TimeWanderer

Software Requirements Specification

Version 1

01.09.2024

Team

Daria Avtomonova  
Anna Kodola

Kateryna Pasenko

**Revision History**

| **Date** | **Description** | **Author** | **Comments** |
| --- | --- | --- | --- |
| 04.10.2023 | Version 1 | Pasenko  Avtomonova  Kodola | Version 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

| **Signature** | **Printed Name** | **Title** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**REVISION HISTORY II**

**DOCUMENT APPROVAL II**

**1. INTRODUCTION 1**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

**2. GENERAL DESCRIPTION 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

**3. SPECIFIC REQUIREMENTS 2**

3.1 External Interface Requirements 3

*3.1.1 User Interfaces 3*

*3.1.2 Hardware Interfaces 3*

*3.1.3 Software Interfaces 3*

*3.1.4 Communications Interfaces 3*

3.2 Functional Requirements 3

*3.2.1 <Functional Requirement or Feature #1> 3*

*3.2.2 <Functional Requirement or Feature #2> 3*

3.3 Use Cases 3

*3.3.1 Use Case #1 3*

*3.3.2 Use Case #2 3*

3.4 Classes / Objects 3

*3.4.1 <Class / Object #1> 3*

*3.4.2 <Class / Object #2> 3*

3.5 Non-Functional Requirements 4

*3.5.1 Performance 4*

*3.5.2 Reliability 4*

*3.5.3 Availability 4*

*3.5.4 Security 4*

*3.5.5 Maintainability 4*

*3.5.6 Portability 4*

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

**4. ANALYSIS MODELS 4**

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

**5. CHANGE MANAGEMENT PROCESS 5**

**A. APPENDICES 5**

A.1 Appendix 1 5

A.2 Appendix 2 5

**1. Introduction**

**1.1 Purpose**

Мета цього документа - створення платформи для зручного пошуку відповідних маршрутів для мандрівників, їх проходження та гейміфікації цього процесу.

**1.2 Scope**

TimeWanderer - це онлайн-платформа, яка надає можливість пошуку туристичних маршрутів та обміну вражень з іншими користувачами платформи.

Користувач може шукати потрібні йому маршрути, читати інформацію про місця, які він відвідує, залишати свої враження у формі коментарів та статей, а також підіймати власний рейтинг та здобувати досягнення шляхом проходження маршрутів.

Адміністратор може створювати, редагувати та видаляти маршрути, статті та місця, одобрювати або відхиляти поданні на розгляд користувачем статті, створення та редагування досягнень.

Користувач не може проходити більше одного маршруту одночасно.

**1.3 Definitions, Acronyms, and Abbreviations**

Table 1 - Definitions

| **Term** | **Definition** |
| --- | --- |
| TimeWanderer | Платформа для зручного пошуку відповідних маршрутів для мандрівників, їх проходження та гейміфікації цього процесу. |
| Користувач | Користувач, який має можливість користуватися даними платформи. |
| Адміністратор | Користувач, який має можливість керувати даними платформи. |
| Досягнення | Відзнака, яку отримує користувач після успішного проходження маршруту. |
| Маршрут | Дорога, яка поєднує декілька ключових місць, які відвідує користувач. |
| Місце | Особлива точка, визначна пам’ятка, або просто красиве місце, яке є однією з ключових точок з яких складається маршрут. |
| Рейтинг | Це кількість балів, що користувач сумарно набрав за весь час |
| Рівень користувача | Число або показник, який відображає ступінь активності, досягнень користувача. |

**1.4 References**

Table 2 - References

| **Document title** | **Last update** | **Source** |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**1.5 Overview**

Весь SRS-документ складається з п’яти частин. Друга частина містить загальний опис проекту. Також, даний розділ визначає типи користувачів і їх можливості.

Третій розділ описує функціональність більш детально, надає специфікацію та описує інтерфейси.

Четвертий пункт містить опис моделі програми.

П'ятий розділ описує модифікацію та затвердження вимог до проекту.

**2. General Description**

**2.1 Product Perspective**

“TimeWanderer” є самостійною онлайн-платформою, яка орієнтована на покращення туристичного досвіду через користувацький контент та відкриття нових маршрутів. На відміну від традиційних туристичних гідів чи картографічних сервісів, TimeWanderer підкреслює індивідуальне дослідження, дозволяючи користувачам знаходити та ділитися туристичними маршрутами, які відповідають їхнім уподобанням. Платформа слугує цифровим супутником для мандрівників, надаючи детальну інформацію про різні місця призначення та пропонуючи структуровану систему для відстеження й обміну туристичним досвідом. Продукт інтегрується з існуючими онлайн-сервісами для отримання та відображення додаткової інформації про місця, такі як карти. Унікальний набір функцій платформи, включаючи проходження маршрутів і здобуття досягнень, відрізняє її від інших туристичних та картографічних додатків, акцентуючи увагу на залученні користувачів та контенті, створеному спільнотою.

**2.2 Product Functions**

Функції продукту класифікуються наступним чином:

**2.2.1 Реєстрація**

Реєстрація користувачів: Програмне забезпечення дозволяє користувачам створювати облікові записи.

**2.2.2 Авторизація**

Авторизація в обліковому записі: Неавторизований користувач може увійти в власний обліковий запис для отримання доступу до основного функціоналу сайту.

**2.2.3 Проходження маршрутів**

Проходження маршрутів: Користувач має можливість почати проходження маршруту, відмічати відвідані локації та завершити маршрут, отримавши досягнення.

**2.2.4 Взаємодія з маршрутами**

Перегляд інформації: Користувач та адміністратор можуть переглядати інформацію про певний маршрут.

Створення, редагування та видалення маршрутів: Адміністратор може створювати, редагувати та видаляти маршрути.

**2.2.5 Досягнення**

Отримання досягнень: Користувач може проходити маршрути і по закінченню отримати нагороду.

Перегляд отриманих досягнень: Користувач може переглядати в обліковому записі отримані досягнення.

**2.2.6 Взаємодія з місцями**

Перегляд інформації: Користувач та адміністратор можуть переглядати інформацію про певне місце.

Створення, редагування та видалення місць: Адміністратор може створювати, редагувати та видаляти місця.

**2.2.7 Взаємодія зі статтями**

Створення статей: Користувач має можливість написати статтю та відправити її на перевірку модератору

Перегляд статті: Користувач, модератор та адміністратор можуть переглядати опубліковані статті на сайті.

**2.2.8 Коментування**

Публікація коментарів: Користувач, адміністратор та модератор можуть залишати коментарі під публікаціями (місця, маршрути та статті), а також відповідати на вже створенні попередніми користувачами.

**2.2.9 Обліковий запис**

Редагування облікового запису: Користувач, адміністратор та модератор мають можливість редагувати дані облікового запису.

Видалення облікового запису: Користувач, адміністратор та модератор можуть видалити обліковий запис.

Додавання друзів: Користувач може додавати інших користувачів в друзі.

**2.2.10 Пошук**

Пошук маршруту, статті або місця: Користувач, адміністратор та модератор мають можливість пошуку потрібних статей, маршрутів або місць за назвою.

**2.2.11 Фільтрація**

Фільтрація за певними категоріями: Користувач, модератора та адміністратор мають можливість фільтрувати місця, статті та маршрути за певними категоріями.

**2.2.12 Модерація статтей**

Перевірка статей: Модератор має можливість відхиляти або схвалювати статті, які користувачі відправляють на перевірку.

**2.2.13 Модерація коментарів**

Перевірка коментарів: Модератор має можливість переглядати коментарі та видаляти ті, які порушують правила сайту.

**2.2.14 Відображення карти**

Відображення на карті місця: Користувач, адміністратор та модератор можуть переглядати розміщення певного місця на карті.

Відображення маршруту: Користувач, адміністратор та модератор можуть переглядати точки маршруту на карті.

**2.2.15 Рейтинг**

Відображення рейтингу: Користувач, адміністратор та модератор можуть переглядати рейтинг користувачів по загальній кількості набраних балів.

**2.2.16 Рівень**

Накопичення рівня: Користувач отримуватиме певну кількість очок за проходження різних завдань.

**2.2.17 Відгук**

Оцінювання: Користувач може оцінити пройдений маршрут або місце на сторінці.

**2.2.18 Зміна паролю**

Встановлення нового паролю: Користувач, адміністратор та модератор можуть встановити новий пароль у разі того, якщо забули його на сторінці редагування облікового запису.

**2.2.19 Відновлення паролю**

Відновлення паролю: Користувач та модератор можуть відновити пароль, отримавши повідомлення на електронну пошту.

**2.2.20 Резервне копіювання**

Резервне копіювання бази даних адміністратором: Адміністратор може зробити резервну копію бази даних.

**2.2.21 Експорт даних**

Експорт даних системи: Адміністратор може зробити експортування даних системи

**2.2.22 Чорний список**

Додавання користувачів до “чорного списку”: Адміністратор та модератор можуть додати користувачів системи до “чорного списку” за порушення

**2.2.23 Керування користувачами**

Керування користувачами: Адміністратор може видаляти та блокувати користувачів системи.

**2.3 User Characteristics**

Адміністратори платформи: Ці користувачі відповідають за управління і контроль за функціонуванням платформи. Вони мають адміністративні права для створення та редагування маршрутів і місць. Також, вони можуть керувати користувачами сайту та створювати резервну копію бази даних.

Авторизований користувачі: Це туристи, мандрівники та інші особи, які використовують платформу для пошуку туристичних маршрутів, читання інформації про місця, які вони відвідують, і обміну власними враженнями. Користувачі можуть залишати коментарі, писати статті, піднімати свій рейтинг і здобувати досягнення, проходячи маршрути. Однак, вони обмежені у проходженні тільки одного маршруту одночасно. Вони пишуть статті і діляться своїм досвідом з іншими учасниками спільноти.

Неавторизований користувач: Це користувачі, які використовують платформу не маючи облікового запису. Вони можуть переглядати сторінку з поясненнями функціоналу сайту, реєструватися та авторизуватися.

Модератори: Це користувачі, які відповідають за перевірку контенту, який публікується на платформі та керувати користувачами. Вони можуть схвалювати або відхиляти статті, які були відправлені користувачами на перевірку, перевіряти коментарі, залишені під сторінками зі статтями, місцями або маршрутами. Також, вони можуть додавати користувачів до “чорного” списку за порушення правил користування сайту.

**2.4 General Constraints**

Нижче наведені всі обмеження, накладені на технології, що використовуються у розробці продукту:

1. MongoDB для локальної бази даних

MongoDB є документо-орієнтованою базою даних, що дозволяє зберігати дані у вигляді документів JSON. Це забезпечує високу гнучкість у зберіганні даних, дозволяючи легко адаптуватися до змін у структурі даних без потреби у змінах схеми бази даних. База даних використовує індекси для прискорення запитів і забезпечує швидкий доступ до даних. Але оскільки MongoDB є документо-орієнтованою базою даних це означає, що всі дані повинні бути організовані у вигляді документів, що зберігаються в колекціях. Це вимагає належного планування структури даних для забезпечення ефективного зберігання і пошуку інформації.

1. React для клієнтської частини

React дозволяє створювати інтерфейс у вигляді незалежних і повторно використовуваних компонентів. Це спрощує розробку та тестування окремих частин інтерфейсу, а також полегшує підтримку і розширення коду. Використання React забезпечує високу швидкість рендерингу. Це покращує продуктивність і зменшує час завантаження сторінки. Оскільки React базується на концепції компонентів, то вимагає розробки інтерфейсу у вигляді незалежних і повторно використовуваних компонентів. Це обмежує можливість використання інших підходів до структурування коду.

1. ExpressJS для серверної частини

ExpressJS є легким і мінімалістичним фреймворком, який дозволяє швидко налаштувати сервер і створити RESTful API. Це спрощує розробку та підтримку backend частини. ExpressJS забезпечує високу продуктивність завдяки асинхронному обробленню запитів, що дозволяє ефективно обробляти великий обсяг одночасних запитів і забезпечує високу швидкість роботи серверної частини. ExpressJS використовує концепцію middleware для обробки запитів і відповідей. Це вимагає розробки ефективної системи маршрутизації і використання middleware для забезпечення безпеки, обробки помилок і управління сесіями.

**2.5 Assumptions and Dependencies**

AS-1: Застосунок буде корисний для людей, які полюбляють подорожувати.

AS-2: Застосунок полегшить пошук цікавих місць для відвідування під час подорожі, або прогулянки.

AS-3: Застосунок буде корисний для людей, які люблять ділитися враженнями від подорожей.

AS-4: Застосунок буде корисний для людей, які бажають дізнатися більше про видатні місця.

AS-5: Користувачі мають базові навички користування комп'ютером та Інтернетом.

DE-1: Щоб отримати доступ до веб-додатку, користувач повинен мати доступ до інтернету.  
DE-2: Для належної роботи веб-додатку, користувач має використовувати сучасний веб-браузер, який підтримує HTML5 та CSS3.

DE-3: Для оптимізації навігації та функціональності на мобільних пристроях буде потрібно адаптивний дизайн, який забезпечить комфортне користування на різних розмірах екранів.

**3. Specific Requirements**

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

Неавторизовані користувачі бачать сторінку, яка розповідає про можливості сайту:

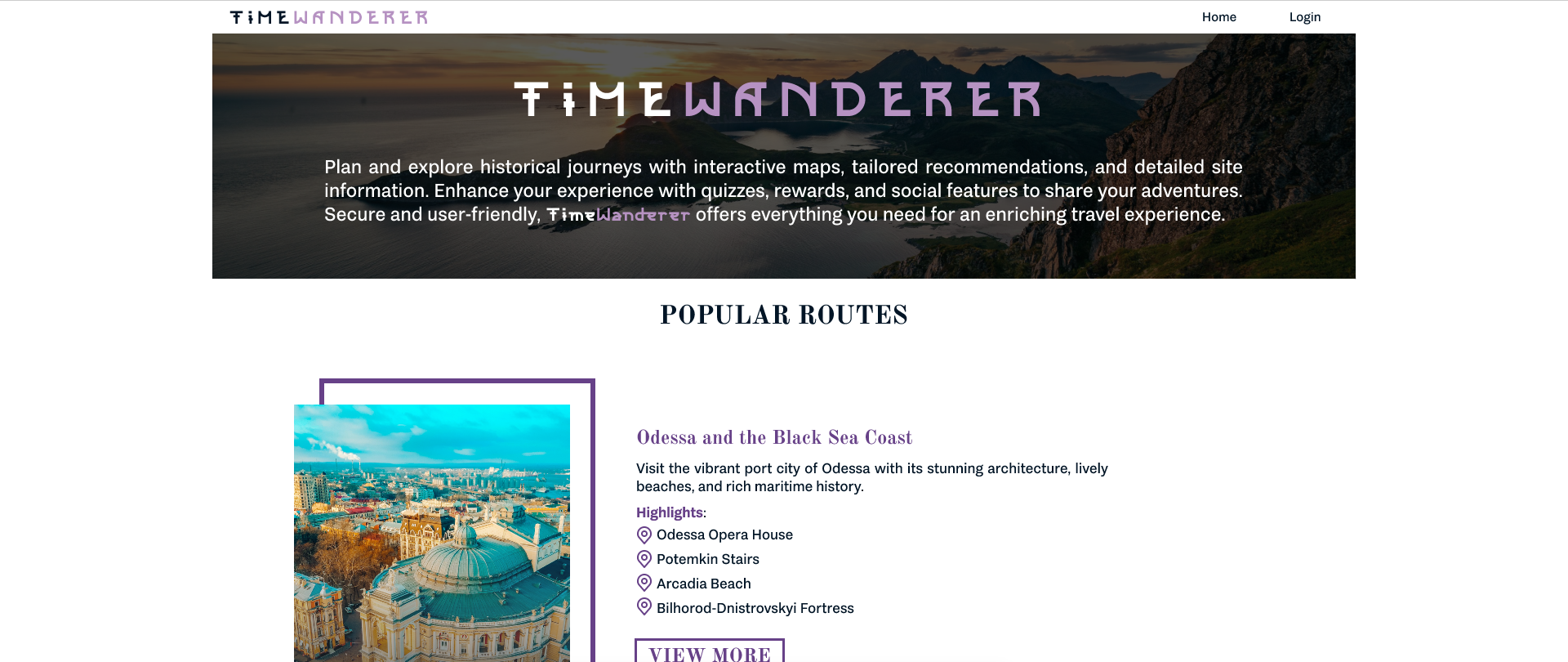


Рисунок 3.1 - Інтерфейс головної сторінки

Не зареєстровані користувачі мають можливість зареєструватися та авторизуватися на сайті.

Користувач може переглядати маршрути, статті та місця. Також, він може писати статті та відправляти їх на модерацію адміністратору. Користувачі можуть залишати коментарі, обмінюючись враженнями з іншими.

Адміністратор має можливість керувати іншими користувачами та даними сайту. Він може погоджувати статті, відправлені на модерацію користувачем. Адміністратор може створювати, редагувати та видаляти маршрути та місця.

**3.1.2 Hardware Interfaces**

Пристрій повинен мати підключений дисплей для відображення користувацького інтерфейсу.

**3.1.3 Software Interfaces**

TomTom API: Використовується для відображення маршрутів на карті, розташування туристичних місць та інших географічних даних. Інтерфейс забезпечує інтеграцію картографічних даних у додаток для покращення користувацького досвіду.

**3.1.4 Communications Interfaces**

Щоб отримати доступ до веб-додатку, користувач повинен мати доступ до інтернету.

**3.2 Functional Requirements**

Функція #1: Пошук та фільтрація маршрутів для проходження

Функція #2: Проходження маршрутів користувачем з відображенням прогресу проходження

Функція #3: Система отримання досягнень та одиниць досвіду за проходження маршрутів

Функція #4: Створення статті та її відправка на модерацію

Функція #5: Перевірка статті модератором для публікації

Функція #6: Створення, редагування та видалення місць адміністратором

Функція #7: Створенння, редагування та видалення маршрутів адміністратором

Функція #8: Написання коментарів під сторінками зі статтями, маршрутами та місцями

Функція #9: Система рекомендацій маршрутів для проходження на сторінці з обраним маршрутом.

Функція #10: Залишення оцінки користувачем після проходження маршруту для формування його рейтингу

Функція #11: Формування та відображення рейтингу користувачів на основі отриманого досвіду

Функція #12: Реєстрація

Функція #13: Авторизація

Функція #14: Обліковий запис, його редагування та видалення

Функція #15: Відновлення паролю

Функція #16: Модерація коментарів та користувачів

Функція #17: Створення облікових записів модераторів адміністратором

Функція #18: Накопичення рівня користувачем під час отримання одиниць досвіду

Функція #19: Резервне копіювання бази даних адміністратором

Функція #20: Експорт даних адміністратором

Функція #21: Додавання користувачів до чорного списку модератором

Функція #22: Керування користувачами адміністратором

Функція #23: Додавання користувачів до друзів для можливості відстежування створених ними статей та пройдених маршрутів

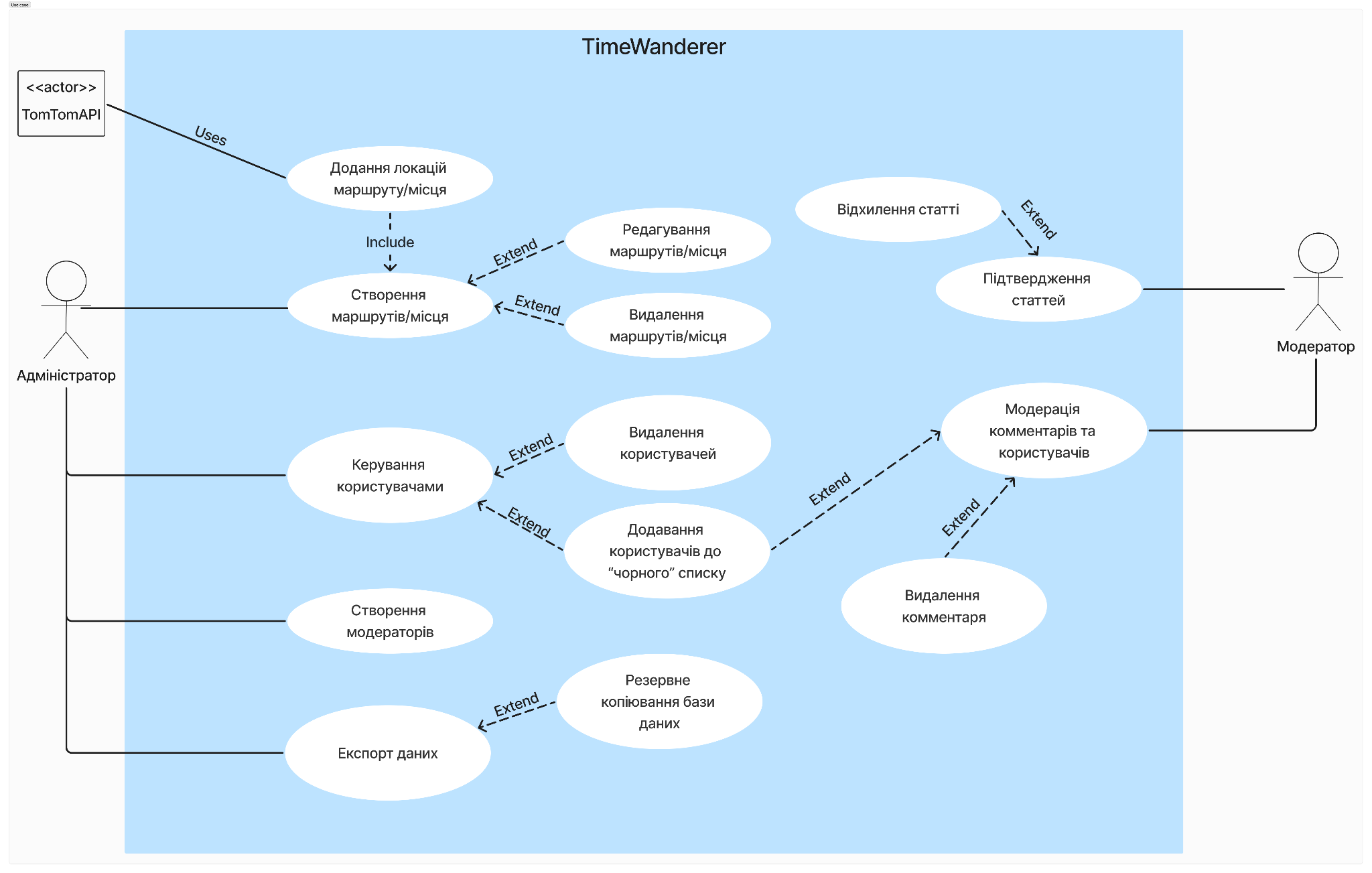
**3.3 Use Cases**

Рисунок 3.2 - Use Case діаграма для адміністратора та модератора

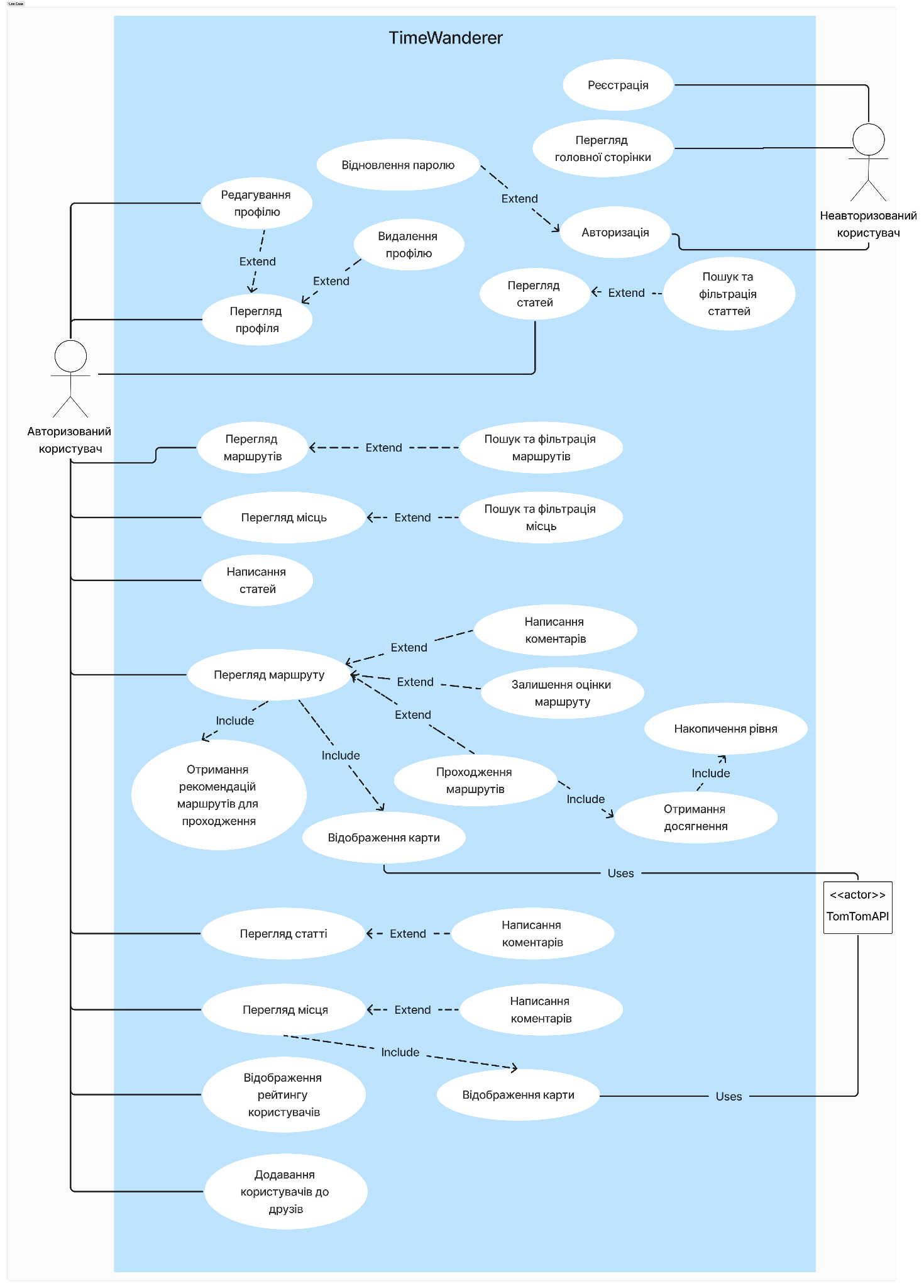


Рисунок 3.3 - Use Case діаграма для авторизованого та неавторизованого користувачів

**3.4 Classes / Objects**

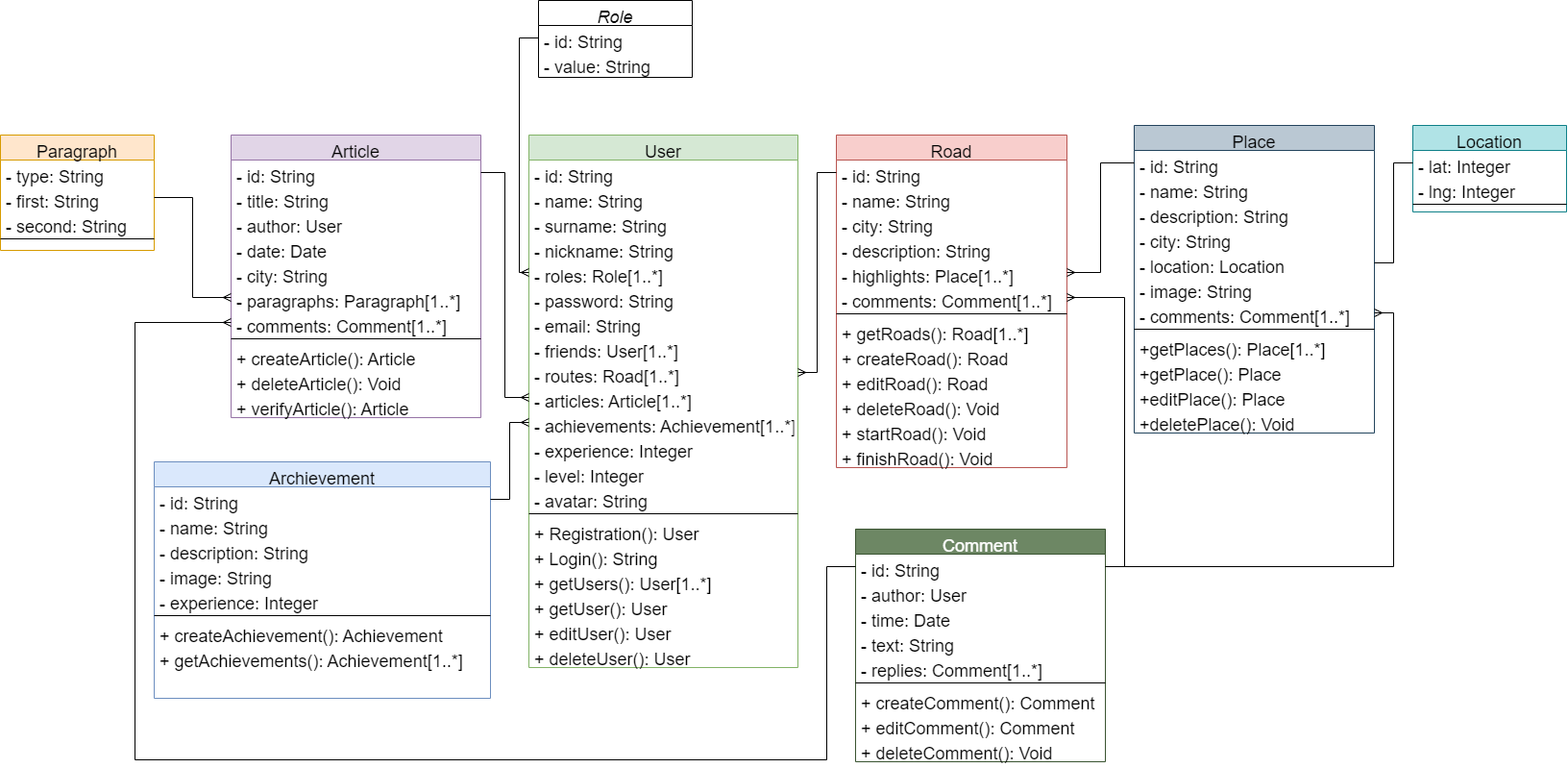


Рисунок 3.4 - Діаграма класів

**3.5 Non-Functional Requirements**

**3.5.1 Performance**

1. Веб-додаток має бути сумісний з основними браузерами, зокрема Google Chrome, Mozilla Firefox, Safari та Microsoft Edge.

**3.5.2 Reliability**

1. Система повинна залишатися стабільною та доступною навіть у випадку введення некоректних або помилкових даних користувачами.

2. У разі збою або помилки система повинна забезпечити коректне відновлення або повідомити користувача про проблему без втрати критичних даних.

**3.5.3 Availability**

1. Кожен зареєстрований користувач повинен мати можливість переглядати та коментувати маршрути, а також додавати відгуки та створювати статті.

2. Всі функції додатку повинні бути доступні користувачам з різних країн, які мають доступ до Інтернету.

3. Інтерфейс додатку має бути локалізований для підтримки основних мов, включаючи англійську та українську.

**3.5.4 Security**

1. Особисті та конфіденційні дані користувачів повинні бути захищені від несанкціонованого доступу.

2. Паролі користувачів повинні зберігатися в зашифрованому вигляді.

3. Користувачі не повинні мати можливості впливати на роботу серверної бази даних або змінювати дані інших користувачів.

4. Система повинна інформувати користувачів про введення некоректних даних через сповіщення або повідомлення про помилки.

**3.5.5 Maintainability**

1. Додаток повинен бути легко розширюваним для додавання нових функцій та можливостей.

**3.5.6 Portability**

1. Інтерфейс додатку повинен адаптуватися під різні розміри екранів і орієнтації пристроїв, включаючи смартфони.

**3.6 Inverse Requirements**

Продукт не потребує особливих вимог.

**3.7 Design Constraints**

Проектування містить певні функції:

1. Дизайн повинен бути мінімалістичний.

2. Дизайн використовує шрифт Passengers Sans.

3. Дизайн повинен бути інтуїтивно зрозумілий користувачу.

4. Сторінки повинні відображати базовий функціонал.

5. Інтерфейс повинен бути адаптивним для різних розмірів дисплею.

5. Інтерфейс має власну кольорову схему:

1. Колір #FFFFFF - це колір заднього фону.
2. Колір #674188 використовується для заголовків та іконок.
3. Колір #B692C2 використовується для заливки кордонів зображень, кнопок.
4. Колір #C7B7DA використовується для деяких частин тексту.
5. Колір #021526 використовується як основний колір тексту.

**3.8 Logical Database Requirements**

3.8.1 Types of Information Used by Various Functions

- Інформація користувача:

* Опис: Дані користувача включають його логін, електронну пошту, пароль, ім’я, рівень, отримані одиниці досвіду, пройдені маршрути, написані статті та його ID.
* Відповідність функціям: Дані користувача важливі для таких функцій, як відображення профіля користувача, формування рейтингу користувачів, системи додавання друзів та залишення коментарів.

- Інформація маршруту:

* Опис: Дані маршруту включають його назву, місця для відвідування, опис та його ID.
* Відповідність функціям: Дані маршруту важливі для таких функцій, як відображення маршруту, проходження маршруту, пошук та фільтрація підходящих маршрутів.

- Інформація місця:

* Опис: Дані місця включають його назву, опис, координати та його ID.
* Відповідність функціям: Дані місця важливі для таких функцій, як відображення інформації про місце, відображення карти з вказівкою місця, побудова карт маршрутів.

3.8.2 Frequency of Use

- Очікується, що дані про користувачів найчастіше запитуються, використовуються та змінюються. Користувачі часто проходять маршрути, що призводить до змін у їхньому рейтингу, який потрібно відповідно оновлювати. Крім того, адміністратори періодично запитують дані користувачів для затвердження їх статті, або для додавання їх до чорного списку.

3.8.3 Accessing Capabilities

- Сервіс надає інтерфейс REST API для надання користувачам доступу до даних.

Приклади ендпоінтів:

* Place endpoints:
  + /places/ (GET)
  + /places/{place\_id}/ (GET)
  + /places/{place\_id}/ (PUT)
  + /places/{place\_id}/ (DELETE)

* User endpoints:
  + /users/profile/ (GET)
  + /users/profile/ (PUT)
  + /users/profile/(DELETE)
  + /auth/registration/ (POST)
  + /auth//login/ (POST)

* Road endpoints:
  + /roads/ (GET)
  + /roads/{road\_id}/ (GET)
  + /roads/{road\_id}/ (PUT)
  + /roads/{road\_id}/ (DELETE)

3.8.4 Integrity Constraints

Унікальні обмеження:

* Жодна з сутностей не може мати однакову назву. Наприклад, не може існувати двох маршрутів з однаковими назвами, як "Таємничі місця Харкова".
* Місця не можуть мати однакову назву.
* Статті не можуть мати однаковий заголовок або назву.

Обмеження валідації:

* Електронна адреса користувача повинна бути дійсною (наприклад, повинна містити символ “@”).
* Пароль користувача повинен містити щонайменше 6 символів.
* Користувач не може розпочати новий маршрут, якщо він вже має незавершений.

Обмеження кількості сутностей:

* Маршрут повинен містити принаймні 2 місця, пов'язані з ним.
* Стаття повинна містити принаймні 1 абзац тексту.

Обмеження доступу:

* Користувач повинен мати певну роль для виконання специфічних дій на платформі (наприклад, адміністратори можуть створювати та редагувати маршрути, а звичайні користувачі не мають таких прав).

3.8.5 Data Retention Requirements

- Майже всі дані зберігаються у документо-орієнтованій базі даних MongoDB. Вони можуть бути змінені користувачем, або адміністратором у потрібний момент. Наприклад, коли користувач вирішив змінити власний логін, або зовсім видалити свій обліковий запис.

3.8.6 Data Models (if provided by the customer)

* Основними моделями даних мають бути:
* User (Користувач);
* Place (Місце);
* Road (Маршрут);
* Article (Стаття);
* Comment (Коментар);
* Achievement (Досягнення);

3.8.7 Database Technologies

- Як основне сховище даних обрано MongoDB. Як сховище файлів, була обрана Firebase Storage.

**3.9 Other Requirements**

Немає інших вимог.

**4. Analysis Models**

**4.1 Sequence Diagrams**

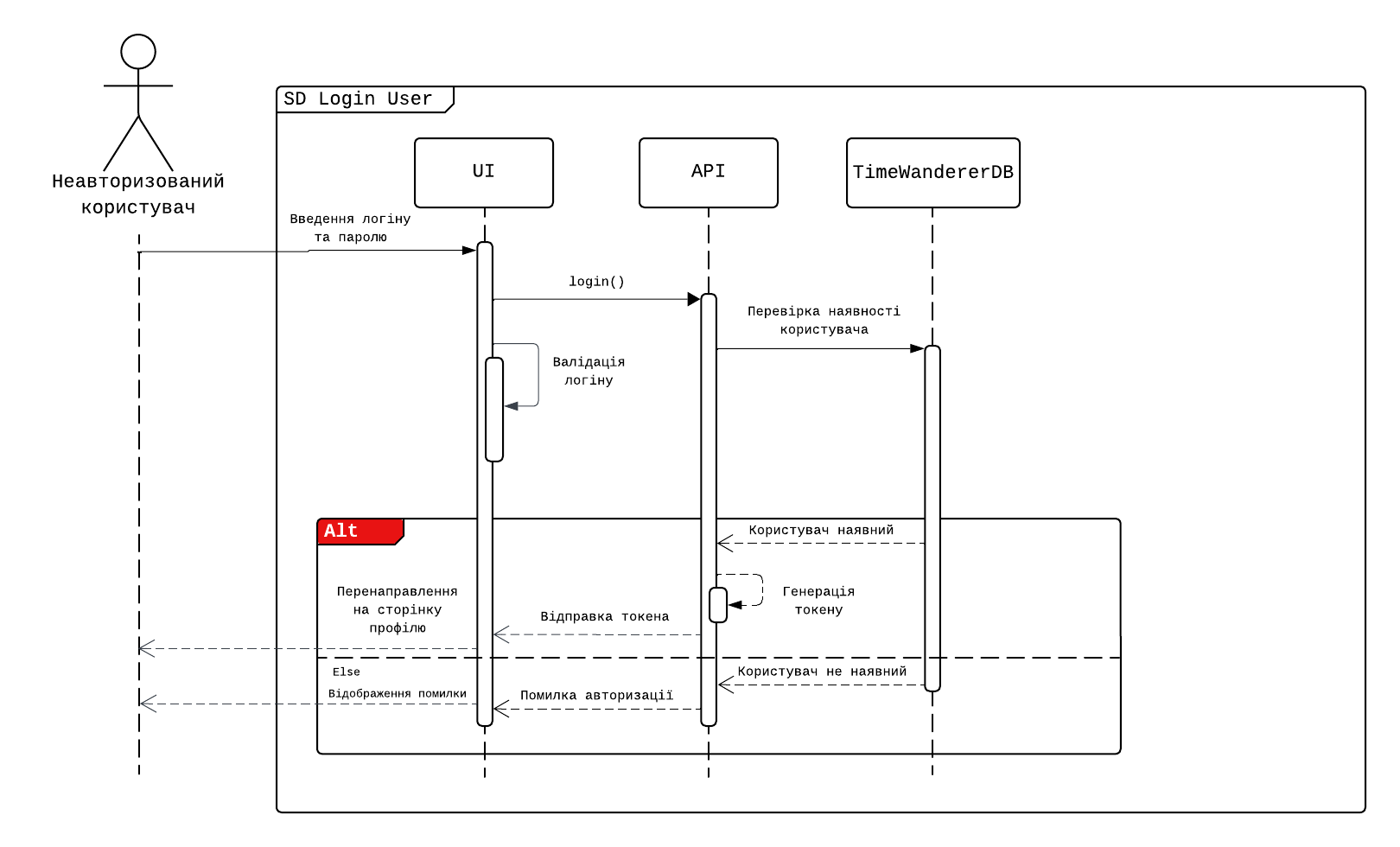
****

Рисунок 4.1 - Авторизація користувача

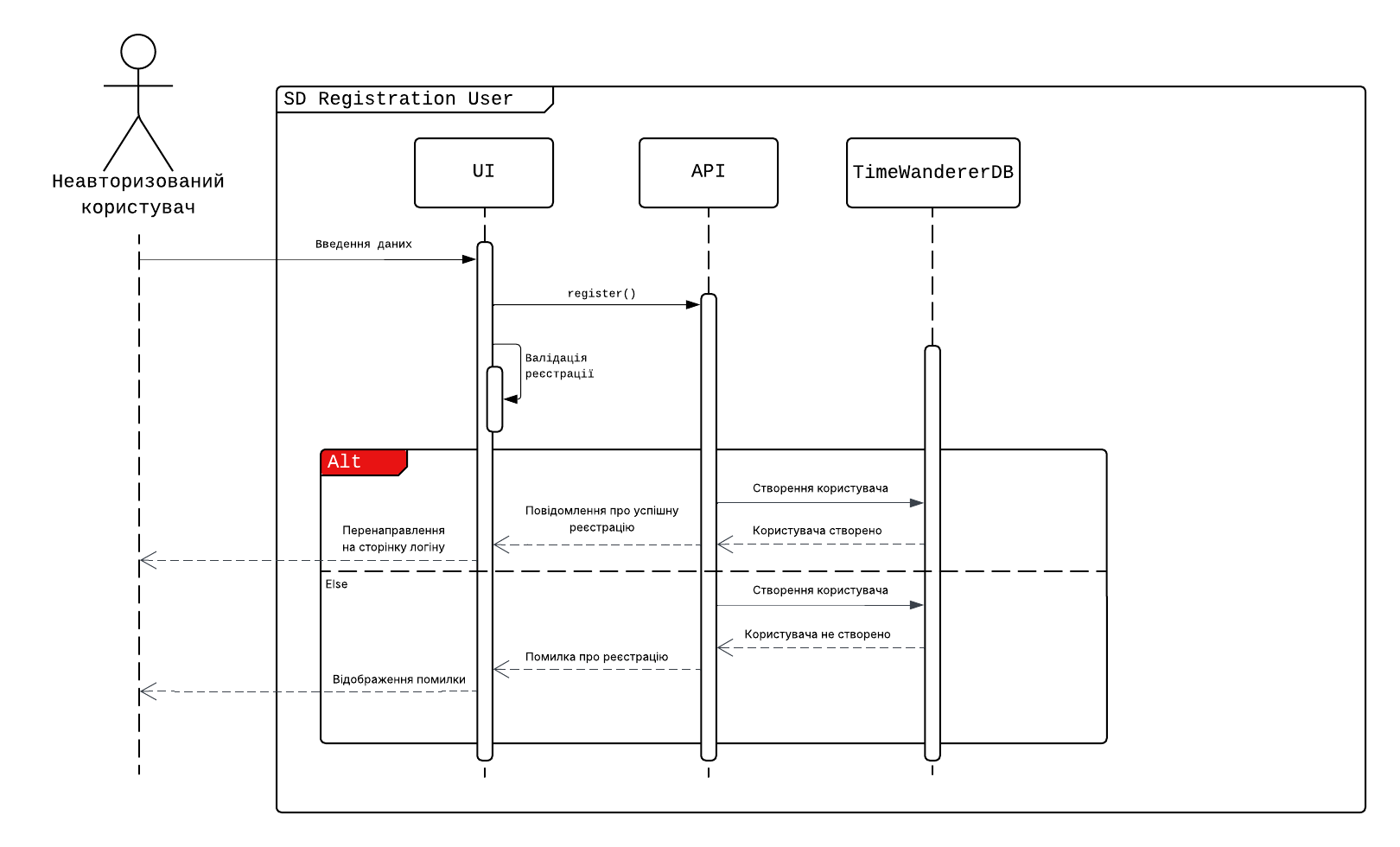


Рисунок 4.2 - Реєстрація користувача

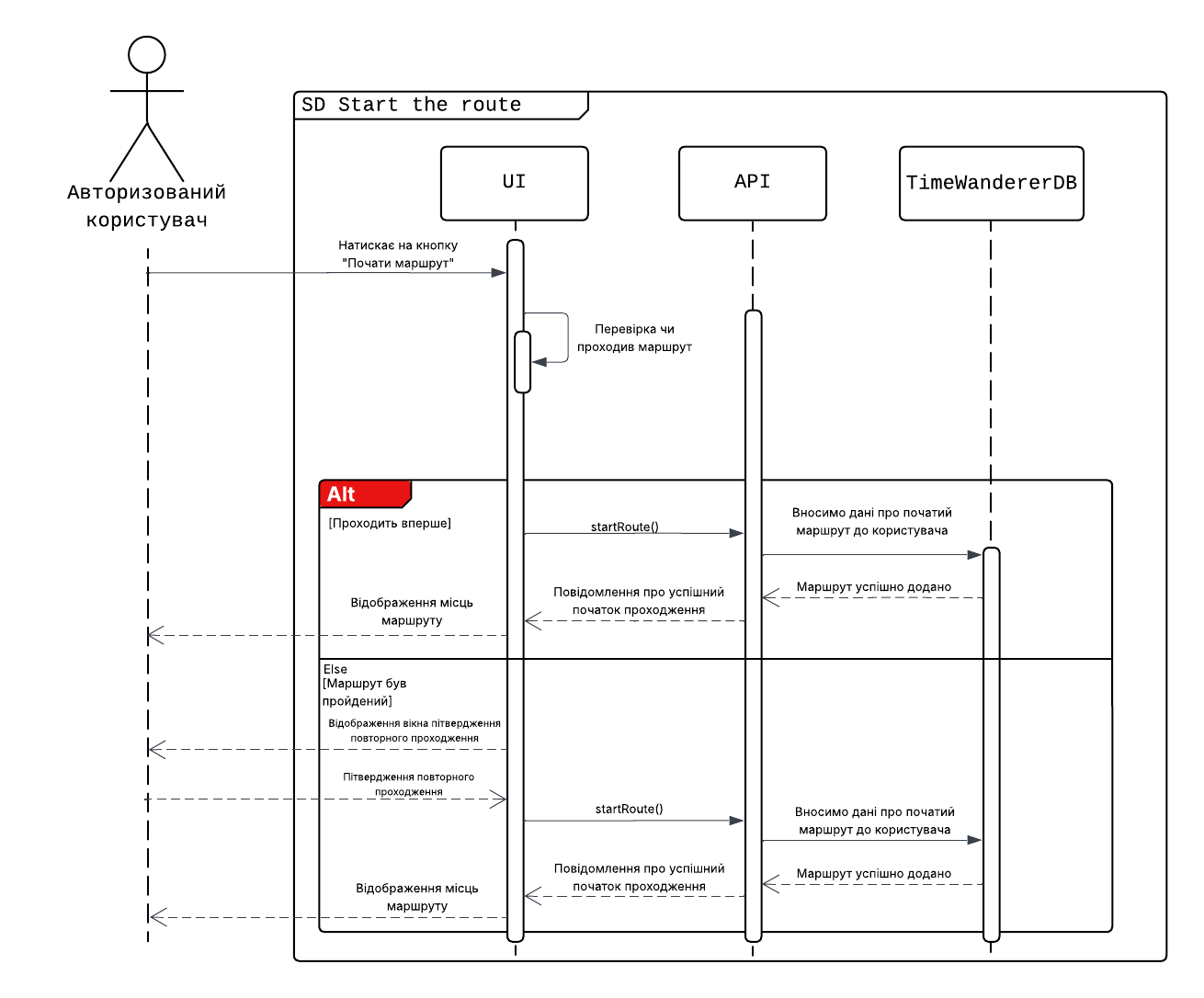


Рисунок 4.3 - Початок проходження маршруту

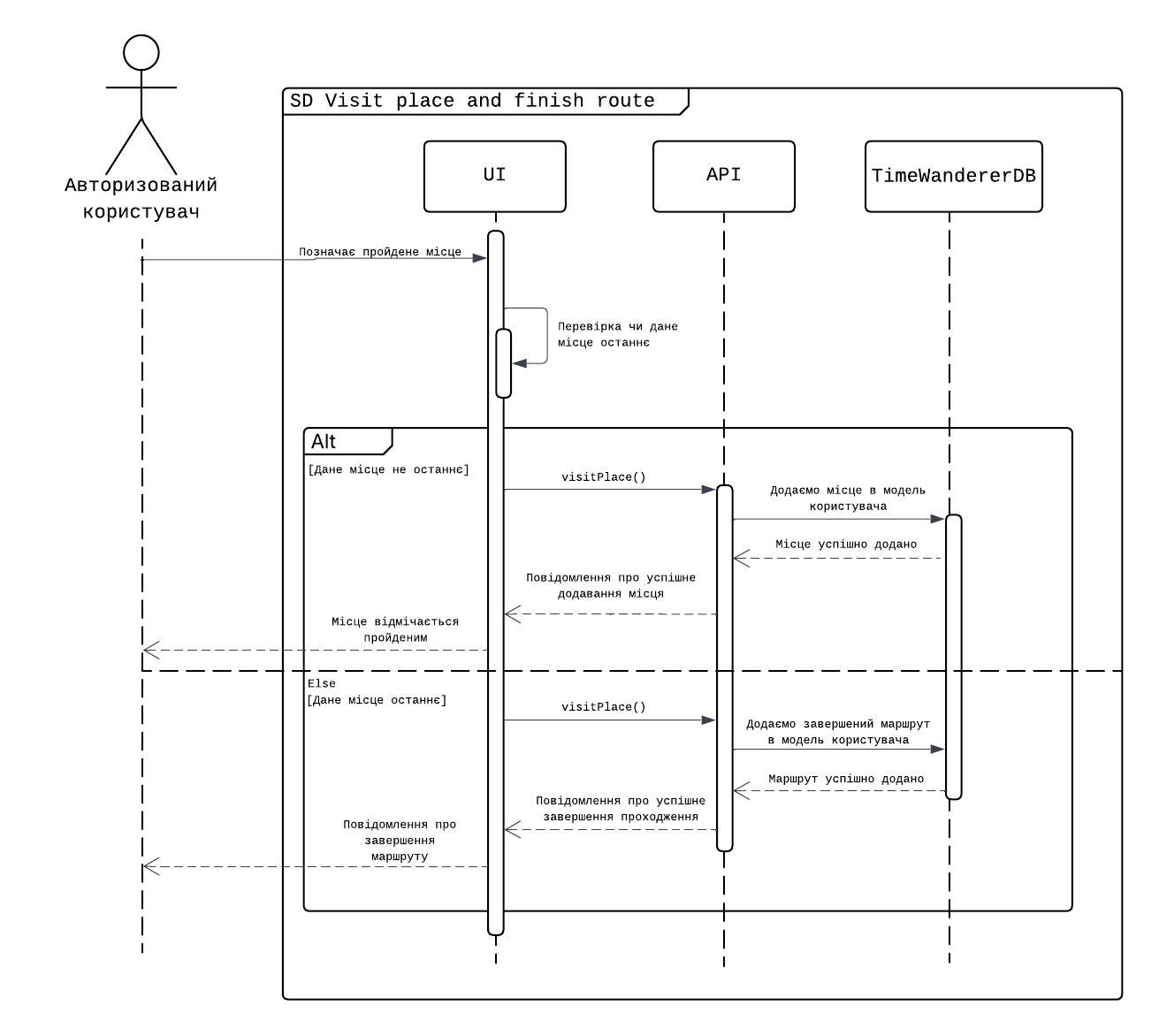


Рисунок 4.4 - Відмічення пройденого місця і завершення маршруту

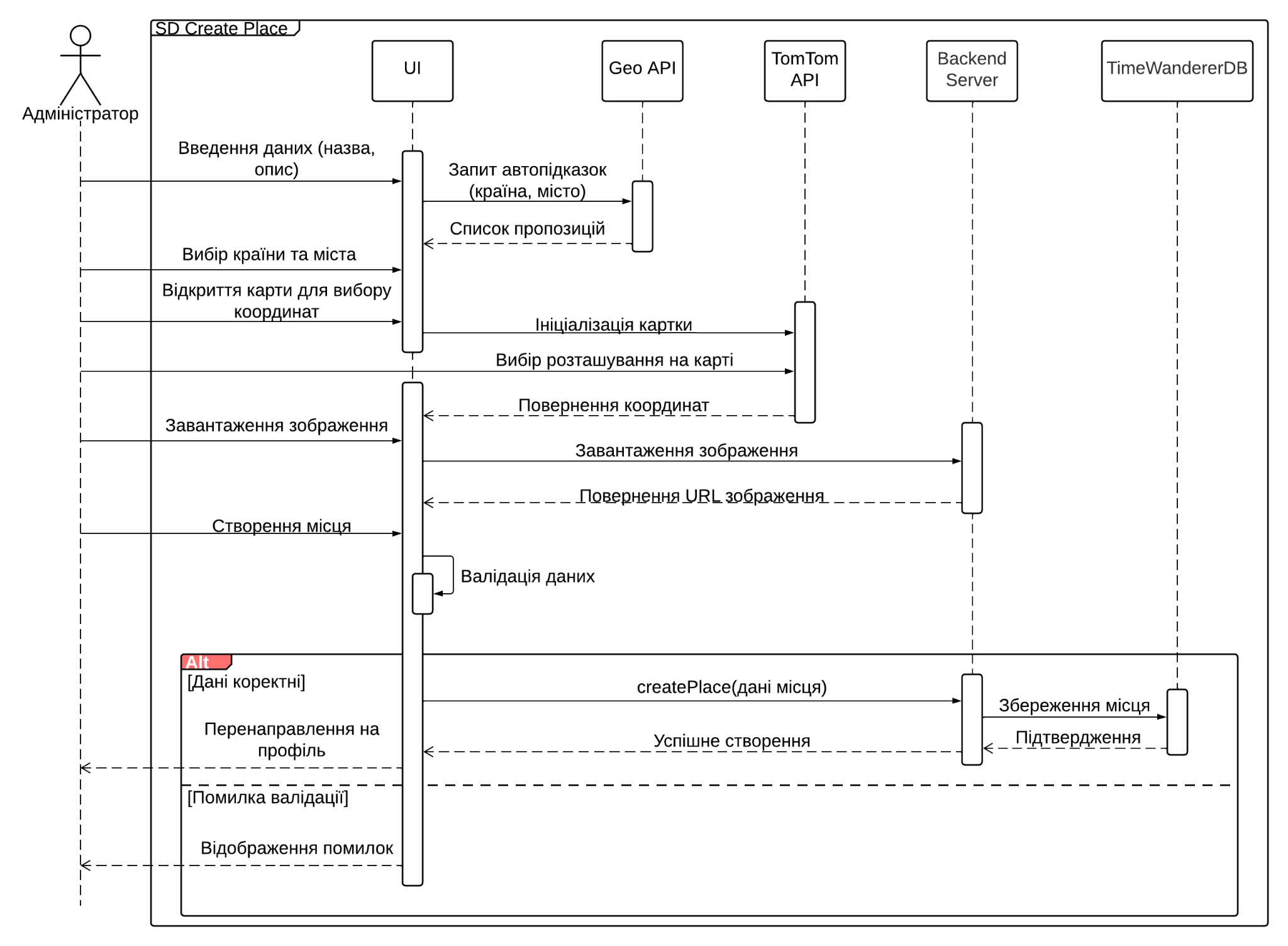


Рисунок 4.5 - Створення місця адміністратором

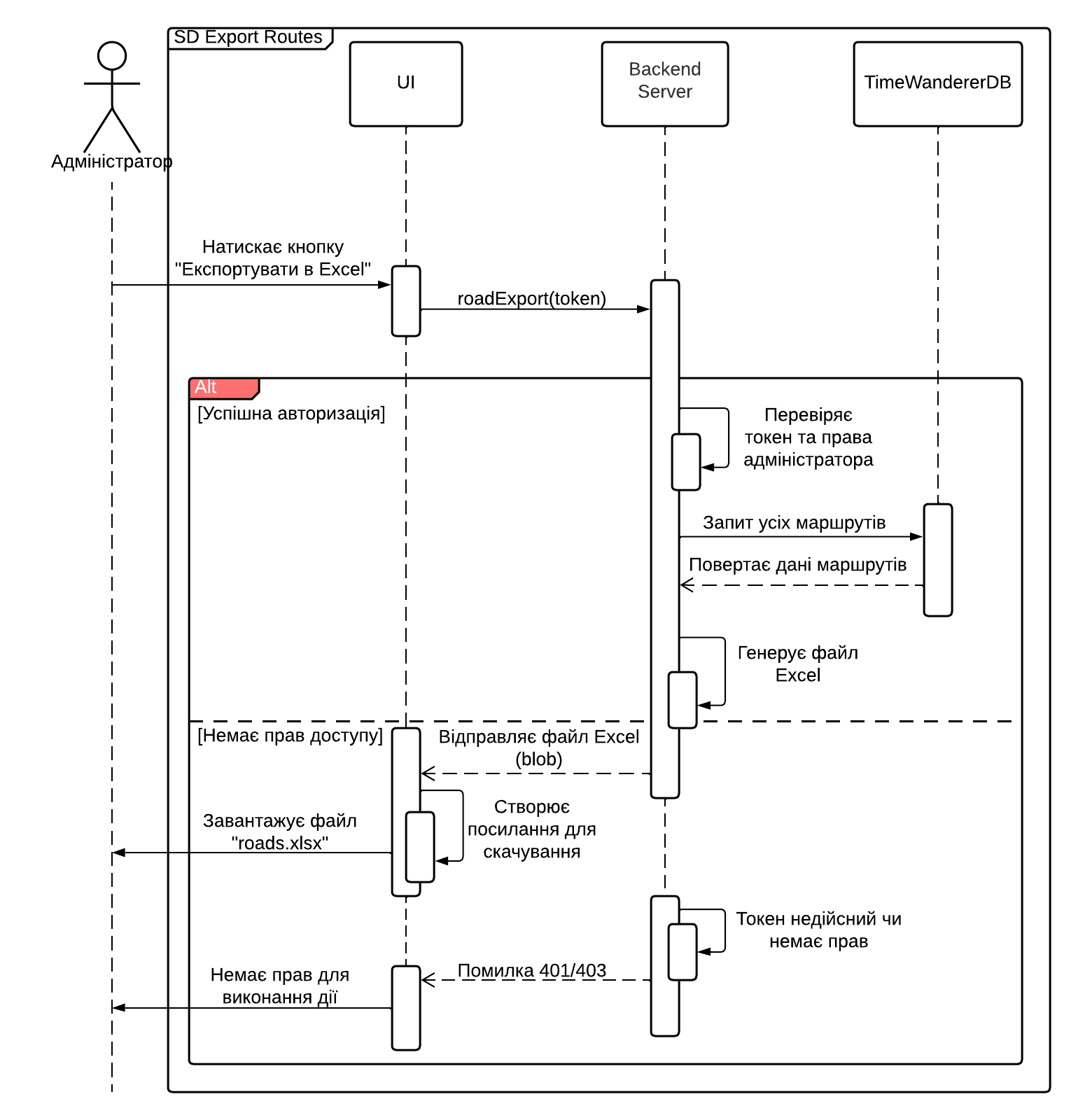


Рисунок 4.6 - Експорт інформації про маршрути адміністратором

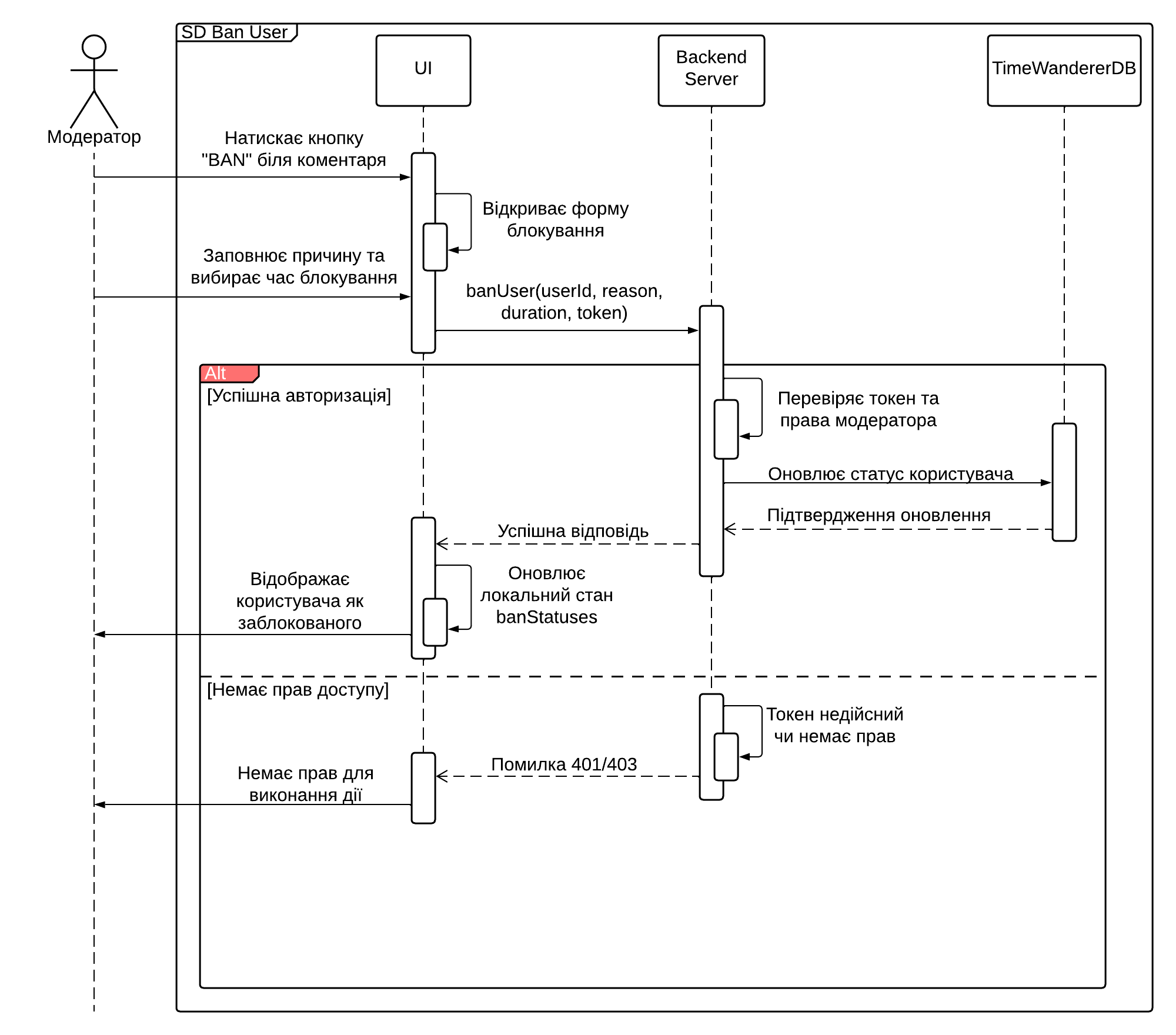


Рисунок 4.7 - Блокування користувача модератором

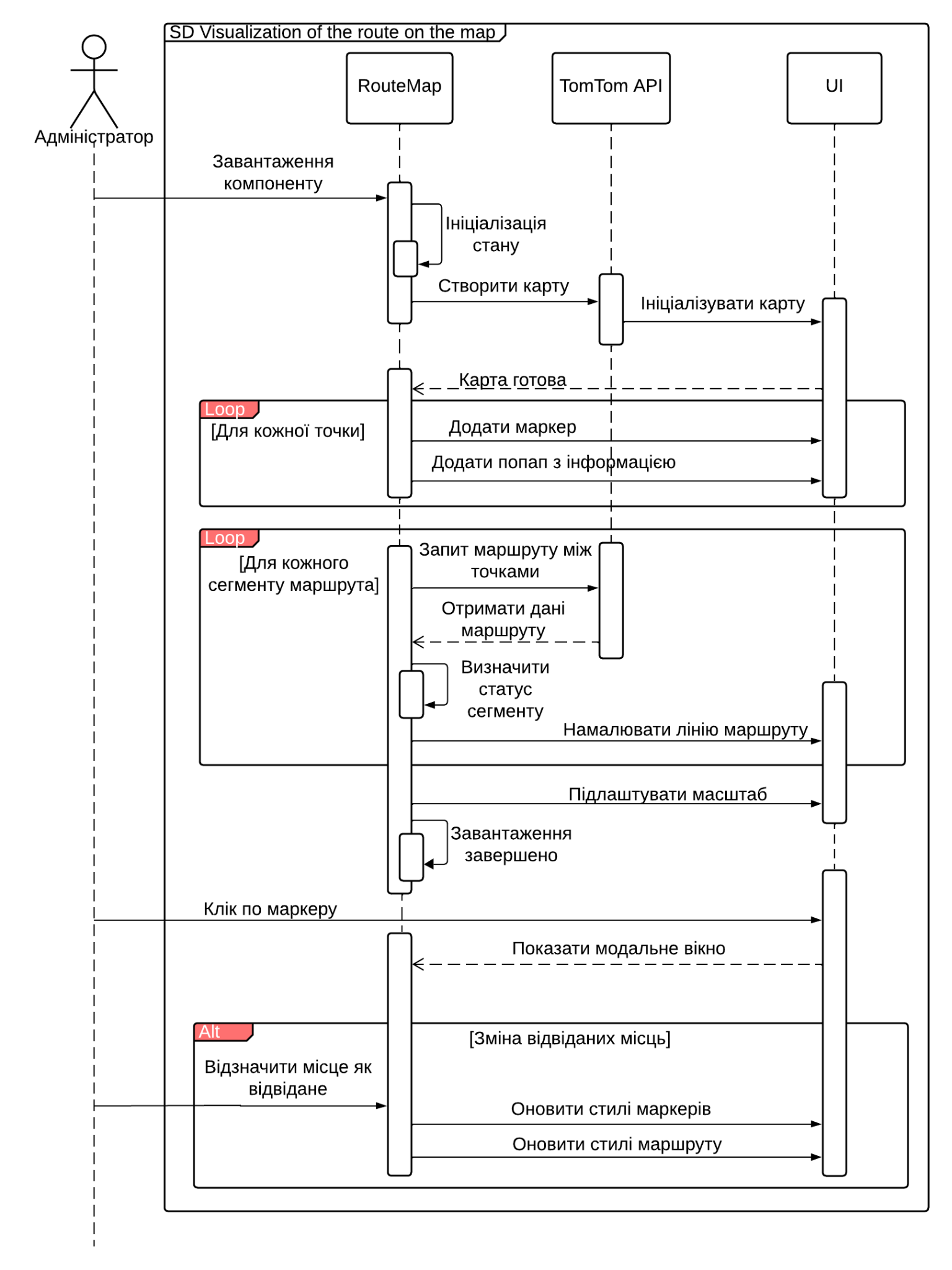


Рисунок 4.8 - Відображення марушруту на карті

**4.2 User Flow Diagram**

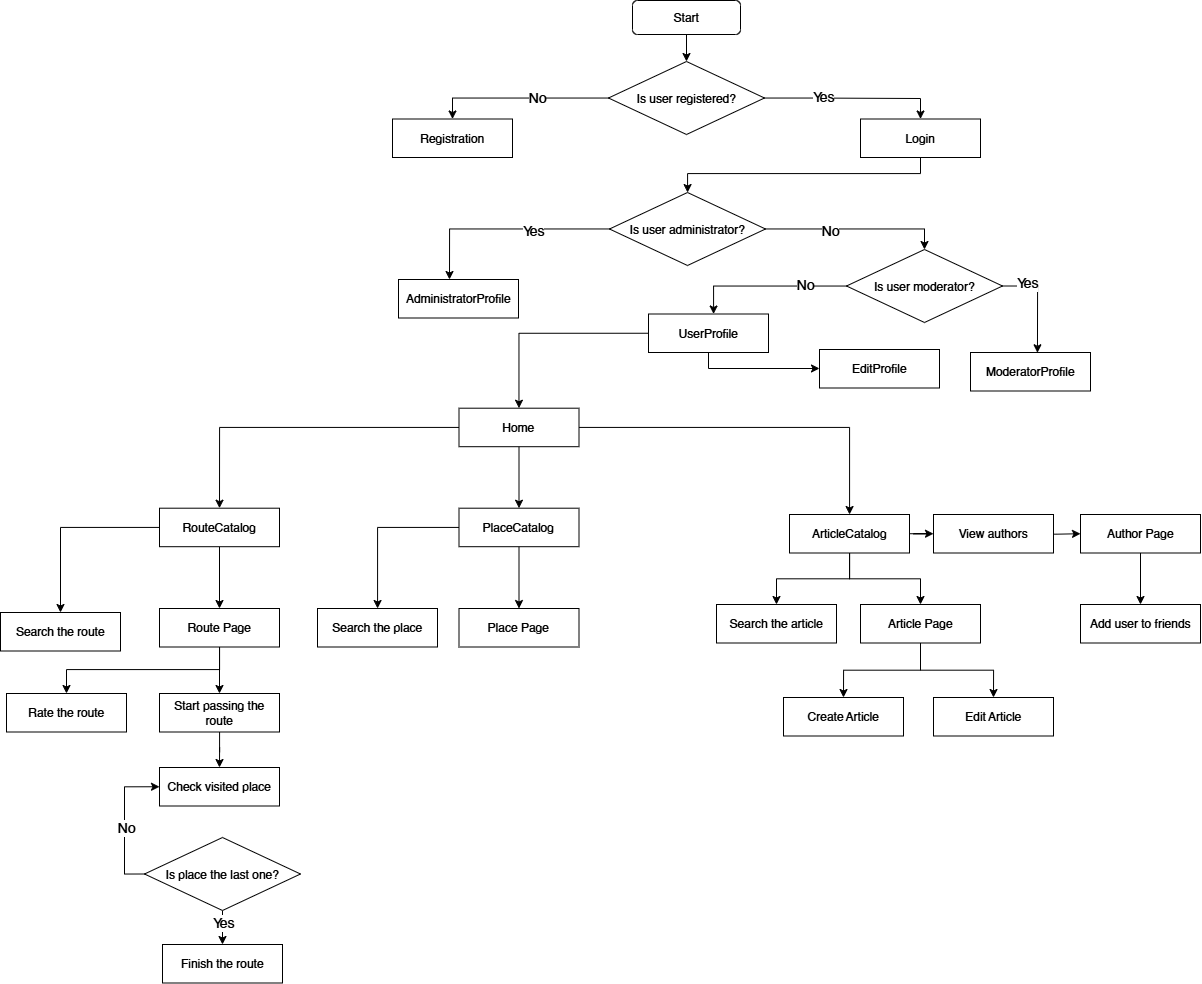
****

Рисунок 4.9 - UserFlow діаграма для клієнстької частини

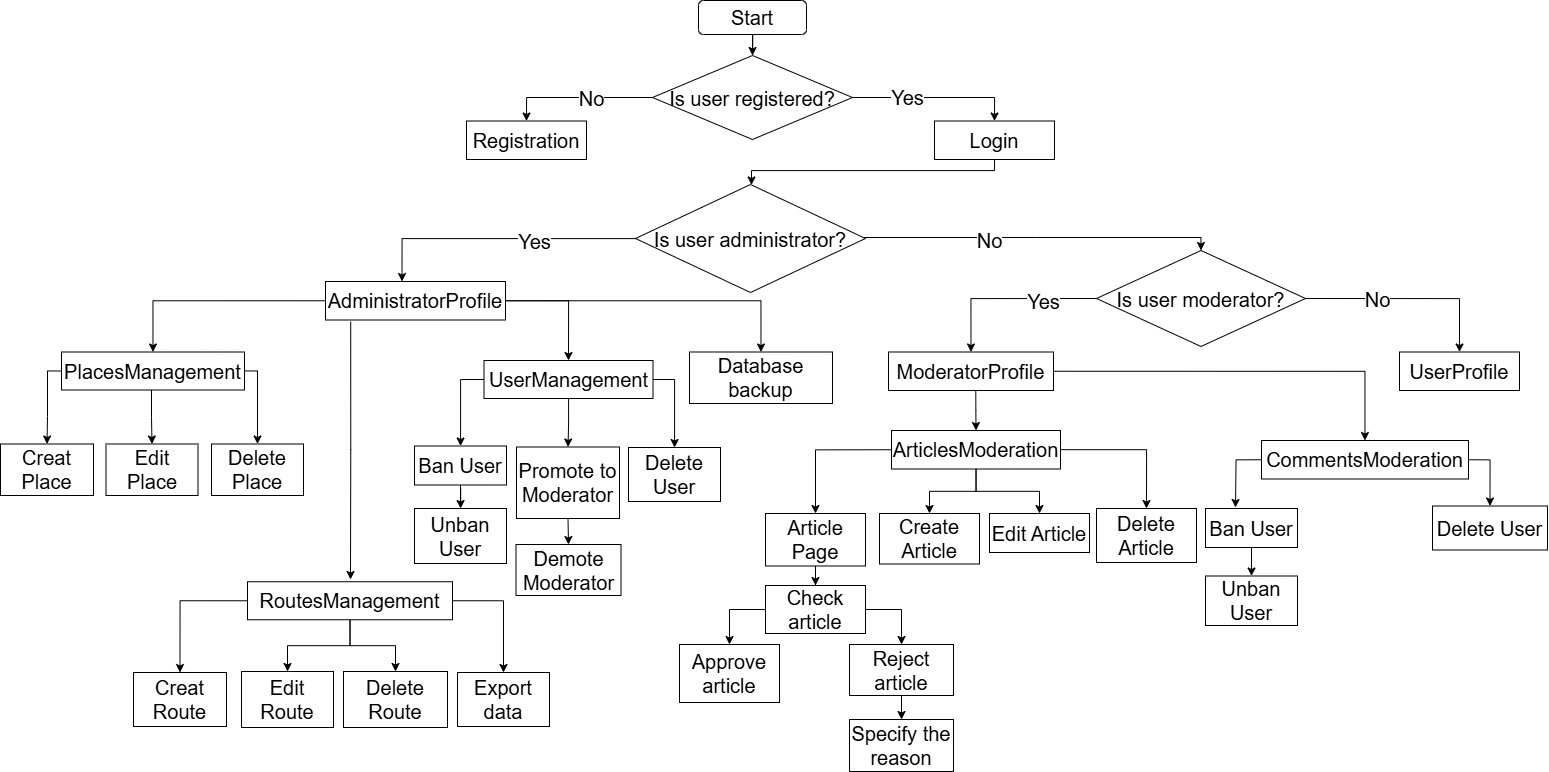
****

Рисунок 4.10 - User Flow для адміністративно-модераторської частини

**4.3 ER-Diagram**

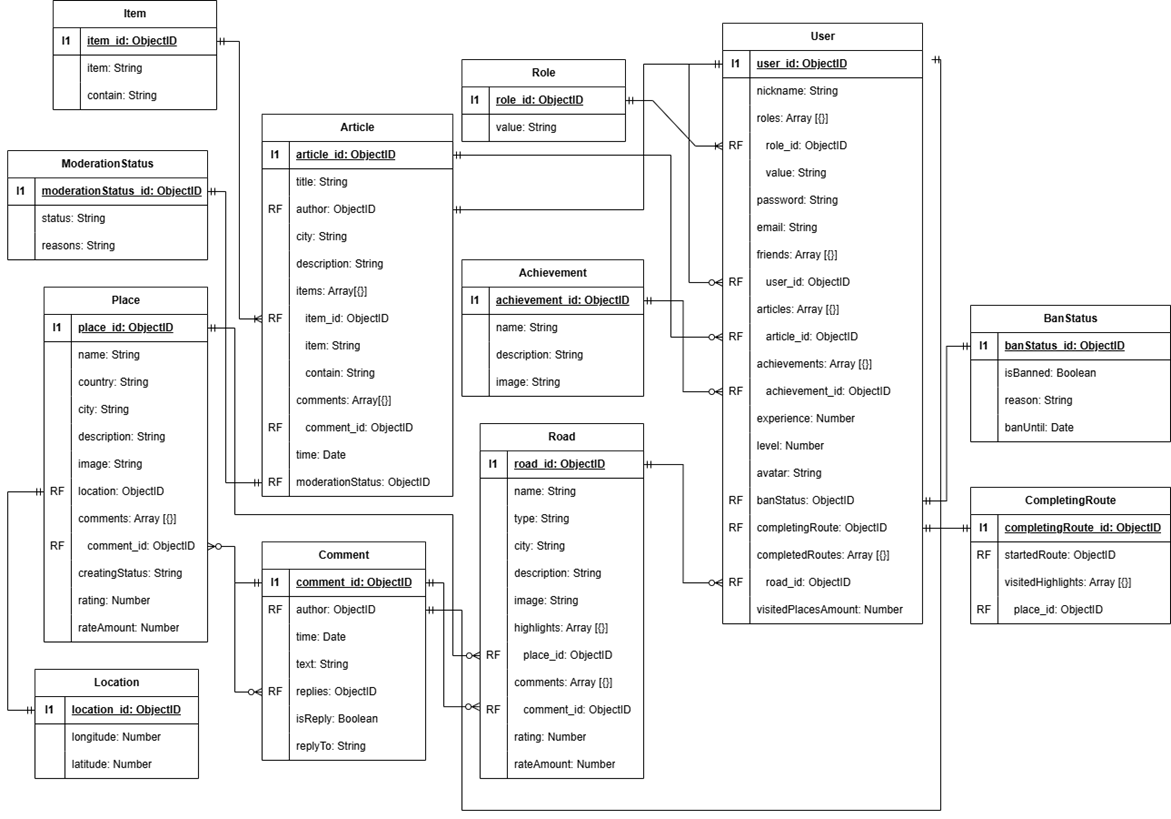
****

Рисунок 4.11 - ER-діаграма

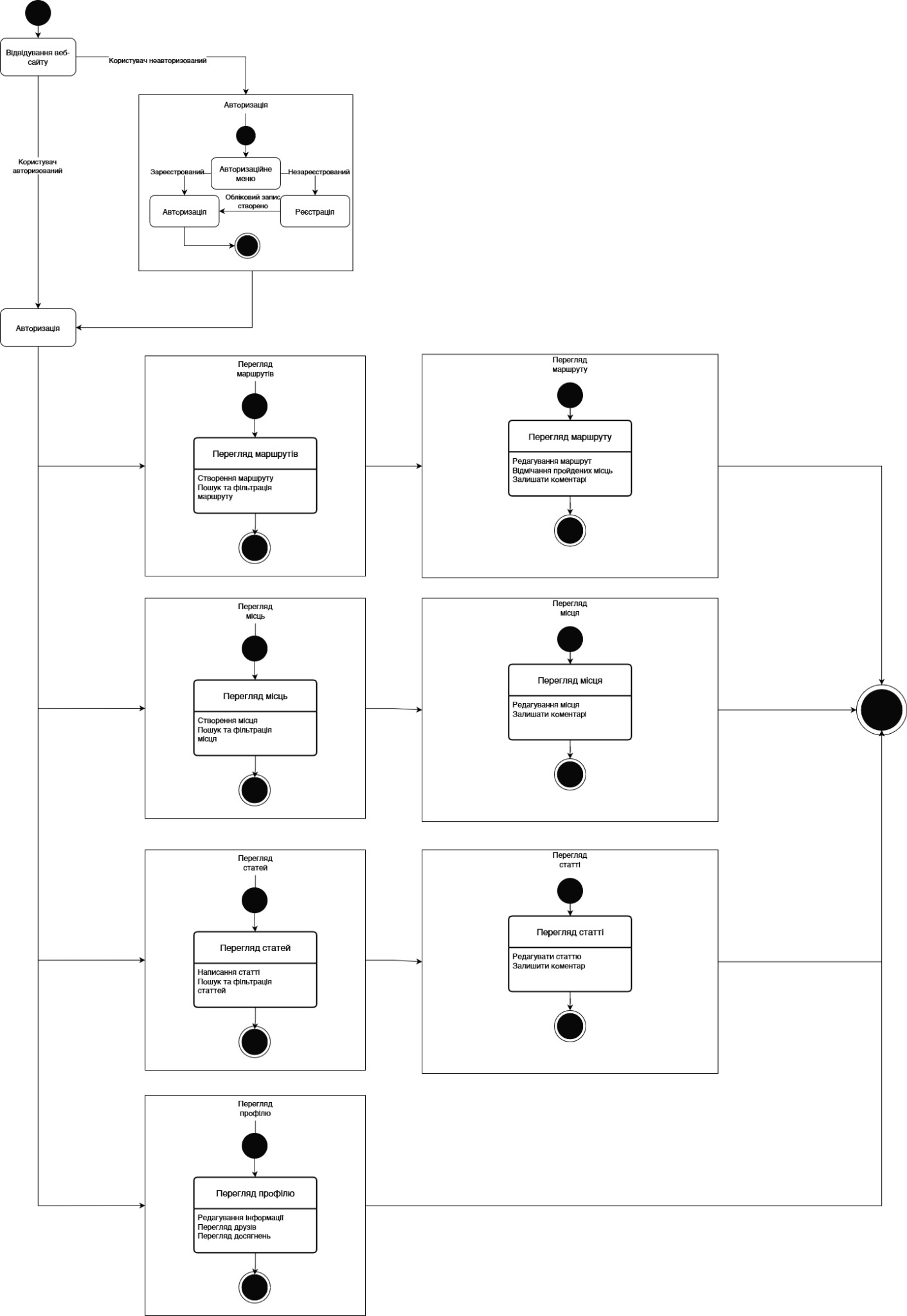
**4.4 State-Transition Diagrams (STD)**

Рисунок 4.12 - State-Transition діаграма

**5. Change Management Process**

Коли обсяг проекту або вимоги змінюються, документація SRS оновлюється.