Вы — бизнес-аналитик в e-commerce компании, которая развивает крупный маркетплейс. Для повышения вовлеченности пользователей и предоставления им более полной информации о товарах, было решено внедрить новый функционал: **"Отзывы на товары"**.

Вам поручено спроектировать REST API для этого функционала, чтобы команда фронтенда (веб-сайт) и команда мобильной разработки могли интегрировать его в свои приложения.

### Задачи:

**1. Анализ требований и декомпозиция (Аналог Use Case)**

* Проанализируйте бизнес-сценарий (ниже).
* Определите ключевые **ресурсы** (сущности), которыми будет оперировать API (не менее 2-х).
* Опишите основные **действия** (сценарии), которые можно выполнять с этими ресурсами (например, "Создать отзыв", "Просмотреть все отзывы к товару" и т.д., не менее 5 действий).

**2. Проектирование эндпоинтов (Аналог Activity Diagram)**

* Для каждого действия из п.1 спроектируйте соответствующий **эндпоинт**: HTTP-метод + URI.
* Представьте результат в виде таблицы.
* Укажите, какие эндпоинты требуют авторизации.

**3. Моделирование данных (Аналог Class Diagram)**

* Опишите структуру основного объекта **"Отзыв" (Review)** в формате JSON.
* Укажите все необходимые поля (не менее 5), их типы данных (string, integer, boolean и т.д.) и отметьте, какие из них являются обязательными при создании.

**4. Создание спецификации (Основной артефакт)**

* Напишите фрагмент спецификации **OpenAPI 3.0** в формате **YAML** для одного, самого важного, на ваш взгляд, эндпоинта.
* В спецификации должны быть описаны: summary, tags, параметры (если есть), requestBody (если нужно) и как минимум два варианта responses (успешный и ошибка).

### Бизнес-сценарий:

Любой пользователь (даже гость) может просматривать список отзывов к конкретному товару. Авторизованный покупатель может оставить только один отзыв на товар, который он ранее купил. Отзыв включает в себя оценку (от 1 до 5) и текстовый комментарий. Автор отзыва может его отредактировать или удалить. Модератор может скрыть (но не удалить) любой отзыв, если он нарушает правила площадки.

### Система оценивания:

|  |  |  |
| --- | --- | --- |
| **Критерий** | **Баллы** | **Комментарии** |
| **1. Анализ и декомпозиция** | **3** | Полнота и корректность выделенных ресурсов и действий. Соответствие бизнес-сценарию. |
| **2. Проектирование эндпоинтов** | **3** | Логичность URI, правильный выбор HTTP-методов (соблюдение принципов REST), указание на необходимость авторизации. |
| **3. Моделирование данных (JSON)** | **3** | Корректная и полная структура JSON-объекта, правильные типы данных, выделены обязательные поля. |
| **4. Спецификация OpenAPI (YAML)** | **2** | Синтаксическая корректность YAML, полнота описания эндпоинта (запрос, успешный ответ, ответ с ошибкой). |
| **5. Оформление и читаемость** | **1** | Чёткость, структурированность ответа, понятные названия. |

### Шкала:

* **12–10 баллов** — **отлично** (все задачи выполнены без ошибок, спецификация детальна и логична).
* **9–7 баллов** — **хорошо** (есть незначительные недочёты, например, неоптимальный выбор HTTP-метода или пропущено необязательное поле в модели).
* **6–4 балла** — **удовлетворительно** (грубые ошибки в REST-подходе или спецификации, неполный анализ требований).
* **3–0 баллов** — **неудовлетворительно** (игнорирование ключевых требований, работа не выполнена).