Львівський національний університет ім. Івана Франка
Факультет прикладної математики та інформатики
Кафедра інформаційних систем

WHY NOT EXPERIENCE

Застосунок-маркетплейс, який з'єднує
місцевих хостів, що пропонують унікальні події та активності,
з гостями, які бажають отримати цей досвід.

Автори команда WhyNot у складі:
Трескот Вадим
Петришин Софія
Євуш Софія
Роман Яремко

Львів 2025

1. Вступ

Цей документ Специфікація програмних вимог (SRS) описує повний перелік вимог до застосунку “Why Not Experience”. Документ визначає межі та функціональні можливості системи, а також нефункціональні вимоги, що дозволяють забезпечити якісну розробку, тестування та подальше використання продукту.

Мета цього документа — зафіксувати очікування замовника та користувачів, сформувати єдине бачення системи для команди розробки, а також надати основу для подальшого проєктування, реалізації та валідації застосунку.

У додатку користувачі можуть знаходити, бронювати та оплачувати різноманітні події й активності, організовані місцевими хостами: майстер-класи, екскурсії, кулінарні зустрічі тощо. Гості, у свою чергу, отримують можливість публікувати власні “досвіди”, управляти розкладом, спілкуватися з гостями та отримувати винагороду через інтегровані платіжні сервіси.

Документ також охоплює нефункціональні вимоги, зокрема системні вимоги, вимоги по безпец та, вимоги по документації. Таким чином, він забезпечує цілісне уявлення про продукт для всіх зацікавлених сторін — від замовника до розробників і тестувальників.

2. Високорівнева архітектура

2.1 Бекенд архітектура

Наша система реалізована з використанням Service-Oriented Architecture (SOA) на макрорівні та Onion Architecture на мікрорівні — всередині кожного окремого сервісу. Такий підхід забезпечує модульність, масштабованість, слабке зв’язування між компонентами та чітке розділення відповідальностей.

Проєкт складається з набору незалежних сервісів, кожен з яких відповідає за певну бізнес-функціональність. Сервіси обмінюються даними через чітко визначені API .

2.1.1 Основні принципи SOA, яких ми дотримуємось:

1. Автономність сервісів: кожен сервіс є незалежним додатком, що має власну бізнес-логіку, модель даних та інфраструктуру.

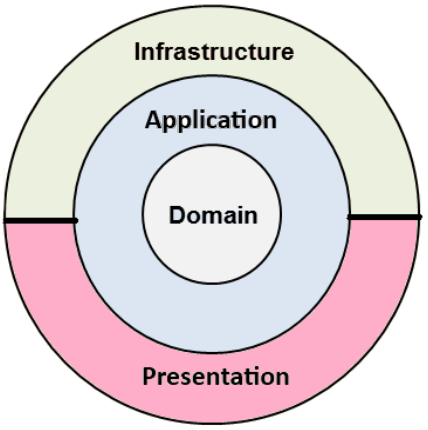
2. Слабке зв’язування: сервіси взаємодіють через стандартизовані контракти (API), без прямої залежності від реалізацій.

3. Повторне використання: сервіси спроєктовані таким чином, щоб їх можна було повторно використовувати в різних сценаріях.

4. Можливість незалежного масштабування: кожен сервіс може масштабуватись окремо в залежності від навантаження.

5. Всередині кожного сервісу ми використовуємо Onion Architecture, яка дозволяє чітко організувати код, забезпечити інверсію залежностей та тестованість бізнес-логіки.

2.1.2 Основні шари Onion Architecture:



1. Domain Layer
   1.1. Містить сутності, value objects, інтерфейси репозиторіїв та бізнес-правила.
   1.2. Абсолютно ізольований від інфраструктури.

2. Application Layer
   2.1. Визначає use cases / application services.
   2.2. Координує виконання бізнес-операцій без знання деталей реалізації.

3. Infrastructure Layer
   3.1. Реалізує інтерфейси з ядра: доступ до БД, зовнішні API, черги, файлову систему тощо.
   3.2. Залежить від внутрішніх шарів, а не навпаки.

4. Presentation Layer
   4.1. Відповідає за прийом і обробку зовнішніх запитів.
   4.2. Взаємодіє з Application Layer.

2.2 Фронтенд архітектура

Застосунок є односторінковим застосунком (Single Page Application, SPA).
Архітектура застосунку побудована за компонентним підходом.

Основні елементи:

1. App — головний компонент, що реалізує маршрутизацію.
2. Pages — логічні сторінки: головна, події, профіль користувача, сторінка хоста.
3. Components — багаторазові UI-елементи (карточки подій, кнопки, модальні вікна).
4. React Router — забезпечує навігацію без перезавантаження сторінки.
5. Context API / Redux — керування глобальним станом (користувач, кошик бронювань, теми інтерфейсу).
6. Services — робота з REST API для отримання даних про події, бронювання та користувачів.

3. Функціональні вимоги

3.1 Каталог функціональних вимог

Нижче наведено таблицю функціональних вимог до системи. Кожна вимога має унікальний ідентифікатор, назву та опис.

ID	Назва	Опис
	Пошук та навігація по аплікації	
FR01	Каталог досвідів	Перегляд публічних подій/досвідів із назвою, хостом, ціною, розкладом, місцем.
FR02	Пошук і фільтри	Пошук за ключовим словом, фільтри за датою, категорією, ціною, локацією, мовою, рейтингом.
FR03	Списки бажаного	Збереження або видалення досвідів зі списку; приватні та доступні для спільного використання списки.
	Залучення хостів та управління оголошеннями	
FR04	Реєстрація хоста	Створення профілю хоста, завантаження документів для верифікації.
FR05	Створення події	Створення події: назва, опис, категорія, програма, ціна, місткість, місце зустрічі.
FR06	Керування медіа	Завантаження та управління фото; встановлення обкладинки події.
FR07	Розклад	Встановлення доступних дат і часу, повторюваних подій.
FR08	Ціни та додаткові витрати	Встановлення базової ціни, ціни за гостя, додаткових опцій (наприклад, матеріали, транспорт).
FR09	Процес публікації	Надсилання оголошення на перевірку; адміністратор схвалює або відхиляє.
FR10	Редагування оголошень	Редагування активного оголошення; відстеження версій; зміни можуть вимагати повторного схвалення.
	Бронювання та оплата	
FR11	Бронювання	Гість може забронювати місце на подію, вказати деталі.
FR12	Управління місткістю	Автоматичне зменшення кількості місць; запобігання надмірному бронюванню; список очікування.
	Повідомлення та сповіщення	
FR13	Сповіщення	Електронні/push-повідомлення/повідомлення в додатку щодо статусу бронювання, нагадувань, змін, термінів відгуків.

FR14	Каденція нагадувань	Автоматичні нагадування перед подією (наприклад, 48 год, 3 год) із місцем зустрічі та чек-листом.
	Проведення події	
FR15	Список відвідувачів	Хост бачить список гостей, контакти, особливі потреби, може відмічати присутність.
FR16	Відмітка відвідування	Обробка запізнень; позначення відсутності (пов'язано з виплатами/поверненням коштів).
	Рейтинги, відгуки, довіра та безпека	
FR17	Оцінки та відгуки	Гості та хости залишають оцінки та відгуки один про одного, система подвійного відгуку.
	Адмін-консоль	
FR18	Модерація користувачів та контенту	Адміністратори модерації можуть переглядати, блокувати користувачів, редагувати контент.
FR19	Конфігурація та політики	Керування політиками скасування.
	Аналітика та звітність	
FR20	Панелі моніторингу	GMV (Gross Merchandise Value, загальна вартість транзакцій), бронювання, конверсії, скасування, теплові карти попиту й пропозиції.
	На рівні платформи	
FR21	Локалізація	Багатомовний контент (інтерфейс); відображення валюти з фіксацією курсу.

3.2 Актори

Актор — це об'єкт, що здійснює обмін даних із системою. Актором може бути користувач, зовнішнє обладнання або інша система. Нижче наведено перелік і ролі акторів у системі.

3.2.1 Системний адміністратор

Системний адміністратор має повний доступ до всієї аплікації та відповідає за управління основним робочим процесом на платформі. Його роль — налаштовувати систему для використання іншими групами користувачів: створення та управління обліковими записами, модерація контенту, перевірку та схвалення досвідів, моніторинг користувачів, управління політиками та забезпечення їх відповідності, а також конфігурація інтеграцій зі сторонніми сервісами.

3.2.2 Хост (Host)

Хост відповідає за створення й підтримку власних оголошень (досвідів/подій). Він може додавати та редагувати опис події, завантажувати медіа, управляти розкладом і цінами, відмічати відвідуваність.

3.2.3 Гість (Guest / User)

Гість може переглядати каталог досвідів, застосовувати пошук і фільтри, бронювати події, отримувати сповіщення, а також залишати оцінки після участі.

3.3 Випадки використання (Use Cases)

Для ключових процесів:

3.3.1 Логін / Реєстрація

Ініціатор: Будь-який користувач (гість або хост).

Опис: Актор входить у систему або створює новий обліковий запис, щоб отримати доступ до функціоналу.

Передумови:
1. Користувач має інтернет-з’єднання.
2. У разі реєстрації — підтверджує e-mail.

Основний потік:
1. Користувач вводить логін і пароль.
2. Система перевіряє правильність даних.
3. У разі успіху — надає доступ до особистого кабінету.

Альтернативні потоки:
1. A-1: Користувач реєструється вперше — створюється новий акаунт з вказаними персональними даними.

Виняткові потоки:
1. E-1: Введені неправильні дані (повторна спроба).
2. E-2: Акаунт заблокований адміністрацією.

3.3.2 Створення події (враження)

Ініціатор: Хост.

Опис: Дозволяє хосту створити новий досвід для публікації.

Передумови: Хост має підтверджений аккаунт.

Основний потік:
1. Хост натискає “Створити подію”.
2. Вводить назву, опис, категорію, місце проведення, програму, місткість і ціну.
3. Завантажує фото.
4. Додає доступні дати та час.
5. Надсилає оголошення на модерацію.

Виняткові потоки:

1. E-1: Відсутні обов’язкові поля.
2. E-2: Завантажений файл перевищує дозволений розмір.

3.3.3 Бронювання події (враження)

Ініціатор: Гість.

Опис: Дозволяє гостю забронювати участь у досвіді.

Передумови:

1. Подія активна й доступні місця.
2. Гість авторизований у системі.

Основний потік:

1. Гість обирає подію, дату та кількість місць.
2. Система показує підсумкову ціну.
3. Гість отримує підтвердження та сповіщення.

Виняткові потоки:

1. E-1: Місця закінчилися під час процесу.

3.3.4 Оцінки

Ініціатор: Гість або Хост.

Опис: Дозволяє сторонам залишати взаємні оцінки.

Передумови: Подія завершена.

Основний потік:

1. Система надсилає нагадування залишити оцінку.
2. Гість/Хост оцінює досвід (зірки).
3. Відгук стає видимим після того, як його залишили або завершився термін.

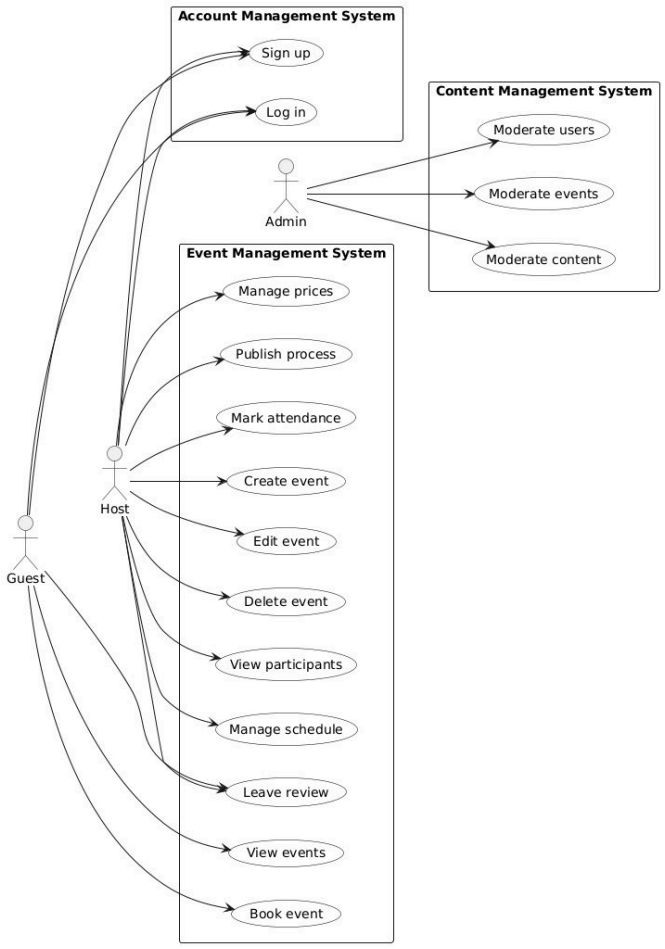
Виняткові потоки:

1. E-1: Відгук порушує правила платформи (передається на модерацію).

3.4 Use-case діаграми

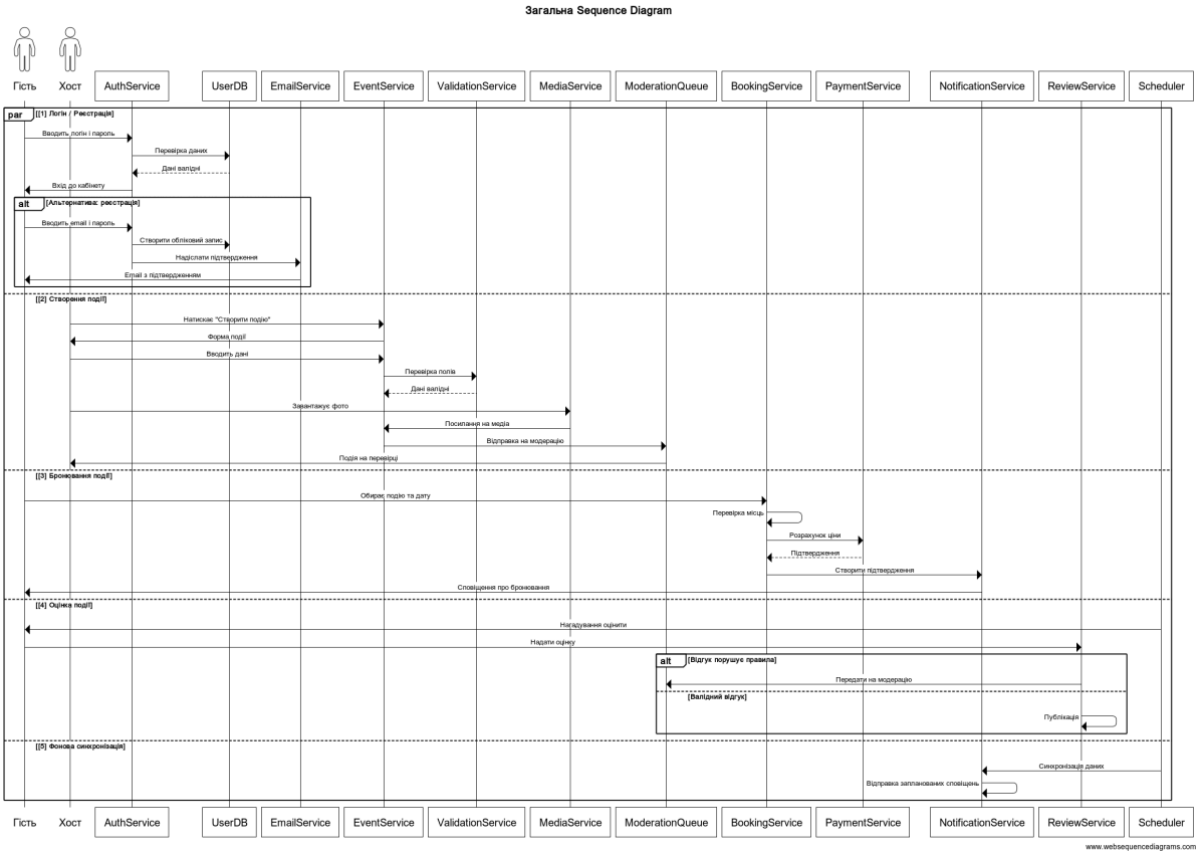
3.4.1 UML діаграма

UML діаграма створена за допомогою платформи PlantText, а за посиланням наведено вихідний код діаграми.



3.4.2 Sequence діаграма

Sequence діаграма створена за допомогою платформи Web Sequence Diagrams, а за посиланням наведено вихідний код діаграми.



4. Нефункціональні вимоги

4.1 Системні вимог

ID	Назва	Опис
SYS01	Вебзастосунок	Додаток зроблений у вигляді вебзастосунку з адаптивними стилями під різні розміри екранів.
SYS02	Сумісність	Вебзастосунок повинен бути сумісним з останніми версіями наступних браузерів: Google Chrome, Mozilla Firefox, Safari.
SYS03	Доступність функціоналу	Основні дії користувача мають виконуватися у максимум 3 кліки від головної сторінки.
SYS04	Адаптивність	Вебзастосунок має коректно відображатися на пристроях з різними розмірами екранів.
SYS05	Локалізація	Інтерфейс користувача та дані будуть українською та англійською мовою. Формати дат, чисел і часу повинні автоматично підлаштовуватися під вибрану мову.
SYS06	Архітектура	Серверна частина застосунку реалізована з використанням Service-Oriented Architecture (SOA) на макрорівні та Onion Architecture на мікрорівні.

4.2 Вимоги по безпеці

ID	Назва	Опис
SEC01	Логування	Всі ключові дії в системі повинні логуватися у центральній системі логування. Логи зберігаються протягом 30 днів або до вичерпання відведеного пам'яті.
SEC02	Повідомлення про збої	Повідомлення про збої на серверній частині вебзастосунку мають надсилатися в середовище Microsoft Teams.
SEC03	Захист з’єднання	Усі з'єднання між клієнтом і сервером повинні здійснюватися по захищеному протоколу HTTPS із використанням SSL-сертифікату (TLS 1.3).
SEC04	Захист паролів	Паролі користувачів повинні зберігатися у вигляді хешу з сіллю.
SEC05	Захист від атак	При взаємодії користувача з сервером має бути реалізований захист від наступних загроз: SQL-ін’єкції, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery).

4.3 Вимоги по документації

ID	Назва	Опис
SEC01	Специфікація проєктування програмного забезпечення	Документ, що визначає проєкт програмного забезпечення, включаючи вимоги до проєктування апаратної/програмної безпеки.