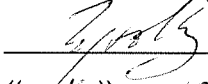


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ У МЕДИЦИНІ ТА ТЕЛЕКОМУНІКАЦІЯХ

УДК 621.37

«Допущено до захисту»

Завідувач кафедри
комп'ютерних технологій у
медицині та телекомунікаціях
к.т.н., доц. Чухов В.В.


« 19 » 12 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему Дослідження можливості створення засобу відстеження фізичної
активності людини

за спеціальністю 163 «Біомедична інженерія»
освітній ступінь «магістр»

Виконавець роботи
студент 2 курсу, гр.БІм-22-1


19.12.23
(дата, підпис)

Власюк В.Р.

Керівник роботи
к.т.н., доц.


19.12.23
(дата, підпис)

Чухов В.В.

Житомир
2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ У МЕДИЦИНІ ТА ТЕЛЕКОМУНІКАЦІЯХ

Пояснювальна записка
до кваліфікаційної магістерської роботи

на тему Дослідження можливості створення засобу відстеження фізичної
активності людини

Виконав: студент 2 курсу,
групи БІм-22-1 спеціальності
163 «Біомедична інженерія»

Власюк В.Р.

Керівник Чухов В.В.

Рецензент Нікітчук Т.М.

Житомир
2023

«Затверджено»

Завідувач кафедри комп'ютерних
технологій у медицині та
телекомунікаціях

Чухов Владислав ЧУХОВ
« 27 » 10 2022 р.

Завдання на кваліфікаційну магістерську роботу

- Власюку Владиславу Руслановичу
(прізвище, ім'я, по батькові)
1. Тема роботи Дослідження можливості створення засобу відстеження фізичної активності людини
затверджено наказом по університету від « 22 » 11 2022 р. № 680
 2. Мета дослідження дослідження можливості створення засобу реєстрації фізичної активності людини
 3. Вихідні дані (характеристики об'єкта, умов дослідження та ін.
додатки Samsung Health, Xiaomi Mi Band 2
 4. Основні задачі дослідження наявні способи відстеження та критерії описання фізичної активності
 5. Термін подання роботи до захисту грудень 2023 р.
 6. Дата видачі завдання « 27 » 10 2022 р.

Науковий керівник

Завдання прийняв
до виконання

Чухов
(підпис)
Власюк
(підпис)

Чухов В.В.
(прізвище, ініціали)
Власюк
(прізвище, ініціали)

АНОТАЦІЯ

На даний момент існує величезна кількість спортивних браслетів, годинників, що відстежують ходьбу, біг, однак рішень, що здійснюють відстеження за допомогою смартфона досить мало. Використання вбудованого акселерометру надає можливість точно відстежувати свою активність без використання сторонніх пристроїв.

В роботі проведено дослідження можливості створення засобу реєстрації фізичної активності людини

ABSTRACT

At the moment, there are a huge number of sports bracelets, watches that track walking, running, but there are quite a few solutions that track using a smartphone. Using the built-in accelerometer allows you to accurately track your activity without using third-party devices.

In the work, a study of the possibility of creating a means of registering a person's physical activity was carried out

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	8
1.1. Загальні відомості про крокомір у смартфоні.....	8
1.2. Огляд існуючих рішень	10
1.2.1. Samsung Health	10
1.2.2. Google Fit.....	12
1.2.3. Runkeeper.....	15
1.2.4. Pedometer - Step Counter by ITO Technologies, Inc.....	17
ВИСНОВОК	20
РОЗДІЛ 2 ПРОЕКТУВАННЯ ДОДАТКУ	22
2.1. Вибір технологій та їх обґрунтування	22
2.1.1. Вибір мінімальної версії системи для розроблюваного додатку ...	24
2.1.2. Вибір мови програмування	26
2.1.3. Вибір допоміжних бібліотек	27
2.1.4. Вибір архітектури додатку	29
2.2. Розробка алгоритму відстеження кроків	31
ВИСНОВОК	36
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ	37
3.1. Реалізація розробленого алгоритму	37
3.2. Реалізація графічного інтерфейсу	41
3.3. Тестування розробленого алгоритму	45
ВИСНОВОК	488
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50

ВСТУП

За останні роки особливої популярності набувають спеціалізовані гаджети для відстеження бігу, ходьби. Відстеження активності у кроках як показало дослідження Стенфордського Університету [2], що опубліковано у «The Journal of American Medical Association», виявив, що люди, які використовують крокоміри, щодня проходять приблизно на 2000 більше кроків, ніж люди, які не користуються крокоміром. Як виявляється, спеціалізовані спортивні браслети використовують звичайний акселерометр, який наявний у будь-якому смартфоні чи планшеті, для визначення типу активності та кількості кроків. Звісно, великі корпорації не створюють мобільні додатки з використанням цього сенсору смартфона, оскільки вони зацікавлені у додатковому продажу спортивних браслетів та аксесуарів для них.

Крокомір, що доступний у кишені смартфона без додаткових аксесуарів (таких як спортивний браслет), - це потужний мотиватор, що сприяє, по-перше, вийти на незаплановану прогулянку парком, оскільки ціль за кроками ще не досягнута. По-друге, крокомір – це помічник у втраті ваги. Дослідження також показали, що носіння крокометра допомагає знизити вагу, оскільки, активність може допомогти схуднути, а також підтримувати ціль по вазі.

Заняття регулярною щоденною активністю призводить до багатьох переваг для здоров'я, таких як зняття стресу, зниження артеріального тиску, контроль рівня цукру в крові, поліпшення рівня холестерину та щільності кісток, зниження індексу маси тіла, менший ризик раку, діабету та серцевих захворювань. Крім того, це може внести величезний внесок у довголіття [3].

Крокоміри надзвичайно корисні для забезпечення об'єктивного та надійного вимірювання вашої щоденної активності. Вам може здатися, що ви робите достатньо, але без крокоміра це лише здогадки. За допомогою крокоміра у смартфоні, користувач може швидко побачити, коли він не досяг своєї щоденної мети. Точне формування представлення користувача з кількістю пройдених кроків та кількістю «спалених» калорій, крокоміри можуть

забезпечити додатковий стимул для досягнення вашої щоденної мети.

Отже, метою цієї кваліфікаційної роботи є дослідження можливості створення засобу реєстрації фізичної активності людини, що без участі додаткових аксесуарів, спортивних браслетів, годинників, буде відстежувати кроки користувача, визначати їх тип (ходьба, біг, біг підтюбцем), кількість пройдених кілометрів та «спалених» калорій, швидкість, час.

Актуальність теми. Збільшення кількості носимих пристроїв що відстежують активність власника та тема власного здоров'я зіграли велику роль у популяризації активного способу життя, однак відстежувати свою активність можна не лише за допомогою спеціалізованих спортивних браслетів. Смартфон, що знаходиться у кожного з нас у кишені, справляється із цією задачею не гірше, ніж спеціалізовані пристрої.

На даний момент існує величезна кількість спортивних браслетів, годинників, що відстежують ходьбу, біг, однак рішень, що здійснюють відстеження за допомогою смартфона досить мало. Використання вбудованого акселерометру надає можливість точно відстежувати свою активність без використання сторонніх пристроїв.

Мета і задачі дослідження. Метою роботи є дослідження можливості створення засобу реєстрації фізичної активності людини.

Для досягнення поставленої мети, необхідно виконати наступні завдання:

- Проаналізувати наявні способи відстеження фізичної активності;
- Сформулювати критерії, що описують тип фізичної активності;
- Розробити програмний продукт за способом у вигляді додатку на смартфон.

Об'єкт дослідження. Програмний продукт у вигляді додатку на смартфон, що здійснює відстеження активності власника.

Предмет дослідження. Спосіб відстеження фізичної активності за допомогою вбудованих сенсорів смартфона.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1. Загальні відомості про крокомір у смартфоні

Смартфони з акселерометром можуть працювати як крокомір для підрахунку кількості кроків, які користувач робить під час ходьби, бігу або бігу підтюбцем.

Переваги крокоміра у смартфоні. Використання крокоміра допомагає відслідковувати кроки, які користувач робить під час занять аеробікою, бігом, ходьбою. Наприклад, якщо користувачу доведеться готуватися до марафону, який проходить за два місяці, він зможе скласти розклад бігу щоранку. Тепер коли користувач щодня може відстежити свій біг, він може побачити чи збільшується витривалість та продуктивність. Якщо користувач відслідковує свою активність за допомогою крокоміру, то він зможе побачити таку інформацію як відстань, яку проходить кожен день, так і кроки та тривалість свого тренування. Крім того, він зможе переглянути свою історію пробіжок чи ходьби через дні та тижні.

Точність підрахунку. Точність підрахунку залежить від того, чи буде користувач використовувати смартфон протягом свого тренування. Так, наприклад, якщо розмовляти по телефону, не використовуючи гарнітуру, та прогулюватися по парку, можливе значне відхилення від реальної кількості кроків. Таке відхилення пояснюється тим, що крокомір використовує акселерометр, а, перебуваючи біля вуха користувача, акселерометр не фіксує жодних змін, оскільки смартфон перебуває у більш-менш стабільному стані. Крім розмов, будь-яке використання смартфону може відобразитися на кроках, що крокомір відстежує. Крім того, прості потрушування смартфону можуть збільшити кількість кроків, знов ж таки за рахунок того, що використовується не спеціалізований датчик.

Крокомір не використовує дані GPS, тому відстеження шляху на карті не передбачено. Однак, перевагою того, що GPS не використовується є факт того, що розхід батареї на багато менший.

Положення смартфону. Для відстеження фізичної активності необхідно

тримати смартфон з користувачем. Для точного підрахунку рекомендується тримати смартфон або у кармані штанів, куртки, чи у сумці. Тримавши смартфон у руці, може сприяти випадковим крокам. Положення смартфона у просторі взагалі не відіграє жодної ролі у підрахунку, - його можна тримати як горизонтально, так і вертикально, догори низом чи навпаки.

Принцип роботи. Як вже було згадано раніше, додатки-крокоміри використовують вбудований акселерометр. Під час роботи додаток буде стежити за змінами значень акселерометра і на основі цих змін буде робити висновок, чи був зроблений крок. Крім того, використовуючи ці значення можна математично обчислити швидкість та відстань з якою користувач пересувається.

Акселерометр надає 3 значення: значення по осі x, y та z. Як тільки користувач зробить свій перший крок, значення по осі x зміниться в залежності від того, з якою силою користувач розпочне своє тренування. Вісь y вказує на відносну висоту смартфона, тобто ту, що вказує на скільки смартфон піднявся угору чи вниз. Відстеживши положення по осі y можна зробити висновок, чи був зроблений крок чи ні. При збільшенні значення по осі y, можна зробити висновок, що користувач виконує крок, а при зменшенні – це є факт того, що крок зроблено.

У останніх версіях мобільної операційної системи Android було додано вбудований сенсор-крокомір, однак він не надає повної інформації про активність користувача протягом тренування. Цей сенсор-крокомір надає лише інформацію про кількість кроків, що були зроблені з моменту запуску смартфона і оновлюються не частіше одного разу на 30 хвилин. До того ж, оскільки операційна система Android є відкритою та смартфони випускаються різними виробниками, цей сенсор може бути програмно відключено з міркувань енергозбереження та мотивації придбання спеціалізованих спортивних аксесуарів, годинників, тощо.

Перевірка точності крокоміра. Для перевірки точності крокоміра достатньо злегка почати натискати на екран смартфона. У випадку якщо кількість кроків почне збільшуватися з великою швидкістю, то це свідчить про

те, що крокомір не досить точний. Чим менше кроків було зафіксовано, тим менше неточність буде при використанні крокоміру у реальному тренуванні.

1.2. Огляд існуючих рішень

Відстеження фізичної активності, а саме відстеження може відбуватися як за допомогою вбудованої у більшість мобільних пристроїв системи глобального позиціонування, так і за допомогою акселерометру. Акселерометр присутній у будь-якому мобільному пристрої і прямою функцією є відстеження орієнтації пристрою у просторі. Однак, його також можна використовувати як засіб для відстеження фізичної активності. Розглянемо найпопулярніші додатки для відстеження фізичної активності, що доступні для завантаження через маркетплейс Google Play.

1.2.1. Samsung Health

Samsung Health – додаток що розповсюджується безкоштовно та розробляється корейською компанією Samsung. Головною задачею цього застосунку є відстеження власного здоров'я для підвищення власного добробуту, включаючи фізичну активність, сон, тощо. Застосунок був представлений у 2012 році, як частина нового бачення у презентації Samsung Galaxy S3.

Головний аспект цього застосунку – підрахунок кроків, - був представлений через рік на презентації Samsung Galaxy S4 2013 року. Встановлення цілей є головною перевагою додатку. За замовчуванням користувачам пропонується проходити 6 тисяч кроків у день, що є рекомендацією Всесвітньої організації охорони здоров'я.

Крім того, застосунок зберігає та оброблює дані про пройдену відстань, загальну кількість спалених калорій, кроків, тощо. Зібрані данні представляються у вигляді графіків, таких як:

- Місячні та денні графіки прогресу;
- Прогрес за день у вигляді гістограми.

Застосунок можна використовувати як автономно, тобто без використання сторонніх пристроїв (носимих), так і з ними, причому носимий пристрій не обов'язково має бути вироблений корейських гігантом електроніки.

За допомогою функції «Будь активнішим», додаток стимулює користувача встати та здійснити фізичну активність, - цю функцію можна відключити. За замовчуванням, 60 хвилин – мінімальний час, який користувач має бути активним у день.

На додаток, застосунок має можливість вручного внесення інформації про сеанс активності (біг, ходьба, тощо), такі як: тривалість, маршрут, відстань, пульс, спалені калорії[5].

Отже, Samsung Health – це потужний додаток для повного моніторингу активності та здоров'я користувача, що пропонує окрім можливості відстеження кроків, ще багато іншого функціоналу.

Переваги:

1. Повний набір інструментів для відстеження здоров'я користувача;
2. Можливість відстеження маршруту користувача за допомогою глобальної системи позиціонування;
3. Можливість автоматичного відстеження кроків, їх типу, відстані, швидкості без глобальної системи позиціонування;
4. Безкоштовні програми для покращення свого фізичного стану, а також мотиваційні програми для збільшення кількості кроків щодня;
5. Можливість планувати тренування за допомогою вбудованого календарю.

Недоліки:

1. Повністю закритий вихідний код;
2. Для роботи крокоміра необхідний обліковий запис;
3. Перенасичений функціонал, що створює дискомфорт при користуванні. Пошук необхідної інформації доступний після довгого блукання у меню та налаштуваннях.

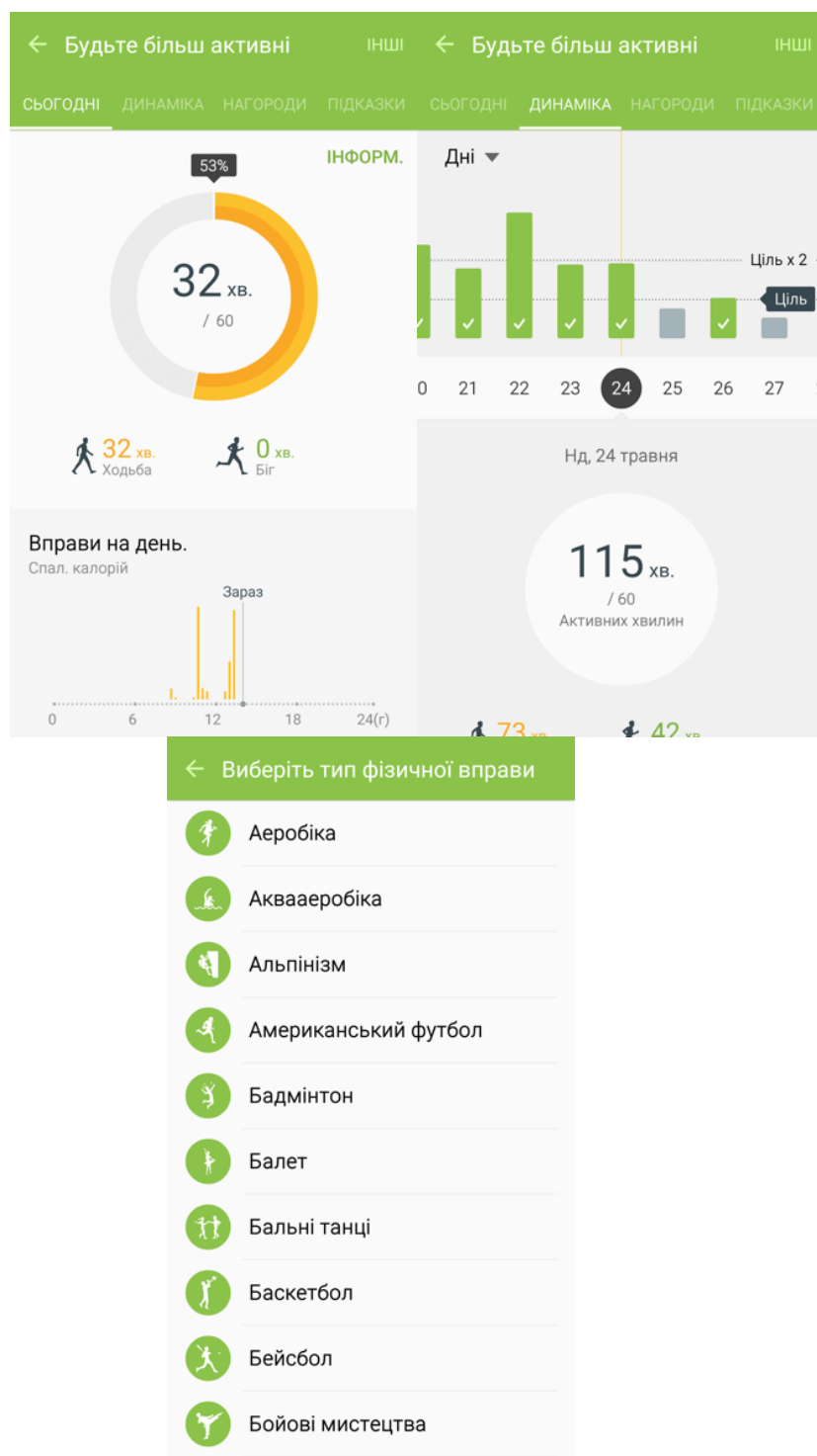


Рис. 1.1. Приклад використання Samsung Health [6]

1.2.2. Google Fit

Google Fit – платформа для відстеження власного здоров'я, що розроблюється американською компанією Google для операційної системи Android. Google Fit – це не лише однойменний застосунок для ОС Android, а й набір інструментів для збору та збереження інформації для подальшого використання за допомогою API. До 2017 року, Google Fit мав веб-версію свого інтерфейсу, однак Google завершила її підтримку. Платформа розроблюється у

партнерстві з провідними спортивними компаніями Nike, HTC, LG, Runkeeper.

Кінцевий користувач може контролювати інформацію про те, хто бачить інформацію про його активність, налаштовувати можливість поширення цієї інформації, тощо.

Платформа Google Fit була представлена у межах конференції розробників Google I/O 14, у межах презентації нової версії ОС Андроїд, як конкурента Health Kit від Apple.

Платформа має величезний набір інструментів для взаємодії з інтерфейсом API, що відкриває можливості для використання отриманої інформації як виробникам смартфонів, так і стороннім розробникам програмного забезпечення. Використання цього API надає доступ до даних про активність від синхронізованих спортивних годинників, фітнес додатків та датчиків, що створює можливість користувачу проводити глибокий аналіз отриманих даних.

Застосунок Google для Android дозволяє відстежувати фізичну активність таку як: біг, ходьба, тощо, як з використанням глобальної системи позиціонування, так і за допомогою сторонніх носимих пристроїв. Партнерами платформи виступили Всесвітня організація здоров'я та Американська асоціація серця) для заохочення своїх користувачів підвищити рівень свого здоров'я. За допомогою додатку можна встановити 2 різні типи цілей: хвилини активності та серцеві бали.

Хвилини активності – тип цілі для користувачів, які не активні. Ця ціль заохочує змінити свій розпорядок дня для покращення свого здоров'я. Так, наприклад, додаток пропонує своїм користувачам використати сходи замість ліфту, здійснити прогулянку замість використання машини. Користувачі, що виконують ці поради набирають так звані «хвилини активності».

Серцеві бали – тип цілі, бали за які нагороджуються лише за більш енергійні вправи, як біг, біг підтюпцем. Біг удвічі більше дає очок, оскільки при цьому типі активності серце працює вдвічі більше, ніж при звичайній прогулянці. Google Fit відслідковує усі ці зміни і при завершенні активності надає точний звіт про виконану фізичну активність аби користувач міг

встановлювати нові цілі для себе [7].

Переваги:

1. Повний набір інструментів для відстеження здоров'я користувача;
2. Можливість автоматичного відстеження кроків, їх типу, відстані, швидкості без глобальної системи позиціонування;
3. Персоналізовані поради для підвищення стану свого здоров'я за допомогою штучного інтелекту;
4. Можливість планувати тренування за допомогою цілей;
5. Можливість підключати будь-які спортивні браслети для передачі інформації у Google Fit;
6. Відкрите API для інших додатків та виробників мобільних смартфонів.

Недоліки:

1. Для роботи крокоміра необхідний обліковий запис;
2. Відсутність персоналізованих тренувань подібних до Samsung Health для покращення здоров'я;
3. Відсутність можливості планування своїх тренувань з використанням календаря;
4. Додаток не має україномовного інтерфейсу.





Рис. 1.2. Приклад використання Google Fit [8]

1.2.3. Runkeeper

Runkeeper - це додаток для відстеження активності за допомогою технології GPS для iOS і Android з більш ніж 50 мільйонами користувачів. Він був запущений в 2008 році генеральним директором Джейсон Джейкобсом і Raizlabs. У лютому 2016 року компанія Runkeeper була придбана компанією ASICS.

Runkeeper дозволяє користувачам відстежувати заняття фітнесом (такі як ходьба, біг і їзда на велосипеді) за допомогою GPS і дозволяє користувачам бачити детальні статистичні дані щодо їх темпу, відстані і часу. Runkeeper відстежує статистику, прогрес, і застосовує коучинг, з вбудованими аудіо сигналами. Користувачі можуть слухати та контролювати музику під час тренування, вимірювати частоту серцевих скорочень.

Програма відстежує продуктивність у реальному часі, дозволяючи користувачам переглядати детальну історію своїх тренувань, отримувати сповіщення про нові особисті "найкращі" та основні етапи, вимірювати прогрес щодо поточних цілей, дотримуватися детальних планів і перетворювати будь-яку діяльність на маршрут, щоб за цим маршрутом можна було здійснити повторні тренування у майбутньому. Runkeeper дозволяє користувачам

ділитися своїм особистим контентом з друзями через розміщення на Facebook і Twitter, а також дозволяє глядачам переглядати живі карти тренувань і гонок. Програма випустила відкритий API для зовнішніх розробників [9].

Переваги:

1. Можливість автоматичного відстеження кроків, їх типу, відстані, швидкості лише з допомогою глобальної системи позиціонування;
2. Персоналізовані поради для підвищення активності та покращення якості кожного тренування;
3. Персоналізовані досягнення;
4. Розгорнуті статистичні дані про тренування з точним вказанням різних показників у реальному часі;

Недоліки:

1. Для роботи з програмою необхідний обліковий запис;
2. Відсутня можливість відстеження активності без системи глобального позиціонування;
3. Відсутність можливості планування своїх тренувань з використанням календаря;
4. Відсутність безкоштовного відкритого API для можливості використання отриманої додатком інформації для подальшого аналізу та розробки планів тренувань;
5. У додатку відсутня можливість встановлювати українську мову.

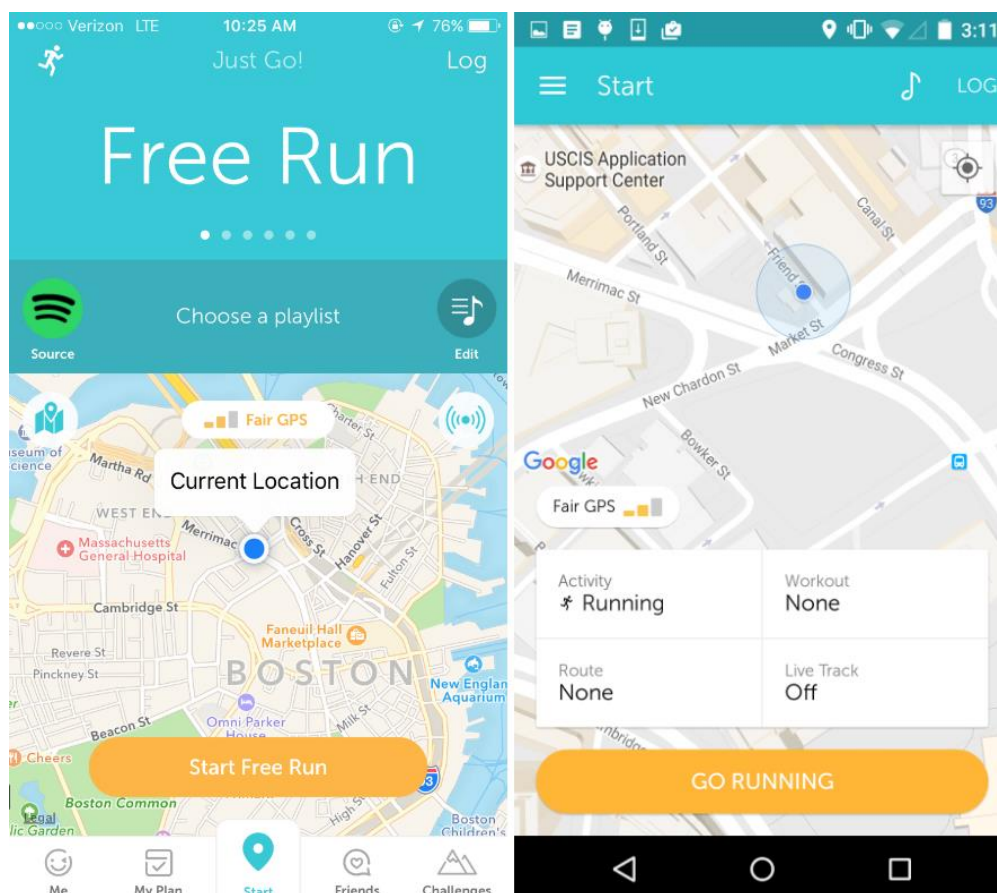
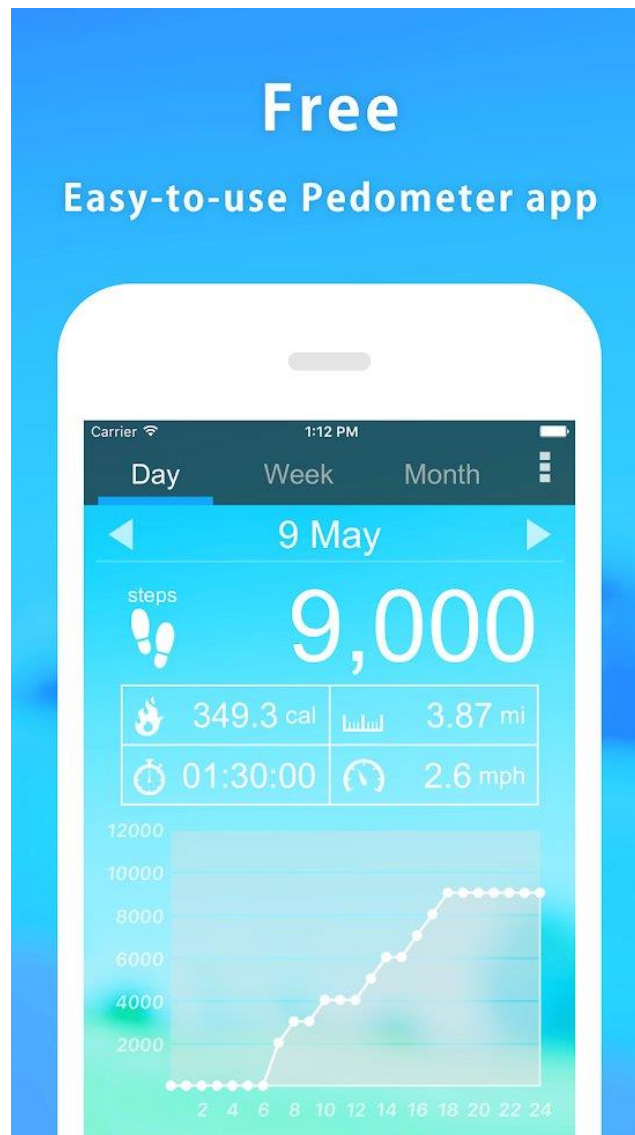


Рис. 1.4. Приклад використання Runkeeper [10]

1.2.4. Pedometer - Step Counter by ITO Technologies, Inc.

Pedometer - Step Counter by ITO Technologies, Inc. – найпростіший із усіх представлених раніше додатків. При запуску програми відразу виводиться екран статистики кроків. Для того, щоб кроки почали рахуватися, потрібно натиснути «пуск». Після цього буде йти підрахунок кроків, калорій, пройденої відстані, часу ходьби і середньої швидкості руху. Можна відразу дивитися статистику за минулу добу, тиждень, місяць.

Додаток має мінімум налаштувань, але дозволяє налаштувати основні функції для крокоміра. Багато пунктів присвячено енергоспоживанню, де можна вибрати, коли додаток буде в режимі сну, яким чином він буде прокидатися, налаштувати чутливість крокоміра. Також можна вибрати оформлення додатка, поділитися своїми результатами в соціальних мережах і інше [11].



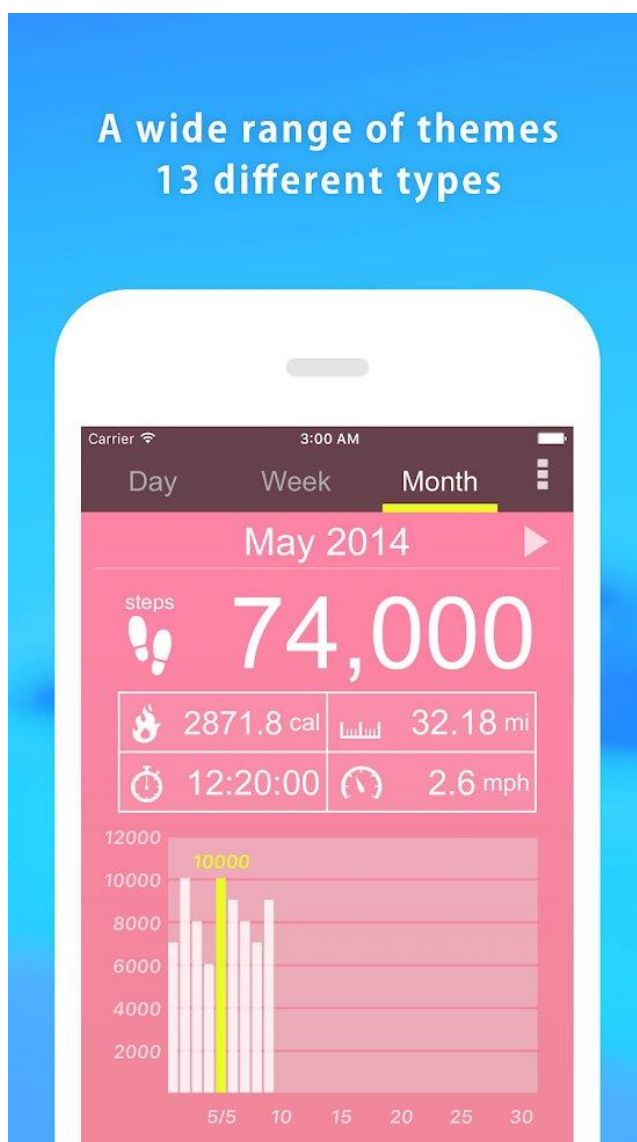


Рис. 1.5. Приклад використання *Pedometer - Step Counter by ITO Technologies, Inc.* [12]

Переваги:

1. Можливість відстеження активності без GPS та підключення до інтернету;
2. Простий та зрозумілий інтерфейс;
3. Розгорнута та детальна статистика про активність користувача;

Недоліки:

1. Для роботи з програмою необхідний обліковий запис;
2. Відсутня можливість відстеження активності з допомогою системи глобального позиціонування;
3. Відсутність можливості планування своїх тренувань з використанням календаря;

4. Відсутня можливість використання української мови.

ВИСНОВОК

У розділі 1 було розглянуто основні аспекти для створення крокоміру (педометру) та досліджено основні готові рішення поставленої задачі. У розділі представлено найпопулярніші існуючі додатки для підрахунку кількості кроків як за допомогою смартфона, так і за допомогою спортивних браслетів, тощо. Кожний із представлених додатків має свої переваги та недоліки, проаналізувавши які можна визначити вимоги до функціоналу розроблюваного додатку.

Отже, розглянемо недоліки готових рішень:

1. Не всі готові рішення мають можливість відстежувати активність без допомоги GPS та мережі Інтернет;
2. Не всі додатки мають відкритий вихідний код, що викликає недовіру користувача, - виникають питання які дані збирає той чи інший додаток, яку інформацію передає на сервер та чи конфіденційно вона зберігається;
3. Деякі додатки не мають можливості відрізнити тип активності;
4. Більшість із розглянутих додатків не мають можливості роботи без облікового запису.

Сформувавши загальний список недоліків, поставимо основні вимоги для розроблюваного додатку:

1. Користувач може користуватися додатком без створення облікового запису;
2. Додаток працює локально та не передає жодних даних до Інтернету;
3. Додаток не використовує систему глобального позиціонування;
4. Додаток відстежує активність користувача у фоновому режимі;
5. Користувач може побачити тип та час кроків, що здійснив;
6. Користувач може встановити цілі «Час активності» та «Кількість кроків» на день;
7. Користувач може проглянути інформацію про швидкість свого руху

впродовж свого тренування (ходьба, біг, що тривав більше ніж 5 хвилин) за допомогою графіку;

8. Користувач може побачити загальну кількість кроків за день, загальний час своєї активності, загальну відстань у кілометрах, що подолав упродовж дня.

Розглянувши недоліки та сформувавши вимоги, перейдемо до створення додатку.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ДОДАТКУ

Предметною областю даної магістерської дисертації є програмний додаток для відстеження фізичної активності за допомогою смартфона, що використовує акселерометр смартфона у якості «детектора» кроків.

У цьому розділі буде обрано технології для подальшої реалізації додатку, описано створений алгоритм для відстеження кроків, розроблено графічний інтерфейс додатку.

2.1. Вибір технологій та їх обґрунтування

Виходячи з поставлених вибог у попередньому розділі, необхідно досягнути максимальної доступності додатку для користувачів. Розглянемо 2 найпопулярніші платформи:

1. iOS;
2. Android OS.

iOS — це власницька мобільна операційна система від Apple. Розроблена спочатку для iPhone, вона стала операційною системою також для iPod Touch, iPad і Apple TV. Apple не дозволяє роботу ОС на мобільних телефонах інших виробників.

iOS є похідною від OS X, отже, є за своєю природою Unix-подібною операційною системою.

Користувацький інтерфейс iOS заснований на концепції прямої маніпуляції з використанням жестів Multi-Touch. Елементи інтерфейсу управління складаються з повзунків, перемикачів і кнопок. Він призначений для безпосереднього контакту користувача з екраном пристрою. Внутрішній акселерометр використовуються деякими програмами для реагування на струшування пристрою, яке є також загальною командою скасування, або обертання пристрою у трьох вимірах, що є загальною командою перемикання між книжковим та альбомним режимами.

Станом на 31 травня 2011 року інтернет-магазин App Store містить понад 500 тисяч застосунків для iOS, які були завантажені понад 15 мільярдів разів. Станом на травень 2010 року, iOS становив 15,4 % ринку операційних систем

для смартфонів, третій після Symbian і Blackberry[13].

Найближчим конкурентом є операційна система Android від компанії Google. У таблиці 2.1 розглянуто актуальне співвідношення смартфонів з різними операційними системами[15].

Таблиця 2.1.

Співвідношення смартфонів за операційними системами

Дата	Android	iOS	Інші
2017-04	71.84%	19.87%	8.29%
2017-05	72.68%	19.32%	8.00%
2017-06	72.89%	19.35%	7.76%
2017-07	73.39%	18.78%	7.83%
2017-08	72.74%	20.32%	6.94%
2017-09	73.52%	19.37%	7.11%
2017-10	73.05%	19.99%	6.96%
2017-11	73.11%	20.34%	6.55%
2017-12	73.54%	19.91%	6.55%
2018-01	74.39%	19.64%	5.97%
2018-02	74.78%	20.15%	5.07%
2018-03	74.24%	20.83%	4.93%
2018-04	75.66%	19.23%	5.11%

Отож, розглянемо альтернативну до iOS операційну систему – Android. Android — (вимов. Андроїд) операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (ОНА).

Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java.

У 84% смартфонів, проданих у 3-ому кварталі 2014 року, була встановлена операційна система Android.

У березні 2017 року ОС Android стала найпопулярнішою ОС, з якої

виходили в інтернет. Так з 37,93% користувачів заходили в інтернет із Android'a, а з Windows лише 37,91% користувачів. В Азії показники ще вищі — 52,2% і 29,2% відповідно[14].

Отже, з Таблиця 3.1 видно, що аби досягнути поставленої мети, - максимальної доступності додатку для користувачів, вибір припадає на мобільну операційну систему Android від американської компанії Google. Крім того, iOS займає теж значну позицію на ринку, однак не має такого відкритого API, як ОС Android, тому додаток буде розроблено для операційної системи Android.

2.1.1. Вибір мінімальної версії системи для розроблюваного додатку

В минулому підпункті було визначено, що додаток буде розроблено для операційної системи Android, тому важливо визначитися з мінімальною версією API, яку підтримуватиме додаток.

У Android існує паралельна підтримка різних версій операційної системи, тобто із виходом нової продовжується підтримка старих версій, а також існує поняття зворотної сумісності, що означає, що додатки розроблені для новіших версій можуть запускатися на мобільних девайсах зі старою версією операційної системи.

Крім мінімальної версії, у розробці додатку є поняття `targetSdk`, тобто версія API для якої будується додаток, і яка є найбільш прийнятною для запуску додатку. За правилами розробки, `targetSdk` не може бути нижче, аніж остання версія ОС.

Розробляючи додаток, мінімальну версію API можна визначити за допомогою `minSdk`. Це значення необхідно вибирати з урахуванням поширеності мінімальної версії API, можливості підтримки нового функціоналу, а також складність підтримки сумісності із кожною версією ОС, що входить у діапазон версій (`minSdk` – `targetSdk`). У Таблиця 2.2 наведено розподіл версій операційної системи Android станом на травень 2019 року[16]. Цей розподіл враховує лише активні пристрої із яких було здійснено вхід до маркетплейсу додатків Google Play.

Таблиця 2.2.

Розподіл версій Android

Версія	Кодове ім'я	API	Розподіл
2.3.3 - 2.3.7	Gingerbread	10	0.30%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.30%
4.1.x	Jelly Bean	16	1.20%
4.2.x		17	1.50%
4.3		18	0.50%
4.4	KitKat	19	6.90%
5	Lollipop	21	3.00%
5.1		22	11.50%
6	Marshmallow	23	16.90%
7	Nougat	24	11.40%
7.1		25	7.80%
8	Oreo	26	12.90%
8.1		27	15.40%
9	Pie	28	10.40%

Для кращого прийняття рішення візуалізуємо дані з таблиці у вигляді кругової діаграми – рис. 2.1.

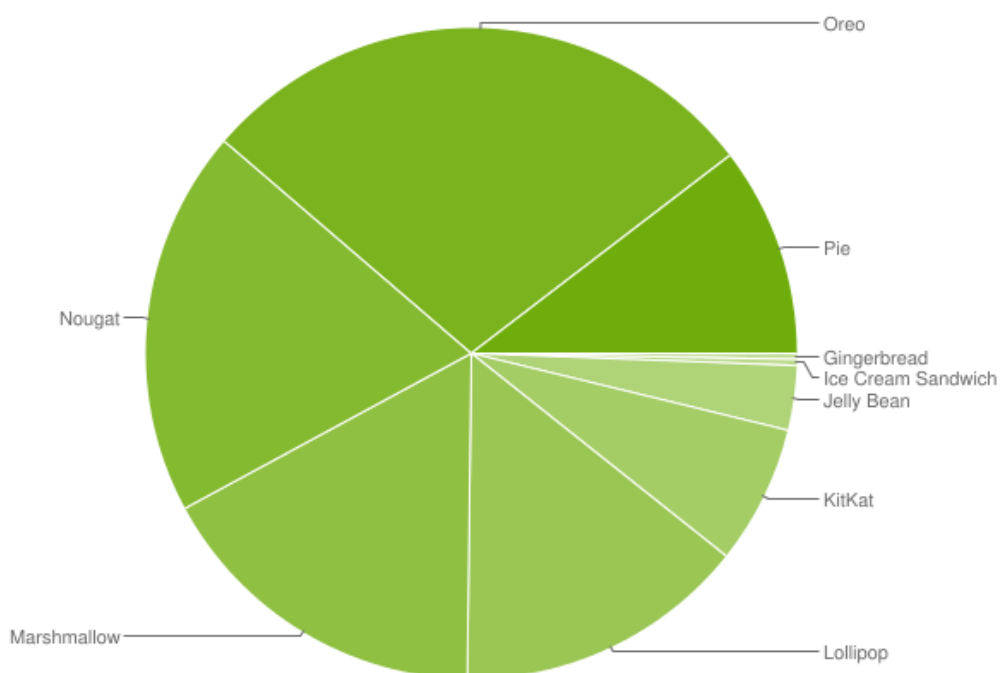


Рис. 2.1. Розподіл версій
операційної системи Android (кругова діаграма) [16]

Отже, враховуючи дані наведені вище, вимоги до додатку і складність підтримки сумісності, було обрано мінімальну версію – 19 (кодове ім'я – KitKat), що забезпечить підтримку майже 95% всіх пристроїв.

2.1.2. Вибір мови програмування

Для розробки додатку доцільно вибрати мови Java або Kotlin, що є офіційними мовами розробки та які підтримує компанія-розробник платформи. Офіційна документація здебільшого адаптована під ці мови та деякі ключові моменти представлено на мові Kotlin, що нещодавно була прийнята як перша офіційна мова ОС, змістивши Java на друге місце. Порівняймо ці дві мови.

Kotlin (Котлін) — статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Також компілюється в JavaScript. Мову названо на честь острова Котлін у Фінській затоці, на якому розміщена частина Кронштадту.

Автори ставили перед собою ціль створити лаконічнішу та типобезпечнішу мову, ніж Java, і простішу, ніж Scala. Наслідками спрощення, порівняно з Scala стали також швидша компіляція та краща підтримка IDE.

Мова розробляється з 2010 року, публічно представлена в липні 2011. Сирцевий код було відкрито в лютому 2012. В лютому було випущено milestone 1, який містив плагін для IDEA. У червні — milestone 2 з підтримкою Android. У грудні 2012 року вийшов milestone 4 та забезпечив підтримку Java 7. Станом на листопад 2015 року основні можливості мови стабілізовані, готується реліз версії 1.0. В грудні 2015 року з'явився реліз-кандидат версії 1.0, а 15 лютого 2016 року відбувся реліз версії 1.0[17].

З 17 травня 2017 року входить в список офіційно підтримуваних мов для розробки додатків для платформи Android.

Java (вимовляється Джава) — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми

компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування[18].

Розглянувши наведену вище інформацію, оберемо Java як мову розробки додатку, оскільки Kotlin хоч і є більш інноваційною мовою розробки додатків для ОС Android, але не вся документація адаптована під неї. Крім того, спільнота розробників на мові Java набагато більша, аніж на мові Kotlin, що сприяє легшому написанню та відладки додатку, пошуку помилок та шляхів їх вирішення

2.1.3. Вибір допоміжних бібліотек

Для розробки додатків для операційної системи Android існує безліч бібліотек із відкритим кодом, що спрощують та іноді пришвидшують процес його розробки, допомагають уникнути повторюваності шаблонного коду, а також реалізувати зручну для розробки та підтримки готових проектів архітектуру.

У проекті використовується наступні бібліотеки:

1. Бібліотека для позиціонування елементів інтерфейсу ConstraintLayout;
2. Бібліотека для зв'язку елементів інтерфейсу із програмною частиною DataBinding;

3. Бібліотека для завантаження, кешування та відображення зображень Fresco.

Розглянемо кожну із них:

Constraint Layout - це група елементів інтерфейсу, яка дозволяє гнучко відтворювати та розміщувати елементи інтерфейсу (віджети). Однією з переваг Constraint Layout є відносне положення.

Відносне позиціонування за допомогою зв'язків є одним з основних блоків створення макетів в Constraint Layout. Ці зв'язки дозволяють розташувати необхідні елементи інтерфейсу одним відносно іншого. Також можливо обмежити елементи по горизонтальній та вертикальній осі:

Горизонтальна вісь: ліва, права, початкова та кінцева частини елементу;

Вертикальна осі: верхня, нижня частина елементу та базова текстова лінія.

Загальна концепція полягає в тому, щоб обмежити певну сторону віджета іншою стороною будь-якого іншого віджета.

Наприклад, для розташування кнопки В праворуч від кнопки А (рис. 2.2) необхідно написати наступний код на мета-мові XML:

```
<Button android:id="@+id/buttonA" ... />
    <Button android:id="@+id/buttonB" ...
        app:layout_constraintLeft_toRightOf="@+id/buttonA" />
```

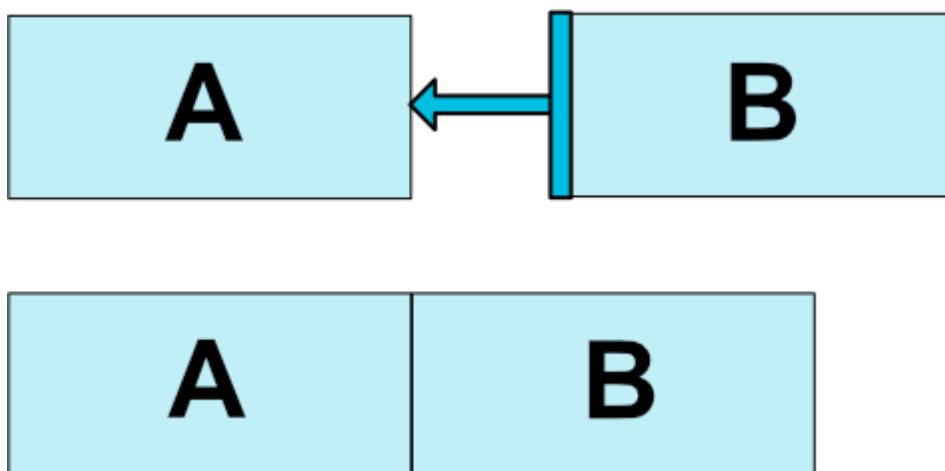


Рис.2.2. Приклад відносного позиціонування [19]

Крім того, можна вставити розриви між зв'язаними віджетами за допомогою тегів в залежності від бажаного результату та орієнтації відступу:

- android:layout_marginStart
- android:layout_marginEnd
- android:layout_marginLeft
- android:layout_marginTop
- android:layout_marginRight
- android:layout_marginBottom

DataBinding - це бібліотека підтримки, яка дозволяє пов'язати компоненти інтерфейсу користувача у макетах із джерелами даних у додатку за допомогою декларативного формату, а не програмного.

Макети часто визначаються в Activity з кодом, який викликає методи рамки інтерфейсу користувача. Наприклад, нижчевказаний код викликає `findViewById()`, щоб знайти віджет `TextView` і прив'язати його до властивості `userName` змінної `viewModel`:

```
TextView textView = findViewById(R.id.sample_text);
textView.setText(viewModel.getUserName());
```

У наведеному нижче прикладі показано, як використовувати Бібліотеку прив'язки даних для призначення текстового віджета безпосередньо у файлі макета. Це виключає необхідність викликати будь-який код Java, показаний вище.

```
<TextView
    android:text="@{viewModel.userName}" />
```

Ця бібліотека дозволяє видалити велику кількість викликів до користувацького інтерфейсу у Activity, що робить саму Activity не такою загрозованою кодом для користувацького інтерфейсу, та й спрощує підтримку додатку. Крім того, бібліотека забезпечує підвищення продуктивності додатку та допомагає уникнути витоків пам'яті (з англ., memory leaks) та ситуацій із значенням `null`.

2.1.4. Вибір архітектури додатку

У попередньому підрозділі було чітко визначено, що буде використовуватися бібліотека `DataBinding`, яка дозволяє пов'язати компоненти інтерфейсу користувача у макетах із джерелами даних у додатку за допомогою

декларативного формату, а не програмного. Однак, використання цієї бібліотеки не обмежується лише зв'язкою компонентів, а й у поєднанні з архітектурою MVVM дозволяє досягти чіткого та легкого для читання і розуміння програмного коду.

Отже, розглянемо детально шаблон MVVM. Model-View-ViewModel (тобто MVVM) - це шаблон архітектури клієнтських додатків, який був запропонований Джоном Госсманом (John Gossman) як альтернатива шаблонами MVC і MVP при використанні технології зв'язування даних (Data Binding). Його концепція полягає в відділенні логіки представлення даних від бізнес-логіки шляхом винесення її в окремий клас для більш чіткого розмежування.

Основними частинами MVVM є:

- View – частина, що взаємодіє з ViewModel та інформує про дії користувача з графічним інтерфейсом;
- ViewModel – обробляє потоки даних, що приходять з View і передає результат обробки до DataModel;
- DataModel - абстрактне джерело даних. ViewModel працює з DataModel для отримання та збереження даних.

Головні переваги цієї архітектури можна визначити такі:

Гнучкість розробки. Цей підхід підвищує зручність роботи в команді, тому що поки один член команди працює над компонуванням і стилізацією екрану - інший, в цей час, описує логіку отримання даних і їх обробки;

Тестування. Така структура спрощує написання тестів і процес створення mock-об'єктів. Також, в більшості випадків відпадає потреба в автоматизованому UI-тестуванні, тому що можна обернути Unit-тестами сам ViewModel;

Розмежування логіки. За рахунок більшого розмежування код стає більш гнучким і простим у підтримці, не кажучи про його читабельності. Кожен модуль відповідає за свою конкретну функцію і тільки.

На рис. 2.3 схематично показано структуру одного модуля додатку з архітектурою MVVM.



Рис. 2.3. Model-View-ViewModel структура класу [20]

2.2. Розробка алгоритму відстеження кроків

У арсеналі Android є 2 спеціалізованих сенсори, що відстежують активність користувача: лічильник кроків та детектор кроків, однак ні один із них не надає інформації у режимі реального часу, тому немає можливості визначити тип та тривалість активності. Крім того, ці сенсори часто блокуються виробниками смартфонів для збереження батареї. Для розробки просунутого крокоміра тривалість та тип активності є важливою складовою, тому використання цих 2 сенсорів не будуть використанні при створенні додатку.

Акселерометр – це датчик, що надає інформацію у реальному часі про положення смартфона за осями x, y та z. Ця інформація складається із 2 компонентів: сили тяжіння, які діють на усі 3 осі та прискорення, яке з’являється внаслідок зміни положення смартфона у просторі. Так, наприклад, коли користувач смартфона прогулюється зі смартфоном у кишені, то і сила тяжіння, що діє на 3 осі, то і прискорення постійно змінюється через рух свого користувача відносно земної системи відліку. Для відстеження активності, у цій роботі пропонується використовувати довжину вектору, що можна визначається за формулою: $\sqrt{x^2 + y^2 + z^2}$, який не залежить від орієнтації смартфона. Варто додати, що довжина вектору враховує прискорення вільного падіння $9,8 \text{ м/с}^2$. На рисунку 2.4 показано графік зміни значень довжини вектору акселерометру при спокійному стані смартфона (смартфон знаходиться нерухомо на столі). Вісь Y – значення довжини вектору, вісь X – час у форматі години:хвилини:секунди:мілісекунди.

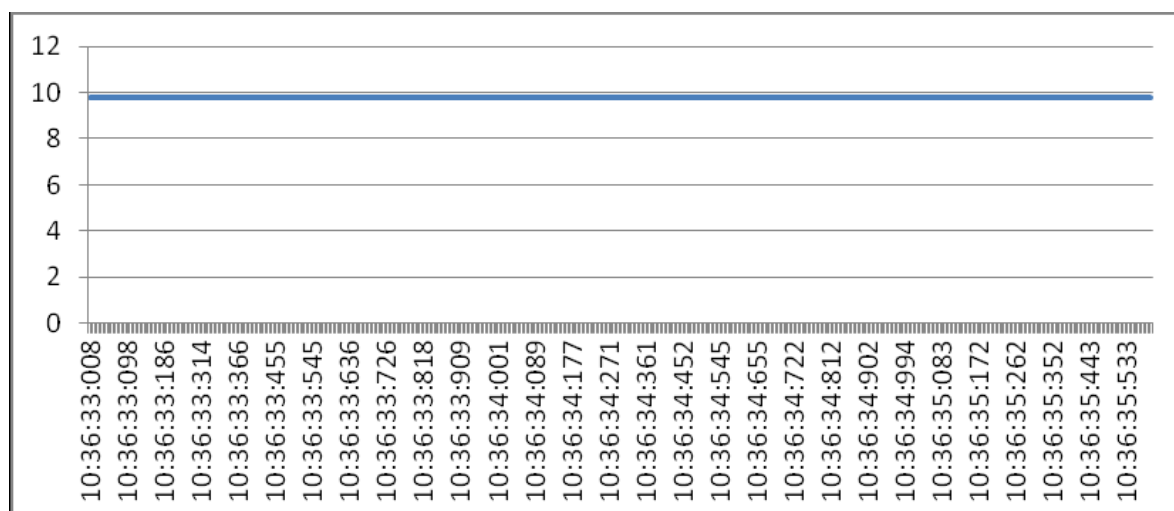


Рис. 2.4. Графік зміни довжини вектору у спокійному положенні

Як видно із графіку, значення по осі Y залишається сталим і дорівнює 9,8 м/с², що відповідає прискорення вільного падіння, що діє на смартфон.

На рисунку 2.5 показано графік зміни довжини вектору від часу, коли смартфон знаходиться у кишені, а його користувач здійснює прогулянку у звичайному темпі.

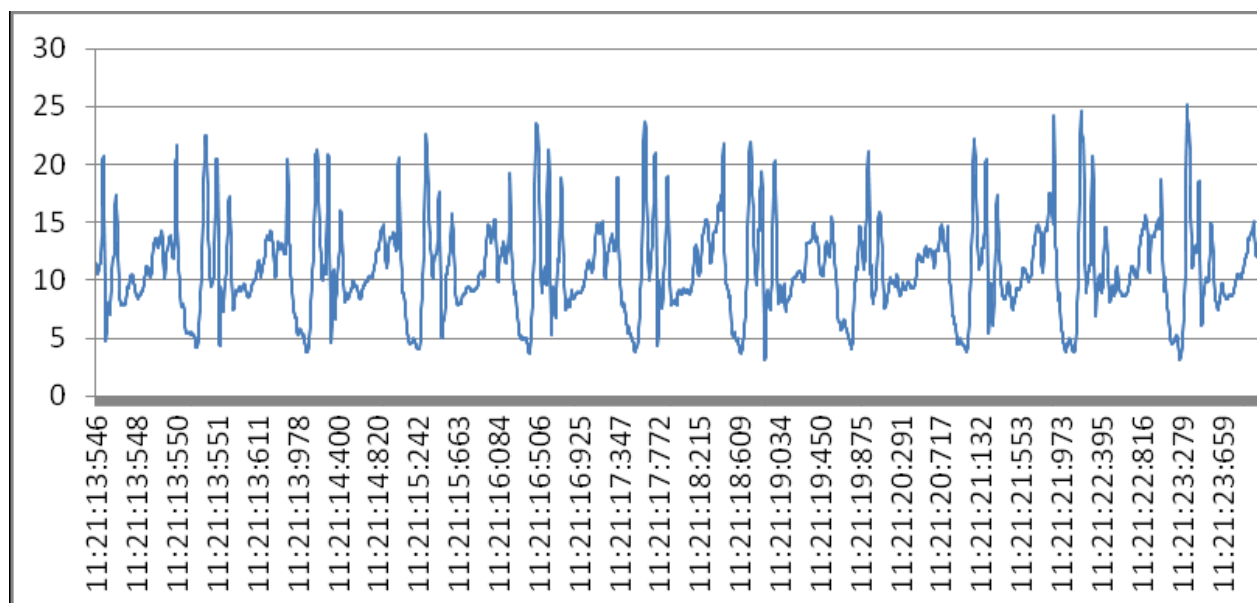


Рис. 2.5. Графік зміни довжини вектору при ходьбі

Показаний вище графік було побудовано на основі даних смартфону за 10 секунд, за які його користувач здійснив 10 кроків тримаючи свій гаджет у кармані своїх штанів. Проаналізувавши графік можна побачити, що:

- Кожний здійснений крок має схожий графік, що схожий та повторюється для інших кроків;
- Для кожного кроку властивий підйом, за яким слідує різкий

спадіння значення;

- Здебільшого, найбільше значення кроку фіксується тоді, коли нога користувача смартфона торкається землі;
- Для виділення кожного нового кроку можна використовувати інформацію про різке зростання значення довжини вектору, а про завершення кроку – різке падіння.

Розглянемо графік (рис. 2.6) значення вектору при швидкій ходьбі чи бігу підтюбцем при тому, що смартфон користувача знаходиться у кармані штанів.

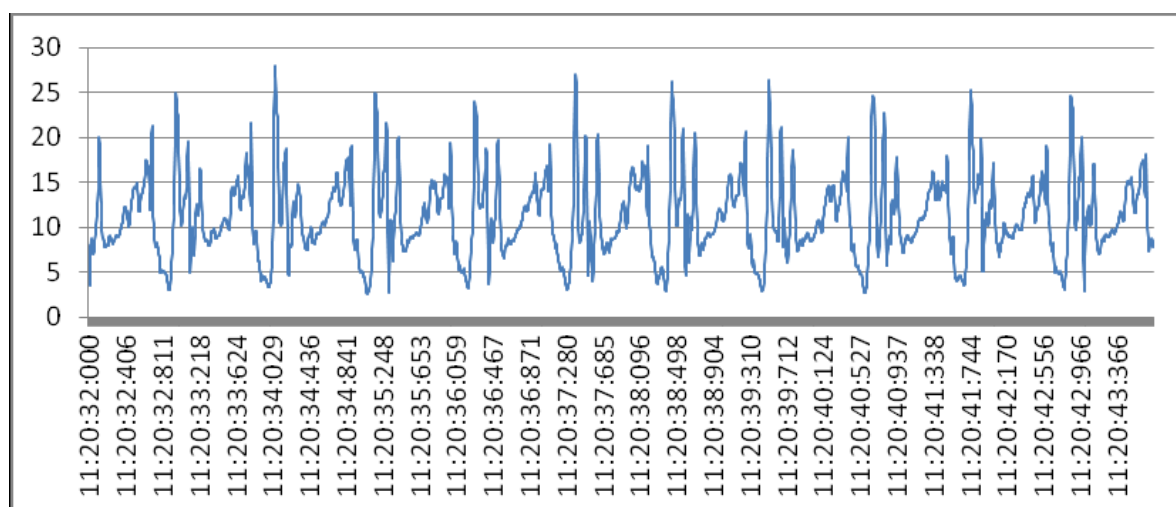


Рис. 2.6. Графік зміни довжини вектору при швидкій ходьбі чи бігу підтюбцем

Отриманий графік дуже подібний до того, що було отримано при звичайній ходьбі з деякими відмінностями:

- Максимальне значення довжини вектору при ходьбі не перевищує 20, тоді як для бігу підтюбцем (швидкої ходи) перевищує 25;
- За той самий час, що і на попередньому графіку (10 секунд), було отримано 11 максимальних значень (тобто 11 кроків). Часовий інтервал між цими вершинами став меншим, аніж при ходьбі.

Розглянемо графік зміни довжини вектору при бігу (рис. 2.7). Смартфон користувача знаходиться у кишені штанів.

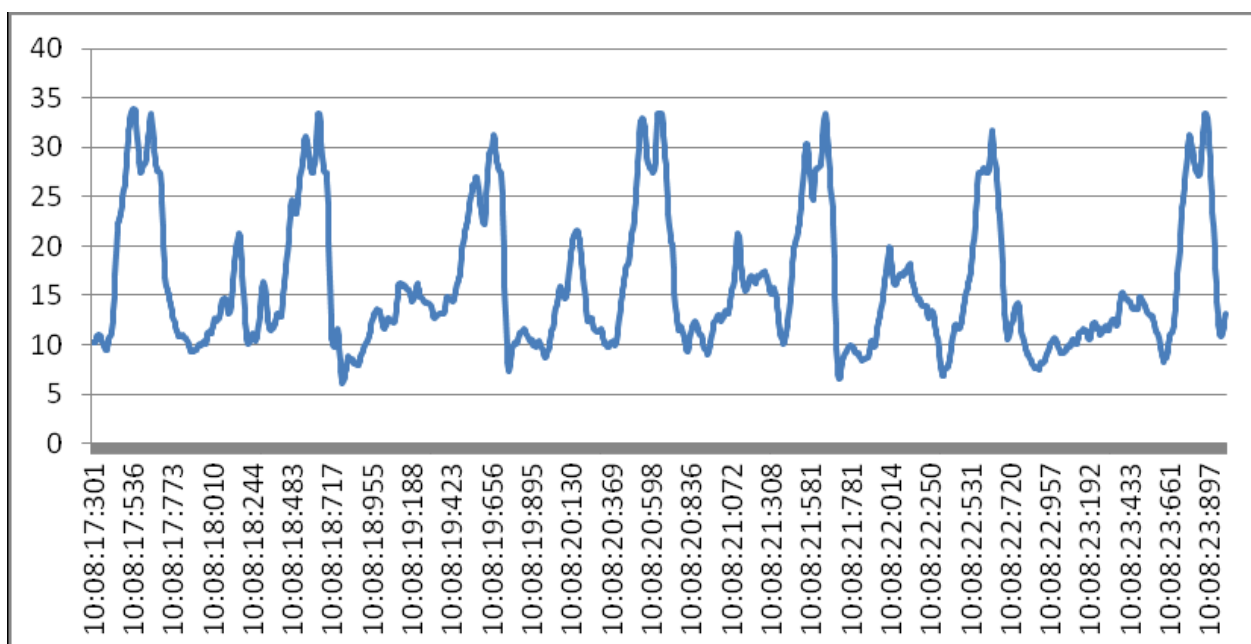


Рис. 2.7. Графік зміни довжини вектору при бігу

Отримавши графіки зміни значення довжини вектору при 3 типах активності, можна зробити наступні висновки:

- Найменше значення вершини при бігу перетинає позначку у 30, тоді як для ходьби – не більше 20, а для швидкої ходьби – не перевищує 25;
- Інтервал між кроками при бігу менший ніж при ходьбі чи бігу підтюбцем.

Звісно, проводячи ці вимірювання знову можна отримати різні результати, які залежать від багатьох факторів: вибір кишені для смартфона (кишеня штанів або кишеня сорочки), типу одягу (у обтяжку чи вільного крою), від типу взуття. Єдине, що не змінюється при проведенні цих тестів – періодичний графік кроків (початок кроку – локальний мінімум значення вектору, за яким обов’язково слідує локальне мінімальне).

Визначення типу кроків. Розглянувши 3 графіки при різному типі активності можна зробити висновок, що при кожному типі є унікальний шаблон. При кожному типі активності акселерометр реєструє локальне максимальне значення вектору у момент коли нога користувача торкається землі. За цим локальним максимальним значенням обов’язково слідує найнижче, що свідчить про те, що крок було завершено. Провівши декілька експериментів було виявлено, що різниця у значенні довжини вектору між типам активності (ходьба, біг підтюбцем та звичайним бігом) дорівнює 5. Так,

для ходьби у звичайному темпі мінімальне значення довжини вектору – 15, а максимальне – 20, для пришвидшеної ходьби (бігу підтюбцем) мінімальне – 20, максимальне – 25, а для бігу – від 25.

ВИСНОВОК

У розділі 2 було проведено аналіз наявних технологій і зроблено вибір для реалізації додатку, а також наведено обґрунтування вибору платформи, мінімальної версії операційної системи Android, мови програмування, бібліотек для спрощення процесу розробки та архітектури для побудови гнучкого та зрозумілого додатку. Обрана операційна система та мінімальна версія операційної системи забезпечить можливість встановлення додатку на 95% всіх пристроїв із операційною системою Android. Використання бібліотеки Databinding надає можливість застосувати MVVM – архітектуру для реалізації додатку, що спрощує реалізацію зв'язків між графічним інтерфейсом користувача та обробкою даних цих даних. Крім того, використання цієї архітектури спрощує тестування додатку та забезпечує розмежування логіки роботи додатку про детально було описано у цьому розділі.

У розділі розкрито принцип роботи алгоритму для виявлення фізичної активності користувача смартфона за допомогою вбудованого акселерометру. Визначено головний показник, за допомогою якого можливо визначити тип активності. Проведено аналіз зміни цього значення від типу активності шляхом заміру цього значення під час ходьби, швидкої ходьби та бігу протягом 10 секунд зі смартфоном (що оснащено акселерометром) у кишені штанів. Виявлено граничні значення цього показника для кожного типу активності: для прогулянки значення знаходиться у межах від 15 до 20, для швидкої прогулянки (бігу підтюбцем) – від 20 до 25, для бігу – від 25.

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ

Процес розробки додатку можна розділити на 3 частини:

1. Реалізація розробленого алгоритму для відстеження активності користувача смартфона зі акселерометром;
2. Розробка графічного інтерфейсу додатку.

3.1. Реалізація розробленого алгоритму

Розроблений алгоритм зі використанням акселерометру смартфона має один великий недолік – для точного відстеження активності додатку потрібно постійно працювати у фоновому режимі, що має негативний вплив на батарею смартфона. Тому у цьому розділі представлений алгоритм буде адаптовано за допомогою наявних інструментів у Android для зменшення навантаження на батарею.

Для того аби відстеження відбувалося у фоновому режимі необхідно створити сервіс – спеціалізований клас, що не має графічного інтерфейсу, однак може виконуватися у фоновому режимі. На рис. 3.1 клас, що виконує функцію сервісу – StepDetector.

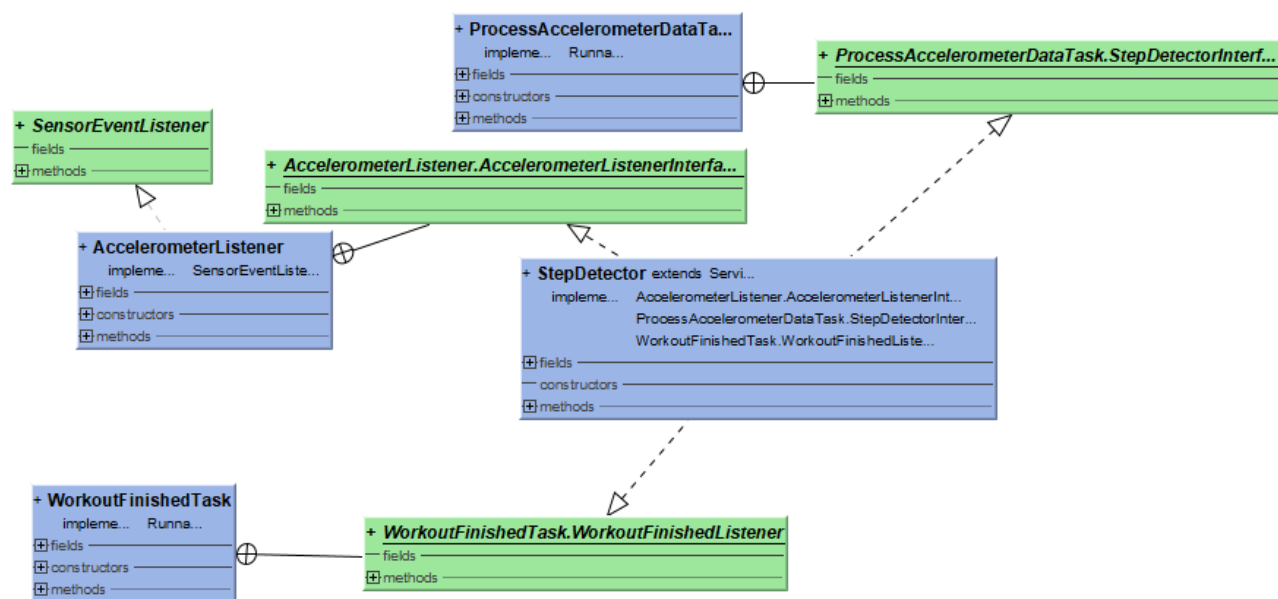


Рис. 3.1. Сервіс для відстеження та обробки отриманої інформації

Для отримання інформації напряму від акселерометру використовується клас AccelerometerListener, який реалізує вбудований інтерфейс

SensorEventListener. За допомогою методу onCreate(), який викликається при створенні сервісу, було створено екземпляр AccelerometerListener, який підключено до SensorManager, який відповідає за усі сенсори смартфона, у тому числі і акселерометр.

Для обробки отриманої інформації використано 2 потоки, що створено за допомогою ScheduledExecutorService. Використання цього класу допоможе планувати виконання задач з певним інтервалом. При плануванні задачі повертається екземпляр класу ScheduledFuture на цю заплановано задачу, яку можна скасувати чи повторити. Планування задачі виконується за допомогою статичного методу ScheduledExecutorService.schedule(). Кожен із 2 потоків буде виконувати одну задачу із певним інтервалом. Перший потік буде використовуватися для того, аби визначити чи завершилось тренування користувача, а другий для обробки отриманої від акселерометру інформації.

Отже, за допомогою методу onCreate() у StepDetector, було створено «пул» із 2 потоків, отримано доступ до акселерометру та зареєстровано «слухач» для отримання його значень. Як тільки слухач було зареєстровано, було заплановано задачу ProcessAccelerometerDataTask для обробки отриманої від акселерометру інформації, що буде виконана через 20 секунд. ProcessAccelerometerDataTask містить у собі методи для обробки отриманої інформації такі як: знаходження максимальних значень у інтервалі для реєстрації кроку та відсіювання псевдокроків.

Реєстрація кроків відбувається наступним чином: якщо значення вектору більше 15 (мінімальне значення, що відповідає ходьбі) та за ним слідує різке зниження значення, то це зараховується як крок. Однак, як показано на графіках у попередньому розділі, за справжнім кроком може спостерігатися псевдокрок. На рис. 3.2. червоним кольором виділено мінімальні значення, що сповіщають про закінчення кроку, жовтим – справжній початок кроку, а зеленим – псевдопочаток кроку.



Рис. 3.2. Псевдокроки на графіку залежності значення довжини вектору від часу

Для фільтрації цих псевдокроків було створено окремий метод, що відсіює ті максимальні значення, що близькі одне до одного (різниця у часі менше 400 мілісекунд).

Після виконання цієї обробки отриманої від акселерометру інформації, викликається метод `onStepDetected()`. Якщо відстежених кроків менше ніж 7, то інформація про ці кроки ігнорується, оскільки вони могли бути отримані внаслідок перевертання смартфона, тощо. Якщо ж їх більше, то фіксується час першого кроку та створюється екземпляр класу `WorkoutSession`, що зберігає інформацію про тренування. У екземпляр класу `WorkoutSession` додається інформація про кожен крок та визначається його тип. Після додавання усіх кроків, фіксується час останнього кроку та швидкість руху користувача у період 20 секунд. Після цього було заплановано задачу збереження інформації у базу даних про тренування, що виконається через 50 секунд. Якщо ця задача уже була запланована, то її скасувати та запланувати її ще раз через 50 секунд. Якщо кроків не було відстежено, задача збереження виконається та інформація про кроки відобразиться у графічному інтерфейсі додатку.

При кожному виклику задачі збереження інформації про кроки перевіряється чи кількість кроків більша за 5. Якщо їх кількість менша, то інформація ігнорується. Якщо тривалість активності більша за 5 хвилин, то до бази даних зберігається детальний опис інформації про тренування. Якщо менше 5 хвилин, то кількість кроків просто фіксується у загальній кількості кроків за день. При збереженні інформації про кроки у якості тренування,

користувач матиме змогу побачити детальний опис тренування з графіком зміни швидкості за час тренування. Крім того, при завершенні тренування користувач отримає повідомлення про завершення активності із можливістю перейти до розгорнутого аналізу отриманої інформації.

Збереження інформації відбувається за допомогою локальної реляційної бази даних SQLite3. Для інформації про кількість кроків за день, фіксуються наступні значення у БД:

- День, місяць, рік активності;
- Кількість кроків кожного тип активності;
- Загальний час активності;
- Загальна відстань за день;
- Загальна кількість кроків за день;
- Інформація про те чи були досягнуті поставлені цілі.

Інформація, що фіксується при збереженні у вигляді тренування:

- Інформація про швидкість у моменти обробки значень, що були отримані від акселерометру;
- Час початку тренування;
- Час кінця тренування;
- Тривалість тренування;
- Відстань;
- Кількість кроків та їх тип;
- Загальний тип тренування, що визначається максимальною кількістю кроків у кожному із типів;
- Кількість «спалених» калорій.

При завершенні тренування, крім збереження інформації, перевіряється чи досяг користувач поставлених цілей. Якщо ціль досягнута, користувач отримає повідомлення про це та побачить у графічному інтерфейсі інформацію про це.

Створений сервіс для відстеження буде стартувати при кожному відкритті додатку та продовжуватиме працювати у фоновому режимі до моменту поки

користувач не закриє додаток через системні налаштування.

3.2. Реалізація графічного інтерфейсу

Відкривши додаток уперше користувач побачить стартовий екран (рис. 3.3), який по мірі активності користувача буде заповнюватися статистичною інформацією (рис. 3.4). Головний екран містить інформацію про активність користувача протягом дня. За замовчуванням відкривається сьогоднішній день, однак користувач може переглядати історію своєї активності за допомогою свайпів вліво та вправо для переходу на день назад чи вперед відповідно.

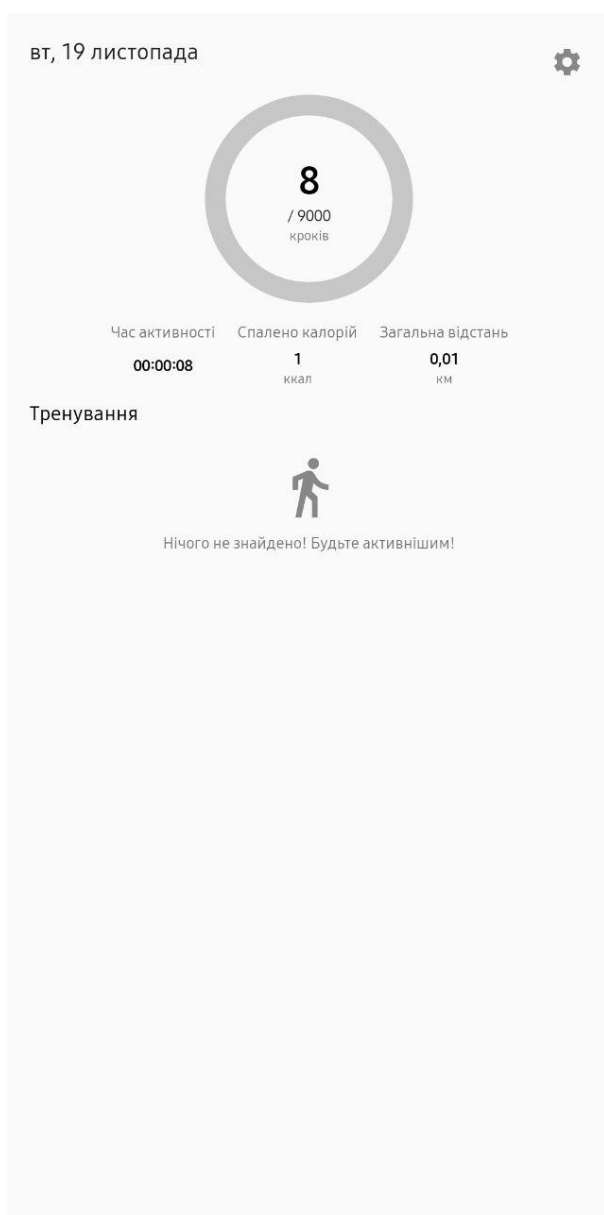


Рис. 3.3. Початковий екран додатку при першому запуску додатку

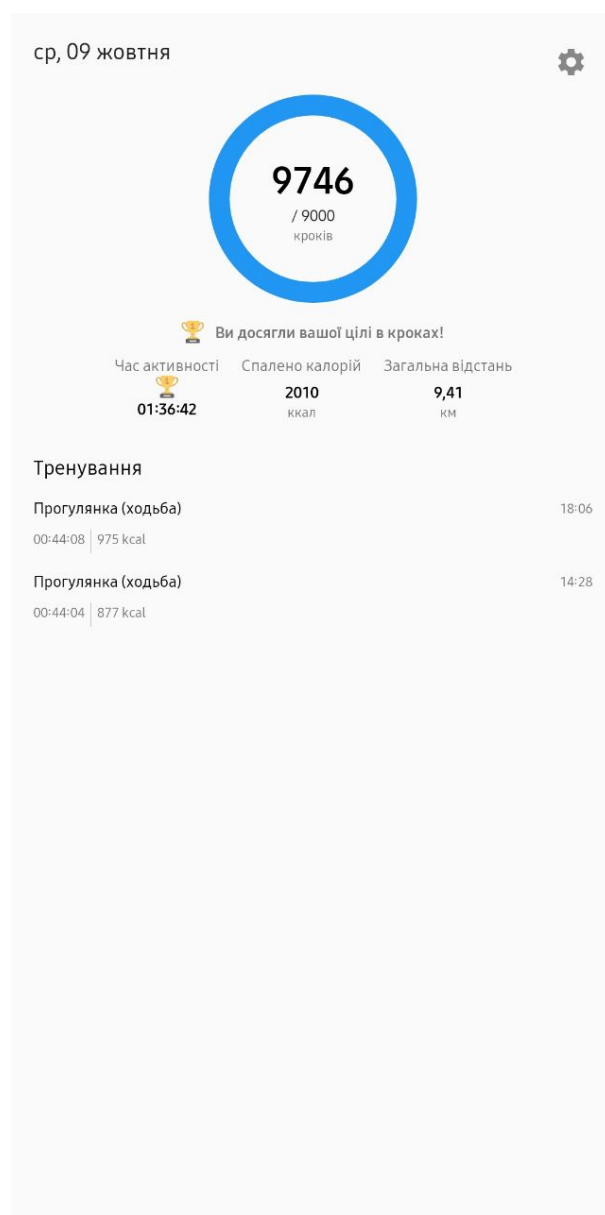


Рис. 3.4. Приклад екрану зі статистичною інформацією за день, що заповнюється по мірі активності

користувача

Головний екран включає інформацію про активність користувача протягом дня таку як:

- Загальну кількість кроків протягом дня;
- Кругову діаграму, де у процентному співвідношенні показано кількість кроків, що пройдено, до кількості кроків, що залишилося пройти задля досягнення поставленої цілі. Якщо цілі досягнуто, то під круговою діаграмою виводиться відповідне повідомлення;
- Загальний час активності користувача;
- Загальну кількість калорій, що було «спалено» протягом дня;
- Загальну відстань, що було пройдено протягом дня;
- Інформацію про тренування, що були відстежені. Під тренуванням мається на увазі прогулянка, біг, швидка прогулянка з тривалістю більше 5 хвилин. Для кожного тренування відображається загальна інформація про тривалість тренування: тип тренування, кількість «спалених» калорій та час початку тренування. Натиснувши на тренування є можливість побачити детальну інформацію (рис. 3.7).

При досягненні цілі користувач отримує сповіщення, приклад якого вказано на рис. 3.5.

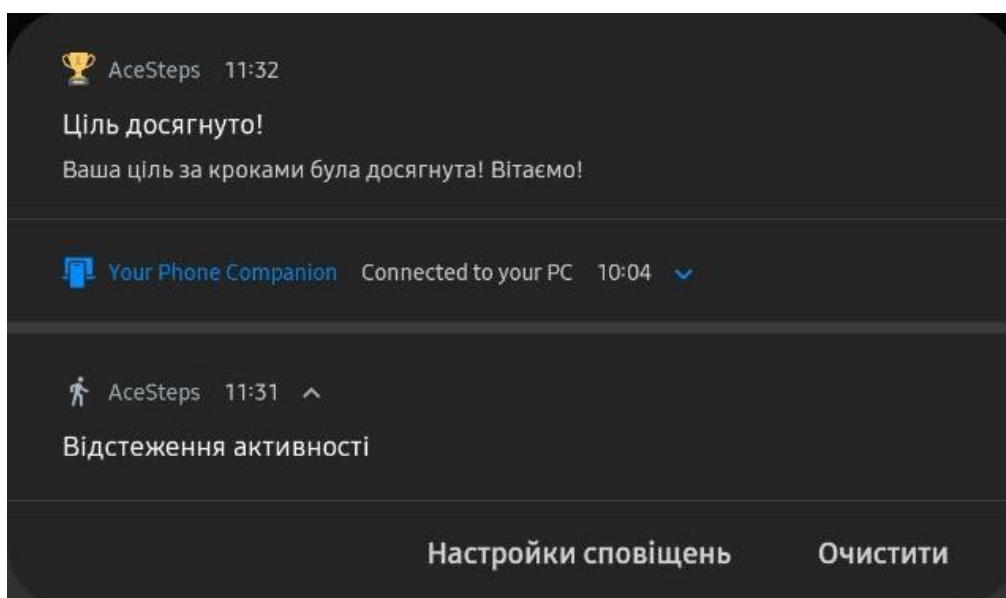


Рис. 3.5. Приклад сповіщення про досягнення цілі

По завершенню тренування користувач отримає сповіщення про час та

вид тренування, приклад якого вказано на рис. 3.6.

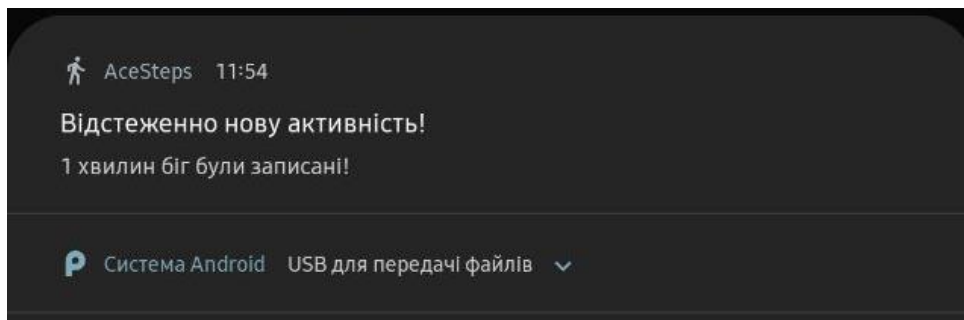


Рис. 3.6. Приклад сповіщення про нове тренування

Крім того, на головному екрані є можливість перейти до налаштувань додатку (рис. 3.8).



Рис. 3.7. Екран інформації про тренування

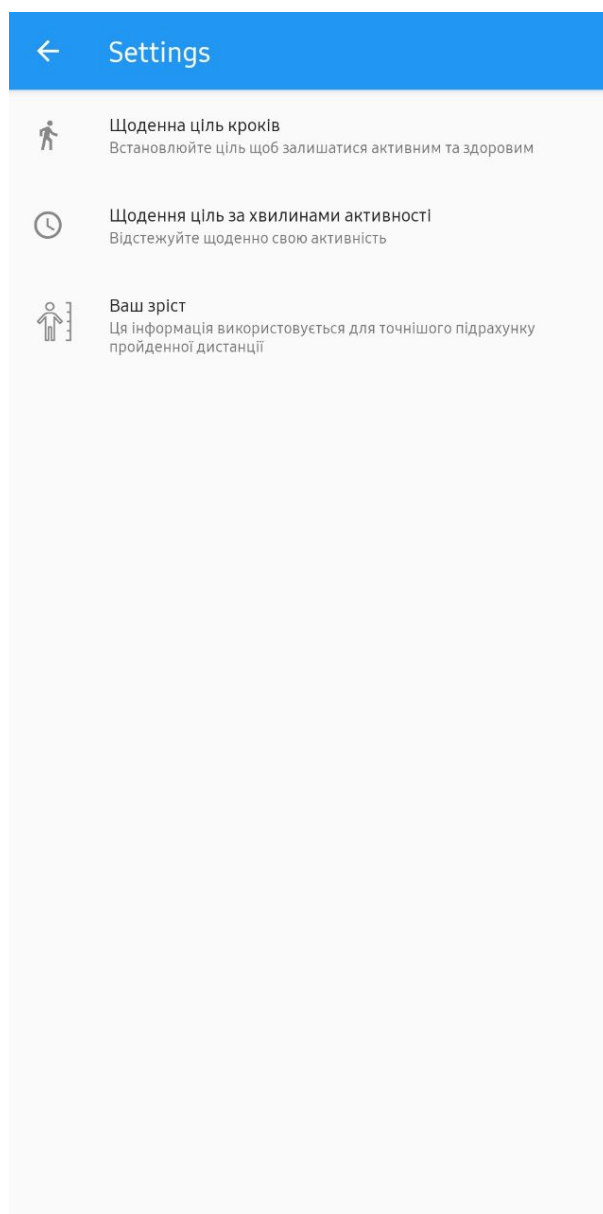


Рис. 3.8. Екран налаштувань

Екран тренування містить наступну інформацію: дату та час тренування, тривалість тренування, загальну відстань, що пройдено та середню швидкість протягом усього тренування. У шапці цього екрану відображається тип активності та є можливість видалити інформацію про тренування, якщо воно було здійснено у поточний день. Звертаємо увагу, якщо користувач видалить інформацію про тренування, то ця операція не впливає на загальну кількість кроків протягом дня.

Крім того, на екрані відображається графік швидкості протягом тренування. Розглянемо у якості прикладу графік на рис. 3.7. Як видно із графіку, користувач виконував прогулянку та близько третьої хвилини швидкість впала до 0, що може свідчити про те, що користувач використовував смартфон чи просто не здійснював жодної активності.

Під графіком розміщується додаткова інформація про тренування: загальна кількість кроків за час тренування, кількість «спалених» калорій та середній час проходження 1 кілометру.

Екран налаштувань дозволяє користувачу додатку встановлювати щоденні цілі у кроках та часі активності, а також свій ріст задля правильного підрахунку відстані. На рис. 3.9, 3.10 та 3.11 показано екрани налаштувань цілі у кроках, цілі у часі активності та росту відповідно.

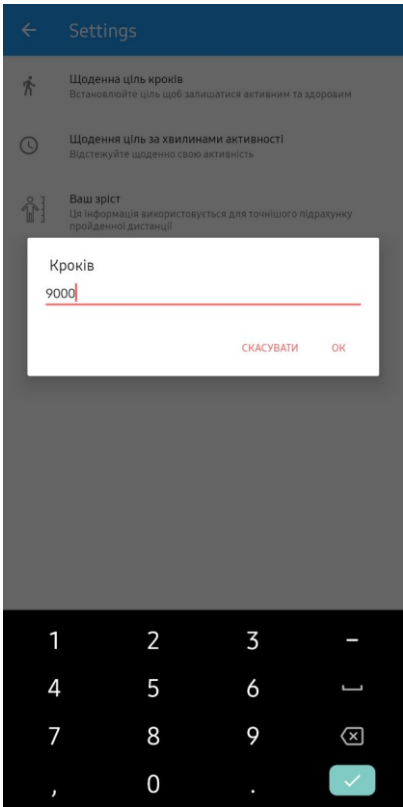


Рис. 3.9. Екран налаштування щоденної цілі у кроках

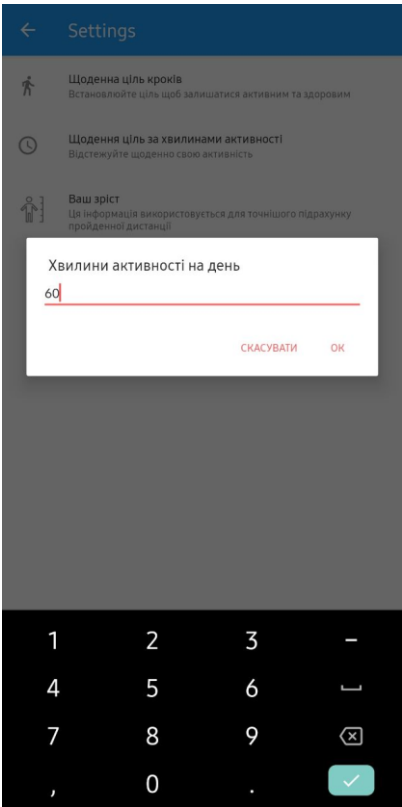


Рис. 3.10. Екран налаштувань щоденної цілі активності у часі

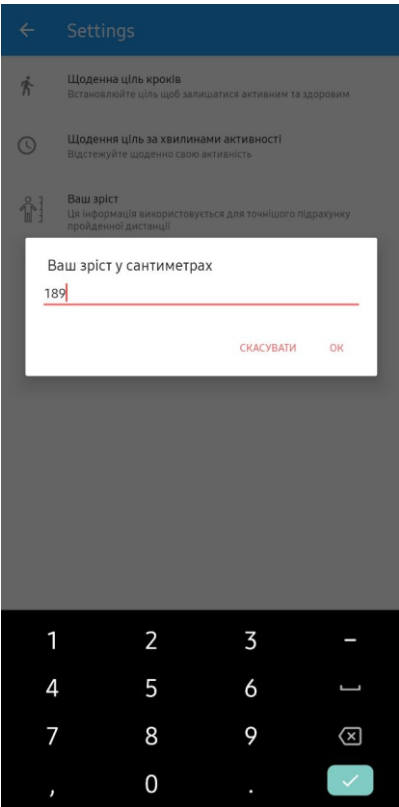


Рис. 3.11. Екран налаштувань росту користувача

Тестування розробленого алгоритму

Розробивши графічний інтерфейс, є можливість порівняти результати роботи алгоритму з аналогічними додатками. Для порівняння розробленого алгоритму обрано додаток Samsung Health, що також здатний записувати кроки у фоновому режимі та спеціалізований спортивний браслет Xiaomi Mi Band 2. Розроблений додаток та додаток Samsung Health працювали одночасно на одному і тому ж пристрої, що знаходився у кишені штанів, а браслет знаходився на лівій руці. Результати відстеження за допомогою 2 додатків та спортивного браслету представлені у таблиці 3.1.

Таблиця 3.1.

Порівняння результатів

Тип активності	Кількість кроків		
	Розроблений додаток	Samsung Health	Xiaomi Mi Band 2

Ходьба	5822	6125	5639
Ходьба	6521	6712	6502
Ходьба	12898	13125	12745
Швидка ходьба	6854	7021	6757
Швидка ходьба	7412	7541	7321
Швидка ходьба	9854	9987	9814
Біг	10054	10127	10027
Біг	12014	12634	11951
Біг	15210	15354	14987

Як видно із таблиці вище, отримані результати близькі до тих, що записані за допомогою конкурентного рішення від Samsung та спеціалізованого спортивного браслету Xiaomi Mi Band 2. Порівняймо отримані за допомогою розробленого додатку результати зі Samsung Health та Xiaomi Mi Band 2 (табл. 3.2).

Таблиця 3.2.

Похибка отриманих результатів

Тип активності	Різниця між розробленим додатком та (%):	
	Samsung Health	Xiaomi Mi Band 2
Ходьба	4.95%	3.25%
Ходьба	2.85%	0.29%
Ходьба	1.73%	1.20%
Швидка ходьба	2.38%	1.44%
Швидка ходьба	1.71%	1.24%
Швидка ходьба	1.33%	0.41%
Біг	0.72%	0.27%
Біг	4.91%	0.53%
Біг	0.94%	1.49%
Максимальна похибка	4.95%	3.25%
Середня похибка	2.39%	1.12%

Як видно з побудованої таблиці, максимальна похибка у ході 9 тестових випробовуваннях розробленого додатку у порівнянні з Samsung Health не перевищує 5%, а середня похибка 2.39%. При порівнянні зі спортивним браслетом Xiaomi Mi Band 2, виявлено, що середня похибка значно менша – 1.12%, а максимальна похибка не перевищує 3.5%.

Отже, розроблений додаток гарантує правильність підрахунку кроків з похибкою, що не перевищує 5% у порівнянні з аналогічними додатками, та не більш ніж 3.5% у порівнянні зі спеціалізованим спортивним браслетом.

Отримані у ході тестування результати показали, що прийняті рішення при розробці додатку були вірні, а результати є у межах допустимої похибки.

ВИСНОВОК

У цьому розділі було реалізовано раніше розроблений спосіб для відстеження фізичної активності, розроблено графічний дизайн додатку та проведено порівняльний аналіз отриманих результатів із конкурентами. Крім того, у даному розділі вирішено наступні задачі:

- оптимізовано розроблений алгоритм для зменшення навантаження на акумулятор смартфона;
- було обґрунтовано вибір системних класів для реалізації задач, що виконуються у фоновому режимі;
- обґрунтовано вибір бази даних для збереження інформації про щоденну активність користувача та тренування, що здійсненні протягом дня;
- описано структуру реляційної бази даних;
- описано процес розробки додатку та інтерфейсу користувача, що продемонстровано на відповідних рисунках. Була надана інформація про кожен елемент, що доступний на кожному екрані додатку, та наведено сценарії використання додатку;
- продемонстровано приклад головного екрану, на якому відображено головну активність користувача протягом дня;
- продемонстровано приклад екрану тренування.

Запропоновані приклади демонструють виконання алгоритму, запропонованого у попередньому розділі та підтверджує справедливність розробленого способу відстеження фізичної активності користувача за допомогою акселерометру смартфона.

ЗАГАЛЬНІ ВИСНОВКИ

Дана магістерська дисертація пропонує спосіб для відстеження активності користувача використовуючи смартфон, що обладнано акселерометром.

У рамках цієї дисертації було проведено порівняльний аналіз наявних рішень, оцінено їх переваги та недоліки, сформовано основні вимоги для розроблюваного додатку.

Проведено аналіз технологій для реалізації способу та оцінено переваги кожного із них, а також описано переваги використання бібліотек, визначено мінімальну версію операційної системи для забезпечення максимальної сумісності для забезпечення найбільшої цільової групи користувачів. Також було обрано мову програмування: серед 2 наявних варіантів – Kotlin та Java, - було обрано другу. Крім того, було обрано додаткові бібліотеки.

Крім того, у розділі 2 проведено аналіз роботи акселерометру смартфона для подальшого використання отриманої інформації для цілей цієї дисертації. Також було запропоновано алгоритм для відстеження фізичної активності, її типів: біг, ходьба, біг підтюпцем. Визначено основні критерії для відстеження типу та вирізнення активності одна від одної.

Виходячи із вимог поставлених у розділі 2, наступний розділ розкриває процес розробки додатку, а саме: реалізацію розробленого алгоритму на операційній системі Android, його адаптацію під цю систему. В рамках цього розділу було реалізовано графічний інтерфейс для перегляду інформації, отриманої за допомогою розробленого алгоритму, та детальний аналіз цієї інформації. Крім того, реалізовано можливість для детального перегляду звіту про кожне тренування та модифіковано алгоритм для відстеження швидкості у режимі реального часу для побудови точного графіку для перегляду по завершенню тренування.

Розроблений додаток було протестовано та отримані результати було порівняно з отриманими за допомогою конкурентного рішення Samsung Health та спеціалізованого спортивного браслету Xiaomi Mi Band 2. У ході тестування виявлено, що похибка не перевищує 5%.

ЛІТЕРАТУРА

1. Varun Nagpal Android Sensor Programming By Example / V. Nagpal // Packt Publishing. – 2019. - №6
2. Harvard.edu: Counting every step you take [Електронний ресурс]. Режим доступу: https://www.health.harvard.edu/newsletter_article/counting-every-step-you-take
3. Eachstepyoutake.com: The Benefits of a Pedometer [Електронний ресурс]. Режим доступу: <http://eachstepyoutake.com/the-benefits-of-a-pedometer/>
4. Mohdaslam.com: Pedometer and Step Counter Mobile Apps : How does it work? [Електронний ресурс]. Режим доступу: <http://mohdaslam.com/pedometer-and-step-counter-mobile-apps-how-does-it-work/>
5. Wikipedia.org: Samsung Health [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/Samsung_Health
6. Itc.ua: S Health для спорту та здорового образу життя на прикладі Samsung Galaxy S6 edge [Електронний ресурс]. Режим доступу: <https://itc.ua>
7. Google Fit Як відстежувати фізичну активність за допомогою годинника [Електронний ресурс]. Режим доступу: <https://support.google.com/fit/answer/6056874?hl=uk>
8. Play.Google.com: Google Fit – здоров'я та трекер активності [Електронний ресурс]. Режим доступу: <https://play.google.com/store/apps/details?id=com.google.android.apps.fitness>
9. Runkeeper [Електронний ресурс]. Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Runkeeper>
10. Runkeeper.com: How to Broadcast Your Runs While on the Go [Електронний ресурс]. Режим доступу до ресурсу: <https://blog.runkeeper.com/362/how-to-broadcast-your-runs-while-on-the-go/>
11. Naandroidi.top: Крокомір: які вибрати програми на андроїд [Електронний ресурс]. Режим доступу: <https://www.naandroidi.top/programy-na-android/krokomir-yaki-vybraty-programy-na-android/>
12. Play.Google.com: Pedometer - Step Counter [Електронний ресурс].

Режим доступа: <https://play.google.com/store/apps/details?id=com.tayu.tau.pedometre>

13. iOS [Электронный ресурс]. Режим доступа: <https://gsmhub.com.ua/glossary/apple-ios>

14. Android [Электронный ресурс]. Режим доступа: <https://www.android.com/everyone/>

15. Statcounter.com: Mobile Operating System Market Share Worldwide [Электронный ресурс]. Режим доступа: <http://gs.statcounter.com/os-market-share/mobile/worldwide>

16. Android.com: Distribution dashboard [Электронный ресурс]. Режим доступа: <https://developer.android.com/about/dashboards/>

17. Kotlin [Электронный ресурс]. Режим доступа: <https://kotlinlang.org/>

18. Java [Электронный ресурс]. Режим доступа: <https://www.java.com/en/>

19. Developer.Android.com: Constraint Layout | Android Developers Website [Электронный ресурс]. Режим доступа: <https://developer.android.com/reference/android/support/constraint/ConstraintLayout>

20. MVVM: Android Architecture Patterns Part 3: Model-View-ViewModel [Электронный ресурс]. Режим доступа: <https://upday.github.io/blog/model-view-viewmodel/>