

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Факультет інформаційних технологій
Кафедра комп'ютерних систем і мереж

Ралюсь Андрій Володимирович

УДК 004.42:004.421

БАКАЛАВРСЬКА РОБОТА

Розробка мобільної системи навігації та побудови маршрутів на гірськолижних курортах засобами Kotlin

Комп'ютерна інженерія

(назва освітньої програми)

123 - Комп'ютерна інженерія

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня Ралюсь А. В.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Кропивницька В. Б., доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри КСМ

д.т.н., професор С.І. Мельничук
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2026 рік

Івано-Франківський національний технічний університет нафти і газу

Факультет Інформаційних технологій

Кафедра Комп'ютерних систем і мереж

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:

Зав. кафедрою КСМ

С.І. Мельничук

«21» квітня 2026 року

**З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Равлюсю Андрію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка мобільної системи навігації та побудови маршрутів на гірськолижних курортах засобами Kotlin

керівник проекту (роботи) Кропивницька В. Б., доцент

затверджені наказом вищого навчального закладу від 21.04.2026 № 180/7

2. Строк подання студентом проекту (роботи) 10 червня 2026р.

3. Вихідні дані до роботи Методичні вказівки, технічна література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз систем навігації та побудови маршрутів на гірськолижних курортах. Проектування мобільно-орієнтованої системи навігації. Реалізація мобільно-орієнтованої системи побудови маршрутів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти розділів роботи

7. Дата видачі завдання 02 лютого 2026 р..

№ з/п	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	<i>Аналіз предметної області та існуючих рішень</i>	<i>Лютий 2026р</i>	
2	<i>Огляд існуючих рішень і методів</i>	<i>Березень 2026р</i>	
3	<i>Проектування системи навігації</i>	<i>Квітень 2026р</i>	
4	<i>Програмна реалізація та тестування системи</i>	<i>Травень 2026р</i>	
5	<i>Оформлення пояснювальної записки</i>	<i>Червень 2026р</i>	

Студент _____
(підпис)

Равлюсь А. В.
(прізвище та ініціали)

Керівник роботи _____

Кропивницька В. Б.

АНОТАЦІЯ

Робота присвячена розробці мобільно-орієнтованої системи навігації для гірськолижних курортів на прикладі «Буковелю». Традиційні картографічні сервіси не враховують специфіку гірськолижного середовища — односторонній рух на схилах, прив'язку до підйомників та рівні складності трас. Це робить їх неефективними для побудови спеціалізованих маршрутів і обумовлює актуальність створення окремого програмного рішення.

Для формалізації завдання інфраструктуру курорту представлено у вигляді орієнтованого зваженого графа. Вершинами графа є станції витягів та ключові точки розв'язок, а ребрами — траси та можливі переходи. Пошук оптимального шляху реалізовано на основі алгоритму Дейкстри. Вага ребер у моделі може динамічно змінюватися залежно від обраного критерію: метричної довжини схилу або орієнтовного часу його проходження.

Програмну логіку обробки графових структур даних і класи моделей створено мовою Kotlin. Клієнтську частину системи реалізовано як адаптивний веб-інтерфейс за допомогою HTML, CSS та JavaScript із використанням бібліотеки Cytoscape.js. Це дозволяє інтерактивно візуалізувати схему курорту, маркувати траси за кольорами складності та підсвічувати розрахований маршрут на мобільних пристроях без встановлення додаткових програм. Додатково до системи інтегровано розрахункові модулі для оцінки вартості квитків (ski-pass) та витрат пального.

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ, ОРІЄНТОВАНИЙ ЗВАЖЕНИЙ ГРАФ,
АЛГОРИТМ ДЕЙКСТРИ, KOTLIN, СТРУКТУРИ ДАНИХ, АДАПТИВНИЙ
ІНТЕРФЕЙС, МАРШРУТИЗАЦІЯ

ABSTRACT

The thesis focuses on the development of a mobile-oriented navigation system for ski resorts, using the "Bukovel" resort infrastructure as a case study. General-purpose mapping services do not account for the specific nature of the skiing environment, such as one-way slope traffic, lift dependencies, and trail difficulty levels. This makes them ineffective for specialized routing and highlights the necessity of developing a dedicated software solution.

To formalize the problem, the resort infrastructure was modeled as a weighted directed graph. Lift stations and key intersections act as vertices, while slopes and transitions act as edges. Optimal route discovery is implemented based on Dijkstra's algorithm. The edge weights can adjust dynamically depending on the selected optimization criteria: physical trail length or estimated travel time.

The computational logic for processing graph data structures and core data model classes are developed in Kotlin. The client-side application is engineered as a responsive web interface using HTML, CSS, and JavaScript, integrating the Cytoscape.js library. This setup enables interactive map rendering, trail difficulty color-coding, and dynamic path highlighting on mobile devices without requiring any native software installation. Additionally, computational modules are integrated to estimate travel fuel consumption and analyze ski-pass costs.

COMPUTER ENGINEERING, WEIGHTED DIRECTED GRAPH,
DIJKSTRA'S ALGORITHM, KOTLIN, DATA STRUCTURES, RESPONSIVE
INTERFACE, ROUTING

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ СИСТЕМ НАВІГАЦІЇ ТА ПОБУДОВИ МАРШРУТІВ НА ГІРСЬКОЛИЖНИХ КУОРТОХ	7
1.1 Особливості навігації на гірськолижних курортах	7
1.2 Загальна характеристика гірськолижного курорту “Буковель”	9
1.3 Аналіз існуючих картографічних і навігаційних сервісів	12
1.4 Аналіз методів та алгоритмів побудови маршрутів	15
1.5 Постановка задачі бакалаврської роботи.....	17
2 ПРОЄКТУВАННЯ МОБІЛЬНО-ОРІЄНТОВАНОЇ СИСТЕМИ НАВІГАЦІЇ	19
2.1 Формування графової моделі гірськолижного курорту	19
2.2 Опис інформаційної структури трас, підйомників і маршрутів	22
2.3 Обґрунтування вибору алгоритму пошуку найкоротшого маршруту	27
2.4 Розроблення структури програмної системи	30
2.5 Проєктування структури адаптивного користувацького інтерфейсу	35
2.6 Вибір програмних засобів реалізації системи	38
3 РЕАЛІЗАЦІЯ МОБІЛЬНО-ОРІЄНТОВАНОЇ СИСТЕМИ ПОБУДОВИ МАРШРУТІВ	42
3.1 Структура та функціональні можливості веб-інтерфейсу	42
3.2 Програмне подання графа та побудова маршруту	46
3.3 Опис роботи основних і додаткових модулів	51
ВИСНОВКИ	57
ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА	59
Додатки	62

					БР.КІ - 45.00.00.000 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Равлюсь А. В.			Розробка мобільної системи навігації та побудови маршрутів на гірськолижних курортах засобами Kotlin	Літ.	Арк.	Аркушів	
Перевір.		Кропивницька В.Б.					3	62	
Реценз.						ІФНТУНГ, КІ-22-2			
Н. Контр.		Лазорів А.М.							
Затверд.		Мельничук С.І.							

ВСТУП

Актуальність роботи. На сьогоднішній день інформаційні та мобільно-орієнтовані системи є важливим засобом підвищення зручності доступу користувачів до різноманітних сервісів. Особливого значення такі системи набувають у сфері туризму, де користувачу необхідно швидко отримувати інформацію про об'єкти інфраструктури, визначати своє місцезнаходження, обирати напрям руху та будувати оптимальний маршрут. Одним із прикладів такої предметної області є гірськолижні курорти, які мають розгалужену структуру трас, підйомників, пересадкових вузлів і сервісних об'єктів.

Гірськолижний курорт є складною просторовою системою, у межах якої турист повинен орієнтуватися між трасами різної складності, підйомниками та іншими елементами інфраструктури. Для досвідчених користувачів така навігація може бути звичною, однак для нових відвідувачів курорту вибір найзручнішого маршруту часто викликає труднощі. Особливо актуальною є задача визначення маршруту між окремими підйомниками або трасами з урахуванням довжини шляху, напрямку руху та особливостей гірськолижного пересування.

На практиці для орієнтування на гірськолижних курортах користувачі можуть застосовувати загальні картографічні сервіси, зокрема Google Maps, а також офіційні сайти курортів. Такі засоби надають загальну інформацію про розташування об'єктів, проте не завжди забезпечують зручну побудову маршруту саме між гірськолижними трасами та підйомниками. Крім того, звичайні картографічні сервіси переважно орієнтовані на автомобільні, пішохідні або громадські маршрути, тоді як гірськолижне середовище потребує врахування специфічної структури трас, напрямків спуску та розміщення підйомників.

У зв'язку з цим доцільною є розробка системи навігації та побудови маршрутів на гірськолижних курортах, яка може використовуватися на мобільних пристроях через адаптивний веб-інтерфейс. Такий підхід дозволяє забезпечити доступ до системи зі смартфона, планшета або персонального

комп'ютера без необхідності встановлення окремого програмного забезпечення. Основою роботи системи є подання інфраструктури курорту у вигляді графової моделі, де вершинами виступають підйомники, станції або ключові точки курорту, а ребрами — гірськолижні траси та можливі переходи між ними.

Для розв'язання задачі побудови маршруту доцільно використовувати алгоритми пошуку найкоротшого шляху у графі. Одним із найбільш поширених алгоритмів такого типу є алгоритм Дейкстри, який дозволяє знаходити найкоротший шлях між вершинами зваженого графа. У межах даної роботи цей алгоритм використовується для визначення маршруту між обраними користувачем об'єктами гірськолижного курорту. Реалізація програмної логіки системи розглядається з урахуванням можливості використання мови програмування Kotlin, яка є сучасним засобом розроблення мобільно-орієнтованих програмних рішень.

Об'єктом дослідження є процес навігації та побудови маршрутів у межах гірськолижного курорту.

Предметом дослідження є алгоритми, моделі та програмні засоби побудови маршрутів між об'єктами гірськолижної інфраструктури в мобільно-орієнтованій системі навігації.

Метою бакалаврської роботи є розробка мобільно-орієнтованої системи навігації та побудови маршрутів на гірськолижному курорті засобами Kotlin, яка забезпечує вибір початкової та кінцевої точки маршруту, визначення оптимального шляху між ними та відображення результату користувачу через адаптивний інтерфейс.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- проаналізувати особливості навігації на гірськолижних курортах;
- розглянути існуючі картографічні та навігаційні сервіси, які можуть використовуватися для орієнтування на гірськолижних курортах;
- дослідити методи та алгоритми побудови маршрутів у графових структурах;

- сформувати графову модель гірськолижного курорту на основі трас, підйомників та інших об'єктів інфраструктури;
- обґрунтувати вибір алгоритму пошуку найкоротшого маршруту;
- розробити структуру системи навігації та побудови маршрутів;
- описати реалізацію основних програмних компонентів системи;
- розробити адаптивний користувацький інтерфейс для відображення маршрутів;
- провести перевірку роботи системи на прикладі побудови маршруту між об'єктами гірськолижного курорту.

Методи дослідження. У роботі використано методи аналізу предметної області для визначення особливостей навігації на гірськолижних курортах; методи теорії графів для подання трас, підйомників і переходів у вигляді графової моделі; алгоритм Дейкстри для пошуку найкоротшого маршруту між заданими вершинами графа; методи об'єктно-орієнтованого програмування для опису структури програмної реалізації; методи проектування користувацького інтерфейсу для забезпечення зручної взаємодії користувача із системою.

Практичне значення отриманих результатів полягає в тому, що розроблена система може бути використана як основа для навігації на гірськолижних курортах. Вона дає змогу користувачам будувати маршрути між трасами та підйомниками, швидше орієнтуватися на території курорту та отримувати інформацію про можливий шлях пересування. Завдяки адаптивному веб-інтерфейсу система може використовуватися на мобільних пристроях через браузер.

1 АНАЛІЗ СИСТЕМ НАВІГАЦІЇ ТА ПОБУДОВИ МАРШРУТІВ НА ГІРСЬКОЛИЖНИХ КУРОРТАХ

1.1 Особливості навігації на гірськолижних курортах

Гірськолижні курорти належать до об'єктів туристичної інфраструктури зі складною просторовою організацією. Їх структура зазвичай включає велику кількість трас різної складності, підйомники, пересадкові точки, сервісні зони, пункти прокату спорядження, заклади харчування, медичні пункти та інші об'єкти, необхідні для комфортного перебування туристів. У зв'язку з цим важливим завданням є забезпечення зручної навігації територією курорту та можливості швидкого вибору маршруту між окремими об'єктами.

Особливістю навігації на гірськолижному курорті є те, що пересування користувача не є довільним, як у звичайному міському середовищі. Рух лижника або сноубордиста залежить від розміщення трас, напрямку спуску, роботи підйомників, складності ділянок, перепаду висот та доступності окремих маршрутів. Тому для побудови маршруту недостатньо враховувати лише географічну відстань між двома точками. Необхідно брати до уваги структуру трас, можливість переходу між ними, напрям руху та взаємозв'язок між підйомниками.

На відміну від автомобільної або пішохідної навігації, гірськолижна навігація має низку специфічних обмежень. По-перше, частина трас може мати односторонній характер руху, оскільки спуск здійснюється згори вниз. По-друге, різні траси мають різний рівень складності, що впливає на вибір маршруту користувачем залежно від його підготовки. По-третє, для переміщення між окремими ділянками курорту часто необхідно використовувати підйомники, які фактично є важливими вузлами маршруту. По-четверте, користувачу важливо отримати не лише найкоротший маршрут, а й зрозумілу послідовність дій: якою

трасою рухатися, до якого підйомника дістатися та в якому напрямку продовжити рух.

Для нових відвідувачів курорту складність навігації полягає також у великій кількості позначень, назв трас і номерів підйомників. Звичайна схема курорту не завжди дає змогу швидко визначити оптимальний шлях між двома точками, особливо якщо користувач погано орієнтується на місцевості. У таких умовах доцільним є використання програмної системи, яка автоматизує процес вибору маршруту та подає результат у зручному для користувача вигляді.

Одним із ефективних способів формалізації задачі навігації на гірськолижному курорті є подання його інфраструктури у вигляді графової моделі. У такій моделі вершинами графа можуть бути підйомники, станції, перехрестя трас або інші ключові точки курорту, а ребрами — гірськолижні траси чи переходи між ними. Кожному ребру може бути задана вага, наприклад довжина траси, орієнтовний час проходження або інший показник, що характеризує складність чи доцільність вибору відповідного шляху. Такий підхід дозволяє застосовувати алгоритми пошуку найкоротшого маршруту для визначення оптимального шляху між вибраними точками.

Важливим аспектом сучасних систем навігації є можливість їх використання на мобільних пристроях. Під час перебування на курорті користувач найчастіше має доступ саме до смартфона, тому інтерфейс системи повинен бути простим, зрозумілим і придатним для роботи на екранах різного розміру. Це може бути реалізовано за допомогою адаптивного веб-інтерфейсу, який відкривається у браузері смартфона або планшета, а також може використовуватися на персональному комп'ютері.

У правилах поведінки Міжнародної федерації лижного спорту та сноуборду FIS Code of Conduct for Skiers and Snowboarders [1] зазначено, що лижники та сноубордисти повинні дотримуватися встановлених знаків і маркування на трасах. Це підтверджує важливість коректного подання

навігаційної інформації, напрямків руху та позначень трас у системах навігації для гірськолижних курортів.

Під час проектування систем навігації для гірськолижних курортів доцільно враховувати не лише просторове розміщення трас і підйомників, а й наявні правила та рекомендації щодо безпеки руху на схилах. Зокрема, у правилах поведінки Міжнародної федерації лижного спорту та сноуборду зазначено, що лижники та сноубордисти повинні дотримуватися встановлених знаків і маркування на трасах. Це підтверджує важливість правильного відображення напрямків руху, позначень трас і навігаційної інформації в програмній системі. Крім того, на гірськолижних курортах широко використовується класифікація трас за рівнем складності, яка відображається на картах, інформаційних табло та вказівниках. Така класифікація дає користувачу змогу оцінити складність маршруту, а в системі навігації може використовуватися як додаткова характеристика під час опису трас і побудови шляху.

Отже, навігація на гірськолижних курортах потребує врахування специфіки трас, підйомників, напрямків руху та рівнів складності маршрутів. Використання графової моделі та алгоритмів пошуку найкоротшого шляху дає змогу автоматизувати процес побудови маршруту. Це створює основу для розроблення мобільно-орієнтованої системи навігації, яка може допомогти користувачам швидше орієнтуватися на території гірськолижного курорту та обирати зручний шлях між його об'єктами.

1.2 Загальна характеристика гірськолижного курорту “Буковель”

Гірськолижний курорт “Буковель” є одним із найбільших і найвідоміших гірськолижних курортів України. Він розташований неподалік від села Поляниця Івано-Франківської області, біля підніжжя гори Буковель, на висоті близько 920 м над рівнем моря. Найвищою точкою курорту є гора Довга, висота якої

становить 1372 м. Завдяки розвиненій туристичній та спортивній інфраструктурі курорт щороку приймає значну кількість відвідувачів, серед яких є як досвідчені лижники, так і туристи-початківці [2].

Курорт розташований на кількох гірських схилах, що формують розгалужену систему трас і підйомників. До основних гір, на яких розміщені траси курорту, належать Довга, Буковель, Бульчиньоха, Бабин Погар та Чорна Клева. Така просторова організація створює складну систему пересування, у межах якої користувач повинен орієнтуватися між різними трасами, підйомниками та пересадковими точками. На рисунку 1.1 наведено схему трас і підйомників гірськолижного курорту “Буковель” за матеріалами офіційного сайту курорту [3]. Ця схема використовується для загального уявлення про просторову структуру курорту та подальшого формування графової моделі системи навігації.



Рисунок 1.1 – Схема трас і підйомників гірськолижного курорту “Буковель”

Як видно з рисунка 1.1 інфраструктура курорту включає гірськолижні траси різного рівня складності. Траси поділяються на легкі, середні та складні, що дає змогу користувачам обирати напрям руху відповідно до власного рівня підготовки

[3]. Довжина окремих трас, їх розташування, перепад висот і зв'язок із підйомниками є важливими характеристиками, які необхідно враховувати під час побудови маршруту.

Важливими елементами інфраструктури курорту є підйомники. Вони забезпечують переміщення туристів між нижніми та верхніми станціями трас і фактично виконують роль ключових вузлів гірськолижної мережі. З погляду побудови маршруту підйомники доцільно розглядати як вершини графа, а траси та переходи між ними — як ребра, що поєднують ці вершини. Такий підхід дозволяє формалізувати задачу навігації та застосовувати алгоритми пошуку найкоротшого шляху.

В Україні вимоги до будови та безпечної експлуатації пасажирських підвісних канатних доріг визначаються НПАОП 60.2-1.02-14 “Правила будови і безпечної експлуатації пасажирських підвісних канатних доріг” [4]. З урахуванням цього підйомники доцільно розглядати не лише як транспортні об’єкти курорту, а й як важливі технічні елементи інфраструктури, які необхідно враховувати під час моделювання маршрутів і формування структури системи навігації.

З погляду розроблення програмної системи навігації курорт “Буковель” є зручним прикладом предметної області, оскільки має значну кількість трас, підйомників і можливих варіантів переміщення між ними. Для користувача важливо не лише бачити загальну схему курорту, а й мати можливість вибрати початкову та кінцеву точку маршруту й отримати рекомендований шлях. Тому інфраструктуру курорту доцільно подати у вигляді графової моделі, де кожен елемент має відповідне функціональне призначення в системі побудови маршрутів.

Отже, гірськолижний курорт “Буковель” має розгалужену структуру трас і підйомників [3], що створює потребу в зручній системі навігації. Наявність трас різної складності, значної кількості підйомників і можливих варіантів пересування обґрунтовує доцільність розроблення мобільно-орієнтованої

системи побудови маршрутів, яка дозволить користувачу швидше орієнтуватися на території курорту та вибирати оптимальний шлях між його об'єктами.

1.3 Аналіз існуючих картографічних і навігаційних сервісів

Для розроблення системи навігації та побудови маршрутів на гірськолижних курортах доцільно проаналізувати наявні картографічні та навігаційні сервіси, які можуть використовуватися туристами для орієнтування на місцевості. Найбільш поширеними серед таких засобів є універсальні картографічні сервіси, зокрема Google Maps [5], а також офіційні інформаційні ресурси конкретних курортів. Вони дають змогу отримати загальне уявлення про територію, розташування окремих об'єктів та наявну інфраструктуру, однак не завжди повною мірою враховують специфіку гірськолижного середовища.

Одним із найпоширеніших картографічних сервісів є Google Maps. Цей сервіс надає користувачам доступ до карт місцевості, супутникових знімків, пошуку об'єктів, побудови автомобільних, пішохідних і транспортних маршрутів. Google Maps має зручний інтерфейс і може використовуватися як на персональних комп'ютерах, так і на мобільних пристроях. Для туристів цей сервіс є корисним під час пошуку розташування курорту, під'їзних шляхів, готелів, закладів харчування та інших об'єктів загальної інфраструктури.

Разом з тим Google Maps має обмеження для задачі побудови маршрутів саме на гірськолижному курорті. Сервіс переважно орієнтований на автомобільну, пішохідну та громадську навігацію, тому не враховує повною мірою структуру гірськолижних трас, напрямки спуску, рівні складності трас, взаємозв'язок між підйомниками та можливість переміщення користувача з урахуванням гірськолижної логіки. Унаслідок цього користувач може бачити загальне розташування курорту, але не отримує готового маршруту між окремими трасами або підйомниками.

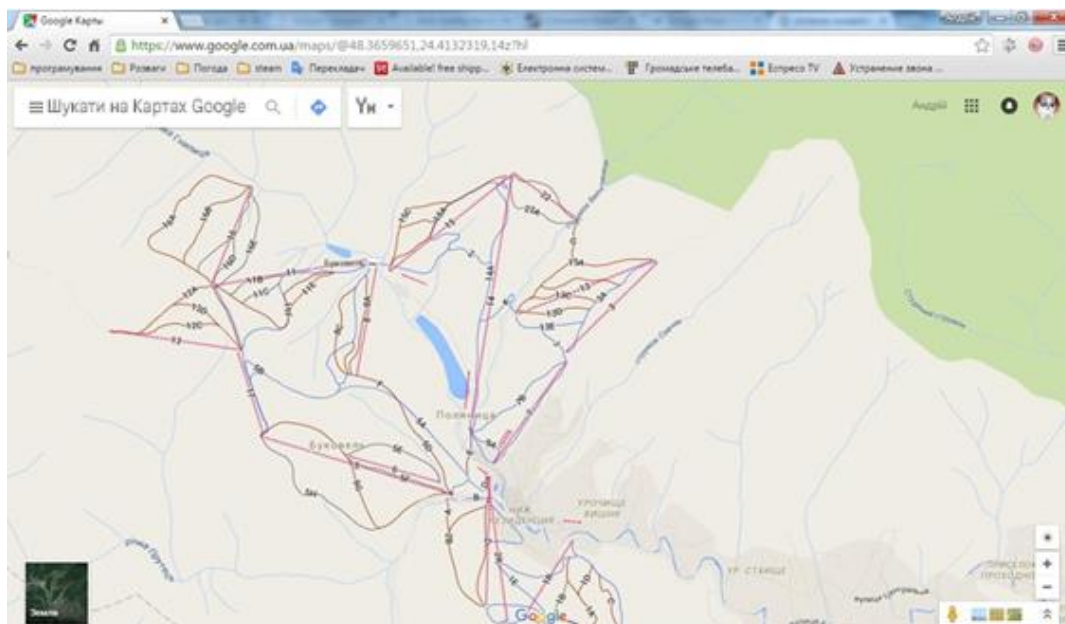


Рисунок 1.2 – Приклад відображення території гірськолижного курорту “Буковель” у сервісі Google Maps

Перевагою офіційного сайту курорту є те, що інформація на ньому безпосередньо пов’язана з реальною інфраструктурою “Буковелю”. Користувач може переглядати інформацію про траси та підйомники, їх стан, доступність та інші важливі характеристики. Однак такий ресурс переважно виконує інформаційну функцію. Він надає користувачу дані про об’єкти курорту, але не завжди забезпечує автоматизовану побудову оптимального маршруту між вибраними точками з урахуванням послідовності руху по трасах і підйомниках.

Для задачі навігації на гірськолижному курорті важливо не лише відобразити карту або схему трас, а й забезпечити користувачу можливість вибрати початкову та кінцеву точку маршруту. Після цього система повинна визначити оптимальний шлях, подати послідовність трас і підйомників, а також відобразити результат у зрозумілому вигляді. Саме така функціональність є важливою для туристів, які погано орієнтуються на території курорту або хочуть швидко визначити зручний шлях до потрібної траси чи підйомника.

Універсальні картографічні сервіси та офіційні сайти курортів мають важливе практичне значення, однак вони не повністю розв’язують задачу

побудови маршруту в межах гірськолижної інфраструктури. Google Maps зручний для загального орієнтування та пошуку місця розташування курорту, а офіційний сайт “Буковелю” містить актуальні дані про траси та підйомники. Проте для автоматизованої побудови маршруту між об’єктами курорту доцільно розробити окрему систему, яка враховує структуру трас, підйомників, напрямки руху та графову модель гірськолижного курорту.

Результати порівняння існуючих сервісів за ключовими критеріями наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих картографічних і навігаційних сервісів

Критерій порівняння	Google Maps	Офіційний сайт курорту “Буковель”	Розроблювана система
Відображення території курорту	Забезпечує загальне відображення місцевості	Містить спеціалізовану інформацію про курорт	Орієнтована саме на структуру трас і підйомників
Відображення трас і підйомників	Може відображати окремі елементи, але не завжди повно	Містить карту трас, витягів і їх статус	Використовує траси та підйомники як елементи графової моделі
Побудова маршруту між трасами	Не забезпечує спеціалізованої побудови маршруту	Переважно надає інформацію, а не будує маршрут	Забезпечує побудову маршруту між вибраними точками
Урахування напрямку руху по трасах	Не враховується	Частково відображається на схемі курорту	Враховується під час формування графової моделі
Урахування складності трас	Не є основною функцією сервісу	Траси позначаються за рівнем складності	Може використовуватися як додаткова характеристика маршруту
Використання на мобільних пристроях	Підтримується	Підтримується через веб-сайт	Передбачається використання через адаптивний інтерфейс
Основне призначення	Загальна картографічна навігація	Інформування про інфраструктуру курорту	Побудова маршруту в межах гірськолижної інфраструктури

Як видно з таблиці 1.1, Google Maps є зручним засобом для загального орієнтування на місцевості, а офіційний сайт курорту “Буковель” містить спеціалізовану інформацію про траси, підйомники та їх стан. Водночас ці сервіси не забезпечують повноцінної автоматизованої побудови маршруту між

вибраними об'єктами гірськолижної інфраструктури. Тому доцільною є розробка мобільно-орієнтованої системи навігації з адаптивним інтерфейсом, яка використовує графову модель трас і підйомників та дозволяє формувати маршрут між обраними точками курорту.

1.4 Аналіз методів та алгоритмів побудови маршрутів

Для побудови маршрутів у графах можуть застосовуватися різні алгоритми. Вибір конкретного алгоритму залежить від того, чи враховуються ваги ребер, чи має граф напрямки руху, а також від того, які критерії використовуються для визначення оптимального маршруту.

Одним із найпростіших підходів є пошук у ширину [6]. Він використовується для знаходження шляху в незваженому графі, тобто в такому випадку, коли всі переходи між вершинами вважаються однаковими. Такий алгоритм дозволяє знайти шлях із найменшою кількістю переходів, однак не враховує довжину трас, час проходження або складність маршруту.

Алгоритм Дейкстри використовується для пошуку найкоротшого шляху у зваженому графі з невід'ємними вагами ребер [7]. У задачі навігації на гірськолижному курорті вагами можуть бути довжини трас або орієнтовний час переміщення між окремими точками. Завдяки цьому алгоритм Дейкстри є придатним для визначення оптимального маршруту між підйомниками, трасами та іншими об'єктами курорту.

Алгоритм Беллмана-Форда також призначений для пошуку найкоротших шляхів у графі. Його особливістю є можливість роботи з від'ємними вагами ребер. Проте для задачі побудови маршрутів на гірськолижному курорті така можливість не є необхідною, оскільки довжина траси або час проходження не можуть мати від'ємного значення [8]. Тому використання цього алгоритму в даній роботі є надлишковим.

Алгоритм A^* є алгоритмом пошуку маршруту, який поєднує фактичну вартість уже пройденого шляху та евристичну оцінку відстані до кінцевої точки. Він часто використовується в задачах просторової навігації, однак потребує правильного вибору евристичної функції [9]. Для базової реалізації системи навігації достатнім є використання алгоритму Дейкстри, оскільки він забезпечує пошук найкоротшого шляху за заданими вагами ребер і має простішу логіку реалізації. Результати порівняння алгоритмів наведено в таблиці 1.2.

Таблиця 1.2 – Порівняння алгоритмів побудови маршрутів у графі

Алгоритм	Основне призначення	Переваги	Недоліки	Доцільність використання в роботі
Пошук у ширину	Пошук шляху в незваженому графі	Простий у реалізації, швидко знаходить шлях за мінімальною кількістю переходів	Не враховує довжину трас, час проходження та інші вагові характеристики	Може використовуватися лише для простих маршрутів без урахування довжин трас
Алгоритм Дейкстри	Пошук найкоротшого шляху у зваженому графі з невід’ємними вагами	Враховує ваги ребер, добре підходить для пошуку найкоротшого маршруту, має зрозумілу логіку реалізації	Не працює з від’ємними вагами, але для даної задачі це не є обмеженням	Є найбільш доцільним для побудови маршруту між трасами та підйомниками
Алгоритм Беллмана-Форда	Пошук найкоротшого шляху у графах, де можуть бути від’ємні ваги	Працює з від’ємними вагами, може виявляти від’ємні цикли	Має більшу обчислювальну складність порівняно з алгоритмом Дейкстри	Для даної задачі є надлишковим, оскільки довжини трас і час проходження не можуть бути від’ємними
Алгоритм A^*	Пошук маршруту з використанням евристичної оцінки	Ефективний для просторової навігації, може швидко знаходити маршрут до цілі	Потребує правильно підбраної евристичної функції, складніший у реалізації	Може бути використаний у подальшому розвитку системи, але для базової реалізації достатньо алгоритму Дейкстри

Як видно з таблиці 1.2, для задачі побудови маршруту на гірськолижному курорті найбільш доцільним є використання алгоритму Дейкстри. Він дозволяє

враховувати вагові характеристики ребер графа, зокрема довжину трас або орієнтовний час переміщення, і не потребує використання від’ємних ваг. Тому цей алгоритм обрано як основу для реалізації пошуку найкоротшого маршруту в системі навігації.

1.5 Постановка задачі бакалаврської роботи

На основі проведеного аналізу предметної області, існуючих картографічних і навігаційних сервісів, а також методів побудови маршрутів у графових структурах можна сформулювати основну задачу бакалаврської роботи. Вона полягає в розробленні системи навігації та побудови маршрутів на гірськолижному курорті, яка забезпечує вибір початкової та кінцевої точки маршруту, визначення оптимального шляху між ними та відображення результату користувачу.

Гірськолижний курорт має розгалужену структуру трас, підйомників і пересадкових точок, тому для розв’язання задачі навігації доцільно подати його інфраструктуру у вигляді графової моделі. У такій моделі підйомники, станції та ключові точки маршруту можуть бути представлені вершинами графа, а траси та переходи між ними — ребрами. Вагами ребер можуть виступати довжина траси, орієнтовний час проходження або інші характеристики, які впливають на вибір маршруту.

Вхідними даними для роботи системи є інформація про об’єкти гірськолижного курорту: перелік підйомників, трас, їх взаємозв’язки, довжини ділянок, напрямки руху та рівні складності. Ці дані використовуються для формування графової моделі курорту, на основі якої виконується пошук маршруту між вибраними користувачем точками.

Система дозволяє користувачеві обрати початкову та кінцеву точки маршруту. На основі цих даних автоматично розраховується оптимальний шлях

за допомогою алгоритму Дейкстри, який є ефективним для зважених графів із невід’ємними вагами.

Вагами ребер графа виступають довжини трас або умовна вартість переходу між об’єктами курорту. Результат обчислення формується у вигляді послідовності ключових точок, трас і підйомників, що значно спрощує орієнтування на місцевості та скорочує час на пошук потрібного напрямку.

Розроблювана система повинна бути орієнтована на використання з мобільних пристроїв. Водночас користувацький інтерфейс може бути реалізований у вигляді адаптивного веб-інтерфейсу, що відкривається у браузері смартфона, планшета або персонального комп’ютера. Такий підхід забезпечує доступність системи для різних типів пристроїв і не потребує обов’язкового встановлення окремого застосунку.

У результаті виконання поставленої задачі має бути отримана програмна система, яка забезпечує:

- подання структури гірськолижного курорту у вигляді графової моделі;
- вибір початкової та кінцевої точки маршруту;
- пошук найкоротшого маршруту між вибраними точками;
- відображення побудованого маршруту в користувацькому інтерфейсі;
- можливість використання системи на пристроях із різними розмірами екрана.

З метою розширення функціональних можливостей системи доцільно передбачити допоміжні розрахункові сервіси: розрахунок вартості катання та орієнтовний розрахунок витрат пального для автомобільного переміщення між віддаленими об’єктами або зонами курорту.

Отже, поставлена задача полягає у створенні мобільно-орієнтованої системи навігації, яка використовує графову модель гірськолижного курорту та алгоритм пошуку найкоротшого шляху. Розроблення такої системи дозволяє поєднати інформацію про траси й підйомники з алгоритмічним підходом до побудови маршрутів і забезпечити користувачу зручний засіб орієнтування на території курорту.

2 ПРОЄКТУВАННЯ МОБІЛЬНО-ОРІЄНТОВАНОЇ СИСТЕМИ НАВІГАЦІЇ

2.5 Формування графової моделі гірськолижного курорту

Для реалізації системи навігації та побудови маршрутів на гірськолижному курорті необхідно формалізувати структуру курорту у вигляді моделі, придатної для алгоритмічної обробки. Оскільки інфраструктура гірськолижного курорту складається з трас, підйомників, станцій та пересадкових точок, її доцільно подати у вигляді графа.

Графова модель дозволяє описати взаємозв'язки між окремими об'єктами курорту та використати алгоритми пошуку найкоротшого шляху. У межах даної роботи гірськолижний курорт розглядається як сукупність вершин і ребер. Вершинами графа є підйомники, станції підйомників, точки перетину трас або інші ключові точки маршруту. Ребрами графа є гірськолижні траси або переходи між окремими вершинами.

Такий підхід відповідає структурі гірськолижного курорту, оскільки турист переміщується між окремими точками не довільно, а відповідно до наявних трас і підйомників. Траси забезпечують рух униз по схилу, а підйомники — переміщення користувача до верхніх станцій. Тому під час формування графової моделі важливо враховувати не лише факт з'єднання між об'єктами, а й напрямок руху.

Для гірськолижної навігації доцільно використовувати орієнтований зважений граф. Орієнтованість графа пояснюється тим, що рух по трасах переважно відбувається в одному напрямку — від верхньої точки до нижньої. Підйомники також мають визначений напрям переміщення — від нижньої станції до верхньої. Зваженість графа означає, що кожному ребру відповідає певна вага, яка характеризує відповідну ділянку маршруту.

Як ваги ребер у графовій моделі можуть використовуватися такі характеристики:

- довжина гірськолижної траси;
- орієнтовний час проходження ділянки;
- умовна складність маршруту;
- кількість переходів між трасами або підйомниками.

У межах базової реалізації системи основним критерієм вибору маршруту є довжина шляху. Тому вага ребра визначається на основі довжини відповідної траси або переходу між об'єктами курорту. Такий підхід дає змогу застосувати алгоритм пошуку найкоротшого шляху для визначення оптимального маршруту між початковою та кінцевою точкою.

Для формування графової моделі використано схему трас і підйомників гірськолижного курорту, на якій відображено основні елементи інфраструктури, напрямки руху та можливі переходи між окремими ділянками курорту. Ця схема є вихідною основою для подальшого подання курорту у вигляді графа.

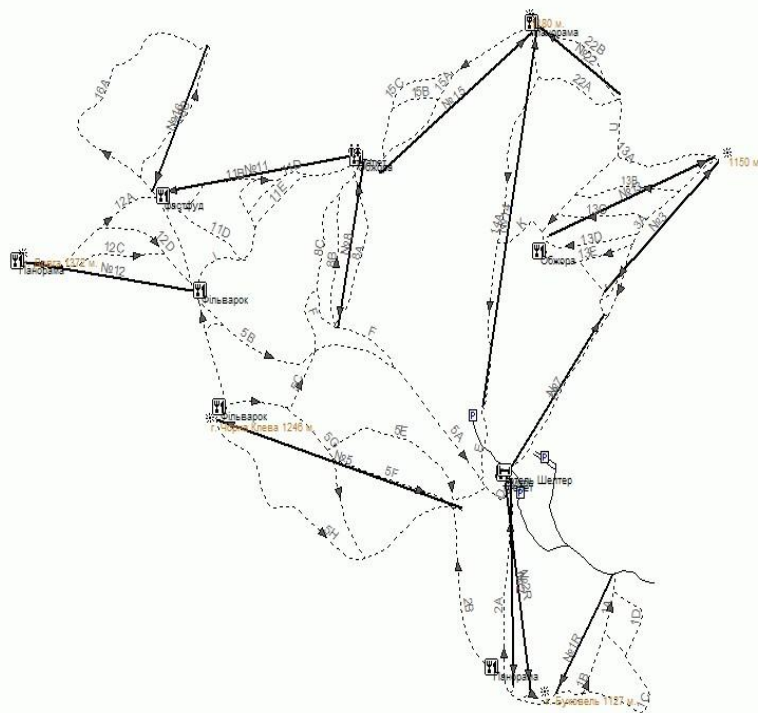


Рисунок 2.1 – Схема трас і підйомників, використана для формування графової моделі курорту

Графову модель гірськолижного курорту можна подати у вигляді множини:

$$G = (V, E),$$

де V — множина вершин графа, що відповідають підйомникам, станціям або ключовим точкам курорту; E — множина ребер, що відповідають трасам або переходам між вершинами.

Кожне ребро графа можна описати у вигляді:

$$e = (v_i, v_j, w),$$

де v_i — початкова вершина маршруту, v_j — кінцева вершина маршруту, w — вага ребра, що визначає довжину або умовну вартість переходу між цими вершинами.

На рисунку 2.2 наведено приклад формування графової моделі гірськолижного курорту на основі трас і підйомників.

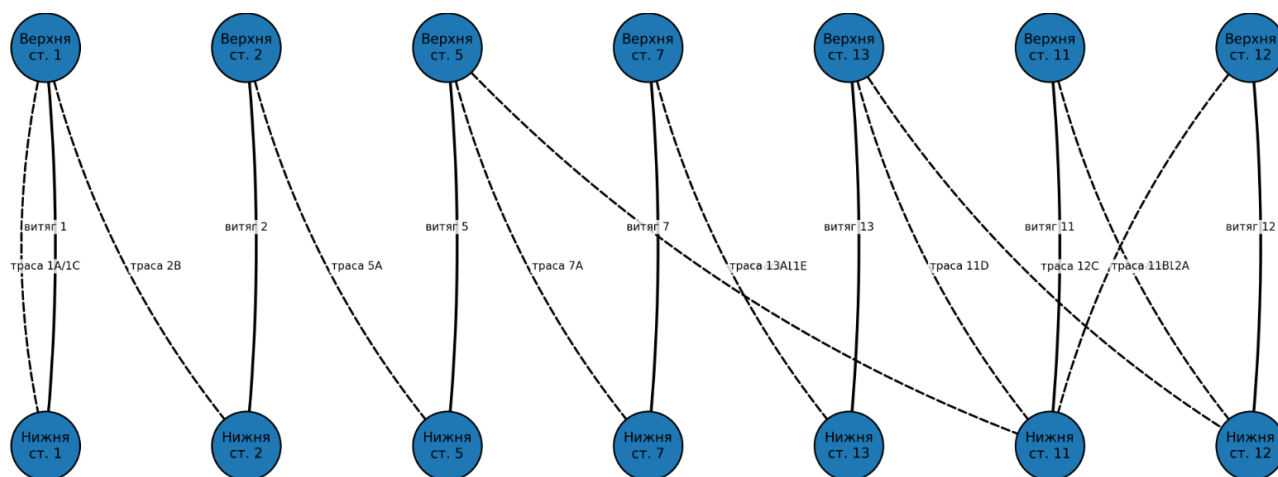


Рисунок 2.2 – Спрощена графова модель трас і підйомників гірськолижного курорту

На рисунку 2.2 вершини графа відповідають нижнім і верхнім станціям підйомників або ключовим точкам маршруту. Наприклад, позначення типу “Нижня ст. 1” і “Верхня ст. 1” відображають нижню та верхню станції відповідного підйомника. У моделі суцільні стрілки позначають підйомники, а пунктирні стрілки — гірськолижні траси або переходи між окремими точками. Ребра графа показують можливі напрямки переміщення між вершинами. Переміщення від нижньої станції до верхньої відповідає руху підйомником, а

переміщення від верхньої точки до нижньої або до іншої станції відповідає руху гірськолижною трасою чи переходом. Таким чином, кожен можливий шлях користувача на курорті може бути поданий як послідовність вершин і ребер графа.

Запропонована модель є спрощеною, оскільки не відображає всі траси та підйомники курорту, а демонструє принцип формалізації інфраструктури для задачі побудови маршруту. У подальшій реалізації системи кожному ребру графа може бути задана вага, наприклад довжина траси, орієнтовний час проходження або умовна складність ділянки. Це дає змогу використовувати графову модель для пошуку найкоротшого маршруту між вибраними користувачем точками.

У запропонованій моделі підйомники та ключові точки курорту виконують роль вузлів маршруту. Траси та переходи між ними забезпечують зв'язки між цими вузлами. Завдяки цьому задача побудови маршруту зводиться до задачі пошуку найкоротшого шляху у зваженому графі.

Для роботи системи користувач обирає початкову та кінцеву точку маршруту. Після цього система знаходить відповідні вершини у графовій моделі та виконує пошук найкоротшого шляху між ними. Результатом є послідовність вершин і ребер, яка відповідає рекомендованому маршруту для користувача.

Отже, формування графової моделі є основою для подальшого проектування системи навігації. Подання трас і підйомників у вигляді орієнтованого зваженого графа дозволяє формалізувати структуру гірськолижного курорту та застосувати алгоритми пошуку найкоротшого маршруту.

2.2 Опис інформаційної структури трас, підйомників і маршрутів

Для побудови системи навігації необхідно визначити інформаційну структуру об'єктів, які використовуються під час формування маршруту. Основними елементами такої структури є гірськолижні траси, підйомники,

станції підйомників, переходи між трасами та маршрут, який формується системою на основі вибраних користувачем початкової і кінцевої точок.

У межах розроблюваної системи гірськолижний курорт подається як сукупність взаємопов'язаних об'єктів. Траси характеризують можливі напрямки спуску, підйомники забезпечують переміщення користувача до верхніх станцій, а маршрут є послідовністю таких елементів, які необхідно пройти для досягнення вибраної кінцевої точки. Такий підхід дозволяє перейти від звичайної карти курорту до структурованої інформаційної моделі, придатної для програмної обробки.

Основними характеристиками гірськолижної траси є її назва або номер, рівень складності, довжина, висота верхньої точки та зв'язок із відповідними підйомниками або станціями. Рівень складності траси використовується для інформування користувача про характер маршруту, а довжина траси може бути використана як вага ребра у графівій моделі. Саме довжина є одним із найважливіших параметрів під час пошуку найкоротшого маршруту.

Характеристики трас гірськолижного курорту “Буковель”, які використовуються для формування інформаційної структури системи [3], наведено в таблиці 2.1.

Таблиця 2.1 – Характеристика трас гірськолижного курорту “Буковель”

Траса	Тип	Довжина,	Верхня висота
1 А	Середня	1602	1126
1 В	Складна	296	1070
1 С	Легка	1587	1126
1 D	Середня	335	995
11 А	Середня	1548	1202
11 В	Складна	804	1197
11 С	Складна	420	1185
11 D	Середня	1593	1204
11 Е	Середня	716	1100
12 А	Середня	1674	1371
12 В	Середня	758	1320

Продовження таблиці 2.1

Траса	Тип	Довжина,	Верхня висота
12 С	Середня	696	1305
12 D	Середня	440	1255
13 А	Середня	1308	1141
13 В	Середня	650	1110
13 С	Середня	570	1075
13 D	Середня	1279	1149
13 Е	Легка	1338	1149
14 А	Легка	2132	1189
15 А	Середня	1528	1189
15 В	Середня	331	1060
15 С	Середня	815	1090
15 D	Середня	1262	1170
16 А	Середня	1741	1205
16 В	Середня	900	1180
16 С	Складна	964	1195
16 D	Складна	528	1205
16 Е	Середня	580	1070
2 А	Середня	915	1110
2 В	Легка	1459	1115
22 А	Середня	501	1135
22 В	Середня	707	1186
3 А	Легка	290	1050
5 А	Легка	1794	1105
5 В	Легка	1446	1236
5 С	Середня	340	1195
5 D	Середня	764	1050
5 Е	Складна	960	1050
5 F	Складна	768	1135
5 G	Середня	2106	1235
5 Н	Середня	1549	1236
7 А	Легка	1048	1013
8 А	Середня	936	1080
8 В	Середня	757	1111

Траса	Тип	Довжина,	Верхня висота
8 C	Середня	750	1105
9 A	Легка	200	927
A	Середня	408	975
B	Легка	40	915
C	Середня	270	895
D	Середня	245	940
E	Легка	282	960
F	Середня	294	1112
G	Середня	480	1103
H	Легка	162	1183
I	Середня	451	1153
J	Легка	221	1015
K	Легка	632	1145
L	Середня	873	977
R	Середня	750	1135
T	Середня	300	1145
U	Середня	330	990

Наведені характеристики дозволяють описати трасу як окремий інформаційний об'єкт системи. У програмній моделі кожна траса може бути представлена записом, що містить її ідентифікатор, назву, складність, довжину, а також зв'язок із початковою та кінцевою точками маршруту. У графівій моделі траса розглядається як орієнтоване ребро, що з'єднує відповідні вершини.

Підйомники в інформаційній структурі системи виконують роль об'єктів, які забезпечують переміщення користувача між нижньою та верхньою станціями. Кожен підйомник може бути описаний номером або назвою, нижньою і верхньою станцією, напрямом руху та зв'язком із трасами, які починаються або закінчуються біля відповідних станцій. У графівій моделі

підйомник також може розглядатися як орієнтоване ребро, що відображає рух від нижньої станції до верхньої.

Станції підйомників і ключові точки курорту в системі розглядаються як вершини графа. Для кожної вершини доцільно зберігати її ідентифікатор, назву, тип об'єкта та зв'язки з іншими вершинами. Типом об'єкта може бути нижня станція, верхня станція, точка перетину трас або інша важлива точка маршруту.

Маршрут у системі є результатом обробки графової моделі. Він формується як послідовність вершин і ребер, що з'єднують початкову та кінцеву точки. До складу маршруту можуть входити траси, підйомники та переходи між ними. Для кожного маршруту доцільно визначати загальну довжину, перелік ділянок, послідовність проходження та, за потреби, інформацію про складність окремих трас.

Загальну інформаційну структуру системи можна подати у вигляді таких основних сутностей:

- “Траса” — містить дані про назву, складність, довжину та зв'язок із точками графа;
- “Підйомник” — містить дані про номер, нижню та верхню станцію;
- “Вершина графа” — описує станцію або ключову точку маршруту;
- “Ребро графа” — описує перехід між двома вершинами та містить вагу;
- “Маршрут” — містить послідовність об'єктів, що формують шлях користувача.

У спрощеному вигляді інформаційну структуру маршруту можна описати так: користувач задає початкову та кінцеву вершини, система знаходить між ними послідовність ребер із мінімальною сумарною вагою, після чого результат подається у вигляді списку трас і підйомників. Така модель є зручною для реалізації алгоритму пошуку найкоротшого маршруту, оскільки всі основні об'єкти курорту мають чітко визначені зв'язки.

Отже, інформаційна структура системи навігації включає дані про траси, підйомники, станції та маршрути. Її основою є графова модель, у якій вершини відповідають ключовим точкам курорту, а ребра — трасам, підйомникам або

переходам між ними. Такий підхід забезпечує можливість алгоритмічної обробки даних і подальшої побудови оптимального маршруту між вибраними користувачем точками.

2.3 Обґрунтування вибору алгоритму пошуку найкоротшого маршруту

Після формування графової моделі наступним етапом є визначення способу пошуку маршруту між вибраними користувачем точками. Для системи навігації важливо не лише встановити наявність зв'язку між окремими трасами та підйомниками, а й визначити такий шлях, який матиме найменшу сумарну довжину або умовну вартість проходження. Саме тому вибір алгоритму пошуку маршруту є одним із ключових етапів проєктування системи. У розроблюваній системі структура курорту подається у вигляді орієнтованого графа, оскільки рух по трасах і підйомниках має визначений напрям: траси забезпечують спуск від верхніх точок до нижніх, а підйомники — переміщення користувача від нижніх станцій до верхніх. Зваженість графа пов'язана з тим, що кожному ребру задається певна вага, наприклад довжина траси, орієнтовний час проходження або умовна вартість переходу.

Оскільки у сформованій моделі всі ваги ребер мають невід'ємні значення, для побудови маршруту доцільно використати алгоритм Дейкстри. У першому розділі було виконано порівняння алгоритмів пошуку маршрутів і визначено, що саме алгоритм Дейкстри є достатнім для розв'язання поставленої задачі. У цьому пункті розглянуто не загальний опис алгоритму, а порядок його застосування в розроблюваній системі навігації.

У межах системи алгоритм Дейкстри використовується для пошуку найкоротшого маршруту між двома точками, які обирає користувач. Початковою та кінцевою точками можуть бути станції підйомників, точки перетину трас або інші ключові вузли графової моделі. Після вибору цих точок система визначає відповідні вершини графа та виконує пошук маршруту з мінімальною сумарною вагою.

Вхідними даними для алгоритму є:

- графова модель гірськолижного курорту;
- множина вершин, що відповідають підйомникам, станціям і ключовим точкам маршруту;
- множина ребер, що відповідають трасам, підйомникам і переходам;
- ваги ребер, які характеризують довжину або умовну вартість проходження ділянки;
- початкова вершина маршруту;
- кінцева вершина маршруту.

Вихідними даними є сформований маршрут, який подається як послідовність вершин і ребер графа. Для користувача цей результат відображається у зрозумілому вигляді: як перелік трас, підйомників або переходів, якими потрібно скористатися для досягнення кінцевої точки.

Особливістю застосування алгоритму в даній системі є те, що результат пошуку повинен бути придатним не лише для внутрішньої програмної обробки, а й для відображення в інтерфейсі користувача. Тому після знаходження найкоротшого шляху система повинна перетворити набір вершин і ребер у послідовність зрозумілих дій: перейти до певного підйомника, піднятися до верхньої станції, спуститися відповідною трасою або перейти до наступної точки маршруту.

У спрощеному вигляді порядок роботи алгоритму в системі можна описати так. Спочатку система завантажує графову модель курорту. Після цього користувач вибирає початкову та кінцеву точки маршруту. Для всіх вершин графа встановлюється початкова відстань, що вважається нескінченною, а для початкової вершини відстань дорівнює нулю. Далі система послідовно вибирає неперевірену вершину з найменшим поточним значенням відстані та аналізує суміжні з нею вершини. Якщо шлях через поточну вершину є коротшим, ніж раніше знайдений, значення відстані оновлюється, а для суміжної вершини зберігається попередня вершина маршруту.

Процес повторюється доти, доки не буде досягнуто кінцеву вершину або доки не залишиться вершин, які можна перевірити. Після завершення роботи алгоритму система відновлює маршрут за збереженими попередніми вершинами та формує результат для виведення користувачу.

Послідовність роботи алгоритму побудови маршруту наведено на рисунку 2.3.

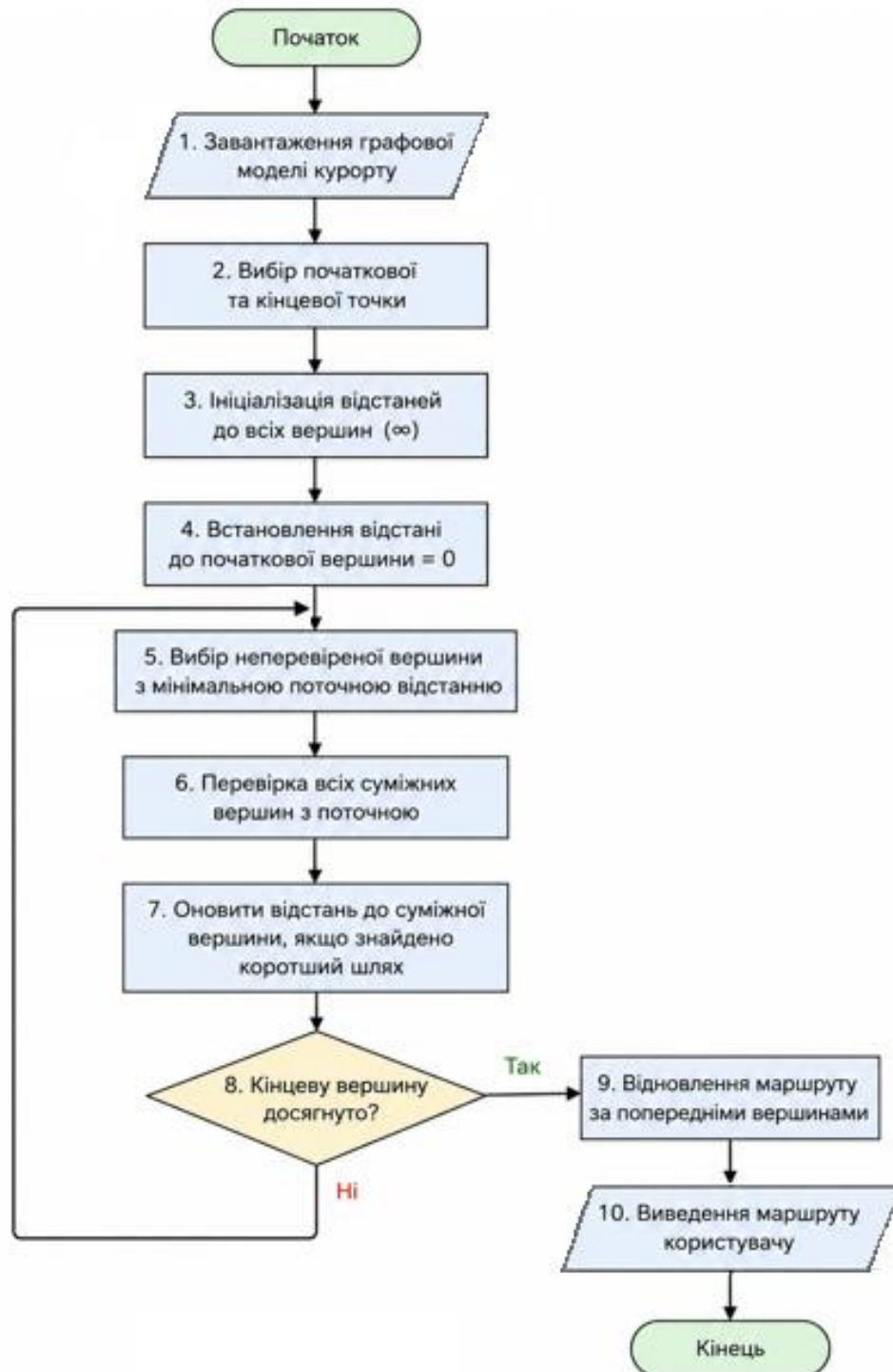


Рисунок 2.3 – Блок-схема алгоритму побудови найкоротшого маршруту

Як видно з рисунка 2.3, робота алгоритму починається із завантаження графової моделі курорту та вибору початкової і кінцевої точки маршруту. У даній роботі графову модель можна подавати у вигляді матриці суміжності, яка показує наявність зв'язків між вершинами, а для пошуку найкоротшого маршруту такі зв'язки можуть доповнюватися вагами, наприклад довжинами трас. Після цього система перевіряє зв'язки між вершинами графа, порівнює довжини можливих переходів і поступово знаходить найкоротший шлях. Після завершення пошуку маршрут відновлюється та виводиться користувачу у вигляді послідовності трас, підйомників або переходів.

Застосування алгоритму Дейкстри в розроблюваній системі є обґрунтованим, оскільки структура курорту вже подана у вигляді орієнтованого зваженого графа, а всі ваги ребер є невід'ємними. Це дає змогу коректно визначати маршрут із мінімальною сумарною вагою. У базовій реалізації такою вагою є довжина ділянки, однак у подальшому вона може бути доповнена іншими параметрами, наприклад складністю траси, орієнтовним часом проходження або кількістю переходів між підйомниками.

2.4 Розроблення структури програмної системи

Після формування графової моделі гірськолижного курорту та вибору алгоритму пошуку маршруту необхідно визначити загальну структуру програмної системи. Структура системи повинна забезпечувати зберігання даних про траси й підйомники, вибір користувачем початкової та кінцевої точки маршруту, обробку графової моделі, виконання алгоритму пошуку найкоротшого шляху та відображення результату в інтерфейсі.

Розроблювана система навігації має модульну структуру. Це дозволяє розділити основні функції системи на окремі компоненти та спростити подальше вдосконалення програмної реалізації. До основних компонентів системи

належать: модуль користувацького інтерфейсу, модуль вхідних даних, модуль графової моделі, модуль побудови маршруту та модуль відображення результату.

Модуль користувацького інтерфейсу забезпечує взаємодію користувача із системою. Через нього користувач переглядає інформацію про курорт, обирає початкову та кінцеву точку маршруту, запускає побудову маршруту та отримує результат. Оскільки система є мобільно-орієнтованою, інтерфейс повинен бути зручним для використання на екранах різного розміру, зокрема на смартфонах і планшетах.

Модуль вхідних даних призначений для зберігання та підготовки інформації про об'єкти гірськолижного курорту. До таких даних належать перелік трас, підйомників, станцій, напрямків руху та характеристик окремих ділянок. У програмній реалізації ці дані можуть бути подані у вигляді матриці суміжності, списку вершин і ребер або структурованого набору даних, який використовується для побудови графової моделі.

Модуль графової моделі відповідає за подання інфраструктури курорту у вигляді орієнтованого графа. У цьому модулі визначаються вершини графа, ребра між ними та ваги ребер. Вершинами виступають станції підйомників або ключові точки маршруту, а ребрами — траси, підйомники та переходи. Така модель є основою для роботи алгоритму пошуку найкоротшого маршруту.

Модуль побудови маршруту реалізує обробку графової моделі та застосування алгоритму Дейкстри. Він отримує від користувацького інтерфейсу початкову й кінцеву точки маршруту, знаходить відповідні вершини у графі, виконує пошук найкоротшого шляху та формує послідовність об'єктів, які входять до маршруту.

Модуль відображення результату призначений для подання знайденого маршруту користувачу. Результат може відображатися у вигляді послідовності трас, підйомників або переходів між ключовими точками. Також користувачу може бути показана загальна довжина маршруту та додаткова інформація про окремі ділянки.

Крім основної функції побудови маршруту між об'єктами гірськолижної інфраструктури, система може містити допоміжні розрахункові сервіси. До них належать розрахунок витрат пального для планування під'їзду до окремих зон курорту та розрахунок орієнтовної вартості катання. Такі функції не змінюють логіку побудови маршруту у графовій моделі, але підвищують практичну корисність системи для користувача.

Загальну структуру системи навігації та побудови маршрутів наведено на рисунку 2.4.

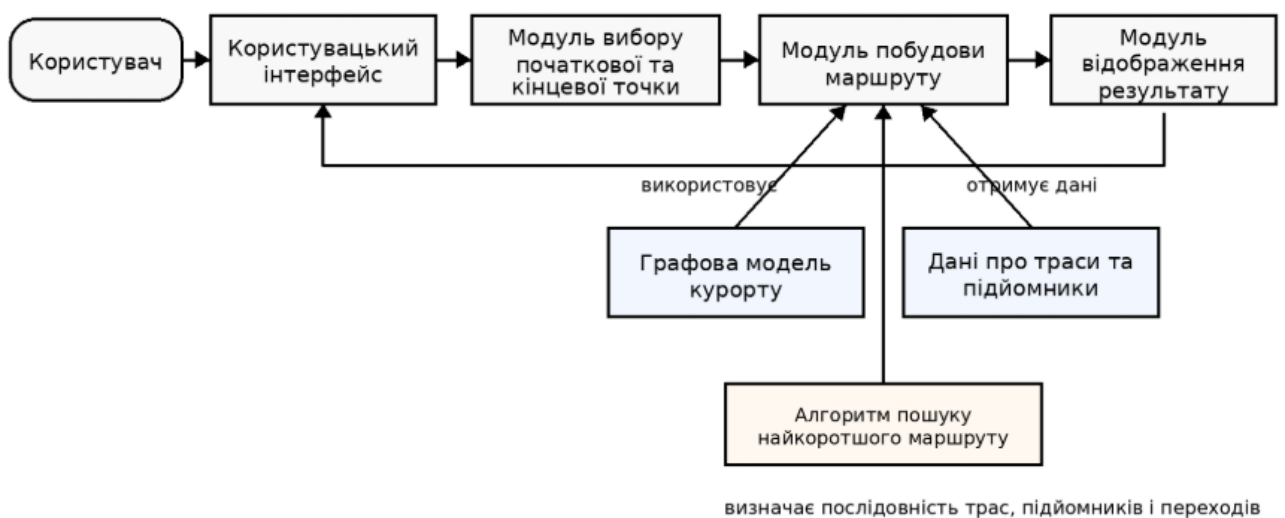


Рисунок 2.4 – Структурна схема системи навігації та побудови маршрутів

Як видно з рисунка 2.4, система навігації та побудови маршрутів має модульну структуру. Користувач взаємодіє із системою через користувацький інтерфейс, у якому здійснюється вибір початкової та кінцевої точки маршруту. Після вибору точок дані передаються до модуля побудови маршруту. Цей модуль використовує графову модель курорту та дані про траси й підйомники, на основі яких виконується пошук найкоротшого шляху.

Графова модель курорту відображає зв'язки між основними об'єктами інфраструктури, зокрема трасами, підйомниками та ключовими точками маршруту. Модуль побудови маршруту застосовує алгоритм пошуку найкоротшого шляху та визначає послідовність трас, підйомників і переходів,

якими користувач може дістатися до вибраної кінцевої точки. Після завершення обчислень результат передається до модуля відображення результату та виводиться користувачу в зручному вигляді.

Для уточнення функціональних можливостей системи доцільно побудувати діаграму варіантів використання. Основним актором системи є користувач, який взаємодіє з інтерфейсом і виконує основні дії: переглядає інформацію про курорт, вибирає початкову точку, вибирає кінцеву точку, запускає побудову маршруту та переглядає результат.

Діаграму варіантів використання системи навігації наведено на рисунку 2.5.

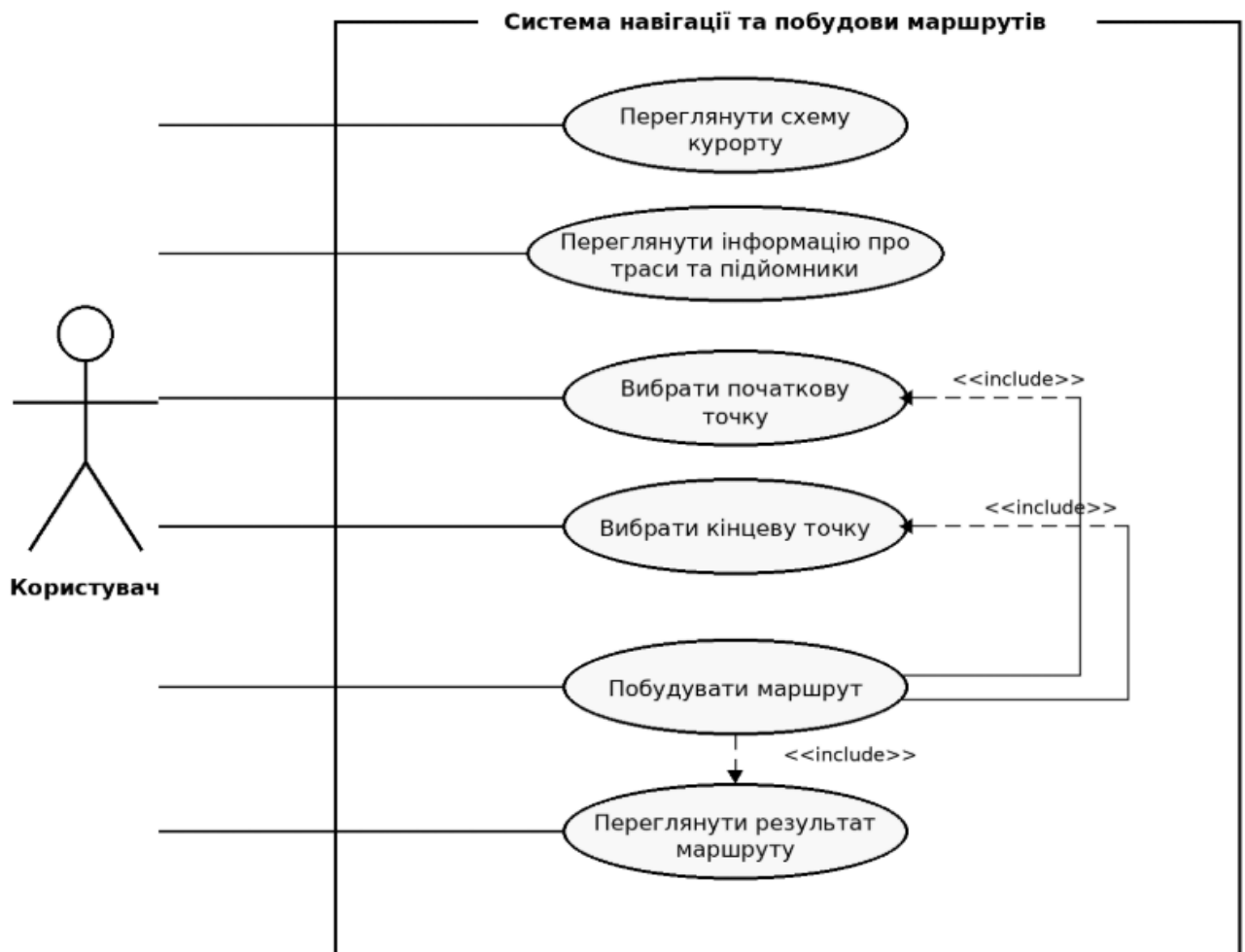


Рисунок 2.5 – Діаграма варіантів використання системи навігації

На рисунку 2.5 наведено UML-діаграму варіантів використання системи навігації [10]. Основним актором є користувач, який може переглядати схему курорту, отримувати інформацію про траси та підйомники, вибирати початкову й кінцеву точки маршруту, запускати побудову маршруту та переглядати отриманий результат. Варіант використання “Побудувати маршрут” пов’язаний із вибором початкової та кінцевої точки, оскільки ці дії є необхідними для формування маршруту.

Для деталізації взаємодії між основними компонентами системи під час побудови маршруту доцільно використати діаграму послідовності. Вона відображає порядок передавання запиту від користувача до програмних модулів системи та повернення результату після виконання алгоритму пошуку найкоротшого шляху. Діаграму послідовності побудови маршруту наведено на рисунку 2.6.

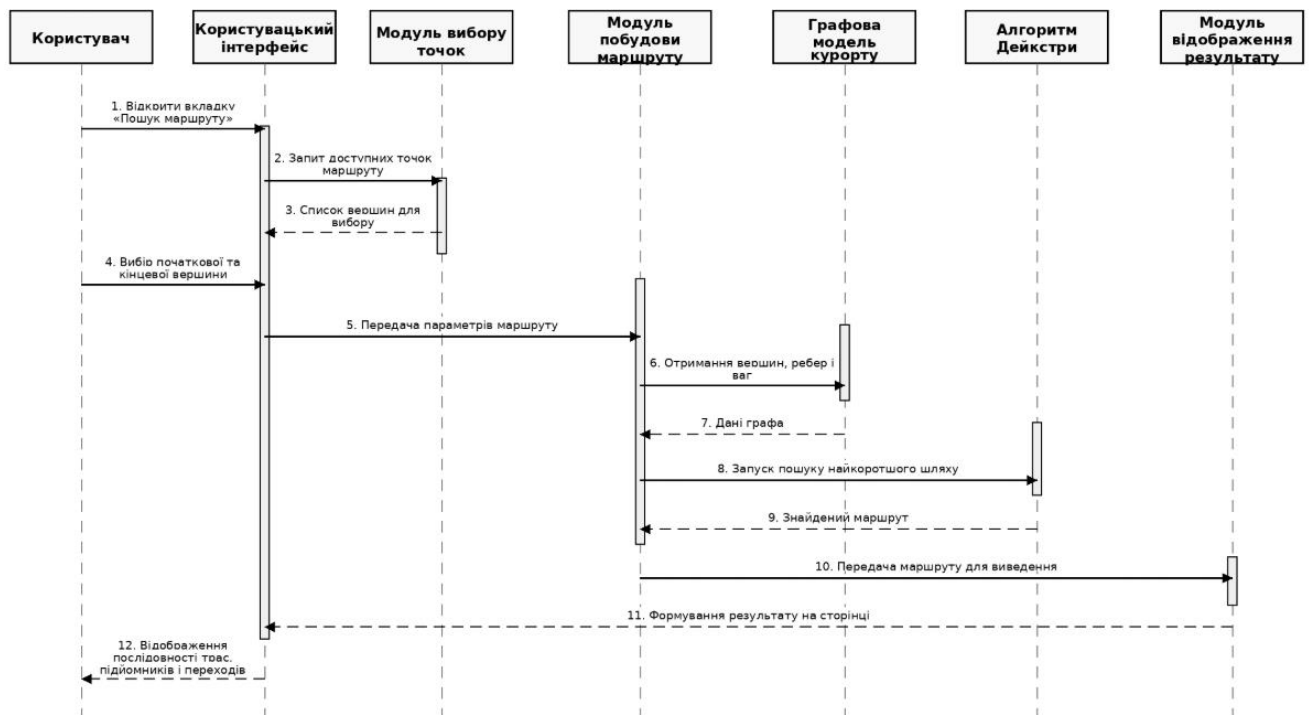


Рисунок 2.6 – Діаграма послідовності побудови маршруту

Як видно з рисунка 2.6, процес побудови маршруту починається з відкриття користувачем вкладки “Пошук маршруту” та вибору початкової і

кінцевої вершини. Після цього користувацький інтерфейс передає вибрані дані до модуля побудови маршруту. Модуль отримує інформацію з графової моделі курорту, запускає алгоритм Дейкстри та після знаходження найкоротшого шляху передає результат для відображення. Користувач отримує маршрут у вигляді послідовності трас, підйомників і переходів

Крім основного модуля побудови маршруту, система містить допоміжні інформаційно-розрахункові сервіси. До них належать розрахунок витрат пального для планування поїздки до курорту автомобільним транспортом та розрахунок вартості катання для попередньої оцінки витрат користувача на ski-pass. Ці сервіси не впливають на алгоритм побудови маршруту, однак розширюють практичні можливості системи.

Отже, структура програмної системи включає набір взаємопов'язаних модулів, кожен із яких виконує окрему функцію. Такий підхід дозволяє чітко розділити роботу з даними, побудову графової моделі, виконання алгоритму пошуку маршруту та відображення результату користувачу. Це спрощує реалізацію системи та забезпечує можливість її подальшого вдосконалення.

2.5 Проектування структури адаптивного користувацького інтерфейсу

Користувацький інтерфейс є важливою складовою системи навігації, оскільки саме через нього користувач взаємодіє з програмною системою, вибирає початкову та кінцеву точки маршруту й отримує результат побудови шляху. Для розроблюваної системи важливо забезпечити простоту використання інтерфейсу, зрозуміле розміщення елементів керування та можливість роботи на пристроях із різними розмірами екрана.

Оскільки система є мобільно-орієнтованою, інтерфейс доцільно проектувати як адаптивний веб-інтерфейс. Такий підхід дозволяє використовувати систему не лише на персональному комп'ютері, а й на

смартфоні або планшеті через браузер. Це спрощує доступ до системи, оскільки користувачу не потрібно встановлювати окремий застосунок на пристрій.

Під час проєктування інтерфейсу враховано основні дії користувача, визначені у діаграмі варіантів використання: перегляд інформації про курорт, перехід до пошуку маршруту, вибір початкової та кінцевої точки, запуск побудови маршруту й перегляд отриманого результату. Крім основної функції пошуку маршруту, інтерфейс системи може містити допоміжні інформаційно-розрахункові сервіси, зокрема розрахунок витрат пального та розрахунок вартості катання.

Головна сторінка системи повинна виконувати навігаційну функцію та забезпечувати перехід до основних розділів. Верхнє меню доцільно використати для швидкого доступу до вкладок “Пошук маршруту”, “Розрахунок витрат пального” і “Розрахунок вартості катання”. Основною вкладкою в межах даної роботи є “Пошук маршруту”, оскільки саме вона забезпечує побудову маршруту між вершинами графової моделі гірськолижного курорту.

У вкладці “Пошук маршруту” користувач повинен мати можливість послідовно виконати основні дії: вибрати початкову вершину, вибрати кінцеву вершину, запустити побудову маршруту та переглянути результат. Результат побудови маршруту доцільно подавати у вигляді впорядкованої послідовності трас, підйомників і переходів. Такий спосіб подання є зрозумілим для користувача та відповідає логіці переміщення в межах гірськолижного курорту.

Загальну структуру адаптивного користувацького інтерфейсу наведено на рисунку 2.7.

Як видно з рисунка 2.7, користувацький інтерфейс системи побудований у вигляді головної сторінки з верхнім навігаційним меню, яке забезпечує перехід між основними функціональними розділами. До таких розділів належать вкладка “Пошук маршруту”, а також допоміжні сервіси “Розрахунок витрат пального” і “Розрахунок вартості катання”.

Основною вкладкою є “Пошук маршруту”. У межах цієї вкладки користувач задає початкову та кінцеву вершини графа, після чого система виконує побудову маршруту. Результат відображається у вигляді послідовності трас, підйомників і переходів. Ліва частина інтерфейсу може використовуватися для відображення схеми курорту або інформаційного блоку, а також для короткого опису можливостей системи.

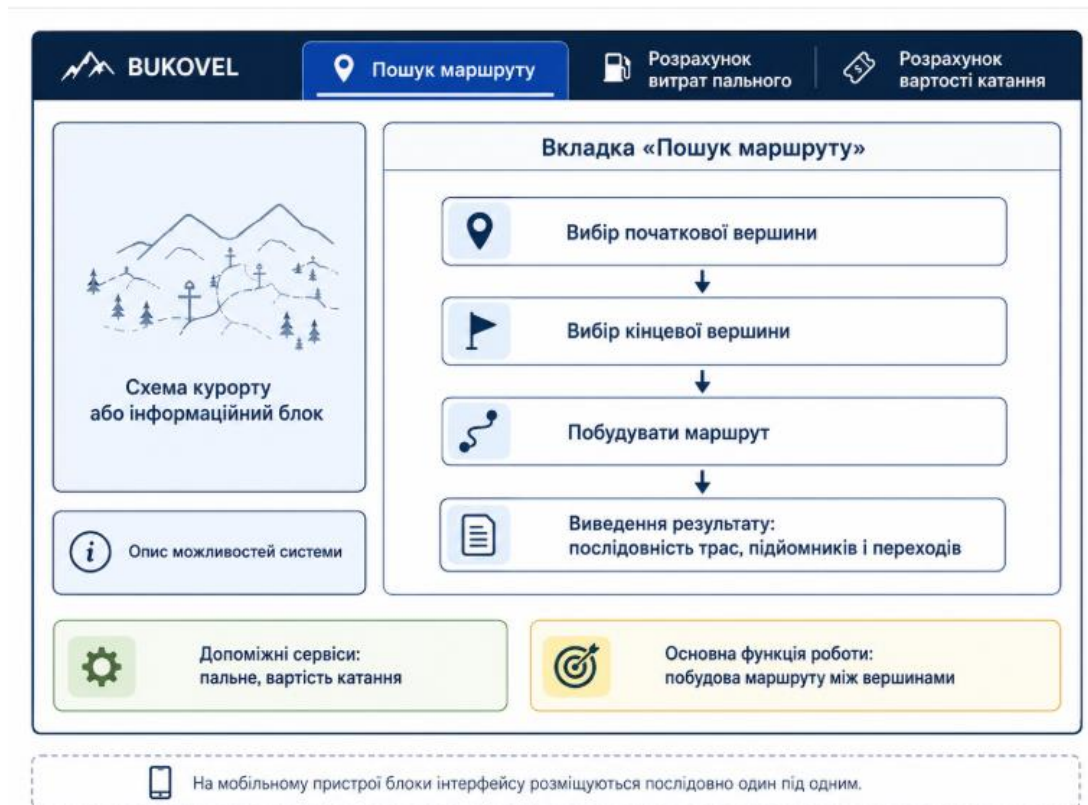


Рисунок 2.7 – Структура адаптивного користувацького інтерфейсу системи

Допоміжні сервіси, пов’язані з розрахунком витрат пального та вартості катання, не змінюють основну логіку побудови маршруту, однак розширюють практичні можливості системи. Розрахунок витрат пального може використовуватися для орієнтовного планування під’їзду до курорту або його окремих зон автомобільним транспортом, а розрахунок вартості катання — для попередньої оцінки витрат користувача.

Для забезпечення адаптивності інтерфейсу його блоки повинні змінювати своє розміщення залежно від ширини екрана. На широкому екрані елементи

можуть розміщуватися горизонтально: схема курорту або інформаційний блок — зліва, а функціональна область пошуку маршруту — справа. На мобільному пристрої ці блоки доцільно розміщувати послідовно один під одним, що забезпечує зручну роботу без горизонтального прокручування.

Отже, на етапі проектування адаптивного користувацького інтерфейсу визначено основні функціональні області системи, структуру вкладки пошуку маршруту та місце допоміжних сервісів у загальному веб-інтерфейсі. Такий підхід забезпечує зручну взаємодію користувача із системою, чітке розміщення основних функціональних елементів та можливість використання інтерфейсу на різних пристроях.

2.6 Вибір програмних засобів реалізації системи

Для реалізації системи навігації та побудови маршрутів необхідно обрати програмні засоби, які забезпечують створення користувацького інтерфейсу, подання графової моделі курорту, обробку даних про траси та підйомники, а також виконання алгоритму пошуку маршруту. Оскільки розроблювана система є мобільно-орієнтованою, важливо забезпечити можливість її використання як на персональному комп'ютері, так і на мобільних пристроях. Тому для інтерфейсної частини доцільно використати веб-технології, а для алгоритмічної частини — мову програмування Kotlin.

Інтерфейсна частина системи може бути реалізована з використанням HTML, CSS і JavaScript. HTML використовується для формування структури сторінок, розміщення навігаційного меню, елементів вибору початкової та кінцевої вершини, кнопок керування та області виведення результату. CSS застосовується для візуального оформлення сторінок, налаштування розміщення блоків і забезпечення адаптивності інтерфейсу для різних розмірів екрана. JavaScript використовується для обробки дій користувача, роботи з елементами сторінки та взаємодії з даними графової моделі.

Використання адаптивного веб-інтерфейсу є доцільним для даної системи, оскільки користувач може відкривати її у браузері на смартфоні, планшеті або персональному комп'ютері. Такий підхід дозволяє забезпечити мобільний доступ до функцій системи без необхідності встановлення окремого застосунку. Це особливо важливо для навігації на гірськолижному курорті, де користувач найчастіше працює із системою саме з мобільного пристрою.

Для зберігання та подання інформації про об'єкти курорту можуть використовуватися структуровані дані. До таких даних належать вершини графа, ребра між ними, назви трас і підйомників, напрямки руху та вагові характеристики окремих ділянок. У програмній реалізації такі дані можуть бути представлені у вигляді матриці суміжності, списку вершин і ребер або структурованого формату JSON. Конкретний спосіб подання даних залежить від реалізації, однак у будь-якому випадку він повинен забезпечувати можливість швидкої обробки графової моделі алгоритмом побудови маршруту.

Для візуалізації графової структури та взаємозв'язків між її елементами може використовуватися бібліотека Cytoscape.js. Вона дає змогу відображати граф у веб-інтерфейсі, працювати з вершинами та ребрами, задавати параметри елементів графа й відображати зв'язки між об'єктами. Для системи навігації на гірськолижному курорті такий підхід є зручним, оскільки траси, підйомники та переходи можна подати як взаємопов'язані елементи графової моделі.

Для опису алгоритмічної частини системи доцільно використати Kotlin, зокрема для подання вершин, ребер, маршруту та логіки пошуку найкоротшого шляху. Такий підхід дозволяє відокремити логіку побудови маршруту від інтерфейсної частини та забезпечити можливість її використання в мобільно-орієнтованій системі.

Kotlin є сучасною мовою програмування, яка широко використовується для створення Android-застосунків, підтримує об'єктно-орієнтований підхід і має зручні засоби для роботи зі структурами даних. Для даної системи це важливо, оскільки графова модель курорту складається з набору об'єктів: вершин, ребер і

маршруту. Такі об'єкти зручно описувати у вигляді класів даних, а алгоритм пошуку маршруту — у вигляді окремої функції або модуля.

Узагальнену структуру основних класів, які можуть використовуватися для подання графової моделі засобами Kotlin, наведено нижче:

```
data class GraphNode(  
    val id: String,  
    val name: String,  
    val type: String  
)  
  
data class GraphEdge(  
    val from: String,  
    val to: String,  
    val name: String,  
    val weight: Double  
)  
  
data class RouteResult(  
    val path: List<String>,  
    val totalWeight: Double  
)
```

У наведеному фрагменті клас `GraphNode` описує вершину графа, наприклад станцію підйомника або ключову точку маршруту. Клас `GraphEdge` описує ребро графа, тобто трасу, підйомник або перехід між двома вершинами. Поле `weight` використовується для збереження ваги ребра, наприклад довжини ділянки або умовної вартості переходу. Клас `RouteResult` описує результат побудови маршруту, який містить послідовність вершин і загальну вагу знайденого шляху.

Для запуску пошуку маршруту може використовуватися окрема функція, яка приймає граф, початкову та кінцеву вершини:

```
fun findShortestRoute(  
    graph: Map<String, List<GraphEdge>>,  
    start: String,  
    finish: String  
) : RouteResult {  
    // пошук найкоротшого маршруту за алгоритмом Дейкстри  
    return RouteResult(path = emptyList(), totalWeight = 0.0)  
}
```

Програмну реалізацію алгоритму пошуку найкоротшого маршруту мовою Kotlin наведено в додатку А. Функція отримує графову модель, початкову та кінцеву вершини, після чого повертає результат побудови маршруту, всередині цієї функції виконується обробка суміжних вершин, порівняння ваг ребер і формування найкоротшого шляху.

Отже, для розроблюваної системи доцільно поєднати веб-технології та Kotlin. HTML, CSS і JavaScript забезпечують створення адаптивного користувацького інтерфейсу, доступного з різних пристроїв. Структуровані дані та графова модель забезпечують подання трас і підйомників у формі, придатній для алгоритмічної обробки. Kotlin застосовано для подання алгоритмічної частини системи, зокрема моделей вершин, ребер і маршруту, а також логіки пошуку найкоротшого шляху. Таке поєднання засобів дозволяє реалізувати мобільно-орієнтовану систему навігації та побудови маршрутів на гірськолижному курорті.

3 РЕАЛІЗАЦІЯ МОБІЛЬНО-ОРІЄНТОВАНОЇ СИСТЕМИ ПОБУДОВИ МАРШРУТІВ

3.1 Структура та функціональні можливості веб-інтерфейсу

Система призначена для допомоги користувачу під час орієнтування на гірськолижному курорті. Вона забезпечує побудову маршруту між вибраними точками, а також містить додаткові розрахункові сервіси. Інтерфейс системи реалізовано у вигляді веб-сторінок, тому його можна відкривати у браузері як на персональному комп'ютері, так і на мобільному пристрої.

Для створення користувацького інтерфейсу використано веб-технології HTML, CSS та JavaScript. HTML застосовано для формування структури сторінок, CSS — для оформлення зовнішнього вигляду інтерфейсу, а JavaScript — для обробки дій користувача та взаємодії з елементами графової моделі. Такий підхід дозволяє розділити структуру, оформлення та логіку роботи сторінки.

Головна сторінка системи виконує навігаційну функцію та забезпечує перехід до основних розділів. У верхній частині сторінки розміщено меню, яке містить такі пункти: “Пошук маршруту”, “Розрахунок витрат пального” та “Розрахунок вартості катання”. Така структура дає змогу користувачу швидко перейти до потрібної функції системи.

Візуальне представлення головної сторінки веб-інтерфейсу системи наведено на рисунку 3.1.

Для перевірки адаптивності інтерфейсу головну сторінку системи було відкрито на мобільному пристрої. Вигляд сторінки на екрані смартфона наведено на рисунку 3.2.

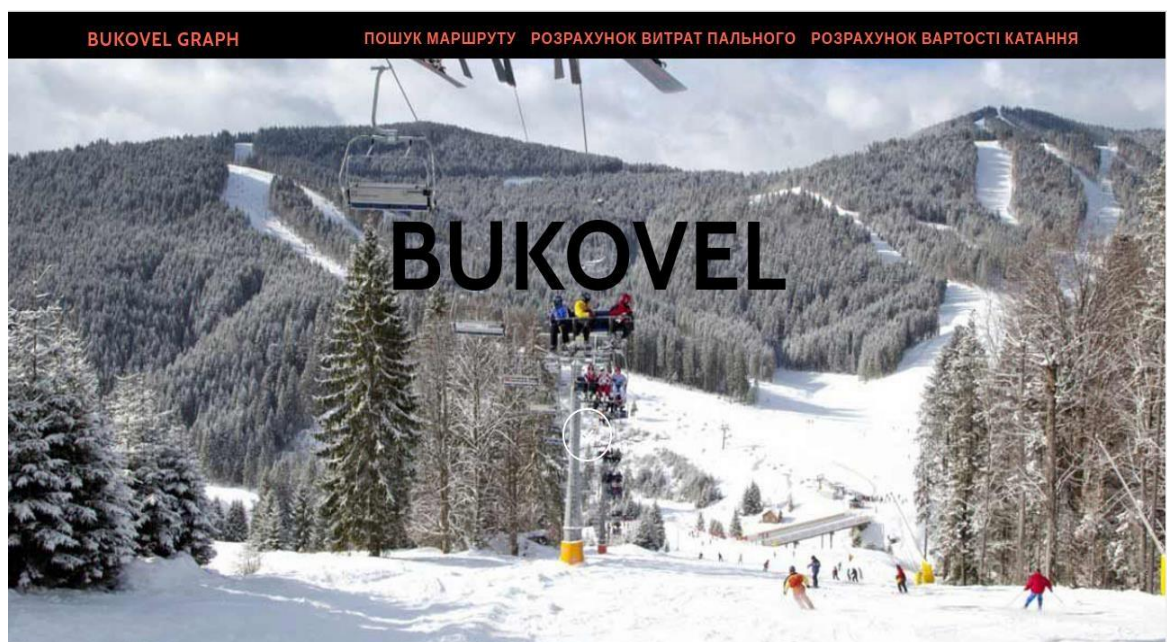


Рисунок 3.1 – Головна сторінка веб-інтерфейсу системи

Як видно з рисунку 3.1, головна сторінка має просту структуру та містить основні навігаційні елементи. Центральна частина сторінки використовується для візуального представлення тематики системи, а верхнє меню забезпечує доступ до її функціональних можливостей. Основним розділом у межах даної роботи є “Пошук маршруту”, оскільки саме він пов’язаний із побудовою маршруту між об’єктами гірськолижного курорту.

Розділ “Пошук маршруту” призначений для вибору початкової та кінцевої точки переміщення. Користувач задає потрібні вершини маршруту, після чого система виконує пошук найкоротшого шляху на основі графової моделі курорту. У графовій моделі вершинами виступають підйомники або ключові точки маршруту, а ребрами — гірськолижні траси, підйомники та переходи між ними.

Для візуального представлення графової структури курорту використано JavaScript-бібліотеку Cytoscape.js. Вона дає змогу відображати вершини та ребра графа, задавати їх зовнішній вигляд, а також виділяти окремі елементи під час побудови маршруту. У межах системи вершини відповідають підйомникам або ключовим точкам маршруту, а ребра — трасам, підйомникам чи переходам між ними.

У початковому стані система відображає графову структуру курорту, у якій представлені основні підйомники та траси. Після вибору початкової та кінцевої точки система виконує побудову маршруту й виділяє знайдений шлях. Інші елементи схеми можуть відображатися менш помітно, що дає змогу користувачу зосередитися саме на рекомендованому маршруті.

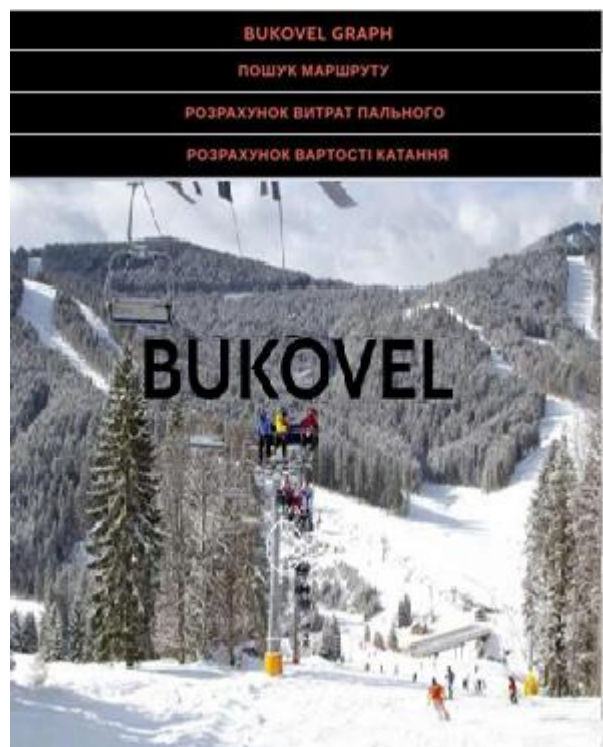


Рисунок 3.2 – Відображення головної сторінки системи на мобільному пристрої

Як видно з рисунка 3.2, на мобільному пристрої елементи верхнього меню розміщуються послідовно один під одним, що забезпечує зручний доступ до основних розділів системи. Основне зображення сторінки також адаптується до ширини екрана, тому користувач може переглядати інтерфейс без необхідності встановлення окремого застосунку.

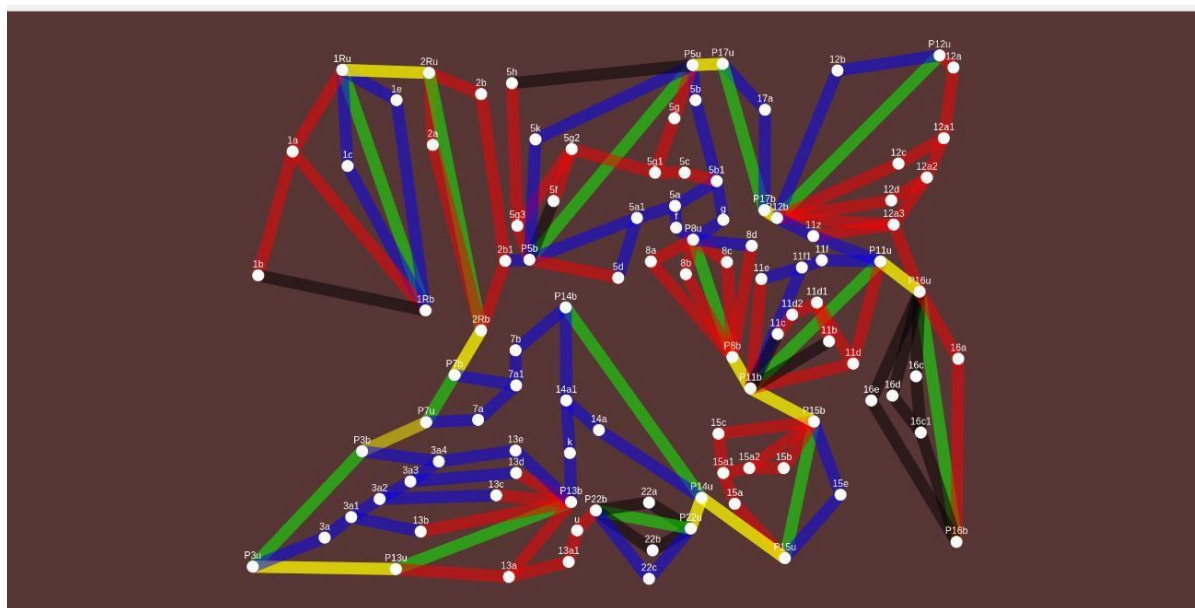
Для забезпечення безпеки туристів та швидкого візуального сприйняття інформації, всі елементи графової моделі (траси) розфарбовані у яскраві кольори відповідно до міжнародної класифікації гірськолижних курортів:

- синій колір — пологі схили для початківців;
- червоний колір — траси середньої складності для любителів;

- чорний колір — стрімкі та складні ділянки для професіоналів;
- зелений колір — лінійні траскторії підйомників.

Візуальний відгук на дії користувача: при виборі початкової чи кінцевої точок маршруту система динамічно підсвічує відповідні об'єкти, а після виконання розрахунку алгоритмом Дейкстри — яскраво виділяє знайдений оптимальний шлях, роблячи всі інші (нерелевантні) ділянки напівпрозорими.

Вигляд сторінки пошуку маршруту наведено на рисунку 3.2.



Розділ “Розрахунок вартості катання” призначений для попереднього оцінювання витрат на користування послугами курорту. Він може використовуватися для порівняння варіантів купівлі ski-pass залежно від тривалості катання та інших умов. Такий сервіс є корисним для користувача, оскільки дозволяє заздалегідь оцінити можливі витрати під час перебування на курорті.

Таким чином, веб-інтерфейс системи об’єднує основну функцію побудови маршруту та допоміжні інформаційно-розрахункові можливості. Основна увага в роботі приділяється функції пошуку маршруту, однак наявність додаткових розділів підвищує практичну цінність системи для користувача. Завдяки веб-орієнтованій структурі інтерфейс може використовуватися на різних пристроях, що відповідає мобільно-орієнтованому характеру розроблюваної системи.

3.2 Програмне подання графа та побудова маршруту

Для реалізації функції пошуку маршруту інфраструктуру гірськолижного курорту подано у вигляді графової моделі. Такий підхід є зручним, оскільки траси, підйомники та переходи між ними можна описати як набір взаємопов’язаних об’єктів. У межах системи вершинами графа виступають підйомники, станції підйомників або ключові точки маршруту, а ребрами — гірськолижні траси, підйомники та переходи між окремими вершинами.

За основу побудови маршруту обрано алгоритм Дейкстри. У даній системі він використовується для визначення найкоротшого шляху між початковою та кінцевою вершинами, які задає користувач. Такий підхід дозволяє не просто відобразити схему курорту, а автоматично визначити послідовність ділянок, якими користувач може дістатися до вибраної точки.

Графова модель курорту є орієнтованою, оскільки напрям руху між окремими об’єктами має значення. Наприклад, рух підйомником здійснюється від нижньої станції до верхньої, а рух трасою — переважно від верхньої точки до

нижньої. Тому під час побудови маршруту враховується не лише наявність зв'язку між вершинами, а й напрям цього зв'язку.

Граф гірськолижних маршрутів туристичного комплексу “Буковель”, який використовується для програмного подання маршрутів, наведено на рисунку 3.4.

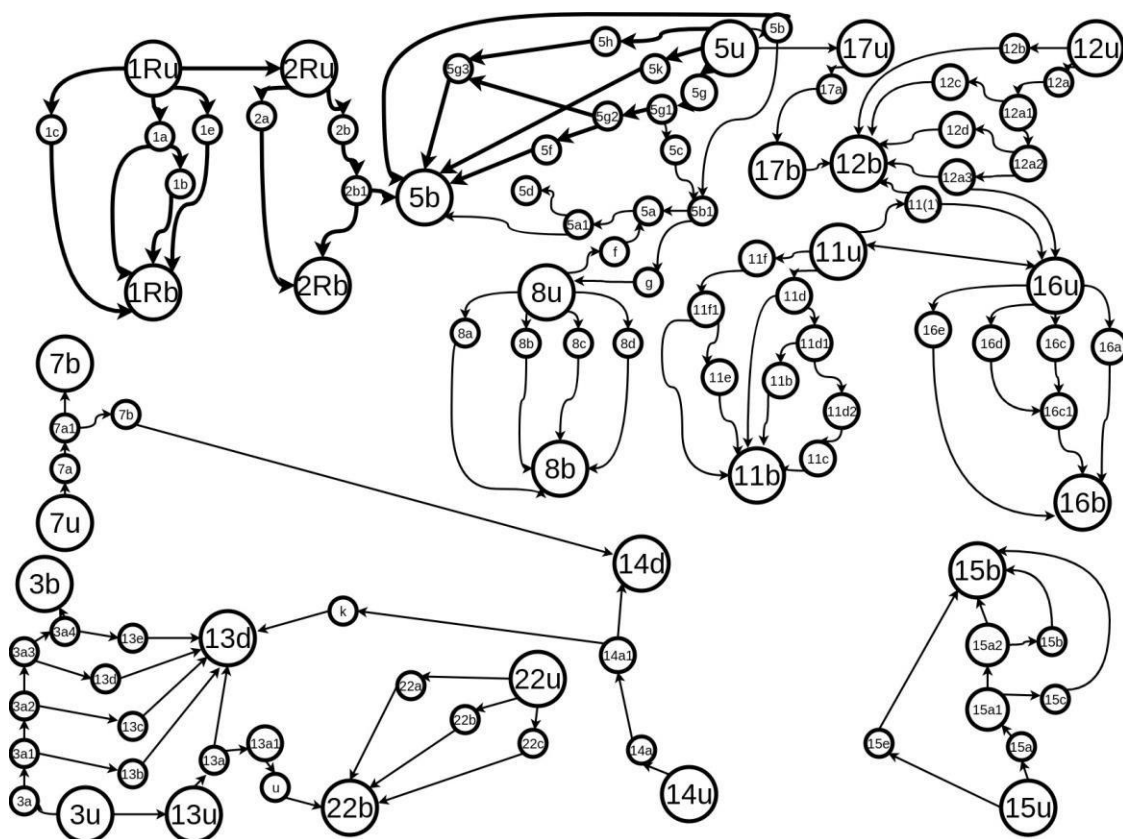


Рисунок 3.4 – Граф гірськолижних маршрутів туристичного комплексу “Буковель”

Як видно з рисунка 3.3, графова структура дозволяє подати підйомники, траси та переходи між ними як єдину систему взаємопов'язаних елементів. Вершини графа відповідають основним точкам маршруту, а ребра визначають можливі напрямки переміщення користувача. Саме така структура використовується для подальшої побудови маршруту.

Для передавання даних, необхідних для побудови графа, використовується формат JSON. Він є зручним для веб-реалізації, оскільки дозволяє зберігати інформацію у вигляді набору об'єктів. У JSON-структурі окремо описуються вершини графа та ребра між ними.

Приклад подання вершини графа наведено нижче:

```
{
  "data": {
    "id": "1Rb",
    "station_name": "1Rb",
    "SUID": "1Rb"
  },
  "position": {
    "x": 308,
    "y": 438
  },
  "selected": false
}
```

У наведеному фрагменті об'єкт data містить інформацію про вершину графа. Такою вершиною може бути нижня або верхня станція підйомника, початок траси або інша ключова точка маршруту. Поле position задає координати розміщення вершини на графічній схемі, що використовується для візуального відображення графа у веб-інтерфейсі.

Приклад подання ребра графа наведено нижче:

```
{
  "data": {
    "id": "1",
    "source": "1Rb",
    "target": "1Ru",
    "company_type": 0,
    "size": 0,
    "time": 1
  },
  "selected": false
}
```

У цьому фрагменті поле source визначає початкову вершину ребра, а поле target — кінцеву вершину. Поле size використовується для збереження довжини відповідної ділянки, а поле time — для збереження часу, необхідного для її проходження. Таким чином, одне й те саме ребро може використовуватися для пошуку маршруту за різними критеріями: за довжиною або за часом.

Крім довжини ділянки, для окремих трас може враховуватися орієнтовний час спуску. Для цього було проаналізовано дані проходження трас кількома лижниками та визначено середній час спуску для відповідних ділянок. Такі дані можуть використовуватися як додаткова вагова характеристика ребра графа.

У базовому режимі побудови маршруту основним критерієм є довжина ділянки, що відповідає параметру size. Водночас дані про середній час спуску

можуть використовуватися для параметра `time` у графовій моделі. Це дає змогу за потреби порівнювати маршрути не лише за довжиною, а й за орієнтовним часом проходження. Таблиці з експериментальними значеннями часу спуску для окремих трас подано в додатку Б.

Для побудови графа та його відображення у веб-інтерфейсі використовується бібліотека `Cytoscape.js`. Вона дає змогу створювати графові структури, задавати вершини й ребра, відображати їх на сторінці та змінювати вигляд елементів залежно від дій користувача. У системі також використовується можливість виконання алгоритму Дейкстри для елементів графа.

Фрагмент запуску алгоритму Дейкстри у веб-частині системи має такий вигляд:

```
var dijkstra = cy.elements().dijkstra({
  root: start,
  directed: true,
  weight: function(e) {
    if (measure === "size") {
      return e.data("size");
    }
    if (measure === "time") {
      return e.data("time");
    }
  }
});
```

У наведеному фрагменті параметр `root` задає початкову вершину маршруту. Параметр `directed: true` вказує, що граф є орієнтованим і під час пошуку потрібно враховувати напрям руху між вершинами. Функція `weight` визначає, яке значення буде використано як вага ребра. Якщо користувач обирає пошук за довжиною, використовується поле `size`; якщо пошук виконується за часом, використовується поле `time`.

Після виконання алгоритму можна отримати відстань до вибраної вершини та сам шлях. Для цього використовуються функції `distanceTo(node)` і `pathTo(node)`. Перша повертає значення відстані від початкової вершини до заданої, а друга — послідовність елементів графа, які утворюють знайдений маршрут.

Таблиця 3.1 – Матриця суміжності графової моделі курорту

У таблиці 3.1 рядки та стовпці відповідають вершинам графа, а значення в комірках показують наявність або відсутність зв'язку між ними. Якщо в комірці записано 1, це означає, що між відповідними вершинами існує зв'язок. Якщо записано 0, прямого переходу між вершинами немає. У випадку використання вагової матриці замість одиниці може задаватися довжина ділянки або час її проходження.

50

графа, ребер графа та результату побудови маршруту. Веб-інтерфейс використовується для вибору початкової та кінцевої точки, а алгоритмічна частина відповідає за обробку графової моделі та пошук найкоротшого шляху. Повну програмну реалізацію алгоритму Дейкстри мовою Kotlin наведено в додатку А. У ній описано класи для подання вершин і ребер графа, а також функцію пошуку найкоротшого маршруту. Це дозволяє показати реалізацію основної логіки побудови маршруту засобами Kotlin і водночас зберегти адаптивний веб-інтерфейс як зручний спосіб взаємодії користувача із системою.

Отже, програмне подання графа в системі базується на використанні JSON-структури, матриці суміжності та бібліотеки Cytoscape.js для роботи з графом у веб-інтерфейсі. Вершини графа відповідають підйомникам або ключовим точкам маршруту, а ребра — трасам, підйомникам і переходам між ними. Алгоритм Дейкстри забезпечує пошук найкоротшого маршруту між вибраними користувачем вершинами з урахуванням довжини або часу проходження ділянок.

3.3 Опис роботи основних і додаткових модулів

Основною функцією системи є пошук маршруту між вибраними користувачем точками гірськолижного курорту. Допоміжними функціями є розрахунок витрат пального та розрахунок вартості катання.

Робота з основною функцією системи починається з переходу користувача до розділу “Пошук маршруту”. У цьому розділі користувач має можливість вибрати початкову та кінцеву точки маршруту. Такими точками можуть бути вершини графової моделі, які відповідають підйомникам, станціям або іншим ключовим об’єктам курорту. Після вибору точок користувач запускає побудову маршруту.

Взаємодія з інтерактивною картою-схемою залежить від використовуваної апаратної платформи: на персональних комп'ютерах вибір об'єктів виконується за допомогою кліку миші, а на мобільних пристроях — за допомогою дотику (тапу) в зоні сенсорного дисплея. Після активації потрібної вершини на екрані з'являється керуюча кнопка „ПОЧАТОК“, натискання на яку ініціалізує початкову точку для алгоритму Дейкстри. Вигляд сторінки вибору початкової точки маршруту наведено на рисунку 3.4.

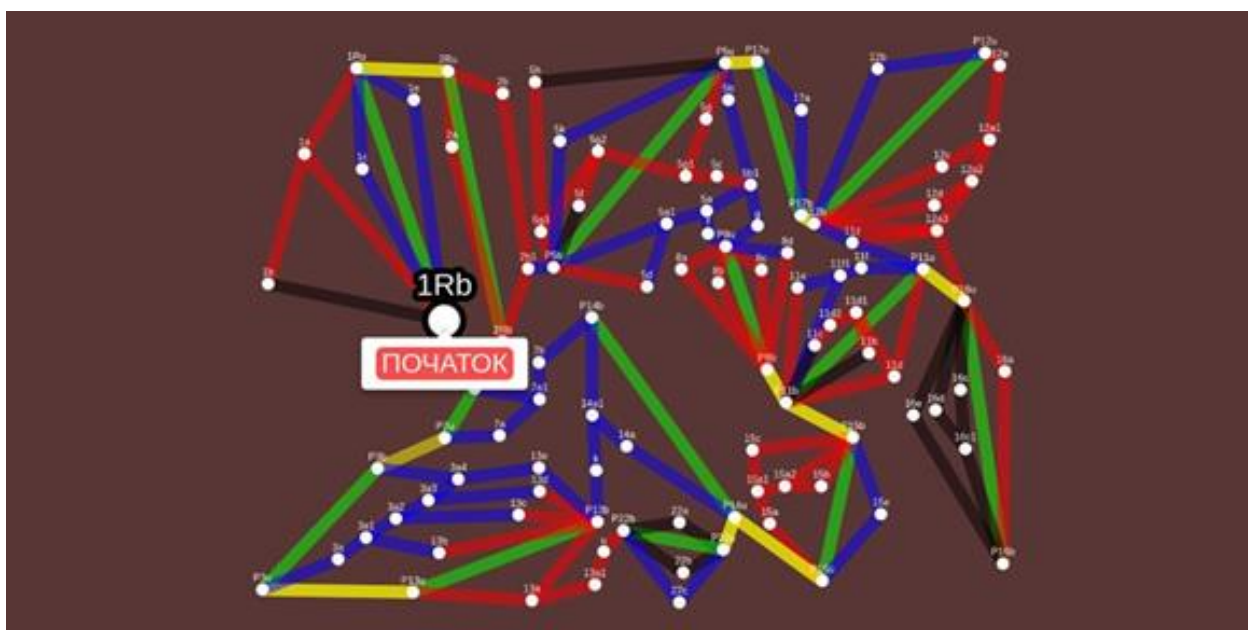


Рисунок 3.4 – Вибір початкової точки маршруту

Для вибору кінцевої вершини (рис.3.5) необхідно провести ту саму операцію як і для вибору початкової вершини. Зміна кольору маркерів на червоний є візуальним підтвердженням того, що ці вершини графа задані як опорні точки для обчислювального алгоритму.

Після введення початкової та кінцевої вершини система передає ці дані до модуля побудови маршруту. Далі виконується пошук найкоротшого шляху на основі графової моделі. Під час побудови маршруту враховуються зв'язки між вершинами графа, напрям руху та вагові характеристики ребер, зокрема довжина або час проходження відповідної ділянки.

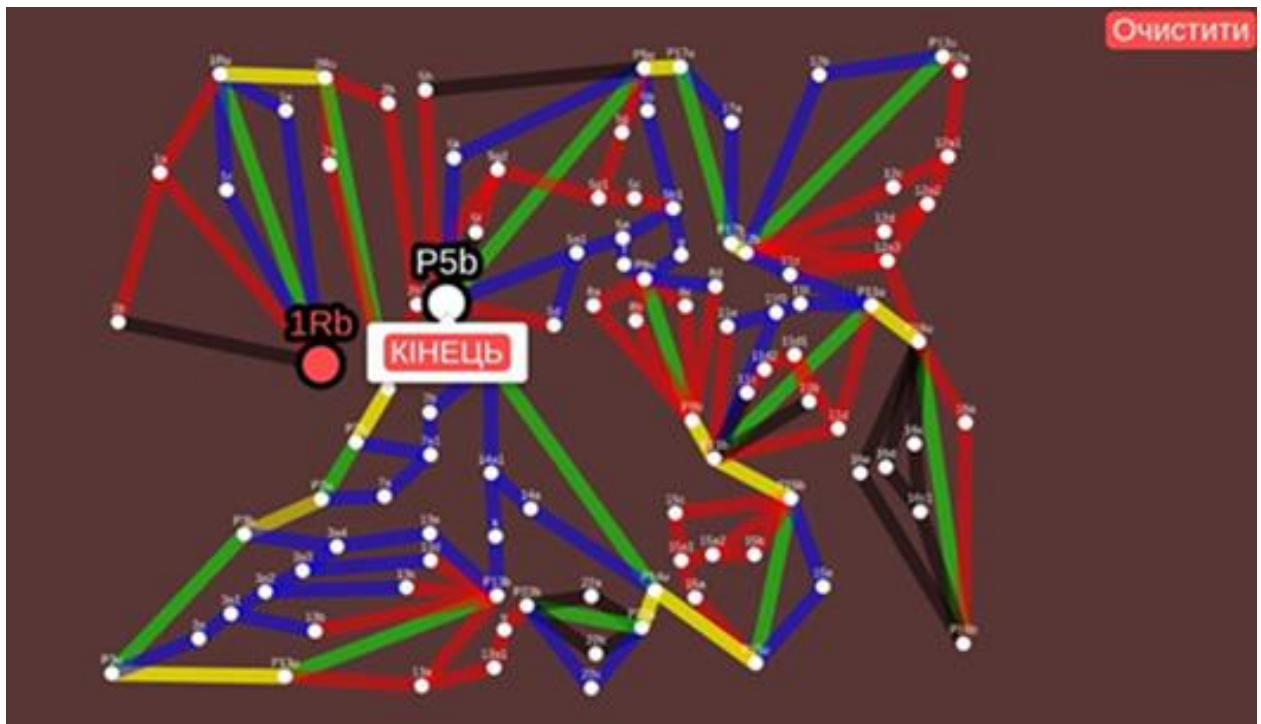


Рисунок 3.5 – Приклад вибору кінцевої вершини

Результатом роботи функції пошуку маршруту є послідовність трас, підйомників або переходів, якими користувач може дістатися від початкової точки до кінцевої. Знайдений маршрут може бути поданий у текстовому вигляді або виділений на графовій схемі курорту. Такий підхід дозволяє користувачу швидко зорієнтуватися в запропонованому шляху та не аналізувати всю схему маршрутів самостійно.

Приклад відображення результату побудови маршруту наведено на рисунку 3.6.

Як видно з рисунка 3.6, після виконання пошуку система відображає знайдений маршрут у зручному для користувача вигляді. Якщо маршрут виділяється на графі, інші елементи схеми можуть відображатися менш помітно. Це дозволяє зосередити увагу користувача саме на тих трасах, підйомниках або переходах, які входять до побудованого маршруту.

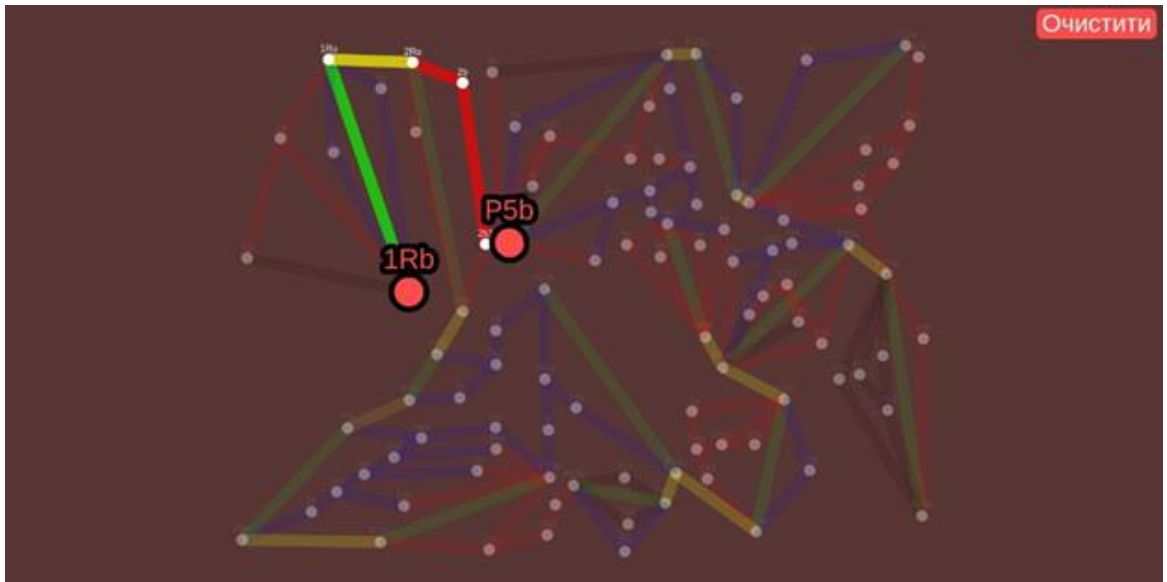


Рисунок 3.6 – Відображення результату побудови маршруту

Крім основного модуля пошуку маршруту, у системі передбачено додаткові розрахункові модулі.

Одним із таких модулів є модуль розрахунку вартості катання. Він призначений для попереднього оцінювання витрат на придбання ski-pass. Користувач може вказати кількість осіб, тривалість катання або вибраний тариф, після чого система обчислює орієнтовну загальну вартість.

Розрахунок підсумкової вартості C_{pass} здійснюється за формулою:

$$C_{pass} = N \cdot P_{day} \cdot (1 - D_{season}) \cdot (1 - D_{age})$$

де:

- N — загальна кількість запланованих днів катання;
- P_{day} — базова вартість одноденного абонементу за стандартним тарифом курорту;
- D_{season} — коефіцієнт сезонної знижки або надбавки (наприклад, тарифи «Низький сезон», «Високий сезон» або «Святковий»);
- D_{age} — коефіцієнт вікової пільги користувача (дитячий, дорослий, VIP або пенсійний квиток).

Програма аналізує введені користувачем параметри за допомогою умовних операторів та пропонує оптимальний варіант купівлі (наприклад, порівнює купівлю кількох окремих одноденних абонементів із придбанням єдиного багаторазового квитка на відповідну кількість днів).

Ще одним додатковим модулем є модуль розрахунку витрат пального. Його призначення полягає в орієнтовному визначенні витрат користувача на пальне під час поїздки до гірськолижного курорту або автомобільного переміщення між його віддаленими об'єктами чи зонами. Користувач може ввести відстань маршруту, середню витрату пального автомобіля та ціну одного літра пального, після чого система обчислює орієнтовну вартість поїздки.

Для розрахунку витрат пального використовується відстань між обраними об'єктами, середня витрата пального автомобіля та ціна одного літра пального. Орієнтовний обсяг пального визначається за формулою:

$$V = \frac{S \cdot R}{100}$$

де

S — відстань автомобільного маршруту між обраними локаціями чи трасами, км;

R — показник середньої витрати пального автомобіля користувача (л/100 км).

Вартість пального визначається за формулою:

$$C = V P$$

де

P — ціна одного літра пального, грн/л.

Також система може визначати орієнтовний запас ходу автомобіля за наявним обсягом пального L , тобто оцінка того, наскільки кілометрів локальних переїздів вистачить наявного в баку палива:

$$L = \frac{V_{\text{зал}} \cdot 100}{R}$$

де

$V_{\text{зал}}$ — поточний об'єм палива, який залишився в автомобільному баку користувача на момент розрахунку.

Інтеграція додаткових розрахункових модулів у веб-інтерфейс системи дає змогу користувачу не лише будувати маршрут між об'єктами гірськолижного курорту, а й орієнтовно оцінювати супутні витрати. Отже, система поєднує основну функцію побудови маршруту з додатковими сервісами, що підвищує її практичну цінність для користувача.

ВИСНОВКИ

У бакалаврській роботі розроблено мобільно-орієнтовану систему навігації та побудови маршрутів на гірськолижному курорті. Розроблена система призначена для спрощення орієнтування користувача між трасами, підйомниками, станціями та іншими ключовими об'єктами гірськолижної інфраструктури.

Під час виконання роботи було проаналізовано особливості навігації на гірськолижних курортах. Встановлено, що така навігація відрізняється від звичайної пішохідної або автомобільної навігації, оскільки потребує врахування напрямку руху трасами, роботи підйомників, складності окремих ділянок і можливості переходу між різними об'єктами курорту. Це обґрунтовує доцільність використання графової моделі для подання структури трас і підйомників.

Для формалізації задачі побудови маршруту інфраструктуру гірськолижного курорту подано у вигляді орієнтованого зваженого графа. Вершинами графа є підйомники, станції та ключові точки маршруту, а ребрами — траси, підйомники або переходи між ними. Вагами ребер можуть виступати довжина ділянки, орієнтовний час її проходження або інші характеристики, які впливають на вибір маршруту.

Для пошуку маршруту обрано алгоритм Дейкстри. Його використання є доцільним, оскільки він дозволяє знаходити найкоротший шлях у зваженому графі з невід'ємними вагами ребер. У розробленій системі алгоритм застосовується для визначення маршруту між початковою та кінцевою точками, які задає користувач.

У роботі визначено інформаційну структуру основних об'єктів системи: трас, підйомників, вершин графа, ребер графа та маршруту. Для подання графової моделі використано структуровані дані, зокрема опис вершин і ребер у

форматі JSON. Таке подання є зручним для програмної обробки та відображення графа у веб-інтерфейсі.

Користувацький інтерфейс системи реалізовано як адаптивний веб-інтерфейс. Це дає змогу використовувати систему не лише на персональному комп'ютері, а й на смартфоні або планшеті через браузер. Для створення інтерфейсної частини використано HTML, CSS і JavaScript, а для відображення графової структури курорту — бібліотеку Cytoscape.js.

Для опису алгоритмічної частини побудови маршруту використано мову Kotlin. У програмній реалізації описано структури даних для вершин, ребер і результату побудови маршруту, а також функцію пошуку найкоротшого шляху. Це дозволяє відокремити логіку пошуку маршруту від інтерфейсної частини системи.

Розроблена система забезпечує вибір початкової та кінцевої точки маршруту, запуск побудови шляху та відображення результату користувачу. Побудований маршрут може бути представлений у вигляді послідовності трас, підйомників і переходів, що спрощує орієнтування користувача на території гірськолижного курорту.

Крім основної функції побудови маршруту, у системі передбачено додаткові розрахункові сервіси. До них належать розрахунок вартості катання та розрахунок витрат пального. Розрахунок вартості катання дає змогу попередньо оцінити витрати на ski-pass, а розрахунок витрат пального може використовуватися для планування поїздки до курорту або автомобільного переміщення між його віддаленими зонами. Ці сервіси не впливають на алгоритм побудови маршруту, але підвищують практичну цінність системи для користувача.

ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. International Ski and Snowboard Federation. FIS Code of Conduct for Skiers and Snowboarders. URL: <https://assets.fis-ski.com/f/252177/49280306c8/fis-code-of-conduct-for-skiers-and-snowboarders-en.pdf> (дата звернення: 06.02.2026).
2. Буковель // Вікіпедія : вільна енциклопедія. URL: <https://uk.wikipedia.org/wiki/Буковель> (дата звернення: 06.02.2026).
3. Статус витягів і трас // Офіційний сайт курорту Bukovel. URL: <https://bukovel.com/status> (дата звернення: 06.02.2026).
4. НПАОП 60.2-1.02-14. Правила будови і безпечної експлуатації пасажирських підвісних канатних доріг : наказ Міністерства енергетики та вугільної промисловості України від 14.01.2014 № 23, зареєстровано в Міністерстві юстиції України 07.02.2014 за № 264/25041. URL: Офіційний вебпортал Верховної Ради України (дата звернення: 08.02.2026).
5. Google Maps. Bukovel. URL: <https://www.google.com/maps/search/Bukovel/> (дата звернення: 15.02.2026).
6. Пошук в ширину // Algo.ua. URL: <https://algo.ua/algorithms/graphs/bfs/> (дата звернення: 22.03.2026).
7. Задача про найкоротший шлях — алгоритм Дейкстри // Mathros. URL: <https://www.mathros.net.ua/znahodzhennja-najkorotshogo-shljahu-v-orijentovanomu-grafi-za-algorytmom-dejkstry.html> (дата звернення: 22.03.2026).
8. Алгоритм Беллмана — Форда // Вікіпедія : вільна енциклопедія. URL: https://uk.wikipedia.org/wiki/Алгоритм_Беллмана_—_Форда. (дата звернення: 22.03.2026).
9. Алгоритм пошуку A^* // Вікіпедія : вільна енциклопедія. URL: https://uk.wikipedia.org/wiki/Алгоритм_пошуку_A^*. (дата звернення: 26.03.2026).

10. M. Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition. PrintSprint. 2004. 183 p.
11. Кузьменко І. М. Теорія графів : навч. посіб. для здобувачів ступеня бакалавра. Київ : КПІ ім. Ігоря Сікорського, 2020. 71 с.
12. Крєневич А. Алгоритми і структури даних. Підручник. К.: ВПЦ "Київський Університет", 2021. 200 с.
13. Guru99. Kotlin Tutorial for Beginners: Learn Kotlin Programming Language [Електронний ресурс]. URL: <https://www.guru99.com/uk/kotlin-tutorial.html> . (дата звернення: 06.04.2026).
14. Problem Solving with Algorithms and Data Structures using Python. URL: <https://runestone.academy/runestone/books/published/pythonds/index.html>. (дата звернення: 06.04.2026).
15. Вихідний код програм. URL: <https://github.com/krenevych/algo>. (дата звернення: 06.04.2026).
16. Brander. Як адаптувати сайт під мобільні пристрої за 5 кроків? URL: <https://brander.ua/blog/yak-adaptuvaty-sayt-pid-mobilni-prystroyi-za-5-kroki> ((дата звернення: 08.04.2026).
17. MDN Web Docs. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 09.04.2026).
18. MDN Web Docs. Responsive web design. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Responsive_Design (дата звернення: 09.04.2026).
19. MDN Web Docs. JavaScript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 09.04.2026).
20. W3Schools. JavaScript HTML DOM Events [Електронний ресурс]. URL: https://www.w3schools.com/js/js_htmldom_events.asp (дата звернення: 09.04.2026).

21.web.dev. Responsive web design basics. URL:
<https://web.dev/articles/responsive-web-design-basics> (дата
09.04.2026).

