

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 МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ МОНІТОРИНГУ
СТАНУ ОРГАНІЗМУ З НАДАННЯМ РЕКОМЕНДАЦІЙ

(тема)

Виконав:
здобувач 4 року навчання,

групи ІТІНФ-21-3

Горбачов І. Д.
(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. Вечірська І.Д.
(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 Горбачову Іллі Дмитровичу
(прізвище, ім'я, по батькові)1. Тема роботи Розробка мобільного застосунку для моніторингу стану організму з наданням рекомендацій

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 червня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література з питань мобільної розробки, обробки біометричних даних та UX/UI-дизайну; матеріали міжнародних науково-технічних конференцій, присвячених цифровій медицині та фітнес-технологіям; а також достовірні інформаційні ресурси мережі інтернет, що містять актуальні наукові та практичні дані з теми дослідження.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз предметної області.

2. Огляд фреймворків Python для створення мобільних застосунків.

3. Вивчення методів збору та обробки біометричних даних.

4. Особливості UX/UI-дизайну мобільних застосунків для здоров'я.

5. Програмна реалізація застосунку.

6. Тестування та оптимізація продуктивності застосунку.

7. Розробка алгоритмів персоналізованих рекомендацій.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми розробки мобільного застосунку для моніторингу організму, постановка задачі, огляд фреймворків Python, опис програмної реалізації застосунку.

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	08.04.25-11.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка застосунку	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. Вечірська І.Д.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 65 с., 13 рис., 30 джерел.

ЗДОРОВ'Є, МОБІЛЬНИЙ ЗАСТОСУНОК, БІОМЕТРИЧНІ ДАНІ, UX/UI-ДИЗАЙН, ФІТНЕС-ТЕХНОЛОГІЇ, АНАЛІЗ СТАНУ, РЕКОМЕНДАЦІЇ, КОНТРОЛЬ ФІЗИЧНОГО СТАНУ.

Об'єктом дослідження є розробка мобільного застосунку для моніторингу та контролю фізичних показників з метою підтримки здорового способу життя.

Актуальність зумовлена зростанням інтересу до здоров'я, розвитком фітнес-індустрії та необхідністю впровадження цифрових рішень у повсякденне життя.

Метою роботи є створення застосунку для моніторингу фізичних показників, аналізу стану організму і формування персоналізованих рекомендацій.

Проаналізовано сучасні засоби мобільної розробки, методи збору та обробки біометричних даних і принципи UX/UI-дизайну.

У результаті роботи розроблено функціональний, доступний і інтерактивний застосунок, що контролює здоров'я, відстежує прогрес, формує звички та надає рекомендації щодо активності, сну й харчування.

HEALTH, MOBILE APPLICATION, BIOMETRIC DATA, UX/UI DESIGN, FITNESS TECHNOLOGIES, BODY CONDITION ANALYSIS, RECOMMENDATIONS, PHYSICAL CONDITION CONTROL.

The object of the research is the development of a mobile application for monitoring and controlling physical indicators to support a healthy lifestyle.

The relevance is caused by growing interest in health, the development of the fitness industry, and the need to implement digital solutions into everyday life.

The purpose of the work is to create an application for monitoring physical indicators, analyzing the body's condition, and generating personalized recommendations.

Modern mobile development tools, biometric data collection and processing methods, and UX/UI design principles were analyzed.

As a result of the work, a functional, accessible, and interactive application was developed that monitors health, tracks progress, builds habits, and provides recommendations for activity, sleep, and nutrition.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	7
Вступ	8
1 Методи аналізу даних у мобільному моніторингу стану організму	10
1.1 Термінологія і основні задачі аналізу даних у мобільному моніторингу стану організму	10
1.2 Метод класифікації стану за допомогою гібридних показників ...	10
1.3 Метод кластеризації змін стану у часовому ряду	11
1.3.1 Кластеризація часових сегментів	12
1.3.2 Генерація рекомендацій	13
1.4 Метод прогнозування стану організму на основі часових рядів ...	15
1.4.1 Формалізація задачі прогнозування	16
1.4.2 Побудова персоналізованої моделі користувача.....	17
1.5 Огляд існуючих цифрових рішень	19
1.5.1 Застосунок Goggle Fit.....	20
1.5.2 Застосунок Apple Health.....	21
1.5.3 Застосунок Fatsecret	22
1.6 Постановка задачі.....	23
2 Мобільний моніторинг здоров'я: аналіз даних.....	25
2.1 Формалізація методів аналізу у моніторингу здоров'я.....	25
2.2 Використання евклідової відстані для оцінки схожості	26
2.2.1 Формалізація евклідової метрики	27
2.3 Інтеграція моделей класифікації у мобільні системи моніторингу	29
2.4 Інтеграція кластеризації для аналізу стану здоров'я.....	32
2.4.1 Приклади алгоритмів кластеризації.....	32
3 Практична розробка мобільного застосунку для моніторингу стану організму з наданням рекомендацій.....	38

3.1	Розробка мобільних застосунків для Android за допомогою мови програмування Python.....	38
3.2	Обґрунтування вибору засобів для розробки застосунку.....	41
3.3	Реалізація застосунку	48
3.3.1	Опис основних компонентів застосунку.....	50
	Висновки	61
	Перелік джерел посилання	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

ВООЗ – Всесвітня організація охорони здоров'я

k-NN – k – Nearest Neighbors (k – найближчі сусіди)

MVC – Model-View-Controller (модель-вигляд-контролер)

LSTM – Long Short-Term Memory (довга короткочасна пам'ять)

ARIMA – AutoRegressive Integrated Moving Average (авторегресійна інтегрована модель ковзного середнього)

OAuth 2.0 – протокол авторизації без передачі пароля

VAR – Vector Autoregression (векторні авторегресійні моделі)

ВСТУП

Сучасні інформаційні технології відіграють ключову роль у підтримці та покращенні багатьох аспектів повсякденного життя, зокрема у сфері охорони здоров'я [1 – 2]. Одним із найперспективніших напрямів розвитку є створення програмних застосунків, які забезпечують збір, аналіз і обробку даних про фізичний стан людини з метою формування персоналізованих рекомендацій, що відповідають індивідуальним потребам користувачів [3 – 4].

Збір інформації може здійснюватися як самостійно – шляхом вимірювання ваги, пульсу за допомогою секундоміра, розрахунку індексу маси тіла, визначення норми калорій та контролю за споживанням вітамінів, мінералів, білків, жирів і вуглеводів, так і за допомогою сучасних пристроїв, таких як фітнес-трекери, сенсори й браслети, що автоматизують цей процес. Окрім цього, значна кількість користувачів звертається до медичних спеціалістів для отримання консультацій, що вимагає значних часових затрат. Особливої уваги заслуговують персоналізовані рекомендації, сформовані на основі зібраних даних, адже їх самостійний збір або використання кількох застосунків із обмеженим функціоналом часто ускладнює та уповільнює процес.

На сучасному ринку представлені популярні мобільні застосунки, такі як MyFitnessPal, Fitbit, Samsung Health та інші, які надають можливості для моніторингу фізичної активності, харчування та загального стану здоров'я. Проте, більшість із них має стандартний набір функцій, не охоплює комплексного аналізу та не забезпечує гнучкої персоналізації рекомендацій, особливо з урахуванням багатовимірних джерел даних [5 – 8].

Отже, розробка нових програмних засобів у сфері здоров'я потребує синтезу сучасних технологій мобільної розробки, методів обробки даних та інтелектуального аналізу інформації, що дозволить отримати більш точні та корисні висновки для користувача.

Актуальність дослідження зумовлена зростаючими проблемами зі здоров'ям і способом життя в суспільстві [9]. За даними Всесвітньої організації охорони здоров'я (ВООЗ), у 2022 році понад 43 % дорослого населення мали надмірну вагу, а 31 % – низький рівень фізичної активності [10]. Ці показники підкреслюють нагальну потребу в ефективних заходах для моніторингу та покращення здоров'я населення, серед яких важливе місце посідає впровадження мобільних застосунків, здатних відстежувати фізичну активність, харчування та інші параметри способу життя без обов'язкового залучення медичних працівників.

Головним завданням цієї роботи є проектування та створення мобільного застосунку для моніторингу стану організму, який інтегрує два різні підходи до аналізу фізіологічних параметрів, що дозволяє формувати персоналізовані рекомендації для покращення здоров'я користувача.

Окрім того, слід зауважити, що ефективність такого застосунку значною мірою залежить не лише від точності збору фізіологічних показників, а й від здатності системи адаптуватися до мінливих умов зовнішнього середовища та індивідуальних змін у режимі життя користувача. Наприклад, сезонні коливання, рівень стресу, якість сну, психоемоційний стан, зміна звичних харчових уподобань або рівня щоденної активності – усе це може суттєво впливати на стан здоров'я, а отже, потребує динамічного коригування рекомендацій. У зв'язку з цим важливим є впровадження механізмів машинного навчання, що дозволяють моделі самостійно навчатися на основі накопичених даних та забезпечувати більш точну індивідуалізацію висновків. Крім того, інтеграція з іншими інформаційними платформами – наприклад, медичними електронними картками чи хмарними базами даних про харчові продукти – відкриває додаткові можливості для багатofакторного аналізу, який може суттєво підвищити ефективність превентивних заходів та мотиваційних стратегій.

1 МЕТОДИ АНАЛІЗУ ДАНИХ У МОБІЛЬНОМУ МОНІТОРИНГУ СТАНУ ОРГАНІЗМУ

1.1 Термінологія і основні задачі аналізу даних у мобільному моніторингу стану організму

Мобільний моніторинг стану здоров'я – це сфера, що об'єднує технології аналізу, обробки даних та інтерфейсу для користувача для надання інформації про стан здоров'я [11-15]. Застосунок аналізує та трансформує усі фізіологічні вхідні дані в доступний для звичайної людини висновок.

Основні задачі пов'язані з мобільним моніторингом стану організму по своїй суті є дуже важливими та складними. Складність полягає у зборі інформації, зазвичай вона не є повною, що в свою чергу може призвести до великої кількості критично неправильних діагнозів. Щоб зробити повноцінну та правильну діагностику потрібно запросити додаткові дані від користувача, також додати аналітику.

Після аналітики та діагнозу завданням застосунку буде зробити комплексні висновки, методи вирішення проблеми, додаткові дієти, вправи для профілактики та приведення до норми фізичного здоров'я людини. На жаль наразі неможливо дати ШІ або програмі інструкції щодо вирішення ментальних проблем. Тож усі висновки, що будуть зроблені мобільним моніторингом будуть видані користувачу.

1.2 Метод класифікації стану за допомогою гібридних показників

В цьому методі буде використано декілька основних показників, що напряму впливають на загальний стан людини, їх можна проаналізувати за

допомогою формули. Кожен показник буде впливати на кінцеву оцінку окремих сфер здоров'я. Розглянемо адаптовану формулу з використанням гібридних показників:

$$C = f(X, S, A, B, D, FS), \quad (1.1)$$

де X – пульс;

S – загальна тривалість сну;

FS – фази сну та їх тривалість;

A – фізична активність людини (вправи, біг, кроки за день, тренування та ін.);

B – рівень стресу людини (визначається на довгому проміжку часу за допомогою тестів або аналізу крові на гормони, а саме кортизол);

D – дієта людини (кількість калорій, вітамінів, мінералів, БЖВ);

C – загальний стан людини.

Функція f може бути реалізована на основі дерева рішень або штучної нейронної мережі, натренованої на персоналізованих даних користувачів. Такий підхід дає змогу отримувати індивідуальні оцінки стану здоров'я з урахуванням широкого спектра фізіологічних та поведінкових параметрів.

1.3 Метод кластеризації змін стану у часовому ряду

Другий метод – це основний прийом для визначення стану здоров'я на дистанції, його використовують у тренуваннях та лікуванні для визначення динаміки змін, шаблонів поведінки та відхилення від норм людини.

Для кожного користувача формується індивідуальний профіль за принципом формули (1.2), що складається з фізіологічних показників у визначені моменти часу:

$$P = \{p_1, p_2, \dots, p_n\}, p_i = [X_i, S_i, A_i, B_i, D_i, FS_i], \quad (1.2)$$

де X_i – пульс;

S_i – загальна тривалість сну;

A_i – фізична активність людини (вправи, біг, кроки за день, тренування та ін.);

B_i – рівень стресу людини (визначається на довгому проміжку часу за допомогою тестів або аналізу крові на гормони, а саме кортизол);

D_i – дієта людини (кількість калорій, вітамінів, мінералів, БЖВ);

FS_i – фази сну та їх тривалість.

1.3.1 Кластеризація часових сегментів

Для аналізу часових рядів фізіологічних та поведінкових даних користувача застосовуються методи кластеризації, зокрема метод k -середніх та ієрархічне кластеризування. Ці підходи дозволяють ефективно структурувати великі обсяги даних, які надходять з пристроїв моніторингу (наприклад, фітнес-трекерів, смарт-годинників або мобільних застосунків), та виділяти типові стани або патерни поведінки користувача.

Метод k -середніх (k -means) належить до класу жорсткої (flat) кластеризації і дозволяє розбити множину векторів на попередньо задану кількість кластерів k . Його основна перевага полягає в простоті реалізації та високій швидкості роботи, що робить його доцільним для обробки потокових або великомасштабних даних. У цьому контексті кожен кластер може відповідати певному характерному стану користувача (наприклад, активність, помірне навантаження, сон, спокій), а центроїди кластерів – узагальненим представленням типового вектора поведінки у відповідному стані.

Ієрархічне кластеризування (Hierarchical clustering), на відміну від методу k -середніх, не потребує попереднього визначення кількості кластерів. Воно формує ієрархічну структуру у вигляді дендрограми, де кожен рівень

представляє певний ступінь подібності між об'єктами. Цей метод є особливо корисним у випадках, коли важливо відстежити взаємозв'язки між кластерами або коли існує потреба в гнучкому масштабуванні рівня деталізації. Завдяки цьому можливе виявлення як загальних груп станів, так і внутрішньої структури в межах кожної групи.

У результаті кластеризації кожному часовому відрізку t_i призначається мітка стану користувача C_n , що формалізується у вигляді формули (1.3):

$$\text{Cluster}(t_i) \in \{C_1, C_2, C_3, \dots, C_n\}. \quad (1.3)$$

Мітки кластерів служать вхідними даними для моделей станів або аналітики. Вони допомагають оцінювати зміни в режимі життя, виявляти відхилення та створювати персоналізовані профілі здоров'я, аналізуючи, наприклад, рівень стресу, якість сну чи прогрес у фізичній активності.

Крім того, аналіз переходів між станами (між кластерними мітками) дозволяє побудувати матриці переходів або марковські моделі, які описують ймовірності переходів від одного стану до іншого. Це відкриває можливості не лише для описового аналізу, але й для прогнозування подій (наприклад, виснаження, порушення сну чи наближення стресового стану).

Інтеграція кластеризаційних методів у загальну систему оцінки дозволяє здійснювати безперервний моніторинг стану користувача, виявляти довгострокові тенденції та своєчасно формувати рекомендації. З практичного погляду, це дає змогу створити адаптивні інтерфейси для фітнес-застосунків або систем підтримки прийняття рішень у сфері цифрового здоров'я.

1.3.2 Генерація рекомендацій

На основі результатів кластеризації та аналізу ключових фізіологічних і поведінкових показників, система формує персоналізовані рекомендації у

вигляді повідомлень або сповіщень. Такі рекомендації мають на меті не лише інформувати користувача про виявлені відхилення, а й запропонувати конкретні дії для покращення стану здоров'я. Вони ґрунтуються на виявлених закономірностях у даних, що регулярно надходять з сенсорів, та на заздалегідь визначених експертних правилах і моделях. Приклади повідомлень:

– «Замала активність. Потрібно зробити вправу або коротке тренування, щоб уникнути зниження метаболічної активності», що сформовано за допомогою моделі аналізу метаболічної активності;

– «Завелика кількість магнію та недостатня кількість кальцію у вашій дієті. За умов дисбалансу ці два елементи можуть взаємно пригнічувати засвоєння. Рекомендується зменшити споживання червоного м'яса та додати до раціону більше кисломолочних продуктів (наприклад, йогурт або кефір)», що сформовано за допомогою моделі оцінки харчового балансу;

– «За останні 4 дні ваш середній сон тривав від 4 годин 20 хвилин до 5 годин 14 хвилин. Це нижче рекомендованого мінімуму. Спробуйте змінити режим дня, зменшити використання екранів перед сном і лягати раніше», що сформовано за допомогою моделлю аналізу режиму сну;

– «Спостерігається стійке підвищення пульсу в стані спокою – можливий накопичений стрес або перевтома. Рекомендується зробити перерву, знизити інтенсивність тренувань і практикувати техніки релаксації (наприклад, дихальні вправи або медитацію)», що сформовано за допомогою моделі оцінки рівня стресу на основі пульсу в стані спокою;

– «Рівень фізичної активності зріс на 35 % порівняно з попереднім тижнем – це позитивна динаміка. Продовжуйте в тому ж дусі, але слідкуйте за відновленням організму після навантажень», що сформовано за допомогою моделі оцінки динаміки фізичної активності.

Такий підхід трансформує систему з пасивного аналітичного інструмента у динамічного цифрового помічника, що активно бере участь у формуванні здорового способу життя. Рекомендації враховують не лише поточні вимірювання, але й контекст у часовому проміжку (наприклад, зміни

за останній тиждень, порівняння з особистими нормами або статистикою користувача).

Система також може адаптувати рекомендації залежно від профілю користувача: вік, стать, спосіб життя, фізична підготовка, наявні захворювання. В подальшому може бути реалізовано використання моделей машинного навчання для генерації рекомендацій на основі схожих випадків інших користувачів, що суттєво підвищить ефективність порад.

Таким чином, моніторинг не лише пасивно фіксує відхилення, а й виступає як інтерактивний механізм підтримки прийняття рішень, надає зворотний зв'язок у режимі реального часу, допомагає змінювати звички та покращувати якість життя.

Системний підхід до аналізу та рекомендацій дозволяє говорити про створення індивідуалізованого алгоритму управління здоров'ям, що ґрунтується на перевірених даних, медичних знаннях і сучасних цифрових технологіях.

1.4 Метод прогнозування стану організму на основі часових рядів

Одним із ключових напрямів у сфері мобільного моніторингу стану здоров'я є використання методів прогнозування, які дозволяють здійснювати передбачення змін фізіологічних показників користувача на основі аналізу часових рядів [17]. Такий підхід забезпечує не лише фіксацію поточного стану, а й оцінку його динаміки та потенційних ризиків, що є критично важливим для реалізації превентивної медицини. Крім того, інтеграція методів машинного навчання дозволяє підвищити точність прогнозів і персоналізацію рекомендацій. Застосування багатовимірного аналізу даних сприяє комплексній оцінці стану здоров'я, враховуючи взаємозв'язки між різними фізіологічними параметрами.

Завдання прогнозування фізіологічного стану зводиться до побудови моделі, яка на основі історичних даних дозволяє передбачити майбутні значення ключових біометричних параметрів. Враховуючи індивідуальність кожного користувача та специфіку його способу життя, побудова такої моделі повинна враховувати адаптивність, гнучкість і здатність до оновлення з часом.

1.4.1 Формалізація задачі прогнозування

Задача прогнозування формалізується як задача регресії у часовому ряді, де на основі значень фізіологічного показника в попередні моменти часу робиться оцінка його майбутнього стану. Загальна модель прогнозу записується у вигляді формули (1.4):

$$F_{t+1} = f(F_t, F_{t-1}, F_{t-2}, \dots, F_{t-n}), \quad (1.4)$$

де f – це функція прогнозу, апроксимується аналітичною моделлю;

F – значення конкретного фізіологічного показника.;

F_{t+1} – це прогнозоване значення на наступний часовий крок;

n – кількість попередніх часових кроків, які враховуються в моделі.

Перед початком прогнозування виконується попередня обробка даних, нормалізація, обробка пропусків, згладжування та перевірка на стаціонарність (за необхідності). Потім будуються моделі, які враховують як внутрішню структуру часових рядів, так і зовнішні фактори (наприклад, вплив сну на рівень стресу). У сучасній практиці найбільш поширені моделі прогнозування приведено нижче.

ARIMA (Autoregressive Integrated Moving Average) – класичний статистичний підхід, що добре працює з лінійними, стаціонарними часовими рядами. Дозволяє враховувати автокореляції та тренди, проте обмежений у

здатності працювати з нелінійними залежностями або складними шаблонами поведінки.

LSTM – тип рекурентних нейронних мереж, спеціально адаптований до обробки довготривалих залежностей у часових рядах. Модель здатна «запам'ятовувати» важливу інформацію з минулого на великому часовому проміжку, що особливо корисно при роботі з фізіологічними показниками, які мають циклічну або індивідуальну динаміку.

Prophet – прогностична модель, розроблена компанією Meta (Facebook). Зручна для використання з часовими рядами, що мають виражені сезонні коливання, тренди або аномалії. Prophet автоматично враховує календарні ефекти (вихідні, свята), легко інтерпретується та дає змогу інтегрувати зовнішні змінні.

Застосування таких моделей дозволяє не лише передбачити можливі зміни в стані користувача, а й завчасно попереджати про потенційні ризики: хронічну втому, перетренованість, порушення сну, накопичення стресу. В поєднанні з системою рекомендацій це створює ефективний інструмент проактивного моніторингу здоров'я. Прогнозовані значення можуть також бути використані як вхід до інших модулів системи:

- для динамічної зміни кластерних міток;
- для адаптації інтерфейсу застосунку залежно від очікуваного стану користувача;
- для побудови довгострокових індивідуальних планів тренувань чи харчування.

1.4.2 Побудова персоналізованої моделі користувача

Для кожного індивідуального користувача формується багатовимірний часовий ряд, який відображає динаміку ключових фізіологічних та поведінкових параметрів у часі за формулою (1.5):

$$\vec{F}_t = [X_t, S_t, A_t, B_t, D_t, FS_t], \quad (1.5)$$

де X_t – пульс;

S_t – загальна тривалість сну;

A_t – фізична активність людини (вправи, біг, кроки за день, тренування та інша);

B_t – рівень стресу людини (визначається на довгому проміжку часу за допомогою тестів або аналізу крові на гормони, а саме кортизол);

D_t – дієта людини (кількість калорій, вітамінів, мінералів, БЖВ);

FS_t – фази сну та їх тривалість.

В залежності від обраної моделі прогнозування може здійснюватися як одномірний прогноз окремих параметрів, так і синхронний аналіз кількох змінних одночасно з урахуванням їх кореляцій та взаємозалежностей.

Одномірний підхід дозволяє детально прогнозувати конкретний показник, наприклад, пульс чи рівень стресу, що корисно для точного моніторингу окремих аспектів здоров'я.

Мультивимірний підхід використовує методи багатовимірного часово-різного аналізу, такі як багатовимірні рекурентні нейронні мережі (наприклад, багатовимірні LSTM), VAR-моделі (Vector Autoregression) або глибинні моделі, що здатні одночасно моделювати динаміку декількох параметрів і враховувати їх взаємний вплив. Це підвищує якість прогнозів і дає змогу виявляти комплексні патерни, які неможливо побачити при аналізі кожного показника окремо.

Застосування багатовимірного прогнозування особливо важливе для побудови персоналізованих рекомендацій і раннього виявлення відхилень у стані здоров'я, що дозволяє своєчасно коригувати режим дня, харчування або фізичну активність.

Після отримання прогнозованих значень система виконує порівняння з нормативними або індивідуальними межами для конкретного користувача. У разі виявлення можливого наближення до критичних значень генерується

відповідне повідомлення або рекомендація, наприклад: при зниженні тривалості сну – порада щодо корекції режиму. При прогнозі підвищення кортизолу – рекомендація щодо зниження навантаження. При очікуваній перевтомі – нагадування про необхідність відпочинку.

Фізіологічний стан людини залежить від великої кількості змінних, що ми не контролюємо, особливу увагу приділяється адаптивному оновленню моделі. Цей підхід дозволяє системі враховувати індивідуальні особливості користувача, а також динамічні зміни його стану в реальному часі. Завдяки цьому прогноз стає більш точним і релевантним у довгостроковій перспективі. Це означає, що з часом система не лише покращує точність прогнозів, а й автоматично адаптується до нових змінних (зміна графіка, способу життя, зовнішніх факторів). Таким чином, метод прогнозування стану організму на основі часових рядів відіграє велику роль у переході від реактивного до проактивного підходу у цифровому здоров'ї, сприяючи вчасному виявленню проблем та підвищуючи якість життя користувача за рахунок персоналізованої цифрової підтримки.

1.5 Огляд існуючих цифрових рішень

У сучасному світі розвитку цифрових технологій активно впроваджуються мобільні додатки для вирішення проблем з низькою фізичною активністю, відсутністю правильної дієти та контролю показників стану людини. Значна частина рішень – орієнтована на людей, що хочуть почати вести правильний образ життя, контролювати свою вагу, показники фізичного стану, якість сну. У цьому розділі розглядаються 3 популярні застосунки, що мають однаковий вплив на покращення різних елементів здоров'я користувача. Ці застосунки використовують мітки кластерів для персоналізованих рекомендацій щодо покращення здоров'я та моніторингу прогресу.

1.5.1 Застосунок Goggle Fit

Goggle Fit – найпопулярніший застосунок, що вирішує проблеми для відстеження загального стану користувача, зображено на рисунку 1.1. Застосунок було розроблено в жовтні 2014 році компанією Google, основною метою було створити аналог програми Apple Health, що вийшов місяцем раніше, для системи Android. На разі Google Fit постійно оновлюється та додає підтримку нових пристроїв та методів аналізу. Основними функціями в цьому застосунку є:

- розрахунок кількості кроків, пройденої відстані, середньої швидкості та часу активності;
- інтеграція з іншими застосунками для збору та аналізу даних;
- відстеження основних показників, таких як пульс, рівень кисню, рівень стресу та інших, завдяки додатковим пристроям.

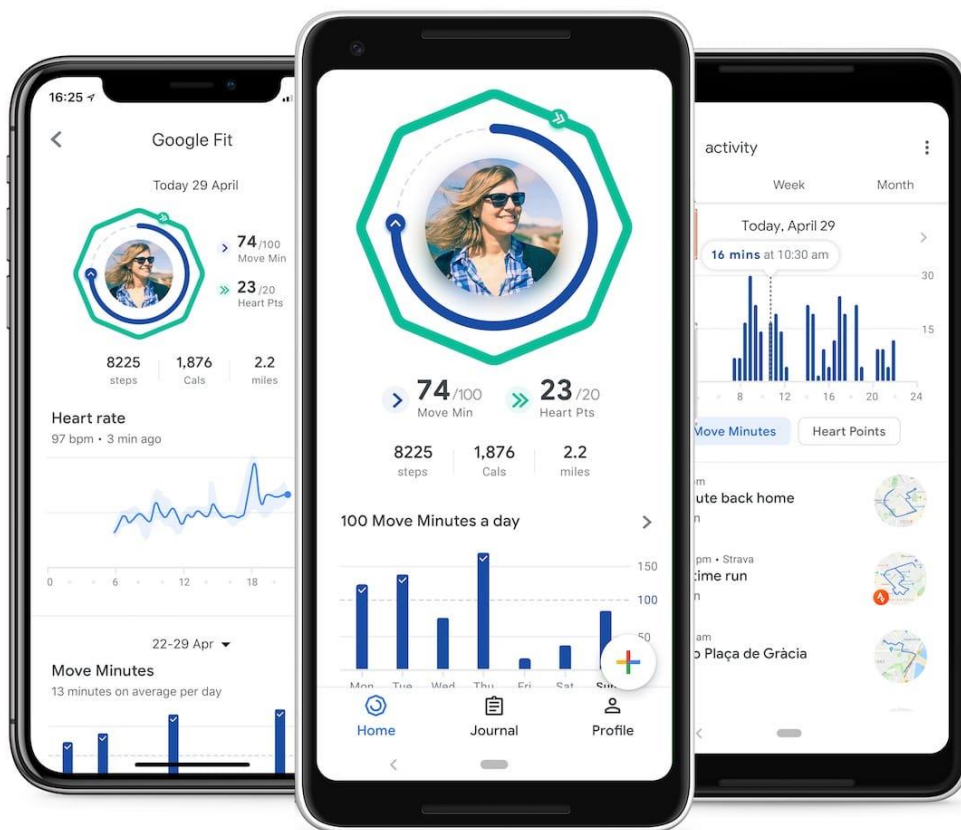


Рисунок 1.1 – Застосунок Goggle Fit

Застосунок базується на часі активності та «heart points», при аналітиці дає користувачу дані про обсяг активності та її інтенсивність. Одним з найбільших переваг над іншими аналогами є підтримка великої кількості сторонніх пристроїв, що відкриває великі можливості для аналітики стану людини. Слід відзначити, що у Google Fit є два великих недоліки – аналітична здібність дуже низька в порівнянні з іншими специфікованими застосунками, також варто відзначити, що через випуск компанією Google, залежність від її екосистеми значна, що впливає на кількість пристроїв, що будуть взаємодіяти з програмою.

1.5.2 Застосунок Apple Health

Застосунок Apple Health – це інтегроване рішення для збору, обробки та візуалізації лікарських та фітнес-даних, розроблене компанією Apple та вперше відкрито в 2014 році, зображено на рисунку 1.2.

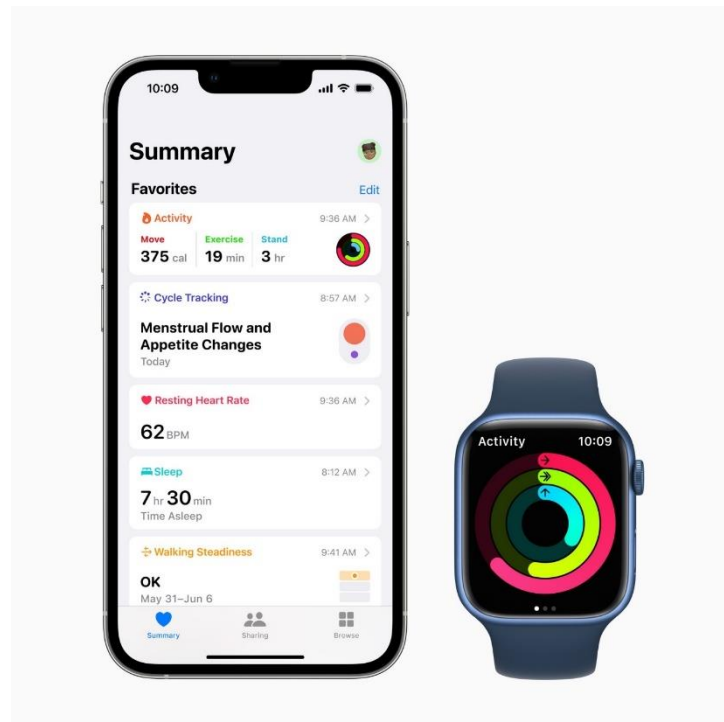


Рисунок 1.2 – Застосунок Apple Health

Основні функції застосунку Apple Health:

- відстеження фізичної активності (кроки, відстань, калорії, час активності);
- аналіз сну, частоти серцевого ритму, рівня кисню в крові, варіабельності серцевого ритму;
- лідерство у веденні журналів харчування, споживання води, ваги, менструального циклу;
- підтримка сервісу «Спільний доступ до даних», що передає інформацію лікарю або членам сім'ї.

Apple Health відрізняється високим рівнем персоналізації, деталізацією даних і конфіденційністю. Дані користувача зберігати на пристрої в зашифрованому вигляді. Переважною вартістю є замкнутість еко-системи – застосунок доступний тільки для користувачів пристроїв Apple, що обмежує використання.

1.5.3 Застосунок Fatsecret

FatSecret – сторонній застосунок і інтернет-платформа харчової групи керування. Він розрахований на таких осіб, що приділяють належну увагу зменшенню або регулюванню свого вагового рівня, дотриманню раціонального харчування, і поєднанню дієтики з фізичними навантаженнями.

Застосунок FatSecret вирізняється простотою використання, наявністю веб-версії та підтримкою кількох мов. Крім того, сервіс доступний безкоштовно, хоча існують і преміум-версії з розширеним функціоналом.

До недоліків слід віднести наявність реклами у безкоштовній версії та відсутність мікромоніторингу медичних показників, як у Google Fit або Apple Health.

За результатами аналізу кожне з розглянутих цифрових рішень має свої сильні сторони та цільову аудиторію. Google Fit орієнтований на користувачів

Android і забезпечує базовий моніторинг активності. Apple Health пропонує розширений інтегрований контроль за здоров'ям, проте виключно для власників пристроїв Apple. FatSecret спеціалізується на контролі харчування, підтримці дієти та надає корисні інструменти для досягнення цілей зі зменшення або підтримки маси тіла.

1.6 Постановка задачі

Об'єктом дослідження є розробка мобільного застосунку для моніторингу та контролю фізичних показників організму з метою підтримки здорового способу життя. Актуальність теми обумовлена зростаючим інтересом до особистого здоров'я, розвитком фітнес-індустрії та потребою впровадження цифрових технологій у повсякденну практику користувачів.

У сучасному світі ритм життя постійно прискорюється, а кількість хронічних захворювань, пов'язаних із малорухливістю, незбалансованим харчуванням і стресами, невпинно зростає, що створює нові виклики як для окремих осіб, так і для систем охорони здоров'я загалом. У цьому контексті виникає нагальна потреба у розробці мобільних рішень для раннього виявлення фізичних порушень та реалізації профілактичних заходів. Зокрема, значну увагу привертає можливість вчасного реагування на зміни фізіологічного стану, що раніше залишалися поза увагою через брак об'єктивних даних у режимі реального часу. Традиційний медичний контроль, що базується на періодичних консультаціях лікарів, обмежений за швидкістю реагування та вимагає значних часових витрат, тоді як безперервний моніторинг із застосуванням мобільних застосунків здатен суттєво підвищити ефективність профілактики. Таким чином, інтеграція цифрових технологій у сферу самоспостереження відкриває нові можливості для індивідуалізованого підходу до підтримки здоров'я.

Мобільні застосунки забезпечують безперервний збір даних із сенсорних пристроїв, автоматичну обробку інформації та динамічну оцінку фізичного стану користувача. Це дає змогу не лише відстежувати ключові показники, а й попереджати про потенційні ризики ще до появи виражених симптомів. Водночас ефективність таких систем залежить від точності збору даних, їх обсягу, а також якості алгоритмів інтерпретації, здатних генерувати персоналізовані рекомендації.

Метою цієї роботи є розробка мобільного застосунку, здатного здійснювати моніторинг фізичних даних показників користувача з майбутнім аналізом стану організму та формуванням індивідуальних рекомендацій, а саме дієт, фізичних вправ, форми нестандартного відпочинку, профілактичних засобів.

Для досягнення цієї мети необхідно вирішити наступні задачі:

- здійснити детальну актуальну аналітику технологій, а саме: мобільного моніторингу, медичних сенсорів та методи аналізу медичних даних;
- обрати перелік фізичних параметрів організму для найінформативнішої оцінки стану організму, з урахуванням доступності їх вимірювання за допомогою пристроїв, розробити моделі первісної обробки та фільтру даних, спрямовані на видалення пропусків та артефактів, що трапляються на приладах з сенсорами постійно;
- дослідити та реалізувати алгоритми класифікації стану здоров'я на основі аналізу індивідуальних профілів користувачів та їх часових рядів;
- створити систему, яка буде формувати рекомендації на результатах класифікації, значеннях показників та минулих даних користувача, реалізувати прототип мобільного застосунку з графічним інтерфейсом, здатний візуалізувати усі дані інформувати користувача про стан організму та пропонувати корисні дії;

2 МОБІЛЬНИЙ МОНІТОРИНГ ЗДОРОВ'Я: АНАЛІЗ ДАНИХ

2.1 Формалізація методів аналізу у моніторингу здоров'я

Нехай $H = \{h_1, h_2, \dots, h_n\}$ – множина фізіологічних показників, що надходять до мобільного застосунку. Кожен показник h_n – часовий ряд з певного інтервалу спостереження. Ці дані будуть включати пульс, кількість кроків, можливо температуру тіла, рівень кисню в крові.

Для кожного з фізіологічних показників треба взяти вектор. І визначати середнє добове, стандартне відхилення, максимум та мінімум. Далі зображено формулу (2.1) узагальненого вектору стану користувача:

$$V = \bigcup_{i=1}^m X_i. \quad (2.1)$$

Ця формула дає загальний опис фізіологічного стану людини в момент часу t .

Метод аналізу полягає у віднесенні вектора стану до одного з наперед визначених класів самопочуття, можна записати у вигляді такої формули (2.2):

$$K = \{k_1, k_2, \dots, k_n\}, \quad (2.2)$$

де кожен k_n – це певний стан організму, а саме: норма, сон, стрес, підвищена активність і інші.

Для розпізнавання класу проводиться класифікаційна функція (2.3):

$$f : V \rightarrow K. \quad (2.3)$$

Кожен вектор V порівнюється з V_k для кожного показника, після чого буде порівнюватися з двома найближчими сусідами, якщо такі є за формулою (2.4):

$$f(V) = \arg \min_{k \in K} \rho(V, V_k), \quad (2.4)$$

де ρ – евклідова відстань між векторами V та V_k ;

K – множина індексів векторів.

Для класифікації найближчого центроїда використовується функція, яку можна записати формулою (2.5):

$$f(V) = \arg \min_i \min_{c \in C} \rho(V, c), \quad (2.5)$$

де V – вхідний вектор ознак;

c – множина центроїдів класу;

ρ – евклідова метрика;

i – індекс кластеру, до якого належить найближчий центроїд c .

Завдяки використанню формаційних методів аналізу, усі зібрані дані на постійній основі перетворюються на біологічно-медичні вектори. Ці вектори порівнюються з нормативними значеннями та забезпечують формування результату для користувача.

Такий підхід є оптимальним з огляду на баланс між швидкістю обчислень і точністю оцінювання стану користувача.

2.2 Використання евклідової відстані для оцінки схожості

Евклідова відстань є однією з найпоширеніших і базових метрик для оцінки схожості між векторами ознак. Вона широко використовується у задачах класифікації, кластеризації та виявлення аномалій. Основною перевагою евклідової метрики є простота реалізації та швидкість обчислення, що особливо важливо для систем реального часу. У випадку біомедичних даних вона дозволяє точно визначити ступінь відхилення фізіологічних

параметрів від норми. Це забезпечує своєчасне виявлення потенційних ризиків або патологій у стані користувача. Метрика найкраще працює з нормалізованими числовими даними, тому попередня обробка є обов'язковою. Завдяки своїй ефективності евклідова відстань є надійним інструментом у медико-аналітичних системах. Її використання забезпечує оптимальний баланс між точністю оцінювання та обчислювальною ефективністю в практичних застосуваннях.

2.2.1 Формалізація евклідової метрики

Евклідова відстань між фізичними ознаками (пульс, температура, рівень кисню, рівень стресу, тощо) між двома векторними величинами, які були розглянуті в розділі 2.1 можна записати у вигляді формули (2.6):

$$\rho E(V, C) = \sqrt{\sum_{i=1}^n (v_i - c_i)^2}. \quad (2.6)$$

Цей підхід передбачає, що всі ознаки однаково важливі та чисельно порівнювані, тобто вектори повинні бути попередньо нормалізовані, щоб виключити вплив різних шкал вимірювання. Тому необхідно задати коефіцієнт впливу для кожної з ознак.

Евклідова метрика є однією з найпростіших та водночас найуживаніших метрик у задачах машинного навчання, статистичного аналізу та біомедичних досліджень. Вона має ряд переваг, які зумовлюють її широку популярність. Насамперед, це інтуїтивно зрозуміла геометрична інтерпретація – відстань між двома точками в багатовимірному просторі. Це забезпечує легкість у візуалізації результатів та швидку побудову моделей. Крім того, евклідова відстань є простою для обчислень, що дозволяє її ефективно використовувати

для великих обсягів даних у режимі реального часу. Водночас евклідова метрика чутлива до викидів у даних.

Зокрема, в контексті медико-біологічного аналізу, де дані часто представлені у вигляді числових векторів фізіологічних показників (температура тіла, частота серцевих скорочень, тиск, рівень кисню тощо), евклідова метрика дозволяє швидко та наочно оцінити ступінь відхилення від нормативних значень.

Однак попри свою простоту, евклідова відстань має ряд суттєвих недоліків, які обов'язково слід враховувати при її застосуванні. Найголовніша проблема полягає в її чутливості до масштабу ознак. Різні параметри організму мають різні діапазони змін. Наприклад, температура тіла людини зазвичай варіюється в межах 36–38 °С, тобто зміна становить лише кілька одиниць, тоді як пульс може коливатися в межах від 40 до 160 ударів на хвилину, що дає значно більшу варіативність. У такій ситуації пульс «домінуватиме» над температурою у розрахунках, навіть якщо відхилення температури є клінічно більш значущим.

Це може призвести до викривлення результатів, коли важливі, але менш варіативні ознаки будуть ігноруватись або мати занижену вагу. Ще один аспект, а саме евклідова відстань, передбачає, що всі ознаки незалежні між собою та мають однакову вагомість, що не завжди відповідає реальним біомедичним даним. Тому застосування цієї метрики без відповідної попередньої обробки даних може знизити точність моделі та призвести до хибних висновків.

Для того щоб евклідова метрика працювала коректно, необхідно виконувати нормалізацію або стандартизацію вхідних даних, зводячи їх до одного масштабу. У деяких випадках також доцільно застосовувати вагову модифікацію евклідової відстані, де ознакам призначаються коефіцієнти важливості відповідно до їхнього клінічного значення.

Отже, евклідова метрика має значні переваги, але її ефективне використання потребує відповідального підходу до підготовки даних. За

умови правильної обробки вона залишається одним з найоптимальніших інструментів у задачах оцінювання схожості у медичних та біоінформаційних системах.

У мобільних застосунках моніторингу здоров'я евклідова метрика використовується: для миттєвої оцінки відхилення від нормальних станів, як критерій при класифікації стану користувача на нормальний, невеликі відхилення від норми, в зоні ризику, критичний, у процесі кластеризації історичних даних для виявлення повторюваних шаблонів поведінки організму.

2.3 Інтеграція моделей класифікації у мобільні системи моніторингу

Для створення адаптивних моделей, здатних класифікувати фізіологічні стани користувача на основі даних, що надходять у режимі реального часу та ефективна інтеграція в точної класифікаційної моделі обчислювально обмежене середовище мобільного пристрою. Потрібно обрати найліпшу типову модель у цьому завданні.

Критерії для вибору методу класифікації включають такі основні характеристики: персоналізація, стійкість до пошкоджених даних, споживання ресурсів, швидкість. Ці вимоги об'єднані узагальнюючим поняттям ефективності в умовах мобільного середовища, де кожен із зазначених критеріїв відіграє визначальну роль для стабільної та корисної роботи програмного забезпечення в реальному часі.

Швидкість – один з найважливіших чинників, адже саме від неї залежить оперативність отримання результату. Чим швидше буде здійснено класифікацію фізіологічного стану, тим скоріше користувач отримає рекомендацію або попередження, а відтак і можливість вчасно відреагувати. Метод, що використовується, має забезпечувати майже миттєву реакцію на нові вхідні дані.

Споживання ресурсів також набуває особливої актуальності у контексті мобільних пристроїв, різних за технічними характеристиками. Обрана модель повинна працювати ефективно навіть за обмежених апаратних можливостей, не перевантажувати центральний процесор, демонструвати низький рівень енергоспоживання та підтримувати автономну роботу без постійного підключення до мережі інтернет.

Стійкість до пошкоджених даних є ключовою умовою застосування класифікації в реальних умовах, де сенсори часто генерують нечіткі або спотворені значення. Наприклад, навіть незначне зміщення фітнес-браслета під час фізичної активності призводить до помилок у вимірюваннях пульсу. У таких випадках система повинна або адаптивно фільтрувати ці аномальні значення, або бути здатною інтерпретувати їх без суттєвого впливу на кінцевий результат.

Персоналізація вимагає, щоб модель класифікації могла адаптуватися до індивідуальних особливостей кожного користувача, враховуючи як сталі фізіологічні параметри, так і поведінкові патерни. Така адаптація може досягатися як шляхом інтерактивного зворотного зв'язку, так і за допомогою механізмів поступового навчання моделі на нових даних.

У мобільному застосунку можливе використання різноманітних моделей машинного навчання, зокрема логістичної регресії, дерева рішень, методу k -найближчих сусідів (k -NN) та нейронних мереж [17]. Усі ці підходи мають свої переваги й недоліки, що обумовлює необхідність обґрунтованого вибору відповідно до умов застосування.

Метод k -найближчих сусідів є доцільним у разі кластеризації даних, адже він відзначається простотою реалізації в мобільному середовищі, не потребує збереження навчального набору після початкової підготовки, здатен обробляти великий обсяг різноманітних фізіологічних ознак та демонструє хорошу точність класифікації завдяки використанню евклідової метрики. Разом із тим, метод має певні обмеження: втрачає ефективність при обслуговуванні великої кількості користувачів, потребує значних обсягів

оперативної пам'яті за наявності тривалої історії даних і може накопичувати застарілу або нерелевантну інформацію.

Нейронні мережі становлять один із найбільш перспективних напрямів у галузі машинного навчання, оскільки дозволяють моделювати складні нелінійні залежності між вхідними даними та результатами. Їх основними перевагами є здатність виявляти приховані закономірності в багатовимірних і гетерогенних даних, гнучкість щодо типів вхідної інформації (включно з числовими значеннями, часовими рядами та зображеннями), а також можливість адаптації до індивідуальних особливостей користувача. Однак до недоліків слід віднести низьку інтерпретованість результатів, вимогу до великих обсягів якісних навчальних даних, високі обчислювальні витрати та складність налаштування й тестування моделей, що створює виклики для їх використання у мобільному середовищі з обмеженими ресурсами.

Інші типи класифікаційних моделей мають обмежену сферу застосування через неспроможність працювати з повним спектром фізіологічних даних. Саме тому нейронні мережі виявляються найбільш перспективними для задач, де потрібна глибока індивідуалізація та адаптивність.

Загальний процес аналізу фізіологічних даних у мобільному застосунку включає кілька ключових етапів: збір даних із сенсорів або введення вручну; обробку й очищення сигналів від шумів; формування вектора ознак на основі нормалізованих параметрів; класифікацію стану з використанням вибраної моделі; та, нарешті, інтерпретацію результату у формі повідомлень або рекомендацій для користувача.

Таким чином, вибір моделі класифікації має ґрунтуватися на сукупності технічних характеристик, що враховують обчислювальні ресурси, інтерпретованість результатів, можливість персоналізації та адаптації, а також стійкість до похибок вхідних даних.

2.4 Інтеграція кластеризації для аналізу стану здоров'я

У мобільному моніторингу стану здоров'я кластеризація відіграє ключову роль. Принцип її роботи полягає в групуванні схожих за характеристиками об'єктів у кластери, що суттєво зменшує обсяг обчислень на етапі класифікації. Завдяки цьому підвищується загальна продуктивність та ефективність моделі.

Кластеризація дає змогу формувати групи фізіологічних станів на основі типових патернів, які повторюються у конкретного користувача. Наприклад, можна виділити такі групи, як стабільний стан, легке навантаження, перевтома, порушення сну та інші.

Таким чином, замість роботи з великою кількістю індивідуальних вхідних даних система оперує середніми представниками кластерів – центроїдами. Це значно спрощує процес аналізу і знижує обчислювальні витрати, що особливо важливо для мобільних застосунків з обмеженими ресурсами.

2.4.1 Приклади алгоритмів кластеризації

На рисунку 2.1 наведено приклади найбільш поширених алгоритмів кластеризації. Кожен із цих методів має свої особливості, які роблять його більш або менш придатним для різних типів даних та завдань. Вибір алгоритму залежить від структури даних, розміру вибірки та вимог до точності і швидкості обробки. Варто враховувати, що в контексті практичного застосування важливу роль відіграють не лише теоретичні характеристики алгоритму, але й його здатність до масштабування та адаптації у динамічних середовищах. Алгоритми кластеризації використовуються для групування схожих об'єктів у кластери, що допомагає спростити аналіз складних даних. Крім того, вони дають змогу виділити закономірності, які важко помітити при

розгляді індивідуальних елементів. Правильний вибір алгоритму забезпечує більш ефективну роботу системи моніторингу, знижує навантаження на обчислювальні ресурси та підвищує якість кінцевих результатів. Також алгоритми кластеризації можуть адаптуватися до змін у даних, що покращує їхню стійкість до шумів та аномалій. Це сприяє більш точному та надійному відображенню стану користувача. Загалом, кластеризація є невід’ємною складовою сучасних систем аналізу даних у медицині, де від своєчасності та точності обробки даних часто залежить ефективність прийняття рішень у процесі лікування. У багатьох випадках вона використовується як базова технологія для побудови складніших моделей прогнозування, зокрема в системах підтримки прийняття рішень, що набуває особливої актуальності в умовах розвитку телемедицини.

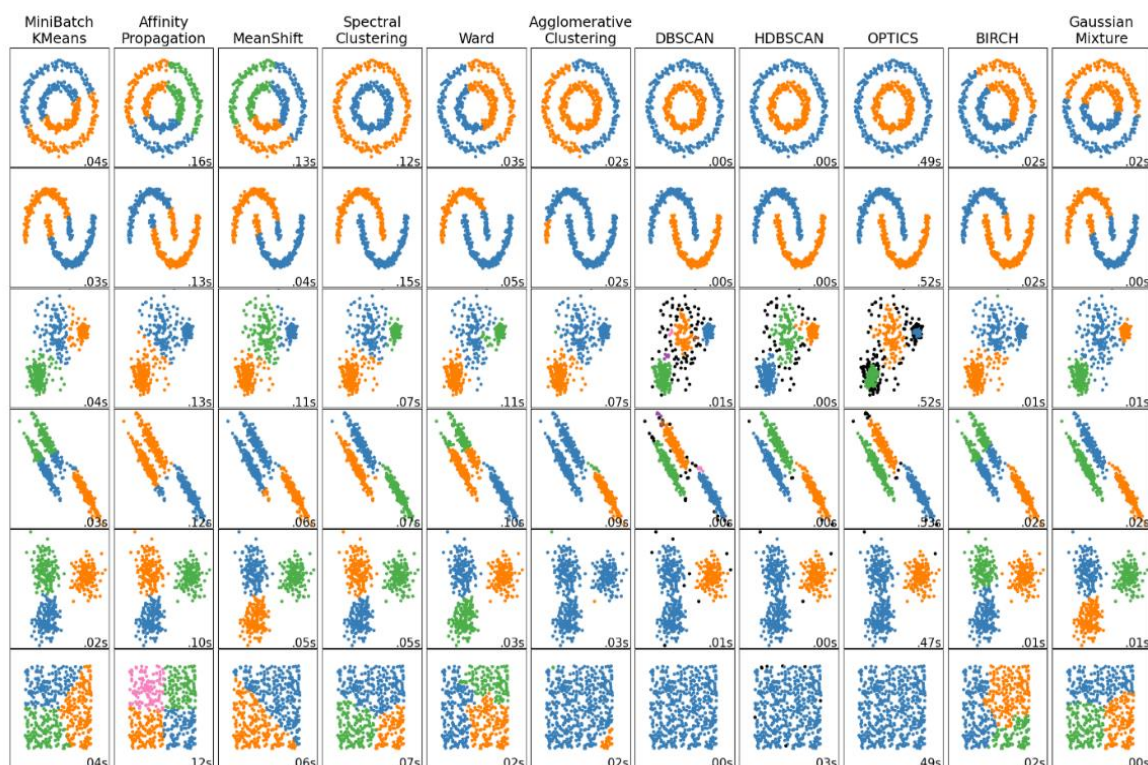


Рисунок 2.1 – Приклади алгоритмів кластеризації

Серед алгоритмів кластеризації найоптимальнішими в контексті аналізу фізіологічних даних вважаються k-NN, DBSCAN і Mean-shift. Метод k-NN

(рис. 2.3) забезпечує швидке групування даних за ознаками у фіксовану кількість кластерів і є доцільним для задач із чітко визначеною структурою. Він широко застосовується у практичних задачах завдяки своїй простоті та надійності. Алгоритм DBSCAN (рис. 2.2) демонструє ефективність у виявленні аномалій або нерівномірно розподілених кластерів, зокрема патологічних відхилень, що робить його корисним для медичних досліджень та моніторингу стану пацієнтів. У свою чергу, Mean-shift (рис. 2.4) не вимагає попереднього задання кількості груп, що робить його особливо зручним для персоналізованих моделей з динамічно змінною структурою даних, а також дає можливість адаптуватися до нових патернів у даних. Кожен із цих алгоритмів має свої переваги та обмеження, а також зони, у яких вони будуть менш ефективними і можуть створювати аномальні кластери з неправильними даними. Тому важливо враховувати особливості кожного методу при виборі для конкретного завдання.

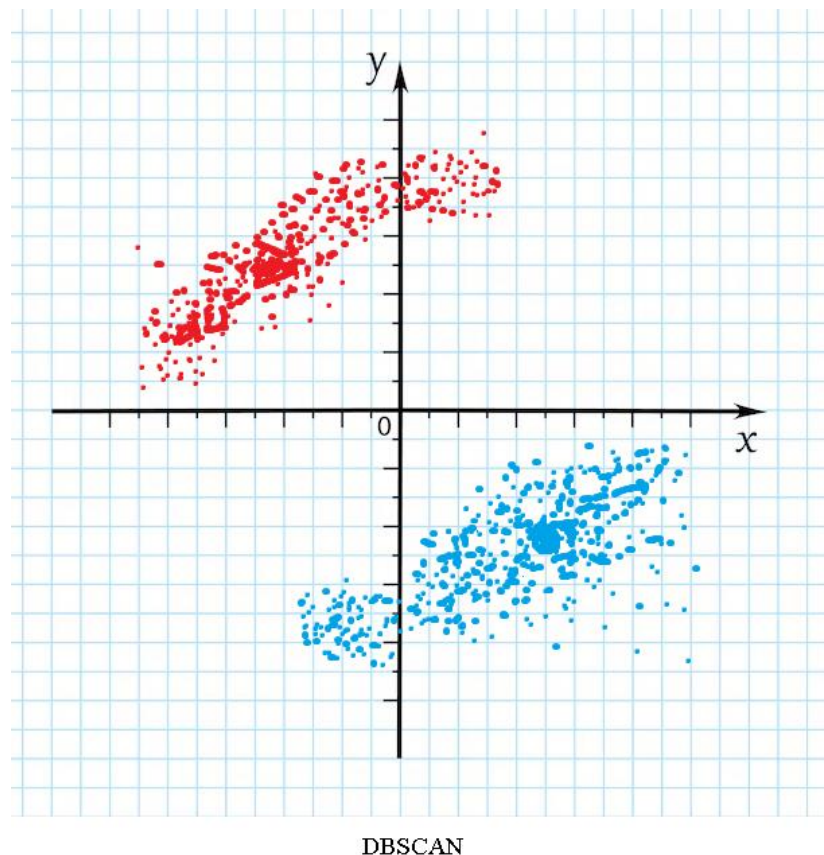


Рисунок 2.2 – Метод кластеризації DBSCAN

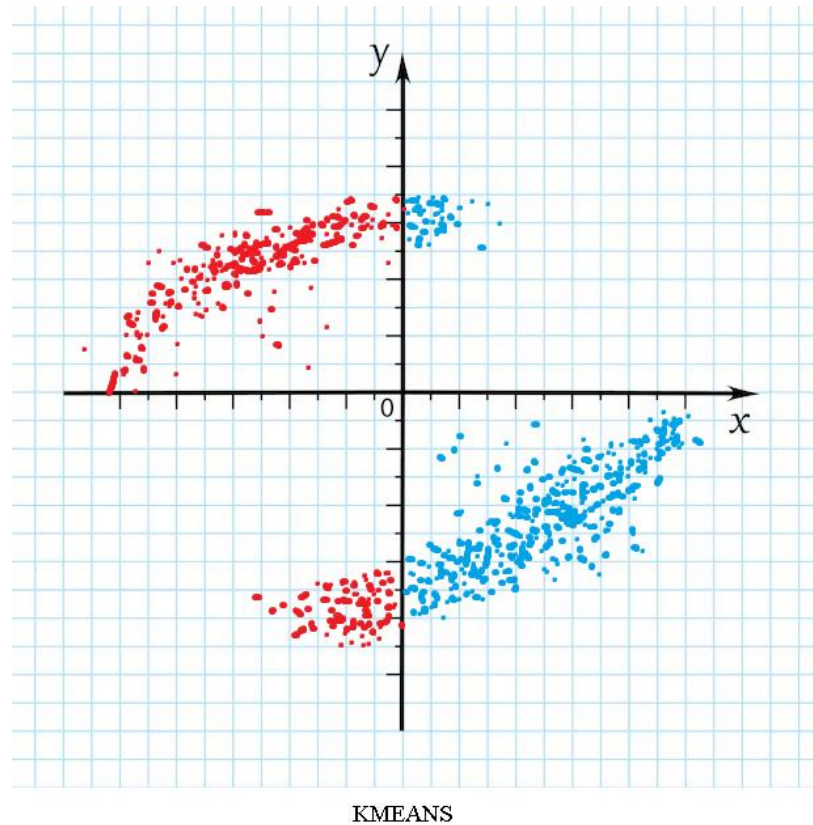


Рисунок 2.3 – Метод кластеризації к-середніх

Метод кластеризації к-середніх (k-means) є одним із найпопулярніших алгоритмів через свою простоту та ефективність у розподілі даних на задану кількість кластерів. Він працює шляхом ітеративного знаходження центрів кластерів – центроїдів та розподілу точок даних за найближчим центроїдом. Проте цей метод має обмеження, особливо у випадках, коли дані містять аномалії або шуми. Аномалії можуть спотворювати положення центроїдів, що призводить до неточного розподілу даних та зниження якості кластеризації. Щоб виправити ці недоліки, використовують попередню обробку даних, зокрема фільтрацію шумів і видалення викидів. Також застосовують більш стійкі до аномалій методи кластеризації або комбінують к-середніх з алгоритмами, які виявляють аномалії, наприклад DBSCAN. Крім того, можна використовувати адаптивні алгоритми, що автоматично коригують кількість кластерів або ваги точок, щоб зменшити вплив аномалій. Таким чином,

правильна обробка аномалій дозволяє підвищити точність і стабільність результатів кластеризації методом к-середніх.

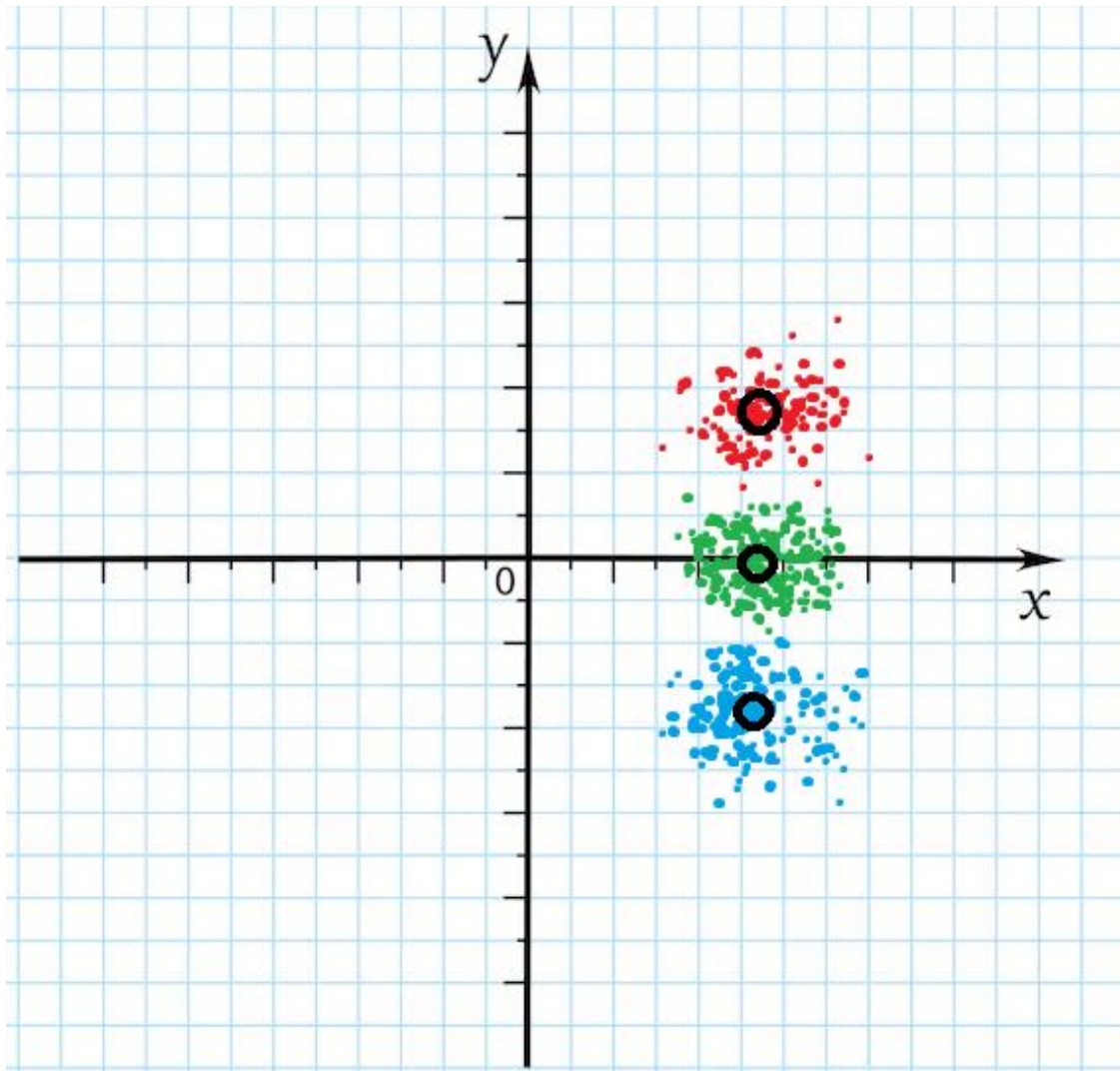


Рисунок 2.4 – Кластеризація методом Mean-Shift

Перевагами кластеризації є оптимізація процесу аналізу, оскільки обробка виконується лише для представників кластерів, що значно зменшує обчислювальні витрати і сприяє підвищенню продуктивності системи. Узагальнення даних підвищує точність моделі, знижуючи вплив випадкових помилок та шумів у вхідних даних, що особливо важливо у складних біомедичних задачах. Це дозволяє зробити аналіз більш стабільним та надійним, зменшуючи ризик помилкових висновків. Крім того, кластеризація сприяє ефективному виявленню закономірностей у великих масивах даних, що полегшує прогнозування змін у стані здоров'я. Такий підхід також забезпечує

швидшу обробку інформації, що критично для оперативного реагування в медичних системах.

Кластеризація спрощує аналіз даних, адже обробляються лише представники кластерів, що економить ресурси, прискорює роботу системи та підвищує її продуктивність. Завдяки узагальненню даних моделі стають точнішими, адже зменшується вплив шумів і випадкових похибок, що особливо важливо для біомедичних задач, де надійність аналізу критично необхідна для уникнення хибних висновків. Кластеризація дає змогу створювати персоналізовані профілі – кожен користувач отримує індивідуальний шаблон, що ідеально підходить для тривалого моніторингу здоров'я чи адаптації до змін у способі життя. Це забезпечує гнучкість системи, підвищує якість рекомендацій і зміцнює довіру користувачів, які бачать, що система враховує їхні особливості.

Однак кластеризація має й недоліки. Вона чутлива до вибору параметрів, як-от кількості кластерів у методі k-середніх: якщо налаштування некоректні, поділ даних може бути неточним, а результати – ненадійними. Надмірне узагальнення загрожує втратою унікальних ознак чи рідкісних патологій, що в медицині може призвести до помилкових діагнозів і знизити довіру до системи. Для коректної роботи потрібна ретельна підготовка даних – нормалізація та стандартизація, без яких моделі можуть давати збої.

Кластеризація ефективна під час оброблення історичних даних користувача та для ініціалізації моделей прогнозування й діагностики. Це підвищує ефективність мобільних застосунків і дозволяє надавати рекомендації, що враховують біоритми, індивідуальні особливості та відхилення. Як результат, покращується якість медичного супроводу, користувачі краще контролюють своє здоров'я, харчування, сон і фізичну активність. Застосунки з кластеризацією діють як персональний помічник, який аналізує дані та пропонує поради для здоровішого життя.

3 ПРАКТИЧНА РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ МОНІТОРИНГУ СТАНУ ОРГАНІЗМУ З НАДАННЯМ РЕКОМЕНДАЦІЙ

3.1 Розробка мобільних застосунків для Android за допомогою мови програмування Python

Розробка мобільних застосунків для Android – складний багатовимірний процес, що охоплює архітектурні, функціональні, візуальні та інженерні аспекти. Особлива увага приділяється використанню альтернативних мов програмування, таких як Python, фреймворків Kivy і KivyMD, бібліотеки Matplotlib для візуалізації та систем керування базами даних MySQL і SQLite [18–23]. Ці технології дозволяють створювати ефективні та масштабовані рішення, які легко адаптуються під різні потреби користувачів.

Такий підхід поєднує швидкість розробки, масштабованість, легкість обслуговування та гнучку архітектуру з високою продуктивністю і привабливим інтерфейсом. Це особливо важливо для команд з обмеженими ресурсами, що прагнуть швидко отримати якісний результат без втрати користувацького досвіду.

Серед недоліків – принцип модульності, коли програма поділяється на незалежні функціональні модулі: інтерфейс, бізнес-логіку, обробку даних, взаємодію з базами, генерацію графіків і мережеві запити. Така структура вимагає чіткої координації розробників і дотримання єдиних стандартів кодування [24–27]. Це підвищує складність управління проектом, але водночас забезпечує гнучкість і можливість паралельної розробки різних компонентів системи.

Кожен модуль пишеться, тестується та оптимізується окремо, що дозволяє зменшити кількість помилок при масштабуванні або оновленні програми. У проєктах, заснованих на Kivu, структура зазвичай така:

- .ру-файли з бізнес-логікою;
- .kv-файли, де описано інтерфейс;
- інші обробні модулі даних, візуалізації або синхронізації.

Цей метод забезпечує простоту масштабування, зручну підтримку та швидкий розвиток завдяки автономності компонентів. Нові учасники команди можуть швидко орієнтуватися в структурі проєкту та вносити зміни без ризику порушення роботи застосунку.

Використання Kivu та KivuMD підтримує кросплатформну розробку: код для Android із мінімальними змінами можна застосувати на Windows, Linux, macOS і частково iOS [23]. Це значно знижує витрати часу та ресурсів на створення окремих програм для різних ОС і особливо корисно при розробці прототипів або MVP для швидкого тестування ідей.

Бібліотека KivuMD реалізує інтерфейс за стандартами Material Design для Android, що забезпечує користувачу зручну навігацію та приємний візуальний досвід. Інтерфейс адаптується до різних розмірів екранів – від смартфонів до планшетів – завдяки відсотковому визначенню розмірів (`size_hint`) та автоматичним відступам (`padding`, `spacing`), що гарантує коректне масштабування.

Декларативна мова .kv відокремлює макет інтерфейсу від логіки програми, спрощуючи розробку, повторне використання елементів, локалізацію і тестування.

Python як високорівнева мова програмування з великою кількістю бібліотек дозволяє швидко реалізувати широкий спектр функціональностей, що є критично важливим у розробці мобільних застосунків. Основні можливості включають:

- обробку вхідних даних з датчиків або форм ручного введення;
- виконання математичних та статистичних обчислень;

- перевірку вірності та цілісності отриманих даних;
- синхронізацію з базами даних через RESTful API;
- побудову діаграм і графіків у реальному часі для візуалізації результатів.

Для зберігання та обробки даних використовуються такі системи:

- SQLite – легка локальна база даних, яка забезпечує автономну роботу без постійного підключення до Інтернету;
- MySQL – потужна серверна база даних, яка дозволяє синхронізувати інформацію між декількома пристроями або користувачами.

Серверна комунікація здійснюється за допомогою протоколів HTTP/HTTPS. Python-бібліотеки requests, httpx або aiohttp забезпечують синхронну та асинхронну передачу даних у форматі JSON або XML. Це дозволяє ефективно налаштувати обмін інформацією в умовах змінного мережевого покриття. Інтеграція бібліотеки Matplotlib дає змогу реалізувати графічну інтерпретацію даних безпосередньо в інтерфейсі користувача. Вона дозволяє створювати:

- лінійні графіки, гистограми та кругові діаграми для представлення змін;
- комбіновані діаграми з кількох показників;
- часові ряди, що особливо корисні для медичних застосунків, які відображають динаміку фізіологічних параметрів.

Візуальне представлення результатів сприяє кращому розумінню стану користувача та прийняттю обґрунтованих рішень на основі аналітики. Мобільні пристрої мають обмежені ресурси, тому оптимізація продуктивності є критично важливою. Ефективні стратегії включають:

- мінімізацію кількості запитів до сервера;
- локальне кешування даних;
- зменшення частоти оновлення інтерфейсу;
- використання асинхронного виконання задач;

- очищення пам'яті від зайвих ресурсів;
- уникнення глибоко вкладених віджетів у графічному інтерфейсі.

Ще одним важливим аспектом є підтримка, масштабування та розширення функціональності застосунку. Добре структурований код, що базується на шаблонах архітектури, таких як MVC або MVVM, дозволяє швидко адаптуватися до нових вимог, оновлень Android та впроваджувати нові модулі з мінімальними витратами.

Інтерфейс користувача (UI) та взаємодія з ним (UX) мають бути інтуїтивно зрозумілими, адаптивними до різних розмірів екранів і відповідати сучасним вимогам доступності. Kivy та KivyMD забезпечують ці вимоги за допомогою гнучкої системи компоновання, відносного масштабування (`size_hint`) та підтримки адаптивного дизайну відповідно до принципів Material Design. Попередньо можна привести таку діаграму (рис. 3.1), яка ілюструє загальну архітектуру мобільного застосунку, побудованого з використанням Python, фреймворків Kivy/KivyMD, та інструментів для обробки, збереження і візуалізації даних:

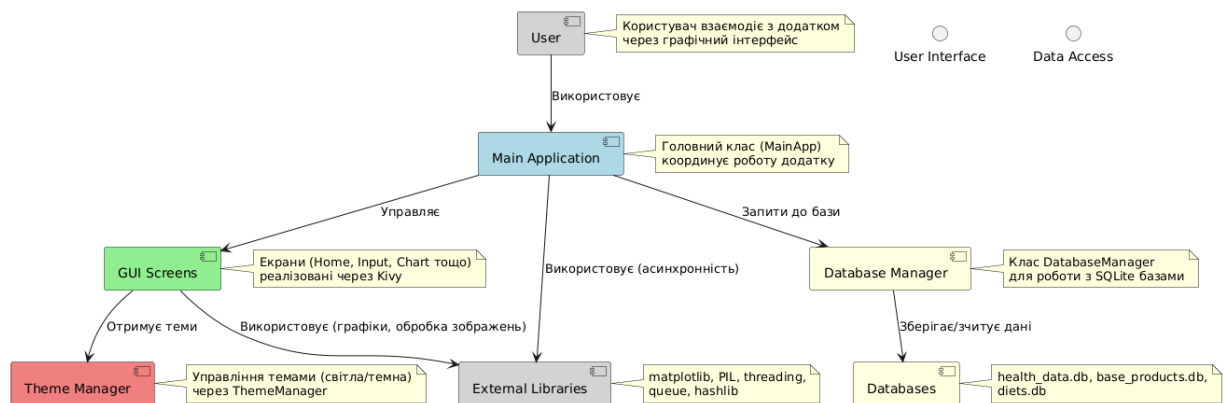


Рисунок 3.1 – Загальна архітектура застосунку

3.2 Обґрунтування вибору засобів для розробки застосун

Для реалізації мобільного застосунку для підтримки здоров'я було обрано наступний стек технологій:

- мова програмування Python;
- фреймворк Kivy, бібліотеки інтерфейсних компонентів KivyMD;
- вбудовані бази даних SQLite;
- додаткові засоби для візуалізації, обробки даних та забезпечення логіки користувацької взаємодії.

Такий вибір забезпечує повну автономність застосунку, кросплатформеність, сумісність, високу гнучкість у розширенні функціоналу, а також прозорість обробки даних, що не потребує серверної інфраструктури або постійного підключення до мережі Інтернет.

Python виступає основною мовою програмування застосунку. Вона є високорівневою, має великий набір бібліотек і вважається універсальною. Активна спільнота розробників та широка база знань роблять її ефективним інструментом для створення як швидких прототипів, так і готових продуктів.

У межах даного проєкту використано 29 бібліотек і модулів, основними можна виділити:

- `sqlite3` – для роботи з локальною базою даних;
- `hashlib` – для створення хешів (перевірка цілісності, зберігання паролів);
- `threading` і `queue` – для реалізації багатопотокового виконання;
- `matplotlib` та `PIL` – для створення графіків та обробки зображень;
- `scikit-learn` і `NumPy` – для реалізації елементів машинного навчання, математичних обчислень та пошуку схожих об'єктів.

Фреймворк Kivy став базою для побудови кросплатформи графічного інтерфейсу користувача. Він підтримує мобільні (Android, iOS) та настільні (Windows, Linux, macOS) середовища, що полегшує адаптацію продукту.

Основні переваги Kivy:

- підтримка OpenGL ES 2.0 для прискореного рендерингу;
- власна система обробки подій і взаємодії з пристроями введення;

– використання декларативної мови KV, що дозволяє відокремити логіку інтерфейсу від бізнес-логіки.

Інтерфейс розроблено за допомогою KivyMD – бібліотеки, що реалізує компоненти за стандартами Material Design від Google. Це забезпечує інтуїтивно зрозумілий, адаптивний, сучасний вигляд інтерфейсу із підтримкою візуальної ієрархії та анімацій.

У реалізації використано наступні елементи: MDLabel, MDTextField, MDFlatButton, MDDialog, ScreenManager, BoxLayout, ScrollView та інші. Завдяки ним реалізовано багатосторінкову навігацію, керування станами, а також динамічні елементи інтерфейсу, які змінюються в реальному часі.

Механізмом локального зберігання даних обрано SQLite – файлову СУБД, яка не вимагає серверного компонента. Її переваги:

- повна підтримка SQL-запитів;
- відповідність стандарту ACID для транзакцій;
- відсутність необхідності у додатковій інсталяції чи конфігурації;
- ефективність при виконанні неінтенсивних запитів.

Завдяки цим особливостям SQLite задовольняє вимоги автономної роботи застосунку для одного користувача: зберігання історії, конфігурацій, налаштувань та результатів сеансів.

Для реалізації функцій аналітики, візуального відображення результатів і покращення користувацького досвіду було інтегровано наступні бібліотеки:

– Matplotlib – для побудови графіків (лінійних, кругових, стовпчикових); рендеринг здійснюється в оперативній пам’яті з подальшим відображенням через текстуру Texture;

– Pillow (PIL) – для масштабування, перетворення та збереження зображень;

– NumPy – для високошвидкісної обробки числових масивів;

– Scikit-learn (модуль sklearn.neighbors.NearestNeighbors) – для реалізації класифікаційних задач та пошуку об’єктів за схожими ознаками.

Ці інструменти дозволяють застосунку здійснювати складні розрахунки, проводити аналіз і виводити результати у графічній формі без потреби у зовнішніх сервісах, повністю автономно. Для підсумування всіх згаданих технологій та їх взаємодія між собою можна привести UML діаграму, що зображена на рисунку 3.2:

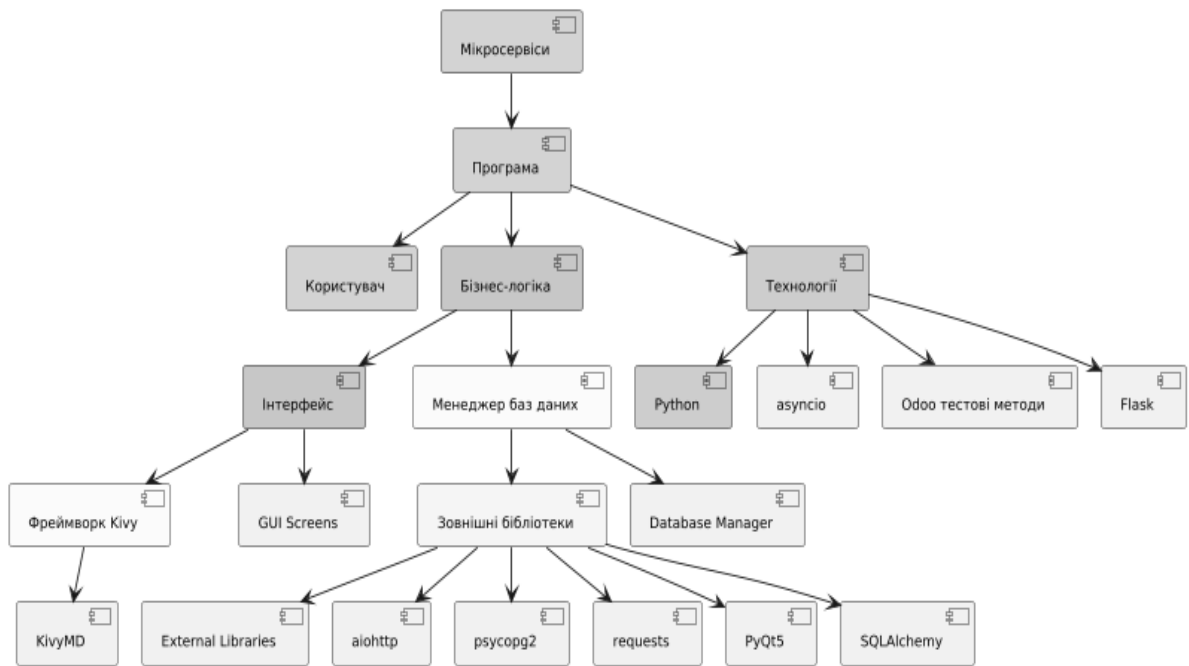


Рисунок 3.2– Діаграма взаємодії усіх компонентів між собою

Застосунок побудовано за принципом багаторівневої архітектури (Layered Architecture) зображено на рисунку 3.3:

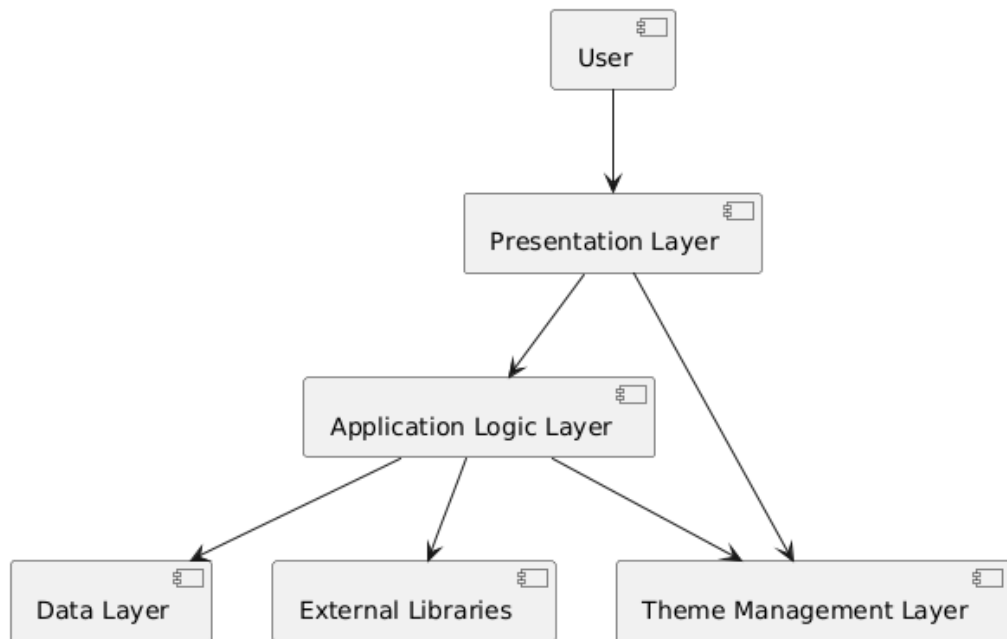


Рисунок 3.3 – Багаторівнева архітектура (Layered Architecture)

У процесі розроблення мобільного застосунку для підтримки здоров'я реалізовано багаторівневу архітектуру, що передбачає поділ функціональних обов'язків між окремими логічними рівнями. Такий підхід дає змогу чітко структурувати програмну систему, забезпечити незалежну змінність компонентів, повторне використання модулів та спростити тестування.

Рівень представлення реалізовано з використанням бібліотек Kivu та KivuMD. Він відповідає за графічний інтерфейс користувача, зокрема за візуалізацію даних, реагування на дії користувача, обробку подій введення, навігацію між екранами та оновлення елементів відображення. Цей компонент забезпечує основний канал взаємодії системи з користувачем і формує його суб'єктивне враження від застосунку.

Рівень бізнес-логіки виконує функцію посередника між інтерфейсом і даними. Саме тут реалізовано керування поведінкою застосунку, обробку подій, вибір алгоритмів, координацію викликів до джерел інформації та визначення подальших дій. Така структура дозволяє ізолювати логіку від зовнішнього вигляду та гарантувати її узгоджене функціонування незалежно від змін в інтерфейсі або даних.

Рівень доступу до даних забезпечує роботу з локальним сховищем SQLite. Клас DatabaseManager, який реалізує відповідну логіку, інкапсулює запити до бази, підтримує захист від SQL-ін'єкцій, контроль цілісності транзакцій і структурну ізоляцію. Це дає змогу централізовано керувати зберіганням інформації, забезпечуючи безпечно та ефективно опрацювання фізіологічних показників.

Рівень тем і стилів виконує роль візуального налаштування інтерфейсу. Він охоплює адаптацію до різних розмірів екранів, підтримку темного(нічного) і світлого режимів, відповідність принципам Material Design. Завдяки цьому досягається високий рівень доступності, зручності, естетики та ергономіки застосунку.

Для реалізації взаємодії між структурними компонентами використано патерн проектування MVC, що включає модель, представлення та контролер. Модель, реалізована у вигляді класу DatabaseManager, здійснює збереження, оновлення й вибірку інформації з бази даних. Представлення формують файли з розширенням .kv, де описано компоненти інтерфейсу користувача, зокрема текстові поля, кнопки, діалоги тощо. Контролер, який написаний мовою Python, керує потоком інформації, отримує дані з моделі, обробляє їх та передає до представлення, забезпечуючи логіку швидкої взаємодії всередині системи застосунку.

Ключовим елементом оптимізації стало впровадження асинхронної обробки даних, яка реалізується за допомогою модулів threading, queue та механізмів фонових потоків. Це дозволяє виконувати ресурсоємні завдання (наприклад, побудову графіків, зчитування великих обсягів даних або аналіз за допомогою NumPy/scikit-learn) без блокування основного потоку, що забезпечує плавність інтерфейсу.

Завдяки використанню буфера пам'яті (io.BytesIO) та рендерингу графіків у фоновому режимі, значно зменшується навантаження на графічний інтерфейс і підвищується загальна продуктивність. Крім того,

багатопоточність використовується для фонові обробки сесій, автоматичного збереження й обробки подій таймерів.

Обрана архітектурна модель має ряд переваг, які є критично важливими для розроблення мобільного застосунку, орієнтованого на автономність, масштабованість і підтримку.

Чіткий розподіл відповідальностей. Кожен шар та компонент виконує лише визначену функцію, що знижує ймовірність помилок, спрощує читання коду та дає змогу швидко виявити проблемні місця.

Гнучкість і масштабованість. Зміна одного шару (наприклад, переходу з SQLite на іншу БД) не впливає на інші частини застосунку. Це дозволяє легко розширювати функціонал без порушення архітектури.

Висока адаптивність інтерфейсу. Завдяки KivyMD користувацький інтерфейс адаптовано під різні платформи та розміри екранів, що позитивно впливає на досвід користувача (UX).

Підвищена продуктивність. Асинхронна обробка дозволяє уникати зависань та затримок, що особливо важливо для мобільних пристроїв з обмеженими ресурсами.

Легка підтримка та тестування. Завдяки модульності та патерну MVC, окремі компоненти можна незалежно тестувати, оновлювати або замінювати.

Підтримка автономної роботи. Всі обчислення, зберігання й обробка даних відбуваються локально, що дозволяє користуватися застосунком без постійного підключення до мережі Інтернет.

Безпека та цілісність. Використання `hashlib`, валідації та розділення логіки зберігання підвищує загальний рівень безпеки застосунку.

Таким чином, архітектура мобільного застосунку забезпечує надійну, ефективну та масштабовану основу для реалізації функціоналу, спрямованого на підтримку здоров'я користувача, навіть в умовах обмеженого підключення та ресурсів.

3.3 Реалізація застосунку

Технічне завдання спрямоване на розробку вдосконаленого мобільного застосунку, орієнтованого на моніторинг і управління фізичним здоров'ям. Загальна мета застосунку полягає у створенні зручної, інформативної та функціональної платформи, за допомогою якої користувачі зможуть відстежувати свої фізичні показники, отримувати персоналізовані рекомендації щодо дієти, керувати своїм раціоном і розробляти дієтичні плани для загального фізичного покращення. Застосунок має забезпечити ефективну взаємодію користувача з функціоналом у зручному інтерфейсі, адаптованому для мобільних пристроїв.

Продукт забезпечує функціонал для управління інформацією, аналізу та планування разом із персоналізованими рішеннями для обробки даних, пов'язаних із поєднанням фізичних вправ і дієти. Користувачам надається можливість вводити дані про пройдені відстані, тривалість сну та спожиті калорії через текстові поля з вбудованими елементами перевірки. Усі ці дані зберігаються локально в базі даних SQLite, що забезпечує безпечне зберігання та швидкий доступ для подальшого аналізу.

У застосунку реалізовано калькулятор макронутрієнтів, що включає білки, жири та вуглеводи. В інтерфейсі всі інформаційні блоки інтегровані з локальною базою даних користувача. Програмне забезпечення містить модуль бази даних продуктів, що дозволяє користувачам переглядати, фільтрувати та вибирати продукти. Доступний пошук за назвою, аналіз поживних властивостей і вибір для використання у калькуляторі БЖВ.

Модуль рекомендацій дозволяє створювати персоналізовані дієтичні плани на основі введених особистих даних, таких як вага, зріст, цільова вага, вік і тривалість. Використовуються математичні формули для розрахунку індексу маси тіла (ІМТ), базового метаболізму (ВМР) і загальних добових енерговитрат (ТДЕЕ). Отриманий план включає розподіл макронутрієнтів, калорій, білків, жирів і вуглеводів, а також перелік страв із бази даних.

Користувач може переглядати заплановані страви та харчування за днями з можливістю позначати спожиті страви.

Графічний модуль відображає коливання фізичних параметрів, таких як кроки, сон, калорії, а також порівняння цільових і фактичних значень поживних речовин. Графіки створюються за допомогою бібліотеки `matplotlib`, кешуються і оновлюються при зміні даних. Реалізовані навігаційні елементи для перемикання між графіками. Передбачено аналіз середніх показників за останні 7 днів, включно із загальною активністю, сном та спожитою енергією.

Користувачі мають можливість перемикатися між темним і світлим режимами інтерфейсу, що покращує зручність роботи за різних умов освітлення. Загалом, застосунок забезпечує персоналізоване та інтерактивне середовище, що підтримує фізичне здоров'я шляхом відстеження, аналізу та планування дієтичних звичок. Програмне забезпечення використовує базу даних `SQLite` і алгоритм `k-NN`, забезпечуючи автономність без необхідності підключення до зовнішніх сервісів.

Мобільна частина програми суттєво впливає на комфорт використання та підвищення ефективності сприйняття інформації. У верхній частині кожного екрана розташований заголовок, що відображає активний сегмент інтерфейсу. Навігація здійснюється через головне меню, що містить кнопки для доступу до модулів: введення даних, графіки, рекомендації, калькулятор БЖВ, база даних, підсумок і налаштування. Кнопки реалізовані як віджети, які змінюють колір при натисканні.

Крім основних функцій, мобільний застосунок інтегрує механізми довготривалого спостереження та аналітики. Система формує щоденні звіти, що містять графічне та текстове представлення змін. Візуалізація даних здійснюється у вигляді графіків, доповнених короткими аналітичними підсумками. Рекомендаційний модуль самонавчається й адаптується до змін поведінки користувача, автоматично коригуючи плани. Таким чином, застосунок створює інтелектуальне середовище для підтримки здоров'я, що адаптується під індивідуальні потреби.

3.3.1 Опис основних компонентів застосунку

Програмний продукт являє собою мобільний застосунок, спеціально розроблений для комплексного моніторингу показників фізичного здоров'я користувача, управління раціональним харчуванням і підтримки здорового способу життя. Основною метою створення цього програмного засобу є забезпечення максимально доступного, інтуїтивно зрозумілого та функціонально насиченого інтерфейсу, який дає змогу реалізувати цілісний підхід до аналізу фізіологічних параметрів, ведення деталізованого харчового щоденника, формування персоналізованих дієтичних рекомендацій і рекомендацій щодо фізичної активності. Архітектура програмного забезпечення ґрунтується на використанні бібліотеки Kivu, що забезпечує повну кросплатформенність, дозволяючи користувачам ефективно працювати із застосунком на мобільних пристроях під керуванням операційних систем Android та iOS, а також на персональних комп'ютерах із різними операційними системами. Такий підхід сприяє універсальності продукту, забезпечуючи його гнучке застосування в різних умовах експлуатації. Застосунок використовує сучасні методи обробки даних, що дозволяють оперативно аналізувати біометричні показники, адаптувати рекомендації до індивідуальних потреб користувача та забезпечувати стабільну роботу навіть за умов обмежених обчислювальних ресурсів. Модульна структура програми сприяє її масштабованості, полегшує впровадження нових функціональних можливостей і забезпечує зручність оновлення для підтримки актуальності відповідно до сучасних вимог і технологічних тенденцій.

Інтерфейсна частина проєкту організована за принципом модульної побудови, де кожен функціональний елемент реалізований у вигляді окремого екрану, оформленого з використанням декларативного підходу у файлі `main.kv`. Поведінкова логіка кожного модуля визначається окремим класом, який наслідує властивості базового класу `Screen`, забезпечуючи тим самим

ізолювану реалізацію внутрішньої логіки кожного розділу програми. Керування переходами між функціональними екранами реалізовано через компонент `ScreenManager`, який дозволяє здійснювати навігацію з анімаційною підтримкою, а також гарантує збереження стану між переходами.

Для забезпечення адаптивності та універсальності застосунку на пристроях із різною роздільною здатністю використано компоненти організації простору, зокрема `BoxLayout`, `GridLayout` та `FloatLayout`, що дозволяє динамічно підлаштовувати вміст під технічні особливості екрана. Передбачено можливість перемикання між темною та світлою кольірними темами, що сприяє підвищенню зручності сприйняття інформації в умовах різного освітлення. З метою забезпечення інтернаціоналізації реалізовано підтримку локалізаційного механізму з використанням словникової структури перекладів, що дозволяє розширити мовну підтримку без внесення змін до основного програмного коду.

Таким чином, розроблений застосунок є гнучким, масштабованим і відкритим до подальшої модернізації, відповідає сучасним вимогам до програмних засобів у сфері цифрової медицини, сприяє формуванню здорового способу життя та забезпечує надійне й ефективне рішення для моніторингу та оптимізації фізичного стану користувача

Для зберігання та обробки даних у застосунку застосовується локальна база даних `SQLite`, яка реалізована через спеціально створений модуль `database.py`, що забезпечує зручну та структуровану взаємодію з інформацією. Структура бази даних складається з трьох ключових сховищ: `health_data.db`, призначене для збереження різноманітних фізичних показників, таких як кількість кроків, тривалість і якість сну, частота пульсу та витрата калорій, що є основою для аналізу стану здоров'я; `base_products.db`, яке використовується для детального зберігання інформації про харчові продукти, включаючи їх назву, калорійність, вміст білків, жирів, вуглеводів та тип прийому їжі, що дозволяє вести точний облік харчування; а також `diets.db`, де зберігаються дані про заплановані та вже спожиті дієти з чіткою прив'язкою до конкретних дат,

що полегшує планування та контроль раціону. Базовий клас `DatabaseManager` виступає центральним елементом управління доступом до даних, забезпечуючи зручний і надійний інтерфейс, а також підтримує індексацію таблиць для значного пришвидшення пошуку необхідної інформації навіть у великих обсягах даних. При першому запуску застосунка база продуктів автоматично наповнюється стандартними записами, що гарантує мінімальний рівень функціональності з самого початку використання, дозволяючи користувачу одразу приступити до роботи без додаткових налаштувань.

Синхронізація з `Google Fit` реалізована через окремий модуль `google_fit_sync.py`, що використовує протокол `OAuth 2.0` для автентифікації. Після збереження токенів у файлі `token.json` застосунок щоденно, в нічний час, у фоновому режимі виконує синхронізацію даних за попередню добу. Усі отримані фізіологічні показники зберігаються у локальну базу даних через відповідні методи, що гарантує збереження історії активності навіть без активного підключення до Інтернету.

Аналіз фізичного стану користувача здійснюється на основі алгоритму найближчих сусідів (`k-NN`), реалізованого в модулі `pulse_knn_analysis.py`. На основі введених параметрів (пульс, калорії, сон, активність) виконується класифікація за історичними даними, після чого користувачеві надаються індивідуальні рекомендації, наприклад щодо необхідності відпочинку. Крім текстового опису, користувач також бачить графік із найближчими сусідами, що допомагає візуально інтерпретувати результати.

Темна і світла теми оформлення реалізовані за допомогою модуля `theme.py`, який забезпечує централізоване управління кольоровими палітрами всіх елементів інтерфейсу, включно з фоном, текстами, кнопками та текстовими полями. Застосування теми відбувається через метод `apply_theme`, що оновлює всі віджети в реальному часі, забезпечуючи плавний перехід між режимами.

Графічне представлення даних забезпечує клас `ChartScreen`, який використовує бібліотеку `matplotlib` для генерації графіків фізичної активності,

сну, спожитих калорій, а також порівняння планових і фактичних значень поживних речовин. Для підвищення продуктивності графіки зберігаються як кешовані зображення у форматі PNG, а їх генерація виконується в окремому потоці, що унеможлиблює блокування головного інтерфейсу. Відображення графіків здійснюється через Canvas з використанням текстур.

Особлива увага приділена обробці помилок. Для журналювання подій використовується модуль logging, а критичні винятки перехоплюються через конструкції try-except із відповідним повідомленням користувачеві у спеціальних полях інтерфейсу (error_label, result_label). Це забезпечує стабільність роботи застосунка та інформативність при збої функцій.

Усі екрани програми пов'язані між собою згідно з логікою взаємодії: головне меню (HomeScreen) забезпечує доступ до введення даних (InputScreen), графічного аналізу (ChartScreen), перегляду дієт (DietViewerScreen), генерації рекомендацій (RecommendationsScreen) та аналізу пульсу (PulseKNNScreen). Дані, введені вручну або отримані через API, зберігаються в базі даних, звідки надалі використовуються для візуалізації, аналізу та формування дієтичних планів. Інтерфейс підтримує гнучке компонування через BoxLayout і ScrollView, що забезпечує зручне розміщення елементів незалежно від розміру екрана пристрою.

Застосунок підтримує кросплатформність, що забезпечує його коректну роботу на різних операційних системах, має повну локалізацію українською мовою, що сприяє зручності використання для україномовних користувачів, забезпечує асинхронну обробку фонових задач для оптимізації продуктивності та реалізує сучасний підхід до моніторингу фізичного стану з елементами штучного інтелекту. Це дозволяє забезпечити високу адаптивність до потреб користувачів, ефективну роботу на різних типах пристроїв і надання персоналізованих рекомендацій на основі аналізу даних. Інтеграція інтелектуальних алгоритмів дає змогу оперативно обробляти біометричні показники, підвищуючи точність моніторингу та якість взаємодії з користувачем. Завдяки модульній архітектурі застосунок легко

масштабується, що забезпечує його гнучкість і можливість подальшого вдосконалення відповідно до сучасних технологічних вимог.

Подальший розвиток застосунку передбачає інтеграцію з носимими пристроями на зразок фітнес-браслетів та смарт-годинників, що дозволить здійснювати більш точний та автоматизований збір даних у реальному часі. Також планується розширення функціоналу за рахунок додавання модуля психологічного стану, який базуватиметься на аналізі щоденникових записів, моделей поведінки користувача та рівня стресу протягом дня, що покращить загальну якість аналізу. З технічної точки зору передбачається перехід до використання хмарної бази даних для забезпечення зручної синхронізації між пристроями та створення резервних копій. Додаткову увагу буде приділено впровадженню механізмів машинного навчання з метою прогнозування ризиків для здоров'я та формування ще точніших рекомендацій на основі динаміки індивідуальних показників. Таким чином, застосунок продовжить еволюціонувати як повноцінна платформа для цифрового моніторингу здоров'я, адаптована до потреб кожного користувача.

Розроблений мобільний застосунок забезпечує комплексний підхід до моніторингу фізичного стану користувача та управління харчуванням. Його архітектура побудована на основі бібліотеки Kivu, що гарантує кросплатформенність, гнучкість інтерфейсу та зручність у подальшій модифікації відповідно до змін вимог користувачів. Застосунок включає модулі синхронізації з Google Fit, аналізу даних за допомогою алгоритмів машинного навчання та графічної візуалізації показників у доступному для сприйняття форматі. Система є масштабованою, зручною для користувача і готовою до подальшого розвитку та інтеграції з новими сервісами і платформами.

У мобільному застосунку реалізовано головне меню, що зображено на рисунку 3.4, яке містить набір кнопок, кожна з яких відповідає за окрему

функцію, пов'язану з моніторингом здоров'я та управлінням харчуванням. Нижче подано опис функціонального призначення кожної з них.

«Введення даних» – відкриває екран, на якому користувач може вручну ввести показники стану здоров'я: кількість пройдених кроків, тривалість сну, спожиті калорії та середній пульс. Після заповнення даних вони зберігаються у базі даних для подальшого аналізу.

«Графіки» – кнопка переводить користувача до екрана, де відображаються графіки статистики: кроки, сон, калорії, пульс. Також тут представлені прогнози, побудовані за допомогою алгоритму k-ближчих сусідів (k-NN). Перехід між типами графіків здійснюється за допомогою кнопок навігації «<» і «>». Приклад графіку можна побачити на рисунку 3.5.

«Створити дієту» – відкриває екран формування індивідуального плану харчування. Користувач вводить такі параметри: поточна вага, зріст, цільова вага, вік і бажаний термін досягнення мети. Застосунок розраховує добову норму калорій і співвідношення білків, жирів і вуглеводів, після чого генерує відповідну дієту.

«Додати продукти» – дозволяє користувачеві розрахувати харчову цінність страви за введеною вагою продуктів або додати нові продукти до бази, вказавши їх назву, калорійність, вміст білків, жирів і вуглеводів.

«Перегляд дієти» – забезпечує доступ до поточного плану харчування. Користувач може переглянути страви, рекомендовані на кожен прийом їжі (сніданок, обід, вечеря), та позначати спожиті за допомогою чекбоксів. Переглянути екран з дієтами можна на рисунку 3.6.

«Зведення» – відкриває екран, де відображається підсумкова статистика за певний період. Включає середні значення кроків, сну, пульсу та калорій, а також загальні цільові й фактичні показники по дієті (калорії, білки, жири, вуглеводи).

«Аналітика» – реалізує функцію аналізу стану здоров'я за допомогою алгоритму k-ближчих сусідів екран зображено на рисунку 3.7. Користувач вводить основні показники (пульс, кроки, сон, спожиті білки, жири, вуглеводи,

калорії), після чого система генерує висновок щодо рівня стресу або можливих відхилень у фізіологічному стані [28 – 30].

«Синхронізація Google Fit» – відкриває екран ручної синхронізації з Google Fit API. Після авторизації система отримує дані про кроки, пульс, сон і калорії за останню добу та зберігає їх у локальну базу даних.

«Налаштування» – відкриває екран, де користувач може перемикає тему інтерфейсу (світла або темна), а також вручну запуснути синхронізацію з Google Fit.

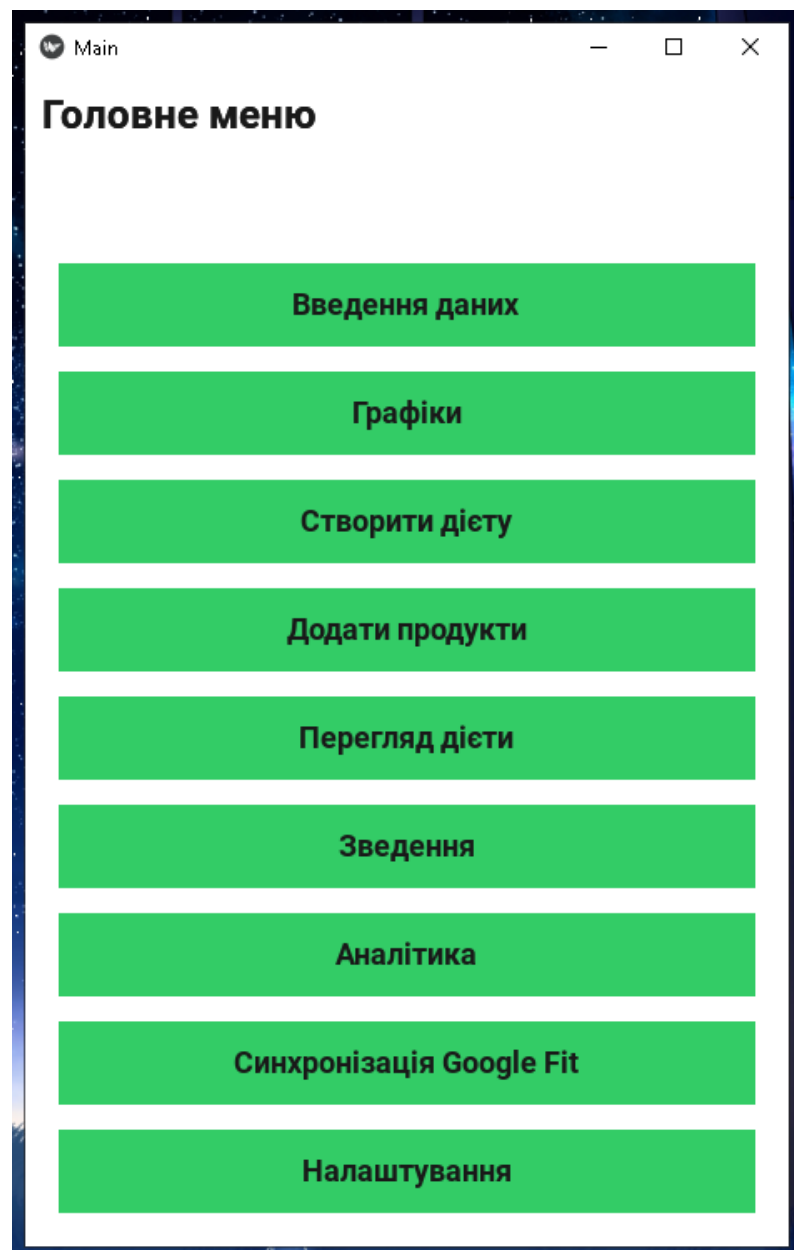


Рисунок 3.4 – Головне меню

Інтуїтивно зрозумілий дизайн для будь-якої вікової групи. Тематика виконана в зеленому кольорі, що надає свіжого та спокійного враження. Кожен пункт представлений у вигляді зеленого блоку з білим текстом, що робить навігацію зручною та візуально привабливою.

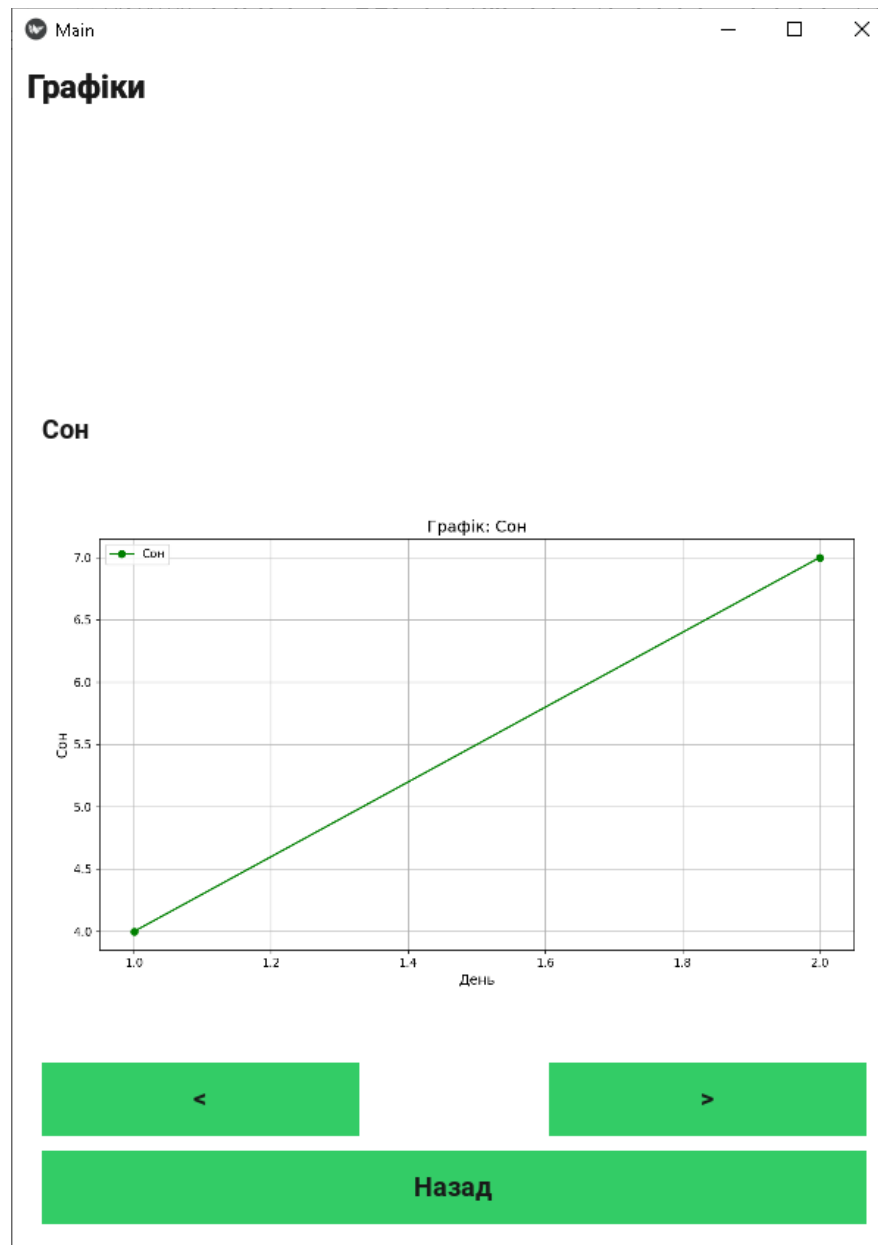


Рисунок 3.5 – Екран графіків

У класі ChartScreen, що розташований у файлі screens.py, реалізовано повну логіку формування та відображення графіків для візуалізації стану здоров'я користувача.

При вході на екран (метод `on_enter`) викликається функція `load_data()`, яка ініціює завантаження даних про стан здоров'я (`health_data`), інформації про дієту (`diet_data`) та спожиті продукти (`consumed_data`) з локальної бази даних за допомогою класу `DatabaseManager`. Одночасно активується фоновий потік `start_graph_worker`, який опрацьовує чергу типів графіків (`graph_queue`).

У застосунку реалізовано декілька типів графіків, перелічених у списку `graph_types`. До них належать графіки для базових метрик (кількість кроків, тривалість сну, кількість спожитих калорій, середній пульс), порівняння цільових і фактичних значень (калорії, білки, жири, вуглеводи), а також прогнозовані значення на основі алгоритму k-ближчих сусідів (k-NN).

Побудова графіків виконується у методі `_generate_chart`, який працює у фоновому потоці для зменшення навантаження на основний інтерфейс. Для побудови використовується бібліотека `matplotlib`. Лінійні графіки будуються для щоденних показників здоров'я. Для порівняльних графіків візуалізуються одночасно два ряди значень – фактичні й цільові. У випадках прогнозування застосовується модуль `PulseKNNAnalyzer`, що аналізує історичні дані з використанням методу k-ближчих сусідів для побудови прогнозів за відповідними параметрами.

Згенеровані графіки зберігаються у форматі PNG у тимчасовій папці `cache`, причому назви файлів формуються на основі хешу відповідного типу графіка. Для оновлення інтерфейсу користувача застосовується метод `_update_graph_ui_with_source`, який завантажує відповідне зображення у віджет типу `Image` (`graph_image`) та оновлює підпис (`graph_label`). У разі відсутності даних або помилки виводиться відповідне повідомлення у спеціальному полі.

Для зручності реалізована функціональність переходу між графіками за допомогою кнопок «<» та «>». Перемикання здійснюється через методи `next_graph()` та `previous_graph()`, які очищають поточний графік (`_clear_graph()`) та додають у чергу новий тип для побудови.

Кешована інформація автоматично очищується при кожному вході на екран завдяки виклику методу `clear_cache()`, що забезпечує актуальність відображуваних даних.

Таким чином, система формування графіків є асинхронною, оптимізованою для продуктивності, підтримує відображення базових, порівняльних та прогнозованих метрик, а також забезпечує інтерактивну навігацію між візуалізаціями.

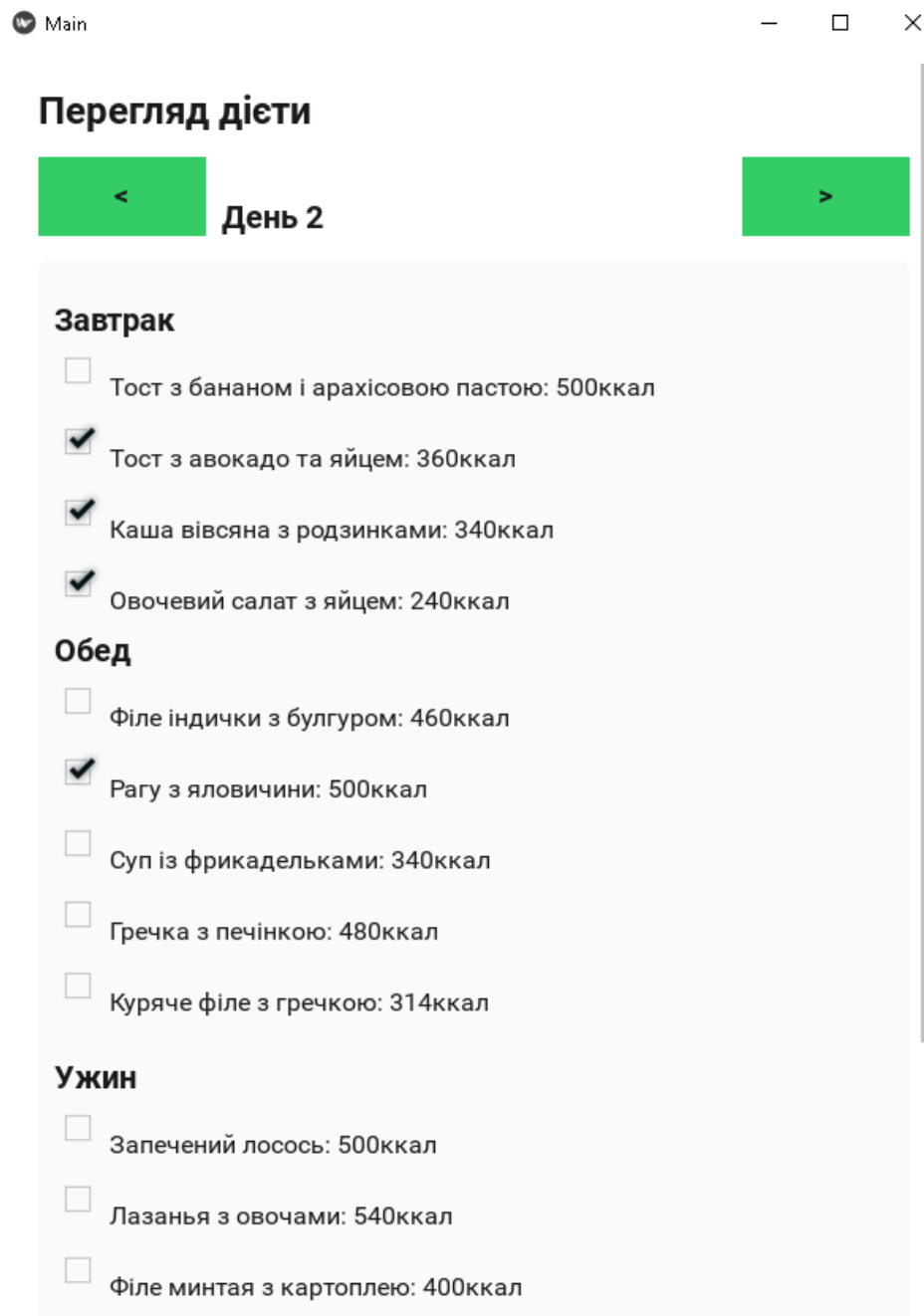


Рисунок 3.6 – Екран перегляду рекомендованої дієти

Екран перегляду рекомендованої дієти відображає дієту за днями, групуючи страви за прийомами їжі (сніданок, обід, вечеря). Кожну страву можна позначити як спожиту за допомогою чекбокса, після чого запис створюється в таблиці `consumed_diets`. Дані завантажуються з бази через `DatabaseManager`, а навігація між днями забезпечується кнопками. Інтерфейс автоматично оновлюється відповідно до вибраної теми (світла або темна).



Main

Аналіз пульсу (k-NN)

Пультс (уд/хв): Середній пульс

Кроки: Кількість кроків

Сон (години): Години сну

Білки (г): Спожиті білки

Жири (г): Спожиті жири

Вуглеводи (г): Спожиті вуглеводи

Калорії (ккал): Спожиті калорії

Аналізувати

Назад

Рисунок 3.7 – Екран рекомендацій з аналізом стану здоров'я

ВИСНОВКИ

У виконаній кваліфікаційній роботі досягнуто поставленої мети – здійснено повноцінну розробку програмно-апаратного засобу у вигляді мобільного застосунку, що забезпечує моніторинг фізіологічних показників користувача з подальшим аналізом стану організму та формуванням індивідуалізованих рекомендацій превентивного характеру.

Проведене теоретичне та прикладне дослідження підтвердило обґрунтованість обраного науково-практичного підходу і реалізацію всіх поставлених завдань.

Здійснено глибокий аналітичний огляд сучасного стану вітчизняних і зарубіжних технологій мобільного здоров'я, зокрема апаратних засобів збору медичних даних, програмного забезпечення для їх обробки та провідних алгоритмів класифікації у сфері штучного інтелекту. Сформульовано системну оцінку переваг і недоліків кожного з підходів в умовах мобільного середовища.

Визначено найбільш інформативні фізіологічні параметри, що підлягають контролю, з урахуванням технічної доцільності та обмежень сенсорних пристроїв. Розроблено методику попередньої обробки, очищення й фільтрації даних, що забезпечує підвищення достовірності подальших обчислень та інтерпретацій.

Сформовано моделі класифікації фізіологічного стану користувача із застосуванням сучасних методів інтелектуального аналізу даних – k -найближчих сусідів, нейронних мереж та алгоритмів кластеризації. Емпірична перевірка засвідчила, що нейронні мережі демонструють високу ефективність при обробці складних багатовимірних залежностей, тоді як метод k -NN вирізняється простотою реалізації та достатньою точністю на контрольованих множинах даних.

Реалізовано багаторівневу архітектурну модель мобільного застосунку, що включає представницький, логічний, інфраструктурний та інтерфейсно-

декоративний компоненти. Використання патерна проектування MVC забезпечило чітке розмежування функціональних обов'язків між компонентами системи, покращення структури коду та підвищення модульності й підтримуваності рішення.

Створено функціонально завершений прототип мобільного застосунку з графічним інтерфейсом, що забезпечує візуалізацію ключових фізіологічних показників, відображення рекомендацій щодо покращення фізичного стану, а також автоматичне інформування користувача у разі виявлення потенційних загроз для здоров'я.

Проведено комплексне тестування системи на основі набору синтетичних і реальних фізіологічних даних, яке підтвердило адекватність архітектури, стійкість алгоритмів класифікації до зміни вхідних умов та доцільність запропонованих функціональних рішень.

Отже, реалізований мобільний застосунок є ефективним інструментом підтримки здоров'я користувачів, що забезпечує безперервний моніторинг, аналітичну обробку даних і надання обґрунтованих рекомендацій. Здобуті результати мають прикладне значення у галузі цифрової медицини і відкривають перспективи для подальших наукових досліджень у сфері персоналізованих систем охорони здоров'я.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Shaik, T., Tao, X., Higgins, N., Li, L., Gururajan, R., Zhou, X., & Acharya, U. R. (2023). Remote patient monitoring using artificial intelligence: Current state, applications, and challenges. arXiv.
2. Jat, A. S., & Grønli, T.-M. (2024). Harnessing the Digital Revolution: A Comprehensive Review of mHealth Applications for Remote Monitoring in Transforming Healthcare Delivery. arXiv.
3. Aljedaani, B., & Babar, M. A. (2020). Challenges in Developing Secure Mobile Health Applications: A Systematic Review. arXiv.
4. Devi, D. H., Duraisamy, K., Armghan, A., Alsharari, M., Aliqab, K., Sorathiya, V., Das, S., & Rashid, N. (2023). 5G Technology in Healthcare and Wearable Devices: A Review. *Sensors*, 23(5), 2519.
5. Tepper, P. (2023). Real-time Health Monitoring: Role of Mobile Technologies in Public Health. Walsh Medical Media.
6. Liu, P., Fels, S., West, N., & Görges, M. (2019). Human Computer Interaction Design for Mobile Devices Based on a Smart Healthcare Architecture. arXiv.
7. Torres-Guzman, R. A., Paulson, M. R., Avila, F. R., Maita, K., Garcia, J. P., Forte, A. J., & Maniaci, M. J. (2023). Smartphones and Threshold-Based Monitoring Methods Effectively Detect Falls Remotely: A Systematic Review. *Sensors*, 23(3), 1323.
8. Ahmed, S., Irfan, S., Kiran, N., Masood, N., Anjum, N., & Ramzan, N. (2023). Remote Health Monitoring Systems for Elderly People: A Survey. *Sensors*, 23(16), 7095.
9. de Souza Ferreira, E., de Aguiar Franco, F., dos Santos Lara, M. M., et al. (2023). The effectiveness of mobile application for monitoring diabetes mellitus and hypertension in the adult and elderly population: systematic review and meta-analysis. *BMC Health Services Research*, 23, 855.

10. Всесвітня організація охорони здоров'я (ВООЗ). (2022). Глобальні тенденції надмірної ваги та фізичної активності населення.
11. Малахов, К. С. (н.д.). Цифрові технології охорони здоров'я в Україні: стан та перспективи.
12. Турбота: персоналізований медичний застосунок для пацієнтів. (н.д.).
13. Іваненко, М. І., & Петров, О. В. (2023). Використання мобільних додатків для моніторингу здоров'я в Україні. Медична інформатика, (3), с. 45-53.
14. Ковальчук, О. І. (н.д.). Розробка мобільного додатку для фізичної активності: досвід України.
15. Степанова, Т. В., & Савченко, О. В. (2022). Мобільні технології у сфері охорони здоров'я: сучасні тенденції та виклики. Український журнал медицини та біології, 5(2), с. 78-83.
16. Жернова, П. Є., & Бодяньський, Є. В. (2019). Нечітка імовірнісноможливісна послідовна кластеризація даних на основі ансамблевого підходу. Прикладная радиоэлектроника, (1-2), с. 40-45.
17. Вечірська, І. Д., & Четвериков, Г. Г. (2012). Математичні аспекти побудови ланцюгів лексичних одиниць на основі алгебри предикатів. Біоніка інтелекту, (2 (79)), с. 84-88.
18. Hunt, J. A Beginners Guide to Python 3 Programming / Cham : Springer, 2019. – XVIII, с. 433
19. Sundnes, J. Introduction to Scientific Programming with Python / Cham : Springer, 2020. – XIV, с. 148.
20. Turner, P. R., Arildsen, T., Kavanagh, K. Applied Scientific Computing with Python / Cham : Springer, 2018. – X, с. 272.
21. Климчук, В. (2022). KivyMD: швидкий старт : навч. посібник. Харків.
22. KivyMD Documentation [Електронний ресурс]. – Режим доступу: <https://kivymd.readthedocs.io>, вільний. – Дата звернення: 30.04.2024.

23. Kivy Official Documentation [Електронний ресурс]. – Режим доступу: <https://kivy.org/doc>, вільний. – Дата звернення: 30.04.2024.
24. Pakula, K. GUI Programming with Python and Kivy : навч. посібник / Self-published via Prospero Coder, 2021. pp. 826.
25. Лі, П. (2018). Розробка мобільних додатків на Python з Kivy: монографія. Лондон: Packt Publishing.
26. Phillips, D. (2014). Creating Apps in Kivy: Mobile with Python / Sebastopol, CA : O'Reilly Media. pp. 188.
27. Васильєва, Н. (2023). Візуальні інтерфейси на Python з KivyMD : навч. посібник. Київ: НБУВ.
28. Tvoroshenko I., and Gorokhovatskyi V. (2022). The Application of Hybrid Intelligence Systems for Dynamic Data Analysis. International Journal of Engineering and Information Systems, 6(2), 40-48.
29. Gorokhovatskyi V., Tvoroshenko I. (2023). Identification of visual objects by the search request. International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, 25-27.
30. Творошенко І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ.