**Project Overview: API Testing for User Management System**

**Objective:**

The goal of this project is to test the basic CRUD operations (Create, Read, Update, Delete) of an API for a user management system. The tests will validate the performance, correctness, and response time of each API endpoint.

**API Endpoints Tested:**

**GET /api/users?page=1**

**Description:** Retrieves a list of users with pagination support.

**Method:** GET

**Assertions:**

Status code equals 200 (OK).

Response time is within an acceptable range (e.g., under 2 seconds).

Valid JSON response structure.

**POST /api/users**

**Description:** Creates a new user with the provided name and job parameters.

**Request Body:**

{

"name": "John Doe",

"job": "Developer"}

**Method: POST**

**Assertions:**

Status code equals 201 (Created).

Response time is under the specified limit.

Response body contains the correct name and job attributes.

**GET /api/users/1**

**Description:** Retrieves the details of the user with ID 1.

**Method**: GET

**Assertions:**

Status code equals 200 (OK).

The response contains the expected user details (i.e., user ID 1).

Response time is within acceptable limits.

**PUT /api/users/1**

**Description:** Updates the first name of user with ID 1.

**Request Body:**

{

"first\_name": "Paul"

}

**Method: PUT**

**Assertions:**

Status code equals 200 (OK).

The first\_name field in the response is updated to "Paul".

Response time is within the defined limit.

**DELETE /api/users/1**

**Description:** Deletes the user with ID 1.

**Method:** DELETE

**Assertions:**

Status code equals 204 (No Content), indicating successful deletion.

The user should no longer exist when queried via GET.

Response time remains within acceptable limits.

**Test Plan Configuration:**

*Number of Threads (Users):* 100

This simulates 100 concurrent users performing the test operations on the API.

*Ramp-Up Period:* 10 seconds

The ramp-up period ensures that the 100 users will start gradually, with a 10-second interval, to simulate more realistic load on the API.

*Loop Count: 1*

Each request will be executed once per thread during the test. This configuration allows for a single iteration of testing per user.

**Timers:**

To simulate real-world user delays, a random timer between 300 to 700 milliseconds is applied to each request:

*Random Timer (300-700 ms):* Introduces a random delay between requests, simulating more realistic user behavior by varying the wait time between 300 and 700 milliseconds for each thread.

**Assertions:**

Response Assertions: Ensures that the response status code and body are correct for each endpoint.

For GET requests, ensures correct status code and valid response body.

For POST and PUT requests, checks for correct status codes and ensures that the created or updated data matches the request.

For DELETE requests, confirms the user was deleted successfully with a 204 status code.

**Result Monitoring:**

*Response Times:* Track the response times of all API calls and ensure that they remain within the desired limits for optimal performance.

*Error Rate:* Monitor any failed requests and analyze the root causes for potential improvements in API design or load handling.