

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ЦИКЛОВА КОМІСІЯ
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи освітнього ступеня «фаховий молодший бакалавр»
за спеціальністю 122 «Комп'ютерні науки»

на тему:

**«Програма для ведення сімейного архіву фотографій
та відео»**

Виконав здобувач освіти групи П-42
Голяченко Роман Сергійович

Керівник роботи:
Устименко Людмила Миколаївна

Рецензент:
Устименко Ярослав Іванович

Зміст

Вступ	6
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ	8
1.1. Постановка основної задачі розробки	8
1.2. Огляд та порівняння аналогічних систем.	9
1.3. Вибір та обґрунтування технологій розробки	15
Висновки до розділу 1	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА	19
2.1. Вибір алгоритмів та їх ефективність	19
2.2. Спроектовані класи програми	22
2.3. Програмна реалізація	25
Висновки до розділу 2	31
РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	32
3.1. Мінімальні вимоги до системи	32
3.2. Інтерфейс користувача	33
3.3. Інструкція для роботи з програмою	37
Висновки до розділу 3	39
ВИСНОВКИ	41
Перелік літературних джерел	42
Додаток А.	45

					ДР.122.042.024.ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Голяченко Р. С.			Програма для ведення сімейного архіву фотографій та відео	Літ.	Арк.
Перевірив		Устименко Л.М.					3
Рецензент		Устименко Я.І.				Аркушів	55
Н. Контр.						ЖАТФК	
Затвердив.		Лавріщев О.О.					

РЕФЕРАТ

Записка: 55 стор., 16 рис., 1 таблиця, 1 додаток, 20 джерел.

Ключові слова: сімейний архів, Windows Forms, .NET 8, Vlc.DotNet, медіафайли, EXIF, асинхронність, інтерфейс користувача.

Дипломна робота присвячена розробці програми для організації та управління сімейними архівами фотографій і відео з використанням Windows Forms на платформі .NET 8. Метою роботи є створення зручного локального додатка для групування медіафайлів за датами, їх перегляду, відтворення та базового редагування. У роботі проаналізовано аналоги (Google Photos, Adobe Lightroom, VLC), обґрунтовано вибір технологій (.NET 8, Windows Forms, Vlc.DotNet) через їхню продуктивність і простоту.

Розроблено алгоритми асинхронного завантаження файлів із складністю $O(n)$, інтеграції EXIF-даних і відтворення відео з керуванням (Play, Pause, Stop). Програма включає два класи: MainForm для основного функціоналу та AboutForm для інформації. Інтерфейс із TreeView, PictureBox і панеллю керування відео забезпечує інтуїтивність. Визначено мінімальні системні вимоги: Windows 10, 2 ГБ ОЗП, VLC Player. Інструкція описує роботу з архівом.

Результати підтверджують працездатність програми, її переваги над аналогами в локальності та простоті. Робота має практичну цінність для збереження сімейних спогадів, відкриваючи можливості для подальшого вдосконалення.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

.NET — платформа розробки від Microsoft, яка включає набір бібліотек і інструментів для створення програмного забезпечення.

API — Application Programming Interface (інтерфейс програмування додатків), набір правил і засобів для взаємодії між програмними компонентами.

C# — об'єктно-орієнтована мова програмування, розроблена Microsoft для платформи .NET.

CLR — Common Language Runtime (загальномовне середовище виконання), компонент .NET для управління виконанням програм.

EXIF — Exchangeable Image File Format (формат обміну даними зображень), стандарт метаданих для зберігання інформації про зображення, зокрема дату створення.

GUI — Graphical User Interface (графічний інтерфейс користувача), спосіб взаємодії користувача з програмою через візуальні елементи.

JIT — Just-In-Time (компіляція "точно вчасно"), технологія в .NET для компіляції коду під час виконання.

LINQ — Language Integrated Query (інтегровані запити в мові), технологія в .NET для роботи з колекціями даних.

NuGet — менеджер пакетів для .NET, що використовується для підключення бібліотек, таких як Vlc.DotNet.

UI — User Interface (інтерфейс користувача), сукупність елементів для взаємодії з програмою.

VLC — VideoLAN Client, медіаплеєр із відкритим кодом, основа для бібліотеки Vlc.DotNet.

Vlc.DotNet — бібліотека .NET, що забезпечує інтеграцію VLC Media Player у додатки Windows Forms.

Windows Forms — технологія в .NET для створення настільних додатків із графічним інтерфейсом.

					ДР.122.042.024.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному світі цифрові технології відіграють ключову роль у збереженні та управлінні особистими даними, зокрема сімейними архівами фотографій і відео. З кожним роком обсяги цифрового контенту, який створюють люди, зростають, що породжує потребу в зручних і функціональних інструментах для його організації, зберігання та швидкого доступу. Сімейні архіви, які включають світлини та відеозаписи важливих моментів життя, є цінним ресурсом, що потребує систематизації для зручного використання як самими власниками, так і наступними поколіннями.

Метою даної дипломної роботи є розробка програми для ведення сімейного архіву фотографій та відео, яка забезпечить користувачам інтуїтивно зрозумілий інтерфейс, можливість групування медіафайлів за датами створення, а також функції додавання, видалення та відтворення вмісту. Програма створюється з використанням технології Windows Forms на платформі .NET 8, що дозволяє забезпечити високу продуктивність, сумісність із сучасними операційними системами Windows та легкість у розгортанні.

Актуальність теми зумовлена зростаючою потребою в персоналізованих програмних рішеннях для управління цифровими архівами. Більшість стандартних засобів операційних систем або комерційних програм не завжди відповідають специфічним потребам користувачів, таких як автоматичне сортування за EXIF-даними чи інтеграція зручного відтворення відео. Розробка спеціалізованого програмного забезпечення дозволить усунути ці недоліки, пропонуючи користувачам простий і ефективний інструмент для роботи з їхніми медіаархівами.

У процесі виконання роботи були поставлені такі завдання:

- проаналізувати вимоги до програми для ведення сімейного архіву;
- розробити інтерфейс користувача з використанням Windows Forms;
- реалізувати функціонал завантаження, групування та відображення медіафайлів;

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

- забезпечити підтримку відтворення відео з базовими елементами керування;
- інтегрувати механізм збереження налаштувань користувача.

Розроблена програма призначена для широкого кола користувачів, які прагнуть організувати свої сімейні архіви в зручний спосіб, зберігши при цьому простоту використання та функціональність. Використання .NET 8 як основи розробки забезпечує сучасний підхід до програмування, включаючи підтримку асинхронних операцій, що покращує продуктивність під час роботи з великими обсягами даних.

Даний вступ окреслює основні аспекти роботи, її мету, актуальність і завдання, які будуть детально розглянуті в наступних розділах дипломної роботи.

					ДР.122.042.024.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ

1.1. Постановка основної задачі розробки

Розробка програми для ведення сімейного архіву фотографій та відео є відповіддю на сучасні потреби користувачів у систематизації та управлінні їхніми цифровими медіаресурсами. Основна задача цього проєкту полягає у створенні зручного, функціонального та простого у використанні програмного забезпечення, яке дозволить користувачам ефективно організувати сімейні архіви, забезпечуючи швидкий доступ до фотографій і відео, а також базові можливості їх редагування та відтворення.

Основна мета розробки полягає в реалізації програми, яка:

- забезпечує автоматичне групування медіафайлів за датами створення (з використанням EXIF-даних для фотографій або часу створення файлу);
- підтримує додавання нових файлів до архіву та видалення застарілих чи непотрібних;
- надає можливість перегляду зображень і відтворення відеофайлів із базовими елементами керування (відтворення, пауза, зупинка, регулювання гучності та позиції);
- дозволяє користувачу обирати папку для зберігання архіву з автоматичним збереженням цього вибору для подальших сеансів роботи;
- відображає прогрес завантаження файлів для зручності роботи з великими обсягами даних.

Для досягнення поставленої мети необхідно вирішити такі ключові підзадачі:

- Визначити, які формати файлів (наприклад, .jpg, .png, .mp4, .avi) будуть підтримуватися, які функції керування медіа необхідні, та як забезпечити інтуїтивність інтерфейсу.
- Обґрунтувати використання Windows Forms і .NET 8 як основи для створення програми, враховуючи їхні переваги в контексті поставлених задач.

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

- Створити архітектуру, яка включає асинхронну обробку файлів, інтеграцію з медіаплеєром (наприклад, VLC), і механізми збереження налаштувань.
- Забезпечити зручне розміщення елементів управління, таких як дерево файлів, область перегляду, кнопки керування та прогрес-бар.
- Перевірити коректність роботи програми з різними обсягами даних і типами файлів, а також оптимізувати її продуктивність.

Основна задача розробки зосереджена на створенні програми, яка буде доступною для користувачів із мінімальним досвідом роботи з комп'ютером, але водночас надаватиме достатній функціонал для ефективного управління сімейним архівом. Важливим аспектом є забезпечення стабільності програми при роботі з великою кількістю файлів, а також її адаптивності до різних конфігурацій операційної системи Windows.

Використання технології Windows Forms у поєднанні з платформою .NET 8 дозволяє реалізувати поставлені задачі завдяки підтримці сучасних засобів розробки, таких як асинхронне програмування, вбудовані бібліотеки для роботи з файлами та зображеннями, а також широкі можливості інтеграції зовнішніх компонентів (наприклад, Vlc.DotNet для відтворення відео).

1.2. Огляд та порівняння аналогічних систем.

Розробка програми для ведення сімейного архіву фотографій та відео потребує аналізу існуючих рішень, які виконують подібні функції. На ринку існує низка програмних продуктів і вбудованих засобів операційних систем, призначених для організації та управління медіаконтентом. У цьому підрозділі розглядаються основні аналоги, їхні переваги, недоліки та порівняння з розроблюваною системою.

					ДР.122.042.024.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.1. Вбудовані засоби Windows (Провідник і "Фотографії")

Операційна система Windows пропонує базові інструменти для роботи з медіафайлами, такі як Провідник і додаток "Фотографії".

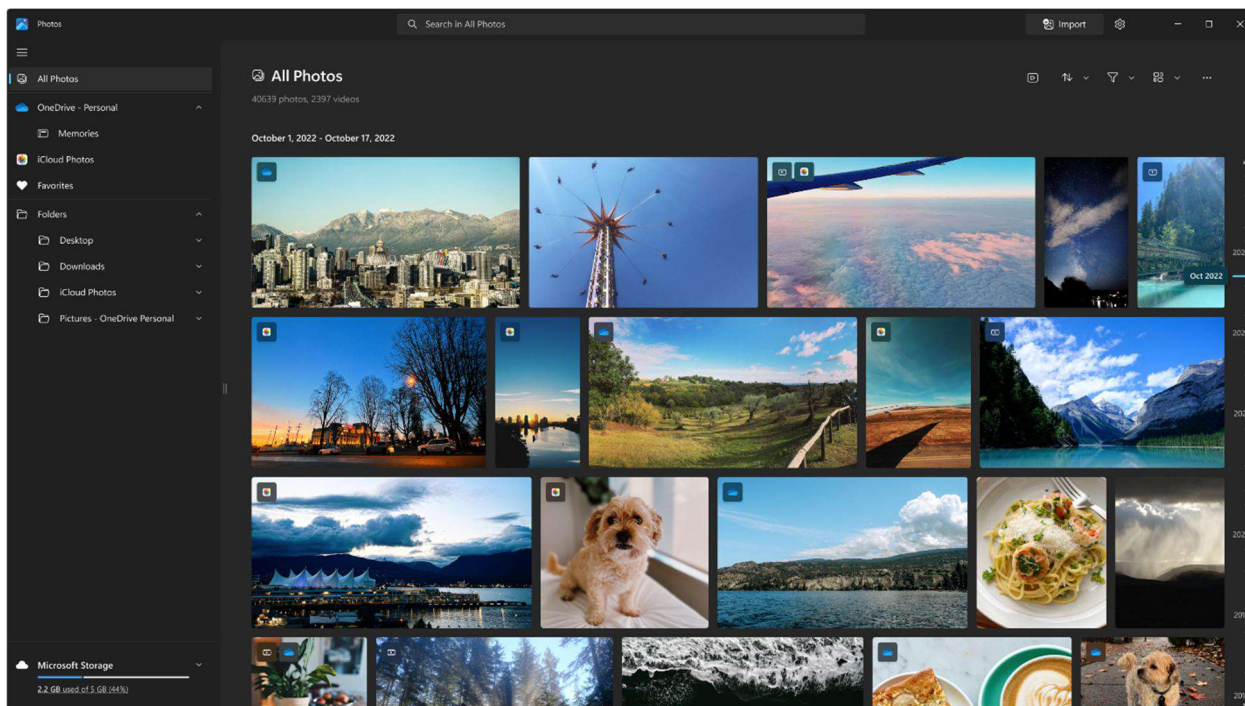


Рисунок 1.2.1 – Додаток "Фотографії" Windows.

Провідник дозволяє переглядати файли у папках, сортувати їх за датою створення чи зміни, перейменовувати та видаляти.

Вбудований у систему, не потребує встановлення, простий у використанні для базових операцій.

Відсутність автоматичного групування за EXIF-даними, обмежені можливості перегляду (без масштабування чи відтворення відео з розширеним керуванням), немає інтегрованого медіаплеєра для зручного відтворення.

Додаток "Фотографії" дозволяє перегляд зображень, базове редагування (обрізка, фільтри), відтворення відео через вбудований плеєр.

Інтуїтивний інтерфейс, підтримка популярних форматів, автоматичне групування за датою (за умови наявності метаданих).

Обмежений функціонал для відео (немає регулювання позиції чи гучності), відсутність дерева папок для навігації, неможливість додавання файлів у конкретний архів без ручного копіювання.

					ДР.122.042.024.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.2. Google Photos

Хмарний сервіс для зберігання, організації та перегляду фотографій і відео з автоматичним групуванням за датами, розпізнаванням облич і пошуком за ключовими словами.

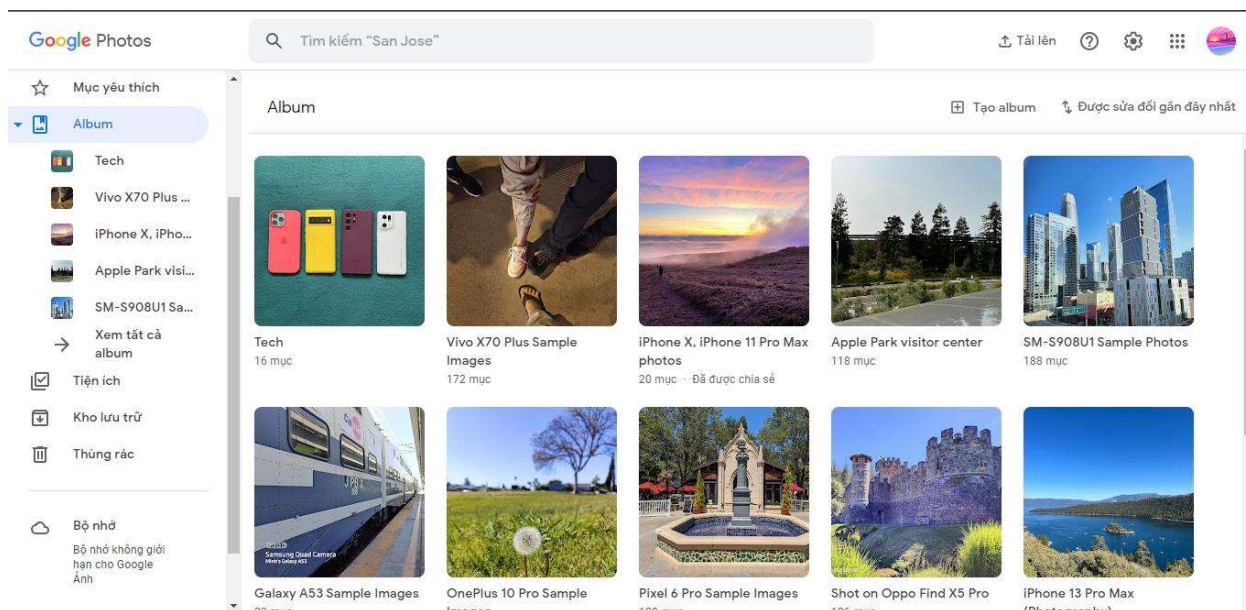


Рисунок 1.2.2 – Інтерфейс Google Photos

Синхронізація між пристроями, потужний пошук, автоматичне резервне копіювання, підтримка відтворення відео.

Потребує підключення до Інтернету, обмежений безкоштовний простір (15 ГБ), залежність від політики конфіденційності Google, відсутність локального управління без завантаження файлів.

1.2.3. Adobe Lightroom

Професійний інструмент для організації, редагування та перегляду фотографій із підтримкою відео, групування за датами, тегами, рейтингами.



Рисунок 1.2.3 – Інтерфейс Adobe Lightroom

Розширені можливості редагування, підтримка EXIF, інтеграція з хмарою, гнучке сортування.

Висока вартість підписки, складність для новачків, орієнтація на професійних користувачів, обмежена підтримка відтворення відео.

1.2.4. VLC Media Player

Функціональність: Багатофункціональний медіаплеєр із підтримкою широкого спектру форматів відео та зображень, базовим управлінням файлами через плейлисти.

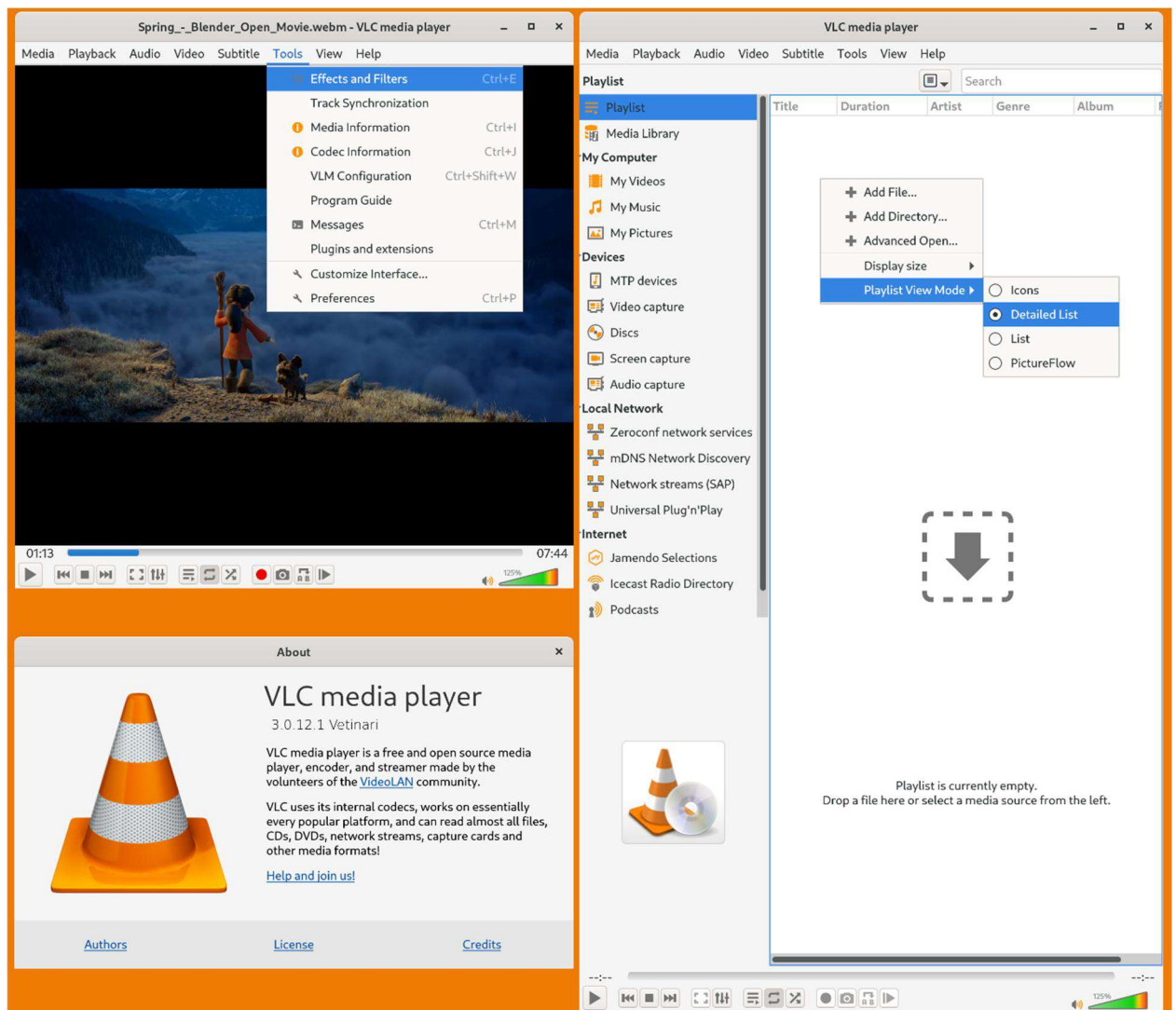


Рисунок 1.2.4 – Інтерфейс VLC Media Player

Безкоштовний, відкритий код, потужний плеєр із функціями регулювання гучності, позиції, швидкості відтворення.

Відсутність функцій архівування чи групування файлів, орієнтація виключно на відтворення, а не на організацію.

1.2.5. Порівняльна таблиця

Аналіз аналогів показує, що більшість рішень або орієнтовані на професійне використання (Adobe Lightroom), або потребують хмарного зберігання (Google Photos), або мають обмежений функціонал для організації архівів (Провідник, "Фотографії", VLC). Жоден із розглянутих аналогів не поєднує в собі локальне зберігання, автоматичне групування за EXIF-даними, повноцінне відтворення відео з елементами керування та простоту використання в одному безкоштовному рішенні.

Таблиця 1.2.5.1.

Порівняльна таблиця аналогів

Програма/ Сервіс	Групування за датою	Відтворення відео	Додавання/ Видалення	Локальність	Простота використання	Вартість
Провідник Windows	Частково (ручне)	Ні	Так	Так	Висока	Безкоштовно
"Фотографії" Windows	Так	Так (базове)	Ні	Так	Висока	Безкоштовно
Google Photos	Так	Так	Так	Ні	Висока	Умовно безкоштовно
Adobe Lightroom	Так	Так (обмежено)	Так	Так	Середня	Платно
VLC Media Player	Ні	Так (повне)	Ні	Так	Середня	Безкоштовно

Розроблювана програма на базі Windows Forms і .NET 8 спрямована на заповнення цієї ніші. Вона пропонує:

- локальне управління архівом без залежності від Інтернету;
- автоматичне групування файлів за датами з використанням EXIF;
- інтеграцію з Vlc.DotNet для повноцінного відтворення відео;
- простий і зрозумілий інтерфейс для користувачів-початківців;
- безкоштовність і відкритий потенціал для подальшого розвитку.

Основною перевагою розроблюваної системи є її спеціалізація на потребах сімейного архівування, що вирізняє її серед універсальних чи комерційних аналогів.

1.3. Вибір та обґрунтування технологій розробки

Для реалізації програми «Програма для ведення сімейного архіву фотографій та відео» необхідно обрати технології, які забезпечать високу продуктивність, зручність розробки, сумісність із сучасними системами та відповідність поставленим задачам. У цьому підрозділі розглядається вибір технологій, зокрема Windows Forms і .NET 8, а також допоміжних бібліотек, таких як Vlc.DotNet, із обґрунтуванням їхньої доцільності.

1.3.1. Вибір платформи розробки

.NET 8 — це сучасна версія платформи розробки від Microsoft, яка підтримує створення кросплатформених і високопродуктивних додатків. Основні причини вибору .NET 8 для даного проєкту:

- .NET 8 включає численні оптимізації, такі як покращений JIT-компілятор (Just-In-Time) і підтримка асинхронного програмування, що забезпечує швидку обробку великих обсягів файлів — ключову вимогу для роботи з медіаархівами.
- Платформа підтримує всі актуальні версії Windows, що робить програму доступною для широкого кола користувачів.
- .NET 8 надає вбудовані бібліотеки для роботи з файлами (System.IO), зображеннями (System.Drawing) і асинхронними операціями (System.Threading.Tasks), що спрощує реалізацію завантаження, групування та відображення медіафайлів.
- Інтеграція з такими компонентами, як Vlc.DotNet, є простою завдяки NuGet-пакетам, що дозволяє додати відтворення відео без написання складного коду з нуля.
- Як одна з найновіших версій на момент розробки, .NET 8 гарантує довготривалу підтримку та відповідність сучасним стандартам програмування.

Альтернативи, такі як .NET Framework або інші платформи (наприклад, Java чи Python), менш доцільні: .NET Framework є застарілим і менш

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

продуктивним, а Java та Python ускладнюють створення нативних настільних додатків для Windows із простим розгортанням.

1.3.2. Вибір технології інтерфейсу

Windows Forms — це технологія для створення настільних додатків із графічним інтерфейсом у екосистемі .NET. Її вибір обґрунтований такими факторами:

- Windows Forms має інтуїтивний дизайнер у Visual Studio, що дозволяє швидко створювати інтерфейс із використанням елементів управління (TreeView, PictureBox, Button тощо), ідеально підходячи для невеликих проєктів, таких як сімейний архів.
- Технологія перевірена часом і підтримується в .NET 8, що гарантує надійність і сумісність із сучасними версіями Windows.
- Windows Forms легко інтегрується з .NET-бібліотеками для роботи з файлами та зображеннями, а також із зовнішніми компонентами, такими як Vlc.DotNet.
- На відміну від сучасніших технологій, таких як WPF чи WinUI, Windows Forms не потребує значних ресурсів, що важливо для користувачів із базовими конфігураціями ПК.

Альтернативи, такі як WPF (Windows Presentation Foundation) або WinUI, пропонують кращу графіку та анімацію, але їхня складність і вищі вимоги до ресурсів роблять їх менш доцільними для простої програми сімейного архіву. Windows Forms забезпечує оптимальний баланс між функціональністю та легкістю розробки.

1.3.3. Вибір бібліотеки для відтворення відео

Для відтворення відеофайлів обрано бібліотеку Vlc.DotNet, яка є обгорткою для VLC Media Player у .NET. Обґрунтування вибору:

- VLC підтримує більшість популярних відеокодеків (.mp4, .avi тощо), що усуває потребу в додаткових кодеках на стороні користувача.
- Надає розширені можливості керування відтворенням (Play, Pause, Stop, регулювання гучності, позиції), що відповідає вимогам до програми.

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

- Vlc.DotNet легко вбудовується як елемент управління в Windows Forms, дозволяючи розмістити плеєр у формі та керувати ним через код.
- Бібліотека є відкритою і безкоштовною, що відповідає концепції розробки доступного програмного забезпечення.

Альтернативи, такі як Windows Media Player (через WMPLib) або створення власного плеєра, менш практичні: WMPLib має обмежену інтеграцію з Windows Forms і меншу підтримку форматів, а власний плеєр потребує значних зусиль і знань у обробці мультимедіа.

1.3.4. Додаткові технології та інструменти

- Використання System.Drawing для читання EXIF-даних (зокрема, PropertyTagDateTimeOriginal) дозволяє групувати фотографії за датою створення, що є ключовою функцією програми.
- Застосування async/await у .NET 8 забезпечує плавну роботу інтерфейсу під час завантаження файлів, що критично для великих архівів.
- Properties.Settings у .NET дозволяє зберігати вибір користувача (наприклад, шлях до архіву) між сеансами роботи.

Комбінація Windows Forms, .NET 8 і Vlc.DotNet є обґрунтованим вибором для створення програми сімейного архіву, поєднуючи простоту, функціональність і сучасні можливості розробки.

					ДР.122.042.024.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 1

Перший розділ дипломної роботи присвячений огляду та аналізу технологій розробки програми для ведення сімейного архіву фотографій та відео. У процесі дослідження було сформульовано основну задачу розробки, яка полягає у створенні зручного програмного забезпечення для локального управління медіафайлами з функціями групування за датами, додавання, видалення та відтворення вмісту. Аналіз існуючих аналогів показав, що вбудовані засоби Windows, такі як Провідник і додаток "Фотографії", а також сторонні рішення, наприклад Google Photos, Adobe Lightroom і VLC Media Player, мають певні обмеження, такі як відсутність повноцінного локального архівування, складність використання або залежність від хмарних сервісів. Розроблювана програма вирізняється спеціалізацією на сімейних архівах, пропонуючи автоматичне групування за EXIF-даними, інтеграцію відтворення відео та простоту інтерфейсу, що робить її унікальною серед розглянутих аналогів.

Вибір технологій розробки обґрунтовано їхньою відповідністю поставленим задачам. Платформа .NET 8 обрана завдяки високій продуктивності, підтримці асинхронного програмування та сумісності з Windows, що забезпечує ефективну роботу з великими обсягами даних і сучасними системами. Windows Forms стала основою для інтерфейсу через простоту розробки, стабільність і низькі системні вимоги, що ідеально підходить для створення настільного додатка з інтуїтивним дизайном. Для відтворення відео використано бібліотеку Vlc.DotNet, яка забезпечує підтримку широкого спектру форматів і розширені можливості керування, легко інтегруючись із Windows Forms. Додаткові інструменти, такі як вбудовані бібліотеки .NET для роботи з EXIF і збереження налаштувань, доповнюють технологічний стек, надаючи програмі необхідну функціональність.

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА

2.1. Вибір алгоритмів та їх ефективність

Розробка програми для ведення сімейного архіву фотографій та відео потребує ретельного вибору алгоритмів, які забезпечать швидкодію, коректність обробки даних і зручність для користувача. У цьому підрозділі розглядаються ключові алгоритми, використані в програмі, їхня роль у реалізації функціональності, а також аналіз їхньої ефективності з точки зору часової та просторової складності.

Завантаження та групування медіафайлів

Одним із основних завдань програми є завантаження медіафайлів із вибраної користувачем папки та їх групування за роками й місяцями створення. Для цього реалізовано алгоритм, який включає кілька етапів:

- **Сканування директорії**

Використовується метод `Directory.GetFiles` із параметром `SearchOption.AllDirectories` для рекурсивного пошуку файлів із підтримуваними розширеннями (`.jpg`, `.png`, `.mp4`, `.avi`). Для підвищення продуктивності застосовується фільтрація за розширеннями через LINQ-запит `.Where`. Часова складність цього етапу залежить від кількості файлів і структури директорії, наближаючись до $O(n)$, де n — загальна кількість файлів у всіх підпапках.

- **Отримання дати створення**

Для кожного файлу викликається асинхронна функція `GetCreationDateAsync`, яка спочатку намагається отримати дату з EXIF-даних (тег `PropertyTagDateTimeOriginal`) для зображень, а в разі невдачі повертає дату створення файлу (`File.GetCreationTime`). Читання EXIF реалізовано через `System.Drawing`, що має константну складність $O(1)$ для одного файлу, але загальна складність для n файлів становить $O(n)$. Асинхронність дозволяє уникнути блокування інтерфейсу, розподіляючи обробку між потоками.

- **Групування**

Файли групуються за роками та місяцями в словник `Dictionary<string, Dictionary<string, List<string>>>>`. Додавання файлу до структури має складність $O(1)$ завдяки хеш-таблиці, а загальна складність групування для n файлів становить $O(n)$. Просторова складність залежить від кількості унікальних років і місяців, але в середньому також близька до $O(n)$.

Ефективність цього алгоритму полягає в його лінійній часовій складності $O(n)$, що є оптимальним для обробки великих архівів. Використання асинхронного підходу додатково підвищує продуктивність, дозволяючи інтерфейсу залишатися чутливим під час сканування.

Відображення прогресу завантаження

Для відображення прогресу завантаження використано механізм `IProgress<int>`, який оновлює `ProgressBar` у головному потоці через `Invoke`. Алгоритм обчислює відсоток оброблених файлів як $(\text{processedFiles} / \text{totalFiles}) * 100$ і передає його в UI. Часова складність цього процесу становить $O(n)$ через ітерацію по всіх файлах, але оновлення UI відбувається з мінімальною затримкою завдяки асинхронності. Просторова складність мінімальна — $O(1)$, оскільки зберігається лише лічильник і загальна кількість файлів.

Відтворення та керування відео

Відтворення відео реалізовано через бібліотеку `Vlc.DotNet`, яка інкапсулює низькорівневі операції `VLC Media Player`. Алгоритми керування включають:

- Метод `vlcControl.Play()` запускає відео, а `SetMedia` задає файл. Операція має константну складність $O(1)$, але залежить від швидкості декодування відео `VLC`.
- `vlcControl.Pause()` і `vlcControl.Stop()` також мають складність $O(1)$, забезпечуючи миттєву реакцію.
- `vlcControl.Position` (0.0-1.0) дозволяє переміщатися по відео за допомогою `TrackBar`. Оновлення позиції через таймер

(UpdateVideoPosition) виконується кожні 500 мс із складністю $O(1)$ на ітерацію, а загальна ефективність залежить від тривалості відео.

- `vlcControl.Audio.Volume` змінює гучність у реальному часі зі складністю $O(1)$.

Ефективність цих алгоритмів висока завдяки низькій часовій складності та інтеграції з оптимізованим VLC-двигуном. Просторова складність мінімальна, оскільки зберігається лише поточний стан плеєра.

Обробка вибору файлу в TreeView

При виборі файлу в TreeView алгоритм `TreeView_AfterSelect` визначає тип файлу за розширенням і виконує відповідну дію (відображення зображення чи відтворення відео). Перевірка розширення та запуск операції мають складність $O(1)$, а загальна ефективність залежить від швидкості завантаження файлу (`Image.FromFile` чи `vlcControl.SetMedia`). Звільнення ресурсів (`Image.Dispose`) також має константну складність, що запобігає витокам пам'яті.

Обрані алгоритми є оптимальними для поставлених задач. Лінійна складність $O(n)$ для завантаження та групування файлів забезпечує прийнятну продуктивність навіть для великих архівів (тисячі файлів), а асинхронність унеможливорює "зависання" програми. Операції відтворення відео та керування мають константну складність $O(1)$, що гарантує швидку реакцію інтерфейсу. Просторова складність у більшості випадків пропорційна кількості файлів ($O(n)$), але завдяки використанню словників і динамічних списків пам'ять використовується ефективно.

2.2. Спроектовані класи програми

Розробка програми для ведення сімейного архіву фотографій та відео на базі Windows Forms і .NET 8 потребує чіткої структури класів, які забезпечують реалізацію функціональності, зручність підтримки та розширюваність коду. У цьому підрозділі описано спроектовані класи, їхню роль у програмі та взаємодію між ними.

Основний клас - MainForm

Клас MainForm є центральним у програмі, успадковуючись від Form із простору імен System.Windows.Forms. Він відповідає за інтерфейс користувача, обробку подій і координацію роботи всіх компонентів. Головна форма програми, яка містить усі елементи управління та логіку взаємодії з користувачем. Координує роботу всіх елементів UI, викликає методи для обробки файлів і відтворення через VlcControl.

```
3 references
partial class MainForm
{
    /// <summary> Required designer variable.
    private System.ComponentModel.IContainer components = null;
    /// <summary> Clean up any resources being used.
    0 references
    protected override void Dispose(bool disposing) {...}
    Windows Form Designer generated code
    private TreeView treeView1;
    private Button btnAdd;
    private Button btnDelete;
    private Panel mediaPanel;
    private Panel panelMenu;
    private PictureBox pictureBox1;
    private ProgressBar progressBar1;
    private Button btnChoosePath;
    private TextBox archivePath;
    private Button buttonAbout;
    private Panel videoControlPanel;
    private Button btnPlayVideo;
    private Button btnPauseVideo;
    private Button btnStopVideo;
    private TrackBar trackBarVolume;
    private TrackBar trackBarPosition;
    private Label labelVolume;
    private Label labelPosition;
    private VlcControl vlcControl; // Використовуємо VLC для відео
    private System.Windows.Forms.Timer positionTimer;
}
```

Рисунок 2.2.1 – Клас MainForm

Поля:

- private VlcControl? vlcControl — об'єкт для відтворення відео (nullable, щоб обробляти випадки невдалої ініціалізації).
- private TreeView treeView1 — дерево для відображення структури архіву (роки, місяці, файли).
- private PictureBox pictureBox1 — область для перегляду зображень.
- private ProgressBar progressBar1 — прогрес-бар для відображення процесу завантаження файлів.
- private TextBox archivePath — текстове поле для відображення та збереження шляху до архіву.
- private Panel mediaPanel — контейнер для PictureBox і VlcControl.
- private Panel videoControlPanel — панель із кнопками керування відео (btnPlayVideo, btnPauseVideo, btnStopVideo) та регуляторами (trackBarVolume, trackBarPosition).
- private System.Windows.Forms.Timer positionTimer — таймер для оновлення позиції відтворення відео.

Методи:

- InitializeUI() — ініціалізація інтерфейсу, створення VlcControl і налаштування подій.
- LoadMediaFilesAsync() і LoadMediaFiles() — асинхронне завантаження та групування файлів у TreeView.
- GetCreationDateAsync(string filePath) — отримання дати створення файлу з EXIF або файлової системи.
- TreeView_AfterSelect(object sender, TreeViewEventArgs e) — обробка вибору файлу для відображення чи відтворення.
- BtnAdd_Click, BtnDelete_Click, BtnChoosePath_Click — обробники для додавання, видалення файлів і вибору шляху.
- BtnPlayVideo_Click, BtnPauseVideo_Click, BtnStopVideo_Click — керування відтворенням відео.

					ДР.122.042.024.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

- `TrackBarVolume_Scroll`, `TrackBarPosition_Scroll` — регулювання гучності та позиції відео.
- `UpdatePositionTimer(bool enable)`, `UpdateVideoPosition()`, `UpdatePositionLabel()` — оновлення позиції відео в UI.
- `MainForm_FormClosing` — очищення ресурсів при закритті форми.

Допоміжний клас: `AboutForm`

Клас `AboutForm` успадковується від `Form` і відображає інформацію про програму. Модальне вікно з інформацією про автора, версію та мету програми. Викликається з `MainForm` через `buttonAbout_Click` і працює як незалежне модальне вікно.

```

3 references
partial class AboutForm
{
    /// <summary> Required designer variable.
    private System.ComponentModel.IContainer components = null;
    /// <summary> Clean up any resources being used.
    0 references
    protected override void Dispose(bool disposing) ...
    Windows Form Designer generated code
    private Label labelAbout;
    private Button buttonClose;
}

```

Рисунок 2.2.2 – Клас `AboutForm`

Поля:

- `private Label labelAbout` — текст із інформацією.
- `private Button buttonClose` — кнопка закриття вікна.

Методи:

- `AboutForm_Load(object sender, EventArgs e)` — заповнення `labelAbout` інформацією (версія, автор, дата).
- `ButtonClose_Click(object sender, EventArgs e)` — закриття форми.

MainForm створює екземпляр AboutForm і відображає його як модальне вікно через ShowDialog().

MainForm ініціалізує VlcControl, додає його до mediaPanel і викликає методи (Play, Pause, Stop) для керування відео.

MainForm керує TreeView, PictureBox, ProgressBar, кнопками та панелями через події та методи, забезпечуючи синхронізацію стану інтерфейсу з діями користувача.

Спроектовані класи MainForm і AboutForm повністю охоплюють функціональність програми. MainForm виступає основним координатором, інтегруючи роботу з файлами, інтерфейсом і відтворенням відео, тоді як AboutForm забезпечує інформаційну складову. Структура класів є оптимальною для настільного додатка середнього розміру, поєднуючи простоту реалізації з можливістю подальшого розвитку.

2.3. Програмна реалізація

У цьому підрозділі описано практичну реалізацію програми для ведення сімейного архіву фотографій та відео на базі Windows Forms і .NET 8. Розглядаються ключові аспекти коду, включаючи ініціалізацію інтерфейсу, обробку файлів, відтворення медіа та керування подіями.

Для роботи програми було розроблено дві форми:

- **Form1.cs** – головна форма для ведення сімейного архіву.
- **AboutForm.cs** – форма "Про програму".

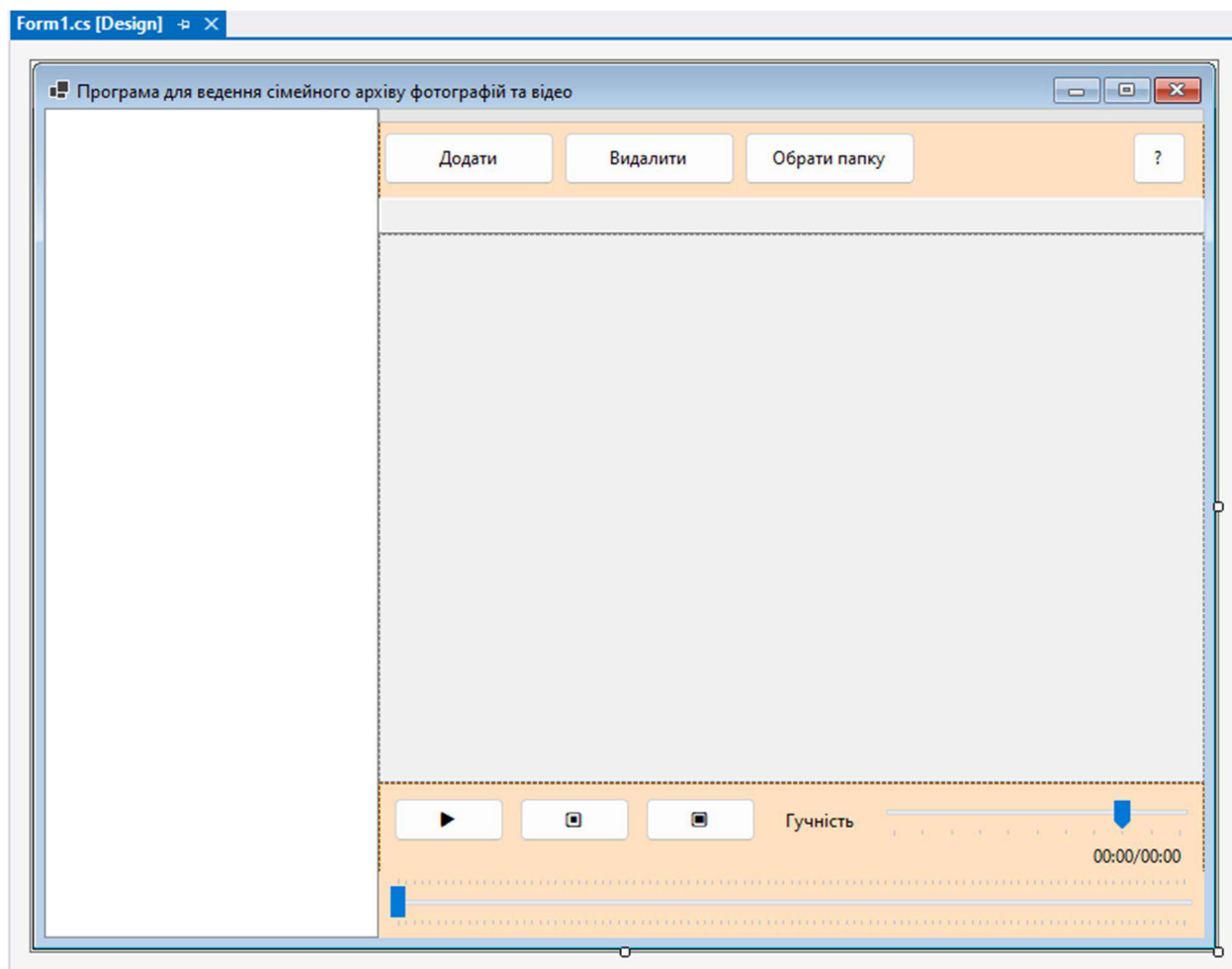


Рисунок 2.3.1 – Головна форма для ведення сімейного архіву

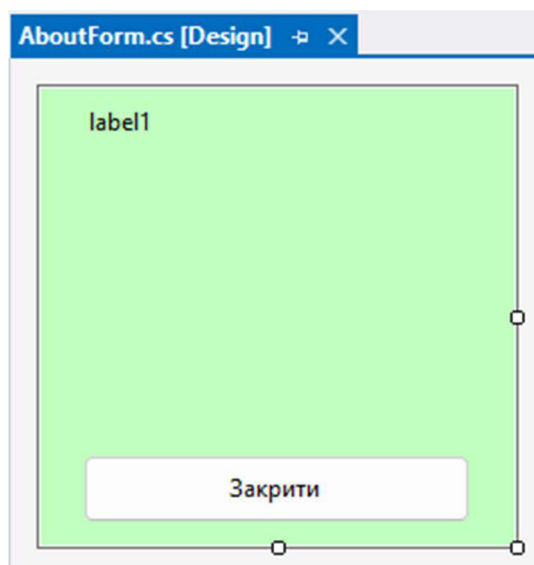


Рисунок 2.3.2 – Форма "Про програму"

Наведено фрагменти коду з поясненнями для основних функцій:

Ініціалізація програми

Програма запускається з класу MainForm, який є головною формою. Ініціалізація інтерфейсу відбувається в методі InitializeUI, викликаний у MainForm_Load.

```
private void MainForm_Load(object sender, EventArgs e)
{
    InitializeUI();
    LoadMediaFilesAsync();
}

private void InitializeUI()
{
    archivePath.Text = string.IsNullOrEmpty(Properties.Settings.Default.ArchivePath)
        ? ChooseArchivePath()
        : Properties.Settings.Default.ArchivePath;
    try
    {
        vlcControl = new VlcControl();
        vlcControl.BeginInit();
        vlcControl.VlcLibDirectory = new DirectoryInfo(@"C:\Program Files\VideoLAN\VLC")
        vlcControl.Dock = DockStyle.Fill;
        vlcControl.Visible = false;
        vlcControl.EndInit();
        vlcControl.EndReached += (sender, e) => this.Invoke((Action)(() =>
        {
            trackBarPosition.Value = 0;
            UpdatePositionLabel();
            UpdatePositionTimer(false);
            videoControlPanel.Visible = false;
            btnPlayVideo.Text = "▶";
            UpdateVideoControlsState();
        }));
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка ініціалізації VLC: {ex.Message}\nВідтворення відео буд
        vlcControl = null;
    }
    if (vlcControl != null)
    {
        mediaPanel.Controls.Add(vlcControl);
    }
}
```

Рисунок 2.3.3 – Ініціалізація інтерфейсу

Поле archivePath отримує шлях із налаштувань або через вибір користувача. VlcControl ініціалізується з обробником EndReached для скидання стану після завершення відео. У разі помилки VLC відключено, що не впливає на решту функцій.

Завантаження та групування файлів

Метод LoadMediaFilesAsync і допоміжний LoadMediaFiles відповідають за асинхронне завантаження файлів і їх групування. Файли скануються асинхронно, групуються в словник за роками та місяцями, а прогрес відображається через ProgressBar. Використання Invoke забезпечує безпечне оновлення UI з іншого потоку.

```
private async Task LoadMediaFiles()
{
    try
    {
        if (!Directory.Exists/archivePath.Text)...

        await Task.Run(() => treeView1.Invoke((Action)(() =>...)));

        string[] supportedExtensions = { ".jpg", ".png", ".mp4", ".avi" };
        var files = await Task.Run(() => Directory.GetFiles/archivePath.Text, ".*", SearchOption
            .Where(f => supportedExtensions.Contains(Path.GetExtension(f).ToLower()))
            .ToList());

        if (files.Count == 0)...

        var groupedFiles = new Dictionary<string, Dictionary<string, List<string>>>>();
        var progress = new Progress<int>(percent =>...);

        int totalFiles = files.Count;
        int processedFiles = 0;

        foreach (var file in files)...

        await Task.Run(() => treeView1.Invoke((Action)(() =>...)));
    }
    catch (Exception ex)
    {
        await Task.Run(() => treeView1.Invoke((Action)(() =>...)));
    }
}
```

Рисунок 2.3.4 – Метод LoadMediaFilesAsync

Відтворення та керування відео

Відтворення реалізовано через VlcControl із кнопками керування в videoControlPanel. При виборі відео в TreeView воно автоматично відтворюється, а btnPlayVideo змінює текст на "II". Кнопка перемикає між відтворенням і паузою, оновлюючи стан UI.

```
1 reference
private void TreeView_AfterSelect(object sender, TreeViewEventArgs e)
{
    if (e.Node?.Tag is string filePath && File.Exists(filePath))
    {
        string extension = Path.GetExtension(filePath).ToLower();

        vlcControl?.Stop();
        pictureBox1.Visible = false;
        if (vlcControl != null) vlcControl.Visible = false;
        UpdatePositionTimer(false);
        videoControlPanel.Visible = false;

        if (extension == ".jpg" || extension == ".png")...
        else if (extension == ".mp4" || extension == ".avi")...
    }
}
```

Рисунок 2.3.5 – Метод LoadMediaFilesAsync

```
private void BtnPlayVideo_Click(object sender, EventArgs e)
{
    if (vlcControl != null && treeView1.SelectedNode?.Tag is string filePath && File.Exists(filePath))
    {
        string extension = Path.GetExtension(filePath).ToLower();
        if (extension == ".mp4" || extension == ".avi")
        {
            try...
        }
    }
}
```

Рисунок 2.3.6 – Метод BtnPlayVideo_Click

Регулювання позиції та гучності

Таймер оновлює позицію кожні 500 мс, а TrackBar дозволяє регулювати гучність у реальному часі.

```
private void UpdatePositionTimer(bool enable)
{
    if (positionTimer == null)
    {
        positionTimer = new System.Windows.Forms.Timer { Interval = 500 };
        positionTimer.Tick += (s, e) => UpdateVideoPosition();
    }
    positionTimer.Enabled = enable;
}

private void UpdateVideoPosition()
{
    if (vlcControl != null && vlcControl.IsPlaying)
    {
        int position = (int)(vlcControl.Position * 100);
        trackBarPosition.Value = Math.Min(Math.Max(position, 0), 100);
        UpdatePositionLabel();
    }
}

private void TrackBarVolume_Scroll(object sender, EventArgs e)
{
    if (vlcControl != null) vlcControl.Audio.Volume = trackBarVolume.Value;
}
```

Рисунок 2.3.7 – Методи регулювання позиції та гучності

Програмна реалізація охоплює всі необхідні функції: асинхронне завантаження файлів, групування за датами, перегляд зображень і відтворення відео з керуванням. Код структурований для стабільності (з обробкою винятків) і чутливості UI (асинхронність і Invoke). Ця основа забезпечує виконання вимог до програми, описаних у попередніх розділах.

Висновки до розділу 2

Другий розділ дипломної роботи присвячений проєктуванню та розробці програми для ведення сімейного архіву фотографій та відео. У процесі роботи було проведено вибір алгоритмів, спроектовано класи та здійснено їхню програмну реалізацію, що дозволило створити функціональне програмне забезпечення, яке відповідає поставленим задачам.

Аналіз і вибір алгоритмів показали, що використання асинхронного підходу до завантаження та групування файлів із лінійною часовою складністю $O(n)$ забезпечує високу продуктивність навіть при роботі з великими архівами. Застосування вбудованих методів .NET для сканування директорій, читання EXIF-даних і відтворення відео через Vlc.DotNet гарантує швидкодію та стабільність. Регулювання позиції та гучності відео з константною складністю $O(1)$ дозволяє реалізувати чутливе керування медіа, що покращує користувацький досвід.

Проєктування класів програми базується на двох основних компонентах: MainForm як центральному класі для управління інтерфейсом і логікою, та AboutForm для інформаційної складової. Структура класів є модульною та простою, що полегшує підтримку коду, а використання подій і обробників забезпечує гнучкість для подальшого розширення функціональності. Координація між компонентами інтерфейсу та медіаплеєром реалізована ефективно, з урахуванням звільнення ресурсів для уникнення витоків пам'яті.

Програмна реалізація підтвердила працездатність обраних підходів. Асинхронне завантаження файлів із відображенням прогресу, групування за датами з підтримкою EXIF, а також інтеграція відтворення відео з кнопками керування були успішно втілені в коді. Обробка винятків і безпечне оновлення інтерфейсу через Invoke забезпечують стабільність програми навіть у разі помилок чи великих обсягів даних. Реалізований функціонал відповідає вимогам до зручності, локальності та повноцінного управління сімейним архівом.

					ДР.122.042.024.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1. Мінімальні вимоги до системи

Програма для ведення сімейного архіву фотографій та відео розроблена з урахуванням доступності для широкого кола користувачів, тому її мінімальні системні вимоги є досить помірними. Для коректної роботи програми необхідна операційна система Windows 10 або новіша версія, що забезпечує сумісність із платформою .NET 8. Рекомендується встановити останню версію .NET 8 Runtime, яка є обов'язковою для запуску програми.

Центральний процесор із частотою не нижче 1 ГГц достатній для базових операцій, таких як завантаження файлів і перегляд зображень. Програма не використовує багатопоточність на рівні апаратного прискорення, тому багатоядерний процесор не є обов'язковим.

Оперативна пам'ять обсягом 2 ГБ дозволяє комфортно працювати з невеликими архівами до кількох сотень файлів. Для роботи з більшими обсягами даних бажано мати 4 ГБ оперативної пам'яті або більше.

Вільний простір на диску залежить від розміру архіву, але для самої програми достатньо 100 МБ для інсталяції.

Відеокарта не є критичним компонентом, оскільки програма не використовує апаратне прискорення для обробки зображень чи відео. Наявність базового графічного адаптера, інтегрованого в процесор, цілком достатня для відображення інтерфейсу.

Для відтворення відео необхідно встановити бібліотеки VLC Media Player версії 3.0 або новіше, які розміщуються за замовчуванням у C:\Program Files\VideoLAN\VLC. Роздільна здатність екрана 1024x768 пікселів забезпечує нормальне відображення всіх елементів інтерфейсу без горизонтального прокручування.

Підтримка стандартних файлових систем, таких як NTFS або FAT32, забезпечує сумісність із більшістю накопичувачів. Для коректного читання EXIF-даних у зображеннях система має підтримувати кодування ASCII, що є стандартним для Windows. Швидкість роботи програми залежить від

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

продуктивності диска, тому SSD-накопичувач значно прискорить завантаження великих архівів порівняно з HDD. Антивірусне програмне забезпечення не повинно блокувати доступ до файлів, інакше можуть виникати помилки під час сканування директорії.

3.2. Інтерфейс користувача

Інтерфейс користувача програми для ведення сімейного архіву фотографій та відео розроблено з урахуванням принципів зручності, інтуїтивності та функціональності, що є ключовими для цільової аудиторії — користувачів із різним рівнем комп'ютерної грамотності. Реалізація інтерфейсу базується на технології Windows Forms, яка забезпечує простий і нативний вигляд у середовищі Windows, а також легкість у налаштуванні елементів управління.

Головне вікно програми (MainForm) поділено на кілька функціональних зон, кожна з яких має чітке призначення. У лівій частині розміщено елемент TreeView (treeView1), який відображає ієрархічну структуру архіву, організовану за роками та місяцями створення файлів. Кожен рік є вузлом верхнього рівня, що розгортається в підвузли місяців, а місяці, у свою чергу, містять список медіафайлів із їхніми назвами. Такий підхід дозволяє користувачу швидко орієнтуватися в архіві та знаходити потрібний файл за датою.

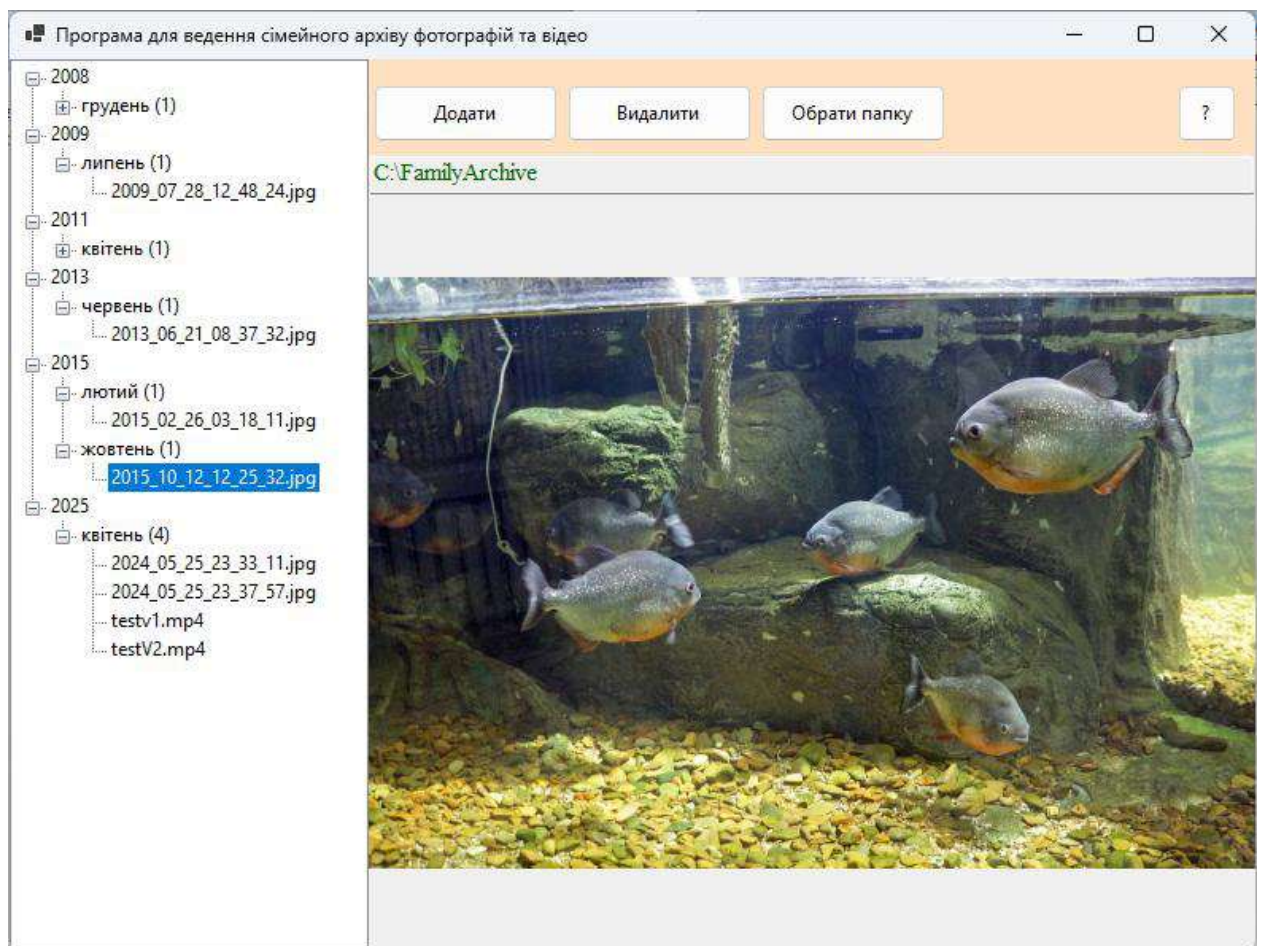


Рисунок 3.2.1 – Головне вікно програми

Праворуч від TreeView розташована основна область перегляду — панель mediaPanel, яка вміщує PictureBox (pictureBox1) для відображення зображень і VlcControl для відтворення відео. Ці елементи мають властивість Dock = DockStyle.Fill, що забезпечує їхнє автоматичне масштабування відповідно до розміру вікна. Для зображень у PictureBox використано режим SizeMode = PictureBoxSizeMode.Zoom, що дозволяє пропорційно відображати файли різних розмірів без спотворення.

Над mediaPanel розміщена панель меню (panelMenu), яка містить основні елементи керування програмою. Текстове поле archivePath відображає поточний шлях до архіву та має властивість ReadOnly, щоб уникнути випадкових змін користувачем. Кнопка btnChoosePath ("Обрати папку") відкриває діалог FolderBrowserDialog для зміни шляху, а кнопки btnAdd ("Додати") та btnDelete ("Видалити") дозволяють додавати нові файли через OpenFileDialog або видаляти вибрані з підтвердженням через MessageBox.

Кнопка `buttonAbout` ("?") відкриває модальне вікно `AboutForm` з інформацією про програму. Прогрес-бар `progressBar1`, розташований у верхній частині `panelMenu`, з'являється під час завантаження файлів і показує відсоток виконаної роботи, після чого автоматично ховається.

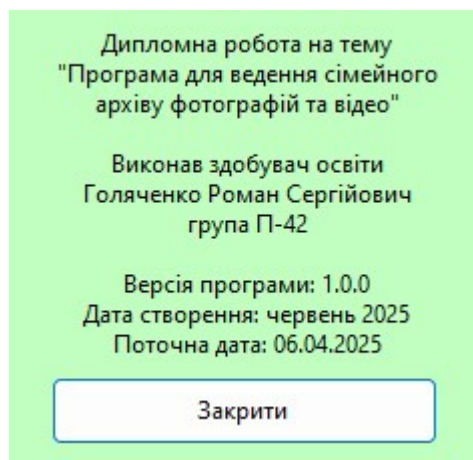


Рисунок 3.2.2 – Вікно з інформацією про програму

У нижній частині вікна розташована панель `videoControlPanel`, яка відображається лише під час роботи з відеофайлами. Вона включає три кнопки: `btnPlayVideo` ("▶" або "⏏" залежно від стану), `btnPauseVideo` ("⏏") і `btnStopVideo` ("■"), які забезпечують базове керування відтворенням. Регулятор `trackBarVolume` із міткою `labelVolume` ("Гучність") дозволяє налаштовувати рівень звуку в діапазоні від 0 до 100, а `trackBarPosition` із міткою `labelPosition` (формат "mm:ss/mm:ss") дає змогу переміщатися по часовій шкалі відео. Панель має властивість `Visible = false` за замовчуванням і активується лише при виборі відео в `TreeView`.

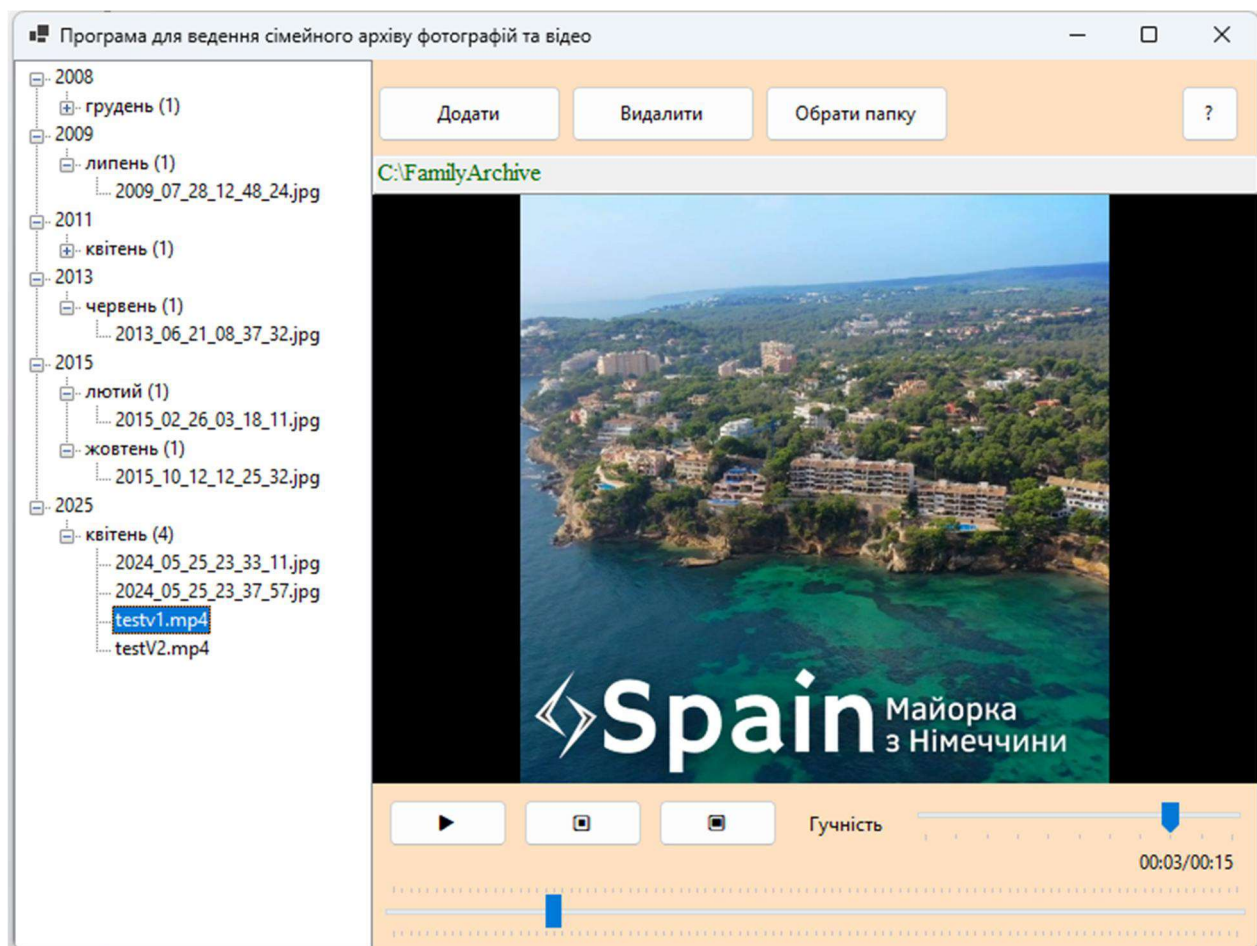


Рисунок 3.2.3 – Панель керування відтворенням відео

Інтерфейс розроблено з акцентом на візуальну чіткість: панелі `panelMenu` і `videoControlPanel` мають світло-помаранчевий фон (`Color.FromArgb(255, 224, 192)`), що контрастує з основним білим тлом форми. Текстове поле `archivePath` використовує шрифт Times New Roman розміром 11.25 для легкого читання, а кнопки в `videoControlPanel` мають жирний шрифт для кращої помітності. Розташування елементів із використанням `Dock` забезпечує адаптивність до різних розмірів екрана, хоча мінімальна роздільна здатність 1024x768 рекомендована для повного відображення без прокручування.

Обробка подій у `TreeView_AfterSelect` автоматично перемикає видимість між `PictureBox` і `VlcControl` залежно від типу вибраного файлу, а також активує `videoControlPanel` для відео. Події кнопок (`Click`) і регуляторів (`Scroll`) синхронізовано з методами `VlcControl`, що забезпечує миттєву реакцію на дії користувача. Асинхронне завантаження файлів через

LoadMediaFilesAsync із оновленням progressBar1 гарантує, що інтерфейс залишається чутливим навіть під час тривалих операцій.

Інтерфейс користувача програми поєднує простоту навігації з повноцінним функціоналом для роботи з архівом. Він дозволяє легко переглядати зображення, відтворювати відео з базовим керуванням і управляти файлами, відповідаючи потребам як початківців, так і досвідчених користувачі

3.3. Інструкція для роботи з програмою

Програма для ведення сімейного архіву фотографій та відео розроблена для простого й зручного використання, тому її освоєння не потребує спеціальних знань. Ця інструкція описує основні кроки для початку роботи, управління архівом і перегляду медіафайлів.

Перед запуском програми переконайтеся, що на вашому комп'ютері встановлено операційну систему Windows 10 або новішу версію, а також .NET 8 Runtime. Для відтворення відео необхідно мати встановлений VLC Media Player (версія 3.0 або новіша) у стандартному розташуванні C:\Program Files\VideoLAN\VLC. Якщо VLC не встановлено, програма повідомить про помилку ініціалізації, але решта функцій залишаться доступними.

Щоб запустити програму, відкрийте виконуваний файл FamilyArchive.exe, який згенеровано після компіляції проєкту. При першому запуску з'явиться діалог вибору папки для архіву, де потрібно вказати директорію для зберігання ваших медіафайлів. Натисніть "ОК", щоб підтвердити вибір, і шлях автоматично збережеться для наступних запусків. Якщо папка порожня або не існує, програма створить її та повідомить про це через спливне вікно.

Після запуску ви побачите головне вікно програми з деревом файлів зліва та областю перегляду справа. Дерево файлів (TreeView) відображає структуру архіву, де файли згруповано за роками та місяцями створення. Щоб переглянути вміст, розгорніть потрібний рік, потім місяць, і клацніть на назву

					ДР.122.042.024.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

файлу. Якщо вибрано зображення (.jpg або .png), воно відобразиться в правій частині вікна. Якщо це відео (.mp4 або .avi), відтворення почнеться автоматично, а внизу з'явиться панель керування.

Для додавання нових файлів натисніть кнопку "Додати" у верхній частині вікна. У діалоговому вікні виберіть один або кілька файлів із розширеннями .jpg, .png, .mp4 чи .avi, після чого натисніть "Відкрити". Файли скопіюються до архіву, і дерево автоматично оновиться. Щоб видалити файл, виберіть його в дереві та натисніть "Видалити". З'явиться запит на підтвердження; після згоди файл буде видалено, а дерево оновлено.

Щоб змінити папку архіву, натисніть "Обрати папку" у верхній частині. У новому діалозі вкажіть іншу директорію та підтвердіть вибір. Програма перезавантажить файли з нової папки, відобразивши їх у дереві. Під час завантаження з'явиться прогрес-бар, який показує відсоток виконаної роботи; після завершення він зникне.

Для керування відтворенням відео використовуйте панель у нижній частині вікна, яка з'являється при виборі відеофайлу. Кнопка "►" (або "II", якщо відео вже відтворюється) запускає або ставить на паузу відтворення. Кнопка "II" додатково зупиняє відео, зберігаючи поточну позицію. Кнопка "■" повністю зупиняє відтворення та ховає панель. Регулятор гучності праворуч дозволяє налаштувати звук від 0 до 100, а регулятор позиції нижче — переміщатися по відео, показуючи час у форматі "mm:ss/mm:ss".

Якщо під час роботи виникають помилки (наприклад, файл не знайдено чи VLC не ініціалізовано), програма сповістить вас через спливне вікно з описом проблеми. У таких випадках перевірте наявність файлів у папці архіву або правильність встановлення VLC. Щоб переглянути інформацію про програму, натисніть кнопку "?" у верхньому правому куті — відкриється вікно з даними про версію, автора та дату створення.

Для завершення роботи просто закрийте вікно програми, натиснувши хрестик у верхньому правому куті. Усі ресурси (зображення, VLC) автоматично звільняться, а шлях до архіву збережеться для наступного

					ДР.122.042.024.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

запуску. Програма не потребує додаткових налаштувань і готова до використання одразу після встановлення.

Висновки до розділу 3

Третій розділ дипломної роботи присвячений аналізу результатів розробки програми для ведення сімейного архіву фотографій та відео, зокрема оцінці її технічних вимог, інтерфейсу користувача та інструкції з використання. Проведена робота дозволила оцінити практичну реалізацію проєкту та підтвердити його відповідність поставленим цілям.

Визначено мінімальні системні вимоги, які є доступними для більшості сучасних комп'ютерів із Windows 10 або новішими версіями, включаючи процесор із частотою 1 ГГц, 2 ГБ оперативної пам'яті та 100 МБ вільного місця на диску. Обов'язкова наявність .NET 8 Runtime і VLC Media Player для відтворення відео забезпечує сумісність і функціональність програми, при цьому її локальна природа усуває потребу в підключенні до Інтернету. Такі вимоги роблять програму легкою для розгортання на базових системах, що відповідає потребам цільової аудиторії.

Інтерфейс користувача, побудований на базі Windows Forms, виявився зручним і інтуїтивно зрозумілим завдяки чіткому поділу на функціональні зони: дерево файлів для навігації, область перегляду для медіа та панелі керування для взаємодії з архівом і відео. Використання елементів управління, таких як TreeView, PictureBox і VlcControl, у поєднанні з асинхронним оновленням прогрес-бару забезпечує плавність роботи та швидкий доступ до файлів. Панель керування відео з кнопками відтворення, паузи, зупинки та регуляторами гучності й позиції додає зручності при роботі з відеофайлами, а адаптивний дизайн із Dock дозволяє комфортно використовувати програму на різних роздільних здатностях екрана.

Інструкція для роботи з програмою детально описує всі етапи — від запуску та вибору папки до додавання, видалення й перегляду файлів. Вона

					ДР.122.042.024.ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

підкреслює простоту освоєння програми, адже автоматичне групування за датами, інтуїтивні кнопки та чіткі сповіщення про помилки роблять її доступною навіть для користувачів-початківців. Інструкція підтверджує, що програма не потребує складних налаштувань і готова до використання одразу після встановлення, що є однією з ключових переваг.

Результати розділу свідчать про успішну реалізацію програми, яка поєднує простоту, функціональність і локальність. Інтерфейс і інструкція адаптовані до потреб сімейного архівування, усуваючи недоліки аналогів, такі як складність чи залежність від хмари. Ці висновки підтверджують практичну цінність розробки та її готовність до використання, а також створюють основу для подальших удосконалень, якщо такі будуть потрібні.

					ДР.122.042.024.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Ось і дійшла до свого логічного завершення подорож у світ цифрових спогадів, де кожен файл — це маленька частинка життя, зафіксована у часі. Дипломна робота, присвячена створенню програми для ведення сімейного архіву фотографій та відео, стала не просто технічним завданням, а справжньою спробою вдихнути порядок і зручність у хаос цифрової ери. Від перших рядків коду до фінального запуску ця праця була сповнена прагненням подарувати користувачам інструмент, який береже найдорожче — пам'ять.

Розпочавши з аналізу технологій, ми проклали стежку через безмежні можливості .NET 8 і простоту Windows Forms, зупинившись на Vlc.DotNet як на вірному супутнику для відтворення відео. Крок за кроком алгоритми оживали, групуючи файли за датами, немов архіваріус, що дбайливо розкладає старі листи по полицях. Класи програми, наче майстерно зведені будівлі, утворили міцний фундамент, на якому розквітнув інтуїтивний інтерфейс — дерево файлів, що шепоче про минулі роки, і панель керування, яка відгукується на кожен дотик.

Програма, що народилася з цієї роботи, — це не просто набір функцій. Це ключ до сімейної скарбниці, де зображення оживають одним клацанням, а відео співає мелодію спогадів із можливістю зупинити мить чи налаштувати її звучання. Вона скромна у своїх вимогах до системи, але щедра у можливостях, пропонуючи кожному користувачу — від новачка до досвідченого — легкість і комфорт у роботі з архівом. Інструкція стала її голосом, що лагідно веде за руку, пояснюючи, як додати нову сторінку до історії чи стерти непотрібне.

Ця дипломна робота — не лише технічне досягнення, а й маленький внесок у збереження людських моментів. Вона показала, що навіть у світі складних систем і швидкоплинних даних можна створити щось просте, але значуще. Програма готова стати вірним хранителем сімейних архівів, а її код — відправною точкою для нових ідей і вдосконалень. Тож нехай кожен запуск цієї програми нагадує про те, що минуле завжди поруч — варто лише відкрити потрібну папку й натиснути "Play".

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Алфьоров М. А. Програмування на C# для початківців. Київ: Видавнича група BHV, 2020. 320 с.
2. Бойко В. І., Савчук О. П. Основи програмування в .NET Framework. Львів: Видавництво ЛНУ імені Івана Франка, 2019. 280 с.
3. Гаврилюк О. В. Розробка графічних інтерфейсів користувача засобами Windows Forms. Харків: ХНУРЕ, 2021. 245 с.
4. Григор'єв Ю. М. Асинхронне програмування в .NET: теорія та практика. Одеса: Одеський національний політехнічний університет, 2022. 198 с.
5. Дейтел П., Дейтел Г. C# 8 для програмістів. Київ: Діалектика, 2020. 960с.
6. Коваленко І. С. Технології обробки мультимедійних даних: підручник. Київ: КПІ ім. Ігоря Сікорського, 2021. 312 с.
7. Корнійчук В. В. Проєктування програмного забезпечення: методи та підходи. Дніпро: ДНУ, 2018. 256 с.
8. Ліберти Дж. Програмування на C# для професіоналів. Львів: Магнолія, 2019. 832 с.
9. Мак-Дональд М. Windows Forms у .NET: створення настільних додатків. Харків: Фабрика знань, 2020. 576 с.
10. Нагін А. І. Робота з файлами та директоріями в .NET: практичний посібник. Київ: Наукова думка, 2021. 184 с.
11. Петров О. О., Сидоренко Т. В. Сучасні технології розробки програмного забезпечення. Київ: Видавництво "Техніка", 2022. 340 с.
12. Ріхтер Дж. CLR via C#: програмування на платформі .NET. Дніпро: Баланс-Клуб, 2020. 896 с.
13. Скоупс А. Основи .NET Core та .NET 8: від теорії до практики. Львів: Новий Світ, 2023. 412 с.
14. Степаненко П. Д. Інтеграція мультимедійних компонентів у настільні додатки. Харків: ХПІ, 2019. 230 с.
15. Троелсен Е., Джепікс Ф. C# 8 і .NET Core 3: вичерпний посібник. Київ: Вільямс, 2021. 1248 с.

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

16. Шилдт Г. С#: повний довідник. Київ: Видавнича група "Школа", 2020. 1056 с.
17. Microsoft Docs. Офіційна документація .NET 8 [Електронний ресурс]. Режим доступу: <https://docs.microsoft.com/dotnet/>. Дата звернення: 01.04.2025.
18. VLC Developer Documentation. VLC Media Player API [Електронний ресурс]. Режим доступу: <https://www.videolan.org/developers/vlc.html>. Дата звернення: 02.04.2025.
19. Stack Overflow. Обговорення інтеграції Vlc.DotNet у Windows Forms [Електронний ресурс]. Режим доступу: <https://stackoverflow.com/questions/tagged/vlc-dotnet>. Дата звернення: 03.05.2025.
20. GitHub. Репозиторій Vlc.DotNet: документація та приклади коду [Електронний ресурс]. Режим доступу: <https://github.com/ZeBobo5/Vlc.DotNet>. Дата звернення: 04.06.2025.

					ДР.122.042.024.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Програмний код модулів

```

using Vlc.DotNet.Forms;
namespace FamilyArchive
{
    partial class MainForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>

```

					ДР.122.042.024.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45