

Retailer Rewards Program Design

This document outlines the design and implementation details of the Retailer Rewards Program.

1. Flowchart

=====

graph LR

A[Customer Makes Purchase] --> B{Calculate Reward Points};

B -- Points > 100 --> C[Award 2 points per dollar over \$100];

B -- Points between \$50 and \$100 --> D[Award 1 point per dollar between \$50 and \$100];

C --> E[Update Customer Points];

D --> E;

E --> F[Store Transaction Data];

F --> G[Return Reward Points to Customer];

2. Design Documentation

=====

2.1 System Architecture:

The system follows a layered architecture:

Presentation Layer (API):

Exposes RESTful endpoints for interacting with the rewards program.

Service Layer:

Contains the business logic for calculating reward points and managing customer data.

Data Access Layer:

Interacts with the database to store and retrieve customer and transaction information.

2.2 Class Diagram:

classDiagram

Customer -- * Transaction

CustomerService -- Customer

TransactionService -- Transactions

CustomerController -- CustomerService

TransactionsController -- TransactionService

```
class Customer {
    +Long customerId
    +String customerName
    +List<Transaction> transactions(OneToMany)
}

class Transaction {
    +Long transaction_Id
    +double transaction_amount
    +LocalDateTime transactionDate
    + Customer customer_Id (foreign key) (ManyToOne)
}

class CustomerService {
    ++addCustomer(Customer customer)
    +getRewardsPointsByCustomerId(Long customerId)
    +getAllCustomersWithRewards()
}

class CustomerController {
    +addCustomer(Customer customer)
    +getRewardPointsByCustomerId(Long customerId)
    +getAllCustomerRewards()
}

class TransactionService {
    ++addTransaction(TransactionDto transactionDto)
}

class TransactionsController {
    +addTransaction(TransactionDto transactionDto)
}
```

2.3 API Endpoints:

Endpoint	HTTP Method	Description
/api/customer/rewards/{customerId}	GET	Retrieves reward points for a specific customer
/api/customer/rewards/	GET	Retrieves reward points for all customers
/api/customer/rewards/	POST	Adds a new customer
/api/transaction/	POST	Adds a new transaction for a customer

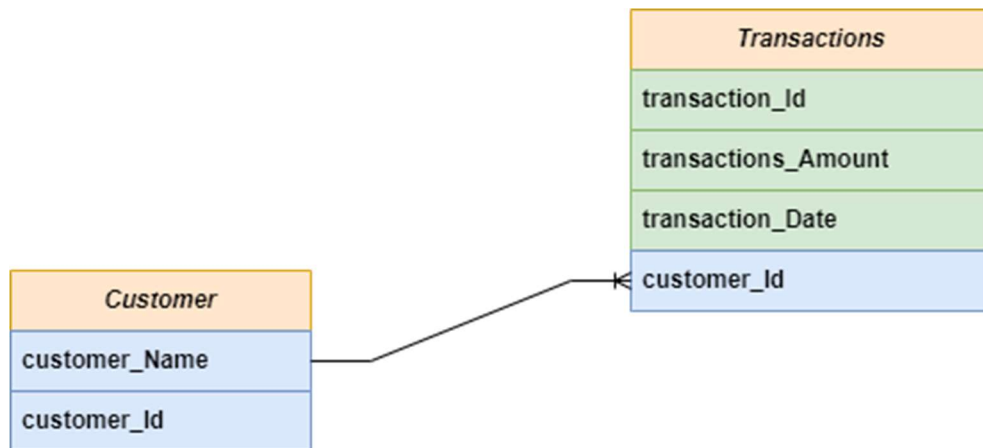
2.4 Data Model: Customer Table:

Column	Data Type	Description
customer_Id	Long	Primary key
customer_Name	String	Customer name

Transactions Table

Column	Data Type	Description
transactions_Id	Long	Primary key
customerId	Customer	Foreign key to Customer table
transaction_Amount	Double	Transaction amount
transaction_Date	LocalDate	Date of transaction

3. Entity Diagram



4. Test Case Coverage

Test Case	Description	Expected Result
Calculate points for transaction under \$50	Input: \$40	Output: 0 points
Calculate points for transaction between \$50 and \$100	Input: \$75	Output: 25 points
Calculate points for transaction over \$100	Input: \$150	Output: 150 points
Add a new customer	Input: Customer details	Output: New customer created
Add a new transaction	Input: transactions details	Output: New transactions created