

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА БОТА В TELEGRAM З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ
МАШИННОГО НАВЧАННЯ ДЛЯ МОНИТОРИНГУ ЗДОРОВОГО
ХАРЧУВАННЯ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-1

Ступак К.О.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Тітова О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____
(підпис)

Кобилін О. А.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Ступаку Костянтину Олександровичу
(прізвище, ім'я, по батькові)1. Тема роботи Розробка бота в Telegram з використанням технології машинного навчання для моніторингу здорового харчування

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 24 травня 2025 р.

3. Вихідні дані до роботи матеріали конференцій, дані інтернет-мережі, програмні інтерфейси (API) сервісів Google Gemini, FatSecret, RapidAPI (Exercisedb), бібліотеки Python для розробки (Aiogram, asyncpg, aiohttp, Pillow, google-gemai), СУБД PostgreSQL.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих рішень та технологій для моніторингу харчування.

2. Проектування архітектури телеграм-бота та вибір технологій.

3. Розробка моделі даних для зберігання інформації користувача.

4. Інтеграція з зовнішніми API (Google Gemini, FoodSecret, RapidAPI).

5. Програмна реалізація основного функціоналу бота в Telegram.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми, постановка задачі, UML діаграми системи, зображення інтерфейсу користувача та середовищ розробки, ілюстрації результатів роботи бота в Telegram.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів та існуючих рішень	15.04.25-20.04.25	
5	Розробка діаграм	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Тітова О.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 64 с., 2 табл., 23 рис., 30 джерел.

БОТ В TELEGRAM, ЗДОРОВЕ ХАРЧУВАННЯ, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, GOOGLE GEMINI, РОЗПІЗНАВАННЯ СТРАВ, МОНИТОРИНГ РАЦІОНУ, ПІДРАХУНОК КАЛОРІЙ, API, POSTGRESQL, ГЕЙМІФІКАЦІЯ.

Об'єктом роботи є програмні рішення для моніторингу здорового способу життя та харчування.

Метою роботи є розробка бота в Telegram, що використовує машинне навчання для розпізнавання страв та забезпечує моніторинг раціону, споживання води й фізичної активності.

Застосовано технології машинного навчання з інтеграцією Google Gemini 2.0 Flash для аналізу зображень, аудіо та тексту. Впроваджено взаємодію із зовнішніми API для отримання даних про рецепти та вправи. Використано PostgreSQL для збереження користувацької інформації.

У результаті створено бота в Telegram, що надає інструменти для: автоматичного визначення страв та калорійності; ведення журналу харчування; доступу до рецептів з фільтрацією; нагадувань про гідратацію; каталогу фізичних вправ; візуалізації прогресу через систему досягнень.

BOT IN TELEGRAM, HEALTHY EATING, MACHINE LEARNING, NEURAL NETWORKS, GOOGLE GEMINI, FOOD RECOGNITION, DIET MONITORING, CALORIE COUNTING, API, POSTGRESQL, GAMIFICATION.

The object of work is the development process of a software solution – a Telegram bot supporting healthy lifestyle and nutrition habits.

The aim is to develop a Telegram bot that uses machine learning for food recognition and provides monitoring of diet, water intake, and physical activity.

Machine learning technologies were implemented with Google Gemini 2.0 Flash integration for analyzing images, audio, and text. External API interactions were employed to retrieve recipe and exercise data. PostgreSQL was used for storing user information.

As a result, a Telegram bot was created providing tools for: automatic meal identification and calorie assessment; dietary journaling; access to filtered recipes; hydration reminders; exercise catalog; progress visualization through an achievement system.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз технологій та існуючих рішень у сфері підтримки здорового харчування	10
1.1 Огляд сучасних застосунків для моніторингу харчування та підтримки здорового способу життя.....	10
1.1.1 Мобільні застосунки для контролю харчування.....	10
1.1.2 Боти в Telegram для моніторингу здорового харчування.....	12
1.1.3 Системи розпізнавання страв з фотографій	14
1.2 Огляд технологій машинного навчання для аналізу зображень їжі.....	16
1.2.1 Використання нейронних мереж для розпізнавання страв... 16	16
1.2.2 Інтеграція з сервісами Google Gemini 2.0 Flash	18
1.2.3 Використання нейронних мереж для розпізнавання страв... 19	19
1.3 Порівняльний аналіз зручності та функціональності існуючих рішень	21
1.4 Постановка задачі	29
2 Проектування та архітектура бота в Telegram для підтримки здорового харчування	30
2.1 Архітектура системи та вибір технологій розробки	30
2.2 Моделі даних для зберігання інформації про користувача та його харчування	32
2.3 Інтеграція з зовнішніми API для роботи з даними про харчування користувача.....	34
2.3.1 Робота з API FatSecret для отримання рецептів.....	35
2.3.2 Робота з API RapidAPI для отримання інформації про спортивні вправи.....	37
2.4 Модель розпізнавання страв та аналізу зображень	40

	6
2.5 Система добового моніторингу раціону.....	42
3 Програмна реалізація бота в Telegram та тестування системи.....	46
3.1 Архітектура системи та вибір технологій розробки	46
3.2 Програмна реалізація застосунку	49
3.3 Інструкція користувача для використання застосунку	52
3.4 Перспективи проєкта	58
Висновки	61
Перелік джерел посилання	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application Programming Interface (інтерфейс прикладного програмування)

ASR – Automatic Speech Recognition (автоматичне розпізнавання мовлення)

БЖВ – білки, жири, вуглеводи

СУБД – система управління базами даних

IDE – Integrated Development Environment (інтегроване середовище розробки)

CNN – Convolutional Neural Network (згортова нейронна мережа)

FSM – Finite State Machine (скінченний автомат)

HTTP – Hypertext Transfer Protocol (протокол передачі гіпертексту)

JSON – JavaScript Object Notation (об'єктна нотація JavaScript)

JSONB – бінарний формат JSON у PostgreSQL

SQL – Structured Query Language (мова структурованих запитів)

CRUD – Create, Read, Update, Delete (створити, прочитати, оновити, видалити – основні операції з даними)

GIF – Graphics Interchange Format (формат обміну графікою)

MIME-тип – Multipurpose Internet Mail Extensions type (багатоцільове розширення для інтернет-пошти)

ПЗ – програмне забезпечення

ДД.ММ.РРРР – формат представлення дати (день.місяць.рік)

ГГ:ХХ – формат представлення часу (години:хвилини)

ВСТУП

Здорове харчування та активний спосіб життя становлять основу загального добробуту й запобігання численним захворюванням. У сучасних реаліях швидкого ритму життя, легкодоступності обробленої їжі та переважно малорухливої роботи, дотримання засад збалансованого харчування перетворюється на справжнє випробування для багатьох людей. Ведення записів про щоденне харчування, пошук придатних рецептів, обчислення калорійності і відстеження водного балансу організму вимагають чималих старань, часових витрат і внутрішньої дисципліни.

Наявні підходи до підтримання здорового харчування коливаються від класичних паперових журналів до різноманітних застосунків. Проте самостійне внесення даних здебільшого виявляється виснажливим та недостатньо точним процесом, що підриває вмотивованість користувачів. Чимало мобільних програм, попри наявність каталогів продуктів, можуть бути позбавлені функціоналу автоматичного визначення страв, вбудованих порад щодо рецептур та фізичних тренувань, або ж нехтують таким важливим компонентом, як водний баланс. До того ж, брак інтерактивних складових та індивідуалізації може призводити до швидкого згасання зацікавленості у використанні подібних засобів.

Для розв'язання цих проблем перспективним напрямом постає впровадження новітніх технологій, зокрема штучного інтелекту й машинного навчання, інтегрованих у зручні та розповсюджені платформи на кшталт месенджерів. Застосування нейромереж для ідентифікації страв за фотографією, голосом чи текстовим описом здатне істотно полегшити процес введення даних і покращити точність розрахунку калорій. Взаємодія з API для отримання рецептів і вправ, а також запровадження елементів ігрової механіки та персоналізованих сповіщень, можуть значно покращити взаємодію з користувачем і сприяти тривалому дотриманню корисних звичок.

Саме тому створення бота в Telegram, який поєднує функціонал розпізнавання їжі за допомогою машинного навчання (Google Gemini 2.0 Flash), ведення обліку харчування, надання доступу до колекцій рецептів (fatsecret API) та вправ (rapidapi), контролю гідратації та досягнень з елементами гейміфікації, є актуальною задачею. Такий інструмент, реалізований на основі популярного месенджера Telegram із використанням СУБД PostgreSQL для збереження інформації, потенційно може перетворитися на зручного й ефективного помічника для широкого загалу користувачів, які прагнуть контролювати своє харчування та підвищувати якість життя.

1 АНАЛІЗ ТЕХНОЛОГІЙ ТА ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ПІДТРИМКИ ЗДОРОВОГО ХАРЧУВАННЯ

1.1 Огляд сучасних застосунків для моніторингу харчування та підтримки здорового способу життя

Сучасний ринок програмного забезпечення пропонує численні інструменти для підтримки здорового способу життя. Особливе місце займають рішення для контролю харчування, оскільки відстеження раціону є ключовим фактором у досягненні цілей, пов'язаних зі здоров'ям, вагою та фізичною формою. Діапазон цих рішень варіюється від базових лічильників калорій до комплексних платформ, що поєднують моніторинг харчування, фізичну активність, споживання води та аналіз результатів.

1.1.1 Мобільні застосунки для контролю харчування

Мобільні програми сьогодні стали чи не найзручнішим способом контролювати раціон. Оскільки смартфон завжди під рукою, уся інформація про прийоми їжі занотовується миттєво. Типові функції таких застосунків включають:

- ведення харчового щоденника: базова функція для запису спожитої їжі та напоїв;
- підрахунок калорій та макронутрієнтів: на підставі введених даних відбувається автоматичний підрахунок загальної кількості спожитих калорій, білків, жирів і вуглеводів за обраний період;
- встановлення та відстеження цілей: користувачі визначають свої завдання (схуднення, нарощування м'язів, підтримка ваги), а додаток розраховує рекомендовану добову норму калорій та БЖВ;

- моніторинг активності та гідратації: фіксація тренувань та кількості спожитої води з урахуванням у загальному енергетичному балансі;
- аналіз прогресу: візуалізація даних через графіки та діаграми для відстеження динаміки досягнення цілей;
- бази рецептів: деякі платформи містять добірки корисних страв з підрахованими БЖВ і калоріям;
- соціальні функції: можливість додавати друзів, порівнювати результати, брати участь у викликах або спільно обговорювати досягнення.

Серед найвідоміших можна виділити MyFitnessPal з великою базою продуктів та активною спільнотою, Lifesum з привабливим дизайном та персональними планами, YAZIO з інтуїтивним інтерфейсом та рецептами, FatSecret з безкоштовним функціоналом та розвиненими соціальними можливостями.

Переваги таких рішень очевидні: вони значно зручніші за паперові щоденники, спрощують пошук інформації про поживну цінність, автоматизують розрахунки та дають наочну картину прогресу, що підвищує мотивацію й дисципліну.

Проте існують суттєві недоліки:

- трудомісткість ручного введення: необхідність постійно шукати та записувати кожен продукт забирає час. Особливо складно оцінити калорійність домашніх страв або їжі з ресторанів;
- обмеженість баз даних: каталоги можуть містити помилки, застарілу інформацію або не включати специфічні регіональні продукти;
- недоліки сканування штрих-кодів: функція працює лише з упакованими продуктами, не допомагаючи з домашньою їжею чи стравами з закладів;
- неточність розпізнавання їжі: існуючі технології розпізнавання за фото часто працюють некоректно, потребують ідеальних умов зйомки або доступні лише в платних версіях;

- складність інтерфейсу: багатофункціональність може ускладнювати навігацію для нових користувачів;
- обмеження безкоштовного функціоналу: корисні можливості часто доступні лише за підпискою;
- відсутність альтернативних способів введення: більшість застосунків не підтримує голосове введення чи інші інтерактивні форми взаємодії;
- потреба в окремому застосунку: необхідність відкривати спеціальну програму замість використання вже звичної платформи.

Таким чином, незважаючи на популярність мобільних застосунків, їхні обмеження вказують на потребу в розробці інтелектуальніших, автоматизованих та зручно інтегрованих рішень. Підхід, запропонований у цій дипломній роботі – створення бота в Telegram з використанням сучасних моделей машинного навчання для мультимодального розпізнавання їжі та інтеграцією в середовище месенджера – спрямований на подолання зазначених недоліків.

1.1.2 Боти в Telegram для моніторингу здорового харчування

Окрім спеціалізованих мобільних застосунків, альтернативним підходом до моніторингу харчування є використання ботів у месенджері Telegram. Зростаюча популярність Telegram як комунікаційної платформи та його доступний API сприяли появі численних ботів, зокрема тих, що допомагають користувачам контролювати свій раціон та спосіб життя.

Основні характеристики та функції:

- інтерфейс на основі чату: взаємодія з ботом відбувається через надсилання текстових команд або повідомлень у чаті. Бот відповідає текстовими повідомленнями, подекуди використовуючи кнопки (inline та reply keyboards) для зручнішого вибору опцій;

- ручне ведення щоденника: переважно логуювання їжі здійснюється через надсилання боту текстового опису страви та кількості. Деякі боти містять невелику вбудовану базу даних або пропонують користувачу самотійно зазначити калорійність;

- нагадування: боти ефективно реалізують функцію надсилання нагадувань про прийоми їжі, потребу випити води або виконати фізичні вправи безпосередньо в чат користувача;

- простий трекінг: зазвичай боти надають базові звіти про спожиті калорії за день або інший період у формі текстових зведень;

- доступність: не вимагають встановлення окремої програми, доступні на будь-якому пристрої з клієнту Telegram.

Переваги:

- нагадування: боти ефективно реалізують функцію надсилання нагадувань про прийоми їжі, потребу випити води або виконати фізичні вправи безпосередньо в чат користувача;

- простота: багато ботів мають спрощений інтерфейс та обмежений набір функцій, що може бути перевагою для користувачів, які шукають нескладний інструмент для запису даних.

Недоліки та обмежування:

- обмеженість введення даних: більшість наявних ботів спираються виключно на текстове введення, що може бути менш зручним, ніж пошук у базах даних мобільних застосунків, особливо для складних страв;

- практична відсутність розпізнавання їжі: функціонал автоматичного розпізнавання страв за фотографією або аудіо в бота в Telegramx для трекінгу харчування майже не представлений або реалізований на дуже базовому рівні з низькою точністю;

- інтерфейсні обмеження: хоча кнопки та інші елементи інтерфейсу Telegram існують, вони не забезпечують такого рівня візуального представлення даних та інтерактивності, як нативні мобільні додатки.

Боти в Telegram пропонують зручний та інтегрований підхід до трекінгу харчування, проте наявні рішення часто поступаються мобільним застосункам за функціональністю, особливо щодо автоматизації введення даних та аналізу. Це створює нішу для розробки досконаліших ботів, які поєднували б зручність месенджера з потужними технологіями, зокрема машинним навчанням для розпізнавання їжі, що і є метою даної дипломної роботи.

1.1.3 Системи розпізнавання страв з фотографій

Один з найбільш перспективних напрямків у сфері автоматизації відстеження харчування – це створення систем, які можуть автоматично розпізнавати страви та продукти харчування за їхніми зображеннями. Такий підхід може значно спростити ведення щоденника харчування, звільнивши користувача від необхідності ручного пошуку або введення яких-небудь даних.

Системи виявлення страв ґрунтуються на методах комп'ютерного зору та машинного навчання, зокрема на використанні глибоких згорткових нейронних мереж. Цей процес зазвичай включає такі етапи:

- збір та підготовка даних: створення широких та різноманітних колекцій даних (наборів даних) з фотографіями страв із відповідними назвами або навіть інгредієнтами та калорійним вмістом;

- тренування нейронної мережі на підготовленому наборі даних для розпізнавання візуальних ознак різних страв з метою класифікації зображень та визначення типу страви на фото [1];

- виявлення страви: подача нового зображення їжі на вхід натренованої моделі для аналізу та прогнозу найменування страви.

Деякі програми для моніторингу харчування на мобільних пристроях (наприклад CalorieMama AI та SnapCalorie) вже використовують функцію

розпізнавання їжі на основі фотографій [2]. Крім того, існують багато наукових проектів та комерційних API-сервісів, що розвиваються у цьому напрямку. Покращені системи можуть використовуватися не лише для класифікацій страви; вони також намагаються сегментувати зображення для ідентифікацій окремих складників або оцінювати об'єм порції для точного підрахунку калорій.

Навіть при значному прогресі автоматичне розпізнавання їжі залишається складним завданням через низку факторів:

- великий діапазон варіацій: одна й та ж страву може виглядати значно інакше в залежності від способу приготування та подачі; освітлення та ракурс зйомки також впливають на їхню зовнішність;

- складні моменти фотографій включають в себе ситуацію з кількома стравами, напоями та додатковими предметами на зображенні. Страви можуть бути частково спожитими або перемішаними;

- необхідність великих обсягів даних: ефективність розпізнавання значно залежить від кількості та якості навчальних даних, що складаються труднощами у зборі та анотуванні для широкого спектру світових кухонь.

Технології розпізнавання страв з фотографій активно розвиваються і вже мають свою практичну цінність. Однак їхня точність та надійність, особливо щодо оцінки калорійності, ще не завжди відповідають вимогам для повноцінної заміни ручного введення. Використання передових мультимодальних моделей, які можуть аналізувати не лише зображення, але й контекст навколишнього середовища (такий як текстовий опис чи аудіокоментар), як це робить Google Gemini, може допомогти подолати деякі з обмежень та значно покращити процес розпізнавання в практичних умовах [3].

1.2 Огляд технологій машинного навчання для аналізу зображень їжі

Як було зазначено у попередньому підрозділі, автоматичне розпізнавання страв за зображеннями є складною задачею комп'ютерного зору. Ключову роль у вирішенні цієї проблеми відіграють сучасні методи машинного навчання, які дозволяють створювати моделі, здатні виявляти складні закономірності у візуальних даних [4]. Серед різноманіття підходів до аналізу зображень особливе місце займають нейронні мережі.

1.2.1 Використання нейронних мереж для розпізнавання страв

Нейронні мережі, особливо їх різновид – глибокі згорткові нейронні мережі (англ. Convolutional Neural Networks, CNN), показали надзвичайну результативність у розв'язанні різноманітних завдань комп'ютерного бачення, як-от класифікація зображень, виявлення об'єктів та сегментація [5]. Їхня будова виявилась напорчуд придатною для опрацювання візуальних даних, зокрема світлин страв.

Головна перевага CNN полягає в здатності самостійно вивчати ієрархічні представлення характеристик безпосередньо з точок зображення. На відміну від класичних методів, де визначальні риси (текстури, обриси, барви) створювалися фахівцями вручну, CNN власноруч визначають важливі візуальні шаблони на різних рівнях узагальнення. Початкові прошарки мережі зазвичай відгукуються на прості елементи, як-от краї та кути, тоді як глибші прошарки поєднують ці елементи для виявлення складніших структур, фактур та об'єктів, притаманних певним стравам.

Основними складниками архітектури CNN, що забезпечують таку функціональність, є:

- згорткові прошарки: застосовують набір фільтрів до вхідного зображення, виявляючи місцеві просторові шаблони. Використання спільних

ваг для фільтрів по всьому зображенню робить модель нечутливою до зміщення об'єкта та значно зменшує кількість параметрів для навчання;

– прошарки підвибірки: зменшують просторові розміри карт ознак, отриманих від згорткових прошарків. Це допомагає зробити представлення стійкішим до незначних викривлень та зміщень, а також знижує обчислювальну складність. Найчастіше застосовується метод максимальної підвибірки;

– повнозв'язні прошарки: розташовуються зазвичай наприкінці мережі. Вони отримують узагальнені ознаки з попередніх прошарків і виконують остаточну класифікацію, визначаючи ймовірність належності зображення до кожного з відомих класів страв [6].

Процес розпізнавання страви за допомогою навченої CNN відбувається так: зображення подається на вхід мережі, проходить крізь послідовність згорткових та підвибіркових прошарків для видобування ознак, після чого повнозв'язні прошарки видають розподіл імовірностей за класами страв. Клас з найвищою ймовірністю обирається як результат розпізнавання..

Навчання таких глибоких моделей потребує значних обчислювальних ресурсів та великих наборів позначених даних (зображень страв з відповідними мітками). Зважаючи на складність збирання та розмічання таких специфічних наборів даних, поширеною практикою є використання перенесеного навчання. Цей підхід передбачає застосування моделей CNN, попередньо навчених на надзвичайно великих загальних наборах зображень (наприклад, ImageNet), та їх подальше донавчання на меншому цільовому наборі даних зі стравами [7]. Це дає змогу значно пришвидшити процес навчання та досягти високої точності навіть за обмеженої кількості специфічних даних про їжу.

Попри потужність CNN, розпізнавання їжі створює для них особливі виклики: велика внутрішньокласова мінливість (різні вигляди однієї страви) та низька міжкласова відмінність (візуальна подібність різних страв). Це вимагає розробки складних архітектур мереж та ретельного добору даних для

навчання задля досягнення надійного розпізнавання в реальних умовах використання. Подальший розвиток технологій, зокрема застосування багатомодальних підходів, що аналізують не лише зображення, а й іншу доступну інформацію, обіцяє покращення точності та надійності таких систем.

1.2.2 Інтеграція з сервісами Google Gemini 2.0 Flash

Враховуючи необхідність розробки складної системи розпізнавання їжі та обмеженість ресурсів, було вирішено не створювати власну модель з нуля, а натомість інтегрувати бот в Telegram із сучасним зовнішнім сервісом, що базується на передових мультимодальних технологіях [8]. Зокрема, увагу привернули моделі від Google сімейства Gemini, особливо варіант, оптимізований для швидкодії та ефективності, подібний до Gemini Flash.

На відміну від традиційних моделей комп'ютерного зору, які зазвичай спеціалізуються виключно на аналізі візуальної інформації, Google Gemini є представником нового покоління моделей, здатних одночасно обробляти та розуміти дані з різних модальностей – текст, зображення, аудіо та відео. Ця мультимодальність відкриває значні переваги для задачі аналізу їжі:

- універсальність підходу: завдяки можливостям текстової генерації та розуміння контексту, система не лише визначає тип страви, але й може оцінювати вірогідний склад інгредієнтів, пропонувати варіанти приготування або відповідати на конкретні запитання щодо зображеної їжі;

- залучення попередньо навчених потужних систем: взаємодія через прикладний інтерфейс дозволяє скористатися обчислювальною потужністю та напрацюваннями масштабного попереднього навчання продуктів Google, уникаючи потреби самостійно збирати величезні масиви даних чи витратити час на навчання власних рішень. Моделі Gemini тренувалися на набагато

ширшому спектрі прикладів, що забезпечує кращу здатність узагальнювати та розпізнавати різноманітні кулінарні вироби;

– полегшена технічна реалізація: комунікація з моделлю здійснюється через спеціальний програмний інтерфейс. Бот надсилає зображення (а за потреби – супровідний текст чи звуковий файл) на сервер Google, який опрацьовує запит за допомогою відповідної моделі та повертає результат у структурованому форматі. Цей результат згодом аналізується та відображається користувачеві у зручному вигляді через інтерфейс бота в Telegram;

– оперативність роботи: використання спеціально оптимізованих версій на зразок «Flash» націлене на забезпечення швидкого часу відповіді (приблизно 0,5 секунд) та доступної вартості обробки запитів, що надзвичайно важливо для підтримки комфортного користувацького досвіду при інтерактивній взаємодії з ботом.

Поєднання бота з сервісами Google Gemini дає змогу втілити значно потужніший та гнучкіший функціонал виявлення та дослідження страв, ніж це було б досягнуто при застосуванні лише традиційних методик комп'ютерного зору чи створенні власної моделі в рамках проекту. Такий підхід дозволяє зосередитись на розвитку зручності користування та логіки функціонування бота, доручивши нетривіальне завдання аналізу зображень досконалому зовнішньому сервісу.

1.2.3 Використання нейронних мереж для розпізнавання страв

Окрім аналізу зображень, створений бот в Telegram дає змогу визначати страви та отримувати інформацію про них на основі текстових описів або голосових повідомлень користувача. Такий різноплановий підхід розширює можливості взаємодії та стає у пригоді, коли фотографування неможливе чи недоречне. Опрацювання таких типів даних потребує

застосування методів опрацювання природної мови та розпізнавання мовлення [9].

Коли користувач надсилає текстовий опис страви система повинна проаналізувати цей текст для виокремлення важливої інформації. Традиційні підходи охоплюють:

- лексичний аналіз і пошук за ключовими словами: виділення назв продуктів, страв, одиниць вимірювання (грами, штуки) та їх зіставлення з внутрішньою базою даних або передача як пошукового запиту до зовнішніх сервісів. Цей спосіб досить простий у впровадженні, але має обмежену дієвість під час роботи зі складними, неоднозначними чи неструктурованими описами;

- методи опрацювання природної мови: досконаліші техніки, як розпізнавання іменованих сутностей, дають змогу ідентифікувати конкретні продукти, складники, способи приготування та кількісні показники в тексті. Аналіз синтаксичних зав'язків допомагає краще зрозуміти структуру опису. Класифікація намірів дозволяє визначити, що саме хоче користувач – записати страву, дізнатися про калорійність, знайти рецепт тощо.

Сучасні великі мовні моделі, зокрема Google Gemini, демонструють високу здатність «розуміти» природну мову в її різноманітних проявах. Вони ефективно аналізують неструктуровані текстові запити, виокремлюють ключову інформацію, «розуміють» контекст і навіть роблять обґрунтовані припущення щодо відсутніх деталей (наприклад, стандартний розмір порції, якщо його не вказано). Залучення такої моделі дає змогу опрацьовувати текстові запити користувачів зі значно вищою гнучкістю та точністю порівняно з класичними алгоритмами.

Для обробки голосових повідомлень першим і найважливішим кроком є перетворення аудіозапису на текст. Цим займаються системи автоматичного розпізнавання мовлення (Automatic Speech Recognition, ASR), які часто називають просто Speech-to-Text (STT) [10].

Мультимодальні моделі, такі як Gemini, потенційно мають перевагу і тут. Вони можуть бути навчені обробляти аудіо безпосередньо або краще «розуміти» контекст, коли голос супроводжує інші дані, наприклад, зображення.

Додавання функцій обробки тексту та голосу істотно розширює можливості бота порівняно з простим розпізнаванням за фото. Користувачі отримують більше варіантів для взаємодії та внесення даних про їжу, що робить систему загалом зручнішою та практичнішою.

1.3 Порівняльний аналіз зручності та функціональності існуючих рішень

Провівши огляд головних видів інструментів для відстеження харчування – мобільних програм та ботів у Telegram – варто здійснити їх зіставний аналіз за вирішальними показниками зручності користування та доступного функціоналу. Такий підхід допоможе яскравіше окреслити переваги й недоліки наявних рішень. Для дослідження було відібрано чотири широковідомі мобільні застосунки (MyFitnessPal, YAZIO, Lifesum, FatSecret) та два представники ботів в Telegram (Meals.Chat, NutriTrack).

Для об'єктивної оцінки було визначено наступні критерії:

- платформа: мобільний застосунок (iOS/Android) чи бот в Telegram;
- методи введення їжі;
- трекінг води: можливість фіксувати споживання води;
- трекінг активності: можливість фіксувати фізичні вправи;
- база рецептів: наявність вбудованої бази рецептів;
- реєстрація: необхідність створення облікового запису;
- ключові платні функції: наявність важливого функціоналу лише у платній версії;

– ключові платні функції: наявність важливого функціоналу лише у платній версії.

Мобільні застосунки традиційно пропонують найбільш розширений функціонал для моніторингу харчування. Як правило, вони мають великі бази даних продуктів, можливість сканування штрих-кодів, деталізовану статистику та візуалізацію прогресу.

MyFitnessPal займає одне місце одного з лідерів ринку завдяки величезній базі даних продуктів, що наповнюється користувачами, та можливості відстежувати воду й активність. Однак, відсутність вбудованої функції розпізнавання їжі за фото та бази рецептів може бути недоліком для деяких користувачів. Інтерфейс насичений функціями, що може вимагати певного часу на освоєння. Необхідна реєстрація (рис. 1.1).

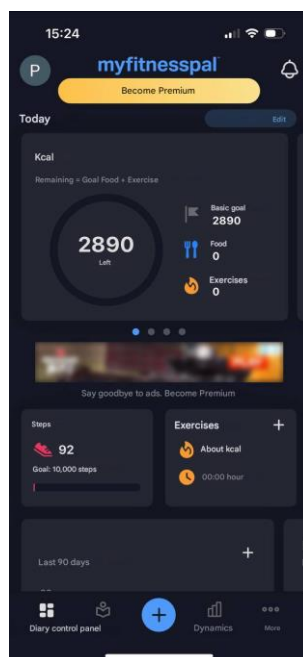


Рисунок 1.1 – Скріншот вигляду UI застосунку MyFitnessPal

YAZIO вирізняється приємним візуальним дизайном та наявністю бази рецептів. Дозволяє відстежувати воду та активність. Як і MyFitnessPal, не має функції розпізнавання за фото. Перевагою є можливість використання базового функціоналу без реєстрації (рис. 1.2).

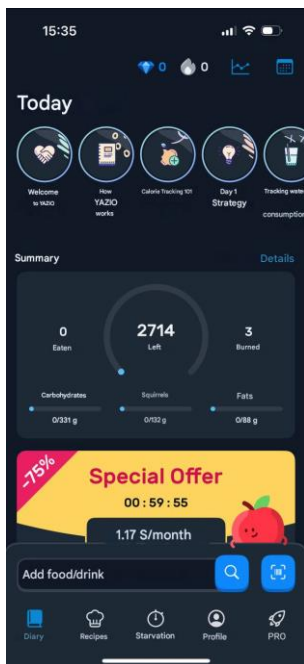


Рисунок 1.2 – Скріншот вигляду UI застосунку YAZIO

Lifesum також пропонує трекінг води та активності, але, як і попередні два, не має розпізнавання за фото. Робить акцент на персоналізованих планах харчування (тільки у платній версії). Вимагає реєстрацію для використання (рис. 1.3).

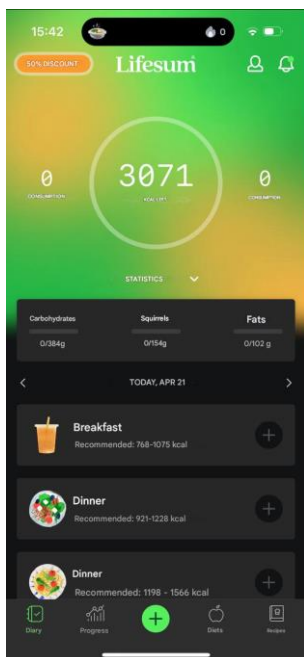


Рисунок 1.3 – Скріншот вигляду UI застосунку Lifesum

FatSecret є єдиним із розглянутих застосунків, що пропонує інноваційні методи введення: розпізнавання їжі за фото та аудіо-описом. Однак, ці функції, а також трекінг води, доступні лише у платній підписці. Додаток має базу рецептів, але не має функціоналу для відстеження фізичних вправ. Можливість використання без реєстрації є плюсом (рис. 1.4).

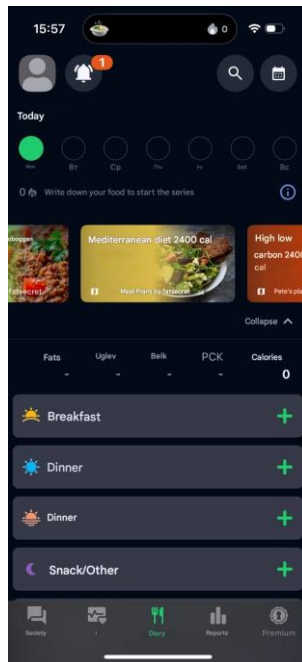


Рисунок 1.4 – Скріншот вигляду UI застосунку FatSecret

Загалом, мобільні застосунки пропонують потужні інструменти аналізу, але зручність введення даних часто обмежена ручним пошуком та скануванням штрих-кодів. Інноваційні методи введення, як-от розпізнавання за фото, є рідкістю або платними. Порівняльна характеристика застосунків показана на таблиці 1.1.

Таблиця 1.1 – Порівняння функціональності мобільних застосунків

Критерій	MyFitnessPal	YAZIO	Lifesum	FatSecret
1	2	3	4	5
Платформа	Моб. застосунок	Моб. застосунок	Моб. застосунок	Моб. застосунок

Продовження таблиці 1.1

1	2	3	4	5
Ручний пошук/введення	Так	Так	Так	Так
Сканування штрих-кодів	Так	Так	Так	Так
Розпізнавання за фото	Ні	Ні	Ні	Так (Платно)
Розпізнавання за аудіо	Ні	Ні	Ні	Так (Платно)
Розпізнавання за текстом	Ні	Ні	Ні	Ні
Трекінг води	Так	Так	Так	Так (Платно)
Трекінг активності	Так	Так	Так	Ні
База рецептів	Ні	Так	Так (Платно)	Так
Обов'язкова реєстрація	Так	Ні	Так	Ні
Інтерфейс	Насичений	Сучасний	Сучасний	Простий

Боти в Telegram пропонують альтернативний підхід, інтегруючи функціонал у звичний інтерфейс месенджера. Це може підвищити зручність та швидкість доступу.

Meals.Chat – бот, що реалізує функцію розпізнавання їжі за фото, що є його основною перевагою. Однак, він суттєво обмежений в іншому функціоналі: відсутні можливості трекінгу води, фізичної активності та бази рецептів. Взаємодія відбувається через надсилання фото та отримання відповіді (рис. 1.5).



Рисунок 1.5 – Скріншот вигляду діалогу з Meals.Chat

NutriTrack фокусується на введенні даних через текстовий опис або аудіоповідомлення, що може бути зручно в певних ситуаціях. Проте, йому бракує розпізнавання за фото та функцій відстеження води, активності чи рецептів. Інтерфейс максимально спрощений і базується на текстових командах (рис. 1.6).

Загалом, існуючі боти в Telegram виграють у зручності доступу та потенційно в альтернативних способах введення (текст/аудіо/фото в окремих випадках), але значно поступаються мобільним застосункам у комплексності функціоналу, аналітичних можливостях та візуалізації даних. Порівняльна характеристика ботів в Telegram показана на таблиці 1.2.

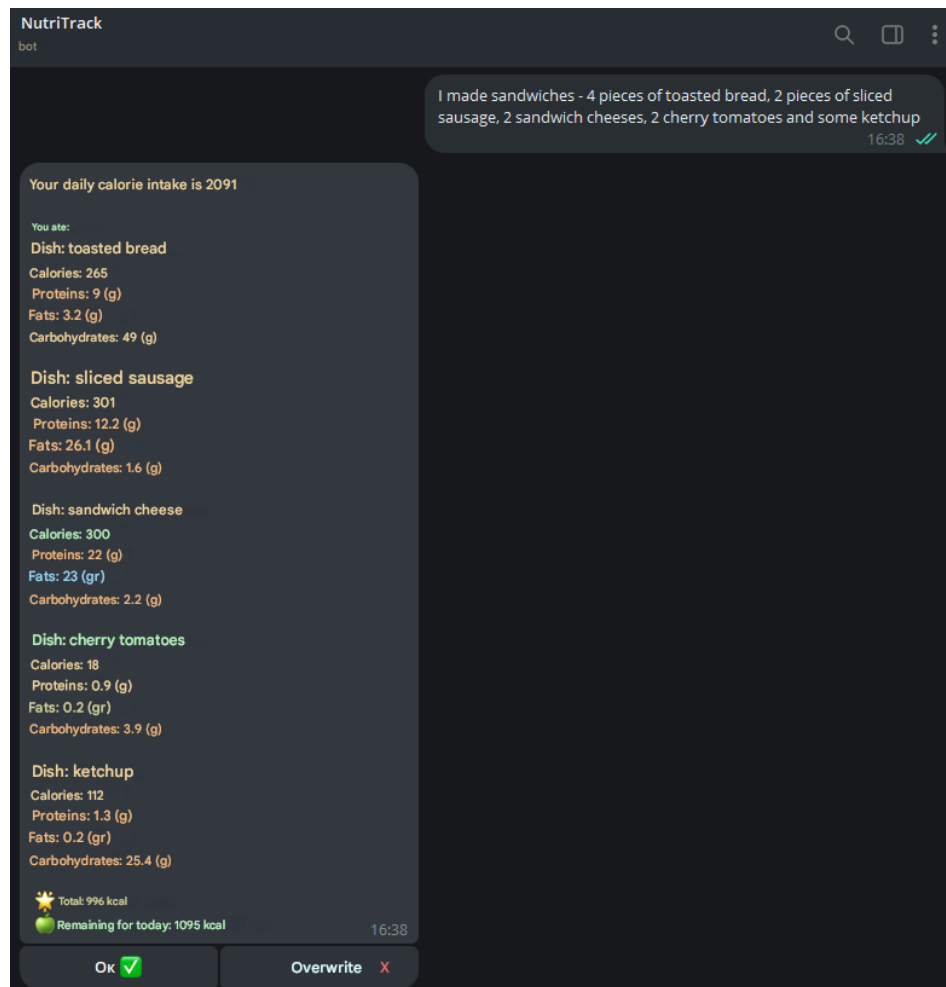


Рисунок 1.6 – Скріншот вигляду діалогу з NutriTrack

Таблиця 1.2 – Порівняння функціональності ботів в Telegram

Критерій	MyFitnessPal	YAZIO
1	2	3
Платформа	Бот в Telegram	Бот в Telegram
Ручний пошук/введення	Ні (через фото)	Так (текст)
Сканування штрих-кодів	Ні	Ні
Розпізнавання за фото	Так	Ні
Розпізнавання за аудіо	Ні	Так

Продовження таблиці 1.2

1	2	3
Розпізнавання за текстом	Ні	Так
Трекінг води	Ні	Ні
Трекінг активності	Ні	Ні
База рецептів	Ні	Ні
Реєстрація	Не потрібна	Не потрібна
Інтерфейс	Чат (фото)	Чат (текст/аудіо)

Проведений аналіз показує існування певного розриву між функціональною повнотою мобільних застосунків та зручністю/інтегрованістю ботів в Telegram. Користувач часто змушений обирати: або багатофункціональний, але потенційно громіздкий застосунок із переважно ручним введенням даних, або простий та зручний бот із суттєво обмеженими можливостями.

Жодне з розглянутих рішень не поєднує в собі одночасно:

- зручність інтерфейсу бота в Telegram;
- потужні та доступні (не лише платні) методи розпізнавання їжі (фото, аудіо, текст);
- комплексний трекінг (їжа, вода, активність);
- додаткові корисні функції (рецепти, гейміфікація).

Це вказує на наявність ніші для розробки бота в Telegram, який би використовував сучасні технології машинного навчання (як інтеграція з Google Gemini) для реалізації гнучкого та точного розпізнавання страв різними способами, поєднуючи це з іншими важливими функціями підтримки здорового харчування та способу життя безпосередньо в інтерфейсі месенджера.

1.4 Постановка задачі

Таким чином, проведений аналіз показав, що існуючі рішення для підтримки здорового харчування мають суттєві обмеження. Мобільні застосунки, хоч і бувають багатofункціональними, часто змушують користувача ретельно вводити всі дані вручну, а зручне розпізнавання їжі за фото зустрічається не так часто. З іншого боку, боти в Telegram є значно доступнішими, проте зазвичай програють у функціональності та інтелектуальних можливостях. Виникає очевидна потреба в інструменті, який би поєднав зручність месенджера з потужністю сучасних технологій аналізу даних. Тому ставиться завдання створити бота в Telegram, що міг би розпізнавати страви різними способами (через фото, аудіо чи текст) за допомогою машинного навчання та водночас пропонував би комплексний моніторинг раціону й активності у зрозумілому інтерактивному форматі.

Об'єктом роботи є програмні рішення для моніторингу здорового способу життя та харчування.

Метою роботи є розробка бота в Telegram, що використовує машинне навчання для розпізнавання страв та забезпечує моніторинг раціону, споживання води й фізичної активності.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати існуючі мобільні застосунки та ботів в Telegram для моніторингу харчування, щоб зрозуміти їхні сильні та слабкі сторони;
- дослідити можливості машинного навчання для аналізу зображень, текстів та аудіо в контексті розпізнавання їжі, приділивши особливу увагу моделям Google Gemini;
- розробити архітектуру майбутнього бота в Telegram, продумати взаємодію його частин, структуру бази даних (на PostgreSQL) та способи інтеграції із зовнішніми API (Google Gemini, fatsecret, rapidapi);
- розробити та реалізувати інтерфейс користувача в середовищі Telegram, забезпечивши інтуїтивно зрозумілу взаємодію.

2 ПРОЄКТУВАННЯ ТА АРХІТЕКТУРА БОТА В TELEGRAM ДЛЯ ПІДТРИМКИ ЗДОРОВОГО ХАРЧУВАННЯ

2.1 Архітектура системи та вибір технологій розробки

На основі аналізу, проведеного у попередніх розділах, для реалізації системи підтримки здорового харчування було обрано платформу бота в Telegram. Цей вибір дозволяє інтегрувати функціонал у звичне для користувача середовище месенджера, забезпечуючи зручний доступ та взаємодію [11]. Подальше проєктування системи було зосереджено на побудові ефективною та гнучкою архітектури для реалізації поставлених завдань у середовищі Telegram.

Загальна структура системи побудована за принципом клієнт-сервер (рис. 2.1). Клієнтську частину представляє програма Telegram користувача, а серверну – програмна логіка бота. Структура містить інтерфейс взаємодії в Telegram, серверну частину бота, сховище даних та зовнішні служби (API). Ключовим структурним принципом є асинхронність, що дає змогу ефективно опрацьовувати численні одночасні запити та не блокувати роботу під час очікування відповідей від зовнішніх служб чи бази даних [12]. Це особливо важливо для забезпечення швидкої реакції бота на дії користувача. Код впорядковано за модульним принципом для кращої структурованості та підтримки. Серверна частина опрацьовує вхідні запити, керує логікою діалогів та станами користувачів, взаємодіє з базою даних та зовнішніми API. База даних забезпечує тривале збереження відомостей. Зовнішні служби використовуються для розширення функціональності: Google Gemini API для визначення страв, Fatsecret API для доступу до рецептів і RapidAPI (Exercisedb) для роботи з базою вправ.

Вибір основних технологій зумовлений вимогами проєкту. Мовою програмування обрано Python завдяки його набору бібліотек для веб-розробки, роботи з API, машинного навчання, асинхронного програмування

(`asyncio`) та зрозумілому синтаксису. Для взаємодії з Telegram використано фреймворк `Aioogram` – сучасний асинхронний інструмент, що надає зручні засоби обробки оновлень, керування станами (FSM) та створення інтерактивних клавіатур [13]. Як систему управління базами даних обрано PostgreSQL через її надійність, відповідність стандартам ACID та ефективну підтримку типу даних `JSONB` для гнучкого збереження параметрів [14]. Взаємодія з базою даних здійснюється асинхронно через бібліотеку `asynccrg`. Реляційну модель обрано для гарантування цілісності даних. Поєднання з Google Gemini дає змогу використовувати передові багатофункціональні можливості для визначення їжі. Бібліотека `Pillow` застосовується для створення зображень зі статистикою, покращуючи візуальне подання даних.

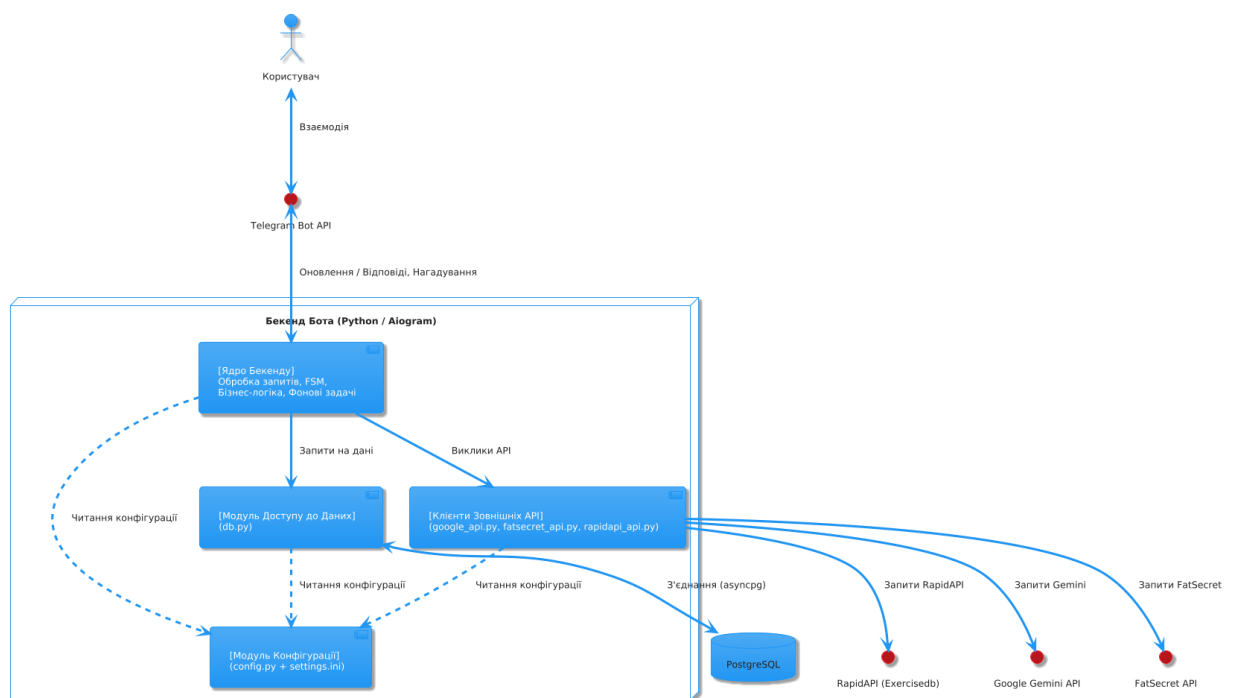


Рисунок 2.1 – UML діаграма архітектури бота в Telegram

Така архітектура на основі асинхронного Python-сервера, що взаємодіє з Telegram через `Aioogram`, зберігає дані в PostgreSQL та інтегрується з потужними зовнішніми API, дозволяє створити функціональний, інтерактивний та зручний бот в Telegram для ефективної підтримки здорового харчування та способу життя користувачів.

2.2 Моделі даних для зберігання інформації про користувача та його харчування

Ефективне зберігання та управління даними становить ключовий елемент розробленого бота. Важливо забезпечити надійний захист інформації про користувачів, їхні особисті параметри та прагнення, записи харчування, обліку спожитої води, фізичних навантажень та індивідуальних налаштувань. З цією метою було розроблено модель даних, яка враховує особливості збереженої інформації та створює умови для її швидкого пошуку й оновлення.

Як зазначалося раніше, для реалізації реляційної бази даних був обраний PostgreSQL. Ця система гарантує цілісність інформації завдяки механізмам основних та зв'язаних ключів, має підтримку транзакцій та пропонує гнучкі можливості для формування запитів через SQL. Суттєвою перевагою PostgreSQL у рамках цієї роботи є наявність типу даних JSONB, який дозволяє зберігати напівструктуровану інформацію, зокрема параметри та цілі користувача, без потреби попередньо визначати сувору структуру таблиці для всіх можливих атрибутів.

На підставі аналізу вимог до збереження даних було створено таку структуру бази даних `nutrition_bot` у PostgreSQL, що складається з декількох пов'язаних таблиць.

Таблиця `users`: Головна таблиця для обліку зареєстрованих користувачів бота:

- `user_id` (BIGINT, PRIMARY KEY): Унікальний ідентифікатор користувача в Telegram, використовується як головний ключ;

- `reg_date` (TEXT): Дата створення облікового запису користувача (записується у форматі ДД.ММ.РРРР);

- `user_params` (JSONB): Об'єкт JSON для збереження фізичних даних користувача (стать, вік, зріст, вага, рівень фізичної активності). Застосування

JSONB дає змогу легко додавати чи змінювати параметри без перебудови структури таблиці;

- `user_goals` (JSONB): Об'єкт JSON для збереження прагнень користувача (бажана вага, денна норма калорій, потреба у воді та активностях). Також забезпечує гнучкість структур;

- `last_achievement_id` (INTEGER, NULLABLE): Ідентифікатор останнього отриманого користувачем досягнення. Може бути зовнішнім ключем, що посилається на таблицю `achievements`.

Таблиця `users_progress`: Призначена для щоденного обліку основних показників досягнень користувача:

- `user_id` (BIGINT): Код користувача (зовнішній ключ, пов'язаний з `users.user_id`);

- `date` (TEXT): День, за який фіксуються досягнення (у форматі ДД.ММ.РРРР);

- `calories` (INTEGER): Загальна кількість спожитих калорій за добу;

- `water` (INTEGER): Загальний обсяг випитої води (в мл) за добу;

- `exercises` (INTEGER): Число виконаних вправ або інший індикатор активності за добу.

Таблиця `users_food`: Докладний журнал спожитої користувачем їжі:

- `user_id` (BIGINT): Код користувача (зовнішній ключ, пов'язаний з `users.user_id`);

- `food_name` (TEXT): Назва спожитої страви чи продукту;

- `food_calories` (INTEGER): Енергетична цінність спожитої страви/продукту;

- `add_time` (TEXT): Точний момент додавання запису (у форматі ДД.ММ.РРРР ГГ:ХХ).

Таблиця `users_hydration`: Містить налаштування користувача щодо нагадувань про вживання води:

- `user_id` (BIGINT, PRIMARY KEY): Унікальний ідентифікатор користувача в Telegram, використовується як головний ключ;

- wakeup_time (TEXT): Час підйому користувача (використовується для визначення періоду активності);
- sleep_time (TEXT): Час, коли користувач лягає спати;
- reminder_frequency (INTEGER): Частота нагадувань (у хвиликах або годинах);
- status (BOOLEAN): Стан активації нагадувань (ввімкнено/вимкнено).

Таблиця achievements: Довідникова таблиця, що містить інформацію про всі можливі досягнення в системі:

- achievement_id (INTEGER, PRIMARY KEY): Унікальний ідентифікатор досягнення;
- name (TEXT): Назва досягнення (наприклад, «Майстер гідратації», «Тижневий марафон»);
- description (TEXT): Опис досягнення.

Ця структура забезпечує збереження всіх необхідних відомостей для роботи бота. Використання asynсrg для взаємодії з цією базою даних дає змогу виконувати запити асинхронно, не зупиняючи роботу бота під час операцій з базою даних. Завдяки реляційній моделі та застосуванню JSONB досягається рівновага між структурованістю, цілісністю даних та гнучкістю для майбутніх доповнень.

2.3 Інтеграція з зовнішніми API для роботи з даними про харчування користувача

Функціональність розробленого бота в Telegram значно розширюється завдяки інтеграції з зовнішніми програмними інтерфейсами (API). Залучення сторонніх сервісів дає змогу працювати з великими масивами спеціалізованої інформації – рецептами здорових страв, даними про поживну цінність продуктів та комплексами фізичних вправ – без необхідності самотужки збирати й підтримувати такі бази. Цей підхід став звичною практикою при

створенні сучасних програмних рішень, адже дозволяє суттєво збагатити власні розробки завдяки можливостям зовнішніх платформ. У даній роботі ключову роль відіграють API для пошуку рецептів, визначення характеристик страв та добору фізичних вправ.

2.3.1 Робота з API FatSecret для отримання рецептів

Задля забезпечення користувачів доступом до каталогу рецептів здорового харчування було обрано API сервісу FatSecret [15]. Цей ресурс надає доступ до обширної колекції кулінарних рецептів різноманітних кухонь світу, включно з докладними відомостями про складники, послідовність приготування та обчислену поживну цінність (калорії, білки, жири, вуглеводи). Надання користувачам змоги знаходити рецепти відповідно до їхніх дієтичних потреб (наприклад, за призначенням – сніданок, обід, вечеря) є важливим елементом сприяння здоровому харчуванню.

Доступ до API FoodSecret вимагає автентифікації за протоколом OAuth 2.0 із застосуванням методу надання доступу «client credentials». Серверна частина бота використовує унікальний ідентифікатор клієнта (Client ID) та приватний ключ клієнта (Client Secret) для отримання тимчасового маркера доступу (Bearer Token) від сервера автентифікації FoodSecret (<https://oauth.fatsecret.com/connect/token>). Отриманий маркер додається до заголовка Authorization кожного наступного запиту до ресурсів API.

Пошук рецептів (/rest/recipes/search/v3) дає змогу здійснювати пошук рецептів за різними ознаками. Бот використовує параметри запиту для відбору результатів згідно з побажаннями користувача, такі як:

- recipe_types: фільтрування за призначенням страви (приміром, «Breakfast», «Lunch», «Dinner», «Snack»);

- `calories_from`, `calories_to`: фільтрування за діапазоном калорійності порції;
- `search_expression`: пошук за ключовими словами в назві або описі рецепту;
- `page_number`, `max_results`: параметри для поділу результатів пошуку на сторінки, що дозволяє отримувати рецепти частинами;
- `must_have_images`: фільтр для відбору лише рецептів із зображеннями.

Отримання подробиць рецепту (`/rest/recipe/v2`) здійснюється наступним чином: після того, як користувач вибирає рецепт зі списку знайденого, бот використовує номер цього рецепту (`recipe_id`) для запиту докладної інформації. Відповідь містить повний опис страви, перелік інгредієнтів із кількістю, покрокові настанови з приготування та вичерпні дані про харчову цінність.

Робота з API FoodSecret реалізована асинхронно за допомогою бібліотеки `aiohhttp` (рис. 2.2). Це дозволяє боту надсилати запити до API й очікувати на відповідь, не зупиняючи опрацювання інших запитів від користувачів. Отримані дані у форматі JSON опрацьовуються та подаються користувачеві у зручному вигляді в інтерфейсі Telegram, здебільшого з використанням кнопок для навігації та вибору. Поєднання з FoodSecret API значно збагачує функціональність бота, пропонуючи користувачам практичні засоби для планування свого раціону.

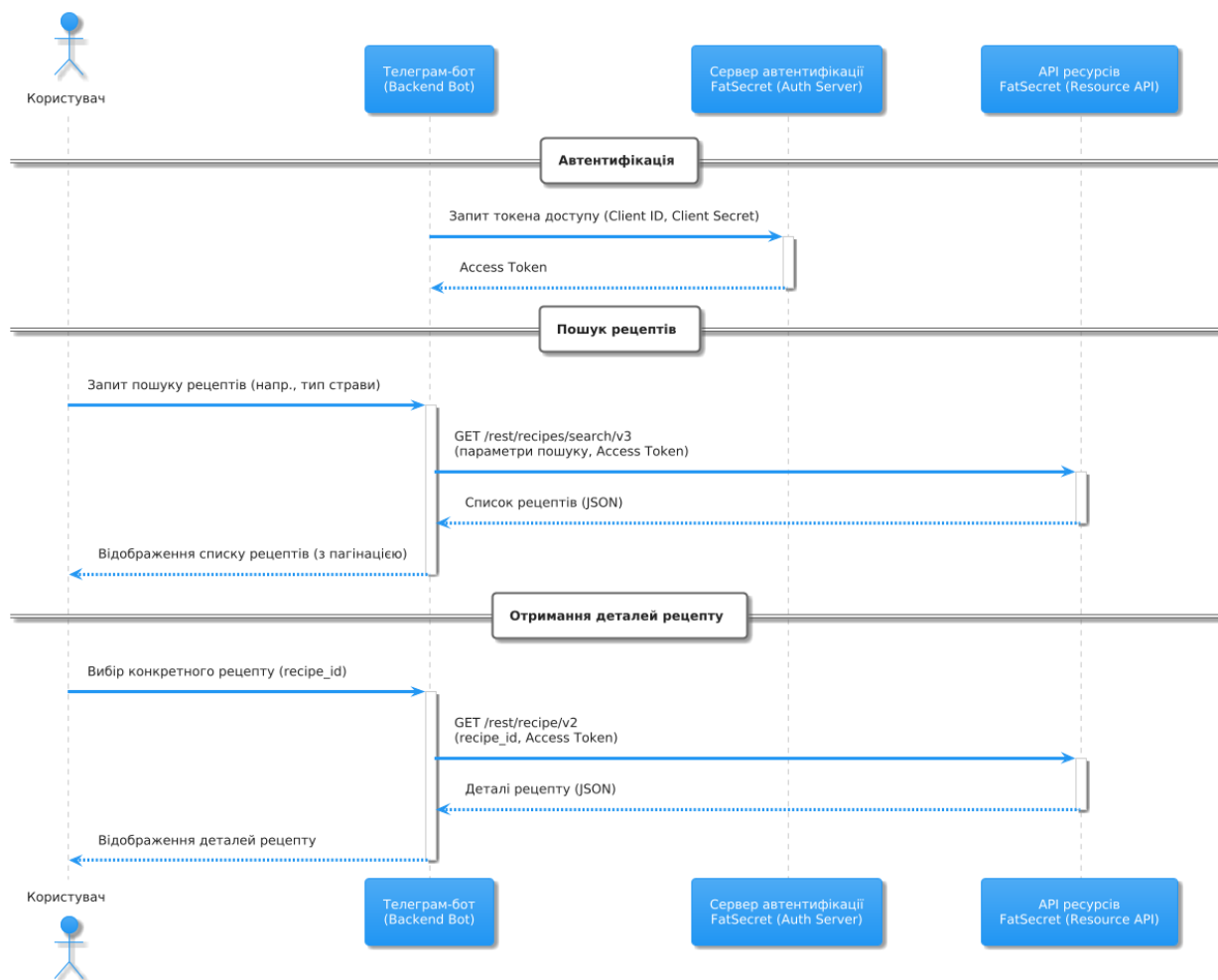


Рисунок 2.2 – UML діаграма взаємодії з API FatSecret

2.3.2 Робота з API RapidAPI для отримання інформації про спортивні вправи

Важливим компонентом здорового способу життя, окрім збалансованого харчування, є регулярна фізична активність. Для створення комплексного рішення, що підтримує користувачів у всіх аспектах здорового способу життя, було вирішено додати функціонал доступу до каталогу фізичних вправ. Найдоцільнішим рішенням стало використання вже існуючої бази даних через зовнішній сервіс, замість створення власної.

Після дослідження доступних варіантів вибір припав на платформу RapidAPI – агрегатор численних програмних інтерфейсів різноманітного призначення. Серед запропонованих на цій платформі ресурсів особливу

увагу привернув ExerciseDB API, який містить розширену колекцію (понад 1300 найменувань) фізичних вправ з докладними візуальними поясненнями. Кожна вправа супроводжується анімацією у форматі GIF, має опис задіяних м'язових груп, необхідного спортивного інвентарю та детальних інструкцій з виконання [16].

На відміну від раніше розглянутого FoodSecret API, взаємодія з сервісами через RapidAPI потребує дещо іншого підходу до автентифікації. При формуванні кожного запиту необхідно додавати спеціальні HTTP-заголовки:

- X-RapidAPI-Key: персональний ключ користувача платформи, що підтверджує повноваження на доступ;
- X-RapidAPI-Host: доменне ім'я конкретного сервісу (у нашому випадку це `exercisedb.p.rapidapi.com`).

Ці заголовки додаються автоматично до кожного звернення бота в Telegram до ExerciseDB. Зберігання та управління ключем реалізовано через єдиний модуль конфігурації системи, аналогічно іншим зовнішнім сервісам.

Основний функціонал, який використовується з ExerciseDB API:

- отримання переліку вправ (`/exercises`) – метод, який дозволяє завантажити список доступних вправ з можливістю посторінкового перегляду. Розроблений бот в Telegram використовує цей функціонал для надання користувачам узагальненого каталогу вправ;
- вибірка за цільовою зоною (`/exercises/bodyPart/{bodyPart}`) – сервіс, який надає можливість відфільтрувати вправи за конкретною групою м'язів чи частиною тіла (наприклад, спина, кардіо, груди, ноги тощо). Ця функція реалізована в боті через інтерактивні кнопки, що дозволяють користувачам швидко знаходити вправи для потрібної їм зони.

Технічна реалізація зв'язку з RapidAPI побудована на принципах асинхронної взаємодії через бібліотеку `aiohhttp`, що забезпечує ефективну обробку запитів без блокування основного потоку виконання. Код у файлі

rapidapi_api.py формує необхідний URL-запит з відповідними параметрами, додає потрібні заголовки автентифікації та надсилає GET-запити до API.

Отримані від сервісу дані у форматі JSON містять структуровану інформацію про кожну вправу – назву, детальний опис, задіяні м’язи, потрібне обладнання та посилання на анімаційну демонстрацію. Розроблений бот обробляє ці дані та переформатовує їх для зручного відображення в інтерфейсі Telegram. Для зручності навігації по об’ємному каталогу реалізовано механізм посторінкового перегляду.

Завдяки інтеграції з ExerciseDB через платформу RapidAPI вдалося значно розширити функціональність бота, створивши для користувачів практичний інструмент пошуку й вибору фізичних вправ (рис. 2.3). Це доповнює можливості контролю харчування та сприяє формуванню цілісного підходу до підтримки здорового способу життя.

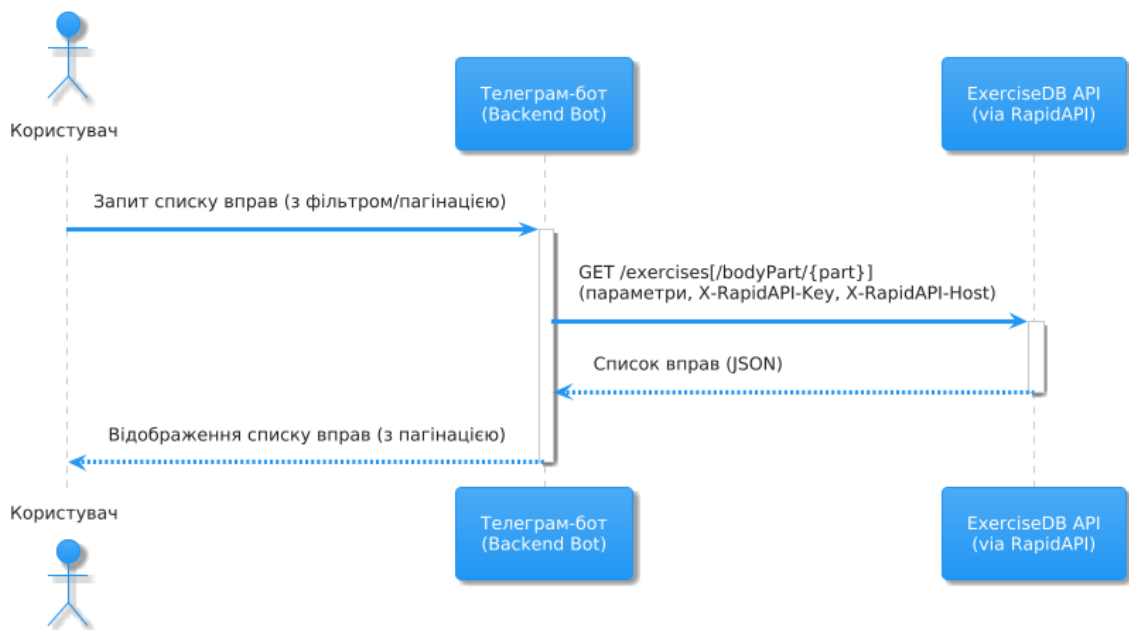


Рисунок 2.3 – UML діаграма взаємодії з API ExerciseDB через Rapid API

2.4 Модель розпізнавання страв та аналізу зображень

Ключовою особливістю створеного бота в Telegram є функція автоматичного визначення спожитої їжі та оцінювання її харчової цінності на основі даних від користувача – світлин, голосових описів чи текстових повідомлень. Впровадження цього компонента суттєво спрощує ведення харчового щоденника, позбавляючи потреби вручну шукати продукти в базах. З огляду на різноманітність зовнішнього вигляду страв, труднощі з визначенням складників та розмірів порцій за візуальними даними, а також бажання обробляти різні формати введення, вирішено застосувати зовнішню мультимодальну модель машинного навчання замість розробки власної системи комп'ютерного зору [17].

Для забезпечення функції розпізнавання обрано рішення з лінійки Google Gemini через відповідний програмний інтерфейс. Як зазначалося раніше, вибір Gemini обґрунтований її здатністю одночасно опрацьовувати та інтерпретувати дані різного типу (текст, зображення, звук), що цілком відповідає вимогам бота. Це дає змогу створити єдиний механізм для опрацювання різноманітних запитів користувача щодо харчування. Використання версії з оптимізацією швидкодії (gemini-2.0-flash) забезпечує прийнятний час відгуку під час інтерактивної взаємодії з ботом.

Взаємодію з API Google Gemini реалізовано в модулі `google_api.py` із застосуванням офіційної бібліотеки `google-generativeai` для Python [18]. Комунікація відбувається асинхронно через метод `client.aio.models.generate_content`, що дозволяє боту не блокуватися під час очікування результатів від моделі.

Процес розпізнавання залежить від типу вхідних даних:

– аналіз зображень: коли користувач надсилає фото страви, бот отримує файл, відкриває його за допомогою бібліотеки `Pillow` (`PIL.Image.open`) і передає зображення разом зі спеціально підготовленим текстовим запитом (`photo_prompt`) до функції `generate_content`;

– аналіз аудіо-описів: у разі отримання голосового повідомлення, бот зчитує аудіофайл як байти та надсилає його моделі як об'єкт `types.Part.from_bytes` із вказаним MIME-типом (`audio/mp3`) [19]. Водночас передається текстовий запит (`description_prompt`) з інструкціями щодо аналізу усного опису;

– аналіз текстових описів: якщо користувач надсилає текстовий опис страви, цей текст форматується й вставляється у відповідне місце запиту `description_prompt` перед відправленням моделі.

Важливу роль у забезпеченні точності та структурованості відповіді від Gemini відіграє техніка формування запитів. Розроблено два основних шаблони (`photo_prompt` для зображень та `description_prompt` для тексту/аудіо). Ці запити містять чіткі вказівки: визначають роль моделі («професійний дієтолог», «експерт з комп'ютерного зору»), встановлюють завдання (аналіз страви, оцінка поживності), перелічують необхідні параметри (назва страви, калорії, білки, жири, вуглеводи, вага) та вимагають подання результату виключно у форматі JSON. Для посилення вимоги щодо формату при зверненні до API застосовується конфігурація `{ 'response_mime_type': 'application/json', 'response_schema': FoodResponse }`. Параметр `response_schema` використовує Pydantic-модель `FoodResponse` для опису очікуваної структури, що дозволяє API Gemini самостійно перевіряти та формувати вихідні дані [20].

Отримавши відповідь, бот перевіряє її наявність, обробляє JSON-представлення (`json.loads`) і отримує словник з назвою страви та оцінкою поживності. Ці дані використовуються для відображення користувачеві та збереження в базі даних (таблиці `users_food`, оновлення `users_progress`). Процес взаємодії з API охоплено блоками `try...except` для опрацювання можливих помилок (збоїв мережі, помилок API, некоректних відповідей), а інформація про помилки зберігається для подальшого аналізу.

Інтеграція з Google Gemini через ретельно розроблені запити та використання можливостей API щодо форматування дозволила створити

потужний і гнучкий механізм розпізнавання харчування з різних типів даних безпосередньо в боті в Telegram.

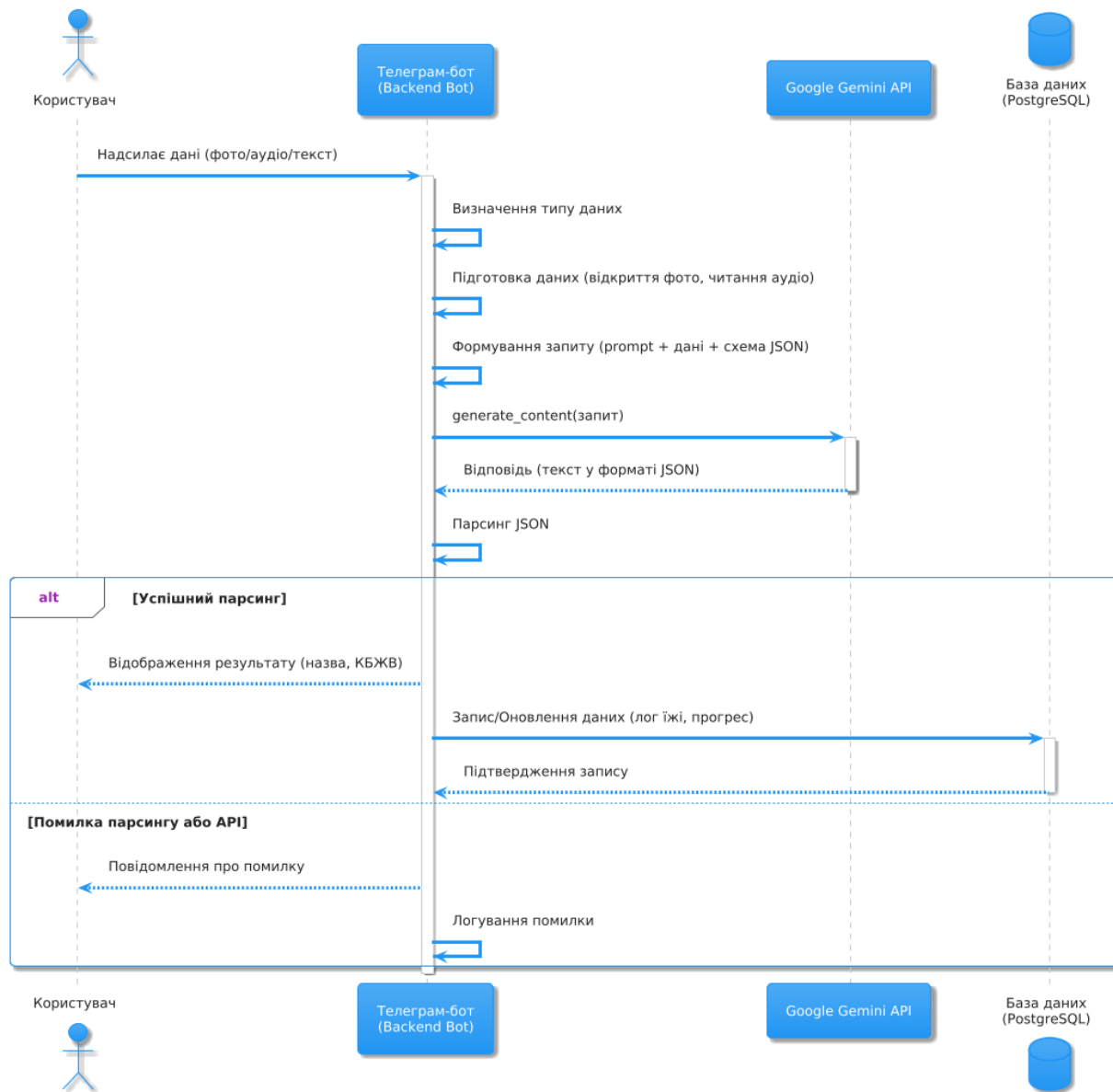


Рисунок 2.4 – UML діаграма взаємодії з Google Gemini API

2.5 Система добового моніторингу раціону

Центральною функцією розробленого бота в Telegram є забезпечення користувачів можливістю здійснювати інтерактивний моніторинг свого добового раціону та інших важливих показників здорового способу життя.

Ця система об'єднує дані, отримані з різних джерел (логування їжі, відстеження води та активності), порівнює їх із встановленими цілями користувача та надає зведену інформацію у зручному для сприйняття вигляді.

Основна мета системи добового моніторингу – надати користувачеві чітке уявлення про його поточний прогрес протягом дня щодо споживання калорій, макронутрієнтів (опосередковано через калорійність), води та фізичної активності. Це допомагає підвищити усвідомленість щодо власних звичок та приймати обґрунтовані рішення щодо подальших прийомів їжі чи активностей для досягнення поставлених цілей (схуднення, набір маси, підтримка форми).

Система добового моніторингу базується на кількох ключових компонентах та процесах.

Збір даних: інформація про спожиту їжу надходить до системи через різні канали:

- автоматичне розпізнавання: результати аналізу фото, аудіо або тексту за допомогою Google Gemini (назва страви, калорійність);

- ручне введення: користувач може самостійно вказати назву продукту/страви та її калорійність [21].

Ці дані записуються у детальний лог (`users_food`) та агрегуються для оновлення добового підсумку калорій. Інформація про спожиту воду та виконані вправи також фіксується через спеціальні нагадування та оновлює відповідні показники.

Щоденні підсумкові дані зберігаються у таблиці `users_progress`. При кожному логуванні їжі, води або активності відповідне поле (`calories`, `water`, `exercises`) для поточного дня та конкретного користувача (`user_id`, `date`) оновлюється. Якщо запису для поточного дня ще не існує, він створюється автоматично (як реалізовано у функції `get_user_progress` в `db.py`). Це дозволяє мати актуальну зведену інформацію про добовий прогрес.

Система отримує індивідуальні цілі користувача (добова норма калорій, води, активності) з поля `user_goals` таблиці `users`. Поточні агреговані показники з таблиці `users_progress` порівнюються з цими цілями для розрахунку відсотка виконання плану на день.

Для наочного представлення добового прогресу використовується модуль генерації зображень (`units/draw.py` на основі бібліотеки `Pillow`) [22]. Цей модуль генерує персоналізоване зображення-звіт (рис. 2.5), яке включає:

- кругові діаграми, що відображають відсоток досягнення добових цілей по калоріях, воді та активності;
- числові показники поточного стану та цілі (наприклад, «1250/2000 KCAL», «1500/2500 ML»);
- інформацію про останнє досягнення користувача (іконка та опис), що додає елемент гейміфікації;
- поточну дату.

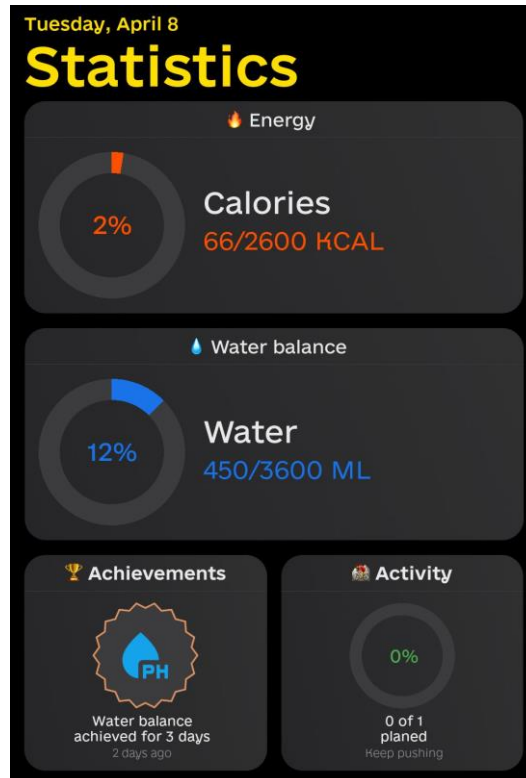



Рисунок 2.5 – Приклад візуалізації добового звіту

Таке візуальне представлення є значно інформативнішим та привабливішим для користувача, ніж простий текстовий звіт.

Користувач може отримати звіт про свій добовий прогрес за допомогою кнопки « Main menu». Бот обробляє цю команду, отримує необхідні дані з бази даних (users_progress, users, achievements), генерує зображення статистики та надсилає його користувачеві.

Таким чином, система добового моніторингу об'єднує функціонал логування, зберігання даних в PostgreSQL, порівняння з цілями та візуалізації результатів за допомогою Pillow. Це створює інтерактивний інструмент, що допомагає користувачам ефективно відстежувати свій раціон та прогрес у досягненні цілей здорового способу життя безпосередньо в інтерфейсі бота в Telegram.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ БОТА В TELEGRAM ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Архітектура системи та вибір технологій розробки

Після ґрунтовного опрацювання архітектурних рішень, структури даних та механізмів взаємодії, які було висвітлено у минулих розділах, наступним етапом стала безпосередня програмна реалізація бота в Telegram [23]. Вибір технологічного стеку та інструментарію суттєво впливає на швидкість розробки, ефективність пошуку помилок та загальну якість кінцевого продукту.

Незважаючи на те, що основний користувацький інтерфейс визначається можливостями платформи Telegram, для створення специфічних візуальних компонентів – статистичних зображень, символіки досягнень та компонування візуальних звітів – на початковому етапі використовувався онлайн-редактор Figma. Цей інструмент дозволив наперед спланувати вигляд майбутніх зображень, дібрати відповідні шрифти та кольорову гаму, а також підготувати графічні елементи (іконки) для подальшого застосування в коді [24]. Робота з Figma допомогла впорядкувати візуальну частину проєкту ще до її програмної реалізації (рис. 3.1).

Для написання основного коду системи було обрано інтегроване середовище розробки PyCharm. Цей редактор є одним із найпотужніших для роботи з мовою Python, яку було обрано основною для проєкту. PyCharm пропонує розробникам широкі можливості: розумне доповнення коду, миттєвий аналіз його якості, зручні засоби рефакторингу, вбудований відлагоджувач для покрокового виконання та аналізу значень змінних, а також підтримку систем контролю версій, зокрема Git [25]. Використання PyCharm значно прискорило процес розробки та допомогло завчасно виявляти помилки (рис. 3.2).

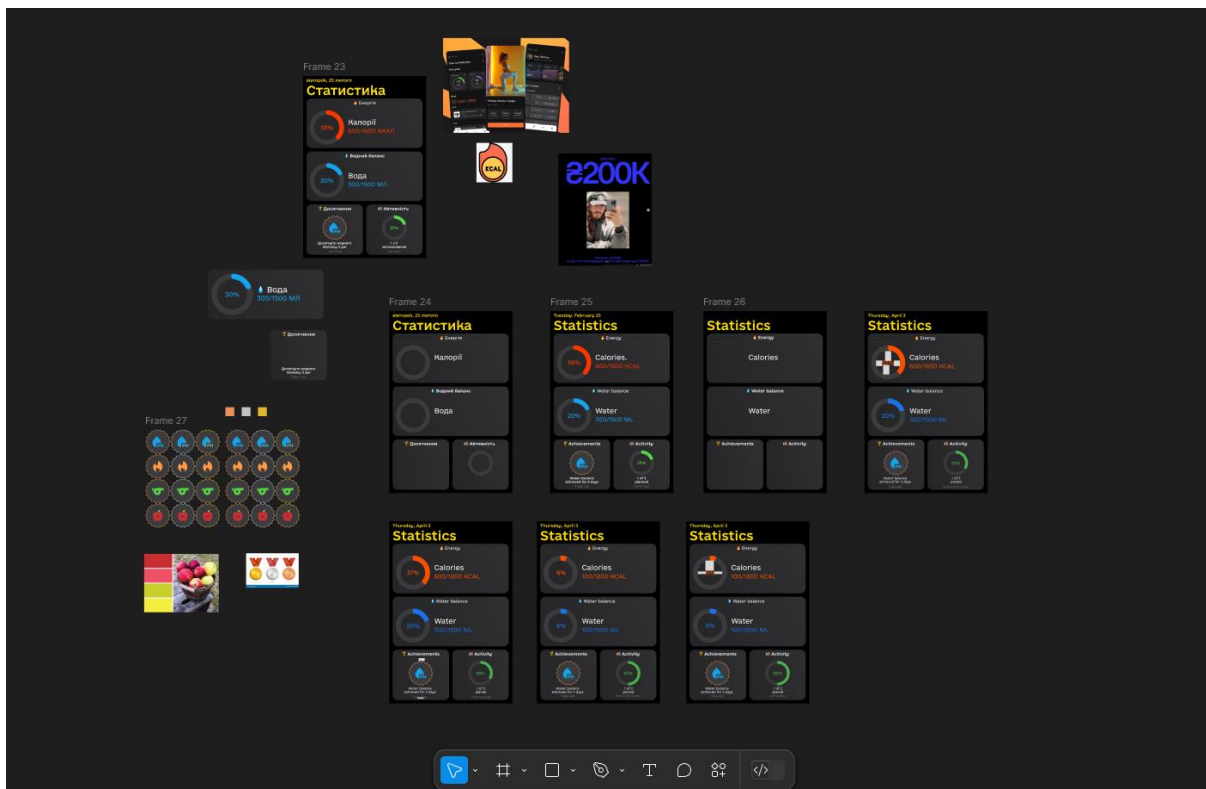


Рисунок 3.1 – Скріншот інтерфейсу Figma з макетами візуальних напрацювань для бота

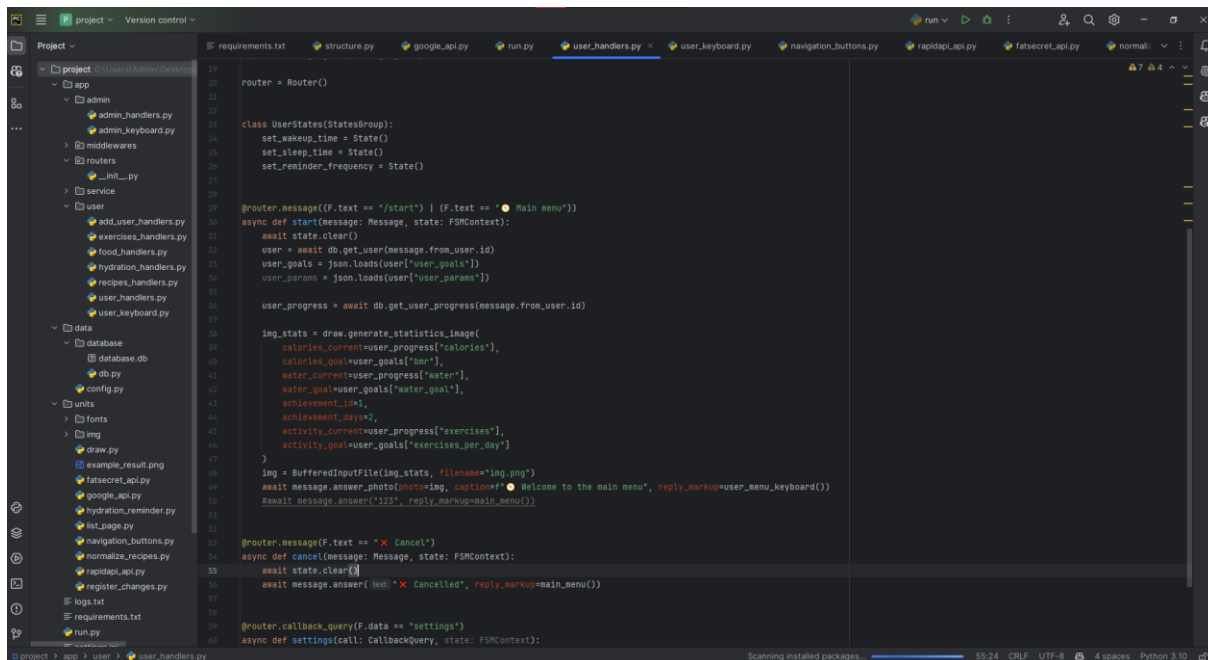


Рисунок 3.2 – Скріншот інтерфейсу редактора PyCharm

Як система управління базами даних використовується PostgreSQL. Щоб спростити адміністрування цієї системи, перегляд структури таблиць,

створення та виконання SQL-запитів, а також контроль стану бази під час розробки й тестування, було застосовано інструмент pgAdmin 4. Ця відкрита платформа надає графічний інтерфейс для роботи з PostgreSQL, що робить взаємодію з базою даних набагато зручнішою порівняно з використанням лише командного рядка. pgAdmin 4 дає змогу легко переглядати вміст таблиць, керувати користувачами та їхніми правами доступу, аналізувати ефективність запитів (рис. 3.3).

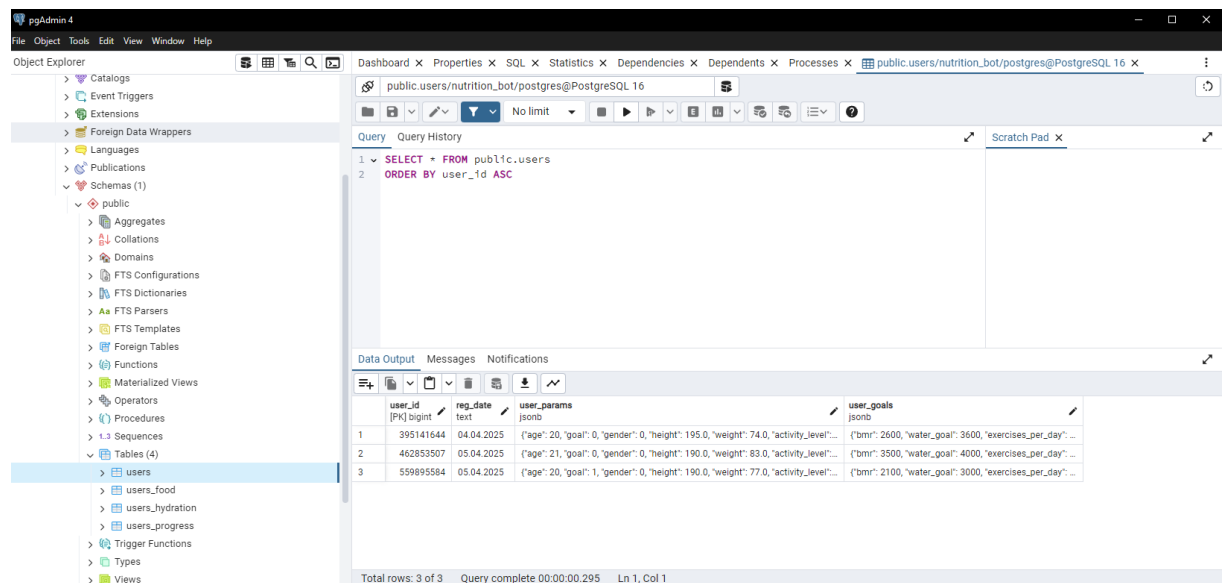


Рисунок 3.3 – Скріншот інтерфейсу pgAdmin 4

Отже, обраний технологічний стек – мова Python, середовище PyCharm для написання коду, система управління базами даних PostgreSQL, інструмент pgAdmin 4 для роботи з базою та Figma для створення прототипів візуальних елементів – становить збалансований і продуктивний набір засобів. Така комбінація дозволила успішно реалізувати всі функціональні складові бота в Telegram відповідно до вимог, окреслених у попередніх розділах кваліфікаційної роботи.

3.2 Програмна реалізація застосунку

Програмна реалізація бота в Telegram для моніторингу здорового харчування була виконана відповідно до архітектури та технологічного стеку, обґрунтованих у попередньому розділі. Основою реалізації слугувала мова програмування Python та асинхронний фреймворк Aiogram. Структура проєкту була організована за модульним принципом для забезпечення логічного розділення функціоналу та полегшення його підтримки.

Точкою входу до застосунку є файл `run.py`. У цьому файлі відбувається ініціалізація основних компонентів системи: створюються екземпляри об'єктів `Bot` та `Dispatcher` з бібліотеки `Aiogram`, встановлюється асинхронне з'єднання з базою даних `PostgreSQL` за допомогою функції `connect` класу `Database` (з модуля `data/database/db.py`), реєструються всі необхідні роутери з обробниками команд та повідомлень, налаштовується проміжне програмне забезпечення (`middleware`), зокрема для обмеження частоти запитів (`throttling`, реалізовано в `app/middlewares/middleware_throttling.py`), та запускається основний цикл обробки оновлень від Telegram (`dp.start_polling(bot)`). Також у `run.py` ініціюється запуск фонових завдань для системи нагадувань про гідратацію.

Логіка обробки взаємодії з користувачем розподілена по обробниках (`handlers`), згрупованих у відповідних директоріях всередині папки `app`. Використання механізму роутерів `Aiogram` дозволило чітко розділити функціонал:

- `app/user/`: містить основні обробники для взаємодії зі звичайними користувачами, розділені за функціями (реєстрація `add_user_handlers.py`, робота з їжею `food_handlers.py`, вправами `exercises_handlers.py`, рецептами `recipes_handlers.py`, гідратацією `hydration_handlers.py`, загальні команди `user_handlers.py`). Тут же знаходиться модуль `user_keyboard.py` для генерації специфічних для користувача клавіатур;

- `app/admin/`: містить обробники для адміністративних функцій (перегляд статистики, модерування користувачів, тощо);
- `app/service/`: включає обробники для специфічних системних подій, таких як обробка помилок (`error_handlers.py`) або повідомлень, що не були оброблені іншими хендлерами (`missed_handlers.py`).

Взаємодія з базою даних PostgreSQL інкапсульована у класі `Database` модуля `db.py`. Цей клас використовує бібліотеку `asynpcrg` для виконання всіх операцій з базою даних асинхронно. Реалізовано методи для основних CRUD-операцій (Create, Read, Update, Delete) стосовно таблиць, описаних раніше: додавання нового користувача (`add_user`), отримання даних користувача (`get_user`), оновлення параметрів та цілей (`update_user_params_and_goals`), отримання та оновлення щоденного прогресу (`get_user_progress`, `update_user_progress`), додавання записів про їжу (`add_food`), отримання та збереження налаштувань гідратації (`get_user_hydration`, `set_hydration_settings`) та інші.

Для реалізації багатоетапних діалогів, таких як процес реєстрації користувача з введенням параметрів та цілей, або налаштування системи нагадувань про гідратацію, активно використовується механізм машини станів (FSM) фреймворку `Aiogram`. Визначення станів (`StatesGroup`) та перевірка поточного стану користувача в обробниках дозволяють боту коректно вести користувача через послідовність кроків, збираючи необхідну інформацію.

Реалізація взаємодії з зовнішніми API виконана у відповідних модулях директорії `units`:

- `google_api.py`: містить асинхронні функції (`get_photo_nutrition`, `get_description_nutrition`, `get_voice_description_nutrition`) для надсилання запитів до API Google Gemini з використанням бібліотеки `google-gemai`. Реалізовано формування запитів з використанням розроблених промптів та обробку відповіді у форматі JSON завдяки параметрам `response_mime_type` та `response_schema` [26];

– `fatsecret_api.py`: реалізує логіку автентифікації (отримання OAuth 2.0 токена) та виконання асинхронних запитів до API FoodSecret для пошуку рецептів (`fetch_recipes`) та отримання їх деталей (`get_recipe_details`) за допомогою бібліотеки `aiohhttp`;

– `rapidapi_api.py`: містить функцію `get_exercises` для асинхронного запиту до ExerciseDB API через платформу RapidAPI з використанням `aiohhttp`, включаючи передачу необхідних заголовків автентифікації .

Система нагадувань про гідратацію реалізована у модулі `units/hydration_reminder.py`. Вона функціонує як асинхронне фонове завдання, яке з певною періодичністю перевіряє налаштування користувачів у базі даних (`users_hydration`) та поточний час, і надсилає нагадування тим користувачам, у яких ця функція активована і настав час для сповіщення.

Генерація візуальних звітів про добовий прогрес здійснюється функцією `generate_statistics_image` у модулі `units/draw.py`. Ця функція використовує бібліотеку Pillow для створення зображення на основі шаблону (`background_template.png`), малювання кругових діаграм прогресу (`draw_circular_progress`), нанесення тексту з поточними показниками, цілями та інформацією про досягнення, використовуючи кастомні шрифти з папки `units/fonts`. Готове зображення повертається у вигляді байтового потоку для надсилання користувачеві через Telegram.

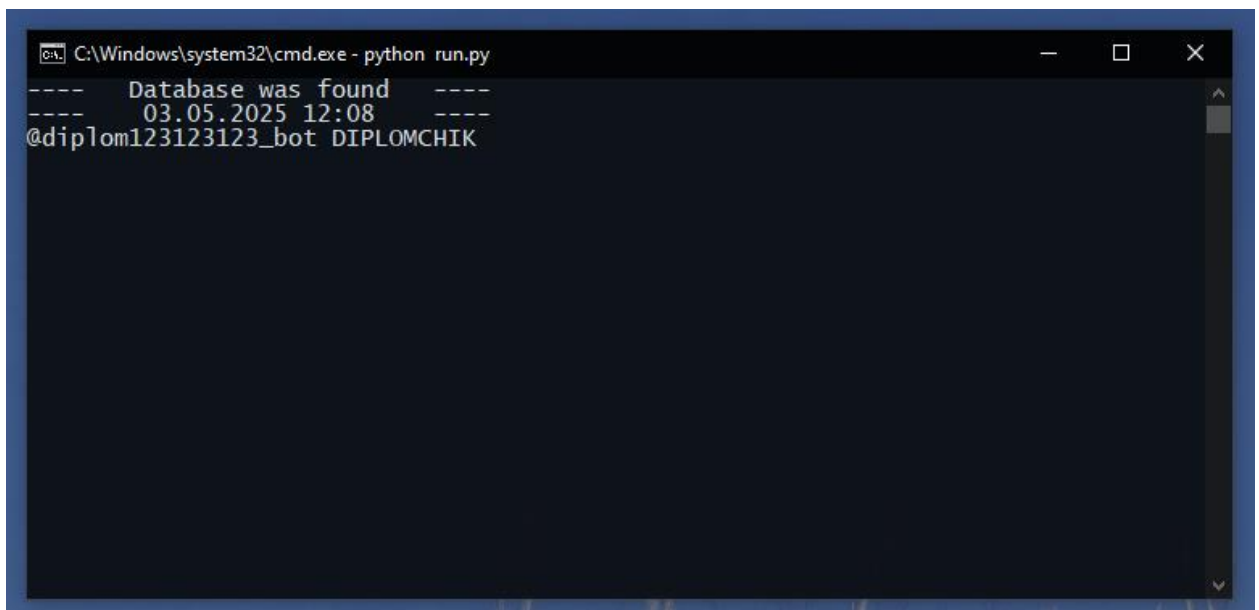
Утилітарні функції, такі як робота з датами та часом (з урахуванням часових поясів за допомогою `pytz`), обробка пагінації для списків рецептів та вправ (`list_page.py`, `navigation_buttons.py`), а також інші допоміжні операції, винесені в окремі модулі для кращої організації коду. Управління конфігурацією та секретними ключами здійснюється через файл `settings.ini` та модуль `data/config.py`.

Програмна реалізація застосунку послідовно втілює спроектовану архітектуру, використовуючи можливості обраних технологій для створення функціонального та інтерактивного бота в Telegram. Модульна структура,

асинхронний підхід та чітке розділення відповідальності між компонентами забезпечують працездатність системи та можливість її подальшого розвитку.

3.3 Інструкція користувача для використання застосунку

Перед початком взаємодії з ботом у Telegram необхідно запустити його бекенд-скрипт [27]. Це здійснюється шляхом виконання файлу `run.py` з кореневої директорії проєкту в середовищі Python. Успішний запуск супроводжується виведенням у консоль повідомлень про знаходження бази даних та інформації про запущеного бота (рис. 3.4).



```
C:\Windows\system32\cmd.exe - python run.py
---- Database was found ----
---- 03.05.2025 12:08 ----
@dip1om123123123_bot DIPLOMCHIK
```

Рисунок 3.4 – Скріншот консолі після успішного запуску `run.py`

Для початка роботи в Telegram користувач знаходить бота в Telegram за його іменем користувача (username) та ініціює діалог, натиснувши кнопку "Start" або надіславши команду `/start`.

При першому запуску бот автоматично переходить у режим налаштування профілю. Користувачеві послідовно пропонується ввести основні антропометричні дані: поточну вагу (в кг), зріст (в см), вік, стать (вибором з варіантів), обрати свою основну мету (схуднення, набір ваги,

підтримка поточної ваги) та вказати рівень щоденної фізичної активності (рис. 3.5). Ці дані є необхідними для розрахунку індивідуальних рекомендацій щодо добової норми калорій та інших показників. Після введення всіх даних профіль користувача зберігається.

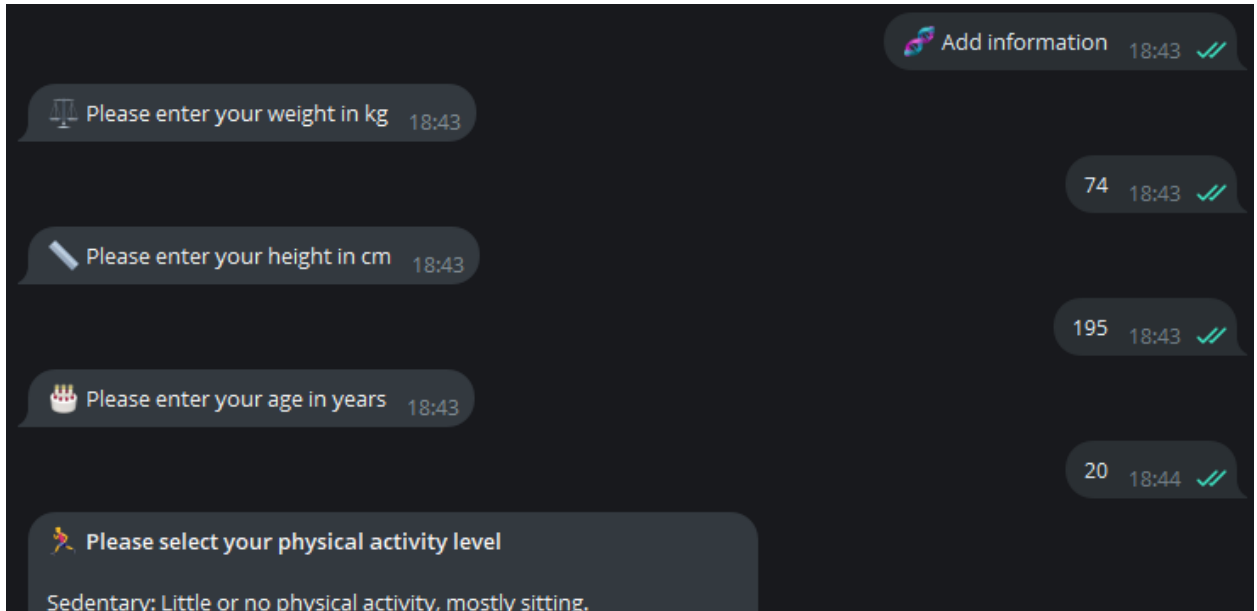


Рисунок 3.5 – Скріншот налаштування профілю користувачем

Після успішного налаштування профілю користувач потрапляє до головного меню бота. На цьому екрані відображається актуальне зображення зі зведеною статистикою за поточний день (спожиті калорії, вода, активність відносно цілей) та основні кнопки для доступу до функцій: «Add Food» (Додати їжу), «Recipes» (Рецепти), «Exercises» (Вправи), «Hydration» (Гідратація) (рис. 3.6).

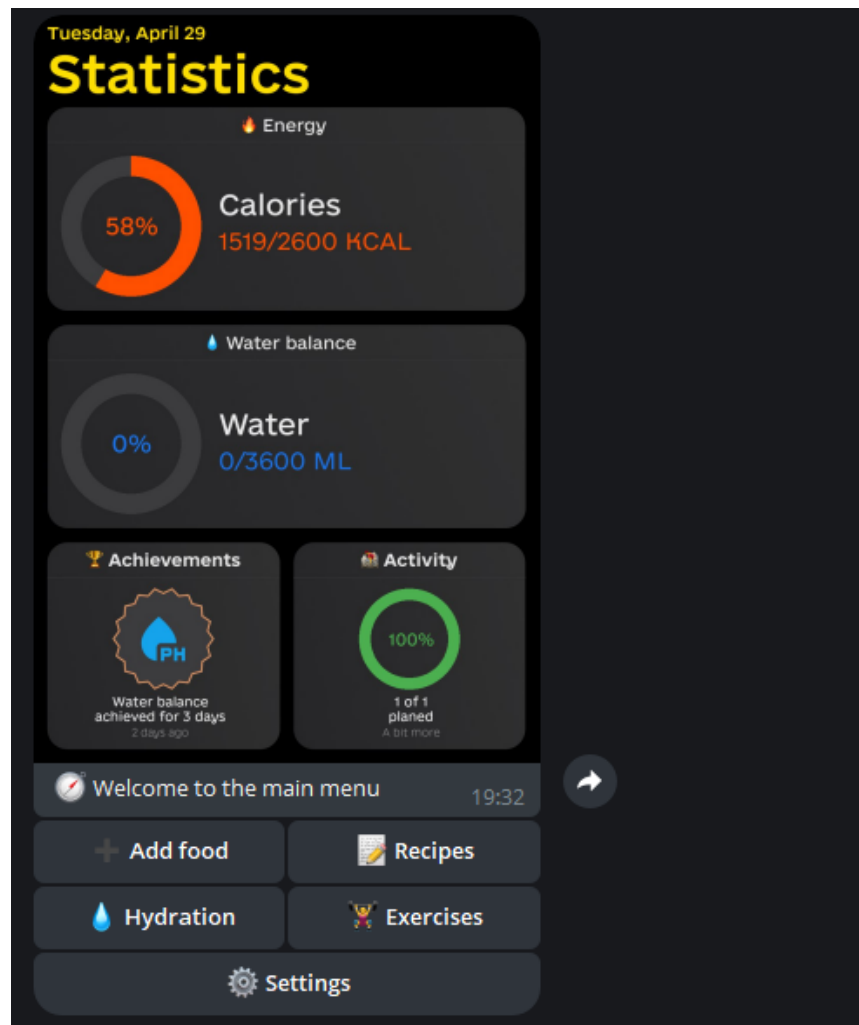


Рисунок 3.6 – Скріншот головного меню бота

Додавання прийому їжі («Add Food»):

– натискання кнопки «Add Food» відкриває меню вибору способу введення інформації про їжу: за фотографією, за аудіо-описом або за текстовим описом (рис. 3.7);

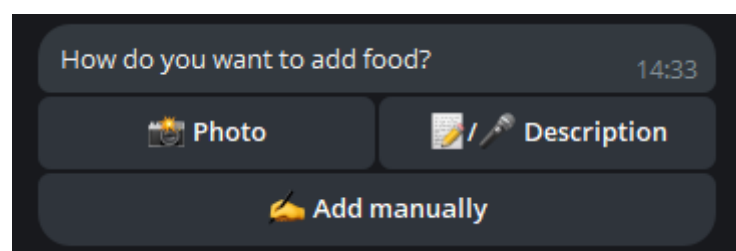


Рисунок 3.7 – Скріншот меню вибору способу додавання їжі

– користувач обирає зручний метод та надсилає відповідні дані (файл фото, голосове повідомлення або текст);

– бот обробляє отримані дані за допомогою інтеграції з Google Gemini та надсилає у відповідь повідомлення з розпізнаною назвою страви, оцінкою її калорійності та вмісту білків, жирів, вуглеводів. Під цим повідомленням знаходиться кнопка «Add to your food intake» (Додати прийом їжі) (рис. 3.8).



Рисунок 3.8 – Скріншот повідомлення від бота з результатами розпізнавання

їжі

Налаштування нагадувань про гідратацію («Hydration»):

– перехід до меню «Hydration» дозволяє налаштувати систему нагадувань. Користувачу необхідно ввести час свого пробудження, час відходу до сну та бажану частоту надсилання нагадувань (наприклад, кожні 120 хвилин) (рис. 3.9);

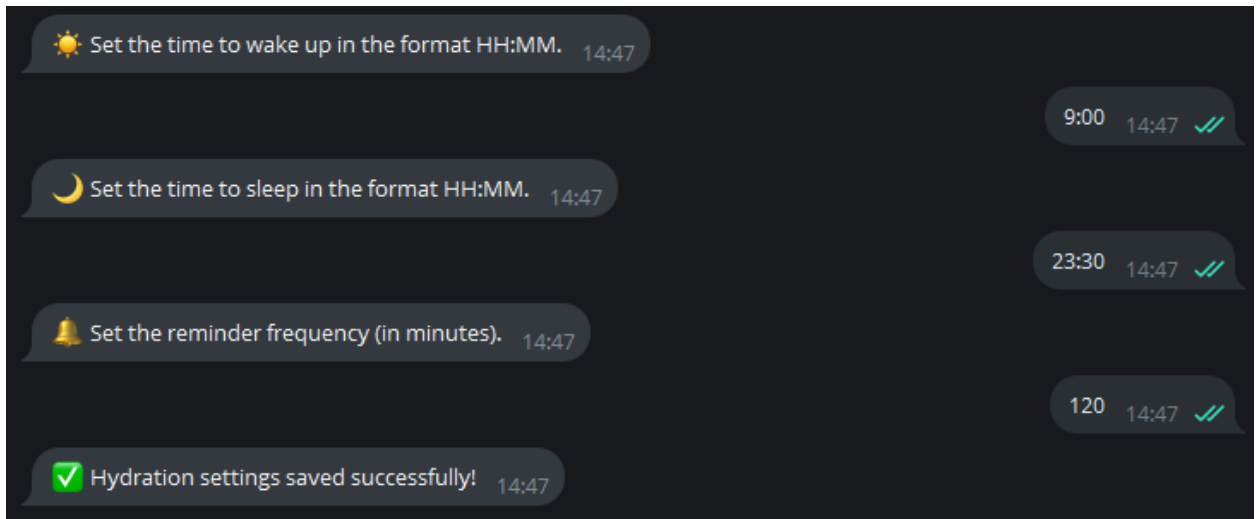


Рисунок 3.9 – Скріншот інтерфейсу налаштування параметрів гідратації

– на основі цілей користувача та введених даних бот розраховує рекомендовану добову норму води та розмір порції для кожного нагадування;

– після активації бот починає надсилати нагадування випити води у вказаному часовому діапазоні (між пробудженням та сном) із заданою частотою. Кожне повідомлення-нагадування містить кнопку «Done» (Додати воду) (рис. 3.10).

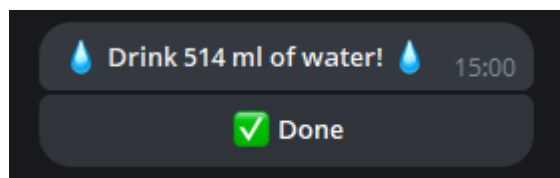


Рисунок 3.10 – Скріншот прикладу повідомлення-нагадування про воду

Пошук рецептів («Recipes»):

– натискання кнопки «Recipes» відкриває інтерфейс пошуку рецептів. Користувач може обрати категорію страви (сніданки, коктейлі, салати тощо) за допомогою кнопок-фільтрів або ввести ключове слово (наприклад, «курка», «авокадо») для пошуку рецептів, що містять цей інгредієнт (рис. 3.11);

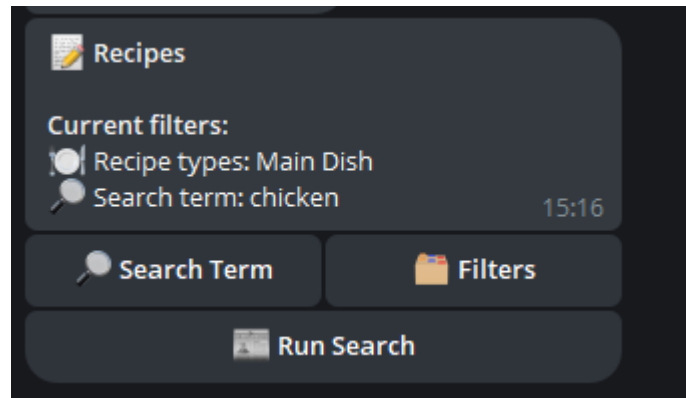


Рисунок 3.11 – Скріншот інтерфейсу пошуку рецептів з фільтрами

– бот відображає список знайдених рецептів. Користувач може переглядати деталі обраного рецепту (інгредієнти, інструкції, харчова цінність), можливість додати харчову цінність обраного рецепту до свого добового логу.

Пошук вправ («Exercises»):

– натискання кнопки «Exercises» дозволяє шукати фізичні вправи. Користувач може вибрати цільову групу м'язів або тип активності (кардіо, біцепс, спина тощо) за допомогою кнопок-фільтрів (рис. 3.12);

– бот відображає список відповідних вправ, надаючи можливість переглянути їх опис та, можливо, анімацію виконання. Користувач може відзначити виконану вправу, що оновлює показник активності у добовій статистиці.

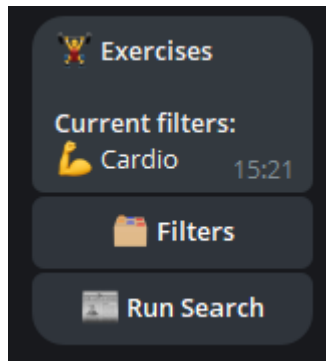


Рисунок 3.12 – Скріншот інтерфейсу пошуку вправ з фільтрами

Актуальна зведена статистика за поточний день завжди доступна у вигляді зображення в головному меню бота. Дотримання цієї інструкції дозволить користувачеві максимально ефективно використовувати всі функції розробленого бота в Telegram для досягнення своїх цілей у сфері здорового харчування та способу життя.

3.4 Перспективи проєкта

Розроблений бот в Telegram для моніторингу здорового харчування з використанням технологій машинного навчання та інтерактивного моніторингу добового раціону є функціональним прототипом, що демонструє потенціал такого роду систем. Однак, як і будь-який програмний продукт, він має значні перспективи для подальшого розвитку та вдосконалення. Нижче наведено можливі напрямки майбутньої роботи над проєктом, що можуть підвищити його цінність для користувачів та розширити функціональні можливості.

Розширення бази даних та покращення розпізнавання:

– інтеграція з додатковими базами даних продуктів: підключення до локалізованих баз даних продуктів харчування, специфічних для певних регіонів або країн, може значно підвищити точність підрахунку калорій для

більшої кількості продуктів, особливо тих, що не мають універсальних штрих-кодів або відсутні в глобальних базах;

– вдосконалення моделі розпізнавання: постійне оновлення та донавчання моделей Google Gemini (або використання новіших, більш досконалих версій, що з'являтимуться) може покращити точність розпізнавання страв, особливо для складних або нетипових випадків. Також можлива розробка власної моделі, донавченої на специфічних датасетах української кухні, для ще кращої ідентифікації локальних страв;

– оцінка об'єму порції: дослідження та впровадження методів для більш точної оцінки об'єму порції на фотографії (наприклад, за допомогою референтних об'єктів на фото або алгоритмів тривимірної реконструкції з декількох знімків) дозволить точніше розраховувати калорійність.

Розширення функціоналу моніторингу та аналітики:

– детальний аналіз мікронутрієнтів: окрім БЖВ, додати можливість відстеження та аналізу споживання вітамінів, мінералів та інших важливих мікронутрієнтів, інтегруючись з відповідними базами даних харчової цінності;

– розширені звіти та довгострокова статистика: надання користувачам можливості переглядати статистику свого харчування, ваги та активності за тиждень, місяць, рік у вигляді графіків та діаграм для відстеження довгострокових тенденцій;

– персоналізовані рекомендації: на основі аналізу даних користувача (цілі, поточний раціон, рівень активності, можливі харчові обмеження) генерувати більш персоналізовані рекомендації щодо корекції раціону, вибору продуктів чи планування фізичних навантажень [28].

Покращення користувацького досвіду та гейміфікації:

– більш розвинена система досягнень: впровадження ширшого спектру досягнень за різні активності (регулярне логування, досягнення цілей по воді/калоріях/активності протягом певного періоду, використання різних функцій бота), що може підвищити мотивацію користувачів;

– інтерактивні виклики та змагання: можливість створювати або приєднуватися до викликів (наприклад, «30 днів без цукру», «10 000 кроків щодня») з друзями або іншими користувачами бота [29];

– інтеграція з фітнес-трекерами та іншими застосунками: дозволити синхронізацію даних про фізичну активність та сон з популярними фітнес-трекерами (Google Fit, Apple Health, Mi Fit тощо) для автоматичного оновлення відповідних показників у боті [30].

Технічні вдосконалення та масштабування:

– оптимізація роботи з API: впровадження кешування відповідей від зовнішніх API для зменшення кількості запитів та прискорення роботи бота, особливо для часто запитуваних даних (популярні рецепти, вправи);

– розгортання на хмарній платформі: для забезпечення стабільної роботи 24/7 та можливості обслуговування великої кількості користувачів, необхідне розгортання бота на надійній хмарній платформі (AWS, Google Cloud, Azure, Heroku);

– локалізація та багатомовність: додавання підтримки інших мов інтерфейсу та розпізнавання для розширення аудиторії користувачів.

Монетизація: якщо проект розвиватиметься як комерційний продукт, можна розглянути різні моделі монетизації, такі як преміум-підписка з розширеним функціоналом (детальна аналітика, персональні плани харчування від дієтологів, відсутність обмежень), або партнерські програми з сервісами доставки здорової їжі чи спортивними клубами.

Реалізація навіть частини з перерахованих напрямків може суттєво підвищити конкурентоспроможність та корисність розробленого бота в Telegram, перетворивши його на комплексного помічника у веденні здорового способу життя.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований бот в Telegram для моніторингу здорового харчування, що використовує сучасні технології машинного навчання та забезпечує інтерактивний моніторинг добового раціону користувача.

Під час виконання роботи було проведено аналіз існуючих рішень у сфері моніторингу харчування, виявлено їх переваги та недоліки, що обґрунтувало актуальність розробки нового інструменту. Було досліджено технології машинного навчання для аналізу зображень, аудіо та текстових даних, зокрема можливості моделей Google Gemini, та спроектовано архітектуру системи, що забезпечує їх ефективну інтеграцію. Обрано технологічний стек на основі мови Python, фреймворку Aiogram та СУБД PostgreSQL, що дозволило реалізувати асинхронну та модульну систему.

Розроблений бот надає користувачам функціонал для розпізнавання страв за фотографіями, аудіо та текстовими описами, ведення добового логу харчування, доступу до бази рецептів та фізичних вправ, налаштування нагадувань про гідратацію та моніторингу прогресу за допомогою візуалізованої статистики та елементів гейміфікації. Інтеграція з API Google Gemini, FoodSecret та RapidAPI (ExerciseDB) дозволила значно розширити функціональні можливості бота.

Результатом роботи є програмний продукт, що демонструє практичне застосування сучасних технологій для вирішення завдання підтримки здорового способу життя. Розроблений бот в Telegram може слугувати зручним та ефективним інструментом для користувачів, які прагнуть контролювати свій раціон та підвищити рівень фізичної активності.

Результати роботи апробовано у вигляді тез доповідей під час Міжнародної науково-практичної конференції «Сучасні інформаційні технології та системи штучного інтелекту MIT@AIS-2025» [31].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tvoroshenko, I., Pomazan, V., Gorokhovatskyi, V., & Kobylin, O. (2023). Application of video data classification models using convolutional neural networks.
2. Kaur, R., Kumar, R., & Gupta, M. (2023). Deep neural network for food image classification and nutrient identification: A systematic review. *Reviews in Endocrine and Metabolic Disorders*, 24(4), 633-653.
3. Team, G., Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., ... & Blanco, L. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.
4. Tvoroshenko, I., Gorokhovatskyi, V., Kobylin, O., & Tvoroshenko, A. (2023). Application of deep learning methods for recognizing and classifying culinary dishes in images.
5. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.
6. Aloysius, N., & Geetha, M. (2017, April). A review on deep convolutional neural networks. In 2017 international conference on communication and signal processing (ICCSP) (pp. 0588-0592). IEEE.
7. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
8. Tvoroshenko, I., & Temchur, K. (2021). Features of software application development for food recognition using deep machine learning methods.
9. КОБИЛІН, О., ВЕЧІРСЬКА, І., & АФАНАСЬЄВ, А. (2024). Аналіз існуючих моделей глибинного навчання в задачах обробки природної мови. *Information Technology: Computer Science, Software Engineering and Cyber Security*, (3), 63-76.
10. Kothadiya, D., Pise, N., & Bedekar, M. (2020). Different methods review for speech to text and text to speech conversion. *International Journal of Computer Applications*, 175(20), 9-12.

11. Tvoroshenko, I. (2020). Information technologies for decision-making on the conditions of spatially distributed objects. In I International Scientific and Practical Conference. Problems and perspectives of modern science and practice, Austria (pp. 45-50).

12. Python Software Foundation, `asyncio` — Asynchronous I/O. URL: <https://docs.python.org/3/library/asyncio.html> (дата звернення 15.04.2025).

13. Aiogram Documentation. URL: <https://docs.aiogram.dev/en/latest/> (дата звернення 16.04.2025).

14. PostgreSQL 16 Documentation, JSON Types. URL: <https://www.postgresql.org/docs/current/datatype-json.html> (дата звернення 17.04.2025).

15. FatSecret Platform API Documentation. URL: <https://platform.fatsecret.com/docs/> (дата звернення 18.04.2025).

16. ExerciseDB API. RapidAPI Hub. URL: https://rapidapi.com/justin-WFnsXH_t6/api/exercisedb (дата звернення 22.04.2025).

17. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description. *Advances in Electrical and Electronic Engineering*, 21(1), 19.

18. Google Generative AI SDK for Python. PyPI. URL: <https://pypi.org/project/google-generativeai/> (дата звернення 23.04.2025).

19. Lyashenko, V., Rabotiahov, A., Kobylin, O., & Kolesnykov, D. (2018, October). Analysis of human speech as a protection tool in infocommunication systems. In 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T) (pp. 79-83). IEEE.

20. Pydantic Documentation. URL: <https://docs.pydantic.dev> (дата звернення 24.04.2025).

21. Кобилін, О. А., Вискребенцева, С. О., & Петрова, Р. В. (2019). Обробка даних, що містять пропуски в задачах кластеризації. Системи управління, навігації та зв'язку, (5), 45-50.

22. Python Imaging Library (PIL) and Pillow contributors. URL: <https://pillow.readthedocs.io/en/stable/> (дата звернення 25.04.2025).
23. Творошенко, І. С. (2021). Технології прийняття рішень в інформаційних системах.
24. Figma: The Collaborative Interface Design Tool. URL: <https://www.figma.com/> (дата звернення 26.04.2025).
25. Van Horn II, B. M., & Nguyen, Q. (2023). Hands-on application development with PyCharm: Build applications like a Pro with the ultimate Python development tool. Packt Publishing Ltd.
26. Gorokhovatskyi, V., Chmutov, Y., Tvoroshenko, I., & Kobylin, O. (2025). Reducing computational costs by compressing the structural description in image classification methods. *Advanced Information Systems*, 9(1), 5-12.
27. Karakonstantyn, D., & Tvoroshenko, I. (2024). About the issue of optimization the performance of the server part of the information system.
28. Ситніков, Д. Е., Ситнікова, П. Е., Тітов, С. В., & Тітова, О. В. (2021). Фільтрація результуючого набору асоціативних правил з точки зору оцінки цікавості. *Системи обробки інформації*, (1 (164)), 83-88.
29. Titov, S. V., & Titova, O. V. (2015). Оцінка юзабіліті освітніх сайтів: методи і технології. *Вісник Харківської державної академії культури*, (47).
30. Google Fit API Documentation. URL: <https://developers.google.com/fit> (дата звернення 28.04.2025).
31. Ступак К.О. Розробка Телеграм-бота з використанням технології машинного навчання для моніторингу здорового харчування. Сучасні інформаційні технології та системи штучного інтелекту MIT@AIS-2025: тези доповідей 1-ї Міжнародної науково-практичної конференції (Харків, 19–22 травня 2024 р.). Харків: ХНУРЕ, 2024. Т. 7. С. 41–42.