Final Capstone Project Work

Justice Ohene Amofa

Goodreads Data Pipeline

Trestle Academy Ghana

INGENN Engineering & Manufacturing Technical Recruiters

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Final Draft	Team	All sections being Filled	03/06/2024

Contents

1. Introduction

1.1 Introduction

The purpose of this document is to define and describe the requirements of the Goodreads Data Pipeline project. It outlines the system's functionality, architecture, and constraints to provide a comprehensive understanding of how the pipeline operates. This document serves as a guide for setting up and running the pipeline, detailing the steps and components involved in capturing, processing, and analyzing data from the Goodreads API.

The Goodreads Data Pipeline is designed to capture data in real-time from the Goodreads API using a Python wrapper. The data is stored locally before being transferred to an AWS S3 Landing Bucket. ETL (Extract, Transform, Load) jobs, written in Apache Spark and orchestrated by Apache Airflow, process the data every 10 minutes. The processed data is then loaded into a Redshift data warehouse, where it is available for analysis through an analytics module.

This document includes detailed instructions on setting up the necessary infrastructure, including EMR clusters, Redshift, and Airflow. It also provides an overview of the ETL flow, module descriptions, and scenarios for handling increased data volume and concurrency. Finally, it acknowledges the support and training provided by Trestle Academy Ghana, which was instrumental in the development of this project.

1.2 Scope of this Document

Data Extraction

- Utilize the Goodreads Python Wrapper to fetch data from the Goodreads API in real-time.
- Store the fetched data on a local disk before transferring it to AWS S3 for further processing.

Data Storage

- Implement a structured data storage strategy using AWS S3 buckets.
  - Landing Zone: For initial storage of raw data.
  - Working Zone: For staging data before transformation.
  - Processed Zone: For storing transformed data ready for loading into the data warehouse.

ETL Process

- Develop ETL jobs using Apache Spark to perform data extraction, transformation, and loading.
- Schedule ETL jobs to run every 10 minutes using Apache Airflow.
- Transform raw data into a format suitable for analysis and store it in the processed zone.

Data Warehousing

- Implement a Redshift data warehouse to store and manage the processed data.
- Load transformed data from the processed zone into Redshift staging tables.

- Perform UPSERT operations to update the Redshift data warehouse tables with new data.

Data Quality and Monitoring

- Incorporate data quality checks within the Airflow DAG to ensure data integrity at each stage of the pipeline.
- Implement monitoring and alerting mechanisms to notify of any data pipeline failures or quality issues.

Analytics and Reporting

- Develop an analytics module to query and analyze data stored in the Redshift data warehouse.
- Provide tools and interfaces for generating insights and reports based on Goodreads data.

Infrastructure Setup

- Detail the setup and configuration of necessary infrastructure components, including:
  - EMR clusters for running Spark jobs.
  - Redshift cluster for data warehousing.
  - Airflow for scheduling and orchestrating ETL jobs.

Testing and Scalability

- Use the `goodreadsfaker` module to generate large volumes of test data to validate the pipeline's performance under heavy load.
- Ensure the pipeline can handle significant data increases and maintain performance.

Documentation and Training

- Provide comprehensive documentation for setting up, running, and maintaining the pipeline.
- Acknowledge the training and support provided by Trestle Academy Ghana, which contributed to the successful completion of this project.

1.3 **Overview**

The Goodreads Data Pipeline project aims to build a comprehensive and scalable system for capturing, processing, and analyzing data from the Goodreads API. Utilizing various modules, the pipeline fetches data in real-time, stores it locally, and periodically transfers it to AWS S3. ETL jobs, written in Apache Spark and orchestrated by Apache Airflow, transform and load the data into a Redshift data warehouse every 10 minutes. The processed data is then available for analysis through an analytics module, enabling users to generate insights and reports.

The project involves setting up and configuring essential infrastructure components, including EMR clusters, Redshift, and Airflow, to ensure efficient data processing and management. Data quality checks and monitoring mechanisms are incorporated to maintain data integrity and alert users of any pipeline issues. Additionally, the pipeline's performance is tested under heavy load using the goodreadsfaker module to ensure scalability and reliability.

Through this project, we aim to create a robust data pipeline that supports real-time data capture, efficient ETL processing, and meaningful analytics, providing valuable insights from Goodreads data.

1.4 **Business Context**

In the digital age, data-driven decision-making is critical for businesses to gain a competitive edge. Goodreads, a popular platform for book enthusiasts, offers a wealth of data that can provide valuable insights into reading trends, book ratings, reviews, and user preferences. Leveraging this data can significantly benefit businesses in the publishing industry, authors, and marketers by enabling them to understand market dynamics, improve customer engagement, and tailor their strategies to meet audience demands.

The Goodreads Data Pipeline project is designed to harness the potential of Goodreads data by creating an efficient and scalable pipeline for real-time data capture, processing, and analysis. By integrating advanced technologies such as Apache Spark, AWS S3, Redshift, and Apache Airflow, the project ensures that data is not only captured and stored efficiently but also transformed and made readily available for in-depth analysis.

For publishers, understanding which genres are gaining popularity or identifying highly-rated books can inform their acquisition and marketing strategies. Authors can gain insights into reader feedback and preferences, allowing them to tailor their writing and engagement efforts. Marketers can leverage this data to create targeted campaigns, enhancing customer engagement and driving sales.

Overall, this project aims to transform raw Goodreads data into actionable insights, empowering stakeholders to make informed decisions, improve customer satisfaction, and drive business growth.

2. General Description

2.1 Product Functions

1. Real-Time Data Capture
2. Local Data Storage
3. AWS S3 Integration
4. ETL Job Management
5. Data Transformation
6. Redshift Data Warehouse
7. Data Quality Checks
8. Monitoring and Alerting
9. Analytics Module
10. Data Quality Verification for Analytics
11. Load Testing

12. EMR and Redshift Setup
13. User Interface

2.2 Similar System Information

As part of the project, it's essential to gather information about similar systems or existing solutions that address similar needs or functionalities. This helps in understanding industry standards, best practices, and potential areas for improvement or innovation. However, for this specific project, gathering information about similar systems might be limited due to the unique nature of the Goodreads Data Pipeline. Nevertheless, some aspects to consider in terms of similar systems include:

1. **ETL Pipelines in Data Engineering**: Explore existing ETL (Extract, Transform, Load) pipelines used in data engineering projects. Look for similarities in data extraction from APIs, data transformation processes, and data loading into warehouses. Examples of popular ETL tools and platforms such as Apache NiFi, Talend, or Informatica might provide insights into common ETL workflows.
2. **Real-Time Data Processing Systems**: Investigate real-time data processing systems that handle streaming data from external sources. Systems like Apache Kafka, Amazon Kinesis, or Google Cloud Dataflow might offer insights into handling real-time data ingestion and processing.
3. **Cloud-Based Data Pipelines**: Explore cloud-based data pipeline solutions offered by cloud providers such as AWS Data Pipeline, Google Cloud Dataflow, or Azure Data Factory. These platforms often provide tools and services for orchestrating ETL workflows, managing data storage, and integrating with various data sources and destinations.
4. **Analytics and Business Intelligence Platforms**: Look into analytics and business intelligence platforms that enable data analysis and visualization. Platforms like Tableau, Power BI, or Looker might provide insights into data modeling, query optimization, and dashboard creation for analyzing data stored in data warehouses like Redshift.
5. **Open-Source Projects and GitHub Repositories**: Search for open-source projects and GitHub repositories related to data pipelines, ETL frameworks, or data engineering workflows. Reviewing code repositories and project documentation can offer valuable insights into implementation strategies, architectural patterns, and best practices.

2.3 User Characteristics

Understanding the characteristics of the users who will interact with the Goodreads Data Pipeline is essential for designing a system that meets their needs and expectations. The following user profiles outline the primary users and their key characteristics:

Data Engineers

Role: Data engineers are responsible for designing, building, and maintaining the data pipeline. They ensure the pipeline operates efficiently and reliably.

Key Characteristics:

- Proficient in programming languages such as Python and SQL.
- Experienced with data processing frameworks like Apache Spark.
- Familiar with cloud services, particularly AWS (S3, EMR, Redshift).
- Knowledgeable about ETL processes and best practices.
- Skilled in using orchestration tools like Apache Airflow.
- Responsible for troubleshooting and optimizing pipeline performance.

Data Analysts

Role: Data analysts use the processed data stored in the Redshift data warehouse to generate insights and reports. They focus on extracting valuable information to support decision-making.

Key Characteristics:

- Proficient in querying databases using SQL.
- Experienced with data visualization tools (e.g., Tableau, Power BI).
- Strong analytical and problem-solving skills.
- Ability to interpret and communicate data insights effectively.
- Interested in accessing timely and accurate data for analysis.
- Works closely with data engineers to ensure data quality and availability.

Data Scientists

Role: Data scientists analyze complex datasets to develop predictive models and uncover patterns. They utilize the data pipeline to access the necessary data for their experiments and models.

Key Characteristics:

- Proficient in programming languages such as Python or R.
- Experienced with machine learning frameworks (e.g., TensorFlow, scikit-learn).
- Skilled in statistical analysis and data modeling.
- Requires access to large, clean, and well-structured datasets.
- Collaborates with data engineers to integrate new data sources and enhance data quality.
- Utilizes advanced analytical techniques to derive insights from data.

Business Intelligence (BI) Developers

Role: BI developers create and manage BI solutions that provide stakeholders with insights and visualizations. They leverage the data pipeline to ensure data is available for BI tools.

Key Characteristics:

- Proficient in BI tools such as Tableau, Power BI, or Looker.
- Experienced in creating dashboards and reports.
- Strong understanding of data warehousing concepts.
- Skilled in SQL and data modeling.
- Focused on delivering accurate and actionable insights to business users.
- Works with data engineers to ensure data pipelines meet BI requirements.

IT Administrators

Role: IT administrators manage the infrastructure and ensure the security, availability, and performance of the data pipeline components.

Key Characteristics:

- Experienced with cloud infrastructure management (AWS preferred).
- Knowledgeable about network security and data privacy.
- Skilled in monitoring and maintaining system performance.
- Responsible for setting up and configuring services like EC2, S3, and Redshift.
- Ensures compliance with organizational and regulatory standards.
- Provides support for infrastructure-related issues.

Stakeholders/Business Users

Role: Stakeholders and business users rely on the insights generated from the data pipeline to make informed business decisions. They are the end consumers of the reports and analytics.

Key Characteristics:

- Interested in high-level insights and actionable information.
- Focused on business outcomes and strategic decision-making.
- Relies on data-driven insights to guide business strategies.
- Requires user-friendly and accessible reports and dashboards.
- Collaborates with data analysts and BI developers to define data requirements.

By understanding these user characteristics, the project can ensure that the Goodreads Data Pipeline is designed and implemented to meet the diverse needs of its users, facilitating efficient data processing, analysis, and decision-making

2.4 User Problem Statement

Stakeholders, including data engineers, data analysts, data scientists, business intelligence (BI) developers, IT administrators, and business users, face significant challenges in efficiently capturing, processing, and analyzing Goodreads data to generate valuable insights. The existing manual processes and disjointed systems hinder their ability to make data-driven decisions, resulting in inefficiencies and missed opportunities.

2.5 User Objectives

Data Engineers

1. **Automate Data Extraction**
    - Objective: Implement an automated system for real-time data capture from the Goodreads API to reduce manual workload and ensure consistent data availability.
2. **Streamline ETL Processes**
    - Objective: Develop efficient ETL jobs using Apache Spark and manage them with Apache Airflow to ensure timely and accurate data transformation and loading.
3. **Facilitate Data Transfers**
    - Objective: Ensure seamless and secure transfer of data from local storage to AWS S3 and further processing stages.

Data Analysts

4. **Improve Data Accessibility**
    - Objective: Centralize data storage in a well-structured format within AWS S3 and Redshift to provide easy and consistent access to clean and organized data.
5. **Enable Efficient Analysis**
    - Objective: Ensure that transformed and processed data is readily available for analysis, supporting the creation of timely and accurate insights.

Data Scientists

6. **Access Comprehensive Datasets**
    - Objective: Provide up-to-date and comprehensive datasets for developing predictive models and conducting advanced data analysis.
7. **Support Advanced Analytics**
    - Objective: Ensure data quality and consistency to facilitate accurate and reliable data science experiments and modeling.

BI Developers

8. **Enhance Data Integration**
    - Objective: Integrate various data sources into a centralized data warehouse, ensuring data consistency and reliability for BI tools.
9. **Enable Robust Reporting**
    - Objective: Provide access to processed data in Redshift to create high-quality dashboards and reports that support business decision-making.

IT Administrators

10. **Optimize Infrastructure Management**
    - Objective: Simplify the setup, configuration, and management of cloud infrastructure components such as EMR, Redshift, and Airflow to ensure scalability, security, and performance.
11. **Implement Monitoring and Alerts**
    - Objective: Establish monitoring mechanisms and alert systems to promptly detect and address any issues within the data pipeline.

2.6 General Constraints

The Goodreads Data Pipeline project operates within certain constraints that must be considered to ensure successful implementation and operation. These constraints include limitations related to technology, resources, security, and operational requirements. Below are the general constraints for the project:

Technological Constraints

1. **API Rate Limits**
    - The Goodreads API has rate limits that restrict the number of requests that can be made within a specific time frame. This constraint affects the frequency and volume of data that can be captured in real time.
2. **Data Storage Limits**
    - Storage limitations on AWS S3 and Redshift, including cost constraints for large volumes of data, need to be managed effectively.
3. **Processing Power**
    - The computational power available through the chosen AWS EMR instances may limit the speed and efficiency of data processing tasks, especially under heavy data loads.

Resource Constraints

4. **Budget Constraints**
    - Financial limitations may impact the ability to scale infrastructure or invest in additional tools and resources. The project must operate within a predefined budget.

Security Constraints

5. Data Privacy and Security

- Ensuring compliance with data privacy laws and regulations, such as GDPR or CCPA, is critical. Measures must be in place to protect sensitive user data and prevent unauthorized access.

6. Access Control

- Implementing robust access controls to restrict data access to authorized personnel only. This includes managing permissions for AWS services and data warehouse access.

Operational Constraints

7. System Downtime

- Scheduled maintenance and unexpected downtimes of AWS services (S3, Redshift, EMR) or the Goodreads API can impact the availability and performance of the data pipeline.

8. Scalability Limits

- The system must be designed to handle current data loads, but scalability for future growth may be limited by existing architecture and resource constraints.

Performance Constraints

Latency Requirements

- The pipeline must process data with minimal latency to ensure timely availability for analysis and reporting. High latency can affect the real-time capabilities of the system.

Quality Assurance

- Ensuring data quality and consistency throughout the ETL process is crucial. The system must include mechanisms for validating and cleaning data, which may add complexity and processing time.

Environmental Constraints

Network Reliability

- Dependence on stable and high-speed internet connectivity for data transfer between local storage, AWS services, and the Goodreads API.

Environmental Impact

- Consideration of the environmental impact of running large-scale cloud infrastructure and making efforts to minimize energy consumption and carbon footprint.

3. Functional Requirements

Functional Requirements

The functional requirements for the Goodreads Data Pipeline project define the specific behaviors and functionalities that the system must provide to meet user needs and project objectives. These requirements ensure that all aspects of data capture, processing, storage, and analysis are effectively covered.

Data Capture and Ingestion

1. **Real-Time Data Capture**
   - The system must capture data from the Goodreads API in real-time using the Goodreads Python Wrapper.
   - The system must handle API rate limits to ensure continuous data capture without exceeding allowed requests.
2. **Local Data Storage**
   - The system must store captured data on the local disk temporarily before transferring it to the cloud.

Data Transfer

3. **Data Transfer to AWS S3**
   - The system must periodically move data from local storage to the AWS S3 Landing Bucket.
   - Data transfer must be secure and efficient to prevent data loss or corruption.

ETL (Extract, Transform, Load) Processing

4. **ETL Job Scheduling**

- The system must schedule ETL jobs to run every 10 minutes using Apache Airflow.
- ETL jobs must be automatically triggered upon data arrival in the AWS S3 Landing Bucket.

5. **Data Transformation**
   - The system must transform raw data into a suitable format using Apache Spark.
   - Transformations must include data cleaning, normalization, and enrichment processes.
6. **Data Loading to Redshift**
   - The system must load transformed data into Redshift staging tables.
   - The system must perform UPSERT operations to update data warehouse tables efficiently.

Data Quality and Monitoring

7. **Data Quality Checks**
   - The system must perform data quality checks after ETL job execution to ensure data integrity.
   - The system must alert relevant personnel in case of data quality issues.
8. **Monitoring and Alerts**
   - The system must monitor the ETL process and infrastructure performance.
   - The system must send alerts for any failures, delays, or anomalies detected during the ETL process.

Data Storage and Management

9. **AWS S3 Data Zones**
   - The system must use a structured data storage strategy with distinct zones: Landing, Working, and Processed.
   - The system must move data between these zones as part of the ETL process.
10. **Redshift Data Warehouse**
    - The system must store processed data in the Redshift data warehouse for efficient querying and analysis.
    - The system must maintain data organization and indexing in Redshift to support fast query performance.

Analytics and Reporting

11. **Analytics Queries**
    - The system must support the execution of predefined analytics queries on the Redshift data warehouse.
    - The system must validate analytics results to ensure accuracy and reliability.
12. **Data Access for Analysis**
    - The system must provide data analysts and scientists with access to the processed data in Redshift.

- The system must support integration with analytics tools such as Tableau or Power BI.

Infrastructure Setup and Management

13. **Airflow Setup**
    - The system must include instructions for setting up Apache Airflow using AWS CloudFormation.
    - The system must configure Airflow to manage ETL job scheduling and monitoring.
14. **EMR Cluster Setup**
    - The system must provide guidance for setting up an AWS EMR cluster to run Spark jobs.
    - The system must install necessary dependencies such as psycopg2 and boto3 on the EMR cluster.
15. **Redshift Cluster Setup**
    - The system must include steps for setting up a Redshift cluster, either manually or using an automated script.

Load Testing

16. **Load Testing with Fake Data**
    - The system must include a module (goodreadsfaker) to generate fake data for testing the ETL pipeline under heavy load.
    - The system must support load testing scenarios to validate performance and scalability.

4. Interface Requirements

4.1 User Interfaces

The user interfaces (UIs) for the Goodreads Data Pipeline project provide interactive platforms for users to interact with and manage various aspects of the data pipeline. These interfaces aim to streamline user workflows, enhance usability, and provide access to essential functionalities. The following user interfaces are included in the project:

Apache Airflow UI

1. **Dashboard Overview**: The Apache Airflow UI provides a centralized dashboard that displays an overview of all DAGs (Directed Acyclic Graphs) and their current status.
2. **DAG Details**: Users can access detailed information about each DAG, including task dependencies, execution history, and logs.
3. **Task Execution**: The UI allows users to manually trigger task executions, monitor task progress, and view task logs in real-time.
4. **Scheduler Configuration**: Users can configure scheduling parameters for ETL jobs, including task intervals, start dates, and execution timeouts.

Redshift Query Editor

5. **SQL Query Execution**: The Redshift Query Editor enables users to write and execute SQL queries directly against the Redshift data warehouse.
6. **Query History**: Users can access a history of executed queries, including query duration, execution time, and results.
7. **Schema Exploration**: The UI provides tools for exploring the Redshift data warehouse schema, including table structures, column names, and data types.

Analytics Tools Integration

8. **Integration with BI Tools**: The project supports integration with popular business intelligence (BI) tools such as Tableau, Power BI, or Looker.
9. **Data Visualization**: Users can create interactive dashboards, reports, and visualizations using BI tools to analyze processed data stored in Redshift.

Custom Monitoring Dashboard

10. **System Health Metrics**: The project includes a custom monitoring dashboard to display key system health metrics, such as CPU utilization, memory usage, and network throughput.
11. **Alerts and Notifications**: Users can set up alerts and notifications for critical system events, such as ETL job failures, data quality issues, or infrastructure downtime.

Command Line Interfaces (CLIs)

12. **EMR and Redshift CLIs**: The project provides command line interfaces (CLIs) for managing AWS EMR clusters and Redshift clusters.
13. **Configuration Management**: Users can use CLIs to configure cluster settings, scale resources, and manage security groups.

4.2 Hardware Interfaces

The Goodreads Data Pipeline project relies on various hardware components to ensure smooth operation and efficient data processing. These hardware interfaces include cloud-based

infrastructure and local machines used during development and testing. Below are the key hardware interfaces involved in the project:

AWS Elastic MapReduce (EMR)

1. **Cluster Nodes**
    - **Master Node**: Manages the cluster and coordinates distributed processing tasks.
    - **Core Nodes**: Handle data storage and processing tasks. Typically, a 3-node cluster with m5.xlarge instances is used, each offering:
        - 4 vCPUs
        - 16 GiB memory
        - 64 GiB EBS storage
2. **Networking**
    - **VPC (Virtual Private Cloud)**: Ensures secure network isolation and connectivity between EMR nodes.
    - **Security Groups**: Manage inbound and outbound traffic rules for the cluster.
3. **Storage**
    - **EBS Volumes**: Attached to each node for persistent storage of intermediate data and logs.

AWS Redshift

4. **Cluster Nodes**
    - **Leader Node**: Coordinates query processing and distribution among compute nodes.
    - **Compute Nodes**: Perform actual data processing and storage tasks. Typically, a 2-node cluster with dc2.large instances is used, each offering:
        - 2 vCPUs
        - 15 GiB memory
        - SSD-based storage optimized for high I/O performance
5. **Networking**
    - **VPC**: Provides network isolation and secure access to the Redshift cluster.
    - **Security Groups**: Control access to the Redshift cluster from other AWS services and external sources.

AWS S3

6. **Storage Buckets**
    - **Landing Zone**: Stores raw data captured from the Goodreads API.
    - **Working Zone**: Temporary storage for data undergoing transformation.
    - **Processed Zone**: Stores transformed and cleaned data ready for loading into Redshift.
7. **Networking**
    - **VPC Endpoints**: Enable secure access to S3 buckets from within the VPC.

Local Development Environment

8. **Development Machines**
    - **Specifications**: Developers typically use machines with the following minimum specifications:
        - Processor: Quad-core CPU
        - Memory: 16 GiB RAM
        - Storage: 256 GiB SSD
        - Operating System: Linux, macOS, or Windows
9. **Networking**
    - **SSH Access**: Secure Shell (SSH) is used to connect to EC2 instances and manage EMR clusters.
    - **Internet Connectivity**: Required for accessing AWS services and the Goodreads API.

EC2 Instances for Airflow

10. **Airflow EC2 Instance**
    - **Instance Type**: Typically an m5.large instance with the following specifications:
        - 2 vCPUs
        - 8 GiB memory
        - 64 GiB EBS storage for the Airflow home directory
11. **RDS Instance for Airflow Metadata Database**
    - **Database Instance**: Typically a PostgreSQL RDS instance.
        - Instance Type: db.t3.medium
        - 2 vCPUs
        - 4 GiB memory
        - SSD storage for fast read/write operations

Hardware Dependencies

12. **Interconnectivity**
    - **VPC Peering**: Ensures seamless communication between EMR, Redshift, EC2, and S3 within a secure VPC environment.
    - **IAM Roles and Policies**: Manage permissions and access control for hardware resources.

4.3 Communications Interfaces

The Goodreads Data Pipeline project employs a variety of communication interfaces to ensure seamless data transfer, coordination, and integration across different components and services. These interfaces facilitate interactions between users, applications, and infrastructure, providing the necessary connectivity for the pipeline's operations.

API Interfaces

1. **Goodreads API**
    - **Purpose**: To fetch real-time data from Goodreads.
    - **Protocol**: RESTful HTTP/HTTPS.
    - **Authentication**: OAuth for secure data access.
    - **Usage**: The Goodreads Python Wrapper interacts with the Goodreads API to capture data such as books, reviews, and ratings.

Database Interfaces

2. **Amazon Redshift**
    - **Purpose**: To store and manage transformed data for analysis.
    - **Protocol**: JDBC/ODBC for querying and data manipulation.
    - **Authentication**: IAM roles, AWS credentials.
    - **Usage**: ETL jobs load data into Redshift, and data analysts use SQL queries for analysis and reporting.

Storage Interfaces

3. **Amazon S3**
    - **Purpose**: To store raw, working, and processed data in different zones.
    - **Protocol**: HTTP/HTTPS for data upload and download.
    - **Authentication**: IAM roles and policies for secure access.
    - **Usage**: ETL jobs interact with S3 buckets to transfer data between local storage, landing, working, and processed zones.

Workflow Management Interfaces

4. **Apache Airflow**
    - **Purpose**: To manage and schedule ETL jobs.
    - **Protocol**: HTTP/HTTPS for web-based UI, RPC for internal communication.
    - **Authentication**: Username and password for web UI access, key-based SSH for remote job submission.
    - **Usage**: Users interact with the Airflow UI to configure and monitor DAGs, while the Airflow scheduler manages job execution.

Network Interfaces

5. **AWS VPC (Virtual Private Cloud)**
    - **Purpose**: To provide a secure and isolated network environment.
    - **Protocol**: TCP/IP for internal communication, SSH for secure access.

- **Authentication**: Key pairs for SSH access, security groups for traffic control.
- **Usage**: VPC connects all AWS services (EC2, EMR, Redshift, S3) ensuring secure communication within the pipeline.

Secure Access Interfaces

6. **SSH (Secure Shell)**
   - **Purpose**: To securely connect and manage remote servers.
   - **Protocol**: TCP/IP over port 22.
   - **Authentication**: SSH key pairs.
   - **Usage**: Developers and system administrators use SSH to access EC2 instances, manage EMR clusters, and submit Spark jobs securely.

Messaging and Notification Interfaces

7. **Amazon SNS (Simple Notification Service)**
   - **Purpose**: To send notifications and alerts.
   - **Protocol**: HTTP/HTTPS, Email, SMS.
   - **Authentication**: IAM roles and policies.
   - **Usage**: SNS sends notifications about ETL job status, system health, and data quality issues to administrators and users.
8. **Amazon SQS (Simple Queue Service)**
   - **Purpose**: To manage task queues for ETL processes.
   - **Protocol**: HTTP/HTTPS.
   - **Authentication**: IAM roles and policies.
   - **Usage**: SQS queues tasks for the ETL jobs, ensuring reliable execution and retry mechanisms.

Data Analysis and Visualization Interfaces

9. **BI Tools Integration (Tableau, Power BI, Looker)**
   - **Purpose**: To visualize and analyze data stored in Redshift.
   - **Protocol**: SQL via JDBC/ODBC, RESTful APIs for data import.
   - **Authentication**: AWS credentials, OAuth for secure data access.
   - **Usage**: Data analysts use BI tools to create dashboards, reports, and visualizations based on the data processed by the pipeline.

Monitoring and Logging Interfaces

10. **AWS CloudWatch**
    - **Purpose**: To monitor and log infrastructure and application performance.
    - **Protocol**: HTTPS for data collection and API access.
    - **Authentication**: IAM roles and policies.

- **Usage**: CloudWatch collects logs and metrics from EC2, EMR, and other AWS services, providing insights into system health and performance.

11. **Airflow Logs**
    - **Purpose**: To provide detailed logs of ETL job execution.
    - **Protocol**: HTTP/HTTPS for web-based access.
    - **Authentication**: Username and password for web UI access.
    - **Usage**: Users access Airflow logs via the web UI to troubleshoot and monitor ETL job execution.

4.4 Software Interfaces

5. Performance Requirements

Performance Requirements

The Goodreads Data Pipeline must meet specific performance criteria to ensure efficient and reliable operation. The following performance requirements outline the necessary standards for data processing, storage, and analysis within the project:

Data Ingestion and Processing

1. **Data Ingestion Rate**
    - The system must be capable of ingesting data from the Goodreads API in real-time, with a maximum latency of 1 second from data capture to local storage.
2. **ETL Job Frequency**
    - ETL jobs must be scheduled and executed every 10 minutes using Apache Airflow.
3. **Data Transfer to S3**
    - Data must be transferred from local storage to the AWS S3 Landing Bucket within 5 minutes of capture, ensuring minimal delay in the ETL pipeline.

Data Transformation and Loading

4. **ETL Job Execution Time**
    - Each ETL job must complete its processing, including data transformation and loading into the Redshift staging tables, within 5 minutes.
5. **Spark Job Performance**
    - Spark jobs must be optimized to handle large datasets, with the ability to process and transform up to 11.4 GB of data per run, equating to approximately 68 GB per hour and 1.6 TB per day.

6. **Data Load to Redshift**
    - The data load process from S3 Processed Zone to Redshift staging tables must complete within 3 minutes per job run.

Data Storage and Retrieval

7. **Redshift Query Performance**
    - Queries executed on the Redshift data warehouse must return results within 2 seconds for simple queries and within 10 seconds for complex analytical queries.
8. **Data Availability**
    - The processed data in Redshift must be available for querying and analysis within 2 minutes of the ETL job completion.

Data Quality and Integrity

9. **Data Quality Checks**
    - Data quality checks performed by Airflow DAGs must complete within 1 minute, ensuring that data integrity is maintained before making it available for analysis.
10. **Error Handling and Recovery**
    - In case of ETL job failures, the system must automatically retry the job up to 3 times, with a maximum retry interval of 2 minutes. Notifications of failures must be sent within 1 minute of detection.

Scalability and Load Handling

11. **Scalability**
    - The system must be capable of scaling to handle data volume increases by 100x without significant degradation in performance. This includes scaling EMR clusters and Redshift nodes as needed.
12. **Concurrent Users**
    - The system must support up to 100 concurrent users querying and analyzing data in Redshift without impacting performance.

System Availability and Reliability

13. **Uptime**
    - The system must maintain an uptime of 99.9%, ensuring high availability and reliability.
14. **Backup and Recovery**
    - Data backups must be performed daily, with the ability to restore data within 1 hour in case of data loss or corruption.

Resource Utilization

15. **Resource Optimization**
    - The system must efficiently utilize AWS resources (EMR, Redshift, S3, EC2) to minimize costs while maintaining performance standards.
16. **Monitoring and Alerts**
    - The system must continuously monitor performance metrics and resource utilization, with alerts configured to notify administrators of any performance degradation or anomalies within 1 minute of detection.

6. Other non-functional attributes

.

6.1 Security

Ensuring the security of the Goodreads Data Pipeline is paramount to protect sensitive data, maintain system integrity, and comply with regulatory requirements. The following security attributes outline the necessary measures to secure data and infrastructure within the project:

Data Security

1. **Data Encryption**
    - **At Rest**: All data stored in S3, Redshift, and local disks must be encrypted using AWS Key Management Service (KMS) with AES-256 encryption.
    - **In Transit**: Data transferred between components (e.g., Goodreads API, S3, Redshift, EMR) must be encrypted using SSL/TLS to ensure secure communication.
2. **Access Controls**
    - **IAM Roles and Policies**: Implement strict IAM roles and policies to control access to AWS resources, ensuring that only authorized users and services have the necessary permissions.
    - **Bucket Policies and ACLs**: Configure S3 bucket policies and access control lists (ACLs) to restrict access to sensitive data.

Network Security

3. **Network Isolation**
    - **VPC Configuration**: Use a Virtual Private Cloud (VPC) to isolate the pipeline's infrastructure from the public internet, and segment the network using private and public subnets.

- **Security Groups**: Implement security groups to control inbound and outbound traffic to EC2 instances, EMR clusters, and Redshift, allowing only necessary traffic.

4. **Secure Communication Channels**
    - **SSH and VPN**: Use SSH for secure remote access to EC2 instances and configure a VPN for secure access to the VPC.
    - **SSHTunnel**: Utilize sshtunnel for securely submitting Spark jobs to the EMR cluster from the EC2 instance.

Application Security

5. **Authentication and Authorization**
    - **OAuth**: Use OAuth for secure authentication when accessing the Goodreads API.
    - **Airflow Access Control**: Secure the Apache Airflow web interface with username and password authentication, and implement role-based access control (RBAC) to restrict access to critical operations.
6. **API Security**
    - **Rate Limiting**: Implement rate limiting for API calls to prevent abuse and ensure fair usage.
    - **API Keys**: Use API keys to authenticate and authorize access to the Goodreads API and other integrated services.

Monitoring and Logging

7. **Audit Logs**
    - **AWS CloudTrail**: Enable AWS CloudTrail to capture API calls and log changes to AWS resources, providing an audit trail for security and compliance.
    - **Airflow Logs**: Maintain detailed logs of Airflow job executions and user activities for monitoring and troubleshooting.
8. **Intrusion Detection**
    - **AWS GuardDuty**: Use AWS GuardDuty to monitor and detect potential security threats within the AWS environment.
    - **VPC Flow Logs**: Enable VPC Flow Logs to capture and monitor network traffic within the VPC for suspicious activities.

Data Integrity

9. **Data Quality Checks**
    - **Airflow DAGs**: Implement data quality checks within Airflow DAGs to validate data integrity after ETL processing and before loading into the data warehouse.
10. **Backup and Recovery**
    - **Automated Backups**: Schedule automated backups of data stored in S3 and Redshift, ensuring data can be restored in case of corruption or loss.
    - **Disaster Recovery Plan**: Develop and maintain a disaster recovery plan to quickly restore critical services and data in case of major incidents.

Compliance and Regulatory Requirements

11. **Compliance**
    - **Data Protection Regulations**: Ensure the pipeline complies with relevant data protection regulations (e.g., GDPR, CCPA) by implementing necessary security and privacy measures.
    - **AWS Compliance Programs**: Leverage AWS compliance programs and certifications (e.g., ISO 27001, SOC 2) to align with industry standards.

Incident Response

12. **Incident Response Plan**
    - **Preparation and Training**: Develop an incident response plan and provide training to the team to effectively respond to security incidents.
    - **Automated Alerts**: Configure automated alerts for security incidents and performance anomalies using AWS CloudWatch and SNS.

6.2 Binary Compatibility

This Goodreads Data Pipeline system will be compatible with any environment that supports the following software and frameworks:

- **Operating Systems**: The system can be deployed on Linux-based operating systems (Ubuntu, CentOS, Amazon Linux) and Windows Server environments.
- **AWS Services**: The pipeline is designed to be fully compatible with Amazon Web Services (AWS), including but not limited to Amazon S3, Amazon Redshift, Amazon EMR, and Amazon EC2.
- **Apache Airflow**: Compatible with Airflow 1.10 and later versions, running on EC2 instances or any other cloud or on-premise infrastructure.
- **Apache Spark**: Compatible with Spark 2.4 and later versions, running on Amazon EMR or any other distributed computing environment.
- **Python**: The system components, including the Goodreads Python Wrapper and ETL scripts, are compatible with Python 3.6 and later versions.
- **JDBC/ODBC Drivers**: The data loading and querying functionalities are compatible with standard JDBC/ODBC drivers for Amazon Redshift.

6.3 Reliability

Reliability is one of the key attributes of the system. Back-ups will be made regularly so that restoration with minimal data loss is possible in the event of unforeseen events. The system will also be thoroughly tested by all team members to ensure reliability.

6.4 Maintainability

The Goodreads Data Pipeline system shall be maintained by the Data Engineering Team at Trestle Academy Ghana. Maintenance responsibilities include monitoring the pipeline, troubleshooting issues, performing regular updates, and ensuring the system's smooth operation. If needed, maintenance tasks can be delegated to another qualified employee within the team to ensure continuity and reliable performance.

6.5 Portability

The Goodreads Data Pipeline system shall be designed to ensure portability across various environments. The system can be deployed on multiple cloud or on-premise infrastructures, including AWS (Amazon Web Services) and local servers. The use of containerization technologies such as Docker ensures that the system can be easily replicated and run in different environments without compatibility issues. This design allows for flexibility in deployment and scalability, accommodating various user and organizational needs.

6.7 Reusability

The Goodreads Data Pipeline system shall be designed to ensure that its components can be reused for various data engineering and analytics tasks. The ETL modules, data transformation scripts, and analytics queries are developed to be modular and adaptable, allowing them to be easily repurposed for different datasets and use cases. This design approach enables the system to be reused for different projects, enhancing its overall utility and value to the organization.

7. Operational Scenarios

Scenario A: Initial Item Definitions

The Goodreads Data Pipeline system shall provide functionality for users to enter information about items into the database during its initial construction and evolution. This functionality will be facilitated through a user-friendly form interface, allowing users to input item details such as title, author, genre, publication date, and other relevant information. The data entered via the form will be processed and stored in the database, ensuring accurate and comprehensive records of all items available for auction.

Scenario B: Customer Check-out

The system shall enable users to record information about customers purchasing specific items and participating in auctions. Users will have the ability to enter customer details, including name, contact information, and bid amount, via a user interface integrated with the database. Additionally, users will input the winning bid for each item, ensuring that the system maintains accurate records of customer transactions and auction outcomes.

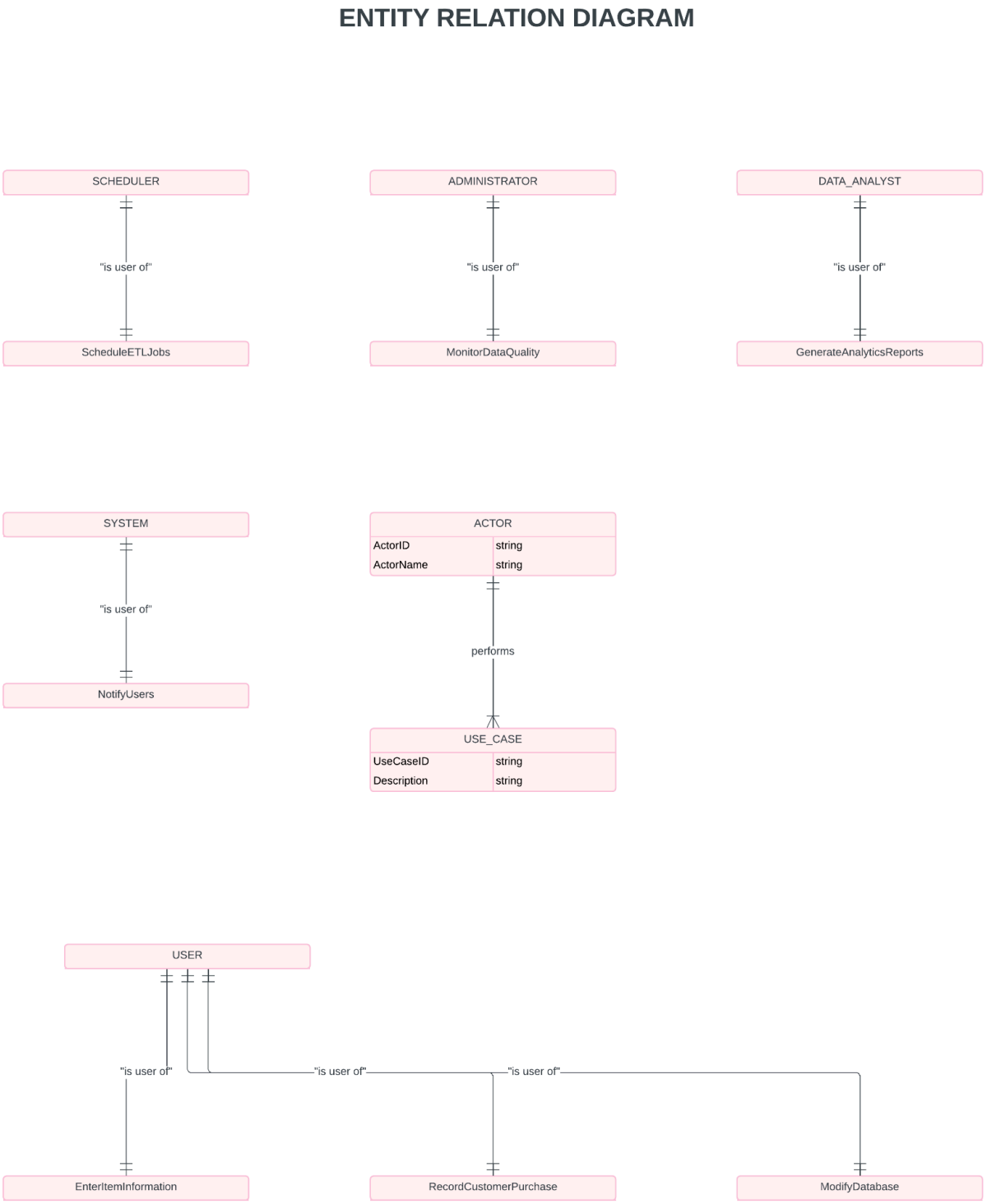
Scenario C: Database Maintenance

The system shall provide users with functionality to perform maintenance tasks on the database, including altering or deleting information after auctions have concluded. Users will have the capability to modify or remove data entries as needed, ensuring the database remains up-to-date and reflective of current inventory and auction status. These maintenance operations will be performed through a user-friendly interface, allowing users to efficiently manage database content and ensure data integrity.

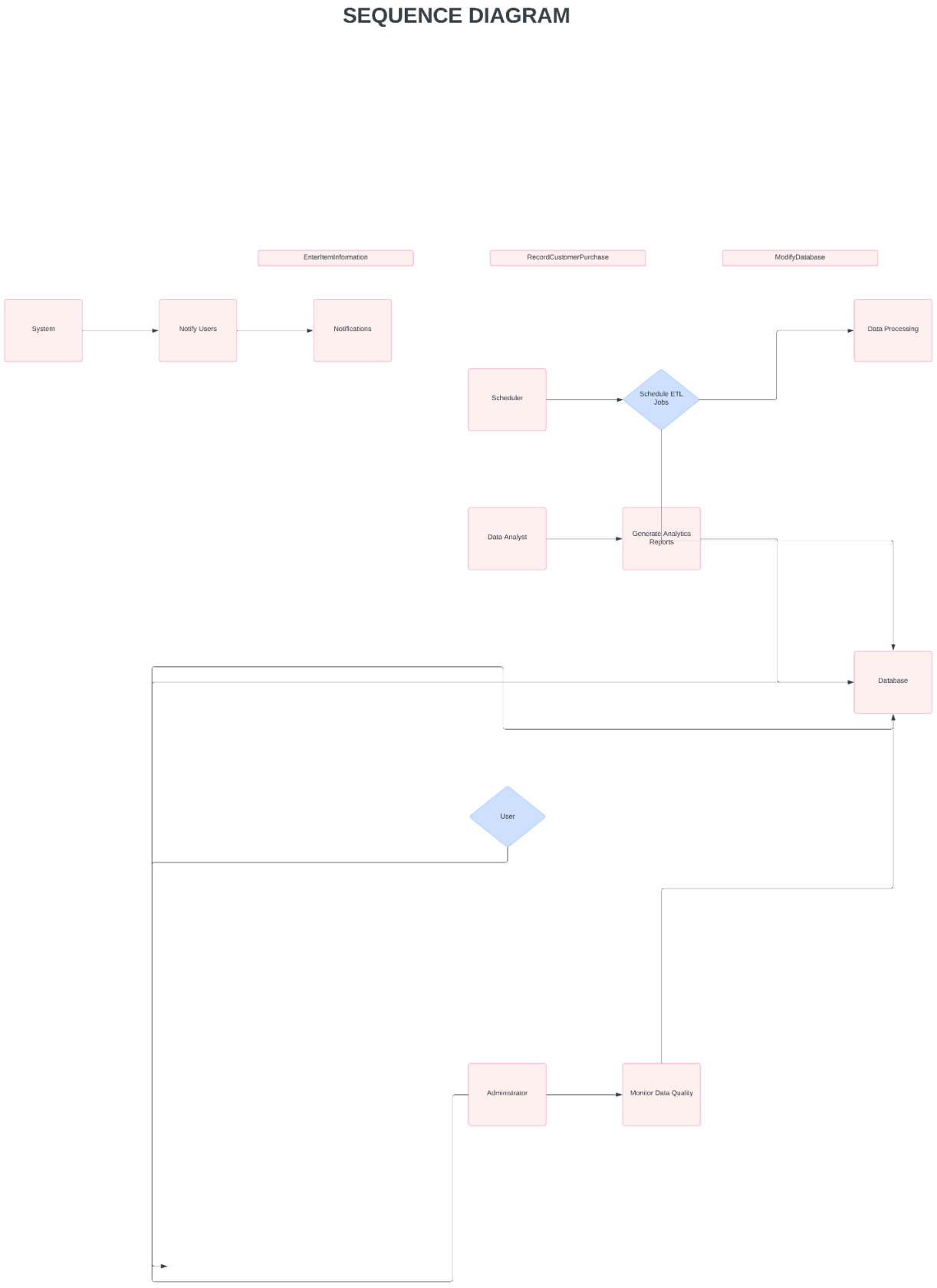
8. Preliminary Use Case Models and Sequence Diagrams

This section presents a list of the fundamental sequence diagrams and use cases that satisfy the system's requirements. The purpose is to provide an alternative, "structural" view of the requirements stated above and how they might be satisfied in the system.

8.1 Entity Relation Diagram



8.2 Sequence Diagram



9. Updated Schedule

Not Available