

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” 2021 р.

Дипломний проект

на здобуття ступеня бакалавра

**за освітньо-професійною програмою “Інженерія програмного забезпечення
комп’ютерних систем”**

спеціальності 121 “Інженерія програмного забезпечення”

на тему: Система трекінгу активностей

Виконав (-ла): студент (-ка) 4 курсу, групи ІП-74
(шифр групи)

Тіхонов Станіслав Олександрович

(прізвище, ім’я, по батькові)

(підпис)

Керівник Волокита І.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль)

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті немає
запозичень з праць інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2021 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної
техніки Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)
Освітньо-професійна програма “Інженерія програмного забезпечення комп’ютерних
систем”
спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ **Сергій СТИРЕНКО**
(підпис)
“_____” _____ 2021 р.

ЗАВДАННЯ
на бакалаврський дипломний проєкт студента

Тіхонова Станіслава Олександрович

1. Тема проєкту Система трекингу активностей
керівник проєкту _____ ст. викладач Волокита Іван Миколайович _____,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від ____ 2021 року № ____

2. Термін здачі студентом закінченого проєкту 31 травня 2021 р.

3. Вихідні дані до проєкту технічна документація, теоретичні дані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)
Опис та аналіз предметної області, аналіз аналогів у даній області, виокремлення основних вимог до ПЗ, проектування архітектури системи трекингу активностей.

1. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень)
структурна схема системи, функціональна схема, принципова схема.

2. Консультанта проекту, з вказівкою розділів проекту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В.П.		

3. Дата видачі завдання 30 вересня 2020р.

Календарний план

№ п/п	Найменування етапів дипломного проекту	Терміни виконання етапів проекту	Примітки
1.	<i>Затвердження теми проекту</i>	<i>10.12.2020-15.12.2020</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12.2020-15.03.2021</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2021-25.03.2021</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>25.03.2021-5.04.2021</i>	
5.	<i>Програмна реалізація системи</i>	<i>5.04.2021-15.04.2021</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2021-20.05.2021</i>	
7.	<i>Захист програмного продукту</i>	<i>25.05.2021</i>	
8.	<i>Передзахист</i>	<i>28.05.2021</i>	
9.	<i>Захист</i>	<i>19.06.2021</i>	

Студент-дипломник _____
 (підпис)

Керівник проекту _____
 (підпис)

Анотація

В бакалаврському дипломному проєкті реалізовано систему трекингу активностей, що призначена для моніторингу спортивних тренувань користувача.

Програма дозволяє будувати маршрут для бігу або вело заїзду, а також слідкувати за пересуванням користувача в режимі реального часу. Програмний продукт був створений на мові JavaScript з використанням технології React Native.

Для візуалізації та отримання картографічної інформації використовується інтегрований сервіс з геоінформаційної системи MapBox.

Annotation

In this project for a Bachelor's Degree, system for activity monitoring is realized.

The software product makes it possible to build running or cycling routes and monitor user's moves in real time system. This program was realized with JavaScript language and React Native technology.

The integrated mapping from MapBox geoinformation system is used for the visualization and output of mapped information.

Опис альбому
до дипломного проєкту
на тему: «Система трекингу активностей»

Київ – 2021 року

Технічне завдання
до дипломного проекту
на тему: «Система трекингу активностей»

Київ – 2021 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	2
2. ПРИЧИНИ ДЛЯ РОЗРОБКИ.....	2
3. ЦІЛЬ ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1 ВИМОГИ ДО ПРОДУКТУ, ЩО РОЗРОБЛЯЄТЬСЯ	3
5.2 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	3
5.3 ВИМОГИ ДО АПАРАТНОЇ ЧАСТИНИ	3
6. ЕТАПИ РОЗРОБКИ	3

					<i>ІАЛЦ.467100.002 ТЗ</i>		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Система трекингу активностей</i> <i>Технічне завдання</i>		
<i>Розробив</i>		<i>Тіхонов С.О..</i>					
<i>Перевірів</i>		<i>Волокита І.М.</i>					
<i>Реценз.</i>							
<i>Н. Контр.</i>		<i>Сімоненко В.П.</i>					
<i>Затв.</i>		<i>Стіренко С.Г.</i>			<i>НТУУ «КПІ ім. І.Сікорського»</i> <i>каф. ОТ, гр. ІП-74</i>		
					<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
						1	3

1.НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування: «Система трекингу активностей». Область застосування: система допомагає будувати маршрути і моніторити спортивні показники під час тренування.

2.ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського дипломного проекту, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

3.МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки є створення системи трекингу активностей.

4.ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література, публікації в спеціалізованих періодичних виданнях, довідники по платформах дистанційного навчання, публікації в мережі Інтернет по даній темі.

5.ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

- Здатність коректно обробляти дані.
- Висока швидкість роботи, в тому числі, обробки запитів.
- Простий та зрозумілий для використання інтерфейс.

5.2. Вимоги до програмного забезпечення

- NodeJS version 12+;
- React Native version 63+;

5.3. Вимоги до апаратної частини

- Підключення до Internet;
- Оперативна пам'ять не менше 2 ГБ;
- Сервер із підтримкою розгортання додатків на Node.js;

					<i>ІАЛЦ.467100.002 ТЗ</i>	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

	Дата
6.1. Вивчення необхідної літератури	24.02.2021
6.2. Складання та узгодження технічного завдання	08.03.2021
6.3. Написання вступу та огляду готових рішень	22.03.2021
6.4. Проектування архітектури додатку	04.04.2021
6.5. Написання програмної частини	10.04.2021
6.6. Тестування і виправлення помилок	07.05.2021
6.7. Підготовка документації дипломного проекту	19.05.2021

					<i>ІАЛЦ.467100.002 ТЗ</i>	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Пояснювальна записка
до дипломного проєкту
на тему: «Система трекингу активностей»

Київ – 2021 року

ЗМІСТ

ВСТУП	15
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ТРЕКІНГУ ТРЕНУВАНЬ	17
1.1 ОПИС ЗАВДАННЯ.....	17
1.2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	18
1.2.1 Nike Run Club	18
1.2.2 Strava	22
1.2.3 Saynax Sports Tracker.....	26
1.2.4 Runmaster.....	30
1.3 Порівняння оглянутих рішень	34
ВИСНОВКИ ДО РОЗДІЛУ 1	37
РОЗДІЛ 2 ВИБІР ТЕХНОЛОГІЙ І ПЛАТФОРМ	38
2.1 REACT NATIVE.....	38
2.1.1 Порівняння React Native з іншими рішеннями	40
2.1.2 Порівняння стилізації React Native і Android додатків	42
2.2 МЕНЕДЖЕР СТАНУ. МОВХ	44
2.2.1 Порівняння Mobx з Redux.....	48
2.3. МАРВОХ	49
2.3.1 Mapbox документація	49
2.3.2 Mapbox Studio	51
2.3.3 Mapbox SDK.....	53
2.3.4 Mapbox бібліотека під React Native.....	53
2.4 СХОВИЩЕ ДАНИХ FIREBASE	53
ВИСНОВКИ ДО РОЗДІЛУ 2	55
РОЗДІЛ 3 ПРОЕКТУВАННЯ І РОЗРОБКА ДОДАТКА.....	56
3.1 Вимоги до системи	56
3.1.1 Функціональні вимоги	56
3.1.2 Вимоги до інтерфейсу.....	57
3.1.3 Нефункціональні вимоги	58
3.2 ПРОЕКТУВАННЯ СИСТЕМИ.....	58
3.2.1 Авторизація	58
3.2.2 Firestore.....	61
3.3 ДЕТАЛІ ПРОЕКТУ.....	67
ВИСНОВКИ ДО РОЗДІЛУ 3	73
РОЗДІЛ 4 ІНСТРУКЦІЯ КОРИСТУВАЧА	74
ВИСНОВКИ ДО РОЗДІЛУ 4	92
ВИСНОВКИ	93

					ІАЛЦ.467100.003 ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Тіхоненв С.О..			Система трекингу активностей Пояснювальна записка	Літ.	Арк.
Перевірів		Волокита І.М.					Аркушів
Реценз.							
Н. Контр.		Сімоненко В.П.				1	81
Затв.		Стіренко С.Г.				НТУУ «КПІ ім. І.Сікорського» каф. ОТ, гр. ІП-74	

СПИСОК СКОРОЧЕНЬ

JSON (JavaScript Object Notation) – Об’єкт позначення Джаваскрипт.

UI (User Interface) – Користувацький інтерфейс.

JS (JavaScript) – мова Джаваскрипт.

API (Application Programing Interface) – Програмний інтерфейс додатку.

NoSql (No Structured Query Language) – Формат нереляційних баз даних.

DI (Dependency Injection) – Інтегрування залежностей.

GPS (Global Position System) – Супутникова система навігації.

ОС (Операційна система).

HTTP (HyperText Transfer Protocol) – Протокол передачі даних.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

В наш час спорт є невід'ємною частиною життя у будь-якій сфері, оскільки саме спорт прямо впливає на здоров'я і характер людини. Стоїть пригадати народну приказку «У здоровому тілі – здоровий дух!». Регулярні заняття безпосередньо впливають на кісті, зв'язки людини, зміцнюють їх, що зменшує ризики отримання травм. Фізичні вправи поліпшують кровоток людини, що приводить до покращеної роботи серця і зменшення ризиків появи інсультів та інфарктів у літньому віці.

Досить багато сервісів для моніторингу тренувань здобули популярності у біговій та вело індустріях. Проблемою являється саме те – що на кожную обрану спортивну сферу приходить окремий сервіс, що призводить до безлічі однотипних додатків у користувачів. Ця робота не буде першою спробою у вирішенні поставленої проблеми, існують ідеологічно схожі рішення і все таки метою даної роботи є поєднання декілька спортивних індустрій в єдину систему, що також вносить розроблену функціональність, що бракує іншим розробкам.

Розроблений застосунок отримав унікальний та інтуїтивний інтерфейс, щоб задовільнити потреби користувачів при занятті спортом: при використанні системи, користувач може використовувати додаток для заміру і моніторингу стану тренування, як при занятті бігом, так і при їзді на велосипеді, адже важливо під час свого тренування підтримувати єдиний темп бігу, знати яку дистанцію вже подолав. Для велосипедиста або бігуна досить важливим може бути прокласти майбутній маршрут, згідно якому той і проведе своє тренування. Також тривіальною задачею для спортсмена є коригування вже налаштованого маршруту, що є великою проблемою в подібних додатках.

Також стоїть зазначити, що багатьом існуючим рішеннях притаманна проблема аварійного закриття застосунку у використанні. Що призводить до втрати прогресу тренування. Причинами можуть бути: низький заряд акумулятору й подальше вимикання смартфона, мала кількість оперативної пам'яті, що призводить до звільнення додатку операційною системою з переліку активних програм тощо. В даній роботі система дозволяє користувачу самому

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

обрати, що він бажає зробити при наступному запуску додатку після аварійного завершення: зберегти, видалити або одразу продовжити тренування.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ТРЕКИНГУ ТРЕНУВАНЬ

1.1 Опис завдання

Завдання даної роботи полягає в розробці системи трекингу користувацьких активностей і тренувань у вигляді зручного додатку для смартфона. Додаток повинен мати зрозумілий і інтуїтивний інтерфейс для користувача. Система має покривати наступну функціональність:

- поєднання декількох режимів трекингу(велосипед, біг, автозаїзд);
- моніторинг тренування – заміри таких показників як «дистанція», «середня швидкість\темп», «поточна швидкість\темп», «загальний час», «час в русі»;
- прокладення маршрутів;
- редагування маршрутів;
- перегляд збережених маршрутів;
- перегляд збережених тренувань;
- одиничне або множинне видалення обраних маршрутів;
- одиничне або множинне видалення обраних тренувань;
- початок тренування з відображенням обраного збереженого маршруту на мапі;
- авторизація користувача у системі;
- вибір теми стилізації додатку;
- вибір мови додатку;
- видалення, збереження або продовження тренування після аварійного завершення додатку;
- видалення, збереження або продовження побудови маршруту після аварійного завершення додатку;
- пошук місця по назві міста для швидкого переміщення мапи.

Щодо прокладення маршрутів, має бути декілька режимів побудови:

- 1) прямий режим побудови «крапками» – користувач вказує кожен наступний координату на мапі, що з'єднується з попередньою – прямою перетином мапи;

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

2) режим напрямків «лініями» - користувач вказую кожну наступну координату на мапі, що з'єднується з попередньою лініями напрямків відповідно автомобільному, велосипедному або пішому рухам.

Редагування має здійснюється перетаскуванням координат(крапок) маршруту на мапі з наступним перерахунком дистанції маршруту.

У системі повинна бути розроблена авторизація користувача як власна, з використанням пошти з паролем, так й вхід за допомогою Google акаунтів. Авторизація дозволяє використовувати додаток на декількох смартфонах з єдиною історією тренувань і маршрутів.

При виборі мови додатку мають бути запропоновані українська(державна) і англійська мови.

Майбутнім завданням розробки має бути:

- пошук місць не лише по назві міста, а також за назвою закладів, вулиць, споруд для покращеної взаємодії користувача з додатком;
- інтеграція сервісу з соціальною мережею;
- об'єднання декількох виділених маршрутів в один.

1.2 Аналіз існуючих рішень

Існує безліч застосунків в обраній сфері, кожен з яких відповідає своїм нормам і реалізує певний функціонал. Розглянемо 4 найбільш популярні застосунки, кожен з яких має свої певні недоліки і переваги.

1.2.1 Nike Run Club

Nike Run Club є одним з найвдалиших проєктів у біговій індустрії. Як пише Nike про свій сервіс: «Nike Run Club - Your Perfect Running Partner»[1].

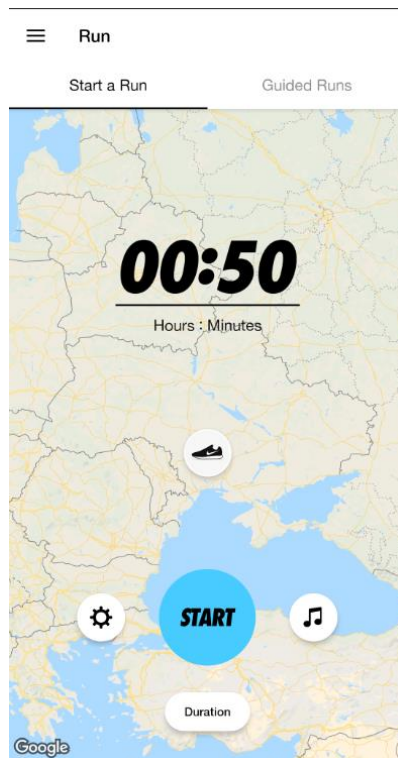


Рисунок 1.1 Головний екран застосунку «Nike Run Club»

Головним завданням даного сервісу є супроводження користувача на пробіжці, записуючи такі показники як: «подолана дистанція», «середній темп», «поточний темп», «час в русі», «кількість потрачених калорій», «серцебиття на 1 хв», «кількість кроків на 1 хв» й зчитування стану GPS модуля смартфона для візуалізації пройденого шляху.

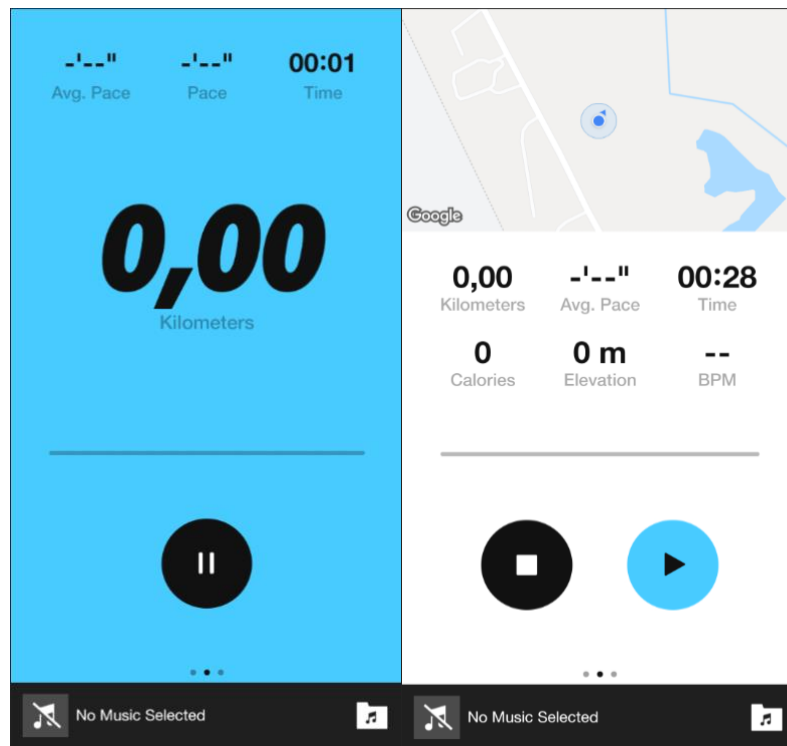


Рисунок 1.2 Екран активного тренування застосунку «Nike Run Club»

Сервіс здобув популярності завдяки:

- 1) Іменованій компанії Nike, що славиться своїми одними з найкращих модельних рядів бігових кросівок;
- 2) Компанія має досить велику аудиторію як серед покупців, так і серед спортсменів-фанатів;
- 3) Простий і інтуїтивний інтерфейс застосунку;
- 4) Відсутність рекламних вставок у додатку.

Серед функціоналу слід виділити: аудіо супроводження відомими спортсменами, що являється корисним для новачків, що тільки починають бігати. Адже супроводження допоможе ознайомитися з правилами дихання, крокування і є важливою підтримкою, мотиватором під час бігу, оскільки перша пробіжка може бути досить стресовою, бо організм не підготовлений до подібних навантажень.

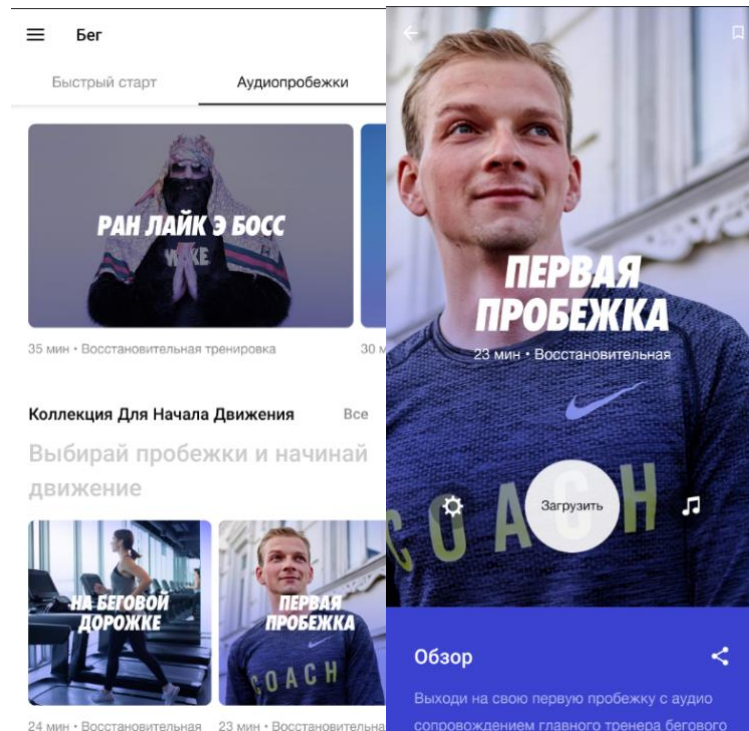


Рисунок 1.3 Экран выбору аудио супроводження пробіжки застосунку «Nike Run Club»

Тривіальною задачею є слідкування за своїм прогресом тренувань, аналізу минулих пробіжок, що буде корисним для самовдосконалення і покращення своїх майбутніх показників.

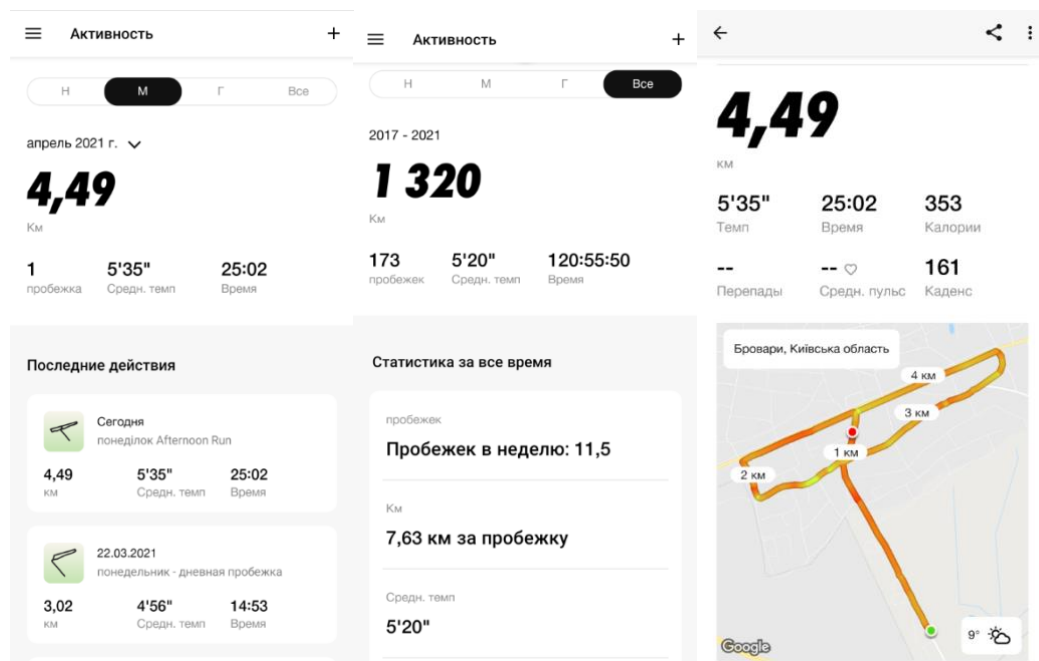


Рисунок 1.4 Экран перегляду історії тренувань застосунку «Nike Run Club»

Для більш досвідчених спортсменів сервіс пропонує брати участь у викликах, котрі публікують як представники самого сервісу, так і сторонні організації. Подібні виклики спонукають ставити спортивні цілі, долати їх і залишатися мотивованим, виховуючи в собі твердість характеру.

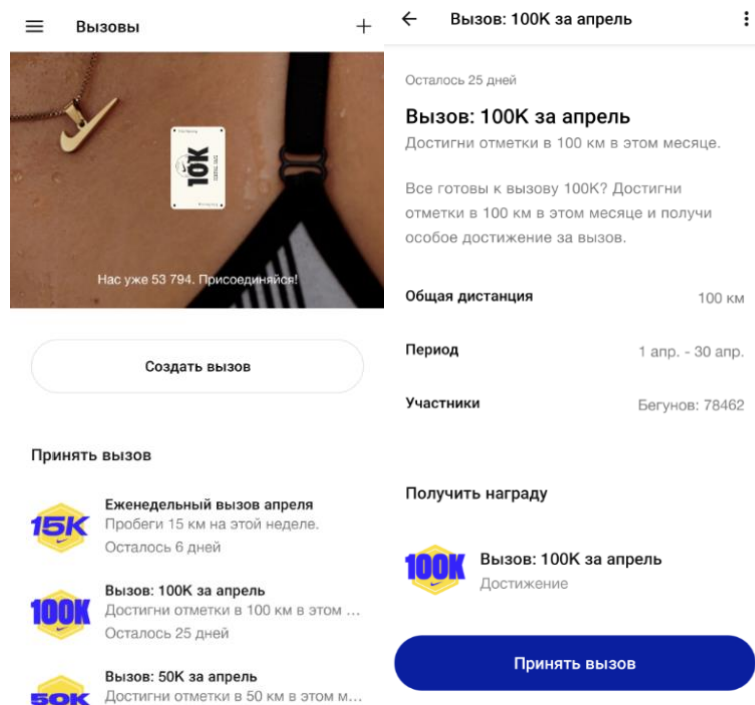


Рисунок 1.5 Экран wyboru виклика застосунку «Nike Run Club»

Функціонал побудови маршрутів у «Nike Run Club» відсутній.

1.2.2 Strava

Strava є одним з найпопулярніших сервісів, що поєднує декілька спортивних індустрій в єдину систему.

Сервіс презентований у 2009 році організацією Strava Inc одразу здобув попит, оскільки розробники прислуховувались до своєї аудиторії, реалізуюючи як необхідний функціонал, що безпосередньо націлений на моніторинг стану тренувань, так і додатковий, що став характерною рисою додатку в той час і досі таким являється, а саме – інтеграція соціальної мережі до застосунку.

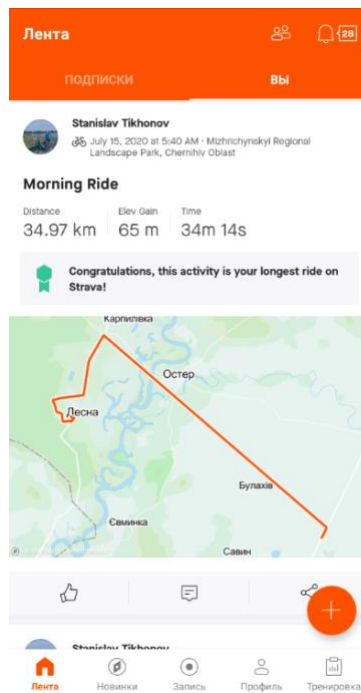
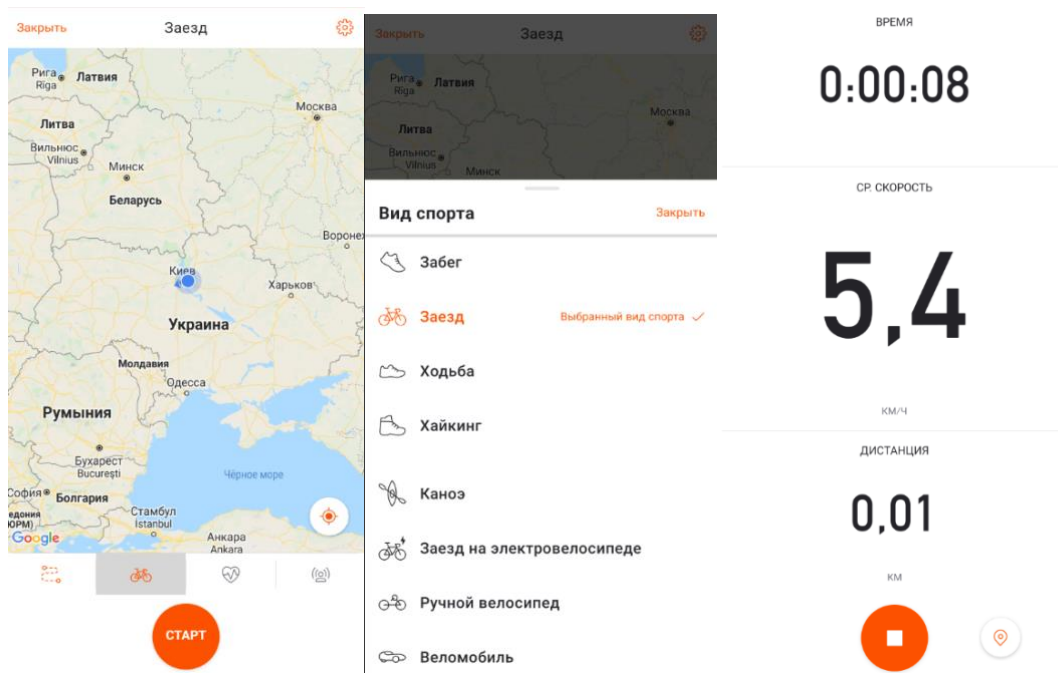


Рисунок 1.6 Головний екран застосунку «Strava»

Станом початку 2021 року, система підтримує більше 30 видів тренувань, найпоширеніші з них: «забіг», «заїзд», «ходіння», «хайкінг» тощо. Головним завданням сервісу як і вищезгаданого «Nike Run Club» [1] являється моніторинг тренувань спортсменів, слідкування за основними показниками як «дистанція», «середня швидкість», «максимальна швидкість», «час в русі», «набір висоти» й зчитування стану GPS модуля смартфона для візуалізації пройденого шляху.



Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

14

Рисунок 1.7 Екран тренування застосунку «Strava»

Як було згадано вище, дана система інтегрована з соціальною мережею, що дозволяє розширювати знайомства і слідкувати за прогресом кумирів, друзів. Можна налаштовувати повідомлення появи тренувань, що одразу потрапляють до користувача.

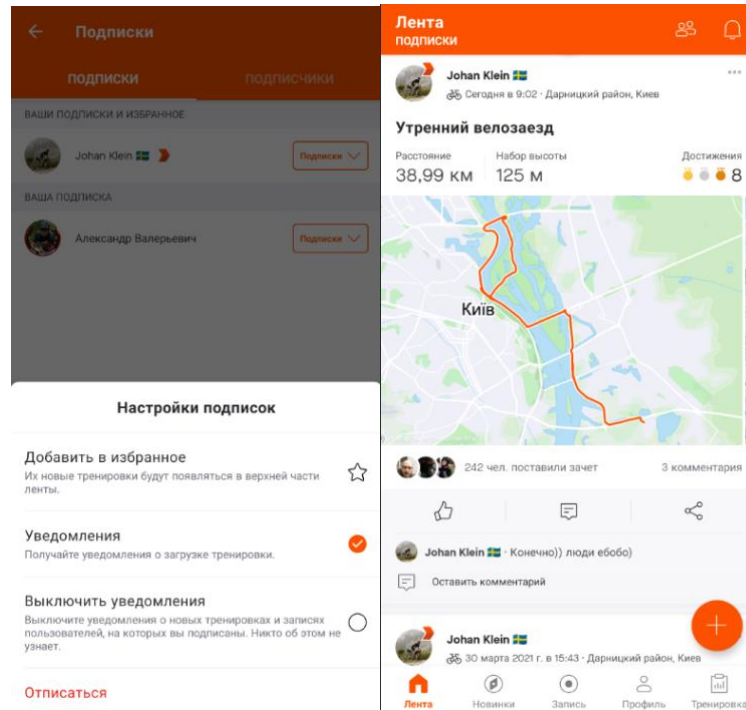


Рисунок 1.8. Екрани налаштування підписок і історією тренувань друзів застосунку «Strava»

У «Strava» виділяється наступний функціонал: в системі є підтримка групових тренувань, достатньо одного смартфона на компанію і мати спільні підписки між собою. В результаті, можна відправити запрошення на вступ до обраного групового тренування будь-кому зі списку підписників. Після підтвердження запрошень – обране тренування відобразиться в історії всіх спортсменів вищезгаданої групи.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

15

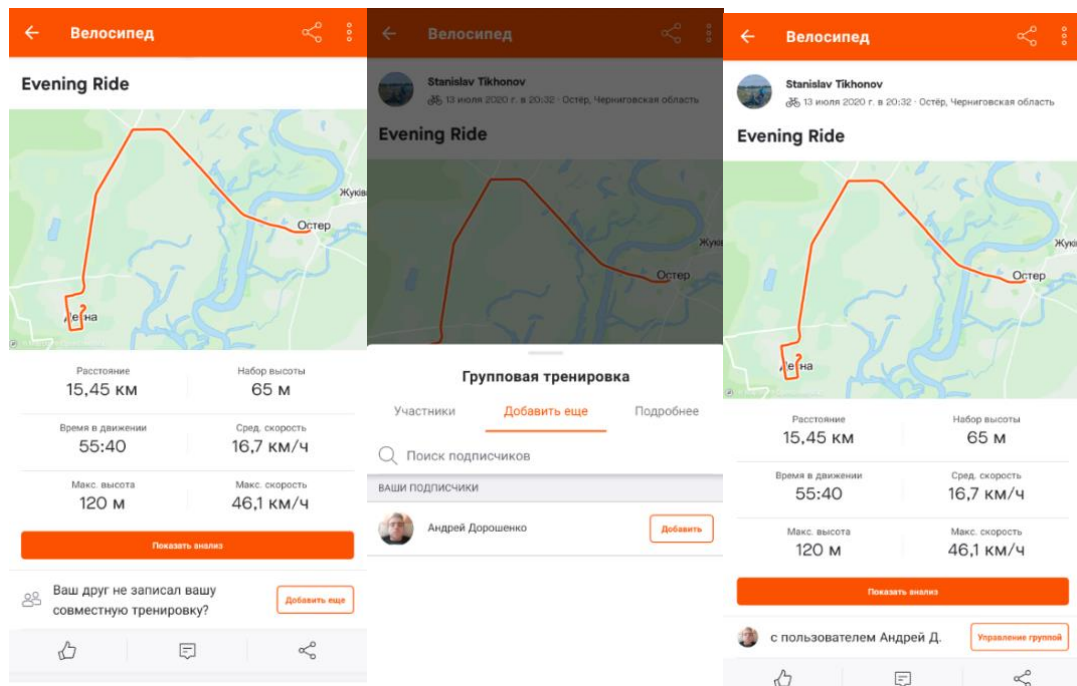


Рисунок 1.9 Экран налаштування групового тренування застосунку «Strava»

В даному застосунку притаманна схожа функціональність «викликів» «Nike Run Club» – задачі. Доступний великий набір задач під різні види спорту (біг, велосипед, плавання тощо), що періодично оновлюється. Такі задачі можуть бути як ставленням цілей, так і онлайн, офлайн (на період пандемії COVID-19 лише онлайн) змаганнями.

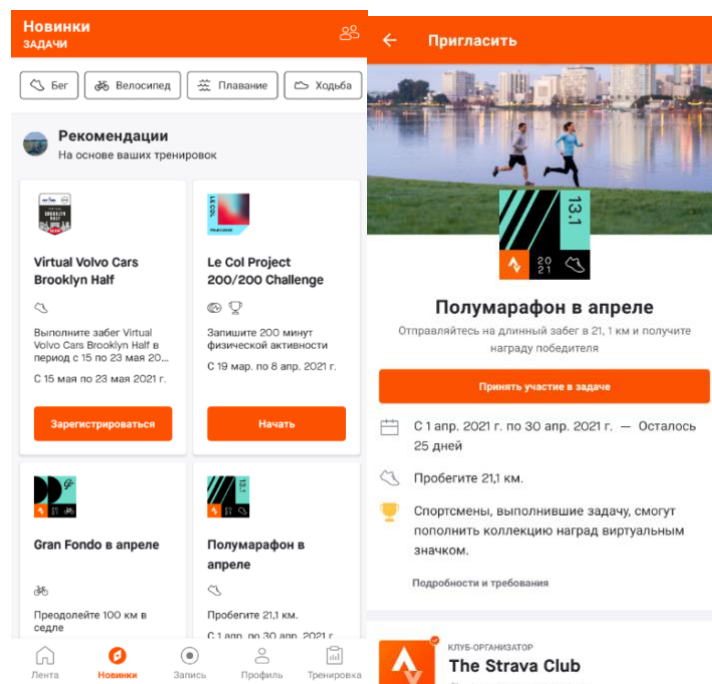


Рисунок 1.10 Экран wyboru задачі застосунку «Strava»

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

16

Щодо побудови маршрутів, у «Strava» присутній такий функціонал, але лише в платній підписці сервісу. Навіть з підпискою побудова маршрутів не є вдалою і має вагомий недолік - для прокладення маршрутів майбутніх тренувань необхідно переходити на окремий сайт організації Strava Inc, авторизуватися до власного акаунту і прокласти маршрут, що є достатньо не зручним при використанні мобільного смартфона.

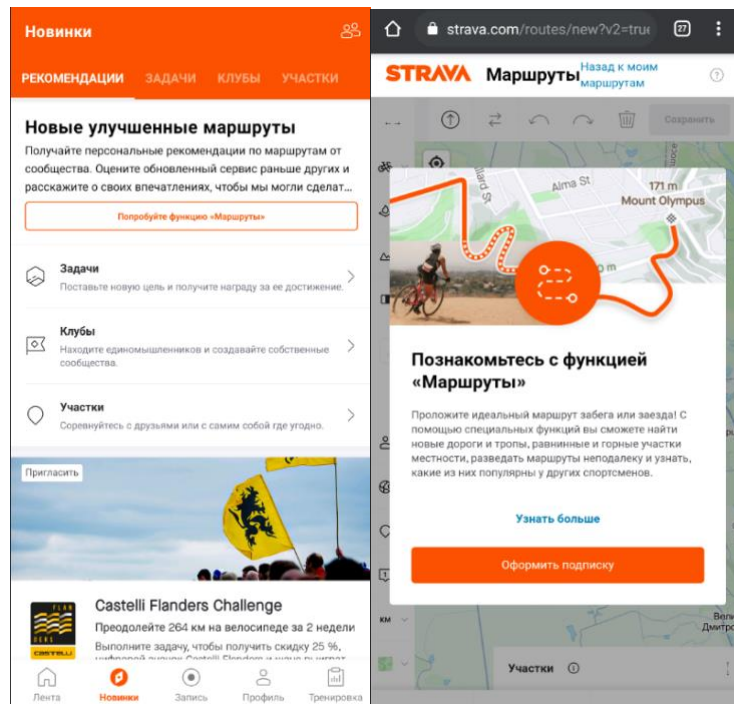


Рисунок 1.11 Екран функції побудови маршрутів застосунку «Strava»

1.2.3 Caynax Sports Tracker

Caynax Sports Tracker – ще один відомий сервіс-трекінг тренувань, що поєднує в собі більше 20 видів спорту. Найпоширеніші з яких: «біг», «стаціонарний велосипед», «велосипед», «бігові лижі», «гребля».

Декілька можливостей завдяки яким даний трекер здобув популярності:

- Швидкий і зручний додаток;
- Відсутність реєстрації і авторизації;
- Невеликий розмір додатку;
- Весь функціонал присутній у безкоштовній версії(стоїть зазначити, присутня реклама, котра вимикається єдиною оплатою в середині додатку).

З запуском додатку першим екраном є мапа з відображенням місце розташування, обиранням виду тренування, клавішею для початку тренування і клавішею зміни теми додатка, що є важливим для тривалих, виснажених тренувань задля збереження зору.

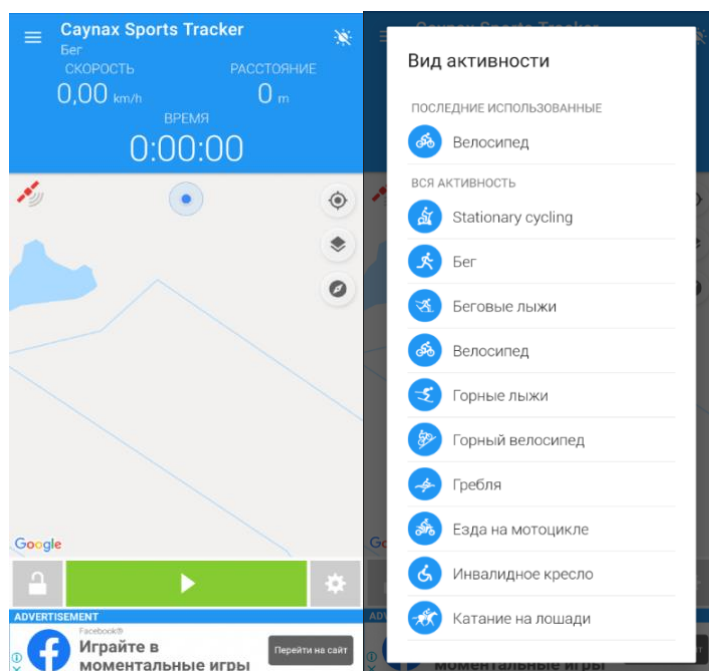
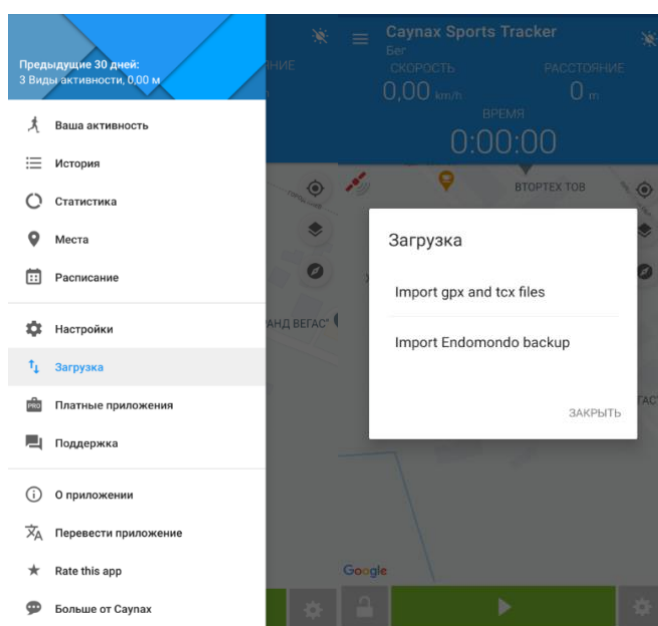


Рисунок 1.12 Головний екран застосунку «Саунax Sports Tracker»

У застосунку відсутній функціонал побудови маршрутів, однак присутня можливість: підгрузити існуючий маршрут формату gpx, tcx або резервну копію сервісу Endomondo і вже за ним провести тренування.



Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

18

Рисунок 1.13 Екран підгрузки існуючого маршруту застосунку «Саунах Sports Tracker»

Даний сервіс має зручний фільтр історії тренувань. Користувач може легко знайти свої минулі тренування за наступними конфігураціями фільтру: вид тренування(наприклад біг), діапазон дат(наприклад поточна неділя)

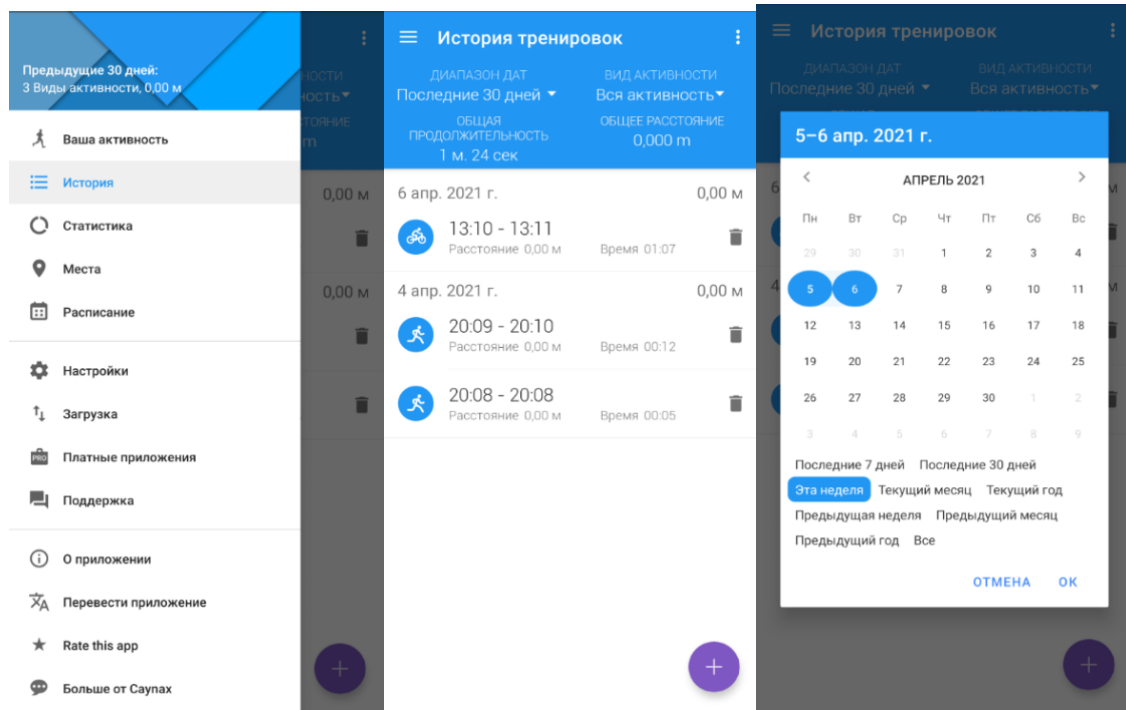


Рисунок 1.14 Екран історії збережених тренувань застосунку «Саунах Sports Tracker»

Стоїть відокремити корисний функціонал додатку: розклад і встановлення тренувань про заплановані тренування. Розклад нагадує календар Google Calendar, що являється зручним і інтуїтивним у використанні. При додаванні нового запису можна вказати:

- діапазон дат;
- час початку тренування(час нагадування);
- вид тренування(біг, велосипед тощо);
- тип тренування(основні тренування – загальний режим слідування GPS активності, цільове тренування – тренування з заданням певної цілі: подолання дистанції, часу або страти певної кількості калорій);

- тренувальні дні(можна налаштувати як одиночне нагадування тренування, так і періодично за певними днями тижня або на кожен день).

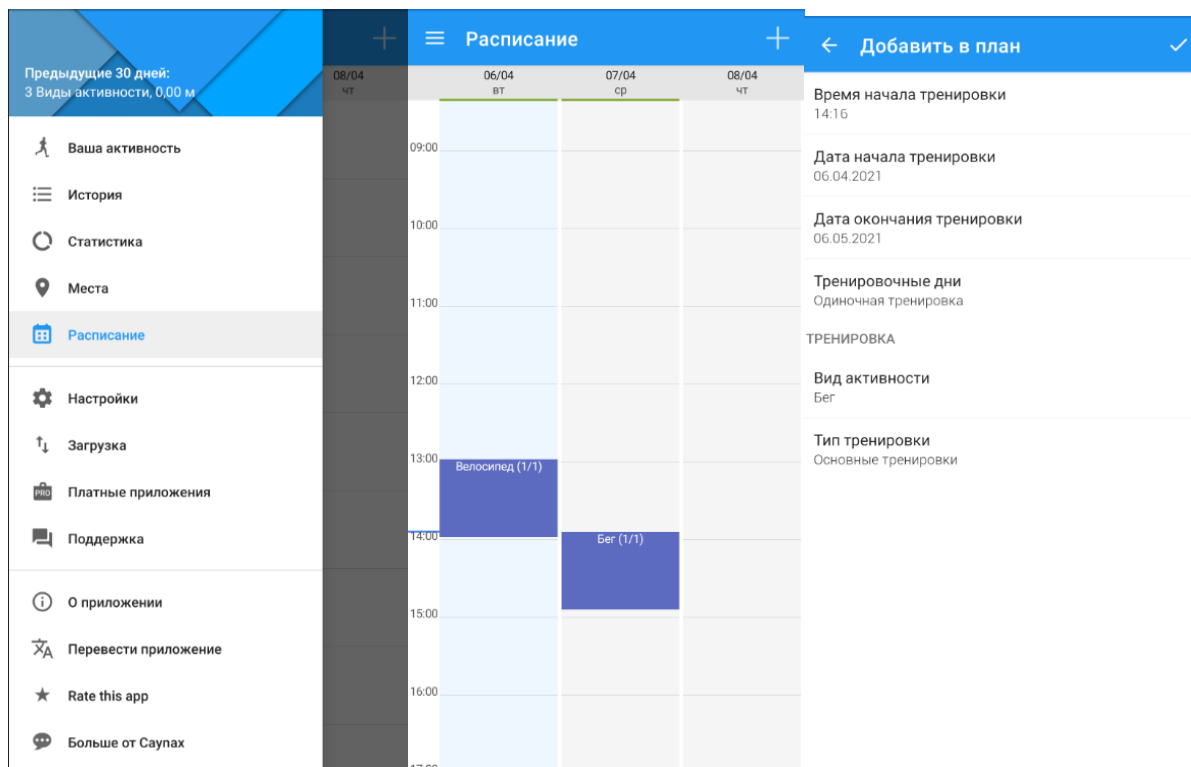


Рисунок 1.16 Экран розкладу тренувань застосунку «Caynax Sports Tracker»

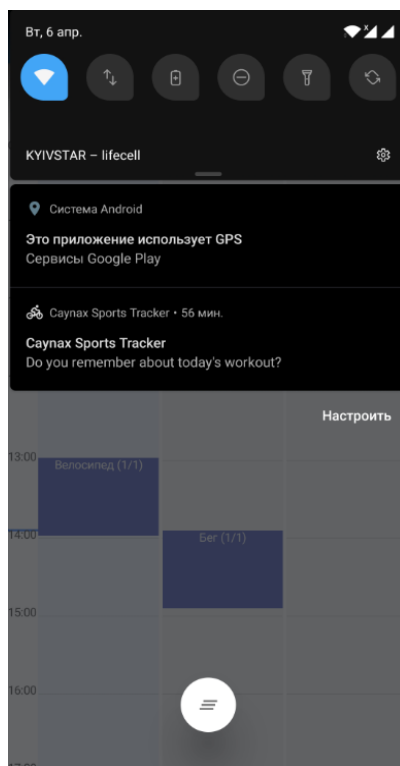


Рисунок 1.16 Экран вхідних повідомлень операційної системи Android

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

20

Як на мене, сервіс має декілька вагомих недоліків:

- 1) відсутність авторизації до власного акаунту, а саме відсутність єдиної історії прогресу тренувань на декількох смартфонах;
- 2) відсутність обробки аварійного завершення додатку під час тренування, що несе за собою втрату прогресу тренування.

1.2.4 Runmaster

Runmaster - ще один відомий сервіс тренувань і моніторингу спортивних показників. Притаманні тренування таких видів спорту як: «біг», «піший туризм», «їзда на велосипеді», «скандинавська ходьба», «сноуборд» ін.

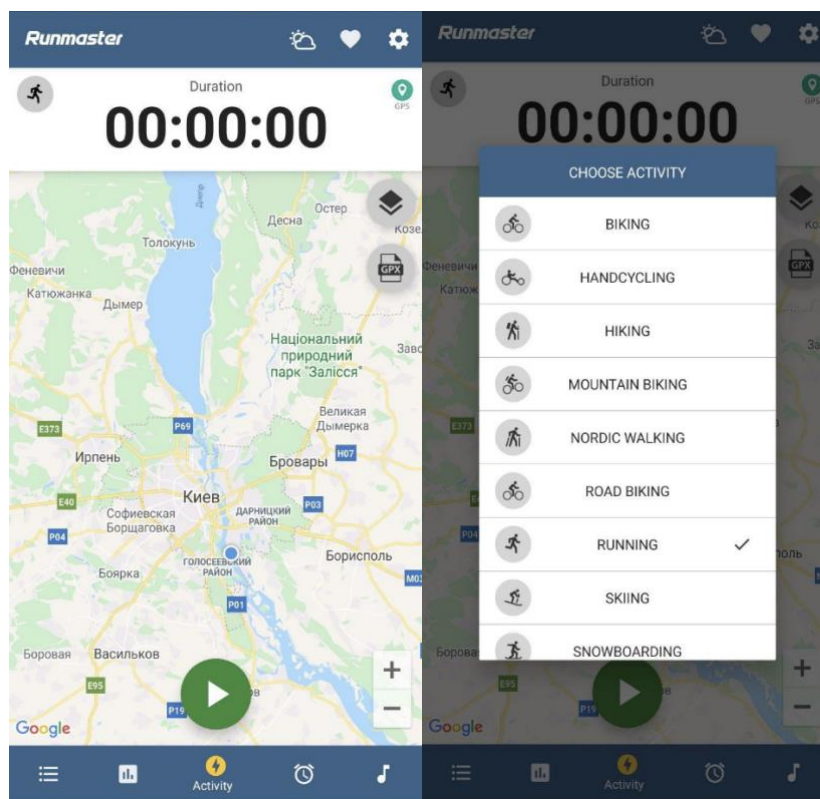


Рисунок 1.17 Головний екран застосунку «Runmaster»

Сервісу наявна багата кількість корисних функцій. Розглянемо деякі з них:

На головному екрані застосунку - одразу можна обрати тип тренування, розпочати тренування, мати під боком доступ до прогнозу погоди одразу на екрані тренування:

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

21

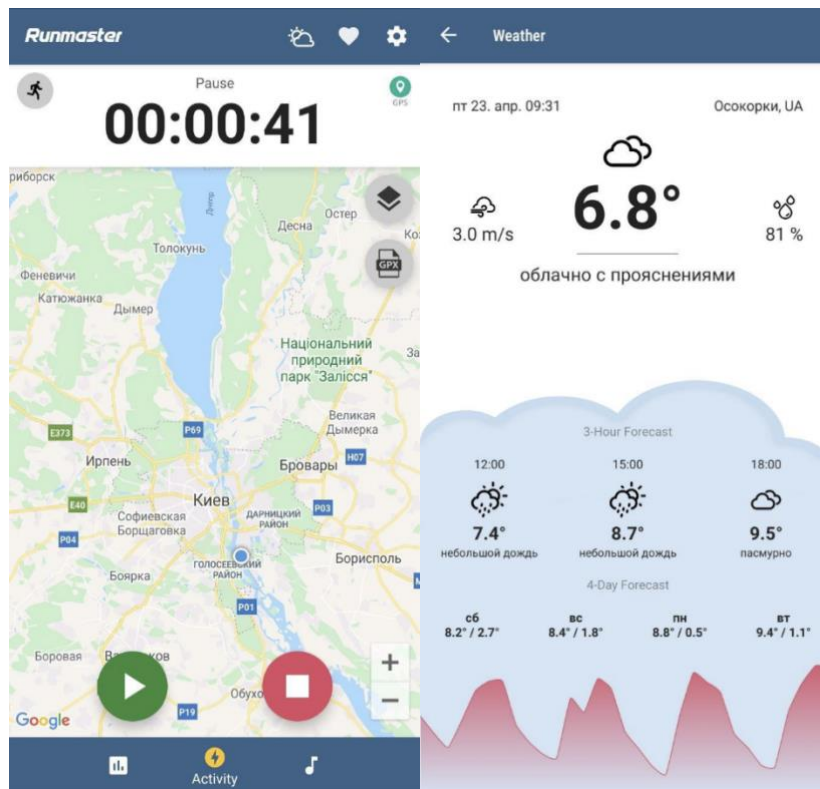


Рисунок 1.18 Экран активного тренування додатку «Runmaster»

Адже під час довгого, виснаженого тренування важливо слідкувати за показниками погоди: температури повітря, вологістю та опадами, оскільки великий шанс захворіти\перемерзнути після такого тренування.

В застосунку присутні налаштування профілю користувача: окрім локального профілю можна підключити Google акаунт, що дозволяє мати єдину синхронізовану історію тренувань на декількох девайсів. Також, при використанні стороннього пульсометру - приладу для слідкування за показником пульсу тіла, можна долучити його до застосунку за допомогою технологій: Bluetooth Smart і Ant+, що в свою чергу, дає можливість індивідуально налаштувати гранично-допустимі показники пульсу в певних зонах: від зеленої (комфортної) до червоної (тяжкої). У випадку перевищення встановлених планок програма проінформує користувача, що бажано знизити навантаження.

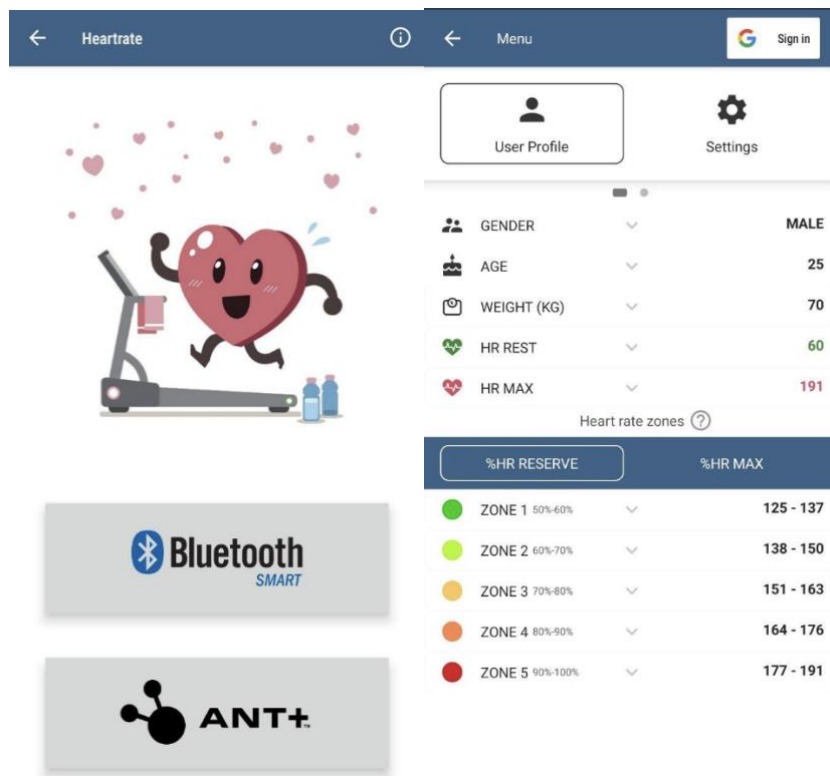


Рисунок 1.19 Екрани підключення пульсометру і налаштування профілю користувача застосунку «Runmaster»

Як і в Саунах Sports Tracker у застосунку відсутній функціонал побудови, редагування маршрутів, однак доступна можливість підгрузити існуючий маршрут формату GPX для подальшого використання при тренуванні.

Присутній зручний календар тренувань (планувальник) у системі для інформування користувача про заплановану пробіжку або велоїзд. При додаванні нового плану можна обрати як тип тренування, так і частоту виклику нагадувань про майбутні тренування за наступними критеріями:

- Одиночне нагадування;
- Щоденне;
- Кожні 2 дня;
- Кожні 5 днів;
- Кожен тиждень в певний час;
- Кожні 2 тижні.

Таким чином, користувач може налаштувати планувальник, що раз в 2 дні буде виходити на пробіжку і раз на тиждень тренуватися на велосипеді.

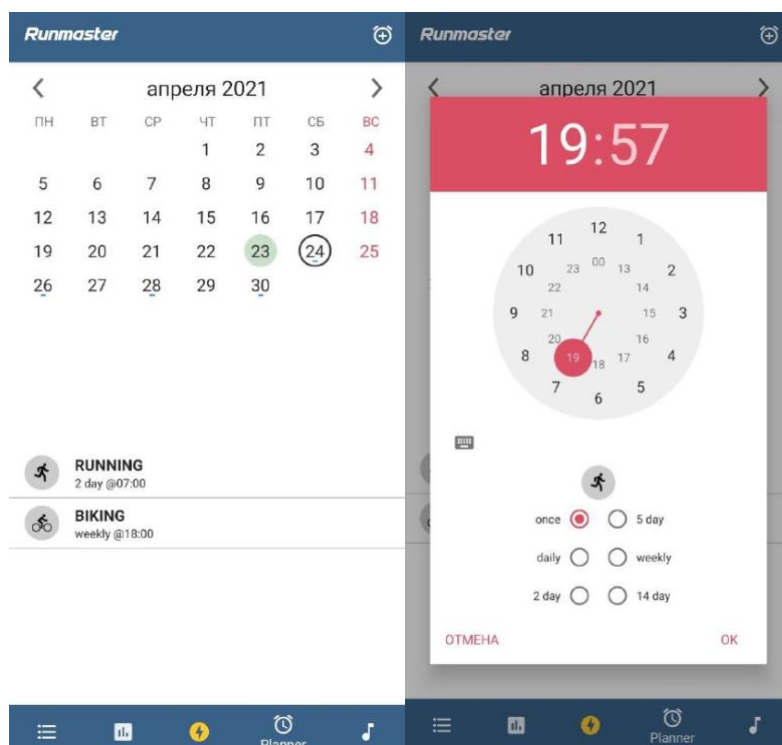


Рисунок 1.20 Екран планування майбутніх тренувань застосунку «Runmaster»

Однозначно серед функціоналу слід виділити зручну статистику тренувань, з зрозумілим графіком тижневих і місячних активностей і вибіркою основних показників: «час», «дистанція», «підйоми», «кількість затрачених калорій» тощо.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

24

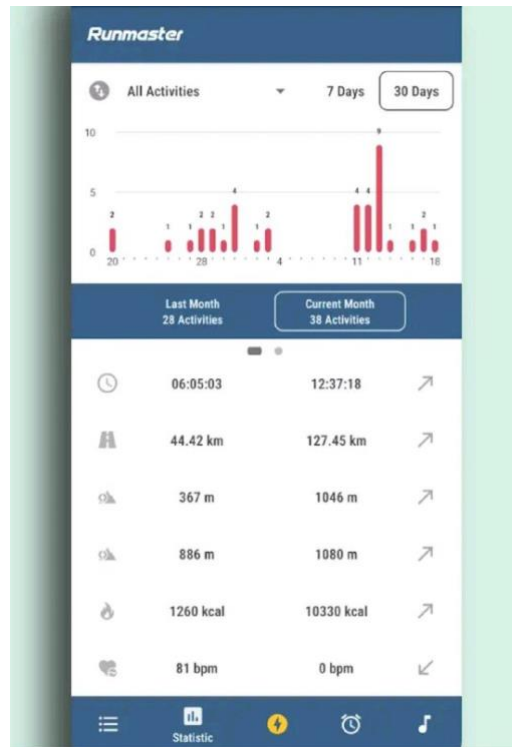


Рисунок 1.21 Екран статистики тренувань застосунку «Runmaster»

Обробка аварійного завершення додатку під час тренування відсутня - що несе за собою втрату прогресу тренування.

1.3 Порівняння оглянутих рішень

Таблиця 1.1 Порівняння застосунків

Назва застосунку	Проведення тренування	Аудіо супроводження	Історія власних тренувань	Історія тренувань друзів
Nike Run Club	+	+	+	-
Strava	+	-	+	+
Caynax Sports Tracker	+	-	+	-
Runmaster	+	-	+	-

Таблиця 1.2 Порівняння застосунків

Назва застосунку	Challenges	Групові тренування	Побудова маршрутів	Підгрузка існуючих маршрутів
Nike Run Club	+	-	-	-
Strava	+	+	+	-
Caynax Sports Tracker	-	-	-	+
Runmaster	-	-	-	+

Таблиця 1.3 Порівняння застосунків

Назва застосунку	Авторизація у додатку	Зміна теми додатка	Розклад тренувань	Обробка аварійного завершення під час тренування
Nike Run Club	+	-	-	+
Strava	+	-	-	+
Caynax Sports Tracker	-	+	+	-
Runmaster	+	-	+	-

Таблиця 1.4 Порівняння застосунків

Назва застосунку	Перегляд погоди у застосунку	Перегляд статистики тренувань	Перегляд тренувань друзів
Nike Run Club	+	+	-
Strava	-	+	+
Caynax Sports Tracker	-	+	-
Runmaster	+	+	-

ВИСНОВКИ ДО РОЗДІЛУ 1

В цьому розділі були розглянуті декілька застосунків для моніторингу тренувань користувача. Були розкриті, як переваги та і недоліки кожного рішення, проте жодне рішення не покриває повну функціональність, що описана в завданні даної роботи. Головні функції – єдина система для декількох спортивних індустрій(VELO і Біг), прокладення і редагування маршрутів для майбутніх тренувань, можливість користувачу обирати необхідну дію(продовжити, зберегти чи видалити тренування) після аварійного завершення додатка при активному тренуванні ін.

Беручи до уваги вищезгадані зауваження, розробку власного додатку можна вважати доцільним, адже функціональність додатку перекриває альтернативні рішення.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

РОЗДІЛ 2 ВИБІР ТЕХНОЛОГІЙ І ПЛАТФОРМ

2.1 React Native

React Native – відносно нове рішення у світі розробки мобільних застосунків. Цей фреймворк презентований Facebook у 2015 році, головною метою якого є прискорене написання кросплатформених додатків для декількох операційних систем з мінімальними відхиленнями у реалізації. На даний момент React Native підтримує наступні операційні системи: Android, IOS, Windows.

В минулому було небагато схожих кросплатформених рішень, а котрі були – сирі, що не підходили для використання у продакшені для кінцевих користувачів. Багато часу займав пошук розробників окремо під Android і IOS, подальше проектування, написання застосунку під усі ОС, що потребує замовник\власник додатку. Для виконання застосунку на базі ОС Android - використовуються мови Java/Kotlin, для IOS – Objective C/Swift. Що вело до повністю незалежної розробки застосунку.

React Native застосунки написані на мові JavaScript, котрий деякий час займав лише нішу Web сайтів і сервісів, згодом, JavaScript з'явився на обробці серверних рішень на платформі NodeJS, що також привело для мінімізації затрат часу написання сервісів і розробники, котрі в змозі підтримувати серверну і клієнтську частину додатків, ставали все більш цінними. Отож поява JavaScript у мобільних застосунків була лише питанням часу. Хоча додаток і написаний на JS, кінцеве, скомпільоване рішення виконується на смартфоні як і нативні рішення з використанням (Java для Android, Objective C для iOS).

Застосунок написаний на RN виконується в окремих потоках:

- 1) Головний потік – він запускається абсолютно у кожному нативному застосунку. Він обробляє відображення елементів користувацького інтерфейсу, жести користувача, різні події для зміни екранів застосунку тощо.
- 2) JavaScript Runtime потік – його зона відповідальності як раз таки виконання написаного JavaScript коду.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.1 Взаємодія нативних застосунків на рівні ОС(Android, IOS)

Як можна побачити на рис. 2.1 і взявши до уваги вище згадані аспекти написання окремих додатків під Android/IOS, виділяються певні недоліки: необхідно вивчити два стеки технологій, дублювати функціональність і двічі тестувати обидва додатки.

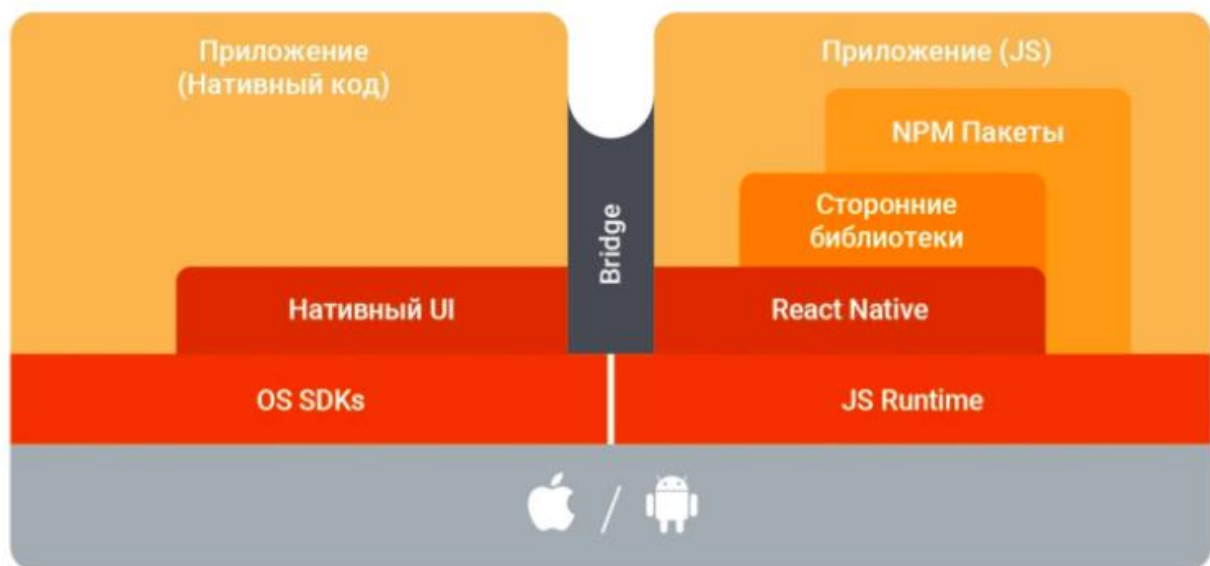


Рисунок 2.2 Взаємодія «React Native» застосунків на рівні ОС(Android, IOS)

Взглянувши на рис 2.2, можна помітити, що фігурує такий рівень абстракції - Bridge. Саме завдяки Bridge(міст) - JS Runtime взаємодіє з потіком ОС. При першому запуску застосунку, створюється головний потік, що

відповідає за користувацький інтерфейс, як було згадано вище, котрий в свою чергу породжує тінювий потік для виконання, обчислень JS частини. Взаємодія побудована на основі підписки подій, взаємодія двонаправлена, змінені дані відправляються у серіалізованому форматі JSON. Стоїть зазначити, що взаємодія асинхронна, тобто в момент відправки даних, наприклад з JS потоку до головного, JS потік не блокується чекаючи результату, а продовжує виконувати JS частину реагуючи на дії користувача.

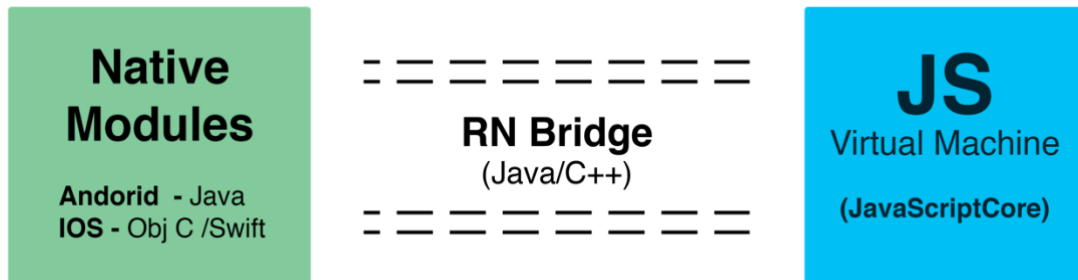


Рисунок 2.3 Більш детальний огляд «React Native» Bridge

2.1.1 Порівняння React Native з іншими рішеннями

Провівши дослідження, можна виділити наступні популярні технології для написання кросплатформених застосунків, крім вищезгаданого React Native: Flutter, Xamarin.

Таблиця 2.1 Порівняння кросплатформених рішень

Технологія	React Native	Flutter	Xamarin
Компанія	Facebook	Google	Microsoft
Мова програмування	JavaScript	Dart	C#, .Net core
Типізація з коробки	-	+	+
Написання адаптованих стилів додатку	+	-	-
Компіляція в бінарний код	-	+	+

В порівнянні React Native з іншими технологіями, одним з головних переваг є можливість написання адаптованих стилів для додатку, пере-використання вже написаних стилів, що приводить к меншим затратам часу написання додатка.

Оскільки JavaScript динамічна мова, в React Native відсутня статична типізація з коробки в порівнянні з типізованими мовами Dart і C#, але стоїть зазначити, що React Native підтримує оточення TypeScript, JSDoc, Flow для анотації типів, написання інтерфейсів, що зменшує час розробки і дозволяє писати більш типізований, безпечний код.

Досить вагомим аспектом у виборі оточення для проекту є підтримка і масштаби спільноти технології. React Native спільнота в декілька раз більше вищезгаданих рішень і підтримує значно більшу кількість готових компонентів, віджетів, кінцевих рішень такі як кнопки, таблиці, списки, бібліотеки для використання чатів, навігації у додатку тощо.

2.1.2 Порівняння стилізації React Native і Android додатків

У React Native досить зручна стилізація екранів застосунку або певних компонентів. Нижче приведений приклад стилізації компонента:

```
import React from 'react';
import { Text, View, StyleSheet } from 'react-native';

const styles = StyleSheet.create({
  center: {
    alignItems: 'center'
  }
});

const Greeting = (props) => {
  return (
    <View style={styles.center}>
      <Text>Hello {props.name}!</Text>
    </View>
  );
};

const LotsOfGreetings = () => {
  return (
    <View style={[styles.center, {top: 50}]}>
      <Greeting name='Rexxar' />
      <Greeting name='Jaina' />
      <Greeting name='Valeera' />
    </View>
  );
};
```

Hello Rexxar!
Hello Jaina!
Hello Valeera!

Рисунок 2.4 Приклад використання вбудованих стилів у застосунку «React Native»

Як можна побачити на рис 2.4, присутні компоненти Greeting і LotsOfGreetings і окрема сутність styles, що являється обгорткою поверх JS об'єктів, що добре кешується і створюється лише в момент ініціалізації додатку. Такий підхід дозволяє пере-використовувати існуючі стилі, також, додавати нові. На жаль, вбудований StyleSheet не підтримує динамічні стилі, що залежать від зовнішніх показників, наприклад від ширини екрану. У такому випадку необхідно дописувати окремий об'єкт з динамічними стилями у самому компоненті, що може порушувати відповідальність компонента, оскільки окрім відображення даних, компонент має обчислювати стилі. Враховуючи вищезгадане зауваження, що у React Native велика спільнота і база готових бібліотек\рішень, розробник може підключити до проекту будь-яку сторонню бібліотеку для стилізації компонентів замість вище згаданого StyleSheet, наприклад популярною є Styled Components:

					ІАЛЦ.467100.003 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

```
const Button = styled.button`
  /* Adapt the colors based on primary prop */
  background: ${props => props.primary ?
    "palevioletred" : "white"};
  color: ${props => props.primary ? "white" :
    "palevioletred"};

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;

render(
  <div>
    <Button>Normal</Button>
    <Button primary>Primary</Button>
  </div>
);
```

Normal

Primary

Рисунок 2.5 Приклад використання styled-components у застосунку «React Native»

Як можна побачити на рис. 2.5, на основі одного параметра «primary» ми в змозі адаптувати стилі кольору тексту і фона кнопки не міняючи, додаючи логіку компонента, достатньо підправити Button, що відповідає за стилізацію, що може знаходитися в окремому файлі.

Styled Components дозволяє писати майже однаковий код для мобільних і веб рішень, саме завдяки цьому він і здобув популярності. Адже якщо ви маєте сайт і застосунок, наприклад продажу товарів, то цілком виправданим рішенням буде взяти наступний стек технологій: React для веб, React Native для мобільної частини, styled-components для стилізації компонентів. У результаті зменшиться велика кількість однакового кода, даний стек приносить швидкість розробки і пере-використання компонентів і стилів.

Щодо стилізації екранів операційної системи Android, усі стилі використовують свій xml формат, що описується в окремих файлах, наприклад styles.xml. У Android відсутня можливість використовувати динамічні стилі так, як це реалізовано в React Native. У випадку Android, необхідно описати xml стилі:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <TextView
9         android:id="@+id/textView1"
10        android:layout_width="0dp"
11        android:layout_height="wrap_content"
12        android:gravity="center"
13        android:textSize="28sp"
14        android:textColor="#3f51b5"
15        android:text="Android Lollipop"
16
17        app:layout_constraintLeft_toLeftOf="parent"
18        app:layout_constraintRight_toRightOf="parent"
19        app:layout_constraintTop_toTopOf="parent"
20        app:layout_constraintBottom_toBottomOf="@+id/textView2"
21    />

```

Рисунок 2.6 Приклад використання стилів у Android застосунку

Як можна побачити у `TextView` присутній атрибут `android:id` - це унікальний ідентифікатор, що необхідний для звертання до `TextView` з Java\Kotlin коду:

```

setContentViews(R.layout.activity_main);
TextView textElement = (TextView) findViewById(R.id.textView1);
textElement.setTextColor(0xFF00FF00); //this is green color

```

У порівнянні з React Native, такий підхід набагато складніший і потребує зайвих рухів.

2.2 Менеджер стану. Mobx

У більш-менш великих проєктах, зазвичай, присутня велика кількість даних, які можуть бути нормалізованими, тобто вже налаштованими для відображення на екрані користувача, так і не нормалізованими – це сирі дані, що були отримані зі стороннього джерела – API, бази даних тощо. Проблема не нормалізованих даних, як було зазначено вище – це сирі дані, котрі необхідно трансформувати у певну структуру\ певний вигляд для відображення. Для організації таких даних, зазвичай, використовуються бібліотеки, що несуть собою State Manager, тобто менеджер стану.

Одним із таких рішень є популярна бібліотека Mobx.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35



± Weekly Downloads

514,111



Рисунок 2.7 Кількість завантажень Mobx бібліотеки

Девіз команди розробників Mobx: «Anything that can be derived from the application state, should be. Automatically»[2]. Тобто, мається на увазі, що усе, що може бути отримано зі стану додатка повинно бути автоматичним, не важливо якого виду дані – нормалізовані чи ні.

Досить велика кількість подібних бібліотек і все ж на фоні інших Mobx виділяє певна ідеологія - Mobx реактивний, застосовує патерн Observer.

Реактивне програмування — це парадигма програмування, побудована на потоках даних і розповсюдженні змін. Це означає, що у мовах програмування має бути можливість легко виразити статичні чи динамічні потоки даних, а реалізована модель виконання буде автоматично розсилати зміни через потік даних. Наприклад, в імперативному програмуванні вираз, $a := b + c$ означає, що a отримує результат виконання $b + c$ безпосередньо під час обчислення виразу, і потім, якщо значення b і c зміняться, це не впливатиме на вже обчислене значення a . Проте, в реактивному програмуванні значення a буде автоматично оновлено за новими значеннями[3].

Концепція Mobx складається, загалом, з трьох речей:

- Observable states – певні поля стану даних, що є спостерігачами;
- Actions – певні функції, що змінюють значення полів спостерігачів, що в свою чергу миттєво змінюються у всіх місцях, де це значення використовується;
- Computed derivations – це гарячі, швидко-обчислювальні дані, що миттєво вираховується при зміні полів спостерігачів. Зазвичай, використовуються для нормалізації даних.

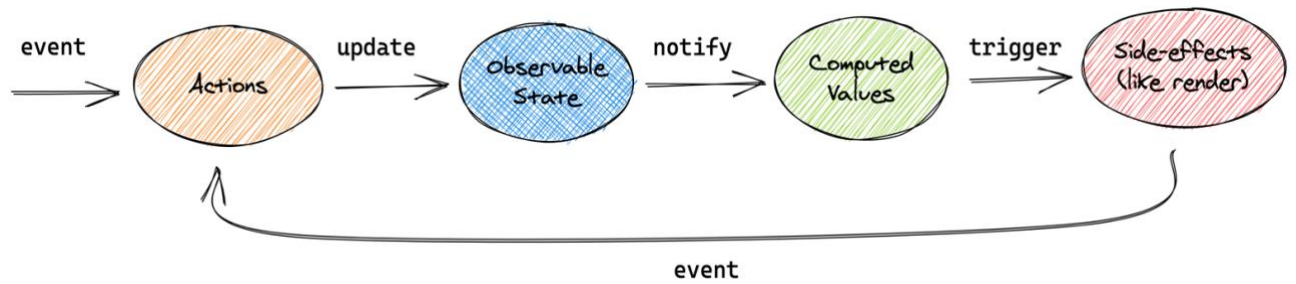


Рисунок 2.8 Схема взаємодії Mobx

Приклад використання Mobx:

```

//Опис класу стану таймеру
class Timer {
  //поле спостерігач - кількість секунд
  secondsPassed = 0

  constructor() {
    //вбудована функція mobx, що обгортає клас таймеру для реактивності
    makeAutoObservable(this)
  }

  //action, що змінює значення спостерігача, а саме збільшує значення на 1
  increase() {
    this.secondsPassed += 1
  }

  //action, що змінює значення спостерігача, а саме повертає значення до 0
  reset() {
    this.secondsPassed = 0
  }

  //computed - значення подоланого часу у мілісекундах. Оновлюється автоматично при
  зміні значення спостерігачу secondsPassed
  get msPassed() {
    return this.secondsPassed * 1000
  }
}

```

```
//створення екземпляру класу стану. Об'єкт що несе в собі стан і функції для виклику
const myTimer = new Timer()

//певний компонент що відповідає за відображення даних
const TimerView = observer(({ timer }) => (
  <>
    <button onClick={() => timer.reset()}>Seconds passed:
{timer.secondsPassed}</button>
    <div>Milliseconds passed: {timer.msPassed}</div>
  </>
))

ReactDOM.render(<TimerView timer={myTimer} />, document.body)

// Виклик екшину для збільшення secondsPassed
setInterval(() => {
  myTimer.increase()
}, 1000)
```

Як можна побачити вище на прикладі використання Mobx, необхідно описати клас стану з певною логікою, що несе в собі концепції, згадані вище, а саме спостерігач, функції для зміни значення спостерігача, computed значення, що автоматично вираховується при зміні спостерігача. Вже у React компоненті (або у будь-якій іншій бібліотеці, фреймворку для відображення) використовується значення зі стану і викликаються функції для зміни значення стану, наприклад reset(). Також є підтримка виклику функцій за межами компонента, оскільки це зона відповідальності об'єкту стану, а у випадку зміни значення стану, компонент TimerView автоматично оновить нове значення.

Саме такий підхід дозволяє спроектувати архітектуру проєкту так, що компоненти будуть відповідати лише за відображення нормалізованих даних або виклик змінення даних для нормалізації.

					ІАЛЦ.467100.003 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

2.2.1 Порівняння Mobx з Redux

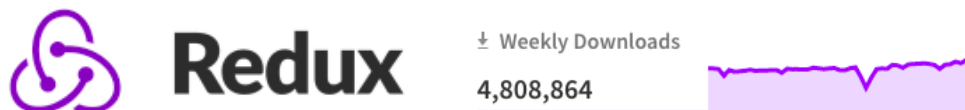


Рисунок 2.9 Кількість завантажень Redux бібліотеки

Redux – ще один популярний менеджер стану, що відповідає за нормалізацію даних у готовий для відображення вигляд. З рис. 2.9, можна зробити висновки, що Redux більш популярний за Mobx, оскільки більш простий для розуміння взаємодії менеджера стану з додатком.

Головна концепція Redux складається з:

- Store – стан додатку
- Actions – функції для виклику зміни стану
- Reducers – функції що приймають зміни з actions і оновлюють стан.

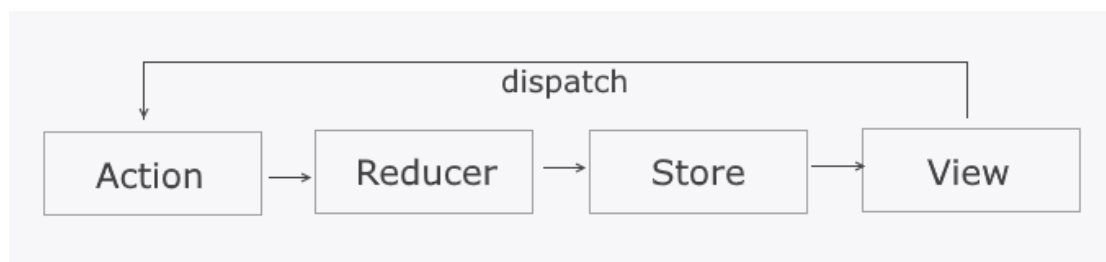


Рисунок 2.10 Схема взаємодії Redux

Концепція трохи схожа з Mobx, але є декілька важливих зауважень:

- 1) Redux за замовчуванням синхронно опрацьовує дані, тобто виклик змін стану йде один за одним. У разі звернення до API у функції action, необхідно дочекатися відповіді від API і тільки після результату передати дані до reducer для опрацювання і нормалізації й перейти до наступного action. Існують різні бібліотеки, що дозволяють викликати actions асинхронно, але такі рішення виходять за ідеологію Redux. Mobx за замовчуванням підтримує асинхронні операції.

2) Redux не реактивний і не несе в собі спостерігачів. Для зміни стану у reducer, необхідно повертати нову структуру, на основі старої, з новими змінами у стані. Тобто якщо у стані є профіль користувача з наступними даними –

{firstname: «Tom», lastname: «Henk», age: 30} і існує action для зміни ім'я – *changeName(name: «Denis»)*, то reducer в свою чергу приймає передане ім'я і має створити і повернути новий об'єкт на основі попереднього зі зміненим ім'ям – *{firstname: «Denis», lastname: «Henk», age: 30}*

3) Беручи зауваження з пункту 2, Redux несе в собі недолік – більша кількість однотипного коду і створення нових структур перед зміною стану, що несе додаткові затрати на використання пам'яті. Коли у Mobx для оновлення поля *firstname* достатньо викликати action *changeName(name: «Denis»)*, що одразу оновить значення ім'я завдяки реактивності.

2.3. Mapbox

Щодо вибору платформи відображення гео-інформації мапою, існує велика кількість сервісів: Google Maps, Street Maps, Mapbox, Yandex Maps тощо. Після більше детального дослідження, було прийнято рішення обрати Mapbox. Переглянемо вибірку критеріїв, за якими було обрано Mapbox:

- Наявність документації;
- Наявність влаштованого інструмента для редагування мап;
- Наявність SDK під усі популярні операційні системи;
- Наявність бібліотеки-обгортки для зручної взаємодії з React Native.

2.3.1 Mapbox документація

В наш час кількість технологій постійно збільшується, існуючі рішення постійно оновлюються. Сервіс, технологія повинна нести собою, окрім функціоналу, документацію як працювати з даним функціоналом, що в свою чергу, зменшує кількість затрачених годин розробника на кінцеву реалізацію

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

специфічних деталей проєкта. Марбох має зручний сайт з актуальною документацією, що постійно оновлюється зі змінами самого Марбох.



Рисунок 2.11 Головна сторінка сайту документації Марбох

Як можна побачити на рис. 2.11, для зручної навігації на сайті присутній пошук і навігаційні клавiши у шапці сайту.

При прокрутці сторінки або натиску навігаційних клавiш з шапки сайту, можна одразу перейти до шуканої інформації.

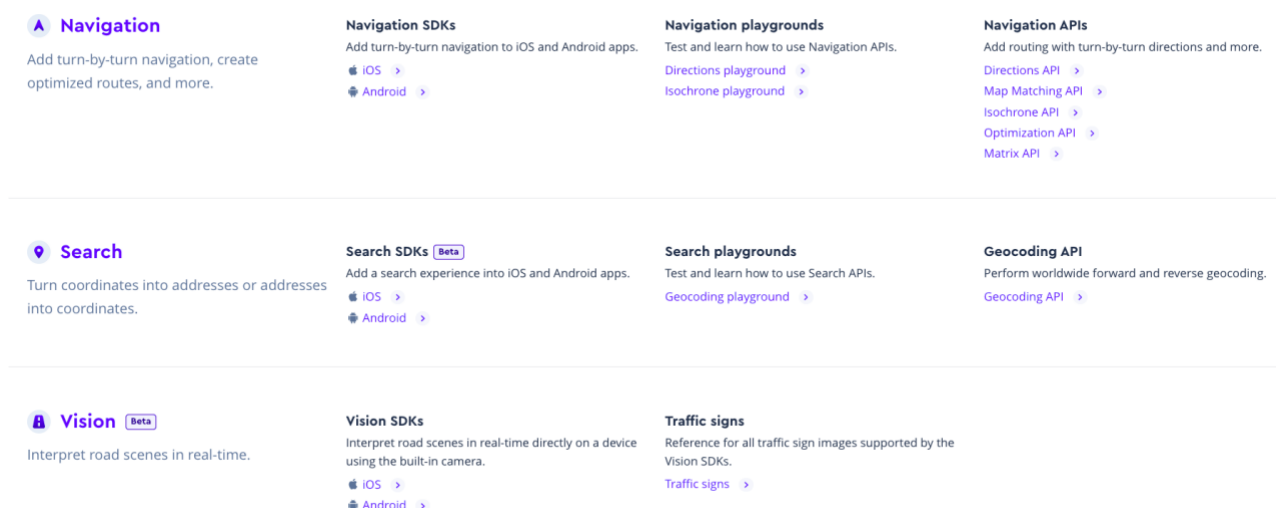


Рисунок 2.12 Навігація по документації Марбох

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

Розглянемо приклад з Directions API. Це API, що відповідає за прокладення маршрутів для пересування. При переході на відповідну сторінку відображається список можливих викликів API, з детальним описом параметрів виклику. Не менш важливим є те, що присутня можливість одразу перевірити виклик API у вбудованому оточенні Playground:

Directions

On this page

- Retrieve directions
- Use HTTP POST to retrieve directions
- Waypoint object
- Route object
- Route leg object
- Route step object
- Step maneuver object
- Lane object
- Voice instruction object
- Banner instruction object
- Incident object
- Instructions languages
- Directions API errors
- Directions API restrictions and limits
- Directions API pricing

Directions API version

This documentation is for v5 of the Directions API. For information about the earlier version, see the [v4 documentation](#).

The Mapbox Directions API will show you how to get where you're going. With the Directions API, you can:

This API supports 4 different routing profiles:

Profile	Description
mapbox/driving-traffic	For automotive routing. This profile factors in current and historic traffic conditions to avoid slowdowns. Traffic information is available for the supported geographies listed in our Traffic Data page . Requests using this profile accept up to three coordinates.
mapbox/driving	For automotive routing. This profile shows the fastest routes by preferring high-speed roads like highways. Requests using this profile accept up to 25 coordinates.
mapbox/walking	For pedestrian and hiking routing. This profile shows the shortest path, including sidewalks and trails. Requests using this profile accept up to 25

Optional parameters

alternatives ☐ return alternative routes

steps ☐ enable turn-by-turn instructions

geometries ☐ format of the returned geometry

geotjson polyline polylinelns

Use the line tool to draw your route.

Request URL

<https://api.mapbox.com/directions/v5/{profile}/{coordinates}>

Response

You haven't created an API request yet.

Рисунок 2.13 Перегляд документації Марбох для прокладення маршруту

2.3.2 Mapbox Studio

Mapbox Studio – це інструмент для редагування мап. Досить поширений сценарій використання додатків, коли недостатньо вбудованого функціоналу мап. Буває що потрібно додати певний функціонал з централізованого інструмента для усіх користувачів: додаткові позначки, змінити колір будівель\доріг, додати відображення 3D будівель тощо.

На головній сторінці відображається список змінених карт користувача. Присутній функціонал пошуку та створення нової карти зі змінами.

Styles

Upload style

New style

Search

All

< >

5 of 5 styles



Outdoors

Published Sep 28, 2020 • Public

🔗 ⋮



Example: Add 3D buildings

Edited Sep 28, 2020 • Published Jun 2, 2020

🔗 ⋮



Example: Add 3D buildings

Published Jun 2, 2020

🔗 ⋮



Example: Add 3D buildings

Edited Jun 2, 2020 • Published Jun 2, 2020

🔗 ⋮



Example: Add 3D buildings

Published Jun 2, 2020

🔗 ⋮

< >

Рисунок 2.14 Головна сторінка Mapbox Studio

Наприклад, можна додати 3D відображення будівель і змінити колір на більш жовтий. Після внесення змін – достатньо натиснути Publish у верхньому правому куту для отримання посилання на мапу, що буде використовуватися у додатку. Такий підхід дійсно зручний оскільки можна мати різні мапи для різних користувачів.

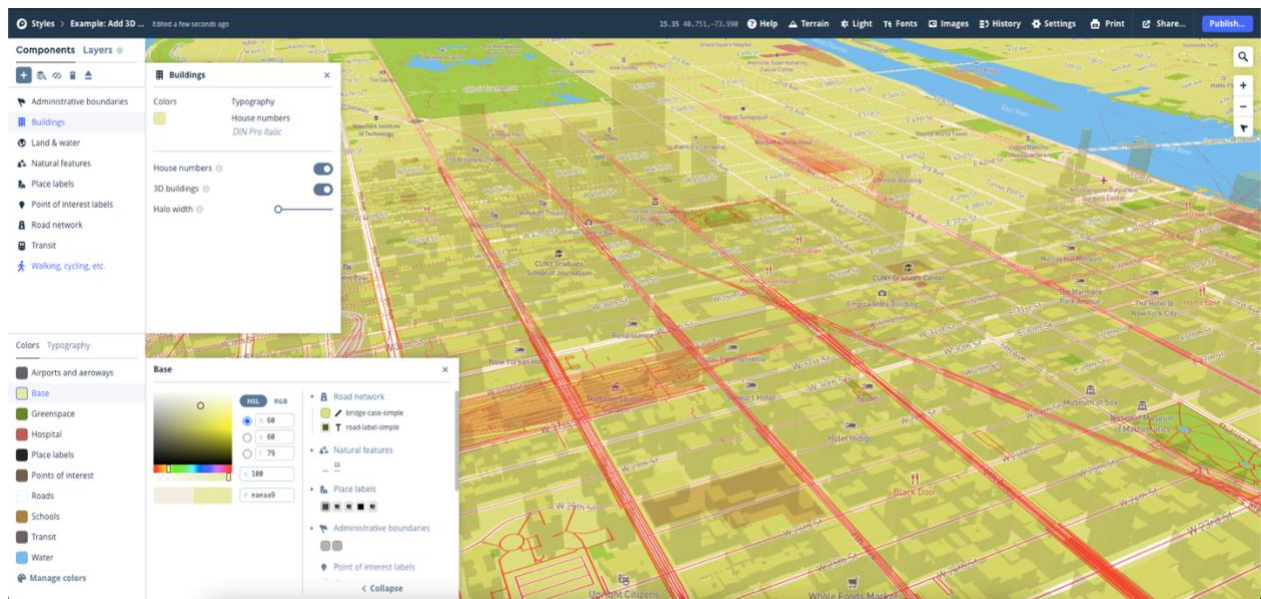


Рисунок 2.15 Сторінка редагування мапи у Mapbox Studio

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

2.3.3 Mapbox SDK

Mapbox має SDK для всіх популярних платформ:

- Web з використанням мови JavaScript;
- Android з використанням мов Kotlin\Java;
- IOS з використанням мов Swift\Object C;

2.3.4 Mapbox бібліотека під React Native

Mapbox також має зручну офіційну бібліотеку під React Native - `@react-native-mapbox-gl/maps`, що постійно оновлюється. Було прийнято рішення використовувати у розробці саме цю бібліотеку.

2.4 Сховище даних Firebase

Firebase – це хмаровий сервіс від компанії Google, що поєднує в собі бази даних, провайдери авторизації, розміщення сайтів, аналітику тощо.

Для баз даних Firebase використовує не реляційні моделі даних, так звані NoSql. База має документо-орієнтований формат схожий на MongoDB.

Нижче приведено вибірку причин за якими було обрано Firebase як сховище даних:

- NoSql база даних;
- Сховище під файли;
- Сервіс авторизації;
- Офлайн доступ до бази даних, робота з кешуванням;
- Інструмент для перегляду\редагування даних для адміністраторів.

Firebase підтримує офлайн доступ до сховища. Тобто, окрім основного хмарного сховища, створюється сховище локально у користувача. Є можливість налаштувати взаємодію зі сховищем так, що пріоритет на запит даних буде у локального сховища на читання, запис. Сховища взаємодіють між собою завдяки двонаправленого механізму підписки подій. В момент появи інтернет з'єднання Firebase опрацьовує події, що виникли під час офлайн взаємодії і оновлює різницю як на хмарі, так і у локальному сховищі у разі необхідності.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

Firebase поєднує велику кількість провайдерів авторизації. За описом завдання дипломної роботи витікає, що для додатку необхідно підключити провайдери: Email\Password, Google, що присутні у Firebase.

Sign-in providers

Provider	Status
 Email/Password	Enabled
 Phone	Disabled
 Google	Enabled
 Play Games	Disabled
 Game Center	Disabled
 Facebook	Disabled
 Twitter	Disabled
 GitHub	Disabled
 Yahoo	Disabled
 Microsoft	Disabled
 Apple	Disabled
 Anonymous	Disabled

Рисунок 2.16 Список провайдерів авторизації Firebase

ВИСНОВКИ ДО РОЗДІЛУ 2

В цьому розділі були розглянуті певні технології, що будуть використовуватися для написання системи трекінгу активностей, провели дослідження як влаштований React Native на рівні взаємодії з ОС, порівняли React Native з іншими схожими кросплатформеними рішеннями. Обрали менеджер стану, платформу для відображення мапи у додатку і платформу для сховища даних й підключення провайдерів авторизації.

Загалом, можна вважати доцільним переходити до наступного етапу - проектування і розробки системи.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ПРОЕКТУВАННЯ І РОЗРОБКА ДОДАТКА

3.1 Вимоги до системи

Вимогами до системи є опис того, що система повинна робити, які функції вона надає, та обмеження функціоналу. Зазначимо вимоги до застосунку:

- Функціональні вимоги – перелік дій або функцій, які система повинна підтримувати, виконувати;
- Вимоги до інтерфейсу – основні вимоги інтерфейсу користувача. Оскільки інтерфейс – головна частина застосунку зі сторони користувача, ці вимоги являються важливим етапом у процесі проектування і розробки додатку;
- Нефункціональні вимоги – визначають якість роботи системи. До них відносяться такі критерії: швидкість, безпека, надійність, адаптивність, масштабованість тощо.

3.1.1 Функціональні вимоги

У системі повинен бути наступний функціонал:

- поєднання декількох режимів трекингу(велосипед, біг, автозаїзд);
- моніторинг тренування – заміри таких показників як «дистанція», «середня швидкість\темп», «поточна швидкість\темп», «загальний час», «час в русі»;
- прокладення маршрутів;
- редагування маршрутів;
- перегляд збережених маршрутів;
- перегляд збережених тренувань;
- одиничне або множинне видалення обраних маршрутів;
- одиничне або множинне видалення обраних тренувань;
- початок тренування з відображенням обраного збереженого маршруту на мапі;
- авторизація користувача у системі;
- вибір теми стилізації додатку;

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

- вибір мови додатку;
- видалення, збереження або продовження тренування після аварійного завершення додатку;
- видалення, збереження або продовження побудови маршруту після аварійного завершення додатку;
- пошук місця по назві міста для швидкого переміщення мапи.

Прокладення маршрутів, має підтримувати декілька режимів побудови:

- 3) прямий режим – користувач обирає кожну наступну координату на мапі, що з'єднується з попередньою – прямою смугою, перетином мапи;
- 4) режим напрямків – користувач обирає кожну наступну координату на мапі, що з'єднується з попередньою смугами напрямків відповідно оптимальному автомобільному, велосипедному або пішому маршрутам.

Редагування має здійснюється перетаскуванням координат маршруту на мапі з наступним перерахунком дистанції маршруту.

У системі повинна бути розроблена авторизація користувача як з використанням пошти з паролем, так й вхід за допомогою Google.

При виборі мови додатку мають бути запропоновані українська(державна) і англійська мови.

3.1.2 Вимоги до інтерфейсу

Інтерфейс системи трекінгу активностей повинен забезпечити наступні вимоги:

- бути інтуїтивно-зрозумілим;
- мати унікальний дизайн націлений на user-friendly;
- мати підтвердження важливих операцій;
- бути UI-first, тобто бути націленим на відображення дії, після чого виконувати логіку.

3.1.3 Нефункціональні вимоги

Застосунок має бути націлений на дотримання наступних вимог:

- коректно працювати без доступу до інтернету;
- підтримувати операційну систему Android;
- система повинна обробляти виключні ситуації в разі їх виникнення;
- система має бути готовою для заміни сторонніх сервісів, у разі такої потреби.

3.2 Проектування системи

3.2.1 Авторизація

Authentication – це сервіс, що забезпечує аутентифікації користувачів. Сервіс використовує протокол OAuth 2.0 та OpenId Connect [4].

OAuth 2.0 – це фреймворк для авторизації, що надає додаткам обмежений доступ до призначених для користувача акаунтів на HTTP сервісах. Принцип роботи полягає у делегуванні аутентифікації користувача сервісу, дозволяючи застосунку отримувати доступ до акаунту користувача [5].

OpenID Connect - це рівень ідентифікації, що працює поверх вищезгаданого протокола. Він дозволяє клієнтам перевірити особу користувача на основі аутентифікації, виконаної сервером авторизації, а також отримати основну інформацію про профіль користувача в оперативному режимі та схожий на REST спосіб [6].

Для аутентифікації користувача, системі необхідно отримати певні облікові дані автентифікації від користувача, так звані credentials. Ці дані можуть бути як електронною адресою з паролем, так і токеном OAuth від постачальника ідентифікаційних даних. У випадку із системою трекингу активностей, система спочатку дістає OAuth токен у постачальника ідентифікаційних даних Google, і після чого передає дані до SDK для

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

аутентифікації Firebase. Таким чином сервіси Firebase перевіряють валідність і цілісність цих даних, повертають інформацію облікового запису користувача, після чого створюють окрему сесію для взаємодії з платформой на рівні застосунку.

Після успішного входу, користувач отримує доступ до основної інформації свого профілю і даних тренувань з маршрутами. Аутентифікація з використанням платформи Firebase досить зручний спосіб інтеграції сторонніх засобів ідентифікації користувача. Вона працює з такими постачальниками як: оператор мобільного зв'язку, Github, Apple, Google, Yahoo, Microsoft, Facebook, Twitter, адреса електронної пошти з паролем.

Взаємодія платформи з вищезгаданими постачальниками ідентифікаційних даних користувача надає системі трекингу активностей можливості для майбутньої масштабованості та інтеграції з іншими постачальниками, окрім Google, як зазначено у функціональних вимогах. Заміна провайдерів відбувається швидко та лише зі сторони клієнтів.

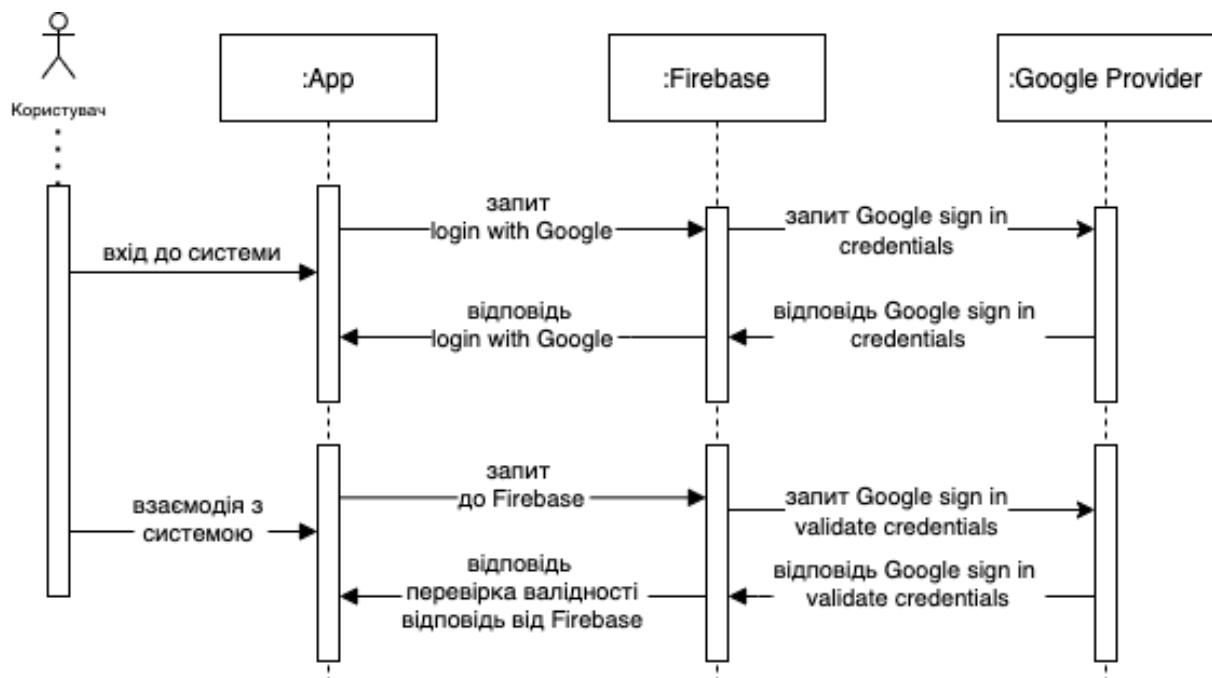


Рисунок 3.1 Sequence-діаграма авторизації користувача

OAuth токен заснований на JWT.

- Алгоритм та тип шифру;
- Певні дані користувача;
- Секретний рядок, що зберігається лише на сервері і не пересилається в запитах.



Адміністратор застосунку має доступ до панелі авторизованих користувачів Firebase. Присутній функціонал пошуку користувача, перегляду таких даних: постачальник, дата реєстрації, пошта, дату останньої авторизації, унікальний ідентифікатор:

					ІАЛЦ.467100.003 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

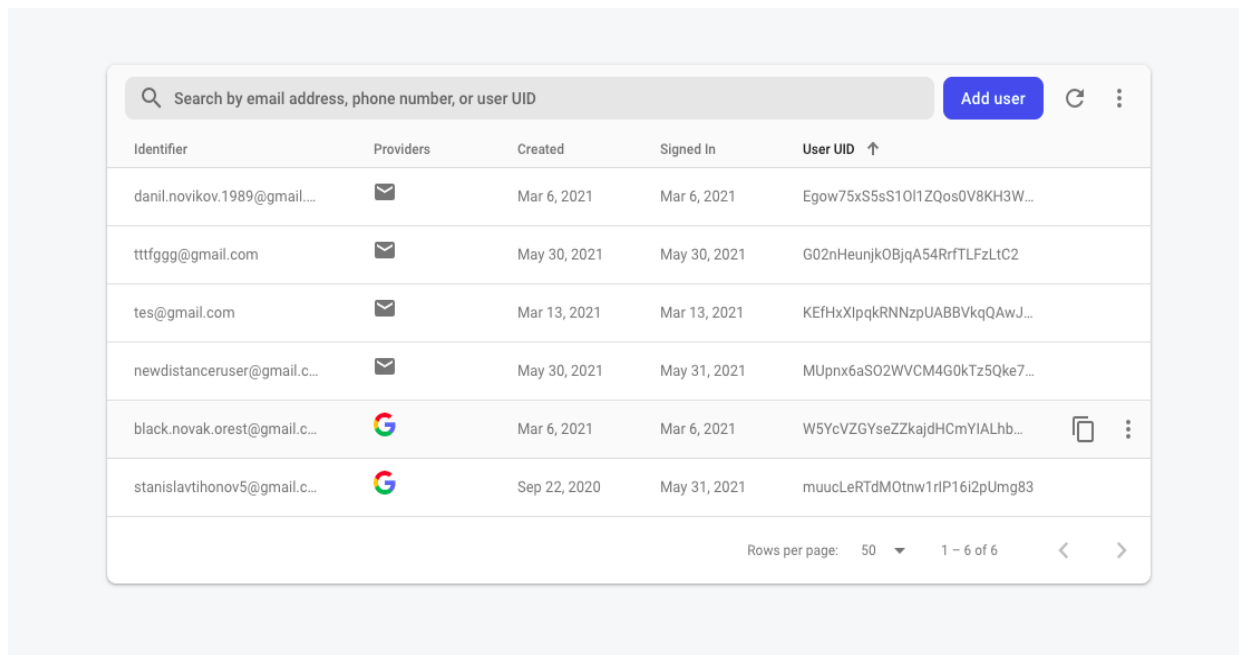


Рисунок 3.3 Панель авторизованих користувачів у Firebase

3.2.2 Firestore

Firestore – це документо-орієнтована NoSQL база даних від Firebase та Google Cloud Platform. Як ми зазначили на етапі вибору платформи сховища, основною перевагою даного сервісу є синхронізація даних з клієнтами завдяки двонаправленого механізму підписки подій у режимі реального часу. База даних має влаштоване автоматичне кешування, в результаті чого робота з нею є максимально зручною, оскільки весь процес синхронізації з хмарами відбувається автономно [7].

Сервіс зберігає дані у форматі документів, що поєднуються у колекції та підколекції. Кожен документ містить структуру об'єктів з полями. Також є можливість створювати підколекції в документах, проектувати ієрархічні структури даних, що будуть масштабуватися при зростанні самої бази даних. Запити при роботі з базою даних підтримують набори фільтрів та параметрів сортування, що індексуються за замовчуванням, тому ефективність запитів пропорційна розміру набору результатів, а не набору даних. Також не менш важливим є те, що інтеграція з Firebase Auth дозволяє захистити і зберегти конфіденційність даних за допомогою налаштувань прав доступу. У Firebase є

зручна консоль – інструмент для адміністраторів, де можна як раз таки налаштувати права доступу до сховища Firestore.

```

1  rules_version = '2';
2  service cloud.firestore {
3    match /databases/{database}/documents {
4
5      match /routes/users/{userId}/{document=**} {
6        allow read: if request.auth != null;
7        allow write: if request.auth != null && request.auth.uid == userId;
8      }
9      match /activities/{directionsMode}/{userId}/{document=**} {
10       allow read: if request.auth != null;
11       allow write: if request.auth != null && request.auth.uid == userId;
12     }
13     match /users/{userId}/{document=**} {
14       allow read: if request.auth != null;
15       allow write: if request.auth != null && request.auth.uid == userId;
16     }
17     match /totals/{directionsMode}/{userId}/{document=**} {
18       allow read: if request.auth != null;
19       allow write: if request.auth != null && request.auth.uid == userId;
20     }
21   }
22 }
23

```

Рисунок 3.4 Описання прав доступу до сховища Firestore в Firebase Console

Як видно з рис. 3.4 – у системі трекингу активностей права налаштовані наступним чином:

- Зчитувати дані можуть лише авторизовані користувачі. Це зроблено із заділом на майбутнє, оскільки майбутній функціонал несе собою інтеграцію з соціальною мережею і профілі користувачів будуть доступні для усіх авторизованих користувачів;
- Записувати або редагувати дані може лише власник цих даних.

У Firebase є зручний моніторинг прав доступу. Адміністратор в змозі продивитися статистику запитів, кількість спроб порушення прав доступу тощо.



Рисунок 3.5 Моніторинг прав доступу до сховища Firestore в Firebase Console

У сховищі Firebase було прийнято рішення зберігати чотири основні моделі даних: activities, routes, totals, users:

Users. Документ профілю користувача має наступний вигляд:

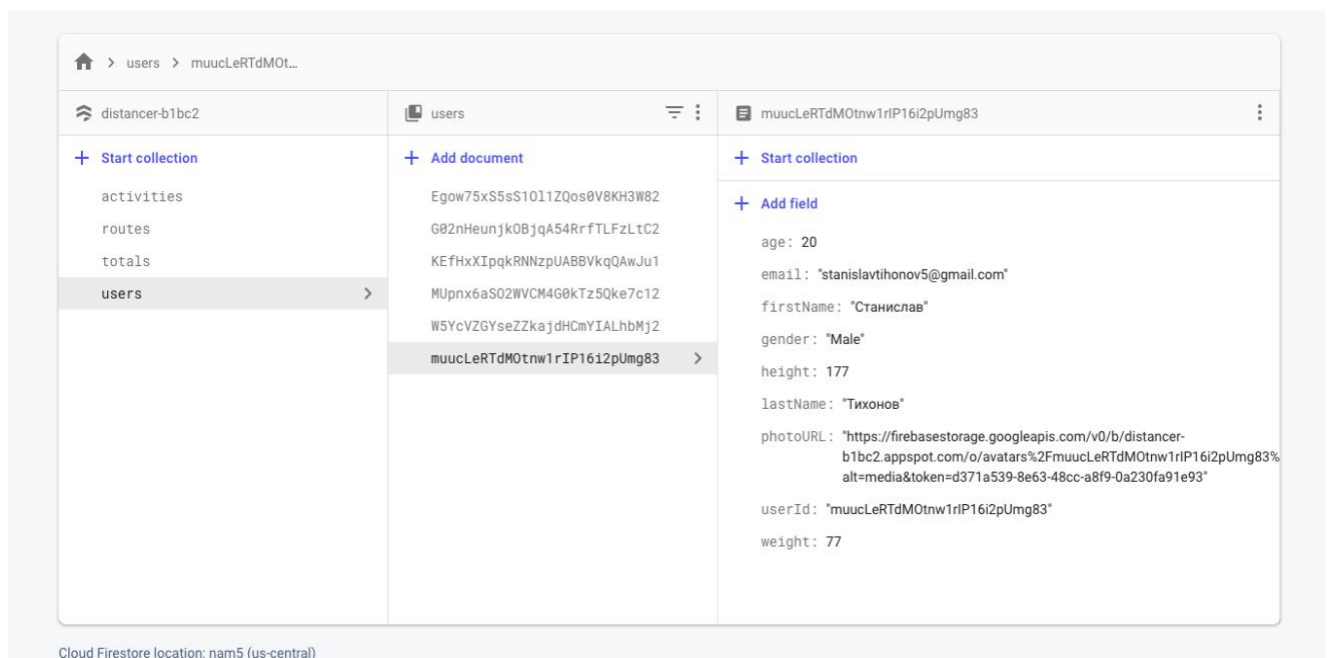


Рисунок 3.6 Документ профілю користувача у Firebase

У колекції users розташовані підколекції документів, назвою якого є ідентифікатор userId, що знаходиться в середині самого документа з інформацією о користувачі.

Таблиця 3.1 Зміст документу User

Поле	Примітка
userId	унікальний ідентифікатор користувача
firstName	ім'я користувача;
lastName	прізвище користувача
gender	стать користувача
age	вік користувача
weight	вага користувача
height	зріст користувача
email	пошта користувача
photoURL	посилання на фото користувача

Після реєстрації усі поля стягуються з постачальника Google або мануально заповнюються на екрані профілю. Присутня можливість редагувати отримані дані.

Routes. Документ маршруту користувача має наступний вигляд:

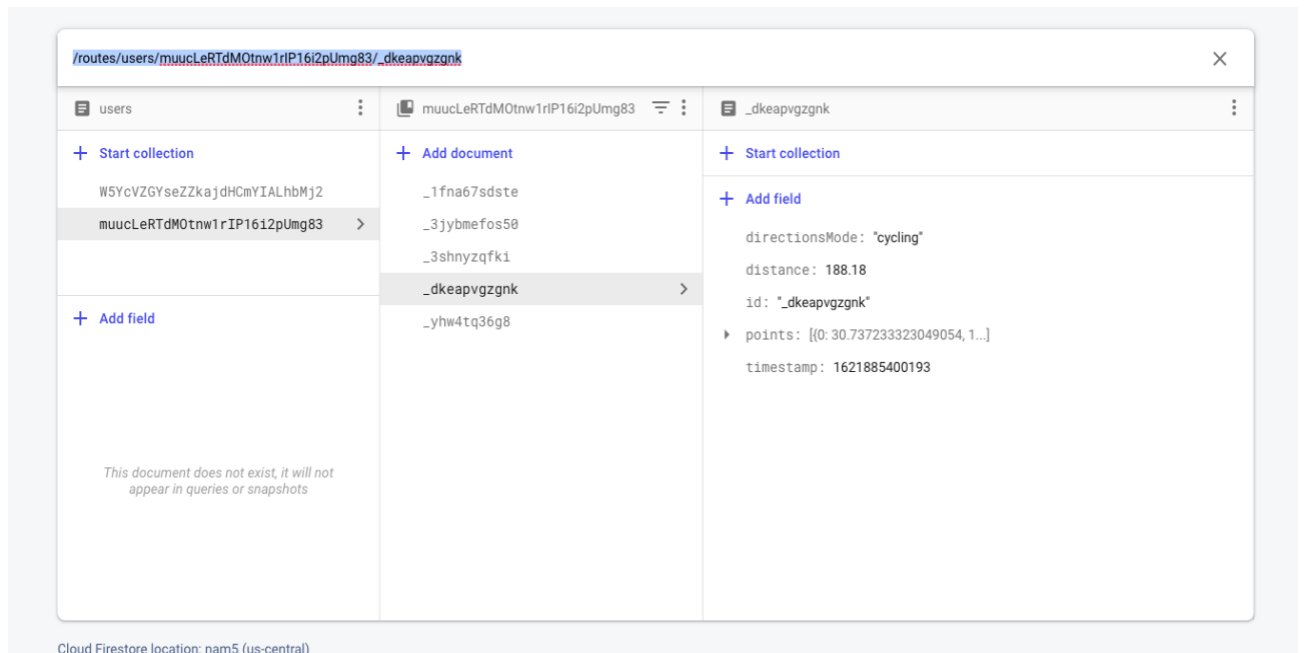


Рисунок 3.7 Документ одного з маршрутів користувача у Firebase

Таблиця 3.2 Зміст документу Route

Поле	Примітка
id	унікальний ідентифікатор документа
distance	дистанція маршруту у кілометрах
directionsMode	тип майбутнього тренування
points	набір географічних координат для відображення смуги маршруту на мапі
timestamp	часова мітка моменту створення маршруту

Activities. Документ тренування має наступний вигляд:

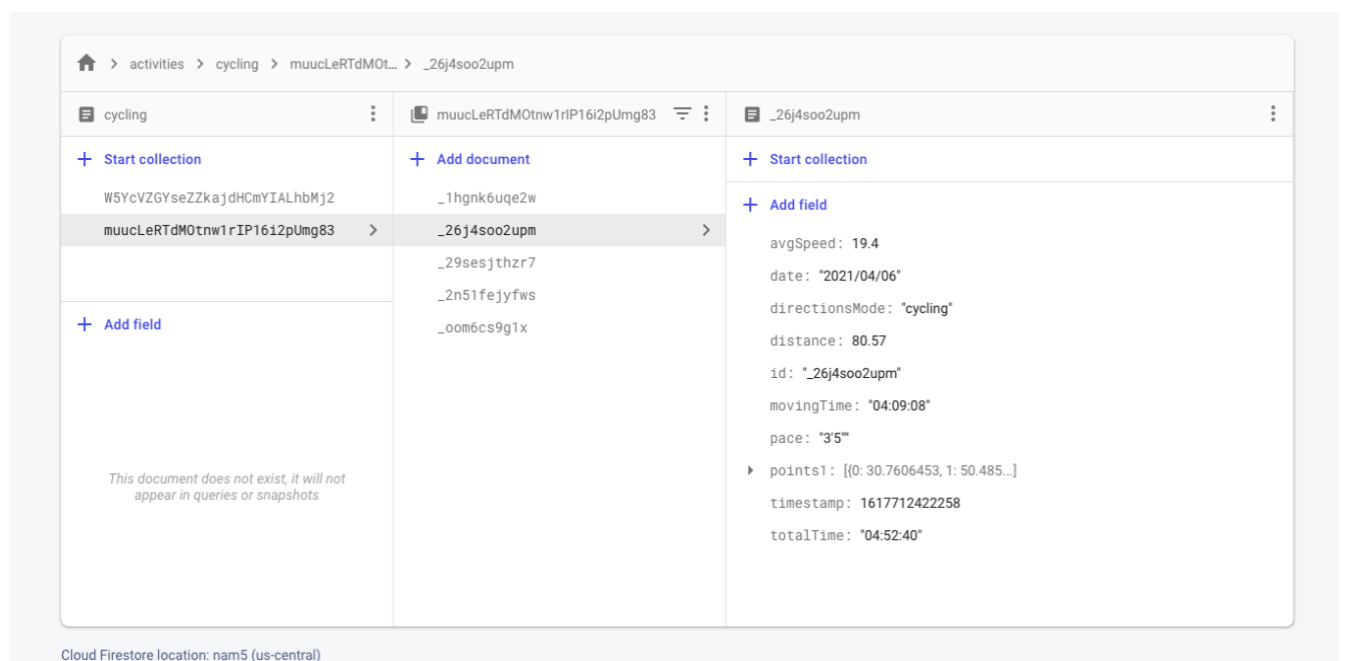


Рисунок 3.8 Документ одного з тренувань користувача у Firebase

Таблиця 3.3 Зміст документу Activity

Поле	Примітка
id	унікальний ідентифікатор документу
distance	подолана дистанція на тренуванні
avgSpeed	середня швидкість користувача на тренуванні
pace	середній темп користувача
movingTime	час тренування у русі
totalTime	загальний час тренування
directionsMode	тип тренування
date	дата тренування
points1	набір географічних координат тренування для відображення смуги маршруту на мапі
timestamp	часова мітка моменту створення тренування

Totals. Дана колекція необхідна для підрахунку середніх показників тренувань у місячній категорії. Поки спроектовані загальні середні показники за місяцями і у загальному обсязі, у випадку зміни функціональних вимог у разі появи іншого виду категорій, наприклад за тижнями, Firebase дозволяє вирахувати потрібну категорію з вже існуючих даних, і у подальшому продовжувати оновлювати вже нову категорію.

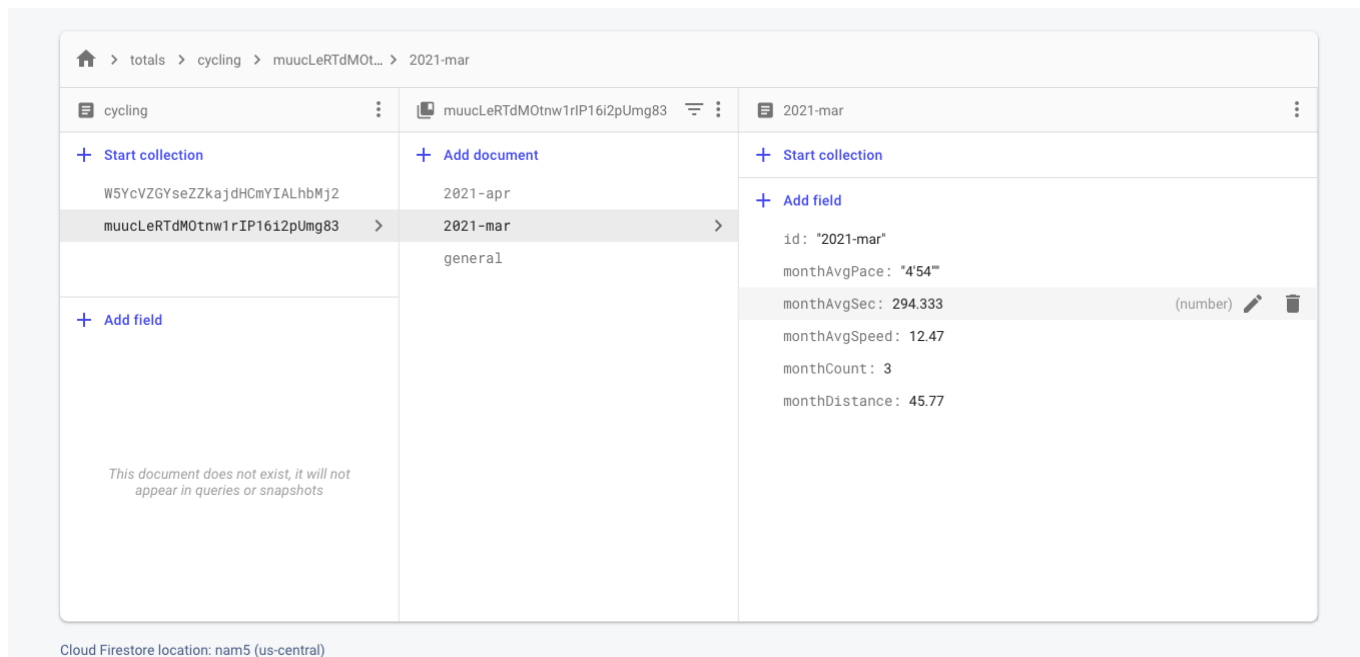


Рисунок 3.9 Документ середніх показників тренувань у місячній категорії користувача у Firebase

Таблиця 3.4 Зміст документу місячних показників

Поле	Примітка
id	унікальний ідентифікатор документу
monthAvgSpeed	середня швидкість тренувань за місяць
monthAvgSec	середня кількість секунд тренувань за місяць
monthAvgPace	середній темп тренувань за місяць
monthCount	кількість тренувань за місяць
monthDistance	подолана дистанція на тренуваннях за місяць

3.3 Деталі проекту

Після проектування програмного забезпечення, була організована наступна структура проекту:



Рисунок 3.10 Структура проекту

Переглянемо основні сутності проекту:

- index.js – точка входу у системі, реєструє кореневий компонент App.js, що знаходиться у src;
- package.json – файл опису проекту. Несе в собі такі важливі дані як назва, версія, список залежностей проекту, команди для запуску\тестування\збірки тощо;
- App.js – легкий компонент, що реєструє DI контейнери застосунку, Mobx сховища, Main.js;
- Main.js – компонент, що реєструє екрани навігації у застосунку;

- src/actions – директорія функцій, сервісів, що взаємодіють з http і firebase;
- src/assets – директорія статичних зображень;
- src/componets – директорія компонентів системи. Вибірка від компонентів мапи або списку тренувань до компонентів клавіш, полів вводу тексту тощо;
- src/config – директорія конфігурацій проекту, реєстрації локалізації, екранів застосунку тощо;
- src/constants – директорія загальних, незмінних даних, наприклад висоти модального вікна, та загальних стилів, що пере-використовуються у застосунку;
- src/hooks – директорія власного допоміжного функціоналу для пере-використання логіки у React компонентах, наприклад для використання анімацій, показу спінера в момент підгрузки даних з сховища, монтування компонентів тощо;
- pages – директорія екранів застосунку;
- stores – директорія сховищ для нормалізованих і ненормалізованих даних застосунку;
- utils – директорія допоміжних функцій застосунку.

Система трекингу користувачів написана на мові JavaScript, основна взаємодія у системі зав’язана на спостерігачах з інтеграції DI.

Dependency Injection (DI) – це процес передачі зовнішньої залежності програмному компоненту. Являється специфічною формою «інверсії управління» (англ. Inversion of control, IoC), коли вона застосовується до управлінню залежностями. У повній відповідності з принципом єдиного обов’язку об’єкт відає хлопоти побудови необхідних йому залежностей зовнішньому, спеціально призначеному для цього загальному механізму [8].

Розглянемо як влаштована локалізація у застосунку. З функціональних вимог слідує, що система підтримує 2 мови: українська і англійська. Нижче

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

наведено структуру зберігання усіх текстових записів, що змінюються в залежності від мови застосунку:

```

export const sign = {
  message: {
    emailAlreadyUsed: 'Дана пошта вже у використанні іншого користувача',
    noCurrentUser: 'Ви не авторизувались',
    userNotFound: 'Користувача не знайдено',
    needAuth: 'Спочатку необхідно авторизуватись',
    needReLogin: 'Спочатку необхідно перев авторизуватись',
  },
  input: {
    email: 'Пошта',
    password: 'Пароль',
  },
};

export const signIn = {
  message: {
    authorized: 'Авторизовано',
    welcomeBack: 'З поверненням',
    signInToContinue: 'авторизуйтесь для продовження',
    or: 'або',
  },
  button: {
    withGoogle: 'Продовжити з Google',
    signIn: 'Авторизуватись',
  },
};

export const signUp = {
  message: {
    welcome: 'Вітаємо',
    signUpToContinue: 'зареєструйтесь для продовження',
    or: 'або',
  },
  button: {
    withGoogle: 'Продовжити з Google',
    signUp: 'Зареєструватись',
  },
};

export const sign = {
  message: {
    emailAlreadyUsed: 'The email address is already in use by another account',
    noCurrentUser: 'You are not signed',
    userNotFound: 'User not found',
    needAuth: 'Log in first',
    needReLogin: 'Please re-log in first',
  },
  input: {
    email: 'Email',
    password: 'Password',
  },
};

export const signIn = {
  message: {
    authorized: 'Authorized',
    welcomeBack: 'Welcome back',
    signInToContinue: 'sign in to continue',
    or: 'or',
  },
  button: {
    withGoogle: 'Continue with Google',
    signIn: 'Sign in',
  },
};

export const signUp = {
  message: {
    welcome: 'Welcome',
    signUpToContinue: 'sign up to continue',
    or: 'or',
  },
  button: {
    withGoogle: 'Continue with Google',
    signUp: 'Sign up',
  },
};

```

Рисунок 3.11 Приклад набору текстових записів, що змінюються в залежності від поточної мови застосунку

Зберігаються записи в об'єктах. У вихідному файлі, що відповідає за реєстрацію локалізації, налаштований DI контейнер, куди, в момент ініціалізації застосунку, інтегруються записи за ідентифікаторами – EN для англійської, UK для української мов:

```

1 import {LOCALES} from '~/constants/constants';
2 import * as en from '~/constants/papyrus/en';
3 import * as uk from '~/constants/papyrus/uk';
4 import {localesDI} from '~/utils/locales-di';
5
6 localesDI.Injectable(LOCALES.EN)(en);
7 localesDI.Injectable(LOCALES.UK)(uk);
8

```

Рисунок 3.12 Приклад реєстрації локалізацій у DI контейнер

У системі дані зберігаються у Mobx сховищах. Як ми визначилися на етапі вибору технологій для системи, Mobx підтримує реактивність, тобто при

виклику функції для зміни даних, дані одразу змінюються у всіх місцях використання. У сховищі є два поля спостерігачі:

- *locale* – ідентифікатор мови, за котрим зберігаються текстові записи у DI контейнері;
- *papyrus* – структура в котрій зберігаються саме текстові записи в залежності від *locale*.

Зміна *locale* з налаштувань користувача трігерить виклик слухача сховища, що стягує з DI контейнеру нові текстові записи за зміненням *locale*. Оскільки, *papyrus* також є спостерігачем, він змінюється в усіх місцях його використання. На рис. 3.11 можна побачити, що структура текстових записів має певну вкладеність. У сховищі присутня екшн-функція *papyrusify*, що приймає шлях до необхідного рядку (котрий буде відображений на екрані) і за шляхом повертає цей рядок. Нижче наведено сховище, що відповідає за локалізацію:

```
import {makeAutoObservable, observe} from 'mobx';
import {DEFAULT_LOCALE} from '~/constants/constants';
import {localesDI} from '~/utils/locales-di';
import {storesDI} from '~/utils/store-di';
import {isExist, isObject} from '~/utils/validation/helpers';

const separator = '.';

const execPapyrus = (accum, key) =>
  (val => (isObject(val) ? val : {_result: isExist(val) ? val : ''})).call(null, accum[key]);

export class LocaleStore {
  constructor() {
    makeAutoObservable(this);
    observe(this, this._listenLocale);
  }

  locale = DEFAULT_LOCALE;
  papyrus = localesDI.Inject(DEFAULT_LOCALE);

  setLocale = locale => {
    this.locale = locale;
  };

  papyrusify = path => {
    const papyrusified = path.split(separator).reduce(execPapyrus, this.papyrus);
    return papyrusified._result ?? papyrusified;
  };

  _listenLocale = ({name, newValue}) => {
    if (name !== 'locale') return;
    this.papyrus = localesDI.Inject(newValue);
  };
}

export const useLocale = () => storesDI.Inject('localeStore');
export const getLocaleStore = () => storesDI.Inject('localeStore');
```

Рисунок 3.13 Mobx сховище локалізації у системі

У місці, де необхідно відобразити локалізований текст, викликається функція *papyrusify* зі шляхом до необхідного рядку (шлях повинен бути

					ІАЛЦ.467100.003 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

однаковим для усіх мов), як ми зазначили вище, у разі зміни мови додатка, *papyrusify* оновить значення рядку.

```
const Greeting = (  
  <>  
    <Text style={greetingStyle}>{papyrusify('signIn.message.welcomeBack')}</Text>  
    <Text style={subGreetingStyle}>{papyrusify('signIn.message.signInToContinue')}</Text>  
  </>  
)
```

Рисунок 3.14 Використання локалізованих текстів у компоненті привітання на екрані авторизації

					ІАЛЦ.467100.003 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 3

У даному розділі було розглянуто вимоги до системи, а саме функціональні, не функціональні й вимоги до інтерфейсу. Було спроектовано моделі Firestore і реалізовано систему трекингу активностей. Велику увагу приділено аутентифікації і авторизації, оскільки вони є ключовими точками безпеки даних додатку. Розглянуто і використано редагування прав доступу до колекцій Firestore.

Таким чином, можна переходити до наступного розділу.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4 ІНСТРУКЦІЯ КОРИСТУВАЧА

У даному розділі надано детальний опис взаємодії користувача з системою трекингу активностей. Також надані пояснення при взаємодії з кожним екраном застосунка, їх функціоналом.

Після встановлення і першого запуску додатку відкривається головний екран, на якому присутня нижня навігаційна панель для переходу між екранами додатку. Доки користувач не авторизувався до системи, навігаційна панель має два стани: View і Menu. Також присутня клавіша для швидкого пересування мапи до поточного місцезнаходження.

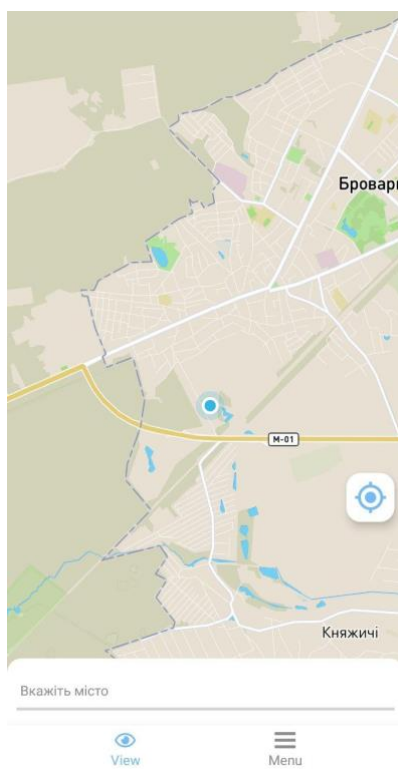


Рисунок 4.1 Головний екран застосунку

Як видно з рис. 4.1, на екрані View присутнє поле пошуку міста. У застосунку є декілька режимів зміни відображення мапи:

- 1) За допомогою пересування мапи дотиком, тобто класичний режим роботи з мапами;
- 2) За допомогою пошуку міста для більш зручної навігації на випадок, якщо шукана зона далеко знаходиться від поточної.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

При натиску на поле пошуку, можна ввести шукане місто, або обрати поточне місцезнаходження користувача, після чого мапа змінить відображення на обрану позицію.

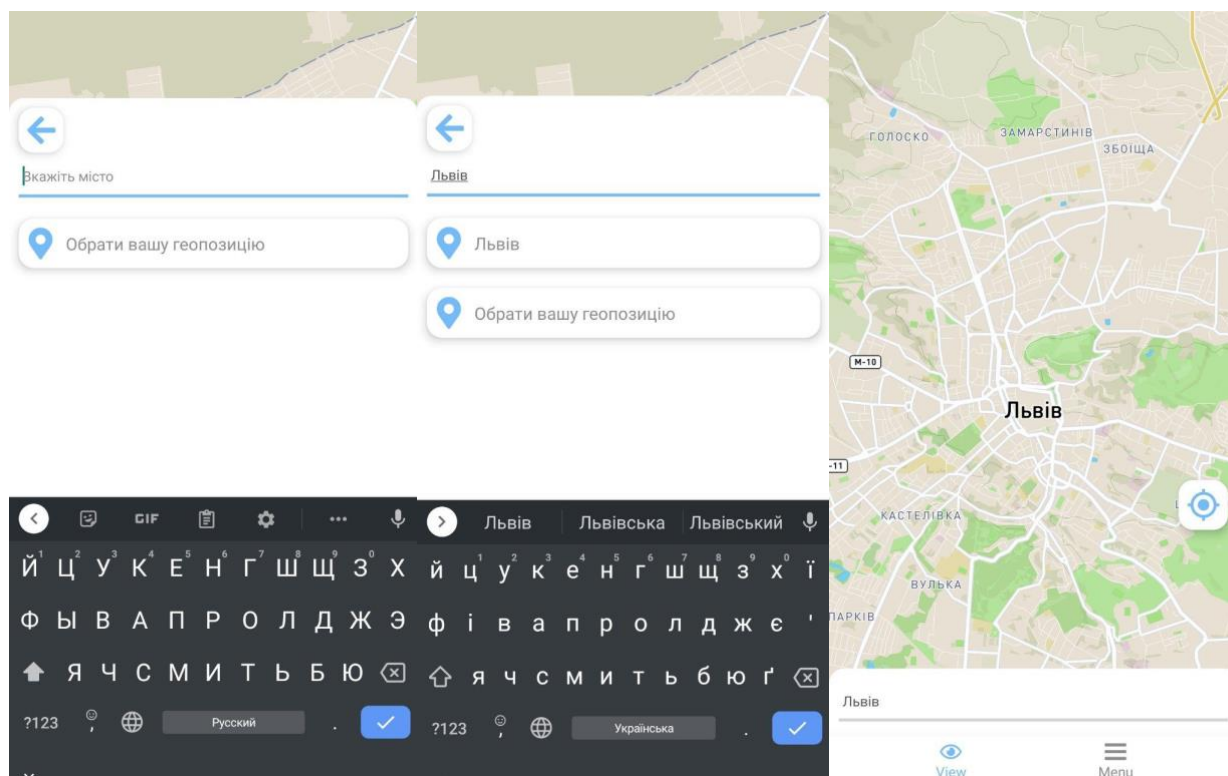


Рисунок 4.2 Пошук міста у застосунку

При переході на Menu екран, що відповідає за налаштування, користувач, доки не авторизований, може:

- змінити тему додатку зі світлої на темну;
- змінити мову додатку з української на англійську;
- авторизуватися до системи: зареєструватися або увійти до існуючого профілю.

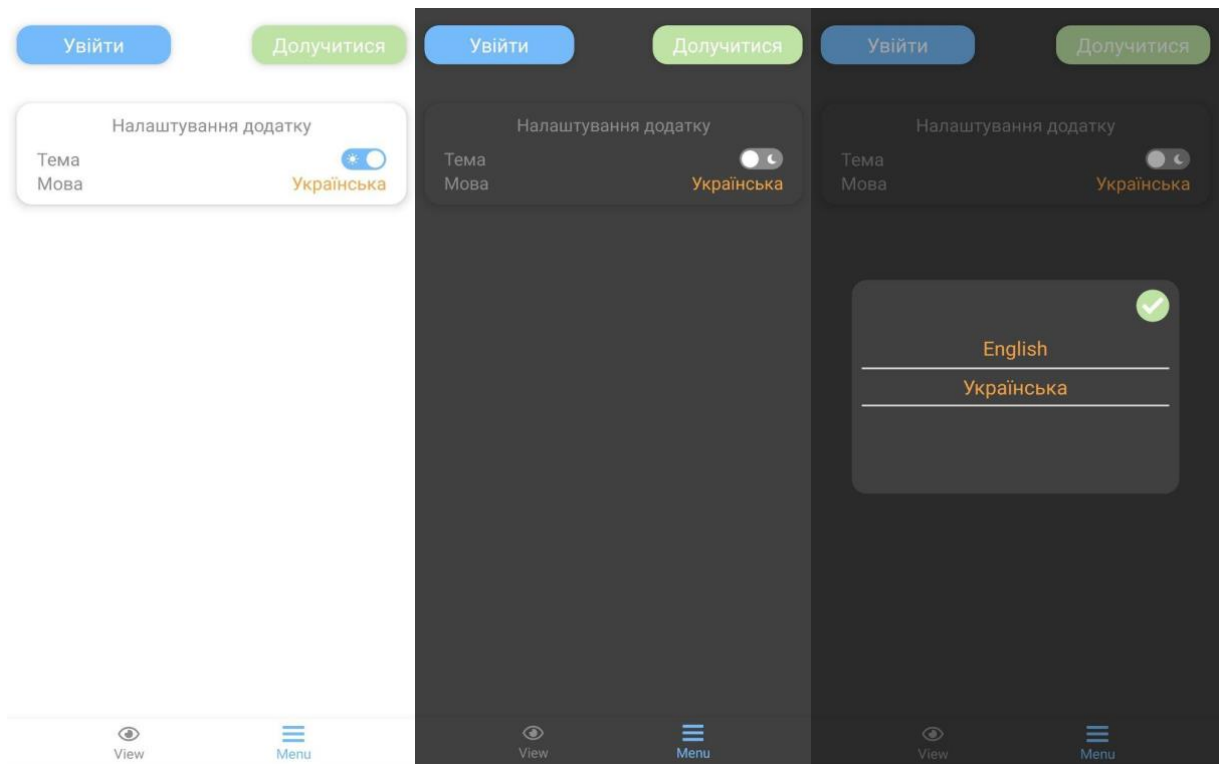


Рисунок 4.3 Екран Меню не авторизованого користувача

Авторизація у системі проводиться двома напрямками: реєстрація нового акаунту, вхід до існуючого. Користувач може авторизуватися як за допомогою пошти з паролем, так і за допомогою Google. У випадку Google авторизації, користувачеві потрібно обрати профіль зі списку або додати новий. При класичній авторизації, користувач має коректно заповнити дані і натиснути відповідну блакитну клавішу для продовження.

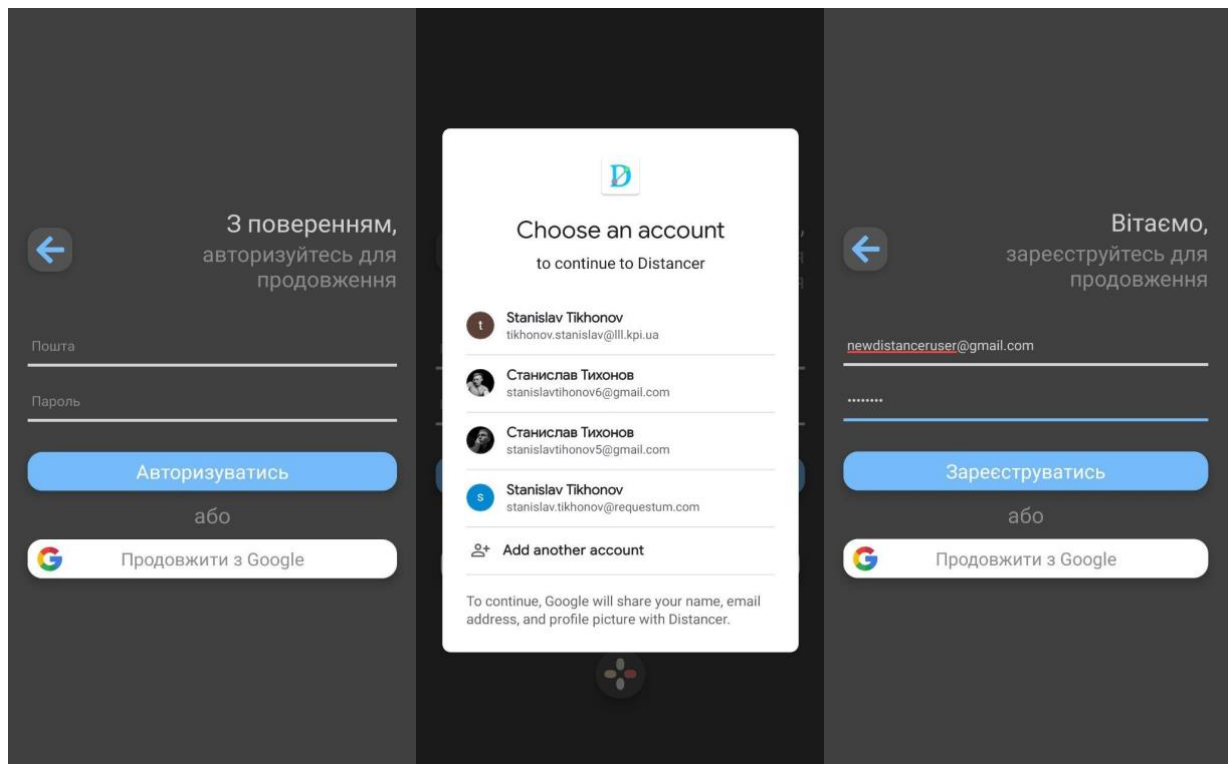


Рисунок 4.4 Екрани авторизації у системі(Вхід, Реєстрація)

Після реєстрації користувач переходить на екран заповнення профілю. Якщо вийти з додатку з незаповненим профілем, при наступному відкритті система нагадає, що потрібно заповнити профіль.

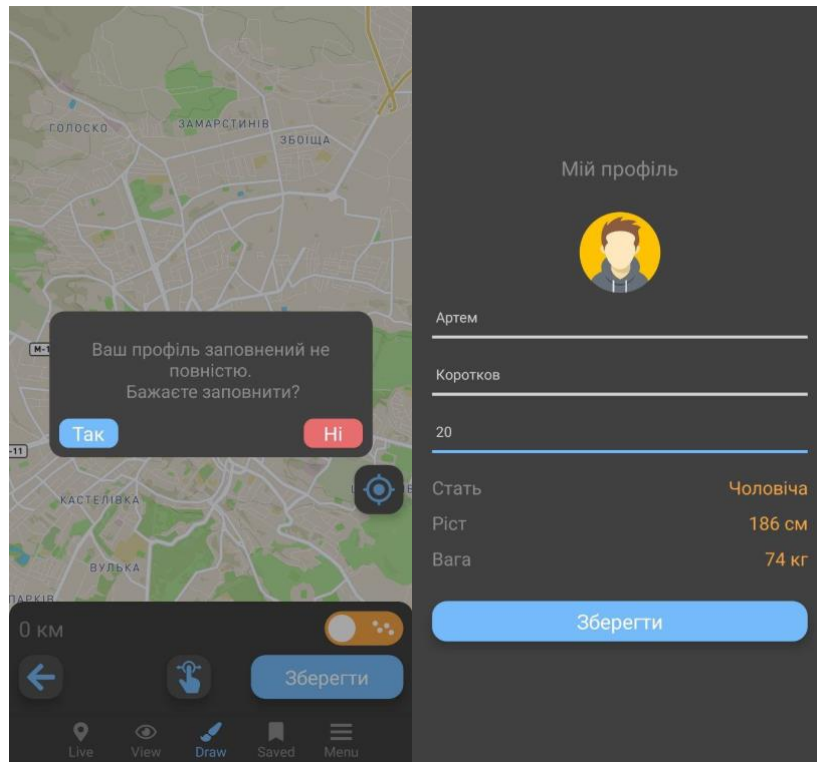


Рисунок 4.5 Екран заповнення профілю у системі

Після авторизації, функціонал додатку розширюється. У навігаційній панелі з'являються нові екрани.

На Live екрані користувач може розпочати тренування. В системі є 3 режими тренування: біг, велозаїзд, автозаїзд. Для початку тренування користувач натискає клавішу «Почати», після чого тренування розпочнеться одразу або після таймеру, в залежності від налаштувань користувача. При пересуванні користувача, зчитуються координати на основі яких малюється смуга подоланого маршруту в режимі реального часу.

В системі передбачена автоматична пауза тренування при бездії користувача, якщо той не пересувається. При активному тренуванні, користувачеві відображається 2 стани тренування: поточні і загальні показники.

При натиску на ліву частину поточних показників змінюється наступна інформація: поточна швидкість у км\год, поточний темп тренування у кількості затрачених хвилин на один кілометр.

При натиску на праву частину, відповідно, змінюється інформація загальних показників: час у русі, подолана дистанція, середня швидкість у км\год, середній темп.

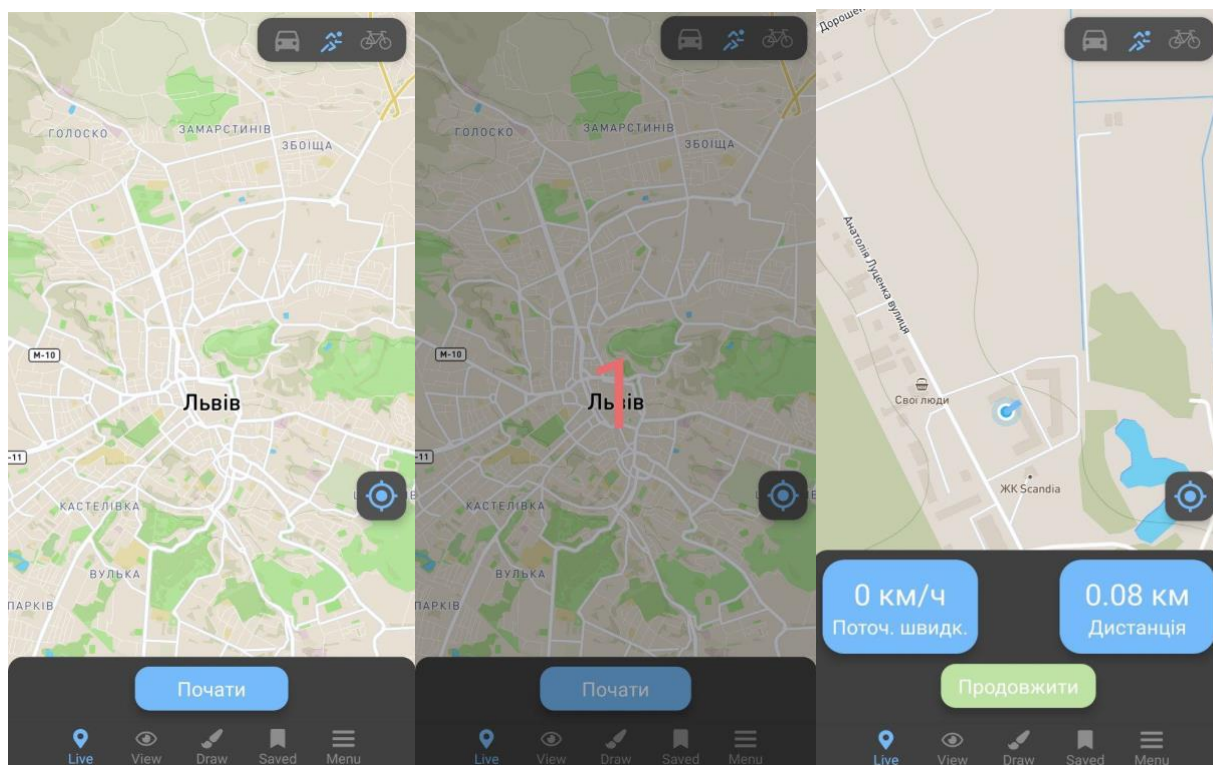


Рисунок 4.6 Live екран у застосунку

Для призупинення тренування достатньо активувати автоматичну паузу в налаштуваннях, або натиснути на помаранчеву клавішу «Зупинити», повторний натиск, вже зеленої, клавіші «Продовжити», відповідно, продовжить тренування.

При довгому натиску клавіш («Зупинити», «Продовжити») у користувача з'явиться вікно з запитом завершення тренування. У разі аварійного закриття додатку весь прогрес зберігається і при новому відкритті, з'явиться вікно з аварійним тренуванням і діями, що можна зробити з вищезгаданим тренуванням: завершити, продовжити, видалити.

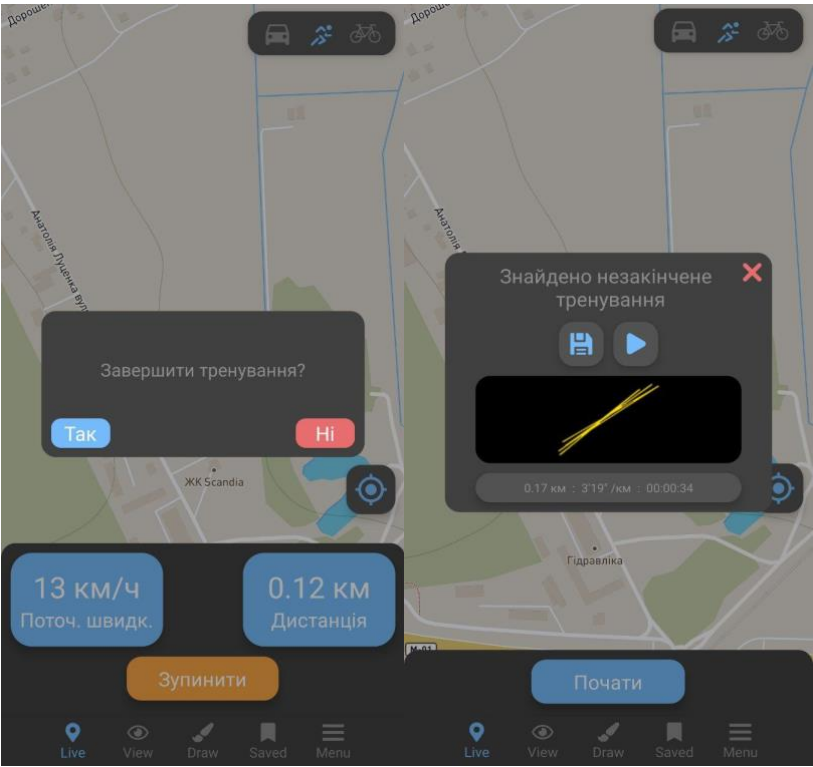


Рисунок 4.7 Функціонал завершення тренування і дій після аварійного закриття.

На Draw екрані користувач може прокладати маршрути для майбутніх тренувань.

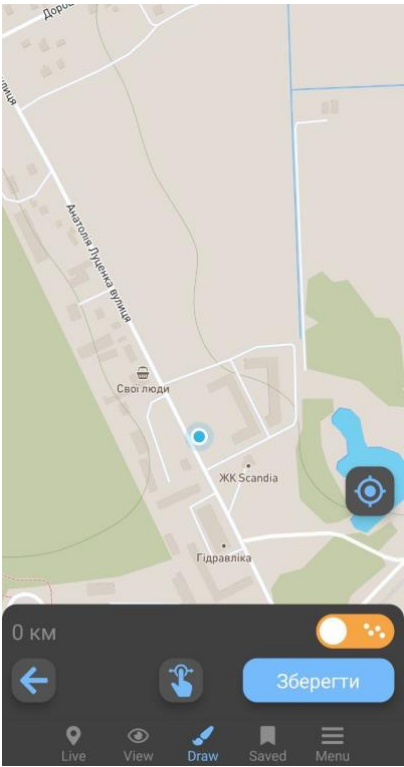


Рисунок 4.8 Draw екран застосунку

У системі передбачено 2 режими прокладення маршруту:

- прямий, тобто прокладається напряму через місцевість натисками на ділянки мапи;
- напрямки, тобто прокладається з урахуванням оптимального маршруту між ділянками мапи в залежності від режиму: пішохідний, велосипедний, автомобільний.

Для переключення режимів прокладення необхідно натиснути на перемикач, що знаходиться зверху клавіші «Зберегти».

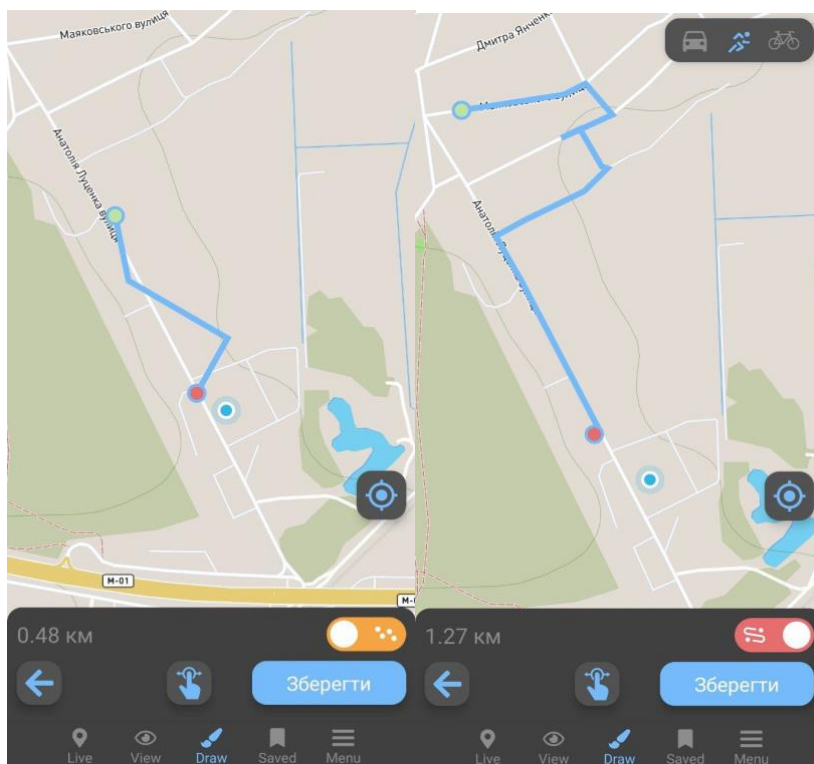


Рисунок 4.9 Режими прокладення маршруту

У системі передбачено функцію повернення на один крок до минулого стану маршруту, для цього, потрібно натиснути клавішу зі стрілкою. При довгому натиску цієї клавіші весь маршрут стирається.

Також передбачено можливість редагувати маршрут на випадок не точного прокладення. Для редагування маршруту користувачеві потрібно натиснути клавішу, що знаходиться у центрі загального вікна. При натиску, клавіша змінить колір на червоний і перейде до активної фази, після чого,

потрібно натиснути на одну з крапок маршруту для вибору активної координати. Після натиску, обрану координату можна перемістити на майбутню позицію на мапі, в результаті чого дистанція маршруту оновиться зі змінами.

В налаштуваннях можна активувати підказки для редагування маршруту.

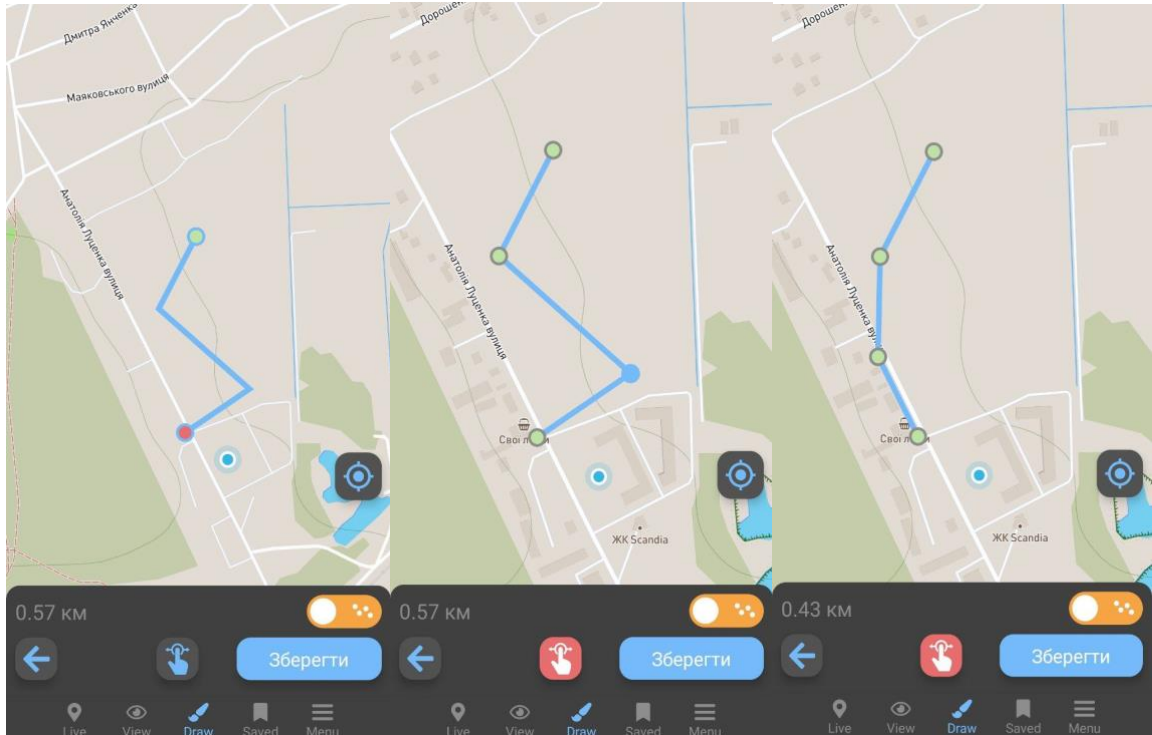


Рисунок 4.10 Режим редагування маршруту

Для збереження побудованого маршруту необхідно натиснути, відповідно, клавішу «Зберегти».

При аварійному закритті додатку, весь прогрес, як і у випадку тренування, зберігається. При наступному відкритті застосунку, з'явиться вікно з аварійним маршрутом і діями, що можна зробити з вищезгаданим маршрутом: зберегти, продовжити прокладення маршруту або видалити.

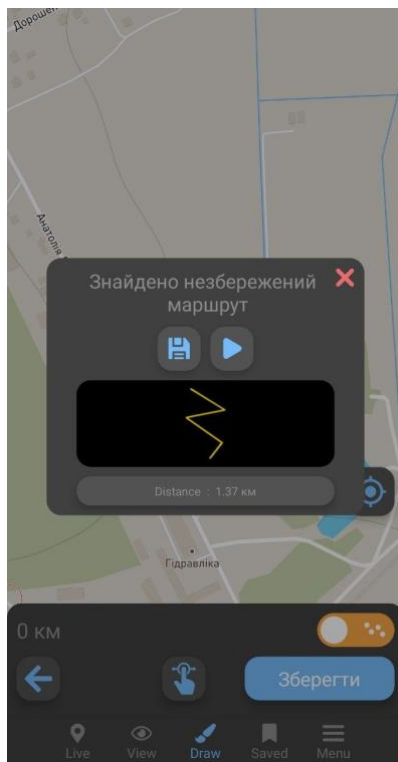


Рисунок 4.11 Застосунок після аварійного закриття під час прокладення маршруту

На Saved екрані користувач переглядає свої збережені маршрути та тренування.

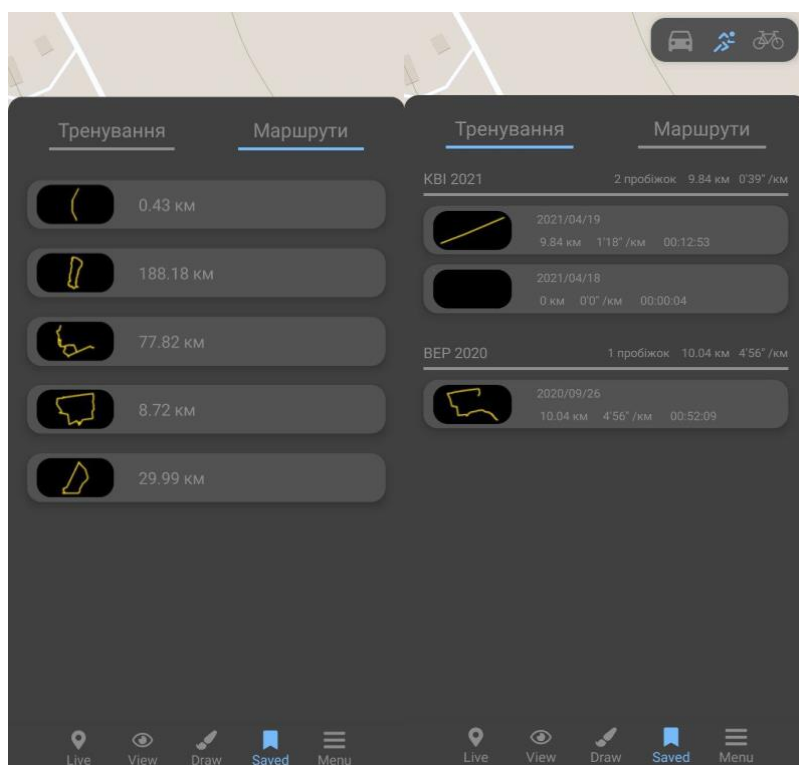


Рисунок 4.12 Saved екран застосунку

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

За замовченням при запуску застосунку відкривається список маршрутів. Для переключення між режимами відображення потрібно натиснути відповідні клавіші «Маршрути», «Тренування». Останній режим зберігається при переключенні екранів, тобто, якщо при активному режимі «Тренування» перейти на інший екран і повернутися до Saved – відобразиться список тренувань.

При активному режимі «Тренування» зверху екрану з'являється панель для зміни списку тренувань в залежності від обраного типу: авто, біг, велосипед.

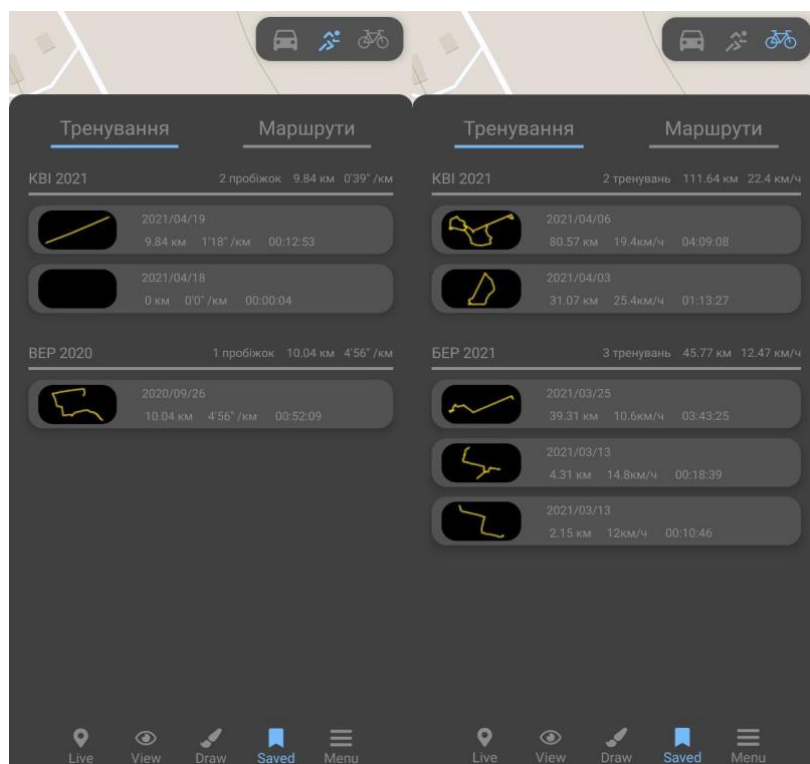


Рисунок 4.13 Перемикавання тренувань в залежності від типу тренування

Як видно на рисунках 4.12 - 4.13, тренування групуються за місяцями, на кожний місяць – відповідна група із записом загальних показників на цей місяць: кількість тренувань, подолана дистанція, середня швидкість або темп, в залежності від типу тренування.

При довгому натиску на тренування або маршрут – активується режим множинного виділення і зверху екрану з'явиться, відповідна, панель з діями. Виділений елемент відзначиться блакитною рамкою. Натиск на наступний, також виділить його. На панелі відображається кількість виділених тренувань,

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

маршрутів. З дій, можна скасувати режим виділення або видалити обрані елементи. При видаленні з'явиться вікно з запитом підтвердження операції.

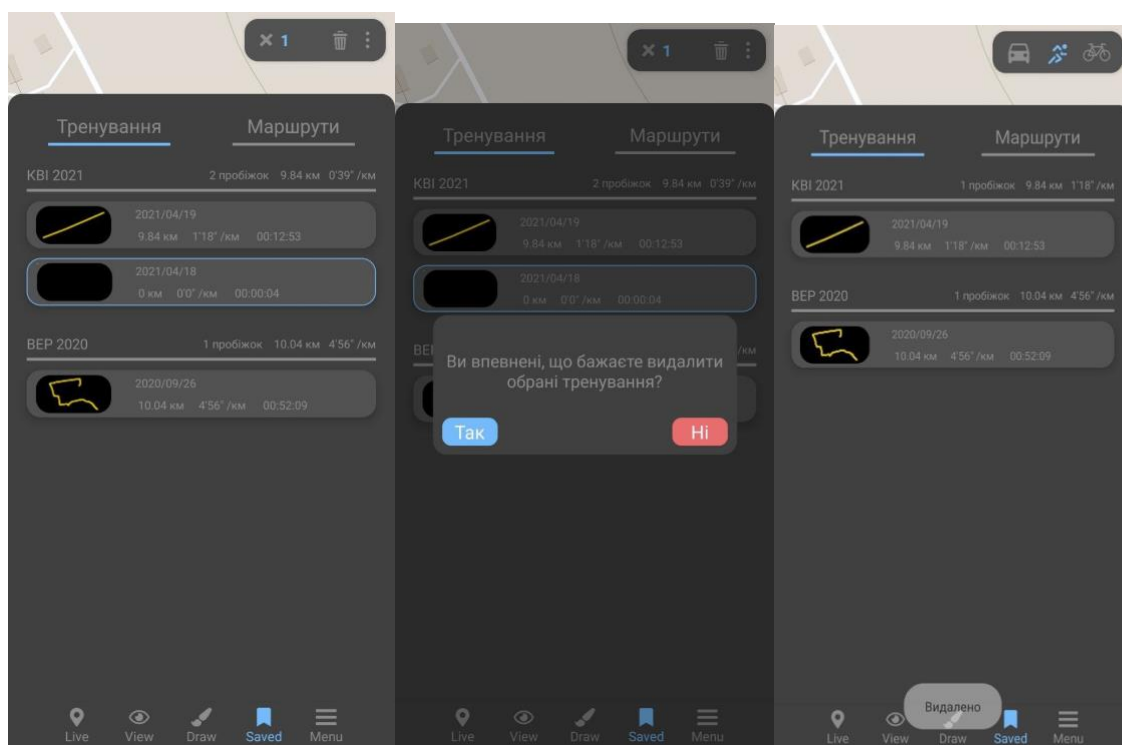


Рисунок 4.14 Режим множинного виділення

При натиску на певний маршрут на Saved екрані відбудеться перехід на екран обраного маршруту.

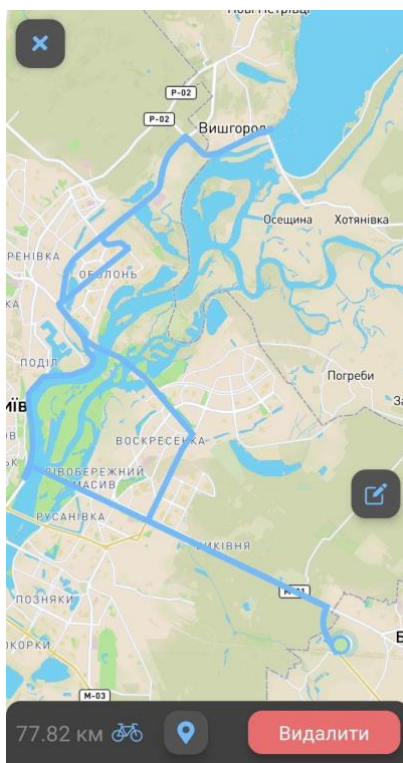


Рисунок 4.15 Екран обраного маршруту у застосунку

На екрані відображається дистанція маршруту и тип майбутнього тренування. При натиску на клавiшу, що знаходиться у центрі загального вікна, відбудеться перехід на Live екран початку тренування зі збереженням смуги маршруту, відмінної від основної. Таким чином, під час тренування на екрані присутній орієнтир запланованого руху спортсмена.

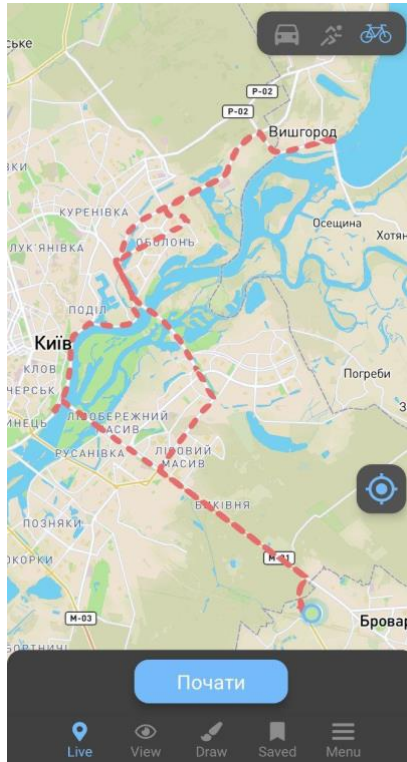


Рисунок 4.16 Екран початку тренування зі збереженою смугою маршруту

Повертаючись до екрану обраного маршруту, зліва вверху присутня клавiша повернення назад до списку збережених маршрутів.

Також на екрані обраного маршруту присутня клавiша для переходу на екран редагування маршруту.

Для видалення маршруту потрібно натиснути, відповідну, червону клавiшу «Видалити», при натиску з'явиться вікно з запитом підтвердження видалення.

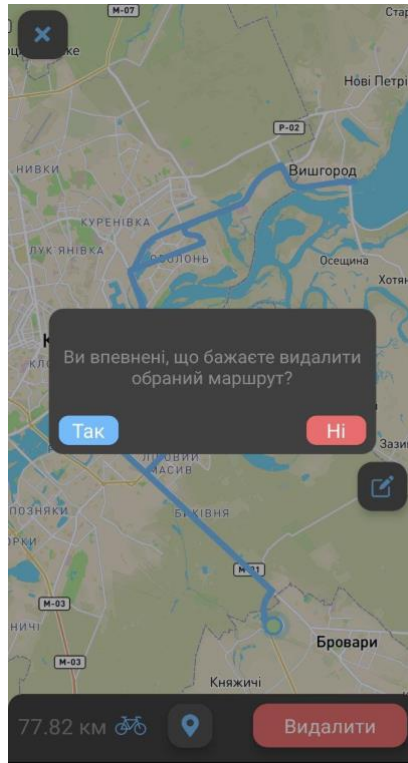


Рисунок 4.17 Видалення маршруту у застосунку

На екрані обраного тренування, також присутні клавiши видалення тренування і повернення до списку тренувань.

Щодо показників тренування, на екрані відображаються: тип тренування, дистанція, темп, середня швидкість, час у русі і загальний час з урахуванням часу зупинок.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						78
Зм.	Арк.	№ докум.	Підпис	Дата		

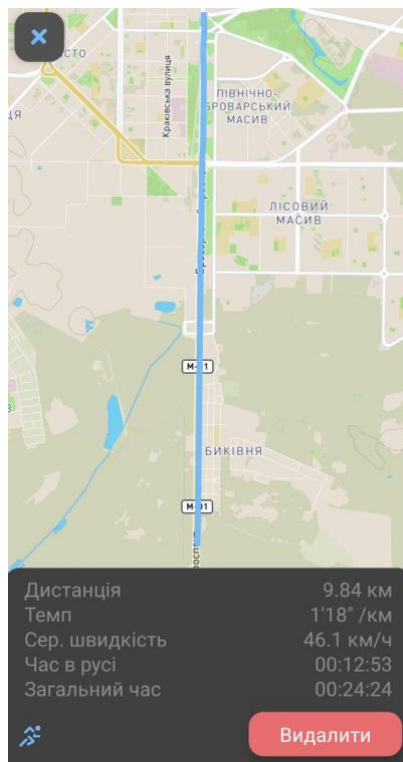


Рисунок 4.18 Екран обраного тренування у застосунку

На Menu екрані, після авторизації, з'явилися нові категорії налаштувань: налаштування акаунту, налаштування тренувань, налаштування маршрутів.

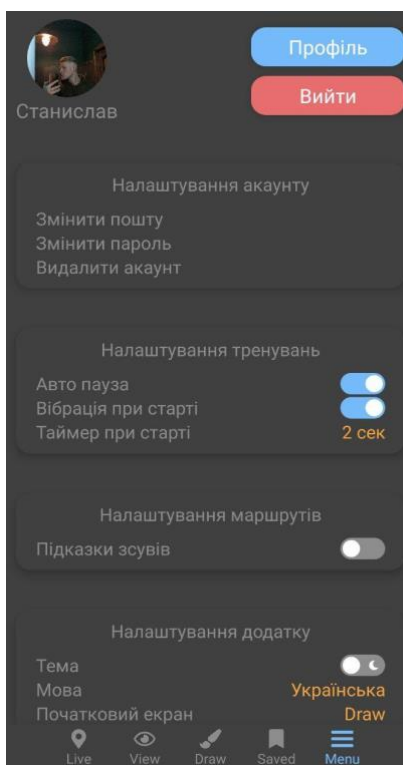


Рисунок 4.19 Menu екран авторизованого користувача у застосунку

					ІАЛЦ.467100.003 ПЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підпис	Дата		

Після авторизації, користувач додатково може:

- 1) Редагувати свій профіль, таким чином як і відбувалось його заповнення.
Для переходу на екран редагування потрібно натиснути блакитну клавішу «Профіль»;
- 2) Вийти з поточного акаунту. При натиску на червону клавішу «Вийти» з'явиться запит на вихід, у разі підтвердження, відбудеться перехід до Меню екрану не авторизованого користувача;
- 3) Змінити пошту або пароль. При натиску на пункт, з'явиться вікно з підтвердженням поточного паролю, у разі успішної перевірки, з'явиться нове вікно для зміни пошти або паролю в залежності від обраного пункту. Слід помітити, даний функціонал можливий лише у користувачів, що авторизувалися поштою з паролем. У випадку Google акаунтів, при натиску на обраний пункт з'явиться, відповідне, повідомлення внизу екрану;

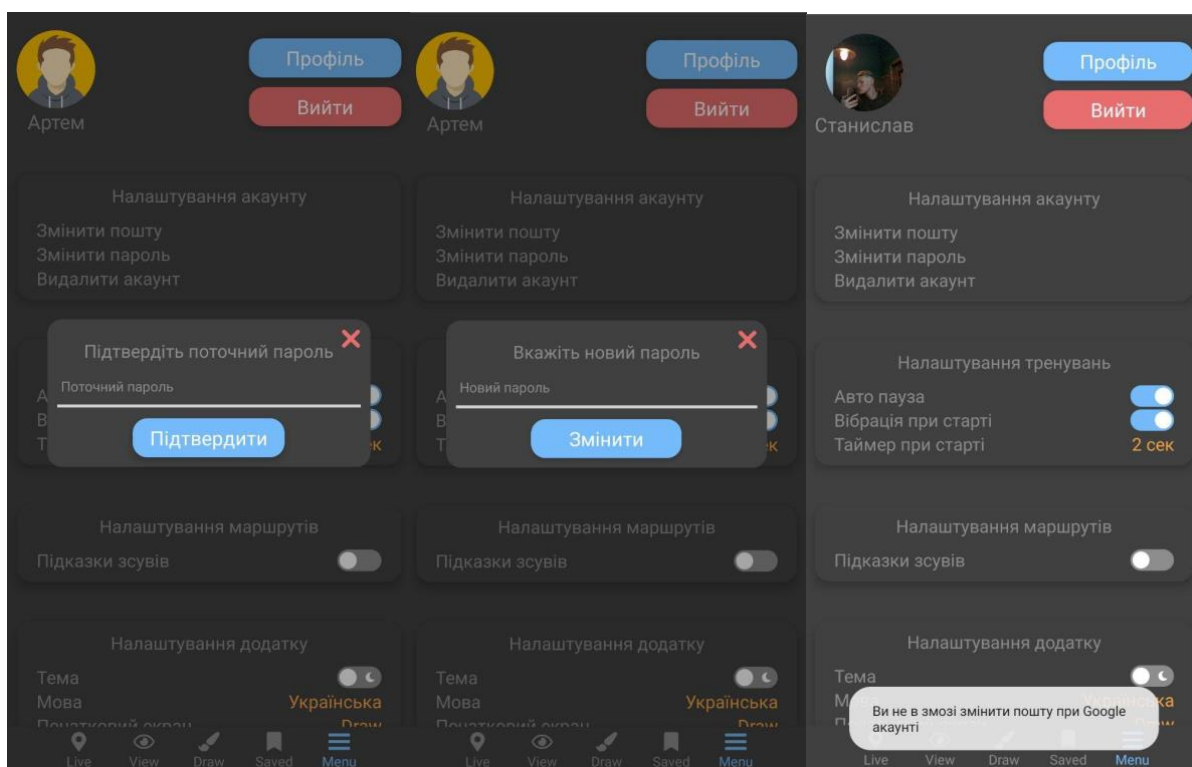


Рисунок 4.20 Зміна паролю користувача

- 4) Видалити акаунт. При натиску червоної клавіши «Видалити», з'явиться запит на підтвердження операції;

- 5) Ввімкнути або вимкнути автоматичну зупинку тренування при бездії спортсмена;
- 6) Ввімкнути або вимкнути вібрацію смартфона на старті тренування;
- 7) Додати таймер зворотного відліку перед початком тренування.

Користувач в змозі обрати значення таймеру у діапазоні від 0 до 10 секунд. У випадку 0, тренування одразу розпочнеться при старті;

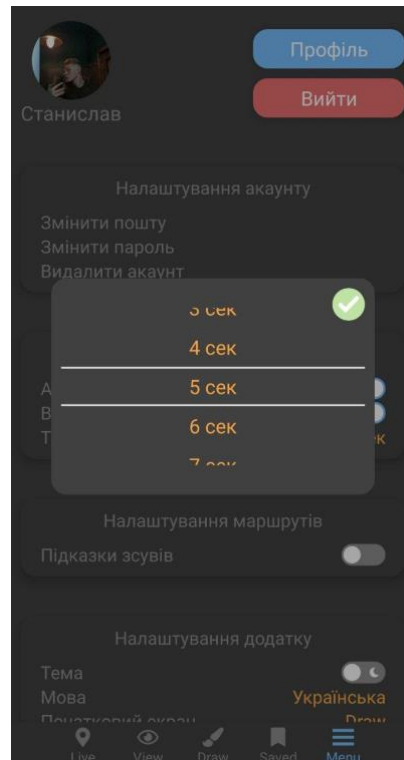


Рисунок 4.21 Вибір значення таймеру на початку тренування у застосунку

- 8) Ввімкнути або вимкнути показ підказок при редагуванні маршрутів, на кожну дію користувача з'явиться, відповідне, повідомлення, з інформацією щодо наступного кроку, внизу екрана;

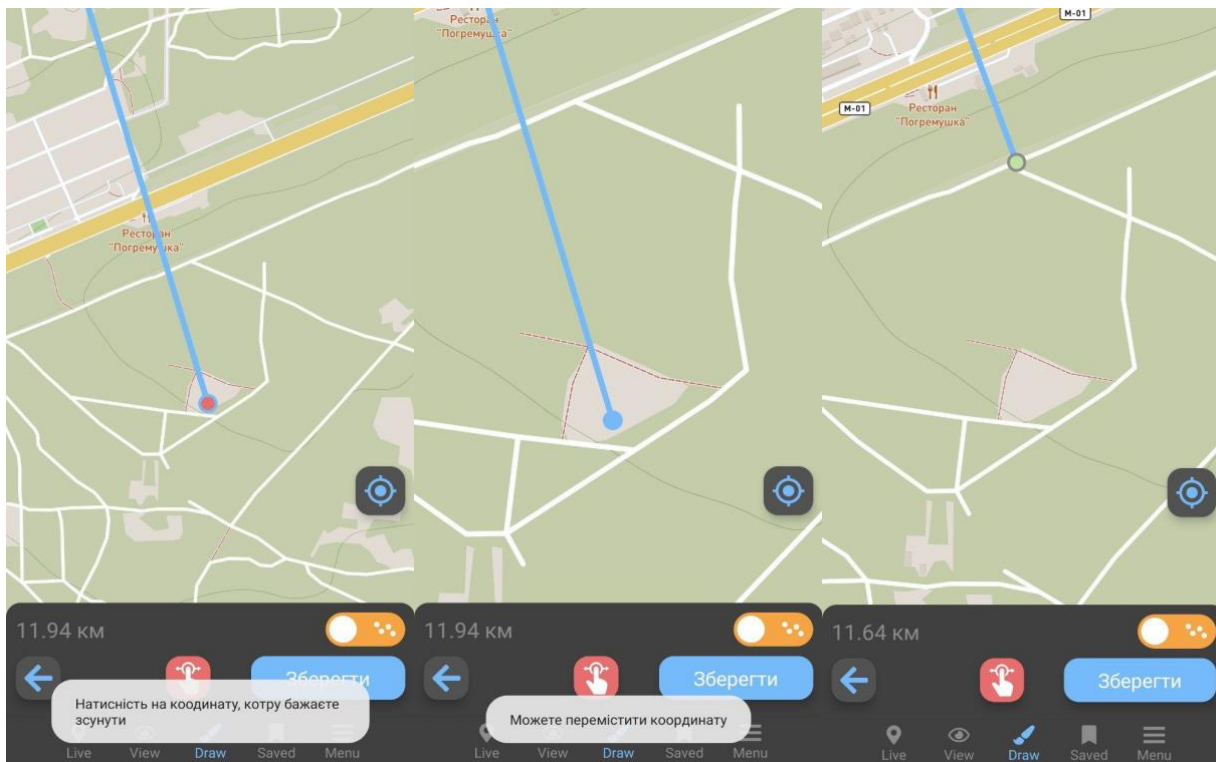


Рисунок 4.22 Демонстрація появи підказок при редагуванні маршруту

- 9) Обрати екран з навігаційної панелі, що буде відображатись першим при запуску додатку.

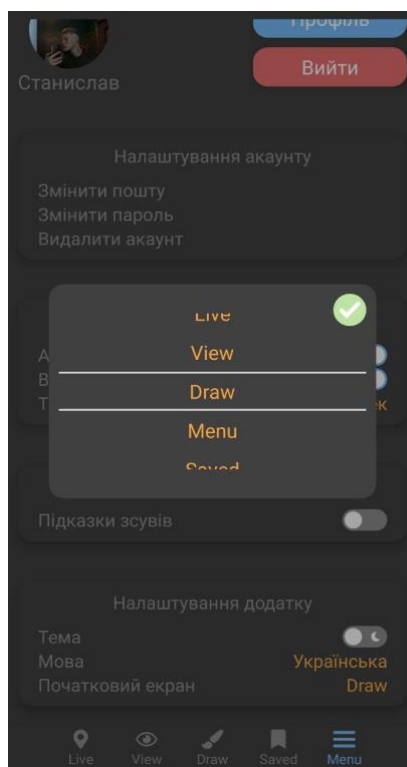


Рисунок 4.23 Вибір початкового екрану при запуску додатку

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

82

ВИСНОВКИ ДО РОЗДІЛУ 4

У даному розділі було детально описано інтерфейси застосунку. Наведені скриншоти кожного екрану й надані пояснення при взаємодії користувача з кожним екраном. Інструкція користувача отримала детальний і ємкий опис, таким чином, у потенційного користувача не виникне проблем при використанні додатку.

Створений інтерфейс вийшов інтуїтивний, зрозумілий та унікальний за дизайном. Ним приємно користуватися і достатньо зручно взаємодіяти під час тренування.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

ВИСНОВКИ

Під час реалізації даної дипломної роботи було створено систему трекингу активностей. Дана система допомагає користувачам будувати маршрути для тренувань і слідкувати за основними спортивними показниками під час тренування.

У першому розділі були розглянуті декілька існуючих рішень для моніторингу тренувань користувача. Під час дослідження, стало зрозумілим, що жодне рішення не покриває повну функціональність, що описана в завданні даної роботи. В результаті, розробку власного додатку було вирішено вважати доцільним, адже функціональність додатку перебиває альтернативні рішення.

У другому розділі були розглянуті певні технології, що будуть використовуватися для написання системи трекингу активностей, було проведено дослідження як влаштований React Native на рівні взаємодії з ОС, було порівнянно React Native з іншими схожими кросплатформеними технологіями. В результаті роботи було обрано менеджер стану для нормалізації даних у застосунку, платформу для відображення мапи у додатку і платформу для сховища даних.

У третьому розділі було розглянуто вимоги до системи, а саме функціональні, не функціональні й вимоги до інтерфейсу. Опісля, спроектовано моделі Firestore і реалізовано сам застосунок.

У четвертому розділі було детально описано інтерфейси застосунку. Наведені скриншоти кожного екрану й надані пояснення при взаємодії користувача з кожним екраном. Інструкція користувача отримала детальний і ємкий опис.

Створений інтерфейс вийшов інтуїтивний, зрозумілий та унікальний за дизайном. Ним приємно користуватися і достатньо зручно взаємодіяти під час тренування.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google Play Store Nike Run Club App [Електронний ресурс] – Режим доступу до ресурсу:
<https://play.google.com/store/apps/details?id=com.nike.plusgps&hl=ru&gl=US>
2. Mobx site [Електронний ресурс] – Режим доступу до ресурсу:
<https://mobx.js.org/README.html>
3. Wikipedia Reactive Programing [Електронний ресурс] – Режим доступу до ресурсу:
https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
4. Authentification [Електронний ресурс] – Режим доступу до ресурсу:
<https://sendpulse.ua/support/glossary/authentication>
5. OAuth 2 [Електронний ресурс] – Режим доступу до ресурсу:
<https://habr.com/ru/company/mailru/blog/115163/>
6. OpenId [Електронний ресурс] – Режим доступу до ресурсу:
[https://docs.microsoft.com/ru-ru/azure/active-directory/develop/v2-protocols-oidc#:~:text=OpenID%20Connect%20\(OIDC\)%20%E2%80%94%D1%8D%D1%82%D0%BE,%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B2%D1%85%D0%BE%D0%B4%D0%B0%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B5%D0%B9%20%D0%B2%20%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5.](https://docs.microsoft.com/ru-ru/azure/active-directory/develop/v2-protocols-oidc#:~:text=OpenID%20Connect%20(OIDC)%20%E2%80%94%D1%8D%D1%82%D0%BE,%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B2%D1%85%D0%BE%D0%B4%D0%B0%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B5%D0%B9%20%D0%B2%20%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5.)
7. Firestore [Електронний ресурс] – Режим доступу до ресурсу:
<https://firebase.google.com/docs/firestore.>
8. DI [Електронний ресурс] – Режим доступу до ресурсу:

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

https://ru.wikipedia.org/wiki/%D0%92%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8

9. What And Why React.js [Электронный ресурс] – Режим доступа до ресурсу: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>.
10. Вступ до NoSQL Баз Даних [Электронный ресурс] // Medium. – 2018. – Режим доступа до ресурсу: <https://medium.com/@mark.rethana/introduction-to-nosql-databases-c5b43f3ca1cc>.
11. How to Make a Good Database Design? [Электронный ресурс] // Medium. – 2020. – Режим доступа до ресурсу: <https://medium.com/swlh/how-to-make-a-good-database-design-584cd3cab5c5>.
12. JavaScript: The Definitive Guide, 7th Edition / David Flanagan // O'Reilly Media, Inc. – 2020 – Режим доступа до ресурсу: <https://www.oreilly.com/library/view/javascript-the-definitive/9781491952016/>.
13. A Smarter Way to Learn JavaScript: The new tech-assisted approach that requires half the effort/ Mark Myers// CreateSpace Independent Publishing Platform– 2014 – Режим доступа до ресурсу: <https://www.amazon.de/dp/1497408180?tag=hackr03c-21&geniuslink=true>.
14. Learning SQL / Alan Beaulieu // O'Reilly – 2012 – Режим доступа до ресурсу: http://www.r-5.org/files/books/computers/languages/sql/mysql/Alan_Beaulieu-Learning_SQL-EN.pdf.
15. How React hooks work - in depth [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.to/eliav2/how-react-hooks-work-in-depth-17o9>.
16. Frustrations with React Hooks [Электронный ресурс] – Режим доступа до ресурсу: <https://blog.logrocket.com/react-hooks-frustrations/>.
17. Clean Code / Robert C. Martin / Prentice Hall – 2008 – Режим доступа до

ресурсу: <https://enos.itcollege.ee/~jpoial/oop/naited/Clean%20Code.pdf>

18. Архітектура ОС Android – Режим доступу до ресурсу:

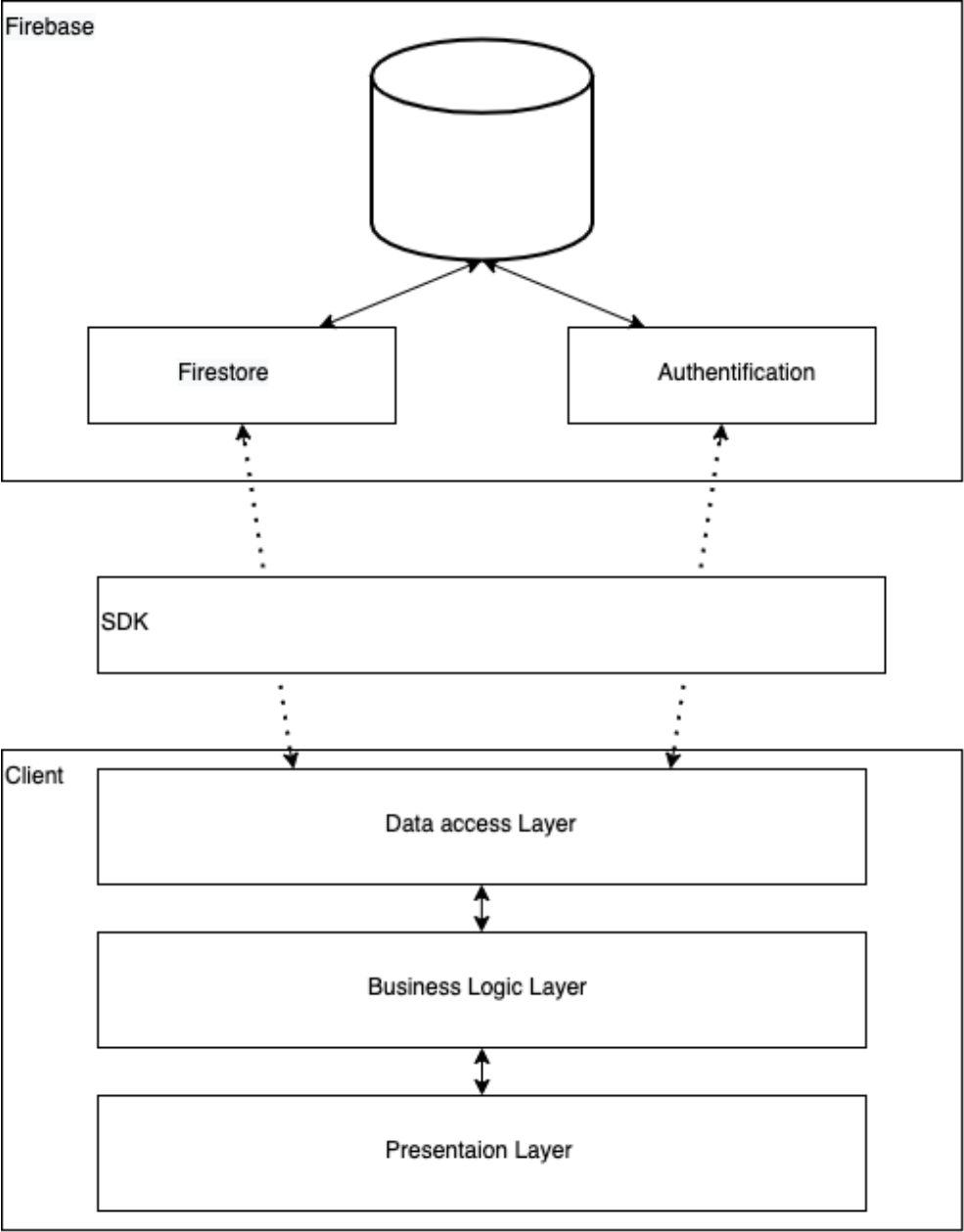
<https://studfile.net/preview/3651536/page:2/>

19. Платформу Firebase [Електронний ресурс] – Режим доступу до ресурсу:

<https://firebase.google.com/>

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

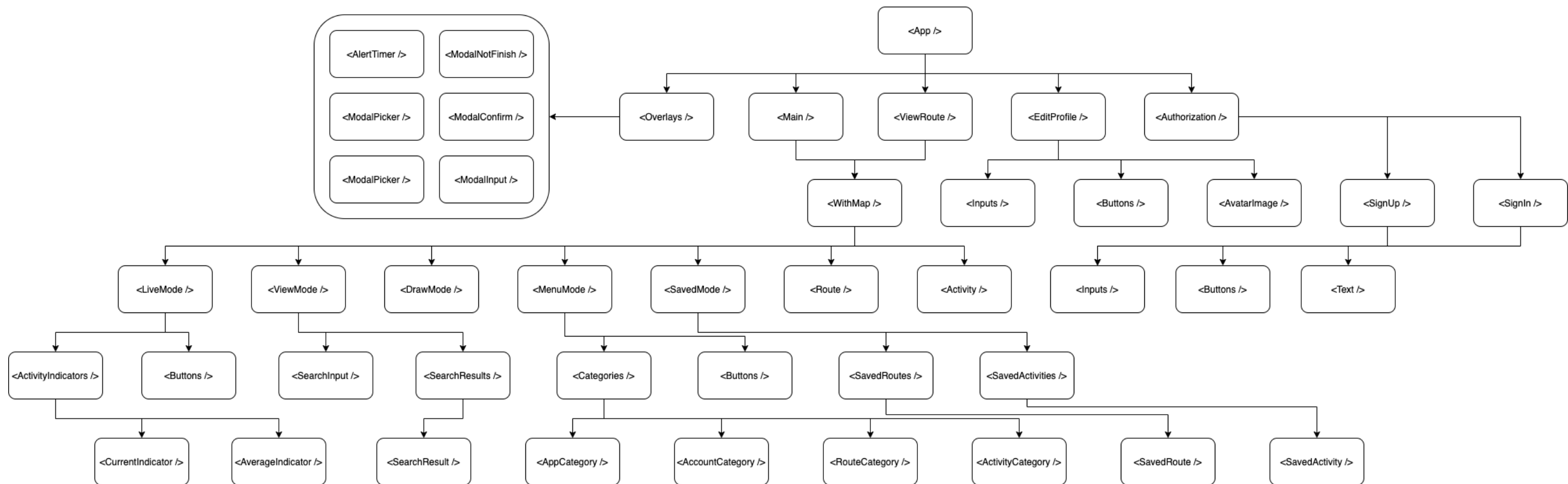
Додаток А
СТРУКТУРНА СХЕМА РОБОТИ
до дипломного проєкту
на тему: «Система трекингу активностей»



					ІАЛЦ.467100.004 Д1							
					Система трекингу активностей Структурна схема	Літера			Маса		Масштаб	
Зм.	Арк.	№ докум.	Підпис	Дата								
Розробив		Тіхонов С.О.										
Перевірів		Волокита І.М.										
						Аркуш 1			Аркушів 1			
					Дипломна робота	НТУУ "КПІ Ім Ігоря Сікорського", ФІОТ, гр. ІП-74						
Н.контр.		Сімоненко В.П.										
Затверд.		Стіренко С.Г.										

Додаток Б
ФУНКЦІОНАЛЬНА СХЕМА РОБОТИ
до дипломного проєкту
на тему: «Система трекингу активностей»

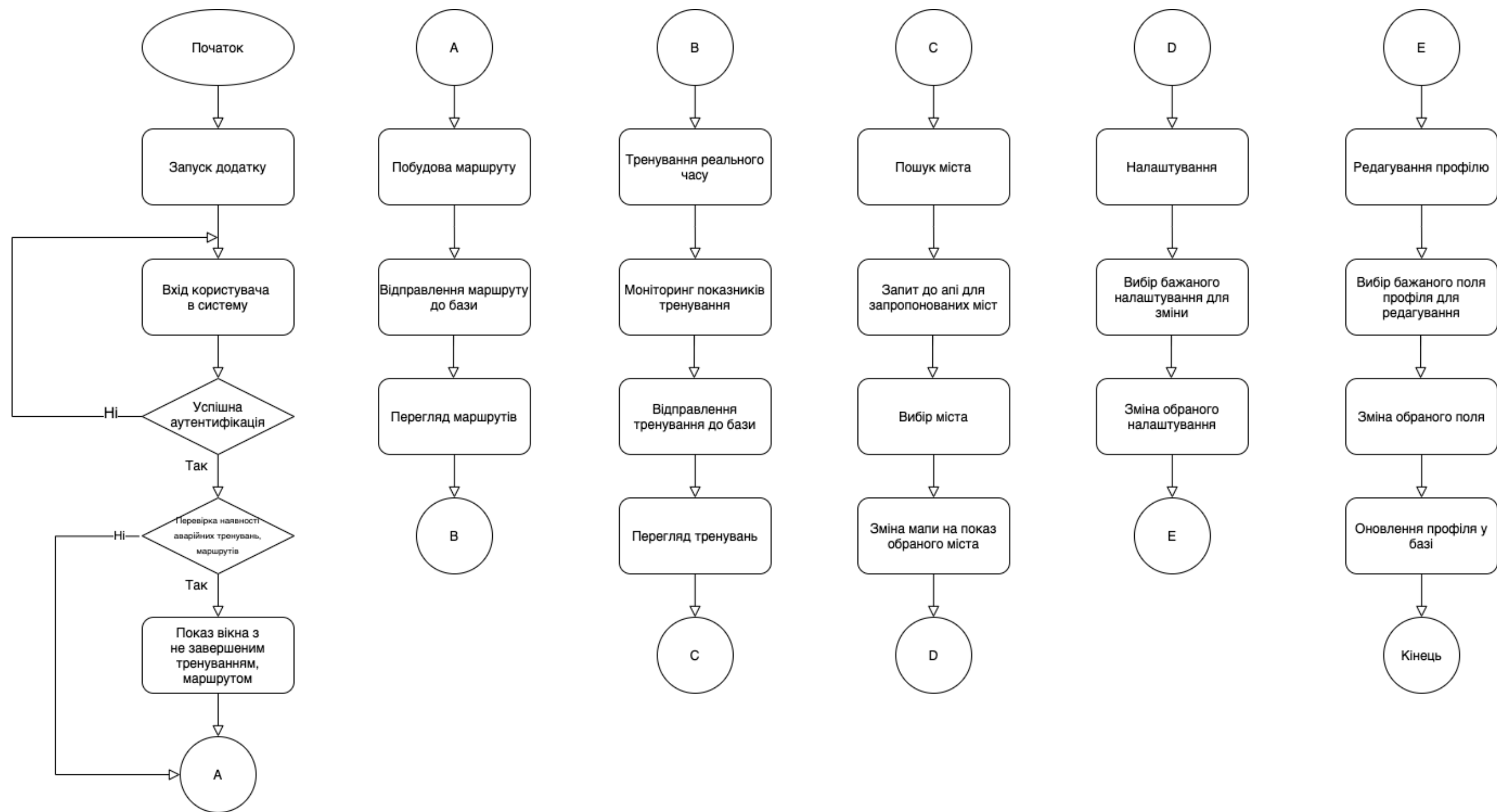
Київ – 2021 року



					ІАЛЦ.467100.005 Д2			
					Система трекингу активностей Функціональна схема (діаграма компонентів)			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Тіхонов С.О.						
Перевірів		Волокита І.М.						
Н.контр.		Сімоненко В.П.			Дипломна робота			
Затверд.		Стіренко С.Г.						
					Літера			
					Маса			
					Масштаб			
					Аркуш 1			
					Аркушів 1			
					НТУУ "КПІ Ім Ігоря Сікорського", ФІОТ, гр. ІП-74			

Додаток В
ПРИНЦИПОВА СХЕМА РОБОТИ
до дипломного проєкту
на тему: «Система трекингу користувачів»

Київ – 2021 року



					ІАЛЦ.467100.006 ДЗ						
					Система трекингу активностей Принципова схема (Блок-схема)	Літера		Маса	Масштаб		
Зм.	Арк.	№ докум.	Підпис	Дата							
Розробив	Тіхонов С.О.										
Перевірів	Волокита І.М.										
					Дипломна робота	Аркуш 1		Аркушів 1			
						НТУУ "КПІ Ім Ігоря Сікорського", ФІОТ, гр. ІП-74					

