

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Системотехніки  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Розробка мобільного застосунку аналізу стану здоров'я людини  
та дослідження алгоритмів його роботи  
(тема)

Виконав:  
студент II курсу, групи ІТПм-22-2  
Рожкова А.В.  
(прізвище, ініціали)

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

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

Освітня програма Інформаційні технології  
проектування  
(повна назва освітньої програми)

Керівник доцент Решетнік В.М.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ  
(підпис)

проф. Гребеннік І.В.  
(прізвище, ініціали)

2024 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«16» січня 2024 р.



Рожкова А.В.

Кваліфікаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.

Матеріали кваліфікаційної роботи не містять відомостей, що заборонені для опублікування у відкритих виданнях.

Попередній захист проведено. 16 січня 2024 р.

Керівник кваліфікаційної роботи

Решетнік В.М.

## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
 Кафедра \_\_\_\_\_ Системотехніки \_\_\_\_\_  
 Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
 Спеціальність \_\_\_\_\_ 122 – Комп'ютерні науки \_\_\_\_\_  
 (код і повна назва)  
 Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
 (освітньо-професійна або освітньо-наукова)  
 Освітня програма \_\_\_\_\_ Інформаційні технології проектування \_\_\_\_\_  
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Рожковій Анастасії Володимирівні \_\_\_\_\_  
 (прізвище, ім'я, по батькові)

1. Тема роботи Розробка мобільного застосунку аналізу стану здоров'я людини та дослідження алгоритмів його роботи  
затверджена наказом університету від 20 листопада 2023 р. № 1157Ст
2. Термін подання студентом роботи до екзаменаційної комісії 18 січня 2024 р.
3. Вихідні дані до роботи Розробити мобільний застосунок аналізу стану здоров'я людини. Застосунок повинен працювати на смартфонах з ОС Android.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Аналіз предметної області, яка характеризує діяльність застосунку аналізу стану здоров'я людини 4.2 Аналіз існуючих алгоритмів аналізу здоров'я людини та існуючих систем для відстежування здоров'я 4.3 Сфера використання мобільного застосунку аналізу стану здоров'я людини 4.4 Постановка задачі 4.5 Визначення системних вимог до застосунку аналізу стану здоров'я людини 4.6 Визначення функціональних вимог до застосунку аналізу стану здоров'я людини 4.7 Опис алгоритму аналізу стану здоров'я людини 4.8 Діаграма варіантів використання застосунку аналізу стану здоров'я людини 4.9 Вимоги до інтерфейсу клієнтської частини мобільного застосунку аналізу стану здоров'я людини 4.10 Компоненти мобільного застосунку аналізу стану здоров'я, які підлягають розробці 4.11 Обґрунтування вибору технології створення мобільного застосунку аналізу стану здоров'я людини 4.12 Обґрунтування вибору СУБД 4.13 Логічне і фізичне моделювання даних застосунку аналізу стану здоров'я людини. Створення бази даних 4.14 Розробка карти застосунку аналізу стану здоров'я людини 4.15 Реалізація алгоритму аналізу стану здоров'я людини 4.16 Тестування розробленого мобільного застосунку аналізу стану здоров'я людини
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 5.1 Титульний лист (1 аркуш формату А4) 5.2 Мета роботи (1 аркуш формату А4) 5.3 Актуальність (1 аркуш

формату А4) 5.4 Постановка задачі (1 аркуш формату А4) 5.5 Концептуальна діаграма (1 аркуш формату А4) 5.6 Декомпозиція концептуальної діаграми (1 аркуш формату А4) 5.7 Декомпозиція функції "Аналіз показників" (1 аркуш формату А4) 5.8 Use case діаграма (1 аркуш формату А4) 5.9 Опис алгоритму (1 аркуш формату А4) 5.10 Опис використаних технологій (1 аркуш формату А4) 5.11 Логічна модель даних (1 аркуш формату А4) 5.12 Екрани застосунку (1 аркуш формату А4) 5.13 Головний екран застосунку (1 аркуш формату А4) 5.14 Екран налаштування показників (1 аркуш формату А4) 5.15 Екран звіту (1 аркуш формату А4) 5.16 Тестування (1 аркуш формату А4) 5.17 Висновки (1 аркуш формату А4)

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання кваліфікаційної роботи	20.11.2023	Виконано
2.	Аналіз завдання та аналогів з теми роботи	20..11.2023	Виконано
3.	Розробка технічних вимог до програми	20.11.2023-29.11.2023	Виконано
4.	Структурне проектування	01.12.2023-03.12.2023	Виконано
5.	Розробка алгоритму	04.12.2023-10.12.2023	Виконано
6.	Вибір середовища розробки програми	11.12.2023-19.12.2023	Виконано
7.	Реалізація застосунку	20.12.2023 – 29.12.2023	Виконано
8.	Тестування застосунку	30.12.2023	Виконано
9.	Розробка «Посібника користувача»	31.12.2023	Виконано
10.	Оформлення пояснювальної записки	02.01.2024- 10.01.2024	Виконано
11.	Представлення на рецензування	за 3 дні	Виконано
12.	Представлення кваліфікаційної роботи в ДЕК	за 2 дні	Виконано

Дата видачі завдання 16 жовтня 2023 р.

Студент \_\_\_\_\_ Рожкова А.В.  
(підпис)

Керівник роботи \_\_\_\_\_ доцент Решетнік В.М.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Атестаційна робота: 89 стор., 39 рис., 1 табл., 3 додатки, 25 джерел.  
Графічний матеріал атестаційної роботи містить 17 плакатів.

АНАЛІЗ ЗДОРОВ'Я ЛЮДИНИ, АЛГОРИТМИ АНАЛІЗУ ЗДОРОВ'Я,  
БАЗА ДАНИХ, ІНТЕРФЕЙС ДОСТУПУ, ANDROID, KOTLIN, MYSQL.

Об'єкт дослідження – процес аналізу стану здоров'я людини.

Предмет дослідження - алгоритми для аналізу стану здоров'я людини на основі показників здоров'я.

Мета роботи - розробка мобільного застосунку аналізу стану здоров'я людини.

Методи дослідження – аналіз літератури, аналіз існуючих застосунків, дослідження існуючих алгоритмів для аналізу здоров'я, моделювання застосунку, концептуальне проектування застосунку.

Результати роботи – мобільний застосунок аналізу стану здоров'я людини.

## ABSTRACT

Attestation work: 89 p., 39 pic., 1 table, 25 source, 3 applications. Graphic material attestation work contains 17 poster.

ACCESS INTERFACE, ANDROID, DATABASE, HEALTH ANALYSIS ALGORITHMS, HUMAN HEALTH ANALYSIS, MYSQL, KOTLIN.

The object of development is the process of analyzing the state of human health.

The subject of development is algorithms for analyzing the state of human health based on health indicators.

The purpose of the work is to develop a mobile application for analyzing the state of human health.

Research methods – literature analysis, analysis of existing applications, research of existing algorithms for health analysis, application modeling, conceptual application design.

The results of the work are a mobile application for analyzing the state of human health.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Аналіз предметної області, яка характеризує діяльність застосунку аналізу стану здоров'я людини .....	9
1.2 Аналіз існуючих алгоритмів аналізу здоров'я людини та існуючих систем для відстежування здоров'я.....	11
1.3 Сфера використання мобільного застосунку аналізу стану здоров'я людини.....	19
1.4 Постановка задачі.....	20
2 РОЗРОБКА ВИМОГ ДО ЗАСТОСУНКУ АНАЛІЗУ СТАНУ ЗДОРОВ'Я ЛЮДИНИ.....	22
2.1 Визначення системних вимог до застосунку аналізу стану здоров'я людини.....	22
2.2 Визначення функціональних вимог до застосунку аналізу стану здоров'я людини.....	23
2.3 Опис алгоритму аналізу стану здоров'я людини .....	26
2.4 Діаграма варіантів використання застосунку аналізу стану здоров'я людини.....	31
2.5 Вимоги до інтерфейсу клієнтської частини мобільного застосунку аналізу стану здоров'я людини .....	35
2.6 Компоненти мобільного застосунку аналізу стану здоров'я, які підлягають розробці .....	36
3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ .....	39
3.1 Обґрунтування вибору технології створення мобільного застосунку аналізу стану здоров'я людини .....	39
3.2 Обґрунтування вибору СУБД .....	40
3.3 Логічне і фізичне моделювання даних застосунку аналізу стану здоров'я людини. Створення бази даних.....	40
3.4 Розробка карти застосунку аналізу стану здоров'я людини.....	44
3.5 Реалізація алгоритму аналізу стану здоров'я людини.....	48
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	62
ДОДАТОК А.....	65
ДОДАТОК Б .....	74
ДОДАТОК В .....	79

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

БД – база даних;

Діаграма прецедентів - діаграма, на якій зображено відношення між акторами та прецедентами в системі.

Коефіцієнт кореляції - показник кореляції (лінійної залежності) між двома змінними;

Android – операційна система для смартфонів;

CNN (Convolutional Neural Network) – згорткові нейронні мережі;

Jetpack Compose - набір інструментів для створення інтерфейсу Android;

Kotlin - об'єктно-орієнтована мова програмування, що працює поверх Java Virtual Machine;

Retrofit - бібліотека для мережевої взаємодії;

SQL (Structured Query Language) – мова структурованих запитів.

## ВСТУП

Станом на 2023 рік існує не так багато застосунків для відстежування здоров'я людини. Окрім того, навіть ті застосунки, які існують, не дозволяють якісно аналізувати стан здоров'я на основі показників здоров'я користувача. Навіть не дивлячись на те, що нейронні мережі все частіше використовуються у сфері медицини, передбачення можливих причин зміни стану здоров'я все ще не доступні для широкого кола користувачів. Виходячи з цього, актуально розробити мобільний застосунок для обробки та аналізу даних про здоров'я.

Для автоматизації процесу аналізу здоров'я актуально розробити мобільний застосунок, в якому реалізовано алгоритм для аналізу здоров'я на основі показників стану здоров'я.

Об'єкт дослідження – процес аналізу стану здоров'я людини. Предметом дослідження є алгоритми для аналізу стану здоров'я людини на основі показників здоров'я.

Мета роботи – розробка мобільного застосунку аналізу стану здоров'я людини.

На сучасному етапі розвитку технологій здоров'я, існуючі механізми відстеження та аналізу здоров'я залишаються обмеженими. Аналіз існуючих рішень визначає необхідність нового підходу для комплексного відстеження стану здоров'я. Застосунки сьогодення часто обмежені конкретними аспектами, тому виникає потреба в розробці універсального інструменту, спроможного охоплювати різні показники і забезпечувати глибокий аналіз даних.

Сучасний ритм життя та акцент на здоров'ї визначають високий попит на засоби для моніторингу та покращення стану здоров'я. Однак існуючі рішення не завжди здатні відповісти на реальні потреби користувачів. Розробка мобільного застосунку для аналізу здоров'я є актуальною не лише з погляду технологічних можливостей, але й з точки зору задоволення потреб суспільства у власному здоров'ї та благополуччі.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Аналіз предметної області, яка характеризує діяльність застосунку аналізу стану здоров'я людини

У сучасному світі існують технології, які дозволяють аналізувати стан здоров'я людини за допомогою штучного інтелекту.

Останнім часом такі технології все більш популярні, але практична користь від них малопомітна та неефективна. Одною із причиною цього є те, що нейронні мережі потребують високої обчислювальної потужності, що не є ефективним.

Головний бізнес-процес предметної області – це проведення аналізу здоров'я людини. Для проведення аналізу здоров'я, користувач (пацієнт) повинен надати інформацію про себе (стать, дата народження) та показники (тиск, пульс, рівень цукру у крові, сатурація). Користувач повинен надавати інформацію про показники здоров'я щодня використовуючи спеціальну форму збору цих даних у застосунку. Таким чином, в ході регулярного надання цієї інформації, проводиться аналіз стану пацієнта і результат аналізу надаються користувачу (пацієнту).

Аналіз стану здоров'я – це систематичний розгляд та інтерпретація різноманітних показників здоров'я, що збираються та аналізуються мобільним застосунком. Це включає в себе обробку широкого спектру даних, таких як фізична активність, біометричні вимірювання, прийом ліків та інші введені користувачем інформаційні дані.

Отже, аналіз стану здоров'я означає вивчення цих даних для виявлення патернів, тенденцій та змін у стані здоров'я користувача. Метою цього аналізу є надання користувачам зрозумілої та корисної інформації щодо їхнього здоров'я, а також виявлення можливих причин змін у стані для покращення загального благополуччя.

Відстеження стану - це систематичне спостереження та реєстрації різних показників здоров'я, що здійснюється за допомогою мобільного додатка. Цей процес включає в себе постійний моніторинг різних факторів, таких як фізична активність, біометричні вимірювання та дані про прийом ліків.

Відстеження стану спрямоване на постійне спостереження за фізичним та загальним станом користувача з метою виявлення патернів, змін і тенденцій. Ця практика надає користувачам докладну інформацію про їхній поточний стан, що може допомогти їм приймати більш осмислені рішення для поліпшення їхнього загального благополуччя. Важливим аспектом відстеження стану є також можливість надання рекомендацій з підтримання здоров'я на основі отриманих даних.

Результати аналізу – це інформація про зміни у стані пацієнта та рекомендації, які могли б покращити стан пацієнта або допомогти з виявленням причини погіршення стану пацієнта. Інформація про зміни у стані пацієнта надається у вигляді візуалізації змін показників протягом деякого часу. Рекомендації надаються у вигляді тексту, який містить в собі інформацію про можливі причини зміни показників пацієнта, а також рекомендації щодо нормалізування стану, наприклад поради щодо зміни режиму, поради щодо харчування, рекомендації щодо відвідування лікаря.

Якщо пацієнт проходить курс ліків, він надає інформацію про це, після чого інформація про приймання ліків надається пацієнтом регулярно. У цьому випадку до результатів аналізу здоров'я додається інформація про приймання ліків (кількість прийнятих препаратів та кількість пропущених приймань). На основі цієї інформації, якщо причиною змін у стані пацієнта може бути пропущення приймання препаратів, до рекомендацій надається відповідна інформація.

У головному бізнес-процесі бере участь пацієнт.

Пацієнт – це користувач, який хоче отримувати інформацію про своє здоров'я, а також отримувати рекомендації щодо покращення свого стану.

Проведення аналізу здоров'я регулюється Законом України «Про інформацію» [1]. Інформація про стан здоров'я – це конфіденційна інформація, і поширення такої інформації не допускається.

Для прискорення процесу аналізу здоров'я, актуально розробити застосунок для аналізу та обробки даних про здоров'я.

## 1.2 Аналіз існуючих алгоритмів аналізу здоров'я людини та існуючих систем для відстежування здоров'я

Останнім часом нейронні мережі все частіше використовуються у сфері медицини. Тому у якості існуючих алгоритмів для аналізу здоров'я людини наведено приклади використання нейронних мереж у медицині.

Згорткові нейронні мережі (CNN) використовуються для аналізу результатів медичної візуалізації [2]. Такі нейронні мережі розробляються з метою обробки зображень, тому вони чудово підходять для аналізу МРТ або рентгенівських знімків. Використання CNN, замість звичайного аналізу зображень лікарем, має переваги. По-перше, нейронні мережі перевершують точність людей. По-друге, нейронні мережі виконують аналіз швидше людини.

Основні компоненти та принципи роботи згорткових нейронних мереж для аналізу медичних зображень виглядають так [3]:

1) Згорткові шари (Convolutional Layers). Ці шари використовуються для виявлення локальних особливостей та ознак в різних частинах зображення. Згорткові фільтри проймаються зображенням, виконуючи операцію згортки, і виділяють важливі особливості, такі як краї, форми тощо.

2) Шари пулінгу (Pooling Layers). Після згорткового шару може використовуватися шар пулінгу для зменшення просторових розмірів зображення та збереження важливих інформаційних ознак. Зазвичай використовується максимальний або середній пулінг.

3) Повні з'єднані шари (Fully Connected Layers). Після декількох згорткових і пулінгових шарів, можуть слідувати один або кілька повністю

з'єднаних шарів. Ці шари використовуються для зіставлення вивчених ознак та класифікації об'єктів на зображенні.

4) Функції активації. Для нелінійності та можливості моделі вивчати більше складних залежностей, до виходів кожного шару додається функція активації, така як ReLU (Rectified Linear Unit).

5) Функція втрати (Loss Function). Використовується для оцінки того, наскільки точно модель передбачає клас об'єкта на зображенні. Для задач класифікації може використовуватися, наприклад, крос-ентропійна функція втрати.

6) Метод зворотного поширення помилок (Backpropagation). Для навчання мережі використовується оптимізатор, такий як стохастичний градієнтний спуск, і алгоритм зворотного поширення помилок. Мережа навчається шляхом зменшення функції втрати та коригування ваг усіх шарів.

Прикладом успішного застосування нейронних мереж для аналізу медичних зображень є система нейронних мереж, яку розробили китайські дослідники. Ця система мереж призначена для аналізу рентгенівських знімків для виявлення ранніх ознак пневмонії, викликаної COVID-19 [4]. Завдяки цій системі лікарі могли швидше діагностувати захворювання, що дозволяло швидше ізолювати та лікувати пацієнтів. Окрім пневмонії, нейронні мережі добре показували себе при виявленні і інших захворювань, наприклад рак, складні патології, пов'язані з мозком, фізичні травми мозку та інші захворювання.

Аналіз медичних зображень – це лише один з прикладів вдалого використання нейронних мереж у сфері медицини. Нейронні мережі також використовуються як віртуальні помічники для догляду за здоров'ям [5]. Такі помічники можуть допомагати користувачу вести записи про здоров'я, а також проводити перевірку стану здоров'я.

Одним із методів аналізу даних про здоров'я є метод ARIMA. Метод ARIMA (Autoregressive Integrated Moving Average) є ефективним інструментом

для аналізу та прогнозу часових рядів, таких як показники здоров'я з часом. ARIMA складається з трьох основних компонентів: авторегресії (AR), інтегрування (I) та ковзного середнього (MA) [6].

Авторегресія (AR) враховує автокореляцію в часовому ряді, тобто взаємозв'язок між значеннями показника з різних моментів часу. Якщо показник здоров'я має який-небудь тренд, AR допоможе виявити цю систематичну залежність.

Інтегрування (I) відповідає за видалення стаціонарності у часовому ряді. Інтегрування зазвичай використовується для видалення трендів, щоб зробити часовий ряд стаціонарним. Якщо тренд присутній, він може бути вилучений шляхом віднімання відповідних значень.

Ковзне середнє (MA) враховує ковзне середнє значень у часовому ряді для зменшення впливу шумів та випадкових величин. Якщо є сезонні коливання, MA може допомогти їх виявити.

Інколи для виявлення взаємозв'язків між різними показниками здоров'я використовуються статистичні методи, такі як коефіцієнт кореляції.

Коефіцієнт кореляції використовується для вимірювання сили та напрямку лінійного взаємозв'язку між двома змінними. У випадку аналізу показників здоров'я, це може допомогти виявити, наскільки два показники взаємодіють або корелюють між собою. Коефіцієнт кореляції може бути позначений як "r" і може приймати значення від -1 до 1:

1) Позитивний взаємозв'язок ( $r > 0$ ): Якщо r позитивний, це вказує на те, що зі збільшенням одного показника інший також збільшується. Чим ближче значення r до 1, тим сильніший позитивний взаємозв'язок.

2) Негативний взаємозв'язок ( $r < 0$ ): Якщо r негативний, це вказує на те, що зі збільшенням одного показника інший зменшується. Чим ближче значення r до -1, тим сильніший негативний взаємозв'язок.

3) Відсутність лінійного взаємозв'язку ( $r = 0$ ): Якщо r дорівнює 0, це вказує на відсутність лінійного взаємозв'язку між показниками. Однак слід зауважити,

що відсутність лінійного взаємозв'язку не означає відсутності будь-якого взаємозв'язку взагалі.

Коефіцієнт кореляції розраховується за формулою [7]:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

де:

n - кількість спостережень,

x і y – значення двох показників,

$\sum$  - сума значень.

Перевагою коефіцієнта кореляції є те, що він нормалізується, тобто не залежить від масштабу показників. Однак важливо розуміти, що кореляція не означає причинно-наслідкового зв'язку і може бути використана тільки для вимірювання ступеня лінійного взаємозв'язку. Інші типи взаємозв'язків можуть залишатися непоміченими кореляційним аналізом.

Для оцінки можливостей існуючих систем відстежування здоров'я людини, проведено аналіз існуючих мобільних застосунків для відстежування здоров'я.

Застосунок Apple Health обрано для аналізу систем для відстежування здоров'я. Цей додаток доволі поширений серед користувачів, а також має багато різноманітних функцій.

Програма «Health» дозволяє автоматично підраховувати кількість кроків та відстань, яку користувач пройшов. Якщо у користувача є годинник Apple Watch, застосунок дозволяє відстежувати активність користувача. Окрім того, користувач може власноруч вводити дані про своє здоров'я або отримувати дані з інших додатків, які сумісні з Apple Health.

На сторінці профілю (рис. 1.1) користувач може ввести дані про себе і редагувати ці данні.

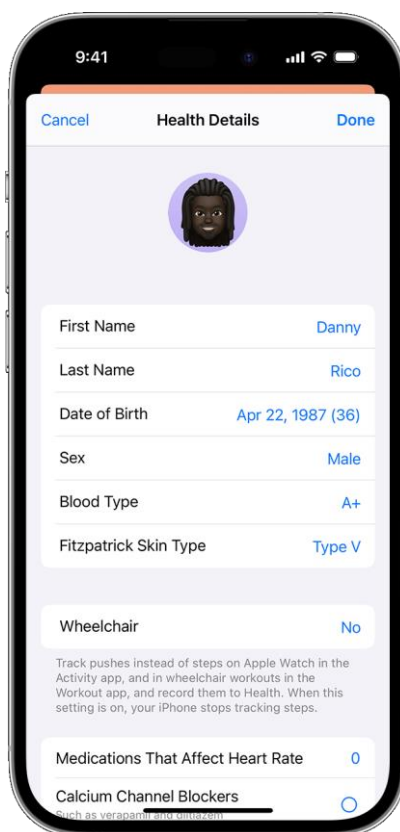


Рисунок 1.1 – Сторінка профілю користувача у застосунку Apple Health

На цій сторінці користувач може ввести такі дані, як ім'я, дату народження, стать, група крові, колір шкіри, а також різні деталі, які стосуються стану користувача. Наприклад, користувач може вказати, що користується інвалідною коляскою. Якщо користувач вказує, що користується коляскою, і при цьому він користується годинником Apple Watch, застосунок буде відстежувати не кількість кроків користувача, а кількість поштовхів коляски.

На основі даних, які користувач вводить на цій сторінці, отримуються розрахунки кількості спалених калорій, кількості кроків та подоланої відстані та активності користувача.

Користувач може редагувати категорії, які він хоче відстежувати. Це можна зробити у вкладці «Улюблені» (рис. 1.2).

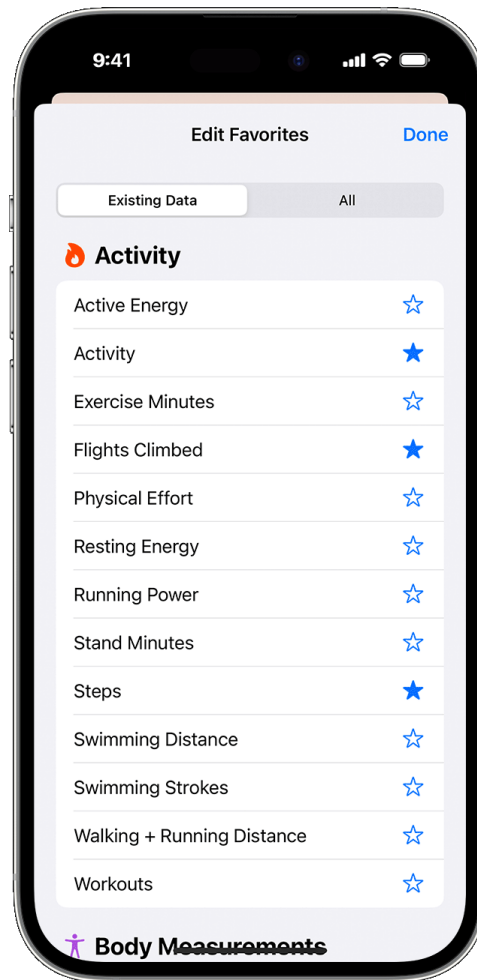


Рисунок 1.2 – Вкладка для редагування категорій у застосунку Apple Health

Таким чином, користувач завжди може обрати, або видалити потрібні йому категорії, які після цього відображаються у розділі «Обране». У цьому розділі можна переглянути щоденні показники за обраними категоріями.

Застосунок Apple Health дозволяє відстежувати зміни показників здоров'я в динаміці у вкладці «Підсумок» (рис. 1.3).

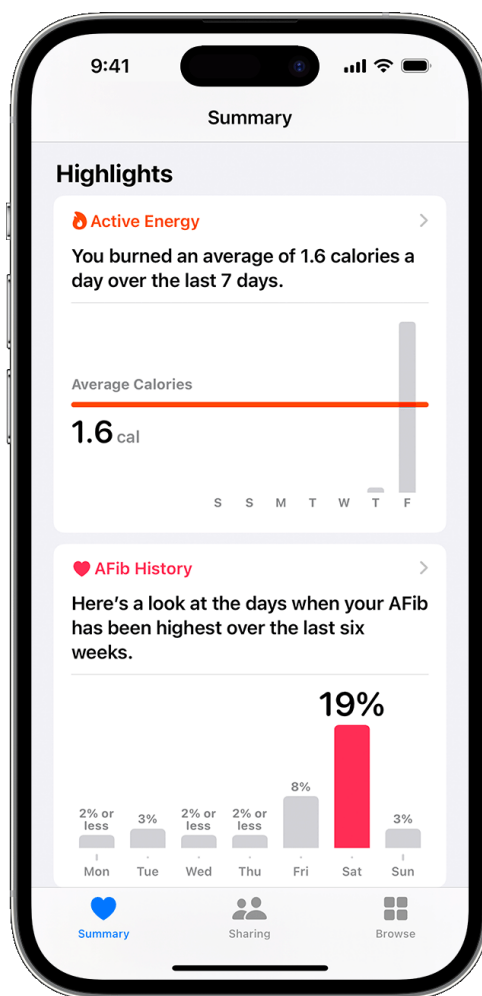


Рисунок 1.3 – Сторінка «Підсумок» у застосунку Apple Health

На цій сторінці надається можливість отримати сповіщення щодо нових тенденцій даних про здоров'я користувача. Також доступні функції «Психічне здоров'я», «Відстежування циклу» та ін.

Розглянутий застосунок є чудовою програмою для відстежування здоров'я користувачем.

Ще одним прикладом застосунку для відстежування здоров'я є застосунок MyTherapy, який нагадує прийняти користувачу ліки або контрацептиви, виміряти рівень цукру у крові або виміряти тиск. Однією із функцій застосунку є фіксування прийняття лік, завдяки чому користувач може відстежувати коли він пропустив приймання ліків.

У застосунку користувач може додавати потрібні йому нагадування (рис. 1.4), та задавати час для виконання активності.

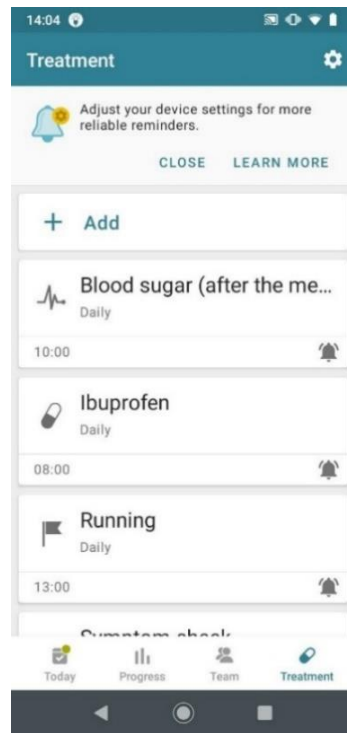


Рисунок 1.4 – Функція додавання нагадувань у застосунку MyTherapy

Для цього користувач вводить потрібні дані (рис 1.5), а саме назву нагадування, тривалість, частоту виконання, час нагадування і деяку іншу менш важливу інформацію, наприклад інформацію про налаштування будильника.

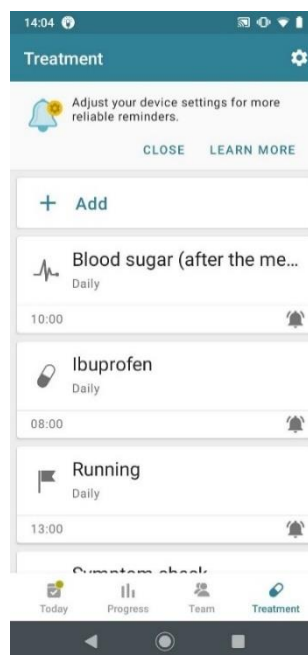


Рисунок 1.5 – Редагування нагадування у застосунку MyTherapy

Після налаштування нагадувань користувачем, він буде отримувати повідомлення.

Додаток MyTherapy – це корисна програма для тих, хто проходить курс препаратів, або для людей, які страждають хворобами, при яких такі показники, як рівень цукру, є дуже важливими для підтримування нормального стану.

Розглянуті додатки направлені на піклування про здоров'я користувача, але метою першого додатку є відстежування показників користувача, а другого – нагадування про виконання потрібних дій для збереження здоров'я. Проведений аналіз існуючих додатків показує, що розробка мобільного застосунку, який поєднував би функції цих програм, є актуальною. Крім того, в розглянутих додатках відсутня функція рекомендацій щодо здоров'я пацієнта на основі його показників, тому актуально додати таку функцію до розробленого застосунку.

Проведений аналіз існуючих алгоритмів, а саме метод ARIMA та статистичний метод, показує, що ці методи можна використати при розробці застосунку аналізу стану здоров'я людини. Ґрунтуючись на коефіцієнтах кореляції та виявлених трендах, вони допоможуть уточнювати результати аналізу стану здоров'я людини.

### 1.3 Сфера використання мобільного застосунку аналізу стану здоров'я людини

Після проведення аналізу існуючих додатків для відстежування здоров'я, можна зробити висновок, що такі додатки користуються попитом серед користувачі, навіть не дивлячись на те, що такі додатки не дозволяють аналізувати показники здоров'я з метою виявлення можливих причин у зміні стану здоров'я.

Актуально розробити мобільний застосунок, який дозволив би відстежувати показники здоров'я користувача та надавати можливість отримувати рекомендації щодо можливих причин у зміні стану.

Реалізація у вигляді мобільного застосунку – це спосіб зробити отримання рекомендацій більш зручним і доступним, тому що завдяки реалізації у такому вигляді користувач може вводити інформацію про свої показники де завгодно і коли завгодно.

Окрім того, такий застосунок зможе допомогти лікарям більш точно ставити діагнози пацієнтам. Користувач може надавати інформацію із застосунку, завдяки чому лікар зможе переглянути графіки зміни показників, інформацію про приймання ліків пацієнтом і результати аналізу програми.

#### 1.4 Постановка задачі

Виходячи з аналізу предметної області та аналізу існуючих застосунків, виявлено, що актуально розробити застосунок, який поєднував би функціонал існуючих програм, а також розробити алгоритми, які зможуть вчасно виявляти зміни в стані користувача і робити висновки щодо змін у стані здоров'я користувача на основі цих змін. Таким чином, метою роботи є розробка мобільного застосунку для аналізу та обробки даних про здоров'я користувача.

Для розробки такого застосунку необхідно визначити план послідовності роботи.

Доступ до функцій програми надаватиме застосунок, який буде реалізовано у вигляді мобільного застосунку.

Користувач системи – це людина(пацієнт), яка хоче отримувати дані про своє здоров'я на основі показників, відстежувати свій стан, або отримувати сповіщення, які допоможуть підтримувати стан здоров'я пацієнта.

Мета розробки застосунку для аналізу та обробки даних про здоров'я:

а) з боку організації:

- 1) автоматизація процесу аналізу здоров'я;
- 2) виявлення можливих причин зміни стану здоров'я за допомогою алгоритмів;

3) підвищення кількості користувачів за рахунок якісного аналізу здоров'я;

б) з боку пацієнта:

1) надання можливості відстежувати показники здоров'я. Зробити процес відстежування показників зручним та доступним. Надання можливості отримувати візуалізацію показників здоров'я;

2) прискорення процесу виявлення можливих причин зміни стану здоров'я з використанням алгоритмів для аналізу здоров'я;

3) надання можливості стежити за прийманням ліків;

4) зробити процес відстежування стану здоров'я зрозумілим і зручним у використанні;

Для досягнення цієї мети, потрібно вирішити такі задачі:

- Дослідження алгоритмів аналізу стану здоров'я;
- розробка системних та функціональних вимог до застосунку для обробки та аналізу стану здоров'я;
- розробка алгоритму для аналізу стану здоров'я;
- розробка вимог до інтерфейсу застосунку для обробки та аналізу стану здоров'я;
- логічне та фізичне моделювання даних ;
- розробка серверної частини застосунку;
- розробка інтерфейсу застосунку;
- реалізація алгоритму для аналізу стану здоров'я;
- тестування розробленого застосунку;

## 2 РОЗРОБКА ВИМОГ ДО ЗАСТОСУНКУ АНАЛІЗУ СТАНУ ЗДОРОВ'Я ЛЮДИНИ

### 2.1 Визначення системних вимог до застосунку аналізу стану здоров'я людини

В ході роботи виявлено системні вимоги до застосунку аналізу стану здоров'я людини.

Застосунок для обробки та аналізу даних про здоров'я за класифікацією по типу архітектури представляє собою систему «клієнт-сервер» [8]. Такий тип архітектури дозволяє розподіляти навантаження між постачальником послуг(сервером) і замовниками послуг(клієнтами). Взаємодія між клієнтом і сервером реалізується завдяки запитам, які користувач відправляє на сервер, після чого від сервера приходять відповідь у вигляді даних або функцій.

Застосунок для обробки та аналізу даних про здоров'я - це інформаційна система, яка призначена для обробки інформації та забезпечення користувачів потрібною інформацією. Ця система представляє собою автоматизовану систему, її головна мета надати рішення на основі автоматизації інформаційних процесів.

Дані в системі представлені у вигляді таблиць, які представляють із себе базу даних.

Застосунок, який розроблюється, призначений для автоматизації процесу обробки даних про здоров'я і виявлення можливих причин зміни стану здоров'я.

На основі мети використання застосунку, виділено системні вимоги до нього:

- серверна частина застосунку для аналізу та обробки даних про здоров'я повинна зберігати інформацію та функції для виконання логіки системи;

- клієнтська частина застосунку для аналізу та обробки даних про здоров'я повинна бути реалізована у вигляді мобільного застосунку, який надає доступ до усіх функцій системи;

- інтерфейс доступу повинно бути реалізовано у вигляді функцій, доступ до яких надається з клієнтського застосунку;
- застосунок для обробки та аналізу даних про здоров'я повинен зберігати усі потрібні дані та обробляти їх;

## 2.2 Визначення функціональних вимог до застосунку аналізу стану здоров'я людини

Для виявлення функціональних вимог до застосунку для обробки та аналізу даних про здоров'я розроблено функціональну модель IDEF0. IDEF0 – це методологія функціонального моделювання, призначенням цієї методології є формування та опис бізнес-процесів [9].

Основний бізнес-процес - «Проведення аналізу здоров'я», для опису цього бізнес-процесу розроблена концептуальна діаграма (рис. 2.1).

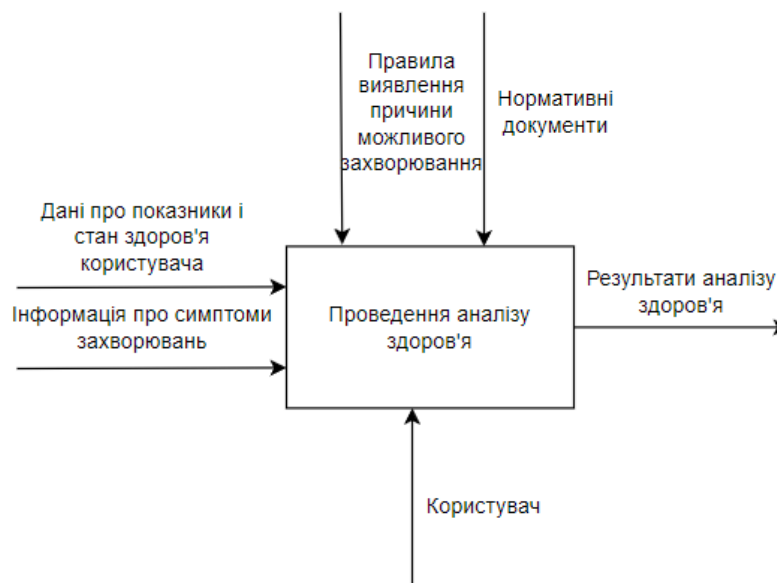


Рисунок 2.1 – Концептуальна діаграма бізнес-процесу «Проведення аналізу здоров'я»

Дані про показники і стан здоров'я користувача та інформація про симптоми захворювань – це вхідна інформація. Правила виявлення причини можливого захворювання і нормативні документи виступають в якості управлінських впливів. Механізмом організації функціонування є користувач. Результати аналізу здоров'я – це вихідні дані.

Для деталізації основного бізнес-процесу було проведено декомпозицію концептуальної діаграми (рис. 2.2), в ході чого виявлено функції системи.

Процес проведення аналізу стану здоров'я починається зі збору даних про показники стану здоров'я користувача. Дані про показники надає користувач на основі вимірювань цих показників (температура, тиск, пульс, рівень цукру у крові, сатурація). Наступним кроком проводиться обробка даних про здоров'я на основі зібраних показників. В ході обробки даних отримується інформація про тенденцію показників. Далі проводиться аналіз показників, в ході аналізу отримується інформація про можливі причини зміни показників і формується звіт з результатами.

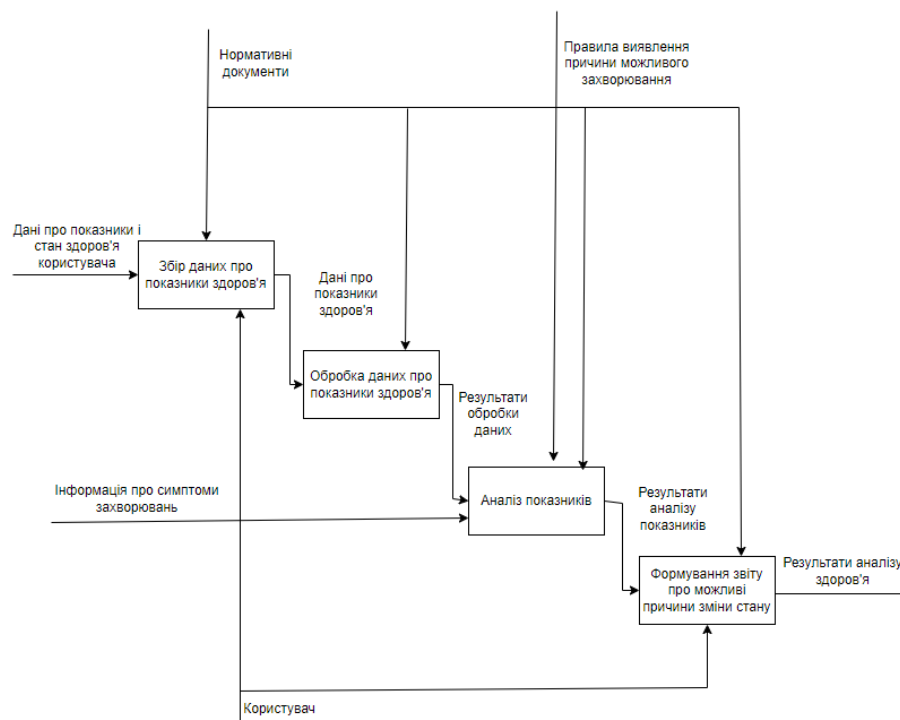


Рисунок 2.2 – Діаграма декомпозиції головного бізнес-процесу

Після виявлення функцій системи, проведено декомпозицію функції «Аналіз показників» (рис 2.3).

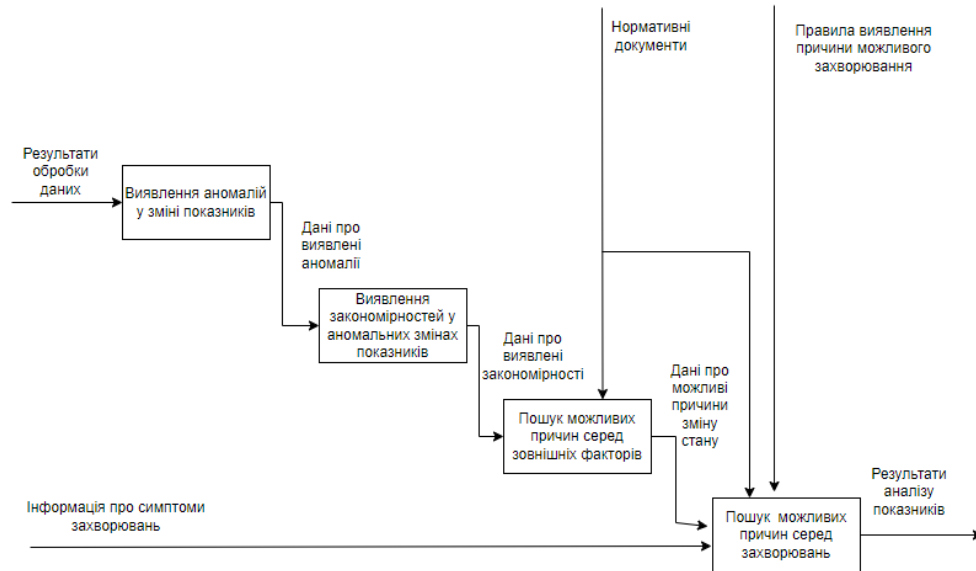


Рисунок 2.3 – Декомпозиція функції «Аналіз показників»

Функція «Аналіз показників» реалізує алгоритм, за допомогою якого отримуються дані про можливі причини зміни стану здоров'я. Процес аналізу показників починається з виявлення аномалій у зміні показників на основі тенденцій. В процесі виявлення аномалій порівнюються показники пацієнта за деякий відрізок часу враховуючи похибку кожного показника, на основі чого робляться висновки щодо різкої зміни у показниках. Далі ці дані використовуються для виявлення закономірностей у різких змінах показників. В ході цього процесу отримується інформація про показники, які можуть бути пов'язані один з одним (наприклад, якщо зміна у показниках ваги пацієнта і температури тіла пацієнта починається одночасно, то вони можуть бути пов'язані і викликаються якимось одним захворюванням).

Наступним кроком проводиться пошук можливих причин зміни показників здоров'я серед зовнішніх факторів, наприклад побічні ефекти ліків,

або кліматичні явища. Після виявлення можливих причин серед зовнішніх факторів, проводиться пошук можливих причин серед захворювань на основі симптомів, які викликають ці захворювання. На виході отримується інформація про результати аналізу показників.

### 2.3 Опис алгоритму аналізу стану здоров'я людини

Під час виконання аналізу здоров'я, виконуються 4 основних кроки:

- Виявлення аномалій у зміні показників;
- Виявлення закономірностей у аномальних змінах показників;
- Пошук причин серед зовнішніх факторів;
- Пошук можливих причин серед захворювань.

Виявлення аномалій у зміні показників – це етап, на якому вхідні дані(показники) аналізуються та знаходиться взаємозв'язок між значеннями показника з різних моментів часу.

Норма таких показників, як температура, рівень цукру, тиск та пульс, різна у кожної людини. Також деякі показники можуть змінюватися протягом дня. Наприклад, температура тіла людини протягом доби змінюється у невеликих межах, залишаючись у діапазоні приблизно від 35,5 до 37,2 °С. Окрім того, показники можуть короткочасно змінюватися у результаті фізичних навантажувальних, приймання їжі, стресу та інших ситуаціях. Тож, головною задачею на цьому етапі є виявлення аномалій серед показників, враховуючі незначні зміни, на які не слід звертати увагу.

Для пошуку аномалій використовується середнє та стандартне відхилення.

Середнє (середнє арифметичне) - це статистичний показник, який представляє собою суму всіх значень в наборі, поділену на їх кількість. Щоб знайти середнє, потрібно додати всі значення та поділити суму на кількість значень. Формула має вигляд [10]:

$$M_1 = \frac{x_1^p + x_2^p + \dots + x_n^p}{n}$$

Стандартне відхилення - це міра розподілу значень в наборі даних відносно їх середнього значення. Високе стандартне відхилення вказує на велику різницю між кожним значенням і середнім значенням, що свідчить про велику дисперсію в даних. Низьке стандартне відхилення, навпаки, вказує на те, що значення тісно групуються навколо середнього значення. Формула для стандартного відхилення виглядає так:

$$S = \sqrt{\frac{\sum_{i=1}^n (X_i - M_1)^2}{n}}$$

де:

$X_i$  - кожне окреме значення в наборі даних,

$M_1$  – середнє арифметичне значення,

$n$  – кількість значень в наборі.

Виявлення закономірностей у аномальних змінах показників – це етап, на якому знаходиться взаємозв'язок між різними показниками. Для знаходження взаємозв'язку використовується коефіцієнт кореляції.

Деякі захворювання вражають відразу декілька органів або систем у організмі людини, і це призводить до зміни відразу декількох показників. При зміні двох показників, важливо виявити чи є взаємозв'язок між цією зміною. Якщо взаємозв'язок є – то, скоріш за все, зміни у стані людини викликаються одним захворюванням. Якщо взаємозв'язка немає, причини зміни показників різні.

Існують різні методи обчислення коефіцієнта кореляції, але основний призначений для вимірювання лінійної залежності - це коефіцієнт Пірсона.

Для знаходження коефіцієнта кореляції потрібно виконати наступні кроки:

1. Обчислити коефіцієнт кореляції Пірсона [11]:

$$r = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}}$$

2. Інтерпретування результатів. Коефіцієнт кореляції приймає значення від -1 до 1. Значення близьке до 1 вказує на сильну позитивну лінійну залежність, -

1 - сильну від'ємну лінійну залежність, а 0 - відсутність лінійної залежності. Далі отримані дані використовуються у подальшому пошуку причин зміни стану.

Пошук причин серед зовнішніх факторів. На цьому етапі проводиться пошук можливих причин у зміні показників серед зовнішніх факторів, таких як реакція організму на зміну погоди, побочні ефекти приймання ліків або погіршення стану на фоні пропуску приймання ліків.

Взагалі, у Міжнародній класифікації хвороб, якою керуються лікарі під час постановки діагнозів, немає термінів «метеочутливість» або «метеозалежність», тобто якщо у людини немає хронічних захворювань і людина дотримується здорового образу життя, зміна погоди не може впливати на її самопочуття. Але є особи, які гостро реагують на температурні перепади, а саме особи з серцево-судинними захворюваннями [12]. Наприклад, при підвищенні температури (як буває в літню спеку) – судини розширюються, кров приливає до тканин та органів, що спричиняє додаткове навантаження на серце. Артеріальний тиск може впасти, що викликає такі симптоми, як запаморочення, потемніння в очах, задишку, холодний піт [13]. Холод також негативно впливає на таких людей. Звуження судин збільшує навантаження на серцевий м'яз, артеріальний тиск зростає.

Окрім людей з серцево-судинними захворюваннями, зміна погоди може впливати на алергіків. Зміни в погодних умовах можуть впливати на концентрацію алергенів у повітрі, що може призводити до алергічних реакцій.

Температура повітря також може впливати на загальний стан людини, екстремальна спека може привести до перегріву, а холод – до гіпотермії (переохолодження).

Приймання ліків може викликати негативні реакції організму. В залежності від препарату, порушення можуть бути з боку травної системи, з боку дихальної системи, з боку нервової системи, з боку серцево-судинної системи та сечовидільної системи. У контексті аналізу здоров'я на основі показників,

викликають інтерес препарати, які можуть впливати на температуру тіла, тиск, рівень цукру у крові і частоту пульсу.

Зміну температури тіла можуть викликати такі препарати, як антибіотики, антипіретики, антидепресанти, м'язові релаксанти [14].

Деякі лікарські засоби можуть впливати на рівень цукру у крові, особливо у людей з діабетом чи тими, хто має підвищений ризик розвитку цукрового діабету. До таких лікарських засобів входять глюкокортикостероїди, гормональні препарати, антипсихотичні ліки, антидепресанти, бета-блокатори [15].

Декілька груп лікарських засобів можуть впливати на артеріальний тиск, іноді призводячи до підвищення (гіпертензія) або зниження (гіпотензія) тиску. Це такі засоби, як антигіпертензивні препарати, антиаритмічні засоби, деякі антидепресанти, амфетаміни та стимулятори центральної нервової системи [16]. Важливо зазначити, що вплив препаратів на артеріальний тиск може варіювати в залежності від індивідуальних особливостей пацієнта та інших факторів.

На частоту пульсу можуть впливати бета-блокатори, антиаритмічні препарати, антигіпертензивні препарати, стимулятори центральної нервової системи, щитовидні гормони та інші препарати [17].

Таким чином, на цьому етапі головною задачею є пошук причин зміни показників користувача серед вище перелічених факторів.

Для того, щоб виявити чи були погодні умови причиною зміни стану здоров'я, першим кроком потрібно отримати дані про зміну погоди у місті, яке користувач вказав як місце свого проживання. Після цього використовуються дані, які були отримані на першому і другому етапі, а саме дані про аномальні зміни показників и дані про кореляцію між показниками, які змінилися. Далі дані про показники порівнюються з симптомами впливу погоди на стан людини, які було розглянуто вище. Якщо є збіги, то ця інформація буде відображатися у звіті.

Під час пошуку причин у зміні показників серед побічних ефектів ліків, використовуватиметься база даних ліків МОЗ. Спочатку здійснюється пошук

діючих речовин ліків, які приймає користувач. Інформація про назву ліків береться з нагадувань, які створював користувач. Після цього порівнюються побочні ефекти діючих речовин ліків зі змінами у стані, які виявлено на першому і другому етапі. Також враховується інформація про пропуск приймання ліків користувачем, і здійснюється пошук за симптомами синдрому відміни від ліків.

У MySQL для пошуку записів за текстовими параметрами можна використовувати повнотекстовий пошук або простий пошук з використанням операторів «LIKE».

4) Пошук можливих причин серед захворювань. На цьому етапі проводиться пошук захворювань, які можуть викликати зміну у стані здоров'я людини, які впливають на один або декілька показників.

Етап 4 схожий на етап 3. На цьому етапі, як і на попередньому, потрібно вести пошук захворювань і знаходити ті захворювання, які викликають симптоми, які є у користувача. Для пошуку захворювань за симптомами використовується база даних Diseases Database [18].

Деякі захворювання можуть викликати зміну одразу декількох показників. У якості прикладу наведено таке захворювання, як системний червоний вовчак. Це аутоімунне захворювання, яке може впливати на різні органи та системи, включаючи суглоби, шкіру, нирки, серце та інші. У ході впливу цієї хвороби на різні системи, показники температури тіла і рівню цукру у крові можуть змінюватися [19]. Тож, якщо рівень цукру і температура тіла користувача змінювалась, і на етапі 2 була виявлена кореляція між цими показниками, потрібно внести цю хворобу до звіту і вказати, що вона може бути причиною зміни стану здоров'я.

Далі на рисунку 2.4 зображена блок-схема розробленого алгоритму аналізу стану здоров'я користувача.

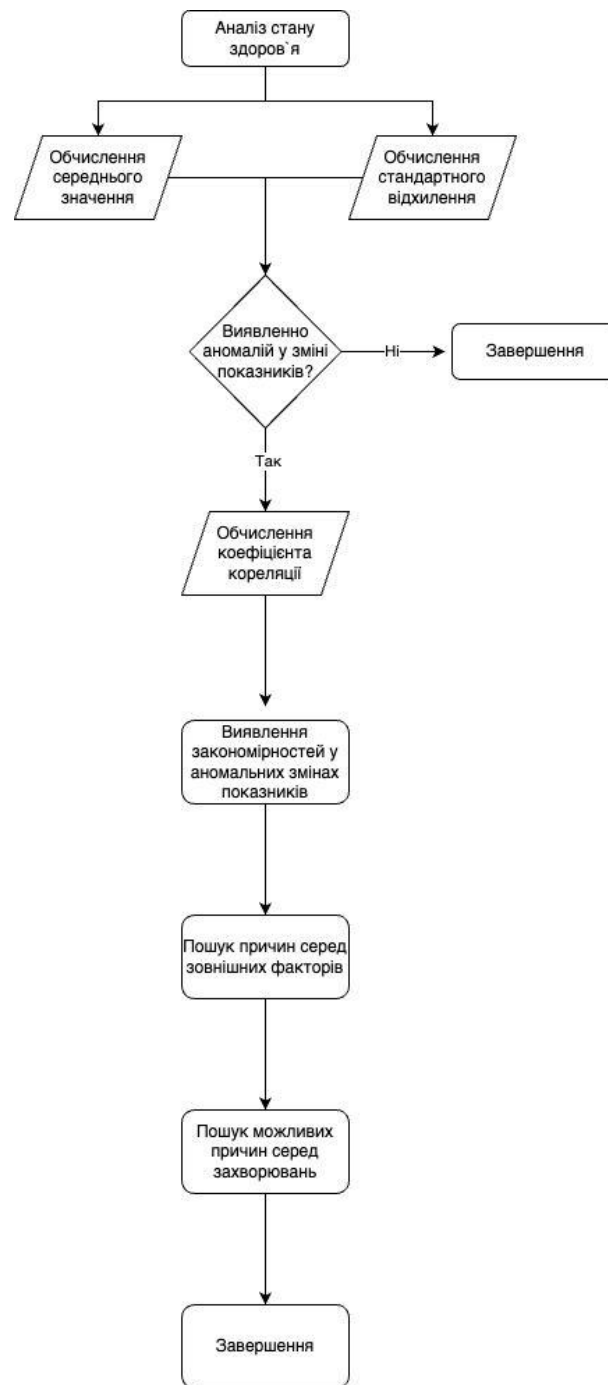


Рисунок 2.4 – Блок-схема розробленого алгоритму.

## 2.4 Діаграма варіантів використання застосунку аналізу стану здоров'я людини

Для графічного зображення функціональності застосунку з точки зору користувача і конкретних сценаріїв використання розроблено діаграму варіантів використання (Use Case Diagram) [20]. Вона використовується для моделювання,

опису і розуміння того, як система взаємодіє з її користувачами або іншими системами.

Основні призначення use case діаграми включають:

- Опис функціональності системи. Use case діаграма надає високорівневий огляд функцій, які система може виконувати. Кожен use case представляє конкретний функціональний аспект системи.

- Визначення взаємодії з користувачами. Діаграма дозволяє ідентифікувати акторів (користувачів або інші системи), які взаємодіють із системою, та показує, які конкретні дії вони можуть виконувати.

- Уточнення сценаріїв використання. Кожен use case може мати декілька сценаріїв використання, які описують різні шляхи взаємодії між користувачем і системою. Це допомагає деталізувати функціональні вимоги.

- Забезпечення бази для тестування. Use case діаграма може слугувати як важлива основа для тестування системи. Кожен use case може бути перевірений на відповідність функціональним вимогам.

- Полегшення розуміння системи. Графічне представлення взаємодії між системою та користувачами допомагає краще розуміти логіку та обсяг роботи системи.

Тож, для аналізу взаємодії користувача з системою розроблено діаграму варіантів використання (рис. 2.5), на якій зображено акторів, взаємовідношення і прецеденти, які зображають основні функції системи.

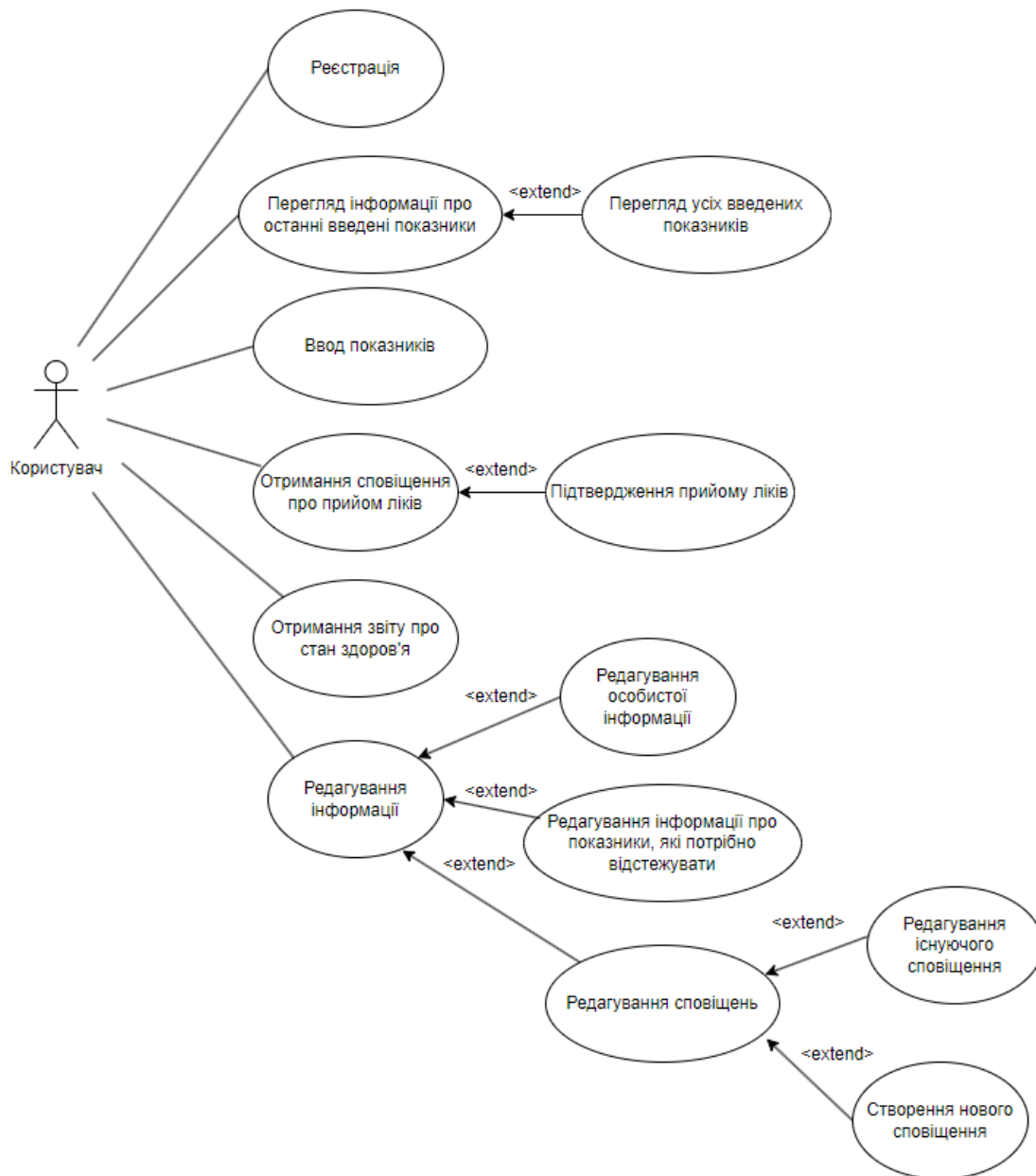


Рисунок 2.5 – Діаграма варіантів використання застосунку аналізу стану здоров'я людини

Користувач – це актор, який користується застосунком і головна мета якого - слідувати за своїм здоров'ям та отримувати звіти з інформацію про стан здоров'я.

Користувач має доступ до таких функцій:

- реєстрація. Функція, яка дозволяє користувачу вносити дані про себе для подальшої обробки цих даних;

- перегляд інформації про останні введені показники. Дозволяє переглянути показники, які вводились користувачем у попередній раз. Функція «Перегляд усіх введених показників» розширює цю функцію, дозволяючи переглянути усі записи показників, які колись вводив користувач.

- введення показників. Ця функція надає можливість користувачу ввести дані про показники, які він відстежує;

- отримання сповіщення про приймання ліків. Функція, яка нагадує користувачу про приймання ліків. Ця функція розширюється функцією «Підтвердження приймання ліків», яка дозволяє користувачу відмітити приймання ліків. Загалом, ці функції потрібні, по-перше, для відстежування користувачем курсу ліків, які йому потрібно приймати, і, по-друге, для подальшого використання даних про приймання ліків при аналізі стану здоров'я, а саме пошук кореляції між зміною показників і побічними ефектами ліків (або ефектами, які викликані пропуском приймання ліків);

- отримання звіту про стан здоров'я. Ця функція надає можливість користувачу отримати результати аналізу здоров'я на основі введених показників, загальної інформації про стан користувача, інформації щодо препаратів, які приймає користувач, і іншої інформації, яка враховується при проведенні аналізу здоров'я;

- редагування інформації. Надає можливість змінювати інформацію, яка використовується при аналізі здоров'я. Включає в себе функцію редагування особистої інформації, яка дозволяє користувачу змінювати інформацію про себе і свій загальний стан здоров'я, і функцію редагування інформації про показники, які потрібно відстежувати, яка дозволяє користувачу обрати або видалити показники, за якими він хоче спостерігати. Також включає в себе функцію редагування сповіщень, яка, в свою чергу включає в себе функції «Редагування існуючих сповіщень» і «Створення нового сповіщення». Ці функції дозволяють додати нагадування про приймання ліків, або відредагувати все існуюче нагадування.

## 2.5 Вимоги до інтерфейсу клієнтської частини мобільного застосунку аналізу стану здоров'я людини

Застосунок аналізу стану здоров'я людини представляє із себе мобільний застосунок під платформу Android. Розробка мобільних застосунків на платформі Android має кілька переваг [21]:

- широке аудиторне охоплення. Android є найпопулярнішою мобільною платформою у світі, що означає, що можна досягти великої аудиторії користувачів;
- різноманітність пристроїв. На Android доступна широка різноманітність пристроїв різних виробників та специфікацій. Це важливо для тестування та забезпечення сумісності застосунку з різними моделями та розмірами екранів;
- відкритість та гнучкість. Android - це операційна система з відкритим вихідним кодом, що дозволяє розробникам вносити зміни до коду та налаштовувати систему під свої потреби;
- розширені можливості. Android надає розробникам широкий набір API для створення різноманітних та інноваційних застосунків, таких як розпізнавання голосу, робота з розташуванням, використання сенсорів та багато іншого.

Перший пункт у цьому списку є дуже важливим, бо, виходячи з профілю користувача, який буде представлено нижче, важливо охопити широке коло користувачів.

При побудові профілю користувача, виділено наступні риси:

- вік від 15 до 60 років;
- основна мова – українська;
- елементарний рівень комп'ютерної грамотності;
- головна мета користувача – стежити за показниками свого здоров'я та отримувати звіти з аналізом здоров'я;

- передбачається, що користувач має смартфон з Android.

На основі аналізу існуючих систем та розробленого профілю користувача, виділено наступні вимоги до інтерфейсу застосунку аналізу стану здоров'я людини:

- зрозумілий інтерфейс для користувача, який володіє елементарним рівнем комп'ютерної грамотності;
- доступ до функцій, які дозволяють стежити за здоров'ям;
- дотримання стилю, виділення елементів інтерфейсу, які є важливими;
- можливість відмінити та підтвердити дію.

2.6 Компоненти мобільного застосунку аналізу стану здоров'я, які підлягають розробці

На основі виявлених функціональних вимог та розробленій діаграмі використання, виявлено компоненти застосунку аналізу стану здоров'я людини, які підлягають розробці:

а) з боку серверної частини:

- 1) доступ до бази даних;
- 2) створення контролерів, що реалізують відправлення запитів до бази даних.

б) з боку клієнтської частини:

- 1) реалізація сторінок, які надають доступ до функцій;
- 2) реалізація функцій, що виконують логіку;
- 3) реалізація алгоритму для аналізу здоров'я.

в) елементи застосунку:

- 1) сторінка реєстрації;
- 2) сторінка з відображенням останніх введених показників і доступом до основних функцій;

- 3) сторінка «Налаштування», де можна відредагувати особисту інформацію, створити нагадування і обрати показники, які потрібно відстежувати;
- 4) сторінка для перегляду усіх введених показників;
- 5) сторінка для підтвердження приймання ліків;
- 6) сторінка зі звітом про стан здоров'я.

г) функціонал:

1) якщо користувач створив нагадування про приймання ліків, він повинен отримувати спливаючі повідомлення у час, який вказано у нагадуванні. У користувача має бути можливість підтвердити приймання ліків;

2) користувач повинен мати доступ до показників, які він вводив. На головній сторінці повинні відображатися показники, які користувач вводив останніми. Також у користувача має бути доступ до списку усіх раніше введених показників, має бути можливість відобразити показники у вигляді графіку;

3) у користувача має бути можливість редагувати інформацію про себе. Повинен надаватися доступ як до редагування особистої інформації, так і до редагування інформації щодо нагадувань, які створював користувач;

4) користувач повинен мати можливість додати показники, які він хоче відстежувати. А також видалити ті показники, які він не хоче відстежувати;

5) користувач повинен мати доступ до функції отримання звіту з аналізом його показників. До звіту повинна входити інформація, яка була отримана у результаті аналізу. Це інформація про погіршення показників, інформація про кореляцію між цими показниками, інформація про кількість пропущених днів приймання ліків, можливі причини погіршення

стану серед зовнішніх факторів (погоді, побочні ефекти ліків), можливі причини серед захворювань.

## 3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Обґрунтування вибору технології створення мобільного застосунку аналізу стану здоров'я людини

При розробці мобільного застосунку аналізу стану здоров'я людини використовується мова програмування Kotlin. Kotlin – це об'єктно-орієнтована мова зі статичною типізацією. Потенційно Kotlin можна використовувати скрізь, де працює Java, але частіше за все ця мова використовується при розробці мобільних застосунків для Android [22]. На сьогоднішній день більшість найпопулярніших програм, які розміщено у Play Market, написано за допомогою мови Kotlin. Враховуючи вищесказане, можна сказати, що обрана мова підходить для досягнення мети, а саме створення мобільного застосунку аналізу стану здоров'я людини для Android.

Для побудування UI використовується фреймворк Jetpack Compose, побудований на базі мови програмування Kotlin. Цей фреймворк дозволяє спростити розробку користувацького інтерфейсу для мобільних застосунків під Android за рахунок меншої кількості коду, потужних інструментів та інтуїтивно зрозумілого синтаксису на основі Kotlin.

Для реалізації архітектури «клієнт-сервер» використовується платформа Node.js. Важливою особливістю Node.js є асинхронний характер [23]. Це означає, що сервер, створений з використанням Node.js, не повинен чекати, поки дані повернуться, при виконанні різних внутрішніх запитів.

Для спрощення взаємодії між віддаленим API з застосунком, використовується бібліотека Retrofit. Retrofit - це бібліотека для роботи з HTTP-запитами в Android-застосунках. Основна її задача - спростити процес взаємодії з віддаленими серверами через HTTP-протокол. Retrofit дозволяє визначити інтерфейс для віддаленого API. Описуються методи цього інтерфейсу, які відповідають різним HTTP-запитам (GET, POST, PUT, DELETE тощо), і Retrofit автоматично генерує реалізацію для цих запитів. Також Retrofit дозволяє

налаштувати різні параметри HTTP-запитів, такі як заголовки, параметри, методи передачі даних (наприклад, JSON або форма) і т.д [24]. Retrofit підтримує асинхронне виконання HTTP-запитів за допомогою об'єктів «Call» та інтерфейсу «Callback». Це дозволяє виконувати мережеві операції в окремому потоці і не блокувати основний потік користувача.

### 3.2 Обґрунтування вибору СУБД

У якості платформи СУБД було обрано MySQL. СУБД MySQL є однією з найпопулярніших та найбільш використовуваних відкритих систем управління базами даних [25]. Використання MySQL при розробці мобільного застосунку має кілька переваг:

- надійність та стабільність. MySQL відомий своєю надійністю та стабільністю. Це важливо для мобільних застосунків, які можуть працювати в умовах обмеженого з'єднання з Інтернетом або на мобільних пристроях з обмеженими ресурсами;

- ефективність та продуктивність. MySQL є швидкою та продуктивною базою даних, що може бути важливим для оптимізації швидкодії мобільного застосунку та забезпечення швидкої відповіді на запити користувачів;

- масштабованість. MySQL підтримує масштабованість, що означає, що можна легко розширювати базу даних, якщо обсяг даних або кількість користувачів зростає;

- кросплатформенність. Більшість мобільних платформ (Android та iOS) підтримують використання MySQL. Це означає, що можна використовувати одну базу даних для підтримки мобільного застосунку на різних пристроях та операційних системах.

### 3.3 Логічне і фізичне моделювання даних застосунку аналізу стану здоров'я людини. Створення бази даних

Для подання структури застосунку аналізу даних здоров'я людини, проведено логічне моделювання даних (рис. 3.1). Для цього використано програму Draw.io, яка дозволяє створювати діаграми.

Логічна модель даних дозволяє виявити елементи даних, їх структура, а також зв'язок та відношення між цими елементами.

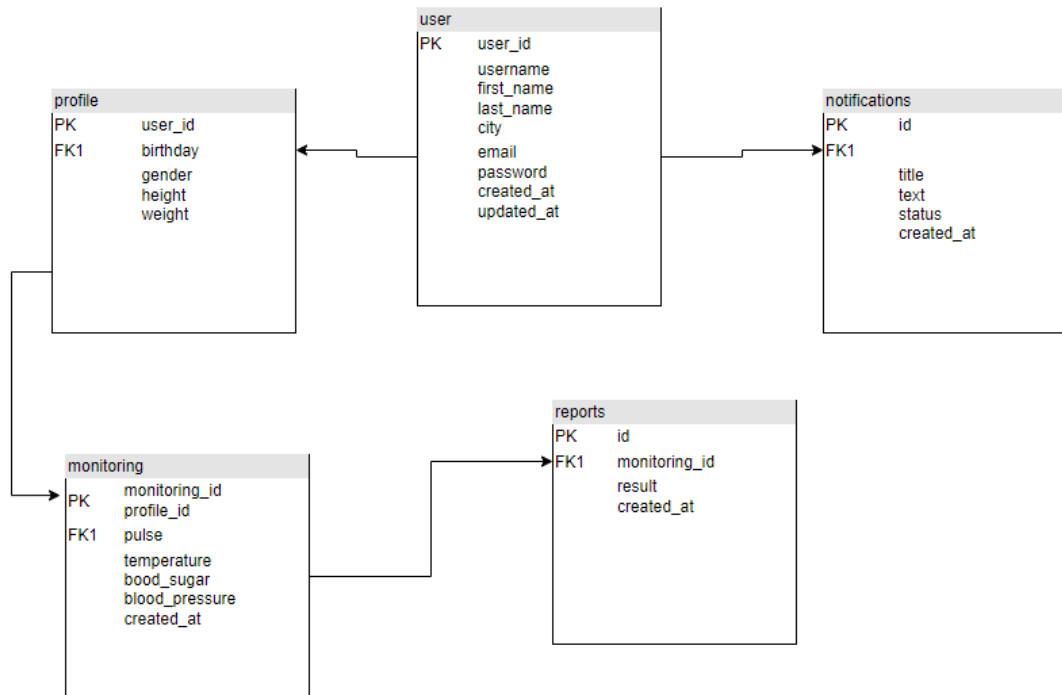


Рисунок 3.1 – Логічна модель даних застосунку аналізу стану здоров'я людини

До логічної моделі увійшли наступні сутності:

- user. Зберігає інформацію про акаунт користувача.
- profile. Зберігає інформацію про користувача, яка буде використовуватися при аналізі здоров'я.
- notifications. Зберігає інформацію про сповіщення.
- monitoring. Зберігає інформацію про показники.
- reports. Зберігає інформації про звіти щодо стану здоров'я.

До логічної моделі увійшли атрибути сутностей, які представлено у таблиці 3.1.

Таблиця 3.1 – Сутності та атрибути сутностей

№	Найменування сутності	Найменування атрибутів	Тип даних	Призначення
1	user	user_id	Лічильник, довге ціле	Первинний ключ
		username	Текст, довжиною 50 символів	Назва користувача
		first_name	Текст, довжиною 50 символів	Прізвище користувача
		last_name	Текст, довжиною 50 символів	Ім'я користувача
		city	Текст, довжиною 50 символів	Місто
		email	Текст, довжиною 50 символів	Адреса електронної пошти
		password	Текст, довжиною 32 символи	Пароль
		created_at	Дата та час	Дата створення акаунту
		updated_at	Дата та час	Дата оновлення акаунту
2	profile	profile_id	Лічильник, довге ціле	Первинний ключ
		user_id	Довге ціле	Зовнішній ключ
		birthday	Дата	Дата народження користувача
		gender	Логічний тип	Стать користувача
		height	Довге ціле	Зріст користувача
		weight	Довге ціле	Вага користувача
3	notifications	id	Лічильник, довге ціле	Первинний ключ
		user_id	Довге ціле	Зовнішній ключ
		title	Текст, довжиною 50 символів	Назва нагадування
		text	Текст, довжиною 300 символів	Опис нагадування
		status	Логічний тип	Статус нагадування
		created_at	Дата та час	Дата створення нагадування
4	monitoring	monitoring_id	Лічильник, довге ціле	Первинний ключ

## Продовження таблиці 3.1 – Сутності та атрибути сутностей

		profile_id	Довге ціле	Зовнішній ключ
		pulse	Довге ціле	Значення показника «Пульс»
		temperature	Число з плаваючою комою	Значення показника «Температура»
		blood_sugar	Число з плаваючою комою	Значення показника «Рівень цукру у крові»
		blood_pressure	Текст, довжиною 10 символів	Значення показника «Тиск»
		created_at	Дата та час	Дата запису показників
5	reports	id	Лічильник, довге ціле	Первинний ключ
		monitoring_id	Довге ціле	Зовнішній ключ
		result	Текст, довжиною 500 символів	Результати аналізу здоров'я
		created_at	Дата та час	Дата створення звіту про здоров'я

За допомогою SqlDBM створено фізичну модель даних (рис. 3.2), до якої увійшло 5 сутностей.

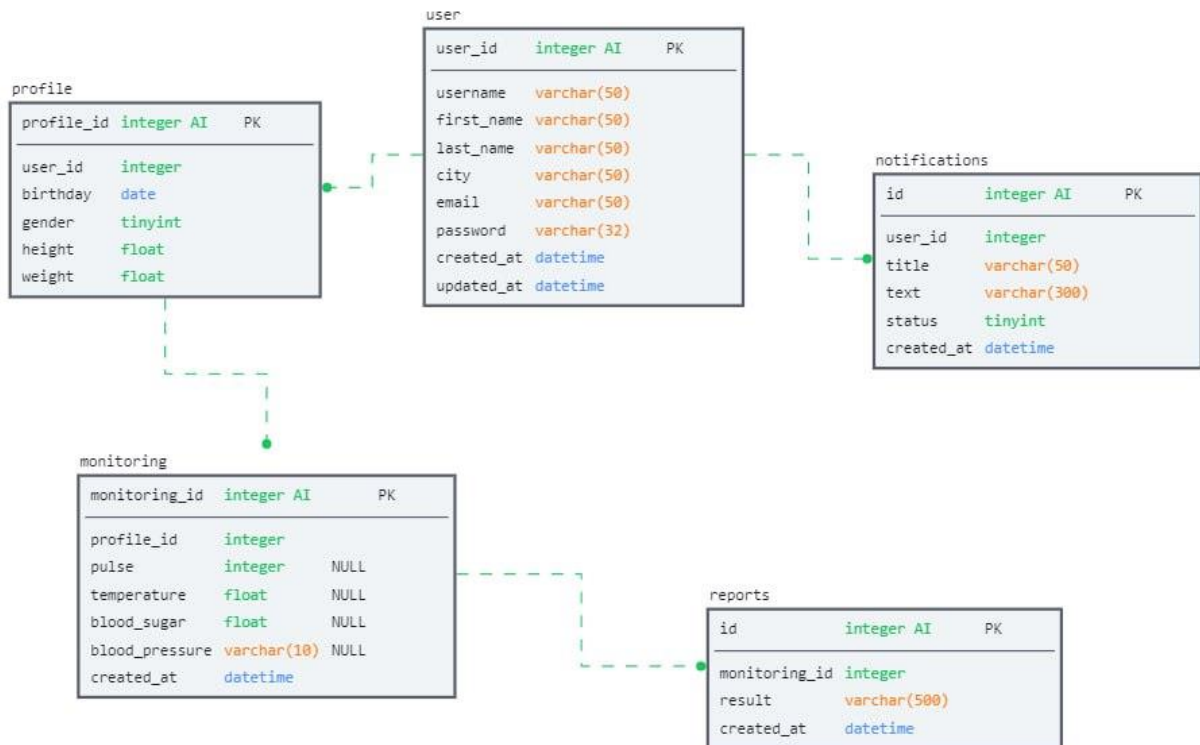


Рисунок 3.2 – Фізична модель даних

Фізична модель орієнтована на технічні аспекти і враховує характеристики конкретної системи баз даних.

### 3.4 Розробка карти застосунку аналізу стану здоров'я людини

Для уточнення вимог до застосунку аналізу стану здоров'я людини, розроблено макет застосунку, який дозволяє краще зрозуміти наповнення екранів застосунку.

Після проходження авторизації, користувач переходить на головний екран застосунку (рис 3.3).

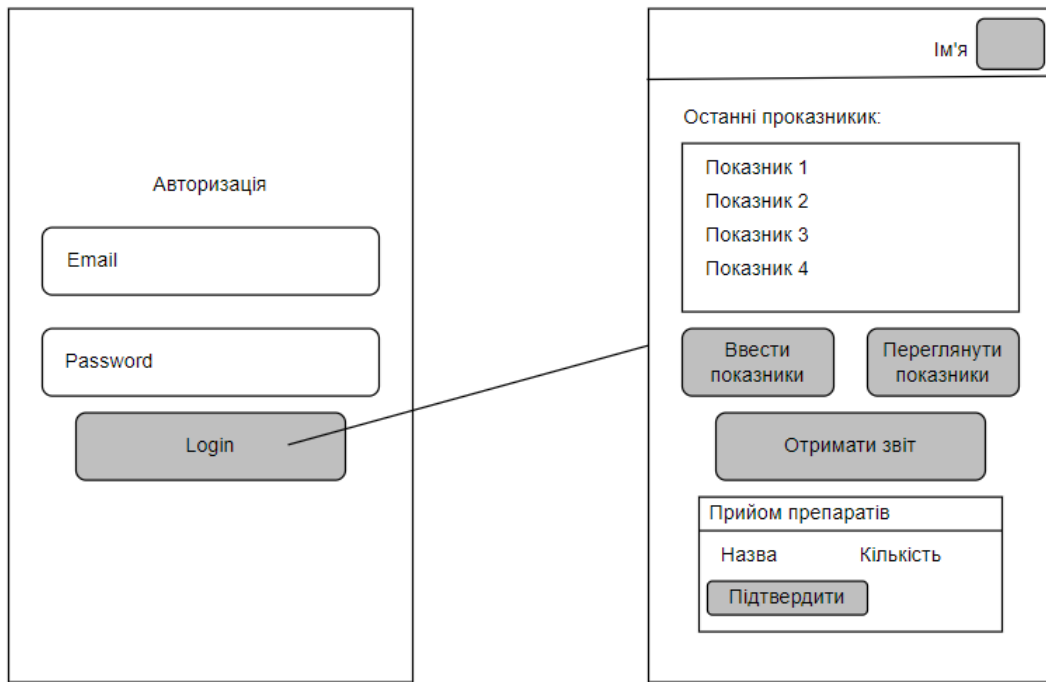


Рисунок 3.3 – Макет екрану авторизації і головного екрану

На екрані авторизації розміщено поля для вводу електронної пошти та паролю. Після вводу потрібних даних і проходження перевірки, користувач переходить на головний екран застосунку. На головному екрані користувач отримує доступ до потрібної йому інформації. Також на цьому екрані користувач може підтвердити приймання ліків.

Головний екран надає доступ до головних функцій застосунку, а саме вводу показників, перегляду показників, отримання звіту про здоров'я, налаштування акаунту (рис 3.4).

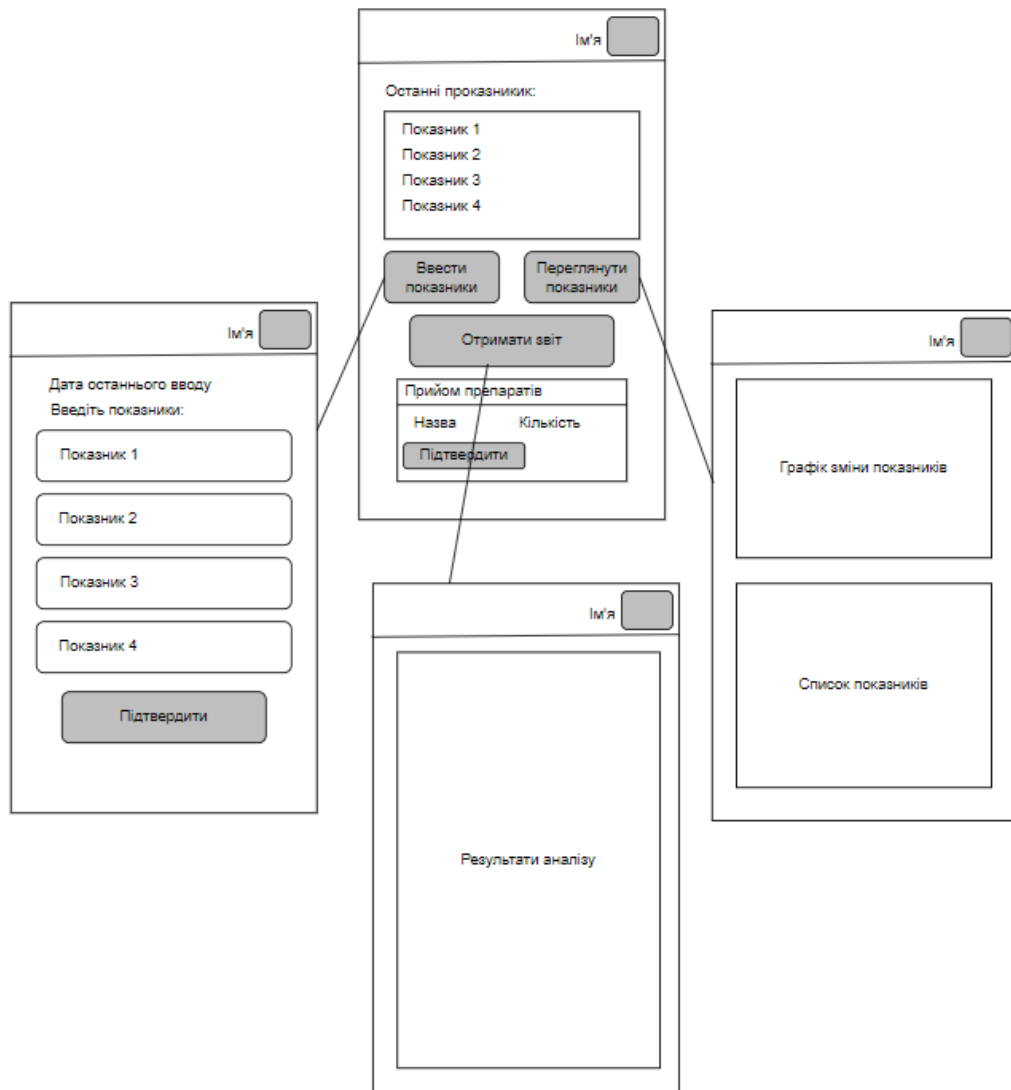


Рисунок 3.4 – Екрани, на які можна перейти з головного екрану

Екран вводу показників містить поля для вводу показників і кнопку підтвердження введених даних. Екран для перегляду показників містить графік з відображенням змін показників, які користувач відстежував за увесь час, а також список усіх показників, які користувач вводив.

При натисканні на кнопку «Отримати звіт», користувач переходить на екран звіту, де відображається інформація про стан здоров'я, яка була отримана в результаті аналізу.

Екран налаштування надає доступ до функцій, які дозволяють відредагувати усі потрібні дані (рис 3.5).

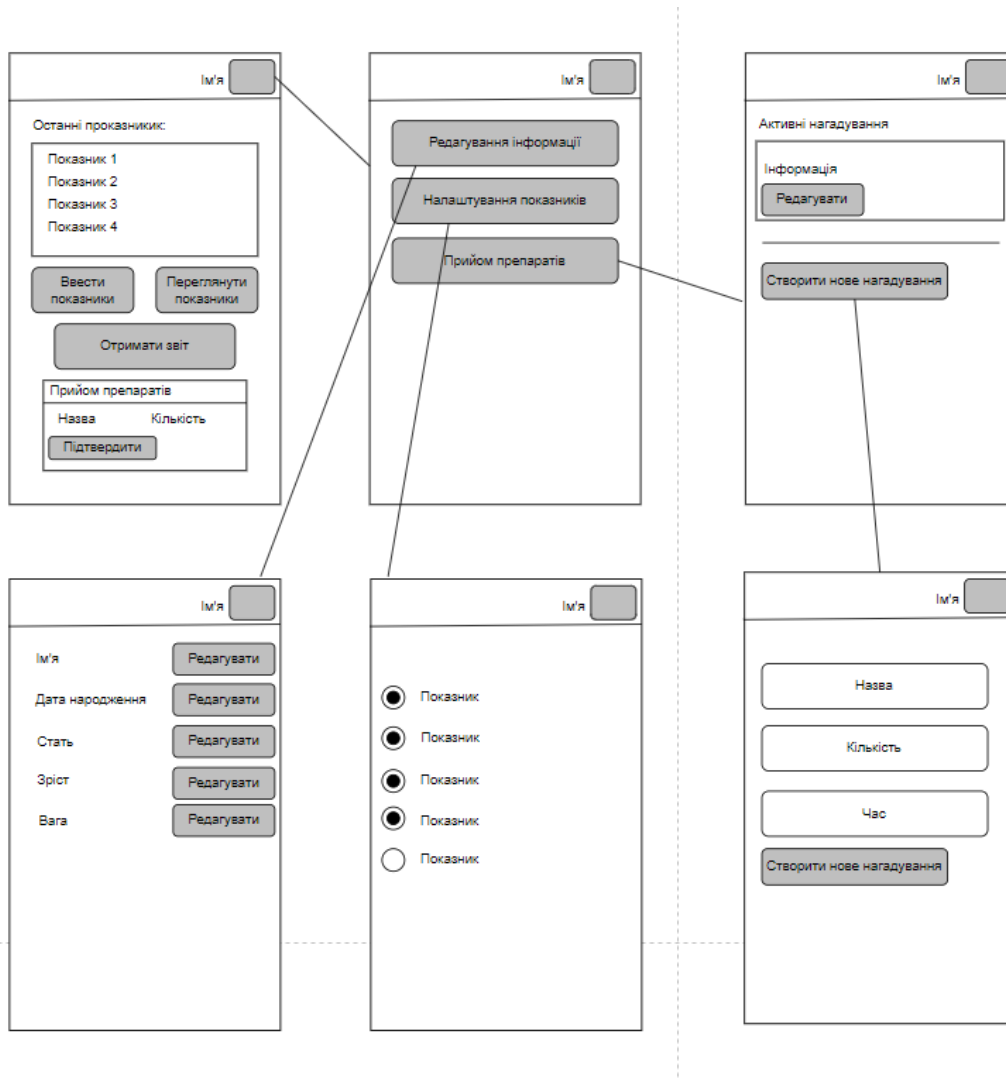


Рисунок 3.5 – Екрани налаштування інформації

Кнопка «Редагування інформації» направляє користувача на екран редагування особистої інформації. На цьому екрані відображається інформація про користувача, а також надається доступ до зміни цієї інформації.

Кнопка «Налаштування показників» направляє користувача на екран, де користувач може обрати показники, які він хоче відстежувати. На цьому екрані показники представлені у вигляді списку з чекбоксами.

Кнопка «Приймання препаратів» направляє користувача на екран управління нагадуваннями. На цьому екрані користувач може переглянути інформацію про вже створені нагадування і відредагувати їх. А також цей екран надає доступ до функції створення нового нагадування. При створенні нового

нагадування користувач переходить до екрану з формою, де потрібно вказати інформацію про препарат, кількість пігулок та час приймання.

### 3.5 Реалізація алгоритму аналізу стану здоров'я людини

Для пошуку аномалій у показниках використовується середнє та стандартне відхилення. Код алгоритму пошуку аномалій наведено у лістингу 3.1.

Лістинг 3.1 – Код пошуку аномальних змін показників

```

1  function detectAnomalies(healthData) {
2      const mean = healthData.reduce((acc, val) => acc + val, 0) /
      healthData.length;
3      const stdDev = Math.sqrt(healthData.map(val => Math.pow(val -
      mean, 2)).reduce((acc, val) => acc + val, 0) /
      healthData.length);
4      const anomalies = healthData.filter(val => Math.abs(val - mean)
      > stdDev);
5      return anomalies;
6  }
```

Цей алгоритм визначення аномалій у наборі даних, функціонує на основі середнього значення і стандартного відхилення. Далі розглянуто рядки окремо:

- рядок 2 - Обчислення середнього значення (середнього арифметичного) для вхідних даних «(healthData)». Використовується функція `reduce` для підсумовування всіх значень та поділу на кількість елементів.

- рядок 3 - Обчислення стандартного відхилення. Спочатку для кожного значення вхідних даних обчислюється квадрат різниці між значенням і середнім (`Math.pow(val - mean, 2)`). Потім використовується функція `reduce` для підсумовування цих квадратів, і весь результат ділиться на кількість елементів, а потім витягується квадратний корінь.

- рядок 4 - Фільтрація значень вхідних даних, для яких абсолютна різниця між значенням і середнім перевищує стандартне відхилення. Отримані значення вважаються аномаліями.

- рядок 5 - Повертає масив аномальних значень, знайдених у вхідних даних.

Для знаходження взаємозв'язка використовується коефіцієнт кореляції Пірсона. У лістингу 3.2 наведено код пошуку кореляції між показниками.

Лістинг 3.2 – Код пошуку кореляції між показниками

```

1  function findPatterns(data1, data2) {
2      const mean1 = data1.reduce((acc, val) => acc + val, 0) /
      data1.length;
3      const mean2 = data2.reduce((acc, val) => acc + val, 0) /
      data2.length;

4      let numerator = 0, denominator1 = 0, denominator2 = 0;
5      for (let i = 0; i < data1.length; i++) {
6          numerator += (data1[i] - mean1) * (data2[i] - mean2);
7          denominator1 += Math.pow(data1[i] - mean1, 2);
8          denominator2 += Math.pow(data2[i] - mean2, 2);
9      }
10     return numerator / Math.sqrt(denominator1 * denominator2);
11 }

```

Цей алгоритм призначений для пошуку патернів або кореляцій між двома наборами даних.

- рядок 2 - Обчислення середнього значення (середнього арифметичного) для першого набору даних (data1). Використовується функція reduce для підсумовування всіх значень та поділу на кількість елементів.

- рядок 3 - Обчислення середнього значення для другого набору даних (data2). Так само використовується функція reduce.

- рядок 4 - Ініціалізація змінних numerator, denominator1 та denominator2 для подальшого використання у виразі розрахунку кореляції.

- рядок 5 - Цикл, який ітерується через всі елементи наборів data1 та data2.

- рядок 6 - Розрахунок чисельника для кореляційного коефіцієнта. Додає до numerator добуток різниці між значеннями обох наборів і їх середніми значеннями.

- рядок 7 - Розрахунок частини знаменника для кореляційного коефіцієнта для першого набору даних.

- рядок 8 - Розрахунок частини знаменника для кореляційного коефіцієнта для другого набору даних.

- рядок 10 - Повертає значення кореляційного коефіцієнта між двома наборами даних. Знаменник розраховується як добуток квадратних коренів відповідних частин знаменників для кожного набору.

У лістингу 3.3 наведено функцію, яка призначена для ідентифікації зовнішніх факторів і хвороб, які мають високу кореляцію з набором даних про здоров'я.

### Лістинг 3.3 – Код для пошуку причин зміни стану здоров'я

```

1  function identifyExternalFactors(healthData, externalData) {
2      return externalData.filter(factor => findPatterns(healthData,
3          factor.data) > threshold);
4  }
5  function searchForDiseases(anomalies, diseasesDatabase) {
6      return diseasesDatabase.filter(disease =>
7          disease.symptoms.some(symptom=> anomalies.includes(symptom)));
8  }

```

- рядок 2 - Використання методу `filter` для відбору тих зовнішніх факторів, для яких значення кореляції (визначене за допомогою функції `findPatterns`) перевищує певний поріг (`threshold`).

- рядок 2 - Функціональний вираз, який визначає умову фільтрації. Для кожного об'єкта `factor` в `externalData` викликається функція `findPatterns`, де `healthData` передається як перший набір даних, а `factor.data` являє собою другий набір даних (зовнішній фактор). Результат порівнюється з пороговим значенням (`threshold`), і тільки ті фактори, для яких умова виконується, залишаються у кінцевому результаті.

## 3.6 Розробка інтерфейсу клієнтської частини застосунку аналізу стану здоров'я людини

При розробці інтерфейсу використовується фреймворк Jetpack Compose. Код у Jetpack Compose представляє із себе набір декларативних інструкцій, написаних мовою програмування Kotlin, які визначають, як повинен виглядати інтерфейс користувача Android-застосунку. Замість використання XML-макетів,

як у традиційному підході Android з використанням View і ViewGroup, Jetpack Compose використовує Kotlin DSL (Domain-Specific Language) для опису інтерфейсу користувача.

У якості прикладу реалізації інтерфейсу за допомогою фреймворку Jetpack Compose, у лістингу 3.4 наведено код реалізації екрану зі звітом про здоров'я.

Лістинг 3.4 – Приклад реалізації інтерфейсу за допомогою фреймворку Jetpack Compose

```
@Composable
fun ReportView(modifier: Modifier = Modifier) {
    Box() {
        Image(modifier = modifier.fillMaxSize(),
            painter = painterResource(id=R.drawable.background),
            contentDescription = "")
        Column(modifier.fillMaxSize()) {
            Row(modifier = modifier
                .fillMaxWidth()
                .padding(vertical = 15.dp),
                horizontalArrangement = Arrangement.End,
                verticalAlignment = Alignment.CenterVertically) {
                Text(modifier = modifier.padding(horizontal = 10.dp),
                    Image(modifier = modifier.padding(horizontal = 10.dp),
                        imageVector = Icons.Default.Settings,
                        contentDescription = "qwe")
            )
        }
        Box(
            modifier = Modifier
                .fillMaxWidth()
                .height(2.dp)
                .background(MaterialTheme.colorScheme.primary)
        )
        Text(
            modifier = modifier
                .fillMaxWidth()
                .padding(horizontal = 10.dp, vertical = 20.dp),
            text = "Звіт",
            textAlign = TextAlign.Center,
            fontSize = 20.sp
        )
        Column(modifier = modifier
            .fillMaxWidth()
            .padding(horizontal = 5.dp)
            .border(BorderStroke(2.dp,
                MaterialTheme.colorScheme.secondary))
        )
    }
}
```

```

        .background(Color(0x99bf8f9e)) {
            Text(modifier = modifier.padding(horizontal = 10.dp,
vertical = 10.dp),
                text = report.value,
                fontSize = 16.sp
            )
        }
    }
}

```

Діла розглянуто елементи коду, який наведено у лістингу:

- `@Composable` – це анотація, яка позначає функцію `ReportView` як компонент `Compose`. `Composable`-функції визначають UI-компоненти, і `Compose` автоматично відслідковує зміни та оновлює інтерфейс при їх зміні.

- `fun ReportView(modifier: Modifier = Modifier)` - Оголошення `Composable`-функції `ReportView` з параметром `modifier`, який використовується для налаштування зовнішнього вигляду компонента. В цьому випадку за замовчуванням використовується порожній `Modifier`.

- `Box() { ... }` - Контейнер-бокс, який може містити декілька дочірніх елементів. У цьому випадку, він містить інші компоненти, такі як `Image`, `Column` та `Text`.

- `Image(modifier = modifier.fillMaxSize(), painter = painterResource(id = R.drawable.background), contentDescription = "")` - Зображення, яке заповнює весь батьківський контейнер (`fillMaxSize()`) і використовує ресурс зображення з ресурсів проекту (зображення задається за допомогою `R.drawable.background`).

- `Column(modifier.fillMaxSize()) { ... }` - Вертикальний контейнер, який також заповнює весь батьківський контейнер. Вмістом є декілька інших компонентів, таких як `Row`, `Box`, і т. д.

- `Text(...)`- блок, що визначають заголовок "Звіт", розташування тексту, розмір і т. д.

- `Column(...)` з використанням різних модифікаторів - в цьому блоку організовано вертикальний контейнер, який містить текстовий блок ("Звіт") та ще один `Column` з текстовим блоком (`Text`) і фоновим кольором.

- Text(...) в останньому Column - виводить текстовий блок, який містить значення змінної report.value з розміром шрифту 16sp.

Код, представлений у лістингу, визначає компонент інтерфейсу для відображення звіту, який містить зображення, ім'я, роздільник, заголовок та текстовий блок зі звітом. Кожен елемент розміщений відповідно один до одного, утворюючи зовнішній вигляд звіту.

Далі наведено знімки екрану реалізованого інтерфейсу.

При вході користувач повинен пройти авторизацію (Рис 3.6).

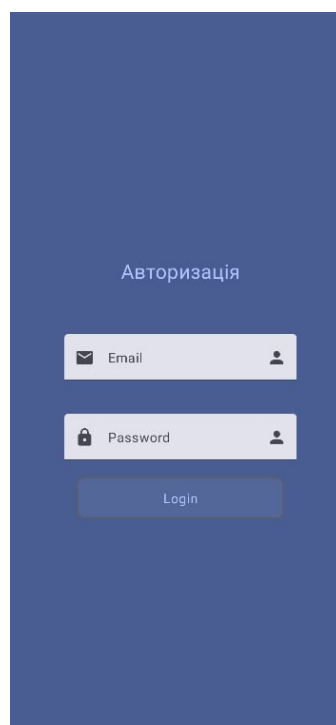


Рисунок 3.6 – Екран авторизації

На цьому екрані розміщено два поля для вводу електронної пошти а паролю. Після вводу пошти та паролю, користувач потрапляє на головний екран застосунку (рис 3.7).

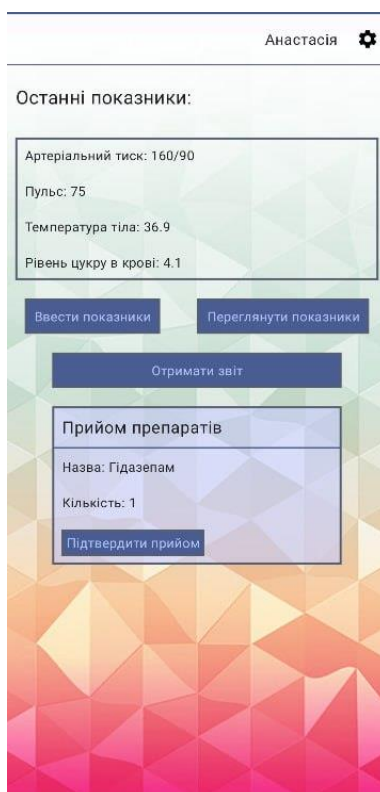


Рисунок 3.7 – Головний екран застосунку

На цьому екрані розміщено інформацію про останні введені показники. Також на цьому екрані розміщено інформацію про препарати, які потрібно прийняти користувачу. Користувач може підтвердити приймання ліків.

Також на цьому екрані розміщені кнопки, які надають доступ до головних функцій і екранів застосунку, це екрани з налаштуваннями, списком усіх введених показників, екран для вводу показників та екран для отримання звіту.

При натисканні на значок шестерні, користувач переходить на екран налаштування (рис 3.8).



Рисунок 3.8 – Екран налаштування

На цьому екрані розміщено кнопки, які надають доступ до функцій редагування інформації, налаштування показників та налаштування приймання препаратів.

Функція редагування інформації дозволяє змінити інформацію про користувача, а саме ім'я, вік, стать, місто проживання, зріст, вагу, а також інформацію для доступу до акаунту (пароль, електронна пошта).

Функція налаштування приймання препаратів дозволяє створити нове нагадування про приймання препаратів або відредагувати існуюче нагадування.

Функція налаштування показників дозволяє додавати або видаляти показники, які користувач хоче відстежувати (рис 3.9).

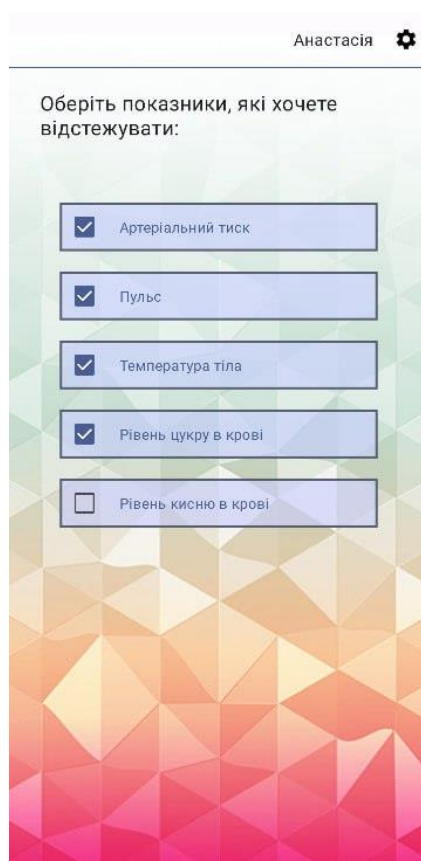


Рисунок 3.9 – Екран з налаштуванням показників

На цьому екрані користувач може обрати показники, які хоче відстежувати. Показники представлені у вигляді списку з чекбоксами. Як можна побачити, показники «Артеріальний тиск», «Пульс», «Температура тіла» і «Рівень цукру у крові» обрані, а показник «Рівень кисню у крові» не обрано. При введенні показників, у користувача будуть запрошуватися дані тільки про обрані показники.

При створенні нагадувань про приймання ліків, користувач отримуватиме пуш-повідомлення (рис 3.10).

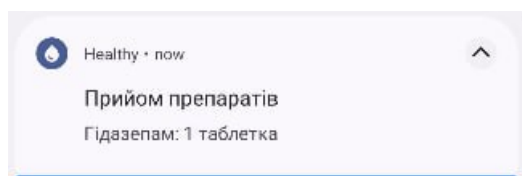


Рисунок 3.10 – Пуш-повідомлення про приймання ліків

Пуш-повідомлення містить інформацію про назву препарату, який потрібно прийняти, і кількість таблеток.

При натисканні на повідомлення, користувач переходить на головний екран застосунку, де він може підтвердити приймання ліків.

Якщо користувач хоче отримати звіт про стан здоров'я, йому потрібно натиснути кнопку «Отримати звіт» на головному екрані, після чого він перейде на екран звіту, де буде сформовано звіт (рис 3.11).

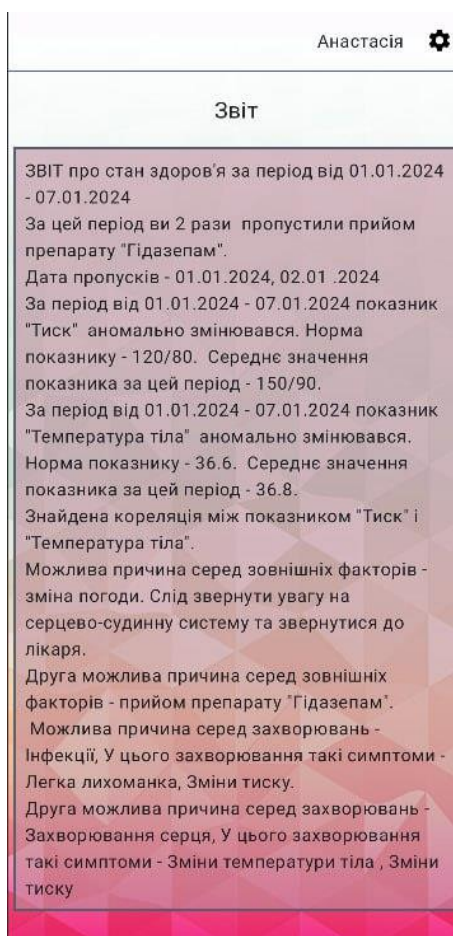


Рисунок 3.11 – Екран звіту

На екрані звіту можна переглянути результати аналізу. Звіт містить інформацію про період відстежування показників, на основі яких було сформовано звіт, інформацію про приймання ліків, дати пропуску приймання ліків. Також у звіті міститься уся інформація, яка була отримана у результаті аналізу здоров'я: Аномальні зміни показників, кореляція між ними, норма показників, середнє значення показників за заданий період, можливі причини

серед зовнішніх факторів, рекомендації щодо здоров'я, можливі причин серед захворювань.

### 3.7 Тестування розробленого мобільного застосунку аналізу стану здоров'я людини

Для тестування розробленого алгоритму проведено модульне тестування. Модульне тестування - це метод програмного тестування, який використовується для перевірки правильності роботи окремих модулів або компонентів програмного забезпечення. Модуль - це найменша незалежна частина програми, яка може бути протестована окремо. Метою модульного тестування є визначення того, чи працює кожен модуль коректно, а також виявлення та усунення можливих помилок або дефектів. Модульне тестування є важливою частиною процесу розробки програмного забезпечення, оскільки воно дозволяє вчасно виявляти та виправляти помилки, забезпечуючи високу якість продукту.

Принцип проведеного тестування полягає у перевірці алгоритму на коректність роботи на основі набору тестових даних.

Для перевірки алгоритму пошуку аномалій серед показників обрано масив даних з показниками, які представлено у лістингу 3.5.

#### Лістинг 3.5 – Масив тестових даних з показниками

```
const testData1 = [72, 75, 74, 73, 76, 150, 72, 74, 75, 72, 76, 73];  
const testData2 = [36.6, 36.6, 38.5, 38.1, 36.7, 36.6, 36.6];  
const testData3 = [4.0, 4.9, 4.9, 4.9, 5.3, 5.3, 4.9];
```

Перший масив – це тестові значення пульсу, другий – тестові значення температури тіла, третій – тестові значення рівня цукру у крові. Очікується, що вихідними даними будуть аномальні значення показників, а саме: у першому наборі даних – 150, у другому – 38.5, 38.1, у третьому – 4.0, 5.3, 5.3.

На рисунку 3.12 представлено результати тестування на основі тестового набору даних.

```

Output:

Test Data 1: [
  72, 75, 74, 73, 76,
  150, 72, 74, 75, 72,
  76, 73
]
Anomalies Detected 1: [ 150 ]
-----

Test Data 2: [
  36.6, 36.6,
  38.5, 38.1,
  36.7, 36.6,
  36.6
]
Anomalies Detected 2: [ 38.5, 38.1 ]
-----

Test Data 3: [
  4, 4.9, 4.9,
  4.9, 5.3, 5.3,
  4.9
]
Anomalies Detected 3: [ 4, 5.3, 5.3 ]

```

Рисунок 3.12 – Результати тестування алгоритму пошуку аномалій серед показників

Як можна побачити, у першому наборі показників за допомогою алгоритму знайдено 1 аномалію, а саме значення 150. У другому наборі показників знайдено 2 аномалії – значення 38.5 і 38.1. У третьому наборі показників знайдено 3 аномалії – 4, 5.3, 5.3. Тобто, можна сказати, що за результатами тестування алгоритм пошуку аномалій серед показників працює як і очікується.

Для перевірки алгоритму пошуку аномалії між різними показниками також обрано масиви даних, які представлено у лістингу 3.6.

#### Лістинг 3.6 – Масиви тестових даних з показниками

```

const testData11 = [36.6 , 36.3, 36.6 , 37.8 , 37.8, 36.6];
const testData12 = [4.9, 4.9, 4.9, 5.3, 5.3, 4.9];
const testData21 = [36.6 , 36.3, 36.6 , 37.8 , 37.8, 36.6] ;
const testData22 = [4.9, 4.9, 4.9, 4.9, 4.9, 4.9] ;
const testData31 = [36.6 , 36.3, 36.6 , 37.8 , 37.8, 36.6] ;
const testData32 = [5.0, 5.1, 4.9, 4.9, 4.9, 4.9] ;

```

При тестуванні алгоритму пошуку кореляції, вхідні дані розбиті на пари. До кожної пари входить масив з показниками температури тіла та масив з

показниками рівню цукру у крові. Очікується, що коефіцієнт кореляції першої пари (testData11, testData12) буде дорівнювати приблизно 1, коефіцієнт кореляції другої пари (testData21, testData22) буде дорівнювати приблизно 0, коефіцієнт кореляції третьої пари (testData31, testData32) буде приблизно -0.6.

Результати тестування наведено на рисунку 3.13.

Output:

```
Data 1: [ 36.6, 36.3, 36.6, 37.8, 37.8, 36.6 ]
Data 2: [ 4.9, 4.9, 4.9, 5.3, 5.3, 4.9 ]
Pattern Similarity (Data 1 and Data 2): 0.9847835588179363
-----
Data 1: [ 36.6, 36.3, 36.6, 37.8, 37.8, 36.6 ]
Data 3: [ 4.9, 4.9, 4.9, 4.9, 4.9, 4.9 ]
Pattern Similarity (Data 1 and Data 3): -5.820992900276011e-15
-----
Data 2: [ 36.6, 36.3, 36.6, 37.8, 37.8, 36.6 ]
Data 3: [ 5, 5.1, 4.9, 4.9, 4.9, 4.9 ]
Pattern Similarity (Data 2 and Data 3): -0.5899445023218312
```

Рисунок 3.13 – Результати тестування алгоритму пошуку кореляції між показниками

Як можна побачити, у першому випадку коефіцієнт кореляції приблизно 0.9, у другому приблизно 0, а у третьому приблизно -0.6. Тобто, можна сказати, що за результатами тестування алгоритм пошуку кореляції між показниками працює як і очікується.

## ВИСНОВКИ

В ході роботи було проведено аналіз предметної області, який дозволив охарактеризувати діяльність застосунку аналізу стану здоров'я людини. Проведений аналіз існуючих застосунків для відстежування здоров'я дозволив виявити недоліки існуючих систем. Також, результати аналізу існуючих алгоритмів для аналізу здоров'я дозволили обрати методи, які дозволили реалізувати алгоритм аналізу стану здоров'я людини.

Під час визначення функціональних вимог, описано бізнес-процеси за допомогою концептуального моделювання.

Розроблена діаграма варіантів використання дозволила відобразити акторів і їх взаємовідносини з прецедентами.

Мета роботи полягає у розробці застосунку аналізу стану здоров'я людини. При розробці мобільного застосунку аналізу стану здоров'я людини використовувалась мова програмування Kotlin, фреймворк Jetpack Compose, платформа Node.js, бібліотека Retrofit та СУБД MySQL.

Розроблений мобільний застосунок дозволяє користувачам відстежувати показники здоров'я і отримувати звіти про здоров'я з рекомендаціями щодо здоров'я. Розроблений мобільний застосунок поєднує в собі функції існуючих аналогів, а також надає доступ до функції аналізу здоров'я за допомогою алгоритмів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про інформацію. – URL: <https://zakon.rada.gov.ua/laws/show/2657-12#Text> (Дата звернення: 20.11.2023)
2. Mintz Y, Brodie R. Introduction to artificial intelligence in medicine. *Minim Invasive Ther Allied Technol.* 2019;28(2):73-81. doi: 10.1080/13645706.2019.1575882
3. Stockman G. C., Shapiro L. G. *Computer Vision*. Prentice Hall, February 2001. 609 p.
4. Tiwari S., Shamik, Anurag J.. "Convolutional capsule network for COVID-19 detection using radiography images." *International Journal of Imaging Systems and Technology* 31.2 (2021): 525-539
5. Topol, E. J. *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, 2019. - 400 с.
6. Д.Ю. Киселев, Ю.В. Киселев, А.В. Вавилин. Функциональное моделирование на базе стандарта IDEF0: метод. Указания . – Самара: Издво СГАУ, 2014. – 20 с
7. Березька К.М. Прогнозування прибутків комп'ютерної фірми на основі ARIMA-моделей / К.М. Березька, В.М. Неміш, Ю.С. Тимчук // Матеріали III всеукраїнської науково-практичної конференції «Прикладна геометрія та інформаційні технології в моделюванні об'єктів, явищ і процесів» (AGIT-2018), м. Миколаїв, 17–19 жовтня 2018 р. – Миколаїв: МНУ імені В.О. Сухомлинського, 2018. – С. 120-121.
8. Bates, D. W., Sheikh, A., & Asch, D. A. *Integrating Health Information Technology: Challenges, Critical Issues, and Practical Solutions*. John Wiley & Sons, 2014. - 320 с.
9. *Patterns in Network Architecture: A Return to Fundamentals* / John Day, 2007. – 464 с.

10. Гмурман В. Е. Теория вероятностей и математическая статистика: Учебное пособие для вузов. 10-е издание, стереотипное. Москва: Высшая школа, 2004. 479 с.

11. Динаміка стану здоров'я народу України та регіональні особливості: аналітично-статистичний посібник /В.М. Коваленко та ін. Київ, 2012. 211 с.

12. Імунобіологічні препарати: Довідник / Смірнов В.В., Сельнікова О.П., Думанський В.Д. Київ: Моріон, 2001. 200 с.

13. Кравчун Н.О. Роль окремих компонентів метаболічного синдрому у розвитку серцево-судинних ускладнень у хворих на цукровий діабет 2 типу. Ендокринологія. 2005. Т. 10. № 2. С. 156–165.

14. Сидорчук Л.П., Амосова К.М. Обґрунтування призначення антигіпертензивного лікування хворим на есенціальну артеріальну гіпертензію залежно від індивідуальної фармакогенетичної чутливості. Український кардіологічний журнал. 2009. № 5. С. 35-51.

15. Cook R, Cook D, Tilley J, Lee K, Marshall J. Multiply organ dysfunction. Baseline and components scores. Crit Care Med. 2001 Nov;29(11):2046-50.

16. Diseases Database. – URL: <http://www.diseasesdatabase.com/> (Дата звернення: 20.11.2023)

17. Буч Г., Джекобсон А., Рамбо Д. UML. Руководство пользователя. М.: ДМК Пресс, 2004.

18. Абрагамович ОО, Абрагамович УО, Синенький ОВ, Кушина АП, Циганик ЛВ. Артеріальна гіпертензія та системний червоний вовчак: ще один погляд на проблему (огляд сучасної літератури та опис клінічних випадків). Український ревматологічний журнал. 2012;4:23–42.

19. Jemerov, D., & Isakova, S. Kotlin in Action. Manning Publications, 2017.- 696 с.

20. Cantelon, M., Harter, M., & Raj, T. Node.js in Action. Manning Publications, 2019. - 416 с.

21. Пік Роджерс, Джон Ломбардо Android. Розробка додатків – М.: ЕКОМ Паблішерс, 2010 – 400 с.
22. Square. Retrofit [Електронний ресурс] / square - URL: <https://square.github.io/retrofit/> (Дата звернення: 20.11.2023)
23. Сапсай Т.Г., Сущук-Слюсаренко В.І. Основи теорії ймовірностей - К.: НТУУ «КПІ», 2014.- 204с.
24. Forta, V. MySQL Crash Course. Addison-Wesley, 2012. - 336 с.
25. . Джек Фолк, Сем Канер, Енг Кек Нгуєн (2001). Тестування програмного забезпечення.