

SP-114 Data Lake Implementation

Software Test Plan

CS 4850, Section 02, Spring 2026

Professor Perry, 3/8/2026

	Name	Role
Team Leader	Caleb Cox	Document project progress; Coordinate communications and development between team members
Team Member	Bryce Wishart	Documentation, development, programming, and testing.
Advisor/Instructor	Sharon Perry	Facilitate project progress; advise on project planning and management.



Caleb Cox
Team Leader



Bryce Wishart
Group Member

Table of Contents

1. Test Objectives	3
2. Testing Scope	3
3. Test Cases	3
3.1 Object Store Credentials	3
3.2 Network Access	4
3.3 Data Transfer	4
3.4 Data Transformation	4
3.5 Data Querying	4
4. Procedures	4
4.1 Sign In	4
4.2 Network Access	5
4.3 Data Transfer	5
4.4 Data Transformation	5
4.5 Data Querying	5
5. Environment	6
6. Test Data	6
7. Software Test Report	7

1. Test Objectives

Testing in this project focuses on the validation of user credentials to access the object store website, ensuring that users can access the data. Once it has been verified that user credentials can login to the store, making sure the object store is accessible outside of the local network is another key focus of test validation.

The data lake implementation features a heavy emphasis on the collection and organization of data for business analytics. As a result, ensuring that the data is properly transformed at each level of the object store is crucial in supporting a functional database.

2. Testing Scope

There are several features that will be evaluated over the course of the creation of this project. These include:

- Object Store Connectivity
- Object Store Storage
- Scraper API Access
- Raw Data Formatting
- Data Querying
- User Credentials for Object Store
- Policy Access in Object Store
- Data Transferring from User to Object Store
- Database Schema for Analyzing Car Listings

3. Test Cases

3.1 Object Store Credentials

- Description: This test case requires the tester to enter the hostname of the object store into an internet browser, then enter a valid user credential set on the login screen. Once logged in, the tester should try to create a new user credential.
- Expected Result: The tester is redirected to the object store dashboard. The account should be allowed to create new users.

3.2 Network Access

- Description: The object store is connected to by a computer or mobile device on a separate wireless network than the one the object store is hosted on, then the tester tries to login using a valid credential.
- Expected Result: The tester is redirected to the object store dashboard.

3.3 Data Transfer

- Description: The scraper program will pull data from a specified location on the host computer, then send the raw JSON file to the proper bucket in the object store
- Expected Result: A JSON file holding car listings will appear within the “lakeraw” bucket of the object store.

3.4 Data Transformation

- Description: The raw car listings from the earlier file are accessed by a program that runs the listings through Apache Parquet to organize them into tables, then writes the tabled listings into a new bucket.
- Expected Result: The Parquet file will appear in a new bucket labeled “lakesilver” in the object store.

3.5 Data Querying

- Description: The tables from the “lakesilver” bucket are accessed by a querying program, which collects all listings that fulfill the necessary criteria for analysis. Once complete, these listings are copied and grouped into a new table, then sent to the object store in the final bucket layer.
- Expected Result: A new data table file is added to the “lakegold” bucket in the object store, with much fewer listings than the files in the “lakesilver” bucket, and with each listing having similar characteristics.

4. Procedures

4.1 Sign In

- Step 1: Enter server hostname.
- Step 2: Enter TestUser credentials.
- Step 3: Navigate to the Access tab.
- Step 4: Click Add User button.
- Step 5: Enter DataLakeReader and TestRead for Username and Password

- Step 6: Give the user read-only privileges.
- Step 7: Save the user settings
- Step 8: Logout of the TestUser account and login to the DataLakeReader account.
- Expected: TestUser can create new accounts, and users with limited privileges, like DataLakeReader, can only access specific policies, like reading objects from the buckets

4.2 Network Access

- Step 1: Open a computer or mobile device connected to a different wireless network than the object store.
- Step 2: Enter the server hostname.
- Step 3: Enter the TestUser credentials.
- Expected: User is redirected to the object store dashboard

4.3 Data Transfer

- Step 1: Create a new JSON file holding some information to store, like car listings from a website.
- Step 2: Run the file uploading python program with the JSON as data.
- Step 3: Wait for the program to finish running.
- Step 4: Log in to the object store.
- Step 5: Navigate to the “lakeraw” bucket.
- Expected: The JSON file should be found within the “lakeraw” bucket

4.4 Data Transformation

- Step 1: Run the silver layer program with access to the “lakeraw” bucket.
- Step 2: Wait for program completion.
- Step 3: Log in to the object store.
- Step 4: Navigate to the “lakesilver” bucket.
- Expected: The JSON file should become a Parquet file and now be in the “lakesilver” bucket

4.5 Data Querying

- Step 1: Run the gold layer program with access to the “lakesilver” bucket.
- Step 2: Wait for the program to be completed.
- Step 3: Log in to the object store.
- Step 4: Navigate to the “lakegold” bucket.
- Expected: A new, smaller parquet file should be found within the “lakegold” bucket

5. Environment

The testing environment for this project requires the use of at least two computers on different networks. This will ensure that the network service portion of the data lake is functioning as intended. One computer should be running the MinIO object store database for holding data, while the other computer should have access to both an internet browser, and a python program that attempts to connect to the server API of the object store.

Additionally, the computer hosting the object store should have a reverse proxy enabled for the server's local hostname, and an external storage drive for hosting the test contents. The reverse proxy allows the server to be reachable on the wider Internet by other computers, like the second computer for testing. For testing purposes, the external storage drive creates an easier diagnostic process for checking server files to find errors and allows for the storage of as much test data as possible.

To confirm the structuring of raw data into tables and queries, external software for reading Parquet table files is needed.

6. Test Data

Test data breaks down into two major categories for the data lake: object store user accounts, and raw data sourced from websites across the Internet. For the first category, our object store server has multiple test accounts for the purpose of validation of user privileges and credentials. The first account is the TestUser account, which has the credentials of an administrator of the object store: being allowed to read and write to buckets, create new users, set policies, and run diagnostics on the server.

Later accounts that are assessed are given less privileges in the object store, to test the functionality of policy limitations, and whether these accounts can still function at their assigned level. Users like DataLakeReader and DataLakeWriter are given only the privilege to access buckets to read data or write in new data, respectively. Once testing is completed, these sample accounts are disabled, to prevent any security risk.

For the test data used in the second category, our test data was small samples from the larger batch of car listings that are stored on the server. After pulling some listings from Carfax, these were transferred across the Internet to the object store. Then, one or two of the JSON files sent over were run through the silver layer Parquet transformation program to assess if the data transformation is working as expected. Once the data is tabled, it is run through the gold layer program to ensure that our project is organizing data according to the specified schema.

This test data helps to validate several aspects of our project, from the functionality of the network accessibility to the execution of data treatment and transformation into tables.

7. Software Test Report

Requirement	Pass/Fail	Severity
User Login	Pass	Critical
Network Access	Pass	Minor
Data Transfer	Pass	Critical
Data Transformation	Pass	Moderate
Data Querying	Pass	Moderate