**Task 3: Test Case Design**

**API Test Cases**

**1. Users Endpoint**

| **Test Case ID** | **Description** | **Method** | **Endpoint** | **Expected Result** |
| --- | --- | --- | --- | --- |
| API-USERS-01 | Retrieve all users | GET | /users | Response Code: 200, Content-Type: JSON, 10 users returned |
| API-USERS-02 | Retrieve user by ID | GET | /users/{id} | Response Code: 200, Content-Type: JSON, User data for ID |
| API-USERS-03 | Retrieve user by invalid ID | GET | /users/9999 | Response Code: 404, Error Message: "Not Found" |

**2. Albums Endpoint**

| **Test Case ID** | **Description** | **Method** | **Endpoint** | **Expected Result** |
| --- | --- | --- | --- | --- |
| API-ALBUMS-01 | Retrieve all albums | GET | /albums | Response Code: 200, Content-Type: JSON, Multiple albums returned |
| API-ALBUMS-02 | Retrieve album by ID | GET | /albums/{id} | Response Code: 200, Content-Type: JSON, Album data for ID |
| API-ALBUMS-03 | Retrieve album by invalid ID | GET | /albums/9999 | Response Code: 404, Error Message: "Not Found" |
| API-ALBUMS-04 | Retrieve all albums for a user ID | GET | /users/{id}/albums | Response Code: 200, Content-Type: JSON, Albums linked to user ID |

**3. Photos Endpoint**

| **Test Case ID** | **Description** | **Method** | **Endpoint** | **Expected Result** |
| --- | --- | --- | --- | --- |
| API-PHOTOS-01 | Retrieve all photos | GET | /photos | Response Code: 200, Content-Type: JSON, Multiple photos returned |
| API-PHOTOS-02 | Retrieve photo by ID | GET | /photos/{id} | Response Code: 200, Content-Type: JSON, Photo data for ID |
| API-PHOTOS-03 | Retrieve photo by invalid ID | GET | /photos/9999 | Response Code: 404, Error Message: "Not Found" |
| API-PHOTOS-04 | Retrieve all photos in an album | GET | /albums/{id}/photos | Response Code: 200, Content-Type: JSON, Photos linked to album ID |

I would break it down as follows:

**Detailed Test Case Descriptions**

**1. Retrieve All Users (API-USERS-01)**

* **Description**: Verify the API returns a list of all users.
* **Method**: GET
* **Endpoint**: /users
* **Expected Result**:
  + **Status Code**: 200
  + **Content-Type**: application/json
  + Response contains 10 user objects.

**2. Retrieve User by ID (API-USERS-02)**

* **Description**: Verify the API returns user details for a given valid user ID.
* **Method**: GET
* **Endpoint**: /users/{id}
* **Expected Result**:
  + **Status Code**: 200
  + **Content-Type**: application/json
  + Response contains details of the specified user.

**3. Retrieve User by Invalid ID (API-USERS-03)**

* **Description**: Verify the API returns a 404 error for an invalid user ID.
* **Method**: GET
* **Endpoint**: /users/9999
* **Expected Result**:
  + **Status Code**: 404
  + Error message indicating "Not Found".