

# Актори та UML діаграми

## 1. Актори

У додатку передбачено три ролі акторів (користувачів): адміністратор, авторизований користувач і неавторизований користувач.

**Адміністратор** має повний доступ до бази даних і відповідає за її наповнення. Він може додавати нові напої та страви, забезпечуючи актуальність інформації в системі.

**Авторизований користувач** — це зареєстрований користувач, який увійшов до системи під своїм обліковим записом. Він може переглядати власний профіль, бачити свої оцінені поєднання пива і страв, а також оцінювати нові варіанти. Окрім цього, авторизований користувач має доступ до пошуку за напоєм або стравою, перегляду асортименту пива й страв, детальної інформації про напої, а також може переходити до покупки. У будь-який момент він може безпечно вийти з облікового запису.

**Неавторизований користувач** має обмежений доступ і може лише переглядати загальнодоступну інформацію. Йому доступні функції перегляду поєднань пива та страв, пошуку за напоєм або стравою, переходу до покупки, перегляду асортименту та детальної інформації про продукти. Також неавторизований користувач може увійти до системи або створити новий обліковий запис, щоб отримати розширені можливості.

## 2. Use-Case

Для ключових процесів системи SipBiteUnite виділено наступні варіанти використання, що описують основні взаємодії користувачів з системою.

### Логін / Реєстрація

**Ініціатор:** Неавторизований користувач.

**Опис:** Користувач входить у систему або створює новий обліковий запис для отримання доступу до розширеного функціоналу, зокрема можливості зберігати улюблені поєднання та оцінювати їх.

**Передумови:**

1. Користувач має доступ до інтернету.
2. У разі реєстрації користувач надає валідну електронну адресу.

## **Основний потік:**

1. Користувач переходить на сторінку входу або реєстрації.
2. Для входу користувач вводить email та пароль.
3. Система перевіряє коректність облікових даних.
4. У разі успішної перевірки система генерує JWT-токен та надає доступ до персоналізованого функціоналу.
5. Користувач перенаправляється на головну сторінку або сторінку профілю.

## **Альтернативні потоки:**

1. A-1: Користувач обирає реєстрацію нового облікового запису — вводить email, пароль, підтверджує пароль та надає базову інформацію про себе. Система створює новий обліковий запис та автоматично авторизує користувача.
2. A-2: Користувач обирає вхід через сторонні сервіси (Google, Facebook) — система використовує OAuth для автентифікації.

## **Виняткові потоки:**

1. E-1: Введено невірний email або пароль — система виводить повідомлення про помилку та пропонує повторити спробу або скористатися функцією відновлення паролю.
2. E-2: Email вже зареєстрований у системі при спробі реєстрації — система пропонує увійти або відновити пароль.
3. E-3: Пароль не відповідає вимогам безпеки — система інформує про необхідні критерії пароля.

## **Пошук та перегляд поєднань**

**Ініціатор:** Будь-який користувач (авторизований або неавторизований).

**Опис:** Користувач шукає оптимальні поєднання пива та страв за допомогою системи рекомендацій або переглядає наявні варіанти в каталозі.

## **Передумови:**

1. Користувач має доступ до інтернету.
2. База даних містить інформацію про пиво та страви.

## **Основний потік:**

1. Користувач вводить назву пива або страви у пошукове поле.
2. Система виконує пошук у базі даних та відображає відповідні результати.
3. Користувач обирає конкретне пиво або страву для перегляду детальної інформації.
4. Система відображає рекомендовані поєднання на основі характеристик обраного продукту.
5. Користувач переглядає список рекомендацій з описом та оцінками.

#### **Альтернативні потоки:**

1. А-1: Користувач використовує фільтри для пошуку (стиль пива, тип страви, рівень гіркоти) — система відображає відфільтровані результати.
2. А-2: Користувач переглядає каталог без конкретного пошуку — система відображає популярні або рекомендовані поєднання.

#### **Виняткові потоки:**

1. Е-1: За запитом не знайдено результатів — система інформує користувача та пропонує змінити критерії пошуку.
2. Е-2: Помилка з'єднання з базою даних — система виводить повідомлення про тимчасову недоступність сервісу.

### **Оцінювання поєднань**

**Ініціатор:** Авторизований користувач.

**Опис:** Користувач залишає оцінку для поєднання пива та страви, що він спробував, допомагаючи іншим користувачам та покращуючи систему рекомендацій.

#### **Передумови:**

1. Користувач авторизований у системі.
2. Поєднання існує в базі даних.

#### **Основний потік:**

1. Користувач переглядає детальну інформацію про конкретне поєднання.
2. Користувач натискає кнопку "Оцінити поєднання".
3. Система відображає форму оцінювання з можливістю поставити оцінку від 1 до 5 зірок.

4. Користувач обирає оцінку та за бажанням залишає текстовий коментар.
5. Система зберігає оцінку в базі даних та оновлює середню оцінку поєднання.
6. Оцінка відображається в профілі користувача у розділі "Оцінені поєднання".

#### **Альтернативні потоки:**

1. A-1: Користувач змінює раніше залишену оцінку — система оновлює існуючу оцінку замість створення нової.

#### **Виняткові потоки:**

1. E-1: Користувач намагається оцінити поєднання без авторизації — система перенаправляє на сторінку входу.
2. E-2: Помилка збереження оцінки — система виводить повідомлення про помилку та пропонує повторити спробу.

## **Додавання нових продуктів (адміністратор)**

**Ініціатор:** Адміністратор.

**Опис:** Адміністратор додає нове пиво або страву до бази даних системи для розширення каталогу.

#### **Передумови:**

1. Користувач має права адміністратора.
2. Адміністратор авторизований у системі.

#### **Основний потік:**

1. Адміністратор переходить до адміністративної панелі.
2. Адміністратор обирає опцію "Додати напій" або "Додати страву".
3. Система відображає форму з полями для введення інформації.
4. Адміністратор заповнює обов'язкові поля: назву, опис, характеристики (IBU, ABV для пива або тип страви, інгредієнти для страв).
5. Адміністратор завантажує зображення продукту.
6. Система валідує введені дані.
7. У разі успішної валідації система зберігає новий продукт у базі даних.
8. Новий продукт стає доступним для всіх користувачів у каталозі.

### **Альтернативні потоки:**

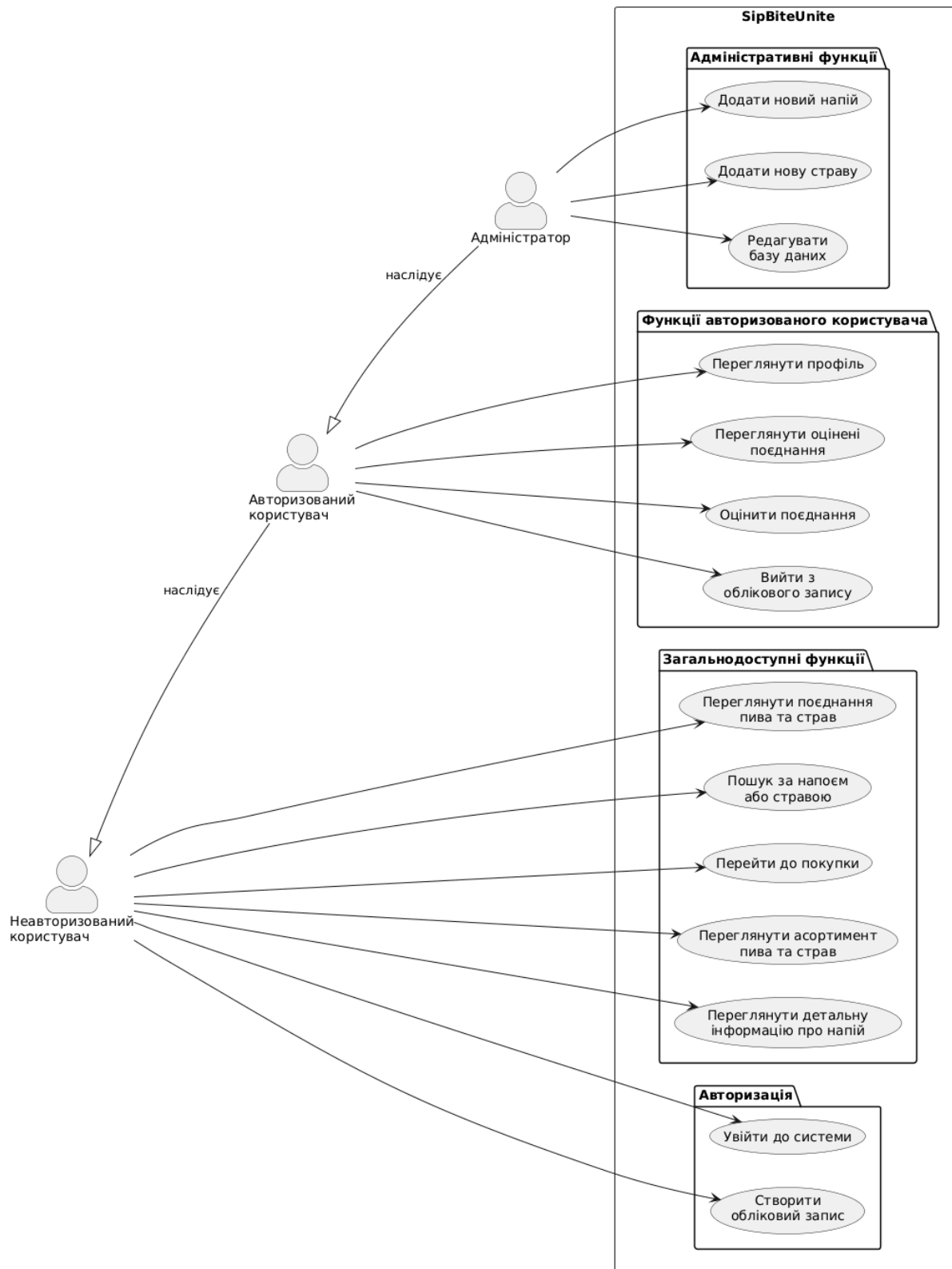
1. A-1: Адміністратор редагує існуючий продукт — система завантажує поточні дані для редагування та оновлює запис після збереження.
2. A-2: Адміністратор видаляє продукт — система помічає запис як неактивний без фізичного видалення з бази даних.

### **Виняткові потоки:**

1. E-1: Не заповнені обов'язкові поля — система виводить повідомлення про необхідність заповнення всіх обов'язкових полів.
2. E-2: Завантажене зображення перевищує дозволений розмір або має невідтримуваний формат — система інформує про помилку та вимоги до файлу.
3. E-3: Продукт з такою назвою вже існує в базі даних — система попереджає про можливий дублікат та пропонує перевірити існуючі записи.

## **3. UML діаграма**

UML діаграма варіантів використання (Use Case diagram) візуалізує функціональні можливості системи SipBiteUnite та взаємодію різних типів користувачів з цими функціями, демонструючи, які дії може виконувати кожна роль в системі.



Код:

@startuml

skinparam actorStyle awesome

left to right direction

actor "Неавторизований\нкористувач" as Guest

actor "Авторизований\нкористувач" as User

actor "Адміністратор" as Admin

rectangle SipBiteUnite {

```
package "Загальнодоступні функції" {  
    usecase UC1 as "Переглянути поєднання  
    пива та страв"  
    usecase UC2 as "Пошук за напоєм  
    або стравою"  
    usecase UC3 as "Перейти до покупки"  
    usecase UC4 as "Переглянути асортимент  
    пива та страв"  
    usecase UC5 as "Переглянути детальну  
    інформацію про напій"  
}
```

```
package "Авторизація" {  
    usecase UC6 as "Увійти до системи"  
    usecase UC7 as "Створити  
    обліковий запис"  
}
```

```
package "Функції авторизованого користувача" {  
    usecase UC8 as "Переглянути профіль"  
    usecase UC9 as "Переглянути оцінені  
    поєднання"  
    usecase UC10 as "Оцінити поєднання"  
    usecase UC11 as "Вийти з  
    облікового запису"  
}
```

```
package "Адміністративні функції" {  
    usecase UC12 as "Додати новий напій"  
    usecase UC13 as "Додати нову страву"  
    usecase UC14 as "Редагувати  
    базу даних"  
}  
}
```

Guest --> UC1

Guest --> UC2

Guest --> UC3

Guest --> UC4

Guest --> UC5

Guest --> UC6

Guest --> UC7

User --> UC8

User --> UC9

User --> UC10

User --> UC11

Admin --> UC12

Admin --> UC13

Admin --> UC14

User -up-|> Guest : наслідує

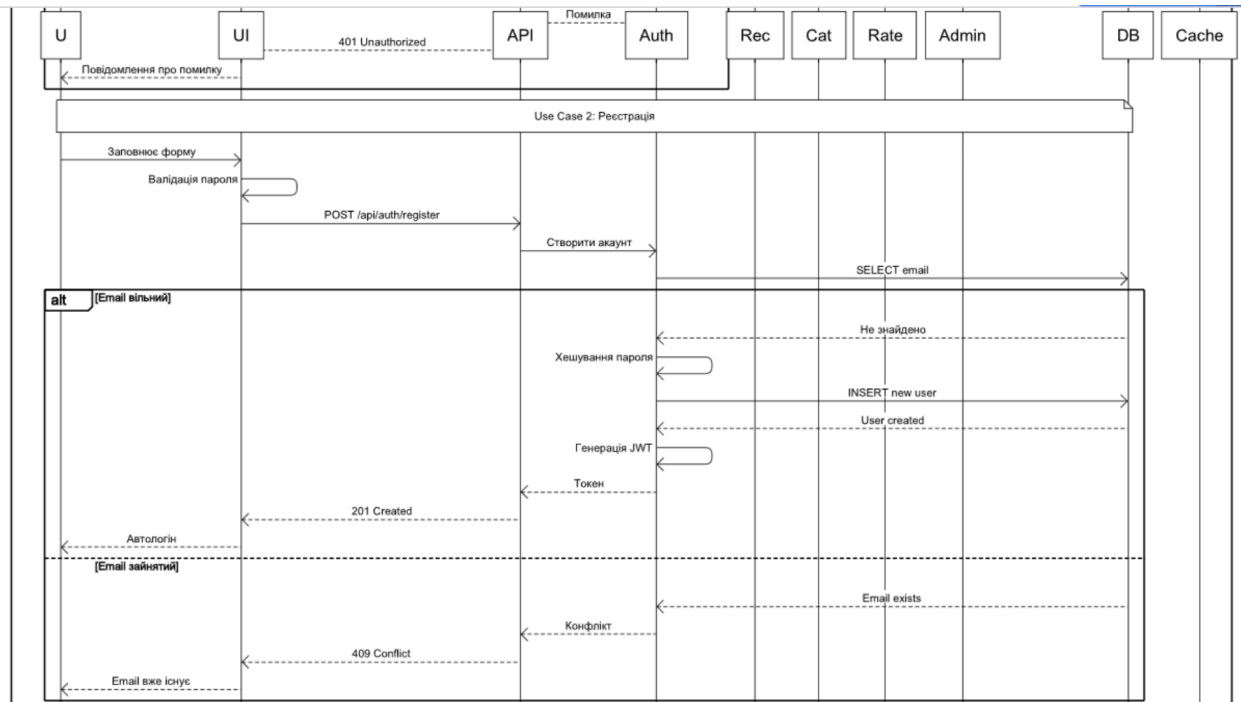
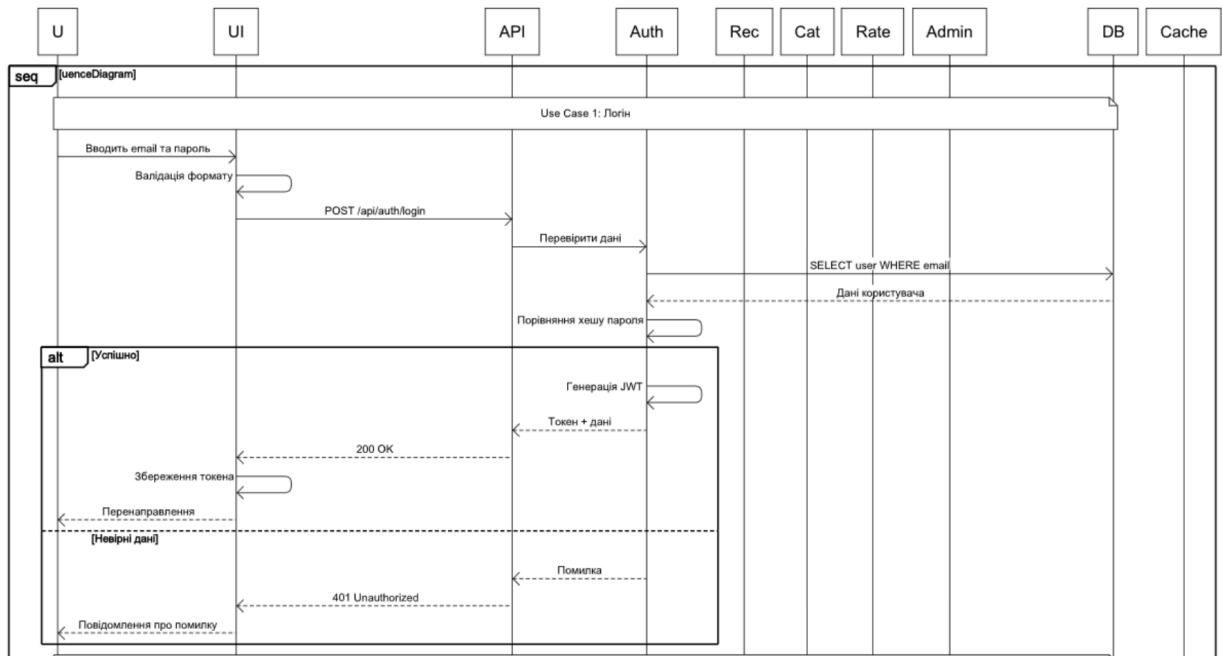
Admin -up-|> User : наслідує

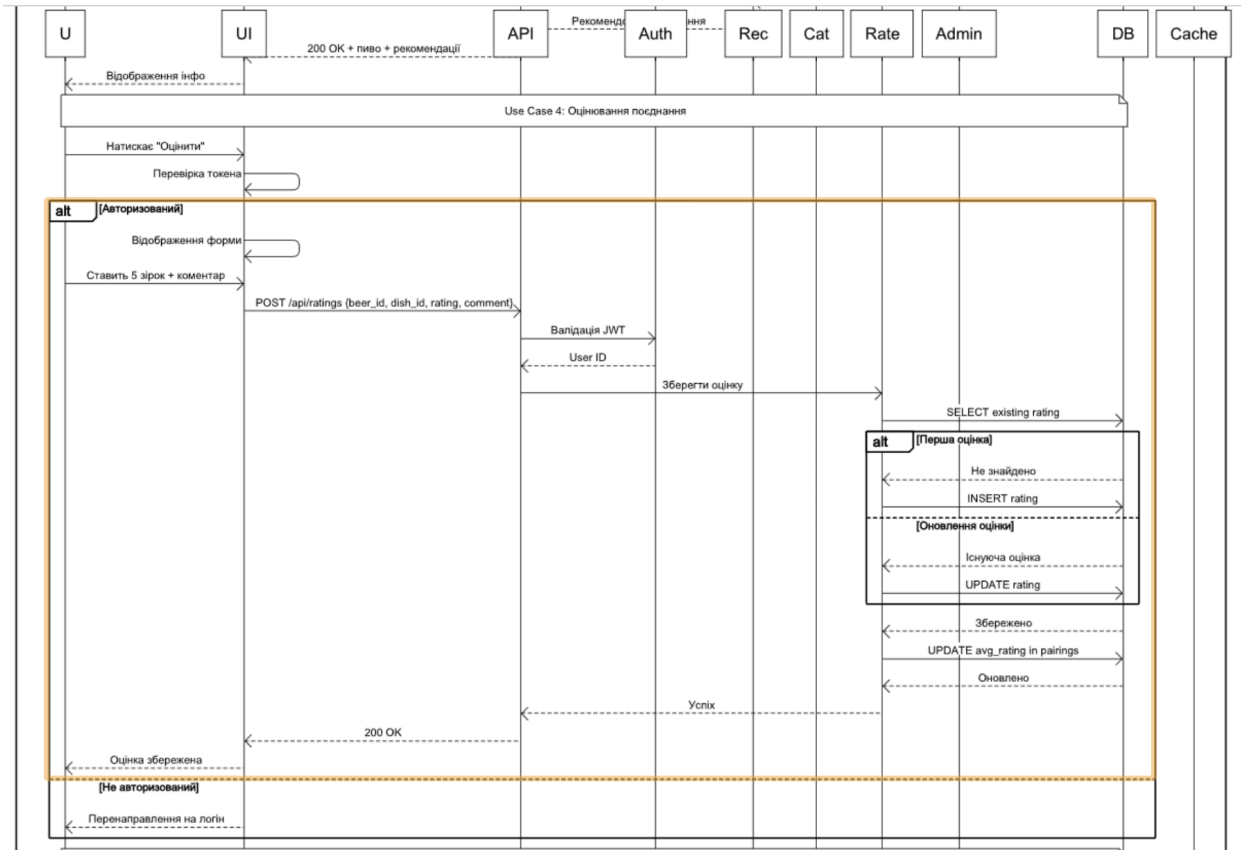
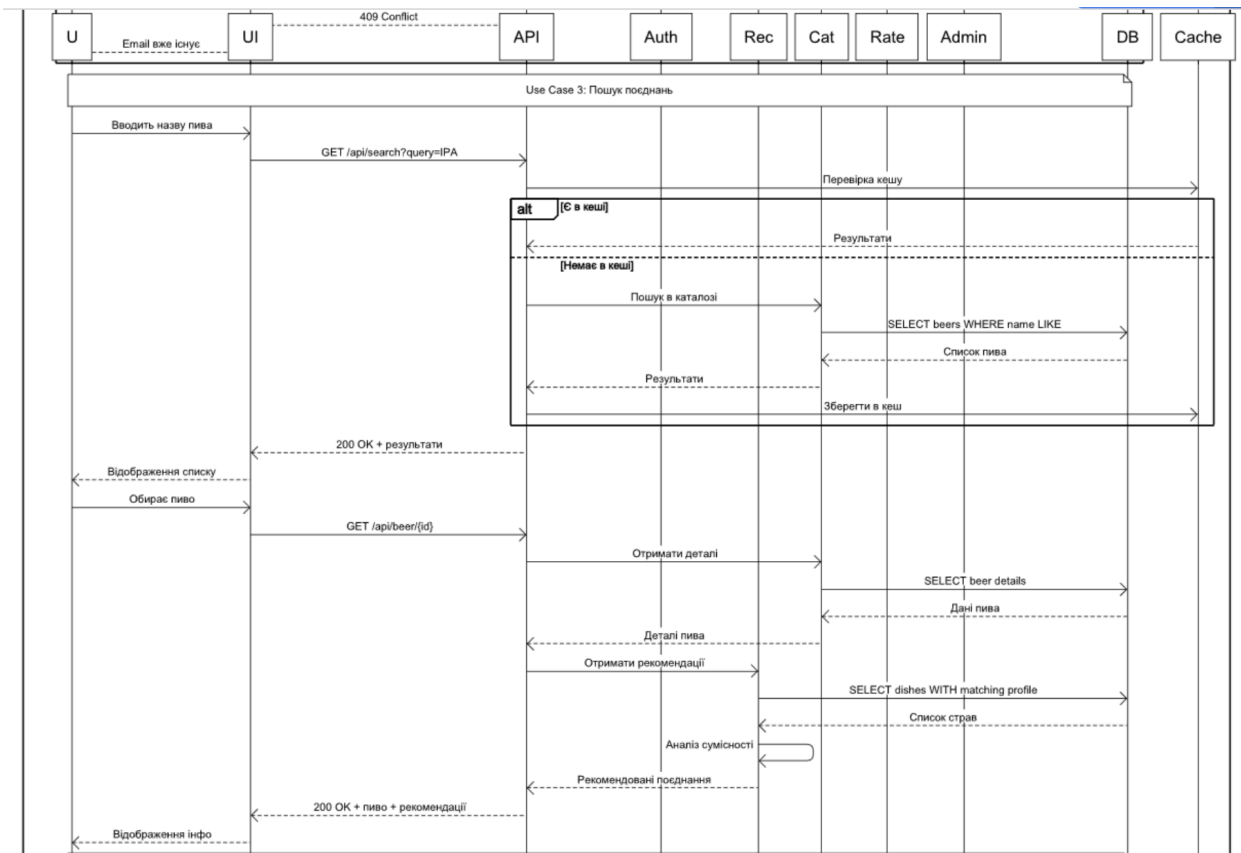
@enduml

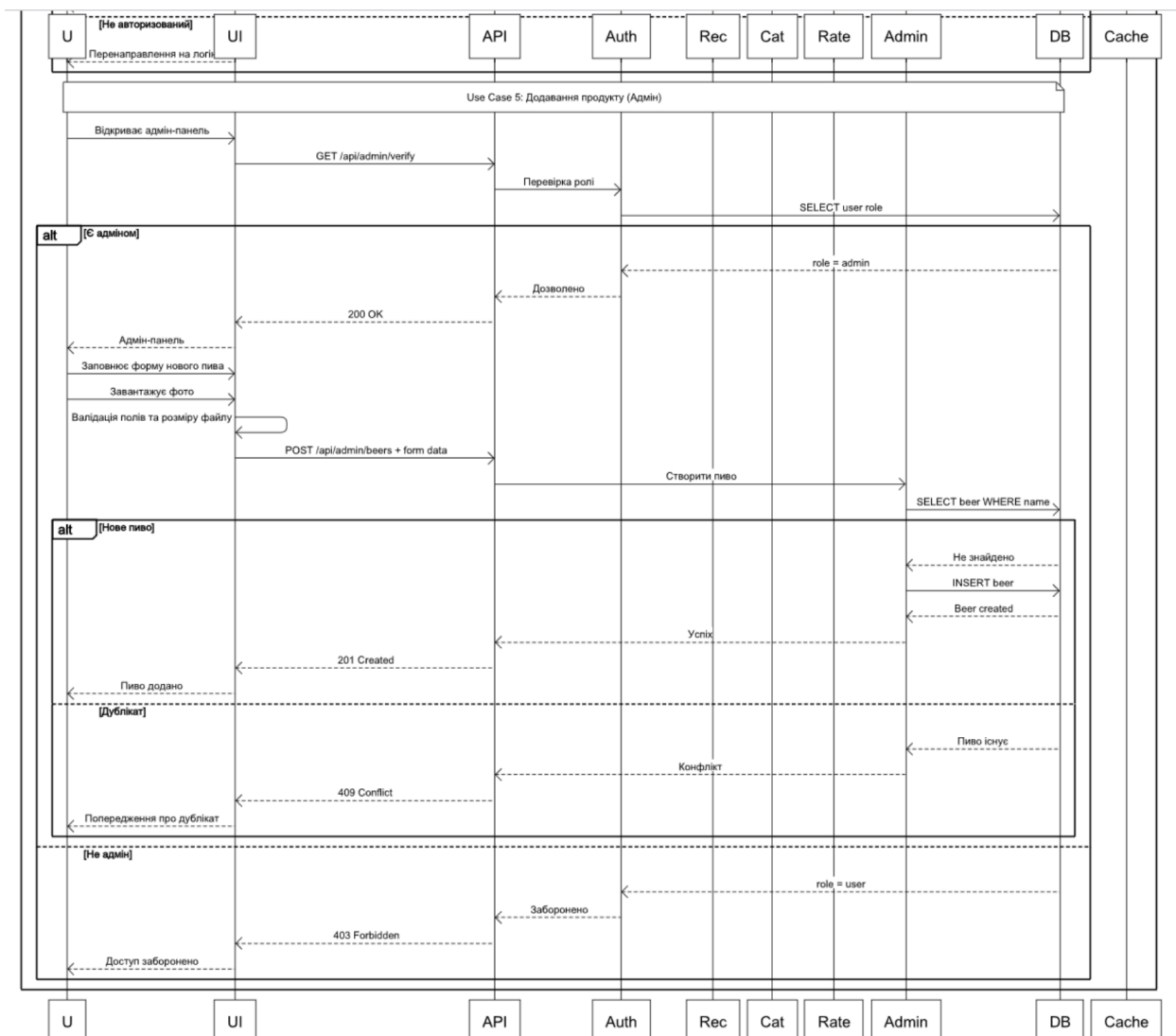
#### **4. Загальна sequence діаграма**

Sequence діаграми (діаграми послідовності) відображають взаємодію між різними компонентами системи в часі, показуючи порядок обміну повідомленнями між об'єктами. Ми створили ці діаграми для візуалізації послідовності дій та комунікації між клієнтом, сервером, базою даних та іншими компонентами SipBiteUnite під час виконання ключових сценаріїв використання. Це допомагає зрозуміти логіку роботи системи, виявити можливі проблеми в архітектурі та забезпечує документацію для розробників, які працюватимуть над реалізацією функціоналу.









Код:

sequenceDiagram

participant U as Користувач

participant UI as Інтерфейс

participant API as Backend API

participant Auth as Сервіс автентифікації

participant Rec as Рекомендаційний сервіс

participant Cat as Сервіс каталогу

participant Rate as Сервіс оцінок

participant Admin as Адмін сервіс

participant DB as База даних

participant Cache as Redis Cache

Note over U,DB: Use Case 1: Логін

U->>UI: Вводить email та пароль

UI->>UI: Валідація формату

UI->>API: POST /api/auth/login  
API->>Auth: Перевірити дані  
Auth->>DB: SELECT user WHERE email  
DB-->>Auth: Дані користувача  
Auth->>Auth: Порівняння хешу пароля

alt Успішно

Auth->>Auth: Генерація JWT

Auth-->>API: Токен + дані

API-->>UI: 200 OK

UI->>UI: Збереження токена

UI-->>U: Перенаправлення

else Невірні дані

Auth-->>API: Помилка

API-->>UI: 401 Unauthorized

UI-->>U: Повідомлення про помилку

end

Note over U,DB: Use Case 2: Реєстрація

U->>UI: Заповнює форму  
UI->>UI: Валідація пароля  
UI->>API: POST /api/auth/register  
API->>Auth: Створити акаунт  
Auth->>DB: SELECT email

alt Email вільний

DB-->>Auth: Не знайдено

Auth->>Auth: Хешування пароля

Auth->>DB: INSERT new user

DB-->>Auth: User created

Auth->>Auth: Генерація JWT

Auth-->>API: Токен

API-->>UI: 201 Created

UI-->>U: Автологін

else Email зайнятий

DB-->>Auth: Email exists

Auth-->>API: Конфлікт

API-->>UI: 409 Conflict

UI-->>U: Email вже існує

end

Note over U,DB: Use Case 3: Пошук поєднань

U->>UI: Вводить назву пива  
UI->>API: GET /api/search?query=IPA  
API->>Cache: Перевірка кешу

```
alt Є в кеші
    Cache-->>API: Результати
else Немає в кеші
    API->>Cat: Пошук в каталозі
    Cat->>DB: SELECT beers WHERE name LIKE
    DB-->>Cat: Список пива
    Cat-->>API: Результати
    API->>Cache: Зберегти в кеш
end
```

```
API-->>UI: 200 OK + результати
UI-->>U: Відображення списку
```

```
U->>UI: Обирає пиво
UI->>API: GET /api/beer/{id}
API->>Cat: Отримати деталі
Cat->>DB: SELECT beer details
DB-->>Cat: Дані пива
Cat-->>API: Деталі пива
API->>Rec: Отримати рекомендації
Rec->>DB: SELECT dishes WITH matching profile
DB-->>Rec: Список страв
Rec->>Rec: Аналіз сумісності
Rec-->>API: Рекомендовані поєднання
API-->>UI: 200 OK + пиво + рекомендації
UI-->>U: Відображення інфо
```

Note over U,DB: Use Case 4: Оцінювання поєднання

```
U->>UI: Натискає "Оцінити"
UI->>UI: Перевірка токена
```

```
alt Авторизований
    UI->>UI: Відображення форми
    U->>UI: Ставить 5 зірок + коментар
    UI->>API: POST /api/ratings {beer_id, dish_id, rating, comment}
    API->>Auth: Валідація JWT
    Auth-->>API: User ID
    API->>Rate: Зберегти оцінку
    Rate->>DB: SELECT existing rating
```

```
alt Перша оцінка
    DB-->>Rate: Не знайдено
    Rate->>DB: INSERT rating
else Оновлення оцінки
    DB-->>Rate: Існуюча оцінка
    Rate->>DB: UPDATE rating
end
```

DB-->>Rate: Збережено  
Rate-->>DB: UPDATE avg\_rating in pairings  
DB-->>Rate: Оновлено  
Rate-->>API: Успіх  
API-->>UI: 200 OK  
UI-->>U: Оцінка збережена

else Не авторизований  
UI-->>U: Перенаправлення на логін  
end

Note over U,DB: Use Case 5: Додавання продукту (Адмін)

U->>UI: Відкриває адмін-панель  
UI->>API: GET /api/admin/verify  
API->>Auth: Перевірка ролі  
Auth->>DB: SELECT user role

alt Є адміном  
DB-->>Auth: role = admin  
Auth-->>API: Дозволено  
API-->>UI: 200 OK  
UI-->>U: Адмін-панель

U->>UI: Заповнює форму нового пива  
U->>UI: Завантажує фото  
UI->>UI: Валідація полів та розміру файлу  
UI->>API: POST /api/admin/beers + form data  
API->>Admin: Створити пиво  
Admin->>DB: SELECT beer WHERE name

alt Нове пиво  
DB-->>Admin: Не знайдено  
Admin->>DB: INSERT beer  
DB-->>Admin: Beer created  
Admin-->>API: Успіх  
API-->>UI: 201 Created  
UI-->>U: Пиво додано  
else Дублікат  
DB-->>Admin: Пиво існує  
Admin-->>API: Конфлікт  
API-->>UI: 409 Conflict  
UI-->>U: Попередження про дублікат  
end

else Не адмін  
DB-->>Auth: role = user

Auth-->>API: Заборонено

API-->>UI: 403 Forbidden

UI-->>U: Доступ заборонено

end