Testing Document

1. Introduction

The purpose of this testing document is to outline the testing approach, scenarios, and results for the aggregation server and its interaction with the GET and Content servers. This document covers tests for functional correctness, concurrent access, data expiry, and Lamport clock synchronization.

2. Objectives

To verify that the aggregation server correctly handles PUT and GET requests.

To ensure concurrent requests are managed appropriately and that there are no race conditions.

To validate that expired data is removed as expected after the given interval.

To confirm that Lamport clock synchronization is functioning as intended across all server interactions.

3. Test Strategy

The test strategy comprises of the following components:

- Unit Tests: Testing individual methods and functionalities in isolation.
- Integration Tests: Validating that different components (Aggregation Server, GET Client, Content Server) work correctly together.
- Concurrent Testing: Simulating multiple concurrent PUT and GET requests to verify thread safety and proper synchronization.
- Edge Case Testing: Testing with invalid data formats, concurrent access, and data expiry to ensure proper handling of edge cases.

The tests are written using JUnit 5 and executed using the ExecutorService to handle multi-threading for concurrent tests.

4. Test Environment

- Java Version: Java 11
- Libraries: JUnit 5
- Frameworks: Core Java
- Tools: Git for version control, IDE IntelliJ IDEA

5. Test Cases

Below are the test cases along with their description, expected results, and their outcomes:

Test Case ID	Test Description	Expected Result	Actual Result	Pass/Fail
TC1	**PUT Request with Valid Data**	Status `201 Created` on the first PUT request	Status `201 Created`	**Pass**
TC2	**GET Request for Weather Data**	Status `200 OK` with expected JSON data	Status `200 OK`, JSON data received	**Pass**
TC3	**PUT Request with Invalid Data**	Status `500 Internal Server Error` for malformed JSON	Status `500 Internal Server Error`	**Pass**
TC4	**Concurrent GET Requests**	All concurrent GETs receive status `200 OK`	All `200 OK` responses received	**Pass**
TC5	**Concurrent PUT Requests**	All PUT requests are serialized, status `201 Created`	All `201 Created` responses received	**Pass**
TC6	**Data Expiry Check**	Data is removed after 30 seconds of inactivity	Data expired as expected	**Pass**
TC7	**Selective Data Expiry Check**	Specific entry is removed after the expiry interval	Specific entry expired correctly	**Pass**
TC8	**Lamport Clock Synchronization**	Correctly incremented Lamport clock on each PUT/GET request	Lamport clock incremented as expected	**Pass**
TC9	**Multiple PUT Requests with Lamport Clock**	Lamport clock value increments correctly across multiple PUTs	Lamport clock incremented as expected	**Pass**
TC10	**Concurrent PUT and GET Requests**	Proper serialization of PUT and GET requests, data consistency maintained	Data consistency maintained, `200 OK` and `201 Created` responses received	**Pass**
TC11	**Content Server Retry Mechanism**	Content server retries on unavailable aggregation server	Retries successfully on server unavailability	**Pass**
TC12	**Multiple Content Servers PUT Requests**	Concurrent Content servers successfully PUT data	Data received correctly by the aggregation server	**Pass**

TC13	**GET Client Concurrent Requests**	Concurrent GET Clients receive consistent JSON responses	Consistent data responses across all clients	**Pass**