Modifications Made to the Project Submission

1. Additional Detailed Execution:

Provided comprehensive details regarding the operation of the ContentServer, AggregationServer, and GETClient components.
outlined each component's roles, how they work together, and how data is exchanged between them.
Updated the design drawing to incorporate important architectural features including data expiration procedures and multi-threading.

2. Update for the ReadMe File:

Added instructions on how to execute, compile, and test the code to the README file.
mentioned particular commands to execute the components, compile the code, and run the unit tests.
Described the overall project procedure, the libraries (Hamcrest and JUnit) that were utilised, and the project's structure

3. Testing:

Implemented thorough integration, edge case, and unit tests for the project.
AggregationServerTest, ContentServerTest, and GETClientTest tests have all been added, and they cover a variety of cases including big data files, missing fields, and empty files.
confirmed that every test runs well, demonstrating the durability and dependability of the code.

4. Comments on the Code and Quality Enhancements:

The goal and operation of each method and code block are explained in comments included throughout the code.
made sure the code complies with the finest standards for maintainability and readability.
enhanced error management in crucial areas of the system, like the ContentServer's handling of missing fields or incorrect input.

5. Design Sketch:

Supplied a design schematic that illustrates the data flow, system architecture, and component interactions graphically.
shown how the AggregationServer uses multi-threading to manage several client queries at once.
contained an explanation of how to use Lamport clocks to keep distributed components consistent.