Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

Бакалаврська дипломна робота на тему:

«МОБІЛЬНИЙ ДОДАТОК ПІДБОРУ МІСЦЬ ДЛЯ ВІДПОЧИНКУ»

Виконала: студентка 4 курсу, групи 1КН-20б
спеціальності 122 – Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)
Сабашна В. В.
(прізвище та ініціали)
Керівник: к. т. н., доцент
Арсенюк І. Р.
(прізвище та ініціали)
«07» червня 2024 р.
Рецензент: PhD, старший викладач
Войцеховська О. О.
(прізвище та ініціали)
«07» червня 2024 р.

Допущено до захисту
Зав. кафедри _____ Яровий А. А.
«07» червня 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
проф., д. т. н. Яровий А. А.
"07" 03 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТЦІ
Сабашній Вікторії Валеріївні

1. Тема роботи: Мобільний додаток підбору місць для відпочинку.
Керівник роботи: Арсенюк Ігор Ростиславович, доцент.
затверджені наказом вищого навчального закладу від "11" 03 2024 року №80

2.Термін подання студентом роботи 27. 05. 2024 р.

3. Вихідні дані до роботи:
- Кількість місць для відпочинку: не менше ніж 500;
- ОС Android 10.0 і вище;
- Розроблюваний додаток має підтримуватись на не менш ніж 16000 пристроях.
- Мова програмування – об'єктно-орієнтована;
- Крос платформне програмне забезпечення;
- Програмне забезпечення Postman;

4. Зміст текстової частини (перелік питань, які потрібно розробити):
- Обґрунтувати актуальність розробки мобільного додатку для підбору місць для відпочинку.
- Проаналізувати сучасні програмні засоби для підбору місць для відпочинку.
- Розробити загальний алгоритм роботи додатку, загальну структурну схему функціонування додатку та UML-діаграму класів.
- Розробити діаграму прецедентів додатку для підбору місць для відпочинку.
- Виконати обгрунтування вибору мови програмування.
- Розробити інтерфейс мобільного додатку.
- Реалізувати програмне забезпечення мобільного додатку та здійснити його тестування.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслен
   - Схема загального алгоритму роботи додатку,
   - Структурна схему функціонування додатку,
   - UML-діаграму класів додатку,
   - Схема життєвого циклу додатку,
   - Діаграма прецедентів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-3	Арсенюк І. Р., к. т. н., доцент		

7. Дата видачі завдання 07.03.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Приміт
		початок	закінчення	
1	Аналіз додатків аналогів для підбору місць для відпочинку	06.05.2024	08.05.2024	
2	Розробка алгоритму для підбору місць для відпочинку	10.05.2024	14.05.2024	
3	Розробка структури додатку для підбору місць для відпочинку	15.05.2024	19.05.2024	
4	Розробка складових додатку для підбору місць для відпочинку	20.05.2024	22.05.2024	
5	Аналіз функціонування додатку для підбору місць для відпочинку	22.05.2024	24.05.2024	
6	Оформлення додатків	27.05.2024	29.05.2024	
7	Розробка інструкції користувача	30.05.2024	03.06.2024	
8	Оформлення матеріалів до захисту БДР	04.06.2024	06.06.2024	

Студент _____ Сабашна В. В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Арсенюк І. Р.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 86 сторінок формату А4, на яких є 24 рисунки, список використаних джерел містить 20 найменувань.

Бакалаврська дипломна робота присвячена розробці мобільного додатку підбору місць для відпочинку.

Виконано аналіз предметної області підбору місць для відпочинку і доведено актуальність створення даного мобільного додатку.

Розроблено модуль отримання та оброблення інформації про геопозицію користувача на основі платформи Android, з метою пошуку найближчих місць для відпочинку. Також реалізовано додаткову функцію надання інформації про користувачів додатку, які знаходяться поруч відповідно до отриманих даних про геолокацію.

Розроблено зручний інтерфейс мобільного додатку, який дозволяє підвищити рівень комфортності роботи з додатком, а надає можливість отримання усієї потрібної інформації про обране місце для відпочинку.

Розроблено алгоритм роботи додатку, який передбачає реалізацію нових функціональних можливостей мобільного додатку підбору місць для відпочинку.

Ключові слова: місця для відпочинку, геолокація, React Native, JavaScript, мобільний додаток.

ABSTRACT

The bachelor's thesis consists of 86 pages of A4 format, on which there are 24 figures, the list of used sources contains 20 titles.

The bachelor's thesis is devoted to the development of a mobile application for the selection of places for rest.

The analysis of the subject area of the selection of places for recreation was performed and the relevance of the creation of this mobile application was proven.

A module for receiving and processing information about the user's geolocation based on the Android platform has been developed in order to find the nearest places for rest. An additional function of providing information about application users who are nearby according to the received geolocation data has also been implemented.

A convenient interface of the mobile application has been developed, which allows you to increase the comfort level of working with the application, and provides the opportunity to receive all the necessary information about the chosen place for rest.

The algorithm of the application has been developed, which provides for the implementation of new functionalities of the mobile application for selecting places for rest.

Keywords: vacation spots, geolocation, React Native, JavaScript, mobile application**.**

ЗМІСТ

ВСТУП

Актуальність досліджень. Сучасний етап світового розвитку характеризується стрімким зростанням темпів життя та збільшенням робочого навантаження на людей. Внаслідок цього, питання ефективного використання вільного часу та організації відпочинку набуває все більшої актуальності. Пошук оптимальних місць для відпочинку, які б відповідали індивідуальним потребам і уподобанням, часто є складним та часозатратним процесом.

ІТ-індустрія пропонує інноваційні рішення для автоматизації процесів та зменшення витрат часу на планування відпочинку. Одним з таких рішень є мобільні додатки, що дозволяють швидко та зручно знайти місця для відпочинку відповідно до заданих критеріїв. З розвитком міст і туристичної інфраструктури, зростає потреба у системах, що допомагають користувачам знаходити ідеальні місця для відпочинку, враховуючи їхні інтереси, бюджет та географічне положення.

Існуючі рішення, такі як туристичні портали та мобільні додатки, часто не повністю задовольняють потреби користувачів, оскільки можуть бути складними у використанні, не забезпечують персоналізованого підбору місць або надають застарілу інформацію. Це зумовлює необхідність створення нового мобільного додатку, який би інтегрував сучасні технології для надання актуальної та персоналізованої інформації про місця для відпочинку.

Метою дослідження є розширення функціональних можливостей додатку підбору місць для відпочинку.

Об'єктом дослідження є процес підбору місць для відпочинку.

Предметом дослідження є програмний додаток підбору місць для відпочинку.

Задачі дослідження:

1. Обґрунтувати актуальність розробки мобільного додатку для підбору місць для відпочинку.
2. Проаналізувати сучасні програмні засоби для підбору місць для відпочинку.

3. Розробити загальний алгоритм роботи додатку, загальну структурну схему функціонування додатку та UML-діаграму класів, діаграму прецедентів додатку для підбору місць для відпочинку.

4. Виконати обгрунтування вибору мови програмування.

5. Розробити інтерфейс мобільного додатку.

6. Реалізувати програмне забезпечення мобільного додатку та здійснити його тестування.

Апробація результатів роботи

Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)».

Публікації

За результатами бакалаврської дипломної роботи опубліковано тези доповіді на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)» (м. Вінниця, Україна) [1].

1 АНАЛІЗ СУЧАСНИХ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПІДБОРУ МІСЦЬ ДЛЯ ВІДПОЧИНКУ

1.1 Обґрунтування актуальності розробки додатку для підбору місць для відпочинку

У сучасному світі зростаюча кількість людей стикається зі значним робочим навантаженням, що підвищує важливість ефективного використання вільного часу та якісного відпочинку. Пошук оптимальних місць для відпочинку стає все більш актуальним, оскільки правильний вибір місця для релаксації може значно підвищити якість життя та сприяти відновленню сил.

Мобільні додатки для підбору місць для відпочинку можуть значно полегшити цей процес, надаючи користувачам зручні інструменти для швидкого пошуку і вибору місць, які відповідають їхнім уподобанням та вимогам. Проте, існуючі додатки часто мають обмежений функціонал або не забезпечують необхідного рівня персоналізації та актуальності інформації.

У великих містах та популярних туристичних регіонах одночасно працюють безліч закладів, включаючи парки, ресторани, музеї, пляжі та інші місця для відпочинку. Кожне з цих місць має свої особливості, години роботи, відгуки відвідувачів та інші важливі параметри, які можуть вплинути на вибір користувача. Відсутність централізованого та зручного джерела інформації ускладнює процес планування відпочинку, що може призвести до втрати часу та незадоволення.

Існують різні ресурси, такі як туристичні сайти, огляди в соціальних мережах, Google Maps та інші, які надають інформацію про місця для відпочинку. Проте, ці ресурси часто не інтегровані, що створює незручності для користувачів, які змушені витрачати багато часу на пошук та порівняння інформації з різних джерел.

Таким чином, актуальним є створення мобільного додатку, який об'єднає всю необхідну інформацію про місця для відпочинку в одному місці, забезпечуючи миттєвий доступ до актуальних даних, персоналізовані рекомендації та зручний

інтерфейс. Це дозволить користувачам економити час на пошуку та плануванні, підвищуючи якість відпочинку та загальний рівень задоволення.

Розробка такого додатку сприятиме не лише поліпшенню організації особистого часу користувачів, але й підвищенню їхньої обізнаності про різноманітні місця для відпочинку, що можуть бути неочевидними або маловідомими. Це, в свою чергу, може мати позитивний вплив на розвиток місцевої туристичної індустрії та економіки загалом.

1.2 Аналіз предметної області підбору місць для відпочинку

Вибір місця для відпочинку є важливою складовою сучасного життя, адже від якості відпочинку залежить загальний рівень задоволеності життям, здоров'я та продуктивність людей. Ринок послуг, пов'язаних з відпочинком, надзвичайно різноманітний і включає в себе безліч опцій, таких як парки, ресторани, музеї, пляжі, оздоровчі комплекси та інші рекреаційні заклади.

Однією з основних проблем сучасного ринку послуг для відпочинку є велика кількість варіантів та джерел інформації, що часто ускладнює процес вибору оптимального місця. Більшість людей стикається з проблемою відсутності централізованої та актуальної інформації, що призводить до неефективного використання часу та ресурсів. Основні джерела інформації про місця для відпочинку включають туристичні сайти, огляди в соціальних мережах, Google Maps та різноманітні тематичні додатки. Проте ці джерела часто не інтегровані між собою, що створює незручності для користувачів.

Централізація інформації про місця для відпочинку може значно спростити процес пошуку та вибору. Наприклад, мобільний додаток, який об'єднає інформацію з різних джерел, зможе запропонувати користувачам актуальні дані, персоналізовані рекомендації та зручний інтерфейс. Такий підхід дозволить зекономити час та ресурси, забезпечуючи більш ефективний та приємний процес планування відпочинку.

Існують різні ресурси, такі як туристичні сайти, огляди в соціальних мережах, Google Maps та інші, які надають інформацію про місця для відпочинку. Проте, ці ресурси часто не інтегровані, що створює незручності для користувачів, які змушені витрачати багато часу на пошук та порівняння інформації з різних джерел. Це ускладнює процес прийняття рішень і може призвести до незадоволення результатами відпочинку.

Мобільний додаток, який об'єднає всю необхідну інформацію в одному місці, зможе значно покращити користувацький досвід. Він має надавати миттєвий доступ до актуальних даних про місця для відпочинку, включаючи години роботи, відгуки відвідувачів, спеціальні пропозиції та інші важливі параметри. Крім того, додаток може пропонувати персоналізовані рекомендації на основі вподобань користувачів, що дозволить зробити вибір більш обґрунтованим і задовольнити індивідуальні потреби кожного користувача.

Ще одним важливим аспектом є інтеграція з іншими сервісами та платформами. Наприклад, можливість зберігати улюблені місця, ділитися ними з друзями та отримувати рекомендації на основі активності в соціальних мережах. Це забезпечить користувачам більш зручний і інтегрований досвід використання додатку.

Слід також враховувати питання актуальності та точності інформації. Багато ресурсів не забезпечують своєчасного оновлення даних, що може призвести до незадоволення користувачів. Наприклад, змінені години роботи, закриття закладу на ремонт або зміни в умовах надання послуг можуть залишитися непоміченими для потенційних відвідувачів.

Важливим аспектом є актуальність та точність інформації про місця для відпочинку. Багато ресурсів не забезпечують своєчасного оновлення даних, що може призвести до незадоволення користувачів. Наприклад, змінені години роботи, закриття закладу на ремонт або зміни в умовах надання послуг можуть залишитися непоміченими для потенційних відвідувачів.

Таким чином, наявні системи інформування не завжди забезпечують достатній рівень зручності та актуальності для користувачів. Це підкреслює

необхідність створення мобільного додатку, який об'єднає всі необхідні функції для швидкого та ефективного підбору місць для відпочинку. Такий додаток повинен забезпечувати користувачів миттєвим доступом до актуальної інформації, персоналізованими рекомендаціями та зручним інтерфейсом.

Актуальність розробки такого додатку обумовлена не тільки потребами користувачів, але й потенційними вигодами для бізнесу та туристичної індустрії. Надаючи користувачам можливість швидко знаходити та обирати місця для відпочинку, додаток сприятиме підвищенню відвідуваності закладів та розвитку місцевої економіки.

1.3 Аналіз програм-аналогів підбору місць для відпочинку

У сфері мобільних додатків існує безліч аналогічних систем, які надають можливості для підбору місць для відпочинку. Наведемо приклади двох можливих конкурентів:

"Google Maps" – цей додаток, в першу чергу, є онлайн-картами, але одна з важливих функцій додатку полягає у наданні інформації про місця для відпочинку. Користувачі можуть шукати ресторани, парки, туристичні пам'ятки та інші місця відпочинку, отримуючи детальні відомості про місцезнаходження, рейтинги, відгуки, час роботи і навіть фотографії. Додаток також дозволяє прокладати маршрути до обраних місць, надаючи інформацію про відстань, час прибуття та можливі транспортні засоби (рис. 1.1). Окрім цього, "Google Maps" надає можливість користувачам зберігати улюблені місця та ділитися ними з друзями. Інтеграція з іншими сервісами Google, такими як Google My Business, дозволяє отримувати актуальну інформацію від самих закладів. Завдяки режиму огляду вулиць користувачі можуть віртуально подорожувати та оцінювати місця заздалегідь. Можливість отримувати сповіщення про пробки та зміни в дорожній ситуації робить додаток корисним не тільки для туристів, але й для місцевих жителів [2].

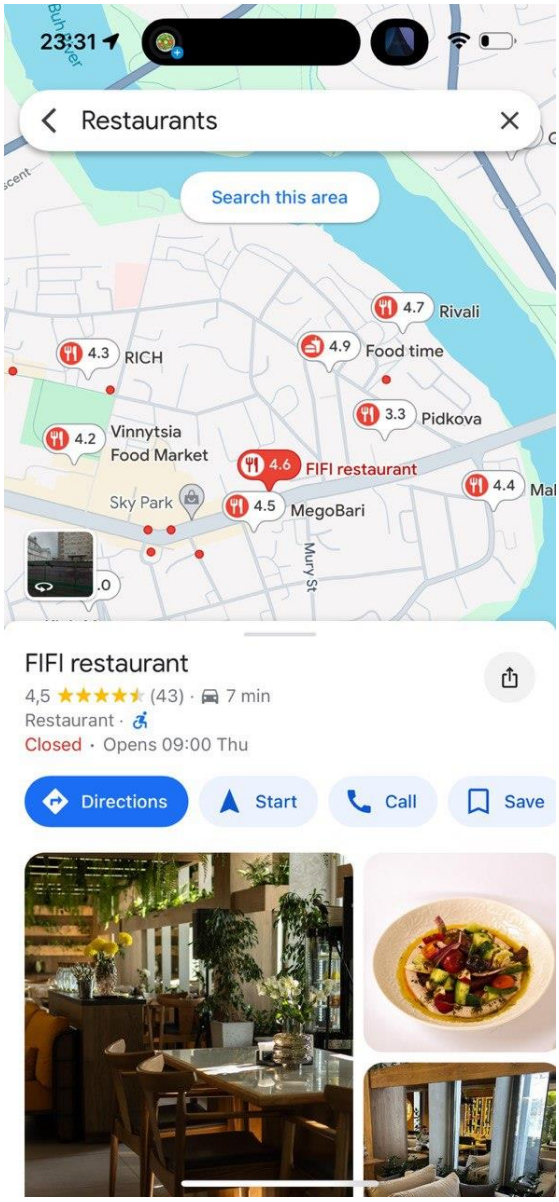

Рисунок 1.1 – Інтерфейс додатку «Google Maps»

"TripAdvisor" – цей додаток є не тільки прикладом можливого конкурента, але й уособлює майже всі інші додатки-аналоги. TripAdvisor пропонує широкий спектр можливостей для планування відпочинку, включаючи огляди, рейтинги, фотографії, детальну інформацію про різні місця для відпочинку, готелі, ресторани, розваги і багато іншого. Користувачі можуть знайти найбільш популярні місця, дізнатися про їхні особливості, а також зберігати обрані локації у власному профілі (рис. 1.2).

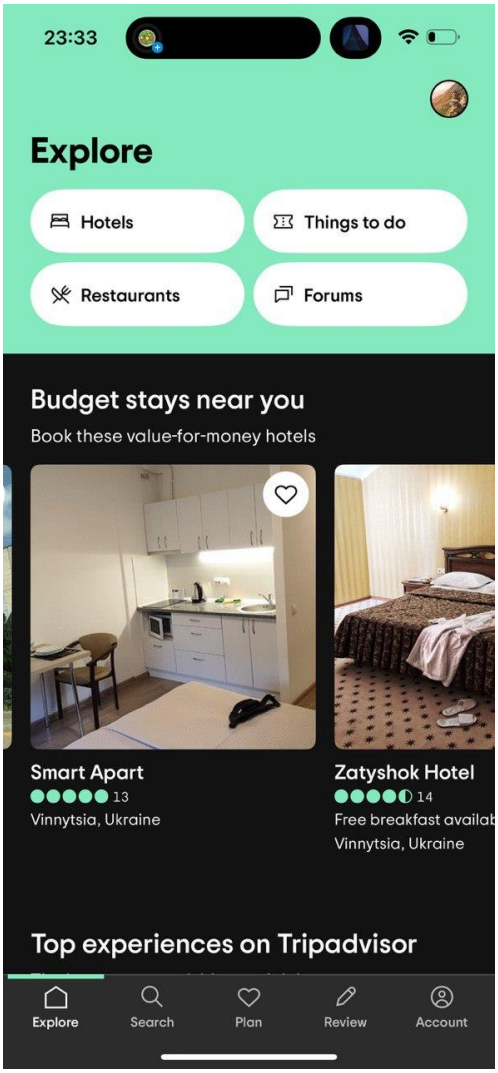

Рисунок 1.2 – Інтерфейс додатку «TripAdvisor»

У першому випадку йдеться про велику організацію, конкуренція з якою у повному сенсі цього слова є складною. Однак головна перевага нового додатку над аналогами полягає у новому підході до проблеми. Додаток буде розрахований на жителів міст і містечок, які хочуть швидко та ефективно знайти місця для відпочинку, не витрачаючи зайвий час на пошук інформації в різних джерелах. [3]

Додаток буде містити функцію персоналізованих рекомендацій на основі вподобань користувача, що робить його ще більш зручним у використанні. Крім того, інтеграція з соціальними мережами дозволить користувачам ділитися своїми враженнями та отримувати поради від друзів. Завдяки цьому додатку, процес планування відпочинку стане значно простішим та швидшим.

1.4 Постановка задачі дослідження

Головна задача даного мобільного додатку - надання користувачу інформації про найближчі місця для відпочинку та можливість знайомств з іншими людьми, які знаходяться поблизу. Зручність і актуальність такої інформації є важливими аспектами, які сприяють задоволеності користувачів та їхньому комфортному відпочинку.

Основні функції мобільного додатку підбору місць для відпочинку включають кілька ключових елементів:

Вхід до системи. Авторизація: Користувач може здійснити вхід до системи через реєстрацію або за допомогою соціальних мереж, що забезпечує легкий та швидкий доступ до додатку.

Перегляд найближчих місць для відпочинку. Геолокація: Додаток збирає інформацію про геолокацію користувача і на основі цих даних пропонує найближчі до нього місця для відпочинку, такі як ресторани, парки, туристичні пам'ятки тощо.

Редагування параметрів пошуку. Налаштування: Користувач може редагувати параметри пошуку місць для відпочинку, вказуючи свої вподобання щодо типу місць, їхнього рейтингу, відстані тощо. Це дозволяє індивідуально налаштовувати пошук і знаходити саме ті місця, які відповідають потребам користувача.

Вибір потрібного місця. Інформація: Додаток надає детальну інформацію про кожне місце, включаючи відгуки, рейтинги, час роботи та інші важливі деталі, що допомагає користувачу зробити обґрунтований вибір.

Отримання сповіщень. Сповіщення: Користувач отримує сповіщення про нові місця для відпочинку або нових користувачів для знайомства, що з'явилися поблизу. Це дозволяє користувачу завжди бути в курсі нових можливостей та пропозицій.

Знаходження найближчих місць для відпочинку. Пошук: Геомодуль знаходить і пропонує користувачу найближчі місця для відпочинку знаходить і пропонує користувачу найближчі місця для відпочинку забезпечуючи швидкий

та зручний доступ до інформації.

Надання даних про доступні місця і користувачів для знайомства

Інформація: Геомодуль надає інформацію про доступні місця для відпочинку і користувачів, які знаходяться поблизу, що сприяє можливості нових знайомств та розширення соціальних контактів.

Додаткові можливості. Персоналізовані рекомендації: На основі попередніх вподобань користувача, додаток може надавати персоналізовані рекомендації, що значно полегшує процес вибору місця для відпочинку.

Інтеграція з іншими сервісами: Можливість інтеграції з іншими популярними сервісами, такими як соціальні мережі, туристичні платформи, дозволяє зібрати всю необхідну інформацію в одному місці.

Візуалізація даних**:** Додаток має зручний та інтуїтивно зрозумілий інтерфейс, що забезпечує легкість у користуванні та швидкий доступ до необхідної інформації.

Створення такого додатку має значні переваги не тільки для користувачів, але й для бізнесу та туристичної індустрії в цілому. Додаток сприяє підвищенню відвідуваності місць для відпочинку, що позитивно впливає на місцеву економіку. Бізнес може ефективніше залучати клієнтів за допомогою актуальних та персоналізованих пропозицій, які відповідають потребам користувачів.

Забезпечення актуальної інформації та зручного доступу до місць для відпочинку сприяє підвищенню економічної активності в регіоні. Відвідувачі частіше вибиратимуть місцеві заклади, що сприятиме розвитку місцевого бізнесу та створенню нових робочих місць.

Додаток використовує сучасні технології, такі як GPS, інтеграція з соціальними мережами та персоналізовані алгоритми, що забезпечує високу якість послуг. Це дозволяє користувачам отримувати актуальну інформацію в режимі реального часу та робити обґрунтовані рішення щодо вибору місць для відпочинку.

Проектування мобільного додатку для підбору місць для відпочинку включає кілька ключових етапів, кожен з яких спрямований на створення функціонального, зручного та ефективного продукту. Перший етап полягає у визначенні загального

алгоритму роботи додатку, який забезпечить його послідовну та логічну роботу. Важливо розробити чітку структурну схему, яка б відображала основні процеси функціонування додатку. Це допоможе краще зрозуміти, як додаток взаємодіє з користувачами, які кроки необхідні для виконання різних завдань, та як забезпечити максимально зручний користувацький досвід.

1.5 Висновок

Було обґрунтовано актуальність розробки мобільного додатку підбору місць для відпочинку. Зокрема, було проаналізовано зростаючу потребу користувачів у якісному відпочинку та труднощі, які виникають при пошуку оптимальних місць через відсутність централізованого джерела актуальної інформації. Створення такого додатку забезпечить миттєвий доступ до персоналізованих рекомендацій і значно полегшить процес планування відпочинку, що відрізняється від існуючих рішень своєю інтеграцією та зручністю. Розглянуто можливості інтеграції додаткових функцій, зокрема, функції знайомств, яка може значно покращити користувацький досвід і залучення нових користувачів.

Було здійснено аналіз предметної області підбору місць для відпочинку. Описано сучасні проблеми ринку, зокрема велика кількість джерел інформації, які не завжди інтегровані між собою, що ускладнює процес вибору оптимального місця для відпочинку. Розглянуто необхідність централізації інформації та створення мобільного додатку, який би об'єднав всі необхідні функції, забезпечуючи миттєвий доступ до актуальних даних, персоналізовані рекомендації та зручний інтерфейс.

Проведено аналіз аналогічних мобільних додатків для підбору місць для відпочинку, таких як "Google Maps" та "TripAdvisor". Аналіз показав, що хоча ці додатки пропонують широкі можливості, вони не завжди зручні для швидкого пошуку інформації про місця для відпочинку. Розробка нового додатку зосереджена на ефективності надання інформації, що дозволить користувачам

швидко знайти оптимальні місця для відпочинку, не витрачаючи зайвий час на пошук у різних джерелах.

Виконано постановку задачі дослідження для розробки мобільного додатку, що надає користувачам інформацію про найближчі місця для відпочинку та можливість знайомств з іншими людьми. Для цього визначені ключові функції, включаючи авторизацію, перегляд місць, налаштування пошуку, надання детальної інформації, сповіщення, та інтеграцію з іншими сервісами. Розробка такого додатку має переваги над існуючими рішеннями, забезпечуючи персоналізовані рекомендації, зручний інтерфейс та актуальну інформацію, що сприяє підвищенню якості відпочинку та економічної активності в регіоні.

2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ПІДБОРУ МІСЦЬ ДЛЯ ВІДПОЧИНКУ

2.1 Розробка загального алгоритму роботи додатку

Проектування мобільних додатків є ключовим етапом у створенні програмного забезпечення, яке відповідає потребам сучасних користувачів. В умовах швидкого розвитку технологій та зростаючої кількості мобільних пристроїв, мобільні додатки стали невід'ємною частиною нашого повсякденного життя. Вони допомагають нам знаходити інформацію, спілкуватися, організовувати свій час, проводити дозвілля та виконувати безліч інших завдань. Одним із популярних напрямків є створення мобільних додатків для підбору місць для відпочинку, які сприяють покращенню якості життя користувачів шляхом надання зручних та ефективних інструментів для планування свого дозвілля.

Успіх будь-якого мобільного додатку залежить від його здатності задовольнити потреби користувачів та забезпечити високий рівень зручності у використанні. Тому проектування мобільного додатку повинно враховувати як технічні аспекти, так і вимоги користувачів. Важливим завданням розробників є створення інтуїтивно зрозумілого інтерфейсу, який дозволить користувачам швидко і легко знаходити необхідну інформацію. Окрім того, додаток повинен бути стабільним, надійним та забезпечувати високий рівень безпеки даних.

Однією з головних переваг мобільних додатків є їхня здатність використовувати можливості сучасних мобільних пристроїв, такі як GPS, камери, сенсори та інші апаратні компоненти. Це дозволяє створювати додатки, які можуть надавати користувачам персоналізовану інформацію та рекомендації. Наприклад, додаток для підбору місць для відпочинку може використовувати дані про місцезнаходження користувача для пропонування найближчих ресторанів, парків, музеїв та інших місць відпочинку.

Інтеграція соціальних функцій також є важливим аспектом проектування мобільних додатків. Можливість обміну враженнями, рекомендаціями та відгуками

з іншими користувачами може значно підвищити цінність додатку для користувачів. Окрім того, соціальні функції можуть сприяти залученню нових користувачів та збільшенню популярності додатку.

Процес проектування мобільного додатку починається з визначення його цілей та завдань. На цьому етапі розробники повинні чітко розуміти, які проблеми користувачів буде вирішувати додаток, які функції він повинен містити та які вимоги до його роботи. Для цього можна проводити дослідження ринку, аналізувати конкурентів, збирати відгуки користувачів та використовувати інші методи збору інформації.

Наступним етапом є розробка концептуального дизайну додатку. Це включає створення схем користувацьких сценаріїв, розробку макетів інтерфейсу та визначення основних функціональних модулів. На цьому етапі важливо врахувати всі можливі варіанти взаємодії користувача з додатком та забезпечити його зручність та інтуїтивність.

Після завершення концептуального дизайну розпочинається етап технічного проектування. Це включає вибір платформ для розробки, визначення архітектури додатку, розробку бази даних та інших компонентів. Важливо обрати технології, які забезпечать високу продуктивність, масштабованість та безпеку додатку. Для мобільних додатків часто використовуються такі платформи як Android та iOS, а для розробки можуть використовуватися різні мови програмування та фреймворки, такі як Java, Kotlin, Swift, React Native та інші.

Технічне проектування також включає розробку детальних технічних специфікацій, які будуть використовуватися для написання коду. Це включає опис функціональних вимог, інтерфейсів користувача, протоколів обміну даними та інших аспектів роботи додатку. На цьому етапі важливо врахувати всі можливі сценарії використання додатку та забезпечити його стабільну роботу в різних умовах.

Одним із важливих етапів проектування є тестування мобільного додатку. Тестування дозволяє виявити та виправити помилки, оцінити продуктивність додатку та переконатися в його відповідності вимогам користувачів. Для

тестування можуть використовуватися різні методи, такі як функціональне тестування, навантажувальне тестування, тестування безпеки та інші. Важливо проводити тестування на різних пристроях та версіях операційних систем, щоб забезпечити високу якість додатку.

Проектування мобільного додатку для підбору місць для відпочинку включає кілька ключових етапів, кожен з яких спрямований на створення функціонального, зручного та ефективного продукту. Перший етап полягає у визначенні загального алгоритму роботи додатку, який забезпечить його послідовну та логічну роботу. Важливо розробити чітку структурну схему, яка б відображала основні процеси функціонування додатку. Це допоможе краще зрозуміти, як додаток взаємодіє з користувачами, які кроки необхідні для виконання різних завдань, та як забезпечити максимально зручний користувацький досвід.

Основні переваги нового додатку:

- Автоматизованість: Користувачам не потрібно кожного разу відкривати додаток і шукати інформацію про місця відпочинку. Додаток автоматично надаватиме персоналізовані рекомендації на основі попередніх вподобань і геолокації.
- Зручність: Простий і інтуїтивний інтерфейс дозволить користувачам швидко знаходити потрібну інформацію без зайвих зусиль.
- Актуальність: Додаток буде надавати тільки актуальну інформацію про місця відпочинку, включаючи години роботи, наявність місць, спеціальні пропозиції та інші важливі деталі.
- Збереження часу: Користувачі зможуть швидко знайти найбільш підходящі місця для відпочинку, що дозволить зекономити час на пошук і планування.

Крім того, додаток включатиме функцію знайомств, що зробить його ще більш корисним і привабливим для користувачів. Це працює наступним чином:

- Додаток допомагає людям долати бар'єри спілкування та знайомитися з новими людьми саме там, де вони знаходяться.

- Можливість знайомств з людьми, які перебувають у тому ж місці, що й ви, маючи спільні інтереси чи цілі. Наприклад, відвідування одного закладу, перебування в музеї чи на виставці тощо.
- Можливість переглянути профіль людини дозволяє вам краще зрозуміти її, перш ніж починати розмову. Можна побачити, чи вона сама, чи з друзями чи партнером.
- Додаток полегшує процес знайомства, адже він відбувається в онлайн-форматі перед тим, як перейти до реального спілкування.

Загалом, цей додаток актуальний для людей, які хочуть долати бар'єри спілкування, знайомитися з новими людьми та зав'язувати стосунки в зручний та простий спосіб. Він надасть користувачам можливість легко знаходити місця для відпочинку та отримувати персоналізовані рекомендації. Такий додаток сприятиме підвищенню якості відпочинку, надаючи актуальну інформацію про місця відпочинку, включаючи години роботи, спеціальні пропозиції та інші важливі деталі. Інтеграція функції знайомств допоможе користувачам знайти однодумців та спростить процес соціальної взаємодії. Це дозволить не тільки ефективно планувати відпочинок, але й розширювати коло знайомств, підвищуючи загальний рівень задоволення від проведеного часу.Таким чином, розробка загального алгоритму роботи додатку є важливим етапом, що забезпечує його функціональність та зручність використання [4].

Схема алгоритму роботи додатку для підбору місць для відпочинку представлена нижче на рисунку 2.1:

Нижче наведено детальний опис алгоритму мобільного додатку для підбору місць для відпочинку:

1. Старт програми:
    - Програма починає свою роботу.
2. Отримання дозволу:
    - На першому етапі програма запитує дозвіл у користувача на використання геолокації.

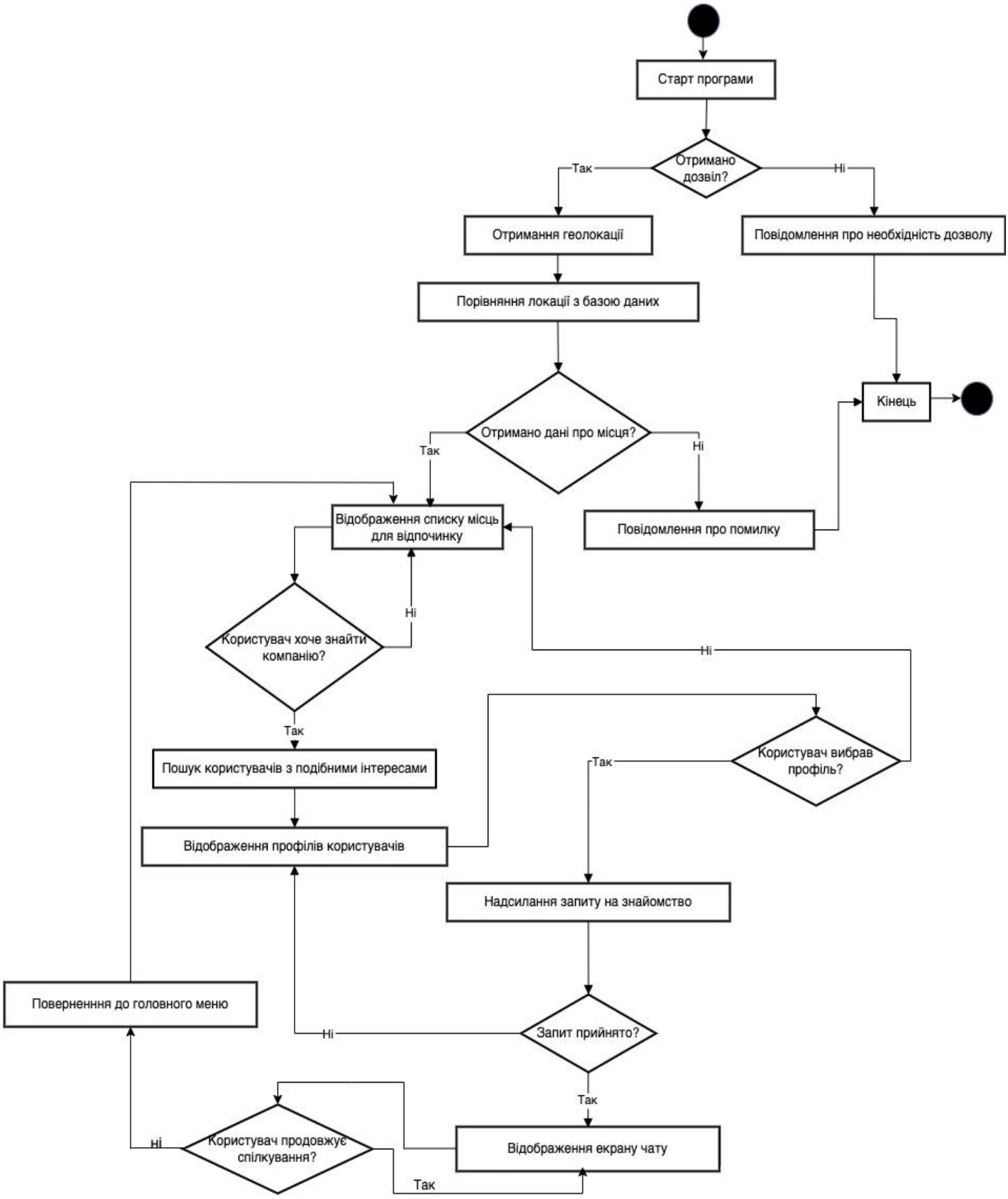

Рисунок 2.1 - Схема алгоритму роботи додатку для підбору місць для відпочинку

- Якщо дозвіл отримано, програма переходить до наступного етапу.
- Якщо дозвіл не отримано, програма відправляє користувачу повідомлення про необхідність дозволу і завершує роботу.

2. Отримання геолокації:
   - Після отримання дозволу програма визначає поточне місцезнаходження користувача.
3. Порівняння локації з базою даних:
   - Програма порівнює отриману геолокацію з даними, що зберігаються у базі даних.
4. Отримання даних про місця:
   - Якщо дані про місця успішно отримано, програма відображає список місць для відпочинку.
   - Якщо дані про місця не отримано, програма відправляє повідомлення про помилку і повертається на початок, або завершує роботу.
5. Користувач хоче знайти компанію?:
   - Програма запитує у користувача, чи хоче він знайти компанію для відпочинку.
   - Якщо ні, програма повертається до початку або завершує роботу.
   - Якщо так, програма переходить до наступного етапу.
6. Пошук користувачів з подібними інтересами:
   - Програма шукає інших користувачів, які мають подібні інтереси і знаходяться поруч.
7. Відображення профілів користувачів:
   - Програма відображає профілі користувачів, які можуть бути потенційними компаньйонами для відпочинку.
8. Користувач вибрав профіль?:
   - Програма запитує у користувача, чи вибрав він один з відображених профілів.
   - Якщо ні, програма повертається до початку або завершує роботу.
   - Якщо так, програма переходить до наступного етапу.

9. Надсилання запиту на знайомство:
    - Програма надсилає запит на знайомство користувачу, профіль якого було вибрано.
10. Запит прийнято?:
    - Програма перевіряє, чи був запит на знайомство прийнятий іншим користувачем.
    - Якщо запит не прийнято, програма повертається до початку або завершує роботу.
    - Якщо запит прийнято, програма відкриває чат для спілкування.
11. Відкриття чату для спілкування:
    - Після прийняття запиту на знайомство програма відкриває чат, де користувачі можуть спілкуватися і домовлятися про зустріч.

Кожен з цих етапів забезпечує послідовну роботу програми, спрямовану на пошук місць для відпочинку та потенційних компаньйонів для користувача.

2.2 Розробка загальної структурної схеми функціонування мобільного додатку для прийому ліків

Структурна схема мобільного застосунку відображає призначення та взаємодію головних структурних модулів програми між собою та з користувачем.

В мобільному застосунку підбору місць для відпочинку є наступні модулі: модуль авторизації, модуль профілю користувача, модуль налаштувань профілю, модуль списку місць поблизу, модуль деталей поточного місця, модуль поточного чату, модуль списку чатів, . Точкою входу в програму є екран авторизації, після якого слідує головне меню з якого починається навігація програмою по інших її складових. Розроблена загальна структурна схема функціонування мобільного додатку підбору місць для відпочинку зображено на рисунку 2.2

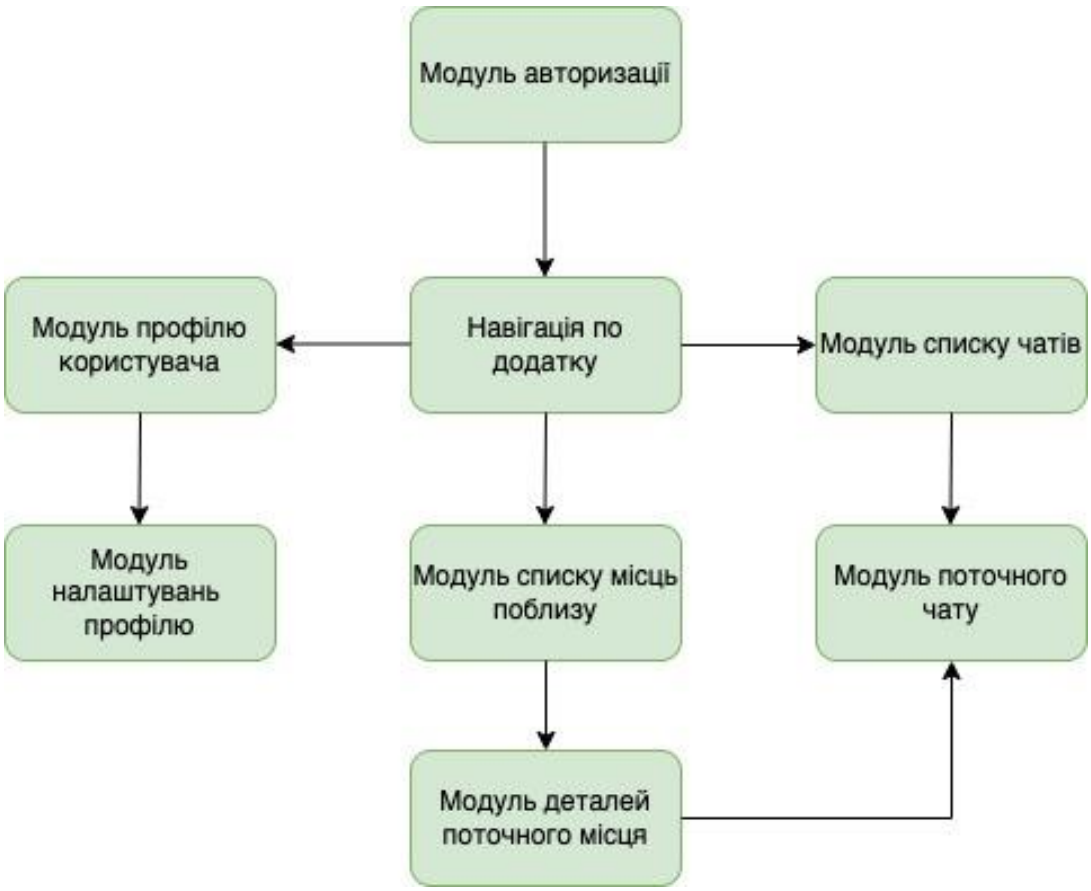

Рисунок 2.2 - Загальна структурна схема функціонування мобільного додатку підбору місць для відпочинку

Ця схема відображає роботу застосунку на найвищому рівні абстракції, тому варто звернути увагу на функціонування гібридних додатків з точки зору їх життєвого циклу.

Функціонування мобільного додатку для підбору місць для відпочинку організовано на основі життєвого циклу аплікації. Цей життєвий цикл контролюється операційною системою і включає різні стани, через які проходить додаток, щоб забезпечити ефективне використання ресурсів і зручний користувацький досвід. Основні методи життєвого циклу додатку: onCreate (); onResume (); onPause (); onDestroy (); [5] Життєвий цикл додатку зображено на рисунку 2.3.

Ініціалізація додатку onCreate():

– Викликається, коли додаток створюється вперше.

– Тут відбувається ініціалізація основних компонентів додатку: налаштування баз даних, створення інтерфейсів користувача, встановлення зв'язків із сервісами.

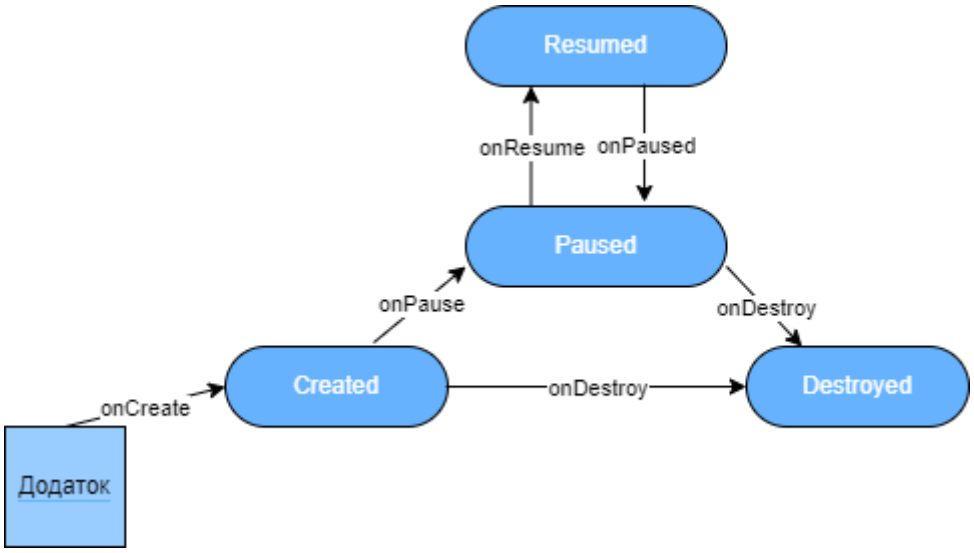

Рисунок 2.3 – Життєвий цикл додатку підбору місць для відпочинку

На цьому етапі додаток готується до роботи, завантажуючи всі необхідні ресурси. Наприклад, він може отримувати поточне місцезнаходження користувача, зчитувати збережені вподобання або налаштовувати з'єднання з сервером для отримання актуальних даних про місця для відпочинку.

Стан активності onStart():

– Викликається, коли додаток стає видимим для користувача, але ще не доступний для взаємодії.

onResume():

– Викликається, коли додаток стає активним і готовим до взаємодії з користувачем.
– Це етап, на якому додаток реагує на дії користувача, наприклад, обробляє натискання кнопок, введення тексту, перегляд місць на карті тощо.

На цих етапах додаток забезпечує максимальну взаємодію з користувачем, відображаючи актуальну інформацію і надаючи доступ до основних функцій, таких

як пошук місць для відпочинку, перегляд деталей, збереження вподобань та початок чату з іншими користувачами.

Стан паузи onPause():

- Викликається, коли додаток перестає бути активним, але ще видимий користувачеві.
- Тут додаток може зберігати поточний стан, призупиняти анімації, зупиняти відтворення відео або аудіо, зберігати дані, введені користувачем, щоб їх можна було відновити пізніше.

Цей етап важливий для забезпечення плавного переходу між додатками і збереження даних користувача в разі тимчасового припинення взаємодії, наприклад, при надходженні телефонного дзвінка або переході до іншого додатку.

Стан зупинки onStop():

- Викликається, коли додаток стає невидимим для користувача.
- На цьому етапі додаток звільняє ресурси, які не потрібні, коли додаток не використовується, наприклад, зупиняє оновлення геолокації, звільняє об'єкти, що займають багато пам'яті.

На цьому етапі важливо зберегти всі необхідні дані, щоб при наступному запуску додатку користувач зміг продовжити роботу з того місця, де зупинився.

Завершення роботи onDestroy():

- Викликається перед тим, як додаток буде знищено.
- Використовується для фінального очищення ресурсів і звільнення пам'яті.

Цей метод завершує життєвий цикл додатку, забезпечуючи його коректне завершення і звільнення всіх ресурсів, що використовуються, для уникнення витоку пам'яті та інших проблем, пов'язаних із ресурсами.

Життєвий цикл додатку для підбору місць для відпочинку є критичним для забезпечення його стабільної та ефективної роботи. Система Android надає різні методи, які дозволяють розробникам контролювати стан додатку і реагувати на зміни в його стані. Використовуючи ці методи, можна забезпечити безперервний та плавний користувацький досвід, зберігаючи при цьому ефективне використання ресурсів пристрою.

- onCreate(): Ініціалізація основних компонентів, налаштування з'єднань з базами даних та серверами.
- onStart(): Підготовка додатку до відображення користувачеві.
- onResume(): Активація інтерфейсу для взаємодії з користувачем, оновлення даних в режимі реального часу.
- onPause(): Збереження стану додатку, припинення ресурсомістких процесів.
- onStop(): Звільнення ресурсів, збереження поточного стану додатку.
- onDestroy(): Фінальне очищення ресурсів, підготовка до знищення додатку.

Кожен з цих етапів є важливим для забезпечення стабільної та ефективної роботи додатку. Розуміння і правильне використання методів життєвого циклу дозволяє розробникам створювати надійні та зручні мобільні додатки, які відповідають очікуванням користувачів і забезпечують високу якість сервісу.

Ця структура життєвого циклу дозволяє додатку для підбору місць для відпочинку залишатися функціональним і ефективним, навіть коли користувач перемикається між різними додатками або виконанням інших завдань на своєму пристрої. Завдяки цьому додаток може оперативно надавати актуальну інформацію, забезпечуючи високу якість відпочинку та підвищуючи задоволеність користувачів.

Для більш детального опису компонентів програми та зв’язків між ними використаємо UML-діаграму класів, для додатку, що зображена на рисунку 2.4.

2.3 Розробка діаграми прецедентів мобільного додатку для підбору місць для відпочинку

Розробка мобільного додатку підбору місць для відпочинку включає не тільки технічні аспекти, але й аналіз користувацького досвіду та взаємодії з системою. Одним із важливих етапів розробки є створення діаграми прецедентів, яка відображає основні дії користувача та відповідні реакції системи.

Це допомагає зрозуміти, як користувачі взаємодіють із додатком, які функції є ключовими та як забезпечити максимально зручний та інтуїтивно зрозумілий інтерфейс.

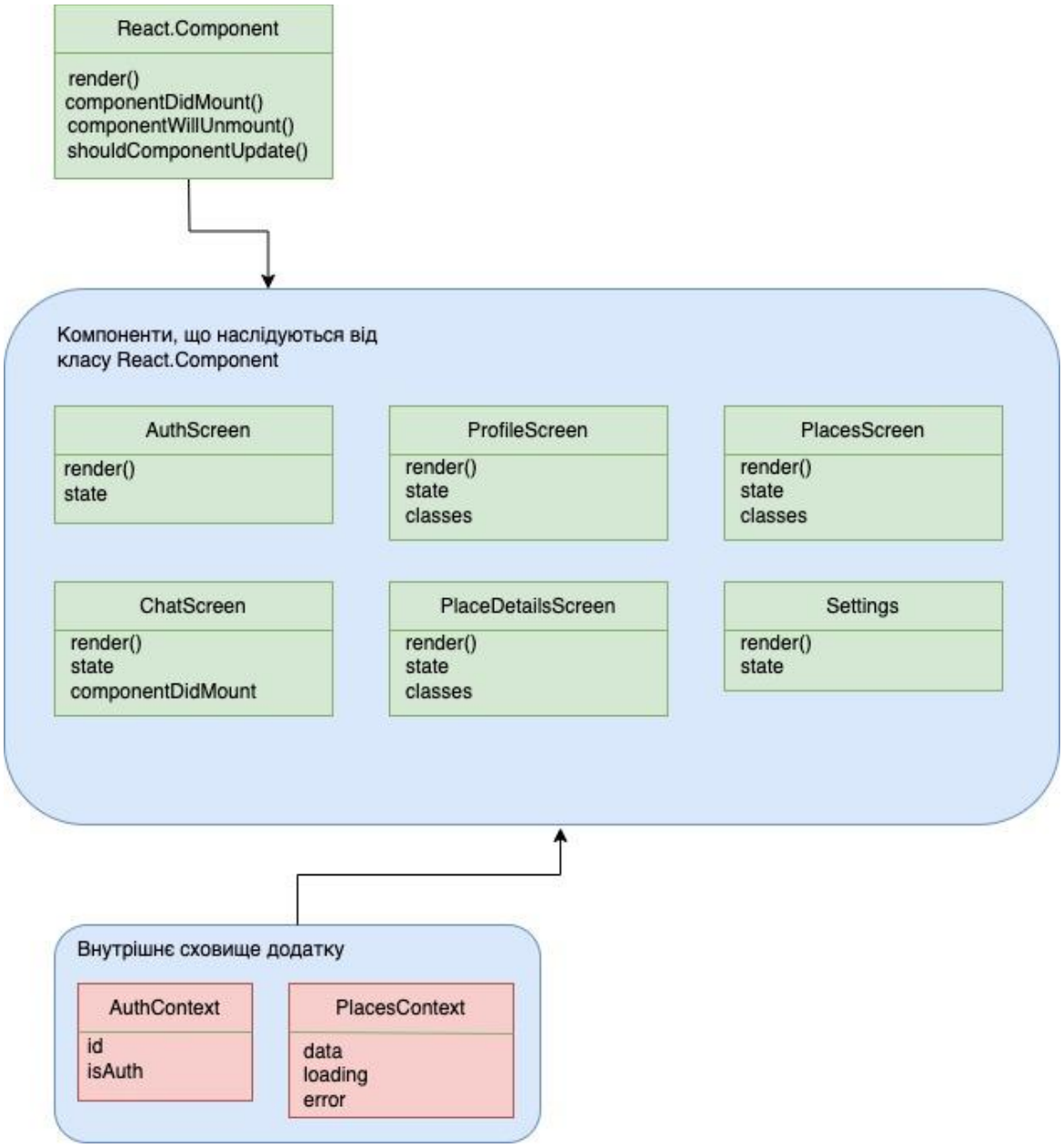

Рисунок 2.4 – UML-Діаграма класів мобільного додатку підбору місць для відпочинку

На рисунку 2.5 наведено діаграму прецедентів мобільного додатку для підбору місць для відпочинку. [6] На діаграмі прецедентів зображено взаємодію користувачів із системою мобільного додатку для підбору місць для відпочинку.

Основні дії, які можуть виконувати зареєстровані та незареєстровані користувачі, включають авторизацію, перегляд списку місць, вибір місця для відпочинку та оповіщення системи. Діаграма складається з кількох основних прецедентів, які детально описані нижче.

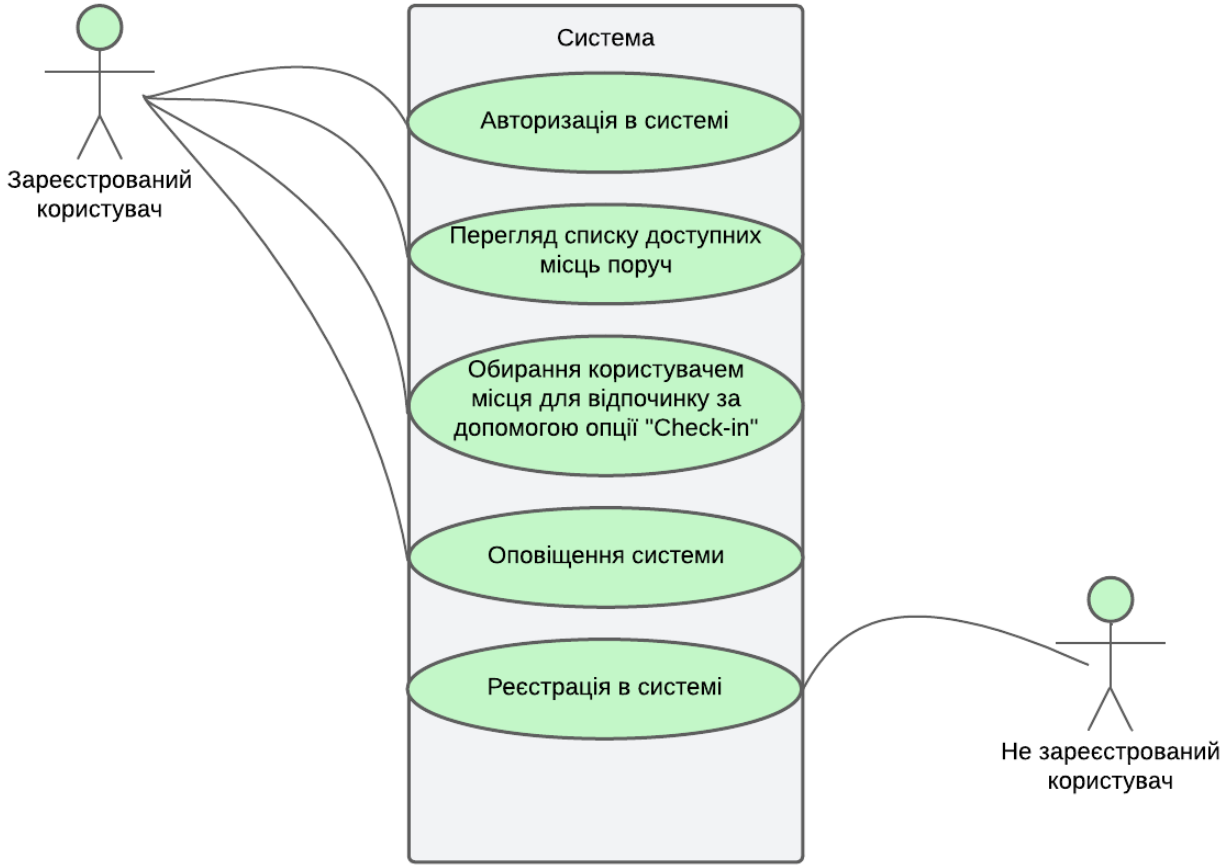

Рисунок 2.5 – Діаграма прецедентів мобільного додатку для підбору місць для відпочинку

Авторизація в системі: Цей прецедент передбачає, що зареєстрований користувач вводить свої облікові дані для входу в систему. Ціль цього прецеденту – надати користувачу доступ до персоналізованих функцій додатку, таких як перегляд найближчих місць та можливість взаємодії з іншими користувачами. Це критичний крок, який забезпечує захист особистих даних користувачів та персоналізацію їхнього досвіду в додатку.

Реєстрація в системі: Незареєстрований користувач може здійснити реєстрацію, заповнивши необхідні дані для створення облікового запису. Цей

прецедент дозволяє новим користувачам отримати доступ до всіх функцій додатку, включаючи перегляд місць, чек-ін та взаємодію з іншими користувачами. Реєстрація є необхідною для забезпечення безпеки та відстеження дій користувачів у системі.

Перегляд списку доступних місць поруч: Після авторизації або реєстрації користувач отримує доступ до списку місць, що знаходяться поблизу його місцезнаходження. Цей прецедент дозволяє користувачам швидко знаходити цікаві місця для відпочинку з урахуванням їхнього поточного місцезнаходження. Функція геолокації грає ключову роль у цьому прецеденті, забезпечуючи точність та релевантність запропонованих місць.

Обрання користувачем місця для відпочинку за допомогою опції "Check-in": Користувач вибирає місце з доступного списку і здійснює чек-ін, що сигналізує іншим користувачам про його присутність. Це дозволяє іншим користувачам бачити, хто знаходиться в даному місці, і потенційно взаємодіяти з ними. Цей прецедент сприяє соціальній взаємодії та полегшує знайомства між користувачами, які мають спільні інтереси.

Оповіщення системи: Система автоматично оновлює статус користувача, коли він залишає місце, використовуючи дані геолокації. Це дозволяє підтримувати актуальність інформації про присутність користувачів у певних місцях без необхідності в ручних діях з боку користувача. Автоматичне оповіщення забезпечує зручність та точність даних, що підвищує загальну ефективність роботи додатку.

Користувачі: Зареєстрований користувач: Цей тип користувача має доступ до всіх функцій додатку, включаючи авторизацію, перегляд списку місць, чек-ін та взаємодію з іншими користувачами. Зареєстровані користувачі можуть максимально використовувати всі можливості додатку для планування свого відпочинку та знайомства з новими людьми.

Незареєстрований користувач: Цей тип користувача має доступ тільки до функції реєстрації. Після успішної реєстрації користувач отримує можливість

використовувати всі функції додатку. Реєстрація є першим кроком для нового користувача, що дозволяє йому стати повноправним учасником спільноти додатку.

Діаграма прецедентів ілюструє основні дії користувачів і те, як вони взаємодіють із системою для досягнення своїх цілей. Це дозволяє зрозуміти, які функції є найважливішими та яким чином вони пов'язані між собою, що сприяє ефективному плануванню та розробці додатку.

2.4 Висновок

Розроблено загальний алгоритм функціонування додатку. Даний алгоритм забезпечує послідовну та логічну роботу додатку, спрямовану на задоволення потреб користувачів. Від початку роботи програми, через отримання геолокаційних даних, до вибору місця для відпочинку і взаємодії з іншими користувачами, кожен етап алгоритму враховує можливі варіанти дій користувачів та відповідні реакції системи. Запропонований алгоритм відрізняється від існуючих автоматизованістю процесів, персоналізованими рекомендаціями, зручним та інтуїтивним інтерфейсом, а також інтеграцією соціальних функцій для покращення користувацького досвіду. Такий підхід дозволяє зменшити ймовірність виникнення помилок та підвищити загальну зручність використання додатку.

Також було розглянуто ключові аспекти проектування мобільного додатку для підбору місць для відпочинку, зокрема загальну структурну схему , яка ілюструє взаємодію основних модулів додатку: модуль авторизації, модуль профілю користувача, модуль налаштувань профілю, модуль списку місць поблизу, модуль деталей місця, модуль чату тощо. Це забезпечує чітке розуміння основних процесів функціонування додатку. Описано життєвий цикл додатку, який включає етапи onCreate(), onStart(), onResume(), onPause(), onStop() та onDestroy(). Це дозволяє контролювати стан додатку та забезпечувати стабільну роботу при переходах між різними станами. Використання UML-діаграми класів допомогло детально описати структуру додатку та зв'язки між його компонентами, що полегшує процес розробки та подальшого підтримання додатку. Перелічені вище

розробки забезпечують додаток гнучкістю, стабільністю та зручністю використання, що вигідно відрізняє його від існуючих рішень.

Було створено діаграму прецедентів, яка ілюструє основні дії користувачів у мобільному додатку для підбору місць для відпочинку. Було розглянуто взаємодію зареєстрованих та незареєстрованих користувачів із системою через прецеденти авторизації, перегляду місць, вибору місця для відпочинку, чек-ін та оповіщення системи. Це дозволило детально зрозуміти, як користувачі будуть використовувати додаток, що сприяє кращому плануванню та оптимізації функцій, роблячи додаток більш інтуїтивним і зручним у порівнянні з існуючими рішеннями.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ПІДБОРУ МІСЦЬ ДЛЯ ВІДПОЧИНКУ

3.1 Обгрунтування вибору мови та середовища програмування

Головна задача розробника – створення ефективного та простого в користуванню додатку. Для цього потрібно відповідально віднестись до задачі вибору мови програмування та середовища програмування. В даній роботі будуть використовуватися дві мови програмування: для написання геомодуля та для реалізації клієнтської частини.

Для написання геомодуля потрібно використовувати нативну мову сумісну із розробкою на мобільні телефони.

Перерахуємо мови програмування на яких можливо написати мобільний додаток:

- Java;
- Kotlin;
- Swift;
- Objective C.

Розглянемо характеристики та особливості кожної мови програмування:

Kotlin — мова програмування зі статичною типізацією, яка працює на JVM та розробляється компанією JetBrains. Вона також може компілюватися в JavaScript. Мета розробників полягала в створенні мови, яка б була лаконічнішою та безпечнішою в типізації, ніж Java, і простішою у використанні, ніж Scala. Спрощення, порівняно зі Scala, призвело до швидшої компіляції та кращої підтримки інтегрованого середовища розробки (IDE) [7].

Використовуючи Kotlin для розробки Android, ви отримаєте такі переваги:

- Менше коду та краща читабельність: Це дозволяє витрачати менше часу на написання та розуміння коду.
- Зріла екосистема: З 2011 року Kotlin постійно розвивався і зараз безперешкодно інтегрований в Android Studio, використовується багатьма компаніями.

- Підтримка Android Jetpack: KTX розширення додають функції Kotlin до бібліотек Android, включаючи функції розширення, лямбда-вирази та названі параметри.
- Сумісність з Java: Ви можете використовувати Kotlin разом з Java без необхідності повного перенесення коду.
- Багатоплатформність: Kotlin дозволяє розробляти додатки для Android, iOS, бекенду та веба, з можливістю спільного використання коду.
- Безпека коду: Краща читабельність зменшує кількість помилок. Компілятор Kotlin виявляє ці помилки, забезпечуючи безпеку коду.
- Легкість вивчення: Kotlin легко вивчити, особливо для розробників Java.
- Велика спільнота: Kotlin має значну підтримку та внески від глобальної спільноти. За даними Google, понад 60% найпопулярніших додатків у Play Store використовують Kotlin. [6]

Swift та Objective C – Також використовуються для написання сучасних мобільних додатків, але є розроблені та заточені в більшій мірі під розробку на операційну систему - IOS [8, 9].

Створення мови програмування Java стало одним із найважливіших досягнень у сфері програмного забезпечення за останні двадцять років. Якщо HTML був необхідний для статичного розміщення веб-сторінок, то Java забезпечила значний прорив у створенні інтерактивних продуктів для Інтернету. Три ключові особливості Java:

- Аплети: Маленькі, надійні та динамічні програми, які можна легко інтегрувати у веб-сторінки. Вони можуть бути поширені серед користувачів так само просто, як і HTML-документи.
- Об'єктно-орієнтована розробка: Java поєднує простий синтаксис з надійним середовищем розробки, що дозволяє швидко створювати нові програми.
- Багатий набір класів: Java надає розробникам широкий спектр інструментів для абстракції системних функцій, таких як робота з вікнами, мережею та введенням-виведенням.

Java успадкувала від C++ потужний механізм об'єктно-орієнтованого програмування. Вона була створена з нуля, без необхідності забезпечувати зворотну сумісність із попередніми версіями, що дозволило розробникам створити зрозумілу та практичну об'єктну модель. Java також зберігає прості типи даних, що підвищує швидкість їх обробки [10].

Простота програмування на Java знижує витрати на розробку в порівнянні з іншими потужними мовами. Її переносимість дозволяє знизити витрати на адаптацію програм для різних платформ. Інструменти для розробки Java-програм часто є безкоштовними, що робить її ідеальною для створення некомерційних програмних продуктів, зокрема в освіті [11].

Java орієнтована на створення надійних програм, що є важливим для фінансових транзакцій у банківських системах. Сувора типізація та автоматичне управління пам'яттю роблять Java безпечною та надійною. Java також підтримує об'єктно-орієнтовану обробку виняткових ситуацій, що дозволяє ефективно обробляти помилки часу виконання [12].

Java розроблялася з урахуванням вимог до створення інтерактивних програм, що працюють з мережею. Вона підтримує багатопоточне програмування, що дозволяє створювати високоефективні інтерактивні системи. Java також вирішує проблеми довготривалості та переносимості програм, забезпечуючи стабільну роботу програм незалежно від змін у системному середовищі. Її легко розширювати завдяки створенню нових стандартних класів та бібліотек, що робить Java гнучкою та придатною для довготривалого використання.

Java є потужним інструментом для розробників, що забезпечує простоту, безпеку та переносимість програмного забезпечення.

Для реалізації клієнтської частини потрібно використати фреймворк, розроблений саме для цієї мети, враховуючи його легкість та стабільність, щоб зменшити ризик використання більшої кількості ресурсів додатком, ніж це насправді потрібно.

Перерахуємо мови фреймворки на яких можливо реалізувати дану задачу:

- Ionic;

- Flutter;
- NativeScript;
- React Native.

Розглянемо характеристики та особливості кожної мови програмування:

Ionic є повним SDK з відкритим кодом, який використовується для розробки гібридних мобільних додатків. Його остання версія побудована як набір веб-компонентів, що дозволяє розробникам вибирати будь-яку структуру користувацького інтерфейсу, таку як Angular, React або Vue.js. Іонік також підтримує використання компонентів без прив'язки до конкретного фреймворка. Він надає інструменти та послуги для розробки гібридних мобільних, настільних і прогресивних веб-додатків на основі сучасних веб-технологій, таких як CSS, HTML5 і Sass. Веб-додатки, створені за допомогою Ionic, можуть бути розповсюджені через нативні магазини додатків за допомогою Cordova або Capacitor [13].

Flutter – це фреймворк з відкритим кодом від Google для створення додатків для платформ Android, iOS та веб. Він використовується як основний спосіб створення додатків для Google Fuchsia. Flutter складається з трьох основних частин:

- Flutter рушій: Це програмний рушій для рендерингу, написаний на C++ з використанням бібліотеки Skia.
- Базова бібліотека (Foundation library): Складається з класів та функцій, написаних на Dart, для побудови програм і взаємодії з Flutter рушієм.
- Віджети: Інтерфейс користувача в Flutter будується з віджетів, які є незмінними об'єктами, що описують частини інтерфейсу. Вся графіка, текст та анімація створюються за допомогою віджетів [14].

NativeScript – це фреймворк з відкритим вихідним кодом, розроблений компанією Telerik для створення додатків для Android та iOS. Додатки розробляються на платформонезалежних мовах, таких як JavaScript або TypeScript, і мають повний доступ до API платформ. NativeScript підтримує Angular і дозволяє включати сторонні бібліотеки з таких джерел, як Cocoapods, Android Arsenal, Maven і npm.js без створення додаткових прошарків [15].

React Native – це платформа для створення мобільних додатків з відкритим кодом від Facebook. Вона використовується для розробки додатків для Android, iOS, веб та UWP, дозволяючи розробникам використовувати React разом із можливостями рідної платформи. React Native використовує JavaScript для маніпуляції нативними компонентами через асинхронний та пакетний міст, що забезпечує швидкість і гнучкість розробки. React Native дозволяє писати нативний код мовами Java для Android та Objective-C або Swift для iOS, що робить його ще більш універсальним [16].

При виборі середовища програмування зазвичай в першу чергу звертається увага на його простоту, легкість та зручність в користуванні, однак при мобільній розробці найважливішим аспектом є - надання всіх можливих функціональних можливостей, які лише можуть виникнути під час розробки додатку, автоматизація всіх можливих потреб, при спробі виконання яких в інших середовищах довелося б шукати шляхи для їх реалізації. чи то пошук команд, чи встановлення додаткових бібліотек. Тому в якості середовища програмування найраще підходить - Android Studio

Android Studio, яка замінила плагін ADT для платформи Eclipse, побудована на базі вихідного коду продукту IntelliJ IDEA Community Edition від JetBrains і розвивається у відкритій моделі під ліцензією Apache 2.0. Підготовлені бінарні складання доступні для Linux, Mac OS X та Windows. Середовище підтримує розробку додатків для смартфонів, планшетів, носимих пристроїв на базі Android Wear, телевізорів Android TV, окулярів Google Glass і автомобільних інформаційно-розважальних систем Android Auto [17].

Для застосунків, розроблених в Eclipse [18] з ADT Plugin, надано інструмент для автоматичного імпорту проектів в Android Studio. Середовище адаптоване для виконання типових завдань, включає засоби для тестування сумісності з різними версіями платформи та інструменти для проектування додатків для пристроїв з різними розмірами екранів.

В Android Studio реалізовано багато додаткових функцій, таких як нова уніфікована підсистема складання, тестування та розгортання додатків, заснована

на інструментарії Gradle і підтримуюча безперервну інтеграцію. Середовище включає колекцію типових елементів інтерфейсу та візуальний редактор для їх компонування, що дозволяє попередньо переглядати різні стани інтерфейсу для різних версій Android і розмірів екранів.

Для створення нестандартних інтерфейсів передбачений майстер створення власних елементів оформлення з підтримкою шаблонів. У середовище інтегровані функції завантаження типових прикладів коду з GitHub. Також воно включає розширені інструменти рефакторингу, перевірки сумісності з попередніми випусками, виявлення проблем продуктивності, моніторингу споживання пам'яті та оцінки зручності використання.

Редактор додано режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Вбудована підтримка оптимізатора коду ProGuard. Надані засоби генерації цифрових підписів та інтерфейс для управління перекладами на інші мови.

3.2 Реалізація модуля отримання геоданних

Основа роботи даного додатку полягає в діяльності модуля отримання геоданих. Оскільки дуже важливо отримати поточне місцезнаходження користувача, для порівняння його із даними про найближчу зупинку для визначення її і в подальшому використанню даних, відповідно до отриманої зупинки.

При першому запуску додатку буде отримано вперше його геолокацію. Для цього потрібно визначити locationListener для того, щоб додаток мав змогу зчитувати дані про геолокацію [19].

```
private final LocationListener locationListener = new
LocationListener() {
    @Override
    public void onLocationChanged(Location loc) {
        // Code to handle location change
    }
```

```
};
```

Наступний важливий крок - отримати locationManager для отримання данних про зміну геопозиції пристрою користувача.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
    locationManager.requestLocationUpdates(
        LocationManager.GPS_PROVIDER,
        LOCATION_REFRESH_TIME,
        LOCATION_REFRESH_DISTANCE,
        locationListener
    );
}
```

Таким чином, ми забезпечили додаток даними про місце знаходження і в подальшому, буде можливе використання функції `getLastLocation()`.Останнє відоме місце розташування пристрою забезпечує зручну базу для запуску, гарантуючи, що додаток має відоме місце до початку періодичних оновлень місцеположення.

При повторних запусках додатку, потрібно забезпечити те, що перед оновленням місцяположення, додаток буде підключатися до служб локації та робити запит про місцезнаходження.

```
@Override
protected void onResume() {
    super.onResume();
    if (requestingLocationUpdates) {
        initiateLocationUpdates();
    }
}

private void initiateLocationUpdates() {
    fusedLocationClient.requestLocationUpdates(
        locationRequest,
        locationCallback,
        Looper.getMainLooper()
```

```
    );
}
```

При виконані попередніх дій повертається колбек функція, що містить часову мітку отримання данних, а також довготу та широту, дані про місце розташування пристрою користувача, які потрібно дістати для подальшого опрацювання.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    locationCallback = new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult result)
{
            if (result == null) {
                return;
            }
            for (Location loc : result.getLocations()) {
                // Your code here...
            }
        }
    };
}
```

При завершенню роботи додатку, було вирішено зупиняти оновлення зчитування даних про геолокацію, для зменшення затрат на використання ресурсів пристрою.

```
@Override
protected void onPause() {
    super.onPause();
    haltLocationUpdates();
}

private void haltLocationUpdates() {

fusedLocationClient.removeLocationUpdates(locationCallback)
;
}
```

Зміна конфігурації пристрою, наприклад зміна орієнтації екрана чи мови, може призвести до знищення поточної активності. Тому додаток повинен зберігати будь-яку інформацію, необхідну для відновлення діяльності. Тому останньою функціює, що входить до геомодуля є - збереження отриманих результатів.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    restoreValuesFromBundle(savedInstanceState);
}

private void restoreValuesFromBundle(Bundle
savedInstanceState) {
    if (savedInstanceState == null) {
        return;
    }
    if
(savedInstanceState.keySet().contains(REQUESTING_LOCATION_U
PDATES_KEY)) {
        requestingLocationUpdates =
savedInstanceState.getBoolean(
                REQUESTING_LOCATION_UPDATES_KEY);
    }
    refreshUI();
}
```

Таким чином, було розроблено геомодуль відповідно до поставлених задач. При запуску додатку він зчитуватиме поточне положення користувача, після чого дані зберігаються як останнє місцеположення, для автоматизації та полегшенння роботи. В той же час вони обробляються та дістаються у вигляді часової мітки та довжини і широти. Що важливо, по завершенню роботи геомодуль зупиняє отримання данних про роботу для зменшення навантаження на акамулятор пристрою та зберігає дані.

Зважаючи на те, що геомодуль відіграє ключову роль у функціонуванні додатку, важливо враховувати низку додаткових аспектів, які забезпечать його ефективну роботу. Серед них – обробка різноманітних сценаріїв використання додатку, інтеграція з іншими модулями додатку та забезпечення надійності й безпеки даних користувача.

Геомодуль повинен бути здатний адаптуватися до різних сценаріїв використання. Це включає обробку ситуацій, коли користувач знаходиться в зоні з поганим покриттям GPS або коли пристрій не має доступу до інтернету. В таких випадках, додаток повинен використовувати кешовані дані або пропонувати альтернативні методи отримання геолокації, наприклад, через мобільні мережі або Wi-Fi. Також важливо враховувати можливість роботи в режимі енергозбереження, що зменшує частоту оновлень геолокації для збереження батареї пристрою.

Геомодуль тісно взаємодіє з іншими частинами додатку, такими як модуль рекомендацій, інтерфейс користувача та система сповіщень. Важливо забезпечити безперебійну роботу цих компонентів, щоб користувачі могли отримувати актуальну інформацію та персоналізовані рекомендації. Наприклад, інтеграція з модулем рекомендацій дозволить додатку надавати користувачам інформацію про найближчі місця для відпочинку, які відповідають їхнім уподобанням. Система сповіщень повинна інформувати користувачів про нові місця або події, що відбуваються поблизу, базуючись на їхньому поточному місцезнаходженні.

Одним з ключових аспектів роботи геомодуля є забезпечення безпеки даних користувача. Це включає захист інформації про місцезнаходження від несанкціонованого доступу та збереження конфіденційності даних. Дотримання цих вимог забезпечить високий рівень довіри користувачів до додатку. Для цього необхідно використовувати сучасні методи шифрування даних, а також обмежити доступ до інформації про місцезнаходження тільки тим сервісам і модулям додатку, які безпосередньо потребують цієї інформації для своєї роботи. Крім того, користувачам повинна надаватися можливість самостійно керувати своїми налаштуваннями конфіденційності, обираючи, які дані можуть бути зібрані та використані додатком.

Геомодуль повинен підтримувати регулярні оновлення для забезпечення його відповідності сучасним вимогам і стандартам. Важливо також забезпечити механізм для отримання зворотного зв'язку від користувачів щодо роботи геомодуля. Це допоможе виявляти та виправляти потенційні проблеми, а також удосконалювати функціональність модуля. Оновлення можуть включати

покращення алгоритмів визначення місцезнаходження, оптимізацію використання ресурсів пристрою та інтеграцію з новими технологіями та сервісами.

Геомодуль є критично важливою складовою мобільного додатку для підбору місць для відпочинку. Його розробка та інтеграція повинні враховувати широкий спектр технічних та користувацьких вимог. Від ефективної роботи геомодуля залежить загальна зручність і функціональність додатку, що безпосередньо впливає на задоволеність користувачів та їхню готовність користуватися додатком у майбутньому. Тому особливу увагу слід приділяти якості отримання та обробки геоданих, інтеграції з іншими модулями додатку, безпеці даних користувачів та регулярному оновленню функціональності геомодуля.

3.3 Реалізація інтерфейсу мобільного додатку підбору місць для відпочинку

Головна задача розробки клієнтської частини - досягнення максимальної зручності для користувача при взаємодії з додатком. Метою додатку є надання інформації про найкращі місця для відпочинку, враховуючи інтереси користувача, а також створення можливості знайомства з іншими користувачами для спільного відпочинку [20].

Вигляд головного екрану наведено на рисунку 3.1

Головний екран додатку складається з декількох ключових секцій, забезпечуючи користувачам легкий доступ до основних функцій.

Навігаційна панель: Розташована в нижній частині екрану.Містить чотири основні вкладки: Home (Головна), Nearby (Поряд), Chat (Чат) і Profile (Профіль).Кожна вкладка дозволяє користувачам швидко переходити між різними розділами додатку.

Блок основної інформації: Займає Верхню частину екрану.Містить промо-банери з інформацією про події та пропозиції. Користувачі можуть бачити актуальні події, місця для бронювання та можливість перегляду найближчих людей.

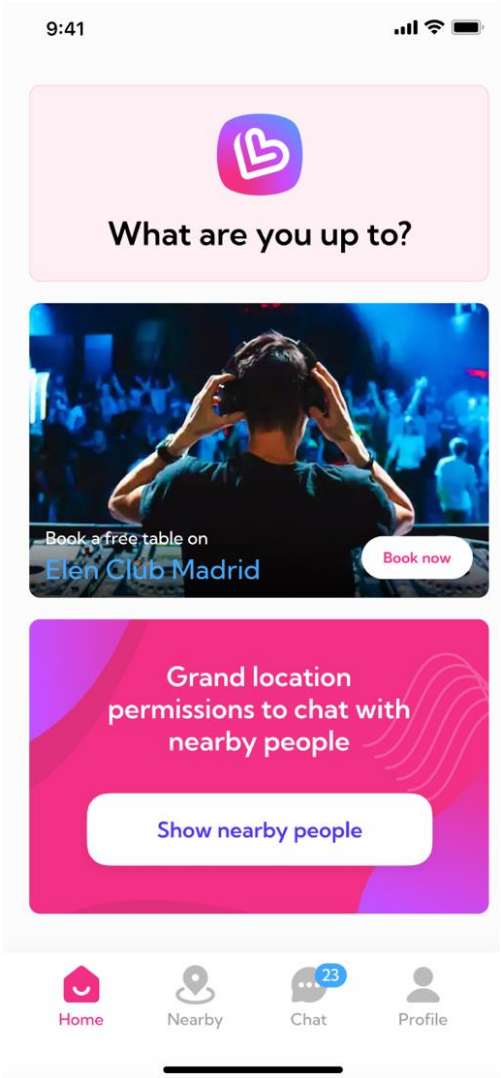

Рисунок 3.1 – Головний екран (Home)

Екран вибору місць (рис.3.2)

Цей екран дозволяє користувачам вибирати місця, що знаходяться поблизу, для відвідування.

Навігаційна панель: Розташована в нижній частині екрану та залишається незмінною для зручності користувача.

Вибір локації: Екран дозволяє користувачу обирати найближчі до нього місця. Також користувач має можливість під час перегляду списку місць бачити інформацію про кількість відвідувачів а саме кількість користувачів додатку з якими можна розпочати чат.

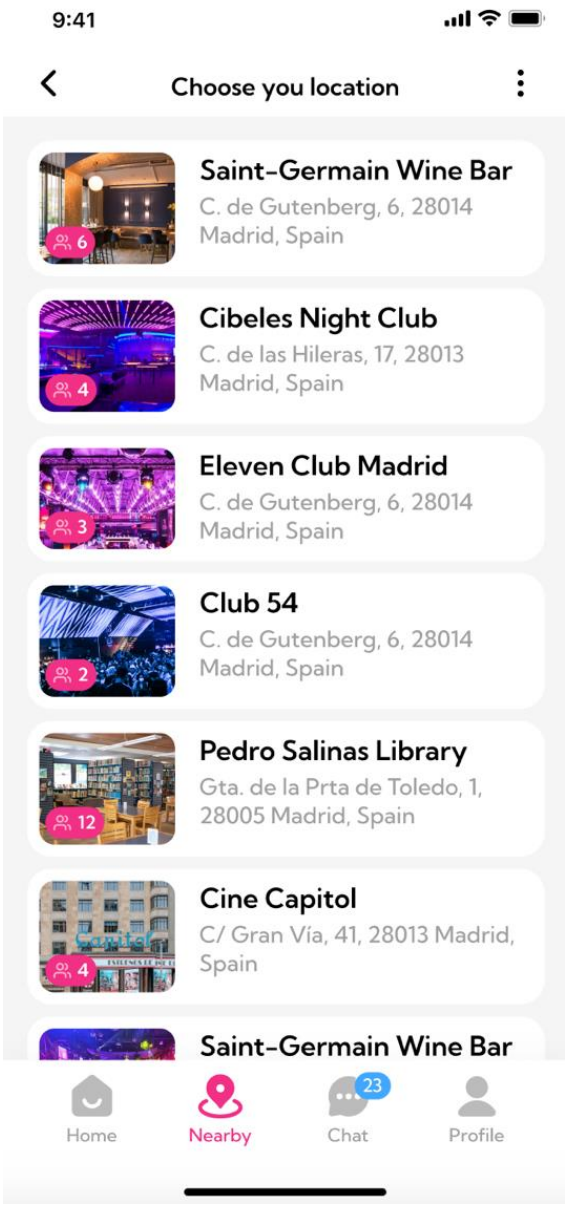

Рисунок 3.2 – Вибір локації (Find Place)

Екран з інформацією про місце для відпочинку наведено на рис.3.3:

Цей екран надає детальну інформацію про вибране місце.

Опис місця: Містить основну інформацію про місце: назву, адресу, короткий опис. Користувачі можуть бачити фотографії місця та прочитати відгуки.

Чек-ін функція: Користувачі можуть "зачекінитися" в місці, натиснувши кнопку "Check in". Після чек-іну користувач може залишити місце, натиснувши кнопку "Leave this place".

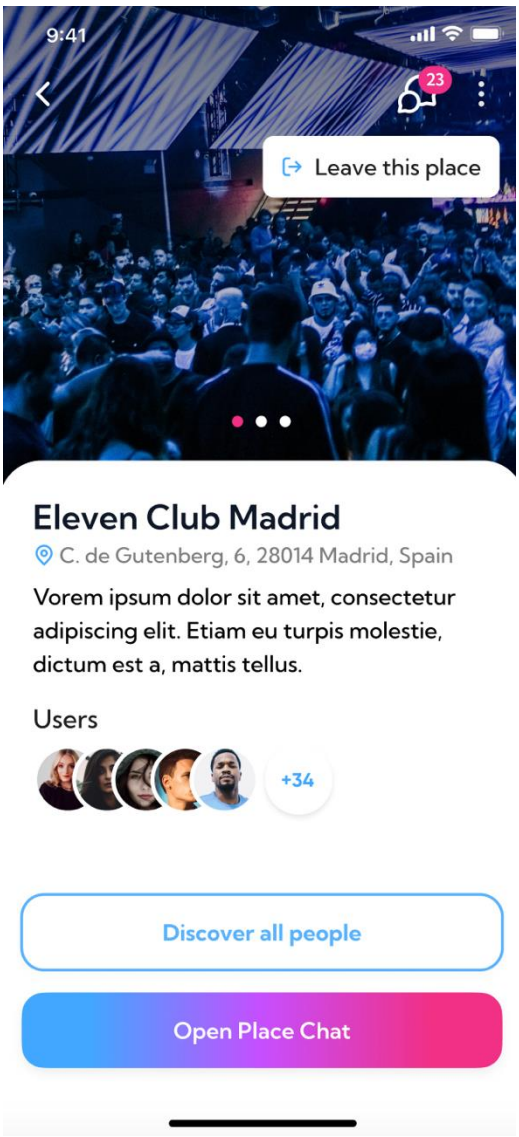

Рисунок 3.3 – Деталі місця (Place Details)

Екран з користувачами додатку які знаходяться у місці де було здійснено чек-ін наведено на рис.3.4:

Даний екран дозволяє користувачам переглядати профілі інших користувачів, що зараз знаходяться в місці в якому було здійснено чек-ін.

Список користувачів: Відображає аватари та імена користувачів, які "відкриті до чату". Можливість перегляду профілів та початку чату.

Фільтри: Користувачі можуть налаштовувати фільтри для пошуку відповідних профілів за інтересами, віком та кількістю фото.

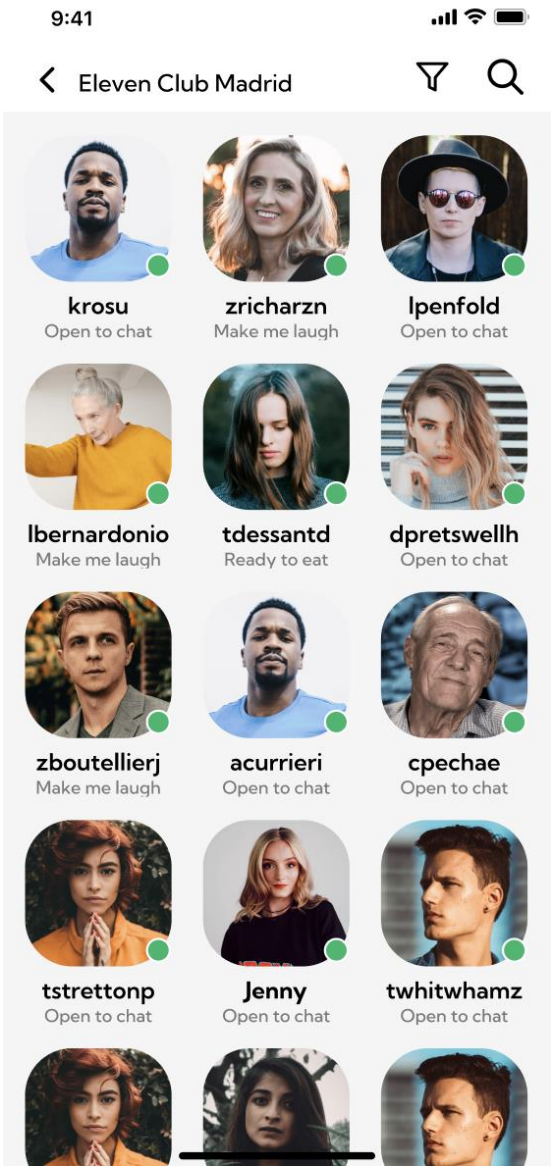

Рисунок 3.4 – Користувачі місця (Place Users)

Чат місця наведено на рис.3.5

Екран чату дозволяє користувачам взаємодіяти один з одним у реальному часі.

Основний блок чату: Містить повідомлення користувачів. Можливість надсилання повідомлень та реакцій.

Навігація: Користувач може повернутися до попереднього екрану або скористатися пошуком по чату.

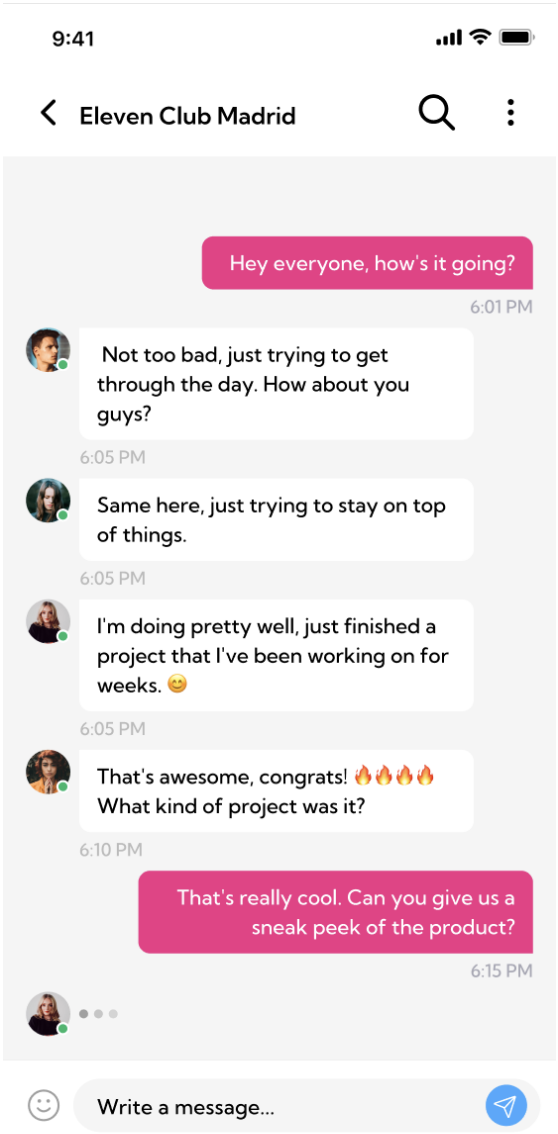

Рисунок 3.5 – Чат місця (Place Chat)

Налаштування профілю наведено на рис.3.6

Екран налаштувань профілю дозволяє користувачам управляти своєю особистою інформацією.

Секції налаштувань: Містить розділи для редагування особистої інформації, статусу, зміни фотографій та пароля. Користувачі можуть змінювати свої дані та налаштування безпеки.

Вихід з профілю: Кнопка для виходу з профілю або видалення профілю.

Цей додаток є зручним інструментом для пошуку та взаємодії з місцями та людьми, забезпечуючи користувачів усіма необхідними функціями для планування своїх соціальних активностей. Головна мета інтерфейсу мобільного додатку

підбору місць для відпочинку з функцією знайомств – забезпечити зручний, інтуїтивно зрозумілий та привабливий користувацький досвід, що сприятиме успішному плануванню відпочинку та налагодженню нових знайомств.

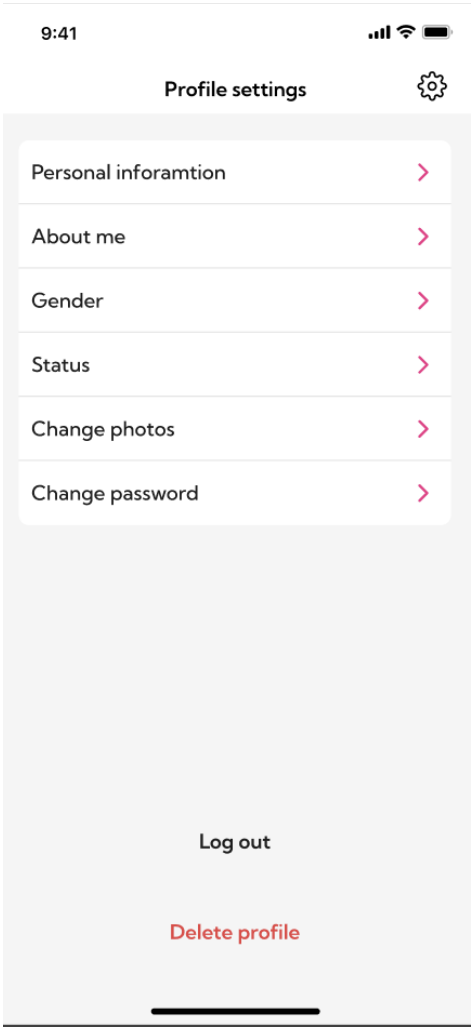

Рисунок 3.6 - налаштування профілю (Profile Settings)

3.4 Тестування мобільного додатку підбору місць для відпочинку та аналіз результатів

При тестуванні додатку для підбору місць для відпочинку необхідно було перевірити коректну роботу всіх модулів. Метою було виявлення та виправлення помилок та неточностей у роботі програми, щоб забезпечити правильність роботи функціоналу та загальну надійність додатку. У процесі тестування здійснювалися наступні кроки:

1. Вхід до додатку з правильними даними для входу.

   Вихідні дані: коректний логін та пароль;

   Очікуваний результат: успішний вхід до додатку;

   Результат входу до додатку з правильними даними:

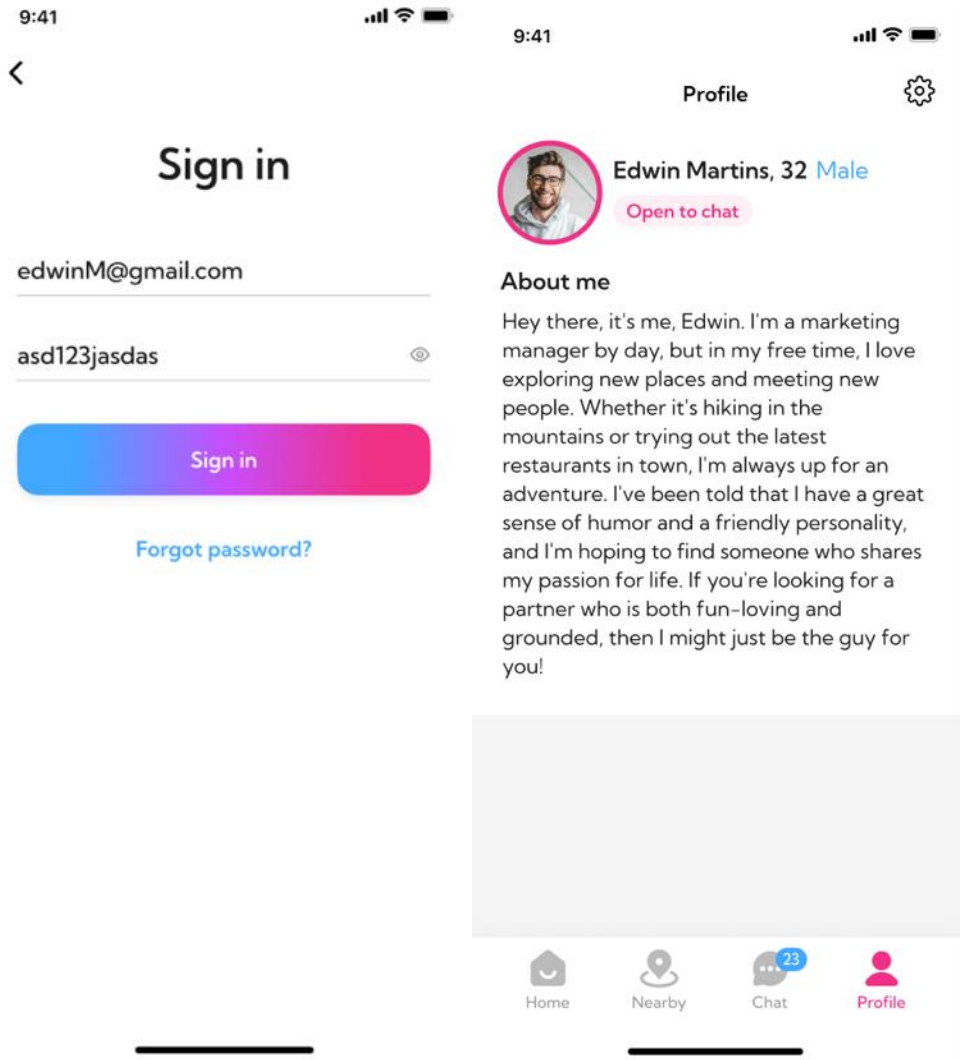

Рисунок 3.7 – результат тестування авторизації до додатку з правильними даними

2. Вхід до додатку з неправильними даними для входу.

   Вихідні дані: хибні логін та пароль;

   Очікуваний результат: відсутність доступу;

   Результат наведено на рисунку 3.8:

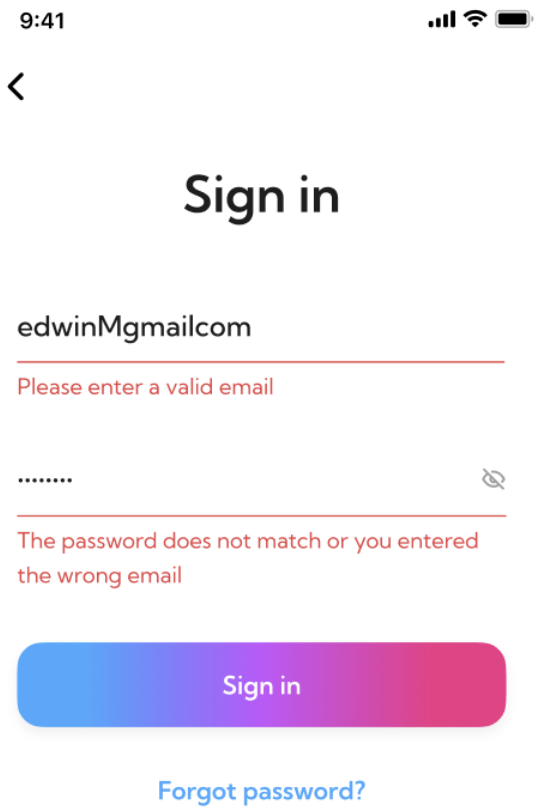

Рисунок 3.8 – Результат тестування авторизації з неправильними даними

3. Пошук місць для відпочинку за геолокацією користувача.

   Вихідні дані: Дозвіл від користувача на використання його геолокації.

   Очікуваний результат: відображення списку місць для відпочинку які знаходяться найближче до геолокації користувача;

   Результат пошуку місць для відпочинку за геолокацією користувача зображено на рисунку 3.9:

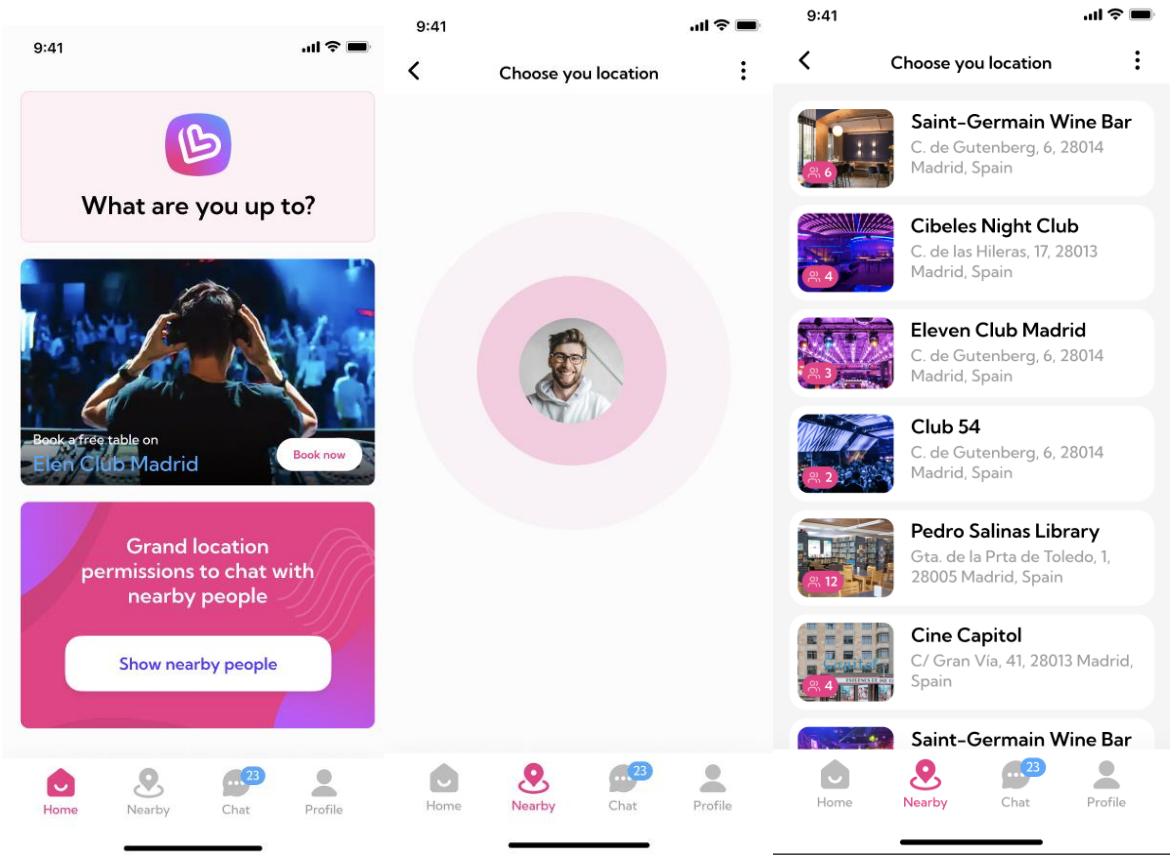

Рисунок 3.9 – результат тестування додавання дозволу додатку на використання геолокації користувача

4. Отримання додаткової інформації про місце для відпочинку за допомогою функції «Check-in»

   Вихідні дані: обране місце для відпочинку користувачем;

   Очікуваний результат: відображення додаткової інформації про місце для відпочинку; можливість відкрити чат з користувачами додатку які також знаходять в цьому ж місці, можливість перегляду профілей користувачів які знаходять в цьому ж місці.

   Результат застосування функції «Check-in» показано на рисунку 3.10:

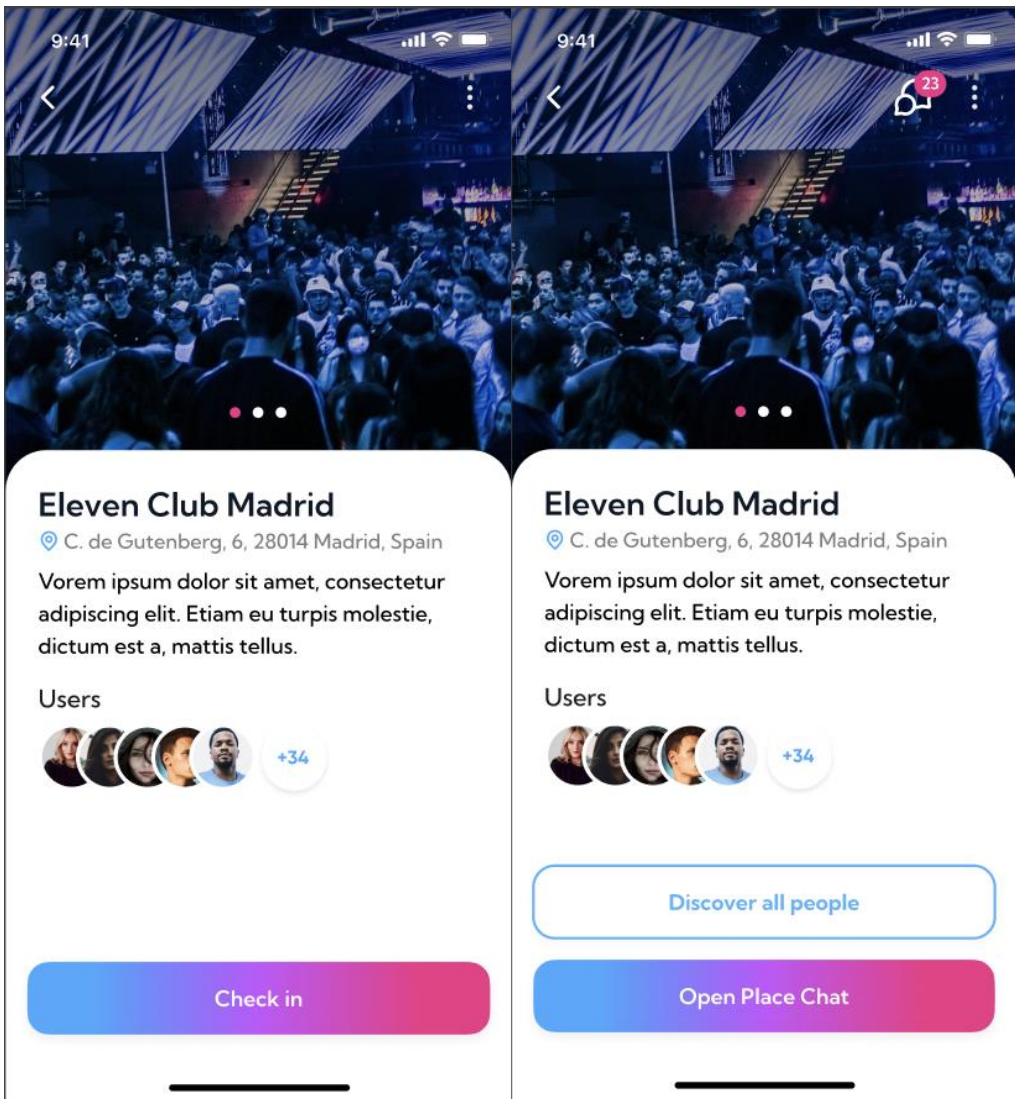

Рисунок 3.10 – Результат тестування функції «Check-in»

5. Початок чату з користувачами в обраному місці

   Вихідні дані: Функція для входу в чат обраного місця;

   Очікуваний результат: відображення спільного чату користувачів які знаходять в обраному місці;

   Результат виконання функції для входу в чат обраного місця зображено на рисунку 3.11:

Розроблений додаток має функціонал, наведений в таблиці 3.1, який було порівняно із функціоналом найпопулярніших мобільних застосунків для підбору місць для відпочинку, таких як GoogleMaps, TripAdvisor.

Таблиця 3.1 – Порівняльна характеристика розробленого ПЗ

Назва програми	Пошук місць для відпочинку поблизу	Налаштування профілю	Функція чату з користувачами в обраних місцях для відпочинку	Функція бронювання місця для відпочинку
GoogleMaps,	+/-	-	-	-
TripAdvisor	+	-	-	+/-
Розроблений додаток	+	+	+	+

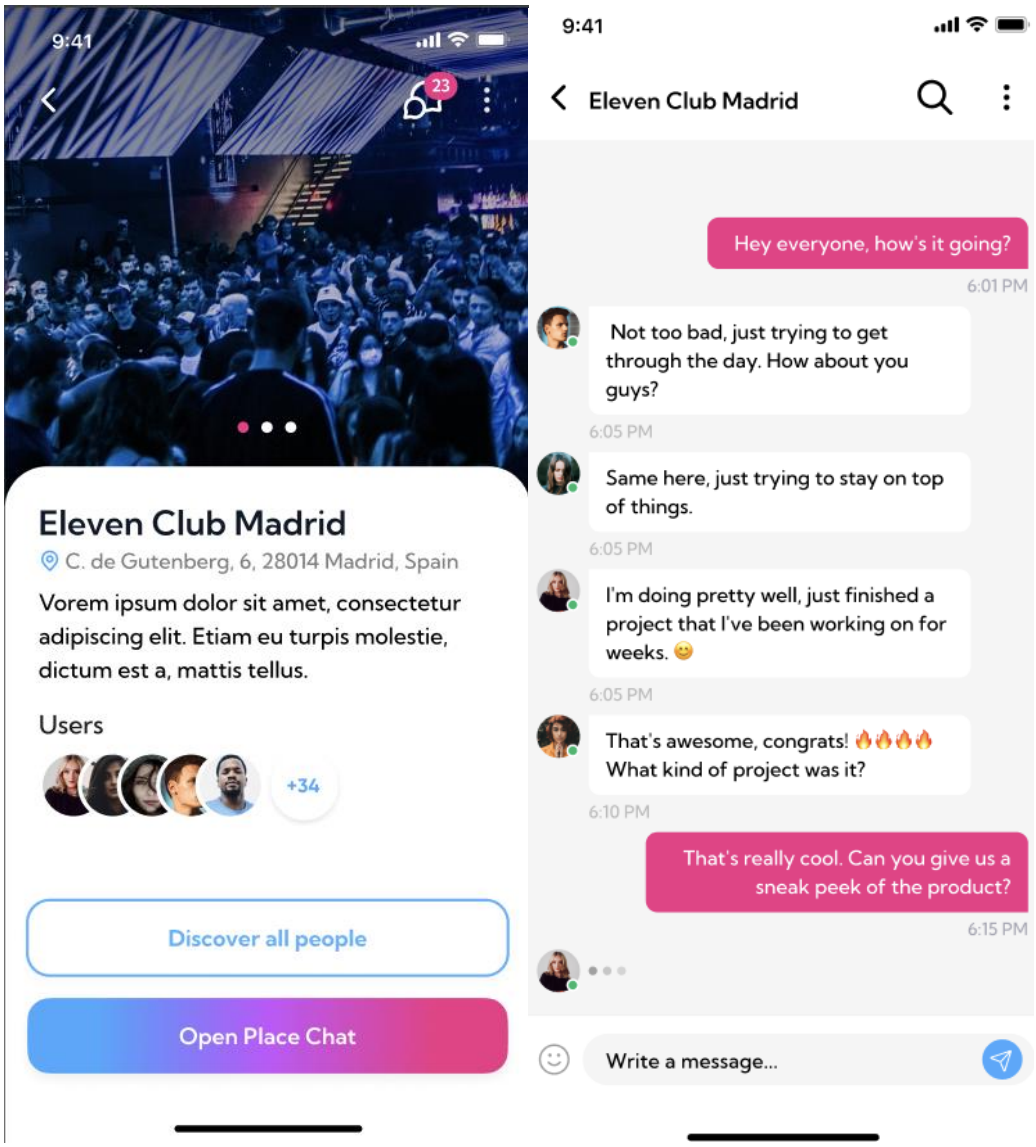

Рисунок 3.11 – результат тестування функції початку чату з користувачами в обраному місці

3.5 Висновок

Виконано обгрунтування вибору мови програмування та середовища розробки для мобільного додатку підбору місць для відпочинку. Розглянуто різні мови, такі як Kotlin, Java, Swift, та Objective C, а також фреймворки, включаючи Ionic, Flutter, NativeScript та React Native. Було вибрано Kotlin для геомодуля та React Native для клієнтської частини через їхні переваги у швидкості, багатоплатформності та зручності використання. Це рішення забезпечує простоту, ефективність та надійність, що відрізняє додаток від існуючих рішень на ринку.

Виконано реалізацію модулю отриманння геоданих, що передбачає точне отримання даних про місцезнаходження користувача. Це було здійснено за допомогою створення LocationListener та використання LocationManager для оновлення геопозиції. Додаток автоматично зупиняє оновлення даних при завершенні роботи для збереження ресурсів пристрою. Це рішення дозволяє забезпечити високу точність, ефективність та економічність, що робить додаток зручним та надійним для користувачів.

Здійснено реалізацію інтерфейсу мобільного додатку для підбору місць для відпочинку. Було створено кілька ключових екранів: головний екран, екран вибору місць, екран з інформацією про місце, екран користувачів, які знаходяться в місці, де було здійснено чек-ін, екран чату та екран налаштувань профілю.
Кожен екран був детально описаний, включаючи основні функції та елементи, що забезпечують зручність користування. Це дозволило створити інтуїтивно зрозумілий і привабливий інтерфейс, який забезпечує користувачів усіма необхідними інструментами для планування відпочинку та взаємодії з іншими людьми. Реалізація інтерфейсу враховує сучасні тенденції UX/UI дизайну та включає всі необхідні функції для зручного користування додатком,

Виконано тестування розробленого програмного додатку. Під час тестування додатку було перевірено коректну роботу всіх модулів. Тестування охоплювало перевірку процесів авторизації, пошуку місць за геолокацією, отримання додаткової інформації про місця та взаємодії з іншими користувачами

через функцію чату. Метою тестування було виявлення та виправлення помилок та неточностей у роботі програми, щоб забезпечити правильність роботи функціоналу та загальну надійність додатку. Кожен з етапів тестування був спрямований на виявлення потенційних проблем та їх швидке виправлення. Це дозволило забезпечити високу якість та надійність роботи мобільного додатку.

Результати тестування показали, що розроблений додаток має функціонал, який є розширеним відносно існуючих сучасних рішень, таких як GoogleMaps та TripAdvisor. Наведено таблицю, яка демонструє порівняльну характеристику функціоналу розробленого ПЗ із цими популярними додатками. Відзначено, що розроблений додаток пропонує більш широкий спектр можливостей, охоплюючи налаштування профілю та функцію чату з користувачами в обраних місцях для відпочинку.

ВИСНОВКИ

У результаті написання бакалаврської дипломної роботи розроблено мобільний додаток підбору місць для відпочинку.

Виконано обґрунтування актуальності розробки зазначеного програмного забезпечення. Його використання людьми, які бажають ефективно використовувати свій вільний час та займатись ефективною організацією відпочинку.

Проведено детальний аналіз існуючих мобільних додатків, таких як Google Maps та TripAdvisor. Хоча ці додатки пропонують широкі можливості, вони не завжди зручні для швидкого пошуку інформації про місця для відпочинку. Було виявлено проблему у вигляді недостатньої кількості функціоналу. Виконано постановку задачі. Розроблений додаток зосереджений на ефективності надання інформації та персоналізованих рекомендаціях, забезпечує зручний інтерфейс та актуальні дані та має додатковий функціонал, який дозволяє спілкуватись з іншими користувачами додатку в режимі чату, що знаходяться в тому ж місці для відпочинку.

Розроблено загальний алгоритм роботи додатку для підбору місць для відпочинку, а також загальну структурну схему функціонування мобільного додатку, життєвий цикл додатку, UML-діаграму класів мобільного додатку підбору місць для відпочинку. Створення цих елементів дозволило чітко визначити, як додаток повинен функціонувати, забезпечуючи зручність користування та ефективність виконання основних задач. Загальна структурна схема та життєвий цикл додатку допомогли визначити послідовність виконання операцій та взаємодію різних компонентів системи. UML-діаграма класів показала структуру даних та їх взаємозв'язки. Це краще і відрізняється від існуючих рішень тим, що пропонує більш структурований та зручний підхід до організації даних і процесів, що сприяє підвищенню ефективності роботи додатку та покращенню користувацького досвіду.

Наведена діаграма прецедентів допомогла чітко визначити всі можливі взаємодії користувача з системою, відображаючи основні функції та процеси додатку, показавши покроковий процес виконання задач, забезпечуючи ясність та структурованість роботи додатку. Це краще і відрізняється від існуючих рішень тим, що надає більш детальне та зрозуміле представлення процесів, що полегшує розуміння та використання додатку для кінцевих користувачів, а також покращує його технічну реалізацію та підтримку.

Здійснено обґрунтування вибору мови програмування та фреймворку. При цьому використано мову програмування JavaScript, фреймворк ReactNative, середовище розробки VisualStudioCode, Firebase для управління базою даних, забезпечення можливості реєстрації без написання повноцінного бекенду, Google API для роботи з геоданими та отриманням місць поблизу користувача.

Структуровано та описано реалізацію функціоналу програмного забезпечення. Проведено тестування програмного забезпечення на відповідність поставленим вимогам. Відповідно до результатів тестування функціонал програмного забезпечення розширено відповідно до поставлених вимог, результати роботи розробленого функціоналу відповідають очікуванням.

Враховуючи вище наведене, можемо констатувати, що мету роботи досягнено. Усі поставлені задачі виконано у повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Арсенюк І. Р., Оптимізація використання ресурсів акумулятора при відстеження геолокації в мобільному додатку react native для операційних систем Android та IOS./ І. Р. Арсенюк, В. В. Сабашна // Молодь в науці: дослідження, проблеми, перспективи (МН-2024), 15.10.23 – 20.05.24 : збірник матеріалів. – Вінниця: ВНТУ, 2024. – Режим доступу: https://conferences.vntu.edu.ua/index.php /mn/mn2024/paper/viewFile/21232/17600
2. Додаток підбору місць для відпочинку «Google Maps» [Електронний ресурс] - режим доступу: https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en_US. Дата звернення: 18.05.2024
3. Додаток підбору місць для відпочинку «Trip Advisor» [Електронний ресурс] - режим доступу: https://www.tripadvisor.com/app Дата звернення: 18.05.2024
4. Configuring App with Schemes [Електронний ресурс] - режим доступу: https://medium.com/nsistanbul/configuring-app-with-schemes-ee4fc6569d2b. Дата звернення: 18.05.2024
5. React Native Component Lifecycle [Електронний ресурс] - режим доступу: https://www.netguru.com/blog/react-nativelifecycle#:~:text=A%20component's%20lifecycle%20can%20be,not%20needed%20and%20gets%20unmounted. Дата звернення: 17.05.2024
6. Guide to app architecture [Електронний ресурс] - режим доступу: https://developer.android.com/topic/architecture Дата звернення: 19.05.2024
7. Використання Kotlin для розробки android [Електронний ресурс] - режим доступу: https://kotlinlang.org/docs/reference/android-overview.html. Дата звернення: 20.05.2024
8. Swift [Електронний ресурс] - режим доступу: https://uk.wikipedia.org/wiki/SWIFT. Дата звернення: 20.05.2024
9. Objective-C [Електронний ресурс] - режим доступу: https://uk.wikipedia.org/wiki/Objective-C Дата звернення: 21.05.2024

10. Заславський В.А. Принцип різнотипності при дослідженні складних систем з високою ціною відмови / В.А. Заславський // Вісник Київського університету. - 2006. - Вип. 12-13. - С.11-14.
11. What Does Java Mean. 2017 [Електронний ресурс] - режим доступу: https://www.techopedia.com/definition/3927/java. Дата звернення: 21.05.2024
12. Вікіпедія [Елекронний ресурс]: [Веб-сайт]- Електронні дані. Дата звернення: 21.05.2024
13. Ionic [Електронний ресурс] - режим доступу: https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework). Дата звернення: 21.05.2024
14. Flutter [Електронний ресурс] - режим доступу: https://uk.wikipedia.org/wiki/Flutter. Дата звернення: 21.05.2024
15. NativeScript [Електронний ресурс] - режим доступу: https://en.wikipedia.org/wiki/NativeScript. Дата звернення: 22.05.2024
16. React Native [Електронний ресурс] - редим доступу: https://en.wikipedia.org/wiki/React_Native Дата звернення: 22.05.2024
17. Android Studio [Електронний ресурс] - режим доступу: https://uk.wikipedia.org/wiki/Android_Studio Дата звернення: 22.05.2024
18. Платформа «Eclipse» [Елекронний ресурс]: [Веб-сайт]- Елекртонні дані. Режим доступу : http://www.eclipse-3.narod.ru - назва з екрану. Дата звернення: 21.05.2024
19. Request location updates [Електронний ресурс] - режим доступу: https://developer.android.com/training/location/request-updates. Дата звернення: 22.05.2024
20. Солоний М., Арсенюк І. Розоробка мобільного додатку інтернет-магазину за допомогою React Native // Матеріали XLVIII науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії ВНТУ, Вінниця, 2019. URL: https://conferences.vntu.edu.ua/index.php/allfitki/all-fitki-2019/paper/view/7012/5949

ДОДАТОК А

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Мобільний додаток підбору місць для відпочинку

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 97,36% Схожість 2,64%

Аналіз звіту подібності (відмітити потрібне):

✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку Озеранський В.С.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Сабашна В.В.

Керівник роботи Арсенюк І.Р.

ДОДАТОК Б
ЛІСТИНГ ПРОГРАМИ

```
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ptn.app">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission
android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission
android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
    <uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission
android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission
android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission
android:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
    <permission
android:name="android.permission.ACCESS_FINE_LOCATION"
android:protectionLevel="dangerous"/>
    <permission
android:name="android.permission.ACCESS_COARSE_LOCATION"
android:protectionLevel="dangerous" />
    <application
      android:name=".MainApplication"
      android:label="@string/app_name"
      android:icon="@mipmap/ic_launcher"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:allowBackup="false"
      android:theme="@style/AppTheme">
      <activity
        android:name=".MainActivity"
        android:label="@string/app_name"

android:configChanges="keyboard|keyboardHidden|orientation|screenSiz
e"
```

```
        android:windowSoftInputMode="adjustResize"
        android:launchMode="singleTop">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category
android:name="android.intent.category.DEFAULT"/>
            <category
android:name="android.intent.category.BROWSABLE"/>
            <data android:host="ptn.page.link"
android:scheme="http"/>
            <data android:host="ptn.page.link"
android:scheme="https"/>
        </intent-filter>
      </activity>
      <activity

android:name="com.facebook.react.devsupport.DevSettingsActivity"
      />
      <uses-library
        android:name="org.apache.http.legacy"
        android:required="false" />
      <meta-data
          android:name="com.google.android.geo.API_KEY"
          android:value="AIzaSyB59AcLcuf2rrghWwEDC1Ba6z5ltIuZfhw"/>

        <!-- <meta-data
android:name="com.google.firebase.messaging.default_notification_cha
nnel_id" android:value="@string/default_notification_channel_id"/> -
->

        <receiver android:name=".timer.AlarmReceiver"
            android:directBootAware="true"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.BOOT_COMPLETED" />
                <action
android:name="android.intent.action.LOCKED_BOOT_COMPLETED" />
            </intent-filter>
        </receiver>

        <receiver android:name=".timer.SyncReceiver"
            android:directBootAware="true"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
```

```
                <action
android:name="android.intent.action.BOOT_COMPLETED" />
                <action
android:name="android.intent.action.LOCKED_BOOT_COMPLETED" />
            </intent-filter>
        </receiver>
        <service android:name=".timer.SyncHeadless"
            android:label="SyncHeadless" />
        <service android:name=".timer.AlarmHeadless"
            android:label="AlarmHeadless" />
            <service
android:name="io.invertase.firebase.messaging.RNFirebaseMessagingSer
vice">
            <intent-filter>
                <action
android:name="com.google.firebase.MESSAGING_EVENT" />
            </intent-filter>
        </service>
        <service
android:name="io.invertase.firebase.messaging.RNFirebaseInstanceIdSe
rvice">
            <intent-filter>
                <action
android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
            </intent-filter>
        </service>
        <service
android:name="io.invertase.firebase.messaging.RNFirebaseBackgroundMe
ssagingService" />

    </application>

</manifest>
}

rootProject.name = 'safeamigos'
include ':react-native-geolocation-service'
project(':react-native-geolocation-service').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-geolocation-
service/android')
include ':react-native-exit-app'
project(':react-native-exit-app').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-exit-app/android')
include ':react-native-image-crop-picker'
project(':react-native-image-crop-picker').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-image-crop-
picker/android')
include ':react-native-splash-screen'
project(':react-native-splash-screen').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-splash-screen/android')
include ':react-native-camera'
project(':react-native-camera').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-camera/android')
include ':react-native-system-setting'
project(':react-native-system-setting').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-system-
setting/android')
```

```
include ':react-native-maps'
project(':react-native-maps').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-maps/lib/android')
include ':react-native-sound'
project(':react-native-sound').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-sound/android')
include ':react-native-geocoder'
project(':react-native-geocoder').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-geocoder/android')
include ':react-native-imei'
project(':react-native-imei').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-imei/android')
include ':@react-native-community_async-storage'
project(':@react-native-community_async-storage').projectDir = new
File(rootProject.projectDir, '../node_modules/@react-native-community/async-
storage/android')
include ':react-native-vector-icons'
project(':react-native-vector-icons').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-vector-icons/android')
include ':react-native-device-info'
project(':react-native-device-info').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-device-info/android')
include ':react-native-linear-gradient'
project(':react-native-linear-gradient').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-linear-
gradient/android')
include ':react-native-gesture-handler'
project(':react-native-gesture-handler').projectDir = new
File(rootProject.projectDir, '../node_modules/react-native-gesture-
handler/android')
include ':@react-native-community_netinfo'
project(':@react-native-community_netinfo').projectDir = new
File(rootProject.projectDir, '../node_modules/@react-native-
community/netinfo/android')
include ':react-native-firebase'
project(':react-native-firebase').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-firebase/android')

include ':app'
        buildscript {
    ext {
        buildToolsVersion = "28.0.3"
        minSdkVersion = 19
        compileSdkVersion = 28
        targetSdkVersion = 28
        supportLibVersion = "28.0.0"
    }
    repositories {
        google()
        jcenter()
        maven {
            url 'https://maven.fabric.io/public'
        }
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.4.1'
        classpath 'com.google.gms:google-services:4.2.0'
        classpath 'com.google.firebase:perf-plugin:1.2.1'
        classpath 'io.fabric.tools:gradle:1.28.1'
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```

```
allprojects {
    repositories {
        mavenLocal()
        google()
        jcenter()
        maven {
            // All of React Native (JS, Obj-C sources, Android binaries) is
installed from npm
            url "$rootDir/../node_modules/react-native/android"
        }
        maven { url 'https://maven.google.com' }
        maven { url "https://jitpack.io" }
    }
}

subprojects {
    afterEvaluate {project ->
        if (project.hasProperty("android")) {
            android {
                compileSdkVersion rootProject.ext.compileSdkVersion
                buildToolsVersion rootProject.ext.buildToolsVersion
            }
        }
    }
}
        import NetInfo from "@react-native-community/netinfo";
import { Alert, AppState } from 'react-native';
export default class ConnectionService {
  static alertOpened = false;
  constructor() {
    NetInfo.isConnected.fetch().then().done((isConnected) => {
      NetInfo.isConnected.addEventListener('connectionChange',
this._handleConnectivityChange);
    });
  }

  static async _checkConnection() {
    try {
      const isConnected = await NetInfo.isConnected.fetch();
      if (!isConnected) {
        if (AppState.currentState !== 'active') return;
        if (ConnectionService.alertOpened) return false;
        this.alertOpened = true;
        Alert.alert(
          'Internet connection error',
          'Please check your internet connection and try again',
          [
            {
              text: 'Ok', onPress: () => {
                this.alertOpened = false;
              }
            }
          ],
          { cancelable: false }
        )
        return false
      } else {
        return true;
      }
    } catch (err) { console.log(err) }
  }

  _handleConnectivityChange = isConnected => {
    if (!isConnected) {
```

```
    }
  };

}
                import Permissions from 'react-native-permissions';
import { PermissionsAndroid, Platform } from 'react-native';

export default class PermissionsService {
  constructor() { }

  static async checkPermissions(permissionType) {
    try {
      const status = await Permissions.check(permissionType);
      if(status === 'authorized') {
        return true;
      } else {
        const res = await Permissions.request(permissionType);
        return res === 'authorized';
      }
    } catch (err) {
      console.log(err)
      return false;
    }
  }

  static async checkPhoneStatePermissions() {
    try {
      const status = await PermissionsAndroid.request(
        PermissionsAndroid.PERMISSIONS.READ_PHONE_STATE,
      );
      return status === PermissionsAndroid.RESULTS.GRANTED;
    } catch (err) {
      console.log(err)
    }
  }

  static async checkDefaultPermissions() {
    let userResponse;
    let granted = true;
    try {
      if (Platform.OS === "android") {
        userResponse = await PermissionsAndroid.requestMultiple([
          PermissionsAndroid.PERMISSIONS.READ_PHONE_STATE,
          PermissionsAndroid.PERMISSIONS.CAMERA,
          PermissionsAndroid.PERMISSIONS.ACCESS_FINE_LOCATION
        ]);
      } else {
        const permissions = [
          await this.checkPermissions('location'),
          await this.checkPermissions('camera'),
          await this.checkPermissions('photo')
        ];
        permissions.forEach(p => granted = granted && p);
        return Promise.resolve(granted)
      }
      Object.values(userResponse).forEach((value) => (granted = granted && (value
=== 'granted' || value === 'authorized')));
      return Promise.resolve(granted);
    } catch (err) {
      return Promise.resolve(false);
    }
  }
}
             import Geocoder from 'react-native-geocoder';
```

```
import ConnectionService from '../ConnectionService';

Geocoder.fallbackToGoogle('AIzaSyB59AcLcuf2rrghWwEDC1Ba6z5ltIuZfhw');

export default class GeocoderService {

  static async getCoordsInfo(location) {
    const { coords: { latitude: lat, longitude: lng } } = location;
    try {
      if (lat && lng) {
        const isConnected = await ConnectionService._checkConnection();
        const res = isConnected ? await Geocoder.geocodePosition({ lat, lng }) :
[];
        return res[0] || {};
      }
      return {};
    } catch (err) {
      return {};
    }
  }
}
import { Alert, Platform } from 'react-native';
import TimerNativeModule from "../../../NativeModules/TimerNativeModule";
import Geolocation from 'react-native-geolocation-service';
export class LocationService {
  static watchId;
  static alertHidden = true;
  static counter = 0;
  constructor() {
  }

  static getCurrentPosition() {
    return new Promise((resolve, reject) => {
      Geolocation.getCurrentPosition(location => {
        if(location.coords.accuracy > 300) {
          if(this.counter < 5) {
            ++this.counter;
            resolve(LocationService.getCurrentPosition());
          } else {
            this.counter = 0;
            resolve(null);
          }
        } else {
          this.counter = 0;
          resolve(location);
        }
        return;
      }, (err) => {
        if ((err.code === 2 || err.code === 5) && Platform.OS === 'android') {
          if (this.alertHidden) {
            this.alertHidden = false;
            Alert.alert(
              'Location error',
              'Please enable location',
              [
                {
                  text: 'Open location settings', onPress: () => {
                    return setTimeout(() => {
                      TimerNativeModule.openLocationSettings();
                      this.alertHidden = true;
                    })
                  }
                },
              ],
```

```
              { cancelable: false },
            );
          }
        }
        ++this.counter;
        if (this.counter === 5) {
          this.counter = 0;
          resolve(null)
        } else {
          setTimeout(() => {
            resolve(LocationService.getCurrentPosition());
          }, 500)
        }
      }, {
        enableHighAccuracy: true,
        showLocationDialog: true,
        maximumAge: 5000,
      })
    });
  }

  static watchPosition(callback) {
    LocationService.watchId = Geolocation.watchPosition(location => {
      if (location) {
        callback(location)
      }
    }, (err) => {
      console.log(err);
    }, {
      enableHighAccuracy: true,
      showLocationDialog: true,
      maximumAge: 5000,
    })
  }

  static clearWatch() {
    navigator.geolocation.clearWatch(LocationService.watchId);
  }
}

export default locationService = new LocationService();
       import AsyncStorage from '@react-native-community/async-storage';

const TIMER_EXPIRATION_DATE = 'TIMER_EXPIRATION_DATE';

export default {
    async setItem(key, value) {
        try {
            return await AsyncStorage.setItem(key, JSON.stringify(value));
        } catch (error) {
            console.error('AsyncStorage#setItem error: ' + error.message);
        }
    },
    async getItem(key) {
        return await AsyncStorage.getItem(key)
            .then((result) => {
                if (result) {
                    try {
                        result = JSON.parse(result);
                    } catch (e) {
                        console.error('AsyncStorage#getItem error deserializing
JSON for key: ' + key, e.message);
                    }
                }
```

```
                return result;
            });
    },
    async removeItem(key) {
        return await AsyncStorage.removeItem(key);
    },
    async saveTimerExpirationTimestamp(timestamp) {
        try {
            return await AsyncStorage.setItem(TIMER_EXPIRATION_DATE, '' +
timestamp);
        } catch (error) {
            console.error('AsyncStorage#setItem error: ' + error.message);
            return null;
        }
    },
    async getTimerExpirationTimestamp() {
        try {
            return parseInt(await AsyncStorage.getItem(TIMER_EXPIRATION_DATE))
        } catch (err) {
            return null
        }
    },
    async storeState(state)  {
        try {
          return await AsyncStorage.setItem( 'state', JSON.stringify(state));
         } catch (err) {
           return null
        }
     },

     async getAllData()  {
         try {
           return await  AsyncStorage.multiGet([TIMER_EXPIRATION_DATE, 'state'])
          } catch (err) {
            return null
         }
      }
}
```

ДОДАТОК В

ІЛЮСТРАТИВНА ЧАСТИНА

«МОБІЛЬНИЙ ДОДАТОК ПІДБОРУ МІСЦЬ ДЛЯ ВІДПОЧИНКУ»

Виконала: студентка 4 курсу, групи 1КН-20б
спеціальності 122 – Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)
Сабашна В.В.
(прізвище та ініціали)
Керівник: к.т.н., доцент каф.КН
Арсенюк І.Р.
(прізвище та ініціали)
«07» 06 2024 р.

Вінниця ВНТУ - 2024 рік

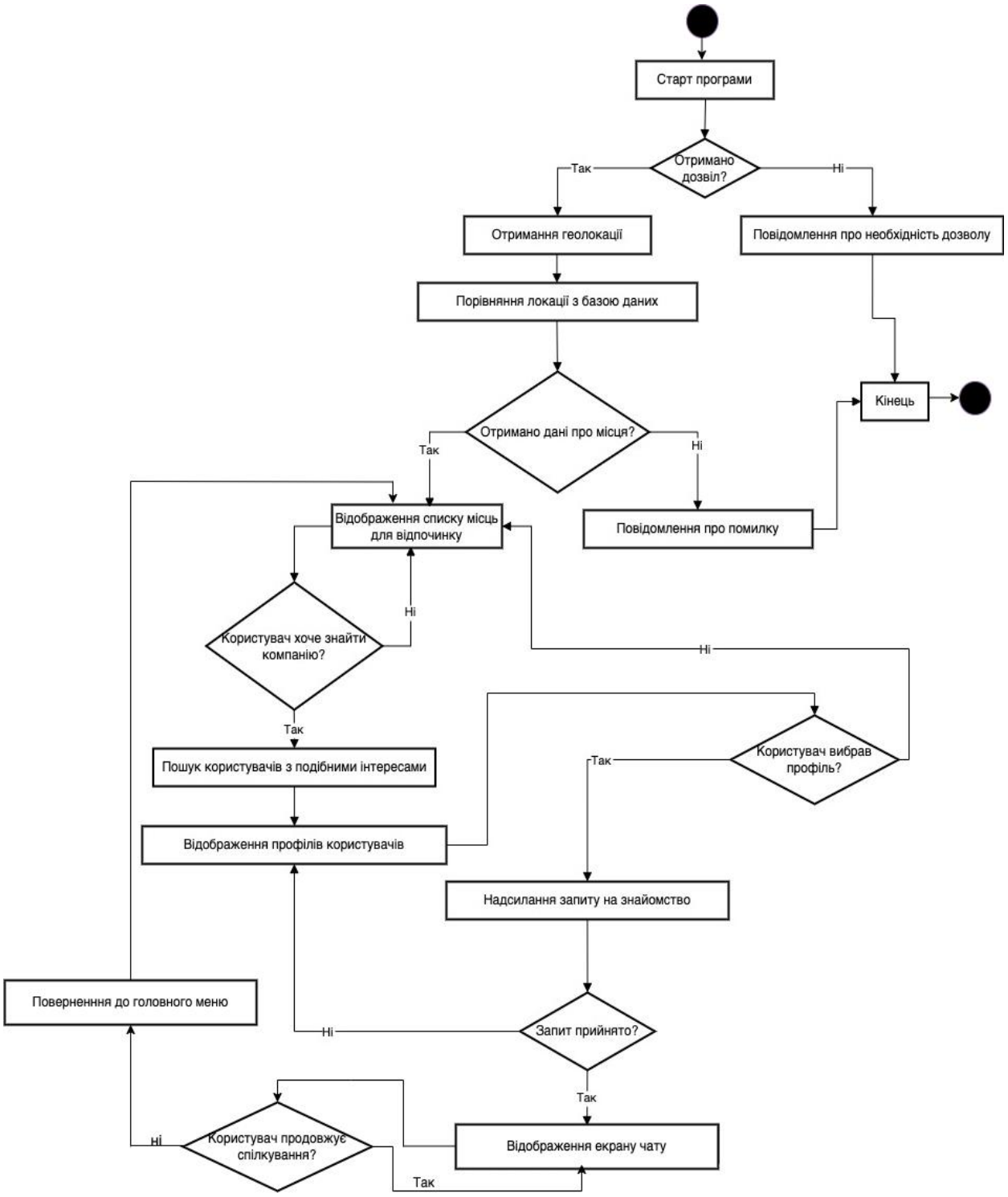

Рисунок В.1 – Схема загального алгоритму роботи мобільного додатку підбору місць для відпочинку

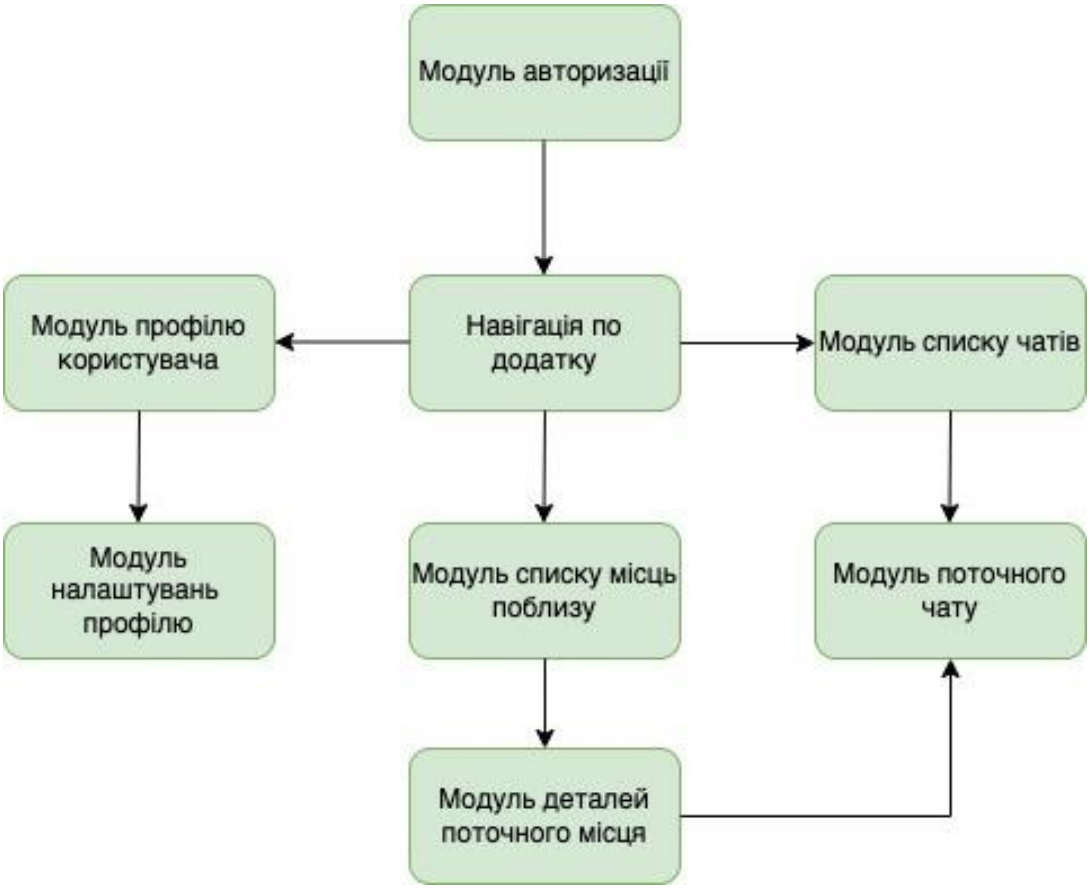

Рисунок В.2 – Структурна схему функціонування мобільного додатку підбору місць для відпочинку

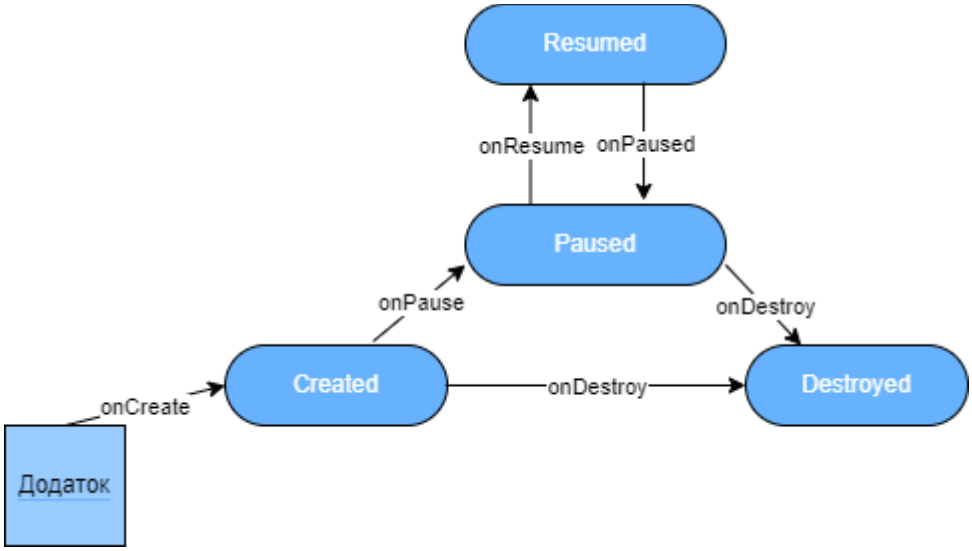

Рисунок В.3 – Схема життєвого циклу мобільного додатку підбору місць для відпочинку

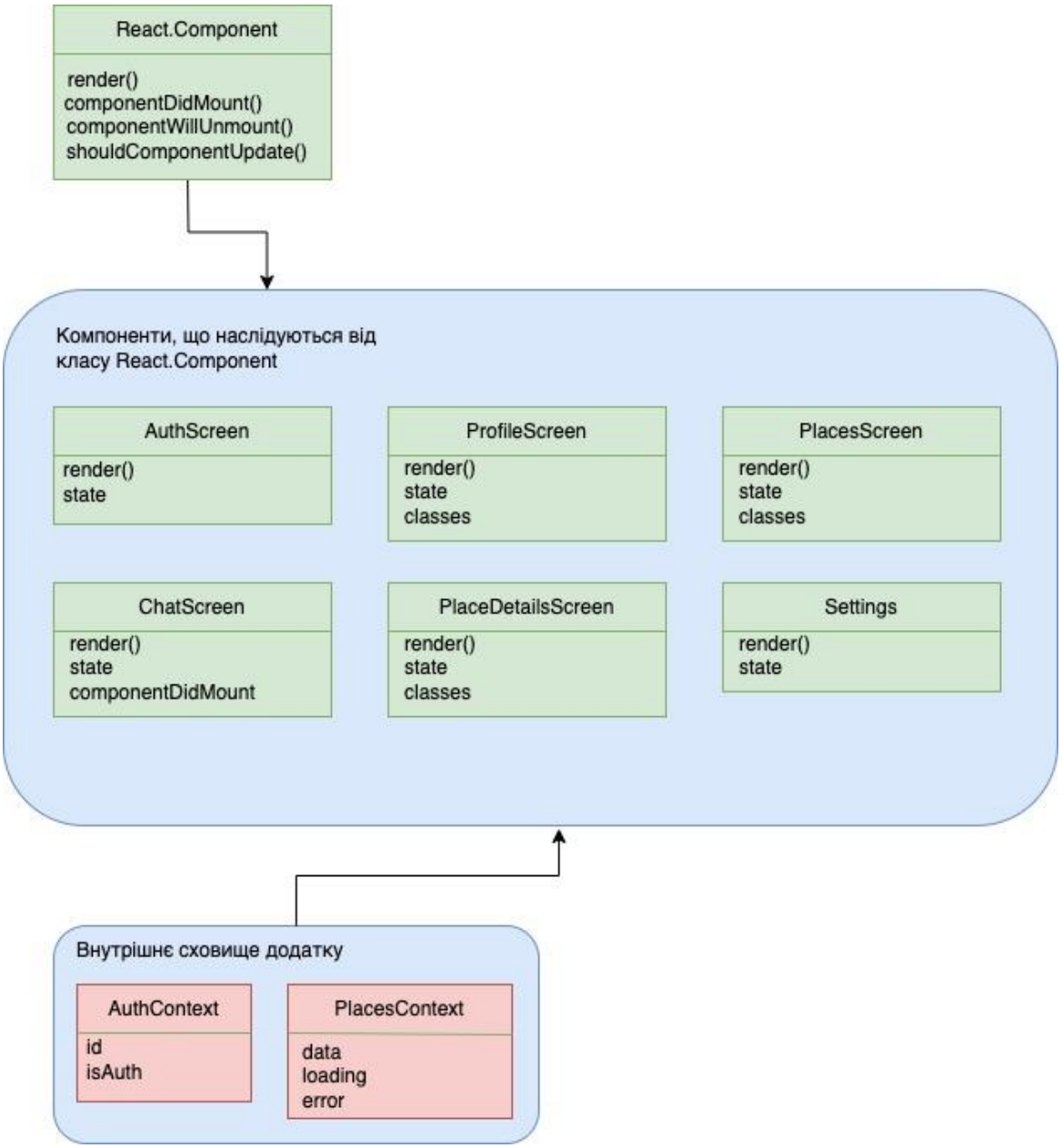

Рисунок В.4 – UML-діаграма класів мобільного додатку підбору місць для відпочинку

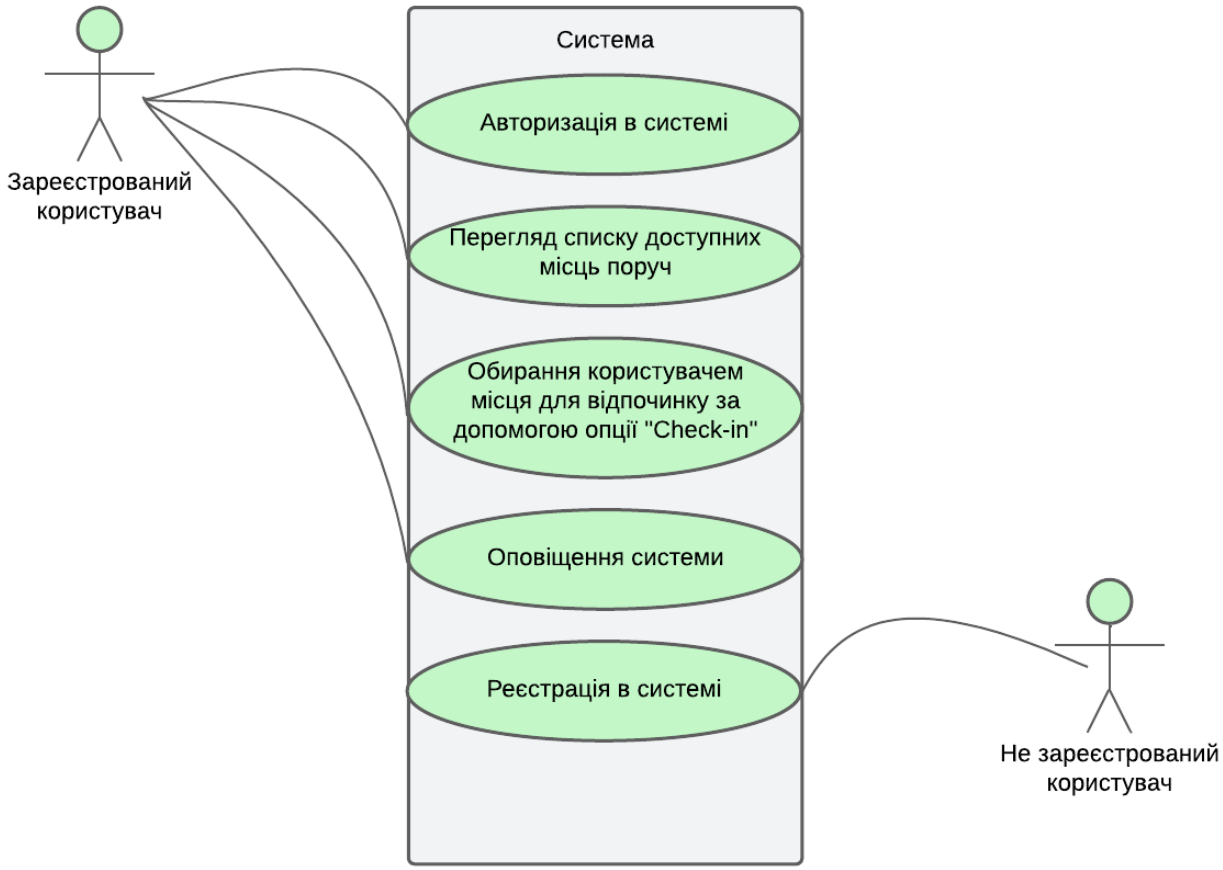

Рисунок В.5 – Діаграма прецедентів мобільного додатку підбору місць для відпочинку

ДОДАТОК Г

ІНСТРУКЦІЯ КОРИСТУВАЧА

1. Запустити на своєму телефоні файл «LinkApp.apk».
2. Натиснути кнопку «Sign in», щоб увійти на вже створений акаунт.
3. Натиснути кнопку «Create account», щоб створити акаунт.
4. Ввести свій логін та пароль у відповідні поля та натиснути кнопку «Увійти».

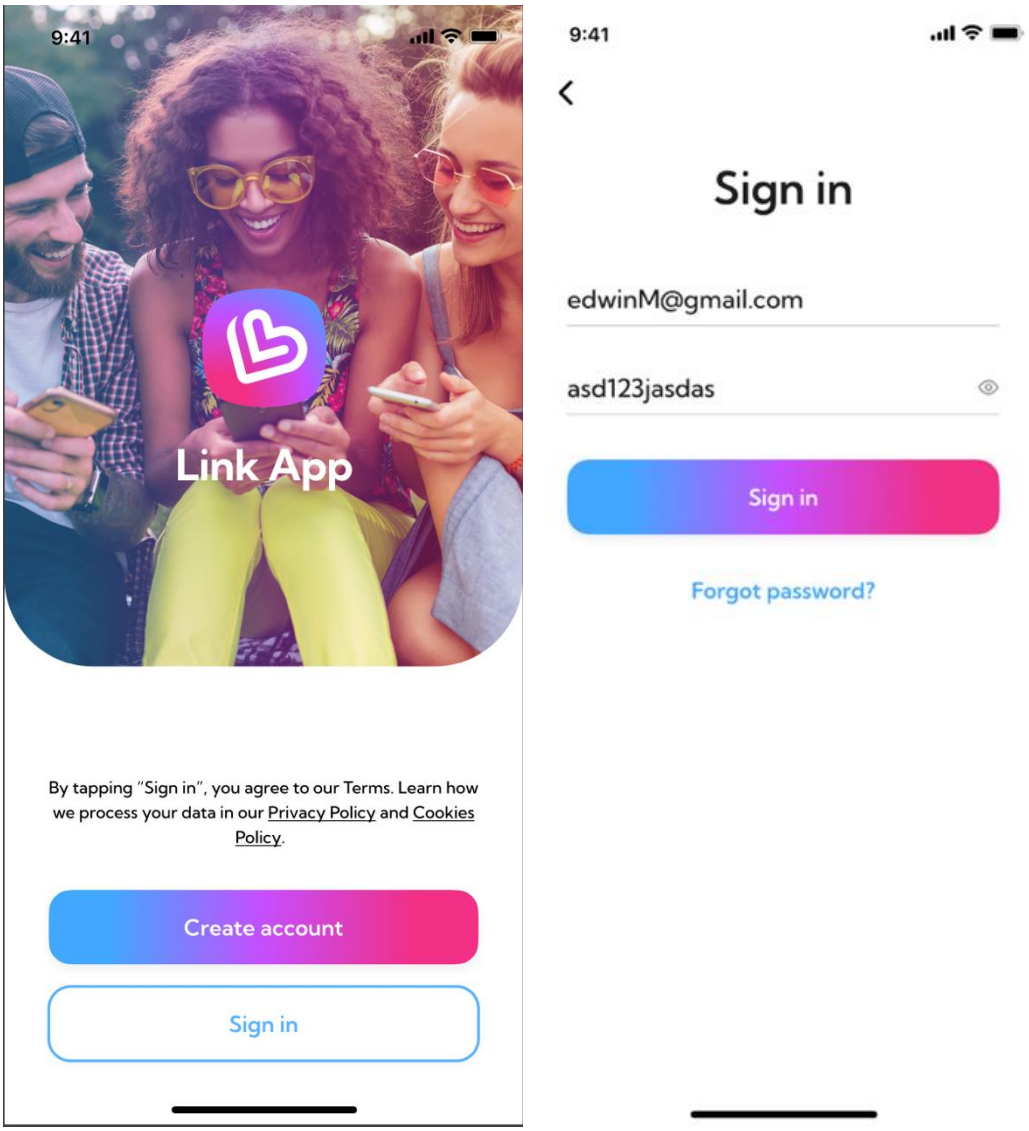

Рисунок Г.1 – Екран авторизації користувача

4.2. Якщо ви ще не зареєстрований користувач, тоді натисніть кнопку «Create account» та заповніть відповідні поля.

5. Для пошуку місць для відпочинку поблизу необхідно надати дозвіл на використання геолокації пристрою натиснувши кнопку «Show nearby people».

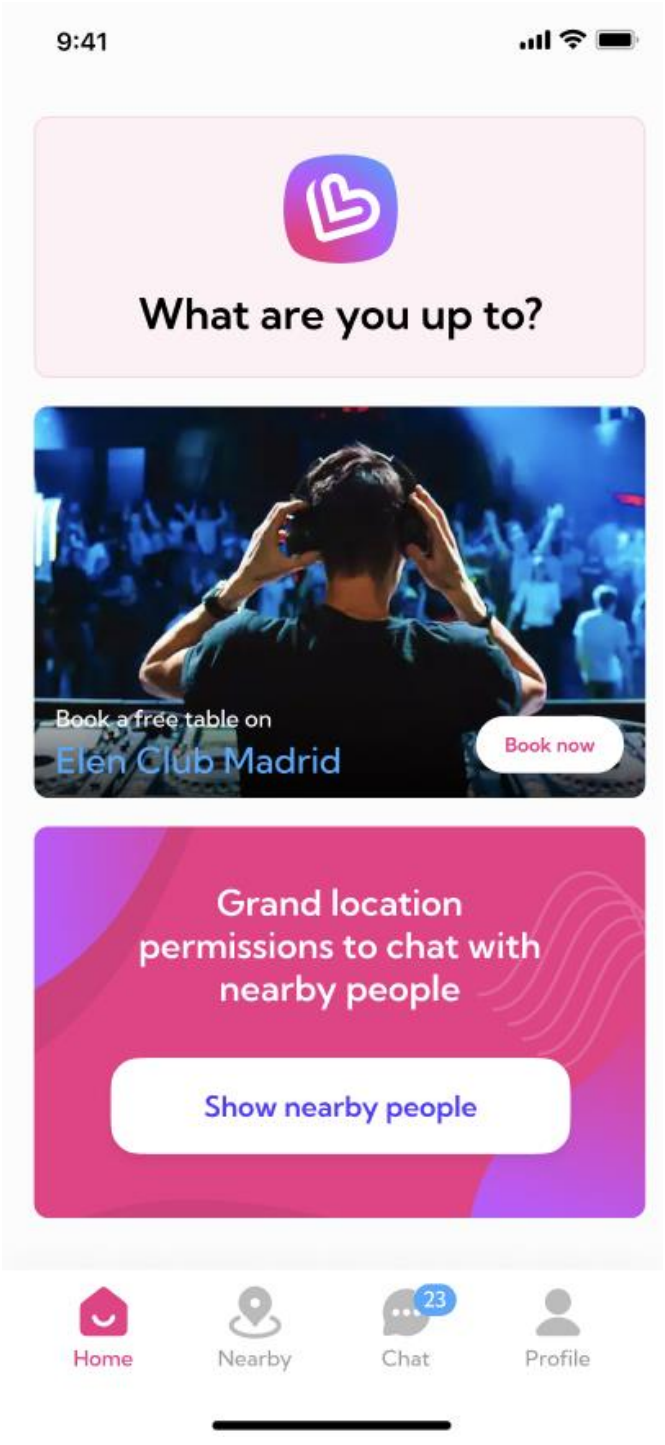

Рисунок Г.2 – Екран надання доступу до геолокації пристрою

6. Далі необхідно обрати місце для відпочинку вибравши зі списку

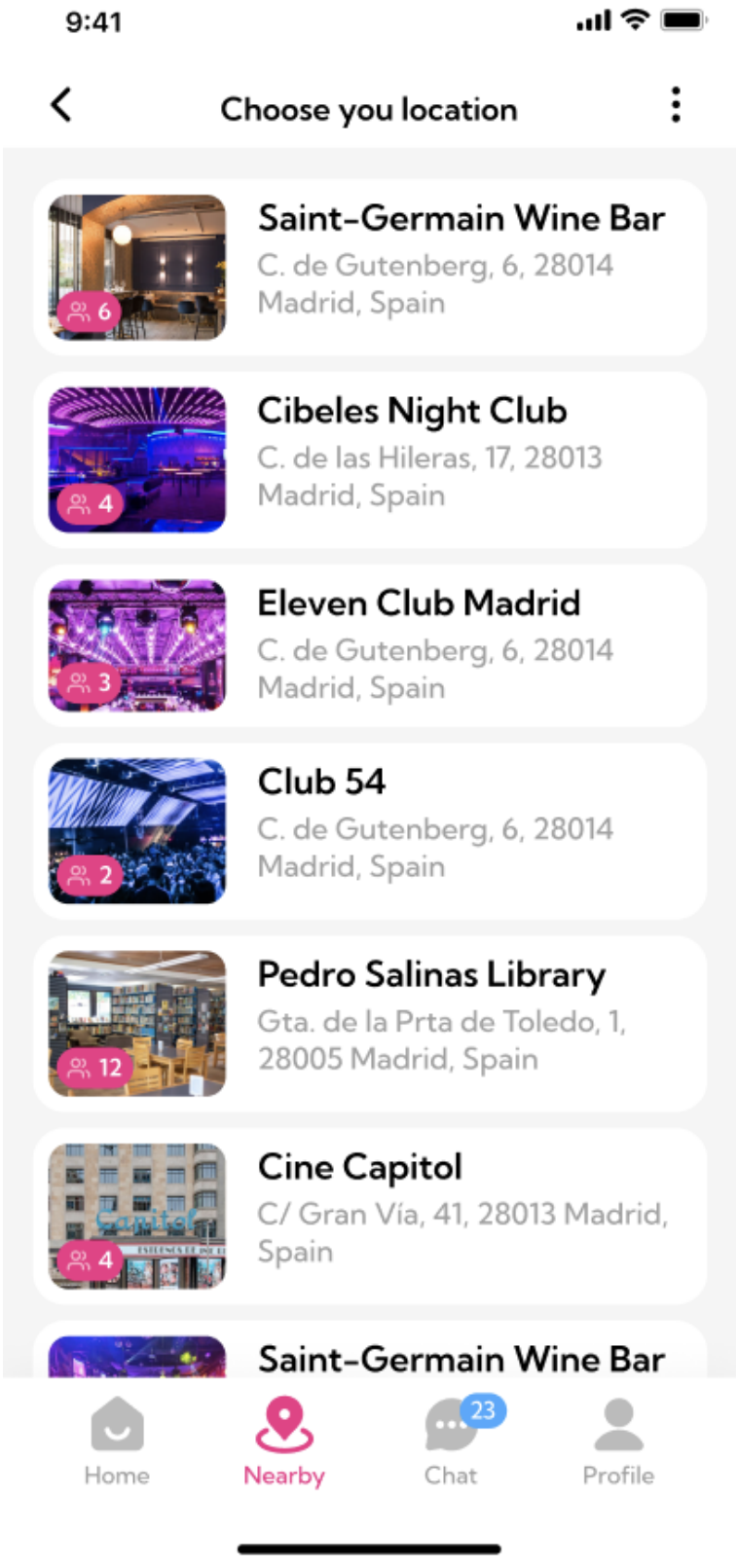

Рисунок Г.3 – Екран списку місць для відпочинку

7. На екрані з детялями про місце для відпочинку можна скористатись функцією «Check-in», щоб показати іншим користувачам додатку, що ви знаходитесь в тому ж місці що й вони, та мати можливість переглядати список профілів користувачів, що знаходяться в обраному місці для відпочинку, а також мати можливість брати участь в спілкуванні в чаті поточного місця.

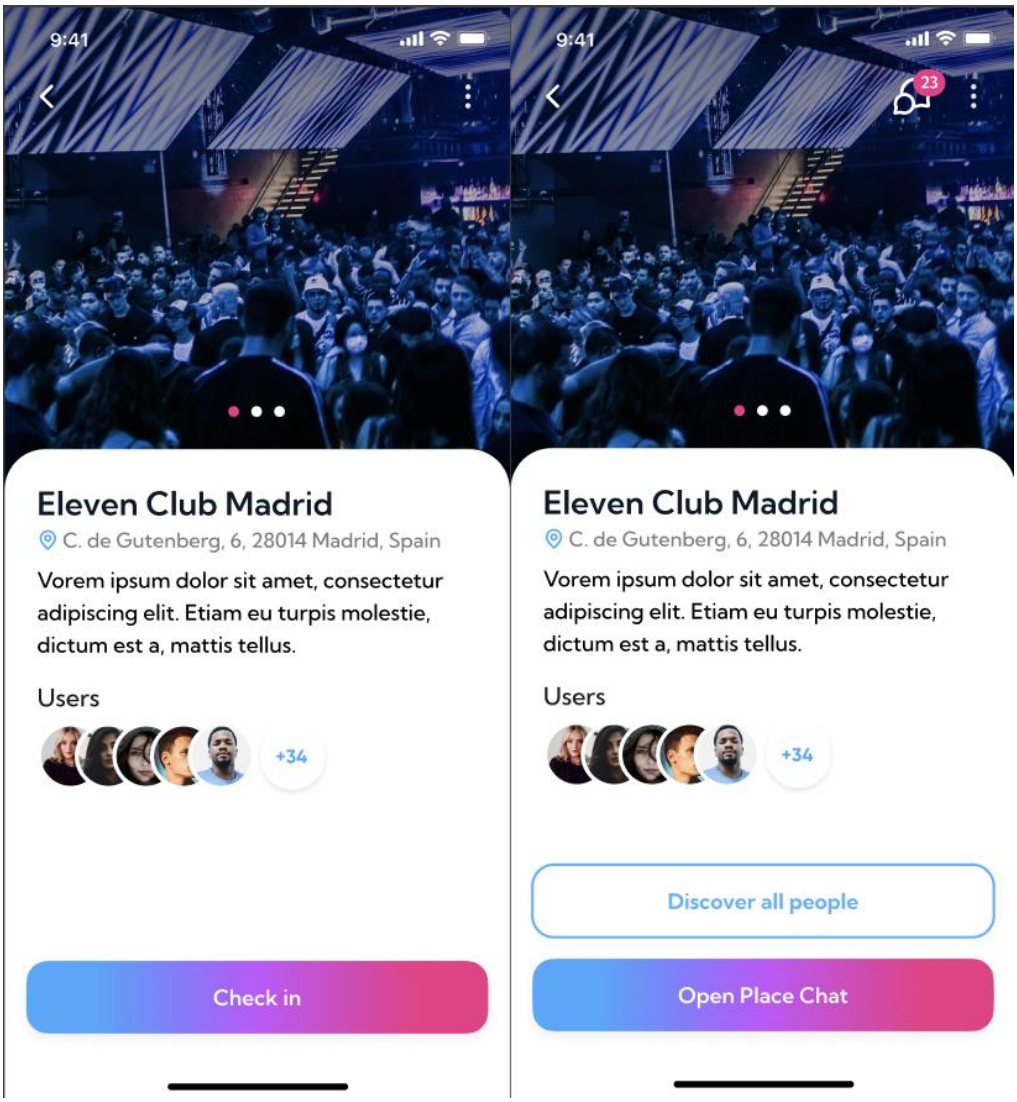

Рисунок Г.4 – Результат функції «Check-in»

8. Далі, щоб переглянути список людей в обраному місці для відпочинку необхідно натиснути на кнопку «Discover all people».

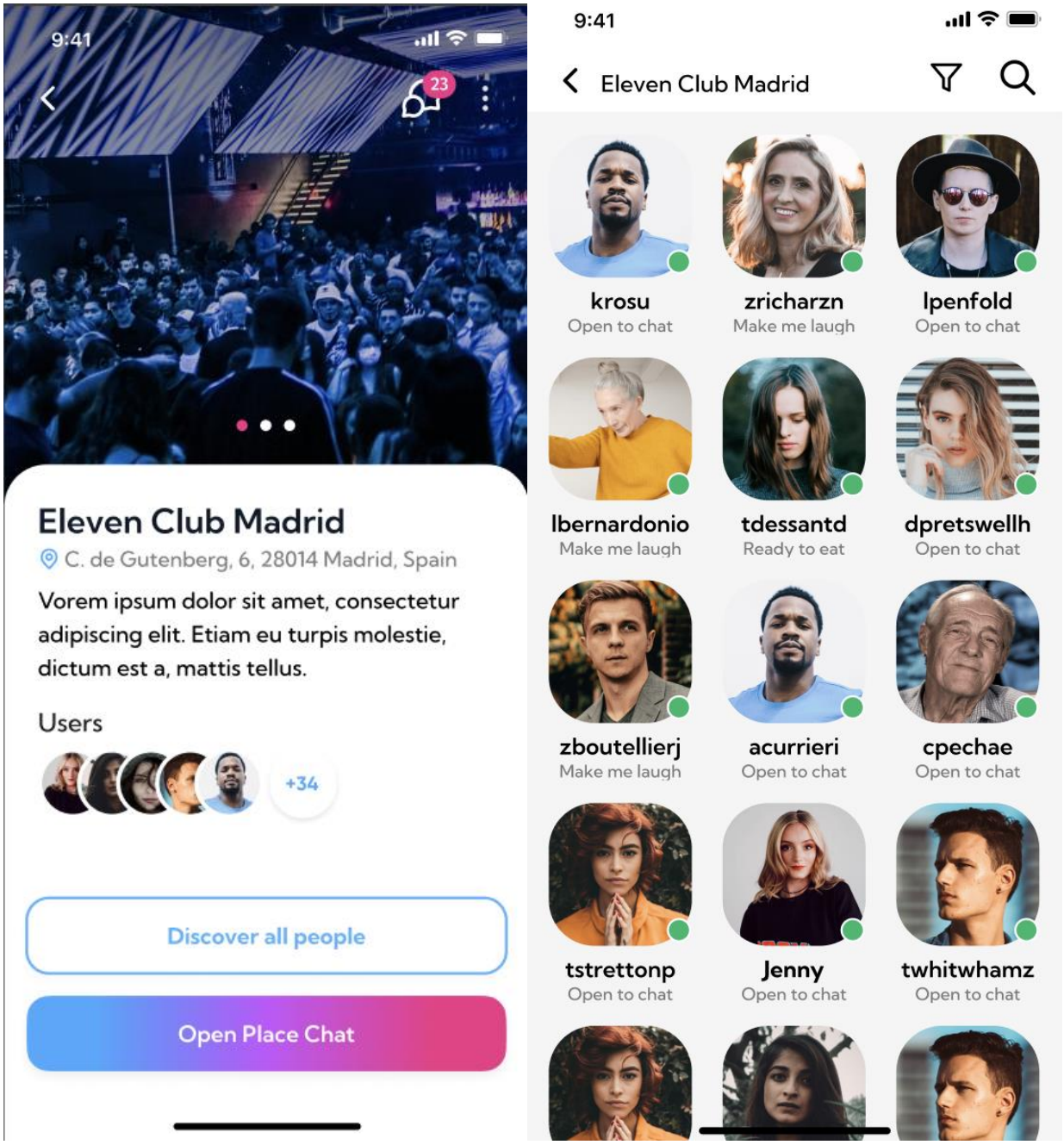

Рисунок Г.5 – Результат опції «Discover all people».

9. Щоб відкрити чат поточного місця необхідно натиснути кнопку «Open Place Chat».

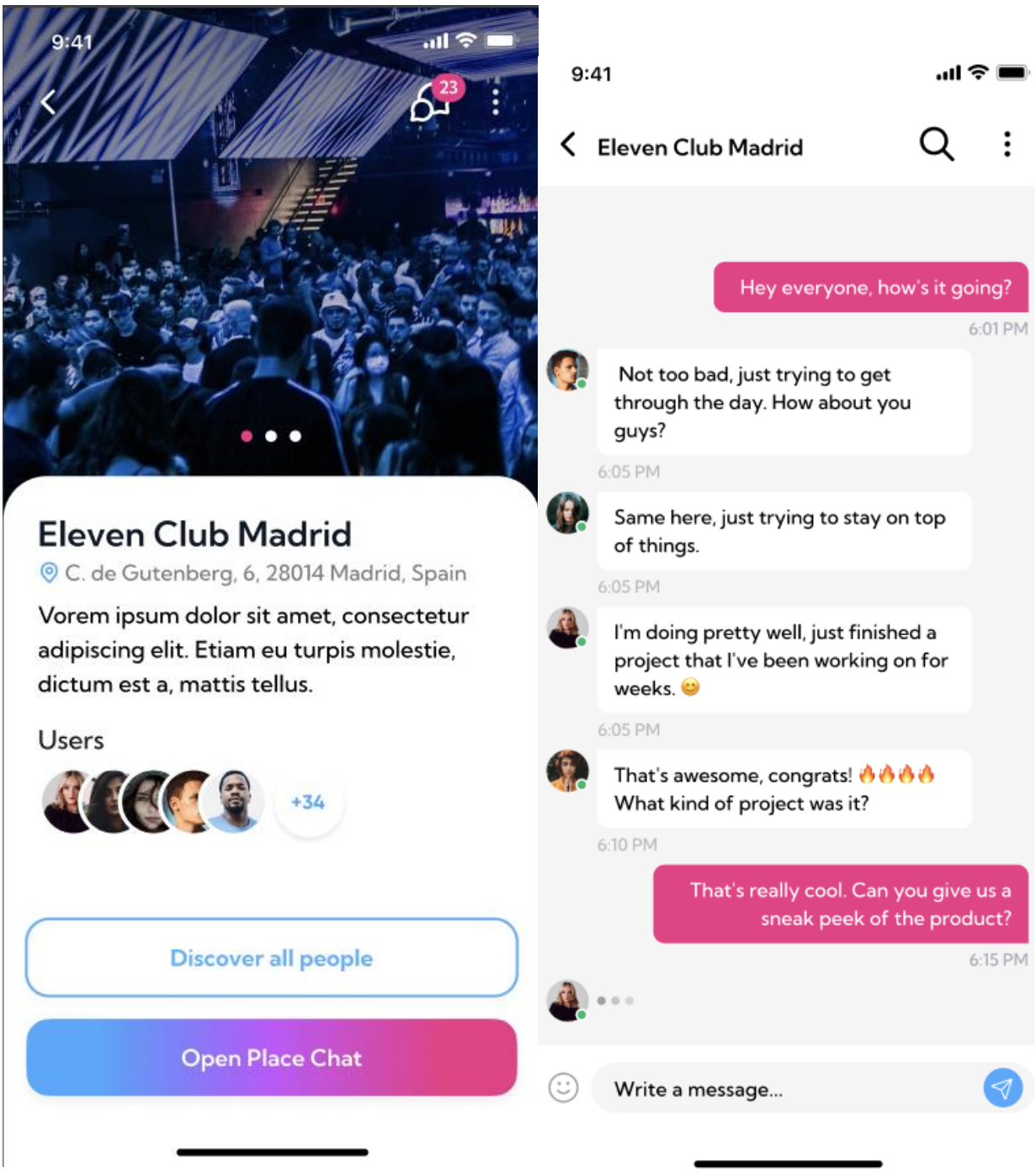

Рисунок Г.6 – Результат опції «Open Place Chat».

10. Далі за допомогою кнопки «Назад» можна повернутися на екран з навігаційним меню в низу екрану.

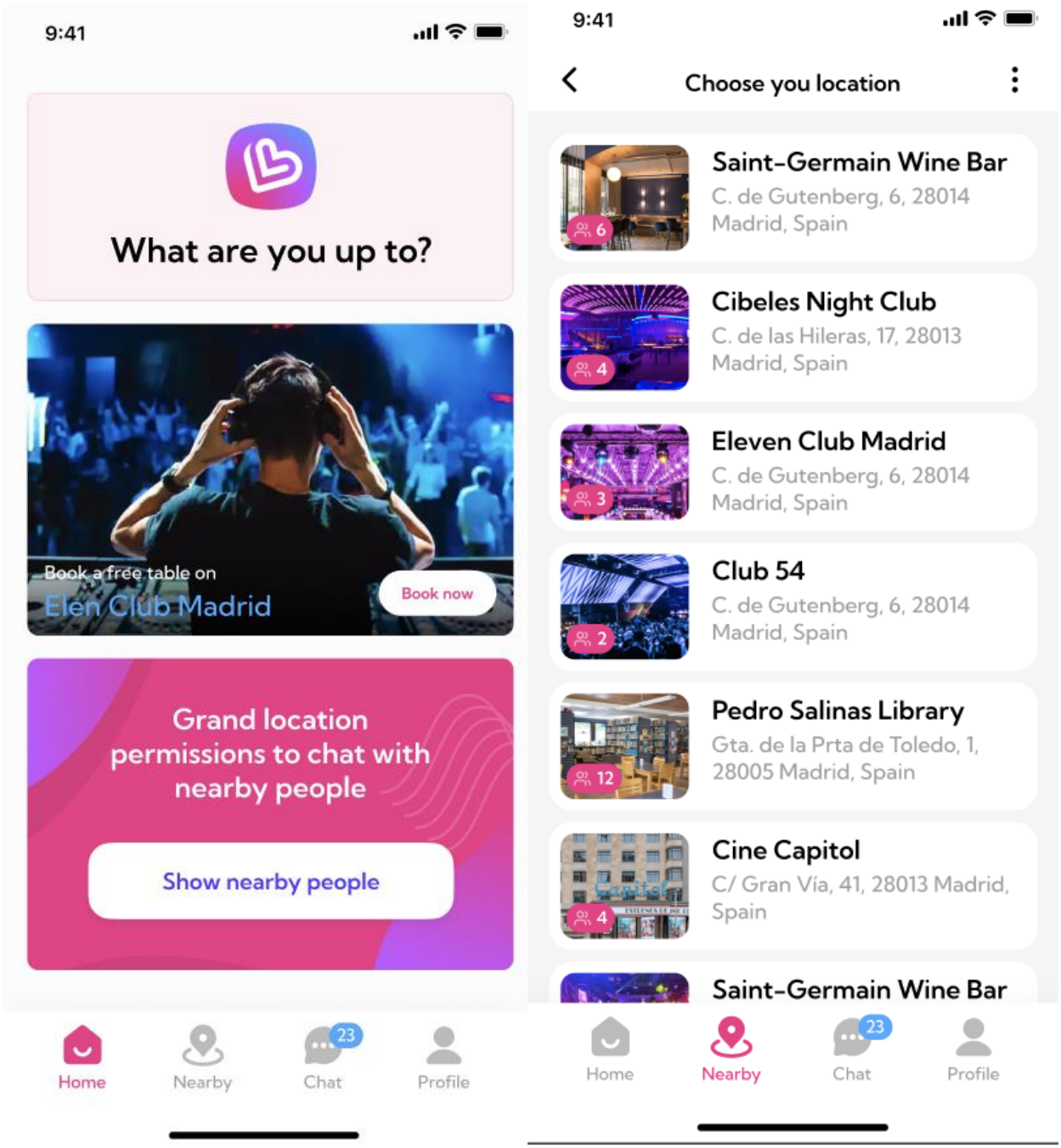

Рисунок Г.7 – Результат натискання кнопки «Назад».

11. Щоб переглянути налаштування профілю необхідно в навігаційному меню в низу екрану натиснути «Profile».

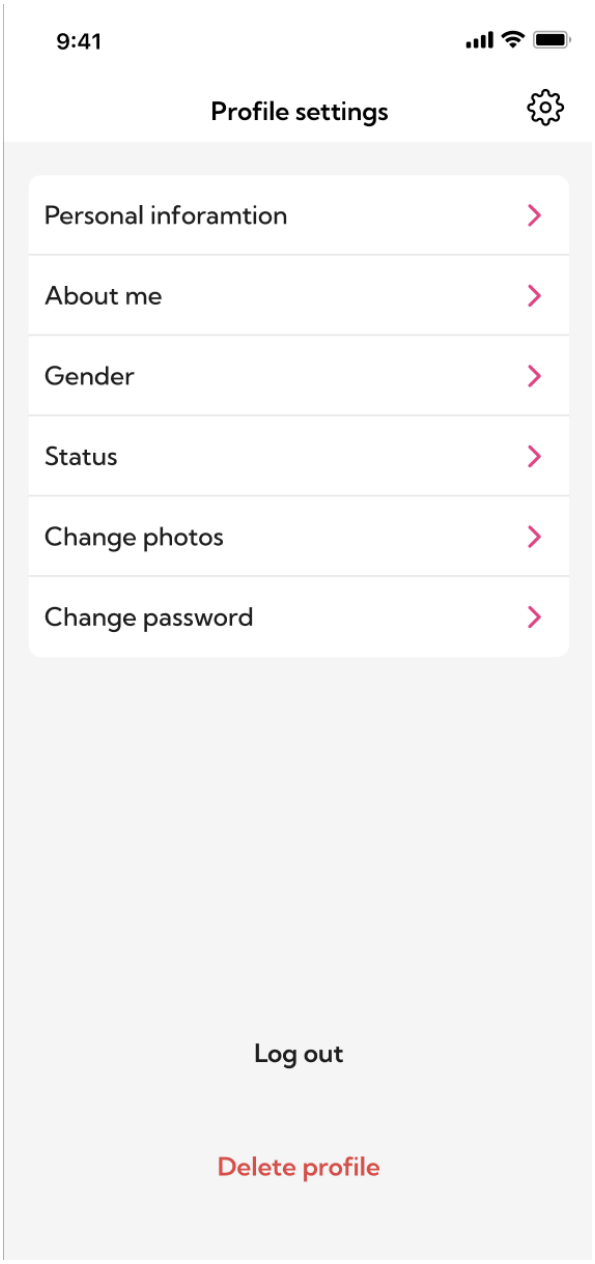

Рисунок Г.8 – Результат натискання кнопки «Profile».