

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра Технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-наукова програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

**“Дослідження процесів управління проектом розробки програмного
забезпечення для цифрової обробки зображень”**

Студента 2-го курсу групи УП-22

Науковий керівник:

Олега СТЕЛЬМАХА

кандидат фізико-математичних наук,
доцент
Григорій СТЕШЕНКО

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри
Технологій управління

Віктор МОРОЗОВ

(підпис)

(ім'я, прізвище)

(дата)

Київ – 2024

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра Технологій управління

Освітній рівень Магістр

Спеціальність 122 “Комп’ютерні науки”

Освітньо-наукова програма “Управління проектами”

ЗАТВЕРДЖУЮ

Завідувач кафедри

професор Віктор МОРОЗОВ

“___” _____ 20__ року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Стельмах Олег Юрійович

Група: УП-22

1. Тема кваліфікаційної роботи: “Дослідження процесів управління проектом розробки програмного забезпечення для цифрової обробки зображень”. Затверджена на засіданні кафедри Технологій управління, протокол №6 від 06.11.2023 р.

2. Строк подання студентом готової роботи – “21” травня 2024 р.

3. Цільова установка та вихідні дані до роботи: дослідження процесів управління проектом із розробки ПЗ для цифрової обробки зображень; пошук оптимальних рішень для досягнення цілей та мети; планування проекту.

4. Зміст роботи: аналіз предметної області, огляд останніх досліджень та публікацій, огляд ринку, визначення проблематики, наукова новизна та інноваційність, постановка задачі дослідження, формалізація ідеї та задуму

проєкту, аналіз зацікавлених сторін та оточення проєкту, аналіз проєктних альтернатив, загальні характеристики проєкту, концептуальні моделі інформаційної системи, формалізація математичних моделей, методи прийняття управлінських рішень, опис ПЗ та технологій для реалізації, огляд архітектури, компонентів та підходів до розробки, розробка БД, огляд алгоритмів, опис інтерфейсу користувача, WBS, OBS, вибір підходу до розробки ПЗ, календарно-ресурсне планування, оцінка витрат, управління якістю та ризиками проєкту.

5. Перелік графічного матеріалу (слайдів): титульна сторінка, мета роботи, огляд ринку, дерево причин та наслідків, концептуальна модель складу інформаційної системи, опис ПЗ та діаграми, структура бази даних, каркасний інтерфейс, WBS, OBS, життєвий цикл, календарне планування, висновки.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	План виконання роботи
1	Затвердження теми роботи та наукового керівника	06.11.2023
2	Дослідження предметної галузі, її проблематики, збір та вивчення науково-інформаційних матеріалів	22.01.2024 – 08.02.2024
3	Створення змісту кваліфікаційної роботи магістра	09.02.2024 - 20.02.2024
4	Підготовка основних розділів роботи	21.02.2024 - 23.04.2024
5	Оформлення та передача на рецензію керівнику	24.04.2024 - 06.05.2024
6	Попередній захист роботи та перевірка на плагіат	07.05.2024 - 16.05.2024
7	Передача друкованої роботи на кафедру та рецензію	17.05.2024 - 20.05.2024
8	Захист магістерської кваліфікаційної роботи	21.05.2024 - 23.05.2024

Дата видачі завдання “06” листопада 2023 р.

Керівник роботи:

к.ф.-м.н., доцент

Григорій СТЕШЕНКО

(підпис)

Завдання прийняв до виконання:

студент групи УП-22

Олег СТЕЛЬМАХ

(підпис)

ЗМІСТ

АНОТАЦІЯ	6
ПЕРЕЛІК ПОЯСНЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
РОЗДІЛ 1	
АНАЛІТИЧНІ ДОСЛІДЖЕННЯ ТА ОЦІНКА ДОЦІЛЬНОСТІ І	
ЖИТТЄЗДАТНОСТІ ПРОЄКТУ	
1.1. Аналіз предметної області	12
1.2. Огляд останніх досліджень та публікацій	15
1.3. Огляд ринку цифрової обробки зображень	17
1.4. Вивчення проблематики предметної області	21
1.5. Наукова новизна та інноваційність	23
1.6. Постановка задачі дослідження.....	24
РОЗДІЛ 2	
РОЗРОБКА ПРОЄКТНОЇ ПРОПОЗИЦІЇ ТА МАТЕМАТИЧНА ПОСТАНОВКА	
ЗАДАЧІ ДОСЛІДЖЕННЯ	
2.1. Формалізація ідеї та задуму проєкту	26
2.2. Аналіз зацікавлених сторін та оточення проєкту	29
2.3. Аналіз проєктних альтернатив	32
2.4. Загальні характеристики проєкту	34
2.5. Концептуальні моделі інформаційної системи.....	39
2.6. Формалізація математичних моделей	41
2.7. Прийняття управлінських рішень в проєкті.....	44
РОЗДІЛ 3	
РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
ПРОЄКТУ	
3.1. Опис програмного забезпечення та технологій для його реалізації.....	48
3.2. Огляд архітектури, компонентів та підходів до розробки.....	50
3.3. Розробка бази даних проєкту	56

3.4. Огляд алгоритмів	60
3.5. Опис інтерфейсу користувача.....	64
РОЗДІЛ 4	
ПЛАНУВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ ПРОЄКТОМ	
4.1. Структура декомпозиції робіт по проєкту.....	70
4.2. Організаційна структура та формування команди проєкту.....	72
4.3. Вибір підходу до розробки програмного забезпечення	76
4.4. Календарно-ресурсне планування.....	78
4.5. Оцінка витрат проєкту.....	82
4.6. Управління якістю проєкту	84
4.7. Управління ризиками проєкту	88
ВИСНОВКИ.....	93
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	96
Додаток А Логіко-структурна схема	101
Додаток Б Аналіз зацікавлених сторін.....	103
Додаток В Реалізація окремих алгоритмів на мові програмування С#	105
Додаток Г Управління якістю проєкту	107

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

“Дослідження процесів управління проектом розробки програмного забезпечення для цифрової обробки зображень”

Студента: Стельмаха Олега Юрійовича.

Науковий керівник: Стешенко Григорій Миколайович.

Рік захисту – 2024.

Метою підготовки даної кваліфікаційної роботи є дослідження процесів управління проектом з розробки та впровадження ПЗ для цифрової обробки зображень на основі концепції, проектної пропозиції, аналітичних досліджень, оцінки доцільності проекту, а також технічної складової створення ПЗ.

Об'єктом дослідження є процеси управління проектом розробки програмного забезпечення для цифрової обробки зображень.

Предметом дослідження є усі процеси управління проектом відповідно до етапів його життєвого циклу, управлінські і технічні рішення для успішної реалізації інноваційної ідеї проекту; програмне забезпечення для цифрової обробки зображень як продукт проекту.

Наукова новизна містить формалізацію задуму та концепції проекту, що пропонує новий підхід до створення продукту у вигляді технічного інструменту, усестороннє дослідження та аналіз складових проекту, процесів управління, побудову необхідних діаграм та моделей для успішної реалізації проекту.

За результатами роботи зроблено висновки щодо можливості створення проекту, виявлено, що напрямок цифрової обробки зображень є дуже перспективним, а проєкт має значний потенціал та велику ймовірність успішного впровадження, подальша розробка вважається доцільною та перспективною.

Робота містить 100 сторінок без додатків, 25 рисунків, 14 таблиць, 45 джерел. Додатки складають 11 сторінок.

***Ключові слова:** цифрова обробка зображень, штучний інтелект, процеси управління проектом, розробка програмного забезпечення, календарно-ресурсне планування, математична модель, управлінські рішення, аналіз якості і ризиків.*

ПЕРЕЛІК ПОЯСНЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект.

DIP (Digital Image Processing) – цифрова обробка зображень.

SaaS (Software as a Service) – програмне забезпечення як послуга.

IaaS (Infrastructure as a Service) – інфраструктура як послуга.

PaaS (Platform as a Service) – платформа як послуга.

ПЗ – програмне забезпечення.

API (Application Programming Interface) – прикладний програмний інтерфейс.

CRI (Color Rendering Index) – індекс передавання кольору.

B2B (Business-to-Business) – бізнес до бізнесу.

B2C (Business-to-Consumer) – бізнес для споживача.

RBM (Results-Based Management) – управління орієнтоване на результати.

CI (Continuous Integration) – безперервна інтеграція.

CD (Continuous Delivery) – безперервна доставка.

ORM (Object-Relational Mapping) – об'єктно-реляційне відображення.

MVC (Model-View-Controller) – модель-представлення-контролер.

TDD (Test-Driven Development) – розробка через тестування.

DI (Dependency Injection) – впровадження залежностей.

IoC (Inversion of Control) – інверсія керування.

TBA (Token-Based Authentication) – автентифікація на основі токену.

JWT (JSON Web Token) – JSON веб токен.

СКБД – система керування базами даних.

SQL (Structured Query Language) – мова структурованих запитів.

WBS (Work breakdown structure) – структура декомпозиції робіт.

OBS (Organization Breakdown Structure) – ієрархічна структура організації.

SEO (Search Engine Optimization) – оптимізація для пошукових систем.

ПРЗ – протиризикові заходи.

ВСТУП

Цифрова обробка зображень – це галузь знань, яка докорінно змінила спосіб взаємодії та аналіз зображень у різних сферах застосування. Вона передбачає використання комп'ютерних алгоритмів для виконання широкого спектру операцій над цифровими зображеннями, що дозволяє покращувати, аналізувати та витягувати цінну інформацію з візуальних даних. Основна мета цифрової обробки – покращення візуального вигляду зображення, отримання корисної інформації та підготовка зображення до виконання інших специфічних цілей. В останні роки в області цифрової обробки зображень відбувся значний стрибок завдяки швидкому розвитку комп'ютерного апаратного забезпечення та програмного забезпечення. Інтеграція алгоритмів штучного інтелекту (англ. artificial intelligence) і машинного навчання (англ. machine learning) ще більше розширила можливості систем на базі цифрової обробки, дозволяючи їм виконувати такі складні завдання, як розпізнавання та класифікація зображень з високою точністю, а область застосування таких методів величезна і постійно розширюється. Проте, розробка таких програмних продуктів – складний та мінливий процес, що вимагає високої ефективності управління. В контексті цієї складності і наростаючих вимог до якості та швидкості, ефективне застосування процесів управління проектом дозволяє зрозуміти та оптимізувати ключові аспекти реалізації, забезпечуючи успішну та ефективну реалізацію проектів у цій галузі.

Актуальність теми: зважаючи на велику потребу та значне застосування даного виду програмного забезпечення актуальним також є розробка нових і покращення існуючих методів. Разом з цим тісно пов'язана тема реалізації таких проектів, що дозволить зробити якісний стрибок в технологіях, методах та алгоритмах. Управління проектами в контексті цифрової обробки зображень має надважливе значення через швидкий розвиток та поширення у різних галузях. Застосування методів цифрової обробки зображень в промисловості, медицині, медіа індустрії, мистецтві, військовій справі та інших сферах вимагає

ефективного управління для успішної інтеграції цих технологій. Дослідження цієї теми відкриває шлях до розуміння кращих практик управління проєктами із розробки комплексного програмного забезпечення у галузі цифрової обробки та впровадження новаторських підходів для подальшого покращення процесів планування, розробки та впровадження.

Метою проєкту є успішне створення та впровадження інформаційної системи для цифрової обробки зображень з інноваційною архітектурою та застосуванням методів штучного інтелекту для надання послуг B2B та B2C сегментам клієнтів і забезпечення бізнес-потреб.

Завдання:

- виконати аналіз предметної області з процесів управління проєктом та цифрової обробки зображень, оцінити доцільність та життєздатність проєкту;
- виконати аналіз процесів управління проєктом та аналіз науково-інформаційних джерел, огляд останніх досліджень та публікацій;
- визначити проблематику, запропонувати шляхи вирішення, сформулювати наукову новизну та інноваційність проєкту;
- виконати огляд ІТ ринку цифрової обробки зображень, визначити ключові напрямки розвитку та перспективи;
- формалізувати ідею, створити паспорт та опис продукт проєкту;
- виконати аналіз зацікавлених сторін та оточення проєкту;
- проаналізувати та порівняти проєктні альтернативи;
- розробити концептуальні моделі інформаційної системи, підготувати формальне представлення у вигляді математичних моделей;
- описати програмне забезпечення, його структуру, обґрунтувати архітектурні рішення та перелік технологій для його реалізації;
- розробити базу даних для продукту, описати використані рішення;
- визначити ієрархічну структуру робіт, виконати огляд планування та супроводу виконання проєкту;

- описати організаційну структуру та обґрунтувати вибір підходу для управління проектом та розробки програмного забезпечення;
- виконати календарно-ресурсне планування проекту;
- визначити ключові аспекти якості та ризику, що впливають на виконання проекту, визначити шляхи для управління ними;
- сформулювати висновки, описати результати та подальші кроки дослідження.

Об'єктом дослідження є процеси управління проектом розробки програмного забезпечення для цифрової обробки зображень.

Предметом дослідження є усі процеси управління проектом відповідно до етапів його життєвого циклу, управлінські і технічні рішення для успішної реалізації інноваційної ідеї проекту; програмне забезпечення для цифрової обробки зображень як продукт проекту.

Методи дослідження: для виконання дослідження актуальним є використання різних методів та підходів, зокрема методів аналізу, експерименту, узагальнення, прогнозування, математичного та комп'ютерного моделювання. Крім того, для виконання окремих завдань використовуються методи декомпозиції, критичного шляху, SWOT та PEST-аналіз, перевірка концепції проекту, створення концептуальних моделей, порівняння різних підходів до розробки програмного забезпечення, визначення та аналіз метрик проекту, методи оцінки якості та визначення і протидії ризикам проекту.

Новизна отриманих результатів: дослідження пропонує перелік процесів управління проектом із розробки програмного забезпечення, їхнє удосконалення, інноваційну концепцію проекту та новий підхід до створення продукту – програмного забезпечення для цифрової обробки зображень як технічного інструменту з великою кількістю параметрів, налаштувань та сценаріїв автоматизації. Пропонується застосування комплексних методів обробки зображень у поєднанні з можливостями штучного інтелекту, розширюваність, адаптивність та інтеграція з сторонніми сервісами та програмами, а також орієнтація на вирішення нетривіальних завдань.

Практичне значення отриманих результатів полягає у виконанні дослідження усіх процесів із розробки програмного забезпечення для цифрової обробки зображень, що являє собою приклад реалізації проєкту та успішного застосування управлінських рішень та підходів у галузі розробки комплексних високонавантажених інформаційних систем. Всебічне дослідження продукту проєкту вказує на те, що він пропонує інноваційний підхід, який може бути успішно застосований партнерами та клієнтами при розробці їхніх власних програмних рішень. Виконані дослідження та огляд інформаційної системи також може використовуватись для виявлення важливих вимог і чинників для подальшого планування, коригування вимог до продукту, оптимізації процесів розробки і управління, та як підтвердження успішності, життєздатності й доцільності реалізації проєкту.

РОЗДІЛ 1

АНАЛІТИЧНІ ДОСЛІДЖЕННЯ ТА ОЦІНКА ДОЦІЛЬНОСТІ І ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1. Аналіз предметної області

Цифрова обробка зображень – це область обробки сигналів та обробки даних, яка вивчає методи обробки та аналізу зображень за допомогою обчислювальних та інформаційних систем. Основна ідея полягає в тому, щоб використовувати алгоритми та програми для виявлення, аналізу, покращення та відтворення зображень, які мають цифрову форму. Методи цифрової обробки зображень (англ. digital image processing) мають надзвичайно важливе значення в сучасній науці та є одними з таких, які постійно розвиваються та вдосконалюються.

Метою обробки зображень може бути як покращення його якості в залежності від певних критеріїв (відновлення зображення, реставрація), так і спеціальне перетворення, що кардинально змінює зображення в залежності від мети обробки. Обробка зображень може бути використана для того, щоб отримати видиме зображення цифрових даних, посилене певним чином для виділення деяких аспектів даних. Поліпшення і обробка зображення може здійснюватися для сприйняття людиною (щоб стало "краще" з суб'єктивної точки зору), для сприйняття комп'ютером (для спрощення подальшого розпізнавання чи інших цілей), для створення спецефектів (отримати естетичне задоволення від красивого ефекту).

Ціль обробки зображень досягається за допомогою різних методів та алгоритмів, наприклад таких як покращення, відновлення, стиснення, сегментація та інші. Методи покращення зображення спрямовані на покращення якості зображення шляхом регулювання його контрастності, яскравості та різкості. Вдосконалені алгоритми вирішують такі проблеми, як шум під час

захоплення зображення, розмиття, спричинене тремтінням камери, або погане освітлення. Методи зменшення шуму, такі як фільтрація, можуть усунути небажані артефакти, тоді як алгоритми підвищення різкості покращують краї та деталі. Регулювання контрастності може перетворити пласке, неживе зображення на зображення з глибиною та яскравістю. Ці вдосконалення не лише покращують естетику зображення, але й полегшують його інтерпретацію людиною, що дуже важливо для таких завдань, як, наприклад, медична діагностика.

Методи відновлення зображень використовуються для відновлення зображень, які були пошкоджені шумом, розмиттям або іншими факторами. Одним з прикладів завдання, яке вимагає відновлення цифрових зображень, є відновлення зображень, пошкоджених внаслідок стиснення з втратами. При виконанні цієї операції деяка інформація втрачається, що може призводити до зниження якості зображення. Наприклад, після стиснення JPEG зображення може мати артефакти чи розмитості. Методи відновлення деталей можуть бути використані для відновлення втрачених деталей, тоді як фільтри для зменшення шуму можуть допомогти зменшити шум, який може з'явитися через стиснення. Іншим прикладом може бути відновлення старих фотографій, зроблених на плівці або на ранніх цифрових камерах, що можуть з часом погіршуватися через вицвітання, подряпини або інші пошкодження. Методи відновлення зображень, такі як інтерполяція та екстраполяція, можуть бути використані для заповнення відсутніх або пошкоджених даних на таких фотографіях, роблячи їх більш чіткими та покращуючи їхню загальну якість. Загалом, такі методи важливі у відео- та фотостудіях, де важлива висока якість візуального контенту. Також це може бути важливо в медичній діагностиці, де точність зображень може мати вирішальне значення.

Методи стиснення зображень мають важливе значення для зменшення обсягу пам'яті необхідного для зберігання, пропускної здатності каналу та часу передачі, зберігаючи при цьому прийнятний рівень якості зображення. Це є критично важливим у завданнях передачі та зберігання цифрових зображень в

інтернеті, зокрема, як приклад, стиснення використовується для потокового відео, для зберігання зображень в цифрових книгах, для зберігання текстур та інших візуальних елементів у відеоіграх.

Методи сегментації зображення використовуються для поділу зображення на значущі частини, що має вирішальне значення для різних завдань, таких як, наприклад, виявлення чи розпізнавання об'єктів у безпілотних системах автомобілів або ідентифікація певних типів клітин на зображенні під мікроскопом. Алгоритми потім класифікують ці сегментовані об'єкти на основі задалегідь визначених критеріїв. Наприклад, система розпізнавання обличчя може сегментувати обличчя конкретної людини з натовпу, а потім класифікувати його на основі збережених даних.

Сфери застосування методів цифрової обробки в наш час значною мірою розширюються, майже повністю витіснивши аналогові методи обробки сигналів зображень. Вони широко використовуються в промисловості, мистецтві, медицині, космосі та при керуванні процесами автоматизації виявлення об'єктів, при розпізнаванні образів і в багатьох інших завданнях [8]. Формування зображень, поліпшення якості та автоматизація обробки медичних зображень, в тому числі зображень, що створюються електронними мікроскопами, рентгенівськими апаратами, томографами тощо, є предметом сучасних досліджень та розробок. Автоматичний аналіз у системах дистанційного спостереження широко застосовується при аналізі місцевості, у лісовому господарстві, наприклад, для автоматичного підрахунку площі вирубок, у сільському господарстві для спостереження за дозріванням урожаю, у розвідці, у системах протипожежної безпеки [8]. Сьогодні важко уявити сферу діяльності, у якій можна обійтися без комп'ютерної обробки зображень. При комп'ютерній обробці зображень вирішується широке коло різних завдань, а автоматизація за допомогою алгоритмів обробки зображень підвищує ефективність і скорочує ручні зусилля та час необхідний для виконання поставлених виробничих завдань. Це не тільки спрощує роботу, але й сприяє покращенню контролю якості, дозволяючи ідентифікувати дефекти та невідповідності в режимі реального часу.

У зв'язку з широким застосуванням програмного забезпечення для цифрової обробки зображень, важливо розвивати нові та вдосконалювати існуючі методи. Ефективне управління проєктами у цій галузі має вирішальне значення для успішної інтеграції цифрової обробки зображень у різних сферах. Зокрема, воно обов'язково включає такі етапи як аналіз потреб користувачів, визначення вимог до програмного забезпечення, розробка та впровадження алгоритмів обробки зображень, тестування та оптимізація продукту. Аналіз предметної області показує, що розвиток сфери цифрової обробки зображень породжує потребу у постійному удосконаленні і розробці нових методів та програмного забезпечення. Це відкриває широкі можливості для досліджень у галузі управління проєктами, спрямованих на оптимізацію процесів розробки та впровадження програм для цифрової обробки зображень.

1.2. Огляд останніх досліджень та публікацій

При ретельному дослідженні науково-інформаційної літератури пов'язаної із застосуванням методів цифрової обробки зображень для вирішення практичних цілей було виділено 5 досліджень, що стосуються застосування у різних цілях і галузях та описують різну проблематику реалізації.

Зокрема, першою серед них є стаття [7], що описує застосування алгоритмів цифрової обробки при реалізації проєкту розробки системи підтримки прийняття рішень для галузі сільського господарства. Висновком є те, що для ефективного управління посівними площами сільське господарство потребує використання нових інформаційних технологій, зокрема геоінформаційних систем та GPS. Недоліками таких систем є висока вартість, необхідність експертного аналізу зображень полів. Сьогодні малі літаки або дрони можуть створювати високоякісні цифрові фотографії. Ці зображення є вхідними даними для спеціалізованих програмних систем, які аналізують зображення та дають думку про стан ґрунтів, посівів тощо. Актуальним

завданням є розробка методів аналізу таких цифрових зображень для встановлення та прогнозування рівня врожайності, виявлення ділянок пересушування ґрунту, забур'яненості, ураженості рослин тощо. Інтеграція цих методів у геоінформаційні системи створять багатофункціональну систему підтримки прийняття рішень у сільському господарстві.

У другому дослідженні [26] порушується питання того, що проблеми сегментації цифрового зображення представляють великі виклики для комп'ютерного зору та наводяться приклади рішень деяких проблем. У статті розглядаються та узагальнюються основні методи сегментації цифрових зображень та наводиться їх класифікація, переваги та недоліки кожного. Це дослідження корисне для визначення належного використання методів сегментації зображення та для покращення їх точності та продуктивності, а також для досягнення основної мети, а саме розробки нових алгоритмів.

В свою чергу, у третій роботі [28] авторами було розглянуто тему класифікації історичних карт. У цій статті обговорюється вичерпний аналіз переваг і недоліків багатьох методів обробки супутникових зображень. Авторами виконано декілька порівняльних аналізів, щоб показати придатність кількох методів. Результатом дослідження є перелік інноваційних рішень для різних проблем, пов'язаних з обробкою супутникових знімків, що можуть бути використані при плануванні і розробці відповідних проєктів.

У четвертому дослідженні [27] розглядається автоматична сегментація коронарної артерії на цифровому зображенні коронарної комп'ютерної томографічної ангіографії (англ. ССТА). В результаті було представлено новий фреймворк сегментації зображень, який поєднує глибоке навчання та алгоритми цифрової обробки зображень для вирішення проблем, з якими стикається галузь, що доводить результативність інноваційних підходів у медичній галузі.

У п'ятій роботі [30] авторами описується вплив методів попередньої обробки та постобробки цифрових зображень на основі фреймворків глибокого навчання у сфері діагностики патологій. Зокрема у роботі наводиться такий факт, що застосування методів глибокого навчання швидко стало основною

методологією для аналізу медичних зображень завдяки хорошій здатності до навчання та перевагам у роботі зі складними шаблонами. Дослідження показали, що інтеграція методів цифрової обробки зображень в моделі глибокого навчання може ще більше підвищити продуктивність і точність, а багато методів, представлених у роботі, особливо методи постобробки, не обмежуються сферою діагностики патологій, а можуть бути застосовані практично до будь-якої області аналізу цифрових зображення.

Проведене дослідження науково-інформаційної літератури, що стосується застосування методів цифрової обробки зображень, підтверджує актуальність застосування таких технологій у різних галузях та підкреслює наявність проблематики у предметній області і потенційних шляхів її вирішення. На основі аналізу існуючих рішень можна зробити висновок про те, що методи цифрової обробки зображень можуть досягати хороших результатів при реалізації інформаційних систем та іншого програмного забезпечення, проте в успіху проєкту окрім технічної складової велику роль також відіграє управління проєктами, зокрема ретельне планування, оцінка ризиків, контроль та моніторинг.

1.3. Огляд ринку цифрової обробки зображень

У 2023 році обсяг світового ринку цифрової обробки зображень оцінювався в 6,2 мільярда доларів США і, як очікується, зростатиме зі середньорічним темпом зростання (англ. CAGR) 19,8% у період з 2024 по 2033 рік. Очікується, що до 2033 року обсяг світового ринку цифрової обробки зображень становитиме близько 37,5 мільярдів доларів [37]. Консалтингові компанії стверджують, що ринок цифрової обробки зображень зазнав значного зростання завдяки зростаючому попиту на передові технології обробки зображень і все більш широкому впровадженню рішень для обробки цифрових зображень в різних галузях. З широким розповсюдженням цифрових пристроїв і

стрімким зростанням обсягів цифрових зображень виникла потреба в ефективних інструментах і методах для обробки та аналізу цих зображень. Також, в останні роки ринок зазнав значного зростання та змін, зумовлених збільшенням попиту на високоякісний візуальний контент та розвитком технологій штучного інтелекту. У сучасному світі, що стрімко розвивається в напрямку цифрової трансформації, програмне забезпечення для цифрової обробки зображень відіграє ключову роль у різних галузях, починаючи від медицини та закінчуючи медіа індустрією. Наприклад, очікується, що у 2024 році світовий попит у сфері медичної візуалізації зросте на ~20% порівняно з 2023 роком. Таке зростання свідчить про важливість технологій для підвищення точності та ефективності медичної діагностики.

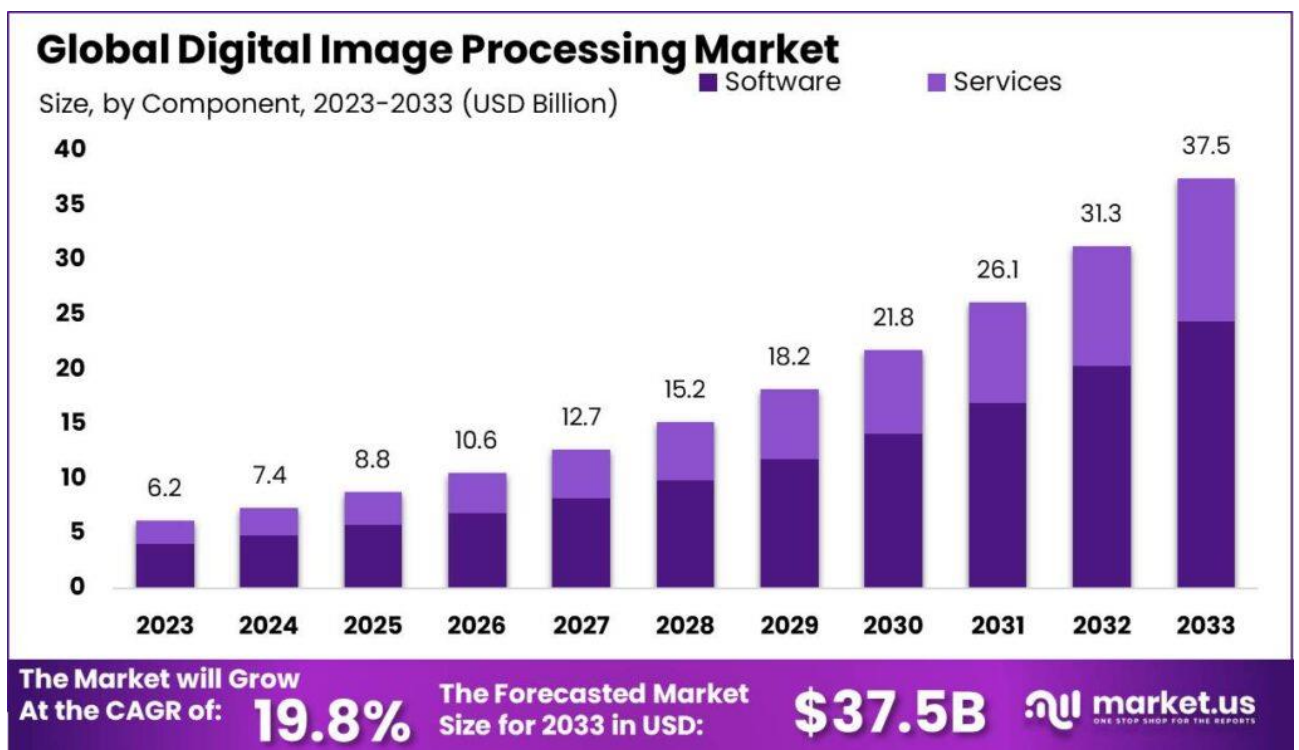


Рис. 1.1. Аналіз глобального ринку цифрової обробки зображень [37]

Як стверджується, найбільш вагомим рушієм для ринку цифрової обробки зображень є поширення технологій штучного інтелекту та машинного навчання. Ці технології покращують здатність систем обробки зображень аналізувати, інтерпретувати та приймати рішення на основі візуальних даних. Інтеграція цих

технологій не тільки підвищує точність і ефективність обробки зображень, але й відкриває нові можливості для інновацій та оскільки вони продовжують розвиватися, очікується, що їх застосування в обробці цифрових зображень буде розширюватися, сприяючи подальшому зростанню ринку.

Ринок поділяється на два основних сегменти – програмне забезпечення (англ. Software) та послуги (англ. Services). Сегмент програмного забезпечення лідирував на ринку у 2023 році, на нього припадало близько 65% світового доходу. Це можна пояснити зростаючим попитом на програмні рішення для обробки зображень у різних галузях економіки, прогресом у програмних технологіях, а також гнучкістю та адаптивністю програмних рішень для вирішення широкого спектру потреб обробки зображень. Крім того, сегмент ПЗ значно зріс завдяки інтеграції алгоритмів штучного інтелекту та машинного навчання, які ще більше розширили можливості ПЗ для обробки зображень, що робить його кращим вибором для підприємств та організацій, які шукають ефективні та адаптивні рішення для своїх потреб в обробці зображень.

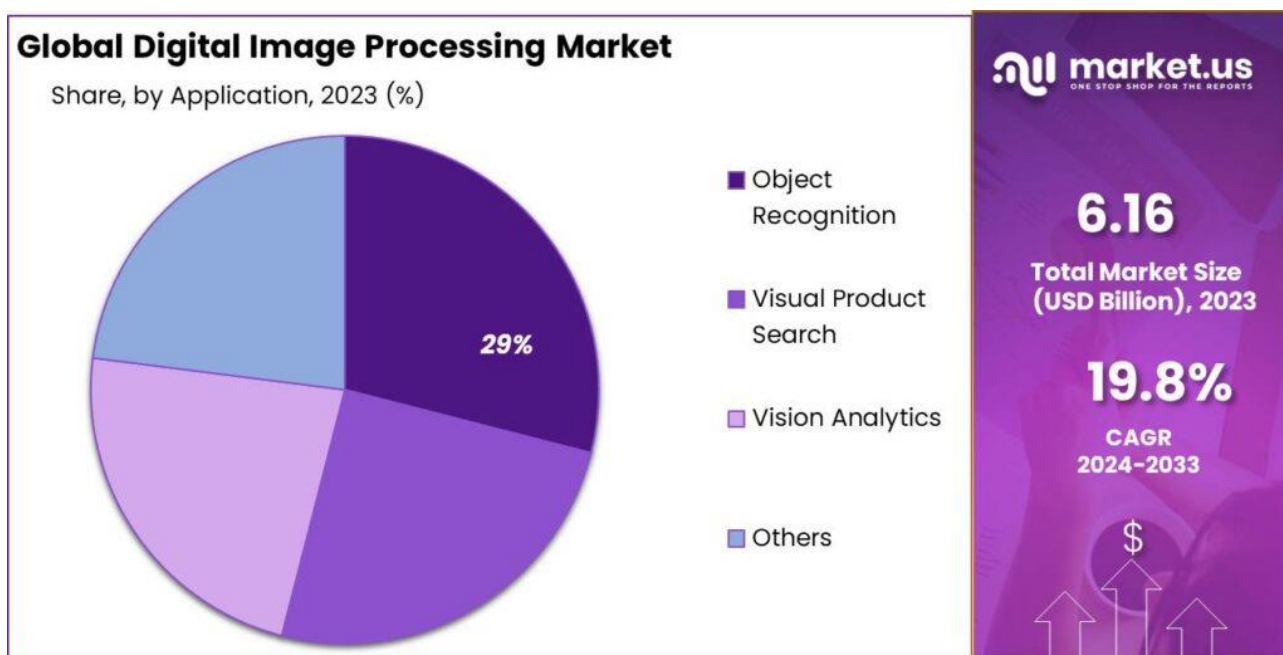


Рис. 1.2. Розподіл ринку за варіантами використання цифрової обробки зображень [37]

Зокрема, у 2023 році серед варіантів застосування домінуючу позицію на ринку займав саме сегмент розпізнавання об'єктів з часткою ринку понад 29%. Наприклад, у роздрібній торгівлі візуальний пошук товарів на основі розпізнавання об'єктів дозволяє клієнтам знаходити продукти лише за фотографією, що значно покращує і спрощує процес шопінгу.

Проте, також прогнозується значне зростання сегменту послуг у найближчі роки завдяки складності та різноманітності завдань з обробки зображень. Консалтингові послуги користуються попитом у зв'язку зі зростанням потреб в експертних рекомендаціях і стратегічних порадах. Багато організацій віддають перевагу аутсорсингу цих завдань експертним постачальникам послуг, які можуть надавати постійні результати та адаптуватися до мінливих вимог. Крім того, такі послуги часто пропонують економічно ефективний підхід, оскільки усувають необхідність власної розробки та підтримки цих рішень.

Однією із головних перешкод, з якими стикається ринок, є висока вартість, пов'язана з впровадженням сучасних систем. Ці витрати включають не лише початкові інвестиції в програмне та апаратне забезпечення, але й операційні витрати, пов'язані з обслуговуванням, оновленнями та іншими чинниками. Крім того, складність інтеграції цих систем в існуючу інфраструктуру може призвести до подальшого зростання витрат, що ще більше ускладнює цей процес для деяких організацій. Цей фінансовий бар'єр може перешкоджати широкому впровадженню рішень для цифрової обробки зображень у різних галузях, впливаючи на загальне зростання ринку.

Зі свого боку, щодо можливостей, ринки, що розвиваються, представляють значні можливості для галузі обробки цифрових зображень. Оскільки країни в Азійсько-Тихоокеанському регіоні, Латинській Америці та Африці, продовжують стрімке економічне зростання та цифрову трансформацію, очікується, що попит на передові технології обробки зображень в цих регіонах також швидко зростатиме.

Серед ключових гравців на ринку, які зарекомендували себе як лідери, варто відзначити Nvidia Corporation, Canon Inc., Panasonic Corporation, Samsung

Electronics, Microsoft, IBM, Adobe Systems, Teledyne Technologies, Cognex Corporation. Саме вони просуваються інновації, стратегічні ініціативи та встановлюють галузеві стандарти.

Підсумовуючи, зростаюча увага до якості зображень, інтеграція передових технологій штучного інтелекту та методів обробки цифрових зображень є важливими факторами, що сприятимуть швидкому розширенню світового ринку галузі у наступні роки.

1.4. Вивчення проблематики предметної області

Застосування методів цифрової обробки зображень приносить інновації, проте його застосування також обмежене наявними проблемами предметної області. Більшість з них напряму пов'язані з якістю отриманого результату та швидкістю роботи програмного забезпечення, що виконує обробку зображень. Проте дуже частою перешкодою впровадження будь-якого методу є складність його реалізації, що напряму пов'язана з тим, що не існує універсальних рішень, які б підходили для всіх задач. Кожна задача може вимагати власного унікального методу обробки зображення, оскільки вона може відрізнитися за різними параметрами, такими як розмір, роздільна здатність, характеристики шуму та багато інших. Тому розробка власного методу для кожної задачі є складним процесом, який вимагає глибокого розуміння принципів обробки зображень та технічної експертизи. Також варто відзначити наступні чинники, що значною мірою впливають на успішність проєктів з цифрової обробки зображень:

- ефективність алгоритмів – розвиток швидких і ефективних алгоритмів для обробки статичних зображень, так і для обробки в реальному часі, наприклад, у системах відеоспостереження;

- розробка надійних методів обробки зображень, які можуть протистояти зловмисним атакам, де ледь помітні зміни в зображенні можуть

обдурити системи штучного інтелекту, з акцентом на підвищення надійності ПЗ, що використовує штучний інтелект для цифрової обробки зображень;

– жодне існуюче рішення не може гарантувати 100% правильності та надійності отриманого результату;

Досить ваговою проблематикою також є труднощі пов'язані з управлінням проєктами. Наприклад, проєкти можуть вимагати швидкого впровадження або високої точності в обробці даних. У високотехнологічних проєктах у цій області існує велика кількість потенційних ризиків, таких як технічні проблеми, втрата даних або конфіденційність. Важливим є необхідність інтеграції з іншими системами, так як методи цифрової обробки у більшості випадків використовуються у системах, які взаємодіють з іншими системами або обладнанням. Та управління командою потребує здатності координувати роботу висококваліфікованих фахівців з різних областей. Завжди існує ризик збільшення термінів та витрат через не гарантованість успішності досліджень та експериментів. Управління проєктами у галузі цифрової обробки зображень вимагає глибокого розуміння технічних аспектів та вміння ефективно керувати процесами розробки програмного забезпечення та досліджень. Важливо враховувати всі аспекти проблематики для успішного виконання проєктів у цій області.

З урахуванням наведених проблем та завдань роботи надалі буде виконано опис проєкту розробки програмного забезпечення для цифрової обробки зображень, що враховуватиме особливості предметної області та потреби користувачів. Також будуть розглянуті підходи до управління проєктами у галузі цифрової обробки зображень, з формалізацією задуму ПЗ, урахуванням ризиків, вимог ефективності та надійності програмного забезпечення.

1.5. Наукова новизна та інноваційність

Застосування методів цифрової обробки зображень у комплексі з штучним інтелектом є складним технічним та математичним завданням, що потребує проведення фундаментальних досліджень нових принципів, методів, теорій та моделей для обробки зображень. Проєкт пропонує новий підхід до створення редакторів цифрових зображень у вигляді технічного інструменту, що дозволяє максимально гнучко налаштувати його під потреби замовника чи партнера та надає функціонал з великою кількістю параметрів, налаштувань та сценаріїв автоматизації.

Також серед функціоналу програмного забезпечення передбачено:

- розширені можливості обробки – широкий спектр функцій та ефектів для покращення та трансформації зображень, включаючи кадрування, корекцію кольору, видалення об'єктів, застосування фільтрів та інше;
- реалізація методів на основі штучного інтелекту для автоматизації та поліпшення обробки зображень, наприклад, для видалення шуму, підвищення роздільної здатності, кольоризації, ресторації, виявлення об'єктів тощо;
- розширюваність – можливість додавати новий функціонал до програми за допомогою плагінів, що дозволить користувачам розширювати функціональність без зміни основного коду та наявність документованого API;
- адаптивність – можливість змінювати параметри та налаштування програми за допомогою конфігураційних файлів, що дозволяє адаптувати програму до різних вимог та потреб користувачів;
- інтеграція із сторонніми сервісами та програмами, що дозволить користувачам ефективно і гнучко досягати поставлених завдань.

Наукова новизна роботи з точки зору управління проєктом включає обґрунтування доцільності проєкту та його життєздатності, моделі дерева причин та наслідків, визначення цілей, структурні моделі організації проєкту, ієрархічні структури декомпозиції робіт, розрахунок очікуваної вартості,

математичні моделі та їх аналіз, а також календарно-ресурсне планування, планування архітектури, підбір технологій для реалізації, проектування бази даних, огляд графічного інтерфейсу, планування і методи контролю якості та управління ризиками у проєктах такого типу.

Основна увага при розробці продукту проєкту сфокусована на забезпеченні інноваційності у трьох напрямках: простота та ефективність застосування комплексних методів обробки зображень у поєднанні з можливостями штучного інтелекту, розширюваність, адаптивність та інтеграцію з сторонніми сервісами та програмами, а також орієнтація на вирішення нетривіальних завдань та надання інструменту для створення нових методів та підходів до обробки зображень. Принципово новим є підхід, який дозволяє клієнтам розробити власне рішення, інтегрувати його як плагін програми, а потім автоматично використовувати його у власному додатку через прикладний програмний інтерфейс.

Для забезпечення цінності продукту для користувача головний фокус при визначенні подальших напрямків розвитку буде спрямований на результатах аналізу отриманих даних щодо застосування продукту користувачами (англ. Usage Statistics). Вони будуть використовуватись для оцінки ефективності проєкту та прийняття рішень на основі об'єктивних даних та зворотного зв'язку користувачів. Це може зробити процес управління проєктом більш обґрунтованим і ефективним. Для реалізації цього ефективним підходом буде застосування статистичного аналізу, моделей та алгоритмів для прогнозування використання, зокрема машинного навчання, та методів візуалізації отриманих даних.

1.6. Постановка задачі дослідження

З урахуванням сформульованої мети в даній кваліфікаційній роботі необхідно вирішити низку наступних задач:

- розробити проєкту пропозицію, формалізувати ідею та задум проєкту, порівняти альтернативи, навести загальну характеристику відповідно до дослідження предметної області, виявленої проблематики та аналізу ринку;
- виконати аналіз зацікавлених сторін та оточення проєкту та інші дослідження;
- виконати формальний опис ІТ системи та подати процес функціонування системи у вигляді схем і математичних моделей, описати методи прийняття управлінських рішень;
- виконати аналіз технологій необхідних для розробки продукту, обґрунтувати доцільність їхнього використання;
- розробити схему взаємодії між компонентами інформаційної системи, діаграми структури бази даних та інші технічні елементи системи;
- виконати опис процесів управління проєктом, їх призначення та визначити найбільш оптимальний підхід до розробки програмного забезпечення, що відповідатиме вимогам проєкту та його меті;
- визначити перелік робіт та оцінити ресурси (людські, матеріальні, фінансові) необхідні для реалізації проєкту, створити календарний план, обрахувати вартість виконання проєкту;
- визначити ключові аспекти якості та ризику, що впливають на виконання проєкту, визначити шляхи для управління ними;
- сформулювати висновки, описати результати та подальші кроки дослідження.

РОЗДІЛ 2

РОЗРОБКА ПРОЄКТНОЇ ПРОПОЗИЦІЇ ТА МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1. Формалізація ідеї та задуму проєкту

Стрімкий розвиток штучного інтелекту та апаратного забезпечення має значний вплив на всі сфери людської діяльності та дозволяє отримувати ще кращі результати при роботі з цифровими зображеннями. Головна ж мета цифрової обробки полягає у використанні таких операцій над зображенням, щоб результат виявився більш підходящим з точки зору конкретного застосування ніж початковий варіант. Головною ідеєю проєкту є розробка програмного забезпечення, що буде виступати інструментом для практичного використання широкого спектру функцій та методів в різних категоріях та з різними цілями, включаючи реалізацію методів на основі штучного інтелекту, методи сегментації, класифікації, відновлення, стиснення, виявлення об'єктів, розмиття, бінаризації, кольоризації та інші. Для користувачів та клієнтів цей проєкт може бути цікавим для цифрової обробки власних зображень через гнучкість та адаптивність програмного продукту, різноманітний функціонал та можливість реалізації власних рішень на його основі.

Для визначення реалістичних цілей проєкту спочатку потрібно виявити і вивчити проблеми та їх наслідки, які можливо та доцільно вирішити за допомогою реалізації проєкту. Ефективним інструментом для такого аналізу є розробка дерева причин та наслідків [11]. На рис. 2.1. показано основні проблеми, з якими стикаються компанії, що не використовують програмне забезпечення для цифрової обробки зображень у свої щоденній діяльності.

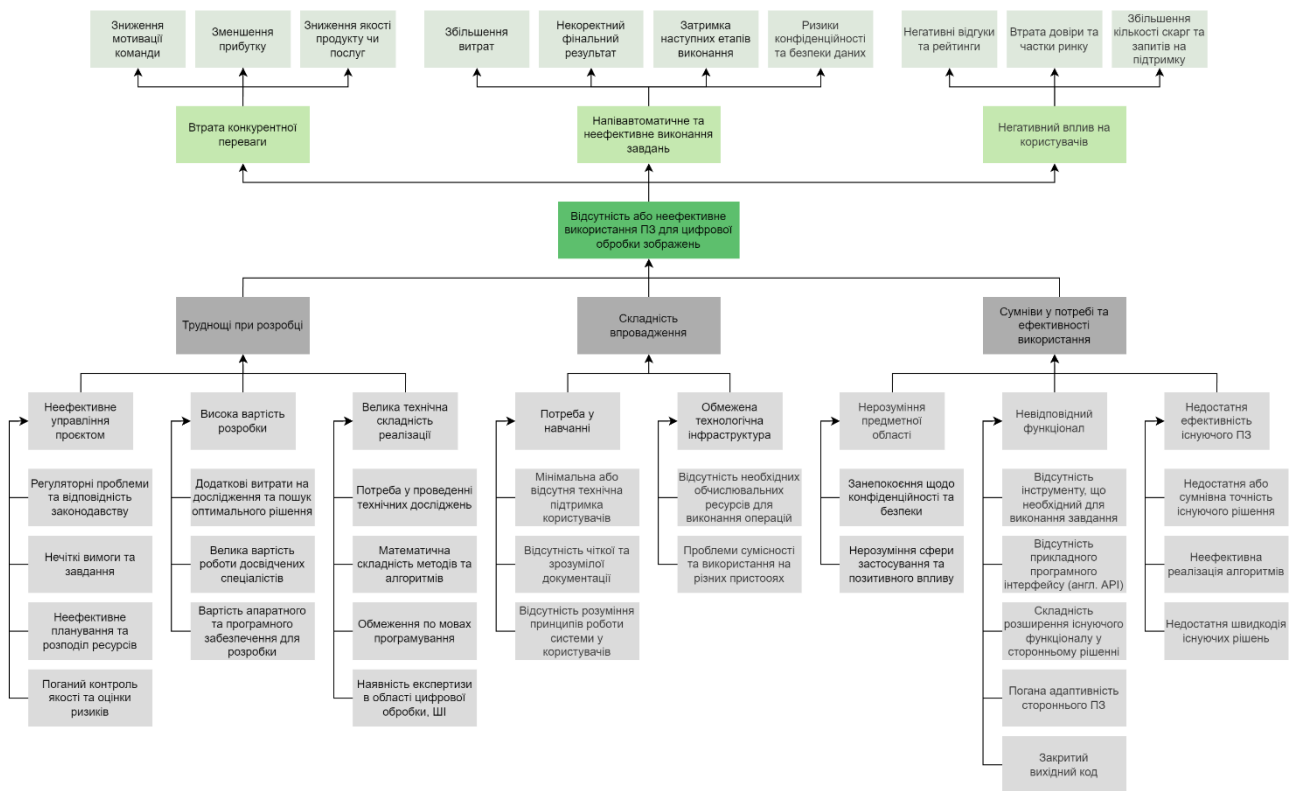


Рис. 2.1. Дерево причин та наслідків (проблем) проекту

Створення програмної системи, що надає такі послуги та має можливість інтеграції з програмним забезпеченням клієнтів може частково вирішити ці проблеми та створити додаткові конкурентні можливості для їхніх підприємств. Відповідно, можна розглянути проблеми через призму нових можливостей та чітко сформулювати основну мету, цілі проекту, ефект, якого має досягти проект та завдання, які має вирішити.

Метою даного проекту є створення програмного забезпечення для цифрової обробки зображень із застосуванням передових інформаційних технологій, таких як машинне навчання та сучасні алгоритми, для надання послуг B2B та B2C сегментам клієнтів і забезпечення бізнес-потреб.

Зі свого боку, постановка цілей допомагає у визначенні критеріїв успішності проекту, що є важливим для забезпечення відповідності проекту вимогам та очікуванням зацікавлених сторін, а також дозволяє уникнути невдач та непорозумінь під час реалізації проекту. Перед командою проекту стоятимуть наступні цілі:

- спроектувати базову архітектуру головних модулів, API та бази даних, спланувати їхню інтеграцію до початку другого кварталу 2024 року;
- розробити та впровадити мінімум 5 категорій та не менше ніж 50 базових функцій та методів для обробки зображень до кінця другого кварталу 2024 року;
- реалізувати та впровадити 10 алгоритмів на основі штучного інтелекту для автоматизації, класифікації та покращення обробки зображень до кінця другого кварталу 2024 року;
- додати функціонал вибору зони інтересу будь-якої форми для зображень у першій версії програми;
- реалізувати підтримку всіх популярних форматів зображень (JPEG, PNG, BMP, TIFF, RAW, HEIC) у першій версії програми;
- забезпечити відповідність програмного продукту міжнародним стандартам (GDPR [13], ISO/IEC 27001 [39]) у першій версії програми;
- реалізувати підтримку функціоналу графічних шарів та збереження послідовності застосованих методів зі значеннями параметрів до серпня 2024 року;
- забезпечити можливість зміни параметрів та налаштування програми за допомогою конфігураційних файлів у відповідності до потреб компанії-замовника до вересня 2024 року;
- розробити документоване API, що дозволить інтегрувати програмне забезпечення з 5 сторонніми сервісами та програмами до кінця 2024 року;
- розробити модель на основі нейронних мереж та комп'ютерного зору із системою вхідних параметрів, що здатна кольоризувати чорно-білі зображення із індексом кольоропередачі (англ. CRI) більше 70 у другій версії програми;
- залучити більше 10 000 активних зовнішніх користувачів за перших два місяці після релізу першої версії програми.

Усі сформульовані цілі було перевірено за допомогою методу SMART [14] для визначення того наскільки вони відповідають очікуванням від проекту та чи

може бути виконана оцінка ступеня їх досяжності. Наступним етапом аналізу цілей стала побудова логіко-структурної схеми, що розробляється для перевірки всіх цілей на відповідність загальній меті проєкту. Вона показана в табл. А.1.

Підсумовуючи, можна сказати, що цілі є реалістичними та відповідають загальній меті проєкту. Логіко-структурна схема є ефективним інструментом для визначення і забезпечення узгодженості та спрямованості усіх дій та завдань проєкту на досягнення його цілей та мети.

2.2. Аналіз зацікавлених сторін та оточення проєкту

Даний аналіз проводиться з метою визначення ключових зацікавлених сторін проєкту та їх потреб. Дослідження дозволяє оцінити вплив результатів проєкту на оточення та того як оточення може впливати на реалізацію проєкту. Крім того, важливо ідентифікувати потреби та вигоди зацікавлених сторін, щоб забезпечити максимальний успіх проєкту. Результати дослідження показано в табл. Б.1.

Загалом, можна зробити висновок, що внутрішні зацікавлені сторони – замовник, виконуюча організація та команда проєкту – мають найвищий рівень позитивного впливу на проєкт. Вони також є найбільш зацікавленими в успіху проєкту, тому важливо приділяти їм особливу увагу в комунікації. Конкуренти можуть мати негативний вплив, який може бути значним, оскільки вони можуть вдосконалювати свій продукт, що зробить вихід на ринок та просування більш складним завданням.

Для стратегічного аналізу зовнішнього середовища було використано PEST-аналіз [23] – інструмент, який дозволяє проаналізувати вплив політичних, економічних, соціальних та технологічних факторів на бізнес-середовище організації або на ринок в цілому. Результат аналізу показано в табл. 2.1.

PEST-аналіз проєкту

Політичні фактори		Економічні фактори	
Фактор	Оцінка впливу	Фактор	Оцінка впливу
Війна (бойові дії на території України)	-3	Рівень інфляції, безробіття, ВВП	-1
Ймовірна зміна податкової та трудової політики	-1	Курси основних валют	-1
Значний рівень корупції та бюрократизації	-2	Темпи зростання економіки	-0.5
Вступ до ЄС та НАТО	+3	Купівельна спроможність населення	-2
Підтримка ІТ інновацій на рівні держави	+2.5	Інвестиційний клімат в галузі	+1
Зміни в законодавстві, що регулюють сферу ІТ	+1	Ступінь глобалізації та відкритості економіки	+2.5
Лібералізація зовнішньоекономічного співробітництва	+2	Рівень розвитку підприємництва та бізнес середовища	+1
Світова політична нестабільність	-1	Вплив конкурентів	-2
Соціальні фактори		Технологічні фактори	
Фактор	Оцінка впливу	Фактор	Оцінка впливу
Рівень підготовки молодих спеціалістів в галузі	+2.5	Рівень інновації та технологічного розвитку галузі	+3
Вимоги до якості продукції та рівня сервісу	+3	Ступінь використання, впровадження та передачі технологій	+2
Рівень міграції та імміграційні настрої	-2	Доступ до новітніх технологій	+2.5
Темпи росту населення	-1.5	Витрати на дослідження і розробки	-2.5
Цифрова грамотність населення	+1	Розвиток конкурентних технологій	-0.5
Доступність комп'ютерів та смартфонів	+1.5	Зростання потужності комп'ютерів	+2

Загалом, PEST-аналіз свідчить про те, що проєкт із розробки програмного забезпечення для цифрової обробки зображень може зіткнутись з низкою викликів у політичній, економічній, соціальній та технологічній сферах.

Фактором, що спричиняє найбільший вплив є війна, яка прямо впливає і на економічну та соціальну сферу, проте за позитивного сценарію, Україна має великий шанс на швидке відновлення та інтеграцію в міжнародні організації та союзи. Доступ до технологій, розвиток галузі та ступінь впровадження технологій свідчать про те, що в країні є потенціал для створення успішних ІТ проєктів. Також Україна має високий рівень ІТ-освіти та талановитих фахівців, що є перевагою для розробки технологій та ПЗ в країні. З іншого боку, ІТ-галузь в Україні має деякі проблеми, зокрема, складну бюрократичну систему та низький рівень інвестицій. Однак, останнім часом уряд України активно сприяє розвитку галузі та створенню сприятливих умов для її функціонування, зокрема, через спеціальні програми підтримки ІТ-стартапів та інноваційних проєктів [22].

Також одним з ефективним інструментів є використання SWOT-аналізу [25] (аналізу сильних сторін, слабких сторін, можливостей та загроз), що має значну цінність для проєкту з різних причин, зокрема, дозволяє комплексно розглянути конкурентні переваги і можливості, виявити потенційні проблеми та загрози, а також розробити стратегію як мінімізувати вплив негативних факторів та покращити прийняття рішень. Аналізуючи ці фактори, можна отримати комплексне уявлення про стан проєкту і перспективи на майбутнє.

Таблиця 2.2

SWOT-аналіз проєкту

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> – Інноваційна бізнес-модель – Використання штучного інтелекту – Гнучкий підхід до управління проєктом – Масштабованість продукту – Зручний інтерфейс та стабільність продукту – Команда з висококваліфікованих фахівців – Відсутність реклами в продукті – Широкий спектр функцій та методів, інноваційні алгоритми – Кросплатформний продукт 	<ul style="list-style-type: none"> – Складність реалізації продукту – Значні витрати на команду – Відсутність репутації компанії – Перший продукт компанії – Обмеженість ресурсів – Ітераційне покращення якості продукту – Ймовірність зміщення термінів розробки – Значні ризики при плануванні проєкту – Складність тестування продукту

Можливості	Загрози
<ul style="list-style-type: none"> – Невелика кількість конкурентів у галузі – Вихід на світовий ринок – Залучення інвесторів – Розширення на ринки країн, що розвиваються – Швидке зростання ринку з середньорічним темпом зростання 19.8% та хороші перспективи – Стратегічні партнерства з клієнтами – Зростаючий попит на хмарні рішення – Поширення IoT і розумних пристроїв – Зростаючий попит на автоматизовану обробку зображень у реальному часі – Досягнення в технологіях комп'ютерного зору 	<ul style="list-style-type: none"> – Наявність сильних конкурентів зі схожим продуктом – Недостатня зацікавленість в продукті – Складність просування – Зміна актуальних технологій – Складність залучення коштів – Вплив форс-мажорних обставин на ринок – Поява нових конкурентів – Економічний спад

Загалом, SWOT-аналіз вказує на те, що сильні сторони продукту та можливості суттєво переважають основні слабкі сторони та загрози, головними серед яких є конкуренція, складність і велика вартість розробки. Важливим є мінімізація впливу слабких сторін, особливо за рахунок ретельного планування процесів управління проектом та специфікації вимог, а також реалізація можливостей.

Отже, за умови ретельного планування та врахування факторів, що мають негативний вплив, проєкт має великий потенціал і можливості для досягнення цілей та впровадження як на ринку України так і світовому ринку.

2.3. Аналіз проєктних альтернатив

Для визначення найкращого варіанту реалізації проєкту та його складових було використано аналіз проєктних альтернатив – процес порівняння та оцінки різних можливих варіантів реалізації проєкту для вибору найкращого з них.

В даному випадку основне питання полягало у виборі підходу до реалізації та платформи реалізації. Серед основних проєктних варіантів є розробка мобільного додатку, десктоп-додатку та веб-додатку. У табл. 2.3. наведено переваги та недоліки цих варіантів.

Таблиця 2.3

Переваги та недоліки проєктних альтернатив

Тип ПЗ	Переваги	Недоліки
Мобільний додаток	<ol style="list-style-type: none"> 1. Зручність використання 2. Полегшення персоналізації і реалізації системи сповіщення 3. Можливий офлайн доступ до деяких функцій 4. Зручність взаємодії та імпорту зображень 	<ol style="list-style-type: none"> 1. Потреба розробки окремих додатків під iOS та Android 2. Ненадійність та проблеми кросплатформних рішень 3. Значні комісії магазинів, де публікується додаток 4. Ймовірно дорожча розробка 5. Оновлення потребують перевірки та публікації в магазинах додатків 6. Обмеження операційних систем
Десктоп-додаток	<ol style="list-style-type: none"> 1. Використання апаратних ресурсів пристрою 2. Скорочення витрат на хмарні ресурси 3. Більш безпечний підхід з точки зору конфіденційності інформації 4. Можливість роботи без інтернету 	<ol style="list-style-type: none"> 1. Ризики пов'язані з неліцензійним використанням ПЗ 2. Труднощі оновлення та небажання користувачів це робити 3. Доступ лише з пристрою на якому встановлена програма 4. Обмеження апаратних ресурсів ПК 5. Потреба в розробці під різні платформи (Windows, MacOS, Linux)
Веб-додаток	<ol style="list-style-type: none"> 1. Висока швидкість виконання будь-яких операцій 2. Доступність з різних пристроїв 3. Незалежність від платформи та сторонніх магазинів додатків 4. Простота оновлення без залучення користувачів 5. Масштабованість та розподіл навантаження 6. Простота синхронізації даних 	<ol style="list-style-type: none"> 1. Потреба доступу до інтернету 2. Збільшені витрати на хмарні ресурси 3. Залежність системи від швидкості інтернету користувача 4. Обмежені можливості доступу до апаратного забезпечення 5. Ризики хакерських чи DDoS атак

На основі цієї таблиці можна виконати порівняння і обрати найбільш підходящий варіант. Відповідно до потреб та вимог проєкту, його цілей та мети

найбільш оптимальним варіантом буде розробка веб-додатку, що має найменш критичні недоліки порівняно з іншими альтернативами, а головне має більший перелік переваг, що значно перевершує інші варіанти саме в контексті розробки програмного забезпечення для цифрової обробки зображень.

2.4. Загальні характеристики проєкту

Місія: інноваційність та доступність програмного забезпечення для цифрової обробки зображень.

Візія: стати лідером на ринку програмного забезпечення для цифрової обробки зображень, завдяки передовим технологіям, зручному інтерфейсу та широкому інноваційному спектру функцій.

Цільова аудиторія поділяється на два базових сегменти.

– B2B – компанії, що бажають покращити свій продукт чи робочий процес використанням технологій цифрової обробки зображень або компанії, що уже використовують таке ПЗ, але шукають альтернативу чи покращення.

– B2C – інші індивідуальні користувачі, що бажають використовувати ПЗ у своїх цілях. До них відносяться: інтернет-користувачі, фотографи, студенти, дизайнери та інші.

Опис цінності: проєкт надає інноваційне програмне забезпечення як послугу (SaaS) [24] та дозволяє оптимізувати та прискорити процеси і операції цифрової обробки зображень, що відповідає сучасним вимогам та потребам. Сприяє розширенню можливостей обробки зображень як для окремих користувачів так і для компаній-клієнтів.

Продуктом проєкту із розробки програмного забезпечення для цифрової обробки зображень буде система, що складається з трьох основних частин: головного модуля у вигляді веб-системи, що має інтерфейс та надає основний функціонал, прикладного програмного інтерфейсу (англ. API), що використовується партнерами та іншими користувачами, та окремих плагінів, що

можуть бути встановлені незалежно один від одного та використовують наданий API. Плагіни надають функціонал у вигляді окремих наборів методів та функцій, що можуть бути зроблені під потреби клієнта або самим клієнтом. Ключовими характеристиками цього продукту будуть наступні:

- Зручність та доступність: додаток матиме простий та інтуїтивно зрозумілий користувацький інтерфейс, що дозволить користувачам легко додавати зображення, налаштовувати сценарії автоматизації, обирати необхідні функції і їх параметри. Він використовуватиме прості інструменти навігації та адаптивний інтерфейс, що враховуватиме різні розміри та роздільну здатність екранів, а також особливості доступності, такі як розміри шрифтів, контрастність та кольорові схеми (наприклад, темний режим). Система буде розроблятися із активним залученням кінцевих користувачів до процесу розробки та покращення ПЗ. Такий підхід дозволить команді розробників зрозуміти потреби та вимоги користувачів і розробити систему, яка відповідатиме їхнім конкретним потребам. Також особливістю є налаштування інтерфейсу під потреби компаній-партнерів, що дозволить зробити його унікальним та таким, що відповідає стилістиці і потребам конкретного підприємства.

- Передові алгоритми та машинне навчання: програмне забезпечення використовуватиме передові алгоритми, методи обробки цифрових зображень та методи машинного навчання для аналізу зображень, їх покращення та виконання інших визначних програмою операцій. Система буде розроблена таким чином, щоб постійно навчатися і вдосконалюватися на основі нових даних та відгуків від користувачів. ПЗ також буде регулярно оновлюватися з урахуванням останніх досліджень і рекомендацій, щоб забезпечити найсучасніші та найкращі результати.

- Якість: щоб гарантувати точність системи, вона буде ретельно протестована і перевірена з використанням великих масивів зображень і перевірених результатів. Тестування включатиме порівняння еталонних зображень із результатами отриманими із програми, загальну перевірку

людиною, а також аналіз роботи системи в різних сценаріях і на різних групах зображень.

– Користувацькі функції: програмне забезпечення матиме функції, що дозволять користувачам налаштовувати параметри та конфігурації методів відповідно до їхніх конкретних потреб. Це означає, що система буде розроблена таким чином, щоб бути гнучкою та адаптивною, дозволяючи користувачам налаштовувати функції та методи відповідно до конкретної групи зображень чи індивідуального зображення. Це може включати в себе коригування вагових коефіцієнтів певних параметрів, а також додавання або вилучення певних критеріїв. Пропонуючи функції, що налаштовуються, програмне забезпечення зможе краще відповідати унікальним потребам та уподобанням клієнтів, що призведе до більш точних та якісних вихідних результатів.

– Масштабованість: програмне рішення можна буде масштабувати відповідно до зростаючої кількості користувачів, які користуються системою та запитів. Для забезпечення масштабованості система буде побудована з використанням модульної архітектури, яка може бути легко розширена або модифікована за потреби. Система також використовуватиме хмарні сервіси, що дозволить їй динамічно масштабуватись залежно від попиту. Масштабованість системи гарантуватиме, що вона зможе продовжувати працювати навіть при збільшенні обсягу даних та кількості користувачів.

– Інтеграція з партнерськими системами: програмне забезпечення матиме змогу інтеграції із системами клієнтів чи партнерів для надання найкращого результату у найбільш зручній формі та задля використання усіх переваг, що надаються внаслідок цієї інтеграції.

– Безпека та конфіденційність: система повинна забезпечити безпеку, конфіденційність і захист даних користувачів від несанкціонованого доступу, модифікації або розголошення. Доступ до системи буде обмежено за допомогою захищених облікових записів та прав доступу. Для захисту конфіденційних даних використовуватиметься шифрування як під час передачі, так і під час зберігання. Користувачі будуть мати контроль над своїми даними, зображеннями

та можливість відмовитися від їх передачі. Програмне забезпечення буде розроблено відповідно до вимог загального регламенту про захист даних (англ. GDPR) [13].

Основними ролями користувачів програмного забезпечення є:

1. Індивідуальні користувачі – реєструють особистий профіль та можуть використовувати базовий функціонал програми.
2. Адміністратори – відповідають за управління доступом користувачів, налаштування та управління програмою на рівні організації або команди та розв'язання технічних питань.
3. Розробники – використовують прикладний програмний інтерфейс (англ. API), що надається програмним забезпеченням для реалізації власних рішень, інтеграції у системи їхніх організацій та мають можливість розробляти плагіни для розширення функціоналу.

Бізнес модель програмного забезпечення для цифрової обробки зображень передбачає різні підходи до різних сегментів та ролей користувачів. Для B2C сегменту та індивідуальних користувачів пропонуватиметься Freemium бізнес-модель [20], що передбачатиме наявність безкоштовної версії з обмеженнями щодо кількості зображень, що можуть бути оброблені та функціоналу, який доступний. Платна підписка в свою чергу пропонуватиме повний набір функцій, кількість зображень що може бути оброблена буде значно збільшена, але обмеження існуватимуть. Також у платній підписці буде доступне використання API, проте в даному сегменті використання обмежене тисячею запитів на місяць для тестових та власних цілей користувачів. Для B2B сегменту пропонуватиметься власна підписка, що передбачає три рівні співпраці. Першим є необмежена кількість застосувань, доступ до всіх функцій та обмежена кількість сценаріїв автоматизації та запитів API. В другому додатково надається можливість використання API із збільшеною кількістю запитів. Проте кількість запитів залежить від вартості. На третьому рівні співпраці вартість та наповнення підписки обговорюється з клієнтами в індивідуальному порядку, що дозволяє

забезпечити найбільшу гнучкість та найкращі умови для кожного конкретного партнера.

Серед переваг веб-системи для користувачів варто відзначити наступні:

- відсутність потреби встановлення програми, доступ здійснюється через браузер;
- хмарне сховище та легке масштабування відповідно до потреб клієнта;
- суттєве скорочення витрат на розгортання і швидке впровадження системи на підприємстві;
- ясність і передбачуваність процесів, делегування процесу розробки і розгортання на компанію-виробника програмного забезпечення;
- економія коштів – оплата лише за користування системою на основі визначеної вище бізнес моделі;
- регулярні оновлення, покращення продукту, нові функції та виправлення помилок;
- високий рівень захисту даних.

Зі свого боку, переваги такого рішення також існують для компанії-розробника даного програмного забезпечення, серед них:

- відсутність проблем із неліцензійним використанням програмного забезпечення;
- велика популярність веб-сервісів серед користувачів;
- великі функціональні можливості веб-сфери, постійний розвиток веб-технологій;
- відсутність проблем у пошуку технічних спеціалістів із розробки, тестування та впровадження;
- легше поширення продукту, глобальне охоплення ринків та збільшення конкурентоспроможності;
- активне залучення користувачів та їх втримання через інтеграцію та зручність надання послуг.

Отже, у випадку створення програмного забезпечення як сервісу (SaaS) [18] замовники та розробники отримують ряд переваг, що є суттєвими і були вирішальними при виборі саме такого підходу до проєктування, розробки системи та її подальшого впровадження.

2.5. Концептуальні моделі інформаційної системи

Інформаційна система розглядається як сукупність взаємопов'язаних елементів, які мають спільну мету або призначення і функціонально утворюють цілісну систему.

За допомогою моделі чорної скриньки [21] можна математично описати програмне забезпечення для цифрової обробки зображень, де ПЗ це чорна скринька з вхідними та вихідними даними. Вхідні дані X представляють множину $\{X_1, X_2, \dots, X_n\}$. В даному випадку це цифрові зображення та метадані. Вихідні дані Y представляють множину $\{Y_1, Y_2, \dots, Y_m\}$, що включає цифровий результат обробки та його метадані. Модель чорної скриньки визначає внутрішні процеси, які трансформують вхідні дані X у вихідні дані Y . Ці процеси включають обробку даних, використання алгоритмів машинного навчання, статистичний аналіз та інші методи цифрової обробки. В даному випадку, деталі внутрішніх процесів не враховуються. Математично це можна описати як функцію F , яка отримує вхідні дані X і видає вихідні дані Y . Представимо цю функцію у наступному вигляді:

$$Y = F(X) \quad (2.1)$$

На основі цієї функції можна створити більш детальну концептуальну модель складу інформаційної системи, що показана на рис. 2.2.

Необхідною умовою існування системи є зовнішнє середовище. Під зовнішнім середовищем розуміють сукупність рівнозначних систем, які

взаємодіють з розглянутою системою, задають правила поведінки системи, обмінюються з нею ресурсами [5].

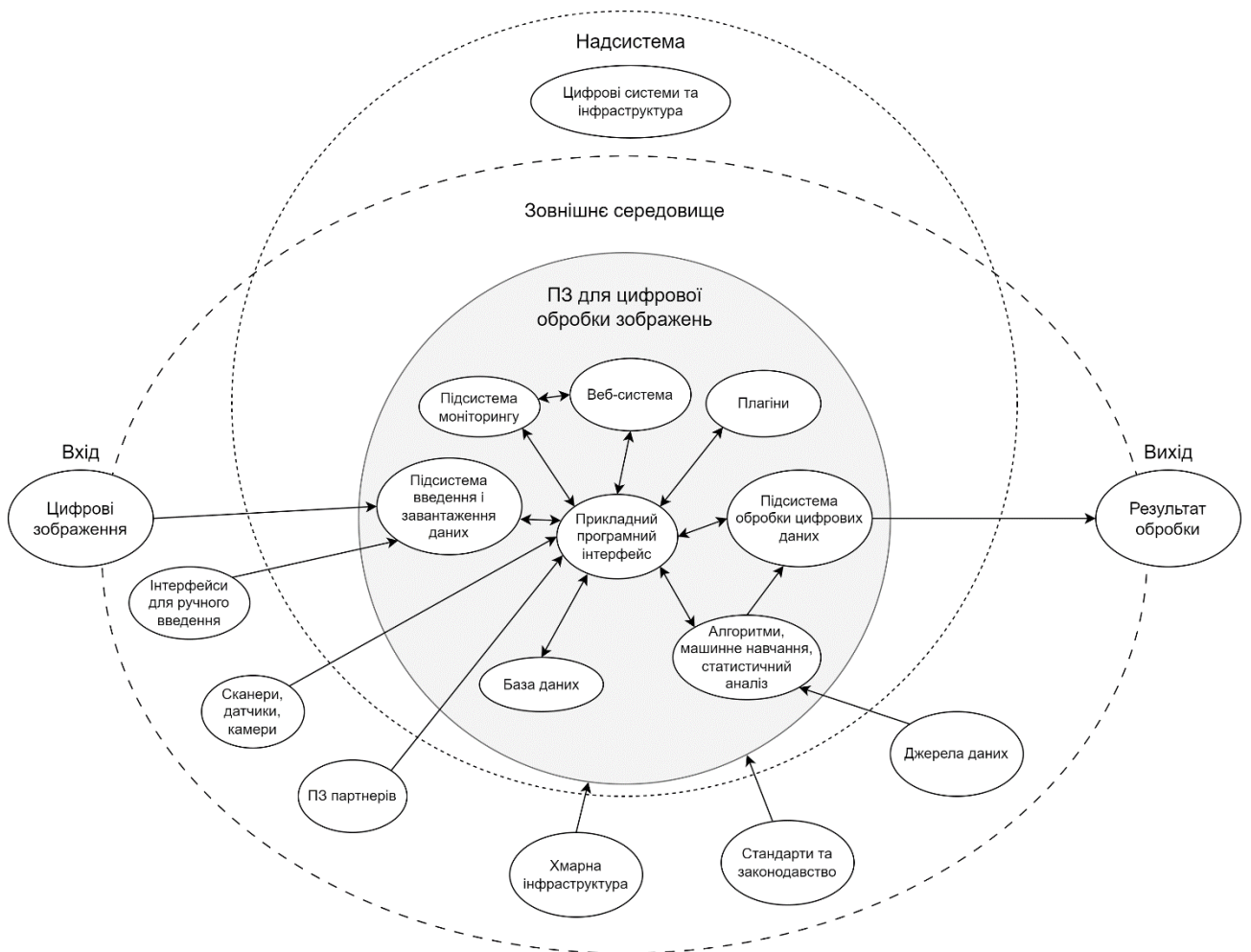


Рис. 2.2. Концептуальна модель складу інформаційної системи

Частина цього зовнішнього середовища, яка включає в себе досліджувану інформаційну систему, вважається надсистемою – більш загальною системою, частиною якої є дана система [5]. До зовнішнього середовища в даному випадку також належить програмне забезпечення партнерів, що може взаємодіяти із системою завдяки інтеграції або реалізовується на основі системи та її прикладного програмного інтерфейсу. Також значну роль відіграє хмарна інфраструктура, її застосування у системі та залежність системи від хмарних сервісів, що надаються. Також система покладається на сторонніх постачальників даних для реалізації алгоритмів на основі машинного навчання.

Проте, якщо безпосередньо розглядати складові інформаційної системи, то варто відмітити опціональну частину – плагіни. Їхня основна мета – розширення функціоналу системи за рахунок легкозв'язної інтеграції з основним функціоналом. В подальшому це може розвинути в платформу, яка надаватиме можливість встановлення плагінів від сторонніх розробників та можливість створення власного плагіну. Варто також зауважити, що відповідно до концептуальної моделі використання графічного інтерфейсу не є обов'язковим для користувачів. Але він існує для забезпечення зручності та для полегшення взаємодії користувача із системою. В інших випадках для реалізації специфічних завдань, система може використовуватись як технічний інструмент.

Загалом, побудова концептуальної моделі є корисним інструментом для аналізу та розуміння системи, вхідних та вихідних даних, внутрішньої структури та взаємозв'язків.

2.6. Формалізація математичних моделей

Завдання цифрової обробки зображень відіграють виняткову роль у багатьох сучасних галузях, становлячи фундаментальну основу для широкого спектра досліджень та прикладних рішень. Це галузь, що динамічно розвивається, теоретична основа якої ґрунтується на міцному математичному фундаменті, які описують властивості та поведінку цифрових зображень. Це дає можливість описувати, аналізувати та вирішувати складні завдання, пов'язані з обробкою та аналізом візуальної інформації. Найчастіше застосування в цій галузі мають методи лінійної алгебри, теорії матриць, математичного аналізу, теорії ймовірностей та статистики та інші.

Математично задачі цифрової обробки зображень можна описати за допомогою числових матриць. Наприклад, цифрове зображення може бути представлене у вигляді матриці пікселів, де кожен піксель має числове значення, яке відображає його яскравість, колір, прозорість тощо. Операції обробки

зображень, можна описати математичними формулами, які використовують ці числові значення пікселів та їхнє розташування у матриці.

У термін "система" вкладають різні поняття. Але він завжди є підмножиною взаємопов'язаних елементів, виділеною з множини елементів будь-якої природи, відповідно до вимог поставленого завдання. У загальному випадку технічна система як об'єкт моделювання характеризується такими параметрами [4]:

1. вхідні (x_1, x_2, \dots, x_i) – значення можуть бути виміряні, але система не має можливості впливати на них;
2. керувальні (w_1, w_2, \dots, w_h) – впливають безпосередньо на технічну систему і дозволяють керувати вихідними параметрами;
3. збурювальні (u_1, u_2, \dots, u_y) – значення змінюються випадково з часом і недоступні для зміни;
4. вихідні (y_1, y_2, \dots, y_j) – описують стан технічної системи або результат її функціонування під впливом вхідних, керувальних та збурювальних параметрів.

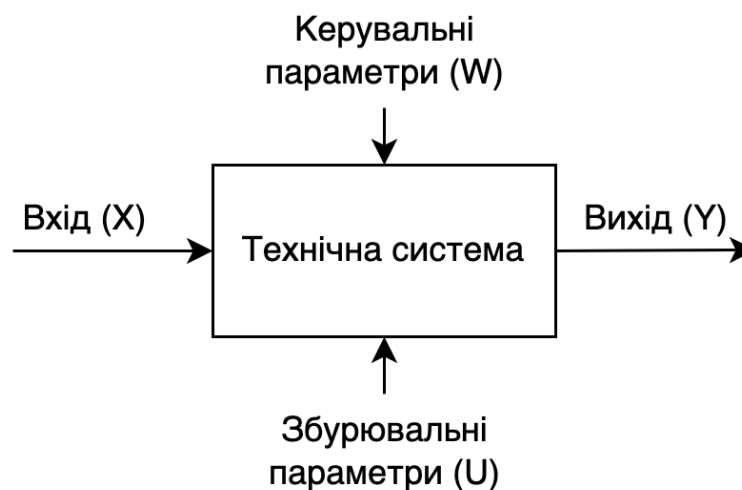


Рис. 2.3. Формальний опис технічної системи

Загалом, задачу цифрової обробки зображень можна формалізувати як перетворення множини вхідних аргументів X у вихідні дані Y за допомогою системи M із взаємозв'язками компонентів системи N (див. рис. 2.2), що

складається з набору алгоритмів, методів, їх вхідних аргументів та іншої додаткової інформації. Базова модель системи описується формулою 2.2.

$$M = \{X, Y, H\} \quad (2.2)$$

При цьому множина вхідних аргументів $X = \{I, M, A_1 \dots A_n, D\}$ та задається наступними елементами:

- I – вхідне зображення або набір зображень заданих у вигляді матриці пікселів;
- M – сигнатура методу, що виконує обробку зображення відповідно до визначеного алгоритму;
- $A_1 \dots A_n$ – перелік вхідних аргументів методу та їх значень відповідно до визначеної сигнатури та накладених обмежень щодо значень, які приймає метод;
- D – метадані, технічна характеристика зображення, додаткова інформація про зображення, така як дата зйомки, інформація про камеру, історія змін та автор.

Множина вихідних значень $Y = \{I', R_1 \dots R_n, K, E\}$ та задається таким переліком елементів:

- I' – оброблене зображення чи набір зображень або інший вихідний результат виконання методу (ЗД модель, виділені ознаки, виконана класифікація);
- $R_1 \dots R_n$ – набір статистичної інформації про виконання методу та аналітичні характеристики вихідного результату.
- K – вихідні метадані про результат обробки;
- E – опис процесу виконання, виконане логування, зазначення наявності помилок та їх опис за наявності.

Прикладом керувальних параметрів системи є порогові значення, набори існуючих алгоритмів, послідовність виклику, кількість ітерацій, гіперпараметри

моделей, коефіцієнти, методи оптимізації. До збурювальних – шум, що може призвести до спотворення результатів обробки, умови освітлення зображення, позиція камери відносно зони інтересу, неточності в датчиках, стиснення

Формалізація задач цифрової обробки зображень в математичному вигляді є важливою складовою розуміння цього процесу та системи, що буде оперувати цими методами в цілому. Це допомагає чітко визначати обмеження, задачі, розробляти алгоритми та аналізувати результати. Математичний підхід забезпечує зрозумілий опис та інструментарій для вирішення широкого спектра поставлених задач.

2.7. Прийняття управлінських рішень в проєкті

Для успішного виконання проєкту важливим завдання є застосування технології прийняття рішень RBM (англ. Results-Based Management) – стратегічного підходу до управління проєктом, який акцентує увагу на досягненні конкретних результатів та впливу [6]. Основна ідея полягає в тому, щоб усі дії і ресурси спрямовувалися на досягнення визначених цілей та планів, а результати були вимірювані, аналізовані та використовувалися для прийняття управлінських рішень. Усі учасники, роблячи прямий чи опосередкований внесок у досягнення цілей, повинні гарантувати, що їхні процеси, продукти та послуги сприяють досягненню бажаних проміжних чи кінцевих результатів.

Для використання цього підходу необхідно визначити:

1. критерії успіху проєкту, пов'язати їх з фазами проєкту;
2. головні потенційні причини невдач проєкту, пов'язати їх з фазами проєкту;
3. заходи щодо усунення можливих причин невдач проєкту [6].

Результати визначення критеріїв успіху та відповідних фаз проєкту показано в табл. 2.4.

Зв'язок критеріїв успіху проєкту і фаз його життєвого циклу

Фаза проєкту	Критерій успіху
Ініціація	Чітке формулювання проєктного завдання, фіналізація концепції та створення паспорту проєкту, отримання схвалення на його реалізацію
Планування	Розроблено та схвалений детальний план проєкту, який включає в себе розподіл ресурсів, обсяг робіт, вимоги до продукту, графік виконання завдань, оцінку ризиків та механізми контролю виконання.
Виконання та контроль	Завершено розробку програмного продукту для цифрової обробки зображень відповідно до визначених технічних вимог, часових обмежень та в рамках планового бюджету. Успішне тестування продукту, відповідність функціональним вимогам.
Впровадження	Успішне впровадження проєкту згідно затвердженого плану, розгортання інфраструктури, випуск MVP версії продукту відповідно до визначених бізнес-цілей.
Завершення	Успішно завершено проєкт, створено необхідну технічну документацію, проведено аудит та оцінку відповідності результатів поставленим цілям та очікуванням.

Наступним кроком є визначення головних потенційних невдач проєкту у відповідності до фаз проєкту. Результат показано в табл. 2.5.

Зв'язок причини невдач проєкту і фаз життєвого циклу

Фаза проєкту	Потенційні причини невдачі проєкту
Ініціація	Відсутність підтримки керівництва, недостатнє обґрунтування доцільності та життєздатності.
Планування	Некоректне планування, розрахунок часу чи бюджету, недостатнє визначення вимог. Недостатні ресурси.
Виконання та контроль	Суттєві зміни у бізнес-вимогах, технічні проблеми, поганий контроль якості, недооцінка обсягу робіт. Критичне перевищення термінів та бюджету.
Впровадження	Технічні проблеми, несумісність ПЗ з існуючими сервісами та системами.
Завершення	Комерційний провал проєкту.
Усі фази	Форс-мажорні обставини, недостатня кваліфікація команди проєкту, труднощі у комунікації.

Для уникнення цих причин варто розробити заходи щодо їх усунення та попередження. Вони показані в табл. 2.6.

Заходи щодо усунення можливих причин невдач проєкту

Потенційні причини невдач проєкту	Заходи
Відсутність підтримки керівництва, недостатнє обґрунтування доцільності та життєздатності.	Створення детального звіту щодо доцільності створення даного проєкту, презентація його керівництву з підтвердженням і обґрунтуванням його життєдіяльності, раціональності та рентабельності; забезпечення прозорості та відкритої комунікації з керівництвом.
Некоректне планування, нестача часу чи бюджету, недостатнє визначення вимог. Недостатні ресурси.	Створення детального календарно-ресурсного плану; створення плану робіт, управління ризиками та забезпечення якості; перегляд бюджет проєкту, аналіз витрат, пошук додаткових джерел фінансування, застосування ефективних методів управління проєктами; перегляд та оптимізація витрачених ресурсів; перегляд обсягу робіт та концентрація зусиль на основних завданнях; залучення додаткових людей до роботи над проєктом.
Суттєві зміни у бізнес-вимогах, технічні проблеми, поганий контроль якості, недооцінка обсягу робіт. Критичне перевищення термінів та бюджету.	Регулярний моніторинг та контроль виконання проєкту, перевірка відповідності поточного стану розробленим планам; застосування Agile підходу, який дозволяє оперативно реагувати на зміни; визначення стандартів якості, створення відповідних документів та використання інструментів для моніторингу якості, ведення статистики та аналіз причин виникнення даних проблем; автоматизація тестування; створення відповідних каналів комунікації.
Технічні проблеми, несумісність ПЗ з існуючими сервісами та системами.	Визначення і розробка чітких технічних вимог перед початком проєкту, що допоможе уникнути будь-яких неоднозначностей щодо функціональності; використання найкращих практик в галузі розробки програмного забезпечення, наприклад, Code Review і CI/CD.
Комерційний провал проєкту.	Першочергове створення MVP продукту, створення реалістичної і стійкої бізнес-стратегії і шляхів монетизації продукту проєкту; фокус на цільовій аудиторії, збір зворотного зв'язку та ітераційне удосконалення продукту; розробка маркетингового та PR плану.
Форс-мажорні обставини, недостатня кваліфікація команди проєкту, труднощі у комунікації.	Розробка протиризикових заходів для всіх фаз життєвого циклу проєкту; неперервний аналіз та оцінка ризиків; залучення досвідчених фахівців зі знанням сучасних методів та технологій розробки ПЗ, а також з відповідними навичками актуальними для проєкту; підвищення кваліфікації працівників; створення плану комунікації та відповідних каналів; забезпечення доступності інформації, регулярні зустрічі.

Загалом, управління по результатам має суттєвий вплив на проєкт та допомагає йому стати більш ефективними, спрямовуючи зусилля команди проєкту на досягнення конкретних та вимірюваних результатів, а також забезпечує більшу прозорість та відповідальність під час їх виконання. Сприяє більш ефективному використанню ресурсів проєкту, недопущенню ризикових подій та досягненню позитивного впливу через чітке визначення та систематичне відстеження результатів діяльності.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

3.1. Опис програмного забезпечення та технологій для його реалізації

Продуктом проєкту є програмне забезпечення для цифрової обробки зображень, що складається з трьох основних частин: головної програмної системи у вигляді веб-системи з клієнтською та серверною частиною, прикладного програмного інтерфейсу (англ. API) та підсистеми взаємодії із модулями розширень, які можуть бути встановлені незалежно один від одного. Архітектура програмного забезпечення буде модульною, що дозволить забезпечити зручність в розробці та підтримці програмного коду, а також сприятиме покращенню розширюваності. Інтерфейс користувача буде частиною ПЗ, що не є обов'язковою для використання функціоналу системи, проте дозволяє робити це більш зручно. Для розробки програмного забезпечення було обрано наступний перелік технологій:

- Microsoft Azure [34] – комплексна платформа хмарних обчислень, головною перевагою якої є широкий спектр хмарних сервісів, зокрема інфраструктура як послуга (англ. IaaS), платформа як послуга (англ. PaaS) та програмне забезпечення як послуга (англ. SaaS).

- Azure SQL Database [32] – хмарна служба реляційних баз даних, яка забезпечує високу доступність, автоматизоване управління операціями, налаштування продуктивності, масштабованість і безпеку веб-додатків.

- Azure Machine Learning [31] – хмарна платформа, що забезпечує підтримку повного циклу машинного навчання, включаючи підготовку даних, створення та навчання моделей, перевірку, розгортання, а також управління та моніторинг.

– ASP.NET Core [41] – кросплатформний веб-фреймворк з відкритим вихідним кодом, що надає модульну та високопродуктивну платформу для створення веб-застосунків. Перевагою є проста інтеграція та вбудована підтримка служб з Microsoft Azure, що дозволяє розробникам використовувати всі переваги хмарної платформи для масштабування, доступності та простоти розгортання.

– Angular [38] – це front-end фреймворк, розроблений командою Angular у Google та спільнотою розробників. Він дозволяє розробникам створювати динамічні односторінкові додатки (англ. Single Page Applications) за допомогою мови програмування TypeScript та компонентно-орієнтованого підходу.

– Entity Framework Core (EF Core) [43] – об'єктно-орієнтована технологія доступу до даних. Це інструмент ORM (об'єктно-реляційне відображення – зіставлення даних з реальними об'єктами), що дозволяє працювати з базами даних, але вводить вищий рівень абстракції, що усуває необхідність у написанні більшості прямих запитів до бази даних.

Ці технології пропонують сучасний і всеосяжний стек технологій для створення надійних, масштабованих, безпечних веб-систем з доступом до баз даних. Більшість технологій належить до екосистеми Microsoft, що значно спрощує розробку та підтримку єдиної екосистеми, яка має значні переваги, зокрема:

– швидкість розробки – всі технології мають вбудовану підтримку в середовищі розробки Visual Studio [45], що усуває потребу у використанні великої кількості різного програмного забезпечення;

– продуктивність програми – всі технології, розроблені однією компанією, успішно взаємодіють між собою, що дозволяє досягти більшої швидкості роботи програми;

– легкість підтримки – чіткий і зрозумілий шлях розвитку та час підтримки, що дозволяє не замислюватись про те, що один із компонентів більше не буде підтримуватися. Перехід на використання нових версій технологій також

не викликає проблем. Більшість нових версій зворотно сумісні зі старими версіями;

- наявність чіткої та повної документації дозволяє легко знайти рішення проблем або розібратись у взаємодії з іншими технологіями екосистеми;
- велика бібліотека, що дозволяє використовувати готові інструменти для багатьох основних потреб.

Отже, усі наведені технології можна використовувати разом для створення сучасного, масштабованого та адаптивного ПЗ з потужною серверною частиною.

3.2. Огляд архітектури, компонентів та підходів до розробки

Програмне забезпечення для цифрової обробки зображень складається з чотирьох основних компонентів:

1. веб-система – є веб-клієнтом, що реалізовує користувацький інтерфейс та взаємодіє з прикладним програмним інтерфейсом;
2. головний програмний модуль, що надає перелік базових функцій;
3. сторонні плагіни, що розширюють головний функціонал системи, є опціональними та реалізуються як окремі сервіси;
4. прикладний програмний інтерфейс, що реалізовується та використовується як і клієнтом так і плагінами.

Загалом, система буде реалізовувати мікросервісну архітектуру – підхід до розробки програмного забезпечення, при якому програма розбивається на невеликі, незалежні компоненти, які називаються мікросервісами [10]. Кожен мікросервіс виконує конкретну функцію і може бути розгорнутий та масштабований окремо. Це надає наступні переваги:

- мікросервіси дозволяють розділити складний проєкт на менші, керовані ефективніше частини, що спрощує розробку, впровадження та підтримку;

– кожен мікросервіс може бути масштабований окремо [15], що дозволяє зробити цей процес гнучким, оптимізувати використання ресурсів та реагувати на зміни в навантаженні;

– різні мікросервіси можуть використовувати різні технології та мови програмування в залежності від їхніх потреб, що дозволяє використовувати найкращі інструменти для кожного компонента системи;

– мікросервіси можна швидко розгортати та впроваджувати, оскільки кожен компонент може бути розроблений та тестований окремо, також команда може працювати над різними мікросервісами паралельно, що значною мірою прискорює процес розробки;

У проєкті з розробки програмного забезпечення для цифрової обробки зображень даний підхід відіграватиме ключові роль так як навіть сама концепція та вимоги до ПЗ передбачають максимальну гнучкість, адаптивність, масштабованість, а також наявність як і внутрішніх так і сторонніх плагінів та розширень.

Відповідно діаграму компонентів показано на рис. 3.1.

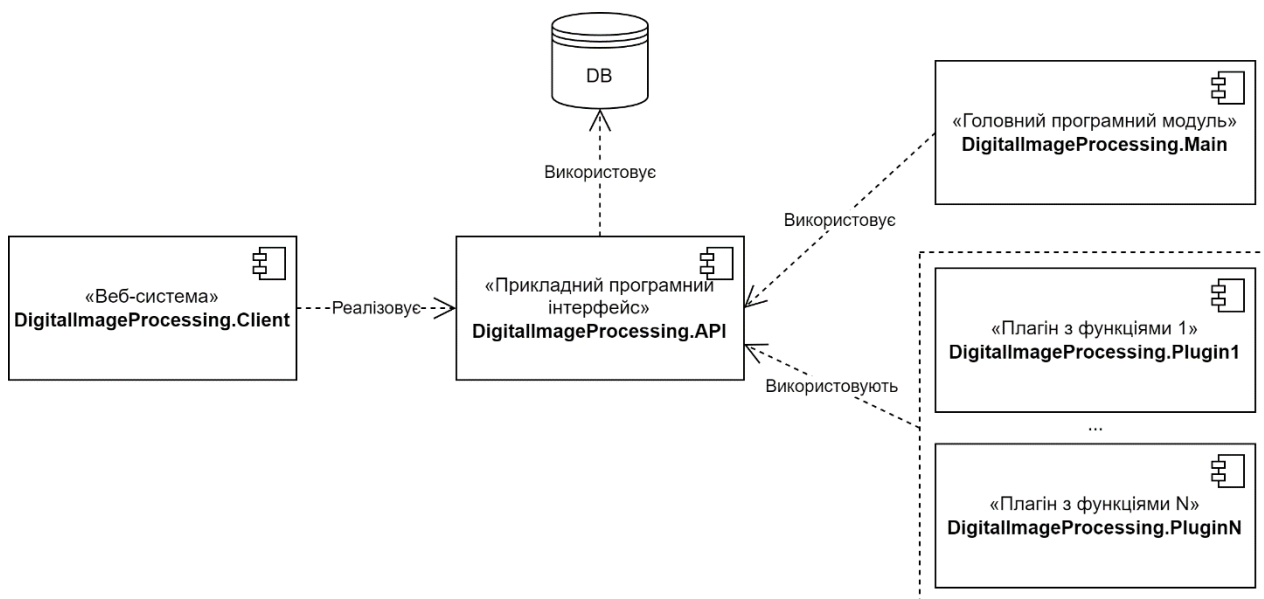


Рис. 3.1. Діаграма компонентів ПЗ для цифрової обробки зображень

В основу веб-системи покладено архітектурний шаблон Model-View-Controller (MVC) [42], що розділяє програму на три основні групи компонентів: моделі, представлення та контролери. Цей шаблон допомагає досягти поділу завдань. Він передбачає, що запити користувачів направляються до контролера, який відповідає за роботу з моделлю для виконання дій користувача та/або отримання результатів запитів. Контролер вибирає представлення для відображення користувачеві та надає йому будь-які потрібні дані моделі.

– Модель у MVC-додатку представляє стан додатку та будь-яку бізнес-логіку або операції, які він повинен виконувати. Бізнес-логіка повинна бути інкапсульована в моделі разом з будь-якою логікою реалізації для збереження стану додатку. Строго типізовані подання зазвичай використовують типи ViewModel, призначені для зберігання даних, що відображаються у цьому поданні. Контролер створює та заповнює ці екземпляри ViewModel з моделі [42].

– Представлення відповідають за представлення вмісту через інтерфейс користувача. У коді представлення має бути мінімальна логіка, і будь-яка логіка в них має стосуватися представлення вмісту [42].

– Контролер — це компонент, який виконує взаємодію з користувачем, працює із моделлю та, зрештою, вибирає представлення для візуалізації [42].

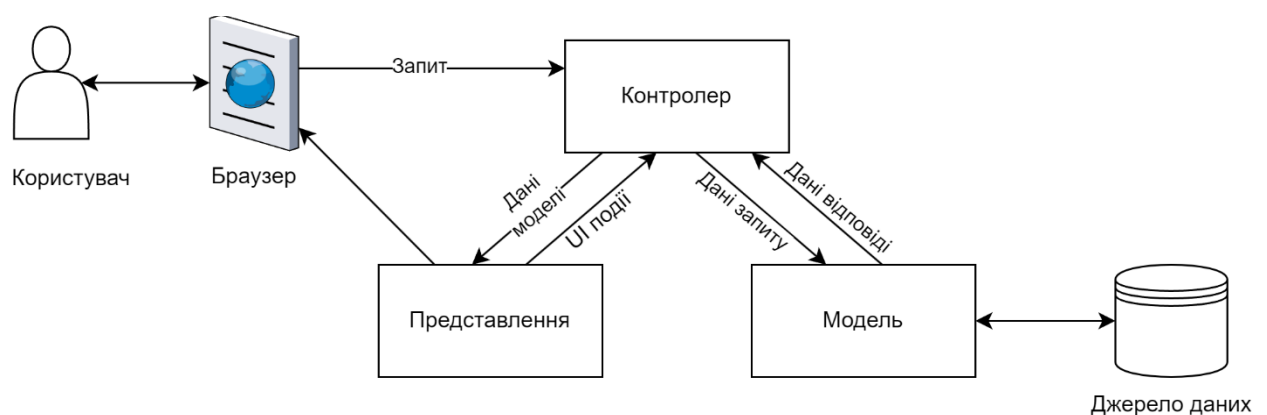


Рис. 3.2. Архітектурний шаблон MVC

Такий розподіл обов'язків допомагає масштабувати програму з точки зору складності, бо це спрощує написання коду, налагоджування і тестування чогось

(модель, представлення або контролер), що має єдине завдання. Набагато складніше оновлювати, тестувати і налагоджувати код, який має залежності, що поширюються на два або більше з цих трьох компонентів. Наприклад, інтерфейс користувача має тенденцію змінюватися частіше, ніж бізнес-логіка. Якщо код представлення та бізнес-логіка об'єднані в одному об'єкті, об'єкт, що містить бізнес-логіку, необхідно модифікувати щоразу, коли змінюється користувацький інтерфейс. Це часто призводить до помилок і вимагає повторного тестування бізнес-логіки після кожної мінімальної зміни користувацького інтерфейсу. Саме тому такий підхід є не бажаним, а розподіл обов'язків – ефективним рішенням цієї проблеми. Серед інших переваг також варто відзначити:

- кожен компонент може розроблятися паралельно з іншими, що значною мірою прискорює сам процес розробки;
- повторне використання коду в різних частинах програми;
- хороша підтримка оптимізації для пошукових систем (SEO);
- легке використання асинхронного коду в розробці;
- спрощене планування, проєктування та підтримка існуючого коду;
- спрощення модульного та інтеграційного тестування, підтримка розробки через тестування (TDD).

Проте, також існують певні недоліки цього архітектурного шаблону, а саме велика кількість файлів коду, складна структура класів і труднощі розуміння, але вони є незначними і правильне використання та розуміння принципів MVC дозволяє уникнути більшості проблем і забезпечити ефективну та організовану розробку веб-додатків.

Програмне забезпечення розробляється з використанням асинхронного програмування для прискорення роботи та уникнення блокування основного потоку програми. Наприклад, складні алгоритми обробки зображень, використання нейронних мереж та запити до бази даних реалізовані асинхронно. Особливістю системи є чітка та не перевантажена архітектура з використанням механізму впровадження залежностей (англ. *dependency injection*) [35] – шаблону проєктування програмного забезпечення, що передбачає надання зовнішньої

залежності програмному елементу, використовуючи інверсію управління (IoC container) для отримання залежностей. ASP.NET Core має вбудовану підтримку впровадження залежностей (DI). У ASP.NET Core MVC контролери можуть запитувати необхідні сервіси через свої конструктори, що дозволяє їм дотримуватися принципу явних залежностей [42]. Він передбачає те, що всі залежності класу повинні бути виражені явно та передані через конструктор, метод або властивість класу [36]. Цей принцип допомагає зробити код більш читабельним, зрозумілим та самодокументованим, оскільки він чітко вказує на те, які саме залежності використовуються у класі. Крім того, він сприяє зменшенню зв'язності між класами та робить код більш гнучким, оскільки залежності можуть бути легко замінені або розширені без зміни основного класу. Загалом використання DI та IoC дозволяє легко тестувати та підтримувати код, а також забезпечує можливість внесення змін через файли конфігурації без потреби у перекомпіляції програми. Впровадження залежностей зменшує зв'язність між класом і його залежностями і відповідає ідеї п'ятого принципу SOLID – принципу інверсії залежностей.

Для реалізації процесу автентифікації буде використовуватись підхід на основі токенів (англ. Token-Based Authentication) – це метод автентифікації, де сервер виділяє токен доступу (англ. access token) для користувача після успішної автентифікації, і цей токен використовується для підтвердження ідентичності користувача при кожному запиті до сервера. Коли клієнту потрібно надіслати запит до API, він використовує цей токен у якості заголовка до запиту. Потім сервер перевіряє токен, знайдений у заголовку запиту, перед тим, як виконати запит [44]. Автентифікація на основі токенів не має стану і є масштабованою. В даному випадку токен є просто криптографічно значущим рядком символів. Найпоширеніший формат токенів – JSON Web Token, або JWT. Процес автентифікації показано на рис. 3.3.

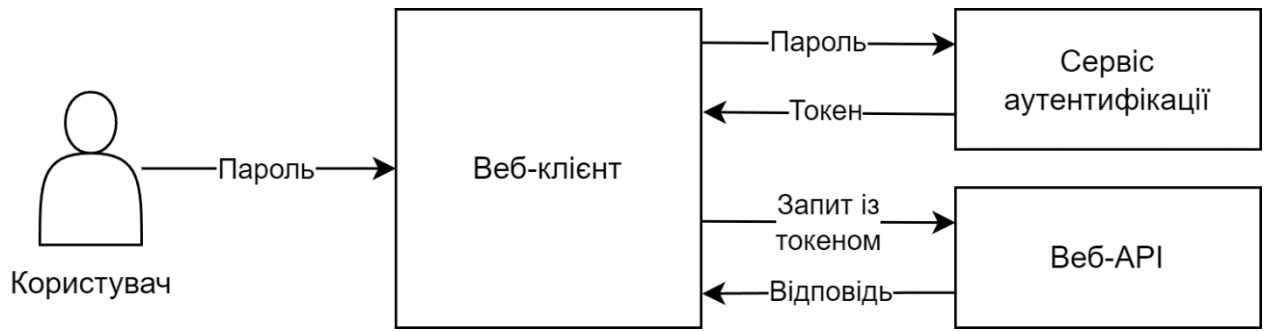


Рис. 3.3. Процес автентифікації у веб-клієнті

Також важливим інструментом для моделювання та розуміння системи є побудова діаграм інформаційних потоків даних. Вони дозволяють візуалізувати, як дані переміщуються в системі, як вони обробляються та як взаємодіють різні компоненти системи. Дані діаграми дозволяють розробникам та іншим стекхолдерам краще зрозуміти структуру та функціонування системи, виявити проблеми та оптимізувати процеси. Вони можуть бути використані на різних рівнях деталізації, починаючи із загального огляду системи до докладного опису окремих процесів та взаємодій. В даному випадку, на рис. 3.4. показано високорівневу діаграму, що в загальному описує систему та основні компоненти.

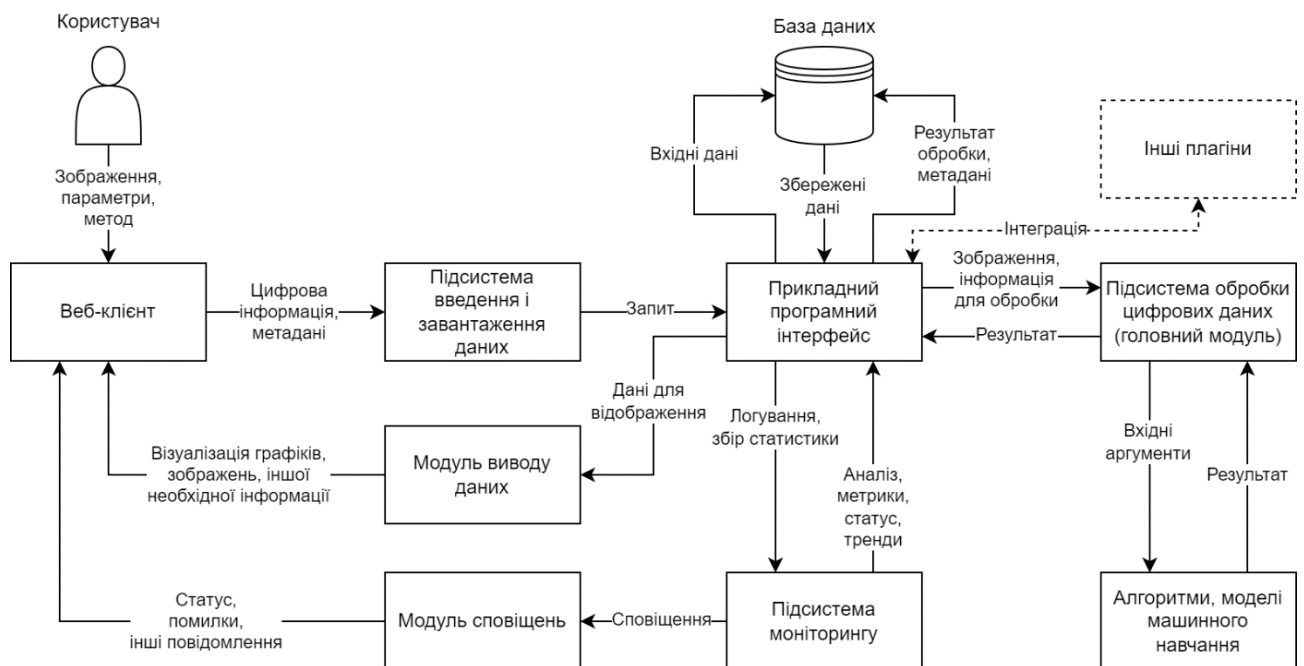


Рис. 3.4. Діаграма інформаційних потоків даних ПЗ

Підсумовуючи, огляд архітектури, компонентів та підходів до розробки є ключовим етапом у процесі створення програмного забезпечення. Від правильного вибору архітектури та компонентів залежить якість, ефективність та масштабованість розробленого продукту. Розглянуті у цьому розділі підходи, такі як мікросервісна архітектура, шаблон проєктування MVC, деталізація процесу автентифікації та діаграма інформаційних потоків даних системи, надають можливість структурувати та організувати роботу з різними компонентами системи, спрощують аналіз та розробку, а також є важливою часткою документації для команди проєкту. Важливим також на даному етапі було врахування специфіки проєкту та потреб користувачів, щоб забезпечити оптимальний початковий функціонал та задоволення від використання розробленого програмного продукту.

3.3. Розробка бази даних проєкту

База даних – це впорядкований набір структурованої інформації або даних, які зазвичай зберігаються в електронному вигляді в комп'ютерній системі та зазвичай управляється системою керування базами даних (СКБД). База даних в програмному забезпеченні відіграє ключову роль у зберіганні, організації та доступі до даних. Вона дозволяє ефективно управляти великими обсягами інформації, забезпечує доступ до даних для різних компонентів системи та забезпечує стабільність роботи та безпеку даних. Основні функції бази даних в програмному забезпеченні включають:

- зберігання даних необхідних для роботи програми, таких як інформація про користувачів, зображення, результати обробки та інші дані;
- організація даних в логічні структури, що спрощує їх зберігання, отримання та модифікацію;
- збереження цілісності даних за допомогою правил і обмежень, а також забезпечення захисту від несанкціонованого доступу;

- захист даних від втрати та пошкодження завдяки резервному копіюванню, відновленню та іншим механізмам безпеки, які гарантують, що інформація залишатиметься доступною навіть у разі збоїв системи;

- виконання функцій управління даними (CRUD), таких як додавання, читання, оновлення, видалення. Окремо варто відзначити також пошук.

Традиційно найбільш популярним типом баз даних є реляційні бази даних – набір даних з визначеними зв'язками між ними у вигляді набору таблиць, що складаються із колонок і рядків, але вони не є єдиним доступним рішенням. NoSQL бази даних, такі як MongoDB, пропонують інший підхід до зберігання об'єктів. Замість того, щоб зіставляти об'єкти з таблицями і рядками, інший варіант – серіалізувати весь граф об'єктів і зберігати результат. Перевагами такого підходу, принаймні на початковому етапі, є простота і продуктивність. Простіше зберігати один серіалізований об'єкт з ключем, ніж розбивати об'єкт на багато таблиць зі зв'язками та оновлювати рядки, які могли змінитися з моменту останнього отримання об'єкта з бази даних. Аналогічно, вибірка та десеріалізація одного об'єкта зі бази даних на основі ключа зазвичай набагато швидша і простіша, ніж складні запити або кілька запитів до бази даних, необхідних для повного складання того ж об'єкта з реляційної бази даних. Відсутність транзакцій або фіксованої схеми також робить бази даних NoSQL придатними для масштабування на багатьох машинах, підтримуючи дуже великі набори даних. Проте, з іншого боку, бази даних NoSQL мають свої недоліки. Реляційні бази даних використовують нормалізацію, щоб забезпечити узгодженість і уникнути дублювання даних. Цей підхід зменшує загальний розмір бази даних і забезпечує миттєву доступність оновлень спільних даних у всій базі даних. Реляційні бази даних також можуть забезпечити реляційну цілісність, застосовуючи такі правила, як зовнішні ключі. Бази даних NoSQL зазвичай не пропонують таких обмежень для своїх даних [44].

Також серед переваг реляційних баз даних варто виділити те, що таблиці є простими для розуміння, а реляційні бази мають суворі правила проєктування (наприклад, нормальні форми) при дотриманні яких можна отримати хороший

результат. Для запитів в таких базах даних використовується мова структурованих запитів (SQL). Саме тому для реалізації в програмному забезпеченні можливості збереження інформації про користувачів та їх зображення з використанням Azure SQL Database було реалізовано реляційну базу даних, в якій зберігається вся інформація про зображення, методи, їх параметри, результати обробки, дані про користувачів та їх налаштування, статистичні дані, логи, сповіщення, метрики та коментарі користувачів.

На рис. 3.5. показано концептуальну діаграму бази даних – перший етап розробки бази даних, який описує високорівневу модель інформаційної системи, в незалежності від конкретних технологій чи інструментів, без орієнтації на систему керування базами даних і модель даних [1]. На даному етапі увага сфокусована саме на предметній області, основних поняттях, їх характеристиках та взаємозв'язках між ними.

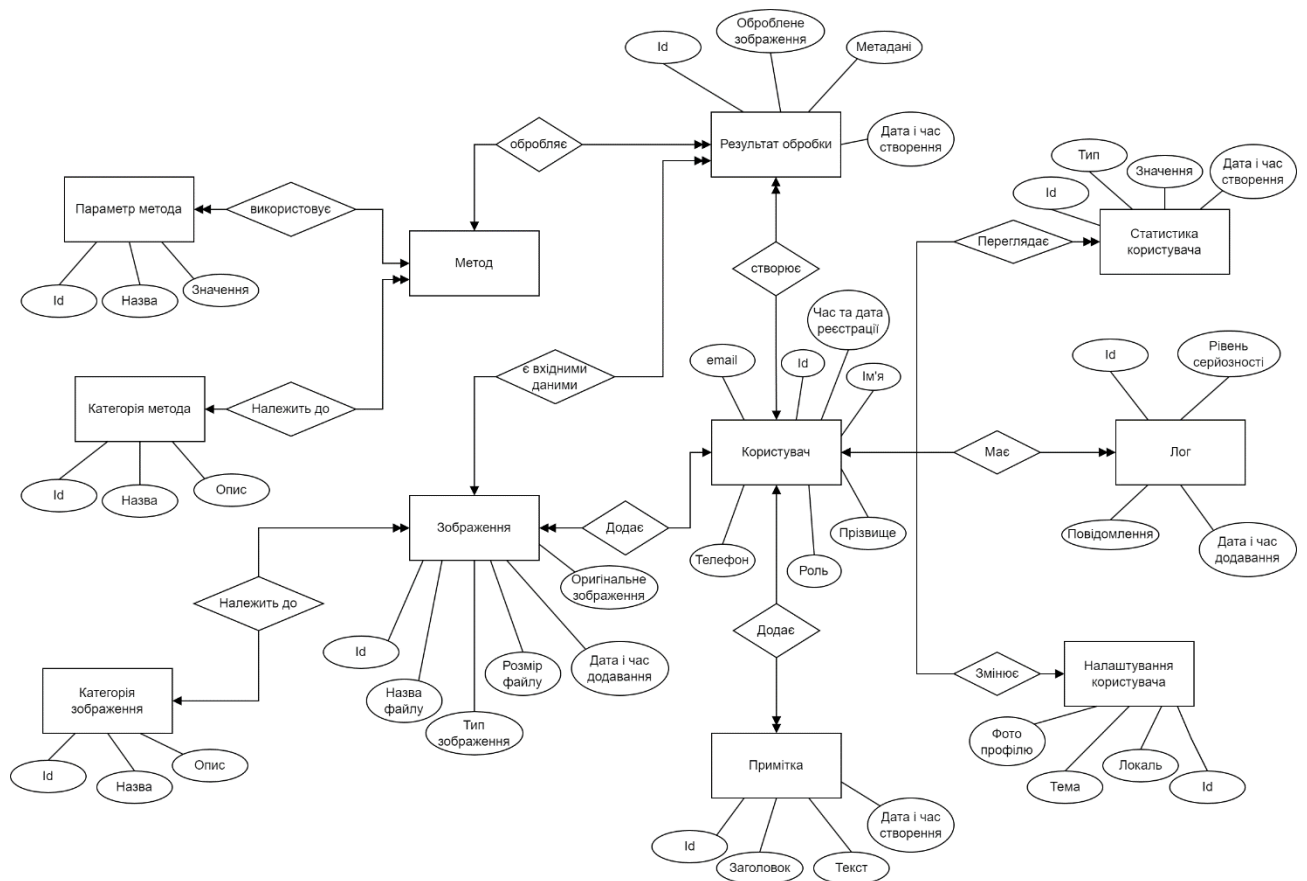


Рис. 3.5. Концептуальна модель бази даних

Другим кроком є створення логічної моделі бази даних – процес створення моделі даних, яка описує структуру та взаємозв'язки між даними в базі даних, незалежно від конкретної СКБД, яка буде використовуватися [1]. В даному випадку це реляційна модель даних. Цей крок є важливим етапом розробки бази даних, оскільки забезпечує чітке розуміння моделі даних, допомагає виявити та усунути потенційні проблеми та значною мірою спрощує процес фізичного проєктування завдяки визначенню первинних і зовнішніх ключів. Результат проєктування показано на рис. 3.6.

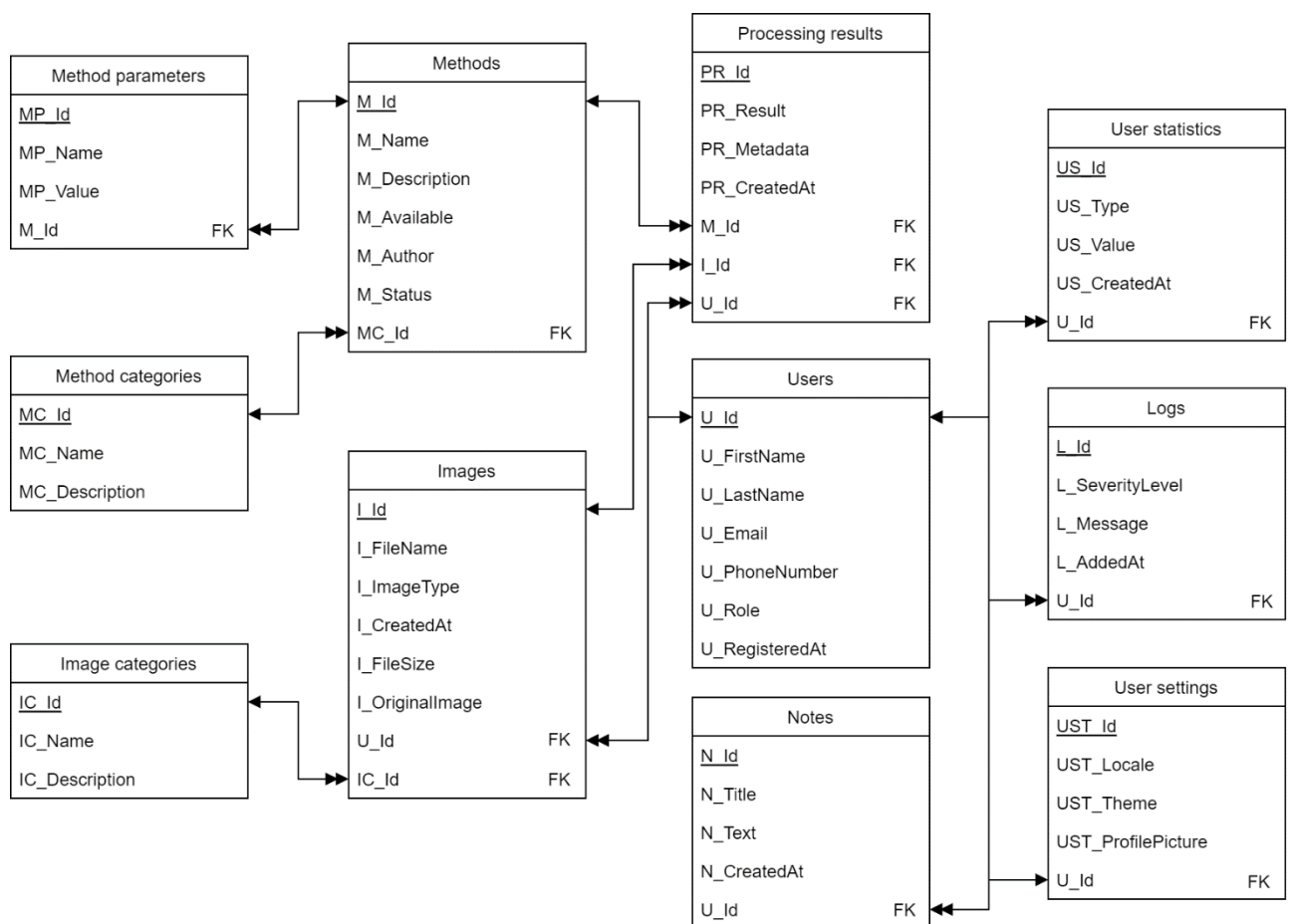


Рис. 3.6. Логічна модель бази даних

Останнім кроком є фізичне проєктування – детальний опис реалізації бази даних з урахуванням особливостей обраної СКБД [1]. В даному випадку це буде Azure SQL Database. Отриманий результат показано на рис. 3.7.

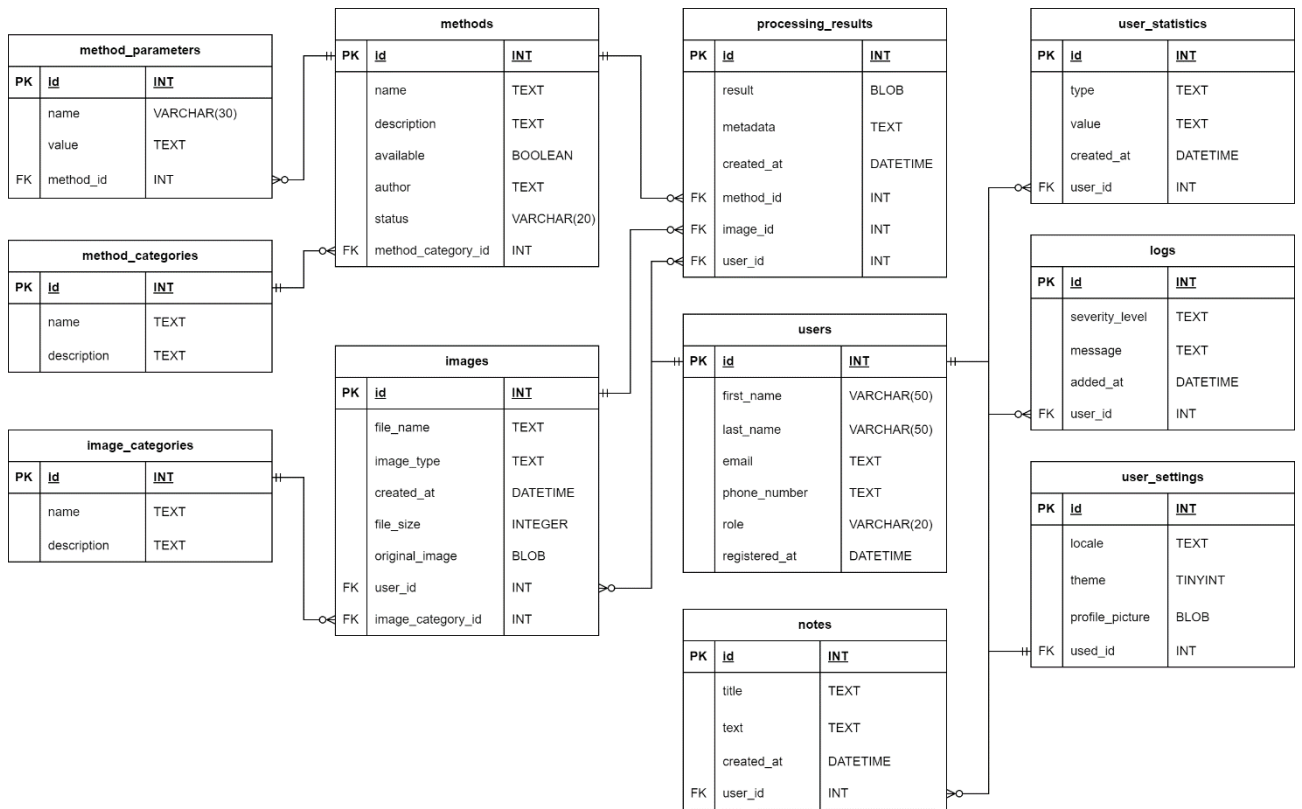


Рис. 3.7. Фізична модель бази даних

Всього в базі даних 11 таблиць, а між ними присутні два основних типи зв'язків: один до одного і один до багатьох.

Під час виконання проєктування бази даних основні завдання полягали в тому, щоб визначити дані, які будуть зберігатись, скоротити надмірність і дублювання даних, забезпечити цілісність та можливість отримання даних по всім необхідним запитам.

3.4. Огляд алгоритмів

Для реалізації алгоритмічної складової програмного забезпечення було створено різноманітні алгоритми обробки цифрових зображень. Загалом це може мати застосування в двох напрямках: як інструмент для отримання фінального результату або ж для попередньої обробки даних перед застосуванням штучного інтелекту. Було реалізовано різноманітні методи серед яких варто відзначити:

теплові карти, білатеральний та медіанний фільтр, корекцію яскравості та контрастності, морфологічні операції, методи сегментації, бінаризації зображення та похідні алгоритми. Приклад реалізації окремих алгоритмів наведено в додатку В.

Реалізованим прикладом застосування методів цифрової обробки зображень може бути просте завдання покращення фото тексту. Для вирішення цього завдання ефективним рішенням є застосування методів порогової обробки, що використовуються для перетворення зображення з градацією сірого у бінарне зображення (чорно-біле) за допомогою встановлення порогового значення. Усі пікселі з інтенсивністю (яскравістю) вище порогового значення вважаються білими, а всі пікселі з інтенсивністю нижче порогового значення – чорними. Існує кілька типів методів порогової обробки, включаючи методи з використанням глобального порогу (англ. *global thresholding*), локального порогу (англ. *local thresholding*) та адаптивного порогу (англ. *adaptive thresholding*). Кожен з цих методів має свої особливості та використовується в залежності від конкретного завдання обробки зображень.

Метод з використанням глобального порогу використовується у випадках, коли розподіл інтенсивності між об'єктами переднього і заднього плану дуже відрізняється, в інших випадках він буде не дуже ефективним. Зі свого боку, метод з використанням адаптивного порогу є більш ефективним та показує дуже гарні результати в певних типах завдань, так як він обирає різні порогові значення для кожного пікселя зображення на основі аналізу сусідніх пікселів. Порівняння методів та практичне застосування показано на рис. 3.8. Зліва показано оригінальне зображення, по центру – після застосування методу з використанням глобального порогу (в даному випадку зі значенням 40) та справа – після застосування методу з використанням адаптивного порогу та інших функцій, зокрема морфологічних операцій.

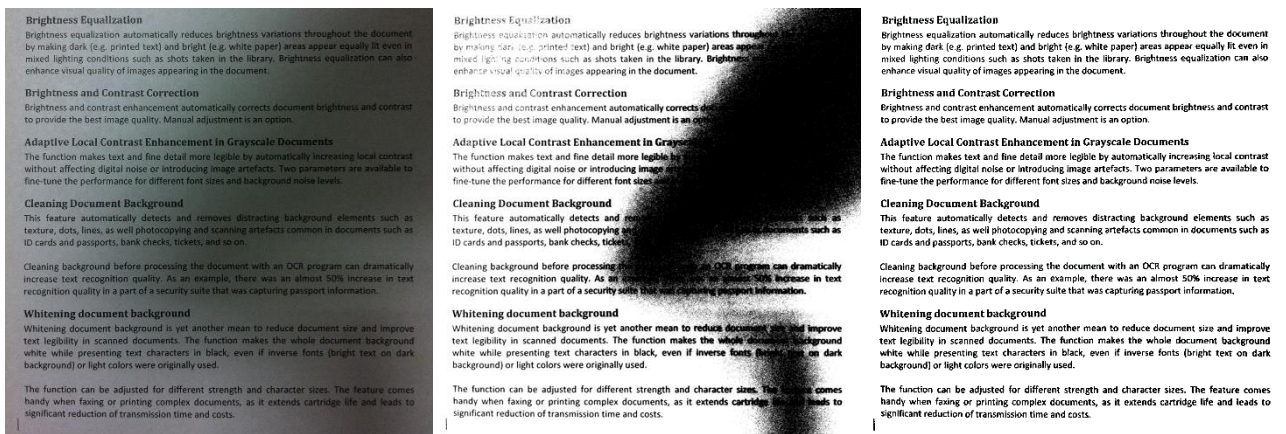


Рис. 3.8. Практичне застосування методів цифрової обробки зображень

Для отримання найкращого результату використовувалось декілька функцій, що були викликані по черзі. Алгоритм показано на рис. 3.9.

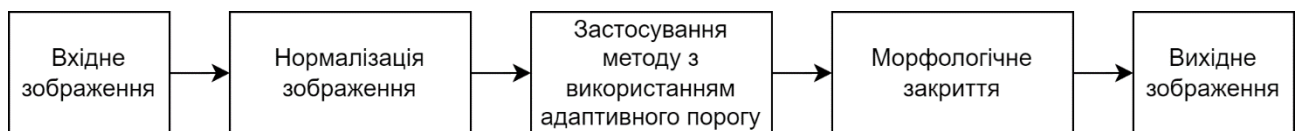


Рис. 3.9. Алгоритм для покращення тексту на фото з видаленням шуму

При розробці алгоритмів на основі штучного інтелекту також використовувалась початкова підготовка даних, зокрема формування набору даних, відсіювання дублікатів, конвертація в потрібний формат, присвоєння унікальних імен та аугментація даних, що дозволяє зробити нейронні мережі більш стійкими до вхідних даних. Потім набори даних поділялись на три головні вибірки: навчальну, валідаційну та тестову із співвідношенням 8:1:1.

– Навчальна вибірка є найбільшою та використовується для навчання моделі. Для цього алгоритм машинного навчання вивчає закономірності в цих даних, щоб потім прогнозувати результати для нових, не бачених раніше даних. Основним завданням є зробити цю вибірку репрезентативною для реальних даних, з якими модель буде працювати в майбутньому.

– Валідаційна вибірка використовується для оцінки продуктивності моделі під час навчання. Саме за допомогою неї обираються найкращі

гіперпараметри моделі, що впливають на її роботу. Це допомагає уникнути перенавчання, а саме стану, коли модель занадто добре запам'ятовує навчальні дані, але показує поганий результат на нових даних.

– Тестова вибірка використовується для остаточної оцінки продуктивності моделі після завершення навчання. Оскільки тестова вибірка не використовується для навчання або налаштування моделі, вона дає найбільш об'єктивну оцінку її точності та продуктивності.

Прикладом оцінки якості алгоритмів є реалізація коефіцієнту схожості Dice [33], що використовується для оцінки якості сегментації. Він визначається як подвоєна площа перетину прогнозованого результату і очікуваного результату поділена на суму площ двох зображень або ж наступною формулою:

$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (3.1)$$

де A – це прогнозована маска сегментації,

B – істинна (очікувана) маска сегментації.

Для обрахунку коефіцієнту Dice для тестової вибірки зображень використовується підхід з середнім зваженим, де вагою для конкретного знімку є сума площ прогнозованої маски та очікуваної маски. Формула обрахунку представлена наступним виразом:

$$\bar{x} = \frac{\sum_{i=1}^n (S_1 + S_2)_i x_i}{\sum_{i=1}^n (S_1 + S_2)_i}, \quad (3.2)$$

де n – кількість зображень у наборі,

S_1 – площа прогнозованої маски,

S_2 – площа очікуваної маски,

x_i – обраховане значення коефіцієнту схожості Dice для конкретної прогнозованої і очікуваної масок.

Загалом, реалізація та практичне застосування алгоритмів цифрової обробки зображень є ключовим у розробці програмного продукту та має прямий вплив на успішність всього проєкту. В даному підрозділі було розглянуто декілька базових прикладів алгоритмів та підходів, що реалізуються в програмному забезпеченні та ще раз підкреслюють його практичну цінність. Правильний підбір та реалізація як і базових, так і інноваційних алгоритмів може зробити продукт проєкту ще більш конкурентоспроможним на ринку, значно покращити функціональність, зручність використання та задовольнити потреби користувачів.

3.5. Опис інтерфейсу користувача

Інтерфейс у програмному забезпеченні є дуже важливою складовою, оскільки він визначає спосіб, яким користувачі взаємодіють з програмою та її функціоналом. Чіткий та продуманий інтерфейс робить програму більш зрозумілою та привабливою для користувачів. Це спрощує навігацію між сторінками, полегшує доступ до функцій та в цілому робить використання програми більш комфортним. Гарно реалізований інтерфейс може значно підвищити задоволення користувачів від використання програми, зробити процес вивчення інтерфейсу більш швидким та сприяти популярності продукту.

Для виконання завдання із розробки інтерфейсу було використано макети, що дозволяють візуалізувати дизайн та функціонал продукту перед його реалізацією, та каркасний інтерфейс, що надає базову структуру та функціонал без візуальних деталей, таких як кольори, шрифти чи інші графічні елементи.

Перед безпосереднім створенням каркасного інтерфейсу та інших макетів було розроблено діаграму навігації – візуальне представлення структури інтерфейсу веб-системи, яке показує шляхи переміщення користувача між різними сторінками або екранами програми. Вона використовується для документування та аналізу структури і функціональності системи, а також

чіткого розуміння того як користувачі будуть взаємодіяти з нею. Результат показано на рис. 3.10.



Рис. 3.10. Діаграма навігації між сторінками програмного продукту

Більш детально деякі з сторінок програмного забезпечення розглянуто у вигляді каркасного інтерфейсу.

Першою з них є сторінка автентифікації та авторизації користувача (рис. 3.11). Вона відповідає за перевірку даних користувача (логіна та пароля) для підтвердження їх коректності, а також за визначення прав доступу після успішної автентифікації. Саме цей етап визначає, які ресурси та функціонал може використовувати користувач у системі.

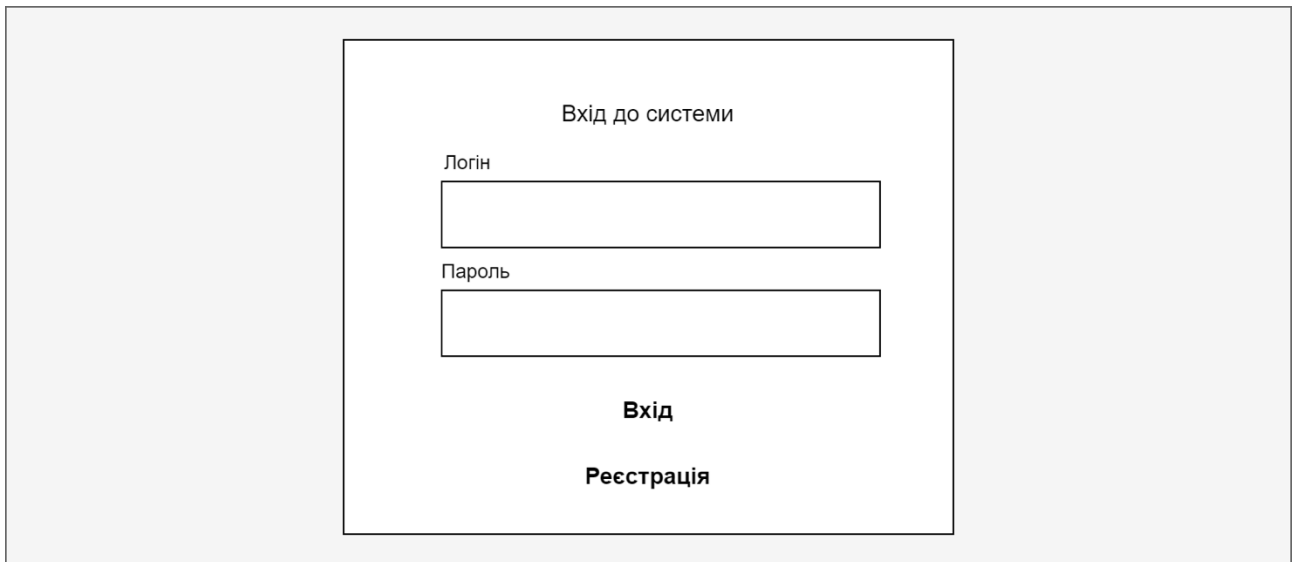


Рис. 3.11. Каркасный інтерфейс сторінки автентифікації та авторизації

Другою сторінкою, що розглядається є головна сторінка ПЗ, яка є вихідною точкою для інших дій та функціоналу у програму. Серед основних функцій: навігаційне меню, перелік останніх зображень, а також функціональні кнопки.

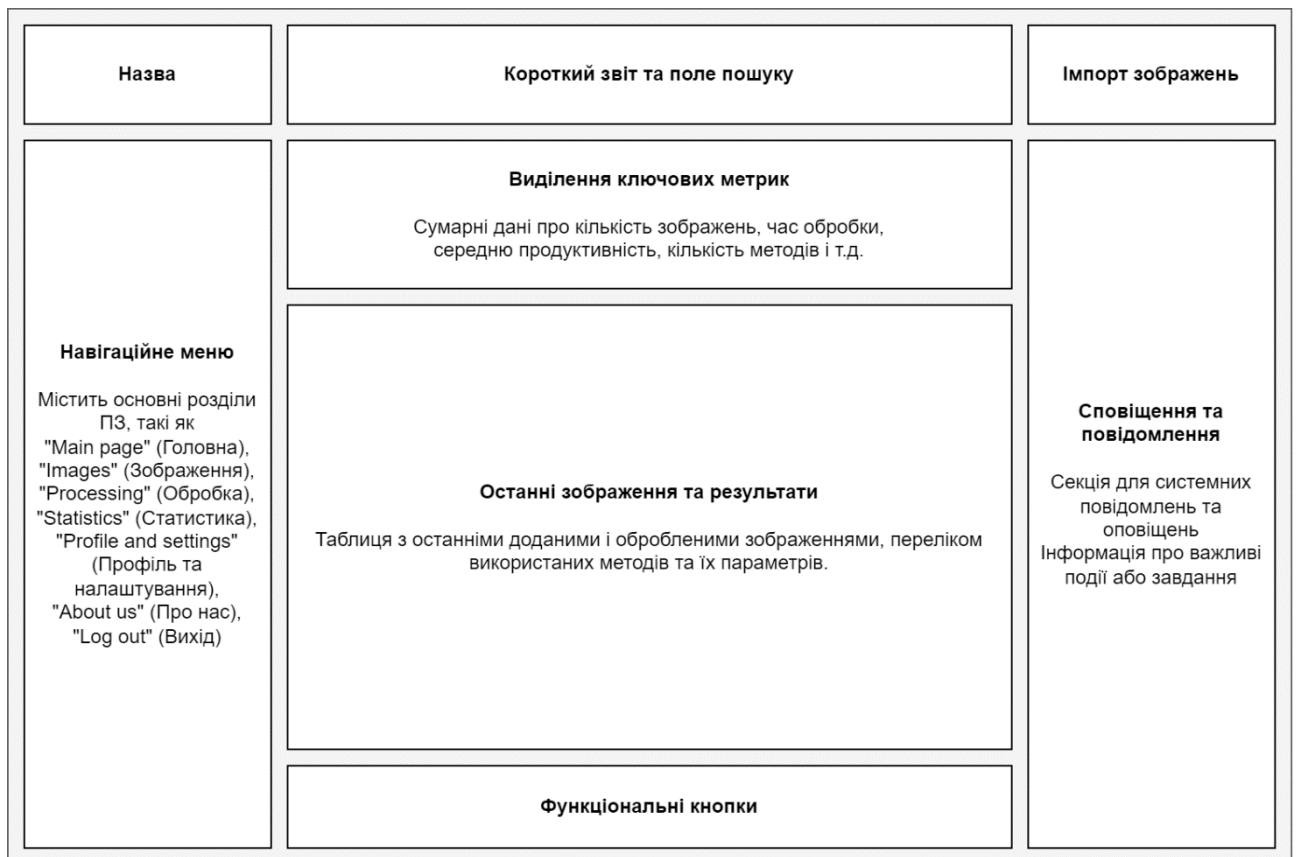


Рис. 3.12. Каркасный інтерфейс головної сторінки системи

Третьою сторінкою є сторінка перегляду зображень (рис. 3.13), де основним елементом є таблиця уже доданих чи оброблених зображень та можливість перегляду самих зображень та їх властивостей. Зі свого боку, сторінка налаштувань (рис. 3.14) передбачає зміну будь-яких налаштувань, що стосуються програми та конкретного профілю користувача.

Назва	Функціональне меню Пошук, фільтри та сортування списку зображень, налаштування колонок, перегляд зображень, інші функції							Імпорт зображення	
Навігаційне меню Містить основні розділи ПЗ, такі як "Main page" (Головна), "Images" (Зображення), "Processing" (Обробка), "Statistics" (Статистика), "Profile and settings" (Профіль та налаштування), "About us" (Про нас), "Log out" (Вихід)	<input type="checkbox"/>	Зображення	Назва файлу	Тип	Дата створення	Розмір
Таблиця зображень									

Рис. 3.13. Каркасний інтерфейс сторінки перегляду зображень

Назва	Основні налаштування Мова, тема, колірна палітра, робочі години		Параметри вибраного акаунта Фото, статус, зміна паролю	
Навігаційне меню Містить основні розділи ПЗ, такі як "Main page" (Головна), "Images" (Зображення), "Processing" (Обробка), "Statistics" (Статистика), "Profile and settings" (Профіль та налаштування), "About us" (Про нас), "Log out" (Вихід)	Основні глобальні налаштування обробки зображень Налаштування, сортування за замовчуванням, типи зображень, вихідна якість, колірний простір, сценарії автоматизації та інше		Налаштування сповіщень та повідомлень Вибір типу відображення, частота,	
			Налаштування пов'язаних програм та інтеграція Можливість підключення інших інструментів або сервісів тощо	
			Налаштування безпеки та доступів Шифрування даних, надання дозволів, управління рівнями доступу до різних частин системи	

Рис. 3.14. Каркасний інтерфейс сторінки налаштувань

Додатково до каркасних інтерфейсів деяких сторінок програмного забезпечення було розроблено функціональний прототип сторінки обробки зображень. Він відповідає за показ достовірного результату використання методів, а також реалізацію можливості виклику функцій з різними аргументами та в різному порядку. Серед базових функцій варто виділити відкриття та збереження зображення, скидання обробленого зображення до оригіналу, перегляд вихідного зображення та скасування попередньої дії. Після вибору користувачем методу та його параметрів і натискання відповідної кнопки в інтерфейсі програми обчислюється результат, який потім можна переглянути та оцінити. Загалом, прототип розроблений таким чином, щоб бути ефективним, уникати помилок та зависань і при цьому залишатись максимально зрозумілим для користувача, достатньо гнучким і розширюваним в розробці

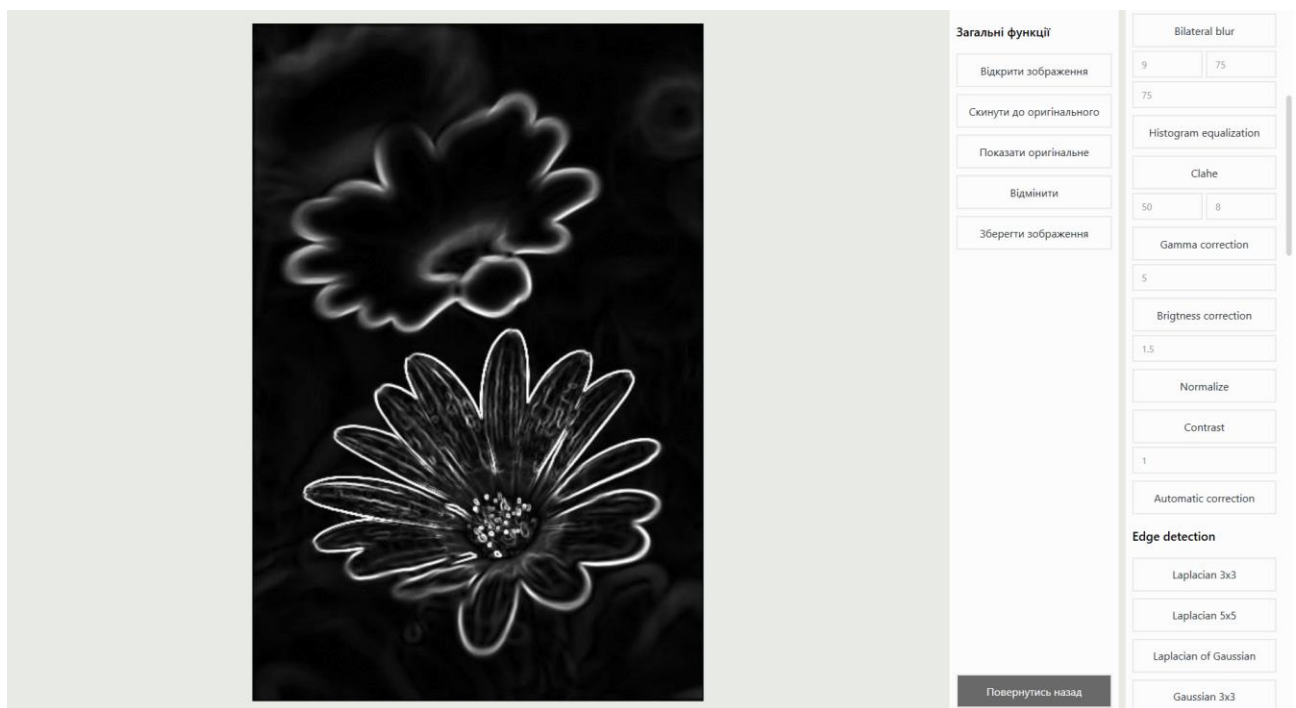


Рис. 3.15. Приклад функціональної сторінки обробки зображень

Серед головних користувацьких функцій, які передбачені інтерфейсом є:

- завантаження зображень з комп'ютера або інших джерел для подальшої обробки;

- вибір різних алгоритмів та методів для обробки зображення, а також зміна їх вхідних аргументів;
- виконання різних операцій редагування, таких як зміна розміру, обертання, вирівнювання тощо;
- можливість попереднього перегляду зображення до та після обробки для оцінки результату;
- збереження оброблених зображень у різних форматах;
- використання автоматичних налаштувань для швидкої та простої обробки;
- перегляд та відновлення попередніх дій та змін над зображеннями;
- перегляд таблиці усіх доданих зображень, методів та їх параметрів;
- зміна налаштувань інтерфейсу та функціоналу програми;
- навігаційне меню для спрощення переходу між сторінками;
- формування статистики, графіків та метрик щодо виконання операцій;
- перегляд сповіщень та повідомлень від програми;
- функції адміністрування та управління користувачами для адміністраторів системи.

Загалом, графічний інтерфейс користувача програмного забезпечення для цифрової обробки зображень буде виконано в єдиному стилі, він містить усі необхідні основні елементи користування та буде адаптивним. Під час виконання етапів розробки та дизайну також враховуються майбутні можливості розширення функціоналу, наприклад, інтеграцію сторонніх плагінів.

РОЗДІЛ 4

ПЛАНУВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ ПРОЄКТОМ

4.1. Структура декомпозиції робіт по проєкту

Структура декомпозиції робіт (WBS) – метод розбиття робіт проєкту на чітко визначені та керовані компоненти [3]. PMBOK визначає WBS як “ієрархічну декомпозицію всього обсягу робіт, який команда проєкту повинна виконати для досягнення цілей проєкту та створення необхідних доробків” [29]. Це дозволяє візуалізувати роботи проєкту, його вміст та взаємозв’язки між частинами. Іншою причиною створення WBS є розбиття робіт проєкту на менші, більш керовані завдання, які можна оцінити в часі, ресурсах та вартості, що значно спрощує планування та контроль виконання проєкту. WBS також дозволяє керівникам проєкту приймати кращі рішення щодо розподілу ресурсів, пріоритетів та ризиків, а в подальшому використовується для визначення складу команди проєкту та календарно-ресурсного планування.

Всього існує декілька типів WBS проєкту:

- по продуктах проєкту – структурує роботи проєкту згідно з основних кінцевих результатів або продуктів, які очікується отримати;
- по фазах життєвого циклу – структурує роботи проєкту відповідно до основних етапів або фаз життєвого циклу, що проходить проєкт;
- по процесах управління проєктом – структурує роботи по окремих процесах або робочих потоках, які задіяні в проєкті [3].

Проєкт розробки програмного забезпечення для цифрової обробки зображень має на меті створити продукт, що складається з декількох головних складових описаних у розділах раніше. Відповідно на основі цього було виконано декомпозицію робіт за продуктами (рис. 4.1.).

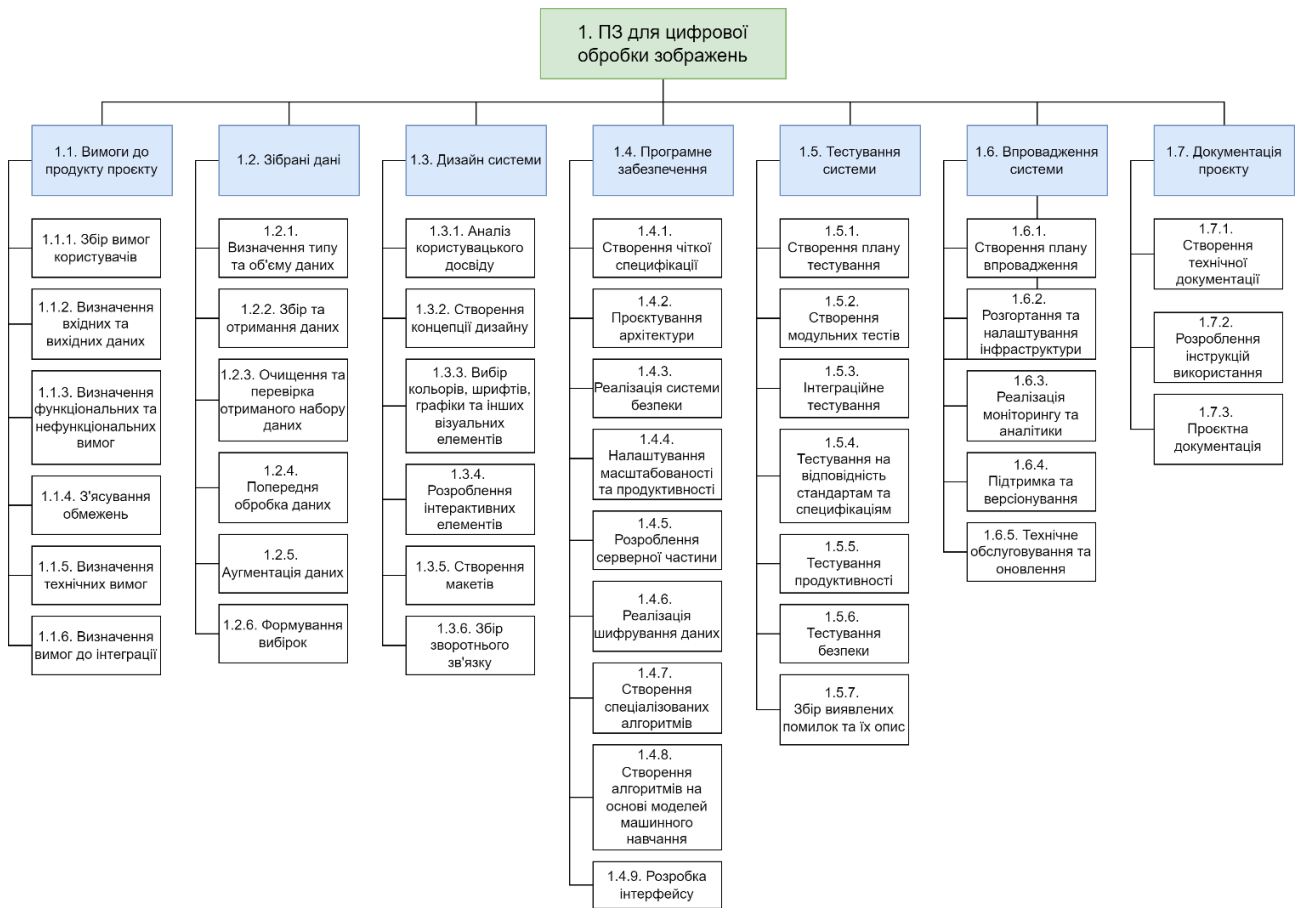


Рис. 4.1. Структура декомпозиції робіт по продуктах проекту

Також було створено структуру декомпозиції робіт проекту по фазах життєвого циклу та виділено 5 фаз: ініціалізація, планування та проектування, виконання та контроль, впровадження та завершення. Отриманий результат показано на рис. 4.2.

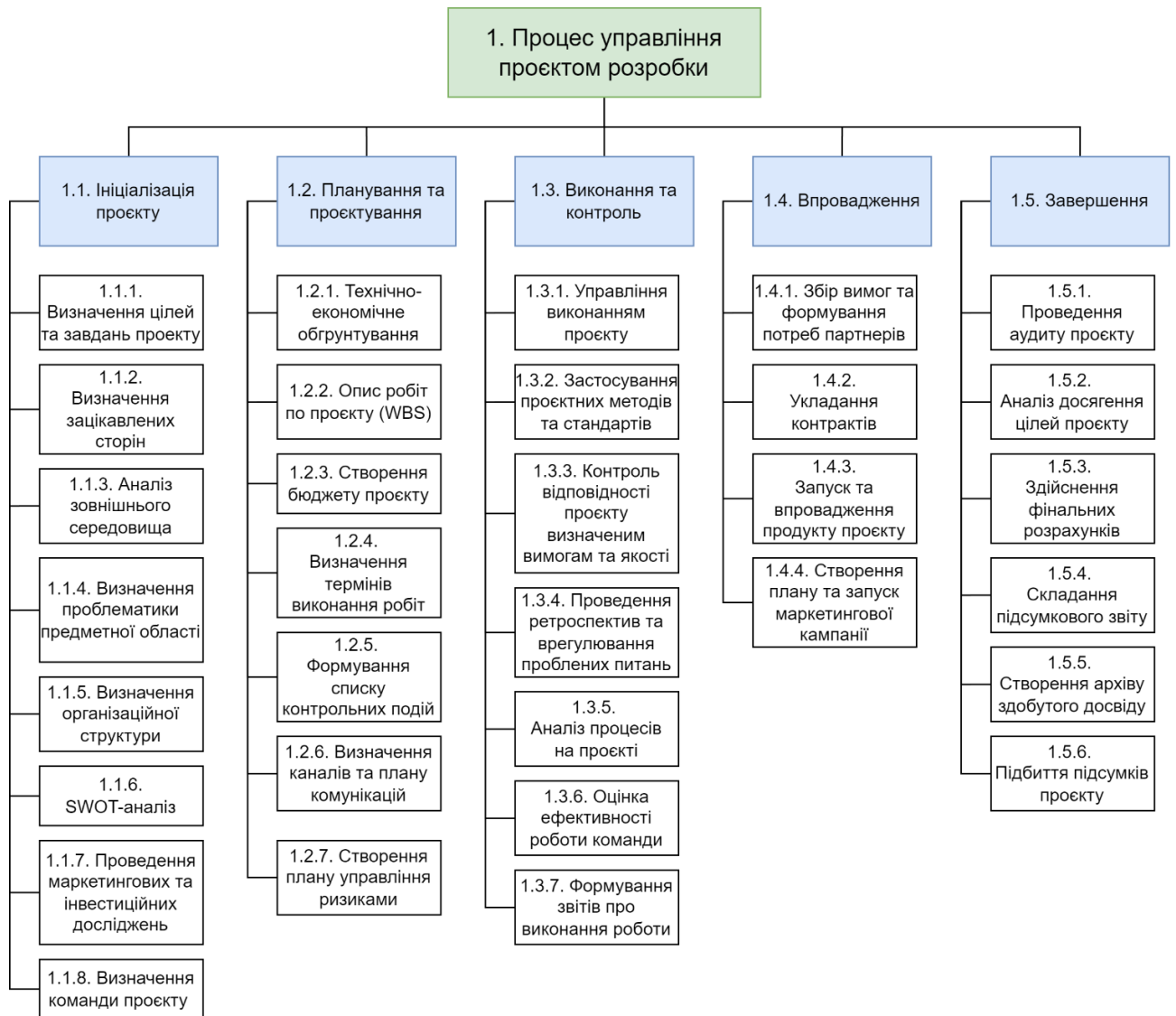


Рис. 4.2. Структура декомпозиції робіт по фазах життєвого циклу проекту

4.2. Організаційна структура та формування команди проекту

Відповідно до визначених робіт по проекту, їх характеристик та обсягу стає зрозуміло, що правильна організаційна структура та склад команди проекту є критичними факторами для успішної реалізації проекту. Швидкість виконання робіт та якість фінального продукту в значній мірі залежать від компетентності команди, їхнього релевантного досвіду та рівня знань. Для даного проекту дуже важливим завданням є створення ефективної та мотивованої команди, що об'єднана спільною стратегічною метою реалізації проекту. Проект буде

розробляться відповідно до проєктної структури, яка є найкращою для управління унікальними, складними проєктами, які мають великий бюджет або велике значення для організації [17].

На рис. 4.3. зображено організаційну структуру та склад команди проєкту зі створення та впровадження ПЗ для цифрової обробки зображень.

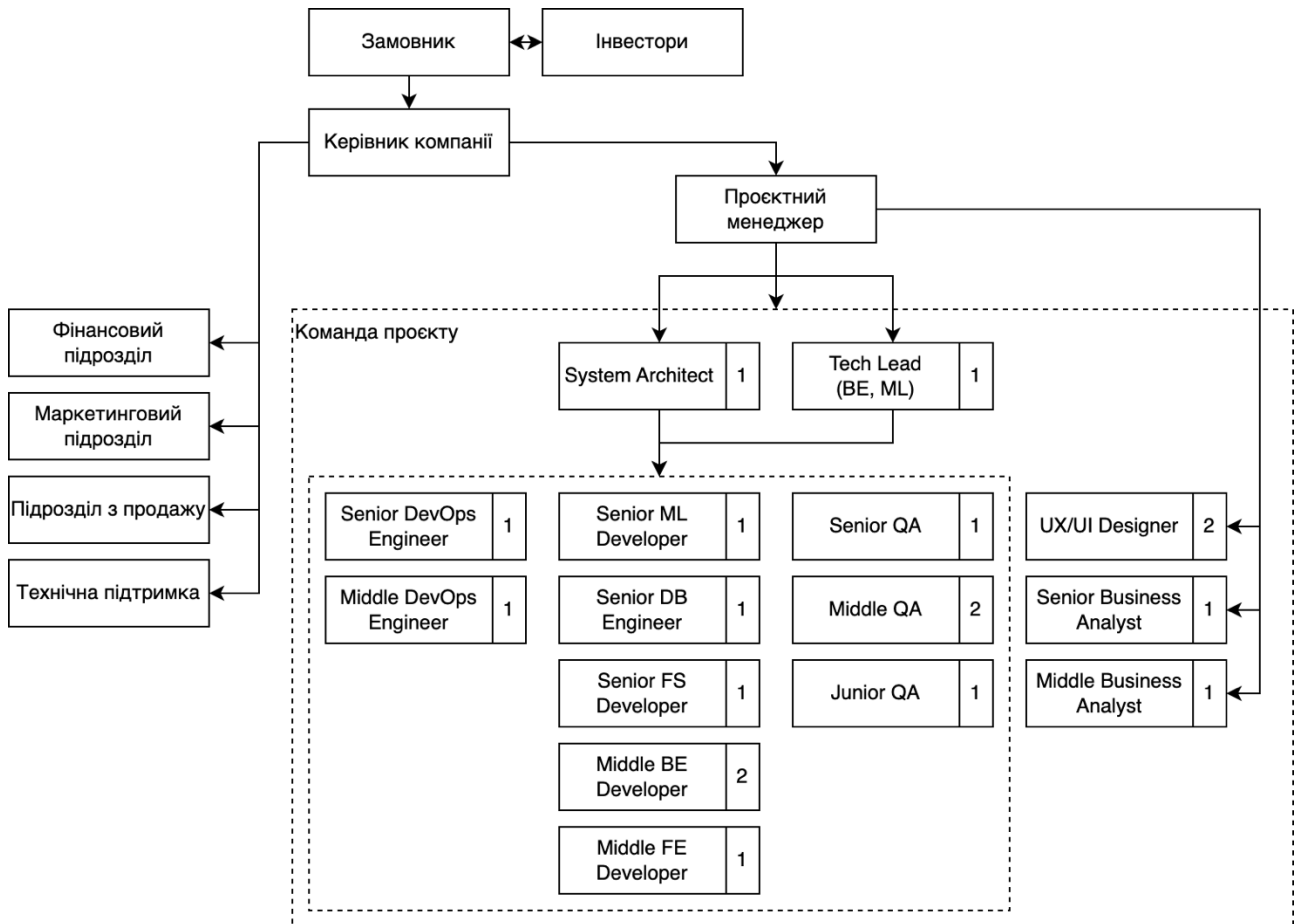


Рис. 4.3. Організаційна структура та команда проєкту

Відповідно до діаграми, ключовими особами, що приймають рішення у команді проєкту будуть безпосередньо керівник проєкту, системний архітектор та тех лід. Саме вони утворюють керівну структуру проєкту. Також на діаграмі показано перелік технічних спеціалістів та інших учасників проєкту, необхідних для його реалізації. Допоміжними структурними відділами компанії також є фінансовий, маркетинговий підрозділи, технічна підтримка та підрозділ з

продажу. Керівник компанії в свою чергу тісно взаємодіє із замовником, керівниками відповідних підрозділів та проєктним менеджером.

Для визначення відповідальності учасників проєкту буде використано матрицю відповідальності виду RACI – інструмент управління проєктами, який використовується для чіткого визначення ролей та обов'язків учасників проєкту щодо завдань та результатів [16].

- R – Responsible (особа, що відповідає за виконання завдання);
- A – Accountable (особа, що відповідає за успішне завершення завдання);
- C – Consulted (особа, що консультиє під час виконання завдання);
- I – Informed (особа, яку інформують про виконання завдання);

Матриця відповідальності низку ключових переваг:

- чіткість і узгодженість структури відповідальності учасників проєкту за реалізацію етапів проєкту;
- сприяння ефективній комунікації між учасниками проєкту, зниження ризику непорозумінь та затримок;
- виявлення потенційних ризиків завдяки чітко визначеним ролям, покращення розуміння ризиків пов'язаних з проєктом;
- прийняття кращих рішень щодо розподілу ресурсів та пріоритетів;
- спрощення моніторингу і контролю виконання завдань та допомога у вчасному виявленні можливих проблем.

Таблиця 4.1

Матриця відповідальності

№	Завдання	Керівник проєкту	Розробник	Тестувальник	Дизайнер	Бізнес-аналітик	DevOps	Архітектор	Тех лід
1	Управління проєктом	RA						C	C

Продовження табл. 4.1

2	Визначення вимог користувачів	A	I		C	R		C	C
3	Формулювання бізнес-вимог	A	I		I	R		C	C
4	Визначення інших вимог та обмежень	A	I		I	R		C	C
5	Визначення технічних вимог	A	C			R		C	C
6	Розробка технічного завдання	I						RA	C
7	Проектування системи та архітектури	I	I	I			I	RA	C
8	Аналіз досвіду користувачів	A			R			I	I
9	Збір даних та їх підготовка	I	R	I				A	I
10	Створення концепції дизайну	A			R	I		I	I
11	Створення макетів дизайну	A			R			I	I
12	Реалізація компонентів системи	I	R	I			R	RA	R
13	Розробка API	I	R				R	AC	C
14	Створення плану тестування	I		R		I		C	A
15	Створення бази даних та алгоритмів	I	R	I			C	AC	RC
16	Створення алгоритмів на основі ШІ	I	R	I			R	A	RC
17	Створення модульних тестів	I	I	R				I	A
18	Розробка серверної та клієнтської частини	I	R	I	C	I		AC	RC
19	Створення автоматизованих тестів	I	I	R			R	C	AC
20	Системне та інтеграційне тестування	I	I	R			I	I	AC
21	Виявлення та виправлення помилок	I	R	R		I		A	RC
22	Підготовка до впровадження	I	R	I			R	A	RC
23	Налаштування інфраструктури	I	C	C			R	A	C
24	Налаштування моніторингу та аналітики	I	R				R	A	C
25	Впровадження системи	I	R	I			R	A	C
26	Створення технічної документації	I	R	R		I	R	RA	RC
27	Технічне обслуговування та оновлення	I	R	R	C	C	R	A	C
28	Аудит проєкту	RA	C	C	C	R	C	C	C
29	Документування результатів проєкту	RA				R		C	C

Після створення цієї матриці варто відзначити що проєктний менеджер є відповідальним за успішне завершення проєкту в цілому, в той же час архітектор і тех лід відповідають за успішне завершення його окремих технічних частин,

зокрема завдань розробки, тестування та впровадження. В цілому, створення матриці відповідальності дозволяє зробити процес керування проєктом більш ефективним і прозорим та чітко визначити залученість тих чи інших членів команди до окремих завдань і процесів.

4.3. Вибір підходу до розробки програмного забезпечення

Методологія управління проєктом визначається як підхід до організації робіт для виробництва продукту або послуги в межах проєкту. Методологія каже про те як виконуються обов'язкові практики управління проєктами, які відображені у стандартах прокатного менеджменту. Методології, що застосовуються в проєктах, мають тенденцію безпосередньо впливати на ефективність та результативність результатів проєкту. А отже, впливають на вартість, якість, обсяг, час і результати. Правильно визначена і суворо дотримувана методологія управління проєктами дає тверду гарантію того, що робота буде виконана вчасно, в рамках бюджету і відповідно до встановлених вимог. Підґрунтям такої гарантії є ефективне використання сукупності правил, принципів, методів, інструментів, які дозволяють управляти проєктом протягом усіх фаз і стадій його життєвого циклу та дозволяють зменшити ймовірність виникнення невдач та послабити вплив ризиків.

Вибір методології управління проєктом суттєво впливає на те, як команда буде організована, як вона буде взаємодіяти, та які інструменти будуть використовуватися для досягнення цілей. Методологія враховує специфіку проєкту, визначаючи підхід до роботи команди, форми контролю, перевірки та оцінки результатів. Від правильної методології залежать якість планування та виконання проєкту, а також його кінцевий успіх.

Враховуючи складність та важливість проєкту, доречним буде застосування гнучкого підходу до розробки програмного забезпечення. Гнучкі методології відомі своєю гнучкістю та адаптивністю, що може бути особливо

корисним при розробці складних систем. Найбільш підходящим для даного типу проєкту є Scrum – підхід до управління проєктами для гнучкої розробки програмного забезпечення, в основі якого лежить ітераційна розробка, постійне вдосконалення продукту та співпраця між командою розробників і зацікавленими сторонами [19]. Scrum має ряд переваг, завдяки яким він добре підходить для такого проєкту.

- Акцент на ітераціях з обмеженим часом, що означає, що проєкт можна розбити на менші, керовані частини, які можна завершити за короткі ітерації, як правило, 2-4 тижні. Такий підхід дозволяє за потреби вносити зміни в обсяг проєкту, вимоги і пріоритети та частіше отримувати зворотній зв'язок від зацікавлених сторін, що може допомогти забезпечити відповідність продукту потребам користувачів та замовників [2].

- Забезпечення найкращої якості продукту завдяки частому тестуванню та перевірці протягом усього процесу розробки, що призводить до меншої кількості дефектів.

- Забезпечення прозорості через регулярні зустрічі, що допомагає всім у команді проєкту мати актуальну інформацію про стан проєкту та їхню роль у ньому.

- Підвищена залученість, більша мотивація команди, відсутність "бюрократизації" та самоорганізація, що сприяють швидкій та продуктивній розробці продукту.

На рис. 4.4. показано процес виконання спринту у проєкті, його складові, обов'язкові зустрічі, а також учасників.

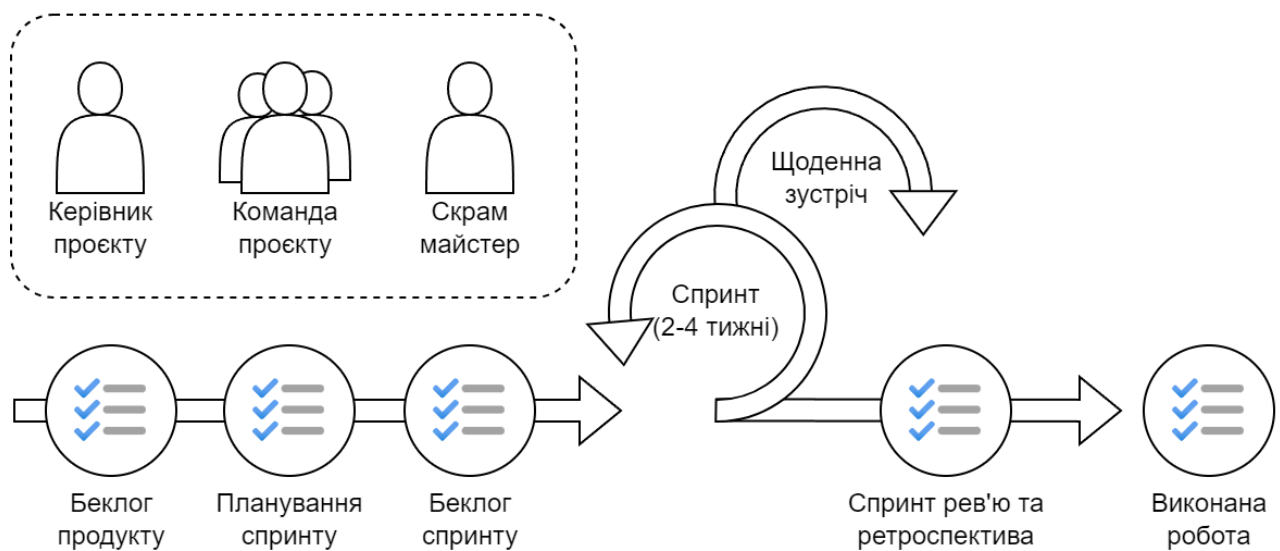


Рис. 4.4. Діаграма виконання спринту у проєкті

Проте варто зауважити, що Scrum найкраще підходить досвідченим, дисциплінованим і вмотивованим командам, які вміють розставляти свої пріоритети та мають чітке уявлення про вимоги проєкту. Отже, це є важливим при підборі спеціалістів та формуванні команди проєкту.

4.4. Календарно-ресурсне планування

Для успішного та коректного виконання календарно-ресурсного планування перш за все необхідно розглянути життєвий цикл проєкту та ключові віхи проєкту. Життєвий цикл проєкту визначає послідовність етапів, через які проєкт пройде від початку до завершення, визначає ключові кроки, вимоги на кожному етапі та ключові точки контролю, які допомагають відстежувати прогрес та дозволяють забезпечити успішне виконання проєкту. Життєвий цикл проєкту може мати різну структуру та кількість етапів залежно від специфіки проєкту, його масштабів та складності. В даному випадку для проєкту з розробки програмного забезпечення для цифрової обробки зображень будуть наявні такі етапи:

- ініціація: визначається мета проекту, проводиться оцінка його життєздатності, формується команда та розробляється початковий план.
- планування: детально розробляється план проекту, який включає в себе визначення завдань, ресурсів, термінів, бюджету, ризиків та інших факторів.
- виконання та контроль: на цьому етапі відбувається безпосереднє виконання робіт проекту згідно з планом, відстежується прогрес проекту, його відповідність плану, виявляються та вирішуються проблеми, вносяться необхідні корективи;
- впровадження: створюється план впровадження, відбувається розгортання програмного забезпечення, яке стає доступним для користувачів та інтеграції з партнерами;
- завершення: проект офіційно завершується, відбувається фінальна підготовка документації, підбиваються підсумки, оцінюється його успішність, готується звітність;

Також через специфіку проекту та застосування методології Scrum, деякі етапи будуть проходити ітеративно, тобто повторюватися кілька разів з поступовим уточненням та вдосконаленням, а саме етапи виконання та контролю, впровадження. Життєвий цикл проекту показано на рис. 4.5.

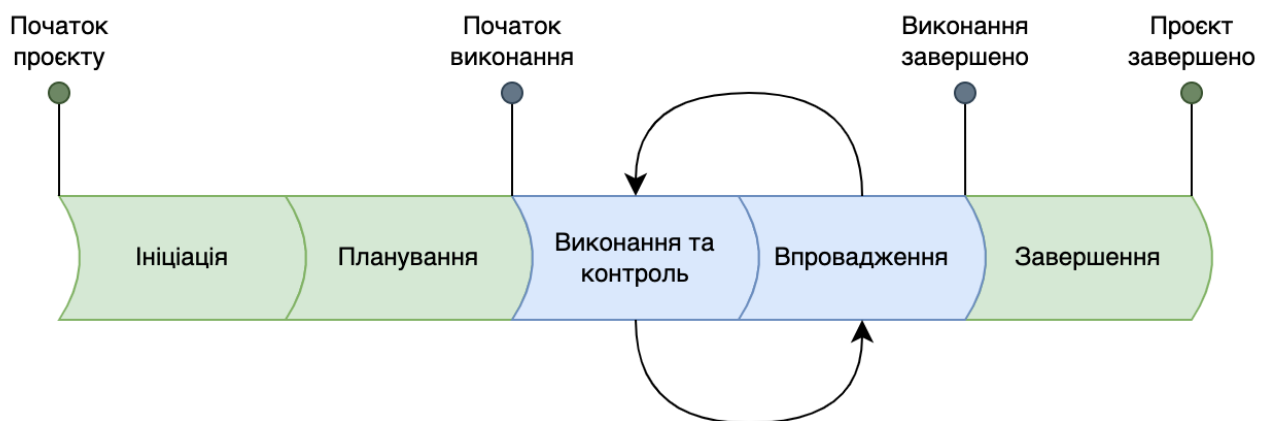


Рис. 4.5. Життєвий цикл проекту

Наступним кроком для розробки плану проекту є визначення ключових віх проекту на основі структури декомпозиції робіт. Результат показано в табл. 4.2.

Віхи проєкту

№	Календарна дата	Назва події (віха)
	05.02.2024	Старт проєкту
	12.02.2024	Керівника проєкту призначено
	26.02.2024	Проведено аналіз предметної області, визначено проблематику
	11.03.2024	Вимоги до продукту, основні характеристики, цілі проєкту визначено
	18.03.2024	Концепцію проєкту затверджено
	25.03.2024	Створено паспорт проєкту
	01.04.2024	Процеси управління проєктом визначено
	15.04.2024	Технічне завдання розроблено
	06.05.2024	Архітектуру системи спроектовано
	13.05.2024	Дизайн системи та макети графічного інтерфейсу завершено
	13.05.2024	Визначено перелік алгоритмів для першочергової розробки
	13.05.2024	Розпочато реалізацію компонентів програмного забезпечення
	14.05.2024	Розпочато розробку API
	14.05.2024	Розпочато проєктування бази даних
	27.05.2024	Створено набори даних для подальшого використання
	01.07.2024	Створено план тестування
	22.07.2024	Розроблено першочергові алгоритми на основі моделей ШІ
	22.07.2024	Проєктування бази даних завершено
	29.07.2024	Модульні тести реалізовано
	29.07.2024	Початок розробки серверної та клієнтської частини
	19.08.2024	Завершено розробку API
	09.09.2024	Завершено розробку серверної та клієнтської частини
	23.09.2024	Завершено розробку алгоритмів
	30.09.2024	Завершено реалізацію автоматизованих та інтеграційних тестів
	30.09.2024	Створено план впровадження ПЗ
	07.10.2024	Хмарну інфраструктуру налаштовано
	14.10.2024	Проведено аналіз ефективності роботи системи
	21.10.2024	Старт залучення клієнтів
	21.10.2024	Виконано інтеграцію та тестування цілісного продукту
	21.10.2024	Моніторинг та логування налаштовано
	28.10.2024	Випущено та впроваджену MVP версію системи
	28.10.2024	Завершено створення технічної документації
	11.11.2024	Виконано аудит проєкту
	18.11.2024	Завершено документування результатів проєкту
	25.11.2024	Визначено план наступних дій
	25.11.2024	Проєкт завершено

Проект розробки програмного забезпечення для цифрової обробки зображень має на меті створити кілька ключових компонентів, які разом утворюють повну систему, проте можуть використовуватись незалежно, відповідно до потреб користувачів. Остання віха "проект завершено" не вказує на те, що розробка продукту завершена назавжди, швидше вказує на те, що буде створено новий проект відповідно до нових вимог та викликів, який продовжить підтримку та розробку нових функцій, ітераційно покращуючи продукт.

Для календарного планування було використано систему управління проектами Microsoft Project [40], а для відображення графіка – діаграму Ганта (рис. 4.6.) [12]. Її основна мета полягає у представленні послідовності завдань, термінів виконання та залежностей між ними. Для створення використовувались структури декомпозиції робіт, ключові віхи проекту та склад команди проекту. Додатково на діаграмі показано критичний шлях, який визначає мінімальний час, необхідний для завершення проекту. Якщо будь-яке завдання на критичному шляху затримається, це призведе до затримки всього проекту. Тому важливо добре відстежувати та керувати цими завданнями для його вчасного завершення.

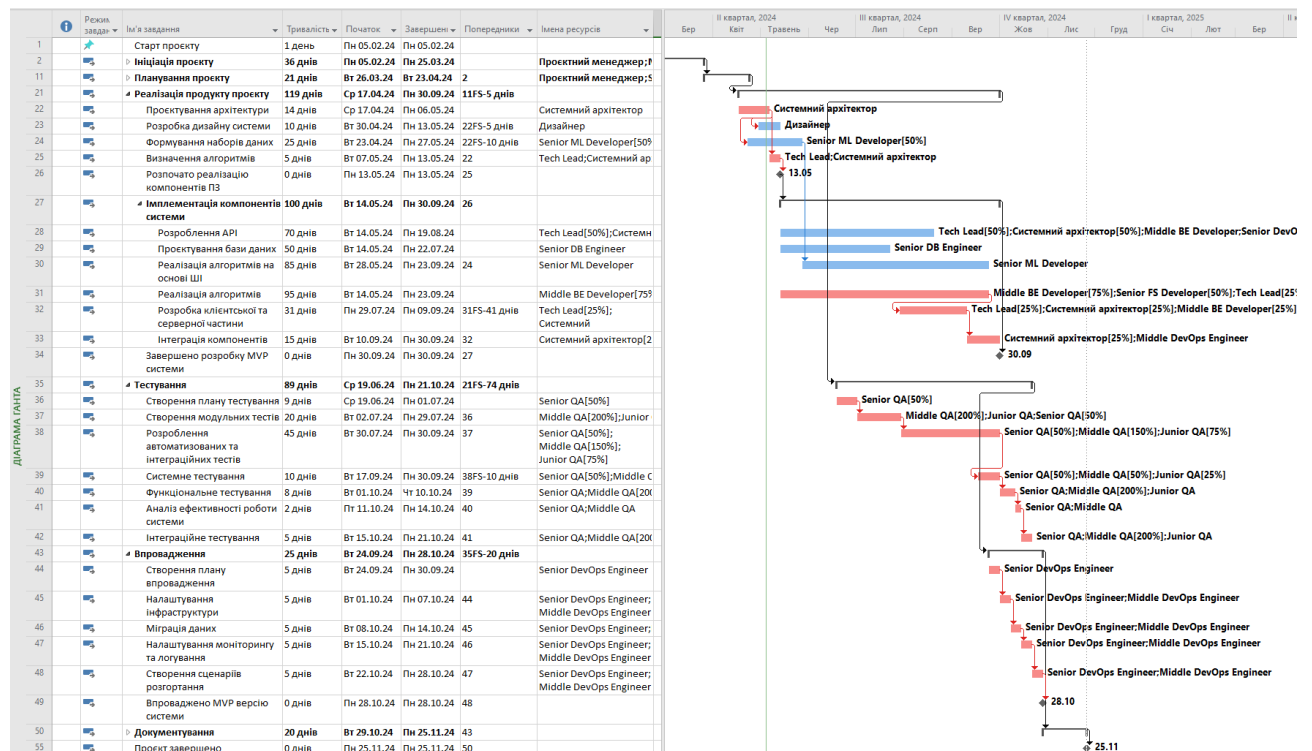


Рис. 4.6. Діаграма Ганта для проекту створення інформаційної системи

4.5. Оцінка витрат проєкту

Додатковими витратами для реалізації проєкту, окрім витрат на команду проєкту є:

- апаратне забезпечення та інфраструктура: придбання та налаштування необхідної апаратної інфраструктури, такої як сервери, пристрої зберігання даних, мережеве обладнання та інші апаратні компоненти, необхідні для підтримки роботи внутрішньої системи;

- програмне забезпечення та ліцензії: придбання ліцензій для різних програмних компонентів та інструментів, необхідних для розробки та функціонування системи, включно з операційними системами, веб-серверами, системами управління базами даних, фреймворками штучного інтелекту та іншими програмними залежностями;

- хмарні сервіси: витрати, пов'язані з постачальниками хмарних послуг, таких як інфраструктура як послуга (IaaS) або платформа як послуга (PaaS), які забезпечують масштабованість, зберігання та обчислювальні ресурси;

- дані: вартість інтеграції зовнішніх джерел даних або API для розширення даних системи та покращення її діагностичних можливостей. Це може включати збір даних, очищення даних та інтеграцію даних;

- навчання та підтримка користувачів: витрати на проведення тренінгів, документацію та технічну підтримку кінцевих користувачів, які будуть взаємодіяти з системою;

- маркетингові та PR витрати: витрати на рекламу, створення контенту, прямі маркетингові витрати, пошукова оптимізація (SEO), проведення вебінарів та онлайн-заходів;

- непередбачувані витрати.

Використовуючи створений календарний графік робіт, організаційну структуру команди та декомпозицію робіт було обраховано усі витрати пов'язані з реалізацією проєкту, як показано у табл. 4.3.

Таблиця 4.3

Оцінка витрат проєкту

Назва витрат	Ціна за одиницю, \$/міс.	Кількість	Тривалість, міс.
Проектний менеджер	3 200	1	10
Команда проєкту			
System Architect	6 500	1	10
Tech Lead (BE, ML)	5 400	1	10
Senior DevOps Engineer	5 500	1	7
Middle DevOps Engineer	3 000	1	6
Senior ML Developer	4 900	1	6
Senior DB Engineer	4 750	1	3
Senior FS Developer	5 000	1	7
Middle BE Developer	2 700	2	7
Middle FE Developer	2 500	1	7
Senior QA	3 200	1	6
Middle QA	1 850	2	6
Junior QA	800	1	6
UX/UI Designer	1 850	2	2
Senior Business Analyst	2 550	1	6
Middle Business Analyst	2 250	1	6
Всього	\$ 423 850		
Інші витрати			
Апаратне забезпечення та інфраструктура	1 000	-	10
Програмне забезпечення та ліцензії	500	-	10
Хмарні сервіси та дані	5 000	-	7
Навчання та підтримка користувачів	1 000	-	1
Маркетингові та PR витрати	15 000	-	3
Непередбачувані витрати	10 150		
Всього	\$ 106 150		
Загалом	\$ 530 000		

В результаті отримано загальну вартість проєкту, що дорівнює \$ 530 000. Детальна і точна оцінка загальної вартості дозволить ефективно керувати бюджетом, уникнути непередбачених витрат, а також забезпечити фінансову стабільність і успіх проєкту.

4.6. Управління якістю проєкту

Управління якістю в процесі управління проєктами передбачає систематичний підхід до забезпечення того, щоб результат проєкту відповідав вимогам якості. Це включає в себе планування, контроль і забезпечення якості на всіх етапах проєкту, з метою досягнення поставлених цілей і задоволення вимог замовника.

Перш за все, було визначено вимоги зацікавлених сторін до якості кінцевого продукту, що показано в табл. 4.4.

Таблиця 4.4

Вимоги зацікавлених сторін до якості проєкту

Пріоритет	Зацікавлена сторона	Вимоги до якості проєкту
1	Замовник	<ol style="list-style-type: none">1. Відповідність програмного забезпечення вимогами користувачів та бізнес-вимогам.2. Реалізація проєкту у визначені терміни, бюджет та з обговореним наповненням.3. Чітке планування проєкту, обґрунтування прийняття рішень.4. Залученість в процесі визначення вимог та контроль ходу виконання проєкту.5. Наявність плану управління ризиками.6. Конкурентоспроможність фінального продукту.
2	Інвестори	<ol style="list-style-type: none">1. Прозорість звітування та використання коштів.2. Прибутковість проєкту.3. Інноваційність, розвиток галузі, стратегічна цінність.4. Чіткий бізнес-план та шляхи монетизації.5. Чітке визначення ризиків та шляхів управління ними.6. Окупність проєкту у визначені терміни.
3	Регуляторні органи	<ol style="list-style-type: none">1. Відповідність законодавству, регулятивним вимогам та стандартам галузі.2. Наявність необхідної документації та можливості проведення аудитів.
4	Користувачі	<ol style="list-style-type: none">1. Доступ до системи з будь-якого пристрою.2. Зрозумілість та адаптивність графічного інтерфейсу.3. Зрозумілість методів, їх налаштувань та параметрів.

		<p>4. Наявність великої кількості функцій, їх поділ на категорії, наявність опису функціоналу для кожної.</p> <p>5. Висока якість вихідного результату, відсутність артефактів і втрати деталей.</p> <p>6. Стабільна робота програми без збоїв навіть при тривалому користуванні.</p> <p>7. Доступна цінова політика та наявність безкоштовної версії програми.</p> <p>8. Хороша швидкість виконання операцій навіть при обробці великих зображень чи великої кількості зображень.</p> <p>9. Система повинна відповідати вимогам безпеки та конфіденційності.</p> <p>10. Наявність базових функцій фільтрування, сортування та пошуку зображень.</p>
5	Компанії-партнери	<p>1. Наявність якісної та однозначної документації по API та інших складових продукту.</p> <p>2. Можливості масштабування, налаштування та інтеграції.</p> <p>3. Сумісність з технологічною платформою партнерів.</p> <p>4. Відповідність стандартам галузі, забезпечення безпеки даних та конфіденційності.</p> <p>5. Відповідність функціональним вимогам.</p> <p>6. Технічна підтримка та взаємодія з розробниками ПЗ.</p>
6	Виконуюча організація	<p>1. Проєкт повинен бути успішним та відповідати вимогам замовника, користувачів та інвесторів.</p> <p>2. Проєкт повинен мати чіткий план управління та однозначне календарно-ресурсне планування.</p> <p>3. Зрозумілий roadmap проєкту та продукту, шляхи подальшого розвитку та реалізації.</p> <p>4. Визначені канали комунікації.</p>
7	Команда проєкту та керівник проєкту	<p>1. Чіткість та зрозумілість поставлених вимог і завдань.</p> <p>2. Сучасний та обґрунтований технологічний стек.</p> <p>3. Наявність актуальної технічної документації.</p> <p>4. Легкість розширення та модифікації ПЗ.</p> <p>5. Можливість професійного росту.</p> <p>6. Успішна реалізація проєкту відповідно до обмежень.</p>
8	Сторонні постачальники даних та послуг	<p>1. Використання даних та послуг у проєкті з розробки ПЗ.</p>

Продуктом даного проєкту є програмне забезпечення для цифрової обробки зображень у вигляді веб-системи. Відповідно до вимог зацікавлених

сторін можливо визначити, які формальні модулі матиме продукт, навести їх опис та вимоги до якості. Результати наведено в табл. 4.5.

Таблиця 4.5

Вимоги до якості модулів проекту

Пріоритет	Модуль	Вимоги до якості
1	Прикладний програмний інтерфейс	<ol style="list-style-type: none"> 1. Наявність документації та чіткого версіонування. 2. Забезпечення безпеки, захист від несанкціонованого доступу. 3. Масштабованість, здатність обробляти велику кількість запитів, стабільність роботи і підтримки. 4. Ефективність у використанні ресурсів, пам'яті, процесору, швидкий доступ до даних. 5. Визначення базових інтерфейсів та шляхів інтеграції.
2	Підсистема обробки цифрових даних	<ol style="list-style-type: none"> 1. Ефективна швидкість обробки цифрових даних, відповідно до вибраних методів та параметрів. 2. Можливість вибору методу, зображення, визначення параметрів. 3. Збереження проміжних результатів та етапів виконання операцій. 4. Детермінований результат у випадку використання не ШІ-алгоритмів. 5. Перелік визначених обмежень для ШІ-алгоритмів. 6. Відмовостійкість та надійність підсистеми під час виконання операцій. 7. Відкритий інтерфейс виклику та створення методів.
3	Підсистема ведення і завантаження даних	<ol style="list-style-type: none"> 1. Можливість додавання зображень різних форматів (JPEG, PNG, BMP, TIFF, RAW, HEIC). 2. Зчитування метаданих зображень. 3. Збереження усіх даних в базі даних. 4. Надійність роботи та відсутність збоїв при роботі із зображеннями різного розміру та формату. 5. Можливість завантаження цілих груп зображень одночасно та паралельно.
4	Підсистема виводу даних	<ol style="list-style-type: none"> 1. Відображення збережених та оброблених даних. 2. Візуалізація графіків статистики і даних моніторингу, їх форматування. 3. Збереження та експорт даних у різних форматах.
5	Модуль керування користувачами	<ol style="list-style-type: none"> 1. Автентифікація (в тому числі двофакторна) та авторизація. 2. Налаштування та керування профілями та системою. 3. Можливість реєстрації та входу до системи. 4. Ведення журналу виконаних дій та історії. 5. Функція відновлення паролю.

6	Графічний інтерфейс	<ol style="list-style-type: none"> 1. Підтримка різних мов та регіональних налаштувань. 2. Єдиний стиль дизайну та графічних елементів. 3. Реалізація елементів пошуку, сортування, фільтрування. 4. Естетична привабливість та сучасний стиль. 5. Легкість використання та навігації між сторінками. 6. Доступність інтерфейсу для усіх категорій користувачів. 7. Швидке реагування на дії користувача, відсутність зависань. 8. Можливість налаштування інтерфейсу, зміни теми та переміщення функціональних елементів. 9. Адаптивність інтерфейсу на різних пристроях.
7	Підсистема моніторингу	<ol style="list-style-type: none"> 1. Анонімізація усіх даних відповідно до вимог та стандартів. 2. Збір метрик та їх візуалізація. 3. Виявлення помилок та нештатних ситуацій при роботі з системою. 4. Ведення журналу виконаних дій, логування. 5. Забезпечення безпеки усіх зібраних даних, їх шифрування. 6. Аналіз використаних апаратних ресурсів та швидкодії.
8	Модуль сповіщень	<ol style="list-style-type: none"> 1. Підтримка різних каналів сповіщень. 2. Забезпечення гарантованої доставки необхідних повідомлень чи сповіщень. 3. Керування підпискою та налаштування сповіщень. 4. Локалізація сповіщень та візуалізація різних типів.

Наступним кроком після збору вимог до якості зацікавлених сторін та модулів є розробка заходів контролю якості. Вони надають чіткий опис того, що повинно бути зроблено для забезпечення відповідності вимогам та якості проєкту і продукту проєкту. Загалом, вони допомагають забезпечити успіх продукту, в даному випадку програмного забезпечення, що відповідає очікування зацікавлених сторін та ринку в цілому. Також важливо відзначити те, що вони допомагають виявити та виправити помилки та дефекти в системі до того, як вона буде введена в експлуатацію, що значно економить час та гроші у довгостроковій перспективі. Результат розробки заходів контролю якості показано в таблиці Г.1.

Загалом, управління якістю в проєкті є важливим елементом успішного управління проєкту, оскільки дозволяє забезпечити відповідність результатів

проєкту вимогам замовника і стандартам якості, що в свою чергу сприяє задоволенню клієнтів, підвищенню репутації компанії та успіху продукту.

4.7. Управління ризиками проєкту

Ризики у проєкті – це потенційні негативні події, які можуть вплинути на досягнення цілей проєкту, викликати зміни в планах проєкту або призвести до негативних наслідків для проєкту [9]. Ризики можуть мати різні форми, такі як технічні проблеми, зміни вимог, фінансові затримки, конфлікти в команді та інші. Головна загроза ризику полягає у його потенційному негативному впливі на проєкт. Ризики можуть призвести до затримок у графіку, збільшення витрат, погіршення якості продукту або навіть до невдачі проєкту. Серед основних причин виникнення ризиків є невизначеність у проєкті, а саме невизначеність у постановці завдань, технологіях, ресурсах тощо. Також суттєвий вплив можуть мати економічні, політичні, соціальні чи технологічні фактори, що може призводити до виникнення негативних подій чи ризиків.

Управління ризиками у проєкті включає в себе ідентифікацію, оцінку, розробку протиризикових заходів та моніторинг для мінімізації їх впливу та забезпечення успішного завершення проєкту.

Ідентифікація ризиків передбачає категоризацію ризиків за типом джерела, створення списку ризикових подій які можуть виникнути протягом життєвого циклу проєкту, а також силу впливу та керованість. Особливо важливим на даному етапі є те, щоб процес ідентифікації ризиків був всеосяжним і усі потенційні ризики були виявлені та описані. Це значно допоможе команді проєкту краще їх оцінити та розробити план дій для їхньої мінімізації чи мінімізації їхнього впливу. Результат ідентифікації показано в табл. 4.6.

Ідентифікація ризиків

№	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	Технологічні ризики	Технологічна застарілість	Середній	Високий
2		Складність реалізації та оптимізації алгоритмів	Високий	Середній
3		Витік конфіденційної інформації	Середній	Високий
4		Надмірна залежність від сторонніх бібліотек та сервісів	Середній	Високий
5		Критичні баги	Високий	Високий
6	Організаційні ризики	Нестача кваліфікованих спеціалістів	Середній	Високий
7		Неефективне управління проектом	Високий	Високий
8		Недостатній досвід команди	Середній	Високий
9		Порушення правил кібербезпеки	Середній	Низький
10		Конфлікти всередині команди	Середній	Середній
11	Економічні та політичні ризики	Зміна конкурентного середовища	Високий	Низький
12		Економічний спад	Середній	Низький
13		Політичні та регуляторні зміни	Низький	Низький
14	Форс мажори	Стихійні лиха	Середній	Низький
15		Воєнні дії	Високий	Низький
16	Комерційні ризики	Недосконалість бізнес-моделі	Високий	Високий
17		Нестача фінансування	Високий	Середній
18		Недостатній попит	Середній	Середній
19		Нечітке позиціонування на ринку	Низький	Середній

Наступним кроком після ідентифікації ризиків є їх оцінка – процес визначення та оцінки ймовірності та впливу потенційних негативних подій. Він допомагає серед усіх ризиків визначити найбільш пріоритетні з точки зору загроз та втрат. Для оцінки ризикової події проставляються оцінки від 1 до 10 (квазікількісна оцінка), що оцінують ймовірність настання, фінансові втрати, затримки в часі та частоти виникнення в проекті. Сила впливу на проект є комплексним показником і обчислюється як добуток ймовірності на фінансові втрати [9]. Оцінювання ризиків проекту показано в табл. 4.7.

Оцінювання ризиків

№	Ризикова подія	Ймовірність	Фін. втрати	Затримки в часі	Частота (за проєкт)	Сила впливу
1	Технологічна застарілість	3	6	5	2	18
2	Складність реалізації та оптимізації алгоритмів	5	8	7	4	40
3	Витік конфіденційної інформації	3	7	2	2	21
4	Надмірна залежність від сторонніх бібліотек та сервісів	5	4	5	3	20
5	Критичні баги	5	7	6	5	35
6	Нестача кваліфікованих спеціалістів	2	8	7	2	16
7	Неефективне управління проєктом	2	9	7	3	18
8	Недостатній досвід команди	3	7	7	4	21
9	Порушення правил кібербезпеки	5	7	3	3	35
10	Конфлікти всередині команди	3	8	9	2	24
11	Зміна конкурентного середовища	4	7	8	3	28
12	Економічний спад	3	8	3	2	24
13	Політичні та регуляторні зміни	1	3	5	1	3
14	Стихійні лиха	3	5	8	1	15
15	Воєнні дії	9	6	8	6	54
16	Недосконалість бізнес-моделі	2	9	5	2	18
17	Нестача фінансування	2	9	9	1	18
18	Недостатній попит	4	6	5	3	24
19	Нечітке позиціонування на ринку	2	4	6	2	8

Після оцінки ризиків варто виявити найбільш пріоритетні та критичні за силою впливу та розробити протиризикові заходи. Вони розробляються для ризиків, які не можна повністю усунути, але які є дуже ймовірними або мають суттєвий фінансовий вплив. Мета протиризикових заходів полягає в тому, щоб

знизити ймовірність виникнення ризику (ПРЗ профілактики), мінімізувати його наслідки (ПРЗ при симптомі) або ж розглянути сценарії реагування при настанні ризикової події (ПРЗ при проблемі) [9]. Усі протиризикові заходи для ризикових подій з силою впливу більше 25 розглянуті в табл. 4.8.

Таблиця 4.8

Протиризикові заходи

№	Ризикова подія	ПРЗ 1 (профілактика)	Симптом (рання ознака)	ПРЗ 2 (при симптомі)	ПРЗ 3 (при проблемі)
1	Воєнні дії	-	Воєнні дії уже тривають	-	Дистанційний формат роботи, децентралізація ресурсів
2	Складність реалізації та оптимізації алгоритмів	Збільшення часу на проведення досліджень, ретельне тестування на більшій кількості даних	Низька продуктивність алгоритму, незадовільні результати виконання	Перегляд стратегії реалізації алгоритмів, удосконалення процесів оцінки та ревію дизайну алгоритмів	Пошук та найм додаткового Senior спеціаліста по розробці алгоритмів, використання сторонніх бібліотек
3	Критичні баги	Впровадження покращених стандартів тестування, використання авт. тестування	Збільшення кількості багів за місяць, систематичні збої в роботі програми	Збільшення обсягу тестування, проведення рефакторингу та інших змін у кодї	Виділення одного спринту на ідентифікацію усіх проблем та їх виправлення
4	Порушення правил кібербезпеки	Обов'язкові курси по правилах кібербезпеки для команди, використання прав доступу, файрволів, створення резервних копій даних, шифрування	Виявлення підозрілої поведінки, вірусу, поява дивних файлів та програм, несанкціонований доступ	Повна перевірка системи на вразливості, зміна паролів і доступів, ізоляція заражених частин	Блокування доступів, відновлення даних, залучення спеціалістів з кібербезпеки, проведення внутрішнього розслідування та аудиту

Продовження табл. 4.8.

5	Зміна конкурентного середовища	Оптимізація процесів розробки, представлення MVP продукту, моніторинг ринку	Зміни у стратегіях конкурентів, зміна ринкової динаміки, нові конкуренти	Поглиблений аналіз стратегій та дій конкурентів для виявлення загроз та можливостей	Перегляд та оновлення стратегії, впровадження нових заходів, вдосконалення продукту
---	--------------------------------	---	--	---	---

Всього було виділено п'ять найбільш пріоритетних ризикових подій та розроблено протиризикові заходи для них. Саме вони можуть мати найбільший вплив на виконання та успішність проєкту. Загалом, протиризикові заходи є важливою частиною стратегії управління ризиками, оскільки вони спрямовані на запобігання або зменшення впливу можливих негативних подій на проєкт та дозволяють суттєво знизити загрозу фінансових чи інших втрат.

ВИСНОВКИ

Дана робота присвячена дослідженню процесів управління та розробці програмного забезпечення для цифрової обробки зображень. Під час дослідження було проведено ретельну оцінку та аналіз предметної області, виконано планування та наведено опис програмного забезпечення. Дослідження успішно виконано відповідно до поставлених завдань, що були визначені в початкових етапах роботи. Було досягнуто наступних результатів.

– В результаті виконання аналізу предметної області та науково-інформаційних джерел виявлено, що методи цифрової обробки зображень успішно застосовуються у багатьох галузях та сферах, а важливим питанням є створення нових та вдосконалення існуючих методів. Також додатково підкреслюється наявність проблематики предметної області і потенційні шляхи її вирішення. На основі аналізу існуючих рішень можна зробити висновок про те, що методи цифрової обробки зображень можуть досягати хороших результатів, проте в успіху проєкту окрім технічної складової велику роль також відіграє управління проєктами, зокрема ретельне планування, оцінка ризиків, контроль та моніторинг.

– Виконано огляд ринку програмного забезпечення для цифрової обробки зображень та визначено основні перспективи його розвитку, складові та напрямки розвитку, зокрема прогнозується середньорічний темп зростання ринку на 19,8% в період з 2024 по 2033 рік. Також виявлено, що велика кількість організацій віддають перевагу аутсорсингу цих завдань, оскільки це є більш економічно ефективним підходом, який усуває необхідність власної розробки та підтримки рішень для обробки зображень.

– Описано інноваційність проєкту, реалізація продукту якого є складним технічним та математичним завданням, що потребує проведення фундаментальних досліджень нових принципів, методів, теорій та моделей для обробки зображень. Зі свого боку проєкт пропонує новий підхід до створення ПЗ

у вигляді технічного інструменту, що дозволяє налаштувати його під потреби клієнта. Наукова ж новизна полягає у всебічному дослідженні та аналізі складових проєкту, процесів управління, побудові необхідних діаграм та моделей для успішної реалізації проєкту.

- Розкрито основну ідею проєкту, розглянуто дерево проблем та наслідків, визначено мету та цілі, які було перевірено за допомогою методу SMART. Додатково створено логіко-структурну схему та виявлено, що всі цілі відповідають загальній меті проєкту.

- Проведено аналіз зацікавлених сторін та оточення проєкту, зокрема виконано PEST-аналіз, який вказує на те що за ретельного планування та врахування факторів, що мають негативний вплив, проєкт має великий потенціал і можливості для досягнення цілей та впровадження на ринку. А найвищий рівень позитивного впливу на проєкт мають внутрішні зацікавлені сторони – виконуюча організація та команда проєкту.

- Проведено аналіз проєктних альтернатив, виявлено, що найбільш оптимальним варіантом є розробка системи, що використовує веб технології.

- Розглянуто основні характеристики проєкту, бізнес-модель, що містить поділ на B2B та B2C сегменти.

- Побудовано концептуальну модель складу системи для аналізу вхідних та вихідних даних, внутрішньої структури та взаємозв'язків. Виконано формалізацію математичних моделей системи та розглянуто питання прийняття управлінських рішень, що базується на RBM підході.

- Проведено аналіз технологій необхідних для розробки продукту, обґрунтовано доцільність їхнього використання. Головними серед них є продукти компанії Microsoft, а саме Azure, мова програмування C#. Для реалізації графічного інтерфейсу користувача використовується Angular.

- Описано основні архітектурні рішення, а саме використання мікросервісного підходу для розробки системи в цілому та архітектурного шаблону MVC – для розробки веб-клієнта.

– Спроектовано базу даних, побудовано три типи діаграм та виявлено, що найкращим рішенням для реалізації є застосування реляційних баз даних. Всього спроектована база даних містить 11 таблиць.

– Описано декілька базових прикладів алгоритмів та підходів, що реалізуються в ПЗ та розглянуто інтерфейс користувача. Розроблено діаграму навігації між сторінками системи, створено та описано каркасний інтерфейс окремих сторінок та показано макет дизайну сторінки обробки зображень.

– Створено структуру декомпозиції робіт проєкту по продуктах та по фазах життєвого циклу.

– Розглянуто організаційну структуру та сформовано команду проєкту відповідно до поставлених цілей, обсягу та структури робіт. Проєктний менеджер є відповідальним за успішне завершення проєкту в цілому, а архітектор і тех лід відповідають за успішне завершення його окремих технічних частин. Додатково створено матрицю відповідальності виду RACI, що дозволила чітко визначити ролі та обов'язки учасників проєкту.

– Описано переваги застосування гнучкого підходу до розробки ПЗ проєкту, а саме методології Scrum.

– Розглянуто життєвий цикл проєкту та виконано календарно-ресурсне планування проєкту. Початок проєкту 5 лютого 2024 року, завершення – 25 листопада 2024 року. Бюджет проєкту становить \$ 530 000.

– Розглянуто та описано управління якістю та ризиками в проєкті, розроблено протиризові заходи для головних ризиків, серед яких воєнні дії, складність реалізації, зміна конкурентного середовища, критичні баги та порушення правил кібербезпеки.

Загалом, оцінюючи виконану роботу, можна зробити висновок, що дослідження процесів управління та продукту проєкту підтверджує те, що поставлені завдання по розробці є реалістичними. Проєкт має значний потенціал та підкреслює значимість інноваційних рішень у галузі цифрової обробки зображень, а наведені результати будуть використані в подальших дослідженнях та майбутній розробці проєкту.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Бази даних та знань в управлінні проєктами [Текст]: Методичні вказівки для практичних та лабораторних робіт (ОП «Управління проєктами»). Кафедра технологій управління факультету інформаційних технологій Київського національного університету імені Тараса Шевченка/ А.О. Хлевний, О.В. Єгорченков– К. :ВИДАВНИЦТВО, 2020. – 54 с.
2. Гнучкі технології управління ІТ проєктами [Текст]: методичні вказівки до виконання практичних, лабораторних робіт та самостійної роботи для студентів освітньої програми «Управління проєктами» для денної форми навчання / Тімінський О.Г., Коломієць А.С., Латишева Т.В.– К.: КНУ імені Тараса Шевченка, 2022. –93 с.
3. Математичне моделювання в ІТ проєктах: методичні вказівки для виконання практичних, лабораторних та самостійних робіт з навчальної дисципліни / Морозов В.В. – К. : КНУ імені Тараса Шевченка, 2022. – 63 с.
4. Математичне моделювання систем і процесів: навч. посіб. – К. : НАУ, 2017. – 392 с.
5. Математичні методи розробки концепцій ІТ проєктів: Методичні вказівки для виконання практичних, лабораторних та самостійних робіт з навчальної дисципліни /Морозов В.В. – К. : КНУ імені Тараса Шевченка, 2022. – 79 с.
6. Методи прийняття управлінських рішень: методичні вказівки до виконання практичних, лабораторних та самостійних робіт з навчальної дисципліни / Морозов В.В. – К.: КНУ імені Тараса Шевченка, 2022. – 39 с.
7. Мінсінь Х. Технології аналізу цифрових зображень для систем підтримки прийняття рішень в сільському господарстві [Текст] / Х. Мінсінь, В. Вацкель // Управління розвитком складних систем. – № 37. – 2019. – С. 164– 167, <https://dx.doi.org/10.6084/m9.figshare.9783227>.

8. Р. Н. Кветний., І. В. Богач, О. Р. Бойко, О. Ю. Софіна, О.М. Шушура. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 2:– Вінниця: ВНТУ, 2013.–235с.
9. Управління ризиками та можливостями проекту [Текст]: методичні вказівки до виконання практичних, лабораторних робіт та самостійної роботи для студентів освітньої програми «Управління проектами» спеціальності 122 «Комп'ютерні науки» для денної і заочної форм навчання / Тімінський О.Г., Коломієць А.С. – К. : КНУ імені Тараса Шевченка, 2021. – 40 с.
10. Гасімов О. Мікросервісна архітектура для початківців. Частина I. GlobalLogic Ukraine. URL: <https://www.globallogic.com/ua/insights/blogs/microservices-architecture-for-beginners-part-one/>.
11. Дерево проблем та рішень. hromada.canactions. URL: <https://hromada.canactions.com/derevo-problem-rishen/>.
12. Діаграма Ганта. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Діаграма_Ганта.
13. Загальний регламент про захист даних (GDPR). GDPR-Text. URL: <https://gdpr-text.com/uk/>.
14. Застосування методики SMART для досягнення цілей. De Visu. URL: <https://devisu.ua/uk/stattia/zastosuvannya-metodiki-smart-dlya-dosyagnennya-ciley->.
15. Змерзлий І. Мікросервісна архітектура. Medium. URL: <https://medium.com/@IvanZmerzlyi/microservices-architecture-461687045b3d>.
16. Матриця відповідальності. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Матриця_відповідальності.
17. Основні форми організаційної структури проекту. Система електронного забезпечення навчання ЗНУ. URL: https://moodle.znu.edu.ua/pluginfile.php?file=/777715/mod_resource/content/0/ТЕМ_А%206_ОСНОВНІ%20ФОРМИ%20ОРГАНІЗАЦІЙНОЇ%20СТРУКТУРИ%20ПРОЕКТУ.pdf.
18. Програмне забезпечення як послуга. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Програмне_забезпечення_як_послуга.

19. Скрам. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Скрам>.
20. Фриміум. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Фриміум>.
21. Чорний ящик. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Чорний_ящик.
22. Шмигаль Д. У Дія.City вже понад 800 резидентів, ІТ-галузь залишається однією з рушійних сил української економіки. Кабінет Міністрів України. URL: <https://www.kmu.gov.ua/news/denys-shmyhal-u-diiacity-vzhe-ponad-800-rezydentiv-it-haluz-zalyshaietsia-odniieiu-z-rushiinykh-syl-ukrainskoi-ekonomiky>
23. Що таке PEST-аналіз. Дія.Бізнес. URL: <https://business.diia.gov.ua/handbook/marketing/so-take-pest-analiz>.
24. Що таке SaaS? Oracle | Cloud Applications and Cloud Platform. URL: <https://www.oracle.com/ua/applications/what-is-saas/>.
25. SWOT-аналіз – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/SWOT-аналіз>.
26. Aly A. A., Bin Deris S., Zaki N. Research Review for Digital Image Segmentation Techniques. International Journal of Computer Science and Information Technology. 2011. Т. 3, № 5. С. 99–106. URL: <https://doi.org/10.5121/ijcsit.2011.3509>.
27. Automatic coronary artery segmentation algorithm based on deep learning and digital image processing / F. Tian та ін. Applied Intelligence. 2021. URL: <https://doi.org/10.1007/s10489-021-02197-6>.
28. Image Processing Techniques for Analysis of Satellite Images for Historical Maps Classification—An Overview / A. Asokan та ін. Applied Sciences. 2020. Т. 10, № 12. С. 4207. URL: <https://doi.org/10.3390/app10124207>.
29. Institute P. M. Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Seventh Edition and the Standard for Project Management (ENGLISH). Project Management Institute, 2021.
30. The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis / M. Salvi

та ін. Computers in Biology and Medicine. 2021. Т. 128. С. 104129. URL: <https://doi.org/10.1016/j.compbiomed.2020.104129>.

31. Azure Machine Learning. Microsoft. URL: <https://azure.microsoft.com/en-us/products/machine-learning>.

32. Azure SQL Database. Microsoft. URL: <https://azure.microsoft.com/en-us/products/azure-sql/database>.

33. Bell D. J. Dice similarity coefficient. Radiopaedia. URL: <https://radiopaedia.org/articles/dice-similarity-coefficient>.

34. Cloud Computing Services. Microsoft Azure. URL: <https://azure.microsoft.com/en-us>.

35. Common web application architectures. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>.

36. Explicit dependencies. Architectural principles. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles#explicit-dependencies>.

37. Global Digital Image Processing Market, Trends and Forecast 2024-2033. market.us. URL: <https://market.us/report/digital-image-processing-market>.

38. Introduction to Angular concepts. Angular. URL: <https://angular.io/guide/architecture>.

39. ISO/IEC 27001:2022. ISO. URL: <https://www.iso.org/standard/27001>.

40. Microsoft Project. Microsoft. URL: <https://www.microsoft.com/uk-ua/microsoft-365/project/project-management-software>.

41. Overview of ASP.NET Core. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>.

42. Overview of ASP.NET Core MVC. Microsoft Learn. URL: <https://learn.microsoft.com/uk-ua/aspnet/core/mvc/overview?view=aspnetcore-8.0>.

43. Overview of Entity Framework Core. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/ef/core/>.

44. Smith S. Architecting Modern Web Applications with ASP.NET Core and Microsoft Azure. Microsoft. URL: <https://dotnet.microsoft.com/en-us/download/e-book/aspnet/pdf>.

45. Visual Studio documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/visualstudio/windows>.

ДОДАТКИ

Додаток А

Логіко-структурна схема

Таблиця А.1

Логіко-структурна схема

Опис	Показники досягнення	Індикатори	Припущення та ризики
<i>Загальна ціль</i> Успішно реалізувати проєкт	Проєкт виконано у визначені терміни з визначеним вмістом та відповідно до запланованої вартості	Інтеграція програмного забезпечення у більш ніж 5 систем клієнтів та залучення більше 10000 активних зовнішніх користувачів	
<i>Конкретні цілі</i> 1. Створити програмний продукт. 2. Спроекувати архітектуру модулів, API та бази даних. 3. Розробити базові функції та методи на основі ШІ.	1. Завершення етапу розробки і тестування. 2. Наявна чітка і документована архітектура усіх базових компонентів ПЗ. 3. Методи та функції працюють відповідно до вимог. Результат є чітким, очікуваним та зрозумілим.	1. Успішні модульні та інтеграційні тести, завершено реліз тестування. 2. ПЗ готове до розширення функціоналу, існує план інтеграції. 3. Розроблено мінімум 5 категорій та не менше ніж 50 базових функцій, а також мінімум 10 ШІ методів.	Досягнення базових конкретних цілей дозволить випустити перший реліз продукту та залучити перших клієнтів і користувачів. Основним ризиками є збільшення термінів та відповідно вартості, невірне проєктування архітектури.
<i>Результати</i> 1. Реалізовано адаптивність функцій програмного продукту. 2. Існує функцію зони інтересу.	1. Для кожної функції існує перелік параметрів та конфігурацій. 2. Реалізовано зону інтересу як окрему функцію. 3. Реалізовано 6 основних форматів зображень: JPEG, PNG,	1. Параметри та конфігурації впливають на виконання і можуть бути змінені. 2. Зона інтересу може бути застосована на будь-якому зображенні і може бути будь-якого розміру.	Дані результати є ґрунтовними для подальшого успіху і розвитку програмного продукту. Можливі труднощі у реалізації та оцінці методів на основі штучного інтелекту.

<p>3. Підтримка усіх популярних форматів зображень.</p> <p>4. Підтримка функціоналу графічних шарів.</p> <p>5. Програмний продукт відповідає міжнародним стандартам захисту даних.</p> <p>6. Можливість кольоризації зображень за допомогою програмного продукту.</p> <p>7. Реалізовано прикладний програмний інтерфейс.</p>	<p>BMP, TIFF, RAW, HEIC.</p> <p>4. Кожне оброблене зображення може складатись з переліку шарів, що містять інформацію для гнучкості робочого процесу.</p> <p>5. Програмний продукт успішно пройшов аудит та перевірку з відповідності міжнародним стандартам захисту даних.</p> <p>6. Функція кольоризації чорно-білих зображень доступна для користувачів.</p> <p>7. Результат виконання методів та функцій може бути отриманий через API.</p>	<p>3. Будь-яке зображення у форматі JPEG, PNG, BMP, TIFF, RAW, HEIC може бути відкрите у програмі.</p> <p>4. Збереження послідовності застосованих методів зі значеннями параметрів.</p> <p>5. Отримано сертифікат відповідності міжнародним стандартам захисту даних, таким як ISO/IEC 27001, GDPR.</p> <p>6. Досягнуто індексу кольоропередачі більше 70 для .</p> <p>7. Успішна інтеграція з компаніями-клієнтами та користувачами.</p>	<p>Ризиками є наявність програмних помилок та некоректних результатів, що матимуть вплив на задоволеність користувачів.</p> <p>Для отримання сертифікатів від незалежних аудиторів повинна бути проведена ретельна робота над підтримкою усіх стандартів.</p> <p>Ризиком для успішної інтеграції та залучення клієнтів є недостатньо ефективна маркетингова кампанія.</p>
<p><i>Дії</i></p> <p>1. Розробка вимог до продукту, планування та дослідження</p> <p>2. Формування проєктної команди</p> <p>3. Отримання даних та обладнання</p> <p>4. Розробка, тестування, впровадження</p> <p>5. Маркетингова та рекламна кампанія</p>	<p><i>Засоби:</i></p> <p>1. Процеси управління проєктом.</p> <p>2. Визначена організаційна структура.</p> <p>3. Апаратне та програмне забезпечення.</p> <p>4. Інструменти розробки, репозиторії, інструменти керування проєктами, інструменти тестування, впровадження.</p> <p>5. Цільова реклама, маркетинг, аналітика та відстеження результатів.</p>	<p><i>Витрати:</i></p> <p>Щомісячна заробітна плата працівникам, купівля фізичного обладнання, витрати на хмарні сервіси та ліцензії, зовнішні послуги, маркетинг та реклама.</p>	<p>1. Чіткі вимоги та планування – більша ймовірність успіху.</p> <p>2. Професійна команда – якісне і вчасне виконання.</p> <p>3. Якісні дані і обладнання мають прямий вплив на продукт і стабільність.</p> <p>4. Виконання усіх етапів відповідно до обмежень збільшує шанси на успіх проєкту.</p> <p>5. Кількість клієнтів та їх залученість впливає на успішність проєкту.</p>

Аналіз зацікавлених сторін

Таблиця Б.1

Потреби та вигоди зацікавлених сторін

№	Зацікавлена сторона	Потреби	Вигоди
1	Замовник проекту	Реалізації ідеї, отримання готового продукту в повному об'ємі та з очікуваною якістю, розширення ринку	Покращення якості надання послуг, підвищення прибутковості та конкурентоспроможності
2	Інвестори	Фінансовий успіх проекту, підтримка інноваційних проєктів та технологій	Повернення вкладень і отримання прибутку
3	Команда проекту	Успішність проекту, професійний розвиток, розвиток бізнесу	Заробітна плата та інші переваги, що надає виконуюча організація, мотивація та задоволення від роботи
4	Проектний менеджер	Досягнення цілей проекту, дотриманням усіх його обмежень (зміст, терміни, якість, вартість, ресурси, ризики), задоволення потреб та очікувань учасників проекту	Власний професійний ріст та фінансова винагорода
5	Виконуюча організація та її керівник	Отримання і накопичення досвіду з реалізації проєктів, підтримка та зміцнення власного становища на ринку, можливості для подальшого партнерства	Отримання прибутку, позитивний вплив на репутацію, залучення нових клієнтів
6	Користувачі	Потужний кросплатформний інструмент для виконання операцій над цифровими зображеннями	Новий зручний інструмент для цифрової обробки зображень, покращення якості, збереження часу, зручний доступ до даних
7	Конкуренти	Збереження свого становища на ринку, розробка власного рішення, перешкоджання у реалізації	-
8	Компанії-партнери	Покращення власного продукту за рахунок інтеграції, участь в інноваційному проєкті, розширення свого ринку, автоматизація операцій, покращення ефективності.	Підвищення рівня конкурентоспроможності на ринку завдяки використанню нових технологій та розробок, збільшення прибутку, покращення репутації

Продовження табл. Б.1

9	Регуляторні органи та аудитори	Інформація про продукт, забезпечення дотримання законодавства та безпеки, прозорість та відповідальність, можливість проводити перевірки	Дотримання законодавства, захист прав споживачів, інновації у галузі, покращення безпеки та якості продукту
10	Сторонні постачальники даних та послуг	Збільшення попиту на їхні послуги та продукти, підвищення рівня конкурентоспроможності на ринку	Отримання прибутку за рахунок постачання необхідних даних та послуг

Реалізація окремих алгоритмів на мові програмування С#

```

public static Task<Bitmap> MedianBlur(this Bitmap bitmap, int kSize)
{
    return Task.Run(() =>
    {
        var originalImage = bitmap.ToImage<Bgr, byte>();
        var newImage = new Image<Bgr, byte>(originalImage.Width, originalImage.Height,
new Bgr(255, 255, 255));
        CvInvoke.MedianBlur(originalImage, newImage, kSize);
        return newImage.ToBitmap();
    });
}

public static Task<Bitmap> BilateralBlur(this Bitmap originalBitmap, int diameter,
double sigmaColor, double sigmaSpace)
{
    return Task.Run(() =>
    {
        var originalImage = originalBitmap.ToImage<Bgr, byte>();
        var resultImage = new Image<Bgr, byte>(originalImage.Size);
        CvInvoke.BilateralFilter(originalImage, resultImage, diameter, sigmaColor,
sigmaSpace, BorderType.Default);
        return resultImage.ToBitmap();
    });
}

public static Task<Bitmap> ApplyHistogramEqualizationAsync(Bitmap bitmap)
{
    return Task.Run(() =>
    {
        var grayImage = bitmap.ToImage<Gray, byte>();
        var histEq = new Mat();
        CvInvoke.EqualizeHist(grayImage, histEq);
        return histEq.ToBitmap();
    });
}

public static Task<Bitmap> ApplyBrightnessCorrectionAsync(Bitmap bitmap, float
brightness)
{
    return Task.Run(() =>
    {
        var image = bitmap.ToImage<Bgr, byte>();
        var result = new Image<Bgr, byte>(image.Width, image.Height);
        CvInvoke.ConvertScaleAbs(image, result, 1, brightness);
        return result.ToBitmap();
    });
}

public static Task<Bitmap> Erosion(this Bitmap bitmap, int iterationsCount)
{

```

```

        return Task.Run(() =>
        {
            var originalImage = bitmap.ToImage<Bgr, byte>();
            return originalImage.Erode(iterationsCount).ToBitmap();
        });
    }

    public static Task<Bitmap> Dilation(this Bitmap bitmap, int iterationsCount)
    {
        return Task.Run(() =>
        {
            var originalImage = bitmap.ToImage<Bgr, byte>();
            return originalImage.Dilate(iterationsCount).ToBitmap();
        });
    }

    public static Task<Bitmap> Open(this Bitmap bitmap)
    {
        return Task.Run(() =>
        {
            return PerformMorphologyOperation(bitmap, MorphOp.Open);
        });
    }

    public static Task<Bitmap> Close(this Bitmap bitmap)
    {
        return Task.Run(() =>
        {
            return PerformMorphologyOperation(bitmap, MorphOp.Close);
        });
    }

    public static Task<Bitmap> Gradient(this Bitmap bitmap)
    {
        return Task.Run(() =>
        {
            return PerformMorphologyOperation(bitmap, MorphOp.Gradient);
        });
    }

    private static Bitmap PerformMorphologyOperation(Bitmap bitmap, MorphOp morphOperation)
    {
        var originalImage = bitmap.ToImage<Bgr, byte>();
        var kernel = CvInvoke.GetStructuringElement(ElementShape.Rectangle, new Size(5, 5),
            new Point(-1, -1));
        var resultImage = originalImage.MorphologyEx(morphOperation, kernel, new Point(-1,
            -1), 1, BorderType.Default, new MCvScalar(1));
        return resultImage.ToBitmap();
    }
}

```

Управління якістю проєкту

Таблиця Г.1

Контроль якості проєкту

Зацікавлені сторони	Модуль продукту	Вимоги до якості продукту	Заходи необхідні для задоволення вимог
Команда проєкту та керівник проєкту Компанії-партнери Користувачі Сторонні постачальники даних та послуг	Прикладний програмний інтерфейс	Наявність документації та чіткого версіонування.	Впровадження підтримки семантичного версіонування (SemVer), докладне документування усіх функцій та можливостей.
		Забезпечення безпеки, захист від несанкціонованого доступу.	Використання автентифікації та авторизації, наявність різних прав доступу та шифрування.
		Масштабованість, здатність обробляти велику кількість запитів, стабільність роботи і підтримки.	Застосування хмарних сервісів, оптимізація запитів, використання кешування, виконання резервного копіювання та відновлення.
		Ефективність у використанні ресурсів, пам'яті, процесору, швидкий доступ до даних.	Асинхронний підхід, паралельність виконання, оптимізація алгоритмів, індексація бази даних.
		Визначення базових інтерфейсів та шляхів інтеграції.	Використання відкритих стандартів, стандартизація інтерфейсів та їх документування.
Користувачі Компанії-партнери Команда проєкту та керівник проєкту Сторонні постачальники даних та послуг	Підсистема обробки цифрових даних	Ефективна швидкість обробки цифрових даних, відповідно до вибраних методів та параметрів.	Оптимізація алгоритмів, паралельна обробка, кешування даних, використання спеціалізованих бібліотек, моніторинг та оптимізація продуктивності.
		Можливість вибору методу, зображення, визначення параметрів.	Параметризація методів обробки, інтерактивність інтерфейсу, кешування проміжних результатів.

Регуляторні органи		Збереження проміжних результатів та етапів виконання операцій.	Кешування проміжних результатів, ведення журналу виконаних дій
		Детермінований результат у випадку використання не ШІ-алгоритмів.	Стандартизація методів, тестування та валідація, документування результатів.
		Перелік визначених обмежень для ШІ-алгоритмів.	Створення та документування обмежень, визначення поведінки.
		Відмовостійкість та надійність підсистеми під час виконання операцій.	Впровадження системи обробки виключень, моніторинг стану системи, планування резервних дій.
		Відкритий інтерфейс виклику та створення методів.	Документування API, забезпечення доступу до методів та прямого виклику.
Користувачі Компанії-партнери Команда проєкту та керівник проєкту	Підсистема ведення і завантаження даних	Можливість додавання зображень різних форматів (JPEG, PNG, BMP, TIFF, RAW, HEIC).	Підтримка наведених форматів, можливість конвертації зображень.
		Зчитування метаданих зображень.	Застосування стандартів зберігання метаданих, підтримка метаданих та їх зчитування.
		Збереження усіх даних в базі даних.	Проєктування та розробка бази даних, реалізація базових операцій.
		Надійність роботи та відсутність збоїв при роботі із зображеннями різного розміру та формату.	Валідація зображень, масштабування, обробка помилок.
		Можливість завантаження цілих груп зображень одночасно та паралельно.	Реалізація вибору декількох зображень, розподілена обробка, оптимізація мережевої взаємодії, паралельна обробка.
Замовник Користувачі Компанії-партнери Виконуюча організація	Підсистема виводу даних	Відображення збережених та оброблених даних.	Наявність користувацького інтерфейсу, таблиця зображень, масштабування, можливість збереження у визначених форматах.

Продовження табл. Г.1

Команда проекту та керівник проекту		Візуалізація графіків статистики і даних моніторингу, їх форматування.	Створення алгоритмів візуалізації даних, адаптивний дизайн графіків, реалізація налаштувань.
		Збереження та експорт даних у різних форматах.	Реалізація вибору формату, налаштування експорту, реалізація інтерфейсу.
Користувачі Компанії-партнери Виконуюча організація Команда проекту та керівник проекту Регуляторні органи Сторонні постачальники даних та послуг	Модуль керування користувачами	Наявність якісної та однозначної документації по API та інших складових продукту.	Стандартизація процесу документування, розробка вимог до документування.
		Можливості масштабування, налаштування та інтеграції.	Використання конфігураційних файлів, хмарних сервісів та визначення зрозумілих інтерфейсів даних.
		Сумісність з технологічною платформою партнерів.	Співпраця та взаємодія з партнерами, тестування сумісності.
		Відповідність стандартам галузі, забезпечення безпеки даних та конфіденційності.	Використання автентифікації та авторизації, наявність різних прав доступу та шифрування.
		Відповідність функціональним вимогам.	Аналіз та контроль виконання вимог, відстеження змін та верифікація.
		Технічна підтримка та взаємодія з розробниками ПЗ.	Надання доступу до технічної підтримки та документації, створення форуму та навчального курсу.
Замовник Користувачі Компанії-партнери Виконуюча організація Регуляторні органи	Графічний інтерфейс	Підтримка різних мов та регіональних налаштувань.	Локалізація інтерфейсу та контенту, підтримка різних культур.
		Єдиний стиль дизайну та графічних елементів.	Створення єдиного набору стилів для різних елементів, використання шаблонів.
		Реалізація елементів пошуку, сортування, фільтрування.	Доступність цих функцій у інтерфейсі програми.

Команда проекту та керівник проекту		Естетична привабливість та сучасний стиль.	Використання сучасних та актуальних стилів, мінімалістичного, неперевантаженого дизайну.
		Легкість використання та навігації між сторінками.	Реалізація коротких та зрозумілих маршрутів, зменшення кількості перемикачів між сторінками.
		Доступність інтерфейсу для усіх категорій користувачів.	Можливість зміни розміру елементів інтерфейсу, доступність альтернативних кольорових гам, реалізація голосових функцій.
		Швидке реагування на дії користувача, відсутність зависань.	Використання асинхронних запитів, прогрес барів у інтерфейсі.
		Можливість налаштування інтерфейсу, зміни теми та переміщення функціональних елементів.	Створення налаштувань інтерфейсу, надання можливості їх редагування.
		Адаптивність інтерфейсу на різних пристроях.	Розробка адаптивного дизайну, тестування на різних пристроях.
Замовник Регуляторні органи Користувачі Компанії-партнери Виконуюча організація Команда проекту та керівник проекту Сторонні постачальники даних та послуг	Підсистема моніторингу	Анонімізація усіх даних відповідно до вимог та стандартів.	Зміна ідентифікаторів, видалення чутливої інформації, агрегація даних.
		Збір метрик та їх візуалізація.	Набір метрик, інструменти збору та візуалізації
		Виявлення помилок та нештатних ситуацій при роботі з системою.	Логування, автоматизоване тестування, моніторинг системи та ресурсів
		Ведення журналу виконаних дій, логування	Створення рівнів логування, анонімізація, захист доступу, запис важливої інформації про діяльність у системі
		Забезпечення безпеки усіх зібраних даних, їх шифрування.	Шифрування даних при зберіганні та передачі в мережі, обмеження доступів
		Аналіз використаних апаратних ресурсів та швидкодії.	Моніторинг апаратних ресурсів, профілювання ПЗ

Закінчення табл. Г.1

Замовник Користувачі Компанії- партнери	Модуль сповіщень	Підтримка різних каналів сповіщень.	Реалізація email, sms, push, web сповіщень, інтеграція з месенджерами
		Забезпечення гарантованої доставки необхідних повідомлень чи сповіщень.	Використання механізмів підтвердження доставки, обробка помилок
		Керування підпискою та налаштування сповіщень.	Реалізація налаштувань підписки
		Локалізація сповіщень та візуалізація різних типів.	Наявність локалізації та підтримка різних культур