Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Створення та корекція зображення за допомогою текстових підказок у діалозі
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-23-1
Кирило Левицький
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Ваган Терзіян
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 25 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Левицькому Кирилу Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Створення та корекція зображення за допомогою текстових підказок у діалозі

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 4 червня 20 25 р.

3. Вихідні дані до роботи дані Інтернет-джерел та відомих наукових проєктів, документація розробленої системи, проведене тестування системи, графічні матеріали, керівництво користувача

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі

2) Аналіз методів системи створення та корекції зображень

3) Розробка функціональних вимог до системи створення та корекції зображень

4) Опис прийнятих проєктних рішень під час розробки системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Аналіз завдання та предметної галузі	22.04.2025	виконано
3	Розробка вимог до системи штучного інтелекту	25.04.2025	виконано
4	Аналіз методів створення генеративних систем	27.04.2025	виконано
5	Проєктування програмного засобу	28.04.2025	виконано
6	Розробка програмного засобу	30.04.2025	виконано
7	Тестування розробленої системи	02.05.2025	виконано
8	Оформлення графічної частини	15.05.2025	виконано
9	Розробка «Керівництва користувача»	15.05.2025	виконано
10	Оформлення пояснювальної записки	19.05.2025	виконано
11	Представлення на рецензування	25.05.2025	виконано
12	Представлення кваліфікаційної роботи в ЕК	04.06.2025	виконано

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф.Ваган Терзіян
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 109 с., 22 рис., 4 дод., 26 джерел.

АНАЛІЗ ЛЮДСЬКОЇ МОВИ, ГЕНЕРАТИВНИЙ ШІ, ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ, ДИФУЗІЙНІ МОДЕЛІ, ЗОБРАЖЕННЯ, СТВОРЕННЯ І КОРЕКЦІЯ ЗОБРАЖЕНЬ, ТРАНСФОРМЕРИ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об’єкт дослідження – є розробка штучного інтелекту для створення і корекції зображень за допомогою текстових підказок в діалозі.

Предмет дослідження – аналіз потреб користувачів у створенні і редагуванні зображень, створення алгоритмів штучного інтелекту для генерації зображень та розробка функціоналу спілкування користувача з системою і отримання результатів.

Мета роботи – створення та впровадження системи, яка дозволить будь-якому користувачеві створювати нові зображення за текстовим запитом і, шляхом додаткових інструкцій отримати поліпшене зображення, яке буде задовольняти вимоги і потреби користувача.

Методи дослідження – аналіз предметної галузі для діяльності системи, аналіз методів для реалізації системи створення і корекції зображень. Розробка функціональних вимог і безпосередньо створення системи створення і корекції зображень за допомогою засобів програмування.

Розроблена система штучного інтелекту для створення і корекції зображень може бути використана великою кількістю користувачів, які шукають нові методи створення графічного матеріалу з мінімальними витратами на створення. Додатково ця система може слугувати джерелом натхнення для творчих людей, яким потрібні нові ідеї, і ця система може надихнути їх на щось особливе.

ABSTRACT

Master’s thesis contains: 109 pp., 22 fig., 4 ann., 26 references.

ARTIFICIAL INTELLIGENCE, DIFFUSION MODELS, GENERATIVE AI, GENERATIVE-COMPETITIVE NETWORKS, HUMAN LANGUAGE ANALYSIS, IMAGE GENERATION AND CORRECTION, IMAGES, TRANSFORMERS.

The object of the study is the development of artificial intelligence for creating and correcting images using text prompts in a dialog.

The subject of the research is the analysis of user needs in creating and editing images, the creation of artificial intelligence algorithms for generating images, and the development of functionality for user communication with the system and obtaining results.

The purpose of the work is to create and implement a system that will allow any user to create new images by text request and, through additional instructions, to obtain an improved image that will meet the user's requirements and needs.

Research methods – analysis of the subject area for the system, analysis of methods for implementing the system for creating and correcting images. Development of functional requirements and the creation of an image creation and correction system using programming tools.

The developed artificial intelligence system for creating and correcting images can be used by large number of users who are looking for new methods of creating graphic material with minimal creation costs. In addition, this system can serve as a source of inspiration for creative people who need new ideas, and this system can inspire them to do something special.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 9
Вступ 10
1 Аналіз предметної галузі 11
1.1 Аналіз предметної галузі для діяльності системи генерації та виправлення зображень за допомогою текстових підказок в діалозі 11
1.2 Аналіз існуючих аналогів системи генерації та виправлення зображень за допомогою текстових підказок в діалозі 13
1.2.1 Midjourney 14
1.2.2 Adobe photoshop 15
1.2.3 Runway AI 16
1.2.4 Microsoft designer 18
1.3 Визначення сфери застосування системи генерації зображень за допомогою засобів генеративного штучного інтелекту 19
1.4 Постановка задачі для системи генерації та виправлення зображень за допомогою текстових підказок в діалозі 20
2 Аналіз методів системи створення та корекції зображень за допомогою текстових підказок в діалозі 22
2.1 Аналіз дифузіних моделей для системи створення і корекції зображень в діалозі 22
2.2 Аналіз генеративних змагальних мереж для системи створення і корекції зображень в діалозі 24
2.3 Аналіз моделей трансформерів для системи створення і корекції зображень в діалозі 25
2.4 Аналіз моделей автокодерів для системи створення і корекції зображень в діалозі 28
3 Розробка функціональних вимог до системи створення та корекції зображень за допомогою текстових підказок в діалозі 30

3.1 Розробка системних вимог до системи створення і корекції зображень в діалозі 30
3.2 Визначення функціональних вимог до системи створення і корекції зображень 30
3.3 Розробка моделі потоків даних для системи створення і корекції зображень 41
3.4 Визначення вимог до інтерфейсу системи штучного інтелекту створення і корекції зображень 43
3.5 Розробка діаграми варіантів використання системи штучного інтелекту створення і корекції зображень 43
3.6 Розробка діаграми класів системи штучного інтелекту створення і корекції зображень 47
3.7 Розробка діаграми послідовності для прецеденту «створення зображення» 51
4 Опис прийнятих проєктних рішень під час розробки системи створення та корекції зображень за допомогою текстових підказок 54
4.1 Опис структури розробленої системи створення і корекції зображень в діалозі 54
4.2 Розробка алгоритму роботи системи створення і корекції зображень в діалозі 55
4.3 Реалізація програмної логіки для бізнес-процесу системи «створення зображення» 57
4.4 Реалізація програмної логіки для бізнес-процесу системи «корекція зображення» 61
4.5 Програмна реалізація логіки спілкування користувача з системою . 64
4.6 Тестування розробленого програмного забезпечення 68
4.7 Захист інформації, що циркулює в системі штучного інтелекту для створення і корекції зображень в діалозі 74
Висновки 75
Перелік джерел посилання 77

Додаток А Графічні матеріали..... 80
Додаток Б Керівництво користувача..... 92
Додаток В Програмний код..... 98
Додаток Г Відомість кваліфікаційної роботи..... 109

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Адмін – користувач із роллю «Адміністратор»;

ШІ – штучний інтелект;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – спосіб для взаємодії двох або більше додатків;

CPU – Central Processing Unit – центральний процесор;

DFD – Data Flow Diagram – діаграма потоків даних;

GANs – Generative Adversarial Networks – генеративно-змагальні мережі;

GPU – Graphics Processing Unit – графічний процесор;

IDEF0 – Icam DEFinition for Function Modeling – діаграма функціонального моделювання;

JPEG – Joint Photographic Experts Group – зображення з втратою якості;

NLP – Natural Language Processing – обробка природньої мови;

PNG – Portable Network Graphics – растровий формат зображення;

RGB – Red, Green, Blue – палітра кольорів;

UML – Unified Model Language – уніфікована мова моделювання;

Use-case – діаграма прецедентів.

ВСТУП

У сучасному світі з кожним днем створюється величезна кількість нових компаній, інтернет–магазинів, каналів і сторінок у соціальних мережах, і їх всім необхідно зацікавити потенційного споживача новим оригінальним контентом без порушення авторських прав і з мінімальними затратами на маркетинг. З цією метою було розроблено величезну кількість систем штучного інтелекту, що дозволяють своїм користувачам створювати різні зображення в стилі, якому вони забажають, і в розмірі зображення, яке вони вказують у запиті.

Мета роботи: проєктування та розробка елементів системи генерації та редагування зображень за допомогою текстових підказок в діалозі між користувачем і штучним інтелектом. Для цього необхідно провести аналіз предметної галузі, знайти і порівняти вже існуючі схожі аналогічні системи і виявити їх недоліки. Наступним кроком необхідно провести функціональне моделювання і визначити вимоги для системи, що розробляється, за допомогою нотації IDEF0. Після цього провести декомпозицію і опис функцій системи, змоделювати і описати діаграму послідовності дій, розробити функції системи та провести тестування розробленого штучного інтелекту, який генеруватиме зображення і матиме можливість редагувати їх, за допомогою текстових підказок.

Галузь застосування – надання послуг створення і редагування зображень фізичним і юридичним особам.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі для діяльності системи генерації та виправлення зображень за допомогою текстових підказок в діалозі

Функціонування штучного інтелекту в Україні регулюється низкою законодавчих та нормативних актів.

У Розпорядженні Кабінету Міністрів України від 02.12.2020 № 1556-р «Про схвалення Концепції розвитку штучного інтелекту в Україні» визначаються мета, принципи та завдання розвитку технологій штучного інтелекту в Україні, як одного з пріоритетних напрямів у сфері науково-технологічних досліджень [1].

Законом України від 01.12.2022 № 2811-IX «Про авторське право і суміжні права» регулюються відносини щодо набуття, здійснення та захисту особистих немайнових та майнових авторських та/або суміжних прав, а також прав особливого роду, пов’язаних зі сферою авторського права та/або суміжних прав [2].

Системи генерації та виправлення зображень за допомогою текстових підказок є частиною штучного інтелекту та комп’ютерного зору. Вони дозволяють користувачам створювати або редагувати зображення за допомогою простих текстових запитів [3].

Такі системи використовуються в різних галузях, зокрема:

– графічний дизайн – для швидкого створення ідей, концептів або ілюстрацій;

– реклама та маркетинг – для генерації візуального контенту без застосування роботи дизайнерів;

– розробка комп’ютерних ігор та анімація – для створення концепт-артів, текстур і середовищ;

– освіта – для наочних матеріалів та тренувальних інструментів;

– розваги – створення унікальних зображень для соціальних мереж, «мемів» або персоналізованого контенту.

Системи генерації зображень ґрунтуються на:

– глибоких нейромережах, зокрема, дифузійних моделях (наприклад, Stable Diffusion, DALL·E, Midjourney, Runway AI тощо);

– обробці природної мови для розуміння текстових запитів;

– генеративно-змагальних мережах (GANs) для створення реалістичних зображень.

Алгоритм роботи подібної системи виглядає наступним чином:

– користувач вводить текстовий опис бажаного зображення;

– алгоритм аналізує запит і створює зображення «з нуля» або змінює наявне;

– користувач може вносити правки через додаткові текстові інструкції [3].

Основними перевагами системи генерації зображень і їх редагування за допомогою текстових підказок в діалозі є:

– швидке створення якісного візуального контенту;

– користувач має можливість отримати унікальне зображення;

– доступність творчої діяльності для людей без навичок у графічному дизайні;

– велика можливість генерувати ідеї та ескізи для творчих людей;

– можливість отримати інструкцію, як можна реалізувати згенероване зображення без допомоги штучного інтелекту;

– гнучкість у редагуванні та адаптації зображень.

Через швидкий розвиток систем штучного інтелекту алгоритми і моделі генерування графічних елементів досі мають недоліки і незручності під час використання. Нижче наведено деякі недоліки таких систем:

– можливі неточності у виконанні складних запитів – системи генеративного штучного інтелекту досі можуть створити некоректне зображення, наприклад, коли людина має шість пальців на одній руці або мати дві праві ноги;

– необхідність оптимізації для отримання бажаного результату – для отримання бажаного результату може знадобитися велика кількість спроб, які не призведуть до якісного зображення, адже все одно будуть зберігатися незначні неточності [4];

– етичні і юридичні нюанси, пов’язані з авторським правом і використанням ШІ-генерованих зображень. Багато систем генеративного штучного інтелекту не дозволяють створювати персонажів або локації з творів, які захищені авторським правом.

Таким чином, системи генерації та редагування зображень за текстовими підказками відкривають нові можливості у сфері творчості, автоматизації та візуального контенту. Вони активно використовуються в бізнесі та розвагах і мають значний потенціал розвитку.

1.2 Аналіз існуючих аналогів системи генерації та виправлення зображень за допомогою текстових підказок в діалозі

Через стрімкий зріст популярності систем штучного інтелекту, який створює контент із нічого, наразі існує безліч подібних систем штучного інтелекту, що дозволяють користувачеві за допомогою текстового запиту створити зображення, відео, музику тощо. Багато з цих сервісів мають свої переваги і недоліки, деякі з них мають вищу багатозадачність, деякі – мають привабливу ціну підписки, деякі системи дозволяють створити зображення, відредагувати частину або змінити зображення повністю. А деякі системи дозволяють тільки створити зображення «з нуля» і неможливо відредагувати вже створене зображення. Також є системи, які дозволяють створити зображення за методом «текст в зображення» і «зображення в

зображення», але, щоб їх відредагувати, треба завантажити найкращу спробу, перейти в потрібний інструмент редагування, завантажити зображення в інструмент, і тільки тоді є можливість відредагувати зображення [5].

У рамках дипломного проєктування буде розглянуто основні популярні системи, що пропонують схожі функції і механіки для створення зображень і їх редагування, за допомогою текстових підказок в діалозі.

1.2.1 Midjourney

Midjourney – незалежна дослідницька лабораторія та назва її власної програми генеративного штучного інтелекту, яка створює цифрові зображення з описів природною мовою, званих «підказками» [6].

Ця система дозволяє створити зображення за допомогою запиту і має власний синтаксис створення зображення. Тобто, користувач здійснює власний запит, зазначає стиль бажаного зображення і також може вказати параметри розміру зображення. На рисунку 1.1 зображено запит і результат виконання запиту в системі генерації зображень Midjourney.

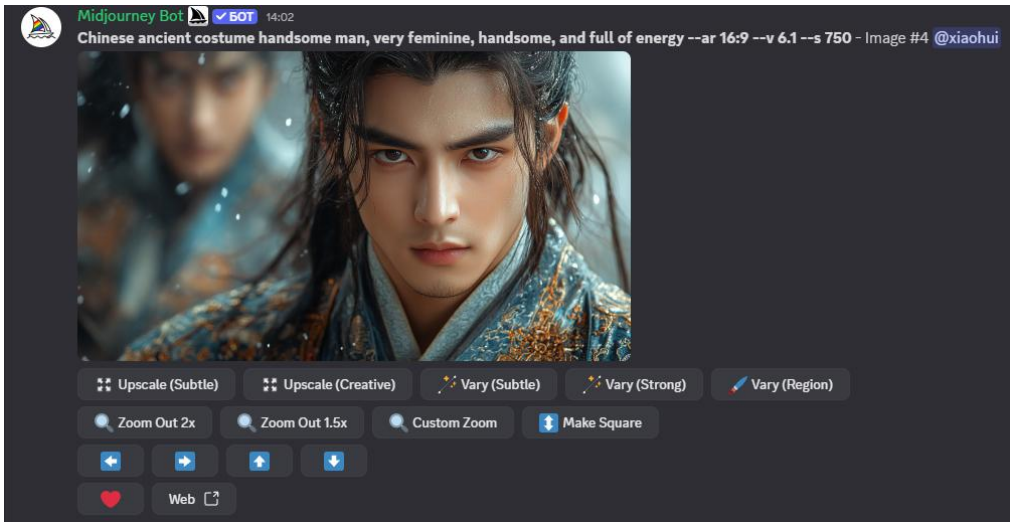

Рисунок 1.1 – Запит і результат виконання запиту в системі генерації зображень Midjourney

На рисунку 1.1 видно основний функціонал системи: користувач вводить запит і команди, згідно з якими зображення має бути формату 16:9, система генерує зображення і повертає його користувачеві, який, в свою чергу, може збільшити зображення, віддалити його, зробити квадратним і перейти в веб-версію, щоб завантажити зображення.

1.2.2 Adobe Photoshop

Adobe Photoshop – графічний редактор, розроблений і поширюваний фірмою Adobe Systems. Цей продукт є лідером ринку в галузі комерційних засобів редагування растрових зображень і найвідомішим продуктом фірми Adobe [7].

Adobe Photoshop Generative Fill – це інструмент на основі штучного інтелекту, який використовує технологію Adobe Firefly для створення та редагування зображень за текстовими запитами. Він дозволяє користувачам швидко додавати, видаляти або змінювати об’єкти на зображенні, вводячи прості текстові підказки [7].

Ця система є дуже популярною серед дизайнерів і людей, які професійно займаються творчістю. Вона не дозволяє створити зображення «з нуля», але має функції генеративного штучного інтелекту, що дозволяє відредагувати виділені об’єкти, що містяться на зображенні, за допомогою текстових підказок. На рисунку 1.2 зображено інтерфейс програми Adobe Photoshop для генеративного редагування зображень.

Generative Fill у Photoshop – це потужний інструмент для автоматичного редагування зображень, який значно прискорює роботу дизайнерів і фотографів. Він надає широкі можливості для творчості, дозволяючи легко додавати, змінювати та видаляти об'єкти за допомогою текстових запитів.

Рисунок 1.2 – Інтерфейс Adobe Photoshop для генеративного редагування зображень

1.2.3 Runway AI

Runway AI – це платформа для генерації та редагування відео та зображень на основі штучного інтелекту. Вона розроблена для креативних професіоналів, зокрема, дизайнерів, художників, режисерів та аніматорів [8].

Runway дозволяє створювати відео з текстових підказок або вхідного зображення. Модель Gen-2 використовує дифузійні технології та машинне навчання для побудови реалістичних або стилізованих відео. Runway також підтримує створення анімованих персонажів, генерацію мови та анімування статичних зображень. На рисунку 1.3 зображено інтерфейс Runway AI для генерації зображень.

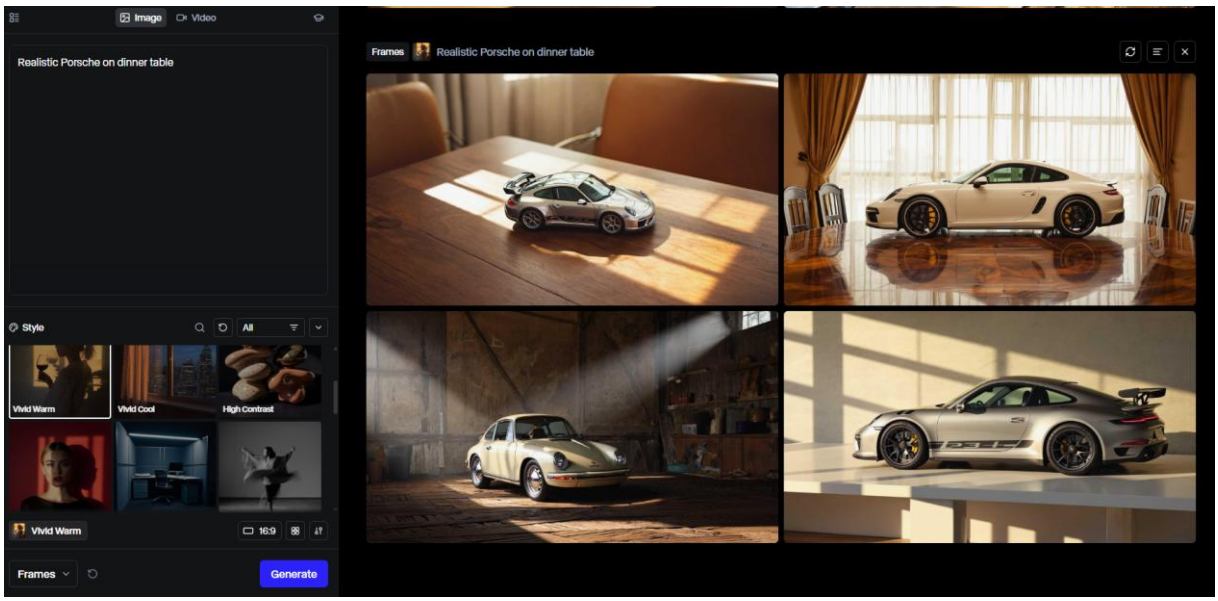

Рисунок 1.3 – Інтерфейс Runway AI для генерації зображень

Функціонал даного сервісу дозволяє обробити отримане зображення відповідно до нового запиту і обраного стилю. На рисунку 1.4 зображено інтерфейс редагування зображення за допомогою генеративного штучного інтелекту.

Рисунок 1.4 – Інтерфейс Runway AI редагування зображення

Runway AI – це зручний інструмент для тих, хто працює з візуальним контентом. Він дозволяє швидко створювати та редагувати зображення й

відео за допомогою штучного інтелекту, відкриваючи нові можливості для творчості та автоматизації.

1.2.4 Microsoft Designer

Microsoft Designer – це інструмент для створення графічного контенту на основі штучного інтелекту. Він поєднує можливості генеративного ШІ (зокрема, DALL·E від OpenAI) та зручні дизайнерські шаблони для швидкого створення візуальних матеріалів [9].

Користувач може вводити текстові підказки і система автоматично створюватиме відповідні зображення за допомогою DALL·E 3. Designer аналізує текстові підказки або завантажені матеріали та пропонує оптимальні макети, кольорові схеми та графічні елементи. Вбудований Copilot аналізує текст і допомагає автоматично генерувати унікальний візуальний контент, зберігаючи при цьому стиль користувача. На рисунку 1.5 зображено інтерфейс Microsoft Designer для генерації зображень.

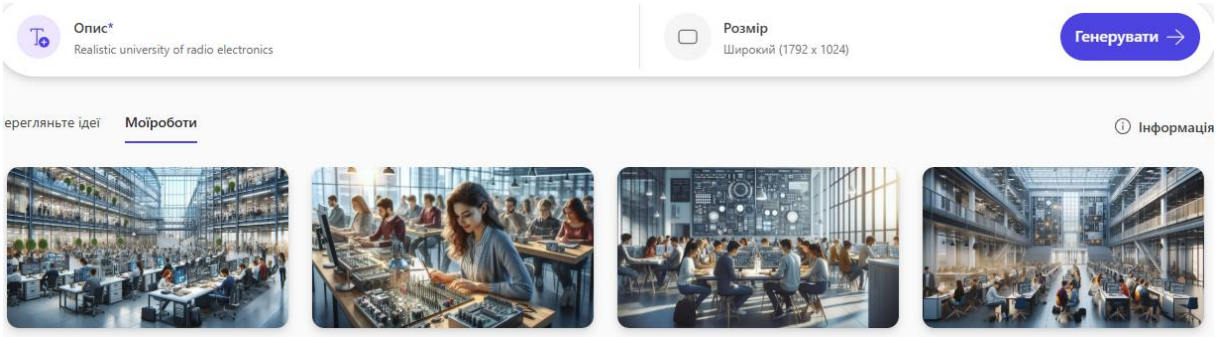

Рисунок 1.5 – Інтерфейс Microsoft Designer для генерації зображень

Microsoft Designer – це зручний інструмент для швидкого створення візуального контенту без потреби у складних графічних редакторах. Можна швидко змінювати фон, додавати нові елементи або коригувати кольори

зображень. Він ідеально підходить для маркетологів, контент-креаторів та користувачів, які хочуть швидко створювати якісні дизайни за допомогою штучного інтелекту.

1.3 Визначення сфери застосування системи генерації зображень за допомогою засобів генеративного штучного інтелекту

Системи генерації зображень за допомогою генеративного штучного інтелекту використовуються в широкому спектрі галузей, де потрібне створення або модифікація візуального контенту. Вони відіграють ключову роль у графічному дизайні, дозволяючи художникам і дизайнерам швидко створювати концепти, ілюстрації та анімації. У сфері реклами та маркетингу такі системи допомагають автоматизувати створення рекламних матеріалів, банерів та контенту для соціальних мереж. Основні галузі застосування таких технологій включають:

– графічний дизайн та ілюстрація – швидке створення концептів, ілюстрацій та художніх робіт без потреби в ручному малюванні;

– фотографія та ретуш – автоматизоване редагування зображень, видалення або додавання об'єктів, зміна фону та покращення якості фото;

– кіно та анімація – створення концепт-артів, візуальних ефектів, фонових сцен та стилізованих кадрів;

– розробка комп’ютерних ігор – генерація текстур, персонажів, локацій та анімацій для відеоігор;

– реклама та маркетинг – створення унікального контенту для рекламних кампаній, соціальних мереж та брендингу;

– електронна комерція – генерація візуального контенту для товарних каталогів, моделювання продуктів у різних умовах;

– архітектура та дизайн інтер’єру – створення візуалізацій приміщень, екстер’єрів та декору;

– мода – розробка ескізів одягу, аксесуарів, візуалізація матеріалів та стилізація образів;

– освіта та наука – створення навчальних матеріалів, наукових ілюстрацій та візуалізацій даних;

– медіа та розваги – персоналізований контент, обкладинки для книг, музичних альбомів, створення «мемів» та артів.

Завдяки генеративному штучному інтелекту ці системи значно спрощують та прискорюють процес створення зображень, роблячи його доступним навіть для людей без спеціальних навичок у дизайні чи малюванні.

1.4 Постановка задачі для системи генерації та виправлення зображень за допомогою текстових підказок в діалозі

Розроблювана система генерації та виправлення зображень за допомогою текстових підказок в діалозі розробляється для операційних систем, що можуть підтримувати роботу з чат-ботом із доступом до мережі Інтернет. Доступ до мережі Інтернет є обов’язковим правилом для користування системою, адже текстовий запит користувача має передаватися на сервер чат-бота, а сервер після обробки запиту і генерації зображення – повернути користувачеві результат генерації відповідно до його текстового запиту.

Система, що розробляється, має забезпечувати автоматичне створення та редагування зображень на основі текстових інструкцій користувача в інтерактивному режимі. Вона повинна розпізнавати запити, інтерпретувати їх відповідно до контексту діалогу та виконувати генерацію чи модифікацію графічного контенту, використовуючи методи генеративного штучного інтелекту.

Для цього модель має бути навчена розпізнавати і генерувати зображення у визначених стилях. Тобто, система має відрізняти

реалістичний стиль від мальованого і на основі цього генерувати подальші об’єкти у визначеному стилі.

Основна мета – забезпечити швидке та точне створення або виправлення зображень відповідно до потреб користувача, враховуючи стилістичні, композиційні та контекстуальні особливості. Система має працювати в реальному часі, зберігаючи послідовність запитів у діалозі та адаптуючи результати відповідно до уточнень користувача.

Для реалізації можливості внесення змін до згенерованого зображення необхідно реалізувати розбиття об’єктів на шари. Це дозволить набагато якісніше вносити зміни і не втрачати вже готові деталі через помилки генеративного штучного інтелекту. Розбиття зображення на шари пропонується робити на:

– головний об’єкт – основний великий об’єкт сцени, що значно виділяється проміж інших і відповідає запиту користувача. Це може бути, наприклад, автомобіль;

– другорядні об’єкти – елементи екстер’єру або інтер’єру в залежності від запиту користувача. Це може бути, наприклад, дорога або підлога;

– тло – елемент дальнього плану, який задає атмосферу композиції. Це може бути, наприклад, ліс, місто або пустеля серед ясного або грозового неба з тучами.

Важливими аспектами роботи системи генерації та виправлення зображень за допомогою текстових підказок в діалозі є розпізнавання контексту запитів, точність обробки інструкцій, реалістичність або стилізованість згенерованих зображень залежно від вимог та можливість інтеграції з іншими інструментами для подальшого редагування. Користувач повинен мати можливість коригувати проміжні результати, додаючи нові текстові підказки, що впливатимуть на кінцевий вигляд зображення.

2 АНАЛІЗ МЕТОДІВ СИСТЕМИ СТВОРЕННЯ ТА КОРЕКЦІЇ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ ТЕКСТОВИХ ПІДКАЗОК В ДІАЛОЗІ

2.1 Аналіз дифузіних моделей для системи створення і корекції зображень в діалозі

Дифузійні моделі – це генеративні моделі, які навчаються відновлювати зображення з шуму, поступово прибираючи його. Під час генерації цей процес іде у зворотному напрямку: починається з шуму і «денойзиться» до фінального зображення, керуючись заданим текстом [10].

Дифузійні моделі трансформують творчі робочі процеси в різних галузях. Ці моделі створюють складні зображення на основі простого тексту або зображення, ітеративно перетворюючи випадковий шум на мистецтво, згенероване ШІ, за допомогою методів дифузії зі зменшенням шуму. Це можна застосувати в багатьох корпоративних сферах, таких, як створення персоналізованого контенту для маркетингу, вигадливих фонів для об'єктів на фотографіях, проектування динамічних високоякісних середовищ і персонажів для ігор та багато іншого [11].

Хоча дифузійні моделі можуть бути корисними інструментами для покращення робочих процесів, при розгортанні в масштабі вони можуть бути надзвичайно обчислювально інтенсивними. Створення однієї партії з чотирьох зображень може зайняти хвилини на неспеціалізованому обладнанні, наприклад, на центральному процесорі, що може заблокувати творчий потік і стати бар'єром для багатьох розробників, які прагнуть відповідати суворим угодам про рівень обслуговування (SLA).

Дифузійні моделі (Diffusion Probabilistic Models) – це клас генеративних моделей, які:

– навчаються відновлювати зображення зі зашумленої версії (тобто денойзити);

– під час генерації – починають з чистого шуму і поступово «виводять» зображення.

Цей підхід дифузійних моделей є протилежністю до автокодерів чи GAN, де зображення створюється безпосередньо або через латентний простір.

Основні переваги дифузійних моделей:

– висока якість зображень (на рівні або вище GAN);

– стабільне навчання;

– контрольованість (через умовне навчання, ControlNet, LoRA тощо);

– гнучкість: можна робити генерацію, редагування, стилізацію, апскейлінг.

Алгоритм роботи моделі для створення зображення:

– зображення поступово перетворюється на чистий шум через гаусівські пертурбації (на кожному кроці додається трохи більше шуму, поки після T кроків матимемо чистий шум);

– наступним кроком модель навчається генерувати зображення шляхом відтворення попереднього стану зображення, тобто відбувається процес прибирання шуму;

– навчена модель генерує зображення шляхом перетворення запиту користувача у вектор (текст у вектор);

– потім цей вектор впливає на генерацію через cross-attention або інший механізм [10].

Таким чином, дифузійні моделі – це сучасний клас генеративних моделей, які відзначаються високою якістю зображень, стабільністю навчання та гнучкістю контролю. Їх ключова ідея полягає в моделюванні процесу відновлення даних із поступово зашумленої версії (реверсивна дифузія), що дає змогу навчати модель на простій задачі денойзінгу, а використовувати – для генерації нових зображень з нуля.

2.2 Аналіз генеративних змагальних мереж для системи створення і корекції зображень в діалозі

Генеративно-змагальні мережі (GAN) – це клас генеративних моделей, у яких два компоненти – генератор та дискримінатор – навчаються у процесі змагання. Генератор вчиться створювати реалістичні зображення, а дискримінатор – відрізняти фейкові зображення від справжніх. Це змагання поступово веде до того, що генератор створює зображення, які неможливо відрізнити від реальних [12].

У діалогових системах, які працюють з мультимодальністю (текст + зображення), GAN можна інтегрувати в генеративний цикл, де користувач надає текстовий запит, а система генерує або змінює зображення відповідно до цього запиту.

Генеративно-змагальні мережі дозволяють використовувати текстові інструкції як додаткове джерело інформації при генерації. Це може виглядати так:

– користувач вводить запит, наприклад: «Намалюй червону машину на фоні гір»;

– текст векторизується (наприклад, через BERT або CLIP);

– вектор тексту подається генератору як умова: G(z|text)→image.

Проведення процесу корекції зображення може поділятися на наступні процеси:

– image-to-image translation: зміна зображення на основі вхідного зображення (наприклад, CycleGAN або Pix2Pix);

– маніпуляції латентним простором: система змінює певні ознаки зображення (наприклад, вираз обличчя, колір предмета) за запитом користувача, змінюючи відповідні латентні вектори [13].

Інструкція користувача для маніпуляції латентним простором:

– користувач вводить запит, наприклад: «Змінити колір машини на синій»;

– система визначає, які латентні параметри відповідають за колір;

– генератор оновлює зображення зі зміненим колірним параметром.

Основні переваги використання генеративно-змагальних мереж:

– висока фотореалістичність (здатні генерувати надзвичайно детальні та реалістичні зображення, особливо обличчя, предмети і текстури);

– миттєва генерація (після навчання генерація дуже швидка – не потрібні багатоетапні процедури, як у дифузійних моделей);

– контроль над стилем та ознаками (за допомогою латентного простору можна змінювати конкретні параметри: емоції, форму, колір, ракурс);

– гнучкість архітектури (GAN добре адаптуються до різних типів задач: від простого шуму до condition-based генерації, стилізації, суперрезолюції).

GAN у діалогових системах забезпечують швидке та реалістичне створення і редагування зображень, проте мають обмеження в контролі та стабільності. Вони ефективні в умовах, де потрібно перетворити текст у зображення або внести конкретні візуальні зміни на основі коротких інструкцій. У поєднанні з NLP-модулями та латентними маніпуляціями GAN залишаються актуальним підходом для мультимодальних діалогових інтерфейсів, хоча сьогодні поступово поступаються місцем дифузійним моделям у задачах з високим ступенем керованості [12].

2.3 Аналіз моделей трансформерів для системи створення і корекції зображень в діалозі

Трансформери – це моделі, які використовують механізм уваги для обробки послідовностей. Спочатку вони були створені для тексту, але згодом адаптовані і для зображень:

– ViT (Vision Transformer): розбиває зображення на патчі й обробляє їх як «слова»;

– ImageGPT, DALL·E, Parti, Flamingo, Emu: трансформери для генерації зображень або мультимодальних задач (текст+зображення);

– BLIP, Flamingo, IDEFICS, GPT–4V: мультимодальні моделі, які об’єднують зображення і мову для відповіді, генерації чи редагування [14].

На відміну від класичних моделей, що обробляють зображення як сукупність пікселів, трансформери, зазвичай, розглядають зображення як послідовність патчів або токенів, схожих за природою на слова в мові. Це дозволяє їм навчатися довготривалим залежностям між частинами зображення, одночасно враховуючи зміст діалогу або інструкцій користувача. Коли система отримує текстову команду, наприклад, «змінити фон на захід сонця», трансформер визначає відповідні ділянки зображення, які мають бути змінені, і генерує або модифікує зображення відповідно до семантичного змісту цієї інструкції. Системи штучного інтелекту для створення зображень працюють наступним чином:

– користувач вводить запит, наприклад: «Зробити картину в стилі Ван Гога». У відповідь трансформер інтерпретує текст та генерує відповідне зображення;

– Image editing – дається інструкція, наприклад: «Змінити небо на нічне», після чого модель аналізує існуюче зображення та редагує його;

– Multimodal dialogue – системи можуть відповідати на запитання, що базуються на зображенні, або виводити підпис до нього, наприклад: «Що на цій картинці?», «Зроби його чорно-білим», «Це панда чи ведмідь?» [15].

Оскільки трансформери не мають жорстко фіксованої структури обробки, вони здатні адаптуватися до широкого спектру задач – від генерації нового візуального контенту до локальних змін в існуючому зображенні. Завдяки цьому такі системи є надзвичайно придатними до діалогових сценаріїв, де користувач може поступово уточнювати свої запити: модель пам'ятає контекст попередніх інструкцій, коригує

зображення відповідно до останньої фрази і водночас зберігає логіку всієї послідовності змін. Переваги систем з використанням трансформера:

– глибоко розуміють природну мову, тому ідеально реалізують наміри користувача;

– можна навчити на парах «інструкція – зображення», що дозволяє моделі добре реагувати на команди;

– здатні утримувати контекст розмови, тобто пам’ятати, про що користувач питав кілька кроків тому, що є критично важливим для діалогу;

– легко доповнювати новими можливостями – наприклад, об’єднувати з дифузійними моделями для кращої якості зображень.

Головні недоліки трансформерів:

– дуже вимогливі до обчислювальних ресурсів – без потужного апаратного забезпечення не запускаються;

– для того, щоб модель добре працювала, її треба навчати на великих датасетах, де є зображення, текст і їх зв’язки;

– редагування зображень недостатньо інтуїтивне й знаходиться в процесі вдосконалення.

Трансформери – це «серце» сучасних діалогових систем із зображеннями. Вони роблять можливою реалізацію простої інструкції, наприклад: «Зробити це красивішим» – і отримати позитивний результат. Проте це потужний інструмент, який вимагає відповідних ресурсів і професійного налаштування [15].

У таких моделях поєднання мовної і візуальної інформації дає змогу будувати системи, здатні взаємодіяти з користувачем на природній мові, при цьому виконуючи точне редагування – як автоматичне, так і під контролем користувача. Саме тому трансформери є базовою технологією для майбутнього діалогових інтерфейсів, де зображення не просто створюються, а редагуються, переосмислюються і взаємно уточнюються у розмові.

2.4 Аналіз моделей автокодерів для системи створення і корекції зображень в діалозі

Автокодери – це клас нейромереж, які навчаються стискати вхідне зображення у компактне відображення (вектор у латентному просторі), а потім відновлювати його назад з цього відображення. Їх роль у системах генерації й редагування зображень у діалозі базується на здатності цього латентного простору виступати як «зона контролю», в якій можна вносити зміни, інтерпретуючи текстову команду, як векторне зрушення або модифікацію [16].

Коли користувач дає інструкцію, наприклад: «Зробити це зображення чорно-білим» або «Змінити обличчя на щасливе», модель із текстового опису формує відповідне латентне зміщення. Це можливо завдяки об'єднанню автокодера з мовною моделлю або трансформером, які вміють переводити інструкцію у зміну характеристик у векторному просторі. Умовні варіанти автокодерів дозволяють включати текст у фазу декодування, впливаючи на результат ще більш точно. Основні переваги використання автокодерів:

– швидкість генерації. Автокодери працюють без ітеративних процедур (як у дифузійних моделях), що забезпечує значно швидшу обробку зображень у реальному часі;

– стиснення та маніпуляція в латентному просторі. Замість редагування пікселів, зміни виконуються у компактному латентному відображенні, що дозволяє ефективно реалізовувати складні трансформації;

– плавні переходи між варіантами. Безперервний латентний простір дає змогу виконувати морфінг – поступові зміни між різними зображеннями або станами;

– невеликі обчислювальні витрати. У порівнянні з GAN або дифузійними моделями автокодери менш вимогливі до ресурсів, що важливо для інтерактивних систем і мобільних застосунків;

– придатні до мультимодальної інтеграції. Легко поєднуються з мовними моделями чи трансформерами для текстово-керованої генерації чи редагування;

– гнучкі у навчанні на нових даних. Автокодери можна швидко перенавчити на нові стилі, домени або конкретні задачі (наприклад, редагування архітектурних фото чи коміксів);

– збереження структури зображення. У редагуванні автокодери не створюють зображення «з нуля», тому краще зберігають композицію, стиль і геометрію оригіналу [17].

У системах, де важлива узгодженість між змістом тексту й візуальним результатом, автокодери можуть інтегруватися з мовними моделями або умовними механізмами, що підвищує керованість процесом. Завдяки безперервному латентному простору автокодери добре підходять для плавних змін і морфінгу між станами зображень. І хоча вони можуть поступатися в деталізації дифузійним або GAN-моделям, саме їх гнучкість, швидкість і контрольованість роблять автокодери особливо корисними для редагування в інтерактивному середовищі.

3 РОЗРОБКА ФУНКЦІОНАЛЬНИХ ВИМОГ ДО СИСТЕМИ СТВОРЕННЯ ТА КОРЕКЦІЇ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ ТЕКСТОВИХ ПІДКАЗОК В ДІАЛОЗІ

3.1 Розробка системних вимог до системи створення і корекції зображень в діалозі

Система штучного інтелекту, що розробляється, повинна запускатися на всіх актуальних пристроях, які дозволяють запуск месенджера Telegram і мають достатню кількість внутрішньої пам’яті для збереження зображення. Цими пристроями є останні версії оперативної системи Windows 10 та Windows 11, а також новітні системи Android та iOs. Усі вище перелічені пристрої обов’язково потребують підключення до всесвітньої мережі Інтернет.

3.2 Визначення функціональних вимог до системи створення і корекції зображень

Визначення функціональних вимог до системи створення і корекції зображень за допомогою IDEF0 – це процес моделювання основних функцій системи з урахуванням входів, виходів, механізмів і керування. Метод IDEF0 дозволяє структуровано описати, що саме має робити система (наприклад, завантаження зображень, редагування, збереження), що є вхідними даними (наприклад, файли зображень), які ресурси використовуються (редактор, ШІ–моделі), якими правилами керується процес (формати, вимоги користувача), які результати очікуються (отримане чи відкориговане зображення) [18], [19].

На рисунку 3.1 зображено концептуальну діаграму для системи штучного інтелекту для створення і корекції зображень за допомогою текстових підказок в діалозі.

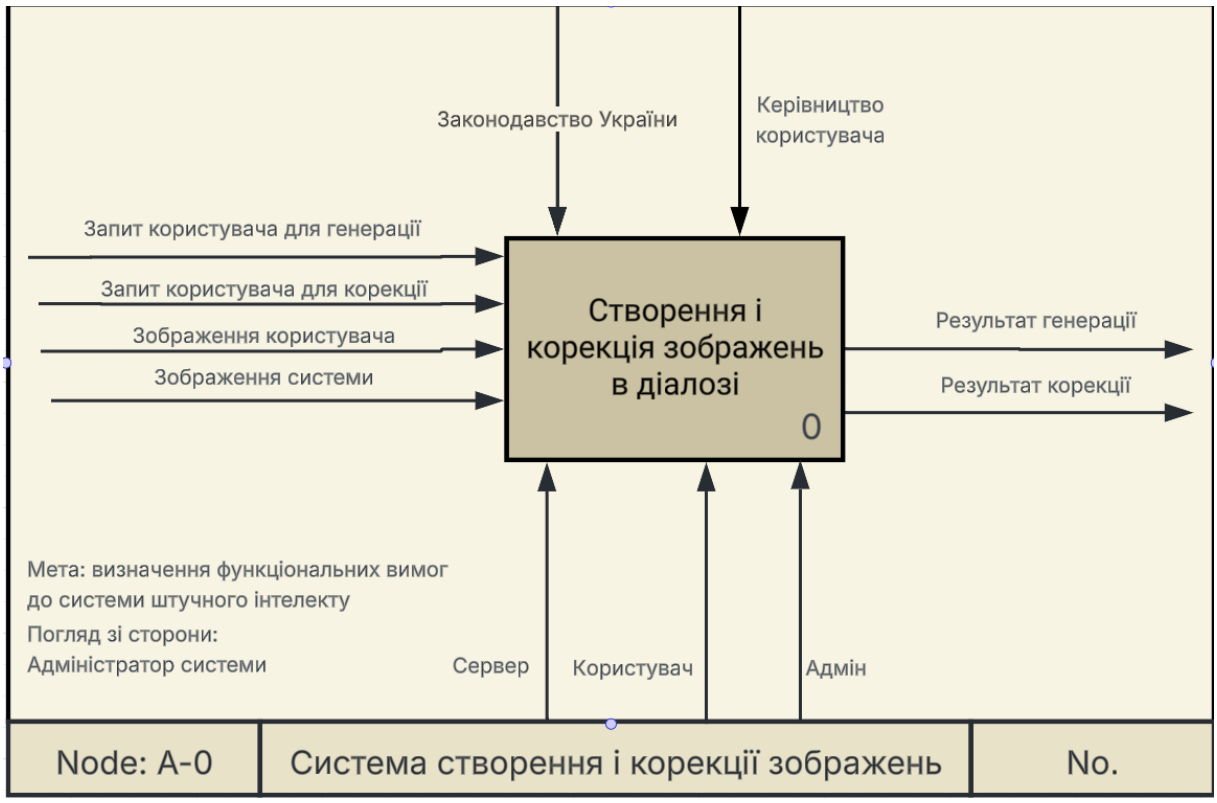

Рисунок 3.1 – Концептуальна діаграма для системи створення і корекції зображень в діалозі

У центрі діаграми знаходиться основна функціональна блок-схема з номером 0, яка представляє головний процес – «Створення і корекція зображень в діалозі». Цей блок описує основну діяльність системи, яка обробляє запити користувача, формує або змінює зображення і видає результат.

Зліва на діаграму заходять основні вхідні дані, які система використовує для виконання своєї функції:

– запит користувача для генерації – це команда або повідомлення, у якому користувач формулює бажане нове зображення;

– запит користувача для корекції – це інструкція щодо змін або редагування вже існуючого зображення;

– зображення користувача – це зображення, надані користувачем для подальшого аналізу, зміни або обробки;

– зображення системи – це внутрішні або збережені зображення, що вже є в базі даних системи і можуть бути використані для корекції чи як шаблони.

Зверху до блоку підходять керуючі впливи, які задають обмеження або правила роботи:

– законодавчі та нормативні акти України – встановлюють правові рамки, зокрема, щодо допустимого вмісту, авторських прав, персональних даних тощо;

– керівництво користувача – містить правила використання системи, інструкції та, можливо, персональні налаштування або обмеження, задані самим користувачем.

Знизу підходять ресурси, які забезпечують виконання функції:

– сервер – технічна інфраструктура, на якій працює система, виконує обчислення, зберігає дані та керує обробкою зображень;

– користувач – не лише ініціює запити, але може виступати джерелом уточнень, зворотного зв'язку чи додаткових дій у процесі діалогу;

– адмін – виконує роль модератора або технічного оператора, який може втручатися в процес, наприклад, у випадках порушення правил або збоїв.

Праворуч виходять результати виконання функції:

– результат генерації – створене нове зображення відповідно до запиту користувача;

– результат корекції – змінене або вдосконалене зображення на основі початкового варіанта.

Наступним кроком проводиться декомпозиція для системи створення та корекції зображень і розробляється головна бізнес-функція на додаткові бізнес-функції, з яких складається головний функціонал системи. Кожна функція, відповідно до нотації IDEF0, має свої вхідні і вихідні дані. Декомпозицію першого рівня зображено на рисунку 3.2.

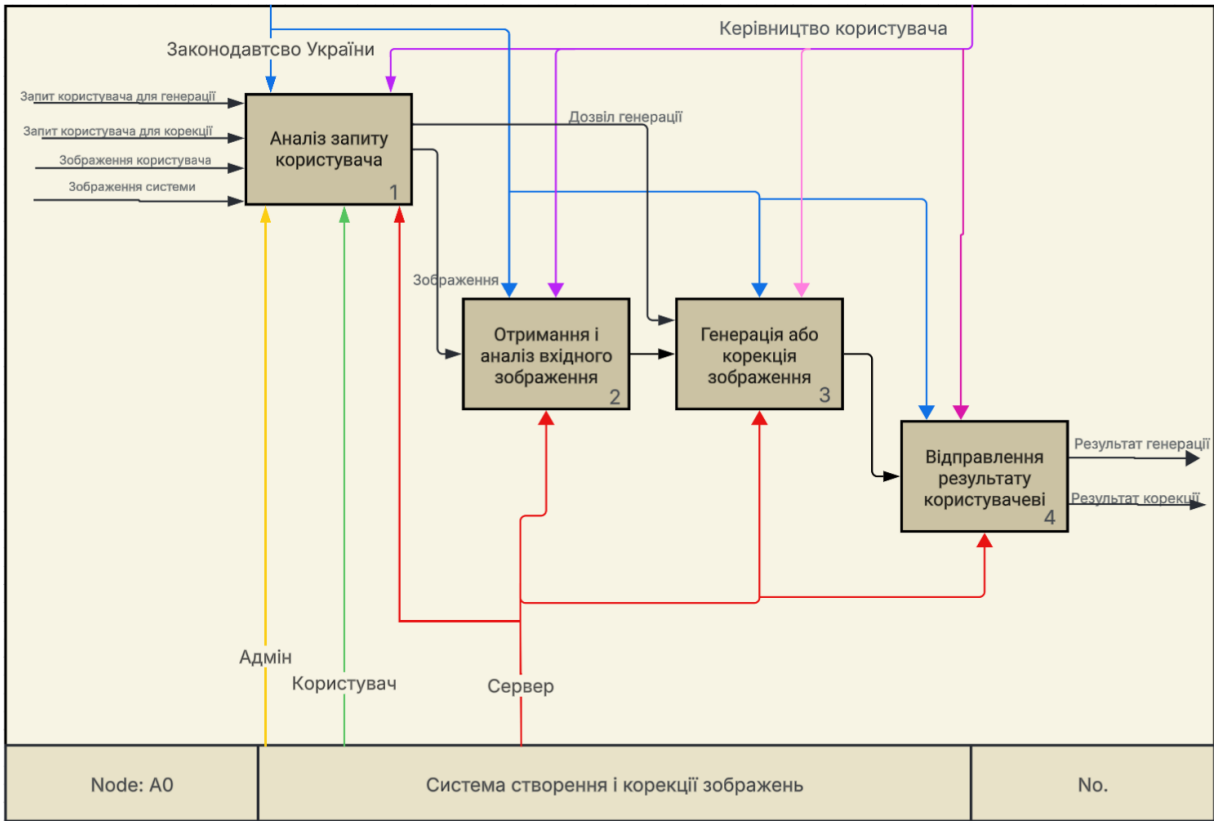

Рисунок 3.2 – Перший рівень декомпозиції для системи створення і корекції зображень в діалозі

Ця концептуальна діаграма IDEF0 моделює систему створення і корекції зображень, розкриваючи її функціонування на першому рівні деталізації. У центрі уваги – процес, який починається з аналізу запиту користувача і завершується передачею результату назад цьому користувачу. Діаграма структурує роботу системи у вигляді послідовності чотирьох логічно пов’язаних функцій, кожна з яких виконує конкретну роль у загальному процесі.

Процес починається з того, що система приймає запит користувача, який може містити інструкції щодо генерації нового зображення або побажання до корекції наявного. Цей запит разом з додатковими вхідними даними – такими як, надані користувачем зображення, зображення з бази даних системи, а також зовнішні чинники, зокрема, законодавство України та персональні обмеження користувача, – потрапляє до блоку аналізу. Тут

здійснюється первинна оцінка: визначається, чи є запит допустимим, які ресурси знадобляться для його виконання, і чи не порушує він юридичних чи етичних норм.

Якщо користувач подав зображення для редагування або система обрала відповідне зображення з бази, процес переходить до наступного етапу – отримання та аналізу вхідного зображення. На цьому кроці перевіряється технічна придатність файлу, його вміст і відповідність очікуванням, заданим у запиті. Ці дані готуються для наступного етапу – власне генерації або корекції зображення.

Третій блок реалізує центральну функцію системи – він відповідає за створення нового зображення або модифікацію наданого. Цей процес враховує як параметри самого запиту, так і результати попереднього аналізу, знову ж таки дотримуючись вимог законодавства та налаштувань користувача. Алгоритми генерації чи обробки зображень працюють на серверній частині системи, яка забезпечує необхідні обчислювальні ресурси.

Останній етап – відправлення результату користувачеві – забезпечує зворотний зв’язок. Тут результат, сформований на попередньому етапі, приводиться у відповідний формат, перевіряється, чи відповідає запиту, і надсилається назад користувачеві. Залежно від типу запиту результат позначається як результат генерації або як результат корекції. У разі потреби до процесу може бути залучений адміністратор для контролю якості або вирішення спірних ситуацій.

Уся система функціонує у взаємодії з чотирма головними зовнішніми факторами: користувачем, адміністратором, сервером і нормативними обмеженнями. Вони формують потоки інформації, керують або обмежують виконання функцій та забезпечують правомірність, стабільність і результативність роботи системи.

Декомпозиція бізнес-функції «Аналіз запиту користувача» в системі створення і корекції зображень розкриває детальні етапи, що забезпечують

коректну інтерпретацію, правову перевірку та маршрутизацію запиту користувача в рамках системи. Цей процес є критичним, оскільки саме він визначає допустимість та напрям подальших дій системи – генерацію нового зображення або корекцію наданого. На рисунку 3.3 зображено декомпозицію бізнес-процесу «Аналіз запиту користувача».

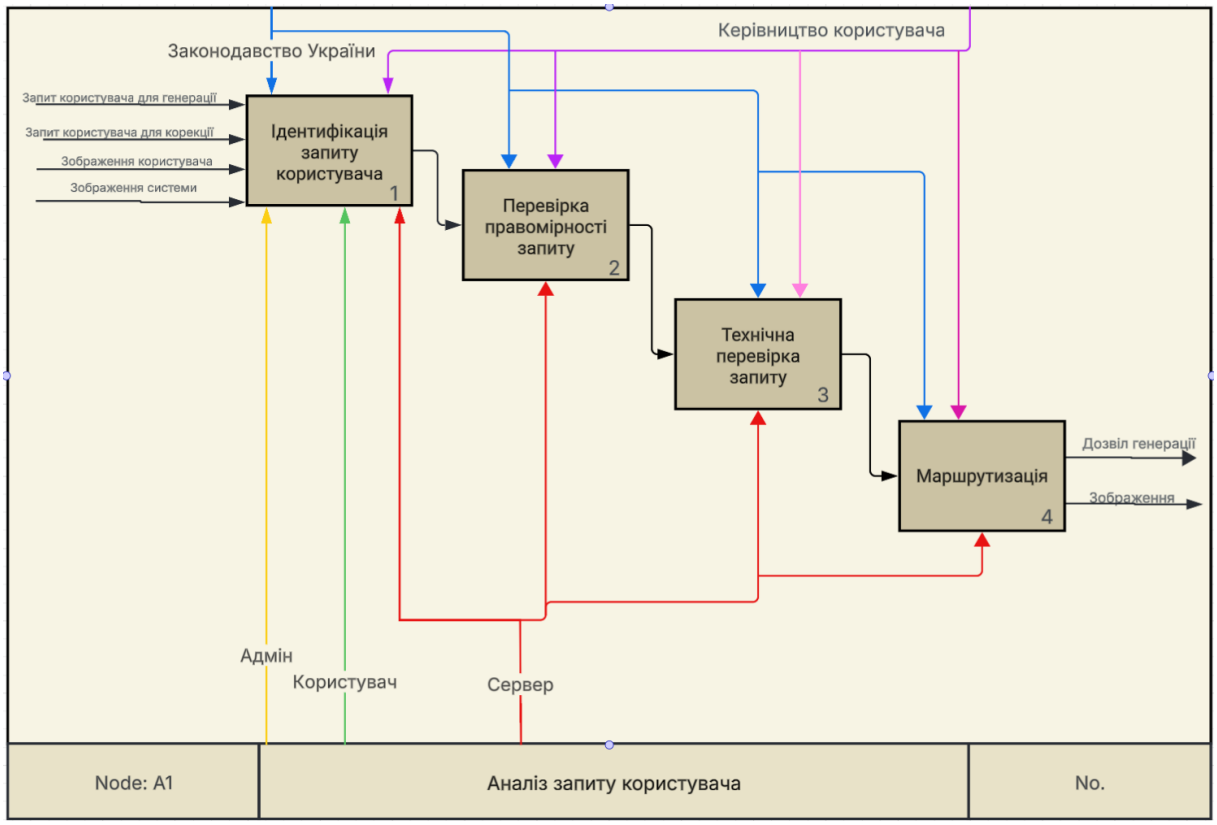

Рисунок 3.3 – Декомпозиція бізнес-процесу «Аналіз запиту користувача»

Ця концептуальна діаграма IDEF0 представляє декомпозицію бізнес-процесу «Аналіз запиту користувача» у контексті системи створення і корекції зображень. Вона демонструє, як вхідні запити користувачів проходять через кілька логічно послідовних етапів, перш ніж система зможе дозволити створення чи редагування зображень.

Процес починається з надходження запитів двох типів: на генерацію або корекцію зображення. До запитів можуть додаватися зображення від самого користувача або вибрані із системної бібліотеки. На першому етапі

система виконує ідентифікацію запиту, визначаючи його тип і зміст. Цей крок здійснюється з урахуванням обмежень, пов’язаних із роллю користувача, вказівками адміністратора, а також зовнішніми правовими нормами, зокрема, законами України.

Після ідентифікації запит надходить на перевірку правомірності, де відбувається аналіз відповідності запиту правовим нормам та внутрішнім правилам, продиктованим керівництвом користувача. Це гарантує, що запити, які містять заборонений або небезпечний вміст, не будуть допущені до подальшої обробки.

Далі відбувається технічна перевірка запиту, що охоплює валідацію вхідних даних – перевірку наявності необхідних параметрів, форматів, а також логічну коректність інструкцій. Усі три попередні блоки працюють із залученням сервера, як технічного інструмента для обробки та передачі даних.

На фінальному етапі здійснюється маршрутизація – на основі результатів правової та технічної перевірки система ухвалює рішення про подальший напрямок запиту: передати його до генератора зображень або до модуля редагування. Якщо всі перевірки пройдені успішно, формується дозвіл на генерацію та відповідне зображення, яке надсилається користувачеві.

Для декомпозиції бізнес-процесу «Отримання і аналіз вхідного зображення» у системі створення і корекції зображень необхідно виділити чотири основні функції. На рисунку 3.4 зображено декомпозицію бізнес-процесу «Отримання і аналіз вхідного зображення».

На концептуальній діаграмі IDEF0 представлено декомпозицію бізнес-процесу «Отримання і аналіз вхідного зображення», що є частиною системи створення та корекції зображень на основі текстових підказок в діалозі. Процес починається з надходження зображення, яке направляється на перевірку параметрів. На цьому етапі відбувається оцінка технічних характеристик, таких, як формат, розмір, цілісність файлу, що виконується

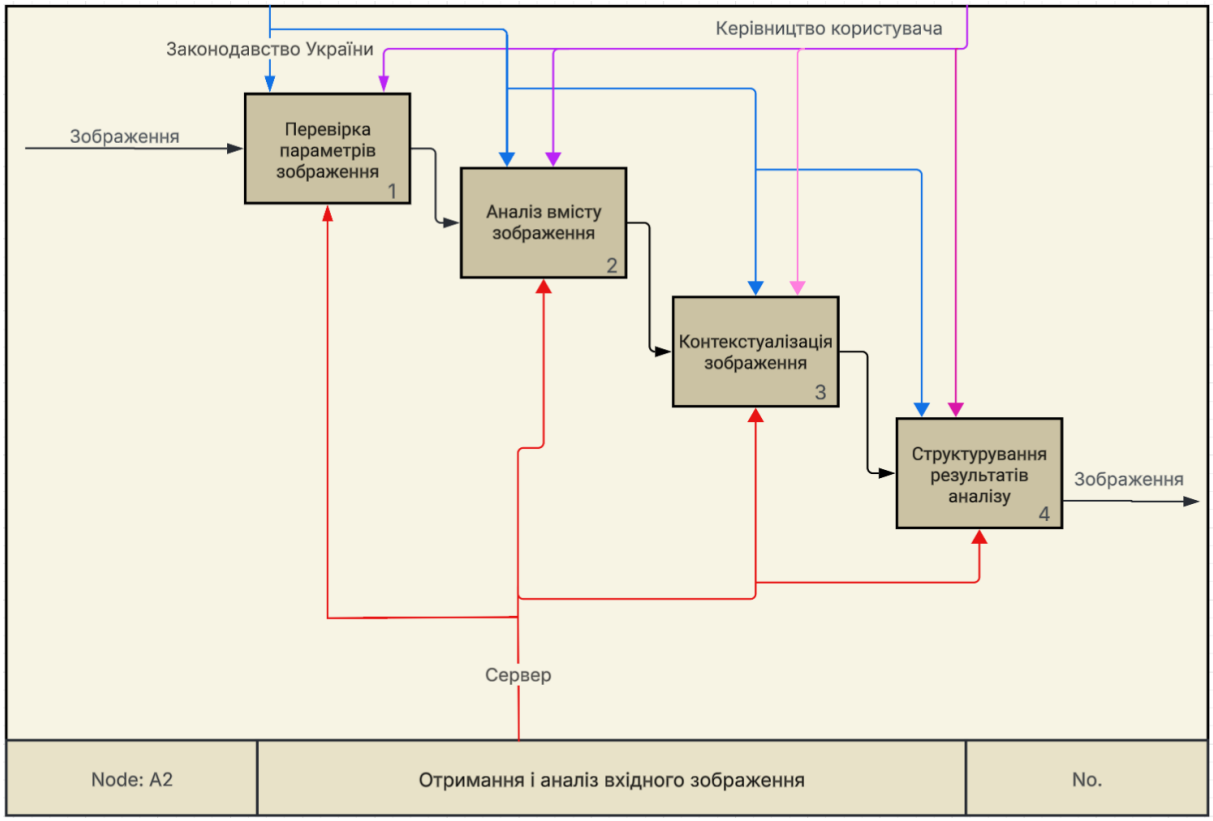

Рисунок 3.4 – Декомпозиція бізнес-процесу «Отримання і аналіз вхідного зображення»

під контролем норм законодавства України і доступу до серверної інфраструктури.

Далі зображення передається до блоку аналізу вмісту, де система за допомогою засобів комп'ютерного зору та штучного інтелекту розпізнає об'єкти, сцени, можливий стиль чи наявність заборонених елементів. Результати цього аналізу переходять до етапу контекстуалізації – прив'язки зображення до поточного запиту користувача, враховуючи його наміри, історію взаємодії з системою або поточну сесію. Тут також враховуються інструкції або настанови від керівництва користувача, що можуть задавати пріоритети чи обмеження.

Завершальним етапом є структурування результатів аналізу. Тут здійснюється інтеграція попередніх результатів у стандартизований формат, придатний для подальшого використання – чи то для генерації нового

зображення, чи то для його корекції. Вихідним результатом цього процесу є підготовлене до наступних дій зображення, яке враховує всі технічні, змістовні й контекстуальні аспекти, визначені в процесі аналізу. Уся система спирається на серверні ресурси та нормативне забезпечення, забезпечуючи юридичну відповідність, стабільність і масштабованість обробки.

Наступним кроком є проведення декомпозиції бізнес-процесу «Генерація або корекція зображення», який також розподіляється на чотири функції. Декомпозиція генерації та корекції зображення зображена на рисунку 3.5.

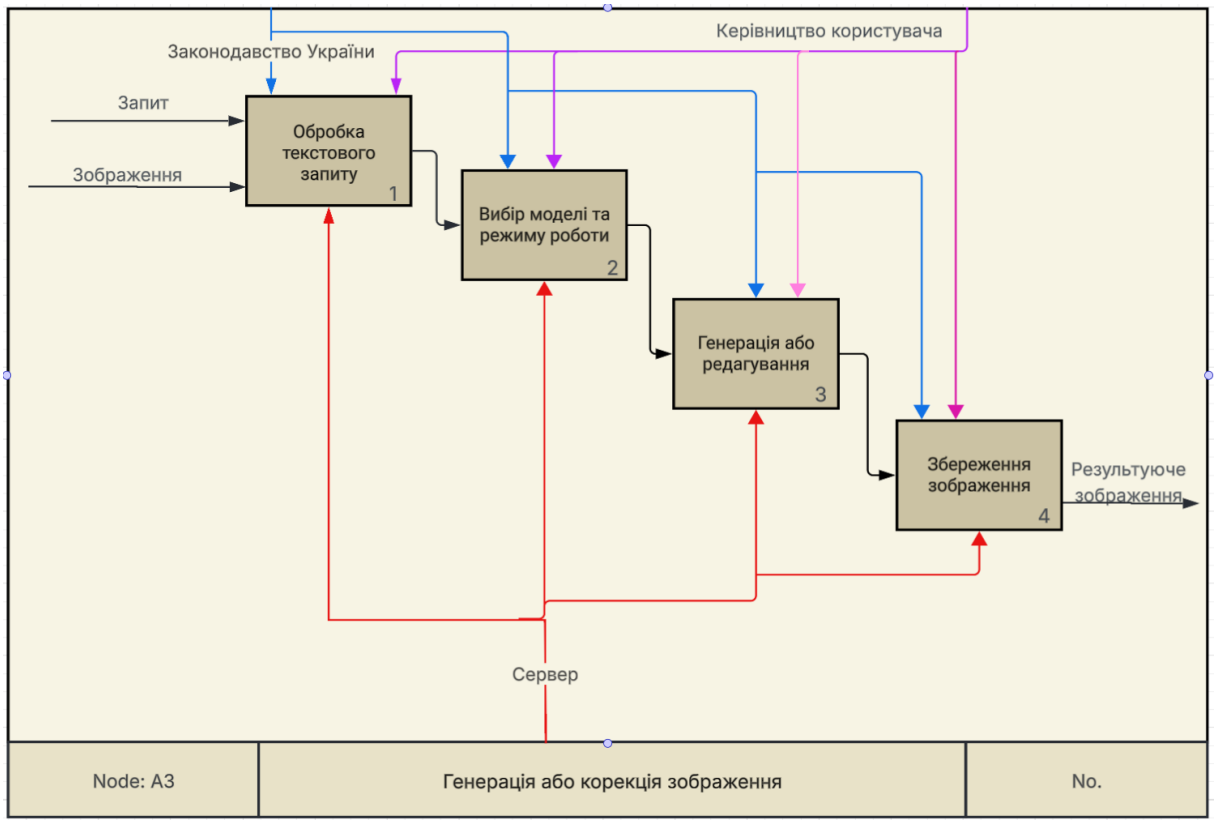

Рисунок 3.5 – Декомпозиція бізнес-процесу «Генерація або корекція зображення»

На наданій концептуальній діаграмі IDEF0 представлено декомпозицію бізнес-процесу «Генерація або корекція зображення» у

вигляді чотирьох послідовних функціональних блоків, які узгоджено перетворюють вхідний текстовий запит і зображення на результуюче зображення.

Процес починається з обробки текстового запиту, де система аналізує запит користувача, перевіряє його відповідність законодавству, отримує доступ до початкового зображення (у випадку корекції), а також враховує керівництво користувача щодо стилю, формату чи результату. Цей етап дає системі змогу зрозуміти намір користувача – чи потрібно згенерувати нове зображення, чи змінити наявне.

Далі виконується вибір моделі та режиму роботи. На цьому етапі система обирає відповідну нейромережу або алгоритм залежно від завдання – генерація «з нуля» або редагування. Також встановлюються параметри якості, стильові налаштування, формат виводу. При цьому зберігається зв’язок з сервером, який забезпечує обчислювальні ресурси.

Третій блок – генерація або редагування – є центральним етапом, де на основі вхідних даних і обраної моделі виконується безпосередня побудова нового зображення або його модифікація. Система застосовує інструкції користувача, дотримується заданого стилю та параметрів.

Після створення зображення здійснюється збереження зображення, яке включає перевірку результату, його фінальне форматування, структурування та збереження у відповідному форматі.

Наступною і останньою декомпозицією бізнес-функції є «Відправлення результату користувачеві», яка складається з двох функцій. Декомпозицію функції відправлення результату користувачеві в чат зображено на рисунку 3.6.

Бізнес-процес «Відправлення результату користувачеві» складається з двох основних етапів, які відображені на діаграмі IDEF0, а саме:

– формування повідомлення для відповіді. На цьому етапі система отримує результуюче зображення, яке було створене або відредаговане раніше. З урахуванням законодавства України та керівництва користувача,

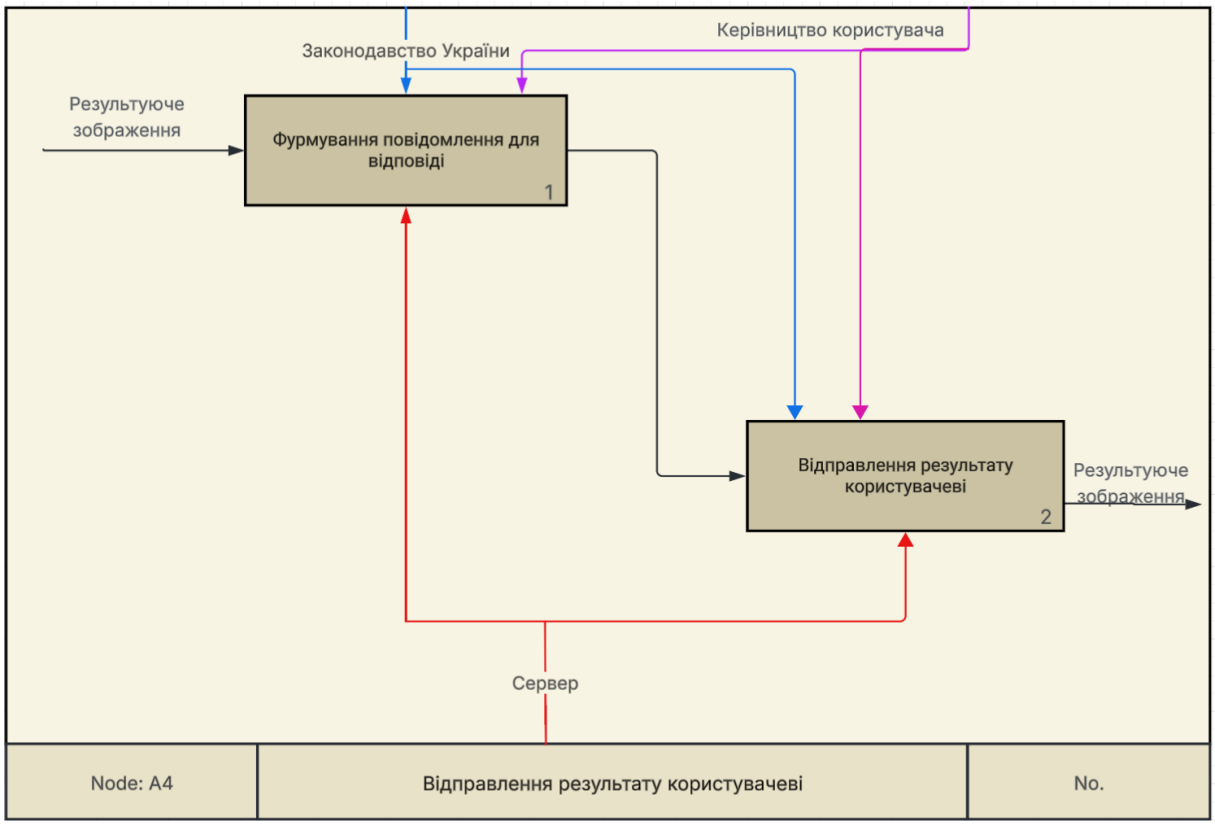

Рисунок 3.6 – Декомпозиція бізнес-процесу «Відправлення результату користувачеві»

система формує повідомлення, яке міститиме це зображення, а також додаткову інформацію для користувача. Цей процес гарантує, що вміст повідомлення є коректним, правомірним і відповідає встановленим правилам оформлення;

– відправлення результату користувачеві. Після формування повідомлення відбувається його відправлення користувачеві через сервер. Сервер виступає технічним посередником між системою і користувачем. На цьому етапі система забезпечує доставку зображення та повідомлення у відповідному форматі, згідно з інструкціями користувача і чинними правовими нормами.

3.3 Розробка моделі потоків даних для системи створення і корекції зображень

Модель потоків даних (Data Flow Diagram, DFD) для системи створення і корекції зображень – це формалізований спосіб представлення того, як інформація (дані) рухається в межах системи, які об’єкти ці дані генерують, обробляють або споживають, та через які процеси проходить. Така модель є важливою частиною структурного аналізу системи, оскільки дозволяє візуалізувати логіку взаємодії між підсистемами без заглиблення у технічні деталі реалізації [20].

У контексті системи створення і корекції зображень, модель потоків даних охоплює головні джерела та споживачів інформації (користувача, адміністратора й сервер), а також центральний бізнес-процес, який виконує обробку запитів, вибір режиму роботи, генерацію чи редагування зображення та передачу результату назад користувачеві. Потоки даних у цій моделі включають вхідні текстові запити, зображення, інструкції користувача, правові обмеження, а також вихідні результати у вигляді згенерованих або відредагованих зображень.

Користувач ініціює потік даних, вводячи запит, який надходить до системи для попередньої обробки. На цьому етапі здійснюється аналіз тексту, перевірка його на відповідність законодавству та перетворення в технічне завдання. Далі система визначає, яку модель або алгоритм потрібно застосувати, враховуючи тип запиту, доступні ресурси та вказівки користувача. Потім відбувається генерація або редагування зображення, після чого результат проходить через етап збереження й додаткової перевірки.

У завершальному етапі збережене зображення форматується у відповідь, яка надсилається користувачеві. Сервер при цьому виконує функцію посередника, забезпечуючи технічну передачу даних між процесами та зовнішніми агентами. Таким чином, модель потоків даних

демонструє логіку і цілісність усієї інформаційної взаємодії в системі створення і корекції зображень. На рисунку 3.7 зображено діаграму потоків даних для системи створення і корекції зображень за допомогою текстових підказок в діалозі.

Рисунок 3.7 – Діаграма потоків даних для системи створення і корекції зображень в діалозі

На діаграмі зображено модель потоків даних для системи створення і корекції зображень у діалозі. У центрі розташований основний процес – «Створення і корекція зображень в діалозі», який отримує, обробляє та передає інформацію між трьома ключовими зовнішніми суб’єктами: адміністратором, користувачем і сервером.

Адміністратор надає системі правила та обмеження, оновлення системи та керівництво користувача, а натомість отримує результати

тестувань та логи помилок і збоїв. Таким чином, адміністративна роль забезпечує нормативну та функціональну підтримку системи.

Користувач взаємодіє з системою через запити: надсилає запити на генерацію зображень, запити на корекцію, а також самі зображення для корекції. У відповідь він отримує результат генерації або корекції, та можливе повідомлення про помилку у разі невдачі генерації.

Сервер у цій моделі є проміжною технічною ланкою, яка надає системі запити користувача та зображення користувача, і отримує назад результуюче зображення або повідомлення про помилку. Таким чином, сервер слугує каналом обміну даними між користувачем і системою.

3.4 Визначення вимог до інтерфейсу системи штучного інтелекту створення і корекції зображень

Визначення вимог до інтерфейсу системи штучного інтелекту для створення та корекції зображень охоплює кілька аспектів, щоб забезпечити зручність, ефективність і функціональність. Нижче наведено основні вимоги для системи створення і корекції зображень:

– юзер-орієнтований інтерфейс – інтерфейс має бути інтуїтивно зрозумілим навіть для користувачів без технічного досвіду;

– швидкий доступ до основних функцій, тобто користувач повинен одразу бачити і вибирати потрібні йому дії з переліку основних функцій (наприклад, створення зображень, корекція кольорів, фільтри, завантаження/експортування файлів).

3.5 Розробка діаграми варіантів використання системи штучного інтелекту створення і корекції зображень

Діаграма використання, ще відома як діаграма прецедентів, створюється на основі двох типів елементів діаграми:

– актор – це зовнішній учасник системи, який взаємодіє з нею для досягнення певної мети. Це може бути людина, інша система або пристрій, який ініціює або бере участь у сценаріях використання. Він знаходиться за її межами, але виконує дії, що запускають певні процеси всередині неї;

– прецедент (варіант використання) – це опис конкретної функціональної можливості системи, яка активується актором для досягнення певного результату. Прецедент описує не внутрішню реалізацію, а поведінку системи з точки зору користувача – як система реагує на запит і який результат повертає [21].

Для системи штучного інтелекту для створення і корекції зображення за допомогою текстових підказок в діалозі було створено діаграму варіантів використання, яку зображено на рисунку 3.8.

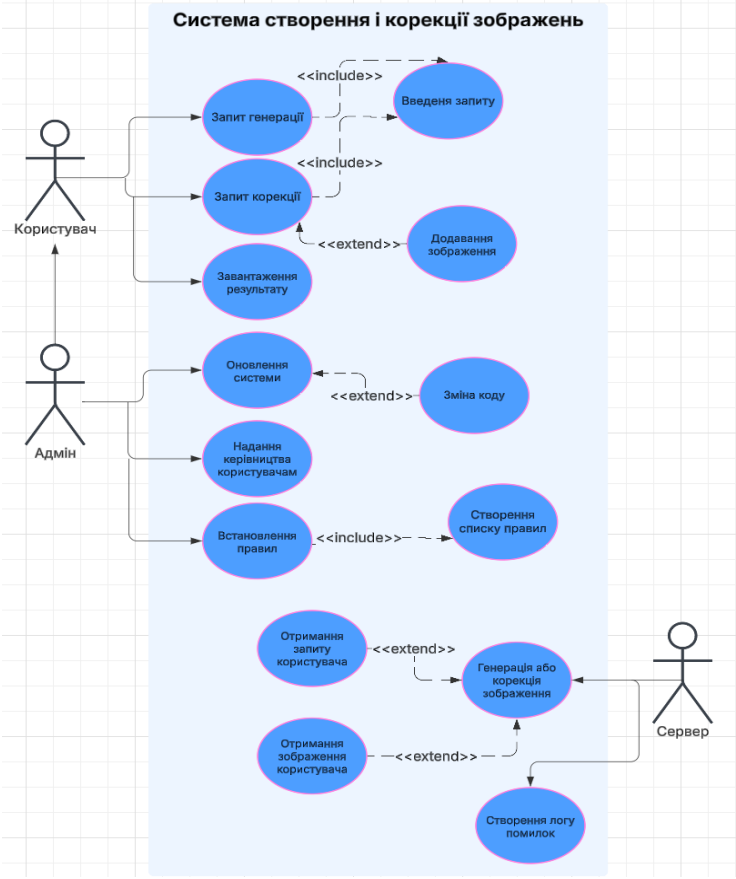

Рисунок 3.8 – Діаграма варіантів використання для системи штучного інтелекту для створення і корекції зображень у діалозі

Наведена діаграма прецедентів для системи штучного інтелекту зображує трьох акторів:

– користувач Telegram – створює команди для сервера генерації зображень і взаємодіє з його інтерфейсом;

– сервер генерації зображень – отримує вказівки користувача, проводить генерацію або корекцію зображення і після чого повертає користувачеві результат;

– адміністратор – наслідує використання функцій користувача Telegram, а також може оновлювати систему штучного інтелекту, передивлятися список помилок і тестувати систему.

Розглянемо бізнес-функції для кожного актора системи штучного інтелекту для створення і корекції зображень.

Користувач взаємодіє з ботом через Telegram, використовуючи команди для створення та корекції зображень. Його основні бізнес-функції:

– створення зображень за описом: користувач може відправляти текстові команди або опис зображення, яке потрібно створити (наприклад, «створити портрет в стилі Ван Гога»);

– корекція існуючих зображень: користувач може завантажити зображення для редагування, наприклад, змінити кольори, додати ефекти, змінити фонове середовище;

– вибір параметрів генерації: користувач може вибирати певні параметри генерації (генеративну модель, розмір зображення, стиль, кольорова палітра) через кнопки або меню;

– отримання результату: після обробки зображення сервером, користувач отримує зображення у форматі медіа (JPEG, PNG) або посилання для завантаження. Завдяки цьому користувач може зберігати створені або відредаговані зображення на своєму пристрої.

Сервер генерації зображень відповідає за обробку запитів, генерування або корекцію зображень згідно з вказівками користувачів. Основні бізнес-функції серверу:

– обробка запитів користувачів: сервер отримує текстові запити або файли з зображеннями від користувачів через бот;

– генерація зображень: сервер за допомогою алгоритмів для створення зображень генерує зображення на основі опису, заданого користувачем;

– корекція зображень: коли користувач завантажує зображення для редагування, сервер проводить корекцію відповідно до вказівок користувача (наприклад, змінює кольори, накладає фільтри);

– збереження історії зображень: сервер має тимчасове сховище для збереження проміжних та готових результатів;

– обробка помилок та некоректних запитів: сервер перевіряє, чи відповідають запити користувачів правильному формату (наприклад, правильно завантажене зображення або опис). А також у разі помилки повертає користувачеві відповідне повідомлення про помилку.

Адміністратор має розширений доступ до всіх функцій користувачів та додаткові можливості для управління системою. Нижче наведено основні бізнес-функції адміністратора:

– оновлення та покращення системи штучного інтелекту: адміністратор може оновлювати штучний інтелект або моделі генерації зображень, додаючи нові стилі, фільтри або алгоритми;

– моніторинг та аналіз помилок: адміністратор має доступ до журналу помилок та може переглядати всі невірні запити або збої в процесі генерації зображень;

– тестування системи: адміністратор може тестувати систему на різних типах запитів та зображень, перевіряти якість генерації або корекції. Також може запускати автоматичні тести для перевірки стабільності та ефективності роботи системи.

Усі описані вище бізнес-функції мають вкладені підфункції, що виконуються безпосередньо для забезпечення роботи основного функціоналу системи.

3.6 Розробка діаграми класів системи штучного інтелекту створення і корекції зображень

Діаграма класів – це один із ключових елементів візуального моделювання системи, що використовується в об’єктно-орієнтованому проєктуванні. Вона відображає структуру програмної системи, зокрема, класи, їх атрибути, методи (функції) та взаємозв’язки між ними. У нотації UML (Unified Modeling Language) діаграма класів є зручною формою представлення логіки роботи системи ще до її реалізації [22].

У контексті системи створення і корекції зображень діаграма класів виконує кілька важливих функцій:

– вона дозволяє чітко визначити головні об'єкти системи: користувачів, запити, серверну логіку, зображення, а також адміністративний функціонал. Це дає змогу розмежувати відповідальність кожного компонента, спростити розробку і підтримку коду, а також забезпечити масштабованість;

– діаграма допомагає зрозуміти, як відбувається обробка запиту: від формування користувачем тексту запиту до генерації зображення на основі заданої моделі. Зокрема, вона показує, які об'єкти беруть участь у передачі інформації, де генерується зображення, як фіксуються помилки та які є шляхи повернення результату користувачу;

– діаграма є засобом комунікації між членами команди: розробниками, аналітиками, дизайнерами інтерфейсу, тестувальниками. Вона дозволяє уникати непорозумінь, оскільки всі учасники бачать єдину узгоджену модель системи;

– в рамках складних систем зі штучним інтелектом, діаграма класів дає змогу формалізувати і контролювати потік даних, щоб забезпечити коректне функціонування алгоритмів, роботу моделей та обробку помилок. Вона є основою для побудови стабільної архітектури системи, яка повинна працювати точно, швидко й надійно [22].

Саме тому для системи створення і корекції зображень за допомогою текстових підказок в діалозі було розроблено діаграму класів, яку зображено на рисунку 3.9.

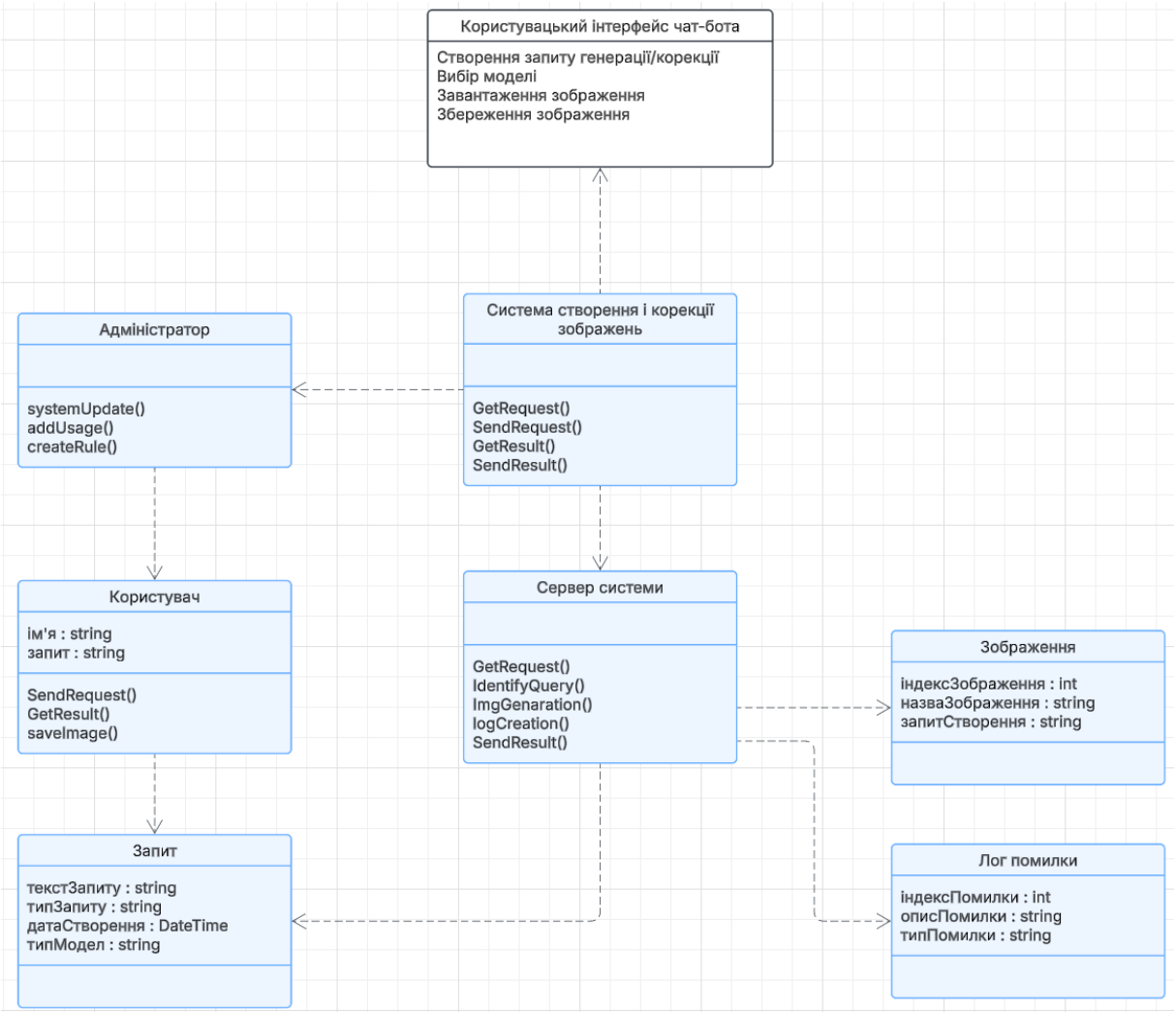

Рисунок 3.9 – Діаграма класів для системи створення і корекції зображень за допомогою текстових підказок в діалозі

Ця діаграма класів описує структуру системи створення й корекції зображень. Центральним елементом є клас «Система створення і корекції зображень», який обробляє запити користувачів та взаємодіє з іншими компонентами. Взаємодія відбувається через методи отримання запиту, надсилання запиту, отримання результату та відправлення результату. Над

системою знаходиться інтерфейс чат-бота, через який користувачі надсилають запити, обирають модель, завантажують і зберігають зображення.

Користувач має атрибути «ім’я» та «запит» і може надсилати запит, отримувати результат та зберігати зображення. Адміністратор є розширенням користувача та має додаткові можливості: оновлювати систему, додавати інструкції використання та створювати правила.

Клас «Запит» описує структуру запиту, що надсилається користувачем. Він містить текст запиту, тип запиту, дату створення та обрану модель. Цей запит обробляється сервером системи, який отримує запит, ідентифікує його, виконує генерацію зображення, створює лог помилок при потребі та відправляє результат.

Згенеровані або скориговані зображення описуються окремим класом «Зображення», який містить індекс, назву зображення та відповідний запит, що його створив. Якщо під час обробки виникають помилки, вони фіксуються у класі «Лог помилки», де зберігаються індекс, опис і тип помилки.

Загалом клас «Користувач» зберігає інформацію таких атрибутів:

– ім’я користувача – текстове поле для імені;

– запит – текстовий запит користувача.

Клас також містить такі функції:

– SendRequest() – функція для надсилання запиту в систему;

– GetResult() – функція для отримання результату обробки запиту;

– SaveImage() – функція для збереження зображення, отриманого у відповідь на запит.

Клас «Адміністратор» є розширенням класу користувача (тобто наслідує функції і атрибути) і має додаткові функції:

– systemUpdate() – функція для оновлення системи;

– addUsage() – функція для додавання інформації про використання або статистику;

– createRule() – функція для створення нових правил обробки запитів або модерації.

Клас «Запит» зберігає такі атрибути:

– текстЗапиту – текстове поле, що містить вміст запиту;

– типЗапиту – тип запиту, який визначає характер операції (наприклад, генерація чи корекція);

– датаСтворення – дата і час створення запиту у форматі DateTime;

– типМоделі – тип моделі штучного інтелекту, яка буде використана для обробки запиту.

Клас «Зображення» має такі атрибути:

– індексЗображення – числовий ідентифікатор зображення;

– назваЗображення – текстове поле з назвою або ідентифікатором файлу зображення;

– запитСтворення – текстове поле або посилання на запит, який спричинив створення зображення.

Клас «Лог помилки» містить наступні атрибути:

– індексПомилки – числовий ідентифікатор помилки;

– описПомилки – текстове пояснення суті помилки;

– типПомилки – класифікація помилки за її типом (наприклад, системна, мережна, логічна тощо).

Клас «Сервер системи» виконує основну логіку обробки запитів і має такі функції:

– GetRequest() – функція для отримання запиту від користувача або інтерфейсу;

– IdentifyQuery() – функція для ідентифікації типу запиту і його параметрів;

– ImgGeneration() – функція для генерації зображення на основі запиту;

– logCreation() – функція для створення запису про можливу помилку під час обробки;

– SendResult() – функція для відправлення згенерованого результату назад до користувача або системи.

Клас «Система створення і корекції зображень» є ключовим обробником запитів. Має такі функції:

– GetRequest() – функція отримання запиту;

– SendRequest() – функція надсилання запиту на сервер або обробку;

– GetResult() – функція отримання результату обробки;

– SendResult() – функція надсилання результату користувачу або інтерфейсу.

Користувацький інтерфейс чат-бота не має атрибутів чи методів у вигляді класу, але виконує такі дії:

– створення запиту генерації/корекції;

– вибір моделі для генерування;

– завантаження зображення в систему;

– завантаження зображення з системи на пристрій користувача.

3.7 Розробка діаграми послідовності для прецеденту «Створення зображення»

Діаграма послідовності дій (sequence diagram) – це тип діаграми UML, який відображає взаємодію між об’єктами системи у хронологічному порядку. Вона показує, як об’єкти надсилають один одному повідомлення (виклики методів) у процесі реалізації певного сценарію або прецеденту, відображаючи порядок і логіку дій у системі [23].

У випадку прецеденту «Створення зображення» діаграма послідовності дій фіксує кроки, які відбуваються від моменту формування користувачем запиту на генерацію до отримання зображення у відповідь. Така діаграма дозволяє візуально прослідкувати, які саме об’єкти

взаємодіють у цьому процесі, в якій черговості відбувається обмін повідомленнями та яка логіка відповідає за генерацію результату. На рисунку 3.10 зображено діаграму послідовності дій для прецеденту «Створення зображення» [23].

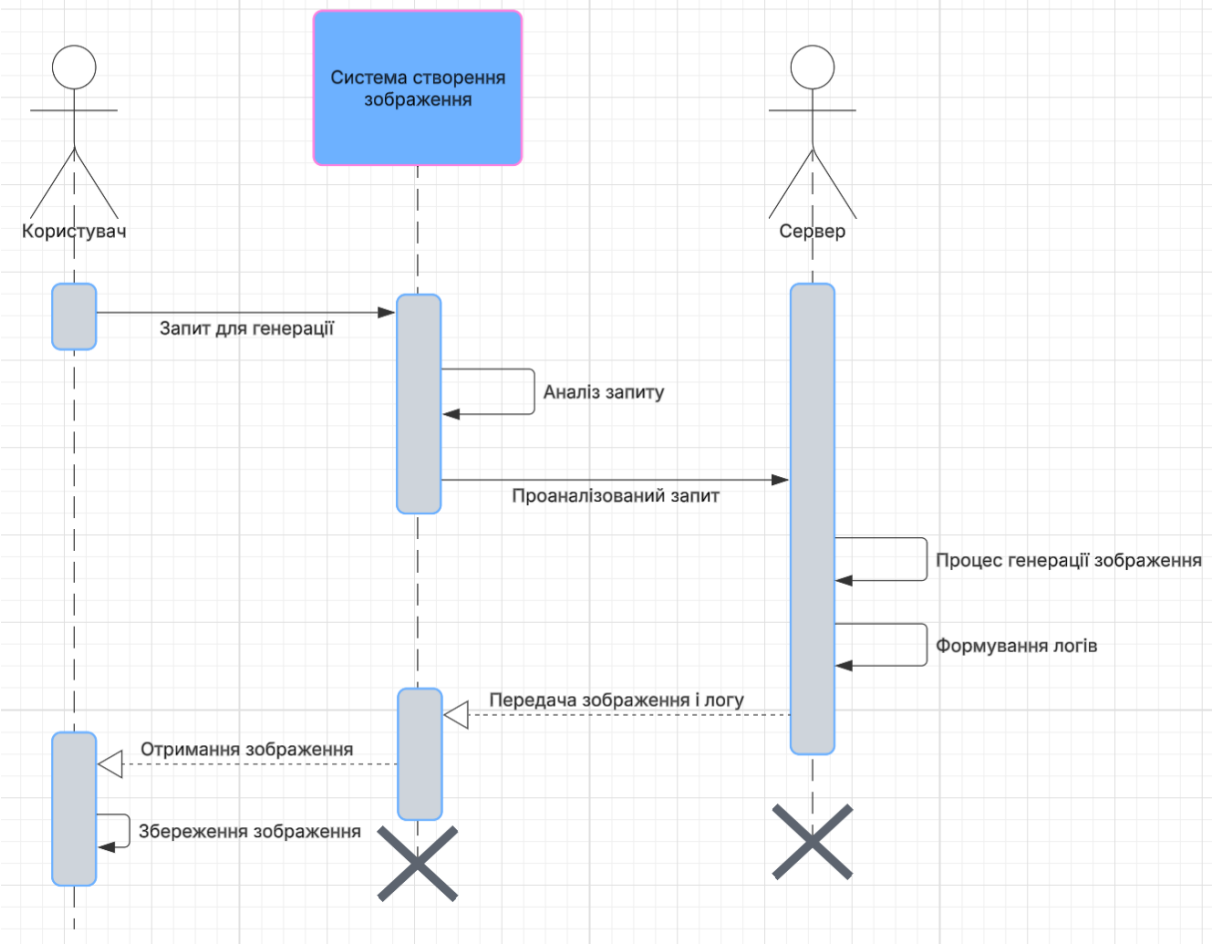

Рисунок 3.10 – Діаграма послідовності дій для прецеденту «Створення зображення»

Ця діаграма наочно показує всю послідовність дій системи, коли користувач натискає кнопку «Створити зображення», а саме:

– користувач ініціює запит через інтерфейс чат-бота, вводячи текстовий опис бажаного зображення, обираючи модель генерації тощо;

– запит передається до системи створення і корекції зображень, яка аналізує його та формує об’єкт запиту;

– далі запит надсилається на сервер системи, де виконується обробка, включно з розпізнаванням типу запиту та запуском моделі генерації зображення;

– сервер викликає відповідні алгоритми чи моделі штучного інтелекту для створення зображення;

– після генерації результат зберігається і посилання або зображення повертається назад до системи;

– система передає відповідь користувачеві;

– у разі помилок також може бути створено об’єкт логу помилки, який фіксується для подальшого аналізу.

4 ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ РІШЕНЬ ПІД ЧАС РОЗРОБКИ СИСТЕМИ СТВОРЕННЯ ТА КОРЕКЦІЇ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ ТЕКСТОВИХ ПІДКАЗОК

4.1 Опис структури розробленої системи створення і корекції зображень в діалозі

Для розробки системи штучного інтелекту для створення і корекції зображення було обрано середовище розробки застосунків IntelliJ IDEA, зокрема, системи штучного інтелекту для генерації та корекції зображень за допомогою текстових підказок. У такій системі IntelliJ використовується як головний інструмент для управління кодом, розгортанням, тестуванням і організацією всієї архітектури.

У межах одного проєкту середовище дозволяє поєднати кілька мов програмування, наприклад, Java або Kotlin для створення API та логіки чат-бота, а також Python для роботи з моделями штучного інтелекту – зокрема, генеративними мережами (GAN, дифузійні моделі), які відповідають за створення або зміну зображень. IntelliJ IDEA підтримує роботу з Python через відповідний плагін.

Мова програмування Python – це мова програмування, яка ідеально підходить для створення систем штучного інтелекту, зокрема, тих, що генерують і коригують зображення в діалозі з користувачем. Вона має простий синтаксис, велику кількість бібліотек для машинного навчання, глибинного навчання, комп'ютерного зору та роботи з зображеннями, що робить її стандартом у сфері ШІ.

Для створення чат-боту в месенджері Telegram буде використовуватися бібліотека Python «python–telegram–bot», яка забезпечує зручний і високорівневий інтерфейс до Telegram Bot API. Вона дозволяє легко створювати бота, що реагує на команди, повідомлення, фото, кнопки та інші дії користувача, а також інтегрувати з логікою штучного інтелекту,

наприклад, для генерації або корекції зображень. Ця бібліотека підтримує як синхронну, так і асинхронну модель програмування, дає змогу працювати з webhook або polling, автоматично обробляє події і має зручну систему фільтрів для розділення логіки залежно від контексту діалогу.

У системах ШІ ця бібліотека виступає як інтерфейс між користувачем і логікою нейромереж. Наприклад, коли користувач пише боту: «Покажи мені портрет у стилі аніме», python–telegram–bot приймає це повідомлення, передає запит у модель зображень, приймає результат і надсилає його назад користувачу.

Telegram – це хмарний месенджер, орієнтований на швидкість, безпеку та розширені функції спілкування. Він підтримує обмін текстовими повідомленнями, медіа, файлами, а також голосові та відеодзвінки. Однією з ключових особливостей Telegram є підтримка ботів – автоматизованих облікових записів, які можуть взаємодіяти з користувачами через простий або інтелектуальний діалог, виконувати команди, обробляти файли, генерувати контент тощо [24].

4.2 Розробка алгоритму роботи системи створення і корекції зображень в діалозі

Карта алгоритму системи (або блок-схема) необхідна для того, щоб візуально та логічно описати порядок дій користувача і поведінку самої системи під час взаємодії. Вона виконує кілька важливих функцій:

– карта допомагає розробникам і дизайнерам чітко розуміти, як саме має працювати система – які етапи проходить користувач, у який момент відбувається генерація або корекція, які є варіанти розвитку сценарію, та де потрібно обробляти помилки або надавати додаткові опції;

– така схема є основою для командної роботи: вона уніфікує бачення процесу між розробниками, тестувальниками, дизайнерами інтерфейсу та

бізнес-аналітиками. Усі учасники команди бачать однакову логіку і можуть спільно ухвалювати рішення щодо покращення UX або функціоналу;

– карта алгоритму – це інструмент перевірки повноти та послідовності. Завдяки їй можна помітити «мертві гілки» (непотрібні або недоступні сценарії), цикли, які не мають виходу, або місця, де користувач може залишитися без результату.

На рисунку 4.1 зображено карту алгоритму системи створення і корекції зображень в діалозі.

Рисунок 4.1 – Карта алгоритму роботи системи створення і корекції зображень в діалозі

Алгоритм роботи системи створення і корекції зображень у діалозі починається з того, що користувач заходить до чату, де йому одразу пропонується зробити вибір між двома напрямками: створення нового зображення або корекція вже наявного. У випадку, якщо користувач обирає генерацію, йому надається можливість обрати одну з доступних генеративних моделей, після чого він формулює запит для створення зображення. Система генерує відповідне зображення на основі запиту і користувач отримує результат. Отримавши зображення, він може або повторити генерацію (змінити запит чи параметри), або перейти до етапу корекції вже згенерованого зображення. Коли результат задовольняє очікування, користувач зберігає зображення.

Якщо ж користувач з самого початку обрав корекцію, то він завантажує своє власне зображення і формує запит для редагування. Після обробки система повертає йому скориговане зображення. Як і у випадку з генерацією, користувач може повторити корекцію або змінити запит, щоб досягти кращого результату. Після збереження задовільного зображення, користувач завершує роботу з чатом.

Таким чином, система забезпечує інтерактивний цикл взаємодії, де користувач у будь-який момент може перейти від генерації до корекції або навпаки, змінювати параметри, повторювати запити, поки не отримає бажаний результат.

4.3 Реалізація програмної логіки для бізнес-процесу системи «Створення зображення»

Реалізація програмної логіки для бізнес-процесу системи «Створення зображення» у рамках чат–бота в Telegram передбачає послідовну обробку запиту користувача, генерацію візуального контенту на основі текстового опису та повернення результату в зручному вигляді. Умовно цей процес

можна описати через декілька логічних етапів, реалізованих у вигляді окремих функцій та взаємодії з зовнішніми модулями штучного інтелекту.

На першому етапі чат-бот приймає текстове повідомлення від користувача, яке розпізнається як запит на генерацію зображення. Вміст повідомлення аналізується на предмет ключових слів, команд або помилок форматування. Якщо запит коректний, він передається в генератор зображень – модель штучного інтелекту, яка підтримує синтез візуального контенту за текстовим описом (наприклад, Stable Diffusion, DALL·E тощо).

Програмна логіка передбачає створення сеансу користувача, де зберігається лише один активний запит. У випадку нового запиту, попереднє зображення автоматично видаляється з пам’яті, а сесія оновлюється. Запит передається в модель з фіксацією параметрів: роздільна здатність, стиль зображення, можливо, додаткові модифікатори (наприклад, «фото реалістичне», «у стилі аніме» тощо).

Після генерації зображення бот отримує зворотній результат у вигляді файлу або URL, який передається користувачеві через API Telegram. У цьому ж кроці може бути реалізовано логіку перевірки – наприклад, контроль розміру файлу або фільтрація неприпустимого вмісту (якщо вбудовано модуль безпеки або модерації).

У разі помилок (наприклад, некоректного запиту, перевищення часу генерації або внутрішньої помилки моделі) бот надсилає користувачеві відповідне повідомлення з поясненням і сесія або перезапускається, або завершується очікуванням нового запиту. Код розробленої функції генерації зображення наведено у лістингу 4.1.

Лістинг 4.1 – Реалізація бізнес-процесу «Створення зображення»

```
# Функція генерації зображення
def generate_image(prompt: str):
    # Токенізація тексту
    text_input = tokenizer(prompt, padding="max_length", 
max_length=77, return_tensors="pt")
```

Продовження лістингу 4.1

```
        input_ids = text_input.input_ids.to(device)
        # Отримання текстових ембеддингів
        text_embeddings = text_encoder(input_ids)[0]
        # Додаємо "порожній" запит для CFG
        uncond_input = tokenizer([""], padding="max_length",
max_length=77, return_tensors="pt")
        uncond_embeddings =
text_encoder(uncond_input.input_ids.to(device))[0]
        # Об’єднуємо ембеддинги
        embeddings = torch.cat([uncond_embeddings,
text_embeddings])
        # Ініціалізуємо шум
        latents = torch.randn((1, unet.in_channels, height //
8, width // 8), device=device)
        latents = latents * scheduler.init_noise_sigma
        scheduler.set_timesteps(num_inference_steps)
        # Diffusion loop
        for t in scheduler.timesteps:
            # Дублюємо latents для умовного/безумовного CFG
            latent_model_input = torch.cat([latents] * 2)
            # Масштабуємо латенти
            latent_model_input =
scheduler.scale_model_input(latent_model_input, timestep=t)
            # Передбачення шуму
            with torch.no_grad():
                noise_pred = unet(latent_model_input, t,
encoder_hidden_states=embeddings).sample
            # Розділяємо CFG
            noise_pred_uncond, noise_pred_text =
noise_pred.chunk(2)
            noise_pred = noise_pred_uncond + guidance_scale *
(noise_pred_text - noise_pred_uncond)
```

Продовження лістингу 4.1

```
            # Крок scheduler'a
            latents = scheduler.step(noise_pred, t,
latents).prev_sample
        # Декодування в зображення
        with torch.no_grad():
            image = vae.decode(latents / 0.18215).sample
        # Перетворення в PIL
        image = (image / 2 + 0.5).clamp(0, 1)
        image = image.cpu().permute(0, 2, 3, 1).numpy()
        image = (image[0] * 255).astype(np.uint8)
        image = Image.fromarray(image)     return image [25]
```

Наведений код програми відображає покрокові дії для успішної генерації зображення, а саме:

– визначення середовища виконання. Спочатку система перевіряє, чи доступний графічний процесор (GPU). Це необхідно, оскільки генерація зображення є обчислювально інтенсивною задачею і виконання на CPU значно повільніше;

– обробка текстового запиту (проведення токенізації). Введений користувачем текст (наприклад, «кіт на скейтборді») передається в токенізатор. Той перетворює речення у послідовність числових представлень – токенів. Ці токени є вхідними даними для мовної моделі CLIP;

– отримання ембеддингів тексту. Токенізований текст подається в текстовий енкодер (CLIP Text Encoder), який формує векторну репрезентацію тексту (ембеддинг). Це вектор, що представляє семантичний зміст запиту;

– побудова безумовних ембеддингів. Для реалізації механізму «classifier–free guidance» одночасно створюється і вектор порожнього тексту. Це дозволяє моделі балансувати між генерацією «з нічого» та генерацією за запитом;

– комбінування ембеддингів. Обидва ембеддинги (з текстом і без) об’єднуються в один тензор. Це дає можливість під час генерації порівнювати, наскільки сильно потрібно слідувати тексту користувача;

– генерація латентного простору. У латентному просторі моделі створюється випадкова матриця шуму. Саме з неї в подальшому формується зображення;

– покрокове очищення шуму. Модель Unet покроково «очищує» шум, кожного разу орієнтуючись на текстовий ембеддинг. На кожному кроці модель прогнозує залишковий шум і видаляє його. Цей процес триває від десятків до сотні ітерацій (залежно від параметрів);

– декодування латентного зображення у візуальну форму. Коли очищення завершене, VAE-декодер перетворює оброблений латентний тензор у зображення у форматі RGB;

– післяобробка і вивід результату. Зображення нормалізується, масштабується і конвертується у формат, придатний до збереження чи відображення. Після цього зображення передається до функції надсилання зображення в чат для користувача.

4.4 Реалізація програмної логіки для бізнес-процесу системи «Корекція зображення»

Бізнес-процес «Корекція зображення» у системі штучного інтелекту передбачає внесення змін до вже наданого користувачем зображення на основі його текстового запиту. Теоретично цей процес включає інтеграцію взаємодії користувача з інтерфейсом, обробку зображення на сервері, застосування алгоритмів штучного інтелекту, а також забезпечення безпечного повернення результату користувачу.

Процес корекції зображення починається з того, що користувач надсилає зображення, яке система приймає та зберігає в оперативній пам’яті сервера. Цей файл стає активним об’єктом для подальших дій і, коли

користувач надсилає повідомлення з текстовим описом змін, то система фіксує цей запит як інструкцію до редагування.

Далі логіка бота передає отримані дані – зображення та текст – до AI-модуля, який використовує сучасні нейромережі для image-to-image трансформацій або image inpainting (наприклад, моделі на базі Stable Diffusion або ControlNet). Модель обробляє зображення, накладаючи зміни згідно з описом, і повертає результат у вигляді нового зображення. Код розробленої функції корекції зображення наведено у лістингу 4.2.

Лістинг 4.2 – Реалізація бізнес-процесу «Корекція зображення»

```
    # Функція корекції зображення
    def imageChange(self, prompt, image, mask_image,
num_inference_steps=50, guidance_scale=7.5):
        # 1. Токенізація тексту
        text_inputs = self.tokenizer(prompt,
padding="max_length",
max_length=self.tokenizer.model_max_length,
return_tensors="pt")
        text_embeddings =
self.text_encoder(text_inputs.input_ids.to(self.device))[0]
        # 2. Підготовка зображення і маски
        init_image =
self.image_processor.preprocess(image).to(self.device)
        mask =
self.mask_processor.preprocess(mask_image).to(self.device)
        # 3. Генерація випадкового шуму
        latents = self.prepare_latents(init_image.shape,
dtype=torch.float16, device=self.device)
        timesteps = self.scheduler.timesteps
        # 4. Прогін дифузійного процесу
        for t in timesteps:
            # Додавання шуму
```

Продовження лістингу 4.2

```
            noisy_latents = self.scheduler.add_noise(latents,
t)
            latent_input = torch.cat([noisy_latents, mask,
init_image], dim=1)
            # Генерація шуму
            noise_pred = self.unet(latent_input, t,
encoder_hidden_states=text_embeddings).sample
            # Крок оновлення latent-змінної
            latents = self.scheduler.step(noise_pred, t,
latents).prev_sample
        # 5. Декодування латентного простору в зображення
        result = self.vae.decode(latents / 0.18215).sample
        image = self.image_processor.postprocess(result)
        return image
```

Наведений код реалізує алгоритм корекції зображення на основі текстового запиту і має такі етапи:

а) обробка текстового запиту користувача. Спочатку система отримує текстову команду (prompt) – наприклад, «зробити небо синім». Цей текст токенізується, тобто розбивається на частини та перетворюється у числові представлення, зрозумілі для моделі. Далі токени подаються в текстовий енкодер, який формує з них векторну репрезентацію – ембеддинг. Цей вектор описує сутність тексту і буде використаний як «інструкція» для нейромережі;

б) підготовка вхідного зображення та маски. Користувач надає зображення, яке потрібно змінити, а також маску – область, яку дозволено змінювати. Обидва ці зображення обробляються і перетворюються в числовий формат, який розуміє модель. Маска – це бінарне зображення, де білі області дозволяють зміну, а чорні зберігають оригінал;

в) ініціалізація латентного простору з випадкового шуму. На цьому етапі система створює випадковий шум того ж розміру, що і латентне

зображення. Це початкова точка для дифузійного процесу. Саме з цього шуму, під керівництвом тексту та маски, буде сформовано нове зображення;

г) основний дифузійний процес – поступове очищення шуму. Для кожного кроку (ітерації) в заздалегідь визначеному графіку часу:

1) до поточних латентів (зображення у вигляді шуму) додається додатковий шум, імітуючи деградацію;

2) цей зашумлений стан об'єднується з маскою і оригінальним зображенням;

3) результат подається в U-Net – нейронну мережу, яка намагається вгадати, який шум був доданий;

4) після цього система виконує оновлення латентного зображення, поступово зменшуючи шум і рухаючись до зображення, яке відповідатиме запиту користувача, але тільки в дозволених маскою областях;

д) побудова остаточного зображення. Коли всі кроки завершені, система має латентне зображення, яке ще не є придатним до перегляду. Воно подається у VAE-декодер – спеціальний модуль, який переводить внутрішнє відображення в звичайне зображення. Потім зображення додатково обробляється, нормалізується та повертається як результат.

4.5 Програмна реалізація логіки спілкування користувача з системою

Програмна реалізація логіки спілкування користувача з системою містить основний інтерфейс системи, який має структуру і набір взаємодій, які забезпечують ефективний та інтуїтивний обмін інформацією між користувачем і системою штучного інтелекту. Код обробки запитів користувачів і відправки зображень в користувачам наведено в лістингу 4.3.

Лістинг 4.3 – Реалізація програмної логіки системи для спілкування користувача з системою

```
    #Реалізація спілкування користувача з системою
    async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
        await update.message.reply_text("Привіт! Я бот для створення і корекції зображень. Надішли мені опис або фото.")
    async def handle_text(update: Update, context: ContextTypes.DEFAULT_TYPE):
        user_id = update.message.from_user.id
        text = update.message.text
        user_sessions[user_id] = {"prompt": text}
        await update.message.reply_text("Запит збережено. Надішли зображення або /generate для створення нового зображення.")
    async def handle_image(update: Update, context: ContextTypes.DEFAULT_TYPE):
        user_id = update.message.from_user.id
        photo = update.message.photo[-1]  # найкраща якість
        file_path = await photo.get_file()
        image_path = f"{user_id}_input.png"
        await file_path.download_to_drive(image_path)
        # Якщо вже є запит – ймовірно, це корекція
        if user_id in user_sessions and "prompt" in user_sessions[user_id]:
            user_sessions[user_id]["image"] = image_path
            await update.message.reply_text("Зображення збережено. Якщо хочеш, надішли маску або /inpaint.")
    async def generate(update: Update, context: ContextTypes.DEFAULT_TYPE):
        user_id = update.message.from_user.id
        prompt = user_sessions.get(user_id, {}).get("prompt", "an astronaut riding a horse")
        await update.message.reply_text("Генерую зображення, зачекай...")
```

Продовження лістингу 4.3

```
        image = pipe_txt2img(prompt).images[0]
        output_path = f"{user_id}_result.png"
        image.save(output_path)
        await
update.message.reply_photo(photo=InputFile(output_path))
    async def inpaint(update: Update, context:
ContextTypes.DEFAULT_TYPE):
        user_id = update.message.from_user.id
        session = user_sessions.get(user_id, {})
        if "prompt" not in session or "image" not in session or
"mask" not in session:
            await update.message.reply_text("Потрібно мати
текст, зображення і маску.")
            return
        await update.message.reply_text("Виконую корекцію...")
        from PIL import Image
        init_image =
Image.open(session["image"]).convert("RGB").resize((1024,
1024))                                  mask_image =
Image.open(session["mask"]).convert("RGB").resize((1024, 1024))
        prompt = session["prompt"]
        image = pipe_inpaint(prompt=prompt, image=init_image,
mask_image=mask_image).images[0]
        output_path = f"{user_id}_corrected.png"
        image.save(output_path)                          await
update.message.reply_photo(photo=InputFile(output_path)) [26]
```

В лістингу наведено код для реалізації спілкування користувача з системою для генерації зображень, який складається з наступних етапів:

а) імпортування бібліотек. Завантажуються всі потрібні модулі: Telegram API, PyTorch, Hugging Face diffusers, transformers та інші;

б) Stable Diffusion для генерації зображення з тексту. Stable Diffusion Inpainting для редагування зображення за допомогою маски;

в) налаштування Telegram-бота. Telegram-бот ініціалізується за токеном. Для обробки команд і повідомлень додаються відповідні хендлери;

г) обробка /start. Коли користувач надсилає команду /start, бот відповідає привітанням та короткою інструкцією;

д) обробка тексту. Якщо користувач надсилає повідомлення з текстом (запитом), бот зберігає цей текст у тимчасовій пам’яті сесії, прив’язаній до користувача. Це буде текстовий запит (prompt) для генерації чи корекції зображення;

е) обробка зображення. Якщо користувач надсилає зображення:

1) воно завантажується локально;

2) якщо до цього був збережений текстовий запит – вважається, що користувач хоче редагувати зображення;

3) зображення зберігається в сесії користувача;

ж) якщо користувач надсилає команду /generate, то тоді система робить наступне:

1) витягує збережений запит;

2) передає його в генеративну модель;

3) згенероване зображення надсилається користувачу;

и) якщо користувач надсилає команду корекції зображення, тоді система робить наступне:

1) перевіряє, чи є всі необхідні компоненти: текстовий запит, зображення та маска;

2) передає все в модель корекції;

3) отримує результат і надсилає його користувачу;

к) зберігання стану сесії. Для кожного користувача створюється окрема «сесія», яка зберігає його запити, зображення і маски. Це дозволяє ботові «пам’ятати» послідовність дій;

л) запуск циклу бота. Бот працює в режимі постійного опитування Telegram-сервера, очікуючи повідомлень від користувачів.

Коли користувач починає роботу бота від своєї особи, то система надсилає в чат привітання разом із кнопками:

– «Створити зображення»;

– «Скоригувати зображення».

Після цього, в залежності від вибору дії система пропонуватиме наступне:

– якщо генерація, тоді запропонує обрати модель генеративного штучного інтелекту із запропонованих і наступним кроком попросить ввести запит для генерації нового зображення;

– якщо корекція, тоді бот попросить скинути зображення і після надсилання зображення попросить ввести запит для редагування зображення.

Після того, як система створить або скорегує зображення, то вона одразу надсилає користувачеві результат своєї роботи і пропонує користувачеві повторно згенерувати або скорегувати зображення, скорегувати створене зображення і зберегти створене/скореговане зображення на диск.

Коли користувач зберігає зображення, то система пропонує йому повторно натиснути кнопки для вибору, що система робитиме, генеруватиме чи корегуватиме зображення, або завершити роботу і продовжити циклічну роботу з чатом.

4.6 Тестування розробленого програмного забезпечення

Для системи створення і корекції зображень було проведено тестування основних бізнес–процесів, які являються основними для роботи всієї системи. В першу чергу було проведено тестування роботи самого бота, чи він запускається. Результат тестування запуску бота зображено на рисунку 4.2.

Рисунок 4.2 – Початок роботи бота

Наступним кроком для тестування є вибір моделі для генерації, написання запиту для генерації зображення і отримання згенерованого зображення. На рисунку 4.3 зображено результат генерації зображення відповідно до запиту.

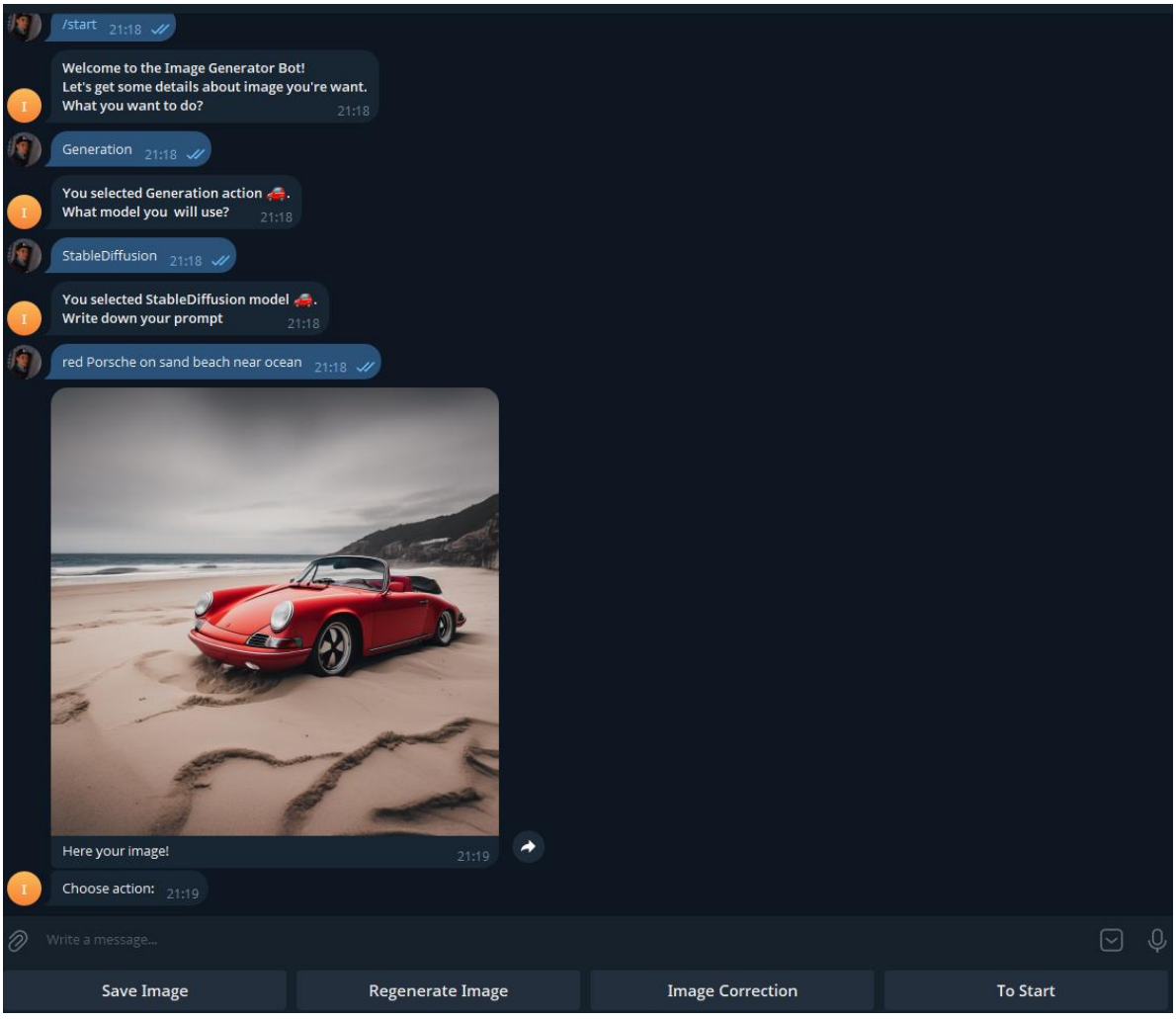

Рисунок 4.3 – Тестування створення зображення

Наведене зображення демонструє, що після створення зображення система пропонує зберегти його на диск, знову створити зображення (система запам’ятала запит користувача і згенерує нове зображення відповідно до запиту користувача), провести корекцію зображення та повернутися на старт роботи бота. За порядком розташування кнопок було проведено тестування збереження зображення. На рисунку 4.4 зображено результат збереження зображення.

Рисунок 4.4 – Тестування збереження створеного зображення

На скриншоті зображено, що система повернула зображення в чат без компресії, тобто без втрати якості. Користувач натискає на меню зображення і зберігає його на диск пристрою. Наступним кроком для проведення тестування повторного створення зображення необхідно створити нове зображення, адже після збереження зображення система повертається на початок життєвого циклу. Тому було створено нове зображення за тим самим запитом. Коли система створила нове зображення, вона пропонує користувачеві свій функціонал, тож після натискання кнопки Regenerate Image система заново створює зображення, відповідно до запиту користувача. Точність системи – приблизно 70 %, тому за одним запитом можна отримати доволі різні зображення. Наприклад, без уточнень старий чи новий автомобіль під час повторної генерації система може перетворити новий автомобіль на старий, і навпаки, старий на новий, або суттєво змінити задній фон зображення. Тому для цієї системи необхідно більш детально

створювати запит, але при цьому не надто навантажений головними об’єктами. Бо тоді система може створити зображення де людина виростає з деталей автомобіля. На рисунку 4.5 зображено тестування повторного створення.

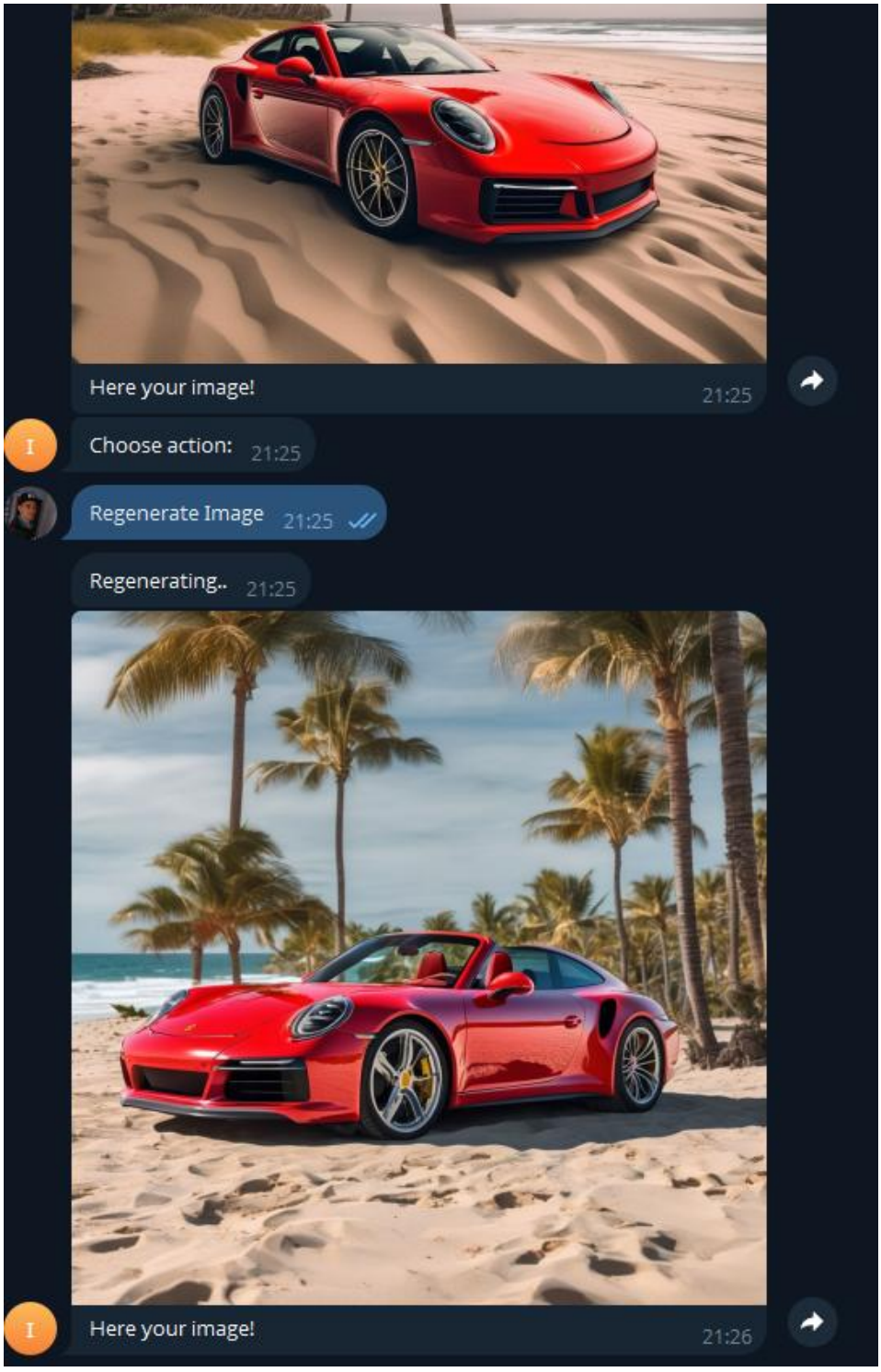

Рисунок 4.5 – Тестування повторного створення зображення

Після повторного створення зображення в системі виникла помилка RunOutTime. Через це було створено нове зображення для проведення тесту для функції корекції щойно створеного зображення. Процес тестування функції корекції для щойно створеного зображення наведено на рисунку 4.6.

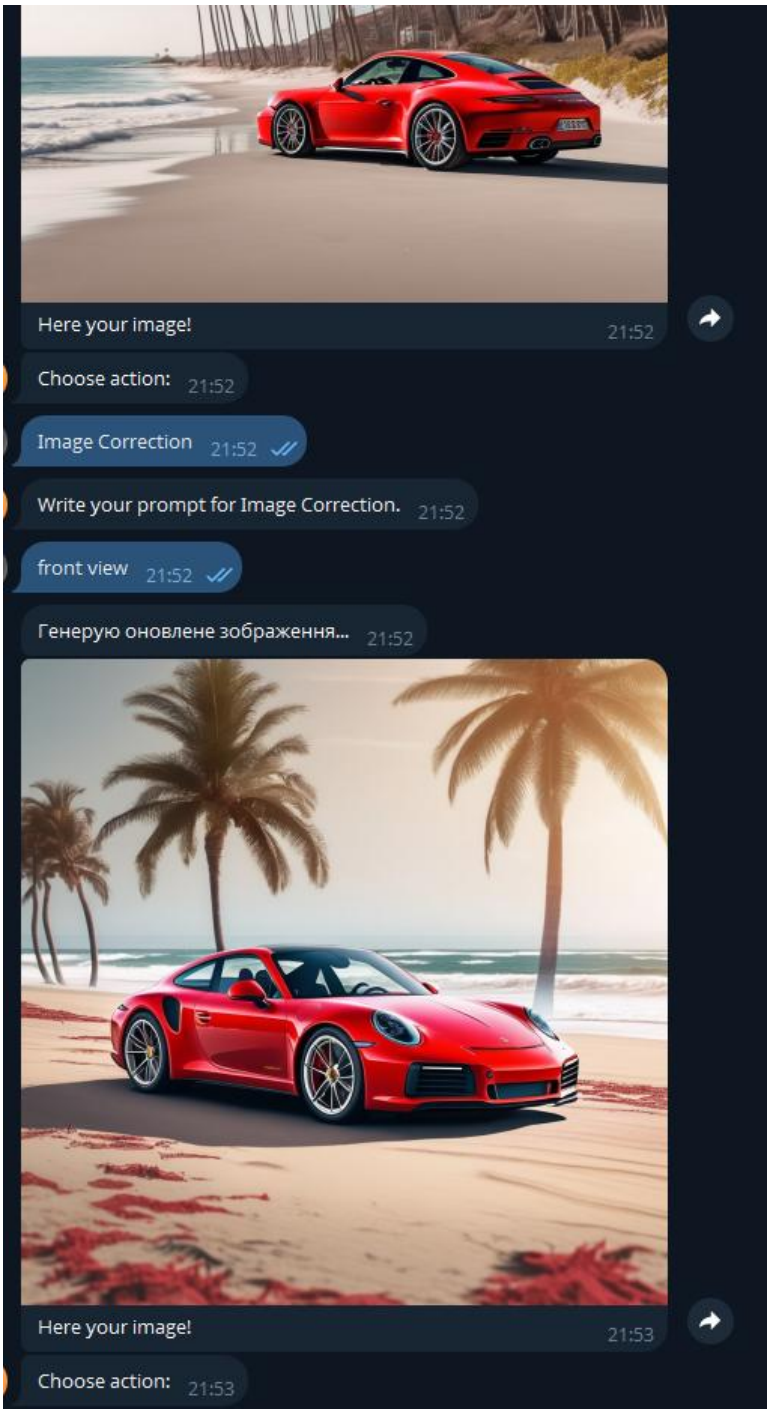

Рисунок 4.6 – Тестування корекції для створеного зображення

Наведений результат свідчить про незначну неточність системи, адже видно, що автомобіль стояв на чистому пляжі, пальми були далеко, а в результаті корекції отримали 3 пальми, але головний об’єкт тепер відображено на передньому плані, як було вказано в запиті.

Наступним кроком було проведено тестування корекції власного зображення. Для цього було використано зображення, яке система створила багато генерацій назад. В запиті не було уточнення, яку модель автомобіля користувач хотів отримати, нову чи стару, але система створила нову модель автомобіля і, відносно запиту, додала пальми на фоні. Результат цього відображено на рисунку 4.7.

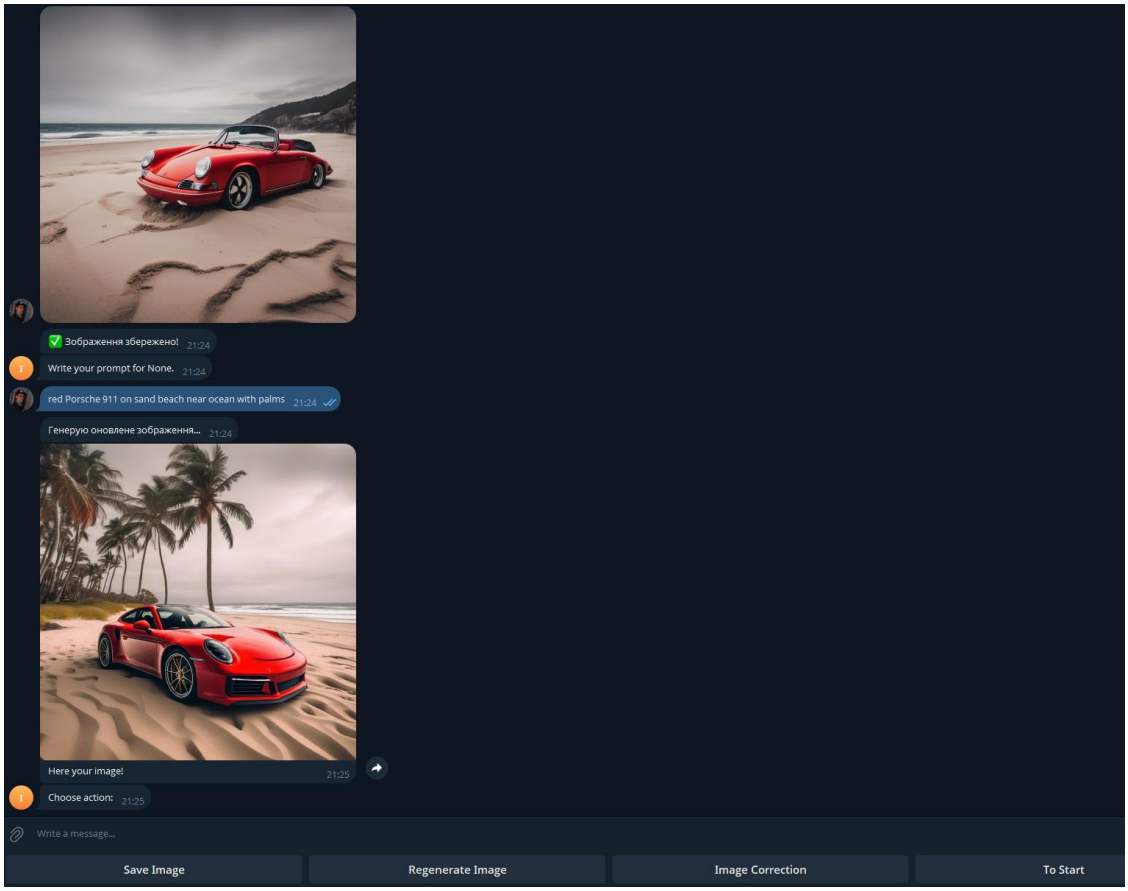

Рисунок 4.7 – Тестування корекції зображення

Через створення нових зображень із шуму неможливо редагувати власні фотографії, адже система навчалася на зображеннях об’єктів і інших

людей, через що, коли система отримує фотографії людей, то система «корегує» зображення і створює зовсім інших людей.

4.7 Захист інформації, що циркулює в системі штучного інтелекту для створення і корекції зображень в діалозі

У системах штучного інтелекту, що взаємодіють із користувачем у форматі діалогу – зокрема, у вигляді чат-бота в Telegram – захист інформації, яка циркулює під час створення та корекції зображень, є критично важливою складовою. Особливо це актуально у випадках, коли система має справу з чутливими або персоналізованими даними користувача, а також з результатами генерації, які можуть містити конфіденційний або приватний зміст.

У спроєктованій системі передбачено тимчасове збереження згенерованих зображень у пам’яті сервера, що здійснюється з метою підтримки активної сесії користувача. Це дозволяє користувачеві протягом однієї взаємодії мати доступ до зображення, створеного в результаті його запиту.

Коли користувач ініціює новий запит на створення або корекцію зображення, система автоматично:

– ідентифікує новий запит як окрему сесію, що замінює попередню;

– видаляє попереднє зображення з оперативної пам’яті, звільняючи ресурси та запобігаючи накопиченню історичних даних;

– не зберігає зображення у довготривалій пам’яті або базі даних, якщо користувач не виконає явної дії збереження (наприклад, завантаження або експорт результату).

Окрім цього, коли система створює лог помилки для майбутнього перегляду адміністратором, то система записує тільки текст запиту користувача для генерації зображення без уточнень, хто є автором запиту.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено всебічний аналіз предметної галузі, що стосується систем генерації та виправлення зображень за допомогою текстових підказок у діалозі.

Проведений аналіз існуючих аналогів продемонстрував, що на ринку вже представлені потужні рішення, такі як Midjourney, Adobe Photoshop (Generative Fill), Runway AI та Microsoft Designer. Кожна з цих систем має свої особливості та області застосування, що варіюються від художньої генерації та створення концепт-арту до автоматизованого редагування зображень та інтеграції з іншими інструментами.

Було визначено основні сфери застосування генеративного штучного інтелекту в графічному дизайні, рекламі, кінематографі, розробці комп'ютерних ігор, архітектурі, моді, освіті та інших галузях. Завдяки можливості швидкого створення візуального контенту та адаптації до конкретних запитів користувача, такі системи значно підвищують ефективність творчих і технічних процесів.

На основі проведеного дослідження сформульовано постановку задачі для системи генерації та виправлення зображень у діалозі, що включає розпізнавання текстових запитів, корекцію зображень у реальному часі та забезпечення високої точності виконання інструкцій.

Після цього було вивчено основні підходи до генерації зображень за допомогою машинного навчання. Ретельно проаналізовано дифузійні моделі, генеративні змагальні мережі, трансформери та автокодери – кожен підхід розглядався з точки зору доцільності використання в діалоговій взаємодії. Виявилося, що дифузійні моделі є найбільш перспективними для створення високоякісних зображень, а трансформери – для розуміння контексту діалогу та забезпечення гнучкості інтерпретації текстових підказок.

На основі проведених досліджень було сформульовано вимоги до системи, спроєктовано її архітектуру, визначено необхідні функціональні можливості, структуру користувацького інтерфейсу та модель обміну даними. Створено діаграми, а саме варіанти використання, діаграма класів і діаграма послідовності, які допомогли сформувати цілісне бачення внутрішньої логіки системи та сценаріїв її використання.

Для розробки системи штучного інтелекту для створення і корекції зображень було обрано дифузійну модель для самої генерації зображень, а для обробки текстових умов користувача використовується трансформер.

Зрештою, було реалізовано основні алгоритми для виконання основних бізнес-процесів, які відповідають за створення та редагування зображень, побудовано механізм обробки діалогу з користувачем. Після цього було проведено тестування для ключових функцій системи і отримано низку згенерованих зображень. Додатково було забезпечено базовий рівень захисту даних користувача і його зображень. Отримана система поєднує в собі інструменти генеративного штучного інтелекту з інтуїтивним інтерфейсом, що робить її доступною для широкого кола користувачів та придатною для подальшого розширення й вдосконалення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про схвалення Концепції розвитку штучного інтелекту в Україні. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/1556-2020-%D1%80 (дата звернення: 31.03.2025).

2. Про авторське право і суміжні права. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/2811–20#Text (дата звернення: 31.03.2025).

3. Левицький К. Ю., Терзіян В. Я. Створення та корекція зображення за допомогою текстових підказок у діалозі: Матеріали 29–го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» Т. 6. Харків: ХНУРЕ. 2025. с. 37–39.

4. Terziyan V. Deep Learning for Cognitive Computing. University of Jyväskylä. 1089 p. URL: https://ai.it.jyu.fi/vagan (date of access: 04.03.2025).

5. Terziyan, V., & Vitko, O. (2023). Causality-Aware Convolutional Neural Networks for Advanced Image Classification and Generation. Procedia Computer Science, 217, 495-506. Elsevier. https://doi.org/10.1016/j.procs.2022.12.245(date of access: 04.03.2025).

6. Official page Midjourney. Midjourney.com. URL: https://www.midjourney.com (date of access: 31.03.2025).

7. Офіційний Adobe Photoshop – програма для редагування фото й розробки дизайну: Creative, marketing and document management solutions. URL: https://www.adobe.com/ua/products/photoshop.html (дата звернення: 31.03.2025).

8. Runway | Tools for human imagination. Runway | Tools for human imagination. URL: https://runwayml.com/ (date of access: 31.03.2025).

9. Microsoft Designer – Stunning designs in a flash. Microsoft Designer – Stunning designs in a flash. URL: https://designer.microsoft.com/ (date of access: 31.03.2025).

10. Дифузійні моделі в ШІ – усе, що вам потрібно знати. Unite.AI. URL: https://www.unite.ai/uk/diffusion-models-in-ai-everything-you-need-to-know/ (дата звернення: 13.05.2025).

11. Дослідження теоретичних основ та застосування дифузійних моделей в ШІ | TheTransmitted. TheTransmitted. URL: https://thetransmitted.com/ai/doslidzhennya-teoretychnyh-osnov-ta-zastosuvannya-dyfuzijnyh-modelej-v-shi/ (дата звернення: 13.05.2025).

12. Що таке генеративний ШІ? – Простий ШІ. Скіммі ШІ. URL: https://skimai.com/uk/що-таке-генеративний-ai/ (дата звернення: 13.05.2025).

13. Як працює генеративний ШІ: переваги та основні сфери застосування | Smodin. Посібник з написання статей – SmodinBlog. URL: https://smodin.io/blog/uk/how-does-generative-ai-work/ (дата звернення: 13.05.2025).

14. Трансформери в ШІ: переваги і можливості – Блог Cpashka. Блог Cpashka – Створіть свою Дропшиппінг Платформу або CPA-сітку. 100 безкоштовних заявок при реэстрації. URL: https://cpashka.biz/blog/transformery-v-shi-perevahy-i-mozhlyvosti/ (дата звернення: 13.05.2025).

15. What Is a Transformer Model? NVIDIA Blog. URL: https://blogs.nvidia.com/blog/what-is-a-transformer-model/ (date of access: 13.05.2025).

16. Weights & Biases. W&B. URL: https://wandb.ai/byyoung3/ml-news/reports/AutoCoder-A-New-Benchmark-in-LLM-Code-Generation--Vmlldzo4MjI5NTYx (date of access: 13.05.2025).

17. Що таке автокодер. Unite.AI. URL: https://www.unite.ai/uk/what-is-an-autoencoder/ (дата звернення: 13.05.2025).

18. Створення схем IDEF0 – Підтримка від Microsoft. Microsoft Support. URL: https://support.microsoft.com/uk-ua/topic/створення-схем-idef0-ea7a9289-96e0-4df8-bb26-a62ea86417fc (дата звернення: 13.05.2025).

19. Головна | Elib LNTU. URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%202%20готовва/page9.html (дата звернення: 13.05.2025).

20. What is a Data Flow Diagram. Lucidchart. URL: https://www.lucidchart.com/pages/data-flow-diagram (date of access: 13.05.2025).

21. UML Use Case Diagram Tutorial. Lucidchart. URL: https://www.lucidchart.com/pages/uml-use-case-diagram (date of access: 13.05.2025).

22. UML Class Diagram Tutorial. Lucidchart. URL: https://www.lucidchart.com/pages/uml-class-diagram (date of access: 13.05.2025).

23. UML Sequence Diagram Tutorial. Lucidchart. URL: https://www.lucidchart.com/pages/uml-sequence-diagram (date of access: 13.05.2025).

24. Bots: An introduction for developers. Telegram APIs. URL: https://core.telegram.org/bots (date of access: 13.05.2025)ю

25. Stability AI. Stability AI. URL: https://stability.ai/ (date of access: 13.05.2025).

26. Generate stunning images with Stable Diffusion XL on the NVIDIA AI Inference Platform | NVIDIA technical blog. NVIDIA technical blog. URL: https://developer.nvidia.com/blog/generate-stunning-images-with-stable-diffusion-xl-on-the-nvidia-ai-inference-platform/ (date of access: 13.05.2025).