Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

МОБІЛЬНИЙ ЗАСТОСУНОК МУЗИЧНОЇ АНАЛІТИЧНОЇ СИСТЕМИ ДЛЯ ЗАНЯТЬ БІГОМ

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки» 122

Керівник магістр комп'ютерних
наук, асистент Бабич Т.А.

_____

(підпис)

«____» _____ 2022 р.

Виконав студент 4 р.н. БП
«Комп'ютерні науки»
Галайда М. В.

«____» _____ 2022 р.

Міністерство освіти і науки України

Київ 2022

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Керівник магістр комп'ютерних
наук, асистент Бабич Т.А.
_____
(підпис)
«____» _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту 4-го курсу, факультету інформатики
Галайді Михайлу Вікторовичу

ТЕМА Мобільний застосунок музичної рекомендаційної системи для занять бігом

Зміст ТЧ до курсової роботи:

Зміст
Анотація
Вступ
1 Рекомендаційні та аналітичні музичні системи для спортивного застосування
2 Реалізація та огляд власної аналітичної музичної системи
Висновки
Література
Додатки

Дата видачі «___» _____ 2022 р.
Керівник _____ (підпис)
Завдання отримав _____ (підпис)

Тема: Мобільний застосунок музичної аналітичної системи для занять бігом

Календарний план роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1	Отримання завдання	17.10	
2	Огляд матеріалів на тему роботи	18.02	
3	Дослідження предметної області	15.03	
4	Реалізація власного застосунку	27.03	
5	Коригування роботи	19.04	
6	Остаточне оформлення роботи	20.05	
7	Захист роботи	27.05	

Студент Галайда М. В.

Керівник Бабич Т. А.

ЗМІСТ

Анотація 5

Вступ 6

РОЗДІЛ 1: Рекомендаційні та аналітичні музичні системи для спортивного застосування 7

1.1. Потреби у рекомендаційних та аналітичних музичних системах 7

1.2. Принципи роботи рекомендаційних музичних систем для бігу 9

1.2.1. Зчитування показників руху зі смартфону та обрахування BPM 9

1.2.2. Підбір музичних композицій з допомогою BPM 11

1.3. Наявні на ринку рекомендаційні музичні системи 12

1.4. Висновки за розділом 1 14

РОЗДІЛ 2: Реалізація та огляд власної аналітичної музичної системи 15

2.1. Опис формату запропонованого рішення 15

2.2. Технології та інструменти, обрані для реалізації 15

2.3. Архітектура та реалізація застосунку 16

2.4. Демонстрація роботи застосунку 19

Висновки 22

Література 23

Додаток А 24

Анотація

Метою роботи є дослідити можливості інтеграції музичних рекомендаціях в мобільні додатки для занять спортом, а саме бігом, на основі отриманих з мобільного пристрою даних про ритм руху користувача.

Ключові слова: рекомендаційна система, відслідковування бігу, музичні рекомендації.

Вступ

Музика та її вплив на рівень виконання фізичних вправ є вкрай досліджуваною та важливою темою у сьогоднішній науці. За мету цієї роботи було поставлено глибше розглянути цю тему та з використанням отриманих результатів запропонувати своє рішення проблеми.

Було досліджено сучасні підходи до вирішення таких питань та розглянуто наявні на ринку рішення.

З урахування результатів дослідження було розроблено власний застосунок що пропонує базовий аналіз впливу музики на заняття бігом.

РОЗДІЛ 1: Рекомендаційні та аналітичні музичні системи для спортивного застосування

1.1. Потреби у рекомендаційних та аналітичних музичних системах

Перед оглядом принципів роботи та наявних реалізацій музичних рекомендаційних систем важливо розуміти, якою є потреба у таких застосунках та чим вона зумовлена.

Кореляція між прослуховуванням музики під час спортивних занять та результатами цих занять багаторазово досліджувалася науковцями відповідних сфер. Наприклад, дослідник Метью Сторк з Університету Британської Колумбії у своєму дослідженні дійшов до висновку, що обрана по вподобаннях користувачем музика впливає як на рівень виконання вправ та витривалість, так і на рівень задоволення, отримуваний людиною в процесі[1]. Інша робота 2020-го року дослідника Нідхала Джебаблі, сфокусована на впливі музики на швидкості бігу, показала, що музика здатна значно підняти середню швидкість бігу, при цьому на міняючи бігових патернів[2].

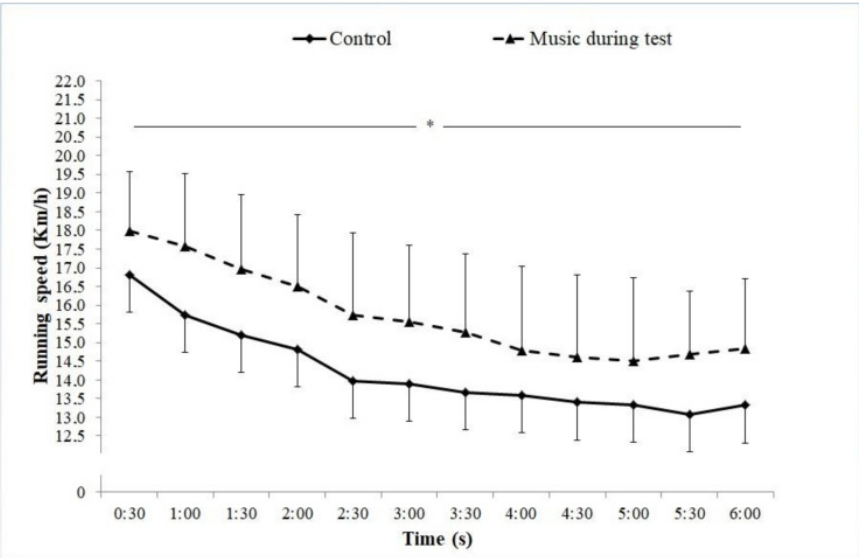

Рисунок 1.1 – порівняння швидкості бігу з та без музики у роботі Н. Джебаблі

Загалом, тема є досить популярною для дослідження – наприклад, на порталі Національної Бібліотеки Медицини США зазначено понад 700 робіт присвячених впливу музики на виконання спортивних вправ [3].

Спираючись на результати окремих досліджень, можна робити висновок, що позитивний вплив музики на виконання фізичних навантажень існує, а, враховуючи кількість дослідницьких матеріалів та консенсус їхніх результатів, достовірність цих результатів є обґрунтованою.

Виходячи з того, що такий зв’язок існує, у своїй роботі я вирішив працювати з двома напрямками:

1. Рекомендація музики на основі виконання фізичних вправ
2. Оцінка впливу окремих музичних композицій на виконання фізичних вправ

Тому, далі у цьому розділі я розгляну принципи роботи сучасних рекомендаційних та аналітичних музичних систем, що працюють з біговими вправами. Після цього я проаналізую наявні на ринку застосунки, що реалізують в собі ці принципи.

1.2. Принципи роботи рекомендаційних музичних систем для бігу

Завдяки природі бігу, створення музичних рекомендацій для бігових вправ з технічної точки зору є досить прямолінійним та незатратним в реалізації. Складається цей процес з таких кроків:

1. Зчитування показників руху з пристрою, обрахування темпу
2. За потреби, переведення одиниць виміру темпу в BPM (Beats Per Minute)
3. Підбір музичного каталогу з допомогою отриманих показників BPM.

Перед розбором кожного з кроків наголошу, що надалі у своїй роботі я досліджуватиму тему в контексті мобільних застосунків, оскільки смартфони є зручними для використання під час бігу та надають достатні можливості для зчитування даних про фізичні вправи, а саме про біг.

1.2.1. Зчитування показників руху зі смартфону та обрахування BPM

Сучасні смартфони є досить потужними комп’ютерами, що дає можливість наповнювати їх великою кількістю функціоналу. Зокрема, найбільш релевантним у нашому випадку інструментом є педометр, що присутній у більшості сучасних смартфонів.

Педометр – це електронний або електромеханічний пристрій, що визначає рух у просторі. Зазвичай бігунами педометр використовується прикріплений до ноги, або ж тримається у руці. За умови коректного використання пристрій чітко вираховує кожен зроблений крок протягом

певного часового періоду – електронний варіант за своєю суттю не відрізняється та працює так само.

Окрім педометра, у смартфонах часто присутні й інші, більш низькорівневі інструменти, як-от детектори руху мобільного пристрою по 3-ох осях.

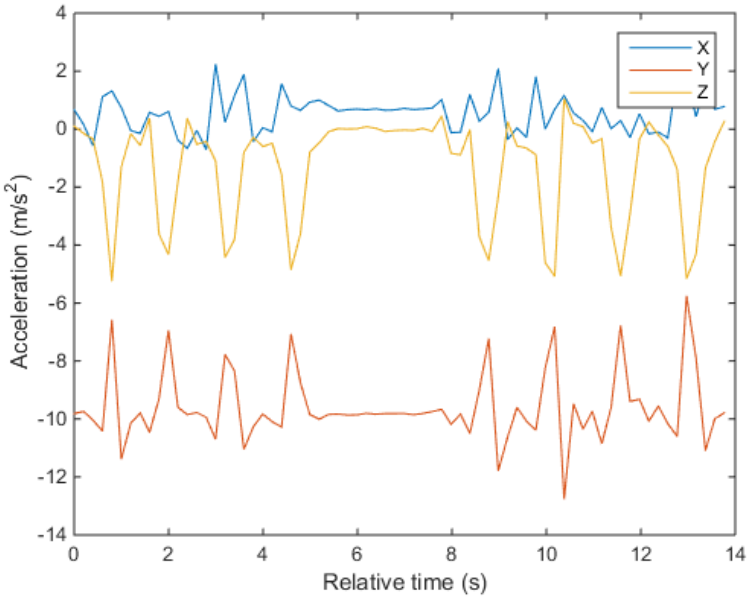

Рисунок 1.2 – вхідні дані з сенсорів руху під час ходьби

На рисунку 1.2 зображений приклад даних, що отримується із сенсорів руху під час ходьби або бігу. Для подальшого використання в нашому контексті такі дані потрібно додатково обробляти: фільтрувати часткові випадки, знаходити середній коефіцієнт між 3-ма осями, тощо. Такий підхід до зчитування даних бігу надає більшу гнучкість та можливість «кастомізації», але, часто педометр є більш оптимальним рішенням, оскільки його використання вимагає мінімальних подальших обробок даних.

Перед використанням зчитаних даних для музичних рекомендацій, фінальним кроком є переведення результатів у формат BPM, тобто збереження кількості кроків не у межах довільного часового проміжку, а у

розрізі однієї хвилини. Цей пункт є важливим, оскільки BPM є фундаментальним поняттям у теорії музики, і саме цей показник використовується для позначення ритму і, відповідно, побудови рекомендаційних музичних систем.

1.2.2. Підбір музичних композицій з допомогою BPM

Маючи на руках показник BPM нашого бігу, наступним кроком є знайти музичні каталоги, які підпадають під такі вимоги:

1. Зберігають «меппінг» окремих композицій (треків) до їхнього BPM
2. Є достатньо обширними, містять в собі великий вибір треків (аби використання каталогу було виправданим та ефективним)

З урахуванням наведених вимог, найбільш розумним вибором є використання великих стрімінгових музичних сервісів, як-от: Spotify, Apple Music, Deezer, Youtube Music, тощо. Такі сервіси є дуже поширеними та використовуються мільйонами користувачів і в той же час мають достатні технічні потужності аби зберігати метадані музичних композицій (зокрема й ритм або BPM), а не лише аудіо.

Однак, варто зазначити, що не завжди музичні стрімінгові сервіси тримають такі метадані у відкритому доступі – так наприклад, Spotify та Apple Music не надають інформації про BPM треків зі своєї бібліотеки.

1.3. Наявні на ринку рекомендаційні музичні системи

Маючи уявлення того, як працюють рекомендаційні системи «під капотом», пропоную оглянути сучасні реалізації таких систем, доступні на ринку.

Одним з найбільших застосунків такого типу на ринку мобільних додатків є **RockMyRun** – музичний фітнес-додаток, що надає широкий вибір треків, підібраних вручну та дає можливість користувачам записувати свої пробіжки у зв'язці з прослуховуваною музикою. Окрім цього, застосунок дає можливість автоматично підбирати треки для тренуванні на основі зчитуваного зі смартфону або смарт-годинника BPM.

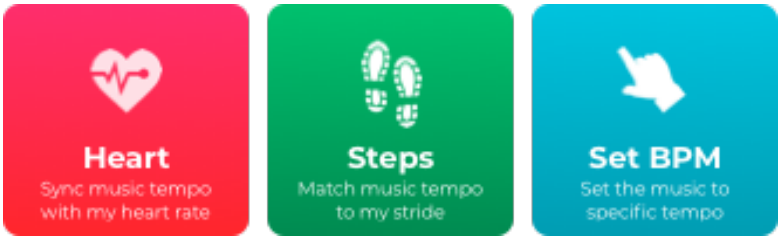

Рисунок 1.3 – функціонал застосунку RockMyRun

Додаток пропонує два автоматичні режими підбору музики: за ритмом кроків та за ритмом биття серця.

Ще один популярний застосунок **Weav Run** надає аналогічний функціонал: вручну або автоматично заданий BPM. На відміну від попереднього згаданого додатку, Weav Run має лише один автоматичний режим. Натомість, WeavRun надає більшу кількість статистики та інформації по вправах користувача, зокрема по середньому значенню BPM, швидкості, дистанціях, тощо.

Рисунок 1.4 – інтерфейс застосунку Weav Run

З більш промовистих імен в контексті рекомендаційних музичних систем варто також згадати **Spotify**. Хоч за своєю суттю Spotify – це звичайний стрімінговий сервіс, не орієнтований на інтеграцію з фітнес-сервісами, довгий час у мобільному застосунку Spotify була доступна функція “Running Mode”, що діяла за принципом попередніх згаданих додатків – мобільний пристрій зчитує кількість кроків на хвилину з допомогою педометра й надає користувачеві музичні рекомендації на основі цих даних.

Великою перевагою цього рішення перед попередніми є те, що воно поєднує в собі функціонал визначення BPM та широкий каталог музичних композицій, до якого інші застосунки не мають настільки відкритого доступу.

1.4. Висновки за розділом 1

У даному розділі було розглянуто питання актуальності теми кореляції музики та результатів виконання фізичних вправ, зокрема бігових.

Розглянуто можливі варіанти використання цих зв’язків на практиці та описано принципи, за якими ці зв’язки використовуються.

На практиці проведено огляд наявних на ринку застосунків, що реалізовують згадані принципи, їх переваги та особливості.

Отримані спостереження буде використано у наступному розділі для проектування та реалізації власного застосунку для аналізу впливу музичних композицій на показники бігу.

РОЗДІЛ 2: Реалізація та огляд власної аналітичної музичної системи

2.1. Опис формату запропонованого рішення

Оскільки на ринку наразі існує досить багато мобільних додатків для зчитування даних про темп бігу та автоматичного підбору музичних композицій на основі цих даних, я зробив висновок, що аналоги таких застосунків на ринку наразі не затребувані.

Натомість, я вирішив зосередитися на іншому напрямку, який я згадував раніше у своїй роботі – фіксування впливу музики на результативні показники бігу користувача.

Як музичний каталог я обрав сервіс Spotify, адже він надає дуже широкий вибір музичних композицій та пропонує у відкритому доступі веб-API, що дозволяє великою мірою контролювати музичний програвач свого акаунту, отримувати інформацію про окремі композиції, виконавців, тощо.

У реалізованому застосунку користувач авторизується з допомогою свого облікового запису у Spotify. Всередині застосунку, користувач має доступ до сторінки з мапою, на якій він також може в режимі реального часу записувати свою пробіжку, пройдену дистанцію, час, швидкість, тощо. При закінченні вправ, додаток надсилає зчитані дані про біг та дані про прослухану музику на сервер, після чого проводиться обрахунок середніх показників виконання фізичних навантажень в межах цілого заняття або окремих музичних композицій.

2.2. Технології та інструменти, обрані для реалізації

Для імплементації мобільного додатку було використано такі технології:

- **React Native** – для ефективної організації застосунку, його швидкої розробки та розгортання.
- **React Redux** – бібліотека для ефективної роботи зі станами, була використана зокрема для ефективного збереження токенів авторизації.
- **React Navigation** – бібліотека для навігації між React-компонентами.
- **Spotify Web Api** – для зручної роботи з широким каталогом музичних композицій.
- **Firebase** – хмарний бек-енд сервіс, був використаний для збереження даних та обробки авторизації.

2.3. Архітектура та реалізація застосунку

Як уже було згадано вище, у застосунку я використав Spotify для доступу до музичного каталогу, і, відповідно, використав авторизацію цього стримінгового сервісу у своєму застосунку.

Для використання Spotify в особистих розробницьких цілях необхідно спершу зареєструвати власний додаток та створити дешборд в сервісі “Spotify for Developers”.

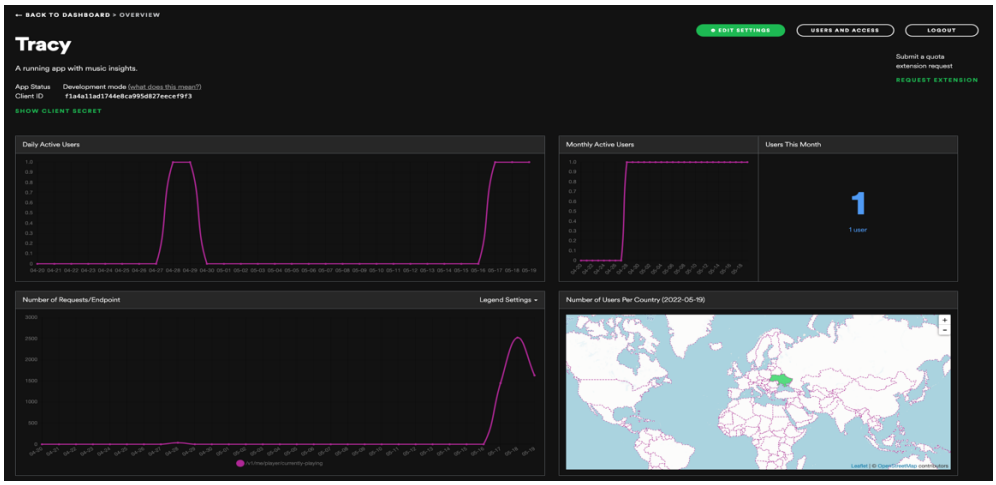

Рисунок 2.1 – розробницький дешборд Spotify

Після цього сервіс надає розробнику доступ до набору секретних кодів, які використовуються для організації авторизації всередині застосунку.

В своїй системі авторизації Spotify використовує принципи OAuth2, а саме підхід “Authorization Code Flow”. Цей підхід складається з кількох кроків:

1. Отримання коду авторизації
2. Використання коду авторизації для отримання токену доступу зі строком придатності 1 год.
3. Паралельно з токеном доступу користувач також отримує токен оновлення, що дозволяє за потреби моментально оновлювати токен доступу

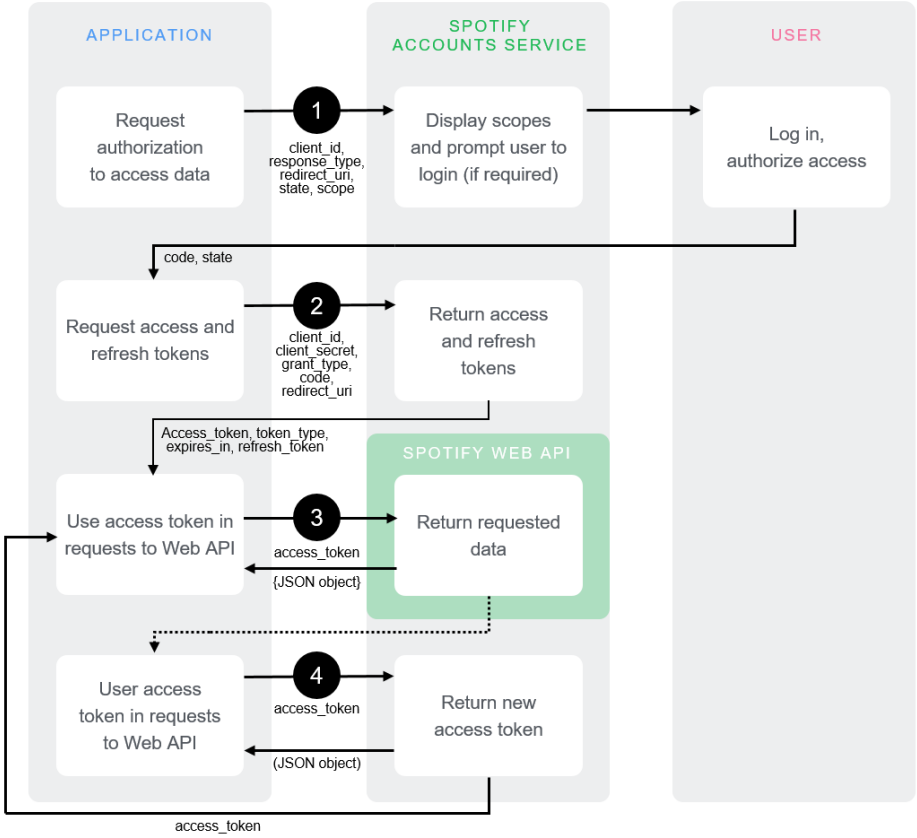

Рисунок 2.2 – Authorization Code Flow в сервісі Spotify

Для реалізації цього процесу я використав сервіс Firebase, а саме його функціонал хмарних функцій (*повний код наведено у додатку А*).

На клієнтській частині застосунку робляться запити на ці функції авторизації, після чого токени зберігаються та використовуються далі у застосунку.

В подальшій роботі додатку Spotify Web Api використовується для отримання даних про поточну музичну композицію – для цього на певні точки API надсилається відповідний запит:

```
async function loadCurrSong() {
  try {
    const res = await fetch("https://api.spotify.com/v1/me/player/currently-playing", {
      method: "GET",
      headers: {
        Authorization: `Bearer ${accessToken}`,
        "Content-Type": "application/json",
      },
    });

    const parsedRes = await res.json();
    setCurrSong(parsedRes.item);
  } catch (error) {
    console.log(error);
    setCurrSong({});
  }
}
```

Рисунок 2.3 – запит для отримання інформації про поточний трек

Отримані дані зберігаються в локальний стан, після чого використовуються для відображення користувачеві.

Паралельно з відслідковуванням музичних треків з програвача, додаток також моніторить локацію користувача з допомогою Expo Location API – цей інструментарій дозволяє отримувати дані про локацію мобільного пристрою в реальному часі та з високою точністю.

```
locSub = await Location.watchPositionAsync({ accuracy: Location.Accuracy.Highest, distanceInterval: 5 }, (loc) => {
  let newCoord = {
    latitude: loc.coords.latitude,
    longitude: loc.coords.longitude,
  };
  setCurrCoord(newCoord);
  setCoordList((prevList) => {
    return prevList.concat(newCoord);
  });
});
setLocationSub(locSub);
```

Рисунок 2.4 – отримання даних про поточну локацію користувача

Отримані дані про геолокацію в подальшому використовуються для вирахування пройденої відстані, швидкості та середнього темпу.

2.4. Демонстрація роботи застосунку

При вході в застосунок користувач бачить проміжний екран завантаження. В цей час відбувається перевірка на наявність токену доступу у локальному сховищі – якщо він знаходиться, додаток завантажує його та пропускає користувача на наступний екран. Якщо ні, користувачу дається можливість авторизуватися.

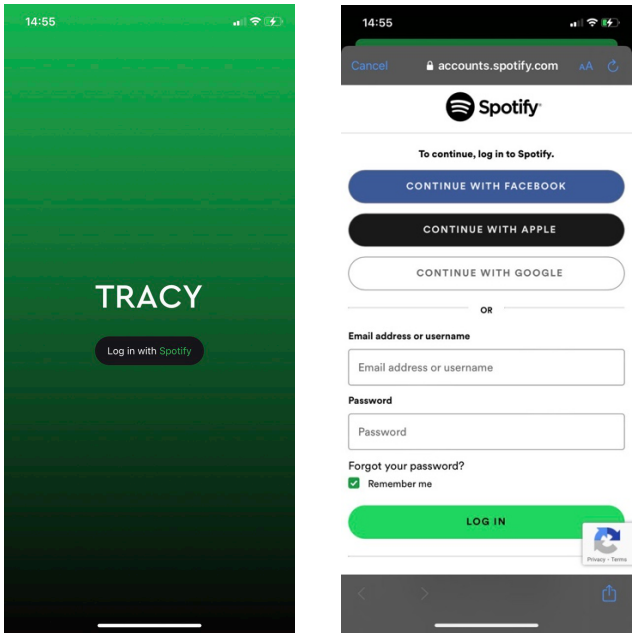

Рисунок 2.5 – авторизація в застосунку

Після авторизації користувач отримує доступ до екрану з картою. На цьому екрані відображається поточний музичний трек з програвача Spotify, пройдена дистанція та витрачений час. Коли користувач натискає кнопку «Start», локація і дані про біг починають записуватися.

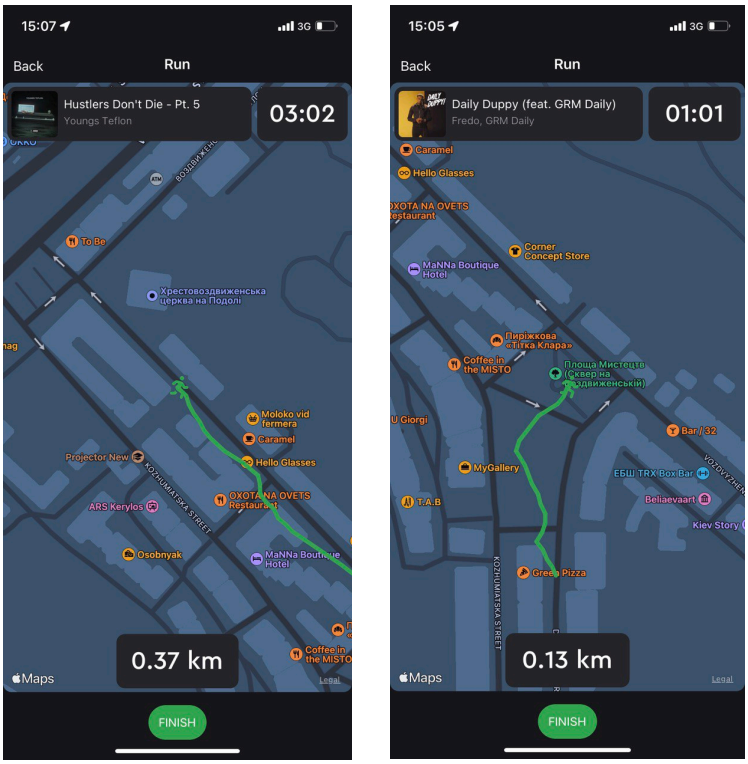

Рисунок 2.6 – запис бігу в застосунку

Коли користувач закінчує біг, він натискає кнопку «Finish» – в результаті, дані про активність зберігаються, надсилаються на сервер та стають доступні на сусідній сторінці «History».

На цій сторінці користувач може переглянути усі свої попередні активності з інформацією про швидкість, дистанцію, час, тощо. Ці дані доступні у розрізі пробіжки в цілому, а також в межах одного прослуханого музичного треку.

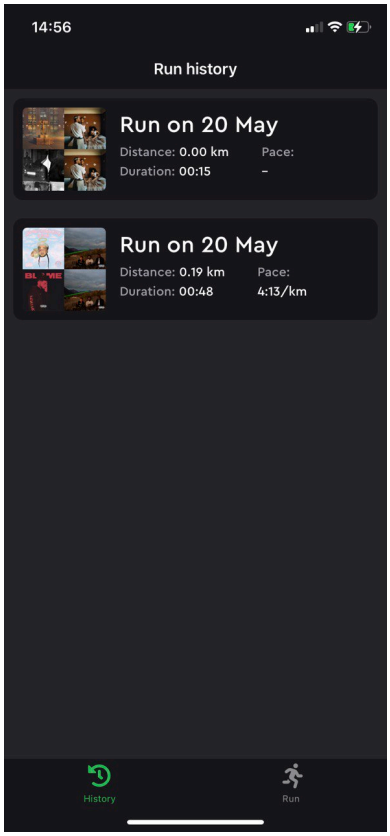

Рисунок 2.7 – історія активностей в застосунку

Висновки

Було проведено огляд сучасних методів музичної рекомендації та підходів до неї. Розглянуто методи зчитування релевантних для такої рекомендації показників із мобільних пристроїв.

З допомогою спостережень та досліджень, отриманих у роботі, було розроблено застосунок, що дозволяє відслідковувати дані про свою фізичну активність, накладаючи їх на інформацію про прослуховування музики, що дозволяє аналізувати вплив музики на виконання фізичних вправ.

Література

1. Music enhances performance and perceived enjoyment of sprint interval exercise (May 2015). Matthew J Stork, Matthew Y W Kwan, Martin J Gibala, Kathleen A Martin Ginis [Електронний ресурс] https://pubmed.ncbi.nlm.nih.gov/25202850/
2. Listening to Preferred Music Improved Running Performance without Changing the Pacing Pattern during a 6 Minute Run Test with Young Male Adults (May 2020). Nidhal Jebabli, Urs Granacher, Mohamed Amin Selmi, Badriya Al-Haddabi, David G Behm, Anis Chaouachi, Radhouane Haj Sassi [Електронний ресурс] https://pubmed.ncbi.nlm.nih.gov/32403345/
3. National Library of Medicine [Електронний ресурс] https://pubmed.ncbi.nlm.nih.gov/?term=music+exercise+performance

Додаток А

Код авторизації

Функції авторизації в Firebase:

```
const functions = require("firebase-functions");
const SpotifyWebApi = require("spotify-web-api-node");
const constants = require("./constants");

const spotifyApi = new SpotifyWebApi({
  redirectUri: constants.redirectUri,
  clientId: constants.clientId,
  clientSecret: constants.clientSecret,
});

const getAuthUrl = functions.https.onRequest((req, res) => {
  const authUrl = spotifyApi.createAuthorizeURL(constants.scopes, constants.state);
  res.json({ url: authUrl });
});

const getAccessToken = functions.https.onRequest(async (req, res) => {
  const code = req.body.code;
  try {
    const grantData = await spotifyApi.authorizationCodeGrant(code);
    res.json({
      accessToken: grantData.body["access_token"],
      refreshToken: grantData.body["refresh_token"],
      expires_in: grantData.body["expires_in"],
    });
  } catch (err) {
    res.json(err);
  }
});

const refreshAccessToken = functions.https.onRequest(async (req, res) => {
  spotifyApi.setRefreshToken(req.body.refreshToken);
  const data = await spotifyApi.refreshAccessToken();
  res.send({
    accessToken: data.body["access_token"],
    expires_in: data.body["expires_in"],
  });
});

module.exports = {
  getAuthUrl,
  getAccessToken,
  refreshAccessToken,
};
```

Функції авторизації на клієнтській частині:

```
const SpotifyApi = {
  fetchAuthUrl: async function () {
    const res = await fetch("https://us-central1-tracy-mh.cloudfunctions.net/getAuthUrl");
    const jsonRes = await res.json();
    return jsonRes;
  },
  fetchAccessToken: async function (authCode) {
    const res = await fetch("https://us-central1-tracy-mh.cloudfunctions.net/getAccessToken", {
      method: "POST",
      body: JSON.stringify({
        code: authCode,
      }),
      headers: {
        "Content-Type": "application/json",
      },
    });
    const jsonRes = await res.json();
    return jsonRes;
  },
  fetchFreshAccessToken: async function (refreshToken) {
    const res = await fetch("https://us-central1-tracy-mh.cloudfunctions.net/refreshAccessToken", {
      method: "POST",
      body: JSON.stringify({
        refreshToken: refreshToken,
      }),
      headers: {
        "Content-Type": "application/json",
      },
    });
    const jsonRes = await res.json();
    return jsonRes;
  },
};

export default SpotifyApi;
```