Exercise: Internal transfers System with an HTTP Interface

System Overview

Using golang, develop an internal transfers application that facilitates financial transactions between accounts. This application should provide HTTP endpoints for submitting transaction details and querying account balances.

A postgres database will be used to maintain transaction logs and account states.

Assumptions

- Consider the currency is the same for all accounts.
- No need to implement authn or authz

API Specifications

Account Creation Endpoint (POST)

Implement an endpoint `/accounts` that accepts JSON-formatted account ID and account initial balance

Example request body:

```
{
    "account_id": 123,
    "initial_balance": "100.23344"
}
```

Expected response is either an error or an empty response.

Account Query Endpoint (GET)

Implement an endpoint `accounts/{account_id}` that returns the account and its balance of the specified account in a JSON format

Example response for account ID 123 (if no error):

```
 {
  "account_id": 123,
  "balance": "100.23344"
 }
```

Transaction Submission Endpoint (POST)

Implement an endpoint `/transactions` that accepts JSON-formatted transaction details, including the source account ID, destination account ID, and transaction amount. The system should then process these transactions to update the account balances in the database.

Example request body:

```
{
    "source_account_id": 123,
    "destination_account_id": 456,
    "amount": "100.12345"
}
```

Technical and Coding Requirements:

1. Adherence to clean code principles is crucial, ensuring the code is readable, maintainable, and well-documented.
2. The system must maintain data integrity and consistency when processing transactions.
3. Implement error handling for various scenarios, including transaction processing failures and invalid input data.

Deliverables

A public git link with all the necessary information to run your application containing a README with:

1. instructions to install, setup, run
2. assumptions taken

DO NOT MENTION TRIPLE-A or put the exercise description in the public

Evaluation Criteria:

- The correctness of HTTP endpoint implementations for both transaction submission and balance queries.
- The system's ability to accurately and quickly process transactions and provide current account balances.
- The quality of the code, considering its organization, documentation, testing and adherence to best practices.