

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

СТЕПАНЮК ОЛЕНА СЕРГІЇВНА

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

доктор технічних наук, доцент

_____ Т. В. Нескородева

« ____ » _____ 20__ р.

**РОЗРОБКА ФІТНЕС-ДОДАТКУ З АНАЛІТИЧНИМИ ФУНКЦІЯМИ НА
ПЛАТФОРМІ IOS**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Нескородева Т. В., д.т.н., доцент
кафедри інформаційних технологій

Оцінка: ____ / ____ / ____

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

(підпис)

АНОТАЦІЯ

Степанюк О. С. Розробка фітнес-додатку з аналітичними функціями на платформі iOS. Спеціальність 122 «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

Дана робота присвячена розробці мобільного фітнес-додатку з аналітичними функціями для платформи iOS.

У вступі наведено актуальність теми дослідження, сформульована мета, предмет, об'єкт та задачі кваліфікаційної роботи. У першому розділі проведений огляд існуючих аналогів та сформовані основні вимоги. Другий розділ містить відомості про використані інструменти та технології. Третій розділ присвячений проектуванню архітектури мобільного додатку. Четвертий розділ містить опис функціоналу додатку. У висновку підбитий підсумок проведеної роботи.

Ключові слова: iOS, Swift, мобільна розробка, аналітика, фітнес-додаток.

55 с., 44 рис., 2 табл., 41 джерело.

ABSTRACT

O. S. Stepaniuk. Development of the fitness application with analytical functions on the iOS platform. Specialty 122 "Computer Science". Vasyl' Stus Donetsk National University, Vinnytsia, 2022.

This work is devoted to the development of a mobile fitness application with analytical functions on the iOS platform.

The introduction presents the relevance of the research topic, formulates the purpose, subject, object and objectives of the qualification work. In the first section the review of existing analogues is carried out and basic requirements are formed. The second section contains information about the tools and technologies used. The third section is devoted to designing the architecture of a mobile application. The fourth section contains a description of the application's functionality. In conclusion, summed up the work done.

Keywords: iOS, Swift, mobile development, analytics, fitness app.

55 p., 44 pics., 2 tables, 41 sources.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Аналіз ринку мобільних технологій	6
1.2 Постановка задачі	7
1.3 Огляд існуючих аналогів	8
РОЗДІЛ 2. ВІДОМОСТІ ПРО ВИКОРИСТАНІ ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ.....	13
2.1 Мова програмування	13
2.2 Інструмент для побудови дизайну додатку.....	14
2.3 Середовище розробки.....	15
2.3 Менеджер залежностей	17
2.4. Використані бібліотеки	18
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	21
3.1 Опис шаблону проектування.....	21
3.2 Використання UML для проектування додатку	25
РОЗДІЛ 4. ОПИС ФУНКЦІОНАЛУ ДОДАТКУ	28
4.1 Опис користувацького інтерфейсу	28
4.2 Аналітика в розробленому додатку	39
4.3 Технологія Push Notifications у додатку	48
ВИСНОВКИ.....	51
СПИСОК ЛІТЕРАТУРИ.....	52

ВСТУП

Сьогодні жодна людина у світі не може уявити ні дня без смартфона. За допомогою мобільних пристроїв людство вирішує багато рутинних задач, автоматизує різноманітні процеси, надає нові можливості для розвитку та досліджень. Завдяки своїй поширеності мобільні пристрої давно є чи не основним способом поширення програмного забезпечення серед користувачів.

Ринок мобільних пристроїв надзвичайно широкий і тому все більше розробників програмного забезпечення звертають на нього увагу. Однією з найпопулярніших операційних систем є iOS від компанії Apple. Це потужна платформа, що надає широкий функціонал, довгострокові оновлення та має чудову продуктивність. На відмінну від Android, вона підтримується лише на пристроях компанії Apple, що значно покращує процес розробки мобільних додатків і їх тестування, бо всі технічні і фізичні характеристики мобільних пристроїв заздалегідь відомі. Екосистема Apple включає крім iOS ще декілька платформ - iPadOS - для планшетів, watchOS - для наручних годинників, macOS - для персональних комп'ютерів та ноутбуків, та tvOS - для відео пристроїв Apple TV. Однією з основних переваг даної екосистеми є безпека користувача, що реалізується шляхом ізолювання додатків та сервісів один від одного та унеможливорює витік особистих персональних даних. Також варто наголосити на тривалій підтримці застарілих моделей мобільних пристроїв.

Сьогодні тема здоров'я - надзвичайно перспективний напрямок мобільних розробок. За останній час з'явились безліч технологій, що відкрили можливість створення нових цікавих рішень. Як приклад, варто назвати HealthKit - рішення від компанії Apple, призначене для зберігання та обміну даними про здоров'я. Даний інструмент допомагає інтегруватися з різними пристроями, що вимірюють пульс, зчитують рівень холестерину в крові, відслідковують фізичну активність та сон, повідомляють користувача, коли показники знаходяться за межами норми, дозволяють коригувати харчування та навіть знімають електрокардіограму.

Метою кваліфікаційної роботи є розробка мобільного додатку для відстеження фізичної активності з аналітичними функціями за допомогою мови програмування Swift та інших засобів, що надає операційна система iOS.

У рамках даної кваліфікаційної роботи основними задачами є:

1. Поглиблене дослідження засобів для створення мобільних додатків для платформи iOS.
2. Аналіз існуючих програмних продуктів у категорії “Здоров’я”.
3. Створення додатку для відслідковування фізичної активності та її аналізу для мобільної платформи iOS за допомогою мови програмування Swift, бібліотек HealthKit та MapKit.

Об’єктом дослідження безпосередньо виступають технології iOS розробки, а предметом дослідження є фітнес-додаток з аналітичними функціями для платформи iOS.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз ринку мобільних технологій

Сьогодні мобільні пристрої забезпечують не лише зв'язок, а й виконання багатьох інших важливих функцій. Для мільярдів людей по всьому світу мобільні телефони забезпечують доступ до інформації, життєво важливих послуг у сфері охорони здоров'я, освіти, суспільства та економіки. Їх можна використовувати для оплати покупок, як квиток у громадському транспорті, відео реєстратор та навіть як ключ від дверей. Значного впливу на розвиток мобільних технологій завдала пандемія. Люди стали уважнішими до свого здоров'я. Це означає, що зростає попит на додатки медичних сервісів, контролю та підтримки стану ментального здоров'я, виховання свідомості та проведення медитацій, різних фітнес-додатках та трекерах фізичної активності. Аналіз кількості встановлених додатків з різних категорій показав, що за 2019-2020 роки найбільш популярними були бізнес-додатки та додатки у категоріях «Медицина» і «Здоров'я та фітнес» [1].

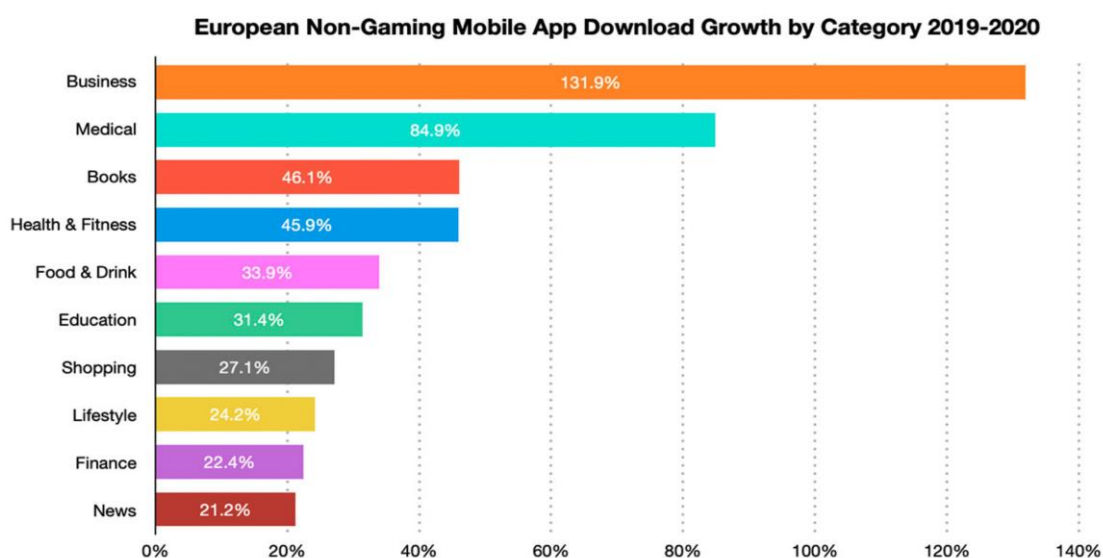


Рисунок 1.1 – Статистика завантажень мобільних додатків за категоріями у 2019 - 2020 році

Подібні результати ми можемо побачити і у статистиці за 2021 рік, що подана на рисунку 1.2[2].

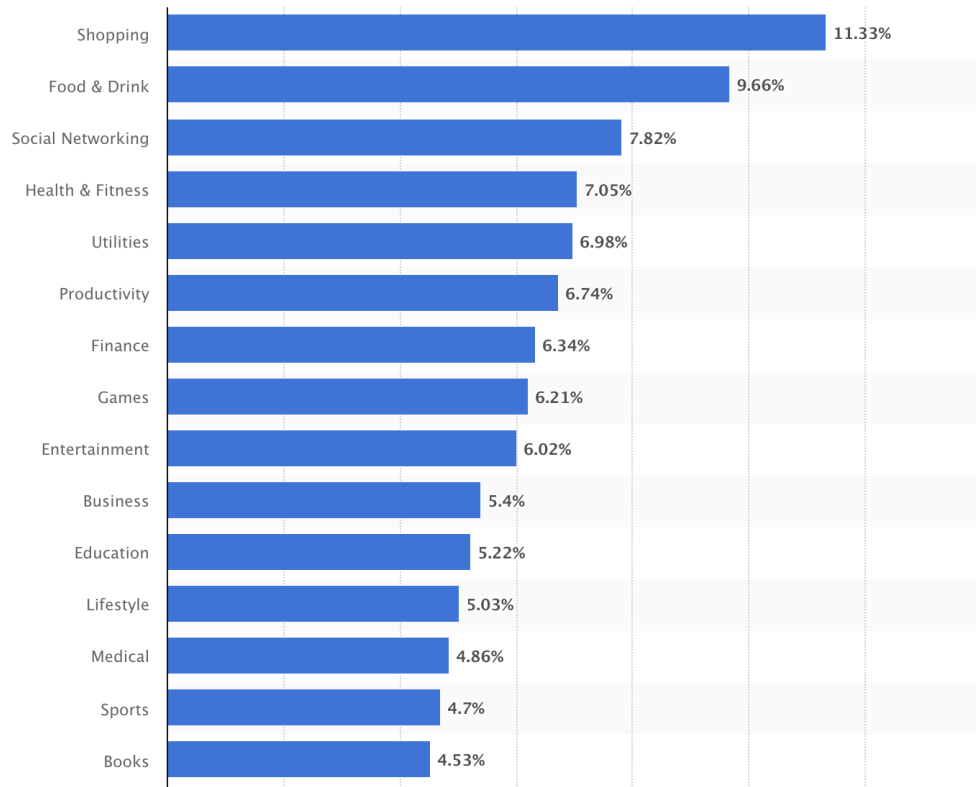


Рисунок 1.2 – Статистика завантажень мобільних додатків за категоріями у 2021 році

1.2 Постановка задачі

Обраною темою є створення додатку для відслідковування фізичної активності користувача та її аналізу. Основними функціональними вимогами при розробці додатку є:

- збір даних про користувача необхідних для аналітики;
- інтеграція із вбудованим додатком Apple Health[3] для отримання медичних показників користувача;
- зберігання даних у центральному репозиторію Apple Health;
- вибір типу тренування користувачем;
- експорт та імпорт показників, зібраних за тренування;
- аналітичний функціонал – інформація про тривалість тренування, подолану відстань, швидкість, кількість витрачених кілокалорій тощо;
- визначення рівня фізичної активності за допомогою аналітичних функцій;

- можливість зупинки тренування з подальшим відновленням без необхідності створювати нове тренування;
- можливість надсилати нагадування про тренування;
- можливість формувати аналітичний звіт про активність за період користування додатком;
- можливість керувати даними про тренування.

Основними вимогами до інтерфейсу є наявність:

- гарної та зручної навігації;
- екрану привітання нового користувача;
- екрану вводу персональних даних(ім'я, вага);
- екрану суміщення з Apple Health;
- екрану з переліком тренувань та випадającym списком для можливості фільтрування за певними характеристиками;
- екрану нового тренування, що має містити карту місцевості, опцію вибору типу тренування, дані про тривалість тренування, подолану відстань, швидкість та кількість витрачених калорій;
- екран редагування тренування;
- екрану перегляду показників певного тренування;
- екрану налаштування додатку.

1.3 Огляд існуючих аналогів

Сьогодні додатки у категорії “Фітнес та здоров’я” мають величезний попит. Мобільні додатки роблять моніторинг стану здоров’я та тренування набагато зручнішими для кожного з нас. Вони забезпечують нам підтримку у покращенні нашого самопочуття, дозволяючи відслідковувати свої показники в будь-який момент з будь-якого місця. Крім того, вони пропонують нам рекомендації щодо виконання конкретних вправ, корегують харчовий раціон в залежності від показників користувача та надають багато іншого корисного функціоналу.

Розглянемо найбільш популярні програмні продукти у даній категорії, їх графічні та функціональні можливості.

1. Strava[4]

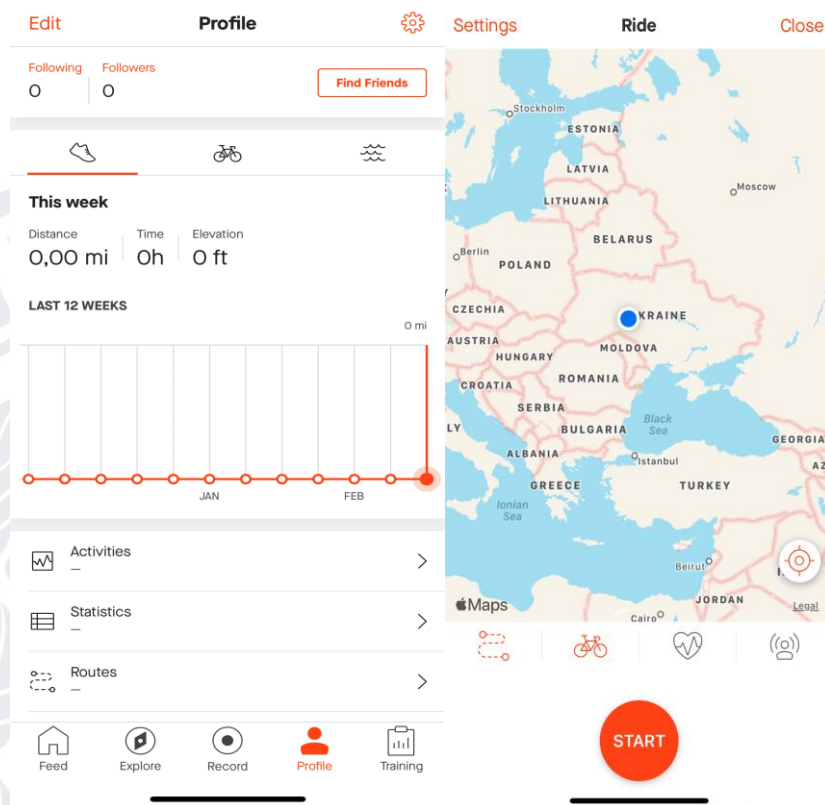


Рисунок 1.3 – Екрани додатку «Strava»

2. Relieve: Run, Ride, Hike & more[5]

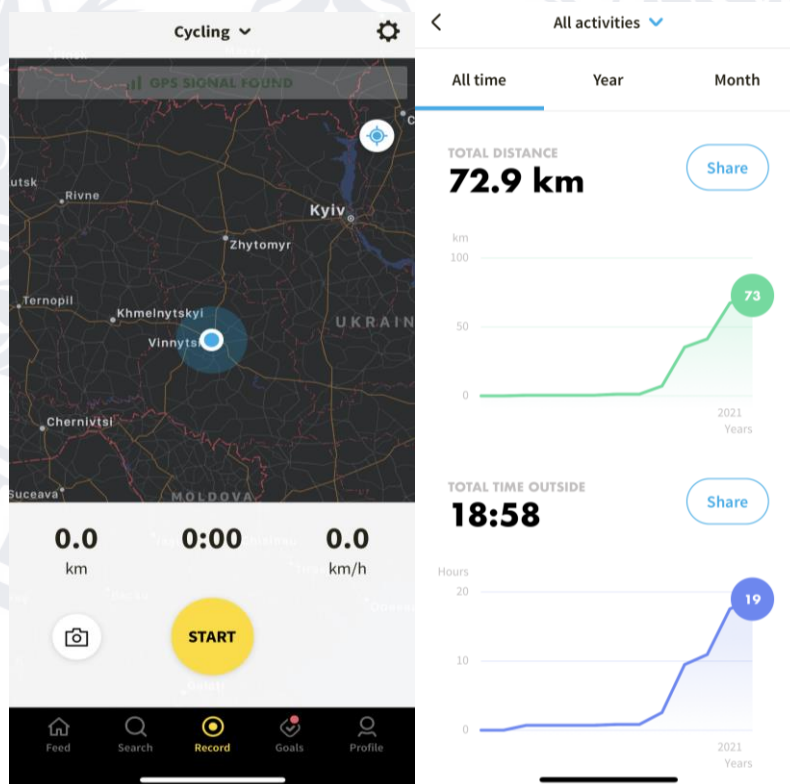


Рисунок 1.4 – екрани додатку «Relieve»

3. Run Track[6]

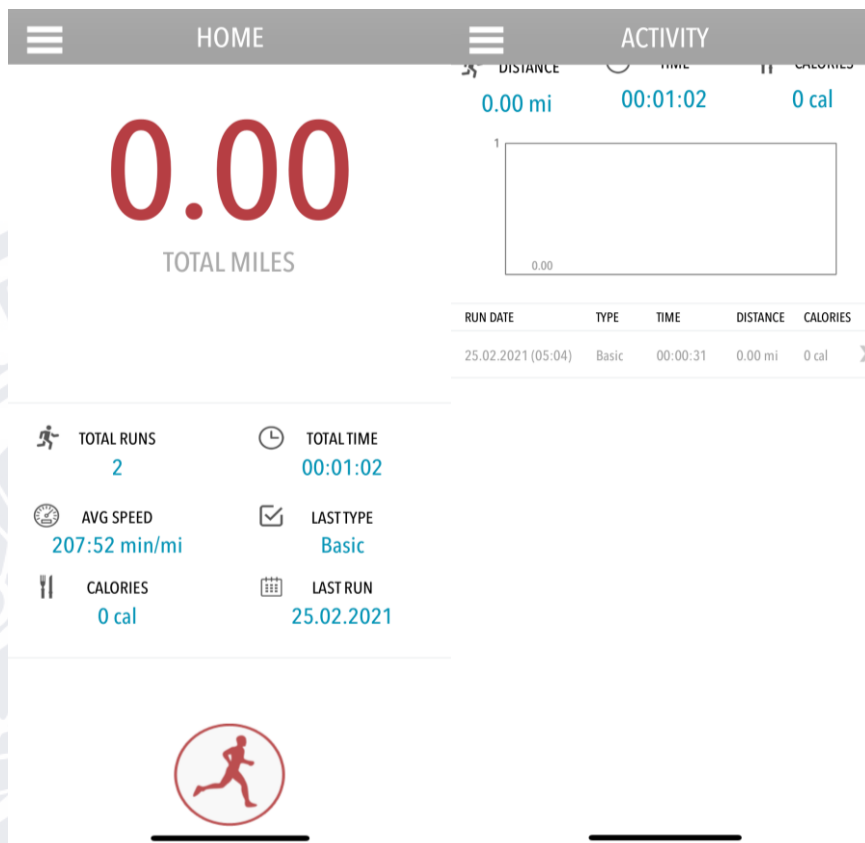


Рисунок 1.5 – екрани додатку «Run Track»

Порівняльну таблицю характеристик перелічених додатків зазначено у таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця характеристик популярних фітнес-додатків для відслідковування фізичної активності

	Strava	Relieve: Run, Ride, Hike & more	Run Track
Наявність екрану з інформацією про користувача	+	+	-
Відслідковування пройденого маршруту на карті	+	+	+

Інтеграція з Apple Health	+	-	-
Збір статистики про тренування	+	+	+
Можливість експорту та імпорту показників, зібраних за тренування	-	-	-
Можливість вибору типу тренування	+	+	-
Можливість вибору одиниць виміру	+	+	-
Визначення рівня фізичної активності	-	-	-
Можливість формування аналітичних звітів за період	-	-	-
Надсилання пуш-повідомлень із нагадуванням про тренування	-	-	-

Із таблиці можемо бачити, що двоє з трьох додатків не синхронізуються з Apple Health та не мають можливості імпорту інформації про користувача, що могло б допомогти при зборі статистики. У всіх перелічених додатках відсутня можливість експорту та імпорту інформації про тренування за допомогою спеціальних файлів з розширенням GPX. Також відсутня можливість визначення рівня фізичної активності, можливість формування аналітичних звітів за період та надсилання пуш-повідомлень із нагадуванням про тренування.

Висновок до розділу 1

Отже, у даному розділі було розглянуто актуальність створення додатку для відслідковування фізичної активності користувача, а також проаналізований ринок мобільних додатків для платформи iOS. Крім цього, були окреслені основні задачі та вимоги при розробці власного фітнес-додатку. На додаток, були детально проаналізовані існуючі аналоги та створено порівняльну таблицю їхніх можливостей.



РОЗДІЛ 2

ВІДОМОСТІ ПРО ВИКОРИСТАНІ ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ

2.1 Мова програмування

Тривалий час традиційною мовою для написання мобільних додатків для платформи iOS залишалась мова Objective-C[7]. Це підмножина мови програмування C, що надає об'єктно-орієнтовані можливості та динамічне середовище виконання. Objective-C переймає синтаксис, типи даних та інші конструкції від мови C, а також представляє новий синтаксис для визначення класів та методів. Проте у 2014 році компанія Apple оголосила про появу нової мови – Swift[8]. З цього моменту почалась нова ера мобільної розробки.

Swift – мультипарадигмова компільована мова програмування, що дозволяє створювати як frontend частини додатків, так і backend частини для будь-яких пристроїв екосистеми Apple. Станом на березень 2022 року Swift входить в 15 найпопулярніших мов програмування у світі за індексом TIOBE[9]. Рейтинг формується на основі обчислення кількості пошукових запитів, що містять згадку про мову.

Основні переваги мови Swift[10]:

- статична система типізації;
- відсутність розбиття коду на інтерфейси та реалізацію, що спрощує читабельність;
- інтеграція з Objective-C кодом;
- швидкість компілювання - за даними, що надає Apple, Swift швидший за Objective-C у 2.6 рази, а також швидший за Python 2.7 у 8.4 рази [11];
- підвищена читабельність коду - Swift відкидає багато застарілих умов, таких як крапка з комою до кінцевих рядків або дужки, які оточують умовні вирази;
- підтримка функціонального та реактивного програмування;
- наявність динамічних бібліотек, що швидко оновлюються та є стабільними;

- підтримка крос-платформної розробки та перспективи у даній області.

Swift дозволяє створювати нативні(англ. native - рідний) для платформи iOS додатки[12]. Оскільки нативні програми спеціально розроблені для певного пристрою, вони мають повну свободу використовувати технічний функціонал, наявний на пристрої, такий як камера, список контактів, GPS і Bluetooth. Нативні програми працюють на операційній системі пристрою. Простіше кажучи, нативні програми вимагають повного доступу до всього апаратного забезпечення та функціональних можливостей пристрою та працюють безпосередньо на пристрої.

Платформи надають розробникам програм стандартизований набір для розробки програмного забезпечення(англ. Source Development Kit), набір, що містить інструменти, зразки коду, бібліотек, документації та посібників, які дозволяють розробникам створювати програми на певній платформі.

Серед переваг нативної розробки варто виокремити:

- швидкість – оскільки нативні програми є рідними для платформи, вони працюють швидше; багато елементів поставляються попередньо завантаженими;
- можливість працювати в автономному режимі;
- забезпечують впізнаваний зовнішній вигляд.

2.2 Інструмент для побудови дизайну додатку

Під час створення програмного додатку було використано інструмент для побудови дизайну додатку – Figma [13, 14]. Figma – це веб-додаток для створення графіки та дизайну інтерфейсу користувача. Його можна використовувати для побудови будь-яких видів графічного дизайну веб-сайтів, розробки інтерфейсів мобільних додатків, створення прототипів, створення публікацій у соціальних мережах і всього, що між ними.

Figma відрізняється від інших інструментів для редагування графіки в основному тим, що вона працює безпосередньо у браузері. Це означає, що можна отримати доступ до своїх проектів і починає проектування з будь-якого

комп'ютера або платформи без необхідності купувати декілька ліцензій або встановлювати програмне забезпечення.

Основними перевагами даного інструменту є:

- веб-орієнтований, що означає, що він працює майже на будь-якій операційній системі з сучасним браузером, включаючи Linux, ChromeOS, Windows і macOS;
- можливість співпраці в режимі реального часу з іншими дизайнерами над тими ж файлами дизайну;
- передача, перевірка, прототип і багато іншого в рамках одного інструменту з використанням додатків або без них;
- гнучкі стилі, компоненти та заміни, які дозволяють швидко виконувати ітерації;
- нова екосистема додатків, яка дозволить реалізувати майже все, що задумано;
- надійний вбудований контроль версій, який дозволяє зберігати та додавати описи до певних версій.

2.3 Середовище розробки

Для розробки мобільного додатку засобами мови Swift необхідне інтегроване середовище розробки(IDE). Компанія Apple надає безкоштовне IDE – Xcode 13 [15]. Даний програмний продукт доступний для завантаження через магазин App Store. Альтернативним IDE виступає AppCode – програмний продукт, запропонований компанією JetBrains [16]. Проте, станом на 2022 рік, AppCode все ще є залежним від інструментів та системи збирання проєктів Xcode, що унеможлиблює самостійну роботу даного IDE.

Xcode містить в собі значну частину документації створеної компанією Apple для розробників, а також низку засобів та інструментів для успішної розробки, тестування та налагодження додатку. Interface Builder [17] – інтегрований застосунок, що використовується для розробки графічних інтерфейсів(GUI). Він дозволяє створювати екрани мобільного додатку без

написання коду. Для цього необхідно розмістити та задати налаштування всіх необхідних елементів та пов'язати з файлами реалізації, де описується логіка взаємодії з ними - контролерами. Ілюстрація роботи даного інструменту представлена на рисунку 2.1.

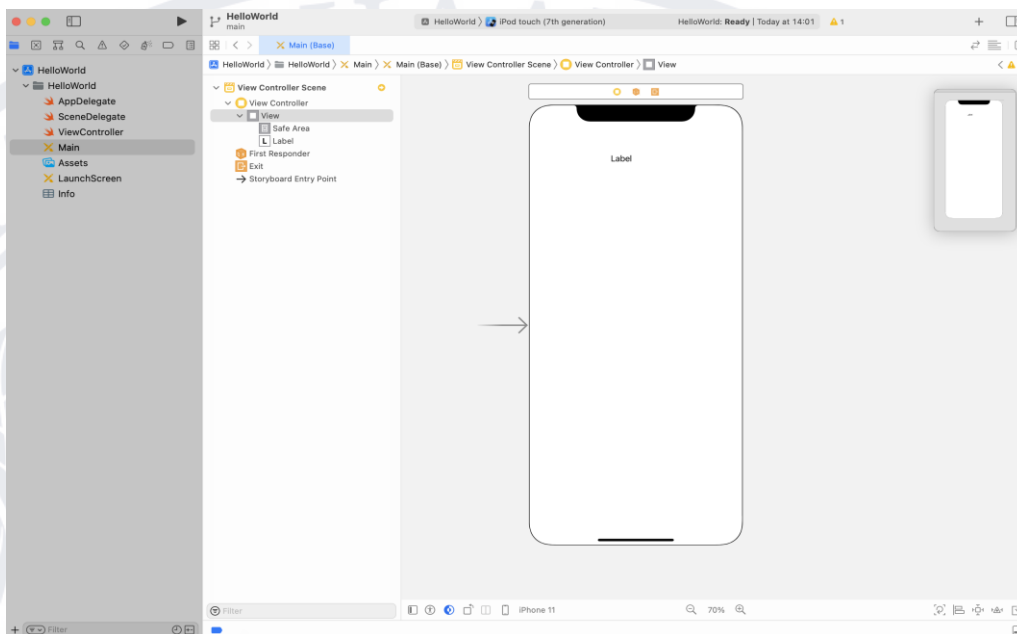


Рисунок 2.1 – Ілюстрація інтерфейсу Interface Builder

Середовище дає можливість запуску додатку без фізичного пристрою. Для цього використовуються симулятори (рис. 2.2). З їхньою допомогою легко можна протестувати додаток на наявність помилок та подивитись його виконання на практично будь-якому пристрої. Також середовище надає можливість напряму запускати та тестувати застосунки на власному мобільному пристрої.

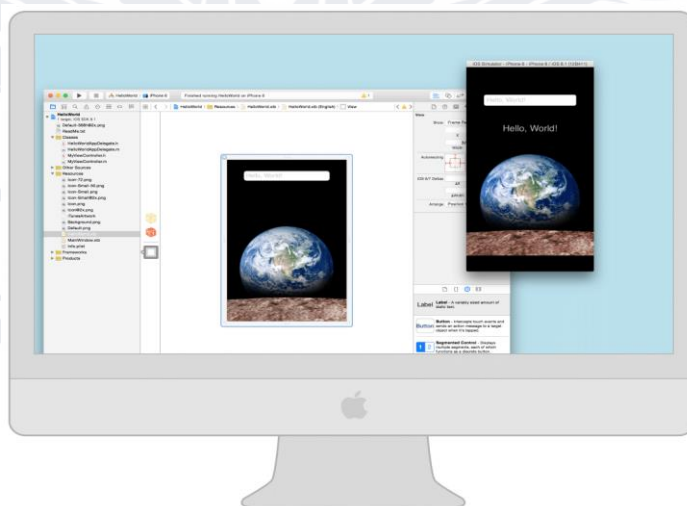


Рисунок 2.2 – Ілюстрація роботи симулятора

Крім цього, Xcode містить інструментарій для визначення помилок, як синтаксичних, так помилок, що виникають під час роботи додатку. Також присутня вбудована система контролю версій GIT, функція імітації геолокації, необхідна для зручної роботи з картами, Playgrounds [18] – середовище, що дозволяє протестувати частину коду без необхідності компіляції цілого проекту.

Xcode – середовище розробки програмного забезпечення, за допомогою якого можна створити додаток будь-якого рівня складності. Серед головних його переваг варто виокремити регулярне оновлення, офіційну підтримку від компанії Apple та наявність обширної та гарно структурованої документації.

2.3 Менеджер залежностей

Існує декілька менеджерів залежностей, які використовуються під час створення мобільного додатку для платформи iOS. У даній роботі використовується найбільш популярний – менеджер CocoaPods [19, 20].

Cocoapods працює з допомогою командного рядка, а також даний інструмент міститься в середовищі розробки AppCode. CocoaPods рекурсивно встановлює залежності між бібліотеками, отримує їхній вихідний код, а також створює та підтримує робочу область Xcode для побудови проекту. Назви всіх доданих бібліотек вказуються у єдиному текстовому файлі «Podfile». Окрім інсталяції з багатьох різних джерел, «головне» сховище специфікацій, що містить метадані багатьох бібліотек з відкритим вихідним кодом, підтримується як GIT-репозиторій і розміщується на платформі GitHub [21].

Podfile для даного проекту зображений на рисунку 2.3.

```
target 'FitnessTracker' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  # Pods for FitnessTracker

  pod 'SnapKit', '~> 5.0.1'
  pod 'Charts', '~> 3.6.0'
  pod 'CoreGPX', '~> 0.8.1'
  pod 'Cache', '~> 6.0.0'
end
```

Рисунок 2.3 – «Podfile» проекту, що розробляється

2.4. Використані бібліотеки

Для розробки заданого мобільного додатку було використано низку бібліотек.

UIKit [22] – бібліотека, що надає основні об'єкти, необхідні для створення додатків більшості операційних систем від Apple. Ці об'єкти використовуються для відображення вмісту на екрані, взаємодії з цим вмістом та керування взаємодією з системою. Також розробник забезпечений інструментарієм для управління взаємодією між користувачем, системою та створюваним додатком. Фреймворк пропонує такі функції, як анімація, підтримка документів, малювання і друк, надання інформації про поточний пристрій, управління текстом, підтримка пошуку та управління ресурсами та багато іншого.

CoreLocation [23] – бібліотека для відслідковування географічного розташування пристрою, висоти розташування та орієнтації пристрою. Даний фреймворк збирає дані за допомогою всіх доступних на пристрої компонентів, включаючи Wi-Fi, GPS, Bluetooth, магнітометр, барометр та мобільний зв'язок. На пристроях iOS користувачі можуть будь-коли змінити налаштування служби визначення місцезнаходження в налаштуваннях, при цьому деякі функціональні частини додатку зупинять роботу.

CoreMotion [24] – бібліотека, що повідомляє дані про рух і навколишнє середовище з бортового обладнання пристроїв iOS, в тому числі з акселерометрів і гіроскопів, а також з крокоміра, магнітометра і барометра. Розробники використовують цю структуру для доступу до даних, згенерованих апаратним забезпеченням, для використання цих даних у додатку. Наприклад, гра може використовувати дані акселерометра та гіроскопа, щоб контролювати поведінку гри на екрані. Багато служб цього бібліотеки дозволяють отримати доступ як до необроблених значень, записаних апаратним забезпеченням, так і до обробленої версії цих значень. Оброблені значення не включають форми упередження, які можуть негативно вплинути на те, як ви використовуєте ці дані. Наприклад, оброблене значення акселерометра відображає лише прискорення, викликане користувачем, а не прискорення, викликане силою тяжіння.

MapKit [25] – бібліотека для роботи з картами. Даний фреймворк дозволяє з легкістю інтегрувати карти у створюваний додаток та задати всі необхідні налаштування.

CoreData [26] – бібліотека, що допомагає зберігати дані на мобільний пристрій. Іншими словами, це фреймворк, який управляє графом об'єктів. Core Data забезпечує стале зберігання даних на диску, що означає, що вони будуть доступні навіть після завершення роботи додатка або виключення пристрою. Це відрізняється від сталості в тимчасовій пам'яті, яке буде зберігати дані тільки до тих пір, поки ваш додаток працює або на передньому плані, або в фоновому режимі.

HealthKit [27] – бібліотека, що надає доступ до центрального репозиторію, де зберігаються особисті дані користувача пристрою та його фізичні показники. Фреймворк дозволяє збирати та зберігати дані про здоров'я та фізичну активність, аналізувати та візуалізувати ці дані. HealthKit також призначений для керування та об'єднання даних із кількох джерел. Наприклад, користувачі можуть переглядати всі свої дані та керувати ними в додатку Health, зокрема додавати дані, видаляти дані та змінювати дозволи програми. Таким чином, ваша програма повинна обробляти ці зміни, навіть якщо вони відбуваються за межами вашої програми.

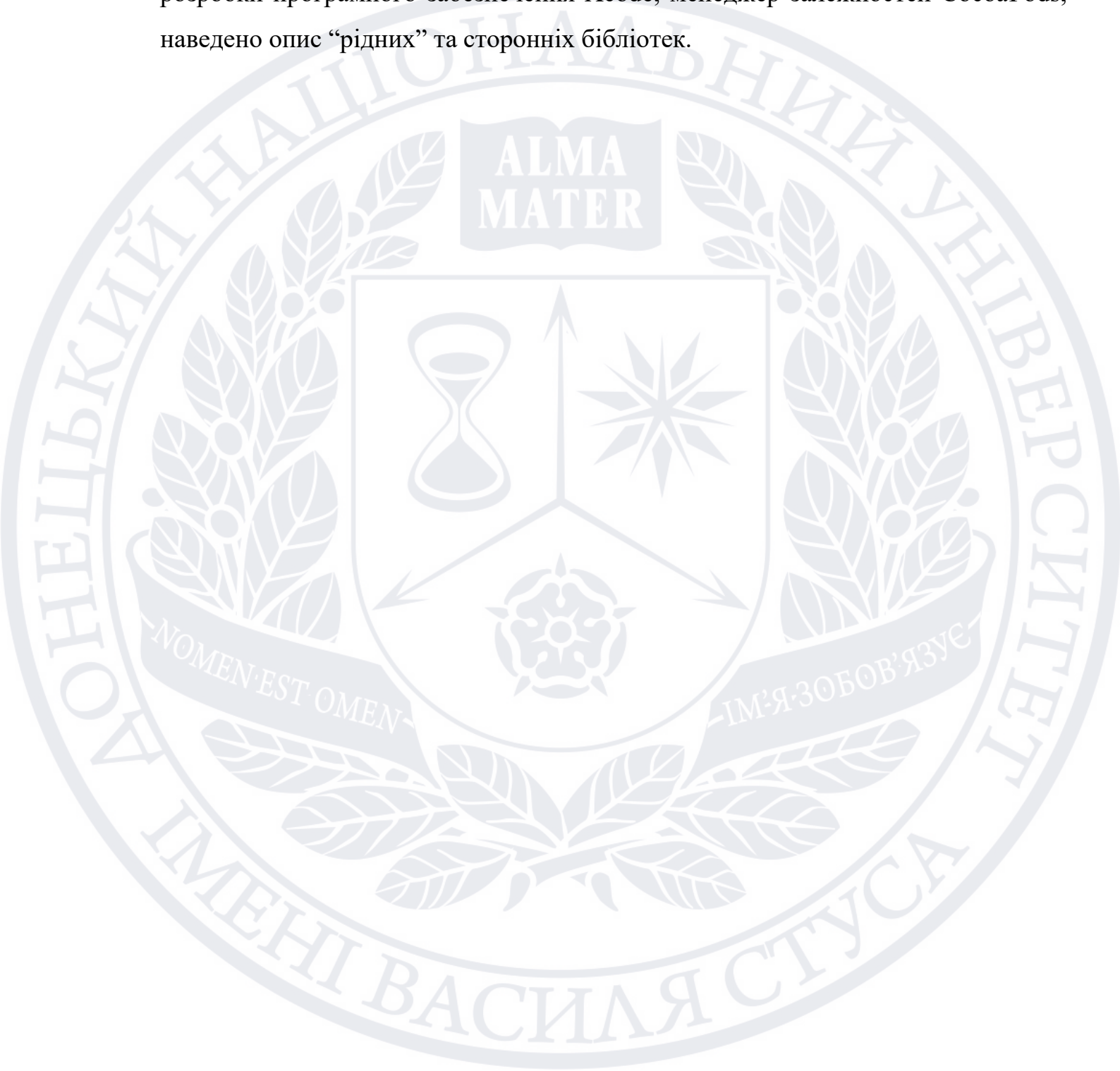
Charts [28] – стороння бібліотека, що допомагає будувати та відображати графіки різних типів.

CoreGPX [29] – стороння бібліотека, що надає можливість реалізувати експорт та імпорт даних у форматі gpx.

SnapKit [30] – стороння бібліотека, що задає правила відображення графічних об'єктів на екранах пристроїв. SnapKit – це доменна мова, яка спрощує автоматичне розміщення графічних об'єктів як на iOS, так і на OS X. Простий і виразний ланцюжок доменної мови дозволяє створювати обмеження з мінімальною кількістю коду, гарантуючи, що його легко читати та розуміти. Використовуючи SnapKit, є можливість створювати, оновлювати, видаляти та керувати обмеженнями макета переглядів інтерфейсу користувача.

Висновки до розділу 2

У даному розділі були розглянуті використані під час розробки інструменти та технології, зокрема мова програмування Swift, інтегроване середовище розробки програмного забезпечення Xcode, менеджер залежностей CocoaPods, наведено опис “рідних” та сторонніх бібліотек.



РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1 Опис шаблону проектування

Сьогодні існує велика кількість різноманітних шаблонів проектування. Вибір шаблону залежить від типу програмного продукту, що розробляється. Одним із базових для розробки під платформу iOS є шаблон Model-View-ViewModel(або Модель-Представлення-Модель Представлення)[31]. Схема взаємодії компонентів даної версії зображена на рисунку 3.1.

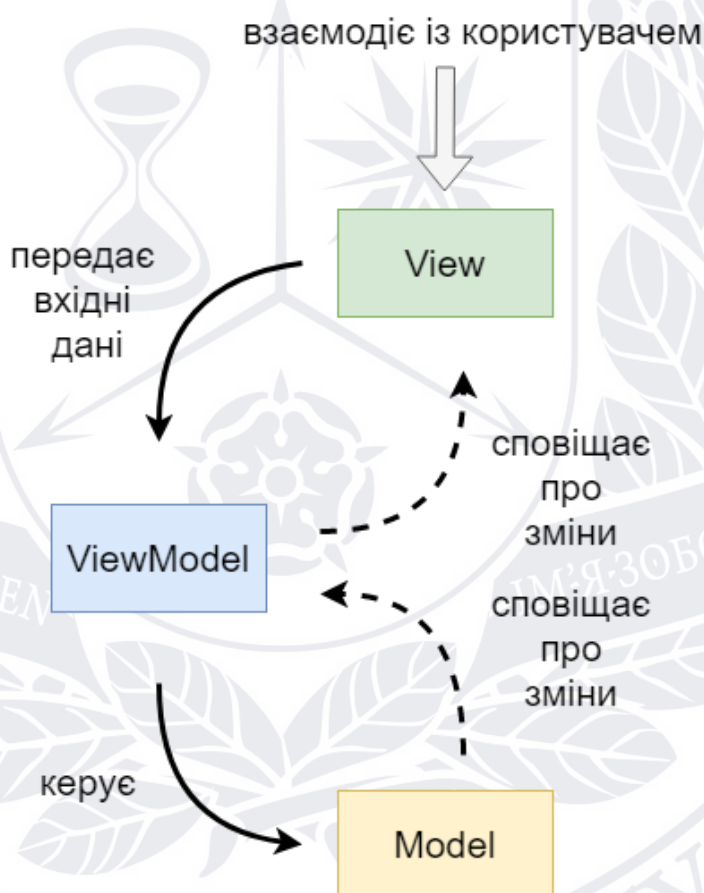


Рисунок 3.1 – Структура шаблону Model-View-ViewModel

Шаблон Model-View-ViewModel допомагає чітко розділити бізнес-логіку та логіку презентації програми з інтерфейсом користувача (UI). Підтримка чіткого поділу між логікою програми та інтерфейсом користувача допомагає вирішити численні проблеми розробки та може полегшити тестування, відлагодження та розвиток програми. Це також може значно покращити можливості повторного

використання коду та дозволяє розробникам і дизайнерам інтерфейсу легше співпрацювати під час розробки відповідних частин програми.

Модель зберігає дані та пов'язану логіку. Це дані, що передаються між частинами контролера або будь-якою іншою пов'язаною бізнес-логікою. Наприклад, об'єкт отримує інформацію із бази даних. Він маніпулює даними і відправляє їх назад до бази даних або використовує для відтворення тих самих даних.

Представлення — це набір видимих елементів, який також отримує введення користувача. Воно включає в себе інтерфейси користувача (UI), анімацію та текст.

Модель представлення (View Model) розташована між шарами View і Model. Тут розміщуються елементи керування для взаємодії з View, тоді як зв'язування використовується для підключення елементів інтерфейсу користувача у View до елементів керування ViewModel.

Шаблон MVVM має низку переваг [32]:

- розділення різних типів коду має полегшити доступ до однієї або кількох більш детальних і сфокусованих частин і вносити зміни без зайвих зусиль.
- Простіше оновлювати програму;
- кожен фрагмент коду є більш детальним, і якщо він реалізований правильно, зовнішні та внутрішні залежності знаходяться в окремих частинах коду від частин з основною логікою, яку можна перевірити;
 - можливість зробити будь-яку з частин більш придатною для повторного використання;
 - рівень презентації та логіка слабо пов'язані;
 - бізнес-логіка відокремлена від View;
 - підтримує розробку на основі тестування (Test Driven Development);

Проте у шаблона є певні недоліки. Серед них:

- відлагодження програми буде трохи складним, якщо у існують складні прив'язки даних;
- складнощі при розробці частини ViewModel.

При розробці мобільного фітнес-додатку був використаний саме MVVM шаблон, оскільки широко застосовується при функціонально-реактивному програмуванні [33].

Структура робочого проекту представлена на рисунку 3.2.

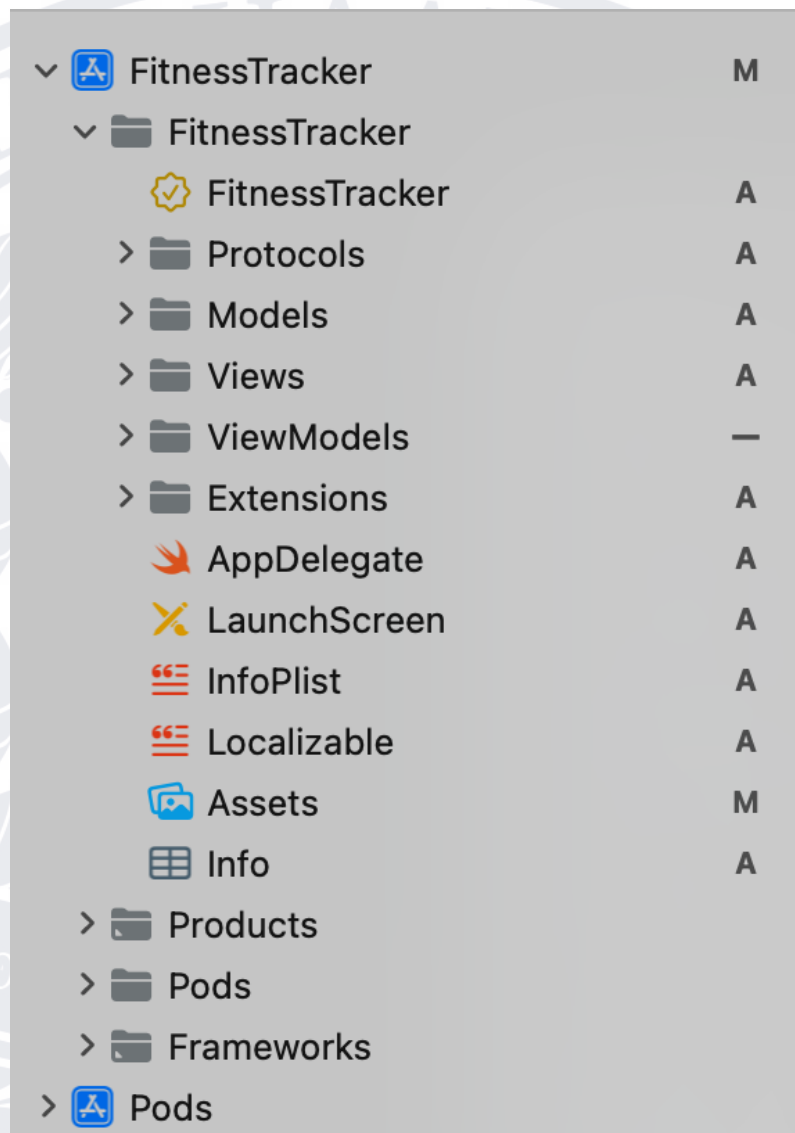


Рисунок 3.2 – Структура робочого проекту

Як бачимо, проект складається із декількох частин – файлів із програмним кодом та файлів-налаштувань. Файли із програмним кодом розділені на 4 групи – протоколи(Protocols), моделі(models), представлення(Views) та групи розширень(Extensions). Окрім цього ми бачимо стандартний клас AppDelegate, що відповідає за ініціалізацію центральних структур додатку, налаштування сцен, керування системними сповіщеннями, реєстрацію будь-яких необхідних служб під час запуску тощо. Також проект містить групу ресурсів(Assets), що

використовуються для написання додатку. Це можуть бути картинки, медіа-файли тощо.

Окрім цього важливою складовою проекту є файл Info.plist або файл списку інформаційних властивостей. Це структурований текстовий файл, який містить важливу інформацію про конфігурацію для пов'язаного виконуваного файлу. Сам файл зазвичай кодується за допомогою кодування Unicode UTF-8, а вміст структурується за допомогою XML. Кореневий вузол XML – це словник, вміст якого – це набір ключів і значень, що описують різні аспекти пакета. Система використовує ці ключі та значення для отримання інформації про програму та її налаштування. Як результат, очікується, що всі пов'язані виконувані файли (плагіни, фреймворки та програми) мають файл зі списком інформаційних властивостей. Розглянемо вміст даного файлу, що містить проект розроблюваного додатку (рис. 3.3).

Key	Type	Value
Information Property List	Dictionary	(23 items)
Localization native development region	String	\$(DEVELOPMENT_LANGUAGE)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	\$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Bundle version string (short)	String	1.0
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Privacy - Health Share Usage Description	String	This app needs access to your health data to save workouts constructed by this app
Privacy - Health Update Usage Description	String	This app needs access to your health data to read and possible correct workouts constructed by this app
Privacy - Location Always and When In Use Usage Des...	String	This app needs your location to track workouts
Privacy - Location Always Usage Description	String	This app needs your location to track workouts
Privacy - Location When In Use Usage Description	String	This app needs your location to track workouts
Privacy - Motion Usage Description	String	This app needs motion related tracking capabilities to record data like steps during a workout
Application supports indirect input events	Boolean	YES
> Required background modes	Array	(1 item)
Launch screen interface file base name	String	LaunchScreen
> Required device capabilities	Array	(1 item)
> Supported interface orientations	Array	(1 item)
> Supported interface orientations (iPad)	Array	(4 items)
> Exported Type Identifiers	Array	(2 items)
> Imported Type Identifiers	Array	(2 items)

Рисунок 3.3 – Системний файл Info.plist

Як бачимо, файл містить інформацію про конфіденційність(Privacy) – додаток запитуватиме користувача, чи згоден він надати ту чи іншу інформацію про себе. Окрім того, файл містить безпосередні налаштування проекту – ідентифікатори проекту, підтримувані орієнтації екрану для мобільних пристроїв та планшетів та інше.

3.2 Використання UML для проектування додатку

UML, скорочення від Unified Modeling Language — це стандартизована мова моделювання, що складається з інтегрованого набору діаграм, розроблених для допомоги розробникам систем і програмного забезпечення для визначення, візуалізації, конструювання та документування артефактів програмних систем, а також для бізнес-моделювання та інші непрограмні системи [34]. Це набір найкращих інженерних практик, які виявилися успішними в моделюванні великих і складних систем. Ця технологія є дуже важливою частиною розробки об'єктно-орієнтованого програмного забезпечення. UML використовує переважно графічні позначення для опису архітектурного дизайну програмних проектів.

Одним з найбільш популярних типів діаграм в UML є діаграма класів. Популярні серед інженерів-програмістів для документування архітектури програмного забезпечення, діаграми класів є різновидом структурної діаграми, оскільки вони описують те, що має бути присутнім в системі, що моделюється.

UML була створена як стандартизована модель для опису підходу до об'єктно-орієнтованого програмування. Оскільки класи є будівельним блоком об'єктів, діаграми класів є будівельними блоками UML. Різні компоненти в діаграмі класів можуть представляти класи, які фактично будуть запрограмовані, основні об'єкти або взаємодії між класами та об'єктами.

Сама форма класу складається з прямокутника з трьома рядками. Верхній рядок містить назву класу, середній рядок містить атрибути класу, а нижній розділ містить методи чи операції, які може використовувати клас. Класи та підкласи згруповані разом, щоб показати статичні відносини між кожним об'єктом.

Оскільки робочий проект містить велику кількість класів, розглянемо кілька основних його частин. На рисунку 3.4 зображена частина діаграми класів, що відповідає за представлення моделі тренування та виконання всіх операцій, пов'язаних із даною областю.

Окремо слід зазначити використання шаблону делегування. Делегати надзвичайно поширені в iOS розробці. Делегат — це будь-який об'єкт, про який слід сповістити, коли сталась певна подія. Наприклад, делегат перегляду таблиці отримує сповіщення, коли користувач торкається рядка, тоді як делегат контролера навігації отримує сповіщення, коли користувач переміщається між контролерами представлення.

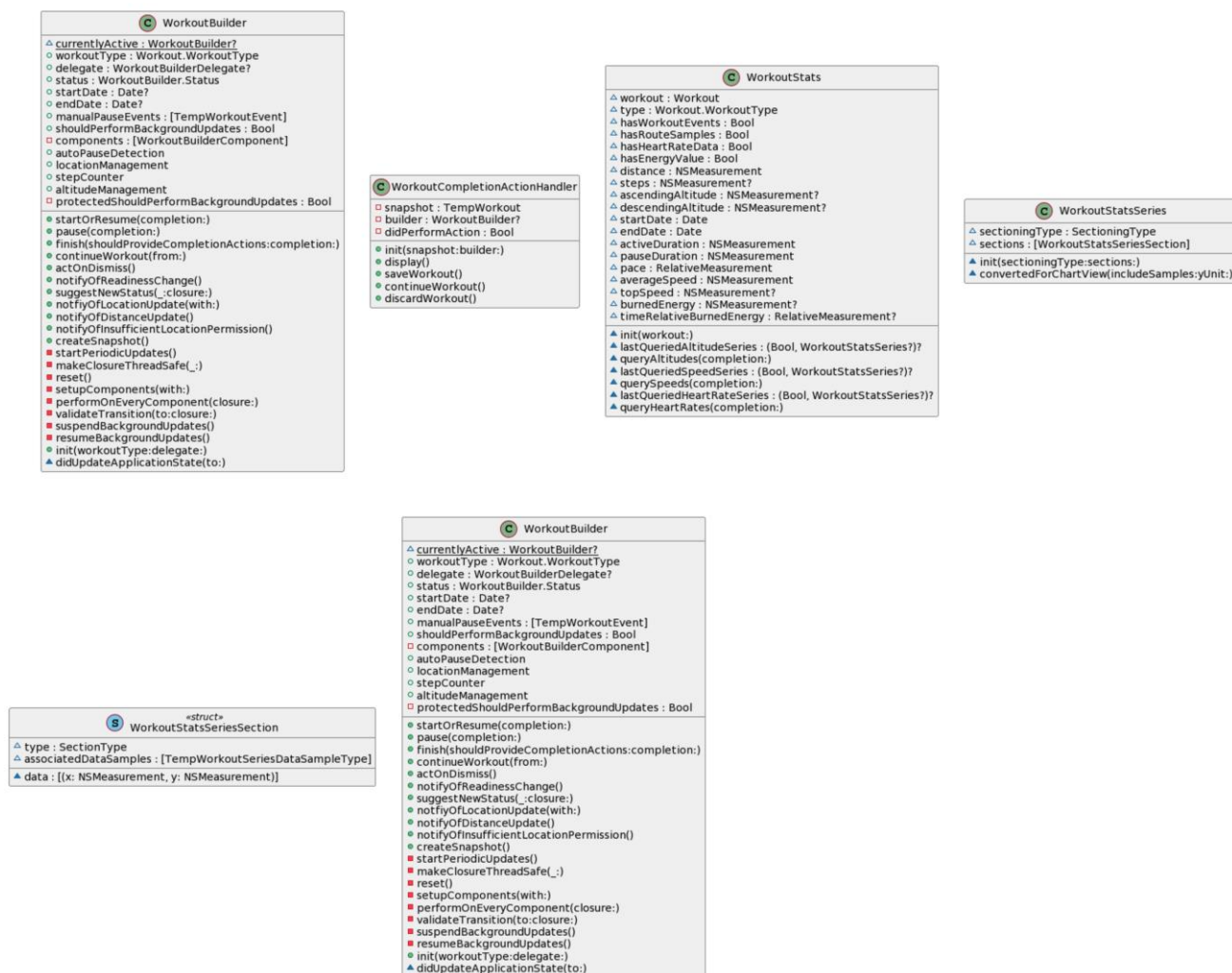


Рисунок 3.4 – Діаграма класів моделі побудови тренувань

Діаграма класів, що відповідають за відображення деяких елементів на екрані, зображена на рисунку 3.5. На даній діаграмі зображені класи, що відповідають за створення нових тренувань, відображення карти, відображення історії тренувань, а також перегляду інформації про конкретне тренування. Окрім того, на діаграмі вказані компоненти, від яких дані класи наслідуються.

Ця залежність відображена стрілкою, що виходить з контролера та направлена на клас або протокол, від якого даний контролер наслідується.

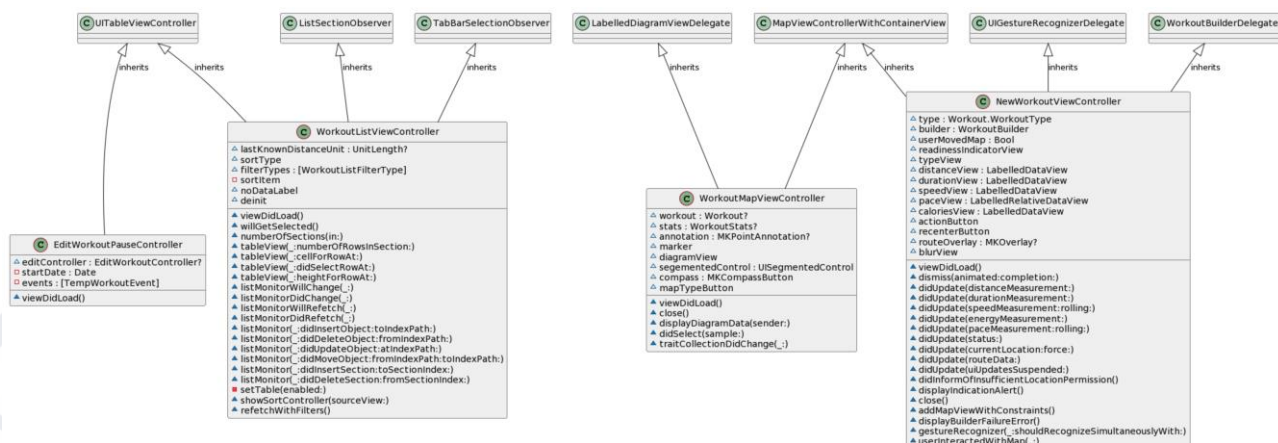


Рисунок 3.5 – Діаграма класів деяких контролерів

Висновки до розділу 3

У даному розділі було розглянуто використовуваний при розробці додатку шаблон проектування, описано його структуру та особливості. Для проектування архітектури додатку була використана мова UML, з допомогою якої було сформовано декілька діаграм класів.

РОЗДІЛ 4

ОПИС ФУНКЦІОНАЛУ ДОДАТКУ

4.1 Опис користувацького інтерфейсу

Кожна платформа має власну індивідуальність. Завжди краще розробляти елементи, які відповідають платформі. Це спрощує вивчення програми користувачами, оскільки вони будуть знайомі з процесом використання інших програм.

Гарно продуманий та зручний графічний інтерфейс є запорукою успіху будь-якого додатку та гарантом збільшення рівня залучення користувача у програмний продукт. Існує певний набір правил та обмежень, що допомагають дизайнерам розробляти правильну візуальну структуру. Apple пропонує низку вказівок щодо дизайну серед яких [35]:

- естетична цілісність – відображає, на скільки зовнішній вигляд та поведінка програми інтегруються з її функціональністю. Наприклад, програма, яка допомагає людям виконувати серйозне завдання, може тримати їх зосередженими, використовуючи тонку, ненав'язливу графіку, стандартні елементи керування та передбачувану поведінку;
- узгодженість – використання звичних людському оку елементів, системних елементів інтерфейсу, добре відомих піктограм, стандартних стилів тексту та єдиної термінології тощо;
- додаток має таку поведінку, яку очікує користувач (наприклад, при повороті екрану, контент відображуваної сторінки змінює своє розташування);
- зворотний зв'язок фіксує дії та показує результати, щоб інформувати людей. Додатки, що надає компанія Apple, це найкраще ілюструють. Інтерактивні елементи можуть підсвічуватись під час натискання, операційні індикатори повідомляють про їх стан, а анімація та звук дають можливість зрозуміти результати дій;

- люди навчаються значно швидше, коли об'єкти програми є метафорами знайомого досвіду, незалежно від того, пов'язані вони з реальним чи цифровим світом. Наприклад, перемикачі, що дозволяють керувати певними функціями, перетягування вмісту, переміщення повзунків або гортання сторінок документів;
- ясність – дизайн повинен спрямовувати. Потрібно намагатися уникати усього, що може заплутати користувача;
- пряма маніпуляція – підтримка анімації, жестів, того, що безпосередньо залучає користувача та реагує на його рухи. Прикладом прямого маніпулювання є обертання екрана;
- користувальницький контроль – додаток спрямовує користувача, попереджає про помилки, але ніколи не контролює користувача. Зворотний зв'язок показує користувачеві результат і захищає від потенційних небезпек. Але надання всіх повноважень додатку не додає впевненості користувачеві. Наприклад, у програмі, якій потрібен доступ до камери, вікно сповіщення попереджає, що «для використання програми потрібен доступ до камери». Це дає користувачу чітке розуміння, що буде використовуватись. Але це також дає можливість заборонити доступ. Користувач контролює – отже, додаток надає впевненості.

Додаток розпочинає свою роботу із екрану привітання нового користувача, зображеного на рисунку 4.1. Екран привітання має містити коротку та чітку інформацію про додаток та має зацікавити користувача. Розроблений додаток має лаконічний дизайн екрану привітання. Він містить коротку інформацію про те, для чого саме призначений програмний продукт та головні його особливості:

- висока точність відслідковування фізичної активності – бігу, ходьби, велосипедний спорт;
- деталізована статистика та аналітика тренувань, наприклад, найвища або середня швидкість;

- захищеність даних - ключовою особливістю додатку є те, що всі дані користувача знаходяться на його пристрої або в захищеному середовищі iCloud [36].

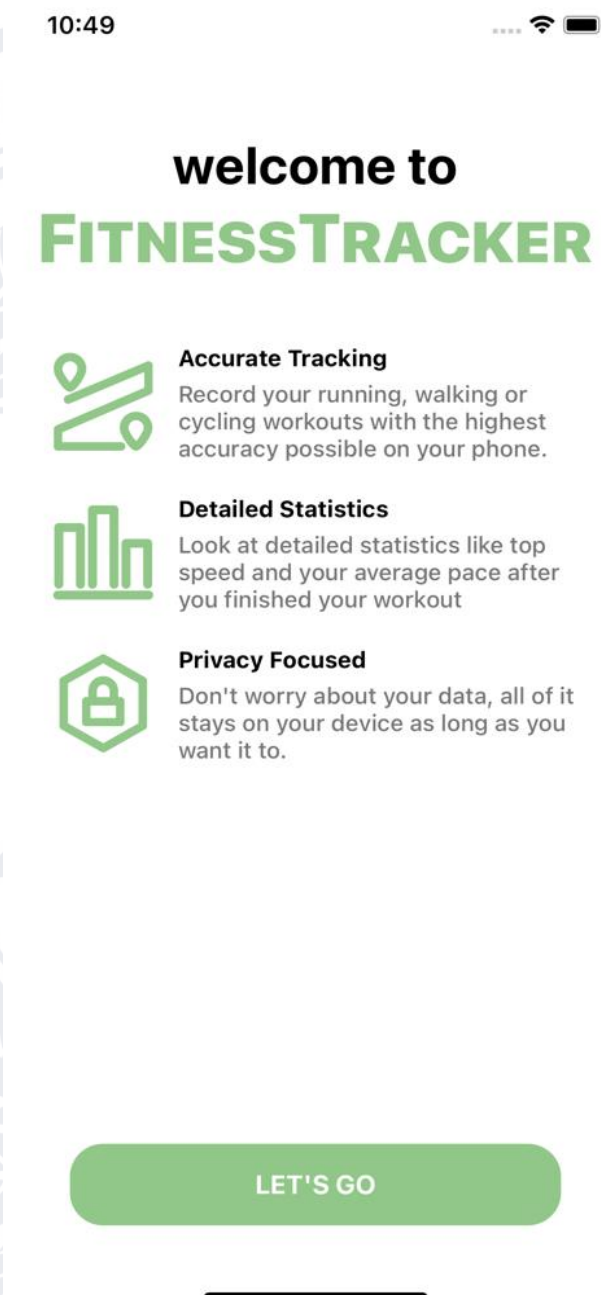


Рисунок 4.1 – Екран привітання користувача

Наступним кроком є ввід даних користувача, таких як ім'я та вага (рис. 4.2). Також є можливість обрати одиниці виміру – кілограми або фунти. Ім'я буде застосовуватись в інтерфейсі для звернення до користувача, проте є можливість пропустити даний етап та зазначити його та інші дані пізніше у налаштуваннях додатку. Вага використовуватиметься для визначення персонального рівня

фізичної підготовки та рекомендацій щодо тренувань. Всі ці дані зберігаються на пристрої користувача та є надійно захищеними.

12:19

Let's talk about You

We need some information about You

Username Name

Preferred System Metric Imperial

Weight Weight lb

Next

Рисунок 4.2 – Екран вводу персональної інформації

Наступний екран(рис. 4.3) пропонує користувачу визначити, чи хоче він синхронізувати роботу додатку із системним додатком Health. Це надасть можливість зберігати дані в центральному репозиторію HealthKit, а також проводити аналітику за тренуваннями, що були записані за допомогою самого додатку HealthKit, тобто користувач надає дозвіл на автоматичний імпорт інформації про тренування. Екран містить детальну інформацію, які переваги надасть дана синхронізація, як зберігається інформація тощо. Робота додатку

націлена на прозорість дій – користувач завжди проінформований про те, як відбувається «на фоні» додатку.

10:28



Want to sync with Apple Health?

If you want your data to be stored in a central space Apple Health might be a good option since it's well integrated with iOS. The synchronization is totally optional though and it should be noted that if you choose to sync with Apple Health your data will possibly leave your phone and be stored on Apple Servers, more on that in the privacy policy. If you choose to enable Auto Import the app tries to import workouts added by other apps into the local database so you keep an overview over all your workouts.

Sync Workouts



Sync Weight



Auto Import Workouts



Next

Рисунок 4.3 – Екран надання дозволу на синхронізацію

Далі користувачу пропонується екран (рис. 4.4), де необхідно надати дозволи на використання такої інформації, як геолокація, доступ до характеристик здоров'я та фізичної активності та дозвіл використання системного функціоналу – крокоміру, гіроскопу тощо. Ці дозволи необхідні для виконання додатком своїх основних функцій.

Користувач в будь-яку хвилину зможе скорегувати дозволи на використання тієї чи іншої інформації за допомогою меню додатку або ж за допомогою системних налаштувань пристрою.

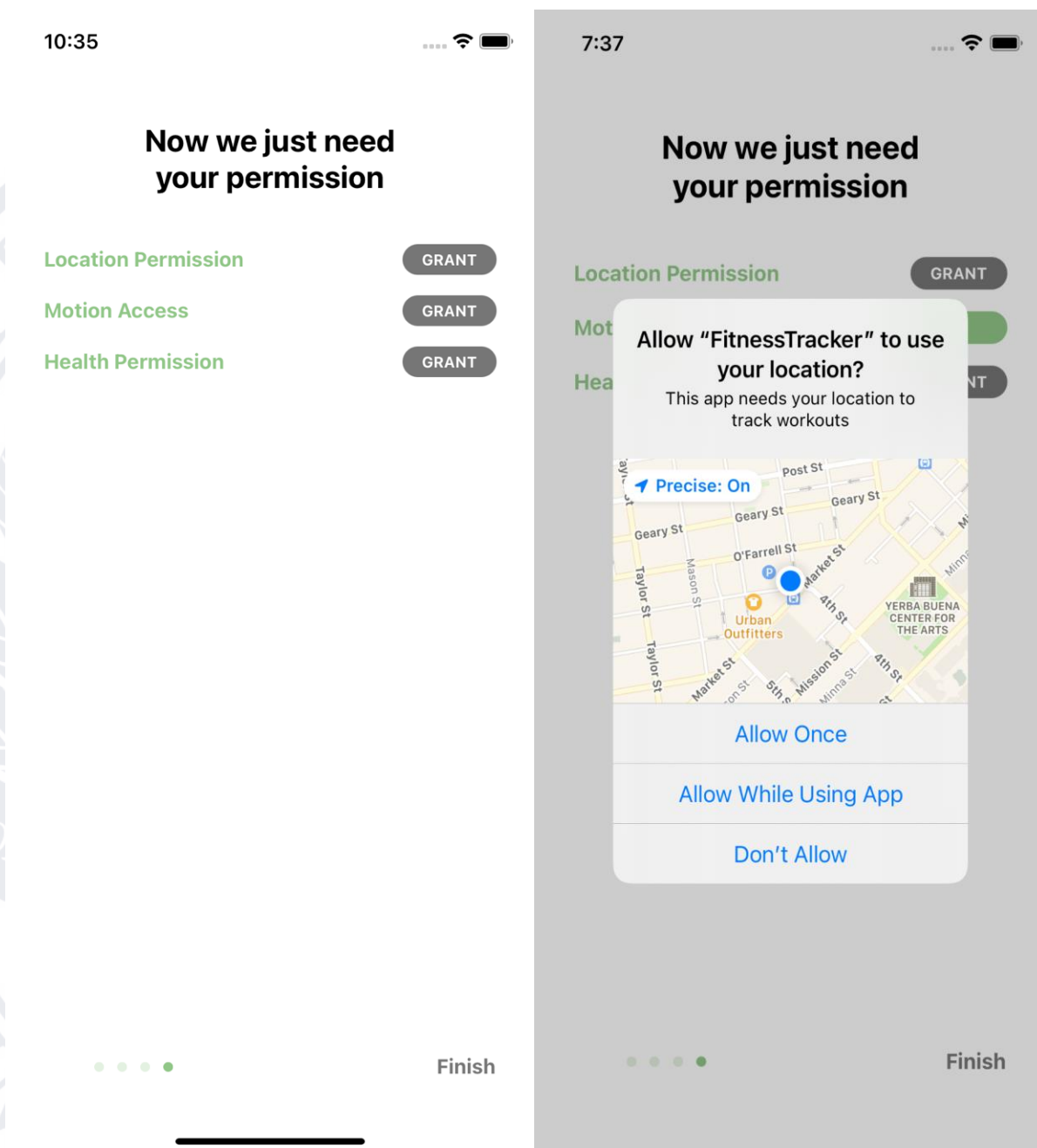


Рисунок 4.4 – Екран надання дозволів на використання системних функцій

Всі ці дії користувач здійснює лише один раз, безпосередньо після встановлення додатку.

Далі користувач потрапляє до основної частини додатку – списку його тренувань (рис. 4.5). Кожна комірка списку містить інформацію про дату тренування, тип тренування, пройдену відстань та тривалість тренування. Також

користувачу доступна карта з пройденим маршрутом у тому разі, якщо він надав доступ на відслідковування геолокації.

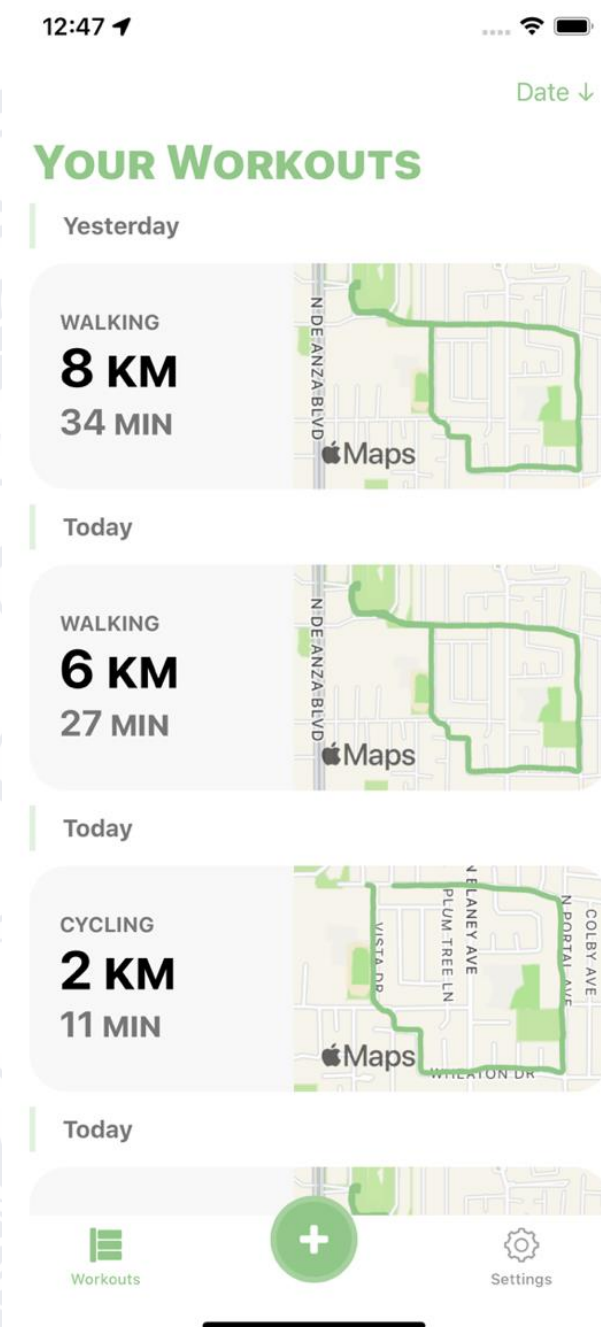


Рисунок 4.5 – Екран списку тренувань

За допомогою випадаючого меню (рис. 4.6) є можливість відсортувати список за датою створення, за пройденою дистанцією або ж за типом тренування (біг, ходьба, ходьба на довгі дистанції, їзда на велосипеді, катання на ковзанах). Окрім того тренування можна відмітити як заїзд для того, щоб можна було порівняти результати за декількома тренуваннями, тобто віднести його до групи.

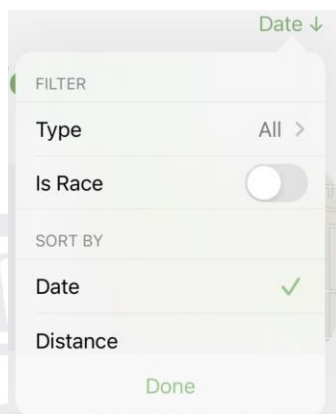


Рисунок 4.6 – Випадаюче меню налаштувань списку

Користувачу доступне загальне налаштування додатку, котре розділене на кілька секцій (рис. 4.7).

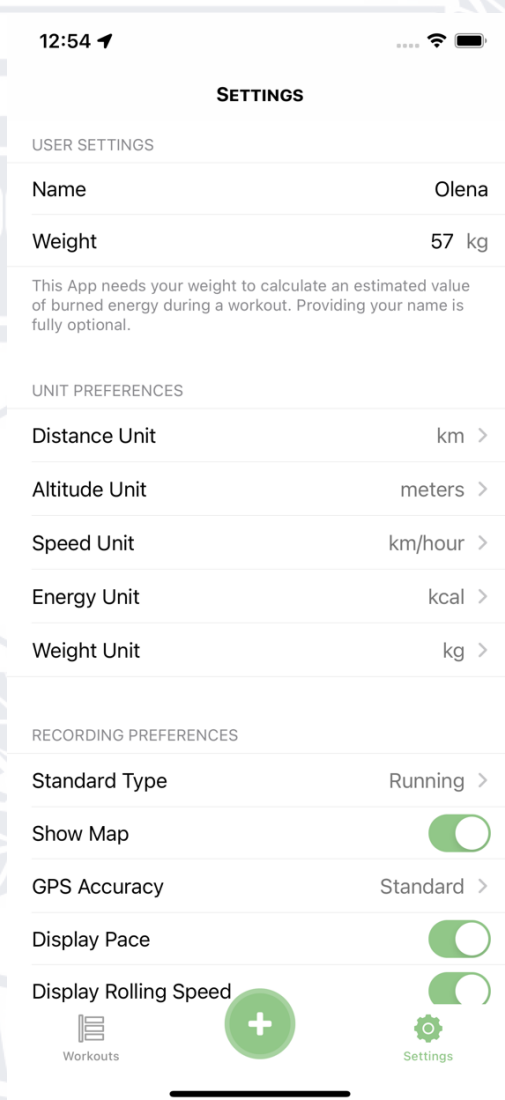


Рисунок 4.7 – Екран загального налаштування додатку

Загалом можна відредагувати форму звертання до користувача у додатку, вагу, одиниці виміру пройденої дистанції, висоти, швидкості, кількості

витрачених калорій та одиниці виміру ваги. Окрім того, можна вказати, який тип тренування використовується за замовченням, бажану точність геолокації (рис. 4.7), варіанти показу певних показників. Точність геолокації варіюється у межах від 1 до 50 метрів або є опція взагалі відключити геолокацію. Цей показник впливає на аналітичні показники тренувань.

GPS ACCURACY

DESIRED ACCURACY

Standard	✓
High	20.0 m
Acceptable	30.0 m
Last Resort	50.0 m
Off (not recommended)	

Рисунок 4.8 – Екран налаштування точності геолокації

Окрім того, користувач може відредагувати дозволи на використання тих чи інших системних функцій або інформації, що були встановлені під час першого налаштування додатку.

Окрему увагу заслуговує секція управління даними (рис. 4.9). Тут користувач має змогу зробити резервне копіювання, завантажити раніше збережені дані або видалити всю інформацію з пристрою.

DATA PREFERENCES

Create Backup	>
Import Backup Data	>
Delete all data	

Рисунок 4.9 – Секція управління даними

Щоб розпочати нове тренування користувачу необхідно натиснути на кнопку нового тренування, що розташоване на панелі екрані внизу посередині. Екран нового тренування представлений на рисунку 4.10. Користувачу надана можливість обрати тип тренування та переглянути готовність додатку розпочати

відслідковування. Вбудована карта дає змогу орієнтуватись на місцевості та планувати свій маршрут. Користувач може побачити об'єкти, що знаходяться неподалік від нього. Для початку тренування необхідно натиснути кнопку «Start».

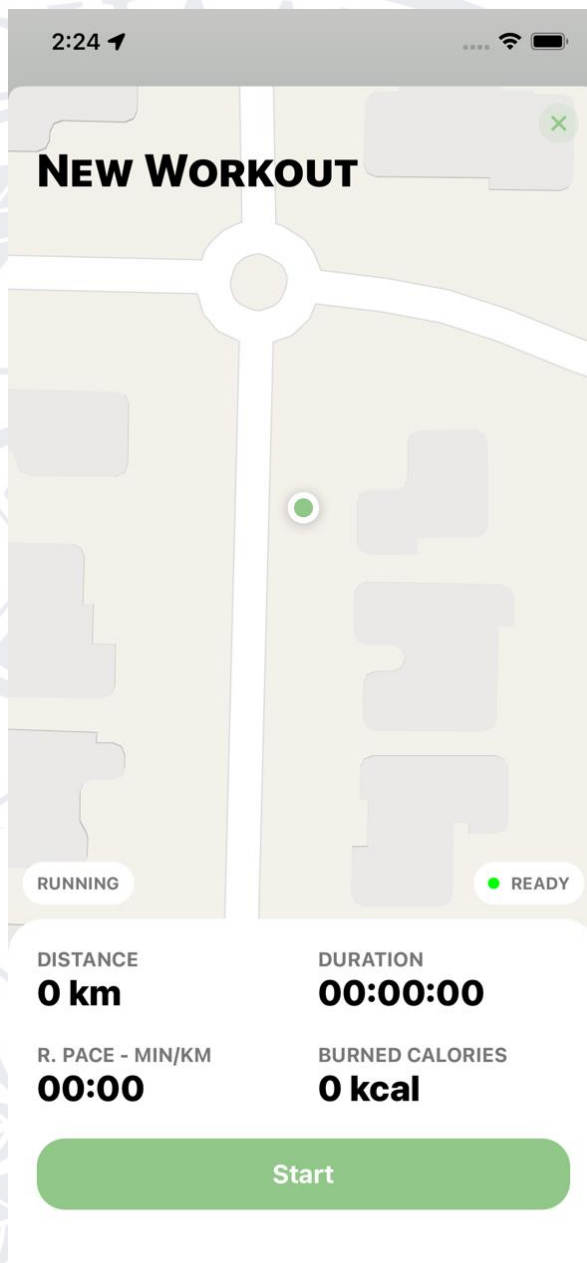


Рисунок 4.10 – Екран нового тренування

Після старту тренування, користувач може переглянути такі показники, як пройдена дистанція, тривалість тренування, швидкість(розраховується як дистанція поділена на витрачений час), а також кількість витрачених калорій, що розраховується як дистанція помножена на вагу користувача та помножена на спеціальну константу, що характеризує тип тренування.

DISTANCE

0.09 km

DURATION

00:00:22

R. PACE - MIN/KM

04:11

BURNED CALORIES

5.17 kcal

Рисунок 4.11 – показники, зібрані під час тренування

Також користувачу доступна опція призупинити тренування, без необхідності переривати відслідковування у додатку. Це може стати в нагоді, наприклад, для перепочинку.

Після закінчення тренування та його збереження до загального списку тренувань користувач має змогу відредагувати його та додати додаткову інформацію. Це можна зробити з екрану інформації про тренування натиснувши кнопку «Edit». Користувач може змінити тип тренування, дату та додати коментарі.

5:02

Cancel Save

EDIT WORKOUT

INFO

Type	Running
Distance	0.01 mi
Steps	Not Set
Start Date	4/16/22, 5:01 PM
Duration	00:00:05
Race	<input checked="" type="checkbox"/>

COMMENT

Comment

Рисунок 4.12 – Екран редагування тренування

4.2 Аналітика в розробленому додатку

Розширені статистичні дані тренування можна переглянути після його закінчення та збереження. Для цього необхідно на головному екрані натиснути на відповідну комірку. Інтерфейс даного екрану зображений на рисунку 4.13.

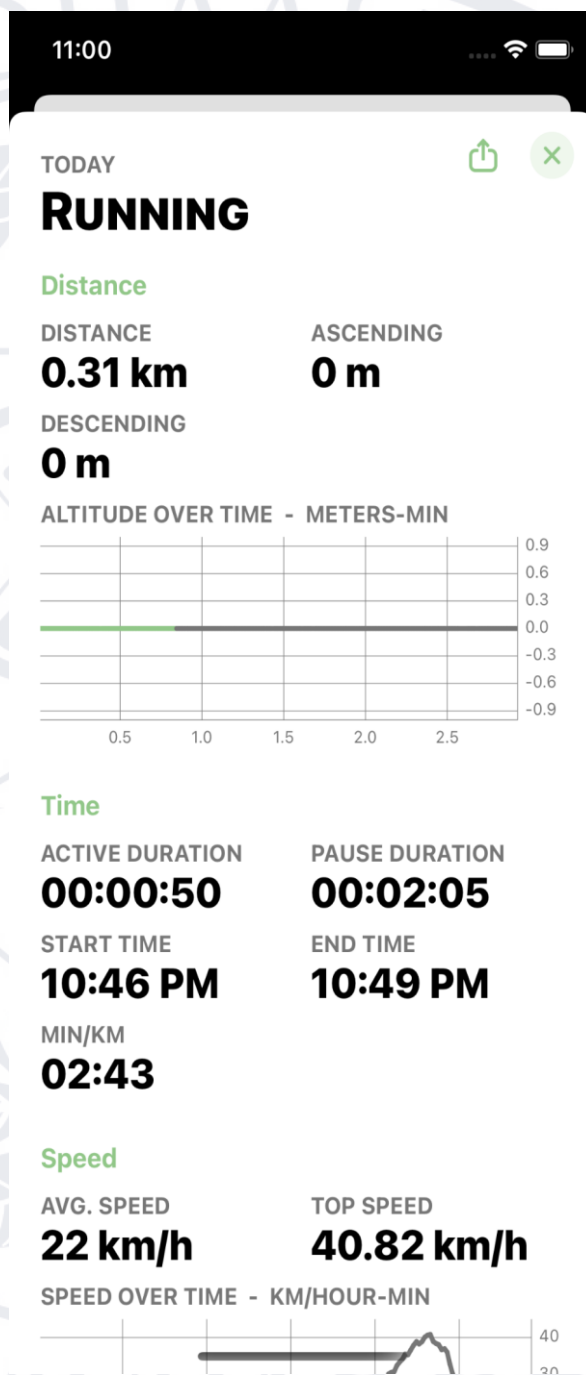


Рисунок 4.13 - Розширені статистичні дані

Даний екран містить кілька секцій. Першою є секція дистанція, що містить дані про пройдену відстань та подолані висоти і низини у метрах. Додатково

можна переглянути графік значень пройдених висот та низин протягом тренування.

Наступною секцією є секція час. Вона містить інформацію про тривалість тренування, тривалість паузи, якщо така була, а також час початку та час кінця тренування.

Далі користувачу пропонується статистика швидкостей за тренування. Надаються значення середньої та найбільшої швидкості. Також додатково можна переглянути, як змінювалась швидкість протягом тренування, на графіку(4.14).

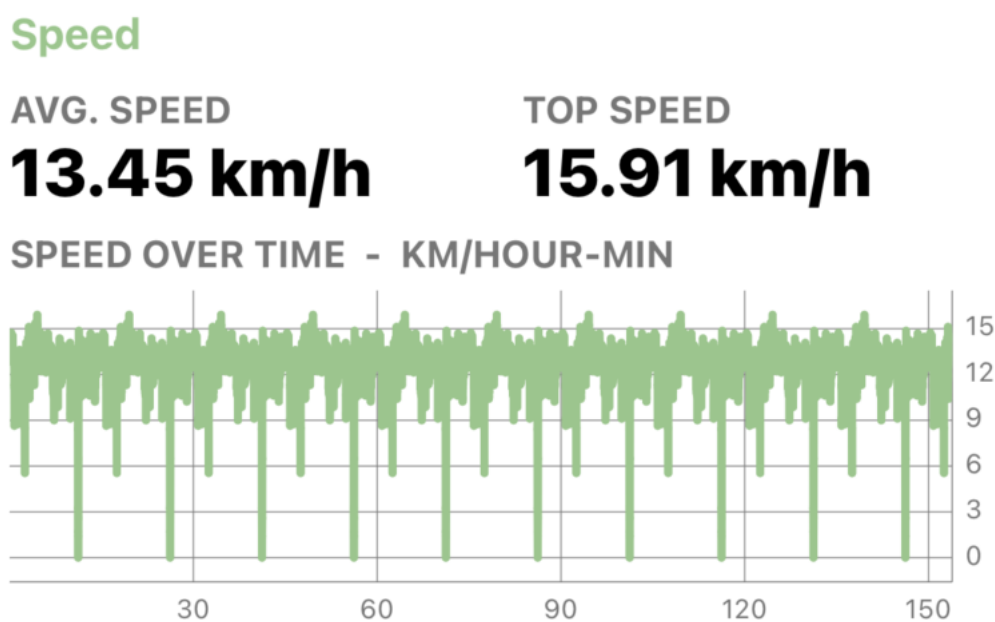


Рисунок 4.14 – Графік швидкостей протягом тренування

Секція «Спалені калорії»(рис. 4.15) надає інформацію про загальну кількість спалених калорій та середню кількість спалених калорій в хвилину.



Рисунок 4.15 – Секція «Спалені калорії»

Окрім того, у додатку можна переглянути деякі показники здоров'я –

кількість серцевих скорочень, мінливість частоти серцевих скорочень, середня частота серцевих скорочень під час ходьби(рис. 4.16). Ці дані збираються за допомогою бібліотеки HealthKit.

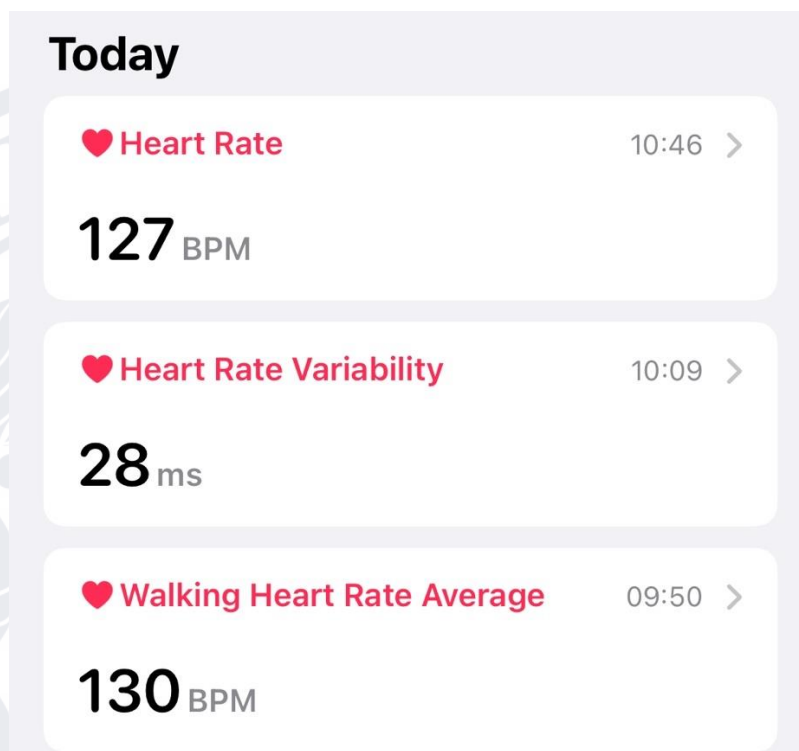


Рисунок 4.16 - Деякі показники здоров'я

Остання секція містить карту, на якій позначено пройдений маршрут. Карта може бути представлена у декількох варіантах: стандартна, гібридна та супутникова (рис. 4.17).

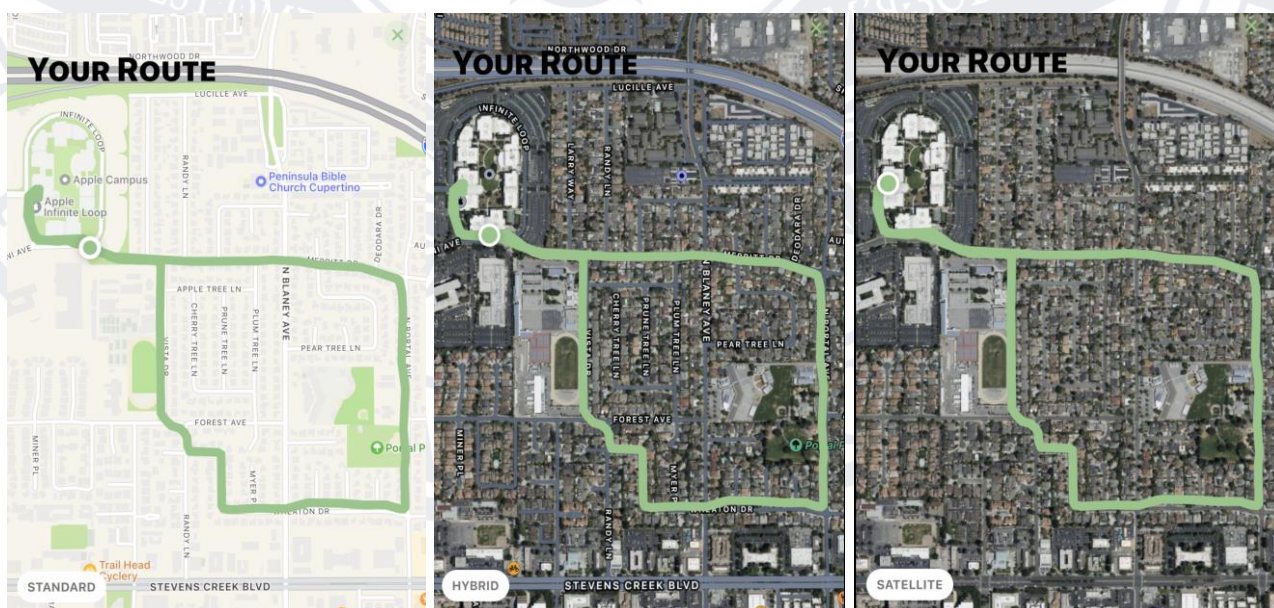


Рисунок 4.17 – Варіанти представлення карти

До обраного тренування можна застосувати ряд операцій, серед яких редагування, видалення та експорт. Експорт показників зібраних під час тренування може здійснюватися кількома способами:

- у форматі CSV [37] - текстовий файл, у якому міститься інформація. Кожен рядок – це окремий рядок таблиці, а стовпці відокремлені один від одного спеціальними символами – роздільниками (наприклад, комою);
- у форматі GPX [38] - загальний формат даних GPS для програмних додатків. Його можна використовувати для опису дорожніх точок, треків та маршрутів. Дані про розташування (висоту, час та іншу інформацію) зберігаються в тегах XML і можуть передаватися між пристроями GPS та програмним забезпеченням.

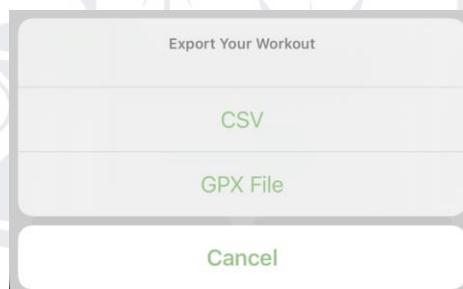


Рисунок 4.18 – Варіанти експорту даних

Приклад даних показників тренувань, що були зібрані за 2020-2022 роки та вивантажені у форматі CSV, зображено на рисунку 4.19.

Date	Activity Id	Type	Distance (km)	Duration	Average Pace	Average Speed	Calories Burned	Climb (m)	Average Heart Rate
2020-11-11 14:05:12	c9627fed-14ac-47a2-bed3-2a2630	Running	10.44	58:40	5:37	10.68	774.0	130	159.0
2020-11-12 14:05:12	be65818d-a801-4847-a43b-2acd4f	Running	12.84	1:14:12	5:47	10.39	954.0	168	159.0
2020-11-13 14:05:12	c09b2f92-f855-497c-b62a-c196b3	Running	13.01	1:15:16	5:47	10.37	967.0	171	155.0
2020-11-14 14:05:12	bc9b612d-3499-43ff-b82a-9b17b7	Running	12.98	1:14:25	5:44	10.47	960.0	169	158.0
2020-11-15 14:05:12	972567b2-1b0e-437c-9e82-fe807f	Running	13.02	1:12:50	5:36	10.73	967.0	170	154.0
2020-11-16 14:05:12	fe2cb3fc-6330-40fa-8a92-0f86f4e7	Running	10.29	59:18	5:46	10.41	764.0	133	155.0
2020-11-17 14:05:12	96acedc9-d3d5-4aac-8df4-f549a6	Running	12.93	1:10:16	5:26	11.04	953.0	159	158.0
2020-11-18 14:05:12	3c91092b-e6f3-4565-b540-6b6537	Running	12.31	1:09:26	5:38	10.64	903.0	134	157.0
2020-11-19 14:05:12	4c163abe-3a57-42fd-b50b-7f3659	Cycling	19.63	1:26:26	4:24	13.63	577.0	210	79.0
2020-11-20 14:05:12	70c7f2ba-f4cb-4403-a9e4-bf2c9b6	Running	12.97	1:13:56	5:42	10.52	964.0	171	156.0
2020-11-21 14:05:12	30aaa821-1d3a-4f2f-9688-8543ce	Cycling	32.61	1:55:15	3:32	16.98	830.0	462	118.0
2020-11-22 14:05:12	51781e97-560e-48d0-9522-44f303	Other	17.65	1:01:28	3:29	17.23	1164.0	219	77.0
2020-11-23 14:05:12	2bd1841f-b428-4683-a41b-2bfb4b	Cycling	36.89	1:58:39	3:13	18.65	937.0	491	122.0
2020-11-24 14:05:12	c9a8e088-441d-4b3f-bfbc-287e87	Cycling	28.17	1:27:07	3:06	19.4	685.0	400	111.0
2020-11-25 14:05:12	12723b6e-571b-4b68-be17-2c797	Cycling	19.41	1:11:33	3:41	16.28	536.0	199	124.0
2020-11-26 14:05:12	503a12b4-9a16-4c63-bbcd-be528	Running	4.38	33:55	7:45	7.75	334.0	60	138.0
2020-11-27 14:05:12	fa9f9d9-4995-4efe-9eb8-00983fd	Running	22.09	2:13:22	6:02	9.94	1652.0	342	148.0
2020-11-28 14:05:12	dc1669de-2d66-4e14-a24f-ba0edc	Running	15.27	1:19:37	5:13	11.51	1143.0	241	150.0
2020-11-29 14:05:12	d4a6b5b2-2dba-464a-a26e-ee0cb	Running	18.7	1:49:19	5:51	10.26	1397.0	280	150.0
2020-11-30 14:05:12	03006a09-870b-4f15-bb6c-9b9797	Running	18.2	1:43:05	5:40	10.59	1355.0	259	145.0
2020-12-1 14:05:12	a929c543-bc2e-446d-9431-40e60f	Other	16.8	1:07:21	4:01	14.97	1075.0	202	94.0
2020-12-2 14:05:12	6c6dd4aa-7567-40c7-8a8c-e6fc33	Running	17.87	1:44:00	5:49	10.31	1269.0	261	141.0
2020-12-3 14:05:12	eb6a80cb-bf87-4426-bc11-9610b7	Running	12.71	1:14:51	5:53	10.19	893.0	180	145.0

Рисунок 4.19 – Приклад представлення даних тренувань у форматі CSV

Однією з особливостей розроблюваного додатку є можливість формування аналітичних звітів за місяць. Користувач має змогу згенерувати звіт за весь період з початку користування додатком до часу безпосередньої генерації звіту. Інтерфейс генерації звіту можна знайти на екрані налаштувань (рис. 4.20).



Рисунок 4.20 – Інтерфейс генерації звіту

Звіт складається з декількох секцій. В першій секції знаходяться графіки розподілу даних за весь період за чотирма показниками.

Перший графік, який зустрічає користувач – це графік пройденої дистанції за період користування додатком (рис. 4.21).

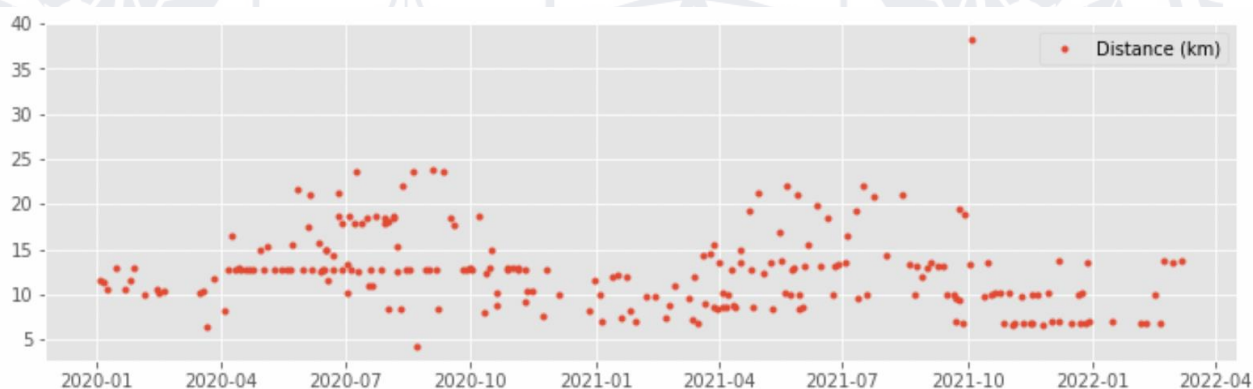


Рисунок 4.21 – Графік пройденої дистанції за період користування додатком

Далі розміщений графік середньої швидкості за період. Даний графік проілюстрований на рисунку 4.22.

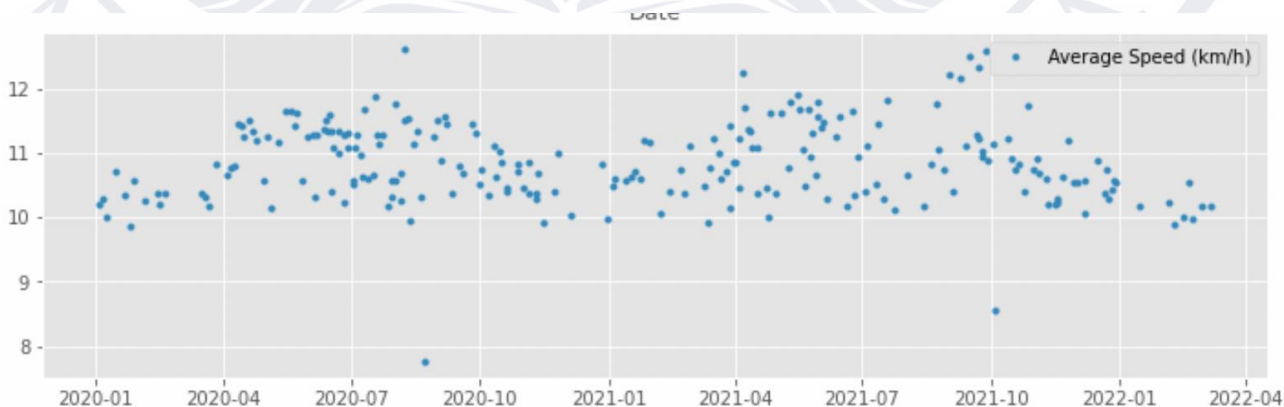


Рисунок 4.22 – Графік середньої швидкості за період

Наступним можемо побачити графік подоланих висот за період (рис. 4.23).

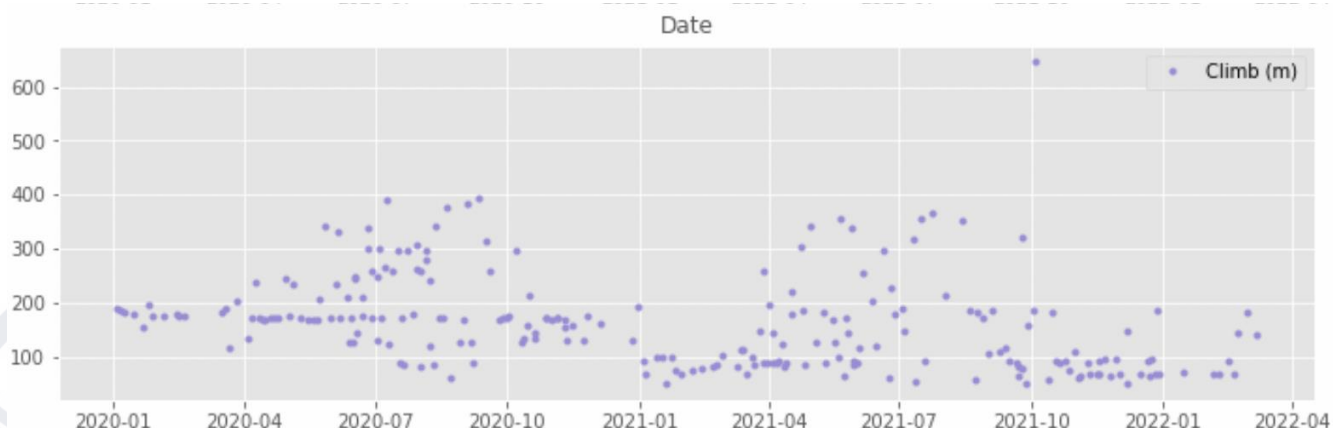


Рисунок 4.23 – Графік подоланих висот за період

Для відслідковування стану здоров'я, зокрема серцево-судинної системи, у звіті присутній графік зміни середньої частоти серцевих скорочень за тренування (рис. 4.24).

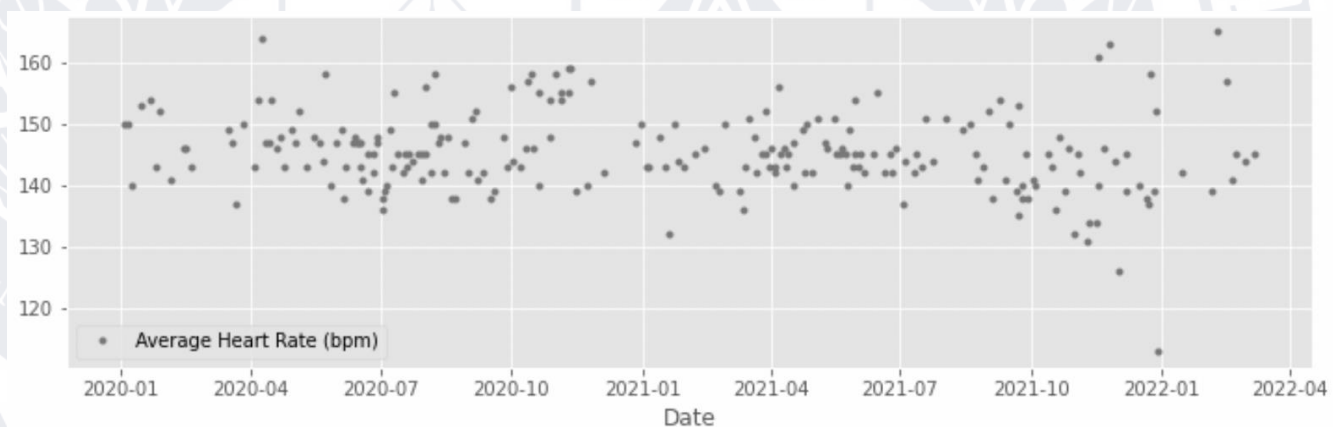


Рисунок 4.24 – Графіки розподілу даних за весь період за чотирма показниками

Далі у звіті можемо побачити як змінювались значення певних характеристик протягом років. На рисунку 4.25 проілюстровано приклад за три роки: 2020, 2021 та 2022.

Distance (km)	Average Speed (km/h)	Climb (m)	Average Heart Rate (bpm)
13.649730	10.840991	196.342342	146.846847
11.781947	10.915929	135.610619	143.920354
9.806250	10.147500	104.000000	147.250000

Рисунок 4.25 – Таблиця значень основних показників за три роки

Окрім того, у звіті присутня гістограма, що ілюструє розподіл кількості серцевих скорочень за категоріями-рівнями: мінімальний, легкий, помірний, важкий, дуже важкий, непомірний. Приклад зображений на рисунку 4.26.

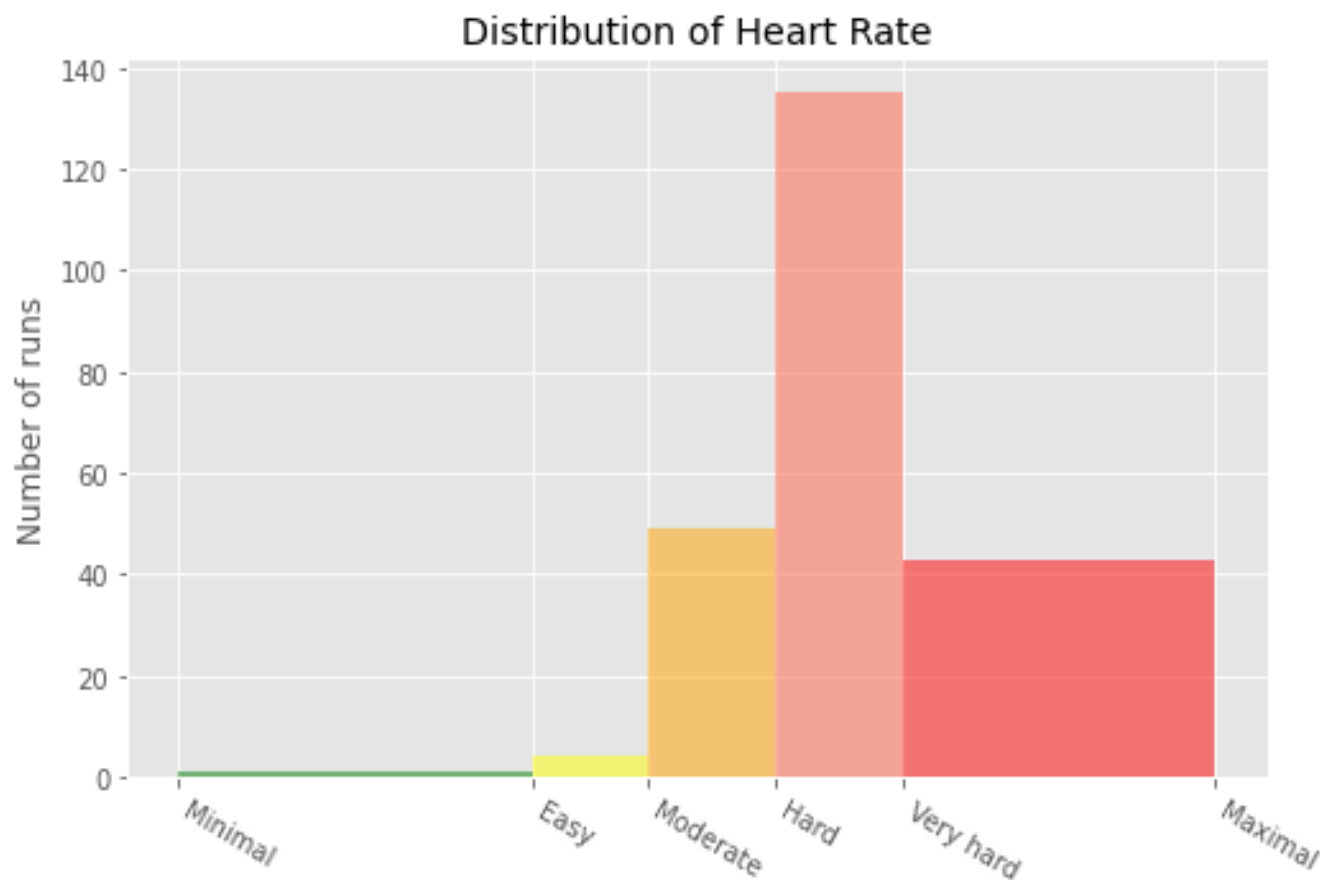


Рисунок 4.26 – Таблиця значень основних показників за три роки

Далі ми можемо побачити таблицю загальної подоланої дистанції та висот за весь період. Дані розподілені за типом тренування, у даному випадку – бігу та ходьби. Приклад зображений на рисунку 4.27.

Type	Distance (km)	Climb (m)
Cycling	406.24	4806
Running	2924.93	37950

Рисунок 4.27 - Загальна подолана дистанція та висота

Для того, щоб побачити коливання наданих показників, у звіт буд доданий графік усереднених даних розподілених за місяцями (рис. 4.28). На ньому можемо побачити інтенсивність тренування у певний період, що характеризується пройденою дистанцією або те, як змінювався показник частоти серцевих скорочень. На рисунку ми бачимо прогрес протягом часу користування додатком у кількості пройденої відстані в порівнянні з історичним середнім значенням. Також бачимо як змінювалась середня частота серцевих скорочень. Загалом було виявлено, що значення були близько середнього значення, і тенденції до збільшення не було виявлено (відстань та/або частота серцевих скорочень) з часом.

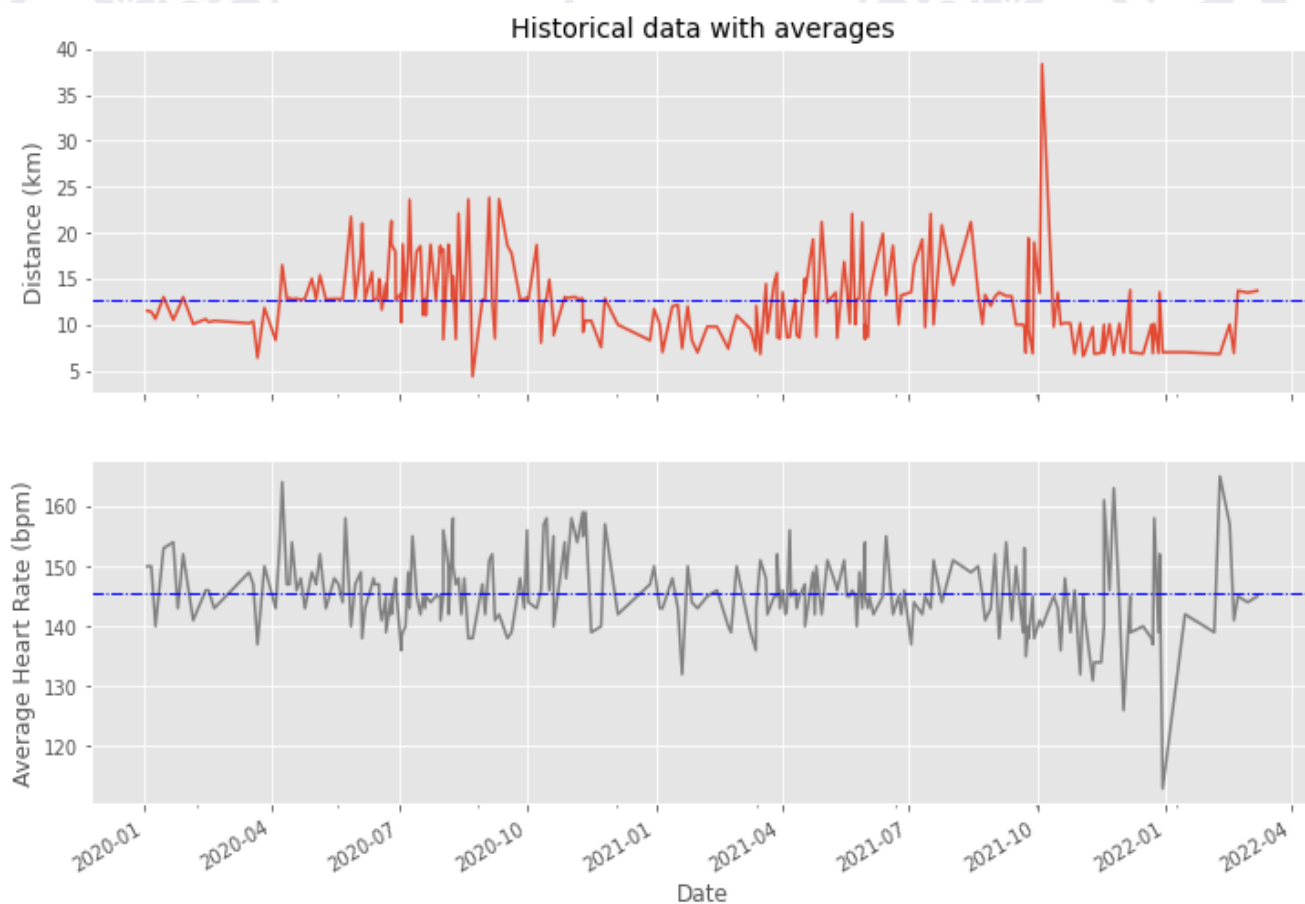


Рисунок 4.28 – Графік усереднених даних розподілених за місяцями

На останок, можемо побачити у звіті таблицю, де обраховані максимальні, мінімальні, середні значення, медіану, стандартне відхилення та інші. Всі показники розподілені за типом тренування. Приклад даної таблиці наведений на рисунку 4.29.

Type		Average Speed (km/h)	Climb (m)	Distance (km)
Cycling	25%	16.805000	195.250000	18.725000
	50%	19.450000	304.000000	22.720000
	75%	21.570000	375.250000	31.762500
	count	16.000000	16.000000	16.000000
	max	23.990000	553.000000	49.180000
	mean	19.180625	300.375000	25.390000
	min	13.630000	163.000000	13.110000
	std	3.094784	125.984589	10.237659
	total	NaN	4806.000000	406.240000
Running	25%	10.410000	91.000000	10.000000
	50%	10.780000	163.500000	12.730000
	75%	11.282500	188.250000	13.690000
	count	232.000000	232.000000	232.000000
	max	12.640000	646.000000	38.320000
	mean	10.853578	163.577586	12.607457
	min	7.750000	49.000000	4.380000
	std	0.620607	87.574597	4.385593
	total	NaN	37950.000000	2924.930000

Рисунок 4.29 – Детальна інформація про тренування за весь період

У додатку реалізована можливість визначення рівня фізичної активності користувача (PAL) за допомогою спеціальних аналітичних функцій. PAL розраховується за наведеною нижче формулою [39]:

$$PAL = \frac{TEE}{BMR},$$

де TEE - загальна витрата енергії протягом 24 годин в кілокалоріях, BMR - загальна швидкість обміну речовин.

BMR в свою чергу розраховується за наступними формулами:

- для чоловіків:

$$10 \times W + 6.25 \times H - 5 \times A + 5$$

- для жінок:

$$10 \times W + 6.25 \times H - 5 \times A - 161$$

В наведених вище формулах W - вага користувача, H - зріст, A - вік.

Далі отримане значення зіставляється із значеннями в таблиці 4.1.

Таблиця 4.1 – Визначення рівня фізичної активності за значенням PAL

Спосіб життя	PAL
Неактивний	< 1,40
Малорухливий	1,40-1,69
Помірно активний	1,70-1,99
Енергійний	2.00-2.40
Надзвичайно активний	> 2,40

Розраховані значення можна переглянути в налаштуваннях додатку в секції, що містить інформацію про користувача.

4.3 Технологія Push Notifications у додатку

Пуш-повідомлення(англ. push notifications) – це спливаючі повідомлення на мобільному пристрої [40]. Видавці програм можуть надсилати їх у будь-який час - користувачам не потрібно перебувати в програмі або використовувати свої пристрої для їх отримання. Вони можуть показувати останні спортивні результати, спонукати користувача до дії, наприклад завантаження купона, або повідомляти користувача про подію, наприклад, про швидкий розпродаж. Пуш-повідомлення виглядають як текстові SMS-повідомлення або мобільні оповіщення, але є доступними лише для тих користувачів, які встановили додаток.

Пуш-повідомлення можуть використовуватись для [41]:

- привернення уваги користувачів до додатку;
- надання своєчасних оновлень або термінових новин;
- демонстрація оновлень.

Пуш-повідомлення можуть супроводжуватись звуком або вібрацією та можуть відображатися в декількох місцях на мобільному пристрої – на екрані блокування, в центрі сповіщень або як спливаючий банер (рис. 4.30). Обрати бажаний варіант можна в налаштуваннях пристрою.

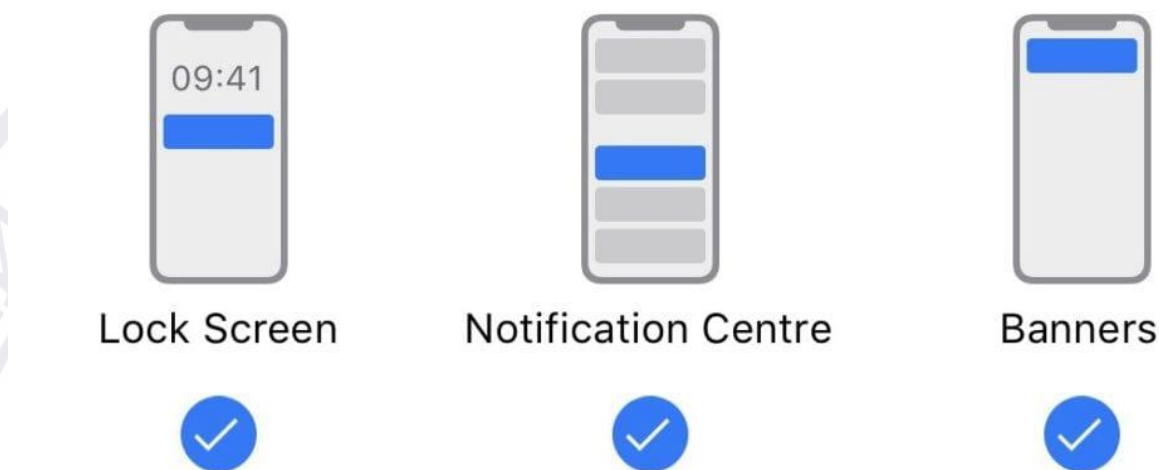


Рисунок 4.30 – Варіанти розміщення пуш-повідомлень

В основному, користувачі бачать сповіщення як банер або спливаюче сповіщення, коли вони використовують свій телефон. Це сповіщення відображається незалежно від того, що робить користувач.

Деякі з учасників надсилання push-повідомлень включають:

- службу push-повідомлень операційної системи. Кожна мобільна операційна система (ОС), включаючи iOS, Android, Fire OS, Windows і BlackBerry, має власну службу;
- розробника програми - видавець програми включає в свою програму OSPNS. Потім видавець завантажує програму в магазин додатків.
- клієнтський додаток – програма для ОС, встановлена на пристрої користувача. Вона отримує вхідні сповіщення.

Розроблюваний додаток може надсилати пуш-повідомлення про необхідність тренування. Наприклад, якщо користувач не використовував додаток принаймні сім днів, тобто не розпочинав тренування, йому надійде відповідне нагадування. Даний функціонал проілюстровано на рисунку 4.31.

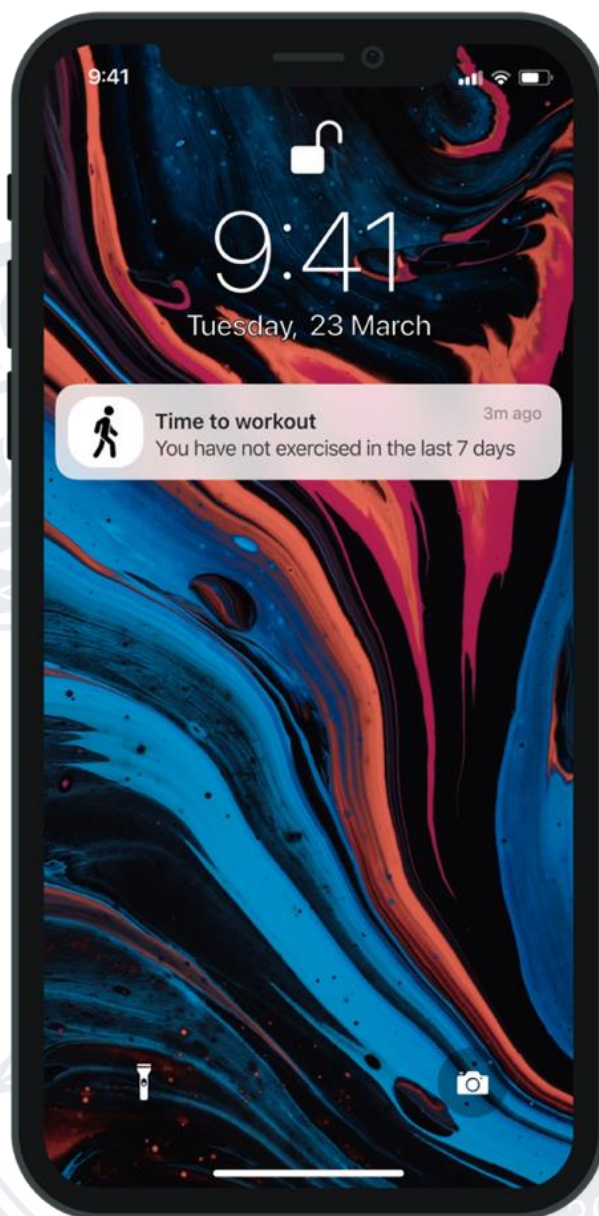


Рисунок 4.31 – Приклад роботи пуш-повідомлень додатку

Висновок до розділу 4

У даному розділі було розглянуто інтерфейс створеного додатку, зазначені його основні характеристики та детально описано функціональність. Також вся зазначена функціональність проілюстрована відповідними рисунками.

ВИСНОВКИ

В ході виконання роботи було розглянуто актуальність створення додатку для відслідковування фізичної активності користувача, створено порівняльну таблицю аналогів, а також проаналізований ринок мобільних додатків для платформи iOS.

Наведено детальний опис використовуваної мови програмування, її основні переваги над альтернативами. Також детально розглянуті основні інструменти та середовище розробки для мобільної платформи iOS.

Під час виконання практичної частини роботи були використані рекомендовані компанією Apple архітектурні підходи та підходи об'єктно-орієнтованого програмування. Також під час розробки були використані функції для визначення фізичної активності користувача на основі показників фізичного стану людини(вага, ріст, стать, інше). Окремо слід відзначити наявність аналітики тренування у додатку. В результаті, було створено фітнес-додаток, що містить широкий функціонал, необхідний для відслідковування фізичної активності користувача та її детального аналізу.

СПИСОК ЛІТЕРАТУРИ

1. European downloads of Business apps surged 132% to 702 Million in 2020. URL: <https://sensortower.com/blog/europe-app-category-growth-2020> (дата звернення: 15.03.2022)
2. Quarterly growth of available apps in the Apple App Store as of 1st quarter 2021, by category. URL: <https://www.statista.com/statistics/271105/growth-of-the-top-apple-app-store-categories/> (дата звернення: 15.03.2022)
3. Health app. URL: [https://en.wikipedia.org/wiki/Health_\(Apple\)](https://en.wikipedia.org/wiki/Health_(Apple))
4. Strava. URL: <https://apps.apple.com/us/app/strava-run-ride-swim/id426826309> (дата звернення: 15.03.2022)
5. Relieve: Run, Ride, Hike & more. URL: <https://apps.apple.com/us/app/relive-run-ride-hike-more/id1201703657> (дата звернення: 15.03.2022)
6. Run Tracker. URL: <https://apps.apple.com/us/app/run-tracker-gps-fitness-tracking-for-runners/id453027672> (дата звернення: 15.03.2022)
7. About Objective-C. URL: <https://developer.apple.com/ObjectiveC/> (дата звернення: 27.03.2022)
8. Swift. URL: <https://developer.apple.com/swift/> (дата звернення: 27.03.2022)
9. TIOBE Index for March 2022. URL: <https://www.tiobe.com/tiobe-index/> (дата звернення: 27.03.2022)
10. Горошко, Юрій Васильович, and Ганна Юхимівна Цибко. "Про перспективи використання мови Swift у навчанні програмування." Науковий часопис НПУ імені МП Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання 19 (2017): 59-63.
11. Swift. A powerful open language that lets everyone build amazing apps. URL: <https://www.apple.com/swift/> (дата звернення: 27.03.2022)
12. Native vs cross-platform development: pros & cons revealed. URL: <https://www.uptech.team/blog/native-vs-cross-platform-app-development> (дата звернення: 27.03.2022)
13. Bin Uzayr, Sufyan (ed.). Mastering UI Mockups and Frameworks: A Beginner's Guide. CRC Press, 2022.

14. Figma. URL: <https://www.figma.com/>. (дата звернення: 27.03.2022)
15. Xcode 13. URL: <https://developer.apple.com/xcode/> (дата звернення: 27.03.2022)
16. AppCode. URL: <https://www.jetbrains.com/objc/> (дата звернення: 27.03.2022)
17. Interface Builder Built-In. URL: <https://developer.apple.com/xcode/interface-builder/> (дата звернення: 27.03.2022)
18. Simulator User Guide. URL: <https://developer.apple.com/Simulator> (дата звернення: 27.03.2022)
19. CocoaPods. GitHub <https://cocoapods.org/> (дата звернення: 27.03.2022)
20. Downey, Eric. "Package Managers." Practical Swift. Apress, Berkeley, CA, 2016. 39-53.
21. GitHub. URL: <https://github.com/> (дата звернення: 27.03.2022)
22. UIKit. URL: <https://developer.apple.com/documentation/uikit> (дата звернення: 27.03.2022)
23. CoreLocation. URL: <https://developer.apple.com/documentation/corelocation/> (дата звернення: 27.03.2022)
24. CoreMotion. URL: <https://developer.apple.com/documentation/coremotion> (дата звернення: 27.03.2022)
25. MapKit. URL: <https://developer.apple.com/documentation/mapkit/> (дата звернення: 27.03.2022)
26. CoreData. URL: <https://developer.apple.com/documentation/coredata> (дата звернення: 27.03.2022)
27. HealthKit. URL: <https://developer.apple.com/documentation/healthkit> (дата звернення: 27.03.2022)
28. Charts. URL: <https://github.com/danielgindi/Charts> (дата звернення: 27.03.2022)
29. CoreGPX. URL: <https://github.com/vincentneo/CoreGPX> (дата звернення: 27.03.2022)

30. SnapKit. URL: <https://github.com/SnapKit/SnapKit#usage> (дата звернення: 27.03.2022)
31. Model-View-ViewModel. URL: <https://en.wikipedia.org/wiki/Model-View-ViewModel> (дата звернення: 30.03.2022)
32. Kouraklis, John. "MVVM as Design Pattern." MVVM in Delphi. Apress, Berkeley, CA, 2016. 1-12.
33. Blackheath, Stephen. Functional reactive programming. Simon and Schuster, 2016.
34. Ciesla, Robert. "UML Class Diagrams." Programming Basics. Apress, Berkeley, CA, 2021. 145-165.
35. Human Interface Guidelines. URL: <https://developer.apple.com/design/human-interface-guidelines>
36. Privat, Michael, and Robert Warner. "Talking to services: icloud and dropbox." Pro iOS Persistence. Apress, Berkeley, CA, 2014. 259-286.
37. Comma separated values. URL: https://en.wikipedia.org/wiki/Comma-separated_values (дата звернення: 10.04.2022)
38. GPS Exchange Format. URL: https://en.wikipedia.org/GPS_Exchange_Format
39. Assessment of physical activity and energy expenditure: an overview of objective measures. URL: <https://www.frontiersin.org/articles/> (дата звернення: 10.04.2022)
40. Mobile Push Notifications: Everything you need to know. URL: <https://www.getvero.com/resources/mobile-push-notifications/> (дата звернення: 10.04.2022)
41. iOS remote push notifications in a nutshell. URL: <https://medium.com/ios-remote-push-notifications-in-a-nutshell> (дата звернення: 10.04.2022)

**Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань**

Степанюк Олена Сергіївна

Прізвище, ім'я, по батькові

Факультет інформаційних і прикладних технологій

Факультет

122 Комп'ютерні науки

Шифр і назва спеціальності

Сучасні інформаційні технології та програмування

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «Розробка фітнес-додатку з аналітичними функціями на платформі iOS» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

дата

підпис