

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мультимедійних систем факультету інформатики

**Розробка мобільного застосунку для відстеження активності на лонгборді  
за допомогою React Native**

**Текстова частина до курсової роботи  
за спеціальністю «Інженерія програмного забезпечення» - 121**

Керівник курсової роботи

с.в. Борозенний С.О.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Виконав студент \_\_\_\_\_

Слободяник М.А.  
(прізвище та ініціали)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Київ 2021  
Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри мультимедійних систем,

доцент, к.ф-м.н.

\_\_\_\_\_ О. П. Жежерун (підпис)

„\_\_\_\_\_” \_\_\_\_\_ 2020 р.

### ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студента Слободяника Максима Анатолійовича факультету інформатики 4-го курсу  
ТЕМА: Розробка мобільного застосунку для відстеження активності на лонгборді за допомогою React Native

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Розділ 1: Аналіз предметної області. Постановка завдання курсової роботи

Розділ 2: Теоретичні відомості

Розділ 3: Опис реалізації програмного продукту

Висновки

Список літератури

Додатки

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2022 р. Борозенний С.О. \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

## Календарний план виконання курсової роботи

**Тема:** Розробка MVP платформи з продажу вживаних книг

### Календарний план виконання курсової роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	28.09.2021	
2.	Аналіз матеріалів за темою	01.01.2022	
3.	Розробка практичної частини до курсової роботи	17.02.2022	
4.	Написання записки до курсової роботи	07.04.2022	
5.	Коригування виконаної роботи	05.05.2022	
6.	Створення слайдів для доповіді та написання доповіді.	10.05.2022	
7.	Кінцеве оформлення слайдів та роботи	15.05.2022	
8.	Захист курсової роботи	20.05.2022	

Слободяник М. А. \_\_\_\_\_

Борозенний С. О. \_\_\_\_\_

“            ”  
\_\_\_\_\_

## Зміст

<b>Зміст</b> .....	4
<b>Вступ</b> .....	5
<b>Розділ 1: Аналіз предметної області. Постановка завдання курсової роботи</b> .....	7
1.1 Аналіз сучасного стану питання та обґрунтування теми .....	7
1.1.1 Загальні відомості .....	7
1.1.2 Аналіз опитування, щодо уподобань лонгбордичної спільноти .....	7
1.2 Огляд існуючих аналогів .....	14
1.2.1 Krak Sesion .....	14
1.2.2 My Fitness Activity Tracker .....	15
1.2.3 Skateboard Tracking .....	16
1.3 Опис груп користувачів .....	17
1.4 Постановка задачі, функціональні вимоги до платформи .....	18
<b>Розділ 2: Теоретичні відомості</b> .....	19
2.1 Хмарні технології.....	19
2.2 JWT .....	21
2.3 ORM.....	23
<b>Розділ 3: Опис реалізації програмного продукту</b> .....	24
3.1 Опис технічного завдання на розробку .....	24
3.2 Обґрунтування вибору засобів розробки .....	27
3.3 Розробка програмного застосунку .....	30
3.3.1 Проектування бази даних .....	30
3.3.2 Створення компонентів React Native.....	31
3.3.3 Розробка серверної частини .....	34
3.4 Опис інтерфейсу застосунку .....	35
<b>Висновки</b> .....	46
<b>Список використаної літератури</b> .....	47

## Вступ

Попри зниження захворюваності на Covid-19 та поступове послаблення пандемічних обмежень, велика кількість населення планети все ще не оговталася від страху захворіти коронавірусом та отримати довгострокові наслідки хвороби. Це змушує людей шукати альтернативи громадському транспорту, який нерідко є місцем скупчення великої кількості людей з мінімальним або взагалі відсутнім дотриманням норм епідеміологічного дистанціювання. З цим фактором зокрема пов'язується зріст більш ніж на 180% попиту на велосипеди та електричні самокати, згідно з даними інтернет магазину Rozetka [1]. Проте вибір екологічного та мобільного транспорту не обмежується лише ними, все більшої популярності набирають скейтборди, лонгборди та електролонгборди адже на відміну від велосипедів та електросамокатів вони мають менші габарити, значно меншу вагу, а також не потребують додаткової міської інфраструктури для тимчасового або постійного зберігання бо можуть поміщатися в спеціальні рюкзаки. Окрім транспортної функції лонгборди також виконують розважальну та спортивну функцію, щороку ставляться та оновлюються рекорди на найбільшу розвинуту швидкість або на найдовший заїзд. Під час катання на лонгборді метрики грають важливу роль, адже завдяки їм можна оцінити свій прогрес, а у випадку поширення власних досягнень заохочувати інших до нових звершень. З огляду на це, у якості теми для своєї курсової роботи я обрав розробку мобільного застосунку для відстеження активності на лонгборді з використанням React native. В першому розділі я проаналізував власне опитування проведене серед учасників лонгбордичної спільноти Києва, щодо необхідності створення застосунку з зазначеним функціоналом. Також цей розділ містить огляд застосунків, що мають схожий функціонал. Завершується розділ описом груп користувачів та постановкою функціонального завдання. В другому розділі я навів теоретичні відомості щодо технологій, термінів та сервісів, що були використані мною під час створення застосунку. Третій розділ описує процес створення застосунку для відстеження

активності на лонгборді, описує основні складові та сервіси застосунку. Завершується розділ оглядом користувацького інтерфейсу створеного застосунку.

## Розділ 1: Аналіз предметної області. Постановка завдання курсової роботи

### 1.1 Аналіз сучасного стану питання та обґрунтування теми

#### 1.1.1 Загальні відомості

Лонгбординг, який також відомий як серфінг на асфальті - це напрямок скейтбордингу який виник у результаті поєднання серфінгу та скейтбордингу. Він виник на Гаваях в 60-х роках 20-го століття, коли перші лонгборди з'явилися у продажу. Вважається, що лонгбординг виник як можливість для серферів тренуватися під час погоди, що непридатна для тренувань на воді. Таким чином лонгбординг паралельно зі скейтбордингом розвивався протягом десятиліть набуваючи все більшої популярності не лише серед тих хто початково хотів присвятити себе цьому спорту, але і серед серферів та сноубордистів, які шукали для себе хобі на період закритого сезону. З того часу і до сьогодні популярність скейтбордингу а лонгбордингу то зростала то спадала, проте відповідно до даних сайту statista[2] зараз скейтбординг а відповідно і лонгбординг знову набирає свою популярність, що в свою чергу збільшує попит на пов'язані товари та послуги в тому числі і на трекеери активності.

#### 1.1.2 Аналіз опитування, щодо уподобань лонгбордичної спільноти

Для визначення актуальності створення додатку для відстеження активності на лонгборді, мною було проведено опитування [3] серед учасників лонгбордичної спільноти де були поставлені питання, що дозволяють визначити вік респондентів, найчастіші сценарії використання лонгбордів, додатки, що є аналогами до пропонованого застосунку, а також метрики, що є важливими для користувачів під час катання. Загалом в опитуванні взяли участь 117 респондентів. Було отримано наступні результати.

Під час створення застосунку важливо розуміти вікову категорію майбутніх користувачів, адже це впливає як на частоту використання додатку, так і на затребуваний функціонал.

Скільки вам років

117 відповідей

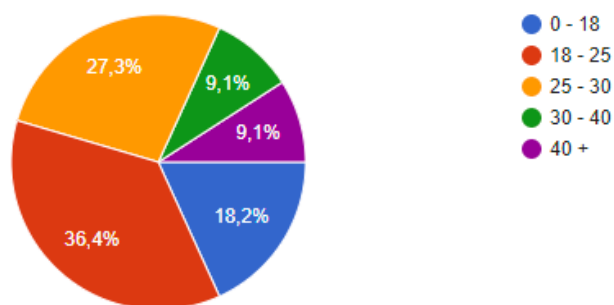


Рис. 1.1 Кругова діаграма розподілу відповідей на питання, про вік учасника

Найбільшою часткою респондентів виявилася молодь віком від 18 до 25 років (36.4% опитаних) та від 25 до 30 років (27.3% опитаних). Меншою за чисельністю виявилася група віком від 0 до 18 років (18.2% опитаних). Групи респондентів від 30 до 40 років (9.1% опитаних) та більше 40 років (9.1% опитаних) виявилися найменш чисельними.

Для яких цілей ви використовуєте лонгборд

117 відповідей

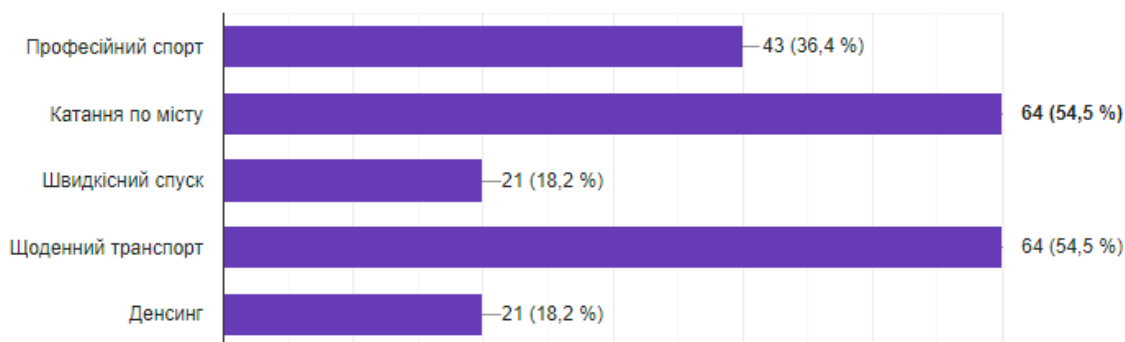


Рис. 1.2 Стовпчикова діаграма розподілу відповідей на питання про основні сценарії використання лонгборду



Найбільше респондентів використовують лонгборд у якості повсякденного транспорту (54.5% респондентів) та для катання по місту (54.5% респондентів). Менш популярними є вузько спрямовані дисципліни які потребують більше навиків та тренувань, такі як професійне катання (36.4% респондентів), швидкісний спуск (18.2% респондентів) та дансинг (18.2% респондентів).

Як довго ви катаєтесь на лонгборді?

117 відповідей

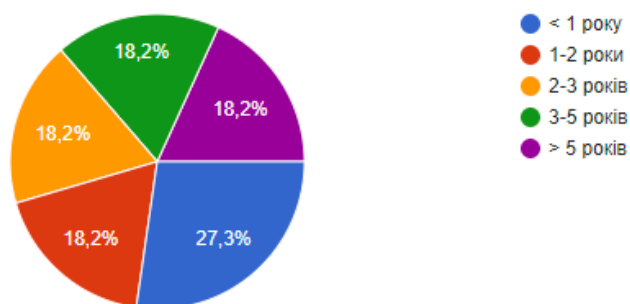


Рис. 1.3 Кругова діаграма розподілу відповідей на питання, щодо кількості років катання на лонгборді

27% опитаних є новачками в лонгбордингу та катаються менше року. Інші групи за досвідом розподілилися порівну та мають по 18% респондентів кожна. Таким чином можна сказати, що опитування повноцінно охоплює різні за професійністю групи

Чи вам було легко знайти лонгбордичну спільноту свого міста?

117 відповідей

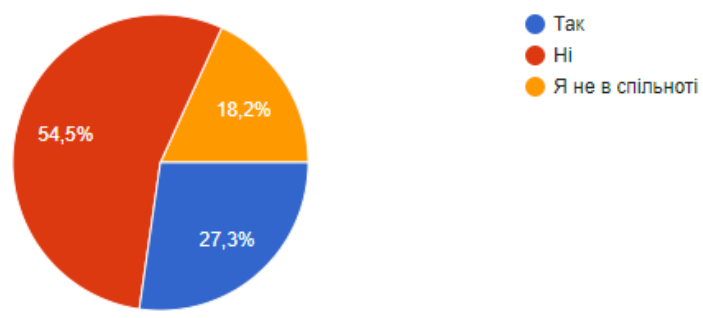


Рис. 1.4 Кругова діаграма розподілу відповідей на питання, щодо складності пошуку лонгбордичної спільноти свого міста.

Оскільки спільнота відіграє важливу роль в прогресуванні в будь-якій справі, мені б хотілося додати до майбутнього застосунку спосіб, який би дозволяв легше знаходити та об'єднуватися з однодумцями. На мою думку для досягнення цієї цілі потрібно максимально спростити пошук спільноти для людей які розглядають лонгбординг як майбутнє захоплення. Проте на даний момент більшість респондентів (54.5% опитаних) зазначають, що пошук спільноти не було легкою справою. Лише 27.3% опитаних легко знайшли спільноту свого міста.

Яким чином відбувається комунікація всередині вашої лонгбордичної спільноти?

117 відповідей

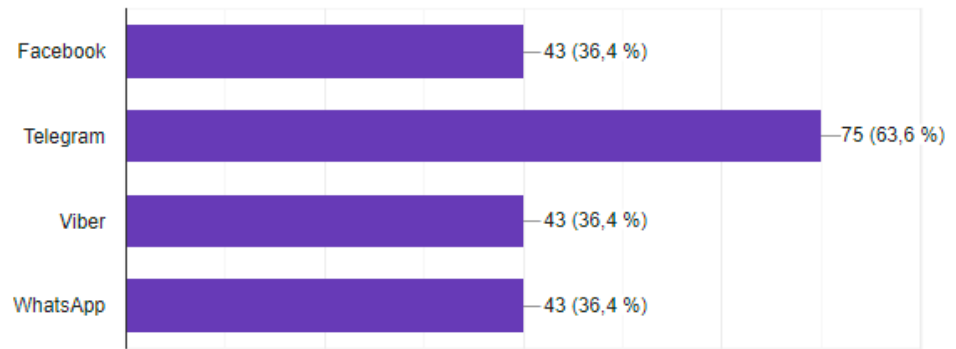


Рис. 1.5 Стовпчикова діаграма розподілу відповідей на питання про основні способи комунікації всередині спільноти

Чи є зручним для вас вид комунікації в спільноті?

117 відповідей

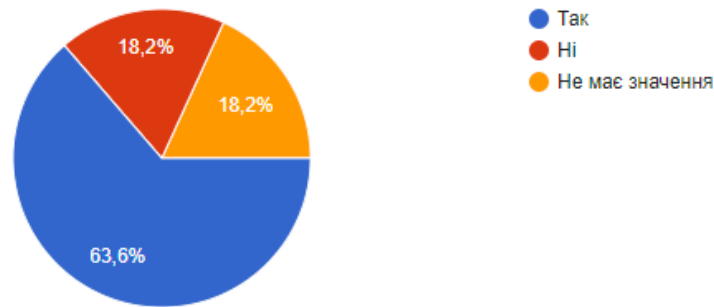


Рис. 1.6 Кругова діаграма розподілу відповідей на питання, щодо зручності спілкування в наявних каналах комунікації спільнот

Спілкування всередині спільноти є не менш важливим фактором її розвитку та прогресування учасників, тож наступне питання стосувалося способів комунікації онлайн та зручності такого спілкування. Згідно з опитуванням найпопулярнішою платформою для комунікації в спільноті є Telegram (63.6% опитаних використовують його). Facebook, Viber та WhatsApp розділили між собою друге місце за уподобаннями та мають по 36.4% відповідей. 63.6% опитаних вважають теперішній спосіб спілкування в спільноті зручним.

Чи вчасно ви дізнаєтеся про події в лонгбордичній спільноті вашого міста?

117 відповідей

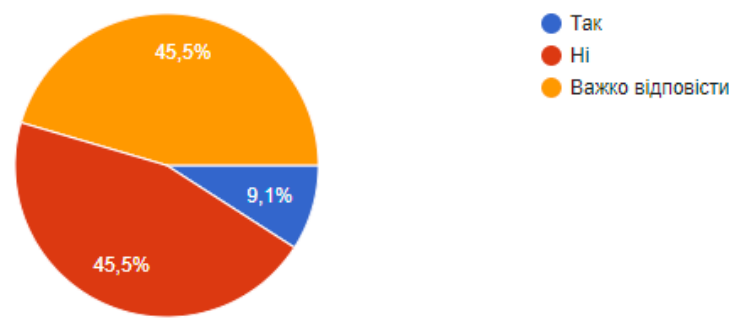


Рис. 1.7 Кругова діаграма розподілу відповідей на питання, щодо своєчасного сповіщення спільноти про майбутні активності

Чи користуєтеся трекером для відстеження активності на лонгборді

117 відповідей



Рис. 1.8 Кругова діаграма розподілу відповідей на питання, щодо користування трекером для відслідковування активності на лонгборді

Оскільки основним функціоналом майбутнього додатку буде трекінг активності, я спробував дізнатися тенденції щодо використання наявних трекерів серед спільноти. Лише 18.2% опитаних не використовують трекер. 36.4% використовують один з наявних на ринку трекерів. 45.9% не знайшли серед пропозицій трекера який би задовольняв їхнім потребам.

Якщо в попередньому питанні ви відповіли так, напишіть який застосунок використовуєте  
28 відповідей

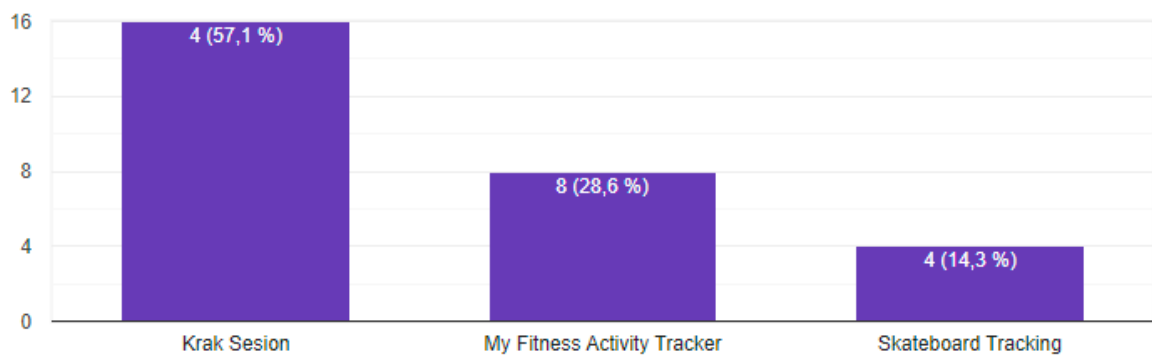


Рис. 1.9 Стовпчикова діаграма розподілу відповідей на питання про трекери які зараз використовуються серед спільноти

Найпопулярнішим трекером серед спільноти на зараз є Krak Session.

Які метрики під час сесії катання є найважливішими для вас?  
117 відповідей

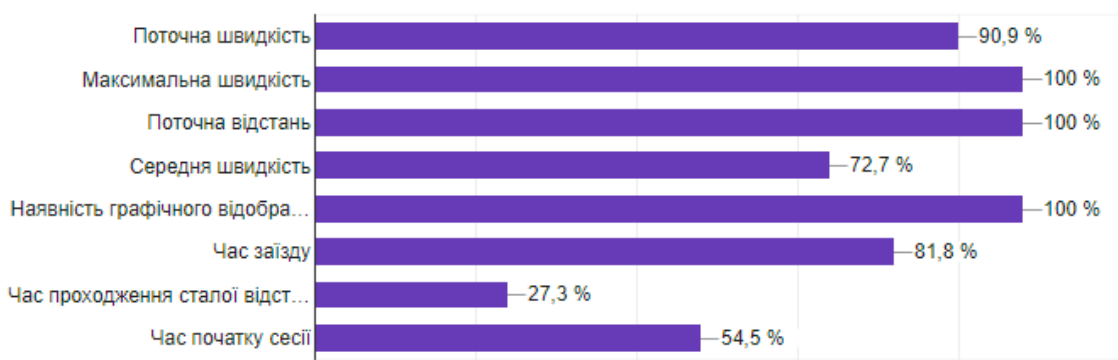


Рис. 1.10 Стовпчикова діаграма розподілу відповідей на питання про найважливіші метрики під час катання

Аналізуючи результати опитування, можна дійти висновку, що спільнота потребує трекеру для відстеження своєї активності на лонгборді. Також затребуваною функцією є спрощення пошуку активностей всередині спільноти. Водночас потреба в

заміні наявних способів комунікації всередині спільноти – відсутня, адже більшість учасників вона влаштовує. Важливими метриками під час трекінгу було визначено такі показники: поточна швидкість, середня швидкість, максимальна швидкість, поточна відстань, наявність графічного відображення маршруту та час заїзду.

## 1.2 Огляд існуючих аналогів

Враховуючи результати опитування проведеного мною в розділі 1.1 я обрав три найчастіше згадувані аналогічні додатки для огляду їхнього функціоналу.

### 1.2.1 Krak Sesion

Krak Sesion - один з спеціалізованих трекерів для скейтбордів наявних в AppStore. Він має базовий функціонал відстеження активності, проте під час сесії доступні лише три показники: швидкість, відстань та час. Графічне відображення пройденого шляху відображається лише після закінчення сесії і вбудована карта не доступна під час сесії. Застосунок має також два додаткових екрани: попередні екрани та статистика за весь час. Статистика містить такі показники як: максимальна швидкість, середня швидкість, загальний витрачений час, кількість сесій та загальний пройдений шлях. Показники максимальної швидкості не доступні для конкретної сесії. Екран попередніх сесій, містить попередні сесії, при відкритті однієї з них, нам стає доступною додаткова інформація.

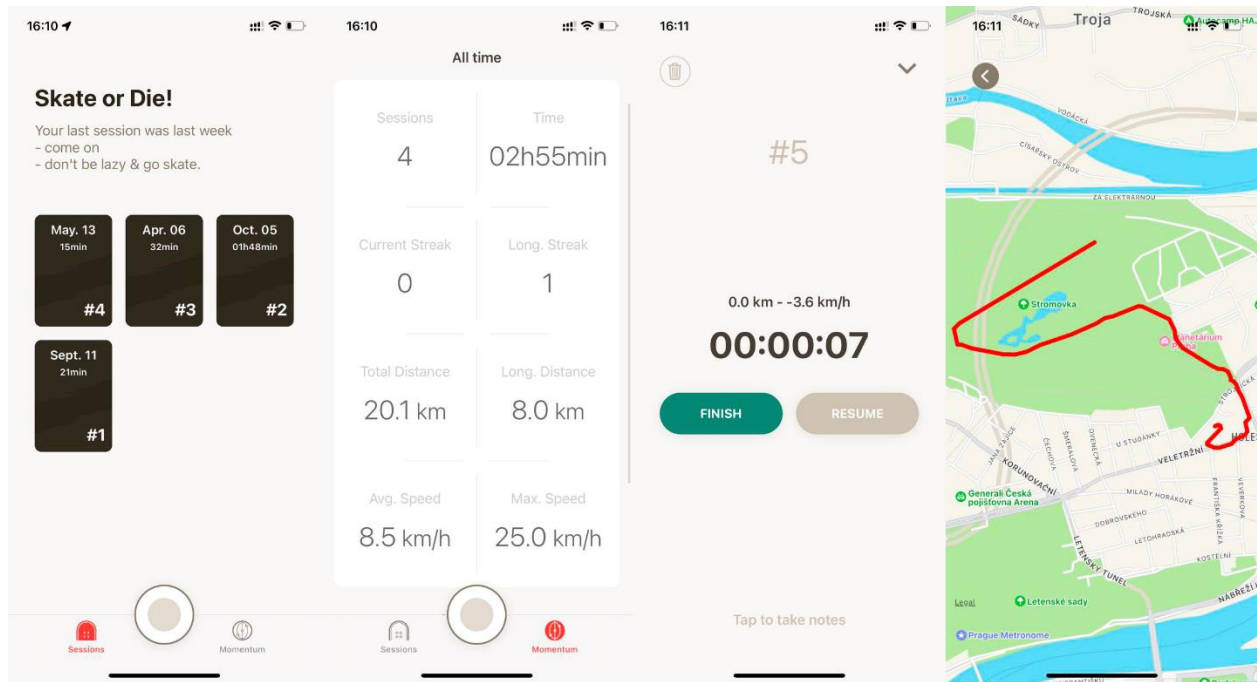


Рис. 1.11 Інтерфейс додатку Krak Sesion[4]

### 1.2.2 My Fitness Activity Tracker

My Fitness Activity Tracker – додаток що позиціонує себе як універсальний трекер. Він має базовий функціонал відстеження активності, під час сесії доступні такі показники: максимальна швидкість, середня швидкість, витрачений час та пройдений шлях. Також доступна вбудована карта, проте вона не відображає пройдений шлях в реальному часі. Також доступний екран з попередніми сесіями та сторінка з загальною статистикою за увесь час. З проблем можу зазначити відсутність обрахування середньої швидкості, адже в інформації завжди відображається помилка – не число.

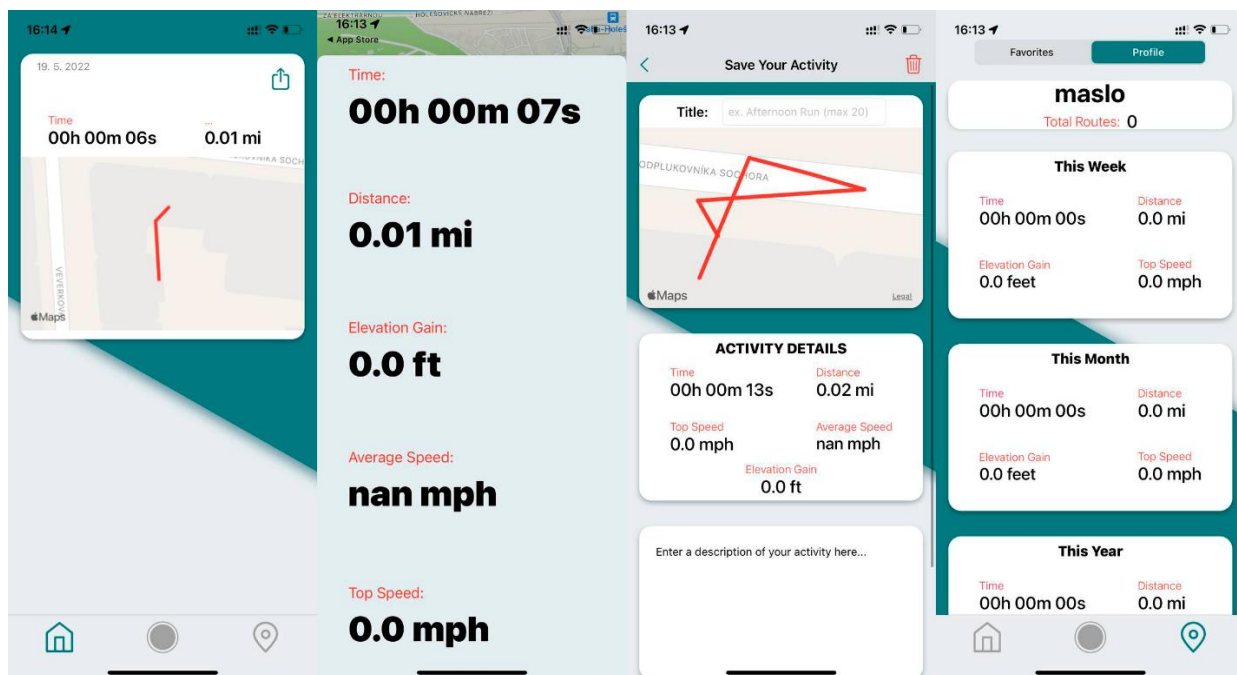


Рис. 1.12 Інтерфейс додатку My Fitness Activity Tracker[5]

### 1.2.3 Skateboard Tracking

Skateboard Tracking - не містить функції запису сесій та лише відображає поточні координати, швидкість руху, середню швидкість і максимальну швидкість.

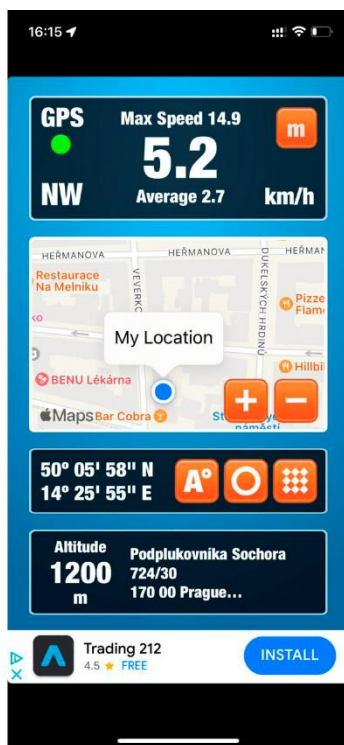




Рис. 1.13 Інтерфейс додатку Skateboard Tracking[6]

### 1.3 Опис груп користувачів

Застосунок матиме одну групу користувачів, а саме User. До групи User належить будь-який користувач, який самостійно зареєструвався в системі. Користувач з групи User може виконувати такі дії в застосунку:

#### 1) Робота з акаунтом:

- реєстрація акаунту за допомогою електронної пошти користувача, імені користувача та пароллю;
- Вхід до акаунту за допомогою електронної пошти користувача та пароллю
- Видалення акаунту;

#### 2) Робота з профілем:

- Редагування інформації акаунту (ім'я користувача, пароль, місце де живе користувач, аватар користувача, опис користувача, telegram ідентифікатор користувача)

#### 3) Взаємодія з спільнотою:

- Текстовий пошук серед учасників спільноти
- Створення підписок на учасників спільноти
- Скасування підписок на учасників спільноти

#### 4) Взаємодія з сесіями

- Перегляд списку власних сесій
- Перегляд детальної інформації про власну сесію
- Перегляд списку сесій учасників спільноти
- Перегляд детальної інформації про сесії учасників спільноти
- Видалення власних сесій

#### 5) Взаємодія з подіями

- Перегляд списку подій
- Перегляд детальної інформації про подію

#### 6) Взаємодія з трекінгом активності

- Початок сесії
- Призупинення сесії
- Перегляд метрик під час сесії
- Перегляд графічного відображення маршруту під час сесії
- Зупинка сесії
- Збереження сесії

#### 1.4 Постановка задачі, функціональні вимоги до платформи

Задачею моєї курсової є розробка мобільного застосунку та серверної частини до нього, що надає можливості для зручного відстеження власної активності на лонгборді та легкої взаємодії з локальною лонгбордичною спільнотою. Проаналізувавши аналогічні додатки для трекінгу, я дійшов висновку, що застосунок повинен відповідати таким функціональним вимогам:

- Реєстрація нових та вхід в систему для вже існуючих користувачів .
- Пошук людей серед спільноти за ім'ям
- Підписка на людей зі спільноти.
- Відстеження власної активності на лонгборді
- Перегляд своїх попередніх сесій активності на лонгборді
- Перегляд графічного відображення шляху подоланого за сесію
- Перегляд метрик зібраних за сесію
- Перегляд попередніх сесій активності на лонгборді людей на яких підписаний користувач
- Створення редагування та видалення власної події.
- Перегляд подій які організовують люди на яких підписаний користувач
- Перегляд та редагування користувачем свого профілю

## Розділ 2: Теоретичні відомості

### 2.1 Хмарні технології

На сьогодні хмарні технології перетворилися з терміну про який говорять як про майбутнє розробки в фактичний стандарт розробки. Безліч компаній в незалежності від свого домену починають мігрувати свої сервіси в хмару. На це є декілька причин. Перш за все утримання сервісів в хмарі коштує дешевше ніж утримання власних дата центрів, адже за рахунок своїх розмірів хмарні провайдери можуть пропонувати меншу ціну за ту ж потужність, не варто також виключати вартість обслуговування серверного обладнання. По-друге, хмарні сервіси дозволяють розташовувати контент максимально близько до кінцевих користувачів тим самим покращуючи користувацький досвід та знижуючи затримки. По-третє хмарні сервіси пропонують значно вищу безпеку даних, гнучкість та захист від природніх катаклізмів, через що вважають ці більш безпечними ніж нерозподілені дата центри. Термін хмарних обчислень був створений в 1996 році, а вже в 2006 році компанія Amazon запустила свої перші сервіси.

Хмарні сервіси поділяються на три основних види (моделі надання послуг):

- IAAS – інфраструктура як сервіс, що фактично означає надання в оренду інфраструктури для розгортання там клієнтського програмного забезпечення. Такий тип надання сервісу є найбільш гнучким, адже всі налаштування, враховуючи операційну систему інфраструктури користувач налаштовує самостійно.
- PAAS – платформа як сервіс, в цьому випадку користувач отримує робочу платформу з попередньо встановленою операційною системою. В такому разі користувач відповідальний лише за дані та застосунки які він запускає в хмарі.

- SAAS – програмне забезпечення як сервіс, надає послуги роботи програмного забезпечення, тож користувач такого сервісу фактично є звичайним користувачем програмного забезпечення та не налаштовує нічого самостійно.

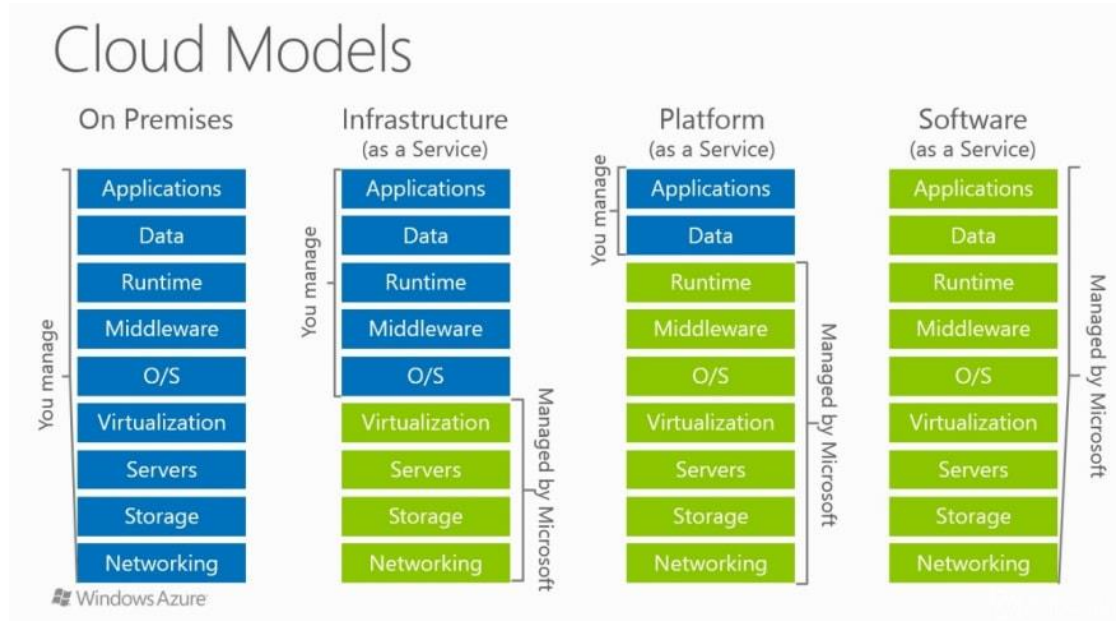


Рис. 2.1 Візуальна ілюстрація відмінностей моделей надання послуг хмарними провайдерами[7]

На жаль, через обмеження безпеки, або обмежень на розміщення даних поза територією певної країни, не всі компанії можуть повністю мігрувати в хмару. Для таких випадків хмарні провайдери розробляють так звані гібридні моделі. Це організація роботи сервісу, коли ви за допомогою спеціального програмного забезпечення поєднуєте свій власний дата центр та сервіси хмарного провайдера. Це дозволяє принаймні частково здешевити вартість утримання власних дата центрів. Найпопулярнішими хмарними провайдерами на зараз є AWS (39% ринку), Microsoft Azure(19% ринку) та GCP(9% ринку).

У своїй роботі я використав сервіси AWS оскільки він є найпопулярнішим надавачем хмарних послуг. Зокрема було використано сервіс S3.

## 2.2 JWT

JWT (JSON Web Tokens) – це JSON об'єкт, що визначається стандартом RFC 7519. Стандартна будова JWT включає в себе заголовок, корисне навантаження а також підпис. Спробуємо розглянути будову на базі токenu: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjhhN2FjYWJiLTdhMDctNDE3Ny1hODMyLWQ5M2UzYWw0NWFhbnR5cCI6ImVtYWlsIjoidmFzeWEzQGdtYWlsLmNvbSI6ImhhdCI6MTY1MTk5NzAwOCwiZXhwIjoxNjUyMDgzNzA4fQ.SOs1q9T6cW_suwffE5wHcS7uuS63ahgWBSNcA29kAIg`

В даному прикладі заголовком є частина токenu: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9`. Вона містить загальну інформацію про токен, таку як алгоритм шифрування та тип токenu. Дані заголовку шифруються.

У випадку розшифрування можемо бачити такий JSON об'єкт: `{ "alg": "HS256", "typ": "JWT" }`

Корисним навантаженням в нашому випадку є ця частина токenu: `eyJpZCI6IjhhN2FjYWJiLTdhMDctNDE3Ny1hODMyLWQ5M2UzYWw0NWFhbnR5cCI6ImVtYWlsIjoidmFzeWEzQGdtYWlsLmNvbSI6ImhhdCI6MTY1MTk5NzAwOCwiZXhwIjoxNjUyMDgzNzA4fQ`

В розшифрованому вигляді це також JSON об'єкт: `{ "id": "8d7acabb-7a07-4177-a832-d93e3ac45ab5", "email": "vasya3@gmail.com", "iat": 1651997308, "exp": 1652083708 }`

Корисне навантаження може містити в собі будь-які дані необхідні для ідентифікації користувача, також воно містить час створення токена та коли він перестане бути дійсним. Підпис обраховується на основі перших двох частин. Після цього всі три

частини поєднуються через крапку, таким чином формуючи кінцевий токен.

**Encoded**
PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Ijhhkn2FjYWJlTdhMDctNDE3Ny1hODMyLWQ5M2UzYWM0NWFiNSIsImVtYWlsIjoiaWZlbnR5cCI6IjE5MTk5NzMwOCwzZXhwIjozNjUyMDgzNzA4fQ.S0s1q9T6cW_suwfFE5wHcS7uuS63ahgWBsNcA29kAIg
```

**Decoded**
EDIT THE PAYLOAD AND SECRET

**HEADER: ALGORITHM & TOKEN TYPE**

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

**PAYLOAD: DATA**

```
{
  "id": "8d7acabb-7a07-4177-a832-d93e3ac45ab5",
  "email": "vasya3@gmail.com",
  "iat": 1651997308,
  "exp": 1652083708
}
```

**VERIFY SIGNATURE**

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Рис. 2.2 Ілюстрація зашифрованого та розшифрованого JWT токена[8]

В своєму застосунку я використовую JWT токен для аутентифікації користувача. У відповідь на надіслані коректні логін та пароль, серверна частина поверне JWT токен для подальшої взаємодії сервером

В мобільному додатку після отримання токена він зберігатиметься в захищеному сховищі та використовуватиметься при кожному подальшому надсиланні запиту.

```

@Inject()
export class JwtStrategy extends PassportStrategy(Strategy) {
  @Inject(AuthHelper)
  private readonly helper: AuthHelper;

  constructor(private readonly configService: ConfigService) {
    super({
      jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
      ignoreExpiration: false,
      secretOrKey: configService.get<string>('auth.secret'),
    });
  }

  async validate(payload: User) {
    return this.helper.validateUser(payload);
  }
}

```

Рис. 2.3 Ілюстрація обробки автентифікації користувача на сервері з використанням JWT токєну

## 2.3 ORM

ORM (Object-Relational Mapping) – технологія, чия суть полягає в створенні віртуальної об’єктної бази даних. Її використання дозволяє уникнути ручного перетворення записів в базі даних в об’єктно-орієнтовані моделі. Також використання ORM дозволяє генерацію схеми бази даних базуючись на моделях об’єктно-орієнтованої парадигми. Оскільки лише за допомогою класу неможливо описати всі тонкощі реляційної моделі, в ORM що працюють з TypeScript використовуються декоратори для додавання додаткових метаданих полям, організації зв’язків тощо.

В своєму застосунку я використовую TypeORM[9]. За допомогою цієї ORM також можливо генерувати міграції, які створюються на базі аналізу різниці між моделлю даних та актуальним станом схеми в СКБД.

## **Розділ 3: Опис реалізації програмного продукту**

### **3.1 Опис технічного завдання на розробку**

Метою виконання технічного завдання – є розробка застосунку для відстеження активності на лонгборді. Суттєвими відмінностями, що покращать користувацький досвід цього додатку в порівнянні з наведеними Розділі 1 аналогами, повинні стати: спеціалізація виключно на трекінгу активності на лонгборді, можливість поширити власну активність серед підписників, можливість перегляду всіх майбутніх подій спільноти свого міста, можливість відслідковувати всі необхідні метрики як під час сесії так і після її завершення. Результуюча робота повинна складатися з мобільного застосунку та серверу для обробки запитів.

Додаток повинен мати такий функціонал:

1. Надавати можливість реєстрації користувача в системі.
2. Надавати можливість входу користувача до системи.



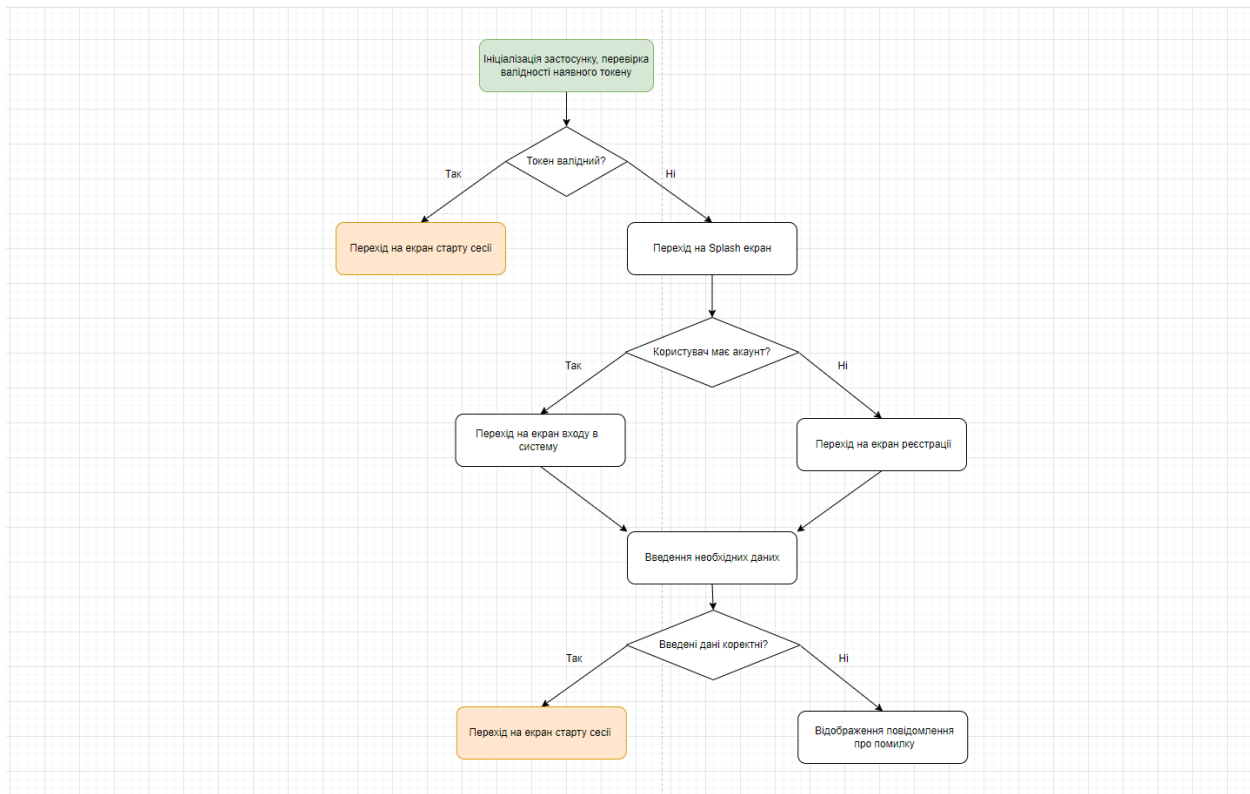


Рис. 3.1 Блок-схема шляху користувача під час процесу реєстрації та входу в систему

3. Надавати можливість текстового пошуку учасників серед спільноти
4. Надавати можливість підписуватися на учасників спільноти
5. Надавати можливість створювати сесії відстеження активності з відслідковуванням таких параметрів:
  - Поточна швидкість
  - Максимальна досягнута швидкість сесії
  - Середня швидкість сесії
  - Відстань подолана за сесію
  - Тривалість сесії

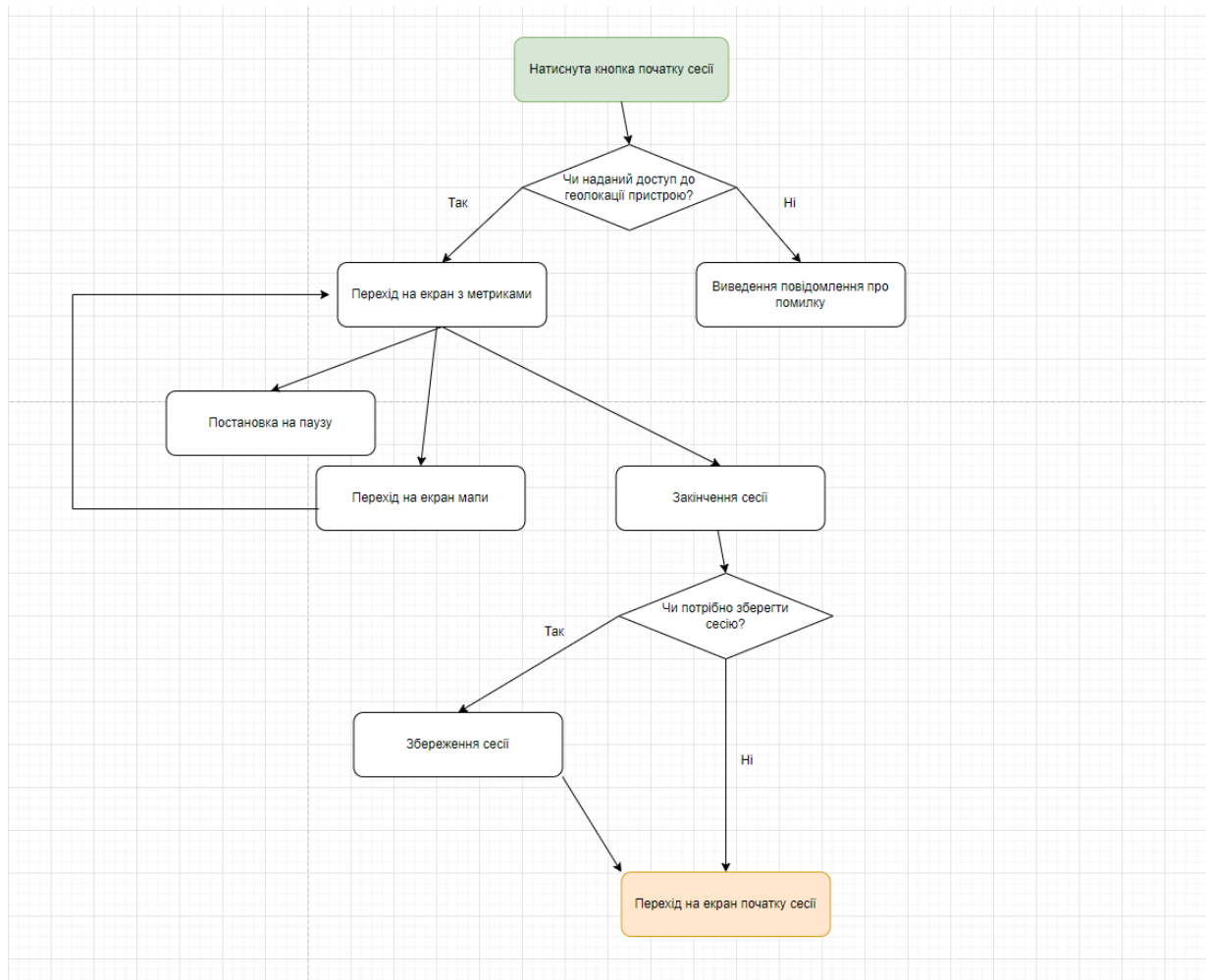


Рис. 3.2 Блок-схема шляху користувача під час відстежувальної сесії

6. Надавати можливість перегляду деталей сесій користувача.
7. Надавати можливість перегляду деталей сесій учасників спільноти
8. Надавати можливість видалення власних сесій
9. Надавати можливість перегляду списку подій.
10. Надавати можливість створювати власні події з такими даними: фото події, заголовок, опис, організатори, ціна за участь, точка старту, точка фінішу.

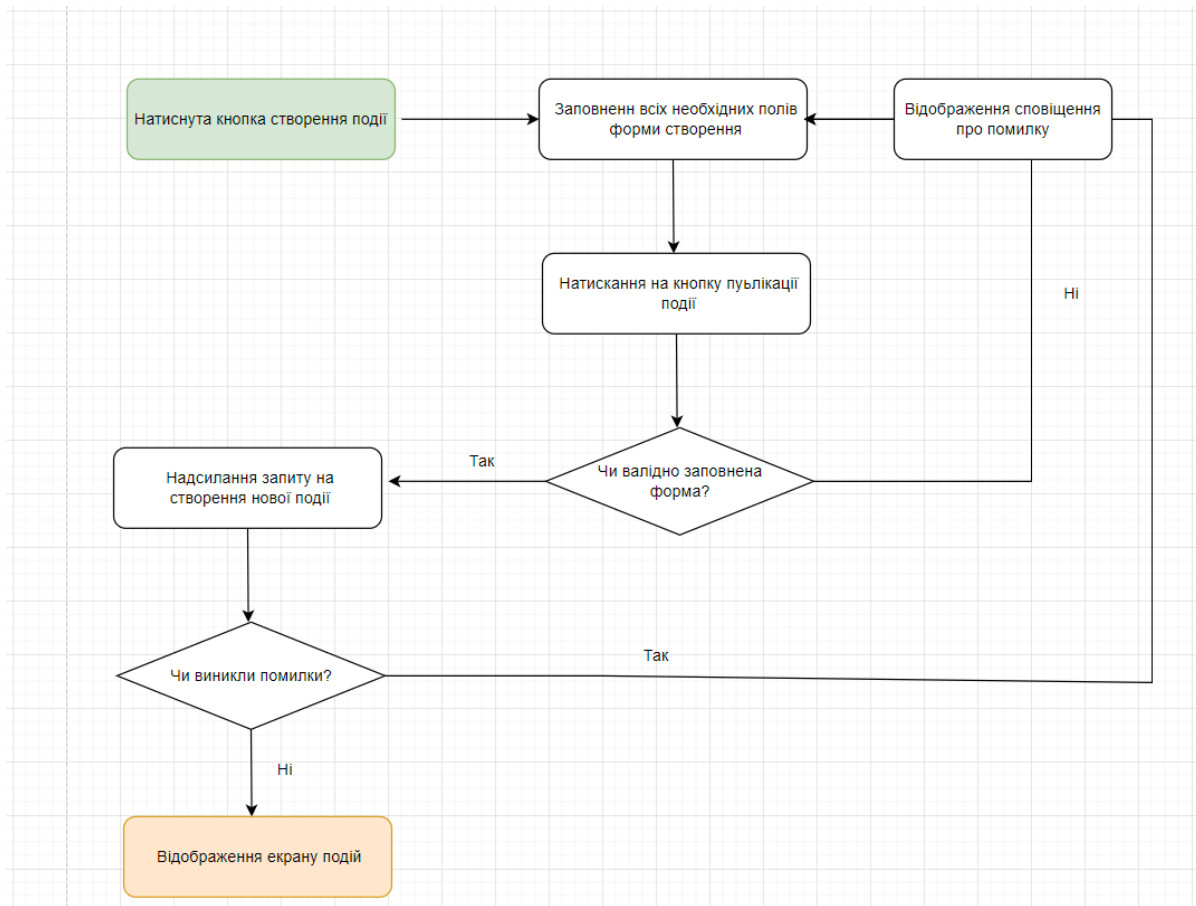


Рис. 3.3 Блок-схема шляху користувача під час процесу додавання події

- 11.Надавати можливість редагування власних подій
- 12.Надавати можливість видалення власних подій.
- 13.Надавати можливість перегляду підписників
- 14.Надавати можливість скасування підписок на учасників спільноти
- 15.Надавати можливість перегляду учасників спільноти на яких підписаний користувач
- 16.Надавати можливість перегляду власного акаунту
- 17.Надавати можливість редагування власного акаунту

### 3.2 Обґрунтування вибору засобів розробки

Робота складається з мобільного застосунку, серверної частини та бази даних. Для мобільного застосунку я обрав реалізацію з використанням бібліотеки React Native[10] у зв'язці з Redux[14] для організації store.

Під час вибору інструменту для розробки мобільного додатку я обирав з поміж бібліотеки React Native, фреймворку Flutter[11] та Створення нативного застосунку окремо для операційних систем android та IOS. При створенні будь-якого програмного продукту який покликаний принести прибуток, потрібно зважати на витрати під час розробки. Створення нативних застосунків є хоча і менш обмеженим проміжними інструментами, однак потребує розробників з вмінням розробляти нативні додатки окремо під кожну платформу, що автоматично збільшує час та ціну розробки. Тому зрештою я обирав між розробкою за допомогою Flutter та React Native, адже обидва ці інструменти дозволяють використовувати одну кодову базу для розробки під різні платформи вносячи лише невеликі зміни. Розробка на Flutter має декілька недоліків, а саме: неможливість тестування IOS застосунку без наявності пристрою компанії apple, що потребує додаткових витрат, а також менша спільнота мови Dart в порівнянні зі спільнотою JavaScript розробників. Тому було обрано реалізовувати мобільний додаток за допомогою React Native. Він базується на бібліотеці React.js розробленої компанією Facebook, проте на відміну від нього фокусується не на розробці веб додатків, а на розробці мобільних. Більш того, якщо використовувати TypeScript під час розробки мобільного додатку, то існує можливість перевикористати велику частину кодової бази мобільного застосунку під час реалізації веб складової. Під час своєї роботи бібліотека трансліює створені ористувачем елементи в нативні елементи обраної платформи, при цьому змінюючи мову логіки. JavaScript викликатиметься під час виконання додатку за допомогою вбудованих нативних інструментів. Також я використав бібліотеку Redux для керування станом застосунку. У разі його використання, всі дані, якими оперує застосунок зберігаються в одному місці - стані. Це спрощує підтримку додатка, та робить його підтримку більш легкою за умови розростання проекту. Під час розробки

я використовував reduxDevTools[15], для відлагодження роботи додатку, цей плагін дозволяє переглядати зміни в стані та його актуальний стан, що пришвидшує та спрощує розробку. Підсумовуючи, мною була обрана така зв'язка бібліотек у зв'язку з актуальністю технологій, швидкістю розробки та легкістю подальшої підтримки. Для серверної частини я обрав реалізацію з використанням фреймворку Nest.js[12] та мови TypeScript.

Nest.js – це бібліотека, яка набирає популярність серед розробників, та цілком може стати новим стандартом в розробці серверних застосунків на JavaScript/TypeScript. Цей фреймворк є модульним, що дозволяє не збільшувати кінцевий бандл застосунку шляхом збереження непотрібних залежностей. Завдяки такій гнучкості і простоті в налаштуваннях проекти можливо швидко запускати та додавати новий функціонал. TypeScript дозволяє проектам залишатися підтримуваним попри зростання кодової бази. У якості ORM я обрав typeORM, через велику спільноту, що полегшить вирішення проблем у разі їхнього виникнення.

У якості бази даних я використав СКБД PostgreSQL[13]

PostgreSQL - об'єктно-реляційна система керування базами даних. Вона має такі переваги:

- Можливість створення тригерів та функцій.
- Наявність індексів, що пришвидшує пошук даних.
- Підтримка бази даних необмеженого розміру, що є корисним для застосунків, що планують активно масштабуватися.
- Наявність індексів, що пришвидшує пошук даних.
- Надійність та стабільність.
- Потужні механізми транзакцій та реплікацій.

ORM TypeOrm має широку підтримку PostgreSQL.

### 3.3 Розробка програмного застосунку

#### 3.3.1 Проектування бази даних

База даних застосунку складається з 7 таблиць

Розглянемо таблицю users. Вона містить первинний ключ id типу uuid.

Зовнішній ключ avatarId типу uuid, а також інші поля:

email типу varchar, username типу varchar, password типу varchar,

description типу varchar, location типу varchar, telegram типу varchar,

Таблиця photos містить в собі всі зображення, що були завантажені через додаток. Вона має первинний ключ id типу uuid, а також інші поля: url типу varchar та key типу varchar.

Таблиця community\_activities містить в собі події спільноти. Вона має первинний ключ id типу uuid. Та зовнішній ключ primePhoto\_id типу uuid, а також інші поля:

name типу varchar, date типу timestamp, location типу varchar, description

типу varchar, startPoint типу varchar, endPoint типу varchar, price типу integer.

Таблиця community\_activities\_photos містить в собі зв'язок “багато до багатьох”, між таблицями community\_activities та photos.

Таблиця community\_activities\_organizers\_users містить в собі зв'язок “багато до багатьох”, між таблицями community\_activities та users. За допомогою неї я зв'язую організаторів подій з подіями.

Таблиця subscriptions містить в собі зв'язок “багато до багатьох” між таблицею users та users. Вона використовується для організації підписки на користувача користувачем.

Таблиця sessions містить в собі сесії користувачів. Вона має первинний ключ id типу uuid. Та зовнішній ключ doer\_id типу uuid, а також інші поля:

name типу varchar, date\_of\_ride типу timestamp, total\_distance типу numeric, max\_speed типу numeric, avg\_speed типу numeric, total\_dutration типу varchar, coordinates типу varchar, price типу integer.

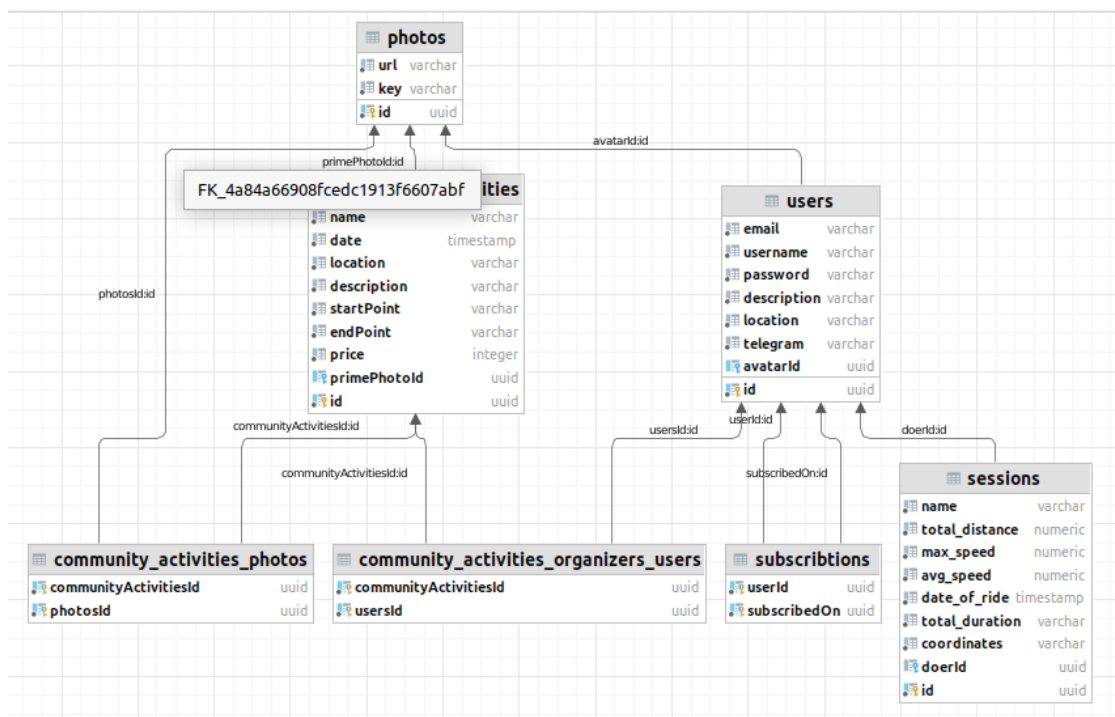


Рис. 3.4 Схема таблиц бази даних

### 3.3.2 Створення компонентів React Native

Під час розробки React Native додатку використовують React.js компоненти. Вони бувають класовими та функціональними. Сучасний підхід до створення React компонентів рекомендує використовувати саме функціональні компоненти, тож саме

такі я використовував в своєму застосунку. Під час розробки Я поділив компоненти на три типи: навігаційні, екрани, та презентаційні компоненти.

Навігаційна компонента відповідає за навігацію всередині застосунку, вони можуть бути вкладеними задля забезпечення необхідного типу навігації. Такі компоненти як правило не взаємодіють зі станом додатку. Екрани – компоненти на які перенаправляє користувача навігаційна компонента. Зазвичай в цих компонентах отримуються всі необхідні дані зі стану додатку та передаються до презентаційних компонентів якщо такі наявні. Презентаційні компоненти не комунікують зі станом та лише відображають дані передані з екранних компонент.

Навігаційними компонентами мого додатку є:

- AppNavigator – кореневий навігаційний компонент додатку. Він розділяє навігацію доступну користувачу, що увійшов в систему та анонімному користувачу.
- AuthNavigator – навігаційний компонент що обробляє переходи між сторінками входу та реєстрації в додатку.
- BTabNavigator – навігаційний компонент, що обробляє взаємодію з навігаційними компонентами в нижній частині екрану.
- ProfileNavigator – навігаційний компонент, що обробляє перехід на сторінки елементів.
- RideNavigator – навігаційний компонент, що обробляє переходи під час активної сесії.
- TTabNavigator – навігаційний компонент, що обробляє взаємодію з навігаційними компонентами в верхній частині екрану.

До екранних компонентів належать такі компоненти:

- CommunityActivitiesScreen – екран відображення списку подій спільноти.
- CommunityActivitiesEditScreen – екран редагування події.
- CommunityActivityScreen – екран відображення деталей події.
- FinishRideScreen – екран закінченої сесії.



- LoginScreen – екран входу в систему застосунку.
- NetworkProfileWrapperScreen – екран обгортка для відображення профілю користувача спільноти.
- NetworkScreen – екран відображення списку учасників спільноти.
- NetworkSessionsScreen – екран відображення сесій користувачів на яких підписаний користувач.
- RegistrationScreen – екран реєстрації нового користувача в системі.
- RideScreen - екран контейнер для відображення інформації та карти під час сесії.
- SessionOverviewScreen – екран перегляду сесії.
- SplashScreen – початковий екран який бачить не авторизований користувач.
- StartRideScreen – екран початку сесії.
- SubscribedOnScreen – екран відображає список користувачів на яких підписаний користувач.
- SubscribersScreen – екран відображає підписників користувача.
- UserProfileWrapperScreen – екран обгортка для відображення профілю користувача спільноти.
- UserSessionsScreen – екран відображення сесій користувача.

До презентаційних компонентів належать такі компоненти:

- Profile – компонент що відображає основну інформацію про користувача.
- ProfileEdit – компонент редагування інформації про користувача.
- Session – компонент відображення інформації про сесію.
- SessionTile – компонент – елемент списку відображення сесій.
- Spinner – компонент що відображається під час завантаження даних.
- CommunityActivityTile – компонент – елемент списку відображення подій.
- HeaderProfilePicture – компонент відображення аватару користувача.
- NetworkUser – компонент відображення користувача спільноти.
- RideInfoPage – компонент відображення інформації про сесію.
- Ridemap – компонент відображення карти під час сесії.

- `SubscribedOnUser` - компонент – елемент списку відображення користувачів на яких підписаний користувач.
- `SubscriberUser` - компонент – елемент списку відображення підписників.

Основним компонентом є `App` він вміщує в собі під'єднання стану застосунку.

### 3.3.3 Розробка серверної частини

Серверна частину додатку розроблена за допомогою фреймворку `Nest.js` та з використанням мови `TypeScript`. Під час розробки додаток був розділений на модулі. `AuthModule` – це модуль, що реалізовує в собі безпекову складову застосунку. Він містить логіку обробки заголовків запитів, виокремлення JWT токенів, та перевірки валідності токенів. Також модуль містить в собі гарди та стратегії, що дозволяють запускати захисні механізми на конкретних ендпоінтах.

`SharedModule` – модуль містить в собі сервіси, що використовують в декількох модулях одночасно.

`CommunityActivityModule` – модуль використовується для обробки запитів пов'язаних з подіями в спільноті. Модуль включає модель даних події для бази даних, сервіс та контролер.

`PhotoModule` – модуль використовується для обробки запитів зі збереження завантаження, та видалення фотографій. Модуль включає модель даних фотографії для бази даних, сервіс та контролер.

`SessionModule` – модуль використовується для обробки запитів пов'язаних з сесіями. Модуль включає модель даних сесії для бази даних, сервіс та контролер.

`UserModule` – модуль використовується для обробки запитів пов'язаних з користувачами. Модуль включає модель даних користувача для бази даних, сервіс та контролер.

Під час розробки серверної частина я намагався використовувати послуги хмарних провайдерів, для полегшення майбутньої міграції в хмару. У рамках цих дій я реалізував збереження фотографій в Amazon S3.

### 3.4 Опис інтерфейсу застосунку

Після запуску додатку користувачу відображається початковий екран де він може обрати вхід до системи або реєстрацію. У разі наявного токена, цей екран автоматично буде пропущений.

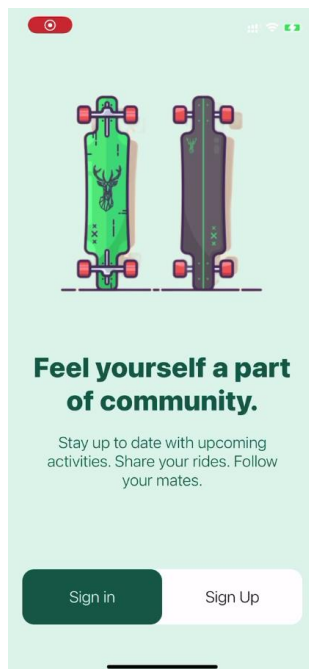
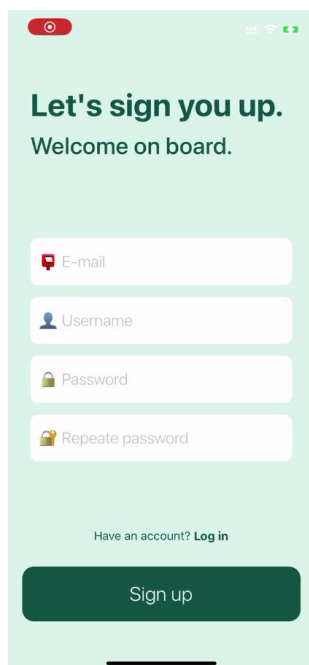


Рис. 3.5 Користувацький інтерфейс: початковий екран

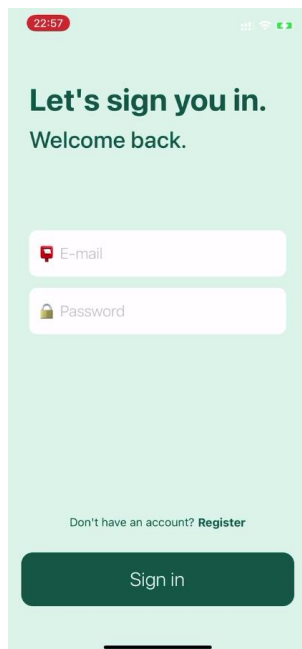
Для переходу на екран реєстрації потрібно натиснути кнопку “Sign UP”.



The image shows a mobile application interface for user registration. At the top, there is a status bar with a red circle icon, signal strength, Wi-Fi, and battery indicators. The main heading is "Let's sign you up." in a bold, dark green font, followed by the subtitle "Welcome on board." in a smaller, dark green font. Below this, there are four input fields: "E-mail" with an envelope icon, "Username" with a person icon, "Password" with a lock icon, and "Repeate password" with a lock icon. Below the input fields, there is a link "Have an account? Log in" in a small, dark green font. At the bottom, there is a large, dark green button with the text "Sign up" in white.

Рис. 3.6 Користувацький інтерфейс: реєстрація нового користувача

Заповнивши всі необхідні поля, користувач повинен натиснути кнопку “Sign up” для завершення реєстрації. У разі наявності акаунту користувач повинен натиснути на напис Log in, для переходу на екран входу в систему.



The image shows a mobile application interface for user login. At the top, there is a status bar with a red circle icon, signal strength, Wi-Fi, and battery indicators. The main heading is "Let's sign you in." in a bold, dark green font, followed by the subtitle "Welcome back." in a smaller, dark green font. Below this, there are two input fields: "E-mail" with an envelope icon and "Password" with a lock icon. Below the input fields, there is a link "Don't have an account? Register" in a small, dark green font. At the bottom, there is a large, dark green button with the text "Sign in" in white.

Рис. 3.7 Користувацький інтерфейс: реєстрація нового користувача

Заповнивши всі необхідні поля, користувач повинен натиснути кнопку “Sign in” для завершення реєстрації.

Після реєстрації або входу користувач потрапляє на екран початку нової сесії.



Рис. 3.8 Користувацький інтерфейс: екран початку нової сесії

Для початку нової сесії необхідно натиснути кнопку “Start session”, після цього користувача перенаправить на екран активної сесії.

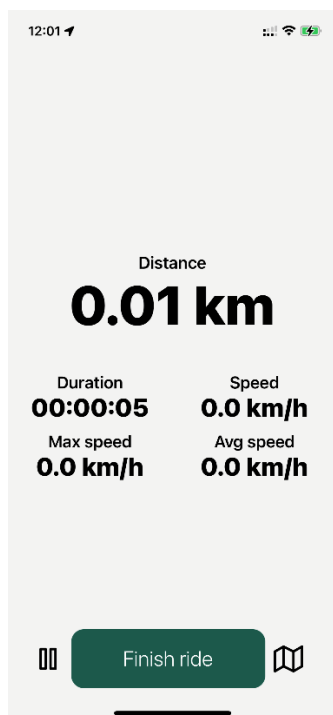


Рис. 3.9 Користувацький інтерфейс: екран активної сесії

За допомогою кнопок в нижній частині екрану користувач може призупинити сесію, закінчити сесію або відкрити екран з мапою сесії де графічно відображається пройдений маршрут.

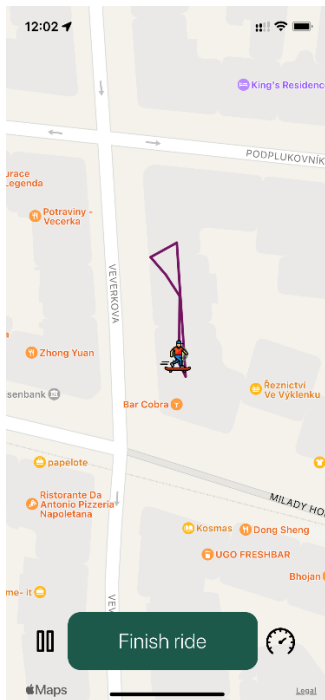


Рис. 3.10 Користувачський інтерфейс: мапа сесії

Для переходу назад на екран з інформацією необхідно натиснути на іконку що позначає спідометр.

Для завершення сесії необхідно натиснути на кнопку “Finish ride”, це перенаправить користувача на екран завершенної сесії.

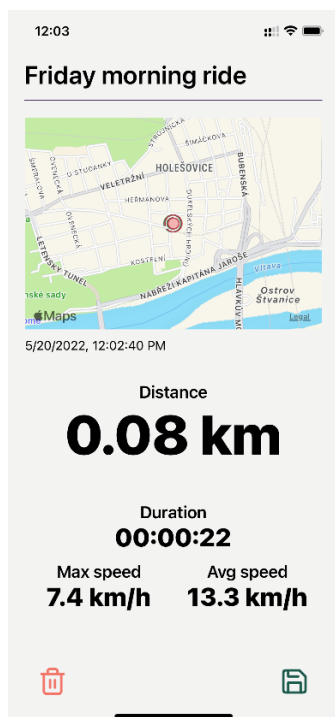


Рис. 3.11 Користувацький інтерфейс: екран завершеної сесії

На цьому екрані доступне збереження та видалення завершеної сесії, після будь-якої з цих операцій користувача перенаправить на екран початку нової сесії.

Щоб перейти на екран з минулими сесіями потрібно натиснути на кнопку “sessions” на нижній смужці елементів керування.



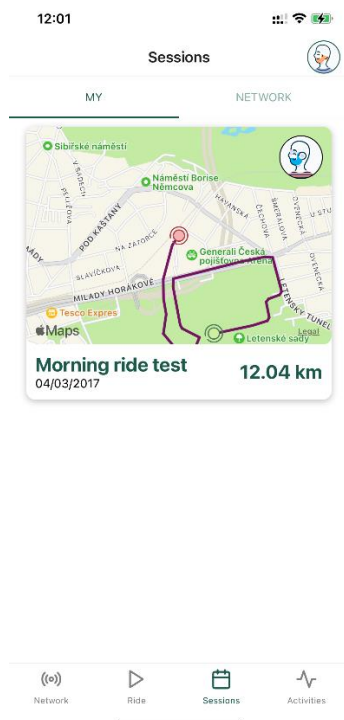


Рис. 3.12 Користувачський інтерфейс: екран минулих сесій

Натиснувши на потрібну сесію можна переглянути її деталі.

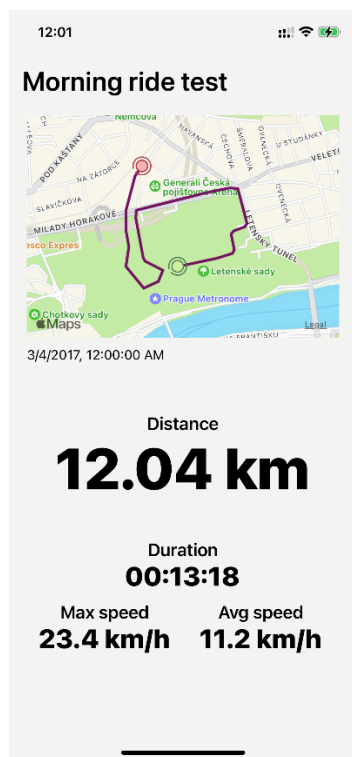


Рис. 3.13 Користувачський інтерфейс: сторінка деталей сесії

Щоб переглянути учасників спільноти потрібно натиснути на кнопку “network” на нижній смужці елементів керування.

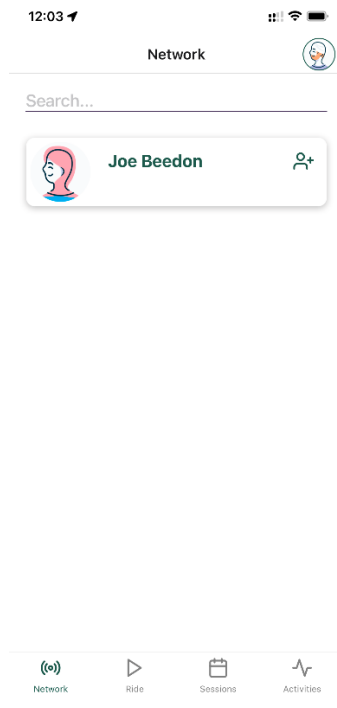


Рис. 3.14 Користувачський інтерфейс: екран спільноти

Користувачам доступний текстовий пошук серед спільноти. Також є можливість підписатися на певного учасника натиснувши відповідну іконку на елементі учасника. Щоб переглянути події потрібно натиснути на кнопку “activities” на нижній смужці елементів керування.

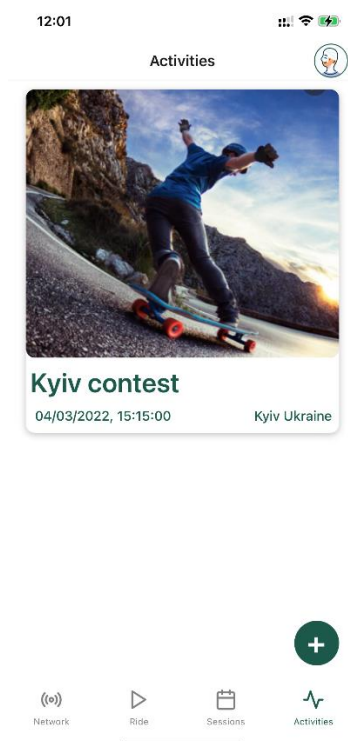


Рис. 3.15 Користувацький інтерфейс: перегляд подій

Щоб додати нову подію потрібно натиснути на кнопку “+” в правому нижньому кутку екрану.

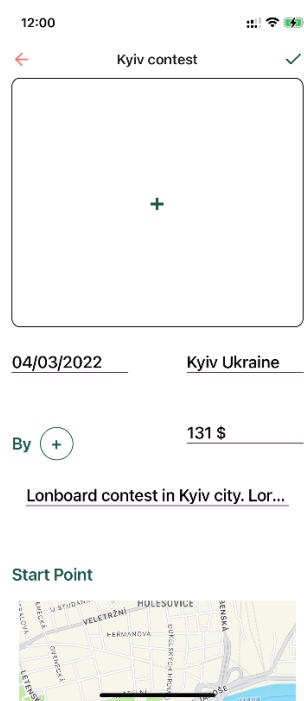


Рис. 3.16 Користувацький інтерфейс: додавання нової події

Для перегляду деталей події, потрібно натиснути на її елемент.

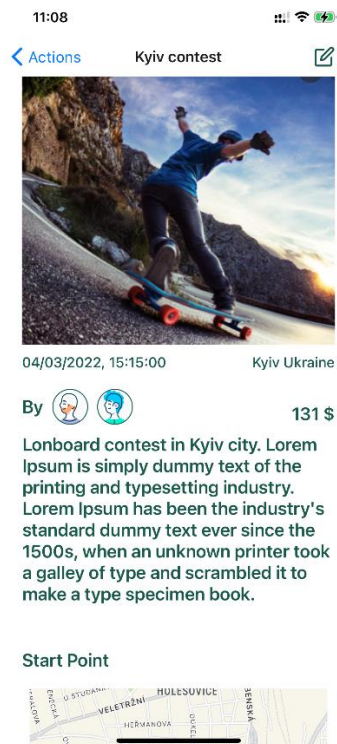


Рис. 3.17 Користувацький інтерфейс: додавання нової події

Для редагування власної події потрібно натиснути на іконку редагування в правому верхньому кутку.

Екран профілю містить інформацію про користувача: нікнейм в телеграмі, опис, та локацію.

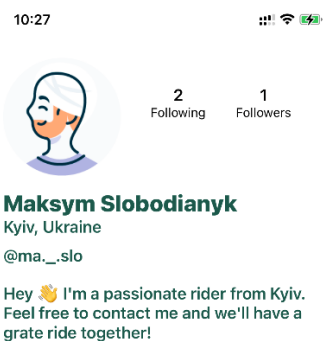


Рис. 3.18 Користувацький інтерфейс: профіль користувача

На цьому екрані користувач має можливість вийти з свого акаунту натиснувши на кнопку “Log out” та переглянути підписників та свої підписки.

## Висновки

По закінченню роботи, було виконано всі поставлені задачі. Провівши опитування серед лонгбордичної спільноти вдалося визначити групи респондентів з яких складається спільнота, знайти аналогічні додатки, що користуються популярністю, а також зібрати побажання щодо бажаного функціоналу майбутнього додатку. Проаналізувавши аналогічні застосунку було знайдено найнеобхідніший функціонал, який повинен містити додаток. На базі цього аналізу було сформовано функціональні вимоги до додатку, а також прописані групи користувачів. Також були досліджені та описані теоретичні відомості, що є пов'язаними з технологіями використаними під час розробки застосунку

За результатами роботи було створено додаток для відстеження активності на лонгборді. Розроблений застосунок має запланований функціонал та здатний відстежувати активність надаючи користувачам важливу інформацію в режимі реального часу та графічно відображати пройдений маршрут.

## Список використаної літератури

1. Публікація Lun.ua щодо попиту на екологічний персональний транспорт [Електронний ресурс]  
<https://misto.lun.ua/bikes-electric-scooters>
2. Сайт статистичних даних [Електронний ресурс]  
<https://www.statista.com/statistics/191308/participants-in-skateboarding-in-the-us-since-2006/#:~:text=Number%20of%20participants%20in%20skateboarding%20in%20the%20U.S.%202006%2D2020&text=This%20statistic%20shows%20the%20number,year's%20total%20of%206.61%20million.>
3. Електронне опитування серед учасників спільноти [Електронний ресурс]  
<https://docs.google.com/spreadsheets/d/1QDdMd6c6Hx5MuDNPwZfHPUFJi9wxZ1m3RK-MNoFZUSo/edit?resourcekey#gid=880628392>
4. Сторінка програми Krak Session в App Store [Електронний ресурс]  
<https://apps.apple.com/us/app/krak-session/id1466105913>
5. Сторінка програми My Fitness Activity Tracker в App Store [Електронний ресурс]  
<https://apps.apple.com/ua/app/my-fitness-activity-tracker/id1578572612>
6. Сторінка програми Skateboard Tracking в App Store [Електронний ресурс]  
<https://apps.apple.com/ua/app/skateboard-tracking/id1034390536>
7. Стаття про різні типи надання послуг хмарними провайдерами [Електронний ресурс]  
<https://dev.to/cloudtech/iaas-vs-paas-vs-saas-41d2>
8. Сайт для розшифрування JWT токена [Електронний ресурс]  
<https://jwt.io/>
9. Офіційна документація бібліотеки TypeORM [Електронний ресурс]  
<https://typeorm.io/>

- 10.Офіційна документація бібліотеки React Native [Електронний ресурс]  
<https://reactnative.dev/>
- 11.Офіційна документація фреймворку Flutter [Електронний ресурс]  
[https://flutter.dev/?gclid=Cj0KCQjw-JyUBhCuARIsANUqQ\\_LGs69glUCJLiFI7simCuFulb3r\\_8p3V4Dhag5PTAqGFqsWBA9y0VMaAtXjEALw\\_wcB&gclsrc=aw.ds](https://flutter.dev/?gclid=Cj0KCQjw-JyUBhCuARIsANUqQ_LGs69glUCJLiFI7simCuFulb3r_8p3V4Dhag5PTAqGFqsWBA9y0VMaAtXjEALw_wcB&gclsrc=aw.ds)
- 12.Офіційна документація фреймворку nestjs [Електронний ресурс]  
<https://nestjs.com/>
- 13.Офіційна документація PostgreSQL [Електронний ресурс]  
<https://www.postgresql.org/docs/>
- 14.Офіційна документація Redux [Електронний ресурс]  
<https://redux.js.org/tutorials/essentials/part-1-overview-concepts>
- 15.Офіційна документація Redux DevTools [Електронний ресурс]  
<https://github.com/reduxjs/redux-devtools#documentation>