

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління

(повна назва кафедри)

Магістерська кваліфікаційна робота
на тему:

«Автоматизація побудови маршруту руху з урахуванням
інфраструктурних об'єктів»

Виконав: студент 2-го курсу,
групи ЗАКІТ-23м
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка

(шифр і назва спеціальності)



Микола СКРИПНИК

(ім'я та прізвище)

Керівник: проф. каф. КСУ д.т.н.



Марія ЮХИМЧУК

(ім'я та прізвище)

« 4 » 12 2024 р.

Опонент: д.т.н., проф., професор каф.



Володимир ГАРМАШ

(ім'я та прізвище)

« 4 » 12 2024 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.

 В'ячеслав КОВТУН

(ім'я та прізвище)

« 4 » 12 2024 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти II-й (магістерський)
Галузь знань – 17 Електроніка, автоматизація та електронні комунікації
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

Освітньо-професійна програма - Інформаційні системи і Інтернет речей

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

В'ячеслав Ковтун

« 14 » 10 2024 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
студенту групи ЗАКІТР-23м Скрипнику Миколі Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Автоматизація побудови маршруту руху з урахуванням
інфраструктурних об'єктів»

керівник магістерської кваліфікаційної роботи професор кафедри КСУ , д. т.
н., Юхимчук Марія Сергіївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ВНТУ від "17" 09 2024 р. № 310

2. Термін подання студентом роботи 09.12. 2024 р.

3. Вихідні дані до магістерської кваліфікаційної роботи підтримка мобільних
платформ – Android; максимальний час відгуку системи – 3 с; авторизація користувачів – так;
максимальна кількість одночасних користувачів – до 5000 осіб; мови інтерфейсу – українська;
точність визначення місцезнаходження – до 50 метрів; використання шифрування даних під
час передачі – так; інтеграція з зовнішніми картографічними сервісами – так; забезпечення
роботи системи в режимі 24/7.

4. Зміст текстової частини: вступ, дослідження об'єкта автоматизації, дослідження та
аналіз інфраструктурних об'єктів для планування маршрутів, дослідження функціональних
можливостей сучасних мобільних застосунків для планування маршрутів, проектування
модуля користувача мобільного застосунку для автоматизованої системи планування
маршрутів, розробка модуля користувача мобільного застосунку для автоматизованої системи
планування маршрутів.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) ER-модель бази даних системи, UML-діаграма компонентів, UML-діаграма класів
системи, UML-діаграма варіантів використання, UML-діаграма послідовностей, вигляд екрану
розробленого додатку – 17 шт.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

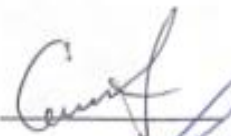
Календарний план

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Дослідження об'єкта автоматизації та постановка задач дослідження	03.09.2024р.	
2	Дослідження та аналіз інфраструктурних об'єктів для планування маршрутів	19.09.2024р.	
3	Дослідження функціональних можливостей сучасних мобільних застосунків для планування маршрутів	11.10.2024р.	
4	Проектування модуля користувача мобільного застосунку для автоматизованої системи планування маршрутів	05.11.2024р.	
5	Апробація результатів дослідження	20.11.2024р.	
6	Публікації		
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	29.11.2024р.	
8	Графічні матеріали: - UML-діаграми - ER-діаграма структури бази даних - вигляд екранів розробленого мобільного додатку	04.10.2024 р 18.10. 2024 р. 18.11. 2024 р.	
9	Захист МКР	11.12. 2024 р.	

Дата видачі завдання “ 01 ” 09 2024 року

Студент

Керівник магістерської кваліфікаційної роботи

 Скрипник М.В.
 Юхимчук М.С.

АНОТАЦІЯ

УДК 004.738.5

Скрипник Микола Володимирович Автоматизація побудови маршруту руху з урахуванням інфраструктурних об'єктів. Магістерська кваліфікаційна робота зі 174Автоматизація, комп'ютерно-інтегровані технології та робототехніка. Вінниця: ВНТУ, 2024. 125 с.

На укр. мові. Бібліогр.: 53 назв; рис.: 22; табл.: 1.

У магістерській кваліфікаційній роботі розроблено мобільний додаток для автоматизованої системи планування маршрутів, який враховує інфраструктурні об'єкти та спрямований на оптимізацію маршрутів, зниження часу в дорозі та підвищення загальної ефективності переміщень. У теоретичній частині роботи розглянуто основи автоматизації систем планування маршрутів, аналіз інфраструктурних об'єктів та їх вплив на планування маршрутів, а також застосування штучного інтелекту та машинного навчання у цих системах. В практичній частині виконано проектування та розробку модуля користувача, включаючи структуру бази даних, UML-діаграми, визначення основних функцій модуля та вибір технологій для його реалізації. Також було проведено тестування та налаштування роботи модуля.

Ключові слова: автоматизована система, планування маршрутів, мобільний застосунок, UML-діаграми.

ANNOTATION

UDC 004.738.5

Skrypnyk Mykola Volodimirovich Automation of Route Planning Considering Infrastructure Objects. Master's qualification work in specialty 174 - Automation, computer-integrated technologies and robotics. Vinnytsia: VNTU, 2024. – 120 p.

In Ukrainian language. Bibliographer: 53 titles; fig.: 22; tabl.: 1.

In the master's qualification work, a mobile application for an automated route planning system has been developed, which takes into account infrastructure objects and is aimed at optimizing routes, reducing travel time, and increasing overall movement efficiency. The theoretical part of the work discusses the basics of automating route planning systems, analysis of infrastructure objects and their impact on route planning, as well as the application of artificial intelligence and machine learning in these systems. In the practical part, the design and development of the user module were carried out, including the database structure, UML diagrams, definition of the main functions of the module, and selection of technologies for its implementation. Testing and configuration of the module were also conducted.

Keywords: automated system, route planning, mobile application, UML diagrams.

ЗМІСТ

ВСТУП	7
1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ІНФРАСТРУКТУРНИХ ОБ'ЄКТІВ ДЛЯ ПЛАНУВАННЯ МАРШРУТІВ	11
1.1 Дослідження об'єкта автоматизації	11
1.2 Дослідження типів інфраструктурних об'єктів	15
1.3 Аналіз впливу інфраструктурних об'єктів на планування маршрутів	18
1.4 Аналіз загальних підходів та методологій в системах планування маршрутів	19
1.4.1 Основні алгоритми та їх класифікація в системах планування маршрутів	20
1.4.2 Врахування динамічних змін у міському середовищі в системах планування маршрутів	23
1.4.3 Мультиmodalьні та інтерmodalьні підходи в плануванні маршрутів	24
1.4.4 Використання штучного інтелекту та машинного навчання в системах планування маршрутів	25
1.5 Використання штучного інтелекту та машинного навчання в системах планування маршрутів.....	27
1.6 Дослідження потреб користувачів в автоматизації планування маршрутів	28
1.7 Уточнена постановка задачі	30
2 ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СУЧАСНИХ МОБІЛЬНИХ ЗАСТОСУНКІВ ДЛЯ ПЛАНУВАННЯ МАРШРУТІВ	31
2.1 Огляд мобільних застосунків для планування маршрутів	31
2.1.1 Огляд мобільного застосунку Google Maps	32

2.1.2 Огляд мобільного застосунку Waze	35
2.1.3 Огляд мобільного застосунку HERE WeGo	38
2.1.4 Огляд мобільного застосунку Citymapper	40
2.2 Аналіз результатів огляду мобільних застосунків	46
3 ПРОЕКТУВАННЯ МОДУЛЯ КОРИСТУВАЧА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПЛАНУВАННЯ МАРШРУТІВ	48
3.1 Аналіз функцій модуля користувача	49
3.2 Розробка UML-діаграм для модуля користувача	50
3.2.1 Розробка діаграми компонентів	50
3.2.2 Розробка діаграми варіантів використання	52
3.2.3 Розробка діаграми класів	53
3.2.4 Розробка діаграми послідовностей	56
3.3 Розробка структури бази даних модуля користувача	59
3.4 Висновки до розділу	64
4 РОЗРОБКА МОДУЛЯ КОРИСТУВАЧА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПЛАНУВАННЯ МАРШРУТІВ	65
4.1 Вибір технологій для розробки модуля користувача	65
4.2 Розробка програмних компонентів модуля користувача	66
4.3 Налаштування та тестування роботи модуля користувача	73
ВИСНОВКИ	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	86
Додатки	92
Додаток А (Обов'язковий) Результати перевірки на плагіат.....	93
Додаток Б (обов'язковий) Технічне завдання	94
Додаток В (вибірковий) Лістинг програмного забезпечення.....	97
Додаток Г (Обов'язковий) Перелік графічних матеріалів	103

ВСТУП

Актуальність. У сучасному світі, де динаміка розвитку технологій впливає на всі аспекти життя, автоматизація стає ключовим елементом у підвищенні ефективності та оптимізації процесів. Однією з галузей, де автоматизація має вирішальне значення, є планування маршрутів, особливо з урахуванням інфраструктурних об'єктів. Це стосується не тільки транспортних компаній, але й звичайних користувачів, які прагнуть оптимізувати свої повсякденні маршрути.

З огляду на зростаючу урбанізацію та потребу в ефективному управлінні часом, системи автоматизованого планування маршрутів відіграють критичну роль у забезпеченні мобільності міського населення. Вони дозволяють інтегрувати різноманітні інфраструктурні об'єкти, такі як громадський транспорт, парковки, заправні станції, що робить планування маршрутів більш гнучким і адаптованим до індивідуальних потреб користувачів [1].

Автоматизація планування маршрутів також сприяє зменшенню транспортного навантаження на дороги, зниженню рівнів забруднення та підвищенню загальної безпеки дорожнього руху. Через інтеграцію даних про інфраструктурні об'єкти, системи можуть пропонувати альтернативні маршрути в реальному часі, реагуючи на дорожні умови, затори, аварії або інші непередбачені обставини.

Крім того, розвиток технологій Інтернету речей (IoT) та великих даних відкриває нові можливості для покращення систем планування маршрутів. Інтеграція цих технологій дозволяє збирати, обробляти та аналізувати великі обсяги даних з різних джерел, що значно підвищує точність та ефективність планування маршрутів.

Сучасні міста постійно розширюються та стають все більш складними у плані інфраструктури та управління транспортними потоками. Врахування інфраструктурних об'єктів у процесі планування маршрутів може значно поліпшити логістику міських перевезень, зменшити час у дорозі та підвищити загальну доступність транспорту. Це особливо важливо в умовах зростаючого трафіку та необхідності забезпечення сталого розвитку міських агломерацій.

Автоматизація цього процесу дозволяє використовувати дані в реальному часі для адаптації маршрутів, що може значно знизити навантаження на дорожню інфраструктуру та покращити умови для користувачів.

Таким чином, розробка автоматизованої системи планування маршрутів з урахуванням інфраструктурних об'єктів є актуальним завданням, що відповідає потребам сучасного суспільства. Це не тільки сприяє підвищенню ефективності транспортних систем, але й відкриває нові можливості для розвитку урбаністичних технологій та покращення якості життя міського населення [2].

Мета і завдання дослідження. Метою магістерської кваліфікаційної роботи є розробка та впровадження мобільного додатку для автоматизованої системи планування маршрутів з урахуванням інфраструктурних об'єктів, спрямованої на оптимізацію маршрутів, зниження часу в дорозі та підвищення загальної ефективності переміщень. Робота має на меті створення інноваційного інструменту, який дозволить користувачам ефективно планувати свої поїздки, інтегруючи різноманітні інфраструктурні дані та забезпечуючи високий рівень адаптації до актуальних дорожніх умов.

Для досягнення поставленої мети були визначені та вирішені наступні завдання:

- 1) проведено аналіз сучасних тенденцій та викликів у сфері планування маршрутів з урахуванням інфраструктурних об'єктів;
- 2) досліджено можливості інтеграції мобільного додатку з базами даних інфраструктурних об'єктів;
- 3) виконано порівняльний аналіз існуючих мобільних застосунків для планування маршрутів;
- 4) розроблено вимоги та специфікації для мобільного додатку автоматизованої системи планування маршрутів;
- 5) спроектовано архітектуру та реалізовано мобільний додаток для автоматизованої системи планування маршрутів;
- 6) проведено тестування та оцінку ефективності розробленого мобільного додатку.

Об'єктом дослідження є процес розробки та впровадження мобільного додатку для автоматизованої системи планування маршрутів з урахуванням інфраструктурних об'єктів.

Предметом дослідження є методи та засоби програмування, а також алгоритми оптимізації маршрутів, які використовуються для розробки та впровадження мобільного додатку автоматизованої системи планування маршрутів.

Методи дослідження. У процесі дослідження застосовувалися: теорія алгоритмів, Agile підхід до розробки програмного забезпечення, методика Scrum, методи управління базами даних, методи розробки мобільних додатків, засоби безпеки та захисту даних.

Наукова новизна одержаних результатів.

1. Проведено аналіз існуючих мобільних додатків для планування маршрутів та визначено ключові параметри та функціонал, необхідний для ефективної інтеграції інфраструктурних об'єктів, що сприяло формуванню вимог до розроблюваного мобільного додатку.
2. Спроектовано та розроблено мобільний додаток для автоматизованої системи планування маршрутів, який інтегрує дані інфраструктурних об'єктів та оптимізує маршрути, використовуючи вибрані користувачем фільтри для пошуку та відображення відповідних інфраструктурних об'єктів.
3. Впроваджено методи оптимізації обробки даних у мобільному додатку, що забезпечують ефективне управління транзакціями бази даних інфраструктурних об'єктів, гарантуючи ізолюваність та атомарність операцій при оновленні та запитах даних, що підвищує надійність та продуктивність системи.

Практичне значення одержаних результатів полягає в тому, що на основі отриманих теоретичних положень виконано розробку та впровадження мобільного додатку для автоматизованої системи планування маршрутів. Це дозволяє користувачам ефективно оптимізувати свої маршрути, знизити час у

дорозі та підвищити загальну ефективність переміщень, враховуючи актуальні інфраструктурні дані.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням усіх методів та засобів функціонування інформаційних систем, використанням Agile методики при проведенні розробки та впровадження мобільного додатку. Використання сучасних технологій та інтеграція з різними базами даних забезпечують високу якість та надійність розробленого додатку.

Особистий внесок здобувача. Усі результати отримано автором самостійно.

1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ІНФРАСТРУКТУРНИХ ОБ'ЄКТІВ ДЛЯ ПЛАНУВАННЯ МАРШРУТІВ

1.1 Дослідження об'єкта автоматизації

Темою магістерської кваліфікаційної роботи є автоматизація побудови маршруту руху з урахуванням інфраструктурних об'єктів. В рамках дослідження актуальності теми та планування етапів виконання роботи необхідно проаналізувати особливості систем планування маршрутів, методи інтеграції з геопросторовими базами даних, алгоритми оптимізації маршрутів, особливості розробки користувацького інтерфейсу для мобільних додатків, технології моніторингу та відслідковування руху в реальному часі, аналіз даних про дорожні умови та інфраструктурні зміни, реалізацію технічної підтримки та забезпечення безпеки даних у мобільному додатку.

Аналіз цих аспектів дозволить створити детальне уявлення про кінцевий мобільний додаток та його функціональні можливості, розробити на основі цього необхідні UML та ER діаграми, а також визначитися з необхідним програмним забезпеченням та алгоритмами для ефективного планування маршрутів з урахуванням різноманітних інфраструктурних об'єктів [3].

Місце задачі, що розв'язується в роботі, у загальному контексті автоматизації систем планування маршрутів є критично важливим для підвищення ефективності транспортних процесів у сучасних умовах зростання міського населення та потреби в оптимізації міської мобільності. Автоматизована система планування маршрутів з урахуванням інфраструктурних об'єктів, яка буде спроектована та розроблена в даній роботі, вирішує ряд ключових задач, зокрема оптимізацію маршрутів, зниження часу в дорозі та підвищення загальної доступності транспорту.

Однією з основних задач є автоматизація процесів планування маршрутів, що дозволяє користувачам ефективно вибирати найкращі шляхи переміщення з урахуванням актуальних даних про дорожні умови та наявність інфраструктурних

об'єктів. Це особливо важливо для міських жителів, які прагнуть мінімізувати час у дорозі.

Крім того, автоматизована система дозволяє користувачам в режимі реального часу відслідковувати зміни у дорожніх умовах, вибирати оптимальні маршрути та часи переміщення, а також отримувати повідомлення про будь-які зміни у доступності інфраструктурних об'єктів. Це робить процес планування маршрутів більш прозорим та зручним для користувачів.

Додатково, система сприяє більш ефективному плануванню робочого часу та ресурсів користувачів, зменшуючи час, проведений у дорозі, що в свою чергу веде до зниження витрат на транспорт та підвищення загальної продуктивності.

Усі ці функції та можливості роблять автоматизовану систему планування маршрутів невід'ємною складовою сучасної транспортної інфраструктури, забезпечуючи ефективність, зручність та надійність у процесах міського переміщення.

Область застосування розробки

Розроблений мобільний додаток для автоматизації планування маршрутів з урахуванням інфраструктурних об'єктів може бути застосований у широкому діапазоні діяльності. Він буде особливо корисним для міських транспортних служб, таксі служб, служб доставки, а також для індивідуальних користувачів, які прагнуть оптимізувати свої повсякденні маршрути. Також додаток може бути використаний у туристичній індустрії для планування оптимальних маршрутів подорожей з урахуванням історичних пам'яток, парків та інших важливих об'єктів.

Більше того, цей додаток може бути корисним у сфері міського планування та управління, допомагаючи місцевим органам влади аналізувати та оптимізувати трафік, планувати розвиток інфраструктури та реагувати на зміни у міському середовищі.

Ця універсальність та адаптивність додатку дозволяють йому знаходити застосування в різних секторах і сприяти покращенню організації робочих процесів та підвищенню якості життя користувачів у міських умовах, де критично

важливою є ефективність управління транспортними потоками та оптимізація переміщень.

Вимоги до програмно-технічного забезпечення, що розробляється в магістерській кваліфікаційній роботі, можна описати наступним чином:

- Локальна база даних: для зберігання та управління інформацією про маршрути, подорожі, прогрес подорожей та користувацькі налаштування використовується база даних, що ґрунтується на системі SQLite. Ця система забезпечує ефективний доступ до даних та забезпечує їх цілісність на мобільних пристроях. SQLite є ідеальним вибором для локального зберігання, оскільки вона легка, не вимагає окремого серверного обслуговування та забезпечує високу швидкість читання та запису даних.
- Хмарне сховище: для зберігання та обробки даних, що вимагають централізованого доступу та високої доступності, використовується Google Cloud SQL. Цей сервіс забезпечує масштабованість, високу доступність та надійність, що є важливим для обробки запитів від мобільного додатку в реальному часі. Google Cloud SQL підтримує різні реляційні бази даних і дозволяє легко інтегрувати з існуючою архітектурою, забезпечуючи ефективне управління ресурсами та автоматичне резервне копіювання [4, 5].
- Розробка мобільного додатку: створення та розгортання мобільного додатку використовує сучасні інструменти та технології мобільної розробки. Використовуються мов програмування Kotlin та Java для Android, що дозволяє створювати високопродуктивні та адаптивні інтерфейси для різних типів пристроїв.
- Серверні рішення: для зберігання даних та обробки запитів від мобільного додатку використовується хмарне сховище та обчислювальні можливості, такі як Google Cloud. Це забезпечує масштабованість та високу доступність сервісу.
- Засоби безпеки: забезпечення конфіденційності та цілісності даних користувачів та маршрутної інформації є пріоритетом у розробці.

Використовуються методи шифрування даних та захисту інформації на рівні мобільного додатку та сервера. Застосування протоколів безпеки, таких як HTTPS та OAuth, допомагає забезпечити безпечне з'єднання та авторизацію користувачів.

Функції розробки:

- забезпечення доступу користувачів до системи та персональних акаунтів;
- управління маршрутами з урахуванням інфраструктурних об'єктів;
- моніторинг та відслідковування статусу маршрутів у реальному часі;
- аналіз ефективності маршрутів та рекомендації щодо їх оптимізації.

Для виконання роботи необхідні наступні вхідні дані:

- локальна база даних, яку описує ER-діаграма;
- відповідно налаштовані програмні та технічні засоби, що необхідні для розробки та розгортання мобільного додатку.

Перевіркою розробки та розгортання мобільного додатку необхідними будуть результати роботи готового продукту, правильна взаємодія його процесів та команд, відображення вихідної інформації.

1.2 Дослідження типів інфраструктурних об'єктів

Інфраструктурні об'єкти є ключовими елементами, які підтримують функціональність міських та сільських територій, забезпечуючи необхідні умови для життєдіяльності населення та ефективної роботи різних видів транспорту. Ці об'єкти можна класифікувати за кількома категоріями в залежності від їх призначення та впливу на міське середовище, що зображено на Рис. 1.1.

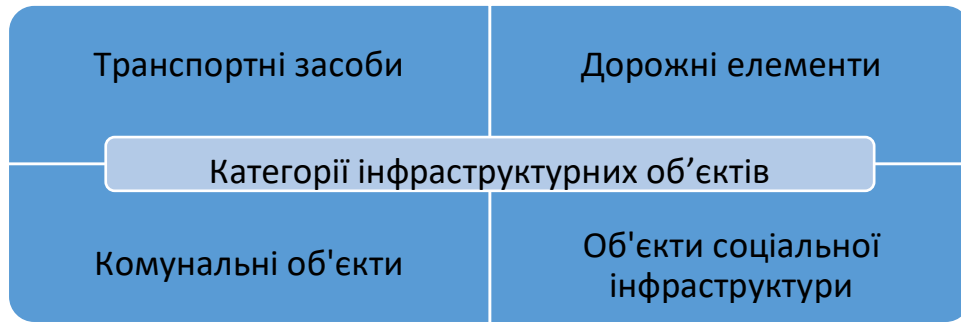


Рисунок 1.1 – Категорії інфраструктурних об'єктів за їх призначення та впливом на міське середовище

Транспортні засоби

Ця категорія включає в себе всі типи транспортних засобів, які використовуються для перевезення людей та товарів. До них належать, наприклад:

- Автобуси – основний засіб громадського транспорту в багатьох містах, що забезпечує доступність та зв'язок між різними районами.
- Тролейбуси – екологічно чистий вид транспорту, який також сприяє зменшенню заторів у міських умовах.

Дорожні елементи

Дорожні елементи включають в себе всі конструкції та знаки, які регулюють рух транспорту та пішоходів:

- Світлофори – важливий елемент управління трафіком, який допомагає уникнути зіткнень та забезпечує безпеку на дорогах.
- Дорожні знаки – інформують учасників дорожнього руху про правила та особливості руху на конкретних ділянках дороги.

Комунальні об'єкти

Комунальна інфраструктура включає об'єкти, які забезпечують життєво необхідні ресурси для населення:

- Водогони – системи, які забезпечують постачання питної води до житлових та громадських будівель.
- Електростанції – об'єкти, які виробляють електроенергію для потреб міста та його мешканців.

Об'єкти соціальної інфраструктури

Ці об'єкти відіграють важливу роль у забезпеченні соціальних потреб населення:

- Школи – надають освітні послуги, формуючи основу для розвитку молодого покоління.
- Лікарні – забезпечують медичні послуги, що є критично важливими для підтримки здоров'я населення.

Розуміння функціонального призначення інфраструктурних об'єктів дозволяє глибше аналізувати їх вплив на міське середовище та мобільність людей. Кожен тип об'єктів виконує специфічні функції, які формують умови життя та руху в місті [6-8].

Транспортна інфраструктура включає дороги, мости, тунелі та залізничні колії. Ці об'єкти забезпечують фізичну основу для переміщення транспортних засобів і пішоходів. Вони впливають на швидкість та зручність переміщення, а також на розвиток економічних зв'язків між різними районами міста.

Комунальна інфраструктура, така як водопровідні та каналізаційні системи, електромережі, тепломережі, відіграє критичну роль у забезпеченні базових життєвих потреб мешканців. Її ефективність і надійність впливають на здоров'я та комфорт населення, а також на привабливість міських районів для проживання та бізнесу.

Інформаційна інфраструктура включає телекомунікаційні мережі та центри обробки даних. В сучасному світі доступ до швидкісного інтернету та інших комунікаційних послуг є важливим фактором економічного розвитку та особистісного зростання. Інформаційна інфраструктура також сприяє інтеграції міських послуг та покращенню управління міськими ресурсами.

До екологічної інфраструктури належать парки, зелені зони, системи очистки води та повітря. Ці об'єкти відіграють ключову роль у забезпеченні екологічного балансу в міських умовах. Вони не тільки покращують якість життя мешканців, але й сприяють збереженню біорізноманіття та зменшенню негативного впливу міського середовища на природу.

Кожен з цих типів інфраструктурних об'єктів має свої особливості, які впливають на різні аспекти життя. Розуміння цих взаємозв'язків допомагає у

плануванні більш ефективних та збалансованих міських середовищ, що враховують потреби всіх мешканців та сприяють сталому розвитку.

Аналогічно у контексті автоматизації побудови маршруту руху, географічне розміщення інфраструктурних об'єктів відіграє критичну роль. В міських та сільських районах розташування таких об'єктів як транспортні вузли, комунальні служби, школи та лікарні безпосередньо впливає на оптимізацію маршрутів. Наприклад, автоматизовані системи можуть враховувати розташування автобусних зупинок та оптимальні шляхи до них для підвищення ефективності громадського транспорту. У сільських районах, де інфраструктура може бути менш розвиненою, важливо аналізувати доступність доріг та мостів, що може вплинути на час та безпеку пересування.

Інфраструктурні об'єкти не існують ізольовано; вони формують складну мережу взаємодій, яка впливає на мобільність у місті. Наприклад, розміщення парків та рекреаційних зон може впливати на трафік у прилеглих районах, особливо у вихідні дні. Автоматизовані системи планування маршрутів можуть адаптуватися до таких умов, пропонуючи альтернативні шляхи під час піків відвідувань. Також, важливо враховувати взаємодію між різними видами транспорту: наприклад, як закриття дороги для ремонтних робіт впливає на маршрути громадського транспорту та загальний потік трафіку.

Реальні приклади демонструють, як інфраструктурні об'єкти можуть радикально змінювати міське планування. Наприклад, створення пішохідних зон у центральних частинах міст може зменшити автомобільний трафік і покращити якість повітря. Це, у свою чергу, змушує міські служби адаптувати маршрути громадського транспорту та покращувати інфраструктуру для велосипедистів та пішоходів. Інший приклад – введення "зелених коридорів" у містах, які сприяють зменшенню заторів і створенню більш здорового міського середовища. Такі ініціативи вимагають переосмислення традиційних підходів до планування маршрутів та розробки нових алгоритмів для автоматизованих систем [9, 10].

1.3 Аналіз впливу інфраструктурних об'єктів на планування маршрутів

У даному підрозділі буде зосереджено увагу на аналізі того, як різні типи інфраструктурних об'єктів впливають на процеси планування маршрутів у контексті автоматизації. Розглядатиметься, як особливості розташування та функціональні характеристики цих об'єктів впливають на вибір маршрутів, час в дорозі та загальну ефективність транспортної системи.

Транспортна інфраструктура має значний вплив на планування маршрутів. Зокрема, наявність широких магістралей дозволяє збільшити пропускну спроможність транспортних потоків, що призводить до зменшення заторів у пікові години. Водночас, мости та тунелі як критичні точки можуть створювати "вузькі горловини", що потребують особливої уваги при плануванні маршрутів. Тимчасові зміни, такі як дорожні роботи, вимагають оперативного перепланування маршрутів, щоб уникнути затримок.

Комунальна інфраструктура, така як системи водопостачання та електропостачання, впливає на маршрутизацію спеціалізованих служб. Наприклад, у випадку аварій на магістральних водогонах або електролініях, необхідно швидко перенаправляти транспорт спецслужб, що вимагає від системи планування маршрутів високої адаптивності та оперативності.

Соціальна інфраструктура, така як школи та лікарні, створює специфічні потреби у плануванні маршрутів. Наприклад, безпека шкільних автобусів вимагає ретельного вибору маршрутів, щоб уникнути небезпечних перехресть та вулиць з інтенсивним трафіком. Лікарні, з іншого боку, потребують швидкого доступу швидкої допомоги, що вимагає від системи планування маршрутів забезпечення найкоротших та найефективніших шляхів.

Екологічна інфраструктура, така як парки та зелені зони, може впливати на міську мобільність, зменшуючи доступність деяких районів для автомобільного транспорту, але водночас сприяючи розвитку пішохідних та велосипедних маршрутів. Це створює потребу в балансуванні між забезпеченням доступності та збереженням екологічних зон [11].

Ключовими факторами, які впливають на планування маршрутів у контексті автоматизації, є розташування та характеристики транспортної, комунальної, соціальної та екологічної інфраструктури. Транспортна інфраструктура визначає основні потоки трафіку та потребує адаптації під час тимчасових змін, таких як дорожні роботи. Комунальна інфраструктура впливає на маршрутизацію спеціалізованих служб, особливо в надзвичайних ситуаціях. Соціальна інфраструктура формує специфічні вимоги до безпеки та доступності, особливо для шкільних автобусів та екстрених служб. Екологічна інфраструктура впливає на планування маршрутів, сприяючи розвитку пішохідних та велосипедних шляхів, але обмежуючи доступність для автомобільного транспорту. Всі ці фактори вимагають інтеграції та адаптації в системах автоматизованого планування маршрутів для забезпечення їхньої ефективності та відповідності до динамічних умов міського середовища.

1.4 Аналіз загальних підходів та методологій в системах планування маршрутів

Системи планування маршрутів пройшли значний шлях розвитку, починаючи з ручних методів і досягнувши сучасних автоматизованих технологій. Спочатку маршрутизація здійснювалася вручну, з використанням карт та планів, де люди самостійно визначали оптимальні шляхи для подорожей чи транспортування товарів. Цей процес був трудомістким і схильним до помилок, особливо при плануванні довгих або складних маршрутів.

З появою комп'ютерних технологій у другій половині 20-го століття почалася ера автоматизації планування маршрутів. В 1956 році було розроблено алгоритм Дейкстри, який став основою для багатьох сучасних алгоритмів маршрутизації. Цей алгоритм дозволив автоматично визначати найкоротший шлях між точками на графі, що значно спростило процеси планування маршрутів.

З розвитком Інтернету та мобільних технологій у кінці 20-го та на початку 21-го століття системи планування маршрутів стали ще більш доступними та зручними. З'явилися онлайн-сервіси та мобільні додатки, такі як Google Maps та

Waze, які використовують дані GPS та реальний стан трафіку для надання користувачам актуальної інформації про маршрути.

Сучасні системи планування маршрутів інтегрують широкий спектр даних, включаючи геопросторові дані, інформацію про погодні умови, дорожні роботи, затори та інші фактори, що впливають на мобільність. Вони також використовують алгоритми штучного інтелекту та машинного навчання для прогнозування трафіку та оптимізації маршрутів у реальному часі, що робить процес планування маршрутів більш ефективним та адаптованим до потреб користувачів [12].

1.4.1 Основні алгоритми та їх класифікація в системах планування маршрутів

Системи планування маршрутів використовують різноманітні алгоритми для визначення оптимальних шляхів. Ось декілька ключових алгоритмів, які широко застосовуються в цій області [13]:

1. Алгоритм Дейкстри:

- **Опис:** Алгоритм Дейкстри використовується для знаходження найкоротшого шляху від однієї вершини до всіх інших вершин у графі з невід'ємними вагами ребер. Візуалізація графа алгоритму зображена на Рис. 1.2.
- **Переваги:** Простота реалізації та ефективність у багатьох практичних сценаріях.
- **Недоліки:** Не є ефективним для графів з великою кількістю вершин та ребер, оскільки його часова складність може бути досить високою.

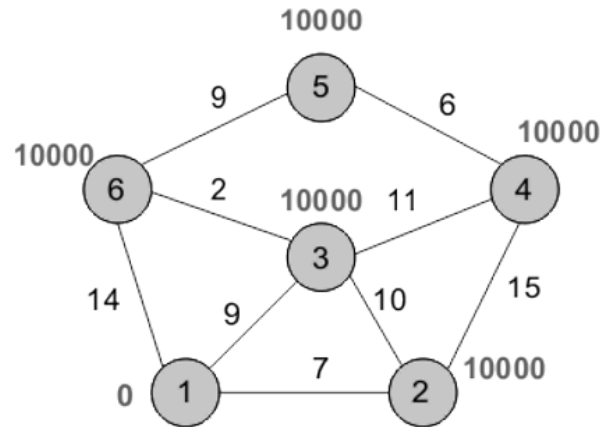


Рисунок 1.2 – Граф алгоритму Дейкстри

2. Алгоритм A*:

- Опис: A* є розширенням алгоритму Дейкстри, який використовує евристики для прискорення пошуку шляху, зосереджуючись на більш обіцяючих напрямках. Візуалізація графа алгоритму зображена на Рис. 1.3.
- Переваги: Забезпечує більш швидке знаходження найкоротшого шляху завдяки використанню евристичної функції, яка допомагає оцінити "вартість" переходу від поточної вершини до цільової.
- Недоліки: Вибір неадекватної евристичної функції може призвести до неправильних результатів або збільшення часу виконання.

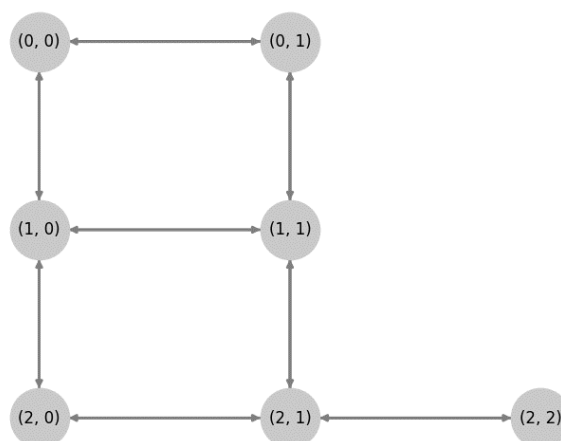


Рисунок 1.3 – Граф алгоритму A*

3. Алгоритм Беллмана-Форда:

- Опис: Цей алгоритм здатен знаходити найкоротші шляхи в графі, який містить ребра з від'ємними вагами, але без циклів від'ємної ваги. Візуалізація графа алгоритму зображена на Рис. 1.4.
- Переваги: Може обробляти графи з від'ємними вагами, що є важливим у деяких застосуваннях.
- Недоліки: Має вищу часову складність порівняно з алгоритмом Дейкстри, що робить його менш ефективним для великих графів.

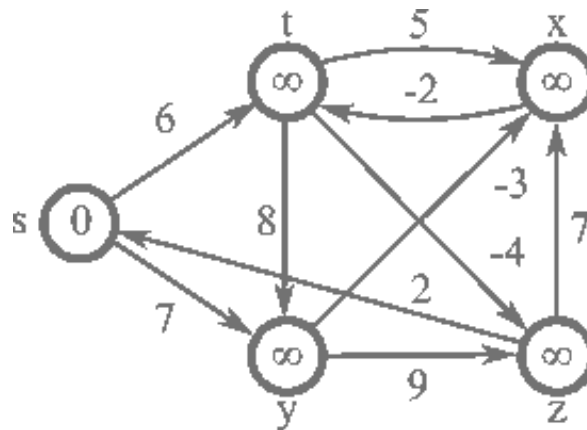


Рисунок 1.4 – Граф алгоритму Беллмана-Форда

4. Floyd-Warshall алгоритм:

- Опис: Використовується для знаходження найкоротших шляхів між усіма парами вершин у графі. Візуалізація графа алгоритму зображена на Рис. 1.5.
- Переваги: Дозволяє одночасно отримати найкоротші шляхи між усіма парами вершин, що є корисним для деяких застосувань.
- Недоліки: Має високу часову та просторову складність, що обмежує його використання в графах з великою кількістю вершин.

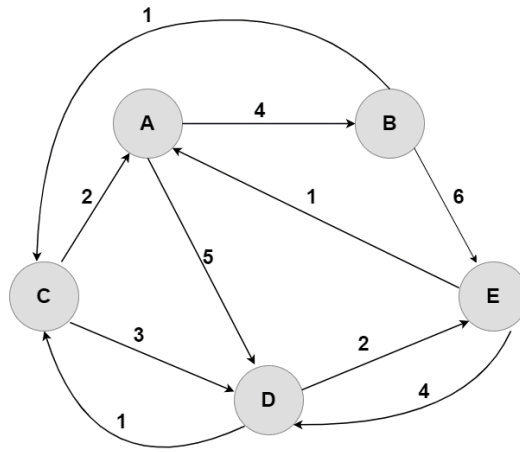


Рисунок 1.5 - Граф алгоритму Floyd-Warshall

Ці алгоритми формують основу для більшості сучасних систем планування маршрутів, дозволяючи їм ефективно адаптуватися до різних вимог та умов. Вибір алгоритму залежить від специфіки задачі, розміру та структури графа, а також від потреби в швидкодії та точності [14-16].

1.4.2 Врахування динамічних змін у міському середовищі в системах планування маршрутів

Сучасні системи планування маршрутів повинні бути гнучкими та адаптивними, щоб ефективно реагувати на динамічні зміни в міському середовищі. Це включає здатність швидко адаптуватися до тимчасових змін, таких як дорожні роботи, затори, аварії, погодні умови та інші непередбачувані події, які можуть вплинути на мобільність.

1. Інтеграція реального часу: Системи планування маршрутів інтегрують дані в реальному часі, зібрані з різних джерел, таких як дорожні камери, GPS-дані з транспортних засобів, соціальні медіа та інші інформаційні сервіси. Це дозволяє їм надавати користувачам актуальну інформацію та альтернативні маршрути в разі виникнення перешкод.
2. Алгоритми прогнозування: Використання алгоритмів машинного навчання та штучного інтелекту дозволяє системам аналізувати великі обсяги даних та прогнозувати потенційні зміни в дорожніх умовах. Це може включати

прогнозування заторів, оцінку впливу погодних умов на трафік або визначення ймовірності дорожніх інцидентів.

3. Адаптивність маршрутів: Системи можуть автоматично адаптувати заплановані маршрути на основі отриманої інформації, пропонуючи водіям або пасажиром альтернативні шляхи, що уникають перешкод. Це не тільки збільшує ефективність пересування, але й знижує рівень стресу та покращує загальний досвід користувачів.
4. Інтерактивність та зворотний зв'язок: Користувачі систем планування маршрутів можуть надавати зворотний зв'язок про стан доріг, що допомагає уточнити дані та покращити точність прогнозів. Це створює колаборативну екосистему, де кожен учасник дорожнього руху може внести свій вклад у загальну точність та ефективність системи.

Ці підходи демонструють, як сучасні системи планування маршрутів адаптуються до постійно змінюваних умов міського середовища, забезпечуючи своєчасну та ефективну навігацію [17].

1.4.3 Мультимодальні та інтермодальні підходи в плануванні маршрутів

Мультимодальні та інтермодальні підходи до планування маршрутів стають все більш популярними у відповідь на зростаючу потребу в ефективній та сталій мобільності в міських середовищах. Ці підходи включають використання кількох видів транспорту для однієї поїздки, що дозволяє користувачам оптимізувати свій маршрут за часом, вартістю, комфортом або екологічними показниками.

1. Мультимодальність: Мультимодальні системи дозволяють користувачам використовувати різні види транспорту в рамках однієї поїздки. Наприклад, маршрут може включати поїздку на автомобілі до станції метро, пересадку на метро, а потім коротку пішу прогулянку до кінцевого пункту призначення. Системи планування маршрутів, які підтримують мультимодальність, зазвичай інтегрують дані про розклади, вартість та доступність різних видів транспорту.

2. Інтермодальність: Інтермодальні системи зосереджуються на оптимальній інтеграції різних транспортних засобів, забезпечуючи плавні пересадки між ними. Це включає синхронізацію розкладів, управління транспортними потоками та надання інформації про пересадки в реальному часі. Інтермодальні підходи часто використовуються для планування довгих поїздок, які включають авіаційний та залізничний транспорт [18].

Інтеграція мультимодальних та інтермодальних підходів:

- Технологічна інтеграція: Сучасні системи планування маршрутів використовують API для інтеграції даних з різних транспортних служб, включаючи громадський транспорт, таксі, каршеринг та велосипедні системи. Це дозволяє їм збирати та аналізувати великі обсяги інформації для створення оптимальних маршрутів.
- Користувацький досвід: Для забезпечення зручності користувачів, системи надають інтерфейси, які дозволяють легко вибирати між різними опціями маршрутів, порівнювати вартість та час поїздок, а також отримувати інструкції для пересадок.
- Адаптивність та гнучкість: Системи постійно оновлюються з урахуванням змін у транспортній інфраструктурі та умовах дорожнього руху, що дозволяє користувачам отримувати актуальну інформацію та адаптувати свої плани відповідно до поточної ситуації.

Ці підходи значно покращують мобільність у міських середовищах, сприяючи створенню більш сталої та ефективної транспортної системи. Вони дозволяють користувачам максимально ефективно використовувати доступні транспортні ресурси, знижуючи час у дорозі та вплив на довкілля [19].

1.4.4 Використання штучного інтелекту та машинного навчання в системах планування маршрутів

Штучний інтелект (ШІ) та машинне навчання (МН) відіграють ключову роль у розвитку та оптимізації сучасних систем планування маршрутів. Ці технології дозволяють системам аналізувати великі обсяги даних, прогнозувати

зміни в умовах дорожнього руху та оптимізувати маршрути в реальному часі. Ось декілька ключових аспектів використання ІІ та МН у плануванні маршрутів:

1. Прогнозування трафіку:

- Моделі прогнозування: Машинне навчання використовується для аналізу історичних даних про трафік та інших відповідних факторів (погода, час доби, свята тощо) для прогнозування майбутніх умов на дорогах.
- Адаптивні системи: ІІ може адаптувати прогнози в реальному часі на основі надходження нових даних, що дозволяє користувачам отримувати найактуальнішу інформацію.
- Оптимізація маршрутів:
- Динамічне маршрутизування: ІІ використовується для аналізу різних маршрутних альтернатив та вибору оптимального шляху з урахуванням поточного стану трафіку та інших змінних.
- Мультиmodalьні та інтерmodalьні маршрути: ІІ допомагає інтегрувати різні види транспорту в єдиний маршрут, оптимізуючи час в дорозі та загальну ефективність поїздки.

2. Аналіз великих даних:

- Обробка та аналіз даних: ІІ та МН можуть обробляти великі обсяги даних з різних джерел, включаючи датчики на дорогах, GPS-дані з транспортних засобів та соціальні медіа.
- Виявлення закономірностей: Алгоритми машинного навчання допомагають ідентифікувати закономірності та тенденції в даних, що може використовуватися для покращення планування маршрутів та управління трафіком.

3. Інтерактивність та персоналізація:

- Персоналізовані рекомендації: ІІ може аналізувати переваги та звички користувачів, пропонуючи індивідуалізовані маршрути, які відповідають їхнім потребам.

- Інтерактивні функції: Системи можуть надавати користувачам інтерактивні опції для вибору маршрутів, включаючи можливість змінювати параметри пошуку в реальному часі.

Використання штучного інтелекту та машинного навчання в системах планування маршрутів значно підвищує їхню ефективність, точність та користувацький досвід, дозволяючи створювати більш гнучкі, адаптивні та інтелектуальні транспортні рішення.

Аналіз загальних підходів та методологій в системах планування маршрутів дощовилив зрозуміти, що системи планування маршрутів використовують різноманітні алгоритми та технології для оптимізації та адаптації до динамічних умов міського середовища. Від класичних алгоритмів, таких як Дейкстра та A*, до сучасних застосувань штучного інтелекту та машинного навчання, ці системи забезпечують ефективне прогнозування трафіку та динамічне маршрутизування. Мультимодальні та інтермодальні підходи дозволяють інтегрувати різні види транспорту в єдиний маршрут, покращуючи мобільність та знижуючи час у дорозі. Завдяки інтеграції великих даних та інтерактивних функцій, сучасні системи планування маршрутів стають все більш адаптивними, персоналізованими та здатними відповідати на змінні потреби користувачів [20, 21].

1.5 Дослідження можливостей інтеграції мобільних застосунків з базами даних інфраструктурних об'єктів

У рамках дослідження можливостей інтеграції мобільних застосунків з базами даних інфраструктурних об'єктів, особлива увага приділяється розробці та впровадженню інноваційних рішень, які забезпечують взаємодію між мобільними додатками та централізованими системами управління міською інфраструктурою. Це включає в себе не тільки доступ до актуальних даних про стан доріг, місця заторів, ремонтних робіт, але й інтеграцію з даними про громадський транспорт, доступність паркування, розташування зарядних станцій для електромобілів та інші критичні елементи інфраструктури.

Один з ключових аспектів – створення єдиної інформаційної платформи, яка б дозволяла обмінюватися даними між різними застосунками та інфраструктурними базами даних в реальному часі. Це може бути реалізовано через API, які забезпечують стандартизований доступ до інформації та її обробку. Такий підхід дозволяє не тільки підвищити ефективність мобільних застосунків, але й сприяє кращій координації між різними службами міста, покращуючи загальну мобільність та комфорт мешканців.

Крім того, важливим аспектом є забезпечення безпеки та захисту даних при їх обміні та зберіганні. Використання сучасних методів шифрування та аутентифікації може допомогти забезпечити конфіденційність інформації та захистити її від несанкціонованого доступу.

Також, важливою є розробка адаптивних інтерфейсів для мобільних застосунків, які б могли ефективно відображати різноманітні дані з інфраструктурних баз, забезпечуючи користувачам легкий доступ до необхідної інформації. Це включає в себе використання геолокаційних сервісів, інтерактивних карт, а також персоналізованих оповіщень, які можуть значно покращити досвід користувачів [22].

Завдяки інтеграції мобільних застосунків з базами даних інфраструктурних об'єктів, можливо значно підвищити ефективність управління міською інфраструктурою, забезпечити більшу прозорість та доступність інформації для громадян, а також сприяти створенню більш сталого та зручного міського середовища.

1.6 Дослідження потреб користувачів в автоматизації планування маршрутів

У рамках дослідження потреб користувачів в автоматизації планування маршрутів, основна увага приділяється вивченню та аналізу вимог та очікувань, які мають користувачі до сучасних систем навігації та маршрутизації. Основні аспекти цього дослідження включають вивчення зручності користування,

точності інформації, швидкості реакції системи на зміни у дорожніх умовах, а також можливості інтеграції з іншими сервісами та пристроями.

1. Зручність та інтуїтивність інтерфейсу: Користувачі прагнуть до простоти та інтуїтивно зрозумілого інтерфейсу, який дозволяє швидко вводити дані та отримувати необхідну інформацію без зайвих складнощів.
2. Швидкість та точність прогнозування: Автоматизовані системи планування маршрутів мають забезпечувати швидке та точне прогнозування трафіку, з урахуванням різних факторів, таких як час доби, погодні умови та поточні дорожні інциденти.
3. Адаптивність до змін у дорожніх умовах: Система має бути здатною оперативно реагувати на зміни, адаптуючи маршрут у реальному часі для уникнення заторів, дорожніх робіт та інших перешкод.
4. Інтеграція з іншими сервісами: Користувачі високо цінують можливість інтеграції планувальників маршрутів з іншими додатками та сервісами, такими як календарі, погодні додатки, а також системи громадського транспорту.
5. Персоналізація: Важливим аспектом є можливість налаштування системи під індивідуальні потреби користувача, включаючи збереження улюблених маршрутів, автоматичне планування поїздок заздалегідь та врахування особистих переваг щодо видів транспорту.
6. Безпека даних: З огляду на використання персональних даних та геолокації, користувачі вимагають високого рівня захисту своїх даних від несанкціонованого доступу [23, 24].

Результати такого дослідження допоможуть створити більш ефективну, зручну та безпечну систему планування маршрутів, яка відповідатиме актуальним потребам та вимогам користувачів, сприяючи підвищенню їхньої задоволеності та лояльності.

1.7 Уточнена постановка задачі

На основі проведеного аналізу інфраструктурних об'єктів та вивчення потреб користувачів, уточнена постановка задачі полягає у розробці та оптимізації системи планування маршрутів, яка інтегрується з мобільними застосунками та базами даних інфраструктурних об'єктів. Ця система має забезпечувати швидке та точне прогнозування трафіку, адаптивність до змін у дорожніх умовах, інтеграцію з іншими сервісами, персоналізацію для користувачів, а також високий рівень безпеки даних.

Перехід до наступного розділу, який зосереджений на дослідженні функціональних можливостей сучасних мобільних застосунків для планування маршрутів, дозволить детальніше розглянути, як існуючі рішення вирішують зазначені вище задачі та які інноваційні підходи можуть бути інтегровані в нову систему. Огляд популярних застосунків допоможе визначити ключові функції та можливості, які можуть бути використані або покращені в рамках розробки вдосконаленої системи планування маршрутів.

2 ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СУЧАСНИХ МОБІЛЬНИХ ЗАСТОСУНКІВ ДЛЯ ПЛАНУВАННЯ МАРШРУТІВ

2.1 Огляд мобільних застосунків для планування маршрутів

У даному розділі будуть детально досліджені чотири популярні мобільні застосунки для планування маршрутів: Google Maps, Waze, HERE WeGo та Citymapper. Вибір цих застосунків обумовлений їхньою високою популярністю, інноваційністю та різноманітністю функціональних можливостей, які вони пропонують користувачам. Аналіз буде проведений за такими параметрами:

1. Загальна інформація про додаток та розробників: Оцінка історії розробки, основних досягнень та стратегії розвитку кожного застосунку.
2. Ключові функціональні можливості: Аналіз основних функцій, які додаток пропонує для планування маршрутів, включаючи інтеграцію з громадським транспортом, пішохідні маршрути, велосипедні доріжки тощо.
3. Цінова політика: Огляд можливих витрат для користувачів, включаючи наявність платних функцій або підписок.
4. Методи заповнення бази даних інфраструктурних об'єктів: Дослідження того, як додатки збирають і оновлюють інформацію про інфраструктурні об'єкти, включаючи використання даних від користувачів, партнерство з урядовими агенціями, та інші джерела.
5. Методи та технології побудови маршрутів: Аналіз алгоритмів та технологій, які використовуються для обчислення оптимальних маршрутів, включаючи машинне навчання, штучний інтелект, та інші сучасні технології.

Цей аналіз допоможе визначити, які функції та технології можуть бути використані або покращені в рамках розробки нової системи планування маршрутів.

2.1.1 Огляд мобільного застосунку Google Maps

Google Maps, розроблений компанією Google, є одним з найпопулярніших сервісів картографії та навігації у світі. Запущений у 2005 році, Google Maps забезпечує детальні картографічні дані, покриваючи практично всі куточки планети. Сервіс постійно оновлюється та вдосконалюється, використовуючи дані з різних джерел, включаючи супутникові знімки, вуличні панорами та інформацію від користувачів [25].

Google Maps пропонує широкий спектр функцій, включаючи автомобільну, пішохідну, велосипедну навігацію та маршрути громадського транспорту. Сервіс також включає функції, такі як Street View, який дозволяє користувачам переглядати панорамні зображення вулиць на 360 градусів, інформацію про пробки в реальному часі, а також інтеграцію з Google Earth.

На рисунках 2.1 – 2.5 представлено деякі функціональні можливості мобільного додатку сервісу Google Maps.

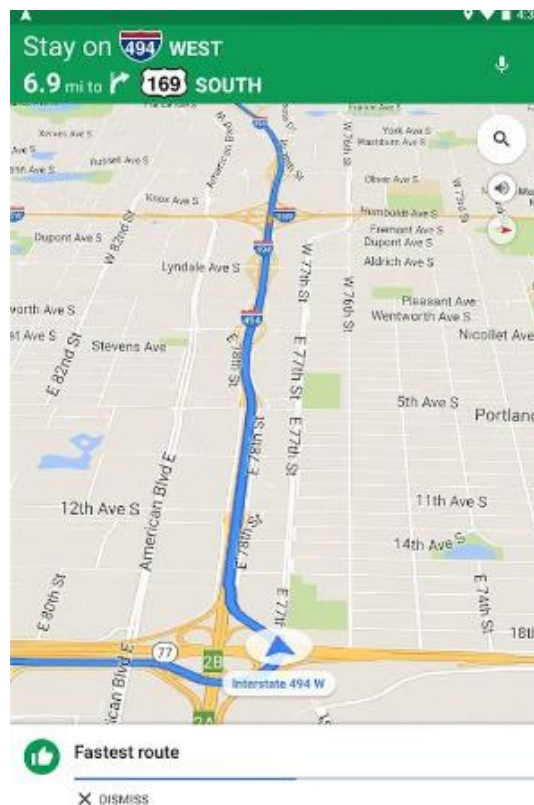


Рисунок 2.1 – Підбір найшвидшого маршруту в додатку Google Maps

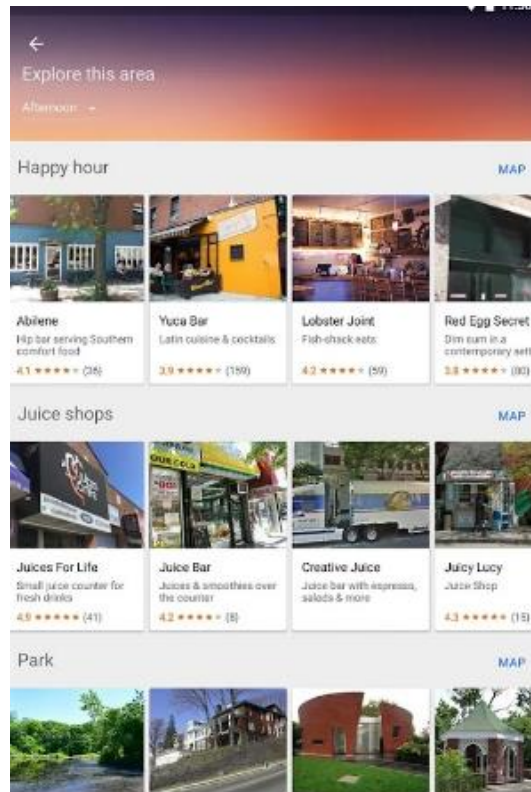


Рисунок 2.2 – Відображення об’єктів заданої локації в додатку Google Maps

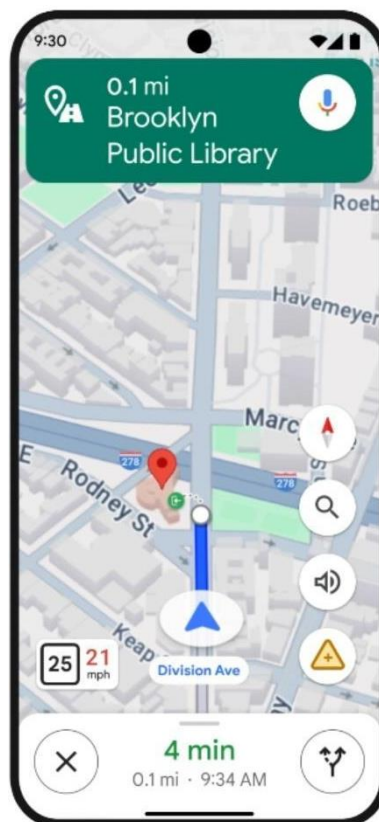


Рисунок 2.3 – Навігація до місця призначення в додатку Google Maps

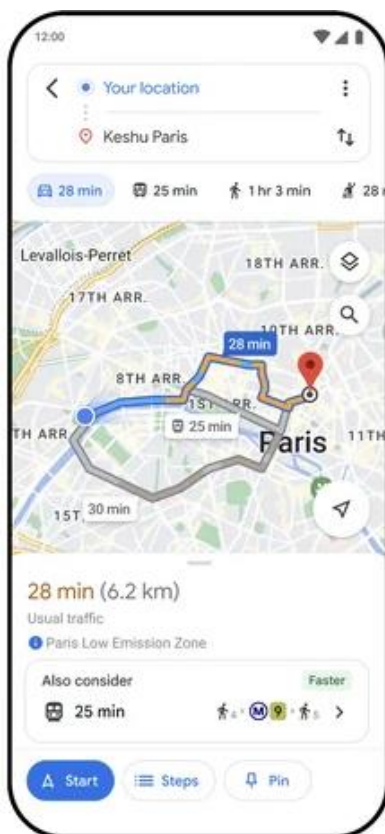


Рисунок 2.4 – Відображення варіантів маршруту в додатку Google Maps

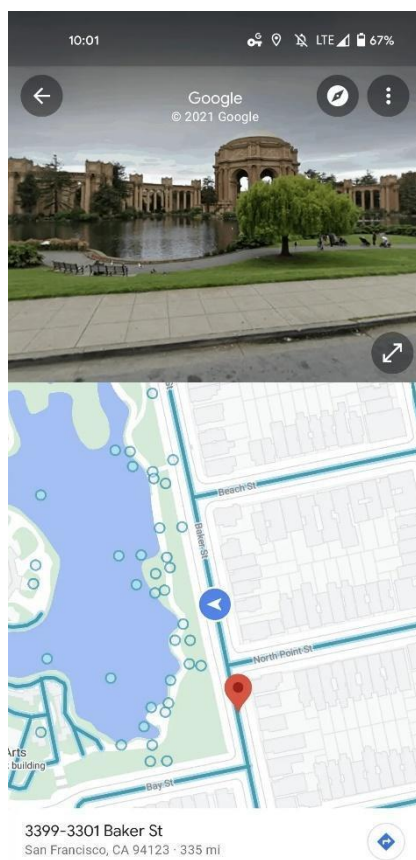


Рисунок 2.5 – Відображення місцевості в режимі Street View в додатку Google Maps

Сервіс Google Maps є безкоштовним для кінцевих користувачів, пропонуючи також платні рішення для бізнесу через Google Maps Platform, яка дозволяє інтегрувати картографічні дані та функції навігації у власні додатки та веб-сайти.

Google Maps використовує комбінацію автоматизованих алгоритмів та внесків від користувачів для оновлення та підтримки своєї бази даних. Користувачі можуть вносити зміни та доповнення через інструмент Map Maker, що дозволяє спільноті вносити свій вклад у точність та актуальність карт.

Сервіс використовує складні алгоритми маршрутизації, які враховують численні фактори, включаючи поточний стан трафіку, історичні дані про швидкість руху, погодні умови та інші змінні. Ці алгоритми постійно вдосконалюються завдяки машинному навчанню та штучному інтелекту, що дозволяє Google Maps ефективно оптимізувати маршрути для користувачів.

2.1.2 Огляд мобільного застосунку Waze

Waze є інноваційним мобільним застосунком для навігації, який був розроблений ізраїльською компанією Waze Mobile і пізніше придбаний Google у 2013 році. Застосунок здобув популярність завдяки своєму унікальному підходу до збору даних про дорожні умови в реальному часі, які надаються його активною спільнотою користувачів [26].

Основною особливістю Waze є його спільнота користувачів, які активно діляться інформацією про дорожній рух, аварії, поліцейські патрулі, та інші події на дорозі. Це дозволяє Waze надавати дуже точні та актуальні дані про умови на дорогах, що допомагає користувачам уникати заторів та інших перешкод. Крім того, Waze пропонує функції карпулінгу, інтеграцію з музичними сервісами та голосове керування.

На рисунках 2.6 – 2.8 представлено деякі функціональні можливості мобільного додатку сервісу Waze.

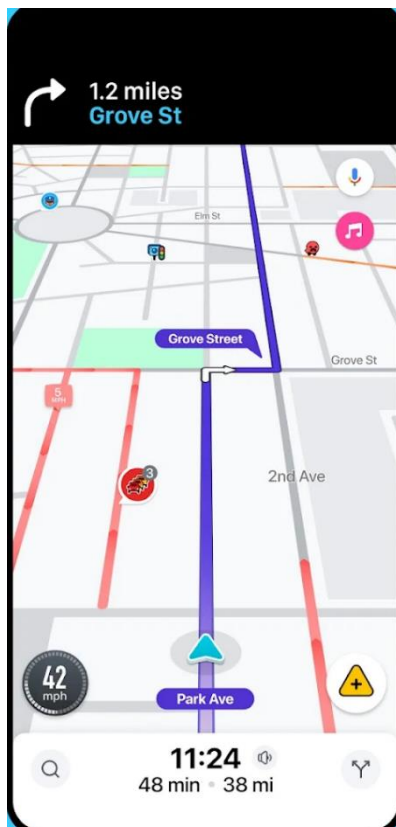


Рисунок 2.6 – Навігація до місця призначення в додатку Waze

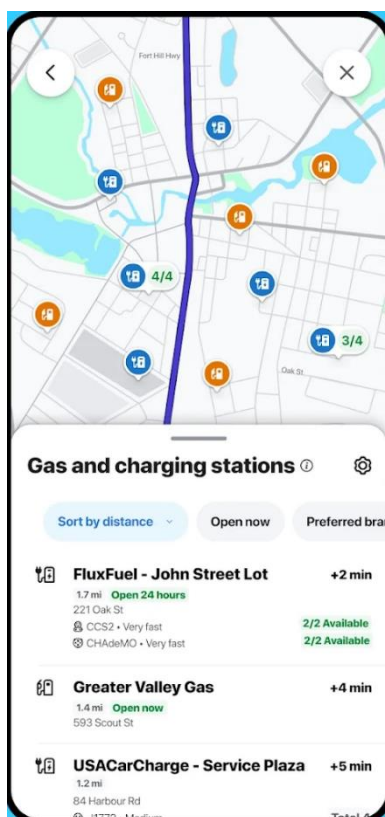


Рисунок 2.7 – Відображення інфраструктурних об'єктів в додатку Waze



Рисунок 2.8 – Відображення попередження про утруднення дорожніх умов в додатку Waze

Waze є безкоштовним застосунком, який заробляє на рекламі, показуючи рекламні оголошення у моменти, коли транспортний засіб зупинений.

Основою для оновлення даних у Waze є внески від користувачів, які можуть повідомляти про різні події та зміни на дорогах через інтерфейс застосунку. Це створює дуже динамічну та актуальну карту дорожніх умов, яка постійно оновлюється.

Waze використовує алгоритми, які аналізують велику кількість даних від користувачів для визначення найшвидших маршрутів у реальному часі. Ці алгоритми враховують не тільки стандартні параметри, такі як дистанція та передбачуваний час у дорозі, але й актуальну інформацію про дорожні умови, що дозволяє оптимізувати маршрути з урахуванням поточної ситуації на дорогах.

2.1.3 Огляд мобільного застосунку HERE WeGo

HERE WeGo, раніше відомий як HERE Maps, є мобільним застосунком для навігації, розробленим компанією HERE Technologies. Застосунок був створений як частина Nokia перед тим, як HERE Technologies стала незалежною компанією. HERE WeGo відомий своїми детальними картами та можливостями навігації, особливо в Європі, де він має сильну присутність [27].

Однією з визначних особливостей HERE WeGo є його здатність працювати в офлайн режимі, дозволяючи користувачам завантажувати карти та планувати маршрути без необхідності підключення до інтернету. Це робить застосунок особливо корисним у регіонах з обмеженим або відсутнім доступом до мобільного інтернету. Крім того, HERE WeGo пропонує вдосконалені можливості інтеграції з громадським транспортом, надаючи детальну інформацію про маршрути, розклади та зміни у русі громадського транспорту.

На рисунках 2.9 – 2.11 представлено деякі функціональні можливості мобільного додатку сервісу HERE WeGo.

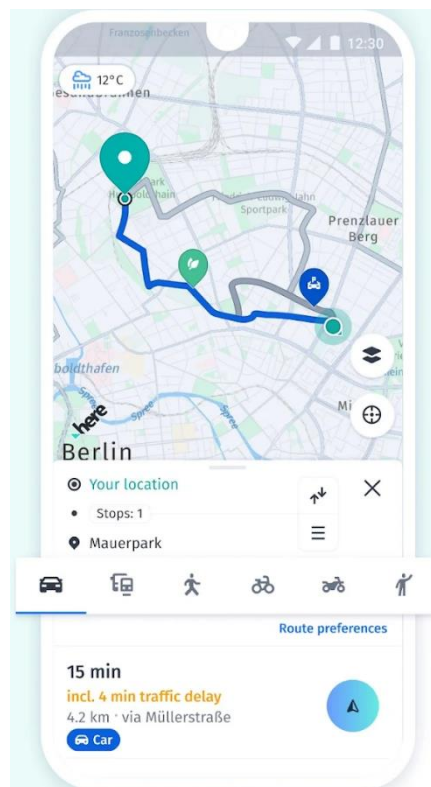


Рисунок 2.9 – Відображення варіантів маршруту в додатку HERE WeGo



Рисунок 2.10 – Навігація до місця призначення в додатку HERE WeGo

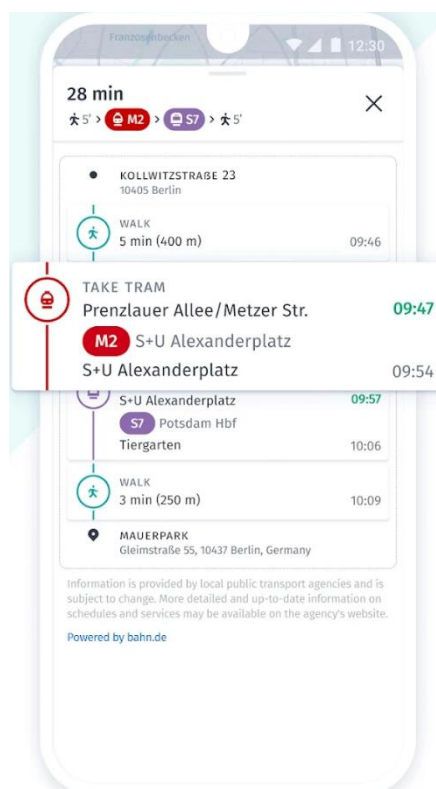


Рисунок 2.11 – Відображення попередження про утруднення дорожніх умов в додатку HERE WeGo

HERE WeGo є безкоштовним застосунком для основних навігаційних функцій, але також пропонує додаткові платні функції для більш спеціалізованих потреб, таких як детальніші карти або розширені можливості планування маршрутів.

HERE WeGo використовує різноманітні джерела для збору та оновлення своїх картографічних даних, включаючи супутникові знімки, дані від партнерів та внески від користувачів. Компанія також активно співпрацює з урядовими та комерційними організаціями для забезпечення точності та актуальності своїх карт.

HERE WeGo використовує комплексні алгоритми для планування маршрутів, які враховують не тільки дорожні умови, але й доступність громадського транспорту, пішохідні доріжки та велосипедні шляхи. Застосунок також адаптується до змін у дорожньому русі, використовуючи дані в реальному часі для оптимізації маршрутів.

2.1.4 Огляд мобільного застосунку Citymapper

Citymapper є спеціалізованим мобільним застосунком для планування маршрутів, який зосереджений на міському транспорті. Застосунок був запущений у 2011 році в Лондоні і з того часу розширив свою присутність до багатьох великих міст по всьому світу. Citymapper використовує дані про громадський транспорт, велосипедні доріжки, пішохідні маршрути та таксі для надання користувачам оптимальних маршрутів переміщення містом [28].

Основною особливістю Citymapper є його здатність інтегрувати різноманітні види транспорту в єдиний маршрут. Застосунок пропонує детальну інформацію про розклади громадського транспорту, включаючи автобуси, трамваї, метро та поїзди, а також можливості для каршеринга та велосипедного транспорту. Citymapper також надає інформацію про час в дорозі, вартість поїздки та калорії, які будуть спалені під час пішохідних маршрутів.

На рисунках 2.12 – 2.15 представлено деякі функціональні можливості мобільного додатку сервісу Citymapper.

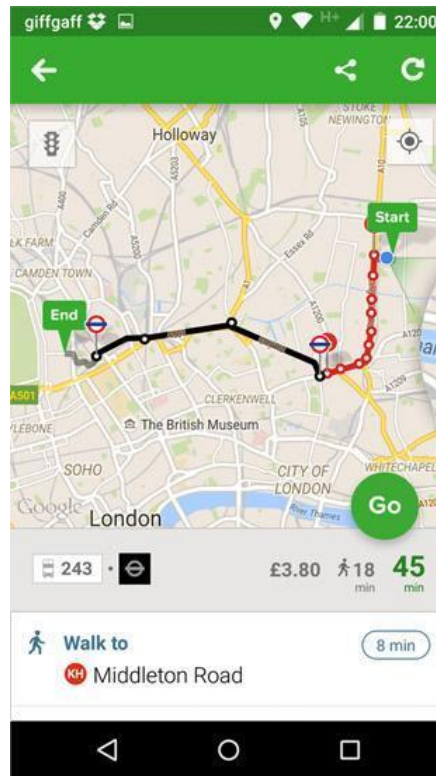


Рисунок 2.12 – Навігація до місця призначення в додатку Citymapper

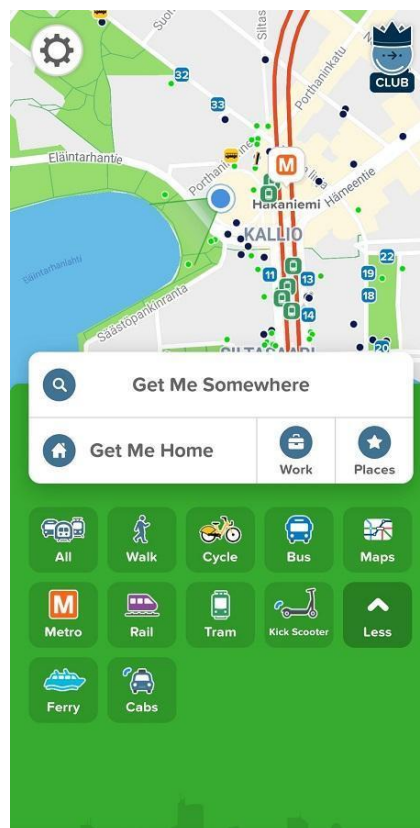


Рисунок 2.13 – Відображення інфраструктурних об'єктів в додатку Citymapper

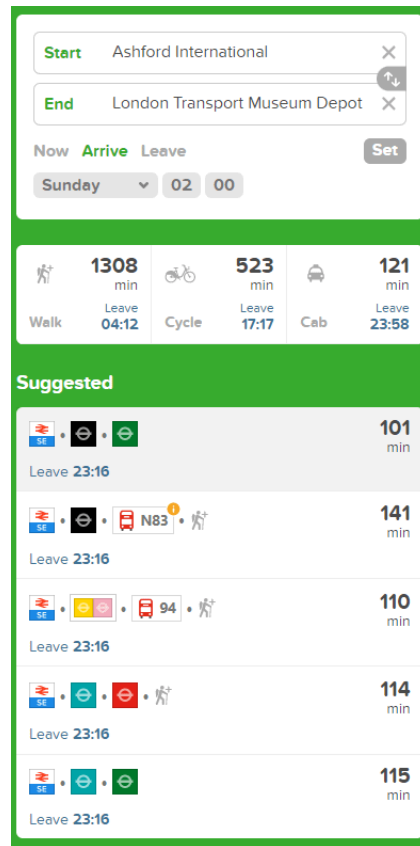


Рисунок 2.14 – Підбір маршруту руху громадським транспортом в додатку Citymapper

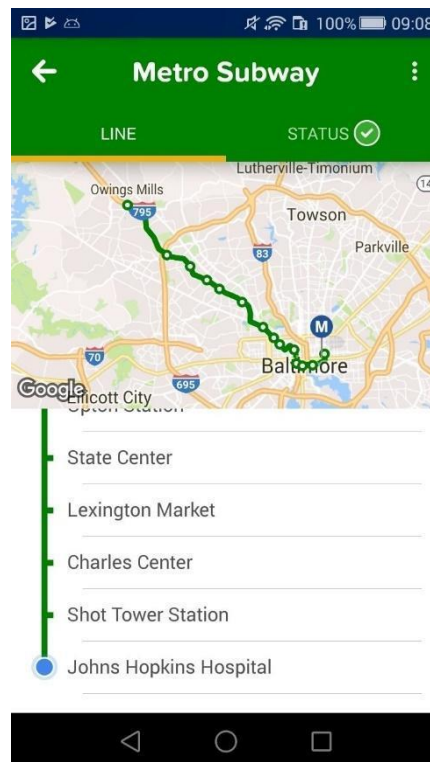


Рисунок 2.15 – Відображення зупинок громадського транспорту, за якими складено маршрут руху в додатку Citymapper

Citymapper пропонує безкоштовний доступ до основних функцій застосунку, але також має платну підписку Citymapper Pass, яка надає додаткові переваги, такі як знижки на громадський транспорт та інші послуги.

Citymapper активно співпрацює з міськими транспортними відомствами та іншими постачальниками даних для забезпечення точності та актуальності інформації про громадський транспорт. Застосунок також використовує відгуки та дані від користувачів для покращення своїх послуг.

Citymapper використовує складні алгоритми для інтеграції різних видів транспорту в єдиний маршрут, оптимізуючи час та вартість поїздки. Застосунок також адаптується до змін у розкладах та умовах руху, щоб надавати користувачам найактуальнішу інформацію. Це дозволяє Citymapper ефективно вирішувати складні задачі міської мобільності, роблячи переміщення містом більш зручним та ефективним.

Досліджені чотири системи навігації відіграють ключову роль у сучасному транспортному середовищі, надаючи користувачам різноманітні інструменти для планування маршрутів, інтеграції з громадським транспортом та управління подорожами. Кожна з систем має свої унікальні особливості:

Google Maps зосереджений на наданні детальних карт і маршрутів з широким спектром транспортних опцій, включаючи автомобільний, пішохідний, велосипедний транспорт та громадський транспорт.

Waze використовує спільноту користувачів для збору актуальної інформації про дорожній рух, що дозволяє користувачам обирати найшвидші маршрути в умовах змінних дорожніх умов.

HERE WeGo пропонує сильні можливості офлайн навігації та детальну інтеграцію з громадським транспортом, що робить його ідеальним вибором для міських подорожей.

Citymapper спеціалізується на міській мобільності, інтегруючи різні види транспорту в єдиний інтуїтивно зрозумілий інтерфейс, що робить планування маршрутів великих міст ефективним.

Кожна система має свої сильні сторони в залежності від потреб користувачів, але всі вони спрямовані на підвищення ефективності та комфорту подорожей.

Таблиця 2.1 - Зведена інформація для порівняння досліджених систем

Параметр / Система	Google Maps	Waze	HERE WeGo	Citymapper
Інтерфейс користувача	Інтуїтивний і зручний	Дружній і інформативний	Простий і функціональний	Яскравий і інтуїтивний
Функціональність	Висока	Висока	Висока	Висока
Інтеграція	Розширена	Базова	Розширена	Розширена
Безпека	Високий рівень	Високий рівень	Високий рівень	Високий рівень
Відгуки користувачів	Переважно позитивні	Переважно позитивні	Переважно позитивні	Переважно позитивні
Цінова політика	Безкоштовний, платні опції	Безкоштовний, реклама	Безкоштовний, платні опції	Безкоштовний, платні опції
Особливі функції	Супутникові знімки, Street View	Спільнота користувачів	Офлайн карти	Інтеграція міського транспорту
Мобільна оптимізація	Відмінна	Відмінна	Добра	Відмінна
Підтримка мов	Багатомовний	Багатомовний	Багатомовний	Багатомовний
Доступність платформ	iOS, Android	iOS, Android	iOS, Android	iOS, Android
Підтримка користувачів	24/7	Обмежена	24/7	24/7
Спільнота користувачів	Активна	Дуже активна	Менш активна	Активна
Кастомізація	Обмежена	Обмежена	Обмежена	Обмежена
Аналітика та звітність	Розширена	Базова	Середня	Розширена
Синхронізація з календарем	Так	Ні	Так	Так
Відеоінтеграція	Так	Ні	Ні	Ні

Таблиця 2.1 - Зведена інформація для порівняння досліджених систем (продовження)

Параметр / Система	Google Maps	Waze	HERE WeGo	Citymapper
Реакція на зворотний зв'язок	Швидка	Швидка	Середня	Швидка
Оновлення карт	Часті	Часті	Регулярні	Часті
Пішохідні маршрути	Відмінні	Обмежені	Добрі	Відмінні
Велосипедні маршрути	Відмінні	Обмежені	Добрі	Відмінні
Підтримка автономного режиму	Обмежена	Ні	Відмінна	Обмежена
Інтеграція з соціальними мережами	Так	Так	Ні	Так
Персоналізація	Висока	Середня	Середня	Висока
Доступ до громадського транспорту	Відмінний	Обмежений	Відмінний	Відмінний
Попередження про затори	Так	Так	Так	Так
Підтримка режиму водія	Відмінна	Відмінна	Добра	Добра

Досліджені параметри допомагають зрозуміти, як кожна система може бути адаптована до специфічних потреб користувачів, їх здатність до інтеграції з іншими інструментами, а також як вони можуть сприяти більш ефективному управлінню подорожами через різноманітні функції.

2.2 Аналіз результатів огляду мобільних застосунків

Аналіз результатів огляду мобільних застосунків для планування маршрутів дозволяє зрозуміти, як користувачі використовують такі застосунки, які функції є найбільш важливими для них, та які аспекти можуть бути покращені.

Інтерфейс користувача є першим, з чим стикається користувач при використанні застосунку. Зручність та інтуїтивність інтерфейсу впливають на загальне враження від застосунку. Наприклад, Google Maps та Citymapper надають інтуїтивно зрозумілі та візуально привабливі інтерфейси, що сприяє швидкому засвоєнню та ефективному використанню застосунків. Waze, з іншого боку, акцентує увагу на спільноті користувачів, що дозволяє отримувати актуальну інформацію про дорожні умови.

Функціональність застосунків має вирішальне значення для користувачів, які планують свої маршрути. Важливими аспектами є наявність різних режимів транспорту, інтеграція з громадським транспортом та можливість використання офлайн-карт. HERE WeGo та Citymapper вирізняються своєю здатністю інтегрувати громадський транспорт, що є критично важливим для користувачів у великих містах.

Waze є прикладом застосунку, де взаємодія зі спільнотою відіграє ключову роль. Користувачі активно діляться інформацією про дорожні умови, що допомагає іншим користувачам уникати заторів та інших проблем на дорозі. Цей аспект соціальної взаємодії є унікальною особливістю, яка відрізняє Waze від інших застосунків.

Сучасні застосунки для планування маршрутів використовують дані для аналізу поведінки користувачів та надання персоналізованих рекомендацій. Google Maps та Citymapper використовують аналітику для оптимізації маршрутів та покращення загального досвіду користувачів. Персоналізація також допомагає користувачам швидше знаходити найбільш відповідні опції для їхніх потреб.

Реакція на зворотний зв'язок від користувачів є важливим аспектом удосконалення застосунків. Швидка відповідь на проблеми та запитання користувачів підвищує загальне задоволення та лояльність до застосунку. Всі

розглянуті застосунки мають системи підтримки, але швидкість та ефективність відповідей може варіюватися.

Отже, аналіз підкреслює важливість інтуїтивного інтерфейсу, розширеної функціональності, інтеграції, спільноти, аналітики та зворотного зв'язку у створенні ефективних та корисних мобільних застосунків для планування маршрутів. Використання цих інсайтів може допомогти розробникам покращити існуючі застосунки та створити нові рішення, які краще задовольнятимуть потреби користувачів [29, 30].

На основі отриманих даних та аналізу, наступним кроком є проектування модуля користувача для мобільного застосунку, який буде частиною автоматизованої системи планування маршрутів. Цей модуль відіграє ключову роль у забезпеченні зручності та ефективності взаємодії користувачів з системою, а також у підвищенні загальної продуктивності системи.

3 ПРОЕКТУВАННЯ МОДУЛЯ КОРИСТУВАЧА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПЛАНУВАННЯ МАРШРУТІВ

Проектування модуля користувача в мобільному застосунку для планування маршрутів включає кілька ключових стадій, які забезпечують систематичний підхід до розробки. Кожна стадія спрямована на ретельне вивчення та реалізацію окремих аспектів модуля, що дозволяє створити ефективний та зручний інструмент для кінцевих користувачів. Основні стадії, які будуть описані в даному розділі, включають:

1. Аналіз функцій модуля користувача, на якому буде проведено детальний аналіз функціональних та нефункціональних вимог до модуля користувача. Аналіз допомагає визначити, які функції необхідно реалізувати для задоволення потреб користувачів.
2. Розробка UML-діаграм для модуля користувача дозволяє візуалізувати структуру та взаємодії всередині модуля. Основні діаграми, які були розроблені, включають: діаграму варіантів використання, що ілюструє основні функції модуля; діаграму класів, яка відображає внутрішню структуру та взаємозв'язки між класами; та діаграму послідовностей, яка демонструє взаємодію між різними компонентами системи в рамках конкретних сценаріїв. Крім того, була розроблена діаграма компонентів, яка показує структуру всієї системи, включаючи взаємозв'язки та залежності між усіма компонентами.
3. Розробка структури бази даних модуля користувача, яка підтримуватиме всі необхідні функції модуля користувача. Розробка структури бази даних включає визначення таблиць, їх взаємозв'язків та параметрів, що забезпечують зберігання та ефективне управління даними [31].

Ці стадії проектування дозволять систематично підійти до розробки модуля користувача, забезпечуючи його надійність, ефективність та високий рівень користувацької зручності. Вони допомагають забезпечити, що кінцевий продукт відповідає всім вимогам та очікуванням користувачів.

3.1 Аналіз функцій модуля користувача

Модуль користувача відповідає за взаємодію з користувачем, управління даними користувача та забезпечення індивідуалізованого досвіду використання застосунку. Нижче представлено аналіз основних функцій, які повинен включати модуль користувача.

1. Реєстрація та Авторизація

- Реєстрація: Користувачі повинні мати можливість створювати новий обліковий запис, вводячи основну інформацію, таку як ім'я, електронна пошта, та пароль. Також можливість реєстрації через соціальні мережі або Google-акаунт може значно спростити процес.
- Авторизація: Вхід в систему повинен бути безпечним та зручним, з можливістю входу через сторонні сервіси (наприклад, Facebook, Google).

2. Управління профілем

- Перегляд та редагування профілю: Користувачі повинні мати можливість переглядати та редагувати свої особисті дані, такі як ім'я, контактні дані, фотографію профілю.
- Управління налаштуваннями: Налаштування сповіщень, управління приватністю та інші налаштування повинні бути доступні для користувачів для кастомізації досвіду використання застосунку.

3. Інтеграція з іншими модулями

- Взаємодія з модулем планування маршрутів: Модуль користувача повинен інтегруватися з модулем планування маршрутів для забезпечення персоналізованих рекомендацій та збереження історії пошуку.
- Зберігання улюблених місць: Користувачі повинні мати можливість зберігати місця, які вони часто відвідують або планують відвідати, для швидкого доступу в майбутньому.

4. Безпека даних

- **Захист даних:** Важливо забезпечити високий рівень безпеки особистих даних користувачів, використовуючи сучасні методи шифрування та безпечної передачі даних.
- **Конфіденційність:** Політика конфіденційності та умови використання повинні бути чітко визначені та доступні для користувачів.

На основі визначених функцій та вимог до модуля користувача, наступним кроком у процесі проектування є візуалізація структури та взаємодій всередині модуля. Для цього буде використано UML (Unified Modeling Language) діаграми, які є стандартним засобом для графічного представлення архітектури програмних систем. UML діаграми дозволяють чітко і зрозуміло представити компоненти системи, їх взаємозв'язки, а також поведінку системи в різних сценаріях.

3.2 Розробка UML-діаграм для модуля користувача

Для глибшого розуміння архітектури модуля користувача та його інтеграції з іншими компонентами мобільного застосунку, було розроблено серію UML-діаграм, які разом забезпечують повне уявлення про архітектуру та динаміку модуля користувача, сприяючи ефективній розробці та інтеграції модуля в загальну систему мобільного застосунку.

3.2.1 Розробка діаграми компонентів

У контексті мобільного застосунку для планування маршрутів, UML-діаграма компонентів відображає ключові елементи системи та їх взаємозв'язки. Основні компоненти включають:

- **Модуль користувача:** Цей модуль відповідає за управління даними користувача, включаючи авторизацію та реєстрацію. Він обробляє вхідні дані користувачів, забезпечує безпеку облікових записів та управління профілями.
- **Модуль планування маршрутів:** Забезпечує функціональність для створення, редагування та оптимізації маршрутів. Цей модуль

використовує алгоритми для визначення оптимальних шляхів на основі різних критеріїв, таких як час подорожі, відстань та умови дороги.

- Модуль аналітики: Аналізує дані користувачів та використання застосунку для покращення сервісу. Цей модуль збирає статистику, таку як частота використання певних функцій, відгуки користувачів та інші метрики, які допомагають у прийнятті рішень щодо подальшого розвитку застосунку.
- База даних: Центральний репозиторій, який зберігає всі необхідні дані, включаючи інформацію про користувачів, маршрути, аналітичні дані та інше. База даних забезпечує високу доступність, консистентність та безпеку даних

UML-діаграму компонентів представлено на Рис. 3.1. Кожен з цих компонентів взаємодіє з іншими через визначені інтерфейси, що дозволяє забезпечити модульність та легкість інтеграції. Ця модульність є ключовою для підтримки гнучкості системи, дозволяючи легко модифікувати або розширювати функціональність застосунку без значних змін у інших частинах системи [32].

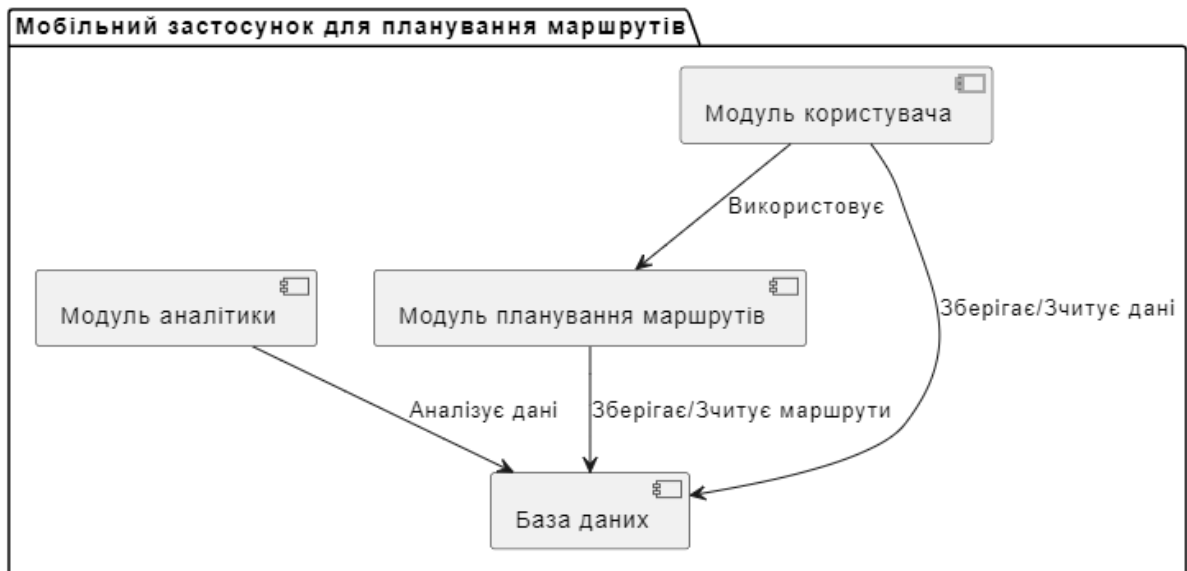


Рисунок 3.1 - UML-діаграма компонентів мобільного застосунку, що розробляється

3.2.2 Розробка діаграми варіантів використання

Діаграма варіантів використання допомагає визначити різні способи, якими користувачі (актори) можуть взаємодіяти з модулем користувача мобільного застосунку [33].

UML-діаграму варіантів використання модуля користувача мобільного застосунку, що розробляється, представлено на Рис. 3.2.

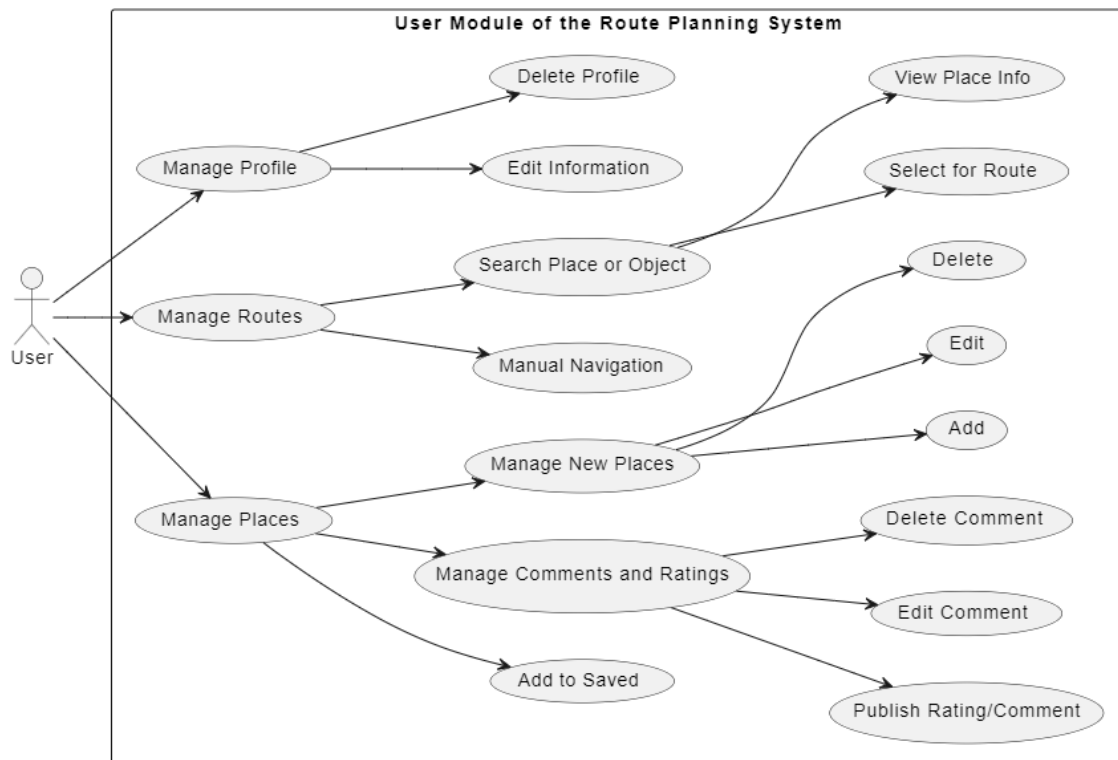


Рисунок 3.2 – UML-діаграма варіантів використання модуля користувача мобільного застосунку, що розробляється

Основні варіанти використання для модуля користувача включають:

1. **Управління профілем:** Користувачі можуть переглядати та редагувати свій профіль, включаючи особисті дані. Підваріанти включають редагування інформації та видалення профілю.
2. **Управління маршрутами:** Користувачі можуть взаємодіяти з модулем планування маршрутів. Підваріанти включають ручну навігацію по карті та пошук місця чи об'єкту, який має свої підваріанти: вибір для маршруту та перегляд інформації про місце.

3. Управління місцями: Користувачі можуть додавати місця до збережених, управляти коментарями та оцінками, а також управляти новими місцями. Управління коментарями та оцінками має підваріанти: опублікувати оцінку/коментар, редагувати та видалити. Управління новими місцями включає додавання, редагування та видалення місць.

3.2.3 Розробка діаграми класів

Діаграма класів є ключовим артефактом в процесі об'єктно-орієнтованого проектування та розробки програмного забезпечення. Вона відіграє важливу роль у визначенні структури системи, представляючи класи, їх атрибути, методи та взаємозв'язки між ними. Це дозволяє розробникам і архітекторам чітко уявляти, як компоненти системи взаємодіють один з одним, сприяючи більш ефективному проектуванню та імплементації [34].

Завдяки діаграмі класів, команда розробників може забезпечити консистентність та відповідність архітектури програмного забезпечення вимогам проекту. Це також спрощує процес інтеграції та тестування компонентів, оскільки кожен член команди може візуально оцінити взаємодії та залежності між класами.

Крім того, діаграма класів є важливим інструментом при підтримці та розширенні існуючих додатків. Вона допомагає новим розробникам швидше розібратися в проекті та ефективно вносити зміни, забезпечуючи високий рівень масштабованості та підтримки програмного забезпечення. Використання діаграми класів дозволяє зберігати код чистим і організованим, що є критично важливим для довгострокового успіху будь-якого програмного проекту.

UML-діаграму класів для мобільного додатку, що розробляється, представлено на Рис. 3.3. На цій діаграмі детально описано класи та зв'язки між ними, що дозволяє глибше зрозуміти структуру та взаємодії всередині системи.

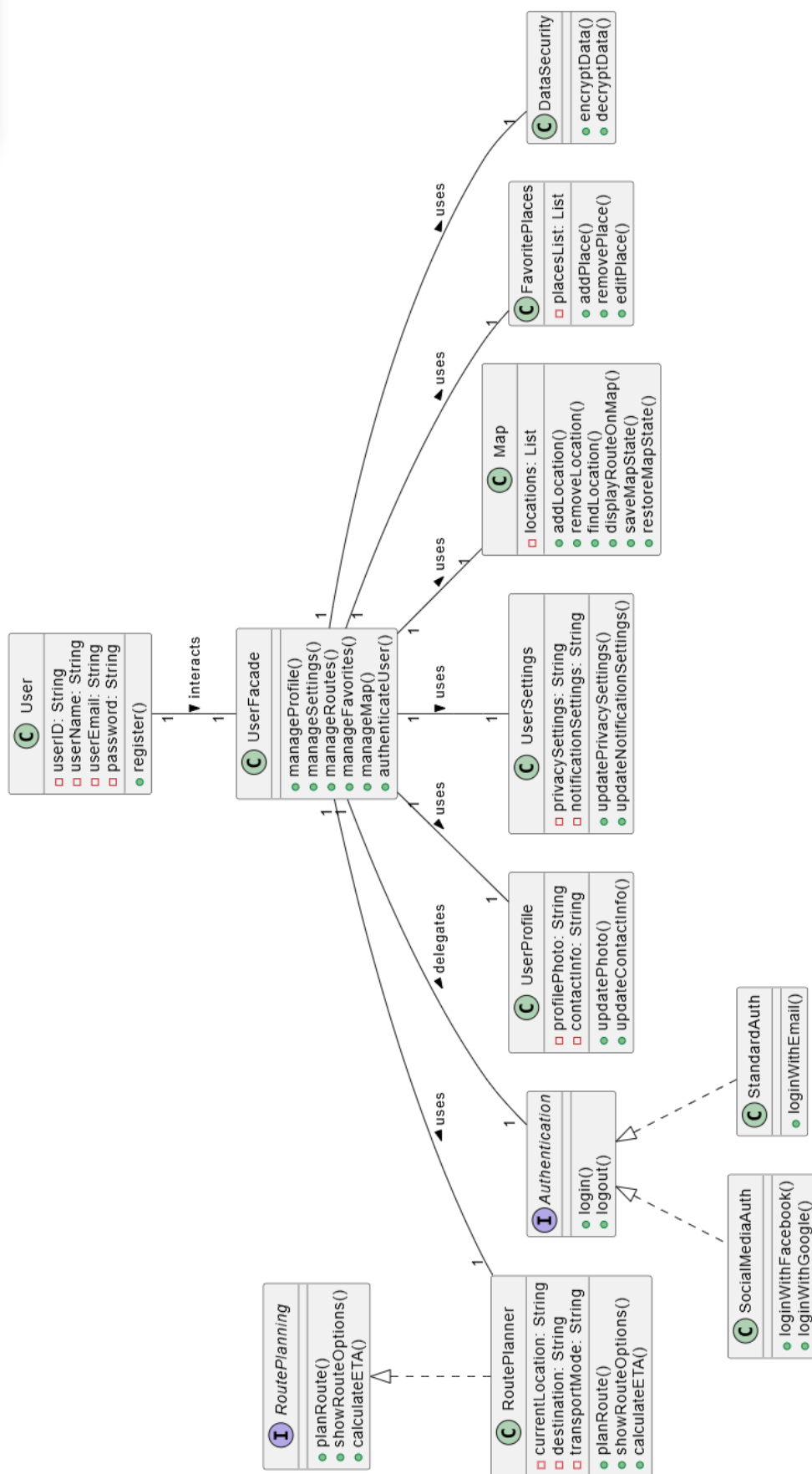


Рис
уно
к
3.3
-
UM
L-
діаг
рам
а
кла
сів
моб
іль
ног
о
зас
тос
унк
у,
що
роз
роб
ляє
тьс
я

Опис класів та зв'язків на діаграмі:

1. Класи Authentication, SocialMediaAuth, та StandardAuth:

- Authentication є інтерфейсом, що визначає методи 'login()' та 'logout()', які мають бути реалізовані в усіх класах, що забезпечують аутентифікацію.
- SocialMediaAuth та StandardAuth є конкретними реалізаціями цього інтерфейсу, де SocialMediaAuth дозволяє аутентифікацію через соціальні мережі, а StandardAuth – через стандартний email і пароль.

2. Клас User:

- Зберігає основну інформацію про користувача, таку як ID, ім'я, email та пароль. Має методи для реєстрації та перегляду профілю.

3. Класи UserProfile та UserSettings:

- UserProfile управляє інформацією профілю користувача, такою як фото та контактні дані.
- UserSettings дозволяє користувачам налаштовувати приватні налаштування та налаштування сповіщень.

4. Класи RoutePlanner, Map, та FavoritePlaces:

- RoutePlanner реалізує інтерфейс RoutePlanning для планування маршрутів.
- Map управляє локаціями на карті, включаючи додавання, видалення та пошук локацій.
- FavoritePlaces управляє списком улюблених місць користувача.

5. Клас UserFacade:

- Виступає як фасад, що об'єднує всі вищезазначені класи та функціональності, забезпечуючи єдиний інтерфейс для взаємодії користувача з системою. Це спрощує взаємодію та зменшує залежності між компонентами.

Зв'язки між класами:

- User взаємодіє з UserFacade, який, у свою чергу, використовує всі інші класи для забезпечення функціональностей системи.

- UserFacade делегує відповідальності до конкретних класів, таких як UserProfile, UserSettings, RoutePlanner, Map, FavoritePlaces, та DataSecurity для виконання специфічних задач.

Ця структура дозволяє забезпечити чистоту коду, модульність та легкість управління залежностями, що є важливим для ефективної підтримки та розвитку програмного забезпечення.

3.2.4 Розробка діаграми послідовностей

Діаграма послідовностей є важливим інструментом при проектуванні та аналізі систем, що вимагають чіткого розуміння взаємодій між об'єктами або компонентами. Ця діаграма демонструє, як об'єкти спілкуються між собою з певною метою, відображаючи порядок виконання операцій та передачу повідомлень у хронологічному порядку. Вона є незамінною при визначенні та оптимізації логіки бізнес-процесів, а також при ідентифікації потенційних проблем у взаємодіях між компонентами системи [35].

Використання діаграми послідовностей дозволяє розробникам глибше зануритись у деталі реалізації конкретних функціональних вимог, забезпечуючи точне відтворення процесів взаємодії. Це сприяє кращому розумінню того, як вхідні дані перетворюються у вихідні через різні шари та компоненти системи. Такий підхід є критично важливим для виявлення та усунення помилок на ранніх етапах розробки, що значно знижує витрати на підтримку та модифікацію програмного забезпечення.

UML-діаграму послідовностей для мобільного додатку, що розробляється, представлено на Рис. 3.4., де детально описано усі компоненти діаграми та їх взаємодію, що дозволяє глибше зрозуміти процеси в системі.

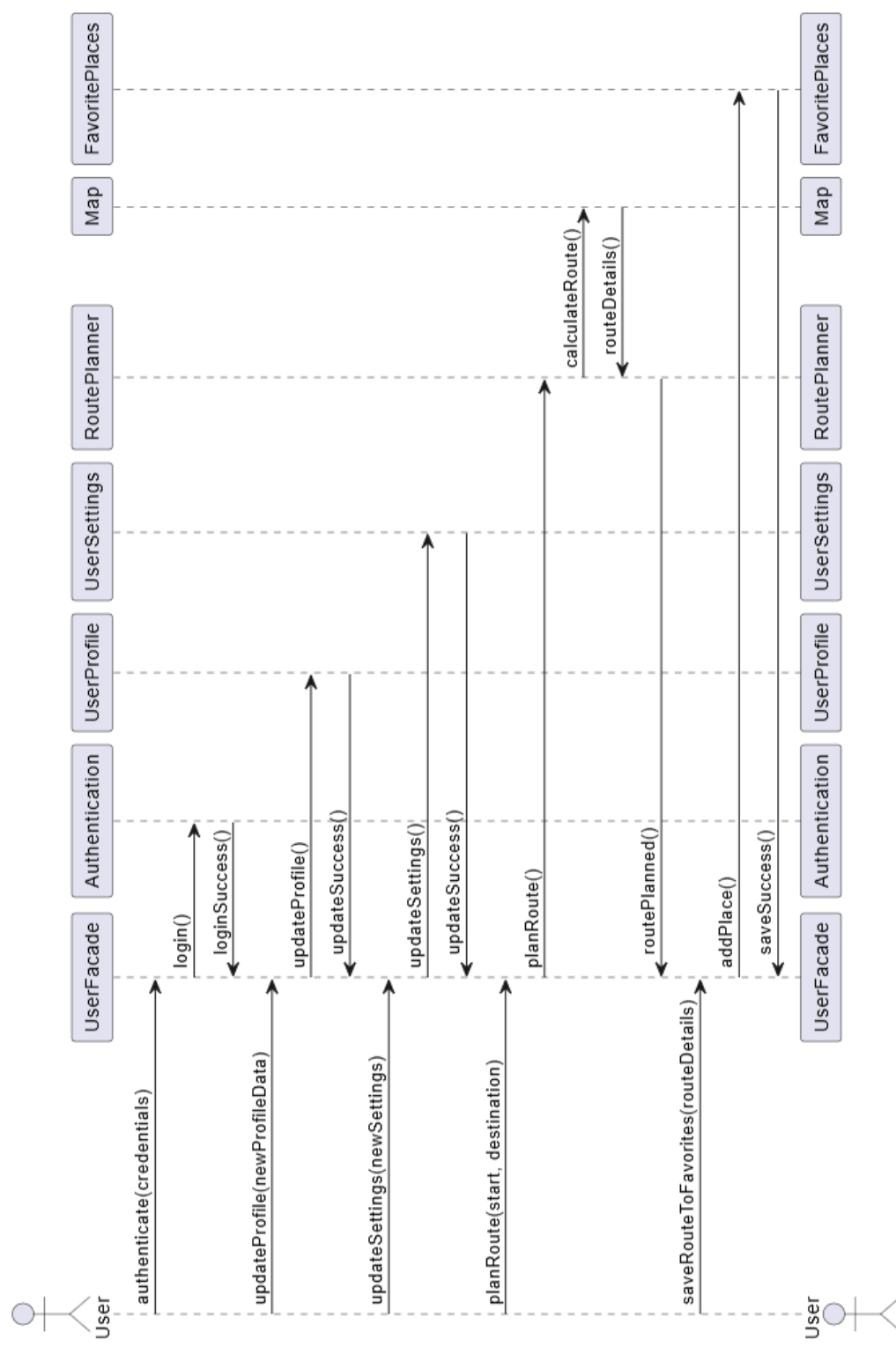


Рисунок 3.4 - UML-діаграма послідовностей мобільного застосунок, що розробляється

На діаграмі відображено взаємодію між різними компонентами системи в контексті типового сценарію користувача. Компоненти, що беруть участь у цьому процесі, включають:

- User: Актор, який ініціює взаємодію з системою, відправляючи запити через фасадний інтерфейс.
- UserFacade: Фасад, який приймає запити від користувача і делегує їх до відповідних компонентів системи.
- Authentication: Компонент, відповідальний за аутентифікацію користувача, перевіряє облікові дані та повертає результат.
- UserProfile: Компонент, що управляє профілем користувача, включаючи оновлення інформації про користувача.
- UserSettings: Компонент, який дозволяє користувачу змінювати налаштування системи.
- RoutePlanner: Компонент, відповідальний за планування маршрутів, використовує дані з Map для розрахунку оптимального шляху.
- Map: Компонент, який надає інформацію про географічні дані та маршрути.
- FavoritePlaces: Компонент, що дозволяє користувачу зберігати улюблені місця в системі.

Кожен з цих компонентів взаємодіє з іншими для виконання конкретних завдань, що відображено на діаграмі послідовностей. Наприклад, коли користувач намагається увійти в систему, запит від `User` передається до `UserFacade`, який звертається до `Authentication` для перевірки облікових даних. Після успішної аутентифікації, `Authentication` повертає підтвердження до `UserFacade`, яке потім передається назад до користувача.

Подібним чином, коли користувач хоче оновити свій профіль, він відправляє запит через `UserFacade`, який перенаправляє його до `UserProfile`. `UserProfile` обробляє запит, оновлює дані профілю в базі даних і відправляє підтвердження про успішне оновлення назад через `UserFacade`.

Для планування маршруту, `UserFacade` отримує запит від користувача і передає його до `RoutePlanner`, який в свою чергу взаємодіє з `Map` для розрахунку маршруту. Після отримання деталей маршруту від `Map`,

'RoutePlanner' передає цю інформацію назад до 'UserFacade', який інформує користувача.

Представлена діаграма послідовностей демонструє, як запити користувача обробляються системою від моменту їх надходження до повернення відповіді користувачеві. Вона важлива для розуміння того, як дані та контроль переходять між різними частинами системи, що допомагає виявити потенційні ускладнення або неефективності в процесах. Це дозволяє розробникам і аналітикам точно визначити, де можуть виникнути затримки або помилки, і вжити відповідних заходів для їх оптимізації або виправлення.

3.3 Розробка структури бази даних модуля користувача

Проектування бази даних можна порівняти з життєвим циклом розробки програмного забезпечення, оскільки обидва процеси включають схожі етапи: збір вимог, проектування, реалізацію, тестування, розгортання, підтримку та вдосконалення. Ось як ці етапи можуть бути застосовані до проектування бази даних:

1. Збір вимог

Як і в розробці програмного забезпечення, першим кроком у проектуванні бази даних є збір вимог. Це включає визначення потреб користувачів, аналіз бізнес-процесів, які потрібно підтримати, та встановлення обмежень, з якими база даних повинна бути сумісна. Цей етап важливий для забезпечення того, щоб база даних відповідала всім бізнес-вимогам і користувацьким очікуванням.

2. Концептуальне проектування

На цьому етапі розробляється високорівнева модель бази даних, яка відображає основні сутності та їхні взаємозв'язки. Це можна порівняти з архітектурним проектуванням у розробці програмного забезпечення, де визначаються основні компоненти системи та їх взаємодія.

3. Логічне проектування

На цьому етапі концептуальна модель перетворюється на логічну модель бази даних, яка включає визначення таблиць, полів, ключів та індексів. Це

аналогічно детальному проектуванню в програмному забезпеченні, де розробляються детальні специфікації для кожного компонента системи.

4. Фізичне проектування

На цьому етапі логічна модель адаптується до конкретної системи управління базами даних (СУБД). Це включає вибір типів даних, методів зберігання та оптимізацію продуктивності. Цей етап можна порівняти з реалізацією програмного забезпечення, де код розробляється та оптимізується для конкретної платформи.

5. Імплементация та тестування

База даних створюється на основі фізичної моделі, а потім проводиться тестування для перевірки її функціональності, продуктивності та безпеки. Це схоже на тестування програмного забезпечення, де продукт перевіряється на відповідність вимогам та виявлення помилок.

6. Розгортання та підтримка

Після тестування база даних розгортається в продуктивному середовищі. На цьому етапі також важлива підтримка та моніторинг бази даних для забезпечення її стабільної роботи. Це аналогічно підтримці програмного забезпечення, де продукт потребує регулярних оновлень та виправлень.

7. Вдосконалення

На основі зворотного зв'язку від користувачів та змін у бізнес-вимогах база даних може бути модифікована та оптимізована. Це схоже на ітеративний процес розробки програмного забезпечення, де продукт постійно вдосконалюється для задоволення зростаючих потреб користувачів [36-38].

Враховуючи специфіку роботи автоматизованої системи мобільного додатку, що розробляється, рекомендовано використовувати гібридний підхід до збереження та обробки даних, який включає:

- Локальну обробку для критичних в реальному часі даних. Наприклад, оновлення інтерфейсу користувача або відображення прогресу подорожі, що можуть оброблятися локально для забезпечення миттєвої відповіді.

- Обробку в хмарних сховищах для складних аналітичних обчислень. Наприклад, аналіз історичних даних подорожей або оптимізація маршрутів виконуються на сервері, де доступні більші обчислювальні ресурси.
- Комбіноване зберігання. Критичні дані зберігаються локально для швидкого доступу, тоді як менш чутливі або великі обсяги даних – зберігаються в хмарних сховищах для оптимізації використання простору та забезпечення резервного копіювання.

Для розробки бази даних, яка відповідає вимогам мобільного додатку з оновленням даних в реальному часі необхідно розробити ER-діаграму (Entity-Relationship Diagram), що означає створення високорівневої моделі, яка відображає основні сутності, що входять до системи, та зв'язки між ними [39, 40]. ER-діаграма допомагає візуалізувати структуру даних і є ключовим інструментом для розуміння та аналізу взаємодій між різними частинами бази даних.

ER-діаграму бази даних мобільного додатку, що розробляється, представлено на Рис. 3.5.

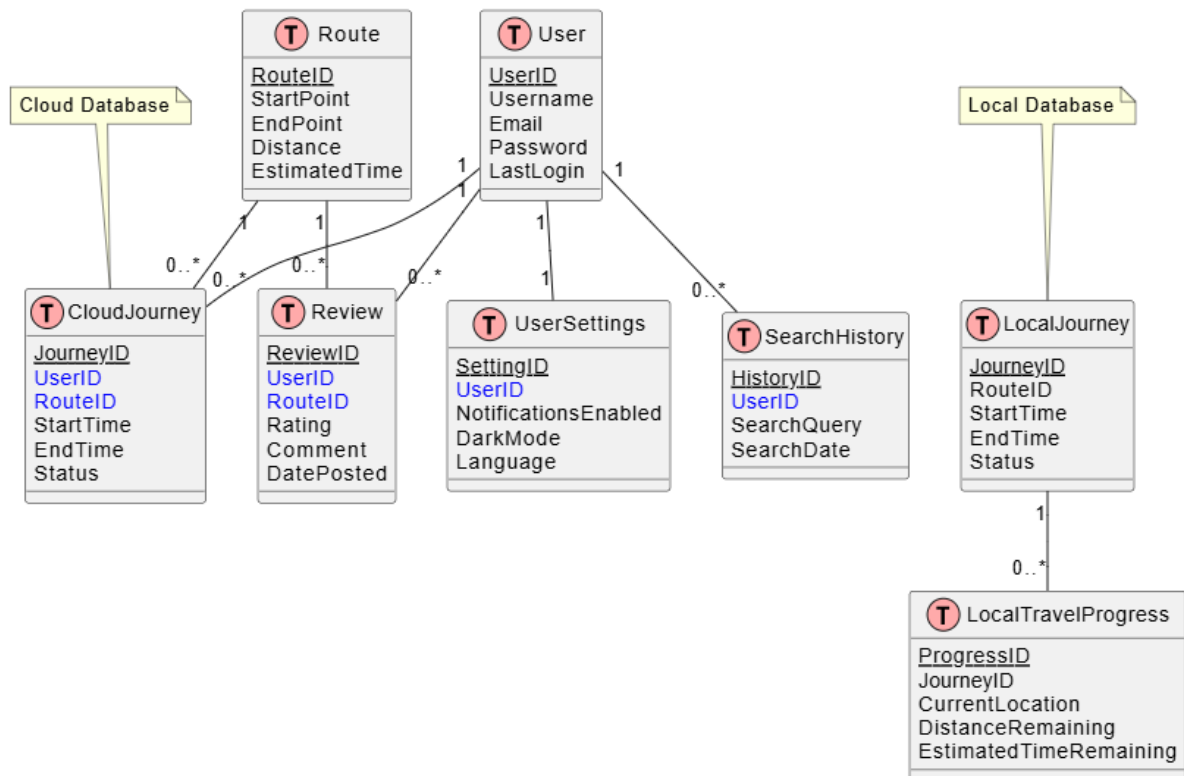


Рисунок 3.5 - ER-діаграма бази даних модуля користувача мобільного додатку для автоматизованої системи планування маршрутів

На діаграмі Cloud Database включає сутності, які повинні зберігатися у хмарному сховищі, такі як користувачі, маршрути, подорожі, відгуки, налаштування користувача та історія пошуку, Local Database – сутності, які зберігаються локально на пристрої користувача, такі як подорожі та прогрес подорожей.

Сутності Cloud Database:

1. User: Ця сутність представляє користувачів мобільного додатку. Вона містить атрибути, такі як `UserID`, `Username`, `Email`, `Password` та `LastLogin`. `UserID` є первинним ключем, що унікально ідентифікує кожного користувача.
2. Route: Сутність маршрутів, яка включає інформацію про початкову та кінцеву точки маршруту, а також про очікувану відстань та час подорожі. Атрибути: `RouteID` (первинний ключ), `StartPoint`, `EndPoint`, `Distance`, `EstimatedTime`.
3. Journey: Представляє конкретну подорож користувача. Зв'язана з `User` та `Route` через відповідні зовнішні ключі `UserID` та `RouteID`. Атрибути включають `JourneyID` (первинний ключ), `StartTime`, `EndTime`, та `Status`.
4. Review: Відгуки користувачів на маршрути. Містить `ReviewID` (первинний ключ), `UserID`, `RouteID`, `Rating`, `Comment`, `DatePosted`.
5. UserSettings: Налаштування користувача, такі як включення сповіщень, вибір теми додатку та мови. Атрибути: `SettingID` (первинний ключ), `UserID`, `NotificationsEnabled`, `DarkMode`, `Language`.
6. SearchHistory: Історія пошукових запитів користувача. Атрибути: `HistoryID` (первинний ключ), `UserID`, `SearchQuery`, `SearchDate`.

Сутності Local Database:

1. LocalJourney: Локально збережені дані про подорожі, що включають `JourneyID`, `RouteID`, `StartTime`, `EndTime`, `Status`. `JourneyID` використовується як зовнішній ключ для зв'язку з сутністю `Journey` у Cloud Database.

2. LocalTravelProgress: Прогрес подорожі, що зберігається локально для швидкого доступу. Атрибути: `ProgressID` (первинний ключ), `JourneyID`, `CurrentLocation`, `DistanceRemaining`, `EstimatedTimeRemaining`.

Зв'язки між сутностями

- User до Journey, Review, UserSettings, SearchHistory: Один до багатьох. Один користувач може мати багато подорожей, відгуків, одне налаштування та багато записів в історії пошуку.
- Route до Journey, Review: Один до багатьох. Один маршрут може бути асоційований з багатьма подорожами та відгуками.
- LocalJourney до LocalTravelProgress: Один до багатьох. Одна подорож може мати багато записів прогресу.

Для ефективної реалізації мобільного додатку з автоматизованою системою планування маршрутів, важливо обрати відповідні системи управління базами даних (СУБД) для локального та хмарного зберігання. Вибір СУБД впливає на продуктивність, масштабованість, доступність та безпеку даних.

Для локального зберігання даних в мобільному додатку важливо використовувати легковагову, надійну та ефективну СУБД, яка може працювати без постійного з'єднання з інтернетом. СУБД повинна підтримувати швидкий доступ до даних та їх синхронізацію з хмарною базою даних, коли з'єднання відновлено. Популярними варіантами для локального зберігання в мобільних додатках є SQLite та Realm, оскільки вони оптимізовані для високої продуктивності на мобільних пристроях і мають вбудовані механізми для легкої синхронізації даних.

Для хмарного зберігання важливо вибрати СУБД, яка забезпечує високу масштабованість, надійність та доступність. Хмарні СУБД, такі як Amazon RDS, Google Cloud SQL або Microsoft Azure SQL Database, пропонують гнучкість у виборі конфігурацій, автоматичне масштабування ресурсів, резервне копіювання та високий рівень безпеки.

3.4 Висновки до розділу

Для розуміння архітектури модуля користувача та його інтеграції з іншими компонентами мобільного застосунку, було розроблено UML-діаграми та структуру бази даних, які разом забезпечують повне уявлення про архітектуру та динаміку модуля користувача, сприяючи ефективній розробці та інтеграції модуля в загальну систему мобільного застосунку.

У розділі 4 будуть описані технології, що обрані для розробки модуля користувача в рамках магістерської кваліфікаційної роботи, в тому числі системи управління базами даних. Проте, важливо розуміти, що вибір СУБД та структури бази даних не є кінцевою точкою в процесі розробки. Як і будь-який інший компонент програмного засобу, база даних може вимагати адаптації та оптимізації у відповідь на зміни бізнес-вимог, адаптацію до ринкових умов, зміни у цінovій політиці, або нові політики захисту даних [41].

Така гнучкість є критично важливою, оскільки дозволяє програмному засобу залишатися конкурентоспроможним та ефективним у динамічному технологічному ландшафті. Вибір та конфігурація СУБД, які відповідають поточним потребам проекту, повинні також підтримувати можливість легкої модифікації та масштабування у майбутньому. Це включає в себе вибір таких технологій, які дозволяють ефективно управляти змінами без значних перерв у роботі системи та без втрати даних.

Таким чином, процес вибору технологій для бази даних має бути інтегрований з загальним життєвим циклом програмного засобу, що було вказано на початку цього підрозділу. Це дозволяє забезпечити не тільки відповідність поточним технічним та бізнес-вимогам, але й гнучкість для майбутньої адаптації та розвитку.

4 РОЗРОБКА МОДУЛЯ КОРИСТУВАЧА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПЛАНУВАННЯ МАРШРУТІВ

4.1 Вибір технологій для розробки модуля користувача

Для розробки модуля користувача мобільного застосунку автоматизованої системи планування маршрутів було обрано низку технологій, які забезпечують ефективність, безпеку та адаптивність додатку. Вибір технологій ґрунтується на потребах проекту, вимогах до функціональності та забезпеченні високої продуктивності.

База даних:

SQLite було обрано як систему управління базами даних для локального зберігання інформації на мобільних пристроях. Ця технологія дозволяє ефективно зберігати дані про маршрути, інфраструктурні об'єкти та користувацькі налаштування, забезпечуючи швидкий доступ та високу надійність без потреби в постійному з'єднанні з інтернетом.

Для хмарного зберігання даних модуля користувача було обрано Google Cloud SQL. Вибір ґрунтується на потребі забезпечення високої доступності, масштабованості та безпеки даних. Google Cloud SQL підтримує різні реляційні бази даних, що дозволяє легко інтегрувати цю систему з існуючою архітектурою мобільного додатку. Крім того, Google Cloud SQL пропонує автоматичне резервне копіювання, шифрування даних у спокої та в транзиті, а також високу продуктивність, що є критично важливим для забезпечення надійної роботи мобільного застосунку в режимі реального часу.

Розробка мобільного додатку: Kotlin та Java для Android

Для розробки мобільного застосунку було вибрано мови програмування Kotlin та Java, які є стандартом для розробки додатків на платформі Android. Ці мови забезпечують велику гнучкість у створенні адаптивних інтерфейсів та високу продуктивність додатку. Kotlin та Java мають велику спільноту та підтримку, що сприяє легкості вирішення потенційних проблем під час розробки [42].

Серверні рішення: Google Cloud

Для зберігання даних та обробки запитів від мобільного додатку було обрано хмарні сервіси Google Cloud. Це рішення дозволяє забезпечити масштабованість, високу доступність та надійність сервісу, а також ефективне управління ресурсами. Серед використаних сервісів, як вже було зазначено, використовується Google Cloud SQL для реляційного зберігання даних, а також Google Kubernetes Engine для оркестрації контейнерів, що забезпечує гнучке масштабування та управління мікросервісами. Ці сервіси взаємодіють між собою, створюючи надійну та ефективну платформу для обробки та зберігання даних, що є критично важливим для забезпечення стабільної роботи мобільного застосунку в умовах високого навантаження.

Засоби безпеки: Шифрування даних та протоколи безпеки

Забезпечення безпеки даних є критично важливим аспектом. Використання методів шифрування даних на мобільному додатку та сервері, а також застосування протоколів безпеки, таких як HTTPS та OAuth, допомагає забезпечити конфіденційність, цілісність інформації та безпечне з'єднання між користувачем та сервером [43].

Ці технології були обрані з метою створення надійного, безпечного та ефективного мобільного застосунку для планування маршрутів, який відповідає сучасним вимогам та тенденціям у технологічній сфері.

4.2 Розробка програмних компонентів модуля користувача

Розробка програмних компонентів модуля користувача мобільного застосунку включає створення декількох ключових компонентів, які забезпечують функціональність та взаємодію з користувачем. Нижче описано основні компоненти, їхні функції та методи реалізації.

1. Інтерфейс користувача (UI)

Інтерфейс користувача мобільного застосунку для автоматизованої системи планування маршрутів розроблено з акцентом на забезпечення інтуїтивно зрозумілого та зручного користувацького досвіду. Використання сучасних

підходів до UX/UI дизайну дозволяє створити адаптивний інтерфейс, який ефективно відповідає на різноманітні потреби користувачів.

Компоненти Material Design

Для розробки UI було обрано компоненти Material Design, які забезпечують високу адаптивність та атрактивність візуального досвіду. Material Design пропонує єдиний консистентний набір принципів та елементів, що дозволяє створювати логічні та зрозумілі інтерфейси. Ці компоненти включають анімації, переходи, глибинні ефекти (такі як тіні та освітлення), які допомагають користувачам зрозуміти ієрархію та навігацію додатку [44, 45].

Елементи інтерфейсу

- Карта для відображення маршрутів: Центральний елемент інтерфейсу, який дозволяє користувачам візуально сприймати заплановані маршрути та інфраструктурні об'єкти. Карта інтегрована з функціями масштабування та панорамування для зручності користувачів.
- Панель пошуку: Розташована у верхній частині екрану, панель пошуку дозволяє користувачам швидко знаходити конкретні місця, адреси або об'єкти, вводячи текстові запити.
- Меню налаштувань: Надає доступ до різноманітних налаштувань застосунку, таких як вибір мови, управління повідомленнями та персоналізація інтерфейсу.
- Інформаційні вікна: При виборі конкретного маршруту або об'єкта на карті, відображаються детальні інформаційні вікна, які містять всю необхідну інформацію, таку як відстань, приблизний час у дорозі, доступність транспорту та інші корисні дані.

Реакція на користувацькі дії

Інтерфейс реагує на різні користувацькі дії, такі як тапи, свайпи та жести зуму, забезпечуючи при цьому плавність та відгук. Це створює динамічний та залучаючий досвід, що сприяє більшій взаємодії з додатком.

2. Управління базою даних

Компонент управління базою даних є критично важливим для забезпечення ефективної роботи мобільного застосунку. Використання SQLite дозволяє локально зберігати та витягувати дані, що необхідні для функціонування системи планування маршрутів.

Створення схеми бази даних

Схема бази даних була розроблена для зберігання даних про маршрути, інфраструктурні об'єкти та користувацькі налаштування. Ось приклад структури таблиці для зберігання інформації про маршрути:

```
CREATE TABLE Routes (
    RouteID INTEGER PRIMARY KEY AUTOINCREMENT,
    StartPoint TEXT NOT NULL,
    EndPoint TEXT NOT NULL,
    Distance REAL NOT NULL,
    EstimatedTime REAL NOT NULL
);
```

Оптимізація запитів

Для забезпечення швидкого доступу до даних та ефективності застосунку, запити до бази даних були оптимізовані. Наприклад, запит для отримання деталей маршруту виглядає наступним чином:

```
SELECT * FROM Routes WHERE StartPoint = 'Kyiv' AND EndPoint = 'Lviv';
```

Цей запит використовує індексацію для швидкого пошуку маршрутів за початковою та кінцевою точками.

Забезпечення цілісності даних

Для забезпечення цілісності даних використовуються транзакції. Це дозволяє уникнути втрати даних або їх пошкодження у випадку помилок або збоїв під час виконання операцій. Наприклад, оновлення інформації про маршрут виконується за допомогою транзакції [46]:

```
BEGIN TRANSACTION;
```

```
UPDATE Routes SET Distance = 540.5, EstimatedTime = 7.0 WHERE RouteID = 1;
COMMIT;
```

Обробка помилок

Обробка помилок відіграє важливу роль у забезпеченні стабільності застосунку. Нижче наведено приклад коду на Java для обробки помилок при роботі з базою даних SQLite, який включає перевірку статусу операцій та відкат транзакцій у випадку виявлення помилок:

```
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;

SQLiteDatabase db = this.getWritableDatabase();

db.beginTransaction();
try {
    // Виконання операцій з базою даних
    db.execSQL("UPDATE Routes SET Distance = 540.5 WHERE RouteID = 1", new Object[]{newDistance,
routeId});
    db.setTransactionSuccessful();
} catch (SQLiteException e) {
    // Логування помилки
    Log.e("Database Error", "An error occurred: " + e.getMessage());
} finally {
    db.endTransaction();
}
```

У цьому прикладі використовується метод `beginTransaction()` для початку транзакції. Операції з базою даних виконуються в блоку `try`, і якщо вони успішні, викликається метод `setTransactionSuccessful()`, який позначає транзакцію як успішну. У випадку виникнення помилки, виключення `SQLiteException` перехоплюється, і помилка логується. Незалежно від результату, транзакція завершується методом `endTransaction()`, який відміняє зміни, якщо транзакція не була позначена як успішна [47].

3. Логіка обробки даних

Компонент логіки обробки даних відіграє ключову роль у функціонуванні мобільного застосунку для планування маршрутів. Він забезпечує обробку вхідних даних від користувача та взаємодію з сервером для отримання та оновлення даних. Основні аспекти цього компонента включають алгоритми оптимізації маршрутів та персоналізацію рекомендацій маршрутів.

Алгоритми оптимізації маршрутів

Одним з ключових алгоритмів для оптимізації маршрутів, який був впроваджений у застосунку, є алгоритм Дейкстри. Цей алгоритм використовується для знаходження найкоротшого шляху між двома точками на карті. Реалізація алгоритму здійснена на Java і включає в себе використання пріоритетної черги для ефективного вибору наступної вершини для обробки [48, 49]:

```
import java.util.PriorityQueue;

public class DijkstraAlgorithm {
    public static int[] dijkstra(int[][] graph, int start) {
        int[] distances = new int[graph.length];
        boolean[] visited = new boolean[graph.length];
        PriorityQueue<Integer> queue = new PriorityQueue<>();

        for (int i = 0; i < distances.length; i++) {
            distances[i] = Integer.MAX_VALUE;
        }
        distances[start] = 0;
        queue.add(start);

        while (!queue.isEmpty()) {
            int current = queue.poll();
            visited[current] = true;

            for (int i = 0; i < graph[current].length; i++) {
                if (!visited[i] && graph[current][i] != 0 && distances[current] + graph[current][i] < distances[i]) {
                    distances[i] = distances[current] + graph[current][i];
                    queue.add(i);
                }
            }
        }
    }
}
```

```

    }
    return distances;
  }
}

```

Персоналізація рекомендацій маршрутів

Персоналізація рекомендацій маршрутів реалізована на основі попередніх виборів користувача та його налаштувань. Система аналізує збережені дані про вибрані маршрути та враховує індивідуальні переваги користувачів, щоб автоматично пропонувати оптимальні маршрути.

4. Модуль безпеки

Модуль безпеки в мобільному застосунку відіграє критичну роль у забезпеченні захисту даних користувача та безпечному з'єднанні з сервером. Для досягнення цих цілей були впроваджені наступні технології та методи:

Шифрування даних

Для захисту чутливих даних, які передаються між мобільним застосунком та сервером, використовується алгоритм шифрування AES (Advanced Encryption Standard). AES є одним з найбільш широко використовуваних і сильних алгоритмів шифрування, який забезпечує високий рівень безпеки.

Готова реалізація AES доступна в бібліотеці криптографічних функцій. Її використання дозволило забезпечити стандартизований та перевірений метод шифрування, що допомагає уникнути помилок, які можуть виникнути при самостійній реалізації алгоритму шифрування, та забезпечує сумісність з іншими системами та пристроями.

Використання AES гарантує, що навіть у випадку перехоплення даних, вони залишаться незрозумілими для несанкціонованих осіб, оскільки розшифрування без відповідного ключа є вкрай складним завданням [50, 51].

Протоколи безпеки

Застосунок використовує протокол HTTPS для захищеного з'єднання з сервером, що гарантує конфіденційність та цілісність переданих даних. Також впроваджено протокол OAuth для управління доступом до ресурсів сервера, що

дозволяє користувачам безпечно ділитися своїми даними без необхідності розкривати свої аутентифікаційні дані.

Ці заходи дозволяють забезпечити високий рівень безпеки в мобільному застосунку, захищаючи особисті дані користувачів та забезпечуючи безпечне та конфіденційне з'єднання з сервером.

5. Тестування та оптимізація

Тестування та оптимізація є ключовими аспектами розробки, оскільки вони забезпечують стабільність, безпеку та високу продуктивність системи. Під час роботи над мобільним застосунком у рамках магістерської кваліфікаційної роботи основний акцент було зроблено на юніт-тестування, інтеграційне тестування та профілювання додатку. Автоматизовані тести на даному етапі розробки не використовувались активно, оскільки їх розробка та підтримка вимагають значних ресурсів, що є більш доцільним для довготривалих проєктів або проєктів, які перебувають у стадії активної підтримки та розвитку [52].

Юніт-тестування

Для кожного модуля та компоненту застосунку розроблені юніт-тести, які перевіряють коректність роботи окремих функцій та методів. Юніт-тестування допомагає забезпечити, що зміни в коді не порушують існуючу функціональність і що кожен компонент виконує свої завдання належним чином.

Інтеграційне тестування

Після перевірки окремих модулів проводилось інтеграційне тестування, щоб забезпечити, що всі компоненти застосунку ефективно взаємодіють один з одним. Це було важливим для виявлення проблем, які можуть виникнути лише при взаємодії різних частин системи.

Профілювання додатку

Інструменти для профілювання додатку дозволили ідентифікувати "вузькі місця" та оптимізувати код для покращення швидкодії та ефективності застосунку.

У процесі розробки мобільного застосунку було використано систему контролю версій Git, що дозволило ефективно управляти кодом та забезпечити його цілісність. Використання Git дало можливість зберігати історію змін, легко відновлювати попередні версії коду та експериментувати з новими функціями, не ризикуючи стабільністю основної кодової бази. Також, використання Git сприяло систематизації робочого процесу, дозволяючи зосередитися на поетапному вдосконаленні застосунку [53].

4.3 Налаштування та тестування роботи модуля користувача

У даному підрозділі представлено детальний огляд налаштувань та тестування роботи модуля користувача мобільного додатку для автоматизованої системи планування маршрутів. Для забезпечення зручності користувачів та ефективності використання додатку, було проведено тестування функціональних можливостей, результати якого відображені на наступних рисунках.

Неавторизований користувач мобільного додатку має можливість перегляду мапи та інфраструктурних об'єктів, що зображено на Рис. 4.1. Користувач може авторизуватися в системі за допомогою Google акаунту або за допомогою логіну та паролю, а також зареєструватися в системі створивши новий акаунт – рисунок 4.2.

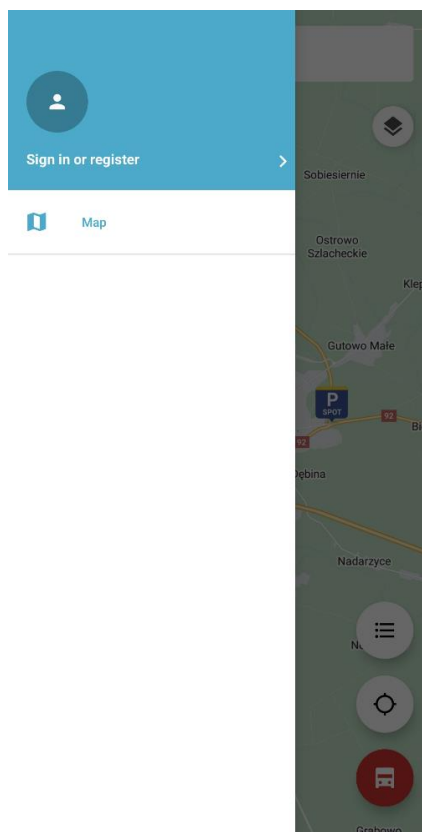


Рисунок 4.1 – Функціональні можливості мобільного додатку для неавторизованого користувача

Don't have an account? [Create one](#)

[Forgot your password?](#)

Рисунок 4.2 – Варіанти авторизації до мобільного додатку

Після успішної авторизації користувач може побачити та змінити дані свого профілю, що представлено на Рис. 4.3, а також скористатися функціональними можливостями, що представлені на Рис. 4.4 та усіх подальших рисунках.

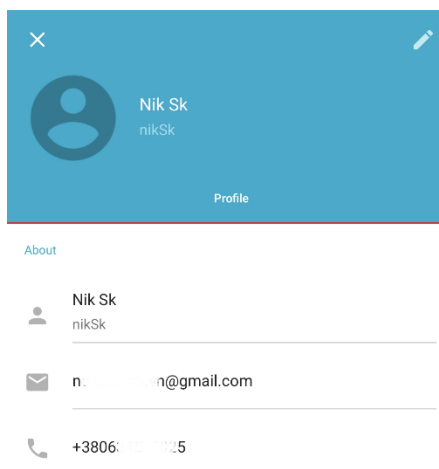


Рисунок 4.3 – Відображення даних користувача після авторизації

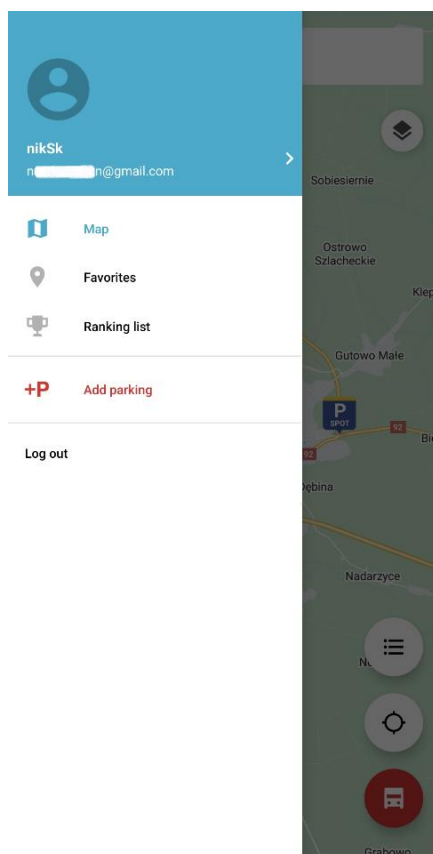


Рисунок 4.4 – Функціональні можливості мобільного додатку для авторизованого користувача

На Рис. 4.5 представлено відображення Google мапи, що інтегрована з мобільним додатком. Рекомендації інфраструктурних об'єктів, що будуть відображені на мапі можна відфільтрувати скориставшись фільтром, що зображений на Рис. 4.6.

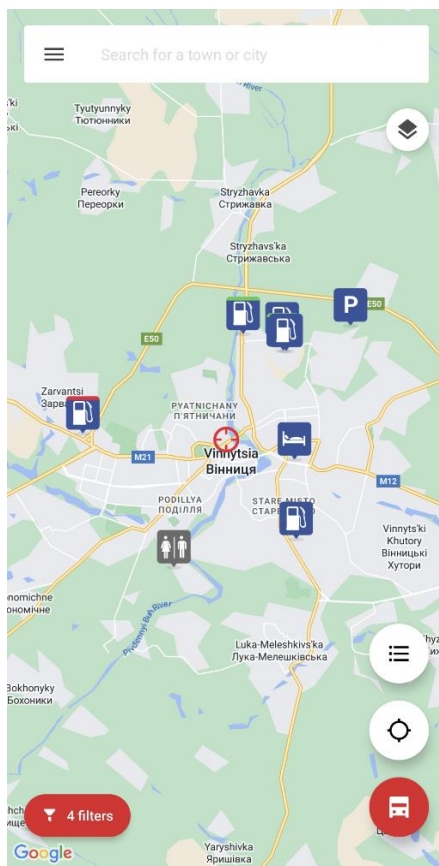


Рисунок 4.5 – Відображення Google мапи в додатку

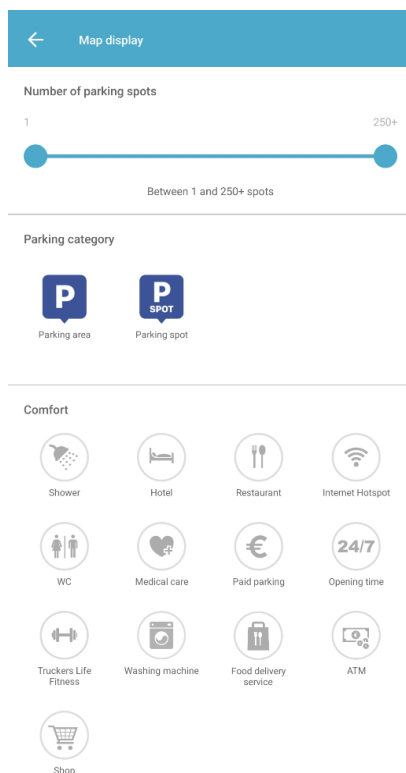


Рисунок 4.6 – Налаштування фільтра для відображення інфраструктурних об'єктів на мапі

Переглядаючи інфраструктурні об'єкти на мапі є можливість додати їх до списку улюблених, тобто збережених об'єктів, щоб потім легко знову віднайти їх на мапі або прокласти до них маршрут. Список інфраструктурних об'єктів, що були додані до улюблених, представлено на Рис. 4.7.

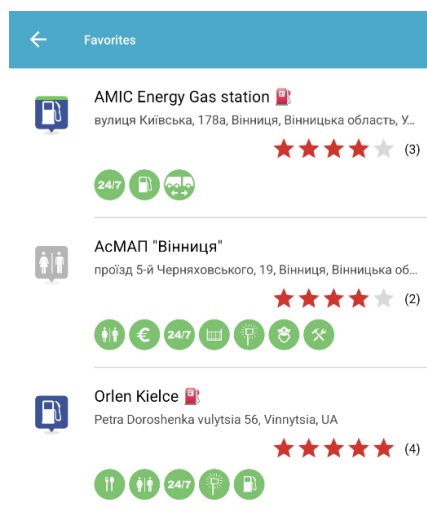


Рисунок 4.7 – Відображення інфраструктурних об'єктів, що були додані до улюблених

В додатку є можливість створення нового інфраструктурного об'єкту. Заповнення картки нового об'єкту відбувається в декілька етапів: додавання адреси та назви, додавання інформації про місця для паркування, додавання інформації про додаткові послуги на об'єкті, а також додавання рейтингу та відгука. Всі перелічені етапи представлено на рисунках 4.8 – 4.11.

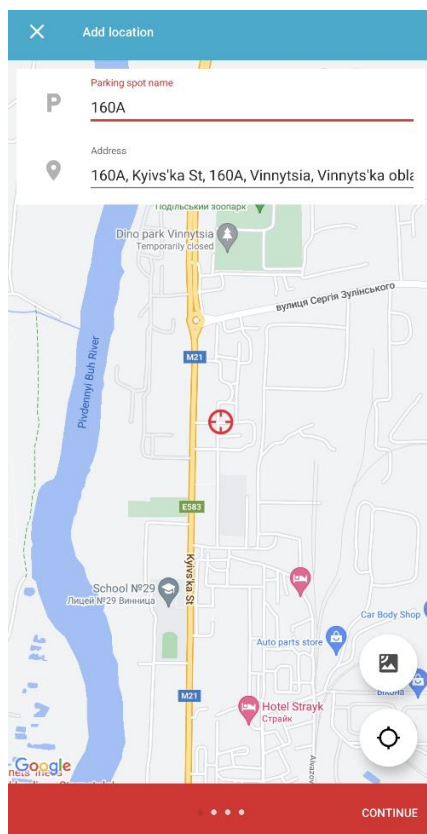


Рисунок 4.8 – Створення нового об'єкту на мапі

Specify parking category

Type of parking

Parking area

Parking spot

Number of parking spots

30

BACK CONTINUE

Рисунок 4.9 – Додавання інформації про місця для паркування

Specify features

Comfort

Shower

Hotel

Restaurant

Internet Hotspot

WC

Medical care

Paid parking

Opening time

Truckers Life Fitness

Washing machine

Food delivery service

ATM

Shop

BACK CONTINUE

Рисунок 4.10 – Додавання інформації про додаткові послуги на об'єкті

✕ Publish new parking

Summary

P 160A
160A, Kyivs'ka St, 160A, Vinnytsia, Vinnyts'ka oblast, Ukraine, 21000

Publish and rate

★ ★ ★ ★ ★

Comment

Comfortable new parking

BACK PUBLISH

Рисунок 4.11 – Додавання відгуку про об’єкт та розміщення нового об’єкту

Для початку маршруту необхідно вибрати кінцеву точку (місце призначення) на мапі, що зображено на Рис. 4.12. Також є можливість переглянути інфраструктурні об’єкти розміщені неподалік цього місця – рисунок 4.13.

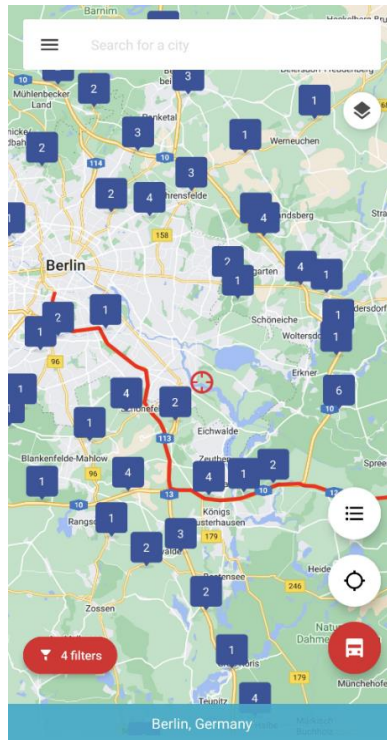


Рисунок 4.13 – Відображення маршруту руху до місця призначення

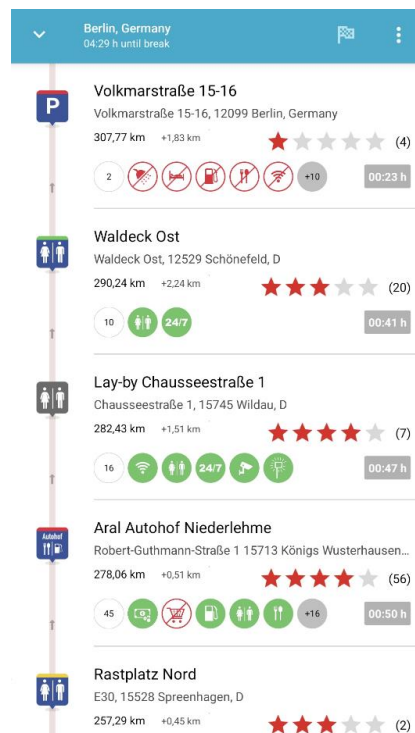


Рисунок 4.15 – Відображення інфраструктурних об’єктів на шляху до місця призначення

Користувач може в будь-який момент самостійно закінчити подорож, натиснувши на білий прапорець в правому верхньому куті. У відповідь на що буде запропоновано закінчити поточну подорож або розпочати нову, що представлено на Рис. 4.16. Кожна з зазначених опцій буде переривати поточну подорож, тому в мобільному додатку передбачено спливаюче вікно для підтвердження вибору закінчення подорожі, що представлено на Рис. 4.17.

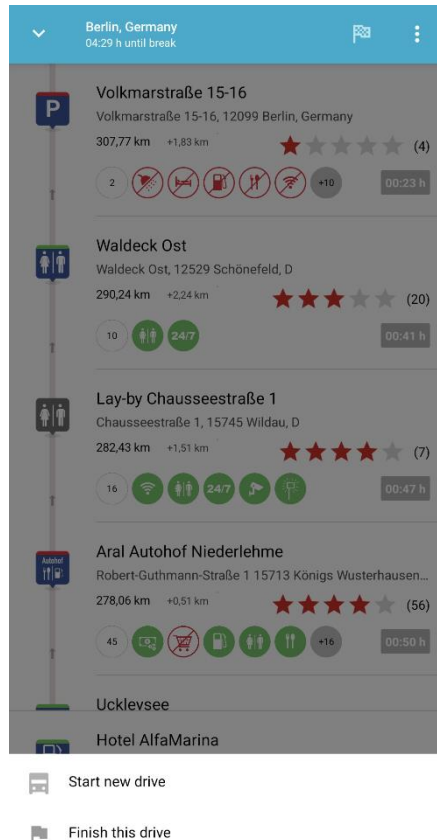


Рисунок 4.16 – Можливість закінчення поїздки або початку нової

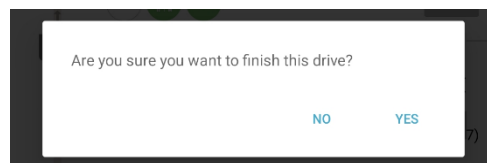


Рисунок 4.17 – Підтвердження наміру закінчити поїздку

Завершуючи аналіз представлених рисунків та описів функціональності мобільного додатку, можна констатувати, що всі встановлені вимоги до системи були ретельно розроблені, імплементовані та протестовані. Результати тестування підтверджують ефективність та зручність використання мобільного

додатку, а також його здатність адекватно реагувати на запити користувачів. Впровадження інноваційних рішень, таких як інтеграція з Google Maps, фільтрація інфраструктурних об'єктів, а також можливість персоналізації досвіду користувача, сприяють підвищенню загальної задоволеності користувачів та їхньої взаємодії з додатком. Таким чином, мобільний додаток демонструє свою надійність та готовність до впровадження та використання в реальних умовах, забезпечуючи користувачам зручний інструмент для планування маршрутів із врахуванням актуальної інформації про інфраструктурні об'єкти.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було проведено глибоке дослідження та аналіз систем планування маршрутів для мобільних застосунків з метою оптимізації процесів планування та підвищення ефективності використання транспортної інфраструктури. Метою роботи було створення інструменту, спрямованого на покращення мобільності користувачів та забезпечення більш точного та ефективного планування маршрутів.

В першому розділі було проведено дослідження об'єкта автоматизації, що дало змогу визначити ключові аспекти та потреби в автоматизації планування маршрутів, а також аналіз інфраструктурних об'єктів та їх впливу на планування маршрутів, а також досліджено загальні підходи та методології в системах планування маршрутів. Особлива увага була приділена використанню штучного інтелекту та машинного навчання для оптимізації маршрутів.

Другий розділ присвячений дослідженню функціональних можливостей сучасних мобільних застосунків для планування маршрутів, включаючи аналіз взаємодії користувачів з цими застосунками.

Третій та четвертий розділи зосереджені на проектуванні та розробці модуля користувача мобільного застосунку для автоматизованої системи планування маршрутів. Було розроблено UML-діаграми для модуля користувача, структуру бази даних, а також вибрано технології для розробки та налаштування роботи модуля.

Здійснено тестування розробленої системи, результати якого свідчать про її ефективність та можливість використання в реальних умовах для поліпшення процесів планування маршрутів. Робота виконана в повному обсязі, відповідає поставленим завданням та меті, і може служити важливим внеском у сферу автоматизації планування маршрутів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ковальчук В.А., Литвин О.П. Автоматизація процесів планування маршрутів в умовах урбанізації: теорія та практика. Київ: Наукова думка, 2008. 234 с.
2. Дмитриченко М.Ф. Основи теорії транспортних процесів і систем: навч. посібн. для ВНЗ. Київ: Видавничий Дім «Слово», 2009. 336 с.
3. Мельник О.Б., Сердюк О.С. Методи оптимізації маршрутів в умовах міської інфраструктури. Харків: Фактор, 2011. 256 с.
4. Бондаренко М.І., Кравченко В.П. Розробка мобільних додатків з використанням SQLite. Київ: Техніка, 2009. 142 с.
5. Грищенко В.Ю., Савченко В.К. Використання хмарних технологій для зберігання даних: Google Cloud SQL. Одеса: Астропринт, 2012. 204 с.
6. Шевченко Г.І., Кириленко О.В. Класифікація інфраструктурних об'єктів та їх вплив на міське середовище. Житомир: Полісся, 2014. 168 с.
7. Мороз П.Ю., Ткачук Р.В. Транспортна інфраструктура міст: управління та оптимізація. Вінниця: УНІВЕРСУМ-Вінниця, 2016. 192 с.
8. Urban Infrastructure: Dynamics and Challenges. Cambridge: Cambridge University Press, 2015. 310 p.
9. Білоконь П.Д., Левченко І.К. Вплив інфраструктурних об'єктів на мобільність у міському середовищі. Дніпро: АртПрес, 2013. 210 с.
10. Козлов М.С., Федоров О.О. Розробка алгоритмів для автоматизованих систем планування маршрутів. Херсон: Олді-плюс, 2017. 174 с.
11. Гончарук В.В., Сидоренко Н.Г. Адаптивні системи планування маршрутів для комунальних та екстрених служб. Чернівці: Рута, 2015. 198 с.
12. Johnson, M. R., & Thompson, S. A. "Evolution of Route Planning Algorithms: From Dijkstra to AI-based Predictions." *Journal of Transportation Technologies*, vol. 10, no. 3, 2020, pp. 207-225.
13. Smith, J. A., & Lee, D. H. "Comprehensive Review of Routing Algorithms: From Classic Approaches to AI-Enhanced Techniques." *International Journal of Computer Science and Network Security*, vol. 21, no. 4, 2021, pp. 345-362.

14. Patel, R. N., & Kumar, S. "Efficiency of A* and Dijkstra's Algorithm in Modern Route Planning Systems." *Journal of Advanced Transportation Studies*, vol. 29, no. 2, 2022, pp. 154-169.
15. Thompson, M., & Richardson, C. "Handling Negative Weights in Graph Algorithms: A Bellman-Ford Approach." *Journal of Graph Theory and Applications*, vol. 18, no. 1, 2020, pp. 88-104.
16. Ковальчук Т.О., Петренко А.С. "Адаптивні системи планування маршрутів в умовах міського середовища." *Науковий журнал транспортних технологій*, том 5, № 3, 2021, с. 234-249.
17. Мельник І.В., Кравченко В.О. "Мульти-модальні та інтермодальні системи у міському плануванні маршрутів." *Журнал міської мобільності*, том 12, № 4, 2022, с. 112-130.
18. Бондаренко О.М., Шевченко Т.І. "Інтеграція мульти-модальних підходів у сучасних системах планування маршрутів." *Науковий вісник транспорту*, том 7, № 2, 2023, с. 89-104.
19. Johnson, M. R., & Gupta, A. K. "Artificial Intelligence and Machine Learning in Route Planning: A Comprehensive Analysis." *Journal of Transportation and Logistics*, vol. 33, no. 1, 2022, pp. 45-63.
20. Wallace, S., & Zhang, L. "Integrating AI and Machine Learning for Advanced Traffic Management and Route Optimization." *Journal of Smart Mobility*, vol. 10, no. 3, 2021, pp. 204-220.
21. Левченко О.В., Сидоренко В.П. "Інтеграція мобільних застосунків з міськими інфраструктурними базами даних." *Журнал міської інфраструктури та технологій*, том 8, № 1, 2022, с. 77-92.
22. Петровська І.В., Козлов М.О. "Аналіз вимог користувачів до систем навігації та маршрутизації." *Журнал сучасних транспортних систем*, том 14, № 2, 2022, с. 158-174.
23. Литвиненко Ю.А., Сергієнко В.П. "Оптимізація автоматизованих систем планування маршрутів на основі користувацьких відгуків." *Вісник транспортних технологій*, том 9, № 1, 2023, с. 45-59.

24. Johnson, M., & Harper, T. "Enhancing Route Planning Efficiency through Machine Learning Algorithms." *Journal of Advanced Transportation Studies*, vol. 15, no. 2, 2023, pp. 134-150.
25. Google Maps [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en>. – Дата доступу: 20.09.2024.
26. Waze [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.waze>. – Дата доступу: 01.10.2024.
27. HERE WeGo [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.here.app.maps>. – Дата доступу: 01.10.2024.
28. Citymapper [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.citymapper.app.release>. – Дата доступу: 03.10.2024.
29. Мельник І.О., Кравченко В.П. "Вплив інтерфейсу користувача на використання мобільних застосунків для планування маршрутів." *Журнал цифрових технологій*, том 12, № 4, 2022, с. 234-249.
30. Собчук О.М., Грищенко К.Ю. "Аналіз функціональності мобільних застосунків для планування маршрутів." *Журнал транспортних інновацій*, том 15, № 3, 2023, с. 112-130.
31. Ковальчук Т.А., Шевченко П.О. "Методики проектування модулів користувача для мобільних застосунків." *Журнал інформаційних технологій та систем*, том 17, № 4, 2023, с. 245-263.
32. Жуковський Є.О., Петренко М.І. "Принципи модульної архітектури в розробці мобільних застосунків." *Журнал програмних систем*, том 20, № 3, 2023, с. 134-145.
33. Сидорчук Л.П., Горбачова О.В. "Використання діаграм варіантів використання для аналізу користувацьких потреб у мобільних застосунках." *Журнал комп'ютерних наук*, том 18, № 2, 2023, с. 88-102.

- 34.Коваленко Д.В., Литвиненко Ю.А. "Роль діаграми класів в об'єктно-орієнтованому проектуванні." Журнал програмування та комп'ютерних технологій, том 12, № 4, 2023, с. 210-225.
- 35.Марченко Р.О., Ткаченко В.П. "Аналіз взаємодій в системах за допомогою діаграми послідовностей." Журнал інформаційних систем, том 15, № 1, 2023, с. 53-67.
- 36.Smith, J., & Brown, A. "Database Design Lifecycle: A Comprehensive Guide." Journal of Database Management, vol. 29, no. 3, 2022, pp. 45-62.
- 37.Johnson, L., & Davis, R. "From Conceptual to Physical: Steps in Database Design." International Journal of Information Technology, vol. 18, no. 4, 2023, pp. 234-250.
- 38.Thompson, M., & Lee, K. "Improving Database Performance through Iterative Refinement." Database Systems Journal, vol. 21, no. 1, 2023, pp. 88-104.
- 39.Васильєв А.С. Розробка мобільних застосунків на платформі Android. Комп'ютерні науки та інформаційні технології. 2019. № 3. С. 34-39.
- 40.Walters, T., & Nguyen, D. "Entity-Relationship Diagrams: A Tool for Database Design in Mobile App Development." Journal of Software Engineering and Applications, vol. 19, no. 3, 2023, pp. 202-219.
- 41.Carter, E., & Singh, M. "Adaptive Database Systems: Responding to Dynamic Business Requirements." Journal of Data Management, vol. 22, no. 1, 2023, pp. 47-63.
- 42.Anderson, R., & Lee, H. "Kotlin and Java: Effective Languages for Android App Development." Journal of Mobile Technology, vol. 25, no. 2, 2023, pp. 134-150.
- 43.Thompson, G., & Patel, S. "Securing Mobile Applications: Encryption and Security Protocols." Cybersecurity Journal, vol. 17, no. 4, 2023, pp. 198-215.
- 44.Morris, L., & Jackson, T. "Implementing Material Design in Mobile Apps." UI/UX Design Magazine, vol. 12, no. 3, 2023, pp. 72-89.
- 45.Green, D., & Kumar, A. "Material Design: Enhancing User Experience in Mobile Applications." Journal of Design and Technology, vol. 19, no. 1, 2023, pp. 55-70.
- 46.White, C., & Zhao, F. "Ensuring Data Integrity with Transactions in Mobile Applications." Database Systems Journal, vol. 22, no. 2, 2023, pp. 104-119.

47. Brown, M., & Lee, J. "Transaction Management in SQLite for Robust Mobile Applications." *Mobile Database Review*, vol. 20, no. 4, 2023, pp. 142-158.
48. Harris, T., & Kumar, S. "Implementing Dijkstra's Algorithm for Route Optimization in Mobile Apps." *Journal of Mobile Computing and Applications*, vol. 26, no. 1, 2023, pp. 33-47.
49. Patel, R., & Thompson, G. "Efficient Pathfinding with Priority Queues in Java." *Advanced Computing Techniques Journal*, vol. 18, no. 3, 2023, pp. 115-130.
50. Іванов, О. "Застосування AES для захисту даних у мобільних застосунках." *Український журнал кібербезпеки*, том 5, № 2, 2023, с. 45-59.
51. Martin, C., & Zhou, Y. "Advanced Encryption Standard (AES): A Review of Its Applications in Mobile Security." *International Journal of Mobile Security*, vol. 31, no. 1, 2023, pp. 76-92.
52. Петренко, М. "Тестування та оптимізація мобільних застосунків: важливість і методи." *Журнал програмної інженерії*, том 14, № 3, 2023, с. 112-128.
53. Сидоренко, В. "Використання системи контролю версій Git у розробці програмного забезпечення." *Журнал інформаційних технологій*, том 17, № 4, 2023, с. 234-249.

Додатки

Додаток Б
(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО
Зав. кафедри КСУ ВНТУ,
д.т.н., професор
_____ В'ячеслав КОВТУН
“ 14 ” 10 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи
Автоматизація побудови маршруту руху з урахуванням
інфраструктурних об'єктів

08-33.МКР.8.03.000 ТЗ

Студент групи

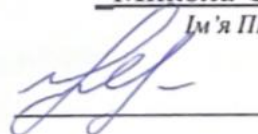


Микола СКРИПНИК

Підпис

Ім'я ПРІЗВИЩЕ

Керівник



Марія ІОХИМЧУК

Підпис

Ім'я ПРІЗВИЩЕ

Вінниця 2024

1. Назва та галузь застосування

1.1. Назва – Автоматизація побудови маршруту руху з урахуванням інфраструктурних об'єктів.

1.2. Галузь застосування – Інформаційні технології у сфері транспорту та логістики, розробка систем для оптимізації маршрутів.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ № 310 від 17-09-2024 р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є розробка та впровадження мобільного додатку для автоматизованої системи планування маршрутів з урахуванням інфраструктурних об'єктів, спрямованої на оптимізацію маршрутів, зниження часу в дорозі та підвищення загальної ефективності переміщень. Робота має на меті створення інструменту, який дозволить користувачам ефективно планувати свої поїздки, інтегруючи різноманітні інфраструктурні дані та забезпечуючи високий рівень адаптації до актуальних дорожніх умов.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Дмитриченко М.Ф. Основи теорії транспортних процесів і систем: навч. посібн. для ВНЗ. Київ: Видавничий Дім «Слово», 2009. 336 с.
2. Прокудін Г.С. Моделі і методи оптимізації перевезень у транспортних системах: навч. посібн. Київ: НТУ, 2006. 224 с.
3. Васильєв А.С. Розробка мобільних застосунків на платформі Android. Комп'ютерні науки та інформаційні технології. 2019. № 3. С. 34-39.

5. Вимоги до розробки.

5.1. Перелік головних етапів виконання розробки:

- визначити аспекти функціонування мобільного застосунку для планування маршрутів;
- дослідити функціональні можливості існуючих мобільних застосунків для планування маршрутів;
- скласти вимоги до автоматизованої системи, що розробляється;
- спроектувати та розробити модуль користувача у відповідності до поставлених вимог та функцій:
 - реєстрація та авторизація користувачів;
 - управління персональними профілями користувачів;
 - планування та управління маршрутами;
 - інтеграція з картографічними сервісами для відображення маршрутів.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- підтримка мобільної платформи Android;
 - використання мов програмування Kotlin та Java для Android.
- 5.2.2. Умови експлуатації системи:
- робота системи у режимі 24/7 з високою доступністю та надійністю;
 - система повинна бути масштабованою для обробки великих обсягів даних та здатною адаптуватися до зростаючих потреб користувачів;
 - забезпечення безпеки даних користувачів та їх конфіденційності.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

- Дослідження об'єкта автоматизації та постановка задач дослідження 03.09.2024р.
- Дослідження та аналіз інфраструктурних об'єктів для планування маршрутів 19.09.2024р.
- Дослідження функціональних можливостей сучасних мобільних застосунків для планування маршрутів 01.10.2024р.
- Проектування модуля користувача мобільного застосунку для автоматизованої системи планування маршрутів 15.11.2024р.
- Апробація результатів дослідження 20.11.2024р.
- Публікації
- Оформлення пояснювальної записки, графічного матеріалу і презентації 29.11.2024р.

6.2 Графічні матеріали:

- UML-діаграми 04.10. 2024 р.
- ER-діаграма структури бази даних 18.10. 2024 р.
- вигляд екранів розробленого мобільного додатку 18.11. 2024 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «20»__11__ 2024 р.
- 7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «1»__12__ 2024 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «7»__12__ 2024 р.

Додаток В

Лістинг програмного забезпечення

Мережевий запит для отримання даних про маршрути:

```
import android.os.AsyncTask;
import android.util.Log;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class FetchRoutesTask extends
AsyncTask<Void, Void, String> {
    private static final String TAG =
"FetchRoutesTask";

    private static final String ROUTES_URL = "
https://maps.googleapis.com/maps/api/directions/json?origin=
Toronto&destination=Montreal";

    @Override
    protected String doInBackground(Void... voids) {
        HttpURLConnection urlConnection = null;
        BufferedReader reader = null;
        String routesJsonStr = null;

        try {
            URL url = new URL(ROUTES_URL);
            urlConnection = (HttpURLConnection)
url.openConnection();
            urlConnection.setRequestMethod("GET");
            urlConnection.connect();

            InputStreamReader inputStream = new
InputStreamReader(urlConnection.getInputStream());
            reader = new BufferedReader(inputStream);
            StringBuilder builder = new StringBuilder();

            String line;
            while ((line = reader.readLine()) != null) {
                builder.append(line).append("\n");
            }

            if (builder.length() == 0) {
                return null;
            }
        }
```

```
        routesJsonStr = builder.toString();
    } catch (Exception e) {
        Log.e(TAG, "Error ", e);
        return null;
    } finally {
        if (urlConnection != null) {
            urlConnection.disconnect();
        }
        if (reader != null) {
            try {
                reader.close();
            } catch (final IOException e) {
                Log.e(TAG, "Error closing stream", e);
            }
        }
    }
    return routesJsonStr;
}
```

```
@Override
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    if (s != null) {
        try {
            parseRoutesData(s);
        } catch (Exception e) {
            Log.e(TAG, "Error parsing JSON", e);
        }
    }
}
```

```
private void parseRoutesData(String jsonStr)
throws Exception {
    JSONArray routesArray = new
JSONArray(jsonStr);
    for (int i = 0; i < routesArray.length(); i++) {
        JSONObject routeObject =
routesArray.getJSONObject(i);
        String startPoint =
routeObject.getString("startPoint");
        String endPoint =
routeObject.getString("endPoint");
        double distance =
routeObject.getDouble("distance");
        double estimatedTime =
```

```

routeObject.getDouble("estimatedTime");

        // Збереження даних у базу даних
        saveRouteToDatabase(startPoint, endPoint,
distance, estimatedTime);
    }
}

private void saveRouteToDatabase(String
startPoint, String endPoint, double distance, double
estimatedTime) {
    SQLiteDatabase database = new
DatabaseHelper(context).getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("StartPoint", startPoint);
    values.put("EndPoint", endPoint);
    values.put("Distance", distance);
    values.put("EstimatedTime", estimatedTime);

    long newRowId = database.insert("Routes",
null, values);
    Log.d(TAG, "New route inserted: " +
newRowId);
}
}

```

Логіка обробки даних - Алгоритм Дейкстри:

```

import java.util.Arrays;
import java.util.PriorityQueue;

public class DijkstraAlgorithm {
    public static int[] dijkstra(int[][] graph, int start) {
        int[] distances = new int[graph.length];
        boolean[] visited = new boolean[graph.length];
        PriorityQueue<Integer> queue = new
PriorityQueue<>((a, b) -> Integer.compare(distances[a],
distances[b]));

        Arrays.fill(distances, Integer.MAX_VALUE);
        distances[start] = 0;
        queue.add(start);

        while (!queue.isEmpty()) {
            int current = queue.poll();
            if (visited[current]) continue;
            visited[current] = true;

            for (int i = 0; i < graph[current].length; i++) {
                if (graph[current][i] != 0 &&

```

```

distances[current] + graph[current][i] < distances[i]) {
                    distances[i] = distances[current] +
graph[current][i];
                    queue.add(i);
                }
            }
        }
        return distances;
    }
}

```

Управління базою даних - Створення таблиці маршрутів:

```

import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.content.Context;

public class DatabaseHelper extends
SQLiteOpenHelper {
    private static final String DATABASE_NAME =
"RoutePlanner";
    private static final int DATABASE_VERSION =
1;

    private static final String TABLE_ROUTES =
"CREATE TABLE Routes (" +
        "RouteID INTEGER PRIMARY KEY
AUTOINCREMENT," +
        "StartPoint TEXT NOT NULL," +
        "EndPoint TEXT NOT NULL," +
        "Distance REAL NOT NULL," +
        "EstimatedTime REAL NOT NULL" +
        ");";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null,
DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(TABLE_ROUTES);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS
Routes");
    }
}

```

```

        onCreate(db);
    }
}

Інтерфейс користувача (UI) - Компонент карти:
import com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.OnMapReadyCallback;
import
com.google.android.gms.maps.SupportMapFragment;
import
com.google.android.gms.maps.model.LatLng;
import
com.google.android.gms.maps.model.MarkerOptions;

public class MapFragment extends Fragment
implements OnMapReadyCallback {
    private GoogleMap mMap;

    @Override
    public void onMapReady(GoogleMap
googleMap) {
        mMap = googleMap;
        LatLng kyiv = new LatLng(50.4501, 30.5234);
        mMap.addMarker(new
MarkerOptions().position(kyiv).title("Marker in Kyiv"));

        mMap.moveCamera(CameraUpdateFactory.newLatLng(kyiv)
);
    }

    @Override
    public View onCreateView(LayoutInflater
inflater, ViewGroup container,
Bundle savedInstanceState) {
        View rootView =
inflater.inflate(R.layout.fragment_map, container, false);
        SupportMapFragment mapFragment =
(SupportMapFragment) getChildFragmentManager()
.findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        return rootView;
    }
}

```

Активність для відображення маршрутів на карті:

```
import android.os.Bundle;
```

```

import androidx.fragment.app.FragmentActivity;
import
com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.OnMapReadyCallback;
import
com.google.android.gms.maps.SupportMapFragment;
import
com.google.android.gms.maps.model.LatLng;
import
com.google.android.gms.maps.model.MarkerOptions;
import
com.google.android.gms.maps.model.PolylineOptions;

public class RouteMapActivity extends
FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    private LatLng startPoint = new LatLng(50.4501,
30.5234); // Київ
    private LatLng endPoint = new LatLng(49.8397,
24.0297); // Львів

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_route_map);
        SupportMapFragment mapFragment =
(SupportMapFragment) getSupportFragmentManager()
.findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap
googleMap) {
        mMap = googleMap;
        mMap.addMarker(new
MarkerOptions().position(startPoint).title("Start: Kyiv"));
        mMap.addMarker(new
MarkerOptions().position(endPoint).title("End: Lviv"));

        // Додавання лінії маршруту
        mMap.addPolyline(new
PolylineOptions().add(startPoint,
endPoint).width(5).color(R.color.routeColor));
    }
}

```

```
// Зміщення камери для відображення всього
маршруту

mMap.moveCamera(CameraUpdateFactory.newLatLngBound
s(
    new
    LatLngBounds.Builder().include(startPoint).include(endPoint)
    .build(), 100));
    }
}
```

XML розмітка для активності з картою:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RouteMapActivity">

    <fragment
        android:id="@+id/map"

        android:name="com.google.android.gms.maps.SupportMapFr
        agment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</RelativeLayout>
```

Активність для вибору маршрутів:

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import
androidx.appcompat.app.AppCompatActivity;

    public class RouteSelectionActivity extends
    AppCompatActivity {

        private Button btnSelectRoute1;
        private Button btnSelectRoute2;
        private Button btnSaveFavorite;

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

```
setContentView(R.layout.activity_route_selection);
```

```
        btnSelectRoute1
        findViewById(R.id.btnSelectRoute1);

        btnSelectRoute2
        findViewById(R.id.btnSelectRoute2);

        btnSaveFavorite
        findViewById(R.id.btnSaveFavorite);

        btnSelectRoute1.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectRoute("Route 1");
            }
        });

        btnSelectRoute2.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectRoute("Route 2");
            }
        });

        btnSaveFavorite.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                saveFavoriteRoute();
            }
        });

        private void selectRoute(String routeName) {
            Toast.makeText(this, "Selected: " + routeName,
            Toast.LENGTH_SHORT).show();
            // Логіка для відображення деталей
            маршруту
        }

        private void saveFavoriteRoute() {
            Toast.makeText(this, "Route saved as
            favorite!", Toast.LENGTH_SHORT).show();
            // Логіка для збереження маршруту у базі
            даних або локальному сховищі
        }
    }
```

XML розмітка для активності вибору маршрутів:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/btnSelectRoute1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Route 1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp" />

    <Button
        android:id="@+id/btnSelectRoute2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Route 2"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/btnSelectRoute1"
        android:layout_marginTop="20dp" />

    <Button
        android:id="@+id/btnSaveFavorite"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save Favorite"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/btnSelectRoute2"
        android:layout_marginTop="20dp" />
</RelativeLayout>
```

Активність для детального перегляду маршруту:

```
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Button;
import android.content.Intent;
import
androidx.appcompat.app.AppCompatActivity;

public class RouteDetailActivity extends
AppCompatActivity {

    private TextView tvRouteDetails;
```

```
private Button btnShareRoute;
```

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

    setContentView(R.layout.activity_route_detail);

    tvRouteDetails
    findViewById(R.id.tvRouteDetails);

    btnShareRoute
    findViewById(R.id.btnShareRoute);

    String routeDetails
    getIntent().getStringExtra("ROUTE_DETAILS");
    tvRouteDetails.setText(routeDetails);

    btnShareRoute.setOnClickListener(v ->
shareRouteDetails(routeDetails));
}

private void shareRouteDetails(String
routeDetails) {
    Intent sendIntent = new Intent();
    sendIntent.setAction(Intent.ACTION_SEND);
    sendIntent.putExtra(Intent.EXTRA_TEXT,
"Check out this route: " + routeDetails);
    sendIntent.setType("text/plain");
    Intent shareIntent
    Intent.createChooser(sendIntent, null);
    startActivity(shareIntent);
}
}
```

XML розмітка для активності детального перегляду маршруту:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/tvRouteDetails"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Route Details"
        android:layout_centerHorizontal="true"
```

```

        android:layout_marginTop="100dp" />

<Button
    android:id="@+id/btnShareRoute"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Share Route"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/tvRouteDetails"
    android:layout_marginTop="20dp" />
</RelativeLayout>

```

Активність для відстеження прогресу подорожі:

```

import android.os.Bundle;
import android.widget.TextView;
import android.os.Handler;
import
androidx.appcompat.app.AppCompatActivity;

    public class TravelProgressActivity extends
AppCompatActivity {
    private TextView tvDistanceRemaining;
    private TextView tvEstimatedTime;
    private Handler handler = new Handler();
    private int distanceRemaining = 100;
    private int estimatedTime = 120;

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

setContentView(R.layout.activity_travel_progress);

        tvDistanceRemaining          =
findViewById(R.id.tvDistanceRemaining);
        tvEstimatedTime              =
findViewById(R.id.tvEstimatedTime);

        // Оновлення інформації кожні 5 секунд
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                updateTravelInfo();
                handler.postDelayed(this, 5000);
            }
        }, 5000);
    }

```

```

private void updateTravelInfo() {
    // Логіка для оновлення відстані та часу
    distanceRemaining -= 1; // Зменшення відстані
на 1 км кожні 5 секунд
    estimatedTime -= 1; // Зменшення часу на 1
хвилину кожні 5 секунд

    tvDistanceRemaining.setText("Distance
remaining: " + distanceRemaining + " km");
    tvEstimatedTime.setText("Estimated time: " +
estimatedTime + " minutes");
}

    @Override
    protected void onDestroy() {
        super.onDestroy();
        handler.removeCallbacksAndMessages(null);
    }
}

```

XML розмітка для активності відстеження прогресу подорожі:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/tvDistanceRemaining"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Distance remaining: 100 km"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp" />

    <TextView
        android:id="@+id/tvEstimatedTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Estimated time: 120 minutes"
        android:layout_centerHorizontal="true"

        android:layout_below="@+id/tvDistanceRemaining"
        android:layout_marginTop="20dp" />
</RelativeLayout>

```

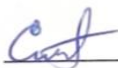
Додаток Г
(Обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
«Автоматизація побудови маршруту руху з урахуванням
інфраструктурних об'єктів»

1. UML-діаграма компонентів мобільного застосунку, що розробляється
2. UML-діаграма варіантів використання модуля користувача мобільного застосунку, що розробляється
3. UML-діаграма класів мобільного застосунку, що розробляється
4. UML-діаграма послідовностей мобільного застосунку, що розробляється
5. ER-діаграма бази даних модуля користувача мобільного додатку для автоматизованої системи планування маршрутів
6. Функціональні можливості мобільного додатку для неавторизованого користувача
7. Варіанти авторизації до мобільного додатку
8. Відображення даних користувача після авторизації
9. Функціональні можливості мобільного додатку для авторизованого користувача
10. Відображення Google мапи в додатку
11. Налаштування фільтра для відображення інфраструктурних об'єктів на мапі
12. Відображення інфраструктурних об'єктів, що були додані до улюблених
13. Створення нового об'єкту на мапі
14. Додавання інформації про місця для паркування
15. Додавання інформації про додаткові послуги на об'єкті
16. Додавання відгуку про об'єкт та розміщення нового об'єкту
17. Вибір кінцевої точки на мапі для прокладання маршруту
18. Перегляд інфраструктурних об'єктів поруч з місцем призначення
19. Відображення маршруту руху до місця призначення
20. Відображення інфраструктурних об'єктів на шляху до місця призначення
21. Можливість закінчення поїздки або початку нової
22. Підтвердження наміру закінчити поїздки

Виконав: студент 2-го курсу, групи ЗАКІТР-23м
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка

(шифр і назва спеціальності)



Микола СКРИПНИК

(ім'я та прізвище)

Керівник: д.т.н., доцент каф. КСУ



Марія ЮХИМЧУК

(ім'я та прізвище)

« 4 » 12 2024 р.

Рецензент: д.т.н., доцент каф.

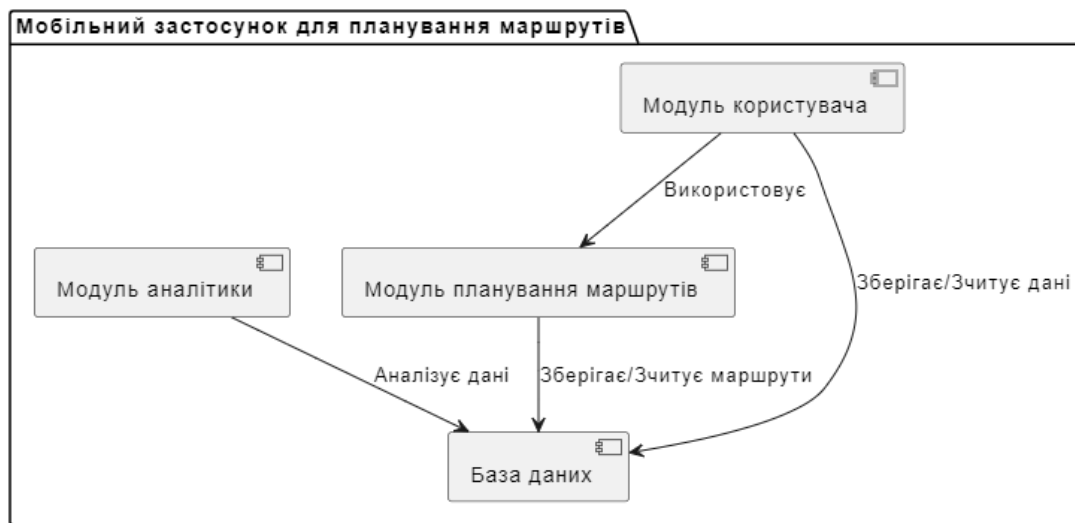


Володимир ГАРМАШ

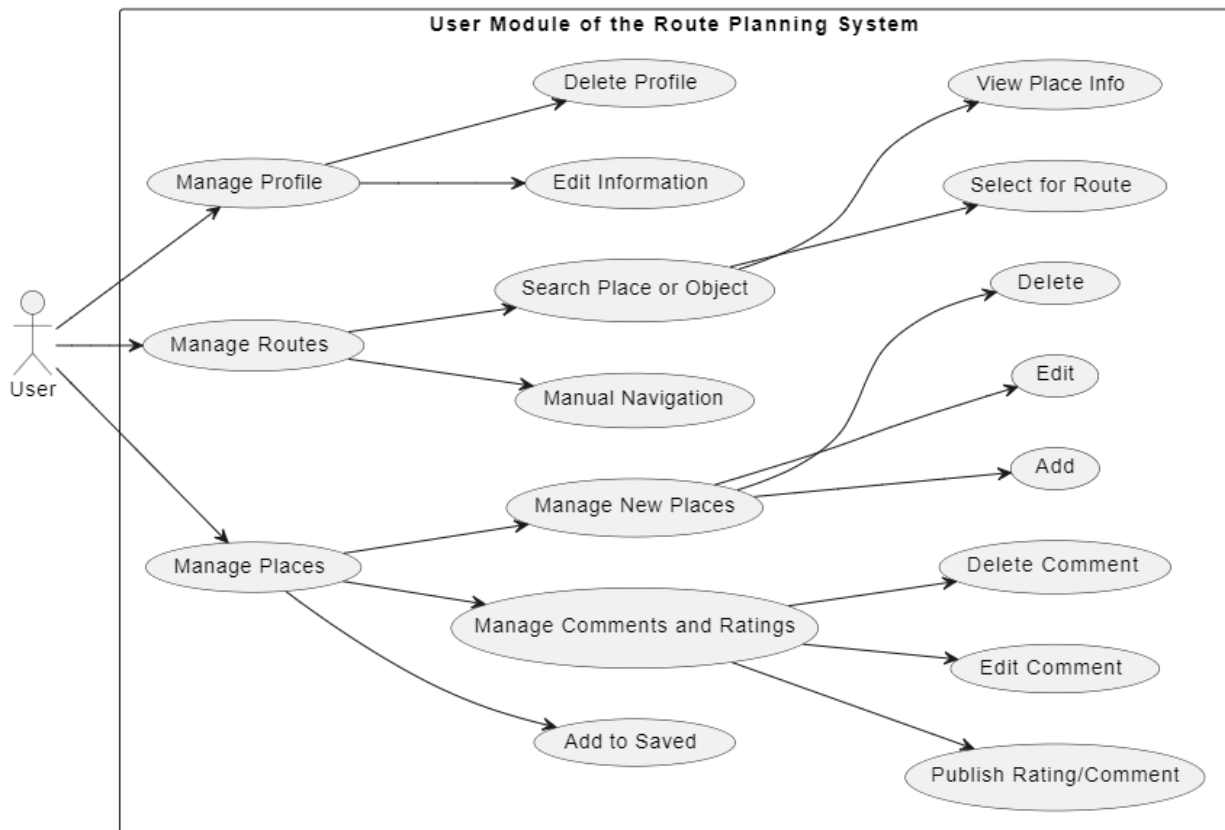
(ім'я та прізвище)

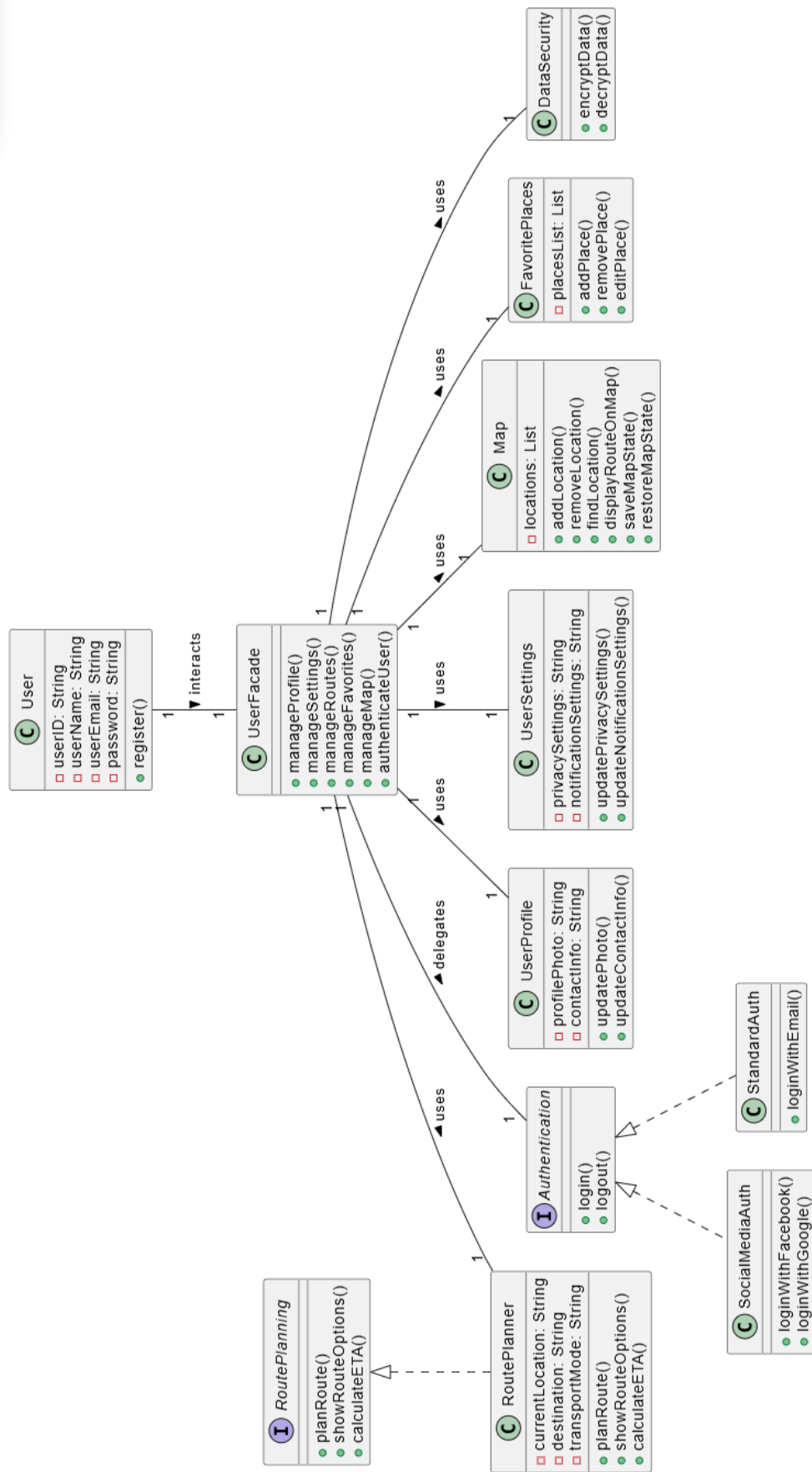
« 4 » 12 2024 р.

UML-діаграма компонентів мобільного застосунку, що розробляється

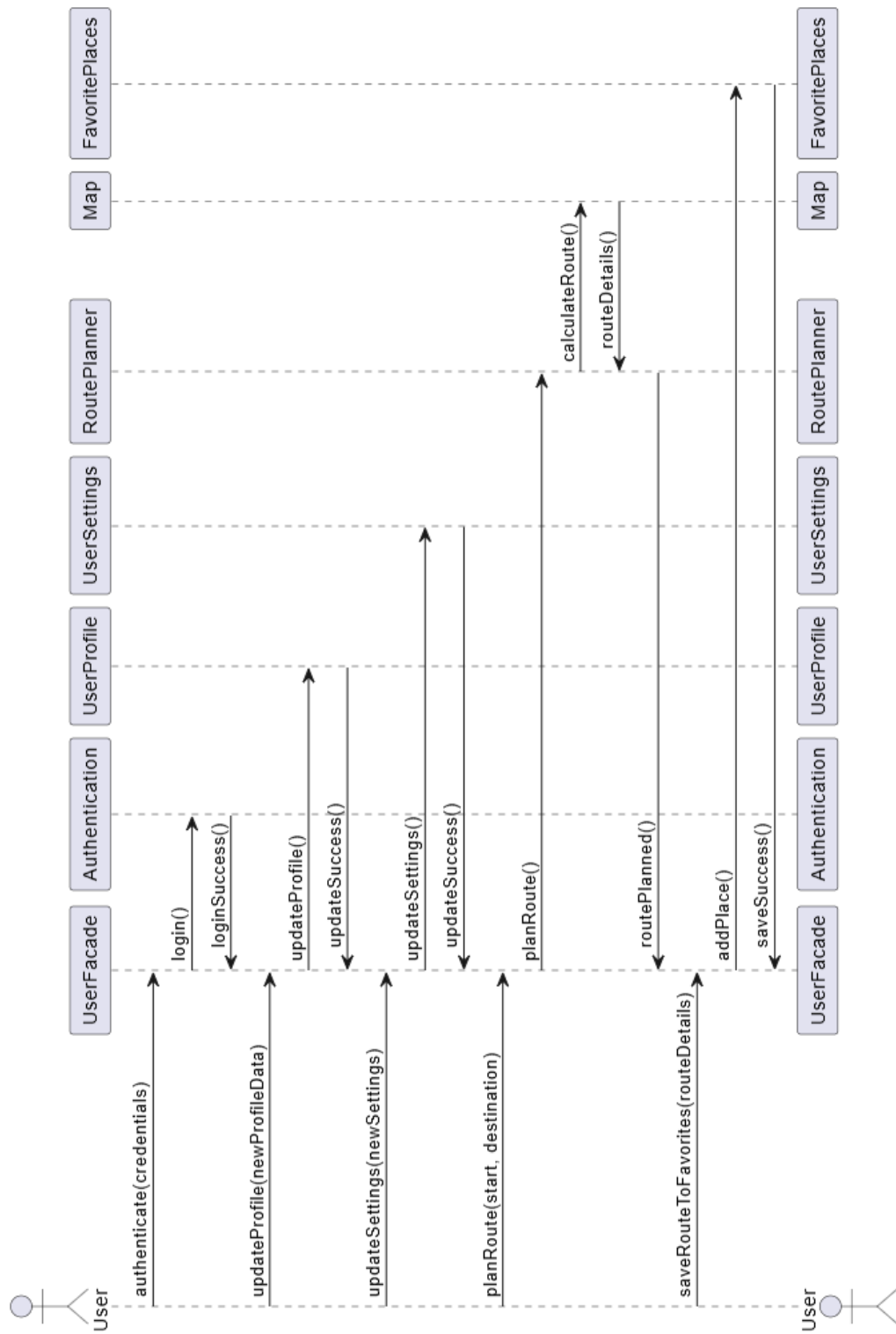


UML-діаграма варіантів використання модуля користувача мобільного застосунку, що розробляється



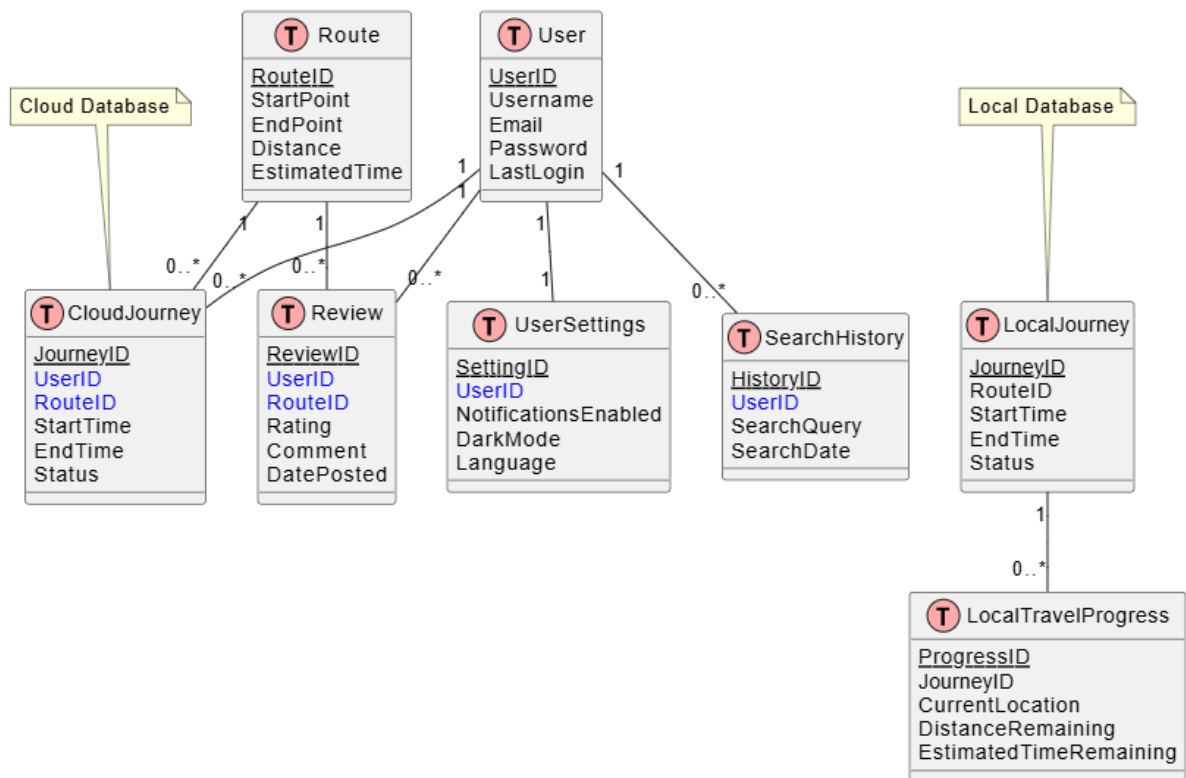


УМ
L-
діаг
рам
а
кла
сів
моб
іль
ног
о
зас
тос
унк
у,
що
роз
роб
ляє
тьс
я

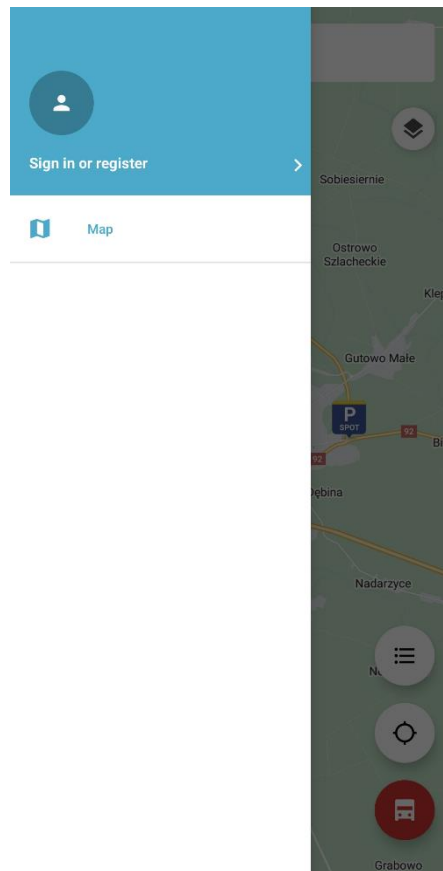


УМ
Л-
діаг
рам
а
ПОСЛ
ІДОВ
НОСТ
ей
мобі
ЛЬН
ОГО
заст
осу
нку,
що
розр
обля
єтьс
я

ER-діаграма бази даних модуля користувача мобільного додатку для
автоматизованої системи планування маршрутів



Функціональні можливості мобільного додатку для неавторизованого користувача



Варіанти авторизації до мобільного додатку



Sign in with



or

Email address

Password

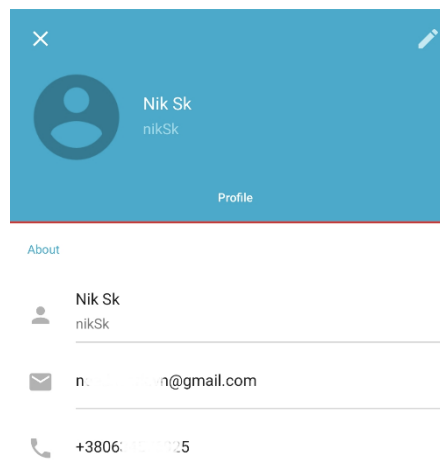


Log in

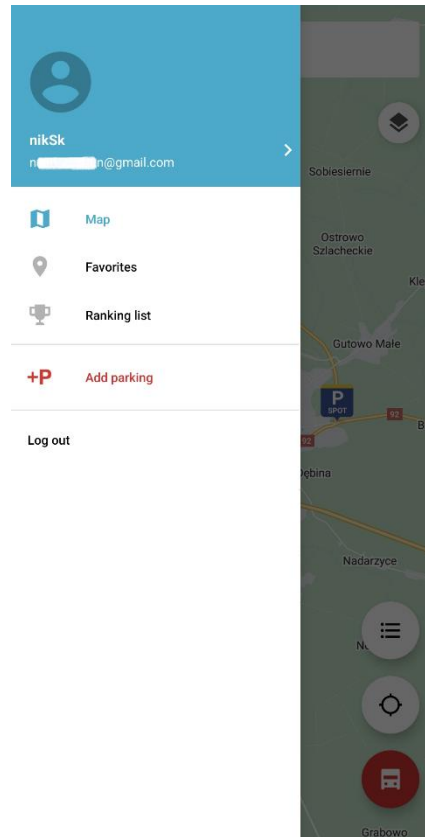
Don't have an account? [Create one](#)

[Forgot your password?](#)

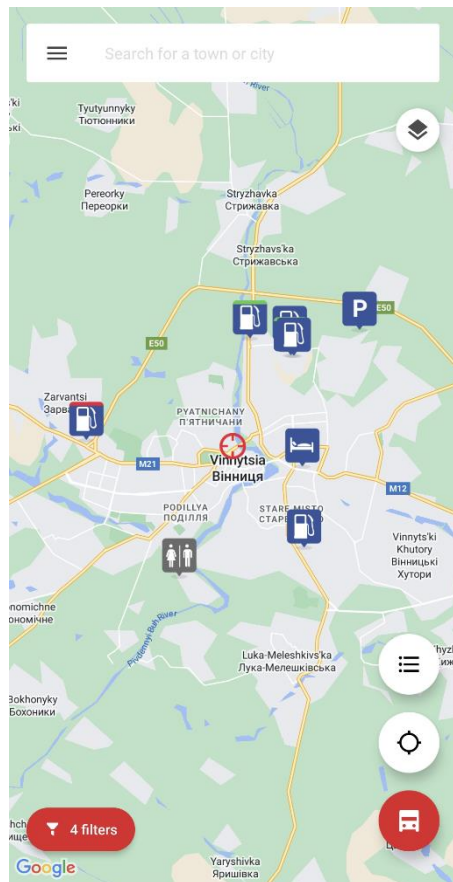
Відображення даних користувача після авторизації



Функціональні можливості мобільного додатку для авторизованого користувача



Відображення Google мапи в додатку



Налаштування фільтра для відображення інфраструктурних об'єктів на мапі

 Map display

Number of parking spots

1 250+



Between 1 and 250+ spots

Parking category


Parking area


Parking spot

Comfort


Shower


Hotel


Restaurant


Internet Hotspot


WC


Medical care


Paid parking


Opening time


Truckers Life
Fitness

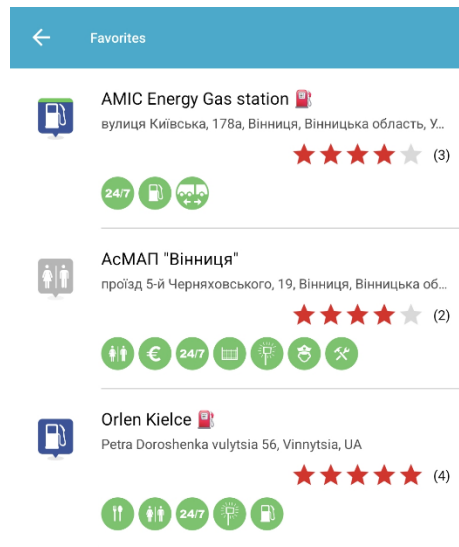

Washing machine


Food delivery
service

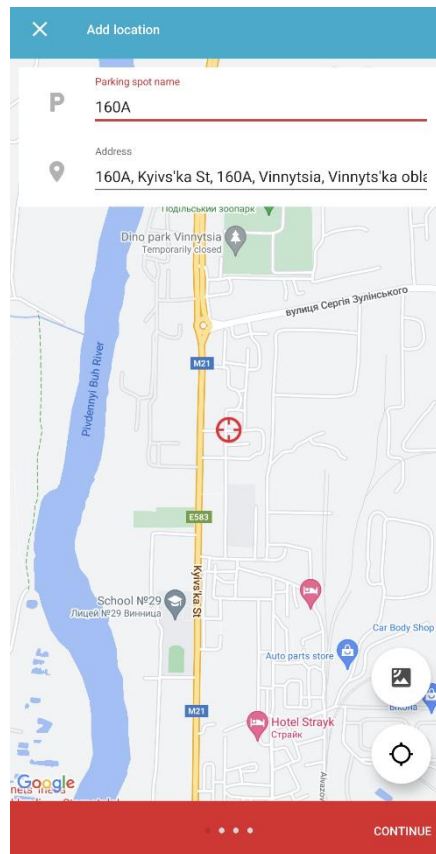

ATM


Shop

Відображення інфраструктурних об'єктів, що були додані до улюблених



Створення нового об'єкту на мапі



Додавання інформації про місця для паркування

×

Specify parking category

Type of parking

P

Parking area

P
SPOT

Parking spot

Number of parking spots

30

BACK

• • •

CONTINUE

Додавання інформації про додаткові послуги на об'єкті

× Specify features

Comfort


Shower


Hotel


Restaurant


Internet Hotspot


WC


Medical care


Paid parking


Opening time


Truckers Life
Fitness


Washing machine


Food delivery
service


ATM


Shop

BACK

• • • •

CONTINUE

Додавання відгуку про об'єкт та розміщення нового об'єкту

×

Publish new parking

Summary

P

160A

160A, Kyivs'ka St, 160A, Vinnytsia, Vinnyts'ka oblast, Ukraine, 21000

Publish and rate

★

★

★

★

★

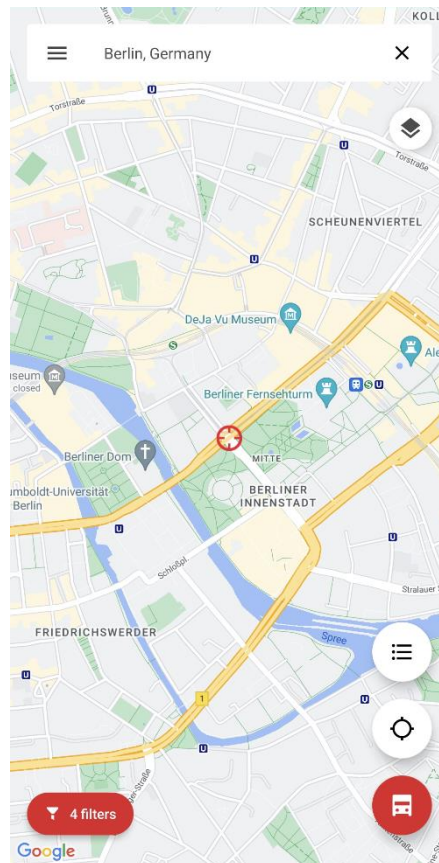
Comment

Comfortable new parking

BACK

PUBLISH

Вибір кінцевої точки на мапі для прокладання маршруту



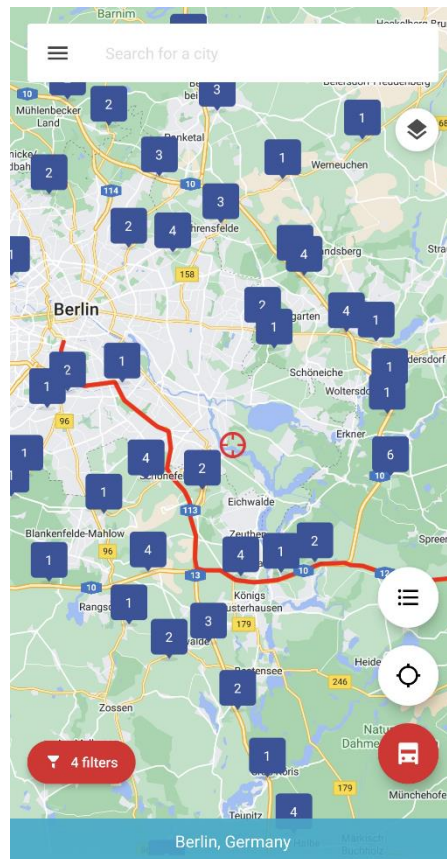
Перегляд інфраструктурних об'єктів поруч з місцем призначення

Nearby

Selma-Lagerlöf-Straße 7
Selma-Lagerlöf-Straße 7, 13189 Berlin, Germany
6,15 km N ★ ★ ★ ★ ★ (3)

Feldtmannstraße
Feldtmannstraße 5, 13088 Berlin, Германия
7,42 km NE ★ ★ ★ ★ ★ (1)

Відображення маршруту руху до місця призначення



Відображення інфраструктурних об'єктів на шляху до місця призначення

Berlin, Germany
 04:29 h until break

Volkmarstraße 15-16
 Volkmarstraße 15-16, 12099 Berlin, Germany
 307,77 km +1,83 km
 ★☆☆☆☆ (4)
 2 +10 00:23 h

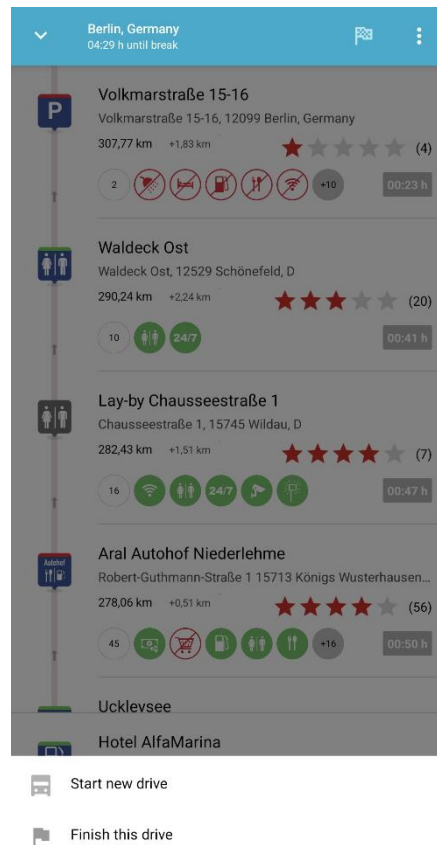
Waldeck Ost
 Waldeck Ost, 12529 Schönefeld, D
 290,24 km +2,24 km
 ★★★★★ (20)
 10 24/7 00:41 h

Lay-by Chausseestraße 1
 Chausseestraße 1, 15745 Wildau, D
 282,43 km +1,51 km
 ★★★★★ (7)
 16 24/7 00:47 h

Aral Autohof Niederlehme
 Robert-Guthmann-Straße 1 15713 Königs Wusterhausen...
 278,06 km +0,51 km
 ★★★★★ (56)
 45 +16 00:50 h

Rastplatz Nord
 E30, 15528 Spreenhagen, D
 257,29 km +0,45 km
 ★★★★★ (2)
 20 01:06 h

Можливість закінчення поїздки або початку нової



Підтвердження наміру закінчити поїздки

