Rewards Calculation System Design Document

Introduction

The Rewards Calculation System is a web application that calculates reward points for customers based on their transaction history. The system allows customers to view their monthly reward points and the total reward points earned over a given period.

Goals

- Calculate reward points for each transaction based on a specific set of rules.
- Sum up the reward points for each customer over a given period.
- Provide an API endpoint to retrieve the reward points for a customer in a specific month and the total reward points.

Architecture Overview

The system follows a layered architecture pattern consisting of the following layers:

- Presentation Layer: Handles HTTP requests and responses. Includes the REST API endpoints for creating transactions and retrieving reward points.
- Service Layer: Implements the business logic for calculating reward points based on transaction history.
- Repository Layer: Provides access to the database for storing and retrieving transaction data.
- Data Layer: Persists transaction data in a relational database.

Technologies Used

- Java: Programming language for implementing the application logic.
- Spring Boot: Framework for building the web application and managing dependencies.
- Hibernate: ORM (Object-Relational Mapping) library for interacting with the database.
- MySQL: Relational database for storing transaction data.
- JUnit: Testing framework for writing unit tests.
- Maven: Build tool for managing project dependencies and building the application.

Database Schema

The database schema includes two main entities: Customer and Transaction.

Customer Entity

- id (Long): Unique identifier for the customer.

- name (String): Name of the customer.

Transaction Entity

- id (Long): Unique identifier for the transaction.
- customerId (Long): Identifier for the customer associated with the transaction.
- amount (Double): Amount of the transaction.
- transactionDate (LocalDate): Date of the transaction.

The Transaction entity has a many-to-one relationship with the Customer entity, where multiple transactions can be associated with a single customer.

API Endpoints

The system exposes the following API endpoints:

Create Transaction

- Method: POST
- Endpoint: /api/v1/transactions
- Request Body: JSON representation of the transaction details (customerId, amount, transactionDate)
- Response: 201 Created if the transaction is created successfully

Create Customer

- Method: POST
- Endpoint: /api/v1/customer
- Request Body: JSON representation of the transaction details (name)
- Response: 201 Created if the customer is created successfully

Calculate Reward Points

- Method: GET
- Endpoint: /api/v1/transactions/{customerId}/points
- Path Variables:
    - customerId (Long): Identifier of the customer
- Query Parameters:
    - yearMonth (String): Year and month in "yyyy-MM" format
- Response: JSON representation of the reward points for the customer in the specified month and the total reward points

Error Handling

The system handles the following error scenarios:

- Invalid or missing input data when creating a transaction, resulting in a Bad Request (400) response.
- Customer not found for the given customer id, resulting in a Not Found (404) response.
- Internal server errors, resulting in an Internal Server Error (500) response.

Testing

The system includes unit tests to ensure the correctness of the business logic and functionality. The tests cover different scenarios, such as calculating reward points for various transaction amounts and verifying the accuracy of the calculations.

Future Enhancements

Here are some potential areas for future enhancements:

- Implement authentication and authorization mechanisms to secure the API endpoints.
- Add pagination and sorting options for retrieving transaction data.
- Implement caching to improve performance for frequent reward points calculations.
- Expand the system to include additional metrics and analytics related to customer transactions and rewards.

Conclusion

The Rewards Calculation System provides a robust solution for calculating reward points based on customer transactions. It leverages a layered architecture, database persistence, and RESTful APIs to achieve accurate and efficient rewards calculation. The system can be easily extended and enhanced to meet future requirements and business needs.