**High-Level Design**

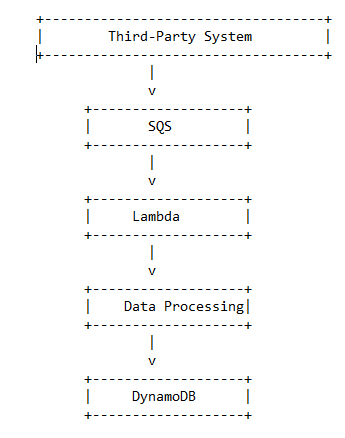
**Project 3**

**Capture transaction data from SQS queue**

**Infrastructure Components:**

* AWS SQS: Message queue service for receiving live transaction data from a third-party system.
* AWS Lambda: Serverless compute service for processing and handling the received messages.
* DynamoDB: NoSQL database for storing Account Master and Source System Master data.

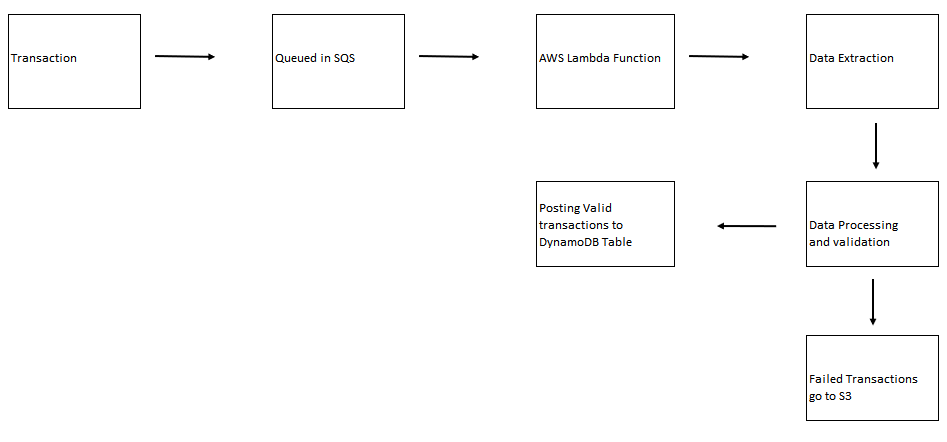
**Data Processing Flow Diagram:**

****

**Detailed Workflow:**

* Third-Party System Integration:
* The third-party system will post live transaction data to the SQS queue in JSON format.
* The SQS queue acts as a buffer, decoupling the transaction processing from the third-party system.
* Message Processing:
* AWS Lambda functions are triggered by messages from the SQS queue.
* The Lambda function retrieves the message from the queue.
* The JSON message is parsed to extract the relevant transaction details.
* Validations are performed on the transaction data.
* Transaction Creation:
* The Lambda function interacts with DynamoDB to fetch the necessary data from the Account Master and Source System Master tables.
* Based on the transaction details, three separate transactions are created for each incoming transaction.
* The transactions are recorded in the Ledger Transaction table in DynamoDB.
* Transaction Credit:
* The transactions are credited to the appropriate accounts based on the transaction type and amount.
* The "Product Sale" account, "Value Added Tax" account, and "Excise Duty" account are credited accordingly.
* Error Handling:
* If a transaction fails for any reason during processing or validation, it is flagged as an error.
* Failed transactions are recorded in a separate file for reload purposes.
* System Maintenance and Reporting:
* Regular maintenance tasks are performed to ensure the system's availability, performance, and data integrity.
* Reporting mechanisms can be implemented to generate transaction reports, monitor system performance, and track any errors or failures

**Dataset Block Diagram:**

****

**Data Models:**

* Account Master Table:
* Stores information about various accounts, including account number, name, description, and type.
* Ledger Transaction Table:
* Stores details of each transaction, such as transaction ID, voucher code, transaction type, date, account number, amount, source system ID, and source system transaction ID.
* Source System Master Table:
* Contains information about different source systems, including system ID and system name.

**Acceptance Criteria:**

* For each incoming transaction, three separate transactions are created in the Ledger Transaction table.
* The transaction amounts are credited to the appropriate accounts, namely "Product Sale," "Value Added Tax," and "Excise Duty."
* In case of any transaction failure, a separate file of failed records is generated for reloading purposes.

The provided HLD is a high-level overview of the system design. It outlines the major components and their interactions.