;
    private string _Password;
    public ViewUsersResult()
    {
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID",
DbType="Int NOT NULL")]

```

```

public int ID
{
    get
    {
        return this._ID;
    }
    set
    {
        if ((this._ID != value))
        {
            this._ID = value;
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FullName",
DbType="VarChar(50)")]
public string FullName
{
    get
    {
        return this._FullName;
    }
    set
    {
        if ((this._FullName != value))
        {
            this._FullName = value;
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Username",
DbType="VarChar(20)")]
public string Username
{
    get
    {
        return this._Username;
    }
    set
    {
        if ((this._Username != value))
        {
            this._Username = value;
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Password",
DbType="VarChar(20)")]
public string Password
{
    get
    {
        return this._Password;
    }
    set
    {
        if ((this._Password != value))
        {
            this._Password = value;
        }
    }
}

```

```

    }

    public partial class ViewFilesResult
    {
        private int _ID;
        private string _FileName;
        private string _Type_of_File;
        private string _Description;

        public ViewFilesResult()
        {
        }

        [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID",
        DbType="Int NOT NULL")]
        public int ID
        {
            get
            {
                return this._ID;
            }
            set
            {
                if ((this._ID != value))
                {
                    this._ID = value;
                }
            }
        }

        [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FileName",
        DbType="VarChar(MAX)")]
        public string FileName
        {
            get
            {
                return this._FileName;
            }
            set
            {
                if ((this._FileName != value))
                {
                    this._FileName = value;
                }
            }
        }

        [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Type_of_File",
        DbType="Char(14)")]
        public string Type_of_File
        {
            get
            {
                return this._Type_of_File;
            }
            set
            {
                if ((this._Type_of_File != value))
                {
                    this._Type_of_File = value;
                }
            }
        }
    }

```

```

    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Description",
DbType="VarChar(MAX)")]
public string Description
{
    get
    {
        return this._Description;
    }
    set
    {
        if ((this._Description != value))
        {
            this._Description = value;
        }
    }
}

public partial class SeachFilesResult
{
    private int _ID;
    private string _FileName;
    private string _Description;

    public SeachFilesResult()
    {
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID",
DbType="Int NOT NULL")]
    public int ID
    {
        get
        {
            return this._ID;
        }
        set
        {
            if ((this._ID != value))
            {
                this._ID = value;
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FileName",
DbType="VarChar(MAX)")]
    public string FileName
    {
        get
        {
            return this._FileName;
        }
        set
        {
            if ((this._FileName != value))
            {

```

```

        this._FileName = value;
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Description",
DbType="VarChar(MAX)")]
public string Description
{
    get
    {
        return this._Description;
    }
    set
    {
        if ((this._Description != value))
        {
            this._Description = value;
        }
    }
}

public partial class ViewOutResult
{
    private int _ID;
    private string _Username;
    private string _Time;
    private System.Nullable<System.DateTime> _Date;

    public ViewOutResult()
    {
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID",
DbType="Int NOT NULL")]
    public int ID
    {
        get
        {
            return this._ID;
        }
        set
        {
            if ((this._ID != value))
            {
                this._ID = value;
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Username",
DbType="VarChar(15)")]
    public string Username
    {
        get
        {
            return this._Username;
        }
        set
    }
}

```

```

        {
            if ((this._Username != value))
            {
                this._Username = value;
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Time",
    DbType="VarChar(15)")]
    public string Time
    {
        get
        {
            return this._Time;
        }
        set
        {
            if ((this._Time != value))
            {
                this._Time = value;
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Date",
    DbType="Date")]
    public System.Nullable<System.DateTime> Date
    {
        get
        {
            return this._Date;
        }
        set
        {
            if ((this._Date != value))
            {
                this._Date = value;
            }
        }
    }
}

public partial class ViewInResult
{
    private int _ID;
    private string _Username;
    private string _Time;
    private System.Nullable<System.DateTime> _Date;

    public ViewInResult()
    {
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID",
    DbType="Int NOT NULL")]
    public int ID
    {
        get
        {
            return this._ID;
        }
    }
}

```

```

    }
    set
    {
        if ((this._ID != value))
        {
            this._ID = value;
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Username",
DbType="VarChar(15)")]
public string Username
{
    get
    {
        return this._Username;
    }
    set
    {
        if ((this._Username != value))
        {
            this._Username = value;
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Time",
DbType="VarChar(15)")]
public string Time
{
    get
    {
        return this._Time;
    }
    set
    {
        if ((this._Time != value))
        {
            this._Time = value;
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_Date",
DbType="Date")]
public System.Nullable<System.DateTime> Date
{
    get
    {
        return this._Date;
    }
    set
    {
        if ((this._Date != value))
        {
            this._Date = value;
        }
    }
}
}
}

```