

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО
ТРЕКЕРУ ФІЗИЧНОЇ АКТИВНОСТІ
SOFTWARE IMPLEMENTATION OF A
MOBILE PHYSICAL ACTIVITY TRACKER

Виконав: студент 4 курсу, групи КНТ-121сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ДРОБІНЯК М.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. СУББОТІН
« » 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ДРОБІНЯКА Миколи Васильовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація мобільного трекеру фізичної активності. Software Implementation of a Mobile Physical Activity Tracker

керівник проєкту (роботи) д.т.н., проф, Олійник Андрій Олександрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року
№ 168

2. Строк подання студентом проєкту (роботи) 03 травня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання « 04 » квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Микола ДРОБІНЯК
(Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
89 с., 28 рис., 3 табл., 4 дод., 24 джерел.

ТРЕКІНГ, ФІЗИЧНА АКТИВНІСТЬ, ЗДОРОВ'Я, МОЛОДЬ,
МОБІЛЬНИЙ ЗАСТОСУНОК, ANDROID, МОВА ПРОГРАМУВАННЯ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Об'єкт дослідження – процес програмної реалізації мобільного трекеру фізичної активності.

Предмет дослідження – програмні засоби для реалізації мобільного трекеру фізичної активності.

Метою роботи є програмна реалізація мобільного трекеру фізичної активності.

Матеріали, методи та технічні засоби: мова програмування Java та середовище розробки – фреймворк Android Studio, система керування базами даних MySQL, мова моделювання та проєктування UML.

Результати. Розроблено мобільний застосунок – трекер фізичної активності.

Практична цінність роботи полягає у програмній реалізації мобільного трекеру фізичної активності у вигляді Android застосунку.

Висновки. Реалізовано мобільного трекеру фізичної активності у вигляді Android застосунку.

Галузь використання – активна молодь.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 28 figures, 3 tables, 4 appendixes, 24 sources.

TRACKING, PHYSICAL ACTIVITY, HEALTH, YOUTH, MOBILE APP, ANDROID, PROGRAMMING LANGUAGE, SOFTWARE

The object of research is the process of software implementation of a mobile physical activity tracker.

The subject of the research is software tools for implementing a mobile physical activity tracker.

The purpose of this work is to provide software implementation of a mobile physical activity tracker.

Materials, methods, and technical tools: Java programming language and development environment – Android Studio framework, MySQL database management system, UML modeling and design language.

Results. A mobile Physical Activity Tracker app has been developed.

The practical value of the work lies in the software implementation of a mobile physical activity tracker in the form of an Android application.

Conclusions. Implemented a mobile physical activity tracker as an Android app.

The field of use is active young people.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	7
Вступ	8
1 Аналіз предметної області	12
1.1 Дослідження ТФА	12
1.2 Механізми програмного ТФА	15
1.3 Порівняння існуючих аналогів	18
1.4 Постановка завдання	24
1.5 Висновки за розділом 1	27
2 Матеріали і методи	28
2.1 Вибір мови програмування	28
2.2 Вибір та обґрунтування середовища розробки	35
2.3 Структурна схема розробки	40
2.4 Висновки за розділом 2	44
3 Основні рішення щодо реалізації компонентів системи	45
3.1 Проєктування моделей системи	45
3.2 Проєктування інтерфейсу	50
4 Експлуатація, тестування та експериментальне дослідження програми	53
4.1 Призначення й умови застосування програми	53
4.2 Вимоги до інтерфейсу	53
4.3 Умови виконання програми	54
4.4 Методика та результати тестування	54
Висновки	58
Перелік джерел посилання	59
Додаток А Технічне завдання	62
Додаток Б Опис програми	67
Додаток В Текст програми	74
Додаток Д Слайди презентації	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ОС – операційна система;

ПЗ – програмне забезпечення;

СУБД – система управління базами даних;

ТФА – трекер фізичної активності;

IDE– integrated development environment\$

GUI – graphical user interface;

HTTPS – hypertext transfer protocol secure;

JVM – Java virtual machine;

UI – user interface;

UML – unified modeling language.

ВСТУП

Розробка мобільного застосунку для відстеження фізичної активності, тобто трекеру фізичної активності (ТФА) є не тільки своєчасним, але й нагальним завданням, враховуючи зростаючу поширеність малорухливого способу життя та пов'язаних з ним ризиків для здоров'я у всьому світі [1]–[4].

Глобальна епідемія малорухливого способу життя: малорухливий спосіб життя став глобальною епідемією, що посилюється технологічним прогресом, урбанізацією та змінами у сфері праці та дозвілля. Всесвітня організація охорони здоров'я (ВООЗ) [3] визначає відсутність фізичної активності як четвертий за значимістю фактор ризику смертності у всьому світі, який, за оцінками, щорічно призводить до 3,2 мільйона смертей [1]–[4].

Зростаючі проблеми зі здоров'ям: малорухливий спосіб життя тісно пов'язаний з низкою проблем зі здоров'ям, включаючи ожиріння, серцево-судинні захворювання, цукровий діабет 2 типу, деякі види раку та психічні розлади. Оскільки ці умови продовжують загострюватися, виникає нагальна потреба у заходах, спрямованих на підвищення фізичної активності та зменшення пов'язаних з цим ризиків для здоров'я [1]–[4].

Технологічні досягнення: з розповсюдженням смартфонів та носіїв, оснащених датчиками, з'явилася безпрецедентна можливість використовувати технології для моніторингу та стимулювання фізичної активності в режимі реального часу. Мобільні застосунки ТФА можуть використовувати ці технології для надання персоналізованих та доступних рішень для відстеження та підвищення рівня фізичної активності [1]–[4].

Пандемія COVID-19: Пандемія COVID-19 ще раз підкреслила важливість підтримки фізичного здоров'я та благополуччя. Карантин і заходи соціального дистанціювання обмежили доступ до традиційних фітнес-залів, в результаті чого фізичні навантаження вдома і на свіжому повітрі стали мати вирішальне значення для підтримки здоров'я і стійкості до інфекційних захворювань [1]–[4].

Переваги застосунків ТФА:

– моніторинг поведінки та зворотній зв'язок: програми ТФА дозволяють відстежувати рівень фізичної активності в режимі реального часу, надаючи користувачам інформацію про їх щоденні рухи, такі як кількість кроків, пройдена відстань та хвилини активності. Ці дані дозволяють користувачам відстежувати свій прогрес з плином часу і отримувати зворотний зв'язок про рівень їх активності, мотивуючи їх на досягнення своїх цілей у фітнесі [1]–[4];

– постановка цілей та підзвітність: програми ТФА дозволяють користувачам встановлювати індивідуальні цілі на основі їхніх прагнень та уподобань щодо фітнесу. Незалежно від того, чи прагнуть вони схуднути, поліпшити стан серцево-судинної системи або підвищити загальний рівень активності, користувачі можуть встановлювати досяжні цілі і відстежувати прогрес в їх досягненні. Функція формування звітів в цих застосунках заохочує дотримання режиму фізичної активності [1]–[4];

– персоналізація та адаптивність: мобільні застосунки ТФА можуть адаптувати рекомендації та відгуки на основі індивідуальних характеристик, таких як вік, стать, рівень фізичної підготовки та стан здоров'я. За допомогою алгоритмів машинного навчання та введення користувачем ці програми з часом можуть адаптувати свої рекомендації відповідно до змін уподобань та обставин, підвищуючи залучення користувачів та їх прихильність [1]–[4];

– інтеграція з пристроями, що носяться: багато застосунків ТФА легко інтегруються з фітнес-трекерами, розумними годинниками та іншими пристроями, що носяться, дозволяючи здійснювати безперервний автоматичний моніторинг фізичної активності. Така інтеграція підвищує точність даних відстеження активності і надає користувачам повний огляд їх щоденних рухів [1]–[4];

– соціальна підтримка та залучення спільноти: деякі програми ТФА мають функції соціальних медіа, що дозволяють користувачам спілкуватися з друзями, родиною чи однодумцями, які поділяють подібні цілі фітнесу.

Соціальна підтримка та залучення громади сприяють підвищенню мотивації, підзвітності та дружньої конкуренції, заохочуючи користувачів залишатися активними та досягати своїх цілей спільно [1]–[4];

- освітні ресурси та методи зміни поведінки: програми ТФА часто включають освітні ресурси, такі як статті, відео та навчальні посібники, щоб надати користувачам знання про переваги фізичної активності та стратегії її включення у повсякденне життя. Крім того, ці програми використовують методи зміни поведінки, такі як постановка цілей, самоконтроль та винагорода, щоб сприяти стійкій зміні поведінки [1]–[4];

- відстеження здоров'я: окрім відстеження фізичної активності, багато програм ТФА включають функції для моніторингу інших параметрів здоров'я, таких як якість сну, харчування, рівень стресу та настроїв. Надаючи користувачам всебічну інформацію про стан здоров'я, ці застосунки сприяють цілісному самопочуттю і допомагають людям робити усвідомлений вибір способу життя [1]–[4];

- віддалена доступність і гнучкість: мобільні застосунки ТФА забезпечують зручний доступ в будь-який час і в будь-якому місці, дозволяючи користувачам дистанційно відстежувати свою фізичну активність і прогрес. Будь то вдома, на роботі чи в дорозі, користувачі можуть взаємодіяти з застосунком, щоб залишатися мотивованими та відповідальними, сприяючи постійності та відданості здоровим звичкам [1]–[4];

- інтеграція в охорону здоров'я та управління захворюваннями: програми ТФА є перспективними для інтеграції в системи охорони здоров'я, полегшуючи дистанційне спостереження за пацієнтами, ведення хронічних захворювань та профілактичні заходи. Надаючи людям можливість брати активну участь у управлінні своїм здоров'ям, ці програми потенційно можуть зменшити витрати на охорону здоров'я, покращити показники здоров'я та підвищити якість життя [1]–[4].

На закінчення хотілося б відзначити, що розробка мобільного застосунку для відстеження фізичної активності є актуальним завданням, враховуючи глобальну поширеність малорухливого способу життя і пов'язані з цим ризики для здоров'я [1]–[4]. Ці програми мають численні переваги у підвищенні фізичної активності та загального самопочуття, включаючи моніторинг поведінки та зворотний зв'язок, встановлення цілей та підзвітність, персоналізацію та адаптивність, інтеграцію з мобільними пристроями, соціальну підтримку та залучення громади, освітні ресурси та методи зміни поведінки, відстеження здоров'я та аналітичну інформацію, віддалену доступність та гнучкість та охорону здоров'я інтеграція та лікування захворювань. Використовуючи силу сучасних технологій, програми ТФА можуть допомогти людям вести здоровий та активний спосіб життя та зменшити тягар хронічних захворювань для суспільства [1]–[4].

Отже актуальною задачею залишається програмна реалізація мобільного трекеру фізичної активності.

Для виконання такої задачі необхідно виконати наступні завдання:

- дослідити область трекінгу фізичної активності;
- дослідити питання трекінгу фізичної активності через програмні застосунки;
- дослідити переваги та недоліки існуючих програмних рішень для трекінгу фізичної активності;
- визначити засоби програмної реалізації нового рішення ТФА;
- виконати проектування розробки нового рішення – програмної реалізації ТФА;
- виконати програмну реалізацію нового рішення;
- виконати тестування розробки;
- сформулювати висновки на основі виконаної роботи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Перший розділ пояснювальної записки до дипломної роботи буде присвячено аналізу предметної області ТФА, в тому числі за допомогою програмних засобів. Таким чином буде досліджено область ТФА. Головною ж метою першого розділу буде проаналізувати можливості розробки нового програмного застосунку (ПЗ) для ТФА на основі виявлених недоліків існуючих аналогів. Можливості розробки нового конкурентоспроможного ПЗ на основі виконаного аналізу буде зафіксовано в форматі функціональних і не функціональних вимог.

1.1 Дослідження ТФА

ТФА включає моніторинг та запис фізичних рухів та вправ під час (або ж після) тренувань. Він включає в себе різні показники, такі як кількість кроків, пройденої відстані, активні хвилини, спалені калорії і інтенсивність вправ. Це відстеження можна здійснювати вручну за допомогою журналів або в електронному вигляді за допомогою спеціальних пристроїв, застосунків для смартфонів або спеціальних фітнес-трекерів [5]–[7].

Важливість відстеження фізичної активності пояснити досить просто. Поінформованість та підзвітність: відстеження фізичної активності дозволяє людям отримувати інформацію про свої щоденні рухи та звички до фізичних вправ. Реєструючи активність, люди отримують уявлення про те, наскільки вони малорухливі, і про можливості підвищення фізичної активності. Це усвідомлення сприяє підзвітності, заохочуючи людей робити усвідомлений вибір руху та фізичних вправ у своєму повсякденному житті [5]–[7].

Постановка цілей та моніторинг прогресу: відстеження фізичної активності полегшує встановлення цілей та моніторинг прогресу. Встановлюючи конкретні, вимірювані, досяжні, актуальні та обмежені за часом цілі, люди можуть встановлювати цілі для підвищення рівня активності,

покращення фізичної форми або досягнення цілей щодо схуднення. Відстеження прогресу в досягненні цих цілей дозволяє користувачам оцінювати свою роботу, відзначати успіхи і вносити необхідні корективи, щоб не збитися з наміченого шляху [5]–[7].

Мотивація та залучення: відстеження фізичної активності може послужити потужним стимулом для зміни поведінки. Спостереження за прогресом у режимі реального часу, наприклад, за збільшенням кількості кроків або досягненням контрольних показників у вправах, зміцнює позитивні звички та підвищує мотивацію продовжувати займатися фізичною активністю. Багато пристроїв та застосунків для відстеження також включають елементи гейміфікації, такі як завдання, значки та нагороди, для підвищення мотивації та залучення [5]–[7].

Аналіз поведінки і закономірностей: ТФА дозволяє отримати цінну інформацію про індивідуальну поведінку і рухах. Аналізуючи дані про активність з часом, люди можуть виявляти тенденції, коливання та кореляції між рівнями активності та такими факторами, як час доби, місце розташування, настрій та умови навколишнього середовища [5]–[7]. Ці знання можуть бути використані при прийнятті рішень, допомагаючи людям оптимізувати свій розпорядок дня для досягнення максимальної ефективності та задоволення [5]–[7].

Підзвітність та соціальна підтримка: обмін даними про фізичну активність з друзями, родиною чи інтернет-спільнотами може підвищити підзвітність та соціальну підтримку. Беручи участь у змаганнях, ділячись досягненнями та отримуючи підтримку від однолітків, люди відчують товариську та мотивацію залишатися активними. Мережі соціальної підтримки можуть надати неоціненну підтримку у важкі часи та допомогти людям залишатися послідовними у виконанні фізичних вправ [5]–[7].

Моніторинг здоров'я і зниження ризику: регулярне відстеження фізичної активності дозволяє людям стежити за своїм загальним станом здоров'я і самопочуттям. Відстеження таких показників, як частота серцевих

скорочень, якість сну і рівень стресу, на додаток до фізичної активності, дає повну картину стану здоров'я людини. Виявлення відхилень від норми може спонукати людей звернутися за медичною допомогою та вжити активних заходів для зменшення ризиків для здоров'я, пов'язаних із малорухливим способом життя та відсутністю фізичної активності [5]–[7].

Налаштування та персоналізація: відстеження фізичної активності дозволяє налаштовувати і персоналізувати цілі і режим занять фітнесом. Люди можуть адаптувати свої цілі щодо фізичної активності відповідно до свого віку, рівня фізичної підготовки, стану здоров'я та конкретних цілей. Відстежуючи прогрес у досягненні індивідуальних цілей, люди можуть регулювати інтенсивність, тривалість і частоту своїх вправ, щоб оптимізувати результати і звести до мінімуму ризик травм або емоційного вигорання [5]–[7].

Зміна поведінки та формування звичок: постійне відстеження фізичної активності сприяє зміні поведінки та формуванню звичок. Встановивши режим регулярного моніторингу та реєстрації дій, люди розвивають самосвідомість, саморегуляцію та самоєфективність у прийнятті та підтримці здорових звичок. Згодом трекінг стає позитивною звичкою, що призводить до тривалого ведення активного способу життя.

Інтеграція з програмами охорони здоров'я та оздоровлення: відстеження фізичної активності може бути інтегровано в програми охорони здоров'я та оздоровлення для підтримки профілактики та лікування хронічних захворювань. Медичні працівники можуть використовувати дані про активність для оцінки фізичної підготовленості пацієнтів, встановлення реалістичних цілей та моніторингу прогресу під час реабілітації або лікування. Оздоровчі програми можуть використовувати дані відстеження для розробки цілеспрямованих заходів, надання персоналізованого коучингу та просування ініціатив у галузі медичної освіти [5]–[7].

Дослідження та аналіз даних: сукупні дані відстеження фізичної активності великих груп населення можуть сприяти дослідженню та аналізу даних у галузі охорони здоров'я, епідеміології та фізичних вправ. Аналізуючи

тенденції та закономірності рівня фізичної активності в демографічних, географічних регіонах та соціально-економічних групах, дослідники можуть виявити фактори, що впливають на поведінку, пов'язану з фізичною активністю, та розробити доказові заходи, спрямовані на заохочення активного способу життя та зменшення відмінностей у здоров'ї [5]–[7].

Таким чином, відстеження фізичної активності є важливим для підвищення обізнаності, підзвітності, постановки цілей, мотивації, залучення, розуміння поведінки, моніторингу здоров'я, адаптації, формування звичок, інтеграції з програмами охорони здоров'я та оздоровлення, досліджень та аналізу даних. Використовуючи потужність технологій та аналітичну інформацію на основі даних, люди можуть оптимізувати свій режим фізичної активності, покращити стан свого здоров'я та підвищити загальну якість свого життя [5]–[7].

1.2 Механізми програмного ТФА

Механізми програмного ТФА включають комбінацію апаратного, програмного забезпечення, алгоритмів та інтерфейсів користувача, призначених для моніторингу, запису, аналізу та представлення даних, пов'язаних з фізичною активністю. Ці механізми дозволяють користувачам відстежувати різні показники, такі як кількість кроків, пройдена відстань, кількість активних хвилин, спалені калорії, інтенсивність вправ та інші важливі параметри. Нижче наведені ключові компоненти і механізми, задіяні в програмі або застосунку ТФА [8], [9]:

Програмні компоненти. Мобільні застосунки: програми ТФА зазвичай розповсюджуються як мобільні застосунки, які користувачі можуть завантажувати та встановлювати на своїх смартфонах або планшетах. Ці програми забезпечують зручні інтерфейси для відстеження, запису та аналізу даних про фізичну активність. Вони також можуть запропонувати додаткові

функції, такі як постановка цілей, соціальні медіа, гейміфікація та інтеграція з іншими програмами для здоров'я та фітнесу [8], [9].

Вебпортали та інформаційні панелі: деякі програми ТФА пропонують вебпортали або інформаційні панелі, де користувачі можуть переглядати та керувати своїми даними про діяльність із робочого столу чи ноутбука. Ці портали надають додаткові функціональні можливості для поглибленого аналізу, створення звітів та обміну показниками активності [8], [9].

Алгоритми та обробка даних. Алгоритми виявлення активності: програми ТФА використовують алгоритми для виявлення та класифікації різних видів фізичної активності на основі даних датчиків. Ці алгоритми аналізують закономірності в показаннях акселерометра, даних GPS та інших вхідних даних датчиків, щоб визначити такі види діяльності, як ходьба, біг, їзда на велосипеді та підйом по сходах [8], [9].

Алгоритми оцінки калорій: багато програм ТФА використовують алгоритми для оцінки кількості спалених калорій під час різних видів діяльності на основі таких факторів, як тип активності, тривалість, інтенсивність, профіль користувача (наприклад, вік, вага, зріст) та дані про частоту серцевих скорочень (якщо такі є) [8], [9].

Об'єднання та інтеграція даних: програми ТФА об'єднують дані з декількох датчиків та джерел, щоб отримати всебічне уявлення про фізичну активність. Методи об'єднання даних об'єднують інформацію від акселерометрів, GPS, пульсометрів та інших датчиків для підвищення точності та надійності [10].

Користувальницькі інтерфейси та взаємодії. Інформаційна Панель та зведені дані: програми ТФА зазвичай мають інтерфейси інформаційної панелі, де користувачі можуть переглядати підсумки своєї щоденної, щотижневої та щомісячної активності. Ці підсумки можуть включати кількість кроків, пройденої відстані, кількість активних хвилин, кількість спалених калорій та іншу відповідну статистику [10].

Журнали активності та історія: користувачі можуть отримати доступ до детальних журналів та історії своїх занять фізичними вправами, включаючи мітки часу, типи активності, тривалість, відстані та кількість калорій. Це дозволяє користувачам відстежувати свій прогрес з плином часу і виявляти тенденції в їх активності [10].

Встановлення цілей та відстеження прогресу: програми ТФА дозволяють користувачам встановлювати персоналізовані цілі для занять спортом та відстежувати прогрес у їх досягненні. Користувачі можуть визначати цільові показники для кроків, відстані, хвилин активної діяльності або витрати калорій і отримувати повідомлення та відгуки, щоб залишатися мотивованими і йти в ногу з часом [10].

Соціальні функції та гейміфікація: багато програм ТФА включають функції соціальних мереж, що дозволяють користувачам спілкуватися з друзями, змагатися, ділитися досягненнями та змагатися з іншими. Елементи гейміфікації, такі як значки, нагороди та таблиці лідерів, підвищують залучення та мотивацію [10].

Конфіденційність та безпека даних. Політика конфіденційності та згода: програми ТФА реалізують політику конфіденційності та отримують згоду Користувача на збір, обробку та зберігання особистих медичних даних. Вони дотримуються правил захисту даних, таких як загальні правила захисту даних та закон про переносимість та підзвітність медичного страхування, щоб гарантувати конфіденційність користувачів [10].

Шифрування даних та безпечна передача: програми ТФА використовують методи шифрування для забезпечення безпеки передачі даних між пристроями та серверами. Вони використовують захищені протоколи, такі як Hypertext Transfer Protocol Secure (HTTPS), для захисту конфіденційної інформації від несанкціонованого доступу та перехоплення [10].

Аутентифікація користувачів та контроль доступу: програми ТФА реалізують механізми автентифікації для перевірки ідентичності користувачів та обмеження доступу авторизованих осіб. Користувачам може знадобитися

увійти за допомогою імен користувачів, паролів, біометричної автентифікації або двофакторної автентифікації, щоб отримати доступ до своїх даних про діяльність [10].

Таким чином, механізми програмного відстеження фізичної активності включають апаратні компоненти (мобільні пристрої, смартфони), програмні компоненти (мобільні застосунки, вебпортالي), алгоритми та методи обробки даних (визначення активності, підрахунок калорій, об'єднання даних), користувальницькі інтерфейси та взаємодії (перегляд панелі моніторингу, журнали активності, постановка цілей), а також заходи щодо забезпечення конфіденційності та безпеки даних (політика конфіденційності, шифрування, аутентифікація). Ці механізми працюють спільно, дозволяючи користувачам відстежувати, аналізувати і підвищувати рівень своєї фізичної активності, що в кінцевому підсумку сприяє поліпшенню здоров'я і самопочуття [10].

1.3 Порівняння існуючих аналогів

Найбільш популярні мобільні застосунки для відстеження фізичної активності іноді прив'язані до конкретного апаратного забезпечення – фітнес-браслетів конкретних виробників. Саме тому, слід розглянути порівняння застосунків, які не прив'язані до конкретних моделей та виробників фітнес приладів [11], [12].

Google Fit – це широко використовуваний застосунок для відстеження фізичної активності, доступний як для пристроїв Android, так і для iOS. Він дозволяє користувачам відстежувати свою фізичну активність, включаючи кроки, відстань, кількість спалених калорій і кількість активних хвилин, за допомогою датчиків, вбудованих в їх смартфони. Google Fit також інтегрується з іншими програмами та пристроями для здоров'я та фітнесу, надаючи повний огляд рівнів активності користувача та показників його здоров'я [11], [12]. Застосунок дозволяє встановлювати цілі, відстежувати

прогрес та отримувати персоналізовану інформацію, щоб допомогти користувачам залишатися мотивованими та активними (рис. 1.1).

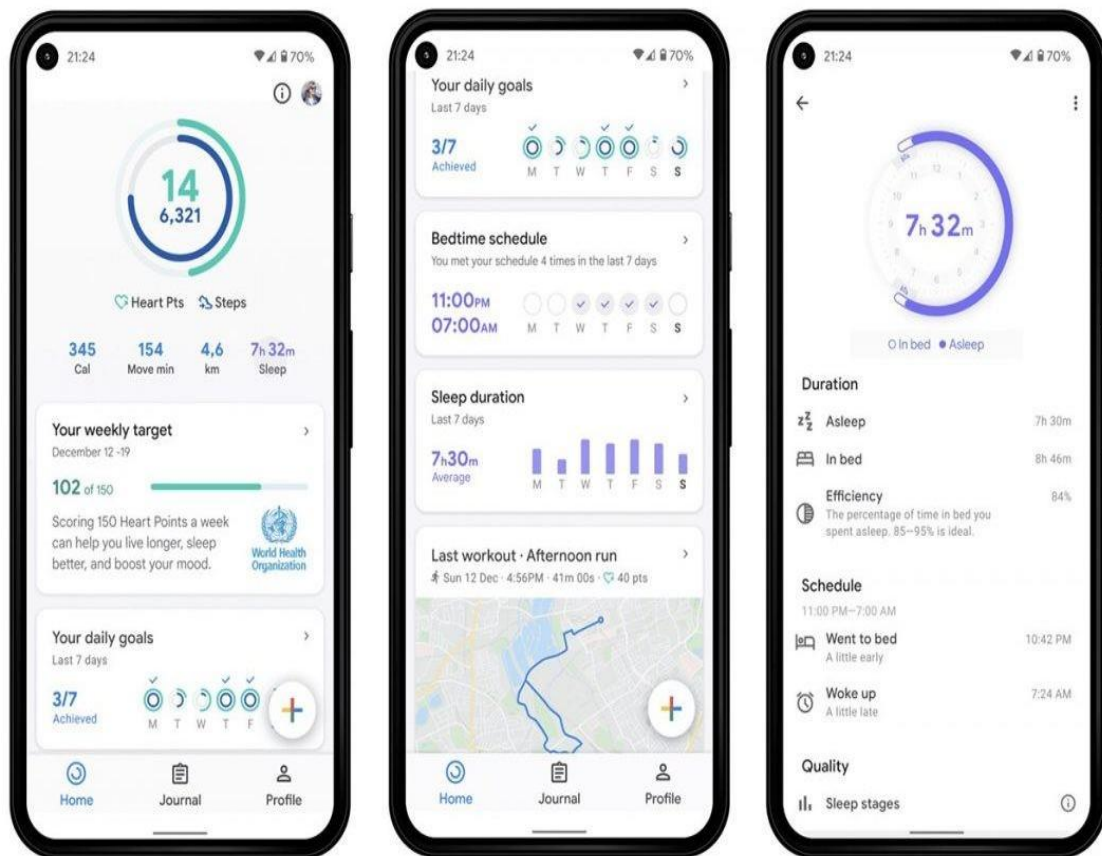


Рисунок 1.1 – Графічні інтерфейси застосунку Google Fit [11]

Apple Health – це вбудований застосунок для підтримки здоров'я та фітнесу, попередньо встановлений на iPhone та Apple Watch [13], [14]. Він автоматично відстежує різні показники, такі як кроки, відстань, пройдений переліт і кількість спалених калорій, за допомогою вбудованих датчиків пристроїв. Apple Health також інтегрується з широким спектром сторонніх застосунків і пристроїв, дозволяючи користувачам об'єднувати дані про своє здоров'я і активності на одній централізованій платформі. Застосунок пропонує такі функції, як підсумок активності, аналіз тенденцій та управління медичними записами, які допомагають користувачам досягати своїх цілей у фітнесі та підтримувати загальне самопочуття [13], [14] (рис. 1.2).



Рисунок 1.2 – Графічні інтерфейси застосунку Apple Health [13]

Strava – популярний застосунок для відстеження фітнесу, орієнтований на заняття бігом та їзду на велосипеді [15], [16]. Він дозволяє користувачам відстежувати свої тренування на відкритому повітрі за допомогою GPS, записуючи такі показники, як відстань, темп, висота над рівнем моря і карти маршрутів. Strava також пропонує функції соціальних мереж, що дозволяють користувачам спілкуватися з друзями, приєднуватися до клубів, змагатися та ділитися своїми досягненнями. Застосунок надає аналітику результатів, таблиці лідерів сегментів та інформацію про тренування, щоб допомогти користувачам покращити свої спортивні показники та залишатися мотивованими [15], [16] (рис. 1.3).

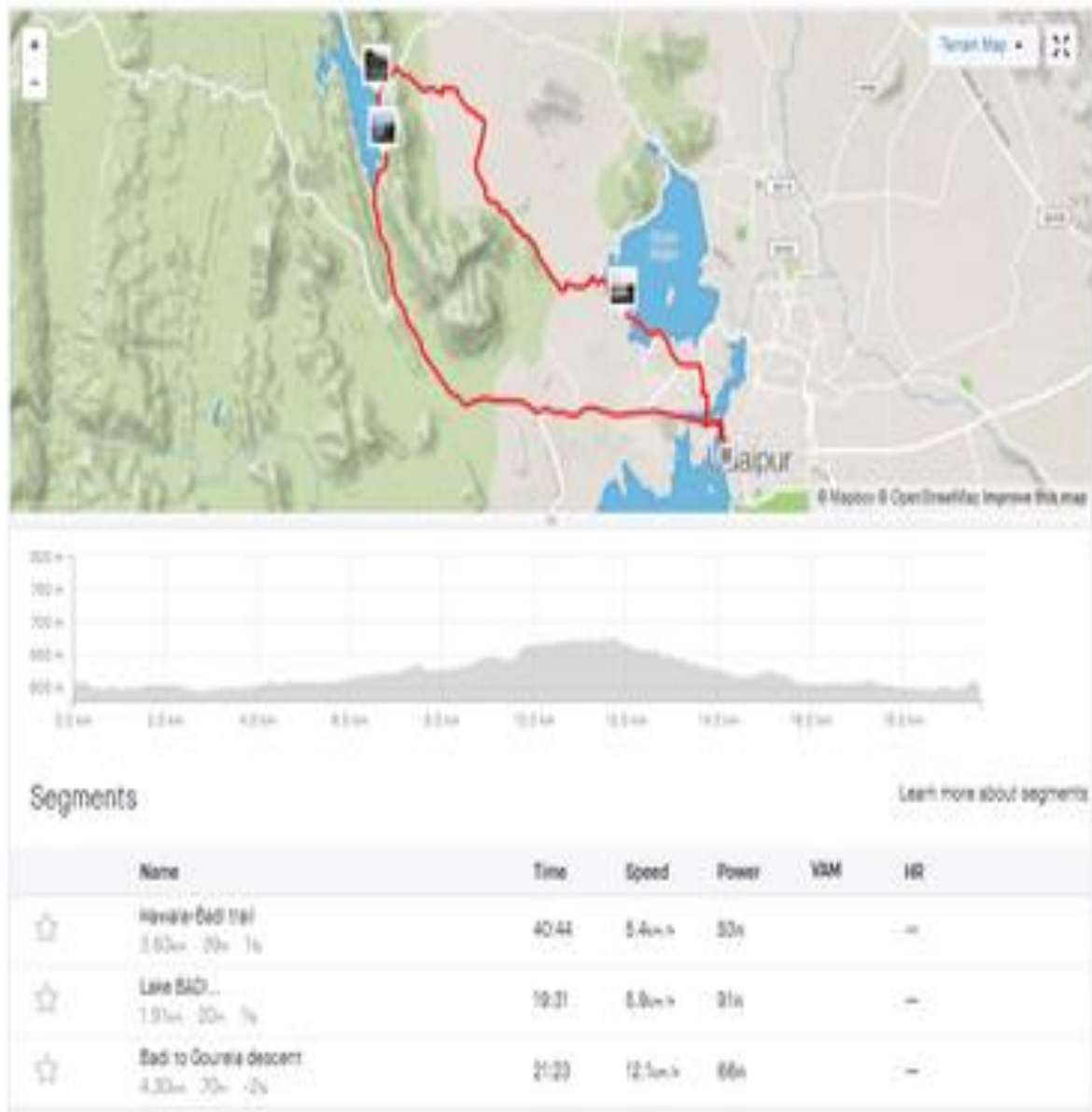


Рисунок 1.3 – Графічні інтерфейси застосунку Strava [15]

Ці три програми є одними з найбільш широко використовуваних і високо оцінених застосунків для відстеження фізичної активності, пропонуючи широкий спектр функцій для підтримки користувачів у моніторингу, аналізі та оптимізації рівня їх фізичної активності.

Виконаємо порівняння існуючих аналогів за критеріями та наведемо його у вигляді табл. 1.1. Порівняння буде проводитися за наступними критеріями:

- платформи – цей критерій вказує на операційні системи (ОС), що підтримуються застосунком, допомагаючи користувачам визначити, чи сумісна програма з їх мобільним пристроєм;

- типи активності – цей критерій описує діапазон фізичних навантажень, які можна відстежувати за допомогою програми, що дозволяє користувачам оцінити, чи відповідає програма їх бажаним режимам тренувань;

- методи відстеження – описує, як програма збирає дані про фізичну активність, будь то за допомогою датчиків смартфонів, носячих пристроїв, GPS або інших засобів;

- інтегрування – вказує, чи може застосунок інтегруватися з іншими програмами для здоров'я та фітнесу або пристроями, дозволяючи користувачам консолідувати дані про свою активність з кількох джерел.

- соціальні функції – дозволяє використовувати всі функції соціальних мереж, пропоновані застосунком, такі як спілкування з друзями, приєднання до клубів, змагання та обмін досягненнями;

- постановка цілей – вказує, чи підтримує програма постановку цілей, дозволяючи користувачам визначати цілі для кроків, відстані, активних хвилин або інших показників і відстежувати свій прогрес у досягненні цих цілей;

- аналіз даних – описує рівень аналітики та інсайтів, що надаються застосунком, включаючи зведення, тенденції та показники продуктивності, які допомагають користувачам інтерпретувати дані про свою діяльність і приймати обґрунтовані рішення;

- спеціалізація – це вказує на будь-яку конкретну спрямованість або спеціалізацію програми, таку як біг, їзда на велосипеді або відстеження загальної активності, орієнтовану на користувачів з особливими інтересами або цілями у фітнесі.

Таблиця 1.1 – Порівняння програмних засобів для дослідження фізичної активності користувачів

Критерії порівняння	Google Fit	Apple Health	Strava	MyTraiFit
Платформи	Android, iOS	iOS	Android, iOS	Android
Типи активності	Ходьба, біг, їзда на велосипеді та інші види діяльності	Ходьба, біг, їзда на велосипеді та інші види діяльності	Біг, їзда на велосипеді, плавання та інші види діяльності	Ходьба, біг, їзда на велосипеді та інші види діяльності
Методи відстеження	Датчики смартфона	Датчики смартфона, Apple Watch	GPS (для активних занять), датчики смартфонів	GPS (для активних занять), датчики смартфонів
Інтегрування	Інтеграція з іншими застосунками і пристроями для здоров'я та фітнесу	Інтегрується з іншими застосунками та пристроями для здоров'я та фітнесу	Інтегрується з деякими фітнес-пристроями, сторонніми застосунками	Інтеграція з іншими застосунками
Соціальні функції	Обмежені соціальні функції	Обмежені соціальні функції	Широкі соціальні можливості, клуби, змагання	Обмежені соціальні функції
Постановка цілей	Так	Так	Так	Так
Аналіз даних	Базова аналітика та аналітичні матеріали	Базова аналітика та аналітичні матеріали	Розширена аналітика продуктивності, таблиці лідерів сегментів	Розширена аналітика продуктивності, таблиці трекінгу
Спеціалізація	Загальне відстеження активності	Загальне відстеження активності	Фокус на бігових та велосипедних тренуваннях	Загальне відстеження активності

Як видно з порівняння, на сьогоднішній день усі існуючі аналоги мають певні обмеження. Більш того, більшість застосунків не надають

повноцінного трекінгу впродовж пролонгованого відрізка часу, а скоріше просто збирають поточну інформацію.

Відповідно актуальною задачею залишається програмна реалізація мобільного застосунку для ТФА.

1.4 Постановка завдання

Розробка мобільного застосунку для відстеження фізичної активності є актуальним завданням, враховуючи глобальну поширеність малорухливого способу життя і пов'язані з цим ризики для здоров'я. Ці програми мають численні переваги у підвищенні фізичної активності та загального самопочуття, включаючи моніторинг поведінки та зворотний зв'язок, встановлення цілей та підзвітність, персоналізацію та адаптивність, інтеграцію з мобільними пристроями, соціальну підтримку та залучення громади, освітні ресурси та методи зміни поведінки, відстеження здоров'я та аналітичну інформацію, віддалену доступність та гнучкість та охорону здоров'я інтеграція та лікування захворювань. Використовуючи силу сучасних технологій, програми ТФА можуть допомогти людям вести здоровий та активний спосіб життя та зменшити тягар хронічних захворювань для суспільства.

Відповідно, можна сформулювати мету даної роботи – розробка мобільного застосунку – трекеру фізичної активності.

1.4.1 Функціональність системи

Система має реалізовувати наступні функціональні вимоги [17], [18]:

а) відстеження активності:

- запис і відстеження різних фізичних навантажень, таких як ходьба, біг, їзда на велосипеді та інші вправи;
- збір даних про показники, включаючи кроки, відстань, тривалість, кількість спалених калорій та кількість активних хвилин;

б) моніторинг в режимі реального часу:

- забезпечення моніторингу фізичної активності в режимі реального часу за допомогою датчиків смартфонів;
- відображення поточного оновлення показників активності під час тренувань;

в) постановка цілей і відстеження прогресу:

- дозволяє користувачам встановлювати індивідуальні цілі активності на основі таких показників, як кроки, відстань або хвилини;
- відстеження та візуалізація прогресу у досягненні цілей за допомогою підсумків, діаграм або показників прогресу;

г) налаштування та персоналізація:

- надання можливості користувачам налаштовувати цілі своєї діяльності, переваги і налаштування повідомлень;
- надання персоналізованих рекомендацій та аналітичних матеріалів на основі даних користувачів та уподобань;

д) аналіз даних та аналітична інформація:

- аналіз даних про активність, щоб отримати уявлення про тенденції, закономірності та взаємозв'язки;
- надання практичних рекомендацій щодо підвищення рівня активності та досягнення цілей у фітнесі.

е) соціальні функції та залучення спільноти:

- надання користувачам можливості ділитися своїми досягненнями, тренуваннями та прогресом з друзями та послідовниками;

ж) гейміфікація та мотивація:

- використання елементів гейміфікації, такі як значки, нагороди та віртуальні завдання, щоб мотивувати користувачів;
- можливість пропонувати стимули для досягнення контрольних показників і підтримки постійного рівня активності;

з) планування тренувань і рекомендації:

- надання інструментів для планування тренувань, програми вправ та тренінгові програми, адаптовані до цілей користувачів та рівня їх фізичної підготовки;
- пропонування рекомендацій щодо правильної форми, техніки та інтенсивності для різних видів вправ.

1.4.2 Вимоги до інтерфейсу

Інтерфейс системи повинен бути ергономічним, інтуїтивно-зрозумілим для користувачів системи. Вся інформація сайту повинна бути чітко структурована і розділена на розділи. Розділ пошуку повинен мати настільки простий інтерфейс, наскільки можливо зробити без втрати його багатофункціональності. Сторінки реєстрації та авторизації користувача повинні мати підказки з приводу заповнення форм [17], [18].

1.4.3 Вимоги до інтерфейсу

Застосунок має відповідати правилам доступності та зручності для користувачів, а саме [17], [18]:

- забезпечення доступності функцій для користувачів з обмеженими можливостями, таких як голосові команди, зчитувачі з екрана та настроювані розміри шрифту;
- розробка зручного інтерфейсу з інтуїтивно зрозумілою навігацією, чіткими візуальними підказками і адаптивним взаємодією.

1.4.4 Вимоги до безпеки

Забезпечення конфіденційності та безпеки даних:

- дотримання відповідних правил та рекомендацій щодо захисту даних для захисту інформації користувача;

- застосування надійних заходів для захисту конфіденційності та безпеки даних користувачів, включаючи шифрування, автентифікацію та безпечну передачу даних.

1.4.5 Не функціональні вимоги до продукту

Обов'язкові вимоги.

Для сервера: серверна операційна система (ОС); підтримка Java Virtual Machine (JVM), підтримка системи управління базами даних (СУБД) MySQL.

Мінімальні системні вимоги та програмне забезпечення (ПЗ), що необхідні для функціонування серверної частини застосунку:

- продуктивний хостинг;
- підтримка JVM.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- смартфон під управлінням ОС Android;
- процесор із тактовою частотою від 2,0 GHz;
- підключення до мережі Internet.

1.5 Висновки за розділом 1

В першому розділі пояснювальної записки було розглянуто предметну область ТФА. А саме що таке ТФА, чому на сьогоднішній день це важливо, та які існують програмні застосунки для виконання ТФА. В першому розділі також реалізовано головну мету – формування набору вимог (функціональних та не функціональних) на основі дослідження існуючих аналогів, для подальшої розробки конкурентоспроможного застосунку.

2 МАТЕРІАЛИ І МЕТОДИ

Другий розділ буде присвячено обґрунтування вибору технічних засобів розробки: мови програмування, середовища розробки, програмних надбудов, тощо. Рішення прийняті в розділі будуть підтверджені та аргументовані.

2.1 Вибір мови програмування

Java вже давно є основною мовою розробки додатків для Android, і її популярність зберігається з кількох вагомих причин. Зв'язок Java з розробкою Android почався з моменту створення платформи в 2005 році. Коли Google придбала Android Inc. і анонсувала проєкт Android з відкритим вихідним кодом в 2007 році, Java був обраний в якості основної мови програмування для створення додатків Android [19], [20]. На це рішення вплинули широке поширення Java, розвинена екосистема, надійний інструментарій і зручні для розробників функції.

Протягом багатьох років Java міцно увійшла в екосистему розробки Android, і мільйони застосунків були створені за допомогою мови програмування Java. Її доступність, стабільність та надійність сприяли її постійній популярності серед розробників у всьому світі. Більш того, офіційне схвалення Google Java для розробки на Android зміцнило її статус стандарту де-факто для створення додатків на Android [19], [20].

Java може похвалитися багатою стандартною бібліотекою та повним набором API, надаючи розробникам потужні інструменти та утиліти для створення складних програм для Android. Java пропонує безліч вбудованих функцій, які полегшують розробку та підвищують продуктивність - від створення мереж та доступу до баз даних до проєктування інтерфейсу користувача (user interface – UI) та роботи з мультимедіа [19], [20].

Кросплатформність та портативність: головна ідея Java сприяє незалежності платформи та портативності коду, дозволяючи розробникам писати код, який можна виконувати на будь-якій платформі за допомогою віртуальної машини Java (Java virtual machine – JVM). Хоча програми для Android в кінцевому підсумку працюють у середовищах Dalvik або ART runtime, синтаксис та семантика Java залишаються послідовними на різних платформах, що полегшує крос-платформну розробку та повторне використання коду [19], [20].

Надійна система типу та управління пам'яттю: надійна система типу Java та автоматичне управління пам'яттю (за допомогою збору сміття) сприяють чистішому та легшому обслуговуванню коду з меншою кількістю витоків пам'яті та помилок під час виконання. Об'єктно-орієнтований характер мови та статичне введення сприяють багаторазовому використанню коду, модульності та масштабованості, полегшуючи розробку складних програм для Android [19], [20].

Java забезпечує ефективну продуктивність під час виконання та можливості оптимізації, дозволяючи розробникам створювати високопродуктивні Програми для Android, які безперебійно працюють на різних пристроях. Використовуючи належні методи кодування і оптимізації, розробники можуть мінімізувати споживання ресурсів, підвищити швидкість відгуку додатків і поліпшити загальний досвід роботи з користувачами [19], [20].

Java забезпечує надійну зворотну сумісність і довгострокову підтримку з боку Oracle і Java community. Розробники Android можуть покластися на Java, щоб забезпечити сумісність зі старими версіями Android та забезпечити сумісність своїх програм із майбутніми оновленнями та вдосконаленнями платформи [19], [20].

Java має розвинену та добре зарекомендовану екосистему бібліотек, фреймворків та інструментів розробки, розроблених спеціально для Android. Розробники мають доступ до широкого спектру ресурсів, включаючи

документацію, навчальні посібники, приклади коду та форуми спільноти, які полегшують навчання, усунення несправностей та співпрацю [19], [20].

Слід розглянути більш широкі наслідки для розробників та бізнесу. Знайомство з розробниками та набір навичок: багато розробників вже володіють Java досконало, або завдяки академічній підготовці, професійному досвіду, або самостійному навчанню. Використання наявних навичок і знань в області Java для розробки під Android зводить до мінімуму час навчання і прискорює процес розробки, що скорочує терміни виведення продукту на ринок і знижує витрати на розробку [19], [20].

Android може похвалитися найбільшою часткою ринку серед мобільних операційних систем, з мільярдами активних пристроїв у всьому світі. Вибираючи Java для розробки під Android, розробники отримують доступ до великої і різноманітної користувальницької бази, що максимально збільшує потенційний охоплення і вплив їх додатків на світовому ринку [19], [20].

Широке поширення Java в корпоративних середовищах ще більше зміцнює її позиції в якості кращого варіанту для розробки на Android. Багато підприємств та організацій інвестували значні кошти в технології, інфраструктуру та талановитих людей на базі Java, що робить Java найбільш підходящою для їхніх ініціатив з розробки Android. Більше того, функції корпоративного рівня Java, такі як масштабованість, надійність та безпека, добре узгоджуються з вимогами масштабних програм Android, розгорнутих у корпоративних середовищах [19], [20].

Сумісність Java з іншими технологіями та платформами сприяє синергії та інтеграції екосистеми, забезпечуючи безперебійну сумісність додатків Android із серверними системами, хмарними сервісами та сторонніми API. Розробники можуть використовувати платформи Java, такі як Spring та Hibernate, для створення надійних серверних сервісів, які доповнюють їхні програми для Android, полегшуючи синхронізацію даних, автентифікацію та реалізацію бізнес-логіки [19], [20].

На закінчення, Java залишається кращим вибором для розробки додатків для Android завдяки своєму історичному значенню, технічним перевагам, зрілості екосистеми та більш широкому значенню для розробників та бізнесу. Багата стандартна бібліотека, незалежність від платформи, надійна система введення, оптимізація продуктивності та сумісність з існуючими технологіями на базі Java роблять його ідеальною мовою програмування для створення складних і високоякісних додатків для Android, що відповідають потребам сучасних мобільних користувачів і підприємств. У той час як альтернативні мови, такі як Kotlin, стали прийнятними варіантами для розробки на Android, незмінна Популярність Java і велика екосистема роблять його надійним і перспективним вибором для розробників, які прагнуть створювати інноваційні та ефективні додатки для платформи Android [19], [20].

Проведемо порівняння мов програмування для реалізації вебзастосунків, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні:

- синтаксис – синтаксис мови програмування відноситься до її граматики та структури. Синтаксис може впливати на читабельність, можливість запису та супроводу коду. Під час порівняння розглядалися синтаксичні відмінності між Java, Kotlin та Python, підкресливши такі фактори, як зрозумілість, стислість та виразність;

- безпека неініціалізованих значень – відноситься до здатності мови безпечно обробляти нульові або неініціалізовані значення, зменшуючи ризик винятків з нульовим покажчиком. Критерій було розглянуто, щоб підкреслити відмінності в механізмах нульової безпеки між Java, Kotlin та Python, оскільки це має вирішальне значення для написання надійного коду;

- сумісність – під сумісністю розуміється здатність мови програмування легко взаємодіяти з іншими мовами або платформами. Під час порівняння сумісність Java, Kotlin та Python порівнювалася з існуючими кодовими базами та бібліотеками, особливо в контексті розробки Android;

- мовні можливості – охоплюють широкий спектр можливостей, включаючи типи даних, структури управління, функції та стандартні бібліотеки. Було досліджено функції та можливості Java, Kotlin та Python, які мають відношення до розробки Android, виділяючи такі аспекти, як спільні програми, функції розширення та безпека null;

- продуктивність – є критичним фактором у розробці мобільних застосунків, оскільки вона впливає на швидкість відгуку, час автономної роботи та загальне сприйняття користувачами застосунків. Розглянуто характеристики продуктивності Java, Kotlin та Python, включаючи ефективність виконання, методи оптимізації та особливості платформи;

- рівень освоєння – відноситься до складності або легкості, з якою розробники можуть освоїти мову програмування і стати професіоналами;

- інструментарій та підтримка – включають доступність інструментів розробки, документації, ресурсів спільноти та офіційних каналів підтримки мови програмування;

- екосистема – включає бібліотеки, фреймворки, інструменти та ресурси, що підтримують розвиток цієї мови. Розглянуто екосистеми Java, Kotlin та Python для розробки на Android, виділивши такі фактори, як доступність бібліотек, підтримка фреймворку та інтеграція з іншими технологіями;

- впровадження платформи – стосується поширеності та прийнятності мови програмування в межах певної платформи чи галузі;

- придатність – відноситься до відповідності мови програмування конкретному завданню або варіанту використання;

- спільнота та ресурси – визначають розмір, активність та якість спільноти розробників, а також доступність таких ресурсів, як навчальні посібники, форуми та документація. Оцінено спільноту та ресурси, доступні для Java, Kotlin та Python, у контексті розробки Android, враховуючи такі фактори, як залучення спільноти, доступність ресурсів та канали підтримки.

Таблиця 2.1 – Порівняння мов програмування

Критерій	Java	Kotlin	Python
Синтаксис	Знайомий синтаксис із крапками з комою та фігурними дужками	Сучасний синтаксис з лаконічними і виразними функціями	Лаконічний синтаксис зі зрозумілим і читабельним кодом
Нульова безпека	Відсутність вбудованих функцій захисту від обнулення	Забезпечує вбудований захист від null з використанням типів, які можуть бути як нульовими, так і ненульовими	Відсутній вбудований захист від null, але може бути реалізована за допомогою додаткових типів
Сумісність	Відмінна сумісність з існуючим кодом Java	Повна взаємодія з кодом Java, підтримує проекти на різних мовах	Обмежена сумісність з Java, вимагає додаткової настройки та інтеграції
Мовні особливості	Розвинена мова з великими можливостями і бібліотеками	Сучасна мова з такими функціями, як спільні програми, функції розширення та захист від null	Багата екосистема з бібліотеками для різних доменів, але менше функцій, специфічних для мобільних пристроїв
Уявлення	Висока продуктивність завдяки ефективному часу виконання та оптимізації	Продуктивність, порівнянна з Java, завдяки оптимізації функцій, специфічних для Kotlin	Як правило, нижча продуктивність порівняно з Java та Kotlin через інтерпретацію або компіляцію в байт-код
Етап навчання	Новачкам легше влаштуватися, особливо з об'єктно-орієнтованими концепціями	Більш плавний процес навчання, особливо для розробників, знайомих з Java	Відносно проста в освоєнні, особливо для розробників з досвідом роботи на Python

Продовження табл. 2.1

Критерій	Java	Kotlin	Python
Інструменти та підтримка	Розвинений інструментарій, велика документація та підтримка спільноти	Зростаюча екосистема за офіційної підтримки Google і успішне впровадження спільнотою	Обмежений набір інструментів та підтримка спільноти розробників Android порівняно з Java та Kotlin
Екосистема	Велика екосистема бібліотек, фреймворків та інструментів, адаптованих для розробки на Android	Розширюється екосистема з підтримкою розробки на Android, що використовує існуючі бібліотеки Java	Обмежена екосистема для розробки на Android порівняно з Java та Kotlin
Впровадження платформи	Широко використовується в якості офіційної мови для розробки на Android	Офіційно підтримується як першокласна мова для розробки на Android і набуває все більшого поширення	Менш широко використовується для розробки на Android, в основному за допомогою сторонніх платформ, таких як Kivy і Beware
Придатність	Добре підходить для створення надійних і масштабованих додатків для Android	Ідеально підходить для сучасної розробки на Android, пропонуючи розширені можливості і продуктивність	Підходить для конкретних випадків використання, таких як прототипування, аналіз даних або нішеві програми
Спільнота та ресурси	Велика та активна спільнота, великі ресурси та канали підтримки	Зростаюча спільнота зі зростаючими ресурсами та підтримкою, яку підтримують JetBrains та Google	Менша спільнота порівняно з Java та Kotlin, обмежені ресурси та підтримка розробки Android

Ця порівняльна таблиця надає всебічний огляд ключових функцій, міркувань та компромісів, пов'язаних із розробкою застосунків для Android за допомогою Java, Kotlin та Python. Хоча кожна мова має унікальні сильні сторони та переваги, розробники повинні ретельно оцінювати вимоги до свого проєкту, досвід команди та особливості платформи, вибираючи найбільш підходящу мову для розробки на Android. Враховуючи переваги, розвинутий інструментарій та досвід роботи, вибір було зроблено на користь саме Java.

2.2 Вибір та обґрунтування середовища розробки

Android Studio широко відома як одна з найкращих середовищ розробки (integrated development environment – IDE) для розробки на ОС Android завдяки кільком ключовим факторам, які сприяють її популярності та ефективності. Ось кілька причин, чому Android Studio виділяється як винятковий вибір для розробки Android [21], [22].

Android Studio – це офіційне IDE для Android, розроблене та підтримуване Google. Як офіційне середовище розробки, воно регулярно отримує оновлення, виправлення помилок та нові функції безпосередньо від Google, забезпечуючи сумісність з останніми версіями Android SDK та оновленнями платформи [21], [22].

Android Studio – IDE, що було створено спеціально для розробки на Android, надаючи повний набір інструментів, функцій і робочих процесів, адаптованих до потреб розробників Android. Воно пропонує спеціалізовані функції для проєктування користувацьких інтерфейсів, написання коду, налагодження, тестування та розгортання програм для Android [21], [22].

Android Studio включає потужний редактор коду з розширеними функціями, такими як підсвічування синтаксису, завершення коду, рефакторинг та аналіз коду. Воно надає розумні пропозиції щодо коду, виявлення помилок та автоматичне форматування, допомагаючи розробникам писати чистий, ефективний та підтримуваний код [21], [22].

Розширений редактор макетів: Android Studio включає в себе розширений редактор макетів, який дозволяє розробникам візуально створювати користувацькі інтерфейси для своїх застосунків Android. Редактор макетів підтримує функцію перетягування, попередній перегляд у режимі реального часу та візуальне редагування XML-файлів макетів, що дозволяє легко створювати ідеальні користувацькі інтерфейси для різних розмірів екрана та орієнтацій [21], [22].

Android Studio включає вбудовані інструменти профілювання продуктивності, які дозволяють розробникам аналізувати продуктивність своїх додатків для Android і виявляти вузькі місця в продуктивності. Ці інструменти дозволяють розробникам відстежувати використання процесора, пам'яті та мережі, оптимізувати продуктивність додатків та покращити взаємодію з користувачами [21], [22].

IDE легко інтегрується з тестовими платформами, такими як JUnit та Espresso, дозволяючи розробникам писати та виконувати модульні тести, інтеграційні тести та тести інтерфейсу користувача для своїх програм Android. Вона надає інструменти для запуску тестів, перегляду результатів тестування і налагодження помилок тестування безпосередньо в середовищі IDE [21], [22].

Android Studio включає потужний емулятор, який дозволяє розробникам тестувати свої програми Android на віртуальних пристроях з різними конфігураціями та рівнями API. Він також підтримує підключення фізичних пристроїв Android для тестування та налагодження, забезпечуючи комплексне середовище тестування для розробки Android [21], [22].

Середовище включає вбудовану підтримку систем контролю версій, таких як Git, що дозволяє розробникам керувати сховищами коду, відстежувати зміни та ефективно співпрацювати з членами команди. Вона надає інструменти для внесення змін, об'єднання гілок, вирішення конфліктів і перегляду історії змін безпосередньо в середовищі IDE [21], [22].

Android Studio підтримується великою документацією, навчальними посібниками та ресурсами, наданими Google та спільнотою розробників

Android. Розробники можуть отримати доступ до офіційної документації, прикладів коду, навчальних курсів та форумів спільноти, щоб вивчити розробку Android, усунути неполадки та бути в курсі найкращих практик [21], [22].

Постійне вдосконалення та оновлення: Android Studio постійно вдосконалюється та оновлюється, регулярно випускаючи нові функції, вдосконалення та виправлення помилок. Google активно запитує відгуки розробників і враховує пропозиції користувачів і побажання по функціоналу в майбутніх випусках, гарантуючи, що Android Studio залишиться передовою і зручною для розробників середовищем розробки для Android [21], [22].

Загалом, поєднання потужних функцій Android Studio, спеціалізованого дизайну, комплексного інструментарію та потужної підтримки з боку Google та спільноти Android робить його однією з найкращих IDE для розробки на Android, дозволяючи розробникам ефективно створювати високоякісні, багатфункціональні програми для Android [21], [22].

Порівняльна характеристика Android Studio з іншими найпоширенішими IDE для розробки мобільних застосунків у таблиці 2.2.

Серед критеріїв порівняння будемо розглядати наступні:

- платформа – цей критерій вказує, чи розроблена IDE спеціально та оптимізована для розробки на Android, чи це середовище розробки загального призначення;
- спеціалізоване середовище розробки – чи призначене IDE спеціально для розробки на Android, чи це більш універсальне середовище розробки Java або редактор коду;
- інтеграція Android SDK – ступінь, в якій IDE легко інтегрується з Android Software Development Kit (SDK) для створення додатків Android;
- дизайн інтерфейсу користувача – доступність та можливості інструментів візуального проєктування інтерфейсу користувача в IDE для розробки інтерфейсів додатків Android;

- профілювання продуктивності – чи включає IDE вбудовані інструменти для профілювання продуктивності та оптимізації додатків Android;
- завершення коду – якість і повнота функцій завершення коду, що надаються IDE для надання допомоги розробникам в написанні коду;
- рефакторинг – наявність інструментів та функцій для рефакторингу коду з метою покращення його структури, читабельності та супровідності;
- налагодження – можливості та ефективність інструментів налагодження, що надаються IDE для виявлення та усунення проблем у програмах Android;
- тестування – ступінь, до якої IDE підтримує платформи тестування та інструменти для автоматизованого тестування програм Android;
- контроль версій – вбудована в IDE підтримка систем контролю версій, таких як Git, для управління сховищами коду та співпраці;
- екосистема плагінів – наявність різноманітної екосистеми плагінів та розширень, які розширюють функціональність IDE та задовольняють конкретні потреби та уподобання;
- підтримка спільноти – розмір, активність та готовність допомогти спільноті розробників, пов'язаній з IDE, включаючи форуми, документацію та канали підтримки;
- рівень навчання – складність або легкість, з якою розробники можуть навчитися ефективно використовувати IDE, враховуючи такі фактори, як складність, знайомство та доступні навчальні ресурси;
- споживання ресурсів – обсяг системних ресурсів (наприклад, процесора, пам'яті), споживаних IDE під час використання, що може вплинути на продуктивність і зручність роботи з користувачем;
- налаштованість – ступінь, в якій IDE може бути налаштована і персоналізована за допомогою тим, плагінів, налаштувань та інших параметрів конфігурації..

Таблиця 2.2 – Порівняння IDE

Критерій	Android Studio	IntelliJ IDEA	Visual Studio Code
Платформа	Специфічний для Android	Java IDE загального призначення	Універсальний редактор коду з розширеннями для розробки на Java і Android
Спеціально розроблений	Так	Ні	Ні
Інтеграція з Android SDK	Повна інтеграція з Android SDK	Повна інтеграція з Android SDK	Розширення для розробки на Android не потрібні
Дизайн інтерфейсу користувача	Редактор макетів для візуального проектування інтерфейсу користувача	Редактор макетів для візуального проектування інтерфейсу користувача	Обмежені можливості візуального проектування користувацького інтерфейсу
Профілювання продуктивності	Вбудовані засоби профілювання продуктивності	Вбудовані засоби профілювання продуктивності	Потрібні розширення для профілювання продуктивності
Завершення коду	Комплексне доопрацювання коду	Комплексне доопрацювання коду	Основне завершення коду за допомогою розширень
Рефакторинг	Повний набір інструментів рефакторингу	Повний набір інструментів рефакторингу	Обмежені можливості рефакторингу за допомогою розширень
Налагодження	Розширені можливості налагодження	Розширені можливості налагодження	Базові можливості налагодження за допомогою розширень
Тестування	Інтегровані інструменти тестування та фреймворки	Інтегровані інструменти тестування та фреймворки	Потрібні розширення для тестування

Продовження табл. 2.2

Критерій	Android Studio	IntelliJ IDEA	Visual Studio Code
Управління версіями	Вбудована підтримка контролю версій	Вбудована підтримка контролю версій	Інтегрована підтримка контролю версій з розширеннями
Екосистема плагінів	Велика екосистема плагінів	Велика екосистема плагінів	Багата торгова площа розширень з різноманітними плагінами
Підтримка спільноти	Велика та активна спільнота	Велика та активна спільнота	Велика та активна спільнота
Процес навчання	Помірне, адаптоване для розробки під Android	Помірне, спеціально розроблене для розробки на Java	Низька вартість, вимагає знайомства з розширеннями
Споживання ресурсів	Ресурсномістке завдяки багатству функціональних можливостей	Ресурсномістке завдяки багатству функціональних можливостей	Легка вага, мінімальне споживання ресурсів
Налаштованість	Легко настраюється за допомогою тим, плагінів і шаблонів	Легко настраюється за допомогою тим, плагінів і шаблонів	Легко настраюється за допомогою розширень, тим і налаштувань

У цій порівняльній таблиці наведено огляд ключових функцій, сильних сторін та рекомендацій для Android Studio, IntelliJ IDEA та Visual Studio Code (з розширеннями) у контексті розробки застосунків для Android. Спираючись на набір сильних характеристик було обрано Android Studio для розробки застосунку.

2.3 Структурна схема розробки

Загальна схема системи за обраною темою на рис. 2.1.

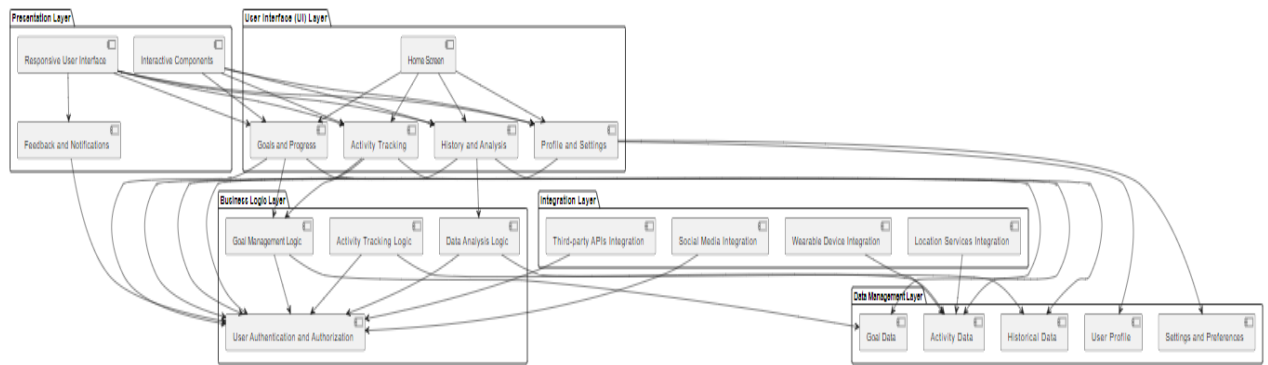


Рисунок 2.1 – Загальна схема системи за обраною темою

Рівень UI:

- початковий екран – відображає огляд щоденної активності користувача, включаючи пройдені кроки, пройдену відстань, спалені калорії та хвилини активного відпочинку;
- відстеження активності – дозволяє користувачам запускати, призупиняти та припиняти відстеження різних фізичних навантажень, таких як ходьба, біг, їзда на велосипеді та інші вправи;
- цілі та прогрес – надає функції для встановлення цілей активності, відстеження прогресу та візуалізації досягнень за допомогою графіків та показників прогресу;
- історія та аналіз – дозволяє користувачам переглядати дані про діяльність за минулі періоди, аналізувати тенденції та отримувати інформацію про свою діяльність та ефективність з часом;
- профіль та налаштування – дозволяє користувачам керувати інформацією про свій профіль, налаштовувати параметри програми та такі параметри, як одиниці виміру, сповіщення та параметри конфіденційності.

Рівень управління даними:

- дані про діяльність – зберігає створені користувачем дані про діяльність, включаючи кроки, відстань, тривалість, спалені калорії та тип діяльності;

- дані про цілі – зберігає задані користувачем цілі активності, включаючи кількість пройдених кроків, дистанцію, тривалість і кількість спалених калорій;

- профіль користувача – зберігає інформацію про користувача, таку як ім'я, вік, стать, вага, зріст та рівень фізичної підготовки;

- налаштування та налаштування – зберігає налаштування користувача та програми, включаючи одиниці виміру (наприклад, метричні чи імперські), налаштування сповіщень та налаштування конфіденційності;

- історичні дані – зберігає історичні дані про діяльність для аналізу, визначення тенденцій та звітування.

Рівень бізнес-логіки:

- логіка відстеження активності – реалізує алгоритми і логіку для відстеження різних фізичних навантажень в режимі реального часу, розрахунку таких показників, як кроки, відстань і спалені калорії, а також обробки взаємодій з користувачами під час сеансів відстеження активності;

- логіка управління цілями – керує цілями діяльності, визначеними Користувачем, відстежує прогрес у досягненні цілей і генерує повідомлення або нагадування, що спонукають користувачів до досягнення своїх цілей;

- логіка аналізу даних – аналізує дані про діяльність, виявляє тенденції та закономірності, генерує ідеї та рекомендації щодо підвищення рівня активності та надає зворотний зв'язок користувачам на основі їх результатів;

- аутентифікація та авторизація користувачів – реалізує механізми аутентифікації, авторизації та контролю доступу для забезпечення безпечного доступу до даних та функцій користувача.

Рівень інтеграції:

- інтеграція служб локації – використовує GPS та служби локації для відстеження заходів на свіжому повітрі, таких як біг, їзда на велосипеді та піші прогулянки, а також для обчислення таких показників, як пройдена відстань та відображення маршруту;

- інтеграція сторонніх API – інтегрується зі сторонніми API для доступу до додаткових джерел даних, таких як інформація про погоду, бази даних про харчування або показники здоров'я, для покращення функціональності програми та зручності використання;

- інтеграція соціальних медіа – дозволяє користувачам ділитися своїми досягненнями у фізичній активності, оновленнями прогресу та підсумками тренувань з друзями та послідовниками на платформах соціальних медіа, сприяючи соціальній взаємодії та мотивації.

Рівень презентації:

- адаптивний користувацький інтерфейс – дані відстеження активності, цілі, прогрес і аналіз представлені в візуально привабливою і зручною для користувача формі, оптимізованої для мобільних пристроїв з різними розмірами екранів і дозволами;

- інтерактивні компоненти – надає інтерактивні елементи, такі як кнопки, повзунки, прапорці та випадаючі списки, щоб полегшити введення користувачем, навігацію та взаємодію з функціями програми;

- зворотній зв'язок і повідомлення – відображає повідомлення зворотного зв'язку, оповіщення, нотифікації і нагадування для доведення важливої інформації до користувачів, надання відгуків про ефективність їх дій і спонукання їх до прийняття необхідних заходів або внесення змін.

На цій структурній схемі представлені основні компоненти та функціональні можливості мобільного застосунки для відстеження фізичної активності, розділені на окремі рівні для полегшення розробки, обслуговування та масштабування. Кожен рівень включає в себе певні обов'язки і взаємодіє з іншими рівнями для забезпечення безперебійної взаємодії з користувачем.

2.4 Висновки за розділом 2

Другий розділ було присвячено роботі по вибору технічних засобів розробки та аргументації цих рішень. Так мовою програмування обрано Java, а IDE – Android Studio.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проєктування моделей системи

Моделювання за допомогою Unified Modeling Language (UML) є цінним інструментом протягом усього життєвого циклу розробки програмного забезпечення, в тому числі при розробці мобільних додатків, таких як додатки для відстеження фізичної активності [23], [24].

На початкових етапах розробки діаграми UML, такі як діаграми прецедентів, активностей та станів можуть допомогти зрозуміти функціональні вимоги програми. Діаграми прецедентів можуть відображати взаємодію між користувачами та системою, тоді як діаграми активностей можуть ілюструвати потік дій всередині програми [23], [24].

Діаграми UML відіграють вирішальну роль у розробці архітектури та структури додатків. Діаграми класів можуть бути використані для моделювання статичної структури програми, показуючи класи, атрибути, методи та взаємозв'язки між ними. Діаграми компонентів можуть ілюструвати компоненти високого рівня та їх взаємодію, тоді як діаграми пакетів дозволяють організувати компоненти в логічні групи [23], [24].

Діаграми стану UML та діаграми послідовностей можуть бути використані для моделювання динамічної поведінки програми. Діаграми стану можуть відображати різні стани та переходи всередині програми, тоді як діаграми послідовностей можуть ілюструвати послідовність взаємодій між об'єктами або компонентами в різних сценаріях [23], [24].

Діаграми UML можуть служити цінним керівництвом на етапі впровадження процесу розробки. Діаграми класів можуть служити основою для написання коду, допомагаючи розробникам зрозуміти структуру програми та взаємозв'язки між класами. Діаграми послідовностей можуть допомогти зрозуміти процес управління та взаємодію між об'єктами або компонентами [23], [24].

Діаграми UML служать ефективною документацією для програми, відображаючи її архітектуру, дизайн та поведінку у візуальному форматі. Вони забезпечують чітке та стисле представлення системи, яке легко зрозуміти розробники, тестери та інші зацікавлені сторони. Діаграми UML також можуть бути використані для інформування членів команди та зацікавлених сторін про проєктні рішення, вимоги та обмеження [23], [24].

UML-моделювання підтримує ітеративні процеси розробки, дозволяючи розробникам з часом удосконалювати і еволюціонувати дизайн програми. У міру зміни вимог або додавання нових функцій UML-діаграми можуть оновлюватися і переглядатися з урахуванням змін, гарантуючи, що дизайн буде відповідати мінливим потребам проєкту [23], [24].

В цілому, UML-моделювання – це універсальний і потужний метод, який підтримує розробку мобільних додатків, таких як додаток для відстеження фізичної активності, забезпечуючи системний підхід до аналізу вимог, проєктування, впровадження та документації. Ефективно використовуючи UML-діаграми, розробники можуть створювати добре структуровані, зручні в обслуговуванні і масштабовані додатки, що відповідають потребам користувачів і зацікавлених сторін [23], [24].

Діаграма класів на рис. 3.1. Ця діаграма класів UML описує основні класи та їх взаємозв'язки в мобільному додатку для відстеження фізичної активності. Вона включає класи для користувача, активності, цілі і трекера активності. Клас Activity Tracker керує даними користувачів, діями та цілями. У кожного користувача може бути кілька пов'язаних з ними дій і цілей [19], [20].

Діаграма прецедентів наведена на рис. 3.2. На цій діаграмі:

- суб'єкт Користувач взаємодіє з додатком для відстеження фізичної активності;
- варіанти використання представляють основні функціональні можливості програми.

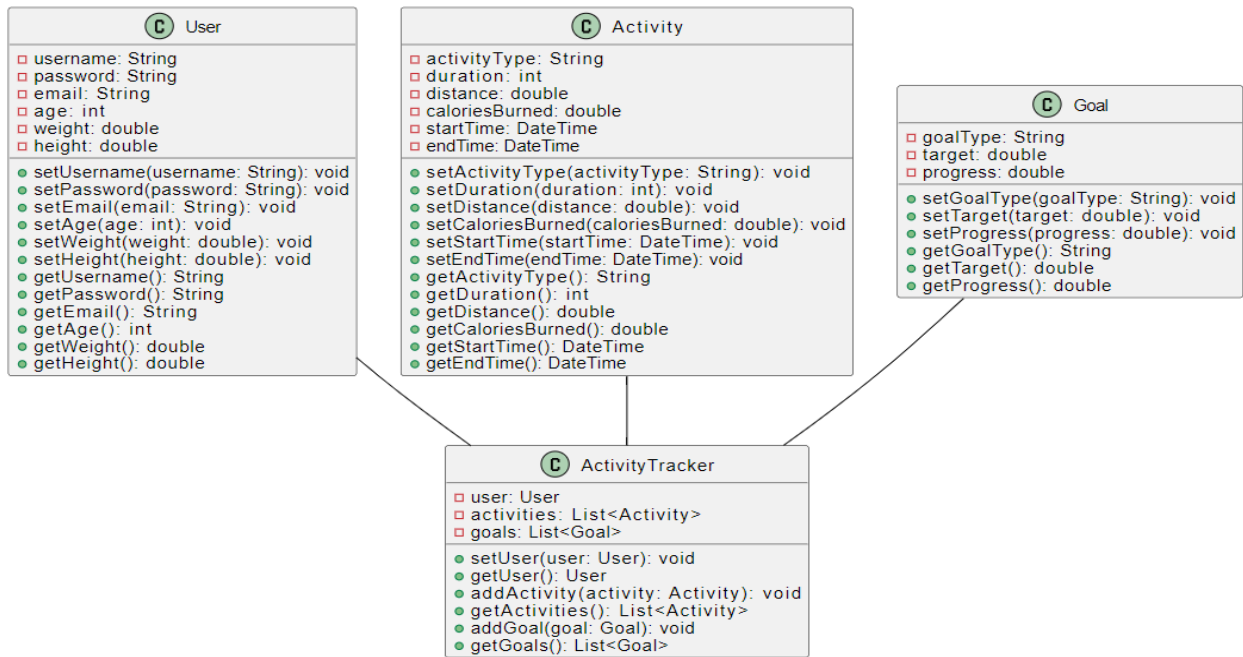


Рисунок 3.1 – Діаграма класів

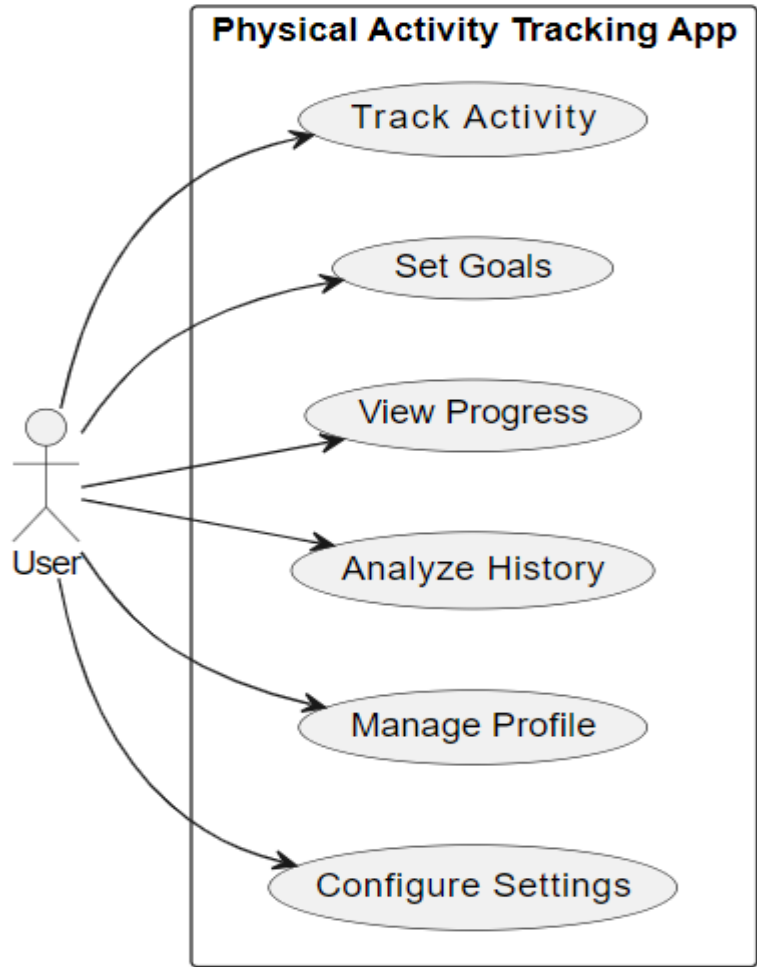


Рисунок 3.2 – Діаграма прецедентів

Варіантами використання є такі:

- відстеження активності: дозволяє користувачеві почати відстежувати свою фізичну активність;
- встановлення цілей: дозволяє користувачеві встановлювати цілі активності, такі як кількість кроків, пройдена відстань або кількість спалених калорій;
- перегляд прогресу: дозволяє користувачеві переглядати прогрес у досягненні поставлених цілей;
- аналіз історії: надає користувачеві інформацію про їх діяльність за останні роки та тенденції;
- управління профілем: дозволяє користувачеві переглядати та оновлювати інформацію про свій профіль;
- налаштування параметрів: дозволяє користувачеві налаштувати параметри програми, такі як одиниці виміру, сповіщення та налаштування конфіденційності.

На цій діаграмі випадків використання описані основні взаємодії між Користувачем і мобільним додатком для відстеження фізичної активності з акцентом на ключові функціональні можливості, що надаються додатком.

Діаграма станів наведена рис. 3.3. На цій діаграмі стану:

- застосунок запускається, коли користувач виходить із системи;
- коли користувач запускає програму, вона переходить від стану очікування до стану виходу;
- якщо Користувач ініціює вхід, він переходить у стан входу. Після успішного входу він переходить у стан LoggedIn, інакше повертається до стану "Idle";
- у стані LoggedIn користувач може виконувати різні дії, такі як відстеження активності, перегляд прогресу, аналіз історії, управління профілем та налаштування параметрів;

– кожна з цих дій призводить до відповідного переходу в інший стан, що представляє різні стани, в яких може перебувати програма залежно від дій користувача;

– наприклад, Користувач може переглядати прогрес, аналізувати історію, керувати профілем або налаштовувати параметри, що призводить до переходів у відповідні стани.

Ця діаграма стану надає огляд різних станів та переходів у мобільному додатку для відстеження фізичної активності, фіксуючи взаємодію користувача та реакцію програми на ці взаємодії.

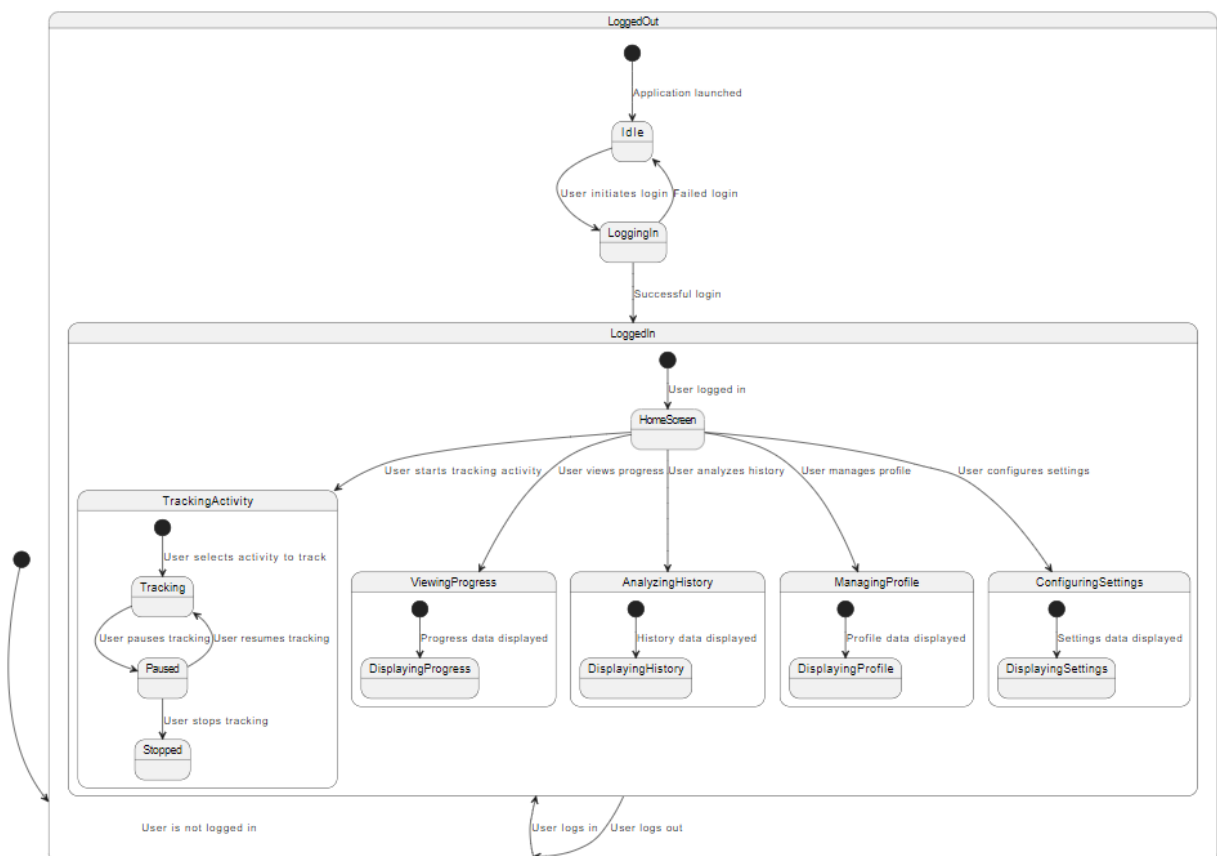


Рисунок 3.3 – Діаграма станів системи

Діаграма діяльності на рис. 3.4. На цій діаграмі дій:

– процес починається з того, що користувач запускає застосунок;
 – якщо користувач увійшов у систему, він перенаправляється на головний екран;

- користувач може виконувати різні дії на головному екрані, такі як початок відстеження активності, перегляд прогресу, аналіз історії, керування профілем або налаштування параметрів;
- кожна дія призводить до відповідної дії або набору дій, представлених гілками на діаграмі;
- подібним чином, якщо користувач вибирає перегляд прогресу, аналіз історії, керування профілем або налаштування параметрів, він перенаправляється на відповідні екрани або процеси;
- якщо Користувач не ввійшов в систему, йому пропонується виконати вхід. Якщо вхід успішний, вони перенаправляються на головний екран; інакше процес зупиняється.

Ця діаграма активності дає загальне уявлення про взаємодію та процеси користувачів у мобільному додатку для відстеження фізичної активності..

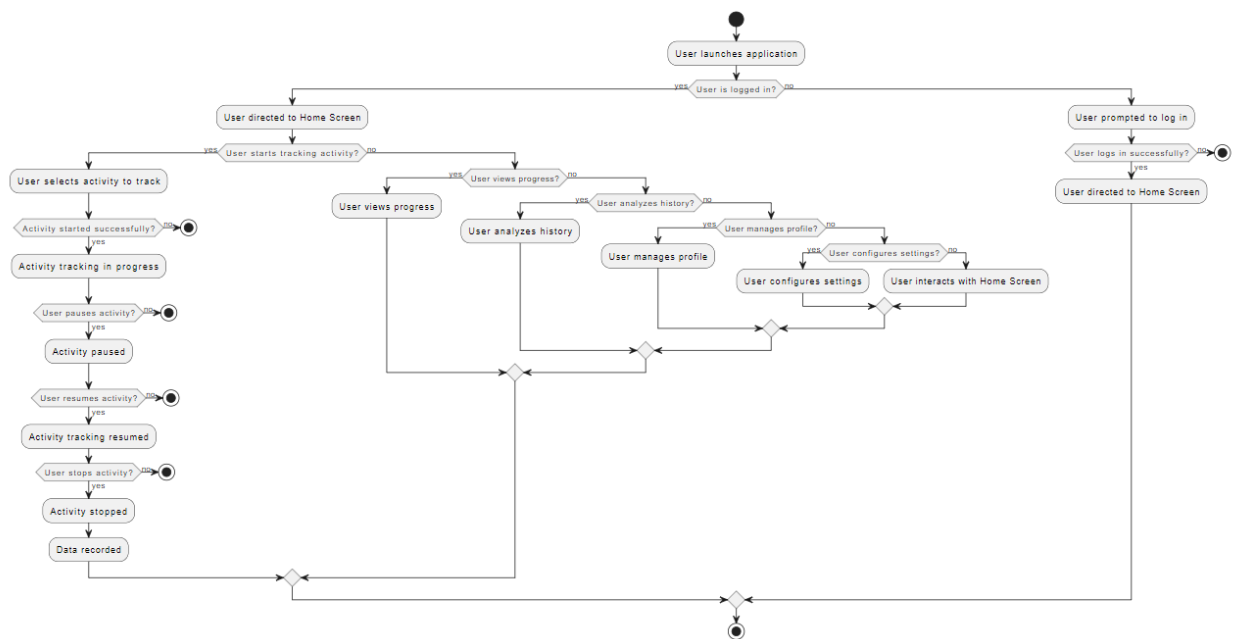


Рисунок 3.4 – Діаграма діяльності: замовлення товару

3.2 Проєктування інтерфейсу

Графічний інтерфейс користувача програми для ТФА відіграє найважливішу роль у забезпеченні інтуїтивно зрозумілої та привабливої

взаємодії користувачів. Ось кілька спеціальних функцій, які в ідеалі повинні бути включені в графічний інтерфейс програми для відстеження фізичної активності.

Зручна інформаційна панель – візуально приваблива інформаційна панель повинна надавати користувачам огляд показників їх щоденної активності, таких як пройдені кроки, пройдена відстань, спалені калорії та хвилини активного відпочинку. Чіткі і лаконічні дисплеї допомагають користувачам швидко зрозуміти, як вони просуваються до своїх цілей у фітнесі.

Інтерфейс відстеження активності – інтуїтивно зрозумілий інтерфейс відстеження активності дозволяє користувачам легко запускати, призупиняти, відновлювати та припиняти відстеження різних фізичних навантажень, таких як ходьба, біг, їзда на велосипеді та тренування. Візуальні підказки та відгуки допомагають користувачам точно відстежувати свої дії.

Постановка цілей і відстеження прогресу – інтерактивні функції для постановки цілей діяльності та відстеження прогресу в досягненні цих цілей мотивують користувачів залишатися активними. Індикатори прогресу, діаграми і графічні зображення наочно відображають прогрес, даючи користувачам відчуття досягнення і мотивації.

Аналіз історичних даних – інструменти для аналізу історичних даних про діяльність дозволяють користувачам переглядати тенденції, закономірності та отримувати уявлення про їхні звички щодо діяльності з часом. Зображення, такі як лінії тренду, гістограми та теплові карти, допомагають користувачам зрозуміти свою ефективність та визначити області для вдосконалення.

Персоналізовані профілі – профілі користувачів дозволяють вводити особисту інформацію, таку як вік, стать, вага, зріст та рівень фізичної підготовки. Персоналізовані налаштування забезпечують відстеження активності і постановку цілей з урахуванням індивідуальних потреб і переваг кожного користувача.

Сповідання та нагадування – своєчасні сповідання та нагадування заохочують користувачів залишатися активними та залученими до роботи з додатком. Нагадування про щоденні цілі, майбутні тренування і досягнення мотивують користувачів до постійного виконання вправ.

Функції обміну інформацією в соціальних мережах і спільнотах – інтеграція з платформами соціальних мереж дозволяє користувачам ділитися своїми досягненнями у фізичній активності, оновленнями про прогрес і зведеннями тренувань з друзями і передплатниками. Обмін інформацією в соціальних мережах розвиває почуття спільності та підзвітності, мотивуючи користувачів залишатися активними та надихати інших.

Спеціальні можливості та можливості налаштування – спеціальні можливості, такі як масштабовані шрифти, налаштування кольорового контрасту та голосові команди, забезпечують зручність використання програми користувачами з різними можливостями. Можливості налаштування тим інтерфейсу, переваг макета і перетворення одиниць виміру покращують зручність роботи користувача і враховують індивідуальні переваги.

Конфіденційність та безпека даних – надійні налаштування конфіденційності та захист даних захищають особисту інформацію та дані про діяльність користувачів. Прозора Політика конфіденційності, надійні механізми аутентифікації і шифрування даних вселяють довіру в користувачів, спонукаючи їх взаємодіяти з додатком.

Включивши ці спеціальні функції в графічний інтерфейс програми для відстеження фізичної активності, розробники можуть створити привабливий, орієнтований на користувача інтерфейс, який мотивує користувачів вести більш здоровий і активний спосіб життя.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Розробка мобільного застосунку для відстеження фізичної активності, тобто ТФА є не тільки своєчасним, але й нагальним завданням, враховуючи зростаючу поширеність малорухливого способу життя та пов'язаних з ним ризиків для здоров'я у всьому світі.

Звідси випливає мета даної роботи – програмна реалізація мобільного трекару фізичної активності..

Система має підтримувати:

- відстеження активності:
- моніторинг в режимі реального часу:
- постановка цілей і відстеження прогресу:
- налаштування та персоналізація:
- аналіз даних та аналітична інформація:
- соціальні функції та залучення спільноти:
- гейміфікація та мотивація:
- планування тренувань і рекомендації.

4.2 Вимоги до інтерфейсу

Застосунок має відповідати правилам доступності та зручності для користувачів, а саме:

- забезпечення доступності функцій для користувачів з обмеженими можливостями, таких як голосові команди, зчитувачі з екрана та настроювані розміри шрифту;
- розробка зручного інтерфейсу з інтуїтивно зрозумілою навігацією, чіткими візуальними підказками і адаптивним взаємодією.

4.3 Умови виконання програми

Обов'язкові вимоги.

Для сервера: серверна ОС; підтримка JVM, підтримка СУБД MySQL.

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини застосунку:

- продуктивний хостинг;
- підтримка JVM.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- смартфон під управлінням ОС Android;
- процесор із тактовою частотою від 2,0 GHz;
- підключення до мережі Internet.

4.4 Методика та результати тестування

Під час тестування було обрано стратегію поетапного виконання алгоритму користування програмою.

Тож на рис. 4.1 наведено головну сторінку застосунку.



Рисунок 4.1 – Головний екран застосунку

На рис. 4.2 наведено результати трекінгу кроків впродовж дня.

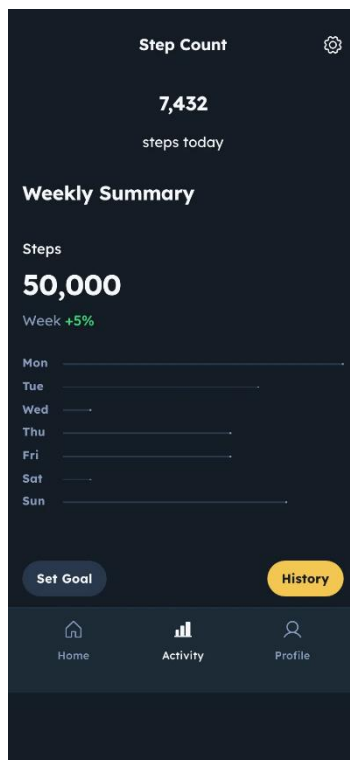


Рисунок 4.2 – Результати трекінгу кроків впродовж дня

На рис. 4.3 наведено налаштування мети – «бажана вага».

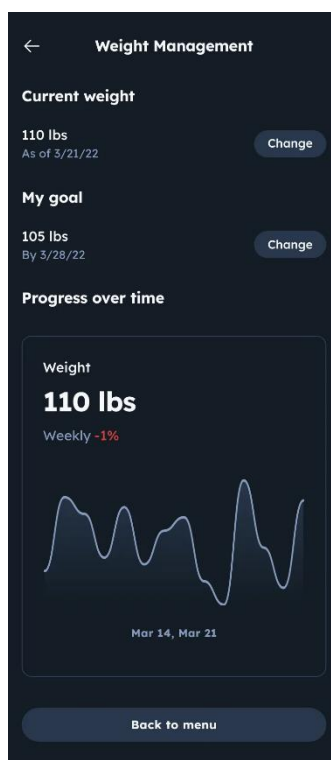


Рисунок 4.3 – Налаштування мети – «бажана вага»

На рис. 4.4 наведено порівняння бажаної ваги та ваги, що є на момент заповнення теркери.



Рисунок 4.4 – Порівняння бажаної ваги та ваги, що є на момент заповнення теркери

На рис. 4.5 наведено додаткові пункти меню, що оформлені у вигляді спливаючого вікна.

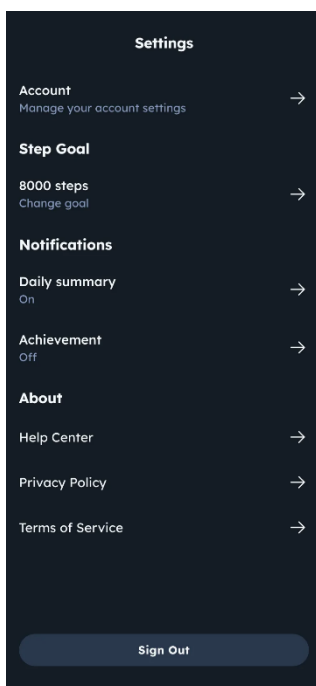


Рисунок 4.5 – Додаткові пункти меню

На рис. 4.6 наведено налаштування застосунку для персоналізації його роботи.

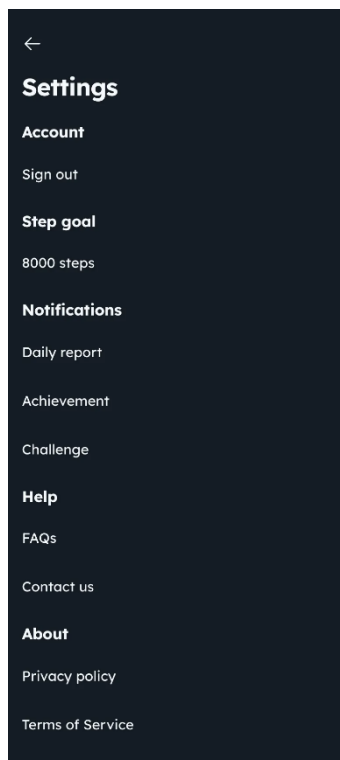


Рисунок 4.6 – Налаштування застосунку

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було досягнуто головну мету – програмно реалізовано мобільний трекер фізичної активності. Для цього було виконано ряд завдань:

- досліджено область трекінгу фізичної активності;
- досліджено питання трекінгу фізичної активності через програмні застосунки;
- досліджено переваги та недоліки існуючих програмних рішень для трекінгу фізичної активності;
- визначено засоби програмної реалізації нового рішення ТФА;
- виконано проектування розробки нового рішення – програмної реалізації ТФА;
- виконано програмну реалізацію нового рішення;
- виконано тестування розробки;
- сформовано висновки на основі виконаної роботи та подальші шляхи покращення розробки.

Розроблений мобільний застосунок був реалізований із використанням мови програмування Java та середовища розробки й фреймворку Android Studio.

Було розроблено сценарій тестування та, відповідно, проведено тестування розробленого мобільного застосунку з використанням цього сценарію на основі логіки взаємодії користувача зі застосунком. Отримані результати засвідчили, що тестування дало позитивний результат.

В результаті розробки реалізовано систему, що відповідає всім вимогам визначеним на початку розробки у вигляді функціональних вимог. Шляхом подальшого покращення розробки є залучення більшої кількості різноманітних сенсорних систем для моніторингу фізичної активності.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Борисевич Л. В. Фізична активність та вільний час у житті студентської молоді [Текст] / Л. В. Борисевич // Реабілітаційні та фізкультурно-рекреаційні аспекти розвитку людини. - 2022. - № 11. - С. 84-89.
2. Самчук О. М. Фізична активність як одна з умов формування здорового способу життя студентської молоді [Текст] / О. М. Самчук, Н. Я. Захожа // Фізичне виховання, спорт і культура здоров'я у сучасному суспільстві. - 2008. - Т. 2. - С. 225–227.
3. Старченко А. Фізична активність, як важливий пріоритет розвитку успішної країни: європейський досвід [Текст] / А. Старченко // Спортивна наука та здоров'я людини. - 2023. - № 1. - С. 139-146.
4. Плошинська А. А. Фізична активність особистості в інформаційному суспільстві: психологічні аспекти державної політики регуляції [Текст] / А. А. Плошинська // Проблеми політичної психології. - 2016. - Вип. 4. - С. 244-252.
5. Чекан О. І. Методика використання інноваційних пристроїв технології ай-трекінгу в роботі з дітьми з особливими освітніми потребами [Текст] / О. І. Чекан, Х. В. Барна, Б. А. Березюк // Інноваційна педагогіка. - 2021. - Вип. 39. - С. 100-105.
6. Голяка Р. Пристрої та моделі магнітного трекінгу для систем доповненої реальності [Текст] / Р. Голяка, Т. Марусенкова, М. Хільчук // Infocommunication technologies and electronic engineering. - 2021. - Vol. 1, № 2. - С. 81-93.
7. Соколюк О. В. Аналіз проблеми моніторингу розвитку оздоровчої рухової активності майбутніх учителів фізичного виховання [Текст] / О. В. Соколюк // Проблеми інженерно-педагогічної освіти. - 2017. - № 54-55. - С. 367-372.
8. Top 5 Fitness apps – real style / [Electronic resource] – <https://hayliepomroy.com/blogs/articles/top-5-fitness-apps-real->

style?tw_source=google&tw_adid=&tw_campaign=17251746453&tw_source=google&tw_adid=&tw_campaign=17251746453&gad_source=1&gclid=Cj0KCQjw_-GxBhC1ARIsADGgDjvDKS6GULfrTBeVEoA5NGz6aIB4gfY02dRaDu7hTSvLim-cOr9BhG0aAgEuEALw_wcB.

9. Fitness tracker app cost estimate / [Electronic resource] – <https://crowdbotics.com/posts/cost-to-build/fitness-tracker-app/>.

10. Best Free Fitness Tracking Apps for Android – 2024 / [Electronic resource] – <https://www.geeksforgeeks.org/free-fitness-tracking-apps-for-android/>

11. Google Fit: трекер активності / [Електронне джерело] – <https://play.google.com/store/apps/details?id=com.google.android.apps.fitness>

12. Google Fit / [Електронне джерело] – <https://www.google.com/fit/>

13. Apple Health / [Electronic resource] – [https://apps.apple.com/app/apple-%D0%B7%D0%B4%D0%BE%D1%80%D0%BE%D0%B2%D1%8C%D0%B5/id1242545199](https://apps.apple.com/app/apple-health/id1242545199)

14. Apple Health review: 50 facts and highlights / [Electronic resource] – <https://versus.com/en/apple-health>

15. Strava / [Electronic resource] – <https://www.strava.com/>

16. Strava: Run, Bike, Hike / [Electronic resource] – <https://play.google.com/store/apps/details?id=com.strava&hl=uk&gl=US>

17. Мороховець М. К. Застосування та розроблення засобів лінгвістичного програмного забезпечення для аналізу та перетворення текстів вимог до програмних систем [Текст] / М. К. Мороховець, Н. М. Щоголева // Кібернетика та системний аналіз. - 2022. - Т. 58, № 5. - С. 16–24.

18. Грицюк Ю.І. Аналіз вимог до програмного забезпечення [Текст] : навч. посіб. / Ю. І. Грицюк ; Нац. ун-т "Львів. політехніка". - Львів : Вид-во Львів. політехніки, 2018. - 453 с.

19. Java Documentation / [Electronic resource]. – <https://docs.oracle.com/en/java/>

20. Learn Java For Android App Development – A Complete Guide / [Electronic resource]. – <https://www.geeksforgeeks.org/learn-java-for-android-app-development-a-complete-guide/>

21. Android Studio / [Electronic resource]. – <https://developer.android.com/studio>

22. Android Studio Tutorial / [Electronic resource]. – <https://www.javatpoint.com/android-tutorial>

23. Unified Modeling Language (UML) Diagrams / [Electronic resource]. – <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>

24. Powerful Diagrams For Every Industry / [Electronic resource] – https://gojs.net/latest/index.html?t=a&gad_source=1&gclid=Cj0KCQjw_-GxBhC1ARIsADGgDjvprJ2h_55QOmFKYanLmrfUN_gJCroPfOHhFd8vwwWKnPtdikVDKywaAjb4EALw_wcB

ДОДАТОК А
Технічне завдання

Вступ

Даний застосунок присвячений для трекінгу (відстеження) фізичної активності користувача на основі даних сенсорних систем його смартфона

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація мобільного трекера фізичної активності», затверджене наказом Національного університету «Запорізька політехніка» № 168 від 24 квітня 2024 р.

A.2 Основні вимоги до програми, що розробляється

Розробка мобільного застосунку для відстеження фізичної активності є актуальним завданням, враховуючи глобальну поширеність малорухливого способу життя і пов'язані з цим ризики для здоров'я. Ці програми мають численні переваги у підвищенні фізичної активності та загального самопочуття, включаючи моніторинг поведінки та зворотний зв'язок, встановлення цілей та підзвітність, персоналізацію та адаптивність, інтеграцію з мобільними пристроями, соціальну підтримку та залучення громади, освітні ресурси та методи зміни поведінки, відстеження здоров'я та аналітичну інформацію, віддалену доступність та гнучкість та охорону здоров'я інтеграція та лікування захворювань. Використовуючи силу сучасних технологій, програми ТФА можуть допомогти людям вести здоровий та активний спосіб життя та зменшити тягар хронічних захворювань для суспільства.

A.2.1 Вимоги до функціональних характеристик

Система має реалізовувати наступні функціональні вимоги:

а) відстеження активності:

- запис і відстеження різних фізичних навантажень, таких як ходьба, біг, їзда на велосипеді та інші вправи;
- збір даних про показники, включаючи кроки, відстань, тривалість, кількість спалених калорій та кількість активних хвилин;

б) моніторинг в режимі реального часу:

- забезпечення моніторингу фізичної активності в режимі реального часу за допомогою датчиків смартфонів;
- відображення поточного оновлення показників активності під час тренувань;

в) постановка цілей і відстеження прогресу:

- дозволяє користувачам встановлювати індивідуальні цілі активності на основі таких показників, як кроки, відстань або активні хвилини;
- відстеження та візуалізація прогресу у досягненні цілей за допомогою підсумків, діаграм або показників прогресу;

г) налаштування та персоналізація:

- надання можливості користувачам налаштовувати цілі своєї діяльності, переваги і налаштування повідомлень;
- надання персоналізованих рекомендацій та аналітичних матеріалів на основі даних користувачів та уподобань;

д) аналіз даних та аналітична інформація:

- аналіз даних про активність, щоб отримати уявлення про тенденції, закономірності та взаємозв'язки;
- надання практичних рекомендацій щодо підвищення рівня активності та досягнення цілей у фітнесі.

е) соціальні функції та залучення спільноти:

- використання функцій соціальних мереж, такі як спілкування з друзями, групові завдання та таблиці лідерів;

- надання користувачам можливості ділитися своїми досягненнями, тренуваннями та прогресом з друзями та послідовниками;

ж) гейміфікація та мотивація:

- використання елементів гейміфікації, такі як значки, нагороди та віртуальні завдання, щоб мотивувати користувачів;

- можливість пропонувати стимули для досягнення контрольних показників і підтримки постійного рівня активності;

з) планування тренувань і рекомендації:

- надання інструментів для планування тренувань, програми вправ та тренінгові програми, адаптовані до цілей користувачів та рівня їх фізичної підготовки;

- пропонування рекомендацій щодо правильної форми, техніки та інтенсивності для різних видів вправ.

A.2.2 Вимоги до інтерфейсу програми

Інтерфейс системи повинен бути ергономічним, інтуїтивно-зрозумілим для користувачів системи. Вся інформація сайту повинна бути чітко структурована і розділена на розділи. Розділ пошуку повинен мати настільки простий інтерфейс, наскільки можливо зробити без втрати його багатофункціональності. Сторінки реєстрації та авторизації користувача повинні мати підказки з приводу заповнення форм.

A.2.3 Вимоги до надійності

Забезпечення конфіденційності та безпеки даних:

- застосування надійних заходів для захисту конфіденційності та безпеки даних користувачів, включаючи шифрування, автентифікацію та безпечну передачу даних.
- дотримання відповідних правил та рекомендацій щодо захисту даних для захисту інформації користувача.

A.2.4 Вимоги до складу та параметрів технічний засобів

Обов'язкові вимоги.

Для сервера: серверна ОС; підтримка JVM, підтримка СУБД MySQL.

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини застосунку:

- продуктивний хостинг;
- підтримка JVM.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- смартфон під управлінням ОС Android;
- процесор із тактовою частотою від 2,0 GHz;
- підключення до мережі Internet.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Розробка мобільного застосунку для відстеження фізичної активності є актуальним завданням, враховуючи глобальну поширеність малорухливого способу життя і пов'язані з цим ризики для здоров'я. Ці програми мають численні переваги у підвищенні фізичної активності та загального самопочуття, включаючи моніторинг поведінки та зворотний зв'язок, встановлення цілей та підзвітність, персоналізацію та адаптивність, інтеграцію з мобільними пристроями, соціальну підтримку та залучення громади, освітні ресурси та методи зміни поведінки, відстеження здоров'я та аналітичну інформацію, віддалену доступність та гнучкість та охорону здоров'я інтеграція та лікування захворювань. Використовуючи силу сучасних технологій, програми ТФА можуть допомогти людям вести здоровий та активний спосіб життя та зменшити тягар хронічних захворювань для суспільства.

Відповідно, можна сформулювати мету даної роботи – розробка мобільного застосунку – трекеру фізичної активності.

Б.2 Функціональне призначення

Система має реалізовувати наступні функціональні вимоги:

а) відстеження активності:

- запис і відстеження різних фізичних навантажень, таких як ходьба, біг, їзда на велосипеді та інші вправи;
- збір даних про показники, включаючи кроки, відстань, тривалість, кількість спалених калорій та кількість активних хвилин;

б) моніторинг в режимі реального часу:

- забезпечення моніторингу фізичної активності в режимі реального часу за допомогою датчиків смартфонів;
- відображення поточного оновлення показників активності під час тренувань;

в) постановка цілей і відстеження прогресу:

- дозволяє користувачам встановлювати індивідуальні цілі активності на основі таких показників, як кроки, відстань або активні хвилини;
- відстеження та візуалізація прогресу у досягненні цілей за допомогою підсумків, діаграм або показників прогресу;

г) налаштування та персоналізація:

- надання можливості користувачам налаштовувати цілі своєї діяльності, переваги і налаштування повідомлень;
- надання персоналізованих рекомендацій та аналітичних матеріалів на основі даних користувачів та уподобань;

д) аналіз даних та аналітична інформація:

- аналіз даних про активність, щоб отримати уявлення про тенденції, закономірності та взаємозв'язки;
- надання практичних рекомендацій щодо підвищення рівня активності та досягнення цілей у фітнесі.

е) соціальні функції та залучення спільноти:

- використання функцій соціальних мереж, такі як спілкування з друзями, групові завдання та таблиці лідерів;
- надання користувачам можливості ділитися своїми досягненнями, тренуваннями та прогресом з друзями та послідовниками;

ж) гейміфікація та мотивація:

- використання елементів гейміфікації, такі як значки, нагороди та віртуальні завдання, щоб мотивувати користувачів;
- можливість пропонувати стимули для досягнення контрольних показників і підтримки постійного рівня активності;

з) планування тренувань і рекомендації:

- надання інструментів для планування тренувань, програми вправ та тренінгові програми, адаптовані до цілей користувачів та рівня їх фізичної підготовки;
- пропонування рекомендацій щодо правильної форми, техніки та інтенсивності для різних видів вправ.

Б.3 Опис логічної структури

Рівень UI:

- початковий екран – відображає огляд щоденної активності користувача, включаючи пройдені кроки, пройденої відстань, спалені калорії та хвилини активного відпочинку;
- відстеження активності – дозволяє користувачам запускати, призупиняти та припиняти відстеження різних фізичних навантажень, таких як ходьба, біг, їзда на велосипеді та інші вправи;
- цілі та прогрес – надає функції для встановлення цілей активності, відстеження прогресу та візуалізації досягнень за допомогою графіків та показників прогресу;
- історія та аналіз – дозволяє користувачам переглядати дані про діяльність за минулі періоди, аналізувати тенденції та отримувати інформацію про свою діяльність та ефективність з часом;
- профіль та налаштування – дозволяє користувачам керувати інформацією про свій профіль, налаштовувати параметри програми та такі параметри, як одиниці виміру, сповіщення та параметри конфіденційності.

Рівень управління даними:

- дані про діяльність – зберігає створені користувачем дані про діяльність, включаючи кроки, відстань, тривалість, спалені калорії та тип діяльності;

- дані про цілі – зберігає задані користувачем цілі активності, включаючи кількість пройдених кроків, дистанцію, тривалість і кількість спалених калорій;

- профіль користувача – зберігає інформацію про користувача, таку як ім'я, вік, стать, вага, зріст та рівень фізичної підготовки;

- налаштування та налаштування – зберігає налаштування користувача та програми, включаючи одиниці виміру (наприклад, метричні чи імперські), налаштування сповіщень та налаштування конфіденційності;

- історичні дані – зберігає історичні дані про діяльність для аналізу, визначення тенденцій та звітування.

Рівень бізнес-логіки:

- логіка відстеження активності – реалізує алгоритми і логіку для відстеження різних фізичних навантажень в режимі реального часу, розрахунку таких показників, як кроки, відстань і спалені калорії, а також обробки взаємодій з користувачами під час сеансів відстеження активності;

- логіка управління цілями – керує цілями діяльності, визначеними Користувачем, відстежує прогрес у досягненні цілей і генерує повідомлення або нагадування, що спонукають користувачів до досягнення своїх цілей;

- логіка аналізу даних – аналізує дані про діяльність, виявляє тенденції та закономірності, генерує ідеї та рекомендації щодо підвищення рівня активності та надає зворотний зв'язок користувачам на основі їх результатів;

- аутентифікація та авторизація користувачів – реалізує механізми аутентифікації, авторизації та контролю доступу для забезпечення безпечного доступу до даних та функцій користувача.

Рівень інтеграції:

- інтеграція служб локації – використовує GPS та служби локації для відстеження заходів на свіжому повітрі, таких як біг, їзда на велосипеді та піші прогулянки, а також для обчислення таких показників, як пройдена відстань та відображення маршруту;

- інтеграція сторонніх API – інтегрується зі сторонніми API для доступу до додаткових джерел даних, таких як інформація про погоду, бази даних про харчування або показники здоров'я, для покращення функціональності програми та зручності використання;

- інтеграція соціальних медіа – дозволяє користувачам ділитися своїми досягненнями у фізичній активності, оновленнями прогресу та підсумками тренувань з друзями та послідовниками на платформах соціальних медіа, сприяючи соціальній взаємодії та мотивації.

Рівень презентації:

- адаптивний користувацький інтерфейс – дані відстеження активності, цілі, прогрес і аналіз представлені в візуально привабливою і зручною для користувача формі, оптимізованої для мобільних пристроїв з різними розмірами екранів і дозволами;

- інтерактивні компоненти – надає інтерактивні елементи, такі як кнопки, повзунки, прапорці та випадаючі списки, щоб полегшити введення користувачем, навігацію та взаємодію з функціями програми;

- зворотній зв'язок і повідомлення – відображає повідомлення зворотного зв'язку, оповіщення, нотифікації і нагадування для доведення важливої інформації до користувачів, надання відгуків про ефективність їх дій і спонукання їх до прийняття необхідних заходів або внесення змін.

На цій структурній схемі представлені основні компоненти та функціональні можливості мобільного застосунки для відстеження фізичної активності, розділені на окремі рівні для полегшення розробки, обслуговування та масштабування. Кожен рівень включає в себе певні обов'язки і взаємодіє з іншими рівнями для забезпечення безперебійної взаємодії з користувачем.

Б.4 Використані технічні засоби

Обов'язкові вимоги.

Для сервера: серверна ОС; підтримка JVM, підтримка СУБД MySQL.

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини застосунку:

- продуктивний хостинг;
- підтримка JVM.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- смартфон під управлінням ОС Android;
- процесор із тактовою частотою від 2,0 GHz;
- підключення до мережі Internet.

ДОДАТОК В
Текст програми

```

public class MainActY extends ActY {

    //Declare the Btn objects.
    Btn recBtn;
    Btn settingsBtn;

    //Declare variable to hold HZ settings.
    int samplingRateHZ = 60;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ActY_main);

        //Define where the Btn objects are.
        recBtn = (Btn) findViewById(R.id.recBtn);
        settingsBtn = (Btn)
findViewById(id.settingsBtn);

        //Handle the actual Btn presses and take
the user to the new action.

        //rec Btn
        final Intent intentrecBtn = new Intent();
        intentrecBtn.setClass(MainActY.this,
recActY.class);

        recBtn.setOnClickListener(new OnClickListener()
        {

```

```

        public void onClick(Vw v) {
startActY(intentrecBtn);}

        });

        //Settings Btn
        final Intent intentSettingsBtn = new
Intent();

        intentSettingsBtn.setClass(MainActY.this,
SettingsActY.class);

        settingsBtn.setOnClickListener(new
OnClickListener()
        {
            public void onClick(Vw v) {
startActY(intentSettingsBtn);}

            });

    }

    @Override
    public boolean onCrtOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

```

```

    }

    public class recActY extends ActY implements
SensorEventListener {

        private SensorMang sensMang;
        private Vw Vw;
        private long lastUpD;

        //Setup file stream.
        FileOutputStream outputFile;

        //Declare Btn Objects
        Btn stopRecBtn;

        int refreshRate = 500;

        @Override
        public void onCrt(Bundle savedInstStt) {
            super.onCrt(savedInstStt);
            setContVw(R.lyut.ActY_rec);

            sensMang = (SensorMang)
getSystemService(SENSOR_SERVICE);
            lastUpD = System.currentTimeMillis();

            //Setup the Btns
            final Intent intentStopRecBtn = new
Intent();

            intentStopRecBtn.setClass(recActY.this,
MainActY.class);

```

```

        stopRecBtn.setOnClickListener(new
OnClickListener()
        {
            public void onClick(Vw v) {
                endRec();
                startActY(intentStopRecBtn);
            }
        });
    }

    @Override
    public boolean onCrtOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
action bar if it is present.
        getMenuInflater().inflate(R.menu.rec, menu);

        return true;
    }

    @Override
    public void onSensorChng(SensorEvent event) {
        if (event.sens.getType() ==
Sensor.TYPE_ACCELEROMETER) {
            getAccelerometer(event);
        }
    }
}

```

```

private void getAccelerometer(SensorEvent event)
{
    float[] Vals = event.Vals;
    // Movement
    float x = Vals[0];
    float y = Vals[1];
    float z = Vals[2];

    long actualTime =
System.currentTimeMillis();

    if ((actualTime - lastUpD < refreshRate))
    {
        return;
    }

    //UpD the fields if the above doesn't dump
out.

    lastUpD = actualTime;
    UpDXYZFields(x,y,z);

    return;
}

public void UpDXYZFields(float x, float y, float
z) {
    txtVw xValtxt = (txtVw)
findVwById(R.id.txtXVal);

```

```

        txtVw        yValtxt        =        (txtVw)
findVwById(R.id.txtYVal);
        txtVw        zValtxt        =        (txtVw)
findVwById(R.id.txtZVal);

```

```

        //Convert the Vals pass this cont to str
that so of the they can be set to the txtFields.

```

```

        String xStringVal = Float.toString(x);
        String yStringVal = Float.toString(y);
        String zStringVal = Float.toString(z);

```

```

        //Set Vals to the txt fields.
xValtxt.settxt(xStringVal);
yValtxt.settxt(yStringVal);
zValtxt.settxt(zStringVal);

```

```

        //Let us go ahead and try to write this
data to a file on the device.

```

```

        //String FILENAME = "XYZ_output.txt";

```

```

    }

```

```

    @Override

```

```

    public void onAccChng(Sensor sens, int Acc) {
    }

```

```

    public void endRec(){

```

```

        //Put some code to give it right start pos
to XYZ pom.

```

```

    }
}

public class SettingsActY extends ActY {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.lyut.ActY_settings);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
action bar if it is present.
        getMenuInflater().inflate(R.menu.settings,
menu);

        return true;
    }
}

```

ДОДАТОК Д
Слайди презентації

Програмна реалізація мобільного трекеру фізичної активності

Виконав

студ. КНТ-121сп
ДРОБІНЯК Микола Васильович

Керівник

д-р техн. наук, професор
ОЛІЙНИК Андрій Олександрович

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес програмної реалізації мобільного трекеру фізичної активності.

Предмет дослідження – програмні засоби для реалізації мобільного трекеру фізичної активності.

Метою роботи є програмна реалізація мобільного трекеру фізичної активності.

Рисунок Д.2 – Слайд 2

Аналіз предметної області. Порівняння існуючих аналогів

Критерії порівняння	Google Fit	Apple Health	Strava	MyTrailFit
Платформи	Android, iOS	iOS	Android, iOS	Android
Типи активності	Ходьба, біг, їзда на велосипеді та інші види діяльності	Ходьба, біг, їзда на велосипеді та інші види діяльності	Біг, їзда на велосипеді, плавання та інші види діяльності	Ходьба, біг, їзда на велосипеді та інші види діяльності
Методи відстеження	Датчики смартфона	Датчики смартфона, Apple Watch	GPS (для активних занять), датчики смартфонів	GPS (для активних занять), датчики смартфонів
Інтегрування	Інтеграція з іншими застосунками і пристроями для здоров'я та фітнесу	Інтегрується з іншими застосунками та пристроями для здоров'я та фітнесу	Інтегрується з деякими фітнес-пристроями, сторонніми застосунками	Інтеграція з іншими застосунками
Соціальні функції	Обмежені соціальні функції	Обмежені соціальні функції	Широкі соціальні можливості, клуби, змагання	Обмежені соціальні функції
Постановка цілей	Так	Так	Так	Так
Аналіз даних	Базова аналітика та аналітичні матеріали	Базова аналітика та аналітичні матеріали	Розширена аналітика продуктивності, таблиці лідерів сегментів	Розширена аналітика продуктивності, таблиці тренінгу
Спеціалізація	Загальне відстеження активності	Загальне відстеження активності	Фокус на бігових та велосипедних тренуваннях	Загальне відстеження активності

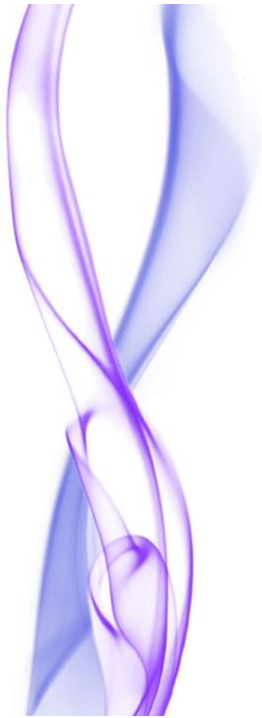
Рисунок Д.3 – Слайд 3

Аналіз предметної області. Постановка завдання

Розробка мобільного застосунку для відстеження фізичної активності є актуальним завданням, враховуючи глобальну поширеність малорухливого способу життя і пов'язані з цим ризики для здоров'я. Ці програми мають численні переваги у підвищенні фізичної активності та загального самопочуття, включаючи моніторинг поведінки та зворотний зв'язок, встановлення цілей та підзвітність, персоналізацію та адаптивність, інтеграцію з мобільними пристроями, соціальну підтримку та залучення громади, освітні ресурси та методи зміни поведінки, відстеження здоров'я та аналітичну інформацію, віддалену доступність та гнучкість та охорону здоров'я інтеграція та лікування захворювань. Використовуючи силу сучасних технологій, програми ТФА можуть допомогти людям вести здоровий та активний спосіб життя та зменшити тягар хронічних захворювань для суспільства.

Відповідно, можна сформулювати мету даної роботи – розробка мобільного застосунку – трекару фізичної активності.

Рисунок Д.4 – Слайд 4



Аналіз предметної області.

Функціональні вимоги

Система має підтримувати:

- відстеження активності:
- моніторинг в режимі реального часу:
- постановка цілей і відстеження прогресу:
- налаштування та персоналізація:
- аналіз даних та аналітична інформація:
- соціальні функції та залучення спільноти:
- гейміфікація та мотивація:
- планування тренувань і рекомендації.

Рисунок Д.5 – Слайд 5

Матеріали та методи.

Вибір мови програмування

Критерій	Java	Kotlin	Python
Синтаксис	Знайомий синтаксис із крапками з комою та фігурними дужками	Сучасний синтаксис з лаконічними і виразними функціями	Лаконічний синтаксис зі зрозумілим і читабельним кодом
Нульова безпека	Відсутність вбудованих функцій захисту від обнулення	Забезпечує вбудований захист від null з використанням типів, які можуть бути як нульовими, так і ненульовими	Відсутній вбудований захист від null, але може бути реалізована за допомогою додаткових типів
Сумісність	Відмінна сумісність з існуючим кодом Java	Повна взаємодія з кодом Java, підтримує проекти на різних мовах	Обмежена сумісність з Java, вимагає додаткової настройки та інтеграції
Мовні особливості	Розвинена мова з великими можливостями і бібліотеками	Сучасна мова з такими функціями, як спільні програми, функції розширення та захист від null	Багата екосистема з бібліотеками для різних доменів, але менше функцій, специфічних для мобільних пристроїв
Уявлення	Висока продуктивність завдяки ефективному часу виконання та оптимізації	Продуктивність, порівнянна з Java, завдяки оптимізації функцій, специфічних для Kotlin	Як правило, нижча продуктивність порівняно з Java та Kotlin через інтерпретацію або компіляцію в байт-код
Етап навчання	Новачкам легше влаштуватися, особливо з об'єктно-орієнтованими концепціями	Більш плавний процес навчання, особливо для розробників, знайомих з Java	Відносно проста в освоєнні, особливо для розробників з досвідом роботи на Python
Інструменти та підтримка	Розвинений інструментарій, велика документація та підтримка спільноти	Зростаюча екосистема за офіційної підтримки Google і успішне впровадження спільнотою	Обмежений набір інструментів та підтримка спільноти розробників Android порівняно з Java та Kotlin
Придатність	Добре підходить для створення надійних і масштабованих додатків для Android	Ідеально підходить для сучасної розробки на Android, пропонуючи розширені можливості і продуктивність	Підходить для конкретних випадків використання, таких як прототипування, аналіз даних або нішеві програми
Спільнота та ресурси	Велика та активна спільнота, великі ресурси та канали підтримки	Зростаюча спільнота зі зростаючими ресурсами та підтримкою, яку підтримують JetBrains та Google	Менша спільнота порівняно з Java та Kotlin, обмежені ресурси та підтримка розробки Android

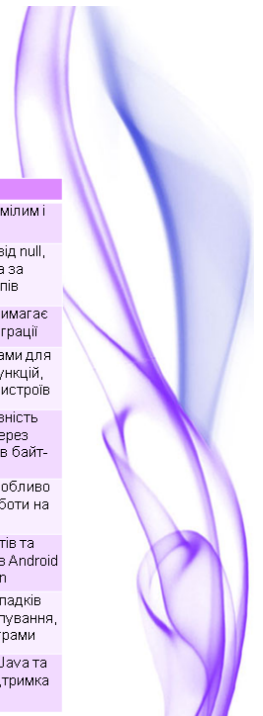


Рисунок Д.6 – Слайд 6

Матеріали та методи.

Вибір середовища розробки

Критерій	Android Studio	IntelliJ IDEA	Visual Studio Code
Платформа	Специфічний для Android	Java IDE загального призначення	Універсальний редактор коду з розширеннями для розробки на Java і Android
Спеціально розроблений	Так	Ні	Ні
Інтеграція з Android SDK	Повна інтеграція з Android SDK	Повна інтеграція з Android SDK	Розширення для розробки на Android не потрібні
Дизайн інтерфейсу користувача	Редактор макетів для візуального проектування інтерфейсу користувача	Редактор макетів для візуального проектування інтерфейсу користувача	Обмежені можливості візуального проектування користувацького інтерфейсу
Профілювання продуктивності	Вбудовані засоби профілювання продуктивності	Вбудовані засоби профілювання продуктивності	Потрібні розширення для профілювання продуктивності
Завершення коду	Комплексне доопрацювання коду	Комплексне доопрацювання коду	Основне завершення коду за допомогою розширень
Рефакторинг	Повний набір інструментів рефакторингу	Повний набір інструментів рефакторингу	Обмежені можливості рефакторингу за допомогою розширень
Налагодження	Розширені можливості налагодження	Розширені можливості налагодження	Базові можливості налагодження за допомогою розширень
Тестування	Інтегровані інструменти тестування та фреймворки	Інтегровані інструменти тестування та фреймворки	Потрібні розширення для тестування
Управління версіями	Вбудована підтримка контролю версій	Вбудована підтримка контролю версій	Інтегрована підтримка контролю версій з розширеннями
Споживання ресурсів	Ресурсомістке завдяки багатству функціональних можливостей	Ресурсомістке завдяки багатству функціональних можливостей	Легка вага, мінімальне споживання ресурсів
Налаштованість	Легко налаштовується за допомогою тим, плагінів і шаблонів	Легко налаштовується за допомогою тим, плагінів і шаблонів	Легко налаштовується за допомогою розширень, тим і налаштувань

Рисунок Д.7 – Слайд 7

Матеріали та методи.

Схема системи

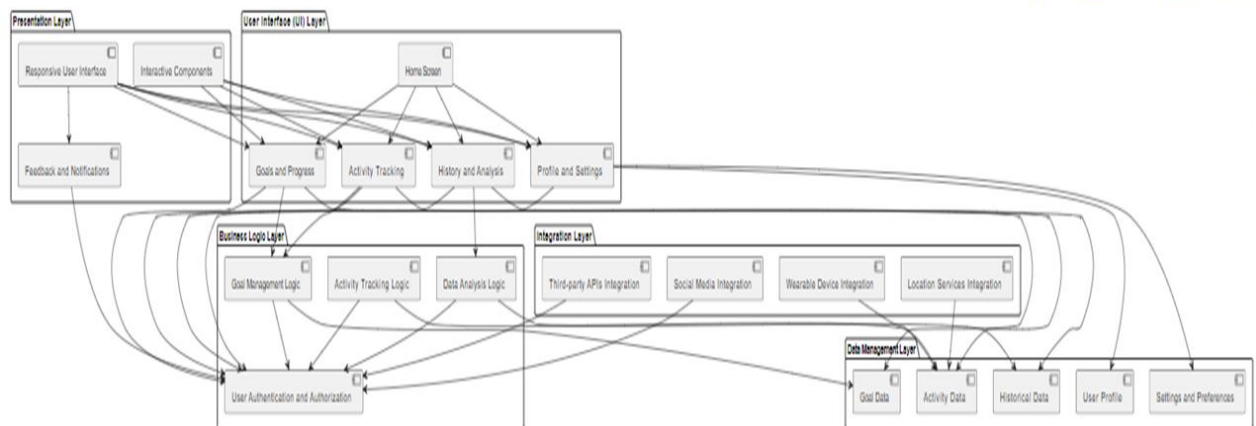


Рисунок Д.8 – Слайд 8

Основні рішення щодо реалізації системи.

Фізичне проектування

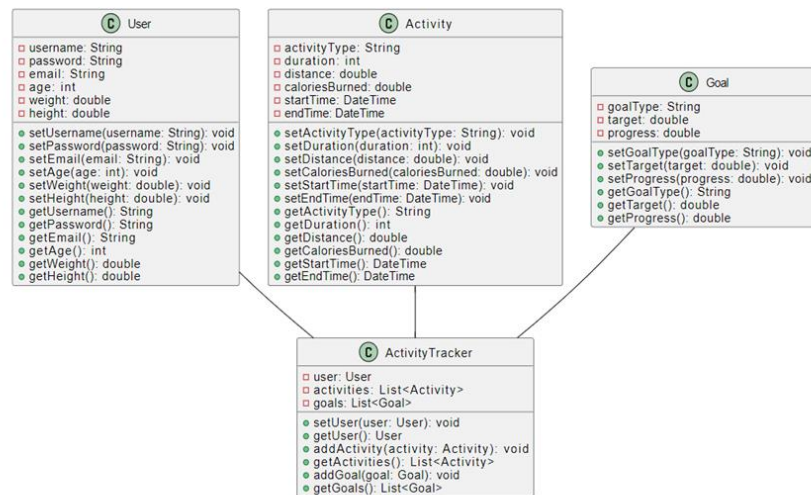


Рисунок Д.9 – Слайд 9

Основні рішення щодо реалізації системи.

Логічне проектування

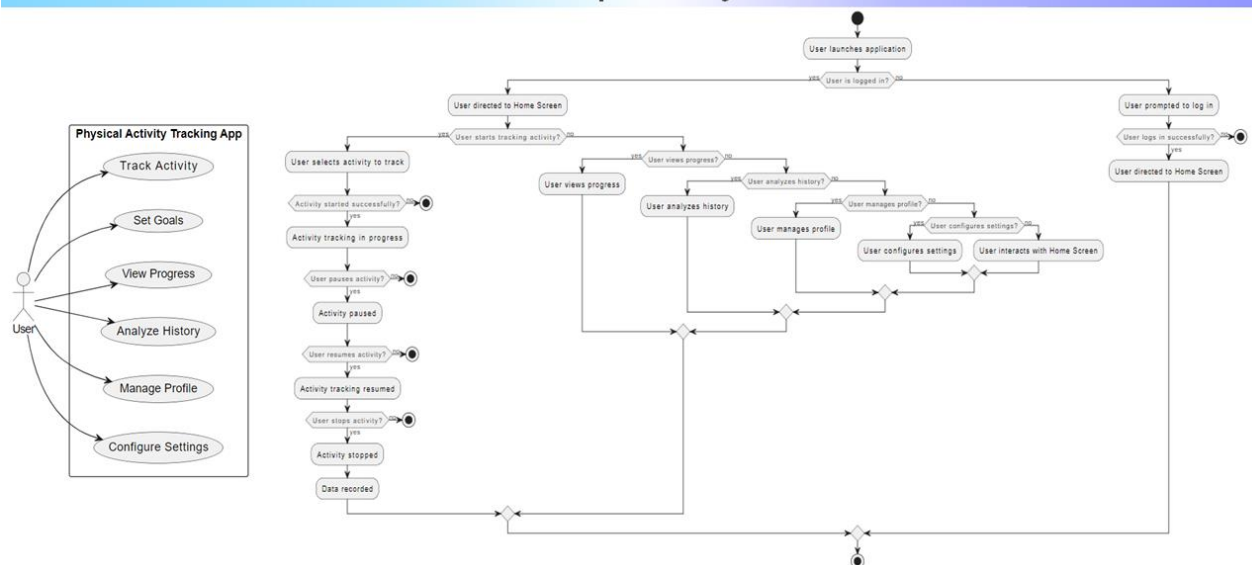


Рисунок Д.10 – Слайд 10

Експлуатація, тестування та експериментальне дослідження програми

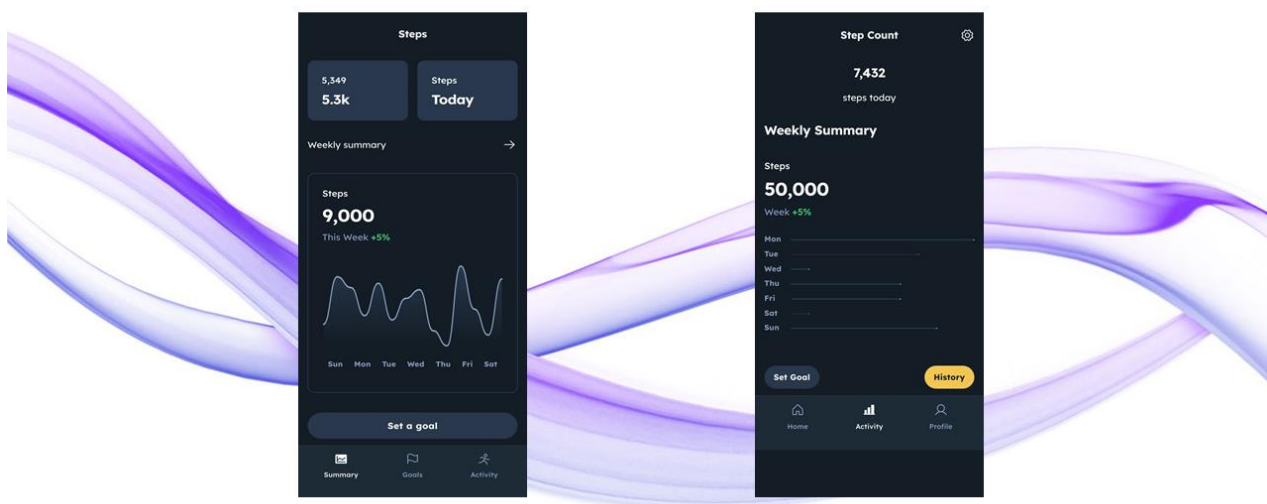


Рисунок Д.11 – Слайд 11

Експлуатація, тестування та експериментальне дослідження програми

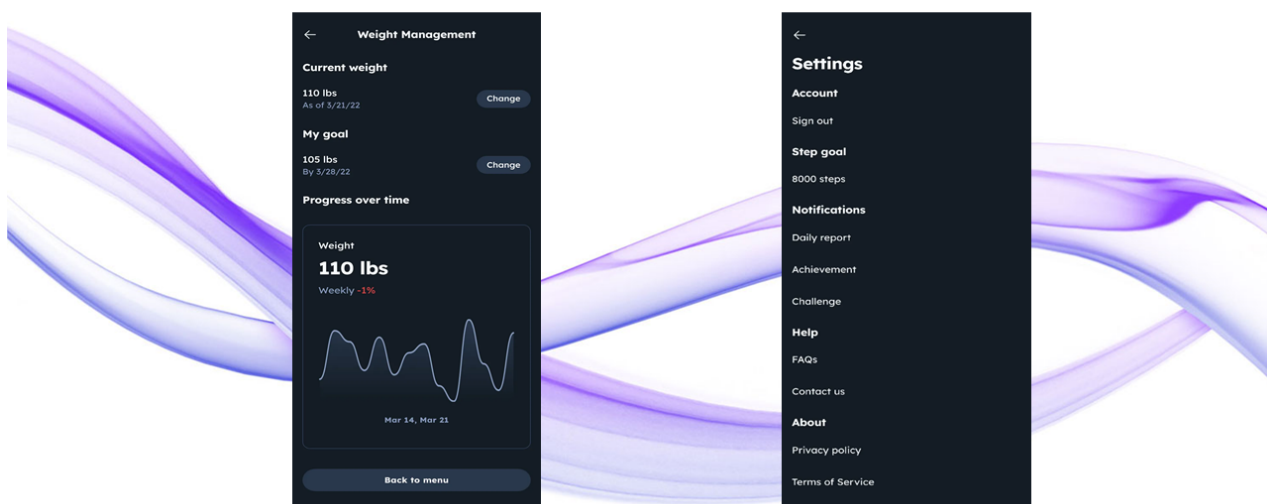


Рисунок Д.12 – Слайд 12



Висновки

В ході виконання роботи було досягнуто головну мету – програмно реалізовано мобільний трекер фізичної активності.

Для цього було виконано ряд завдань:

- досліджено область трекінгу фізичної активності;
- досліджено питання трекінгу фізичної активності через програмні застосунки;
- досліджено переваги та недоліки існуючих програмних рішень для трекінгу фізичної активності;
- визначено засоби програмної реалізації нового рішення ТФА;
- виконано проєктування розробки нового рішення – програмної реалізації ТФА;
- виконано програмну реалізацію нового рішення;
- виконано тестування розробки;
- сформовано висновки на основі виконаної роботи та подальші шляхи покращення розробки.

В результаті розробки реалізовано систему, що відповідає всім вимогам визначеним на початку розробки у вигляді функціональних вимог.

Шляхом подальшого покращення розробки є залучення більшої кількості різноманітних сенсорних систем для моніторингу фізичної активності.

Усі завдання роботи повністю виконано.

Рисунок Д.13 – Слайд 13