GoodTrip

Специфікація вимог до програмного забезпечення

Версія 1.0

Телюк Ілля Олександрович

Танасійчук Владислав Євгенійович

Бургард Іван Костянтинович

27.05.2025

# Історія версій

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Опис | Автор | Коментар |
| 27.05.2025 | Версія 1.0 | Ілля Телюк | Створення документу |
| 29.05.2025 | Версія 1.1 | Ілля Телюк, Владислав Танасійчук, Іван Бургард | Додавання нових пунктів |
|  |  |  |  |
|  |  |  |  |

# Зміст

[Історія версій 2](#_ds17lzrzxt2c)

[Зміст 3](#_26o7x7b4w8vm)

[1 Вступ 4](#_nmroqhtwf51m)

[1.1 Огляд продукту 4](#_57htb13z8zpr)

[1.2 Мета 4](#_9rxzdsze0xks)

[1.3 Межі 4](#_9fcicc94t2tx)

[1.4 Посилання 5](#_jlvpdvbn2a4d)

[1.5 Означення та абревіатура 5](#_9cdr88t3yh4y)

[2 Загальний опис 6](#_x8ql1d9t703c)

[2.1 Перспективи продукту 6](#_uygz42f9kvot)

[2.2 Функції продукту 7](#_fht1ku8xszp)

[2.3 Характеристика користувачів 8](#_wivdm7wa7gb4)

[2.4 Загальні обмеження 8](#_8a8r19aommva)

[2.5 Припущення та залежності 9](#_ca5sdpu3j1wr)

[3 Конкретні вимоги 10](#_k5bzdjprlcpf)

[3.1 Вимоги до зовнішніх інтерфейсів 10](#_w5it2rzdutwo)

[3.1.1 Інтерфейс користувача 10](#_6tuuopdhzge8)

[3.1.2 Інтерфейс користувача 12](#_idbtevz3al5z)

[3.1.3 Програмний інтерфейс 12](#_fcvn4shc84je)

[3.1.4 Комунікаційний протокол 13](#_hj08wf8mc1u9)

[3.1.5 Обмеження пам’яті 13](#_31mi92vv4bo)

[3.1.6 Операції 13](#_zltya5a3m8k)

[3.1.7 Функції продукту 13](#_opys32pizqwq)

[3.2 Вимоги до зовнішніх інтерфейсів 14](#_hrrln2dgyniv)

[3.3 Атрибути програмного продукту 15](#_r8r93vxg2v4z)

[3.3.1 Надійність 15](#_81cm6wekikct)

[3.3.3 Безпека 15](#_nt7eqvgl6c7t)

[3.3.4 Супроводжуваність 16](#_q5tbzzn4jedo)

[3.3.5 Переносимість 16](#_erltk69hj2f8)

[3.3.6 Продуктивність 16](#_oqe0qa1phihi)

[3.4 Вимоги бази даних 16](#_e46omfeoa0cu)

[3.4 Інші вимоги 17](#_d8ul64bncq5i)

[4 Додаткові матеріали 18](#_urxmrrn02nn6)

[4.1 Схеми баз даних 18](#_3c6j4m3o6suq)

[4.2 Діаграма прецедентів 18](#_gv5jjn26dzq8)

# 

# 1 Вступ

## 1.1 Огляд продукту

GoodTrip — це сучасна вебплатформа, яка дозволяє користувачам отримувати персоналізовані туристичні рекомендації на основі вподобань, історії подорожей, соціальної активності та поточних умов. Продукт орієнтований на взаємодію між користувачами та забезпечує зручне керування маршрутами, відгуками та профілями. Адміністратори платформи отримують широкі можливості для модерації та управління даними.

1.2 Мета

Метою розробки програмного забезпечення GoodTrip є створення функціонального вебзастосунку, який дозволяє користувачам отримувати персоналізовані рекомендації туристичних локацій на основі вказаних уподобань, поточного місцезнаходження та відстані до об’єктів. Система повинна забезпечити просту, зручну та ефективну взаємодію користувача з платформою — від реєстрації до збереження маршрутів і взаємодії з іншими мандрівниками.

Рекомендації формуються за допомогою спеціального модуля на Python, який обробляє інформацію про:

* категорії локацій, обрані користувачем у профілі;

## геолокацію користувача (через браузер або вибране місто/регіон);

## дистанцію до туристичних об'єктів, із фільтрацією за радіусом пошуку.

Основні цілі системи:

* автоматизований підбір локацій — надання актуальних рекомендацій з урахуванням інтересів та відстані, без потреби вручну переглядати сотні варіантів.
* зручний доступ до туристичної інформації — інтерфейс дозволяє переглядати фото, описи, рейтинги, обрані списки.
* можливість взаємодії з контентом — залишення відгуків, оцінювання локацій, коментування.
* профіль користувача — редагування даних, вподобань, обраного, маршрутів.
* соціальні функції — підписки, перегляд профілів, скарги.
* інструменти для адміністратора — повне керування контентом та користувачами через адміністраторський інтерфейс.

Архітектура платформи передбачає поділ на окремі підсистеми:

* Клієнтська частина (React, TypeScript, Zustand, i18next, Tailwind CSS, MUI, Shadcn);
* Серверна частина (ASP.NET + Clean Architecture, EF Core, MediatR, AWS Cognito/S3/SQS);
* Рекомендаційна підсистема (FastAPI, pandas, numpy, scikit-learn).

Таким чином, ExplorerAI має на меті не лише покращити користувацький досвід під час планування подорожей, а й створити основу для майбутнього масштабування функціоналу — наприклад, за рахунок більш глибокого аналізу поведінки, відгуків, сезонності та соціальних факторів.

## 1.3 Межі

Програмна система GoodTrip — це сучасна вебплатформа, що надає користувачам персоналізовані туристичні рекомендації на основі їхніх уподобань, поточного місцезнаходження та радіусу пошуку. Платформа орієнтована на зручну взаємодію між користувачами, дає змогу переглядати локації, формувати власні маршрути, зберігати улюблені місця, залишати відгуки та підписуватися на інших користувачів. Уся взаємодія здійснюється через односторінковий вебзастосунок з адаптивним і багатомовним інтерфейсом.

Система підтримує два основні типи користувачів — звичайних користувачів і адміністраторів:

* користувачі мають змогу самостійно створити обліковий запис, налаштувати вподобання (категорії локацій), отримувати персоналізовані рекомендації, взаємодіяти з контентом та іншими користувачами;
* адміністратори отримують доступ до окремої адмін-панелі, де можуть модерувати відгуки, обробляти скарги, управляти локаціями та категоріями, а також блокувати або редагувати облікові записи користувачів.

Особливістю GoodTrip є рекомендаційна підсистема, яка функціонує як окремий мікросервіс. Вона реалізована на Python із використанням FastAPI, pandas, numpy та scikit-learn. На поточному етапі система формує рекомендації за такими критеріями:

* обрані користувачем категорії (вподобання);
* географічне положення (місцезнаходження або обране місто);
* відстань до локацій у заданому радіусі.

Алгоритм працює у запитному режимі: серверна частина передає параметри до Python-сервісу, який повертає відсортовану вибірку релевантних локацій. Архітектура побудована таким чином, щоб у майбутньому мати можливість легко розширити алгоритм — наприклад, за рахунок врахування історії активності, сезонності або поведінкових патернів.

Архітектурно система GoodTrip реалізована за принципами модульної мікросервісної архітектури, де кожна основна функціональність винесена в окремий сервіс:

* інтерфейс: клієнтська частина на React + TypeScript, з використанням Zustand, Axios, Tailwind CSS, shadcn/ui, MUI, i18next, Formik, Yup;
* серверна частина: побудована на ASP.NET Web API (C#), з використанням EF Core, Riok.Mapperly, MediatR і PostgreSQL;
* аутентифікація та авторизація: реалізовано через AWS Cognito з підтримкою JWT;
* файлове сховище: для зображень використовується AWS S3;
* комунікація між сервісами: здійснюється через REST API та черги повідомлень AWS SQS.

Мікросервіси серверного додатку розподілено за доменними зонами:

* Identity – управління користувачами, тегами, категоріями, а також автентифікація;
* Engagement – підписки, фоловінг
* Content – уся інша функціональність, пов’язана з контентом.

Завдяки такому підходу, GoodTrip забезпечує гнучке масштабування, розділення відповідальностей, зручну підтримку та розширення системи без необхідності зміни всієї інфраструктури.

## 1.4 Посилання

1. Посилання на репозиторій клієнту -  
   <https://github.com/GoodTripLtd/GT-Client.>
2. Посилання на репозиторій Identity сервісу -  
   <https://github.com/GoodTripLtd/GT-Identity-Microservice>.
3. Посилання на репозиторій Engagement сервісу -  
   <https://github.com/GoodTripLtd/GT-Engagement-Microservice>.
4. Посилання на репозиторій Content сервісу -  
   <https://github.com/GoodTripLtd/GT-Content-Microservice>.
5. Посилання на репозиторій реалізації алгоритму рекомендацій - <https://github.com/GoodTripLtd/GT-Recommend>

# 2 Загальний опис

## 2.1 Перспективи продукту

Програмна система GoodTrip для персоналізованих туристичних рекомендацій має значний потенціал у підвищенні ефективності планування подорожей та покращенні взаємодії користувачів. Ось деякі ключові перспективи продукту:

* централізоване зберігання та управління даними: всі дані про локації, уподобання користувачів та відгуки будуть зберігатися в єдиній системі. Це забезпечить зручний та безпечний доступ до актуальної інформації, а також полегшить пошук, перегляд та обробку даних;
* контроль відповідності політикам та стандартам: система може бути налаштована відповідно до внутрішніх регламентів та стандартів щодо контенту, модерації відгуків та взаємодії користувачів. Це дозволить уникнути порушень внутрішніх правил та забезпечує якість інформації та сервісу на стабільно високому рівні;

Програмна система GoodTrip буде актуальною для широкого кола користувачів, які прагнуть оптимізувати процес планування своїх подорожей, отримувати персоналізовані рекомендації та взаємодіяти з іншими ентузіастами подорожей. Очікується, що система стане надійним і затребуваним інструментом для автоматизації туристичного планування та покращення користувацького досвіду.

## 2.2 Функції продукту

Програмна система GoodTrip для персоналізованих туристичних рекомендацій складатиметься з веб-додатку. Функціональні можливості буде поділено відповідно до ролей користувачів системи:

Адміністратор:

* переглядати користувацькі скарги;
* видаляти відгуки та коментарі до відгуків;
* керувати категоріями: додавати, видаляти або оновлювати категорії вподобань;
* керувати локаціями: додавати, видаляти або оновлювати інформацію про туристичні об'єкти;
* керувати обліковими записами користувачів;
* блокувати або розблоковувати користувачів.

Зареєстрований користувач:

* входити до системи та авторизуватися;
* реєструвати новий обліковий запис;
* переглядати профіль;
* редагувати налаштування профілю;
* змінювати вподобання профілю для отримання персоналізованих рекомендацій;
* переглядати список обраних локацій;
* керувати власними відгуками: додавати, редагувати, видаляти;
* переглядати чужі відгуки;
* додавати реакції на відгук;
* додавати коментарі до відгуків;
* скаржитися на відгук;
* переглядати список своїх підписок і підписників;
* видаляти підписку;
* відкривати профіль іншого користувача;
* підписуватися на іншого користувача;
* здійснювати пошук користувача за прізвищем, ім’ям та іменем користувача.

## 2.3 Характеристика користувачів

Ґрунтуючись на функціональних можливостях системи, виділено основні ролі користувачів, які взаємодіятимуть із системою GoodTrip:

* адміністратор — користувач, який має можливість переглядати користувацькі скарги, видаляти відгуки, керувати категоріями (додавати, видаляти або оновлювати категорії), керувати локаціями (додавати, видаляти або оновлювати локації), керувати обліковими записами користувачів, а також блокувати або розблоковувати користувачів;
* зареєстрований користувач — користувач, який має можливість входити до системи та авторизуватися, реєструвати новий обліковий запис, переглядати та редагувати свій профіль, змінювати вподобання профілю, переглядати список обраних локацій, керувати власними відгуками (додавати, редагувати, видаляти), переглядати чужі відгуки, додавати реакції та коментарі до відгуків, скаржитися на відгук, переглядати список своїх підписок і підписників, видаляти підписку, відкривати профілі інших користувачів та підписуватися на них, а також здійснювати пошук користувача за прізвищем, ім’ям та іменем користувача.

## 2.4 Загальні обмеження

Нижче вказані всі обмеження, які будуть накладатися на технології, що будуть використовуватися при розробці продукту. Програмна система для персоналізованих туристичних рекомендацій буде складатися з мікросервісів, що охоплюють клієнтську та серверну частини, а також окремий рекомендаційний сервіс. Вибраний підхід, заснований на модульній мікросервісній архітектурі, сприятиме швидшій розробці продукту, забезпечуючи гнучкість і зручність у керуванні та масштабуванні. Для розгортання застосовується Docker із деплоєм на платформу AWS. Взаємодія всередині системи реалізована через REST API та черги повідомлень AWS SQS, що забезпечує чітке розділення відповідальностей. Як база даних використовується реляційна СУБД PostgreSQL. Для управління структурою даних застосовується Entity Framework Core, а міграції виконуються за допомогою окремого сервісу. Усі запити до бази даних обробляються через Entity Framework Core. Система оперує єдиною базою даних, яка містить усі необхідні дані для основної функціональності, тоді як рекомендаційний сервіс може використовувати власні механізми зберігання даних при необхідності. Клієнтська частина буде розроблятися за допомогою React у поєднанні з TypeScript. Для генерації стилів буде використано технологію Tailwind CSS, а для компонентів інтерфейсу — shadcn/ui та Material UI.

2.5 Припущення та залежності

Припущення:

* П-1: платформою будуть користуватися туристи, які прагнуть отримувати персоналізовані рекомендації та взаємодіяти з іншими користувачами;
* П-2: платформа буде використовуватися для пошуку локацій, формування маршрутів та обміну досвідом подорожей;
* П-3: користувачі будуть надавати свої вподобання та актуальне місцезнаходження для отримання релевантних рекомендацій.

Залежності:

* З-1: працездатність додатка буде залежати від завантаженості сервісів AWS (Cognito, S3, SQS) та від швидкості інтернет-з'єднання користувача;
* З-2: залежність коректного відображення сайту від версії браузера та пристрою, на якому відкривається сайт;
* З-3: залежність якості рекомендацій від повноти та коректності наданих користувачем вподобань та географічних даних.

# 3 Конкретні вимоги

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Під час першого доступу до вебзастосунку GoodTrip користувач автоматично перенаправляється на сторінку авторизації, де має ввести електронну пошту та пароль для входу в систему. (див. рис. 3.1).

Зображення, що містить текст, знімок екрана, Шрифт, число

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.1 – Форма логіну

Після успішної авторизації користувач, в залежності від своєї ролі переходить або до сторінки зі своїм профілем (звичайний користувач), або до сторінки зі скаргами (адміністратор).

У системі GoodTrip користувач реєструється самостійно через форму реєстрації, яка містить обов’язкові поля: ім’я, прізвище, електронна пошта, ім’я користувача (username), пароль та підтвердження паролю (див. рис. 3.2).

Зображення, що містить текст, знімок екрана, Шрифт, число

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.2 – Форма реєстрації для користувача

Кожен користувач має персоналізований функціонал на відповідних сторінках системи. Зокрема, сторінка рекомендованих місць (див. рис. 3.3) дозволяє налаштовувати фільтри пошуку і сортування за власними вподобаннями, здійснювати пошук локацій за містом або відносно поточного місцезнаходження, а також переглядати місця, додані до «Обраного», для зручного доступу.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Веб-сторінка

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.3 – Сторінка рекомендованих локацій

Для покращення взаємодії користувача з додатком, веб-інтерфейс використовує повідомлення у вигляді toast-сповіщень (див. рис. 3.4) для інформування про результати дій або необхідність підтвердження критичних операцій..

Зображення, що містить текст, Шрифт, знімок екрана, логотип

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.4 – Сповіщення для користувача

### 3.1.2 Інтерфейс користувача

Оскільки вебзастосунок GoodTrip розгортається як статичний клієнт на AWS S3 і взаємодіє з серверною частиною через хмарні сервіси, система не має виділеного фізичного обладнання чи прямих апаратних інтерфейсів. Уся обробка даних та взаємодія між компонентами відбувається у хмарному середовищі.

### 3.1.3 Програмний інтерфейс

Для доступу до системи GoodTrip користувач повинен використовувати сучасний веббраузер з підтримкою JavaScript, HTML5 та CSS3, оскільки клієнтська частина є односторінковим додатком (SPA), реалізованим з використанням сучасних вебтехнологій. Наявність підтримки зазначених стандартів є критичною для коректного функціонування інтерфейсу.

### 3.1.4 Комунікаційний протокол

Уся взаємодія між модулями серверної частини реалізована через патрени CQRS та MediatR, що дозволяє чітко розділяти запити (читання) і команди (запис), а також мінімізувати зв'язність між компонентами. Вебклієнт взаємодіє з бекендом через HTTPS-запити, які проходять через AWS API Gateway та маршрутизуються до відповідних Lambda-функцій.

Для з’єднання з базами даних, які хостяться на Amazon RDS, використовується стандартне TCP/IP-з’єднання, оптимізоване для хмарного середовища.

### 3.1.5 Обмеження пам’яті

Веб-сайт не має можливості безпосередньо контролювати або обмежувати обсяг доступної пам’яті на пристрої користувача. Обмежень щодо використання пам’яті для роботи веб-сайту не встановлено.

### 3.1.6 Операції

У веб-додатку використовується архітектура REST, яка працює на основі HTTP-протоколу і чітко дотримується принципів CRUD-операцій. Основні методи:

* GET — витягнути дані або переглянути ресурс;
* POST — створити новий об'єкт;
* PUT — оновити існуючий або створити, якщо такого ще нема;
* DELETE — знести непотрібний ресурс.

Ідентифікація й адресація ресурсів здійснюється через чітко структуровані URL-шляхи з параметрами. Усе чітко, без зайвого сміття — кожен запит робить свою справу.

### 3.1.7 Функції продукту

ГФ-1: Реєстрація та автентифікація – користувач має можливість створити обліковий запис, авторизуватись через AWS Cognito, з подальшою ідентифікацією через JWT.

ГФ-2: Керування профілем – користувач може редагувати власні дані профілю, додавати зображення, обирати вподобані теги та категорії для персоналізації системи.

ГФ-3: Система рекомендацій – на основі обраних тегів, категорій та поведінкових патернів система пропонує локації, які можуть зацікавити користувача.

ГФ-4: Соціальна взаємодія – користувач має можливість підписуватись на інших користувачів, переглядати їхній контент, створюючи власну мережу інтересів.

ГФ-5: Контент-менеджмент – користувач може створювати, редагувати та видаляти власний контент (відгуки, нотатки, фото з подорожей тощо).

ГФ-6: Система тегів та категорій – користувачі мають можливість обирати тематику контенту за допомогою тегів і категорій, що впливає на алгоритм рекомендацій.

ГФ-7: Мультимовність інтерфейсу – система підтримує кілька мов (через i18next), що дозволяє працювати з нею користувачам з різних мовних середовищ.

ГФ-8: Зберігання медіафайлів – користувачі можуть додавати зображення, які зберігаються в AWS S3.

## 3.2 Вимоги до зовнішніх інтерфейсів

Для роботи з GoodTrip необхідне стабільне інтернет-з’єднання, оскільки всі дані зберігаються на віддалених сервісах. Інтерфейс працює як статичний SPA-додаток, що деплоїться на AWS S3, а вся серверна логіка виконується через AWS Lambda з маршрутизацією запитів через API Gateway. Час відгуку системи не перевищує 60 секунд. Доступ до функціоналу залежить від ролі користувача, авторизація ідентифікує права доступу через JWT, що видається AWS Cognito.

Безпека і цілісність даних забезпечується через Amazon RDS для кожного мікросервісу, тоді як передача повідомлень і координація між мікросервісами відбувається через Amazon MQ.

GoodTrip побудований з урахуванням майбутнього масштабування, тому розширення системи не вимагає перебудови існуючої архітектури. У найближчих релізах планується реалізація таких фіч:

* інтеграція карт та побудова маршрутів прямо в інтерфейсі;
* розширені фільтри та сортування, зокрема для нетипових туристичних інтересів;
* аналітика поведінки користувача для покращення персоналізації;
* інтеграція із зовнішніми API, зокрема календарями подій і погодними сервісами;
* підтримка push-нотифікацій та реального часу через WebSocket.

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

* система GoodTrip повинна забезпечувати високу надійність, зокрема під час формування персоналізованих туристичних рекомендацій на основі вподобань користувача, історії подорожей та обраних локацій;
* у разі помилок (наприклад, проблеми з мережею, API чи обробкою даних) інтерфейс клієнтської частини повинен відображати користувачу зрозуміле повідомлення з поясненням ситуації. Технічні деталі або коди помилок мають бути приховані, а інформація — адаптована для звичайного користувача;
* комунікація між мікросервісами має бути ізольованою, щоб збої одного сервісу (наприклад, Recommendation) не впливали на роботу інших частин системи (наприклад, автентифікації або відображення збережених локацій).

3.3.2 Доступність

* веб-застосунок GoodTrip підтримує три мови інтерфейсу: англійську, українську та німецьку, що реалізовано за допомогою бібліотеки i18next. Користувач має змогу самостійно обирати мову інтерфейсу у налаштуваннях профілю;
* увесь основний функціонал системи доступний тільки для авторизованих користувачів. Гості можуть лише зареєструватися або увійти в систему;
* роль адміністратора має розширені можливості: перегляд скарг, редагування категорій, управління локаціями та блокування облікових записів;
* ролі звичайного користувача та гостя обмежені в доступі відповідно до своїх дозволів. Роутинг на клієнті реалізовано динамічно залежно від ролі користувача, що гарантує недоступність адміністративного функціоналу для неавторизованих або неуповноважених користувачів.

### 3.3.3 Безпека

* кожен обліковий запис захищений паролем, який користувач встановлює під час реєстрації та може змінити у налаштуваннях профілю. Паролі валідуються за допомогою Yup: мінімум 6 символів, принаймні одна цифра;
* авторизація реалізована через JWT та AWS Cognito, що унеможливлює доступ до даних без верифікації;
* доступ до сторінок обмежено згідно з роллю користувача (admin, user, guest), це реалізовано на клієнті через useMemo та Zustand, а також на сервері — через проміжний шар для авторизації (middleware);
* вся передача даних здійснюється за захищеним HTTPS-з’єднанням. Особисті дані та дії користувача (уподобання, маршрути, збережені локації) зберігаються у захищеній базі даних Amazon RDS із ізоляцією на рівні мікросервісів.

# 3.3.4 Супроводжуваність

* архітектура клієнтської частини побудована на основі модульної структури з ізольованими модулями (AuthenticationModule, PlacesModule, ProfileModule, AdministrationModule тощо), що дозволяє розширювати або змінювати функціональність окремих частин без ризику порушення цілісності всієї системи;
* завдяки розмежуванню обов’язків між клієнтською (React, Zustand, Tailwind CSS) та серверною частиною (ASP.NET + AWS Lambda), обслуговування системи є більш керованим і прогнозованим;
* компоненти UI реалізовані з використанням shadcn/ui та MUI, що дозволяє легко змінювати зовнішній вигляд або логіку окремих елементів інтерфейсу.

### 3.3.5 Переносимість

* клієнтська частина системи реалізована як SPA і розгортається у вигляді статичних файлів на AWS S3, що гарантує її доступність з будь-якого пристрою та регіону за наявності інтернет-з’єднання;
* інтерфейс повністю адаптивний — побудований із застосуванням Tailwind CSS і протестований у Chrome, Firefox, Safari, Edge;
* система підтримує коректне відображення на смартфонах, планшетах і десктопах завдяки респонсивному дизайну.

### 3.3.6 Продуктивність

* перехід між основними сторінками (профіль, обране, локації) не повинен перевищувати 2 секунд, що буде досягнуто завдяки використанню SPA, code splitting і lazy loading компонентів;
* після заповнення форми реєстрації та натискання кнопки "Зареєструватися", новий користувач має бути одразу перенаправлений на модальне вікно підтвердження реєстрації, а лист з кодом підтвердження має надійти на вказану електронну пошту протягом 5 хвилин;
* повідомлення (toast-нотифікації) показуються лише у відповідь на дії самого користувача, наприклад, після успішного додавання відгуку, збереження локації в обране чи зміни налаштувань профілю. Система не використовує фонові повідомлення чи технології реального часу;
* користувач не повинен мати можливість редагувати або видаляти свій відгук після того, як адміністратор його видалив або заблокував — інтерфейс динамічно приховує такі дії відповідно до статусу об'єкта.

## 3.4 Вимоги бази даних

Для зберігання даних у системі GoodTrip використовується PostgreSQL, розгорнутий через Amazon RDS окремо для кожного мікросервісу. Кожен мікросервіс (Identity, Content, Engagement тощо) має власну ізольовану базу даних, що забезпечує кращу масштабованість і безпеку даних.

Міграції бази даних реалізовані за допомогою Entity Framework Core. Для кожного мікросервісу передбачено власну бібліотеку міграцій, яка використовується:

* у WebAPI, з можливістю запуску через Swagger або автоматично під час старту;
* у консольному утилітарному застосунку, що використовується під час локальної розробки та CI/CD.

Усі доступи до бази даних здійснюються виключно через Entity Framework Core, без прямого виконання SQL-запитів або ручного мапінгу. Це забезпечує централізоване управління моделями домену та цілісність структури даних у межах кожного мікросервісу.

## 3.5 Інші вимоги

Система повинна обмежувати доступ до функціоналу лише для авторизованих користувачів. Усі основні можливості веб-застосунку (перегляд рекомендацій, керування профілем, взаємодія з локаціями) мають бути доступні лише після успішного входу в систему. Сторінка логіну повинна бути реалізована як окрема публічна частина застосунку, що не потребує автентифікації.

Інтерфейс користувача повинен бути інтуїтивно зрозумілим, адаптивним та комфортним у використанні для людей різного віку та технічного рівня. Усі елементи управління повинні бути візуально чіткими, послідовно розташованими та доступними для взаємодії з різних типів пристроїв (комп'ютер, планшет, смартфон).

# 4 Додаткові матеріали

## 4.1 Схеми баз даних

Зображення, що містить текст, схема, число, Шрифт

Вміст, створений ШІ, може бути неправильним.

Рисунок 4.1 – Схема бази даних

## 4.2 Діаграми прецедентів

Зображення, що містить текст, схема, малюнок, візерунок

Вміст, створений ШІ, може бути неправильним.

Рисунок 4.2 – Use Case діаграма для ролі Адміністратора

Зображення, що містить текст, схема, малюнок, ряд

Вміст, створений ШІ, може бути неправильним.

Рисунок 4.3 – Use Case діаграма для ролі Звичайного користувача