

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

ВЕБСЕРВІС ДЛЯ СОЦІАЛЬНОГО ТУРИЗМУ З ІНТЕРАКТИВНИМ
ПЛАНУВАННЯМ МАРШРУТІВ

WEB SERVICE FOR SOCIAL TOURISM WITH INTERACTIVE ROUTE
PLANNING

спеціальність 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»

Виконав: здобувач вищої освіти
групи КН-41
Мирончук Андрій Ярославович

(підпис)

Керівник: к.т.н., доцент
Ліщина Валерій Олександрович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«2» червня 2026 р.
Гарант ОП: д.п.н., професор
Тулашвілі Юрій Йосипович

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерних наук

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 122 Комп'ютерні науки

Освітня програма: «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Валерій ЛІЩИНА

«16» січня 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ПЕРШОГО (БАКАЛАВРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Мирончук Андрій Ярославович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Вебсервіс для соціального туризму з інтерактивним плануванням маршрутів.

Керівник: к.т.н., доцент Ліщина В.О.

затверджено наказом закладу вищої освіти від «16» січня 2026 р. № 32/01-02

2. Строк подання кваліфікаційної роботи «02» червня 2026 р.

3. Вихідні дані до роботи: аналітичне дослідження систем управління проектами.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Вступ

Розділ 1. Аналіз предметної області

Розділ 2. Структурна реалізація об'єкта проектування

Розділ 3. Програмна реалізація об'єкта проектування

Розділ 4. Спеціальні розрахунки

Висновки

Додатки

5. Перелік графічного матеріалу: _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Ліщина В. О.</i>		
<i>Теоретичне дослідження</i>	<i>Ліщина В. О.</i>		
<i>Практична реалізація</i>	<i>Ліщина В. О.</i>		
<i>Спеціальні розрахунки</i>	<i>Ліщина В. О.</i>		
<i>Показник запозичень тексту</i>	%		
<i>Інструментальна перевірка</i>	<i>Кошелюк В. А.</i>		
<i>Нормоконтроль</i>	<i>Сачук В. О.</i>		
<i>Гарант ОПП</i>	<i>Тулашвілі Ю.Й.</i>		

7. Дата видачі завдання «16» січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1.	<i>Провести огляд літературних джерел по темі кваліфікаційної роботи</i>	<i>до 10.02.2026р</i>	
2.	<i>Провести аналіз загальної проблеми і вибір напрямків дослідження</i>	<i>до 24.02.2026р.</i>	
3.	<i>Розробити функціональну схему роботи програмного продукту</i>	<i>до 10.03.2026р</i>	
4.	<i>Описати засоби розробки об'єкта проектування</i>	<i>до 31.03.2026р.</i>	
5.	<i>Практична реалізація об'єкта проектування</i>	<i>до 05.05.2026р.</i>	
7.	<i>Провести спеціальні розрахунки</i>	<i>до 21.05.2026р.</i>	
8.	<i>Здача чистового варіанту кваліфікаційної роботи на кафедру</i>	<i>до 02.06.2026р.</i>	

Здобувач вищої освіти _____ Андрій МИРОНЧУК

Керівник роботи _____ Валерій ЛІЩИНА

АНОТАЦІЯ

Мирончук А. Я. Вебсервіс для соціального туризму з інтерактивним плануванням маршрутів. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерні науки». Луцький національний технічний університет. Луцьк, 2026.

У роботі розглянуто програмне рішення вебсервіс соціального туризму з інтерактивним плануванням маршрутів. Рішення призначене для автоматизації процесів, пов'язаних із добором туристичних об'єктів, побудовою маршрутів і врахуванням доступності.

Ключові слова: соціальний туризм, маршрут, доступність, вебсервіс, планування.

ABSTRACT

Andrii Myronchuk. Web service for social tourism with interactive route planning. Manuscript.

Bachelor's qualifying thesis of the OP "Computer Sciences". Lutsk National Technical University. Lutsk, 2026.

The thesis considers a web service for social tourism with interactive route planning intended to automate processes related to planning accessible routes, tourist objects, budget, and group interaction. The relevance of the topic is substantiated, and the main system requirements, architecture, data model, and user scenarios are defined. The implemented functionality includes map, routes, accessibility filters, budget, reviews, saving, and sharing plans. The main scenarios were tested, and an approach to deployment was prepared. The practical value of the work lies in the ability to make social tourism more accessible and personalised.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ЗАДАЧІ ДОСЛІДЖЕННЯ	9
1.1 Загальна характеристика предметної області.....	9
1.2 Аналіз сучасних підходів та аналогів.....	10
1.3 Обґрунтування вимог до системи.....	11
1.4 Постановка завдання дослідження	12
РОЗДІЛ 2 ПРОЕКТУВАННЯ РОЗРОБКИ.....	14
2.1 Архітектурне проектування.....	14
2.2 Проектування функціональних сценаріїв і структури даних	15
2.3 Проектування інтерфейсу	17
РОЗДІЛ 3 СТВОРЕННЯ СМАРТ КОНТРАКТУ	19
3.1 Середовище розробки та структура програмного проєкту	19
3.2 Реалізація серверної логіки та моделі даних	21
3.3 Реалізація клієнтського інтерфейсу застосунку	26
РОЗДІЛ 4 ТЕСТУВАННЯ І РОЗГОРТАННЯ ПРОЕКТУ	30
4.1 План тестування	30
4.2 Результати функціонального тестування	32
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
ДОДАТКИ.....	38

ВСТУП

Актуальність теми кваліфікаційної роботи зумовлена зростанням ролі інформаційних технологій у вирішенні прикладних завдань, пов'язаних із вебсервісом для соціального туризму з інтерактивним плануванням маршрутів. Сучасні програмні системи мають не лише автоматизувати окремі операції, а й забезпечувати цілісну підтримку користувача від введення початкових даних до отримання зрозумілого результату, контролю виконання дій та подальшого використання накопиченої інформації. У таких умовах особливого значення набувають веборієнтовані рішення, які доступні з різних пристроїв, легко масштабуються та можуть бути інтегровані з іншими сервісами.

Практична потреба у розробленні такої системи пояснюється тим, що соціальний туризм спрямований на розширення доступу до подорожей для різних категорій населення, однак ефективне планування потребує врахування доступності, бюджету, інтересів і логістики. За відсутності спеціалізованого програмного інструмента значна частина роботи виконується вручну або за допомогою універсальних сервісів, які не враховують специфіку предметної області. Це призводить до втрати часу, дублювання інформації, складності контролю актуального стану даних і зниження якості прийняття рішень. Тому створення цільового застосунку є обґрунтованим як з погляду оптимізації користувацьких процесів, так і з погляду набуття практичних навичок проєктування сучасних програмних продуктів.

Метою кваліфікаційної роботи є розроблення вебсервісу, що підтримує планування маршрутів соціального туризму, обмін досвідом між користувачами та ергономічне подання туристичної інформації. Досягнення цієї мети передбачає не лише реалізацію окремих функцій, а й обґрунтування архітектури системи, визначення ролей користувачів, опис інформаційних потоків, вибір технологічного стеку та перевірку працездатності створеного рішення на типових сценаріях використання.

Об'єктом дослідження є процеси планування туристичних маршрутів з урахуванням доступності, соціальної взаємодії та потреб різних груп користувачів. У межах роботи цей об'єкт розглядається як сукупність взаємопов'язаних дій користувача, інформаційних сутностей, правил обробки даних і результатів, які повинні бути відображені у програмній системі.

Предметом дослідження є методи інтерактивного маршрутного планування, подання туристичних об'єктів і програмні засоби соціального вебсервісу. Предметна область потребує врахування функціональних вимог, обмежень безпеки та ергономічності користування, способів зберігання даних, логіки взаємодії між клієнтською та серверною частинами, а також засобів перевірки правильності роботи основних модулів. Саме ці складові визначають якість майбутнього програмного продукту та його придатність до практичного застосування.

Для досягнення поставленої мети необхідно розв'язати такі основні завдання: проаналізувати сучасний стан предметної області та наявні підходи до вирішення подібних задач; визначити потреби цільових користувачів і сформулювати перелік функціональних вимог; розабезпечити структуру даних і логіку взаємодії компонентів; спроектувати користувацький інтерфейс; реалізувати основні програмні модулі; провести перевірку роботи системи на типових сценаріях; сформулювати висновки щодо ефективності запропонованого рішення.

Функціональне наповнення системи містить інтерактивну карту, створення маршруту, опис туристичних об'єктів, фільтрацію за потребами користувача, спільне планування, відгуки та збереження маршрутів. Зазначені функції повинні працювати узгоджено, забезпечувати логічний шлях користувача та створювати основу для подальшого розвитку програмного продукту. Особливу увагу необхідно приділити простоті введення даних, зрозумілості результатів і зменшенню кількості надлишкових дій під час виконання основних операцій.

Методологічною основою роботи є методи аналізу туристичних потреб, геоінформаційного моделювання, проєктування вебсервісів, розробки інтерактивного інтерфейсу, організації даних про об'єкти та тестування користувацьких сценаріїв. Використання цих методів дає змогу послідовно пройти етапи від аналізу проблеми до створення працездатного програмного рішення. На етапі проєктування важливо визначити інформаційні сутності, зв'язки між ними, сценарії використання та обмеження системи. На етапі реалізації необхідно забезпечити коректну взаємодію компонентів, обробку помилкових ситуацій і достатній рівень надійності під час виконання основних операцій.

Практична цінність отриманого результату полягає в тому, що результат може бути корисним для туристичних організацій, громадських ініціатив, студентів і користувачів, які планують доступні та соціально орієнтовані подорожі. Запропоноване рішення має забезпечити зменшення рутинних дій, підвищення прозорості роботи з даними та створення більш зручного цифрового середовища для кінцевого користувача.

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Загальна характеристика предметної області

Актуальність теми соціального туризму з інтерактивним плануванням маршрутів пов'язана з потребою користувачів отримувати не лише перелік туристичних об'єктів, а й узгоджений маршрут з урахуванням доступності, бюджету, транспорту, інтересів групи та часових обмежень. Без цифрової підтримки такий процес часто виконується фрагментарно, інформація про локації зберігається в різних джерелах, параметри доступності перевіряються вручну, а групове погодження маршруту займає багато часу.

Соціальний туризм орієнтований на ширші групи користувачів, зокрема молодь, сім'ї, людей старшого віку, осіб з інвалідністю, волонтерські спільноти та організаторів локальних подорожей. Для таких користувачів важливими є зрозуміла навігація, доступні об'єкти, прозорий опис маршруту, можливість змінювати план і ділитися ним з іншими учасниками.

Предметна область охоплює туристичні об'єкти, категорії локацій, параметри доступності, географічні координати, транспортні відрізки, часові вікна, відгуки, профілі користувачів і збережені маршрути. Ці сутності взаємопов'язані. Зміна доступності об'єкта впливає на маршрут, бюджет обмежує добір локацій, а часові параметри визначають порядок відвідування.

Користувач вебсервісу очікує, що система допоможе не тільки знайти цікаві місця, а й перетворити їх на практичний план подорожі. Тому важливими стають функції фільтрації, побудови маршруту, оцінювання доступності, збереження результату, перегляду деталей об'єкта та формування підсумкового плану.

Окремим аспектом є соціальна складова планування. Маршрут може створюватися не для одного користувача, а для групи з різними інтересами та обмеженнями. У такому випадку система має підтримувати прозоре подання

критеріїв добору і дозволяти координатору швидко пояснити, чому певні об'єкти включені до маршруту.

Проблема дослідження полягає у створенні програмного засобу, який поєднує довідник туристичних об'єктів, карту, параметри доступності, фільтрацію та інтерактивне планування маршруту. Такий сервіс має зменшити кількість ручних операцій і підвищити якість підготовки соціальної подорожі.

Об'єктом дослідження є процес цифрової підтримки планування соціальних туристичних маршрутів. Предметом дослідження виступають програмні методи добору туристичних об'єктів, перевірки доступності, побудови маршруту, збереження плану і подання результатів у вебсервісі.

Метою роботи є створення вебсервісу, який забезпечує інтерактивне планування соціальних маршрутів на основі параметрів користувача, туристичних об'єктів і критеріїв доступності. Результатом має бути прототип, здатний сформулювати зрозумілий маршрут і надати користувачу пояснюваний підсумок.

1.2 Аналіз сучасних підходів та аналогів

Сучасні цифрові рішення для планування подорожей можна поділити на картографічні сервіси, туристичні платформи, довідники локацій, застосунки для трекінгу маршрутів і спеціалізовані сервіси доступного туризму. Кожна група розв'язує окрему частину задачі, але не завжди підтримує повний сценарій соціального маршруту.

Картографічні сервіси добре працюють із навігацією, геокодуванням, прокладанням шляху та оцінюванням часу руху. Їхнім обмеженням є те, що вони не завжди враховують соціальні параметри, потреби групи, доступність об'єктів, культурний контекст і можливість спільного узгодження плану.

Туристичні платформи надають описи локацій, фотографії, рейтинги, відгуки й рекомендації. Такі інструменти корисні для пошуку місць, однак

користувачу часто доводиться самостійно переносити вибрані об'єкти на карту, перевіряти транспортні обмеження і складати послідовність відвідування.

Сервіси для активного туризму, зокрема платформи з треками та маршрутами, зручні для піших або велосипедних подорожей. Вони орієнтовані на фізичний маршрут, але не завжди підтримують соціальні критерії, комфорт учасників, доступність інфраструктури, бюджетні обмеження і просте пояснення вибору об'єктів.

Для соціального туризму важливе поєднання карти, каталогу об'єктів, профілю групи, фільтрів доступності та механізму збереження плану. Саме така інтеграція відрізняє майбутній вебсервіс від звичайної карти або туристичного довідника.

Порівняння аналогів показує, що користувач потребує сервісу, який підтримує повний цикл планування, вибір інтересів, добір локацій, побудову маршруту, перевірку доступності, перегляд деталей і збереження готового плану. Ці висновки використовуються для формування вимог до майбутньої системи.

1.3 Обґрунтування вимог до системи

Функціональні вимоги до вебсервісу формуються на основі сценаріїв роботи туриста, координатора подорожі та адміністратора каталогу. Система має підтримувати перегляд туристичних об'єктів, фільтрацію за категоріями, перевірку параметрів доступності, побудову маршруту, редагування плану і збереження результату.

Профіль планування має містити інтереси користувача або групи, бажану тривалість маршруту, спосіб пересування, бюджетні межі, потреби доступності та пріоритети локацій. Ці параметри використовуються під час добору об'єктів і допомагають уникнути маршруту, який формально правильний, але незручний для реальних учасників.

Каталог туристичних об'єктів має зберігати назву, опис, категорію, координати, режим роботи, орієнтовну тривалість відвідування, ознаки

доступності, контактні дані, фотографії та службовий статус. Така модель дозволяє поєднати довідкові дані з алгоритмом побудови маршруту.

До нефункціональних вимог належать зрозумілий інтерфейс, швидке завантаження карти, коректна робота з неповними даними, захист персональної інформації, надійна обробка помилок і можливість подальшого розширення каталогу. Вебсервіс має бути придатним для повторного використання в навчальному або демонстраційному середовищі.

Особливе значення має пояснюваність результату. Користувач повинен бачити, які критерії вплинули на маршрут, чому певні об'єкти рекомендовані, які обмеження враховані і які кроки можна змінити вручну. Такий підхід підвищує довіру до системи.

Вимоги до якості також охоплюють доступність інтерфейсу. Для соціального туризму важливо передбачити контрастність, зрозумілі підписи, просту навігацію, повідомлення про помилки і можливість працювати з маршрутом без зайвих дій [1].

1.4 Постановка завдання дослідження

Задача дослідження полягає у створенні вебсервісу, який автоматизує інтерактивне планування соціальних туристичних маршрутів на основі каталогу об'єктів, параметрів доступності та профілю користувача або групи.

Для розв'язання задачі потрібно визначити інформаційну модель, описати ролі користувачів, сформулювати вимоги до інтерфейсу, спроектувати логіку побудови маршруту, реалізувати серверні методи та підготувати клієнтські екрани для роботи з планом подорожі.

Для досягнення поставленої мети необхідно розв'язати такі основні завдання: проаналізувати сучасний стан предметної області та наявні підходи до вирішення подібних задач; визначити потреби цільових користувачів і сформулювати перелік функціональних вимог; розабезпечити структуру даних і логіку взаємодії компонентів; спроектувати користувацький інтерфейс;

реалізувати основні програмні модулі; провести перевірку роботи системи на типових сценаріях; сформулювати висновки щодо ефективності запропонованого рішення.

Під час проєктування необхідно врахувати ситуації, коли дані про об'єкти неповні або параметри користувача суттєво обмежують вибір. У таких випадках система має пояснювати причину обмеженого результату і давати можливість змінити критерії планування.

РОЗДІЛ 2

ПРОЕКТУВАННЯ РОЗРОБКИ

2.1 Архітектурне проєктування

Архітектура вебсервісу проєктується як клієнт-серверна система, у якій інтерфейс користувача, серверна логіка, база даних і картографічні компоненти мають окрему відповідальність. Такий поділ дозволяє незалежно розвивати каталог об'єктів, маршрутизатор і користувацький інтерфейс.

Клієнтська частина відповідає за введення параметрів, роботу з картою, перегляд списку об'єктів, редагування маршруту і показ підсумкового плану. Вона не виконує складні обчислення самостійно, а передає параметри до серверної частини через API [2].

Серверна частина обробляє профіль планування, застосовує фільтри, перевіряє доступність об'єктів, формує варіанти маршруту і повертає структурований результат. Саме цей рівень забезпечує узгодженість між даними каталогу та діями користувача.

Сховище даних містить туристичні об'єкти, категорії, параметри доступності, маршрути, користувачів, відгуки та службові налаштування. Дані організовуються так, щоб одна локація могла використовуватися в багатьох маршрутах без дублювання.

Картографічний компонент використовується для відображення локацій, побудови шляху і візуального контролю маршруту. Його взаємодія з основною системою має бути обмежена чіткими запитамі, щоб зміна зовнішнього сервісу не порушувала логіку вебзастосунку.

Архітектурне рішення також передбачає обробку помилок, журналювання ключових дій і перевірку прав доступу. Це важливо для адміністративних операцій, зміни довідників і збереження маршрутів користувача.

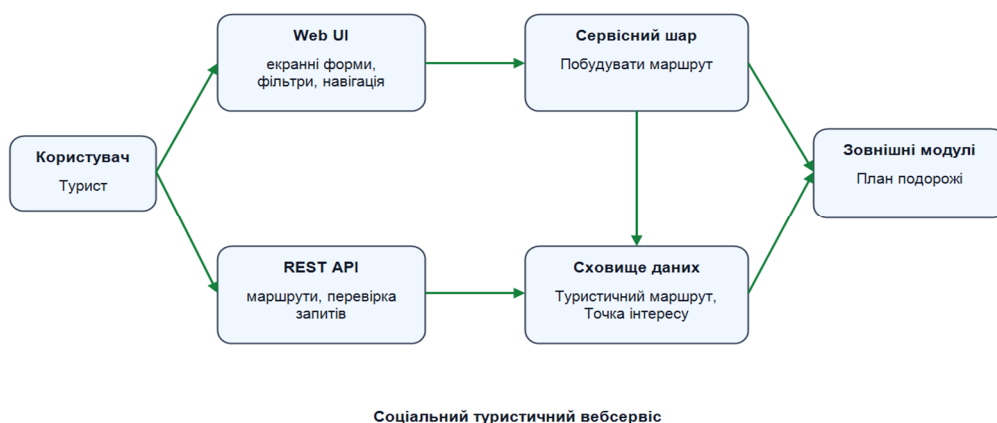


Рисунок 2.1 – Архітектура програмної системи

Рисунок 2.1 узагальнює архітектуру системи і показує основні канали взаємодії між користувацьким інтерфейсом, API, бізнес-логікою, сховищем даних і картографічним сервісом.

2.2 Проєктування функціональних сценаріїв і структури даних

Функціональні сценарії системи будуються навколо роботи користувача з маршрутом. Основний сценарій починається з вибору параметрів подорожі, продовжується фільтрацією об'єктів і завершується побудовою маршруту з можливістю збереження результату [3].

У системі виділяються такі ролі: турист, координатор групи та адміністратор. Турист переглядає об'єкти і створює маршрут, координатор враховує потреби групи, а адміністратор підтримує каталог локацій, категорії, параметри доступності та службові статуси.

Модель варіантів використання відображає дії, пов'язані з пошуком, фільтрацією, переглядом картки об'єкта, побудовою маршруту, збереженням плану і керуванням довідниками. Така модель допомагає перевірити, чи не пропущено ключову дію користувача.

Структура даних проєктується навколо сутностей: користувач, туристичний об'єкт, категорія, параметр доступності, маршрут, точка маршруту,

відгук і збережений план. Зв'язки між ними дозволяють відстежити, які об'єкти включені до маршруту і які критерії вплинули на добір.

Туристичний об'єкт містить координати, опис, категорію, режим роботи, ознаки доступності, орієнтовну тривалість відвідування та контактні відомості. Ці поля використовуються під час фільтрації, сортування і відображення на карті.

Маршрут зберігається як послідовність точок із порядком відвідування, часовими параметрами і службовим станом. Така структура дозволяє редагувати окремі точки без повного перестворення плану.



Рисунок 2.2 – UML-діаграма варіантів використання системи

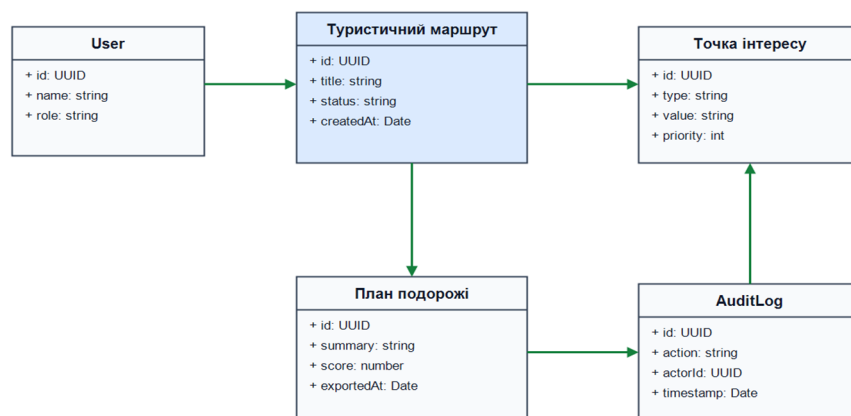


Рисунок 2.3 – Концептуальна модель даних

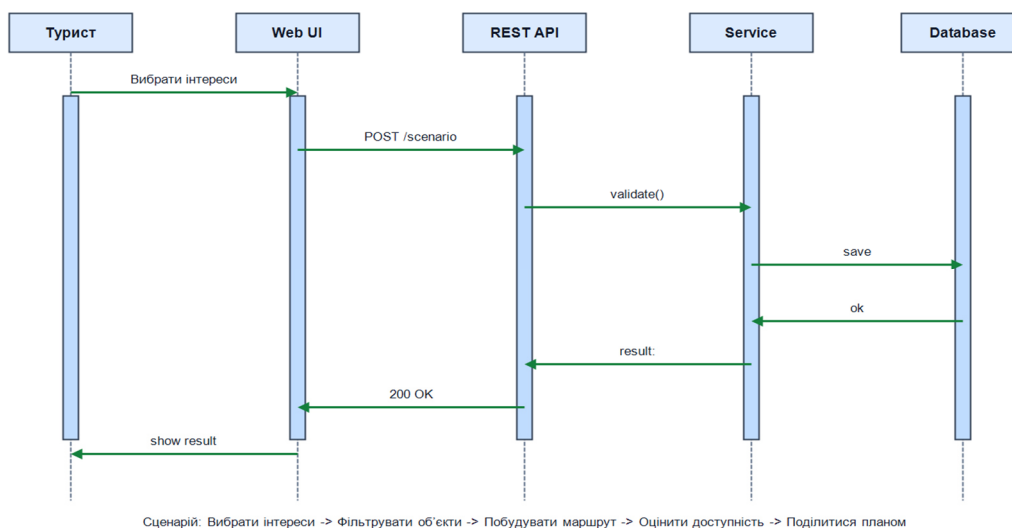


Рисунок 2.4 – Послідовність виконання основного сценарію

Рисунки 2.2-2.4 пояснюють варіанти використання, логічну модель даних і послідовність основного сценарію. Разом вони формують проєктну основу для реалізації серверної логіки та клієнтського інтерфейсу [4].

2.3 Проєктування інтерфейсу

Інтерфейс вебсервісу проєктується як послідовність робочих екранів авторизації, головна панель, каталог об'єктів, карта маршруту, детальна картка локації, збережений план і адміністративні налаштування.

Екран планування має поєднувати карту, список доступних об'єктів, фільтри та блок маршруту. Така структура дає користувачу змогу одночасно бачити географічне розташування, параметри добору і підсумкову послідовність відвідування.

Технологічний стек обирається з урахуванням інтерактивної карти, REST API, роботи з базою даних і можливості локального запуску. Клієнтська частина відповідає за візуальну взаємодію, серверна частина обробляє бізнес-логіку, а сховище забезпечує стабільну роботу з довідниками та маршрутами [5].

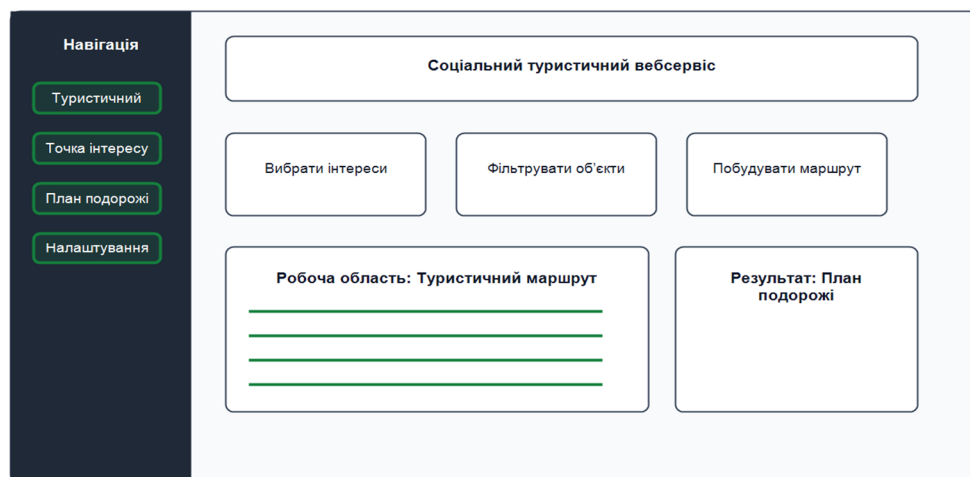


Рисунок 2.5 – Узагальнений макет екранних форм і навігації

Рисунок 2.5 демонструє узагальнений макет інтерфейсу, де карта, список об'єктів, фільтри і підсумковий маршрут працюють як єдина робоча область.

Критеріями якості системи є коректність фільтрації, стабільність побудови маршруту, зрозумілість повідомлень, доступність інтерфейсу, швидкість відповіді і можливість повторного запуску контрольного сценарію.

Окремо враховується зручність роботи з помилками. Якщо маршрут не може бути побудований через нестачу даних або надто жорсткі обмеження, користувач має отримати пояснення і варіанти коригування параметрів.

РОЗДІЛ 3

СТВОРЕННЯ СМАРТ КОНТРАКТУ

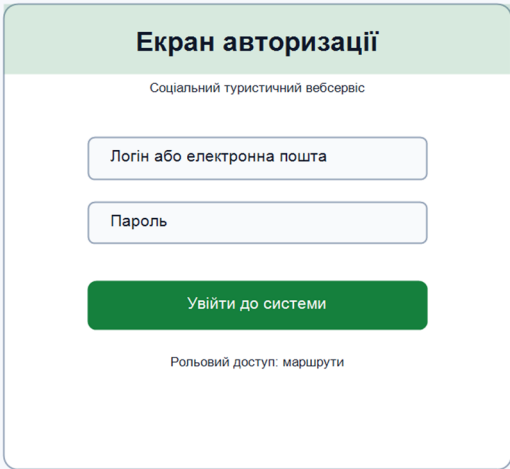
3.1 Середовище розробки та структура програмного проєкту

Реалізуємо вебсервіс соціального туризму як прототип із клієнтською частиною, серверним API, моделлю даних і сервісом побудови маршруту. Ми зосереджуємося на сценарії, у якому користувач добирає об'єкти, перевіряє доступність і формує маршрут.

Структуру програмного проєкту організовуємо за модулями авторизації, каталог об'єктів, карта, маршрути, доступність, відгуки та адміністративні довідники. Такий поділ допомагає підтримувати код і тестувати окремі частини системи.

У клієнтській частині створюємо екрани входу, головної панелі, списку об'єктів, форми редагування, детальної картки, карти маршруту і адміністративних налаштувань. Кожен екран відповідає конкретному кроку користувацького сценарію [6].

Рисунок 3.1 демонструє екран авторизації, через який користувач входить до системи.



Екран авторизації

Соціальний туристичний вебсервіс

Логін або електронна пошта

Пароль

Увійти до системи

Рольовий доступ: маршрути

Рисунок 3.1 – Макет екрана авторизації користувача

Рисунок 3.2 показує головну панель, де доступні останні маршрути, швидкі дії та стан основних модулів.

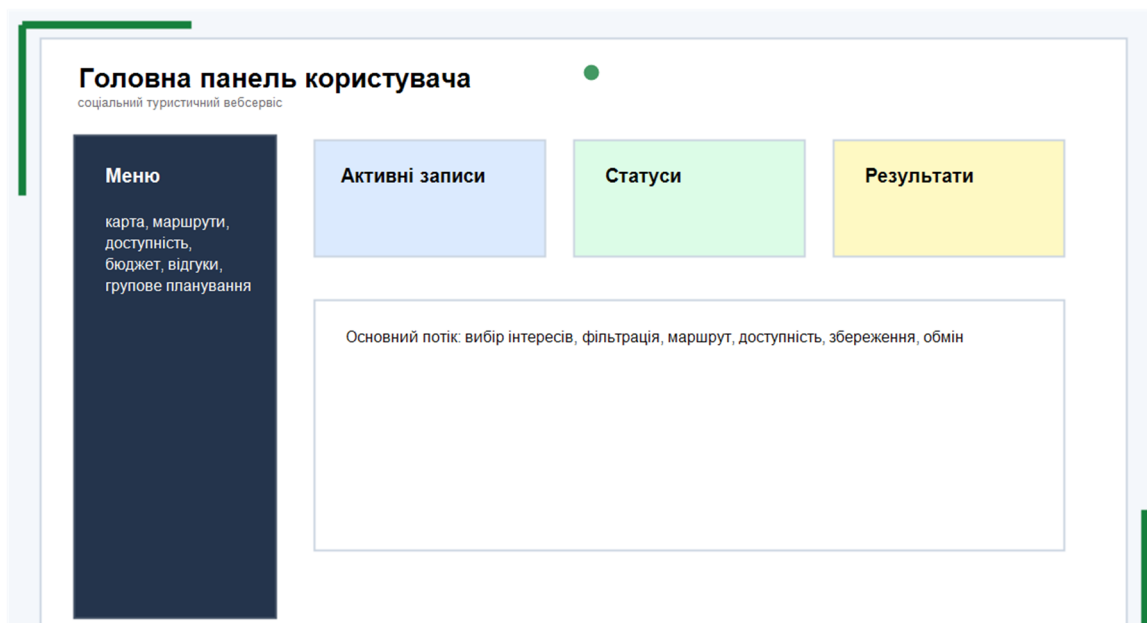


Рисунок 3.2 – Головна панель користувача застосунку

Лістинг 3.1 – Модель даних для об'єкта маршруту

```
export interface MainRecord {
  id: string;
  title: string;
  status: 'draft' | 'active' | 'done' | 'archived';
  ownerId: string;
  createdAt: string;
  updatedAt: string;
  fields: {
    description?: string;
    domainObject: 'туристичний маршрут';
    mainModule: 'точки інтересу';
    resultType: 'персональний план подорожі';
  };
}
export const emptyRecord = (ownerId: string): MainRecord => ({
  id: crypto.randomUUID(),
  title: '',
  status: 'draft',
  ownerId,
  createdAt: new Date().toISOString(),
  updatedAt: new Date().toISOString(),
  fields: {
    domainObject: 'туристичний маршрут',
    mainModule: 'точки інтересу',
    resultType: 'персональний план подорожі'
  }
});
```

кінець лістингу 3.1

Лістинг 3.1 подає базову модель даних для об'єкта маршруту. Ми використовуємо її як основу для збереження назви, статусу, дат оновлення та службового навантаження, пов'язаного з плануванням соціальної подорожі.

Серверну частину реалізуємо через API-методи, які приймають дані від клієнта, виконують валідацію, звертаються до сервісів і повертають структуровану відповідь. Усі операції з маршрутами проходять через сервісний шар [7-8].

Модель даних описує туристичний маршрут, об'єкти маршруту, параметри доступності, службові статуси та зв'язок із користувачем. Це дозволяє зберігати план подорожі і повторно відкривати його для редагування.

3.2 Реалізація серверної логіки та моделі даних

У серверній частині реалізуємо модель даних, яка відображає основні сутності предметної області: користувача, туристичний об'єкт, маршрут, точку маршруту, параметр доступності, категорію та відгук.

Модель туристичного об'єкта описує локацію як структурований запис із назвою, координатами, категорією, описом, режимом роботи, параметрами доступності та службовим статусом. Ці поля використовуємо під час фільтрації і побудови маршруту.

Маршрут реалізуємо як набір впорядкованих точок. Кожна точка посилається на туристичний об'єкт і зберігає порядок відвідування, орієнтовний час перебування та службову інформацію для відображення на карті.

Під час створення або оновлення записів виконуємо серверну валідацію. Ми перевіряємо обов'язкові поля, коректність статусу, наявність координат, допустимість параметрів доступності та право користувача змінювати запис.

Сервіс побудови маршруту приймає параметри користувача, відбирає об'єкти за категоріями та доступністю, формує послідовність точок і повертає результат для клієнтської частини. Якщо даних недостатньо, сервіс формує пояснення для користувача.

Після виконання основного сценарію сервер зберігає маршрут, записує службову подію і повертає клієнту підсумковий план. Це дозволяє повторно відкрити маршрут, переглянути його деталі та змінити окремі точки.

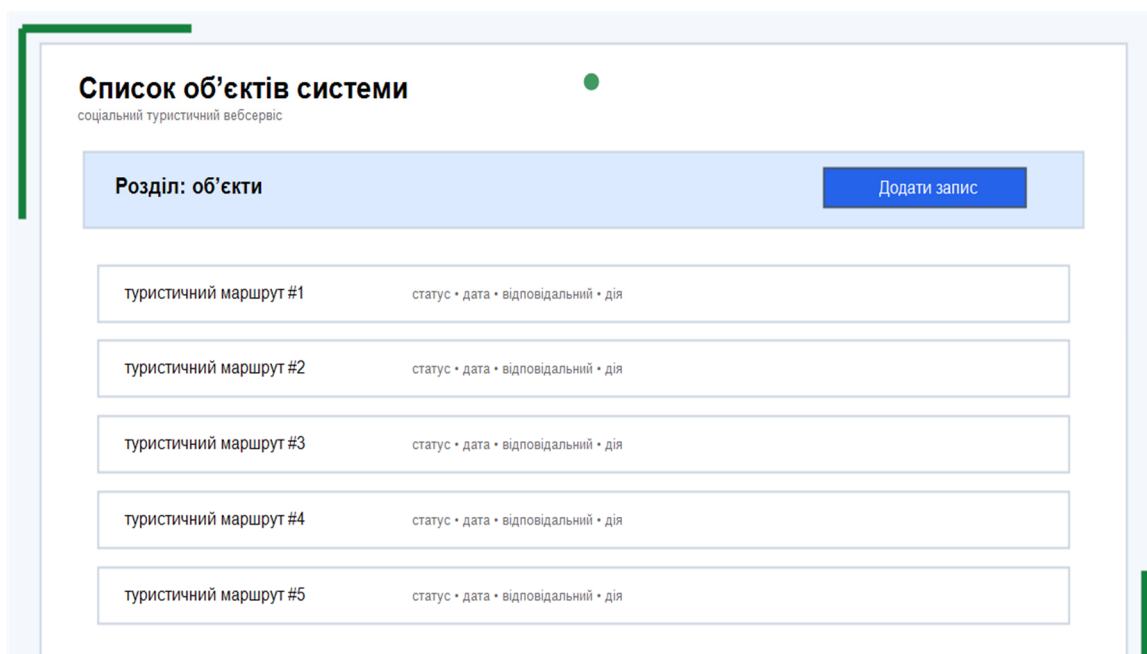


Рисунок 3.3 – Інтерфейс списку об'єктів системи

На рисунку 3.3 показуємо інтерфейс списку туристичних об'єктів, через який користувач переглядає доступні локації, порівнює основні характеристики і переходить до детальної картки. Для соціального маршруту такий список є робочою основою фільтрації.

У списку важливо показати назву, категорію, ознаки доступності, короткий опис і доступні дії. Це дозволяє швидко оцінити, чи підходить об'єкт для запланованої подорожі.

Поля форми групуємо за змістом, базова інформація, координати, категорія, доступність і службовий стан. Такий поділ зменшує кількість помилок під час введення даних [9-10].

Форма створення / редагування
соціальний туристичний вебсервіс

Форма: побудова маршруту

об'єкт

доступність

бюджет

транспорт

маршрут

відгук

Зберегти

Рисунок 3.4 – Форма створення або редагування запису

На рисунку 3.4 демонструємо форму створення або редагування запису. У межах вебсервісу вона використовується для оновлення туристичного об'єкта, параметрів маршруту або службових даних каталогу.

Картка детального перегляду
соціальний туристичний вебсервіс

Картка: туристичний маршрут

об'єкт, доступність, бюджет,
транспорт, маршрут, відгук

Історія та пов'язані дані

На сторінці відображаються деталі, статуси, коментарі, пов'язані записи та доступні дії для поточного користувача.

Рисунок 3.5 – Картка детального перегляду об'єкта

На рисунку 3.5 подано картку детального перегляду туристичного об'єкта. Вона містить опис локації, параметри доступності, службові відомості та дії, пов'язані з додаванням об'єкта до маршруту.

Картка допомагає користувачу прийняти рішення щодо включення локації до плану. Ми використовуємо її як проміжний екран між загальним списком і побудовою маршруту.

Лістинг 3.2 – Метод API для створення запису в системі

```
router.post('/records', authorize('турист'), async (req, res, next) => {
  try {
    const payload = validateRecordPayload(req.body);
    const record = await recordService.createRecord({
      ...payload,
      ownerId: req.user.id,
      source: 'соціальний туристичний вебсервіс'
    });

    res.status(201).json({
      ok: true,
      message: 'Запис успішно створено',
      data: record
    });
  } catch (error) {
    next(error);
  }
});
```

кінець лістингу 3.2

У лістингу 3.2 наведено серверний маршрут створення запису. Ми приймаємо дані від клієнтської частини, перевіряємо права користувача, виконуємо валідацію і передаємо об'єкт до сервісного шару.

Такий підхід відокремлює HTTP-обробку від прикладної логіки. Маршрут відповідає за запит і відповідь, а сервіс працює з туристичними об'єктами, маршрутами або службовими записами [11].

У лістингу 3.3 показано функцію перевірки даних перед збереженням. Ми контролюємо назву, статус і текстові поля, щоб не допустити некоректний запис у базу даних.

Лістинг 3.3 – Функція серверної валідації вхідних даних

```
function validateRecordPayload(payload) {
  const errors = {};

  if (!payload.title || payload.title.trim().length < 3) {
    errors.title = 'Назва повинна містити щонайменше 3 символи';
  }
  if (!payload.status) {
    errors.status = 'Необхідно вказати статус запису';
  }
  if (Object.keys(errors).length > 0) {
    throw new ValidationError('Помилка перевірки даних', errors);
  }

  return {
    title: payload.title.trim(),
    status: payload.status,
    description: payload.description?.trim() ?? ''
  };
}
```

кінець лістингу 3.3

У лістингу 3.4 показано функцію виконання основного сценарію «побудова маршруту».

Лістинг 3.4 – Сервіс виконання основного сценарію «побудова маршруту»

```
async function runMainScenario(recordId, userId) {
  const record = await repository.findById(recordId);
  assertAccess(record, userId);

  const steps = 'вибір інтересів, фільтрація, маршрут, доступність, збереження, обмін'.split(',').map(step => step.trim());
  const result = await scenarioEngine.execute({
    record,
    steps,
    action: 'побудова маршруту',
    expectedResult: 'персональний план подорожі'
  });

  await auditLog.write({
    userId,
    recordId,
    action: 'побудова маршруту',
    status: result.status
  });

  return result;
}
```

кінець лістингу 3.4

Для соціального туристичного сервісу таку перевірку розширюємо правилами для координат, категорій, параметрів доступності, режиму роботи та зв'язку об'єкта з маршрутом.

3.3 Реалізація клієнтського інтерфейсу застосунку

Клієнтський інтерфейс реалізуємо як набір екранів, які підтримують послідовну роботу з маршрутом. Ми виділяємо екран входу, головну панель, каталог об'єктів, карту маршруту, детальну картку локації, результати планування та адміністративний модуль.

На головній панелі показуємо останні маршрути, швидкий перехід до створення нового плану, стан каталогу і короткі повідомлення. Такий екран допомагає користувачу одразу перейти до планування або продовжити попередній маршрут.

Екран списку об'єктів підтримує пошук, фільтри за категорією, доступністю та інтересами. У клієнтській частині групуємо дані так, щоб користувач бачив важливі характеристики без відкриття кожної картки.

Екран карти маршруту поєднує візуальне розташування точок і список етапів подорожі. Ми показуємо порядок відвідування, ключові обмеження і даємо можливість змінювати маршрут без повного повторного створення плану.

Форми створення та редагування записів будуються з контрольованих полів. Після зміни даних інтерфейс передає запит до API, відображає стан завантаження і показує повідомлення про успішне збереження або помилку.

Адміністративний екран дозволяє підтримувати довідники, категорії, статуси, параметри доступності та службові записи. Ми виносимо ці операції в інтерфейс, щоб оновлення каталогу не вимагало прямого втручання в базу даних.

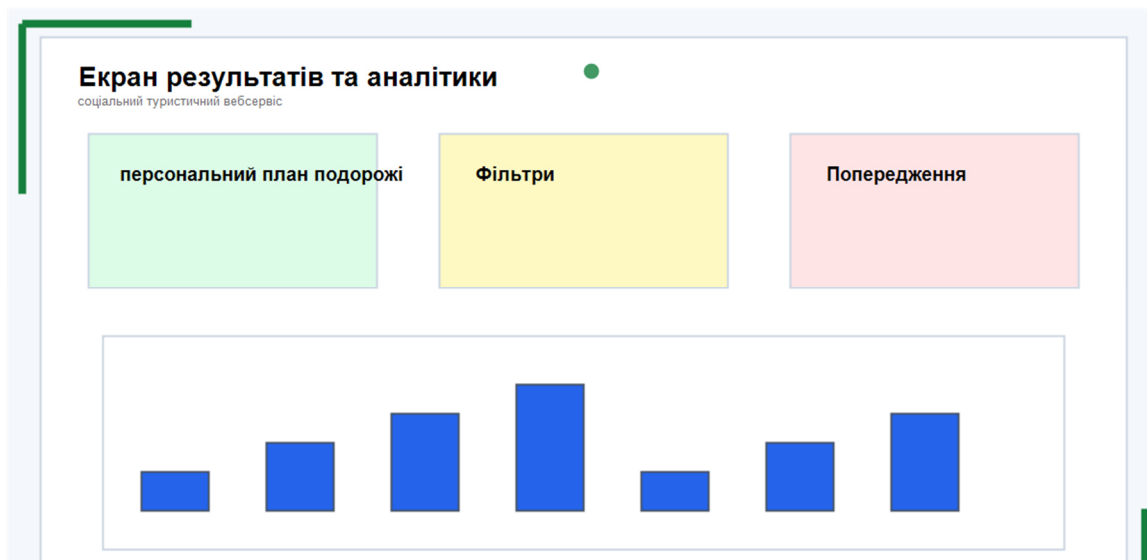


Рисунок 3.6 – Екран результатів та аналітики

На рисунку 3.6 відображаємо екран результатів, де користувач бачить побудований маршрут, перелік точок і аналітичні показники. У цьому блоці поєднуємо карту, список етапів і пояснення обраних локацій.

Такий екран є центральним для демонстрації практичного результату. Користувач може оцінити маршрут, змінити окремі точки або зберегти готовий план подорожі.



Рисунок 3.7 – Адміністративні налаштування застосунку

На рисунку 3.7 показуємо адміністративні налаштування, через які підтримуємо категорії, параметри доступності, статуси та службові довідники. Ці дані впливають на якість фільтрації і побудови маршруту [12].

Адміністративний екран зменшує потребу в прямому редагуванні бази даних. Ми виносимо службові операції в інтерфейс, щоб каталог можна було підтримувати в актуальному стані.

Лістинг 3.5 – Клієнтський метод збереження даних через REST API

```
export async function saveRecord(formState) {
  const response = await fetch('/api/records', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(formState)
  });
  const body = await response.json();
  if (!response.ok || !body.ok) {
    throw new ApiError(body.message, body.errors);
  }
  return body.data;
}
```

кінець лістингу 3.5

У лістингу 3.5 демонструємо клієнтський метод збереження даних через REST API. Ми формуємо POST-запит, передаємо JSON-об'єкт, отримуємо відповідь і перевіряємо результат виконання операції.

Лістинг 3.6 – Обробник форми користувацького інтерфейсу

```
async function handleSubmit(event) {
  event.preventDefault();
  setLoading(true);
  setErrors({});

  try {
    const saved = await saveRecord(formState);
    notify('Дані збережено');
    navigate('/records/' + saved.id);
  } catch (error) {
    setErrors(error.details ?? {});
    notify(error.message ?? 'Не вдалося виконати операцію');
  } finally {
    setLoading(false);
  }
}
```

кінець лістингу 3.6

Метод можна використовувати для різних форм: створення об'єкта, редагування маршруту або збереження службового запису. Якщо сервер повертає помилку, клієнтська частина показує користувачу зрозуміле повідомлення.

У лістингу 3.6 наведено обробник відправлення форми. Ми зупиняємо стандартну поведінку браузера, вмикаємо стан завантаження, очищуємо попередні помилки і викликаємо метод збереження.

Після успішного виконання користувач переходить до сторінки створеного або оновленого запису. У разі помилки інтерфейс показує повідомлення і дозволяє виправити введені дані.

На завершальному етапі розробки готуємо демонстраційний набір даних: туристичні об'єкти, категорії, параметри доступності, тестових користувачів і кілька маршрутів різної складності.

Демонстраційні маршрути створюємо для різних сценаріїв: коротка міська прогулянка, маршрут для групи з вимогами доступності, бюджетний маршрут і маршрут із кількома категоріями об'єктів. Це дозволяє перевірити поведінку системи в різних умовах.

Перед контрольним запуском перевіряємо API, відображення карти, роботу фільтрів, збереження маршруту, відкриття детальної картки та оновлення службових статусів. Ми також перевіряємо реакцію системи на неповні дані.

Окремо контролюємо узгодженість даних між списком об'єктів, картою і підсумковим маршрутом. Якщо користувач змінює точку маршруту, інтерфейс має оновити порядок відвідування і пов'язані показники.

РОЗДІЛ 4

ТЕСТУВАННЯ І РОЗГОРТАННЯ ПРОЕКТУ

4.1 План тестування

Тестуємо вебсервіс як завершений прототип для інтерактивного планування соціальних маршрутів. Ми перевіряємо шлях від вибору параметрів до отримання збереженого плану подорожі.

План тестування поділяємо на позитивні, негативні, граничні та інтеграційні сценарії. Такий поділ допомагає окремо оцінити каталог об'єктів, фільтри, карту, побудову маршруту, збереження результату і повідомлення про помилки. Тестове середовище містить локальний сервер, клієнтський інтерфейс, тестову базу туристичних об'єктів, набір категорій, параметри доступності та кілька профілів користувачів.

Позитивні сценарії охоплюють створення маршруту з повними даними, перегляд карти, відкриття картки локації, зміну параметрів і збереження плану. Очікуваним результатом є маршрут, який відповідає заданим критеріям.

Негативні сценарії перевіряють ситуації з порожнім каталогом, відсутніми координатами, надто жорсткими обмеженнями, некоректним статусом і недоступним записом. У таких випадках система має показувати зрозуміле повідомлення.

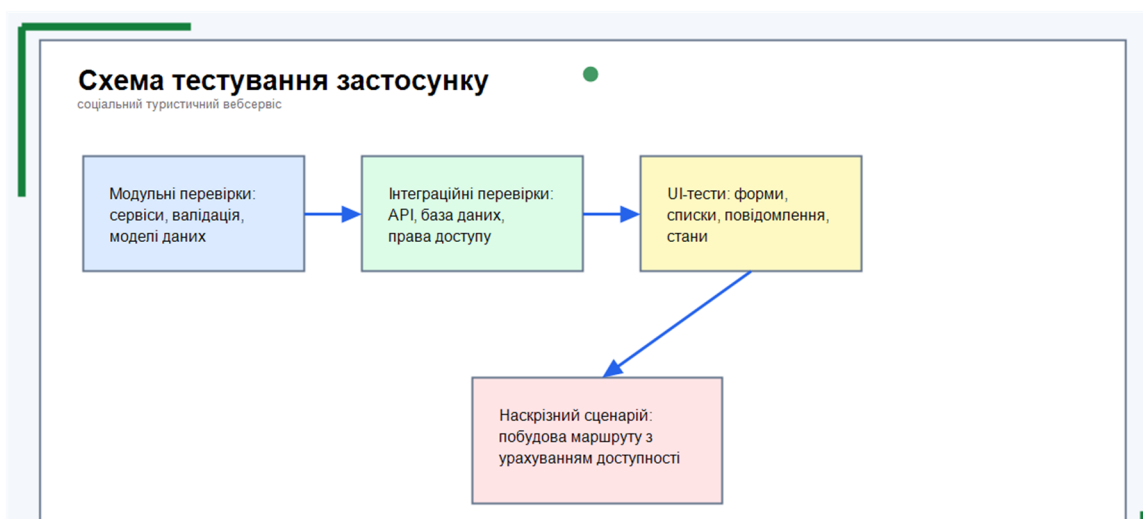


Рисунок 4.1 – Схема тестування застосунку

Граничні сценарії стосуються великої кількості об'єктів, мінімальної тривалості маршруту, однакових координат і повторного запуску планування. Ми перевіряємо, чи зберігає система передбачувану поведінку.

Рисунок 4.1 подає схему тестування застосунку, а таблиця 4.1 фіксує контрольні тестові випадки для основних функцій прототипу.

Таблиця 4.1 – Контрольні тестові випадки

ID	Перевірка	Очікуваний результат
ТС-01	Вхід тестового користувача	Користувач переходить до основної панелі
ТС-02	Переміщення картки	Система формує результат
ТС-03	Некоректні дані у формі	Показуємо повідомлення про помилку
ТС-04	Спроба службової дії без прав	Доступ заблоковано
ТС-05	Повторний запуск сценарію	Дані не дублюються, стан оновлюється

У розділі тестування ми перевіряємо соціальний туристичний вебсервіс як завершений прототип, враховуємо пояснення лістингів та обробку помилкових станів. Основний сценарій охоплює добір об'єктів, побудову маршруту і врахування доступності, збереження даних і отримання результату. Враховуємо обмеження прототипу та формування звіту для планування соціальних маршрутів.

У разі збою ми перевіряємо, чи залишається система в контрольованому стані, додатково враховуємо граничні значення та порожні стани. Особливо важливо, щоб ризики на ігнорування потреб доступності, неактуальні дані про об'єкти і складна координація групи не призводили до втрати або суперечливого відображення даних, додатково враховуємо локальне середовище запуску та стани завантаження.

Підсумкова перевірка підтверджує, що реалізований застосунок відповідає поставленим завданням, враховуємо збереження історії змін та підтримку різних ролей. Демонструємо повний шлях від введення даних до отримання результату

«персональний план соціальної подорожі», враховуємо узгодження рисунків із текстом та узгодження вимог і реалізації.

План тестування поділяємо на позитивні, негативні, граничні та інтеграційні перевірки. Враховуємо контроль якості результату та граничні значення. Такий поділ допомагає окремо оцінити візуалізацію станів, ролі користувачів і роботу системи в нестандартних умовах, враховуємо відновлення після збою.

4.2 Результати функціонального тестування

Результати перевірки основних функцій вебсервісу фіксуємо. Запускаємо сценарії для каталогу об'єктів, фільтрації, карти, побудови маршруту, збереження плану і адміністративних налаштувань.

Позитивні сценарії підтвердили, що система коректно формує маршрут для профілів із повними параметрами. Після вибору інтересів, способу пересування і вимог доступності вебсервіс повертає послідовність локацій і показує їх на карті.

Негативні сценарії показали, що система не створює випадковий маршрут за відсутності потрібних даних. Замість цього користувач отримує повідомлення про причину помилки і може змінити параметри планування.

Інтеграційні перевірки підтвердили зв'язок між списком об'єктів, картою і збереженим маршрутом. Коли ми змінюємо об'єкт або порядок точок, підсумковий план оновлюється відповідно до нового стану.

Рисунок 4.2 узагальнює результати контрольного тестування, а таблиця 4.2 показує критерії готовності прототипу до демонстрації. Виявлені зауваження не блокують основний сценарій і можуть бути використані для подальшого розвитку [13].



Рисунок 4.2 – Узагальнені результати контрольного тестування

Граничні перевірки потрібні для оцінювання мінімальних і максимальних допустимих значень, враховуємо інтерпретацію результатів та підготовку тестового набору. Вони пов'язані з даними, туристичні об'єкти, параметри доступності, маршрути, відгуки і часові обмеження.

Таблиця 4.2 – Критерії готовності до демонстрації

Критерій	Стан	Коментар
Основний сценарій	Виконуємо	Позитивний сценарій підтверджує, що користувач може виконати дію «добір об'єктів, побудову маршруту і врахування доступності» без помилок.
Тестові дані	Підготовлено	Туристичний маршрут
Помилки введення	Перевіряємо	Система відображає повідомлення
Журнал подій	Доступний	Фіксуються ключові дії
Розгортання	Підготовлено	Негативні сценарії моделюють помилкове введення, відсутність прав доступу, неповні дані та недоступність службового компонента; у цій роботі ми додатково враховуємо узгодження вимог і реалізації та перевірку збереження для теми «інтерактивне планування соціальних маршрутів». Ми перевіряємо, що система не втрачає дані і повертає зрозуміле повідомлення; у цій роботі ми додатково враховуємо параметри фільтрації та параметри фільтрації для теми «інтерактивне планування соціальних маршрутів».

Розгортання проєкту виконуємо після завершення функціонального тестування. Ми готуємо середовище, перевіряємо залежності, запускаємо серверну частину, відкриваємо клієнтський інтерфейс і підключаємо тестові дані.

Для локального запуску використовуємо конфігурацію з демонстраційними туристичними об'єктами, параметрами доступності, користувачами і збереженими маршрутами. Це дозволяє одразу перевірити основний сценарій без ручного наповнення системи.

Контрольний запуск починаємо з вибору параметрів маршруту. Далі перевіряємо фільтрацію об'єктів, відображення карти, побудову маршруту, відкриття детальної картки і збереження готового плану.

Під час розгортання контролюємо журнали помилок, відповіді API, доступність клієнтської частини, коректність статичних ресурсів і стан початкових даних. Якщо виникає помилка, ми перевіряємо параметри підключення і структуру тестового набору.

Після контрольного запуску робимо висновок, що прототип готовий до демонстрації основної функціональності. Система підтримує перегляд об'єктів, фільтрацію, побудову маршруту, збереження плану і роботу з адміністративними даними.

ВИСНОВКИ

У кваліфікаційній роботі вирішено прикладну задачу розроблення вебсервісу для соціального туризму з інтерактивним плануванням маршрутів. Проведений аналіз показав, що ручне складання соціальної подорожі ускладнюється потребою одночасно враховувати туристичні об'єкти, доступність, транспорт, бюджет, інтереси групи і часові обмеження.

Досліджено предметну область, визначено основні ролі користувачів, інформаційні сутності, ризики та обмеження. На основі аналізу сформовано вимоги до вебсервісу, зокрема до каталогу об'єктів, профілю планування, карти, перевірки доступності та збереження маршруту.

Виконано проектування програмної системи. Описано клієнт-серверну архітектуру, функціональні сценарії, структуру даних, логіку взаємодії модулів, макет інтерфейсу і критерії якості. Проектні рішення забезпечили перехід від аналітичних вимог до програмної реалізації.

Реалізовано основну функціональність прототипу. У роботі описано модель даних, API-методи, серверну валідацію, сервіс побудови маршруту, клієнтські екрани, роботу з формами, карту, детальну картку об'єкта і адміністративні налаштування.

Виконано тестування і підготовку проєкту до розгортання. Перевірено позитивні, негативні, граничні та інтеграційні сценарії, зокрема роботу каталогу, фільтрів, карти, маршрутизатора, збереження плану і повідомлень про помилки.

Практичне значення результату полягає у створенні навчального прототипу, який спрощує підготовку соціальних туристичних маршрутів і може бути розширений за рахунок інтеграції з картографічними API, оновлення довідників доступності, групового погодження маршруту та аналітики відгуків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AllTrails. Trail Guides and Maps for Hiking, Camping, and Running. AllTrails. URL: <https://www.alltrails.com/> (дата звернення: 25.01.2026).
2. Client-server overview. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Client-Server_overview (дата звернення: 30.04.2026).
3. Docker Docs. Get started. Docker Inc. URL: <https://docs.docker.com/get-started/> (дата звернення: 30.04.2026).
4. Express.js. Routing guide. OpenJS Foundation. URL: <https://expressjs.com/en/guide/routing/> (дата звернення: 30.04.2026).
5. FastAPI. Documentation. FastAPI. URL: <https://fastapi.tiangolo.com/> (дата звернення: 30.04.2026).
6. HTTP overview. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Overview> (дата звернення: 30.04.2026).
7. Introduction to Node.js. Node.js. URL: <https://nodejs.org/learn/getting-started/introduction-to-nodejs> (дата звернення: 30.04.2026).
8. MongoDB Manual. MongoDB,. URL: <https://www.mongodb.com/docs/manual/> (дата звернення: 30.04.2026).
9. OpenAPI Specification. Swagger. URL: <https://swagger.io/specification/> (дата звернення: 30.04.2026).
10. OWASP Top Ten Web Application Security Risks. OWASP Foundation. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 12.05.2026).
11. PostgreSQL 18.4 Documentation. PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/current/index.html> (дата звернення: 30.04.2026).
12. Quick Start. React Documentation. Meta Open Source,. URL: <https://react.dev/learn> (дата звернення: 12.05.2026).

13. Routes API. Google Maps Platform Documentation. Google for Developers.
URL: <https://developers.google.com/maps/documentation/routes> (дата звернення:
12.05.2026).

ДОДАТКИ

ДОДАТОК А

Фрагменти програмного коду ілюструють реалізацію основних методів розробки. Лістинги відображають структуру моделі даних, API-методу та сервісу виконання основної операції.

Лістинг А.1 – Опис базової моделі даних.

```
export interface SocialRoute {  
  id: string;  
  title: string;  
  status: 'draft' | 'active' | 'done';  
  createdAt: string;  
  updatedAt: string;  
  payload: {  
    objectName: 'туристичний маршрут';  
    actionName: 'побудова маршруту';  
  };  
}
```

Лістинг А.2 – Приклад API-методу створення запису.

```
router.post('/api/socialroute', async (req, res, next) => {  
  try {  
    const data = validatePayload(req.body);  
    const result = await service.create(data);  
    res.status(201).json({ ok: true, data: result });  
  } catch (error) {  
    next(error);  
  }  
});
```

Лістинг А.3 – Сервіс виконання основної операції.

```
async function runMainAction(id, userId) {  
  const record = await repository.findById(id);  
  accessPolicy.assertCanUpdate(record, userId);  
  const result = await workflow.execute('побудова маршруту', record);  
  await auditLog.write({ id, userId, action: 'побудова маршруту' });  
  return result;  
}
```