

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАПИСУ ТА
ОБРОБКИ ВІДЕО З ВЕБКАМЕРИ

(назва теми)

SOFTWARE DEVELOPMENT FOR RECORDING AND PROCESSING VIDEO
FROM A WEB CAMERA

Виконав: студент(ка) 4 курсу, групи КНТ-130
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація) Інженерія
програмного забезпечення

ДІДЕНКО Н.О.

(ПРИЗВИЩЕ та ініціали)

Керівник СЕРДЮК С.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ПОДКОВАЛІХІНА О.О.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ДІДЕНКО Никіти Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для запису та обробки відео з вебкамери. Software Development for Recording and Processing Video from a Web Camera

керівник проєкту (роботи) к.т.н., доцент, СЕРДЮК Сергій Микитович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові.)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 03 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка структури програмного забезпечення. 3. Основні рішення щодо реалізації компонентів системи.

4. Керівництво програміста. 5. Керівництво оператора.
Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	СЕРДЮК С.М., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання 15 квітня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2 тиждень	Розділ 1
3	Розробка архітектури програми.	3 тиждень	Розділ 2
4	Розробка програми.	4–5 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми.	6 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	7 тиждень	Розділ 5, Додатки
7	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Нікіта ДІДЕНКО

(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Сергій СЕРДЮК

(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
72 с., 4 табл., 30 рис., 2 дод., 13 джерел.

ВЕБКАМЕРА, ВІДЕО, ЗАПИС, ЗНІМКИ, VISUAL STUDIO 2019, C#,
ФОРМАТ, WINDOWS.

Об'єктом дослідження є процеси отримання та переробки інформації з вебкамери.

Предметом дослідження є методи отримання та переробки інформації з вебкамери.

Мета роботи – розробка програми, що забезпечує користувачу змогу отримувати, обробляти та зберігати інформацію з вебкамери.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування; мова програмування C#; середовище розробки Microsoft Visual Studio 2019; персональний комп'ютер з процесором з частотою 1 ГГц під управлінням операційної системи (ОС) Microsoft Windows 10.

Результати. Розроблено програмне забезпечення, яке дозволяє користувачам записувати відео, застосовувати фільтри та створювати скріншоти з вебкамери. Застосунок має інтуїтивно зрозумілий інтерфейс, призначений для користувачів будь-якого рівня кваліфікації.

Галузь використання – системи спостереження та обробки відео.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 72 pages, 4 tables, 30 figures, 2 appendixes, 13 sources.

WEBCAM, VIDEO, RECORDING, PICTURES, VISUAL STUDIO 2019, C#, FORMAT, WINDOWS.

The object of study is the process of obtaining and processing information from a webcam.

The subject of study are methods of obtaining and processing information from a webcam.

The purpose of the work is a development of a program that enables the user to receive, process and save information from a webcam.

Materials, methods and technical means: object-oriented programming; programming languages C#; Microsoft Visual Studio 2019 development environment; a personal computer with a 1 GHz processor running the Microsoft Windows 10 operating system (OS).

Results. Software has been developed that allows users to record videos, apply filters and take screenshots from the webcam. The application has an intuitive interface designed for users of any skill level.

The field of use are surveillance and video processing systems.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Огляд існуючих методів запису відео з вебкамери.....	10
1.2 Аналіз існуючих програмних аналогів.....	10
1.3 Розробка технічного завдання.....	21
1.3.1 Підстава для розробки.....	21
1.3.2 Мета розробки та її призначення	21
1.3.3 Основні вимоги до програмного продукту	21
1.3.4 Вимоги до технічних засобів, операційної системи і мови програмування	22
1.3.5 Вимоги до інформаційної та програмної сумісності	23
1.3.6 Вимоги до програмної документації	23
1.3.7 Порядок приймання і тестування.....	23
2 Розробка структури програмного забезпечення	24
2.1 Аналіз технічного завдання.....	24
2.2 Аналіз функціональності та моделювання сценаріїв дій користувача ..	25
2.3 Огляд особливостей мови програмування та середовища розробки	27
3 Основні рішення щодо реалізації компонентів системи.....	30
3.1 Основні рішення щодо розроблення проєкту застосунку	30
3.1.1 Розробка класу FiltersController	31
3.1.2 Розробка класу CameraController	34
3.1.3 Розробка класу Form1.....	38
3.1.4 Обробка виключних ситуацій	40
4 Керівництво програміста.....	45
4.1 Призначення та умови застосування програми.....	45
4.2 Характеристики програми	45

4.3 Структура програми	45
4.4 Вхідні та вихідні дані	47
4.4.1 Вхідні дані	47
4.4.2 Вихідні дані	48
4.5 Повідомлення програмісту	48
5 Керівництво оператора	49
5.1 Призначення програми	49
5.2 Умови виконання програми	49
5.2.1 Апаратні вимоги програми	49
5.2.2 Вимоги до користувача	49
5.3 Виконання програми	49
5.4 Повідомлення оператору	52
Висновки	53
Перелік джерел посилання	54
Додаток А. Текст програми	56
Додаток Б. Слайди презентації	64

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AForge.Video	– бібліотека C# для роботи з відео;
AForge.DirectShow	– бібліотека C# для роботи з відео;
Accord.Video.FFMPEG	– бібліотека C# для обробки відео;
Accord.Imaging.Filters	– бібліотека C# для роботи з фільтрами;
MVC	– Model-View-Controller;
MVVM	– Model-View-ViewModel;
OBS Studio	– Open Broadcaster Software Studio;
VS	– Visual Studio;
Гб	– гігабайт;
ГГц	– гігагерц;
Мб	– мегабайт;
ООП	– об'єктно-орієнтоване програмування;
ОС	– операційна система;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер;
ТЗ	– технічне завдання.

ВСТУП

У сучасному світі швидкого технологічного прогресу застосування програмного забезпечення для запису та обробки відео стає все більш актуальним. З розвитком соціальних мереж, відеоігор, стрімінгових сервісів та інших віртуальних платформ виникає підвищений попит на інструменти, що дозволяють користувачам записувати та редагувати відео з вебкамери. Це особливо актуально в умовах пандемій, або інших обмежень, коли віддалена комунікація та онлайн-взаємодія стають необхідними.

Зростаюча потреба в ефективних інструментах для роботи з відео з вебкамери створює велику популярність серед програм, що надають такі можливості. Зручність та функціональність цих програм стають важливими факторами для їх успіху серед користувачів будь-якого рівня кваліфікації.

Однак, наразі існуючі програмні засоби для запису відео можуть бути складні у використанні, не надійні або недостатньо функціональні. Також, багато з них не мають інтуїтивно зрозумілого інтерфейсу, що може ускладнювати роботу з ними для користувачів будь-якого рівня кваліфікації.

Тому розробка програмного забезпечення для запису та обробки відео з вебкамери, яке має простий та зрозумілий інтерфейс, є актуальною завданням.

Метою даної дипломної кваліфікаційної роботи бакалавра є створення програмного продукту, який задовольнятиме потреби користувачів у зручному та функціональному інструменті для роботи з відео з вебкамери.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд існуючих методів запису відео з вебкамери

У нашій цифровій епосі існує безліч методів запису відео з вебкамери, кожен з яких має свої особливості та переваги. Одним із найбільш поширених методів є використання спеціалізованого програмного забезпечення, такого як "Logitech Capture", "ManyCam", "OBS Studio" та "CyberLink YouCam". Це програмне забезпечення надає користувачу широкий спектр функцій для запису та обробки відео з вебкамери. Воно зазвичай має інтуїтивно зрозумілий інтерфейс та дозволяє легко керувати процесом запису, вибирати різні налаштування якості та формату відео, а також застосовувати різні ефекти та фільтри. Крім того, ці програми часто підтримують функції стрімінгу відео в реальному часі на популярні платформи, такі як YouTube та Twitch, що робить їх популярним вибором для відеоблогерів, стрімерів та звичайних користувачів вебкамер.

Ще одним поширеним методом є використання вбудованих програм та сервісів ОС, які зазвичай постачаються разом із вебкамерою або ОС. Ці програми зазвичай надають базові функції запису відео та створення знімків. Хоча вони можуть не мати такого рівня функціональності та гнучкості, як спеціалізоване програмне забезпечення, вони часто є достатньо зручними для звичайних користувачів. Такі вбудовані програми можуть включати в себе програми, які надаються разом із операційною системою, або ж додаткові програми, які користувач може завантажити з вебмагазину своєї ОС.

Наприклад, у Windows зазвичай доступні такі вбудовані програми, як Камера або Movie Maker, які надають базові функції запису відео та створення знімків. У macOS користувачі можуть скористатися програмою Photo Booth, яка також дозволяє записувати відео та робити знімки з веб-камери. Вони можуть мати обмежений функціонал, проте вони зазвичай є достатньо простими у використанні та можуть задовольняти потреби більшості користувачів в щоденному використанні вебкамери.

Також існують різноманітні онлайн-сервіси та вебзастосунки, які дозволяють записувати відео з вебкамери безпосередньо у веббраузері, без необхідності встановлення додаткового програмного забезпечення. Сервіси цього типу зазвичай надають простий і зручний спосіб запису відео, проте вони можуть мати обмеження на функціональність та якість запису. Деякі з популярних онлайн-сервісів для запису відео з вебкамери включають YouTube Live, Facebook Live, а також онлайн-платформи для відеоконференцій, такі як Zoom та Skype. Вони забезпечують можливість миттєвого запису та трансляції відео в реальному часі через Інтернет, що робить їх зручними для ведення вебінарів, онлайн-трансляцій та інших подібних заходів.

Кожен з цих методів має свої переваги та недоліки, і вибір конкретного підходу може залежати від потреб користувача, рівня технічних навичок та доступності обладнання. Далі в роботі буде розглянуто практичну реалізацію програмного забезпечення для запису та обробки відео з веб-камери, з урахуванням зазначених методів та їх можливостей.

Далі в роботі буде розглянуто і проаналізовано існуючі аналоги програмного забезпечення для запису відео з вебкамери з метою виявлення їх переваг та недоліків, на основі яких буде розроблено технічне завдання для подальшої розробки власного програмного продукту.

1.2 Аналіз існуючих програмних аналогів

З розвитком технологій та поширенням інтернету зросла потреба в інструментах для запису та обробки відео. Існує безліч програмних рішень, призначених для цих цілей, кожне з яких має свої особливості та переваги. У цьому розділі проведемо аналіз кількох існуючих аналогів для запису відео з метою визначення їх функціональних можливостей, зручності використання та відповідності до потреб користувачів.

Одним із відомих аналогів є Windows Camera [1], [2] – програма для операційної системи Windows, яка призначена для захоплення зображень і

відео з веб-камери. Ось деякі з основних функціональних характеристик цієї програми:

- захоплення фотографій. Windows Camera дозволяє користувачам зробити фотографії за допомогою вбудованої або підключеної веб-камери;
- запис відео. Програма дозволяє записувати відео з веб-камери в різних роздільних здатності та форматах;
- режими зйомки. Windows Camera зазвичай надає різні режими зйомки, такі як одиночний знімок, серійна зйомка та таймер;
- налаштування камери. Користувачі можуть налаштовувати параметри камери, такі як яскравість, контрастність, баланс білого та інші для досягнення оптимальних результатів зйомки;
- застосування ефектів. Можливе застосування різних ефектів та фільтрів до зображень та відео в реальному часі або після зйомки;
- сумісність із пристроями Surface. Windows Camera зазвичай інтегрована з пристроями Surface, що забезпечує оптимальну взаємодію та додаткові функції для користувачів цих пристроїв;
- автоматичне збереження у хмару. Деякі версії Windows Camera можуть автоматично зберігати фотографії та відео у хмарному сховищі, такому як OneDrive, для забезпечення доступу до них з будь-якого пристрою;
- інтеграція з Windows 10: Програма зазвичай повністю інтегрована з операційною системою Windows 10, що забезпечує простоту використання та зручність доступу до нього.

Це основні функціональні характеристики Windows Camera, які зазвичай надаються у програмах для захоплення зображень та відео на пристроях під керуванням Windows.

Як і будь-яке програмне забезпечення, Windows Camera має свої сильні та слабкі сторони. До переваг Windows Camera слід віднести:

- інтеграція з Windows. Windows Camera повністю інтегрована з операційною системою Windows, що робить її легкою у використанні та доступною для користувачів Windows;

- простота використання. Інтерфейс користувача Windows Camera зазвичай інтуїтивно зрозумілий, що робить процес захоплення зображень і відео простим навіть для новачків;
- різноманітність режимів зйомки. Програма надає різні режими зйомки, включаючи одиночне фото, серійну зйомку та відео, що дозволяє користувачам вибрати найбільш підходящий режим для своїх потреб;
- підтримка хмарного сховища. Деякі версії Windows Camera автоматично зберігають знімки та відео в хмарному сховищі, полегшуючи доступ до них з будь-якого пристрою;
- застосування ефектів. Користувачам надається можливість застосовувати різні ефекти та фільтри до зображень та відео, що дозволяє їм створювати унікальні та цікаві знімки.

До недоліків слід віднести:

- обмежений функціонал. Windows Camera зазвичай пропонує обмежений набір функцій у порівнянні з іншими програмами для захоплення зображень та відео;
- відсутність просунутих можливостей. Для просунутих користувачів Windows Camera може здатися недостатньо функціональною, оскільки в ній може вистачати деякі просунуті можливості та налаштування;
- обмежена підтримка форматів. Програма може обмежувати вибір форматів для збереження файлів зображень та відео, що може бути недостатнім для деяких користувачів;
- не завжди стабільна робота. У деяких користувачів можуть виникати проблеми з неправильною роботою програми або її нестабільністю, особливо на старих або менш потужних пристроях;
- обмежена підтримка пристроїв. Не всі пристрої можуть підтримувати всі функції та можливості Windows Camera, що може обмежувати її використання на деяких пристроях.

Це деякі з переваг та недоліків Windows Camera, які варто враховувати при виборі програми для захоплення зображень та відео на пристроях під керуванням Windows.

Наступний аналог, це Camtasia [3], [4] – програмне забезпечення для створення відеоуроків, запису екрана комп'ютера, монтажу та редагування відео, а також створення презентацій. Розглянемо більш детально функціональність Camtasia:

- запис екрана. Camtasia дозволяє записувати всі дії на екрані комп'ютера, включаючи рухи миші, натискання клавіш та аудіо з різних джерел;
- редагування відео. Програма надає широкий набір інструментів для редагування відео, включаючи обрізання, злиття, накладання звукових доріжок, додавання тексту, анімацій та ефектів;
- додавання мультимедіа. Camtasia дозволяє вставляти зображення, аудіофайли та відеофрагменти в проект, що робить його корисним інструментом для створення навчальних матеріалів та презентацій;
- створення анімації та ефектів. Програма надає можливості для додавання різних анімацій, переходів та спецефектів, які можуть надати вашому відео професійного вигляду;
- експорт та публікація. Camtasia дозволяє експортувати готове відео у різні формати та опублікувати його на популярних платформах для відеохостингу, таких як YouTube та Vimeo;
- навчальні інструменти. Програма включає інструменти для створення інтерактивних тестів, питань та інших навчальних матеріалів, що робить її корисною для навчання та дистанційного навчання;
- шаблони та налаштування. Camtasia надає готові шаблони та налаштування, які спрощують процес створення відео і роблять його більш професійним;

- інтеграція з іншими програмами. Програма інтегрується з іншими програмами та сервісами, такими як Google Drive та Dropbox, що спрощує обмін та спільну роботу над проєктами.

Переваги Camtasia:

- простота використання. Camtasia має інтуїтивно зрозумілий інтерфейс користувача, що робить його доступним для новачків в області створення відео;

- широкі можливості редагування. Програма надає широкий набір інструментів для редагування відео, включаючи налаштування звуку, додавання тексту та анімацій, а також накладання спецефектів;

- записування та редагування одночасно. Camtasia дозволяє записувати відео з екрана та одночасно редагувати його, що заощаджує час при створенні контенту;

- експорт у різні формати. Програма підтримує експорт відео до різних форматів, що дозволяє адаптувати контент під різні платформи та пристрої;

- навчальні інструменти. Camtasia включає інструменти для створення інтерактивних навчальних матеріалів, таких як тести та питання, що робить його корисним інструментом для навчання та навчальних курсів.

Недоліки застосування:

- вартість. Camtasia є комерційним продуктом і вимагає покупки ліцензії для повноцінного використання, що може бути недоступним для деяких користувачів;

- вимоги до ресурсів. Запис та редагування відео може бути ресурсомістким процесом, особливо при роботі з великими файлами та складними ефектами;

- обмежені можливості безкоштовної версії. Безкоштовна версія Camtasia має обмежений набір функцій у порівнянні з повною версією, що може бути недостатньо для деяких користувачів;

- необхідність навчання. Хоча Camtasia має інтуїтивно зрозумілий інтерфейс, для максимального використання всіх можливостей може знадобитися час і навчання.

Ще одним потужним програмним забезпеченням для запису та трансляції відео є OBS Studio (Open Broadcaster Software Studio) [5], [6]. Основні функціональні характеристики OBS Studio наступні:

- записування та трансляція відео. OBS Studio дозволяє записувати відео з екрана комп'ютера, камери та інших джерел, а також стримувати відео на популярні платформи для стримінгу, такі як Twitch, YouTube та ін.;

- сагатоканальний запис. Програма підтримує запис кількох джерел одночасно, що дозволяє створювати складні мультимедійні проекти з різними шарами відео та аудіо;

- інтерфейс з використанням джерел. OBS Studio має інтуїтивно зрозумілий інтерфейс, який дозволяє легко додавати, керувати та налаштовувати різні джерела відео, такі як екрани, зображення, відеофайли, текстові написи та ін.;

- налаштування якості та кодеків. Користувач може налаштувати дозвіл, бітрейт та інші параметри якості відеозапису та стриму відповідно до вимог платформи або власних уподобань;

- аудіофункції. Можливості роботи з аудіо включають налаштування гучності, додавання звукових ефектів, підтримку мікрофонів і аудіоінтерфейсів;

- плагіни та розширення. OBS Studio підтримує широкий вибір плагінів та розширень, які додають додаткові функції та можливості, такі як налаштування відеоефектів, керування сценаріями та інші;

- розширені налаштування трансляції. Користувач може налаштувати різні параметри трансляції, включаючи бітрейт, кодування, формати відео та аудіо, а також підтримку трансляції з роздільною здатністю до 4K.;

- підтримка платформ для стрімінгу. OBS Studio інтегрується з популярними платформами для стрімінгу, такими як Twitch, YouTube, Facebook Live та інші, що полегшує процес трансляції та взаємодії з аудиторією.

До переваг OBS Studio слід віднести:

- безкоштовність. OBS Studio доступний безкоштовно для використання без обмежень, що робить його доступним для широкого кола користувачів;

- багатоканальний запис. Програма дозволяє записувати та транслювати кілька джерел одночасно, що особливо корисно для створення мультимедійних проектів з різними шарами відео та аудіо;

- гнучкість налаштування. OBS Studio надає великі можливості налаштування якості відео та аудіо, що дозволяє користувачам адаптувати програму під свої потреби та вимоги.

- широкий вибір плагінів. Програма підтримує безліч плагінів та розширень, які додають додаткові функції та можливості, такі як налаштування відеоефектів, керування сценаріями та інші.

- підтримка платформ для стрімінгу. OBS Studio інтегрується з популярними платформами для стрімінгу, такими як Twitch, YouTube, Facebook Live та інші, що полегшує процес трансляції та взаємодії з аудиторією.

До недоліків OBS Studio слід віднести:

- складність для новачків. Для нових користувачів OBS Studio може здатися складним в освоєнні через велику кількість функцій і налаштувань;

- вимоги до ресурсів. Програма може споживати багато ресурсів комп'ютера, особливо під час запису або трансляції у високій якості, що може викликати проблеми на повільних або старих комп'ютерах;

- необхідність налаштування. Для досягнення оптимальної якості відео та аудіо потрібно провести ретельне налаштування програми, що може вимагати часу та технічних навичок;

- відсутність вбудованої підтримки для деяких функцій. Деякі нові функції можуть вимагати встановлення додаткових плагінів або розширень, що може ускладнити процес використання для користувачів.

З метою порівняння аналогів, для урахування важливості для досвіду користувача і функціональності програми запису відео, введемо наступні критерії порівняння:

- інтуїтивний інтерфейс. Цей критерій був обраний через його важливість для досвіду користувача. Інтуїтивний інтерфейс забезпечує більш гладку та ефективну взаємодію користувача з програмою, що підвищує його задоволення від використання та знижує час, витрачений на навчання;

- розширені налаштування. Враховуючи різноманітність ситуацій, в яких може знадобитися відеозапис, важливо мати можливість налаштувати параметри запису під конкретні потреби. Цей критерій дозволяє оцінити гнучкість та функціональність програми;

- підтримка стрімінгу. Стрімінг відео стає все більш затребуваним способом обміну контентом. Програми, що підтримують цю функцію, можуть бути привабливішими для користувачів, які хочуть ділитися своїм відео безпосередньо в реальному часі;

- можливості керування аудіо. Звук відіграє важливу роль у відеозаписах, тому наявність інструментів для керування аудіо параметрами важлива для створення високоякісного контенту;

- безкоштовність. Для багатьох користувачів безкоштовні програми мають велику цінність, особливо якщо вони мають гарні функціональні можливості. Цей критерій відбиває інтереси широкого кола користувачів;

- підтримка фільтрів. Фільтри дозволяють користувачам покращувати та кастомізувати свої відеоролики, роблячи їх більш привабливими та оригінальними. Це важливий аспект створення якісного контенту;

– можливість робити знімки з камери. Захоплення зображень з камери розширює функціональність програми та надає користувачам додаткові можливості для творчості та самовираження;

– перегляд відео після запису. Перегляд записаного відео важливий для оцінки якості та можливості внесення необхідних коригувань перед його публікацією. Цей критерій забезпечує зручність та ефективність процесу створення відеоконтенту.

У межах даної кваліфікаційної роботи бакалавра було створено програмне забезпечення під назвою Web Recorder. У табл.1 наведено його порівняння з вищезгаданими аналогами.

Таблиця 1.1 – Аналіз програм-аналогів

Програми Критерії	Web Recorder	Camtasia	Windows Camera	OBS Studio
1	2	3	4	5
Безкоштовність	Так	Ні	Так	Ні
Знімки камери ³	Так	Так	Так	Ні
Інтуїтивний інтерфейс	Так	Ні	Так	Ні
Розширені налаштування	Ні	Так	Так	Так
Підтримка стрімінгу	Ні	Так	Ні	Так
Можливості керування аудіо	Ні	Так	Ні	Так

Кінець таблиці 1.1

1	2	3	4	5
Підтримка фільтрів	Так	Так	Ні	Так
Перегляд відео після запису	Ні	Так	Так	Ні

За результатами аналізу, розглянемо шляхи вирішення недоліків програм запису відео. Для поліпшення розуміння інтерфейсу можна розглянути додаткові навчальні матеріали, відео-уроки та спрощення структури інтерфейсу, або зробити додаткові підказки для користувача.

Додавання додаткових параметрів налаштувань може значно розширити функціонал програми, надаючи користувачам більше можливостей. При цьому, важливо також забезпечити базові налаштування, такі як можливість визначення шляху для збереження файлів, регулювання кількості кадрів для запису та вказання розширення відео. У випадку програми Web Recorder, що розроблюється в даній роботі, можливості розширення для запису відео визначаються можливостями відеопристрою.

Розширення функціоналу для керування аудіо може виправити недоліки та забезпечити користувачам більший контроль над аудіопроцесами. Важливо врахувати базові налаштування, такі як можливість вибору аудіо-пристрою, регулювання гучності та вибір аудіоформату для запису. У випадку програми Web Recorder, можливості керування аудіо не передбачається.

На базі проведеного аналізу, необхідно розробити ТЗ на проєктування власного програмного продукту.

1.3 Розробка технічного завдання

1.3.1 Підстава для розробки

Підставою для розробки служить завдання на дипломну кваліфікаційну роботу бакалавра на тему "Розробка програмного забезпечення для запису та обробки відео з вебкамери", затверджене наказом № 168 від 24 квітня 2024 р. по НУ «Запорізька політехніка».

1.3.2 Мета розробки та її призначення

Метою є розробка програмного застосунку, що призначений для запису відео з вебкамери користувача та здійснення відеоспостереження.

1.3.3 Основні вимоги до програмного продукту

Програмне забезпечення повинно мати один тип користувачів – власника відеосистеми. Користувач має володіти наступними навичками: вміння користуватися персональним комп'ютером та вміння підключати пристрій до комп'ютера. Програмний продукт має надавати користувачу наступні можливості:

- зберігати записи з вебкамери;
- робити знімки з вебкамери;
- записувати інформацію з вебкамери;
- зберігати знімки з вебкамери;
- обирати вебкамеру для здійснення записувідео;
- вказувати шлях для зберігання файлів;
- обирати формат для збереження відео;
- корегувати зображення (налаштовувати відтінки сірого, змінювати яскравість, створювати ефекти інвертності кольорів та сепії).

Вимоги до інтерфейсу:

- інтуїтивно зрозумілий інтерфейс для зручності користування користувачем;

- наявність зручних елементів керування для вибору вебкамери, налаштування параметрів запису та збереження файлів.

Вхідними даними в програмі є інформація з відеокамери, або дані, що буде вводити користувач.

Основною вимогою до вихідних даних є забезпечення повідомлень для користувача про успішне або невдале збереження скріншоту або відео.

Вихідними даними в програмі є збережена інформація та результати виконання керуючих команд користувача.

При роботі програми можуть виникати збійні ситуації, які мають бути програмно оброблюватися:

- запис в неіснуючий каталог;
- відсутність відео-камери (програма не може знайти камеру).

1.3.4 Вимоги до технічних засобів, операційної системи і мови програмування

Програма, має розробляється у вигляді веб-застосунку. Застосунок має бути написана на мові C# у середовищі розробки Microsoft Visual Studio 2019. Програма призначена для роботи на персональному комп'ютері під керуванням ОС Windows 10.

Мінімальні системні вимоги обумовлюються забезпеченням нормальної роботи ОС Windows 10 і мають бути наступними [7]:

- процесор із тактовою частотою 1 ГГц або швидший чи SoC;
- 1 гігабайт оперативної пам'яті (для 32-розрядної версії) або 2 ГБ (для 64-розрядної версії);
- обсяг вільного дискового простору 16 ГБ (для 32-розрядної версії) або 20 ГБ (для 64-розрядної версії);

- графічна плата із підтримкою DirectX 9 або новішої версії з драйвером WDDM 1.0 або новішим;
- дисплей: 800 x 600.

Також, щоб забезпечити функціонування програми ФМ необхідні наступні засоби:

- 500 Мб вільного місця на жорсткому диску для встановлення програми;
- маніпулятори вводу (клавіатура, миша);
- вебкамера.

1.3.5 Вимоги до інформаційної та програмної сумісності

Розроблені вихідні коди повинні бути написані мовою C#. Середовище розробки програмного продукту Microsoft Visual Studio 2019.

1.3.6 Вимоги до програмної документації

Програмна документація має включати: технічне завдання, керівництво програміста, керівництво оператора, опис програми та тексти основних програмних модулів.

1.3.7 Порядок приймання і тестування

Для налагодження та тестування програми необхідно використовувати можливості Visual Studio, щодо виявлення та усунення помилок в процесі розробки. Тестування даного застосунку має проводитися у вигляді виконання усіх можливих сценаріїв роботи оператора з вебкамерою та програмним продуктом.

2 РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз технічного завдання

Для реалізації функціональності запису відео з веб-камери та відеоспостереження можуть використовуватися різні технології. Це може бути веб-додаток з використанням JavaScript та бібліотеки для доступу до веб-камери через браузер, або додаток на основі стандартних бібліотек мови програмування (наприклад, Python із OpenCV). Шаблони проєктування, такі як MVC (Model-View-Controller) [8] або MVVM (Model-View-ViewModel) [9] можуть використовуватися для організації структури програми.

У даному випадку, технічне завдання спрямоване на створення програмного застосунку для запису відео з веб-камери мовою програмування C# з використанням середовища розробки Visual Studio.

Реалізація програми на C# передбачає використання .NET Framework або .NET Core. Для доступу до функціональності веб-камери та обробки відео можна скористатися бібліотеками, такими як AForge.NET Framework, OpenCVSharp або Windows Media Foundation. Ці бібліотеки надають широкий набір функцій для роботи з відео та зображеннями, включаючи захоплення відео з веб-камери, збереження відео та накладення ефектів на зображення. Як шаблони проєктування також можна використовувати MVC або MVVM. Згідно ТЗ середовищем розробки має бути Visual Studio 2019. Це зручне середовище розробки для роботи з .NET Framework та C#, що спрощує процес створення та налагодження програми.

Для створення інтерфейсу користувача в Visual Studio можна використовувати Windows Forms або WPF (Windows Presentation Foundation). Вони надають зручні інструменти для розробки інтерфейсу, а також можливості для створення інтуїтивно зрозумілого і привабливого досвіду користувача.

Одним із вузьких місць може бути забезпечення сумісності з різними операційними системами та браузерами, оскільки доступ до веб-камери та

файлової системи може різнитися залежно від платформи. Труднощі можуть представляти також обробка та зберігання великих обсягів відеоданих, а також забезпечення стабільної роботи програми в реальному часі.

На платформі C# одним із вузьких місць може бути забезпечення сумісності з різними версіями .NET Framework та операційними системами Windows. Також може знадобитися додаткова робота з обробки винятків та управління ресурсами, такими як пам'ять та процесорний час, особливо при роботі з великими обсягами відео.

Для забезпечення сумісності з різними версіями .NET Framework та операційними системами можна використовувати механізми умовної компіляції та перевірки сумісності API. Для ефективної роботи з відеоданими можна застосовувати асинхронне програмування та механізми багатопоточності для оптимізації продуктивності та управління ресурсами. Для керування великими обсягами даних можна застосовувати компресію відео та оптимізацію зберігання файлів. Для обробки помилок та виключень можна використовувати механізми обробки виключень та тестування програмного коду.

Таким чином, в процесі роботи над кваліфікаційною роботою бакалавра необхідно ретельно вивчити документацію з обраних технологій та бібліотек, а також провести тестування програми на різних платформах та сценаріях використання, щоб виявити та усунути можливі проблеми. Також важливо передбачити механізми зворотного зв'язку та підтримки для користувачів програми. В ТЗ вже передбачена реалізація реакції системи на помилки та вивід відповідних повідомлень користувачу.

2.2 Аналіз функціональності та моделювання сценаріїв дій користувача

Основною функціональною частиною програми є головне вікно, в якому користувач може здійснити вибір веб-камери, визначити шлях для

збереження відео, застосовувати фільтри, створити скріншот та почати або закінчити запис відео. Кнопка закінчити запис не буде доступна до тих пір, поки користувач не почне запис. Use-case діаграма можливих дій користувача представлена на рис. 2.1.

В програмі існує сім варіантів подій для користувача. Початок запису та скріншот пов'язані зв'язком <<include>> з перевіркою шляху збереження, тому що кожного разу коли користувач буде починати запис чи робити скріншот буде відбуватись перевірка шляху збереження. Помилка збереження пов'язана зв'язком <<extends>>, тому що помилка виводиться не кожного разу.

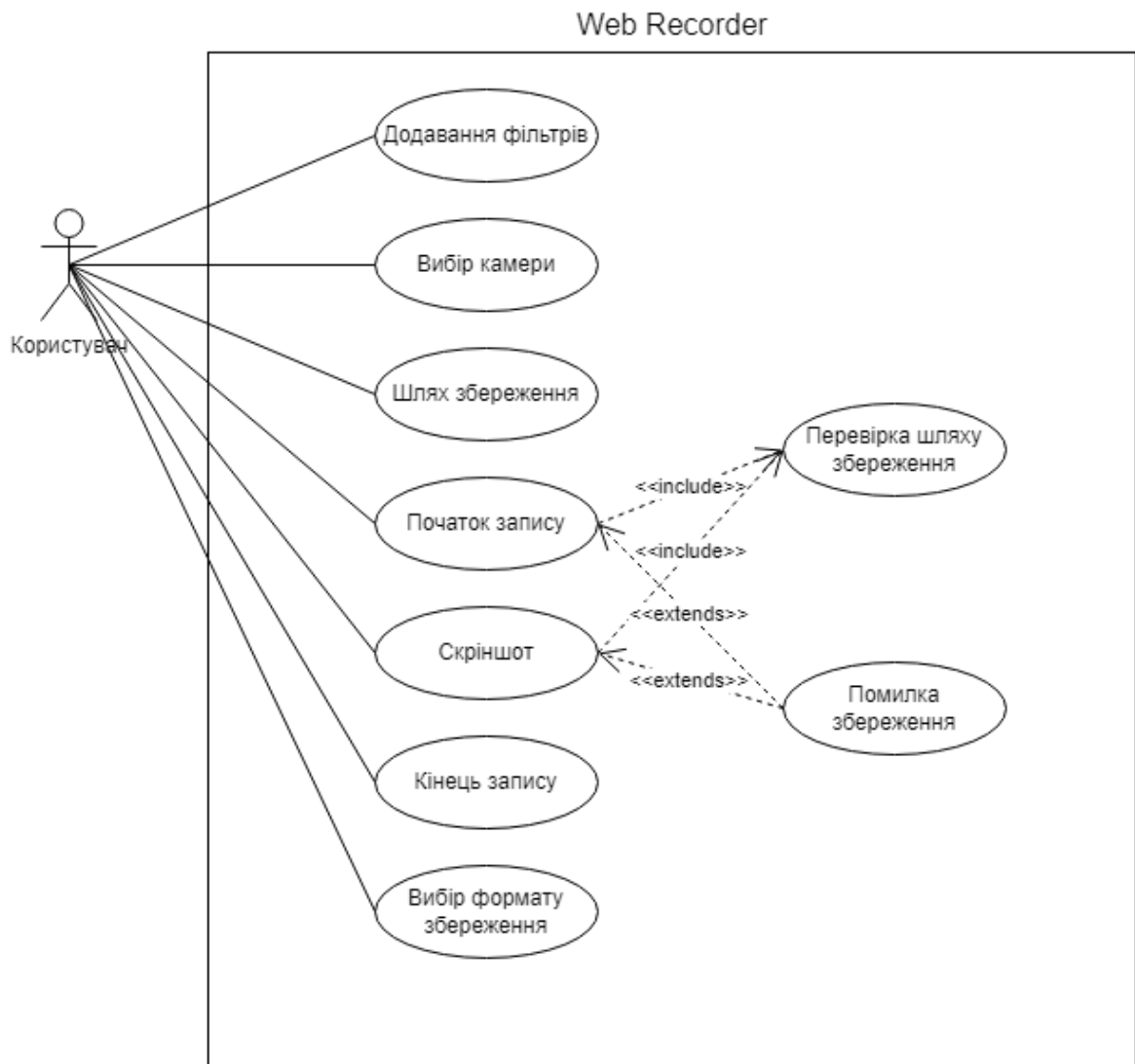


Рисунок 2.1 – Use-case діаграма.

2.3 Огляд особливостей мови програмування та середовища розробки

Вибір мови програмування для розробки програмного забезпечення має ключове значення, оскільки він впливає на ефективність розробки, продуктивність програми, доступність інструментів та ресурсів, а також майбутню підтримку та розширення проекту. У контексті цієї кваліфікаційної роботи бакалавра була обрана мова програмування C# з використанням середовища розробки Visual Studio. Цей вибір мови обґрунтовується наступними причинами:

- простота вивчення та використання. Мова програмування C# відома своєю простотою і зрозумілим синтаксисом, що робить його ідеальним вибором для студентів та розробників-початківців. Це також полегшує розуміння коду та його підтримку у майбутньому;
- широка підтримка та інтеграція. C# є однією з основних мов, підтримуваних платформою .NET Framework, що забезпечує доступ до багатьох бібліотек та інструментів для розробки різноманітних застосунків. Завдяки інтеграції з Visual Studio, розробнику надається зручне середовище для створення, налагодження та розгортання програмного забезпечення;
- багаті можливості для роботи з мультимедіа. C# має потужні інструменти та бібліотеки для обробки та роботи з відео та аудіо даними. Це дозволяє легко реалізувати функціональні вимоги до даного проєкту, пов'язані із записом та обробкою відео з вебкамери. Наприклад, у .NET Framework та .NET Core є бібліотеки, такі як System.Drawing, System.Media, System.Windows.Media, які надають безліч класів та методів для роботи із зображеннями, звуками та відео. Для більш низькорівневої роботи з мультимедійними даними C# доступні DirectX і Media Foundation. Ці технології надають потужні інструменти для захоплення, обробки та відтворення відео та аудіо. Крім того, у спільноті розробників C# існує безліч сторонніх бібліотек та фреймворків, що спеціалізуються на обробці

мультимедійних даних. Наприклад, AForge.NET [10], Accord.NET [11], Emgu CV [12] – це популярні бібліотеки для комп'ютерного зору та обробки зображень у C#;

- підтримка сучасних технологій та платформ. C# активно підтримується Microsoft та постійно оновлюється, що дозволяє використовувати сучасні технології та підходи у розробці. Крім того, C# дозволяє створювати кросплатформні програми, що може бути важливим для майбутнього розширення проєкту на інші ОС;

- широке співтовариство та ресурси. C# має велику спільноту розробників, готових надати поради, підтримку та вирішення проблем. Завдяки цьому, доволі легко можна знаходити відповіді на запитання та рішення для завдань.

Виходячи з вищезазначеного, вибір мови програмування C# для розробки програмного забезпечення для запису та обробки відео з вебкамери обґрунтований його простотою використання, доступністю ресурсів та інструментів, а також його здатністю ефективно реалізовувати вимоги проєкту. У поєднанні із середовищем розробки Visual Studio, C# надає нам всі необхідні інструменти для успішної реалізації поставлених завдань.

Вибір середовища розробки має вирішальне значення для успішного виконання даної роботи. Згідно з ТЗ, для створення програми було обрано середовище розробки Visual Studio 2019 [13]. Цей вибір обґрунтований такими причинами:

- інтеграція з мовою програмування C#. Visual Studio надає повну інтеграцію з мовою програмування C#, що забезпечує безперешкодну розробку, налагодження та тестування коду. Це значно підвищує продуктивність розробника та прискорює процес створення програмного забезпечення;

- потужні інструменти для розробки. Visual Studio пропонує широкий набір інструментів для розробки, включаючи редактор коду з підсвічуванням синтаксису, автодоповнення, налагоджувач, систему

керування версіями, інтегровані тести та багато іншого. Це дозволяє ефективно управляти проєктом та розробляти високоякісне програмне забезпечення;

- кросплатформова розробка. Visual Studio підтримує кросплатформну розробку, що дозволяє створювати програми, які можуть працювати на різних операційних системах, включаючи Windows, MacOS і Linux. Це розширює аудиторію та потенційне застосування програмного забезпечення, що розробляється;

- зручність використання. Visual Studio має інтуїтивний і зручний інтерфейс, що робить процес розробки приємним і ефективним. Інтегроване середовище розробки пропонує численні функції та інструменти, що спрощують рутинні завдання та покращують продуктивність;

- підтримка спільноти. Visual Studio має величезну спільноту розробників, які готові поділитися досвідом, надати поради та вирішення проблем. Це забезпечує доступ до широкого спектру ресурсів та оновлень, що сприяє успішній реалізації проєкту.

Таким чином, вибір середовища розробки Visual Studio 2019 для нашого проєкту обґрунтований його повною інтеграцією з мовою програмування C#, потужними інструментами розробки, підтримкою кросплатформи, зручністю використання та підтримкою спільноти розробників. Visual Studio надає всі необхідні інструменти та ресурси для успішної реалізації програмного забезпечення та забезпечує комфортне та продуктивне робоче оточення для розробників.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

Цей розділ присвячений опису ключових аспектів розробки програмного забезпечення для запису та обробки відео з використанням вебкамери. У цьому розділі будуть розглянуті основні рішення, що базуються на обраних для реалізації різних компонентів системи технології. Метою даного розділу є представлення докладної інформації про класи, методи, бібліотеки та інструменти, використані в процесі розробки, а також пояснення прийнятих рішень та їх ролі у забезпеченні функціональності та продуктивності системи.

3.1 Основні рішення щодо розроблення проєкту застосунку

Під час розробки програми була використана об'єктно-орієнтована парадигма програмування. Це дозволило ефективно організувати код за допомогою класів та об'єктів, що представляють різні складові системи. Кожен клас відповідає за конкретний функціонал і має власні методи та властивості, що дозволяє досягти високої модульності і підтримувати код у чистому та структурованому вигляді. Такий підхід сприяє полегшенню розширення та підтримки програми в майбутньому.

Було розроблено наступні класи:

- `FiltersController` – клас, що відповідає за накладання фільтрів на зображення. Він містить методи для застосування різних типів фільтрів та управління їх параметрами;

- `CameraController` – клас, відповідальний за взаємодію з камерою. Включає в себе всі необхідні методи та властивості для керування камерою, а також обробку отриманих зображень;

– Form1 – клас, що відповідає за інтерфейс користувача програми. У ньому реалізована логіка взаємодії з користувачем, включаючи елементи керування та відображення зображень.

Клас Form1 використовує CameraController для керування взаємодією з камерою та отримання зображень. При цьому CameraController використовує об'єкт класу FiltersController для накладання фільтрів на отримані зображення. Така структура програми є прикладом композиції класів, де один клас включає в себе об'єкти інших класів для досягнення певного функціоналу.

Діаграма класів зображена на рис. 3.1, який демонструє взаємозв'язок між класами та їхні ролі у програмі.

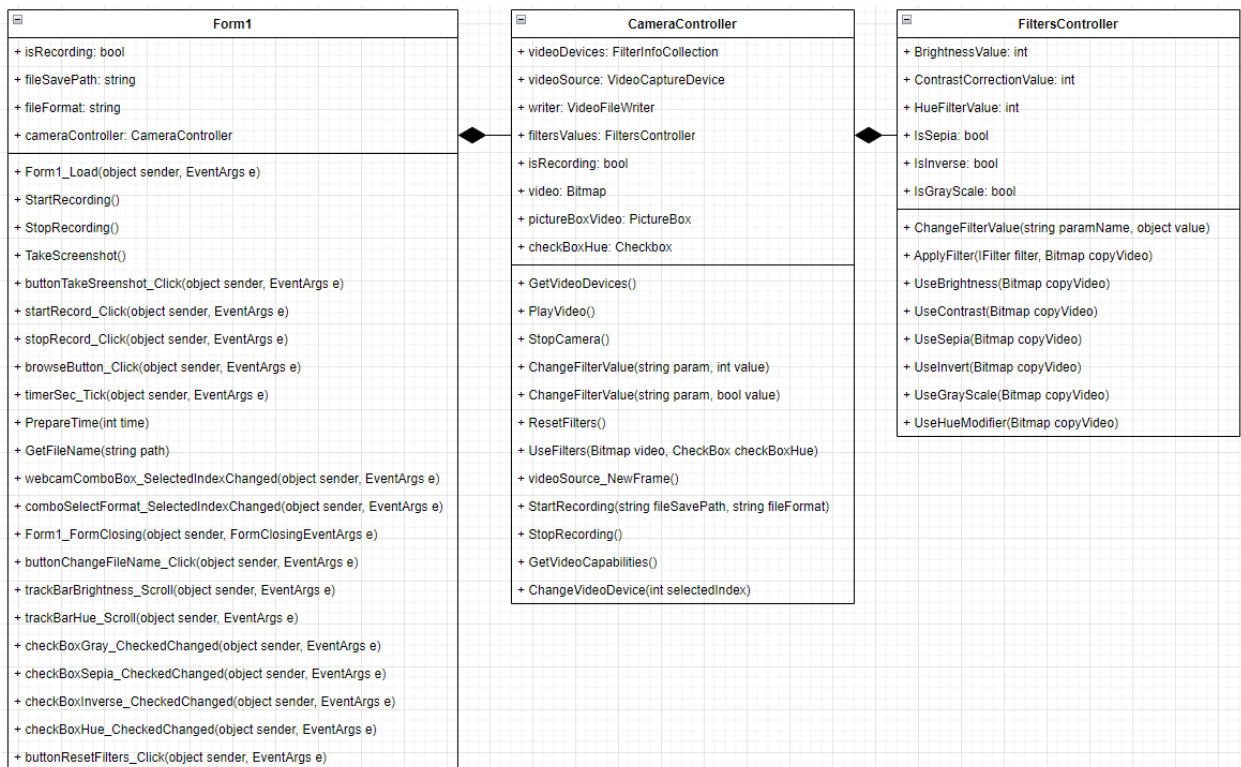


Рисунок 3.1 – Діаграма класів

3.1.1 Розробка класу FiltersController

Клас FiltersController було створено для додавання фільтрів.

Основні поля та методи класу FiltersController наведені у табл. 3.1.

Таблиця 3.1 – Основні поля та методи класу FiltersController

Поля та методи класу	Призначення
1	2
public:	
int BrightnessValue { get; set; }	Поле, яке зберігає значення яскравості для фільтрації
int ContrastCorrectionValue { get; set; }	Поле, яке зберігає значення корекції контрасту для фільтрації
int HueFilterValue { get; set; }	Поле, яке зберігає значення відтінку для фільтрації
bool IsSepia { get; set; }	Поле, яке вказує, чи увімкнений фільтр сепії
bool IsInverse { get; set; }	Поле, яке вказує, чи увімкнений фільтр інверсії кольорів
bool IsGrayScale { get; set; }	Поле, яке вказує, чи увімкнений фільтр відтінків сірого
void ChangeFilterValue(string paramName, object value)	Метод, який змінює значення параметра фільтра за його ім'ям
Bitmap ApplyFilter(IFilter filter, Bitmap copyVideo)	Метод, який застосовує заданий фільтр до відео потоку
Bitmap UseBrightness(Bitmap copyVideo)	Метод, який застосовує фільтр яскравості до відео потоку
Bitmap UseContrast(Bitmap copyVideo)	Метод, який застосовує фільтр корекції контрасту до відеопотоку
Bitmap UseSepia(Bitmap copyVideo)	Метод, який застосовує фільтр сепії до відеопотоку

Кінець таблиці 3.1.

1	2
Bitmap UseInvert(Bitmap copyVideo)	Метод, який застосовує фільтр інверсії кольорів до відеопотоку
Bitmap UseGrayScale(Bitmap copyVideo)	Метод, який застосовує фільтр відтінків сірого до відеопотоку
Bitmap UseHueModifier(Bitmap copyVideo)	Метод, який застосовує фільтр модифікації відтінку до відеопотоку

Приклад коду методу `ChangeFilterValue`, що належить класу `FiltersController`, наведено на рис. 3.2.

```
public void ChangeFilterValue(string paramName, object value)
{
    PropertyInfo propertyInfo = GetType().GetProperty(paramName);

    if (propertyInfo != null)
    {
        if (propertyInfo.PropertyType == typeof(int))
        {
            propertyInfo.SetValue(this, Convert.ToInt32(value));
        }
        else if (propertyInfo.PropertyType == typeof(bool))
        {
            propertyInfo.SetValue(this, Convert.ToBoolean(value));
        }
    }
}
```

Рисунок 3.2 – Код методу `ChangeFilterValue`

Метод `ChangeFilterValue` в класі `FiltersController` дозволяє динамічно змінювати значення фільтрів на основі отриманих параметрів. Цей метод є ключовим для динамічної зміни параметрів фільтрів у реальному часі без необхідності вручну втручатися в код. Завдяки використанню механізму рефлексії, він автоматично визначає властивість фільтра за її іменем, а потім

перевіряє тип значення та, в залежності від типу (int або bool), встановлює нове значення.

Одним із важливих переваг цього метода є його гнучкість та універсальність. Він дозволяє маніпулювати різними типами фільтрів, незалежно від їх кількості та конфігурації, і здійснювати це безпосередньо з рівня контролера фільтрів. Такий підхід спрощує розробку та підтримку коду, оскільки не потребує написання окремих методів для кожного фільтра.

Крім того, метод `ChangeFilterValue` робить можливим динамічне адаптування параметрів фільтрів до змінюючихся потреб користувача або умов експлуатації програми. Це забезпечує зручність та гнучкість в конфігурації фільтрів, що дозволяє досягти оптимальних результатів в обробці відео з вебкамери.

3.1.2 Розробка класу `CameraController`

Клас `CameraController` – це ключовий компонент системи, відповідальний за управління відеокамерою та реалізацію основної функціональності, пов'язаної з записом відео та застосуванням фільтрів до відеопотоку. Цей клас забезпечує інтерфейс для взаємодії з відеокамерою, включаючи запуск запису відео, зупинку запису та налаштування параметрів камери. Крім того, він відповідає за завантаження та застосування фільтрів до відеопотоку в реальному часі.

Основні функції `CameraController` реалізовані за допомогою використання бібліотек `AForge.NET` та `Accord.NET`. Бібліотека `AForge.NET` надає різноманітні інструменти для роботи з відео, включаючи можливості запису, відтворення, захоплення та обробки відеопотоку. Завдяки бібліотеці `Accord.NET` користувач може зручно застосовувати різноманітні фільтри до зображень та відео, що дозволяє йому створювати цікаві ефекти та оптимізувати обробку відеоданих.

Поля та методи класу CameraController розглянуто та задокументовано у табл. 3.2.

Таблиця 3.2 – Опис методів та полів класу CameraController

Поля та методи класу	Опис
1	2
public:	
CameraController(PictureBox pictureBoxVideoLink, CheckBox checkBoxHueLink)	Конструктор, який приймає PictureBox та CheckBox для взаємодії з елементами інтерфейсу
void GetVideoDevices(ComboBox comboBox)	Метод, який заповнює ComboBox інформацією про доступні відеопристрої
void PlayVideo()	Метод, який налаштовує обробник подій NewFrame для відеопристрою та починає відтворення відео
void StopCamera()	Метод, який зупиняє відтворення відео
void ChangeFilterValue(string param, int value)	Метод, який додає фільтр, який використовує число
void ChangeFilterValue(string param, bool value)	Метод, який додає фільтр, який використовує значення істини
void ResetFilters()	Метод, який збрасує всі фільтри
void StartRecording(string fileSavePath, string fileFormat)	Метод, який розпочинає запис відео
void StopRecording()	Метод, який зупиняє запис відео, та розпочинає звичайне відтворення відео
void ChangeVideoDevice(int selectedIndex)	Метод, який змінює вибраний відеопристрій та знов розпочинає відтворення відео
private:	

Кінець таблиці 3.2

1	2
FilterInfoCollection videoDevices	Об'єкт для зберігання інформації про доступні відеоустрої.
VideoCaptureDevice videoSource	Об'єкт для представлення відеопристрою та отримання відеопотоку.
VideoFileWriter writer	Об'єкт для запису відео у відеофайл.
FiltersController filtersValues	Об'єкт класу FiltersController, який містить значення параметрів фільтрів для обробки відеопотоку.
bool isRecording	Поле, в якому зберігається стан запису (йде запис чи ні)
Bitmap video	Змінна для зберігання поточного кадру відеопотоку у форматі Bitmap.
PictureBox pictureBoxVideo	Поле, яке зберігає в собі посилання на pictureBox
CheckBox checkBoxHue	Поле, яке зберігає в собі посилання на checkBox
Bitmap UseFilters(Bitmap video, CheckBox checkBoxHue)	Метод, який застосовує фільтри з об'єкта FiltersController до відеопотоку.
void videoSource_NewFrame(object sender, NewFrameEventArgs eventArgs)	Метод, який викликається при кожному новому кадрі відеопотоку. Він отримує об'єкт NewFrameEventArgs, який містить кадр відео та застосовуються фільтри, якщо вони увімкнені
VideoCapabilities GetVideoCapabilities()	Метод, який отримує інформацію о максимальній роздільній зданості відеопристрія

Приклад коду методу StartRecording, що належить класу CameraController, наведено на рис. 3.3.

```

private void StartRecording()
{
    if (fileSavePath == null)
    {
        MessageBox.Show("Вкажіть шлях для збереження відео");
        return;
    };

    cameraController.StartRecording(fileSavePath, fileFormat);

    startRecord.Enabled = false;
    tabControl1.Enabled = false;
    stopRecord.Enabled = true;

    isRecording = true;
    timerSec.Enabled = true;
}

```

Рисунок 3.3 – Код методу StartRecording

Метод StartRecording виконує ключову функцію у класі, контролюючи процес розпочатку запису відео. При натисканні користувачем кнопки «Почати запис» він перевіряє, чи вказаний шлях для збереження відео, та, у разі відсутності такого шляху, виводить повідомлення про необхідність вказати його. Після цього метод викликає внутрішній метод cameraController.StartRecording(), який розпочинає процес запису відео за вказаним шляхом та форматом файлу.

Далі, метод StartRecording відключає кнопку «Почати запис» та вкладки у вікні програми, щоб уникнути непередбачених дій користувача під час запису. Замість цього, він активує кнопку «Зупинити запис», що дозволяє користувачеві припинити запис у будь-який момент. Крім того, змінна isRecording встановлюється в значення true, що дозволяє системі відстежувати, що зараз проводиться запис відео.

Найважливішою функцією цього метода є його підписка на подію videoSource_NewFrame відео-джерела. При кожному отриманні нового кадру відеопотоку, ця подія викликає метод videoSource_NewFrame, який відповідає

за збереження кадрів у відеофайлі. Таким чином, метод StartRecording гарантує, що кожен новий кадр буде записаний у файл і відеопотік буде правильно оброблений та збережений.

3.1.3 Розробка класу Form1

Для розробки інтерфейсу було використано Windows Forms, яка є стандартною бібліотекою інтерфейсу користувача в середовищі програмування .NET. Клас Form1 був створений з метою забезпечення взаємодії користувача з програмою. У нього включені різноманітні елементи управління, такі як кнопки, тексти, списки тощо, які дозволяють користувачу взаємодіяти з програмою. Крім того, в класі Form1 реалізовано логіку взаємодії із іншими класами програми, такими як CameraController та FiltersController, для керування роботою камери та застосування фільтрів до відеопотоку від камери. Клас Form1 є центральним елементом користувацького інтерфейсу програми, який забезпечує зручну та ефективну взаємодію користувача з функціоналом програми. Поля та методи класу наведено у табл.3.3.

Таблиця 3.3 – Поля та методи класу Form1

Поля та методи класу	Опис
1	2
public:	
Form()	Конструктор, ініціалізується об'єкт cameraController, comboBox заповнюється інформацією про доступні відеопристрої
private:	
void Form1_Load(object sender, EventArgs e)	Метод, який працює при завантаженні форми, відео з камери транслюється у pictureBox

Продовження таблиці 3.3

1	2
void StartRecording()	Метод, який починає запис
void StopRecording()	Метод, який завершує запис
void TakeScreenshot()	Метод, який робить скріншот
void buttonTakeScreenshot_Click(object sender, EventArgs e)	Кнопка, яка викликає метод TakeScreenshot
void startRecord_Click(object sender, EventArgs e)	Кнопка, яка починає запис
void stopRecord_Click(object sender, EventArgs e)	Кнопка, яка зупиняє запис
void browseButton_Click(object sender, EventArgs e)	Кнопка, для вибору шляху для збереження
void webcamComboBox_SelectedIndexChanged(object sender, EventArgs e)	Метод, який відпрацює при зміні відео девайсу
void comboSelectFormat_SelectedIndexChanged(object sender, EventArgs e)	Метод, який відпрацює при зміні формату збереження
void Form1_FormClosing(object sender, FormClosingEventArgs e)	Метод, який відпрацює при закритті форми
void timerSec_Tick(object sender, EventArgs e)	Таймер
string PrepareTime(int time)	Метод, який працює разом з таймером
string GetFileName(string path)	Метод, який із шляху збереження повертає ім'я файлу
void buttonChangeFileName_Click(object sender, EventArgs e)	Метод, який працює при зміні статусу натискані для зміни імені файлу
void trackBarBrightness_Scroll(object sender, EventArgs e)	Метод, який працює при зміні статусу trackBarBrightness і викликає відповідну функцію для додавання фільтру
void trackBarContrast_Scroll(object sender, EventArgs e)	Метод, який працює при зміні статусу trackBarContrast і викликає відповідну функцію для додавання фільтру

Кінець таблиці 3.3

1	2
<code>void trackBarHue_Scroll(object sender, EventArgs e)</code>	Метод, який працює при зміні статусу <code>trackBarHue</code> і викликає відповідну функцію для додавання фільтру
<code>void checkBoxGray_CheckedChanged(object sender, EventArgs e)</code>	Метод, який працює при зміні статусу <code>checkBoxGray</code> і викликає відповідну функцію для додавання фільтру
<code>void checkBoxSepia_CheckedChanged(object sender, EventArgs e)</code>	Метод, який працює при зміні статусу <code>checkBoxSepia</code> і викликає відповідну функцію для додавання фільтру
<code>void checkBoxInverse_CheckedChanged(object sender, EventArgs e)</code>	Метод, який працює при зміні статусу <code>checkBoxInverse</code> і викликає відповідну функцію для додавання фільтру
<code>checkBoxHue_CheckedChanged(object sender, EventArgs e)</code>	Метод, який працює при зміні статусу <code>checkBoxHue</code> і викликає відповідну функцію для додавання фільтру
<code>buttonResetFilters_Click(object sender, EventArgs e)</code>	Метод, який відпрацює при натисканні на кнопку для сбросу фільтрів

3.1.4 Обробка виключних ситуацій

Розглянемо перелік виключень які можуть виникнути при роботі застосунку:

- перевірка чи вказано шлях для збереження перед записом або скріншотом. Перевірка відбувається за допомогою конструкції `if-else`. Якщо перевірка виявилась хибною, то запис або скріншот не зберігаються. Код перевірки наведено на рис. 3.4;

```

if (fileSavePath == null)
{
    MessageBox.Show("Вкажіть шлях для збереження відео");
    return;
};

```

Рисунок 3.4 – Код перевірки наявності шляху

- перед тим як зробити скріншот відбувається перевірка на наявність зображення в pictureBox;
- перед початком запису відео програма перевіряє наявність підключених відеокамер та коректність вибору відеокамери для запису. У випадку, якщо вибрана відеокамера недоступна або відсутня, програма виведе повідомлення про помилку і припинить процес запису. Код перевірки наявності відеокамер наведено на рис. 3.5.

```

public class CameraController
{ private FilterInfoCollection videoDevices; // Колекція відеокамер
  private VideoCaptureDevice videoSource; // Об'єкт для роботи з
відеокамерою
  public CameraController()
  { // Ініціалізація колекції доступних відеокамер
    videoDevices = new
FilterInfoCollection(FilterCategory.VideoInputDevice);
    // Перевірка, чи є хоча б одна відеокамера підключена
    if (videoDevices.Count == 0)
    { MessageBox.Show("Немає доступних відеокамер. Перевірте
підключення.", "Помилка",
      MessageBoxButtons.OK, MessageBoxIcon.Error);
      return;
    }
    // Вибір першої відеокамери для роботи
    videoSource = new
VideoCaptureDevice(videoDevices[0].MonikerString);
  }
}

```

Рисунок 3.5 – Код перевірки наявності відеокамер

– обробка помилок, пов'язаних з доступом до файлової системи. Перед записом відео або створенням скріншоту програма виконує перевірку доступності вказаного шляху для збереження відео. У разі, якщо зазначений шлях відсутній або недоступний, програма виведе відповідне повідомлення і припинить процес запису або створення скріншоту. Код перевірки доступності вказаного шляху наведено на рис. 3.6;

```
public void StartRecording()
{
    // Перевірка доступності шляху для збереження відео
    if (!Directory.Exists(Path.GetDirectoryName(fileSavePath)))
    {
        MessageBox.Show("Вказаний шлях для збереження відео
недоступний", "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    // Логіка початку запису відео за вказаним шляхом
    // ...
}
```

Рисунок 3.6 – Код перевірки доступності вказаного шляху

– обробка помилок під час взаємодії з відеопотоком. Під час запису відео програма періодично перевіряє доступність відеопотоку та коректність його отримання. У випадку виникнення помилки з отриманням відеопотоку (наприклад, відключення відеокамери або втрата зв'язку), програма ініціює відповідні заходи, включаючи припинення запису та виведення повідомлення про помилку користувачу. Код перевірки обробки помилок під час запису відео наведено на рис. 3.7;

– обробка помилок, пов'язаних зі зміною параметрів запису. Під час запису відео користувач може змінити параметри запису, такі як формат відео, роздільна здатність тощо. Програма обробляє помилки, які можуть виникнути

при зміні цих параметрів, забезпечуючи коректну роботу програми та виводячи інформативні повідомлення для користувача.

```
public void StartRecording()
{ // Перевірка доступності шляху для збереження відео
  if (!Directory.Exists(Path.GetDirectoryName(fileSavePath)))
  {
    MessageBox.Show("Вказаний шлях для збереження відео
недоступний", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
  }
  // Початок запису відео
  try
  {
    // Логіка початку запису відео за вказаним шляхом та форматом
    // ...
    isRecording = true;
  }
  catch (Exception ex)
  { | // Обробка помилок під час початку запису
    MessageBox.Show($"Виникла помилка при початку запису
відео: {ex.Message}", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
  }
}
```

Рисунок 3.7 – Код перевірки помилок під час запису відео

Для того, щоб переконатися, що програма вірно обробляє виняткові ситуації та виводить інформативні повідомлення, було проведено відповідні тести. Програма не дає почати запис або зробити скріншоти доки користувач не вкаже шлях для збереження файлів.

В процесі тестування запису відео, було перевірено, що відеозапис працює правильно, зберігається у визначеному форматі та з правильним налаштуванням. Приклад перевірки наведено на рис. 3.8.

При тестуванні можливості робити скріншоти, переконався, що можливо створювати скріншоти з відеопотоку та впевнився, що скріншоти зберігаються у вірному форматі та якості. До скріншота додається час коли він був зроблений, щоб можна було робити багато скріншотів без зміни імені файлів (рис. 3.9).

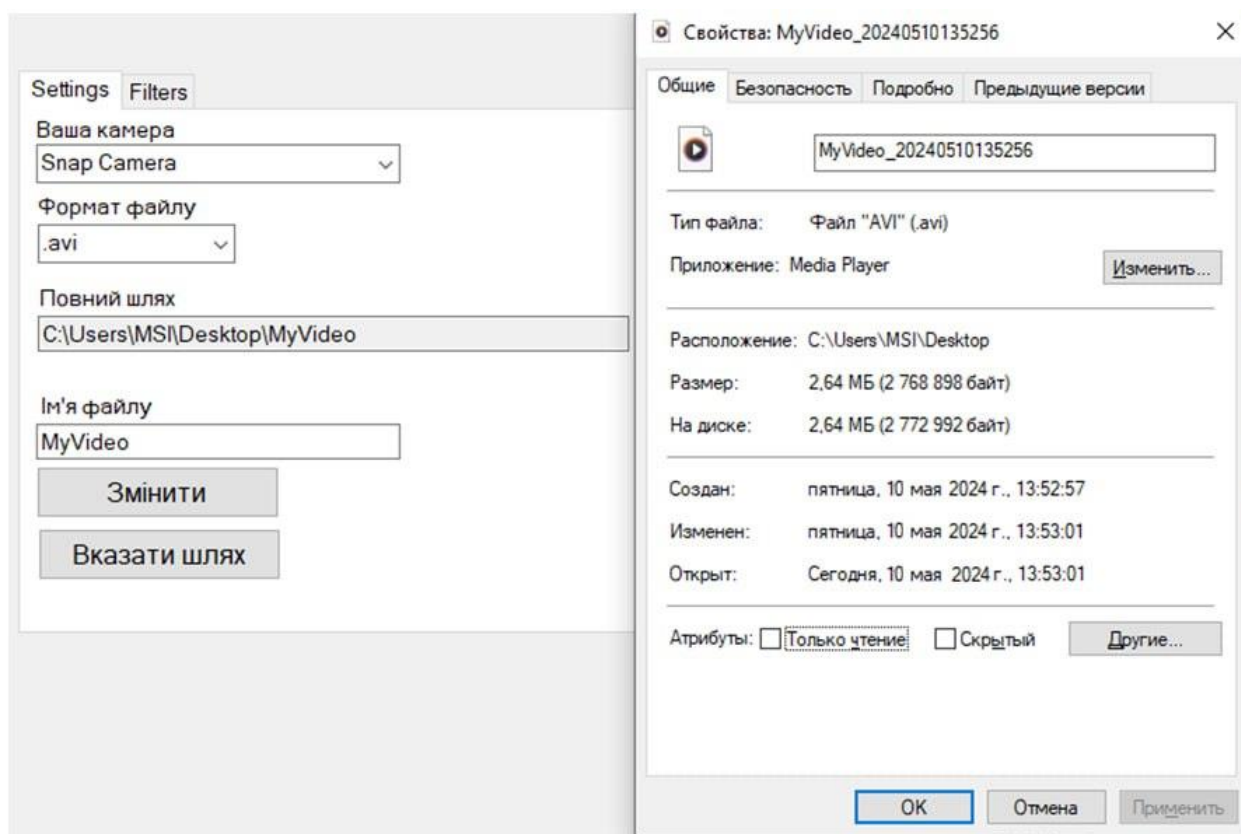


Рисунок 3.8 – Перевірка налаштувань та формату

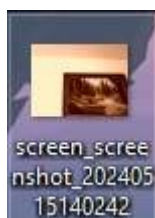


Рисунок 3.9 – Скріншот фрагменту відео

Також, в процеси тестування, було перевірено роботу кожного фільтра. Впевнився, що фільтри можна легко додавати та видаляти та, що

зміни в фільтрах відображаються на відеопотоці і скріншотах. На рис. 5.4 видно, що фільтр впливає на скріншот.

Коди основних класів програми у скороченому вигляді наведено у Додатку А.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення та умови застосування програми

Програма для запису відео з вебкамери розроблена з метою забезпечення зручного та надійного інструменту для захоплення відеоматеріалів, отриманих від вебкамери. Це важливий інструмент для користувачів, які потребують швидкого і простого способу запису відео без необхідності використання складних програм або професійного обладнання.

4.2 Характеристики програми

Програма може бути виконана при умові, що її запускають на персональному комп'ютері. Функціонал програмного застосунку наведено у п. 1.3.3. Вимоги до технічних засобів та операційної системи наведені у п. 1.3.4.

Для запуску програмного забезпечення необхідно виконати файл WebRecorder.exe під управлінням операційної системи Windows 10.

Програма створена мовою C# у середовищі Visual Studio 2019 на платформі .NET Framework 4.7.2. Об'єм оперативної пам'яті, що використовується при виконанні дорівнює 500 Мб.

Об'єм пам'яті для збереження бази даних на жорсткому диску – залежить від об'єма інформації, що записується.

4.3 Структура програми

При виконанні кваліфікаційної роботи бакалавра було використано сторонні бібліотеки Accord.Video.FFMPEG [11] та Aforge.Video [10], які надають можливість обробляти зображення.

Accord.Video.FFMPEG є одним із модулів бібліотеки Accord.NET і забезпечує можливості для роботи з відео, зокрема, для зчитування та запису відео за допомогою бібліотеки FFmpeg.

AForge.NET – це відкрита бібліотека для роботи з обробкою зображень, комп'ютерного зору та штучного інтелекту. Вона надає розробникам доступ до різноманітних інструментів і функцій, що допомагають у розробці застосунків, пов'язаних з обробкою зображень та відео. В даній роботі було використано деякі класи з цієї бібліотеки для роботи з відеопотоком та відеозаписом.

Було використано наступні класи:

а) VideoCapabilities – це клас, який представляє можливості відеопотоку або відеопристрою. Він містить інформацію про різні можливості відеозйомки:

1) розмір кадру. VideoCapabilities дозволяє отримати інформацію про розмір кадру, який підтримується відеопристроєм. Це важливо для налаштування роздільної здатності відеозйомки та підбору оптимальних налаштувань для обробки відеоданих;

2) кількість кадрів в секунду. VideoCapabilities надає інформацію про кількість кадрів, які можуть бути захоплені або відтворені в секунду. Це важливо для визначення плавності відеопотоку та підбору оптимальних параметрів відеозйомки;

3) кодеки. Клас також дозволяє отримати список підтримуваних кодеків, які можна використовувати для кодування або декодування відеоданих. Це важливо для вибору правильного кодеку, який відповідає вимогам якості та об'єму даних відеопотоку;

4) інші характеристики. VideoCapabilities може містити інші характеристики, такі як підтримка режимів відеозйомки (наприклад, режими HD або Full HD), підтримка функцій автоматичного фокусування або стабілізації зображення та інші параметри, які впливають на якість та функціональність відеопристрою;

б) VideoFileWriter – цей клас з бібліотеки AForge.NET представляє об'єкт, що використовується для запису відео в файл. Він взаємодіє з відеопристроєм, таким як веб-камера, і забезпечує можливість створення та запису відеофайлу на основі відеопотоку, отриманого з вказаного відеопристрою. VideoFileWriter надає функціональність запису відео з використанням різних кодеків та налаштувань, що дозволяє користувачу зберігати відеоматеріали у вибраному форматі та якості;

в) FilterInfoCollection. Цей клас відноситься до бібліотеки AForge.NET і використовується для отримання інформації про доступні відеопристрої в системі. Він дозволяє отримувати перелік доступних відеопристроїв та їх характеристик;

г) VideoCaptureDevice – цей клас з бібліотеки AForge.NET є об'єктом, призначеним для взаємодії з відеопристроєм, таким як вебкамера. Він надає можливість отримувати відеопотік з вказаного відеопристрою та здійснювати різні операції, пов'язані з введенням відео. VideoCaptureDevice використовується для отримання відеопотоку, який може бути використаний для подальшої обробки або запису. Крім того, цей клас забезпечує можливість контролювати параметри відеозйомки, такі як роздільна здатність, кадрова частота, експозиція та інші характеристики відеопристрою. Він є важливою складовою для створення програм, які працюють з відеопотоком, такими як програми запису відео, відеочати або системи відеоспостереження.

4.4 Вхідні та вихідні дані

4.4.1 Вхідні дані

Введення даних відбувається під час налаштування параметрів для запису відео та створення скріншотів. Користувач має можливість вказати шлях для збереження відео та скріншотів, обрати тип відеопристрою для запису відео, вибрати формат файлу, в якому буде збережено відео, а також встановити фільтри, які будуть застосовані до відеопотоку. Кожен з цих

параметрів впливає на процес запису та визначає характеристики створених відео- та зображення-файлів, що дозволяє користувачеві налаштувати роботу програми згідно з його потребами та вимогами.

4.4.2 Вихідні дані

Вихідні дані включають в себе кілька компонентів, які формують результат роботи програми. Серед них секундомір запису відео, що відображає час тривалості запису відеоматеріалу, скріншоти, які створюються користувачем у процесі роботи програми та записані відеофайли, які містять зібраний відеопотік. Кожен з цих компонентів є важливим елементом взаємодії з програмою та забезпечує користувачеві необхідну інформацію про результати її роботи.

4.5 Повідомлення програмісту

Застосунок не містить повідомлень для програміста.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програми

Програма розроблена з метою надання користувачеві зручного і простого інструмента для запису відео з вебкамери та створення скріншотів. Основне призначення полягає в тому, щоб допомогти користувачеві зафіксувати важливі моменти або проводити відео-звіти з вебкамери з можливістю застосування різноманітних фільтрів для покращення зображення.

5.2 Умови виконання програми

5.2.1 Апаратні вимоги програми

Програма вимагає наявності операційної системи Windows 10, а також наявності вільного місця на жорсткому диску не менше 50 Гб для зберігання відео та скріншотів. Рекомендується використовувати комп'ютер з процесором Intel Pentium 4 2,0 ГГц або еквівалентним, а також не менше 2 Гб оперативної пам'яті.

5.2.2 Вимоги до користувача

Користувач має мати базові навички роботи з комп'ютером та веб-камерою, а також розуміння процесу збереження файлів на диску.

5.3 Виконання програми

Програма запускається на комп'ютері під управлінням операційної системи Windows 10 за допомогою виконуваного файлу WebRecorder.exe шляхом подвійного кліку лівої клавіші комп'ютерної миші на відповідний ярлик на робочому столі або в меню "Пуск".

При запуску програми користувачу відображається головне вікно, яке зображено на рис. 5.1.

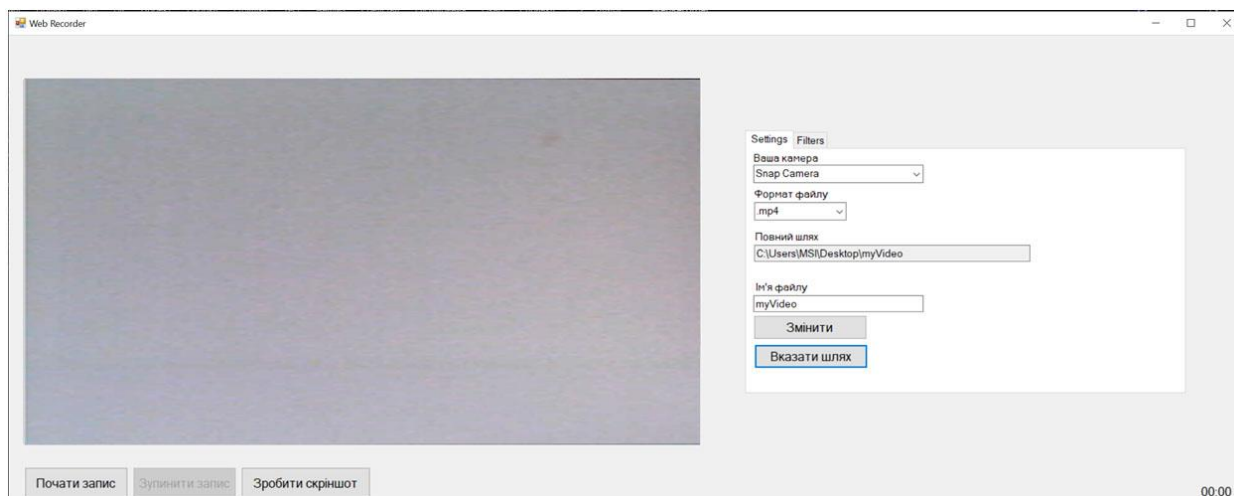


Рисунок 5.1 – Головне вікно

Після запуску програми користувач відкриває головне вікно, де вказує шлях для збереження відео та скріншотів, обирає пристрій для запису відео, встановлює формат файлу та застосовує фільтри за необхідності. Параметри, що може налаштовувати користувач для збереження відеоінформації наведено на рис. 5.2.

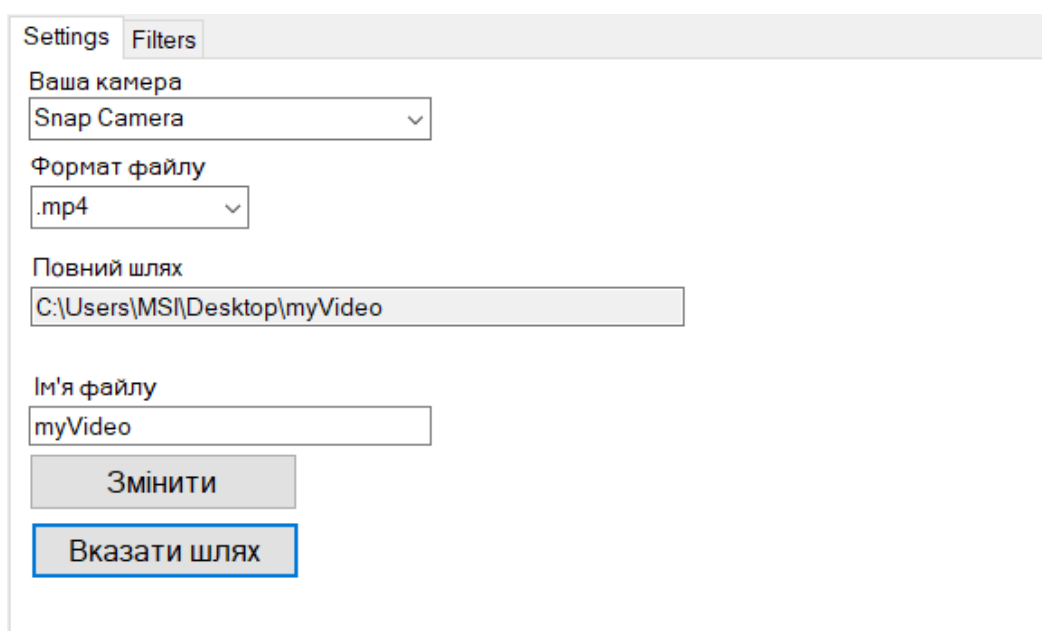


Рисунок 5.2 – Параметри налаштування

Параметри, що може налаштовувати користувач для встановлення фільтрів наведено на рис. 5.3.

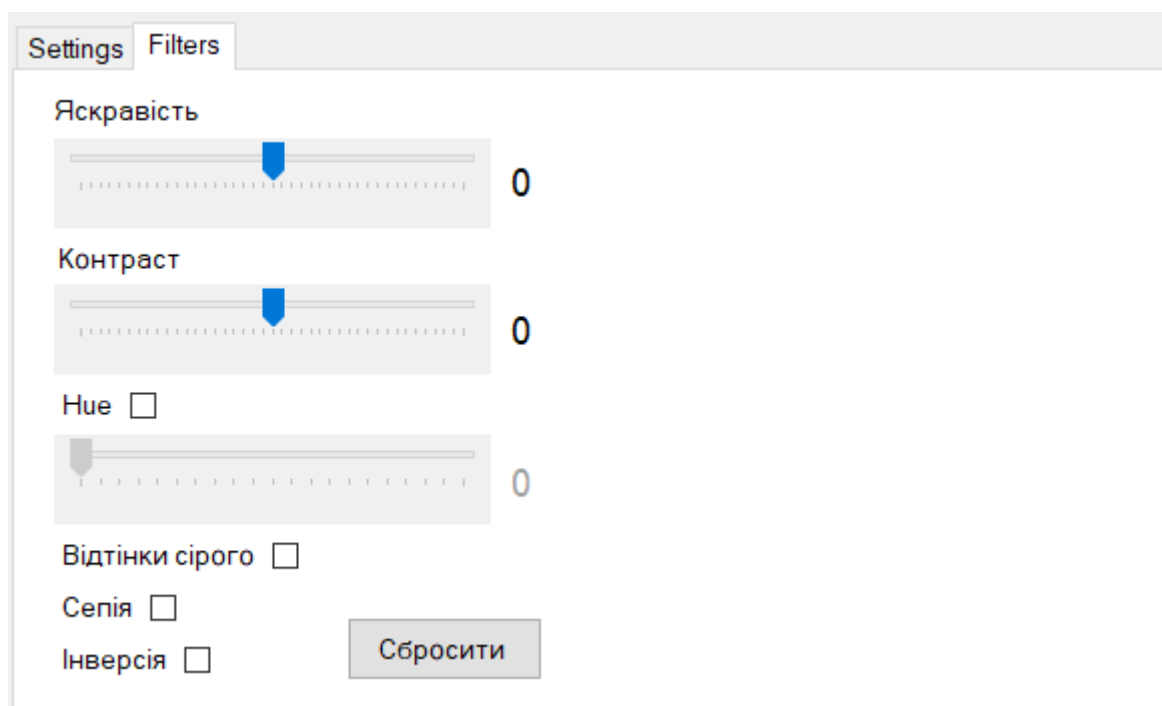


Рисунок 5.3 – Застосування фільтрів

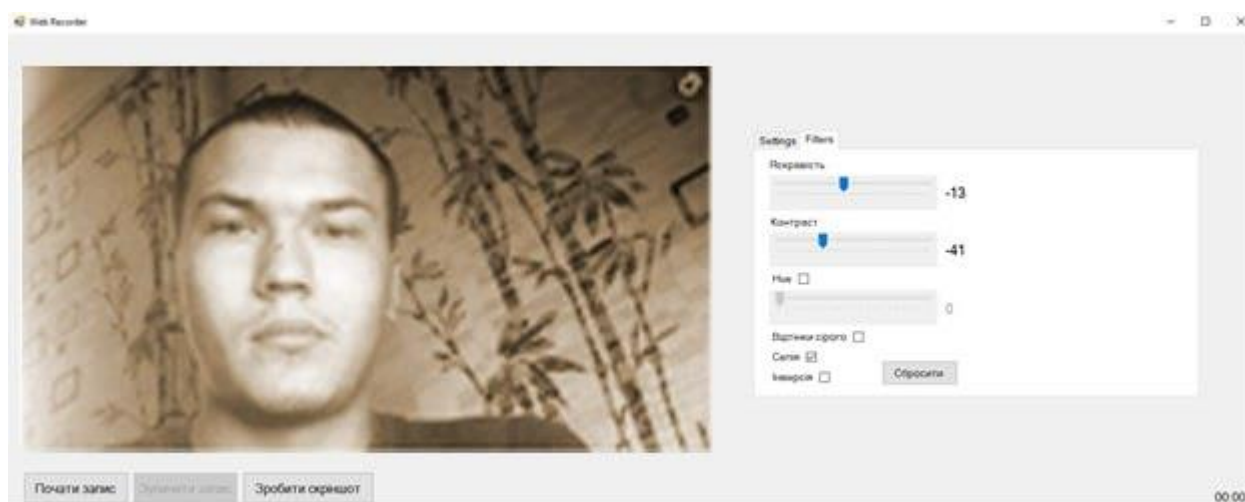


Рисунок 5.4 – Приклад комбінації фільтрів

Під час запису користувачеві заборонено змінювати налаштування. У правому нижньому куті є таймер, який відображає час того, скільки йде запис (рис. 5.5).

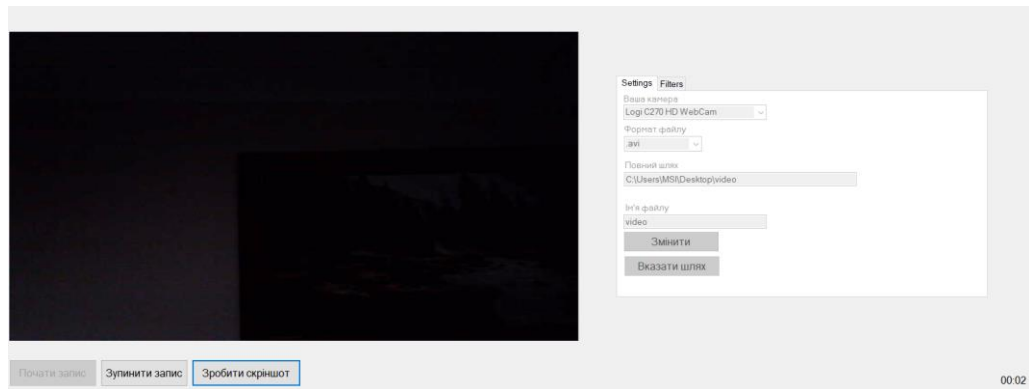


Рисунок 5.5 – Під час запису налаштування недоступні

5.4 Повідомлення оператору

Повідомлення оператору виводяться, якщо він не обрав шлях для збереження файлів (рис.5.6).

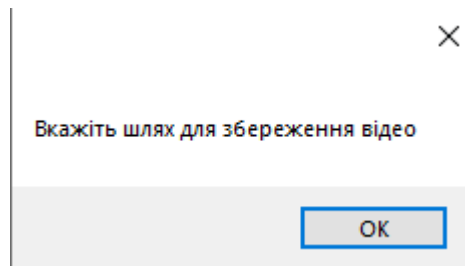


Рисунок 5.6 – Приклад повідомлення оператору

ВИСНОВКИ

Під час виконання дипломної кваліфікаційної роботи бакалавра було вирішено актуальну задачу розробки програмного застосунку, що забезпечує запис відео та створення скріншотів з вебкамери.

Були успішно виконані наступні завдання:

- проведено аналіз програмних аналогів;
- розроблено ТЗ;
- проведено аналіз функціональності та моделювання сценаріїв дій користувача;
- реалізовано застосунок з використанням бібліотек Accord.Video.FFMPEG та Aforge.Video;
- реалізовано обробку виключних ситуацій;
- розроблено відповідну програмну документацію.

Створений проєкт відповідає меті розробки, всім вимогам, що були пред'явлені до програмного засобу.

Практична цінність розробленого програмного застосунку полягає в тому, що він надає користувачеві можливості зафіксувати важливі моменти або проводити відео-звіти з вебкамери з можливістю застосування різноманітних фільтрів для покращення зображення.

Програмна документація включає: технічне завдання; керівництво програміста; керівництво оператора та тексти основних програмних модулів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Microsoft Windows Camera app: What it can do, features and more. [Electronic resource]. – Access mode: http://timesofindia.indiatimes.com/articleshow/93130725.cms?utm_source=contentofinterest&utm_medium=text&utm_campaign=cppst.
2. Windows Camera [Electronic resource]. – Access mode: <https://windows-camera.en.softonic.com/>.
3. Camtasia - Recording from the Web [Electronic resource]. – Access mode: <https://guides.library.ucsc.edu/DS/Resources/Camtasia#:~:text=Camtasia%20is%20a%20recording%20software,filming%20webinars%20or%20video%20tutorials>.
4. Pros and Cons of Using Camtasia for Training Videos [Electronic resource]. – Access mode: https://waterbearlearning.com/pros-cons-camtasia-training-videos/#google_vignette.
5. Open broadcaster software (OBS) [Electronic resource]. – Access mode: <https://www.bradley.edu/sites/it/departments/ldt/it-tools/obs/#:~:text=Open%20Broadcaster%20Software%2C%20or%20OBS,creative%20video%20and%20broadcast%20applications>.
6. OBS Studio review [Electronic resource]. – Access mode: <https://www.techradar.com/reviews/obs-studio>.
7. Вимоги до системи для Windows 10 [Електронний ресурс]. – Режим доступу: <https://support.microsoft.com/uk-ua/windows/вимоги-до-системи--для-windows-10-6d4e9a79-66bf-7950-467c-795cf0386715>.
8. MVC: Model, View, Controller [Electronic resource]. – Access mode: <https://www.codecademy.com/article/mvc>.
9. Model-View-ViewModel (MVVM) [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>.
10. AForge.NET [Electronic resource]. – Access mode: <https://www.aforge.net/framework/>.

11. Accord.NET framework [Electronic resource]. – Access mode: <http://accord-framework.net/>.
12. Documentation Emgu CV [Electronic resource]. – Access mode: <https://www.emgu.com/wiki/index.php/Documentation>.
13. Visual Studio documentation [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>.

ДОДАТОК А
Текст програми

A.1 Код CameraController

```

internal class CameraController
{
    private IVideoDeviceManager videoDeviceManager;
    private IVideoProcessor videoProcessor;
    private IVideoRecorder videoRecorder;
    private bool isRecording;
    public CameraController(IVideoDeviceManager deviceManager, IVideoProcessor processor,
IVideoRecorder recorder)
    {
        videoDeviceManager = deviceManager;
        videoProcessor = processor;
        videoRecorder = recorder;
    }
    public void GetVideoDevices(ComboBox comboBox)
    {
        videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
        if (videoDevices.Count > 0)
        {
            foreach (FilterInfo device in videoDevices)
            {
                comboBox.Items.Add(device.Name);
            }
            comboBox.SelectedIndex = 0;
        }
        else
        {
            MessageBox.Show("No video devices found");
        }
    }
    public void Start()
    {
        videoDeviceManager.Start();
        videoDeviceManager.VideoFrameReceived += VideoFrameReceivedHandler;
    }
    public void Stop()

```

```

{
    videoDeviceManager.VideoFrameReceived -= VideoFrameReceivedHandler;
    videoDeviceManager.Stop();
}

public void StartRecording(string fileSavePath, string fileFormat)
{
    if (!isRecording)
    {
        isRecording = true;
        videoRecorder.StartRecording(fileSavePath, fileFormat);
    }
}

public void StopRecording()
{
    if (isRecording)
    {
        isRecording = false;
        videoRecorder.StopRecording();
    }
}

private void VideoFrameReceivedHandler(object sender, VideoFrameEventArgs eventArgs)
{
    Bitmap processedFrame = videoProcessor.ProcessFrame(eventArgs.Frame);
    if (isRecording)
    {
        videoRecorder.RecordFrame(processedFrame);
    }
}

public void ChangeVideoDevice(int selectedIndex)
{
    StopRecording();
    Stop();
    videoDeviceManager.ChangeDevice(selectedIndex);
    Start();
}
}

```

A.2 Код класу Form1

```

public partial class Form1 : Form
{
    private bool isRecording;
    private string fileSavePath;
    private string fileFormat;
    private FiltersController filtersController;
    private CameraController cameraController;
    public Form1()
    {
        InitializeComponent();
        // Ініціалізація контролерів
        filtersController = new FiltersController();
        cameraController = new CameraController(pictureBoxVideo, checkBoxHue);
        // Отримання відеопристроїв
        cameraController.GetVideoDevices(webcamComboBox);
        // Налаштування інтерфейсу
        stopRecord.Enabled = false; fileFormat = ".avi";
    }
    // Обробник події завантаження форми
    private void Form1_Load(object sender, EventArgs e)
    { // Відтворення відео
        cameraController.PlayVideo();
    }
    // Почати запис відео
    private void StartRecording()
    {
        if (fileSavePath == null)
        {
            MessageBox.Show("Вкажіть шлях для збереження відео");
            return;
        }
        cameraController.StartRecording(fileSavePath, fileFormat);
        startRecord.Enabled = false;
        tabControl1.Enabled = false;
        stopRecord.Enabled = true;
    }
}

```

```

        isRecording = true;
        timerSec.Enabled = true;
    }
    // Зупинити запис відео
    private void StopRecording()
    {
        if (isRecording)
        {
            isRecording = false;
            startRecord.Enabled = true;
            stopRecord.Enabled = false;
            tabControl1.Enabled = true;
            timerSec.Enabled = false;
            labelTimer.Text = "00:00";
            cameraController.StopRecording();
        }
    }
    // Зробити знімок екрану
    private void TakeScreenshot()
    {
        if (fileSavePath == null)
        {
            MessageBox.Show("Вкажіть шлях для збереження знімку");
            return;
        }
        if (pictureBoxVideo.Image != null)
        {
            string screenshotName = $"screenshot_{DateTime.Now:yyyyMMddHHmmss}.png";
            pictureBoxVideo.Image.Save($"{fileSavePath}_{screenshotName}", ImageFormat.Png);
        }
    }
    private void buttonTakeScreenshot_Click(object sender, EventArgs e)
    {
        TakeScreenshot();
    }
    private void startRecord_Click(object sender, EventArgs e)
    { StartRecording();
    }

```

```

private void stopRecord_Click(object sender, EventArgs e)
{
    StopRecording();
}

// Обробник зміни обраного відеопристрою
private void webcamComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    cameraController.ChangeVideoDevice(webcamComboBox.SelectedIndex);
}

// Обробник зміни обраного формату файла
private void comboSelectFormat_SelectedIndexChanged(object sender, EventArgs e)
{
    fileFormat = comboSelectFormat.Text;
}

// Обробник події закриття форми
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    StopRecording();
    cameraController.StopCamera();
}

// Обробник події таймера
private void timerSec_Tick(object sender, EventArgs e)
{
    // Оновление часу таймера
    string[] time = labelTimer.Text.Split(':');
    int minutes = int.Parse(time[0]);
    int seconds = int.Parse(time[1]);
    seconds++;
    if(seconds == 60)
    {
        seconds = 0;
        minutes++;
    }
    labelTimer.Text = PrepareTime(minutes) + ':' + PrepareTime(seconds);
}

// Метод для форматування часу
private string PrepareTime(int time)
{
    if (time < 10)
        return '0' + time.ToString();
}

```

```

        return time.ToString();
    }
    // Метод для отримання імені файла із шляху
    private string GetFileName(string path)
    {
        string[] buff = path.Split("\\");
        string fileName = buff[buff.Length - 1];
        return fileName;
    }
    // Обробник натискання кнопки зміни імені файла
    private void buttonChangeFileName_Click(object sender, EventArgs e)
    {
        if (fileSavePath == null) return;
        textBoxSavePath.Text = fileSavePath;
        string[] buff = fileSavePath.Split("\\");
        buff[buff.Length - 1] = textBoxFilePathName.Text;
        fileSavePath = string.Join("\\", buff);
        textBoxSavePath.Text = string.Join("\\", buff);
    }
    // Обробники зміни параметрів фільтрів
    private void trackBarBrightness_Scroll(object sender, EventArgs e)
    {
        labelBrightScroll.Text = trackBarBrightness.Value.ToString();
        cameraController.ChangeFilterValue("BrightnessValue", trackBarBrightness.Value);
    }
    private void trackBarContrast_Scroll(object sender, EventArgs e)
    {
        labelContrastScroll.Text = trackBarContrast.Value.ToString();
        cameraController.ChangeFilterValue("ContrastCorrectionValue", trackBarContrast.Value);
    }
    private void trackBarHue_Scroll(object sender, EventArgs e)
    {
        labelHueScroll.Text = trackBarHue.Value.ToString();
        cameraController.ChangeFilterValue("HueFilterValue", trackBarHue.Value);
    }
    private void checkBoxGray_CheckedChanged(object sender, EventArgs e)
    {
        cameraController.ChangeFilterValue("IsGrayScale", checkBoxGray.Checked);
    }

```

```

    }
    private void checkBoxSepia_CheckedChanged(object sender, EventArgs e)
    {
        cameraController.ChangeFilterValue("IsSepia", checkBoxSepia.Checked);
    }
    private void checkBoxInverse_CheckedChanged(object sender, EventArgs e)
    {
        cameraController.ChangeFilterValue("IsInverse", checkBoxInverse.Checked);
    }
    private void checkBoxHue_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBoxHue.Checked)
        {
            trackBarHue.Enabled = true;
            labelHueScroll.Enabled = true;
        }
        else
        {
            trackBarHue.Enabled = false;
            labelHueScroll.Enabled = false;
        }
    }
    private void buttonResetFilters_Click(object sender, EventArgs e)
    {
        // Скидання параметрів фільтрів
        cameraController.ResetFilters();
        checkBoxGray.Checked = false;
        checkBoxSepia.Checked = false;
        checkBoxInverse.Checked = false;
        checkBoxHue.Checked = false;
        trackBarBrightness.Value = 0;
        trackBarContrast.Value = 0;
        trackBarHue.Value = 0;
        labelBrightScroll.Text = "0";
        labelContrastScroll.Text = "0";
        labelHueScroll.Text = "0";
    }
}

```

ДОДАТОК Б
Слайди презентації

ДИПЛОМНА РОБОТА

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАПИСУ ТА ОБРОБКИ ВІДЕО З ВЕБКАМЕРИ

Розробив: ст. групи КНТ-130
Керівник: доцент, к.т.н.

Нікіта ДІДЕНКО
Сергій СЕРДЮК

2024

Рисунок Б.1 – Слайд №1

Мета роботи

розробка програми, що забезпечує користувачу змогу отримувати, обробляти та зберігати інформацію з вебкамери

Об'єкт дослідження — є процеси отримання та переробки інформації з вебкамери

Предмет дослідження — методи отримання та переробки інформації з вебкамери



НУ "ЗП", Кафедра програмних засобів

Рисунок Б.2 – Слайд №2

Задачі, що вирішуються

- провести аналіз предметної області;
- провести огляд програмних аналогів;
- розробка технічного завдання;
- розробка структури програмного забезпечення;
- розробка структури класів програмного забезпечення;
- програмна реалізація застосунку;
- розробка необхідної документації.



НУ "ЗП", Кафедра програмних засобів

Рисунок Б.3 – Слайд №3

3

Аналіз програм-аналогів

Програми Критерії	Web Recorder	Camtasia	Windows Camera	OBS Studio
Безкоштовність	Так	Ні	Так	Ні
Знімки камери	Так	Так	Так	Ні
Інтуїтивний інтерфейс	Так	Ні	Так	Ні
Розширені налаштування	Ні	Так	Так	Так
Підтримка стрімінгу	Ні	Так	Ні	Так
Можливості керування аудіо	Ні	Так	Ні	Так
Підтримка фільтрів	Так	Так	Ні	Так
Перегляд відео після запису	Ні	Так	Так	Ні

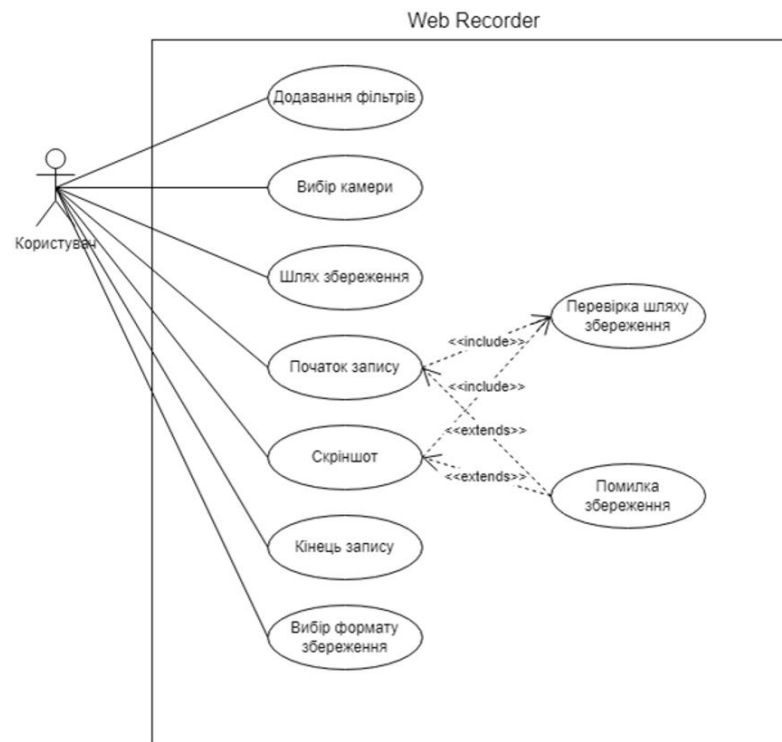


НУ "ЗП", Кафедра програмних засобів

Рисунок Б.4 – Слайд №4

4

Use-case діаграма



НУ "ЗП", Кафедра програмних засобів

5

Рисунок Б.5 – Слайд №5

Бібліотеки, їх модулі та класи, що використані у програмі

Назва	Опис
Accord.Video.FFMPEG	Модуль бібліотеки Accord.NET. Забезпечує можливості для роботи з відео, зокрема, для зчитування та запису відео за допомогою бібліотеки FFmpeg
AForge.NET	Відкрита бібліотека для роботи з обробкою зображень, комп'ютерного зору та штучного інтелекту.
– VideoCapabilities	Клас, який представляє можливості відеопотоку або відеопристрою
– VideoFileWriter	Надає функціональність запису відео з використанням різних кодеків та налаштувань
– FilterInfoCollection	Отримання інформації про доступні відеопристрої в системі
– VideoCaptureDevice	Отримання відеопотоку, який може бути використаний для подальшої обробки або запису. Крім того, цей клас забезпечує можливість контролювати параметри відеозйомки

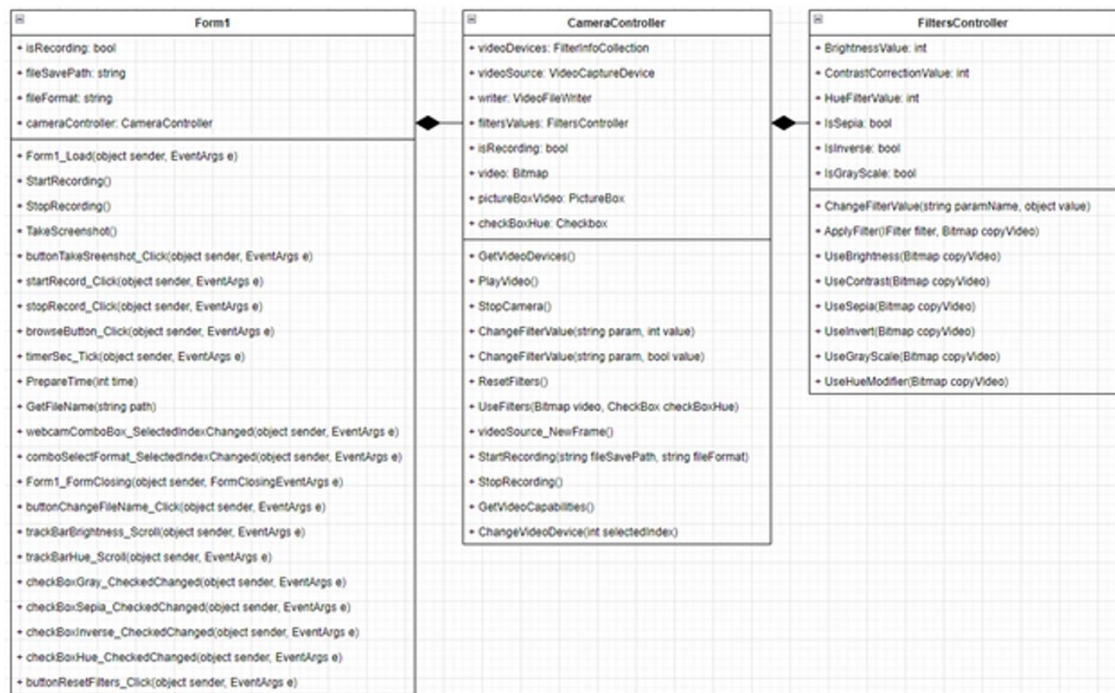


НУ "ЗП", Кафедра програмних засобів

6

Рисунок Б.6 – Слайд №6

Діаграма класів



НУ "ЗП", Кафедра програмних засобів

7

Рисунок Б.7 – Слайд №7

Приклад полів та методів класу CameraController

Поля та методи класу	Опис
public:	
CameraController(PictureBox pictureBoxVideoLink, CheckBox checkBoxHueLink)	Конструктор, який приймає PictureBox та CheckBox для взаємодії з елементами інтерфейсу
void GetVideoDevices(ComboBox comboBox)	Метод, який заповнює ComboBox інформацією про доступні відеопристрої
void PlayVideo()	Метод, який налаштовує обробник подій NewFrame для відеопристрою та починає відтворення відео
void StopCamera()	Метод, який зупиняє відтворення відео
void ChangeFilterValue(string param, int value)	Метод, який додає фільтр, який використовує число
void ChangeFilterValue(string param, bool value)	Метод, який додає фільтр, який використовує значення істини
void ResetFilters()	Метод, який збрасує всі фільтри
void StartRecording(string fileSavePath, string fileFormat)	Метод, який розпочинає запис відео
void StopRecording()	Метод, який зупиняє запис відео, та розпочинає звичайне відтворення відео
void ChangeVideoDevice(int selectedIndex)	Метод, який змінює вибраний відеопристрій та знов розпочинає відтворення відео



НУ "ЗП", Кафедра програмних засобів

8

Рисунок Б.8 – Слайд №8

Приклад коду методу StartRecording

```

private void StartRecording()
{
    if (fileSavePath == null)
    {
        MessageBox.Show("Вкажіть шлях для збереження відео");
        return;
    };

    cameraController.StartRecording(fileSavePath, fileFormat);

    startRecord.Enabled = false;
    tabControl1.Enabled = false;
    stopRecord.Enabled = true;

    isRecording = true;
    timerSec.Enabled = true;
}

```



НУ "ЗП", Кафедра програмних засобів

9

Рисунок Б.9 – Слайд №9

Приклад обробки помилок

```

public void StartRecording()
{
    // Перевірка доступності шляху для збереження відео
    if (!Directory.Exists(Path.GetDirectoryName(fileSavePath)))
    {
        MessageBox.Show("Вказаний шлях для збереження відео  
недоступний", "Помилка",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    // Логіка початку запису відео за вказаним шляхом
    // ...
}

```

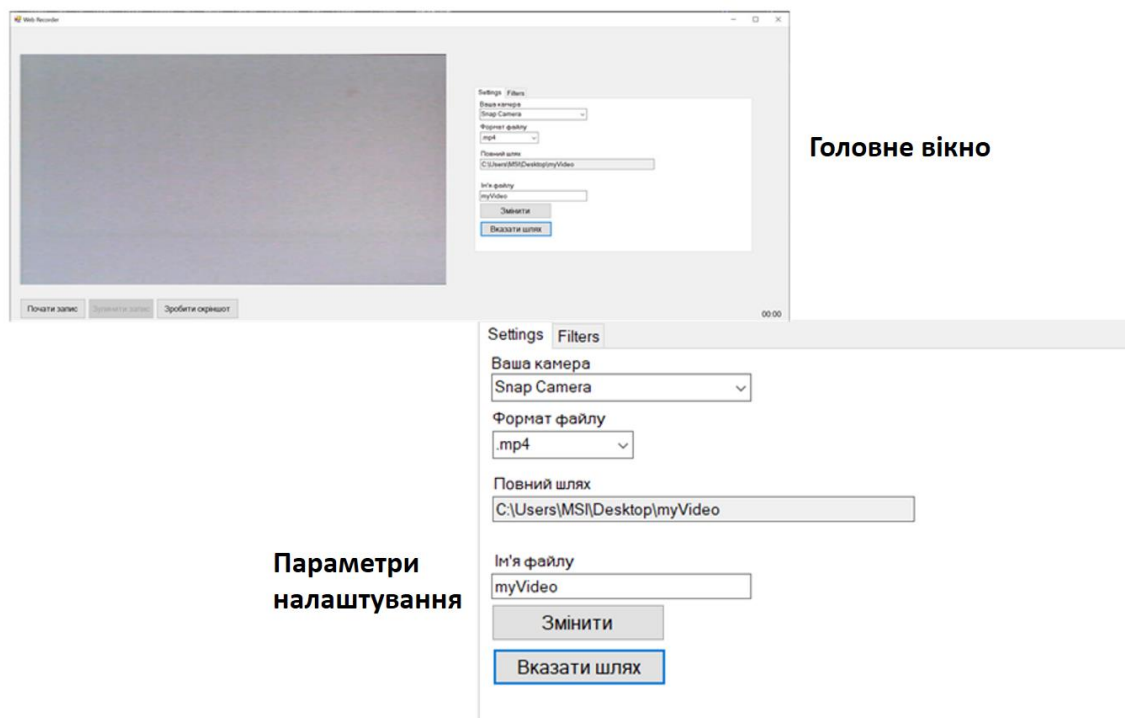


НУ "ЗП", Кафедра програмних засобів

10

Рисунок Б.10 – Слайд №10

Інтерфейс програми



Головне вікно

Параметри
налаштування

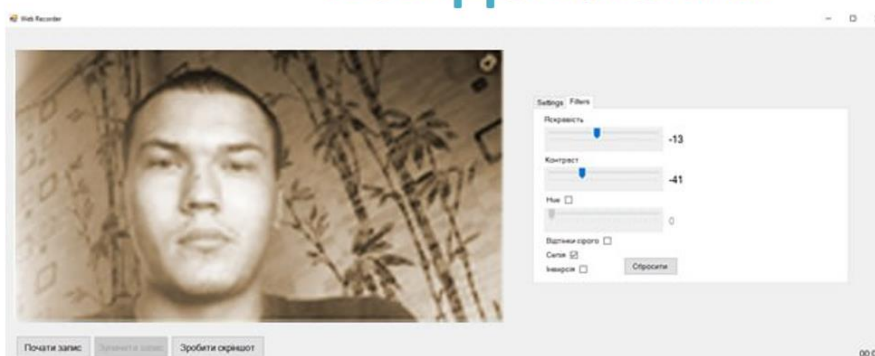


НУ "ЗП", Кафедра програмних засобів

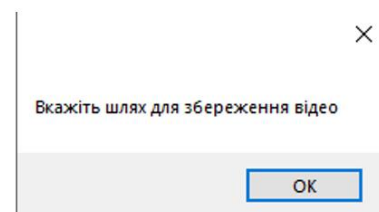
11

Рисунок Б.11 – Слайд №11

Обробка зображень та повідомлення



Приклад комбінації фільтрів



Повідомлення



НУ "ЗП", Кафедра програмних засобів

12

Рисунок Б.12 – Слайд №12

Висновки

- проведено аналіз предметної області;
- проведено огляд програмних аналогів;
- розроблено технічне завдання;
- розроблено структуру програмного забезпечення;
- розроблено структуру класів програмного забезпечення;
- програмно реалізовано застосунок;
- розроблено необхідну документацію.



НУ "ЗП", Кафедра програмних засобів

13

Рисунок Б.13 – Слайд №13

**Доклад закінчено,
дякую за увагу!**



НУ "ЗП", Кафедра програмних засобів

14

Рисунок Б.14 – Слайд №14