Reward Dining: The Course Reference Domain

I.1. Introduction

The labs of this course teach key concepts in the context of a problem domain. The domain provides a real-world context for applying the techniques you have learned to develop useful business applications. This section provides an overview of the domain and the applications you will be working on within it.

I.2. Domain Overview

The Domain is called Reward Dining. The idea behind it is that customers can save money every time they eat at one of the restaurants participating to the network. For example, Keith would like to save money for his children's education. Every time he dines at a restaurant participating in the network, a contribution will be made to his account which goes to his daughter Annabelle for college. See the visual illustrating this business process below:

Figure 1. Papa Keith dines at a restaurant in the Reward Network

Figure 2. A percentage of his dining amount goes to daughter Annabelle's college savings

I.3. Reward Dining Domain Applications

This next section provides an overview of the applications in the Reward Dining domain you will be working on in this course.

I.3.1. The Rewards Application

The "rewards" application rewards an account for dining at a restaurant participating in the reward network. A reward takes the form of a monetary contribution to an account that is distributed among the account's beneficiaries. Here is how this application is used:

1. When they are hungry, members dine at participating restaurants using their regular credit cards.
2. Every two weeks, a file containing the dining credit card transactions made by members during that period is generated. A sample of one of these files is shown below:

```
AMOUNT    CREDIT_CARD_NUMBER    MERCHANT_NUMBER    DATE
-----------------------------------------------------------
100.00    1234123412341234      1234567890      12/29/2010
49.67     1234123412341234      0234567891      12/31/2010
100.00    1234123412341234      1234567890      01/01/2010
27.60     2345234523452345      3456789012      01/02/2010
```

3. When the application starts, several rewards are generated by the `RewardDiningPopulator` class in the `rewardsdining.rewards` package.

I.3.1.1. Public Application Interface

The `RewardNetwork` is the central interface clients such as the `RewardDiningPopulator` use to invoke the application:

```
public interface RewardNetwork {
    RewardConfirmation rewardAccountFor(Dining dining);
}
```

A `RewardNetwork` rewards an account for dining by making a monetary contribution to the account that is distributed among the account's beneficiaries.

The sequence diagram below shows a client's interaction with the application illustrating this process:

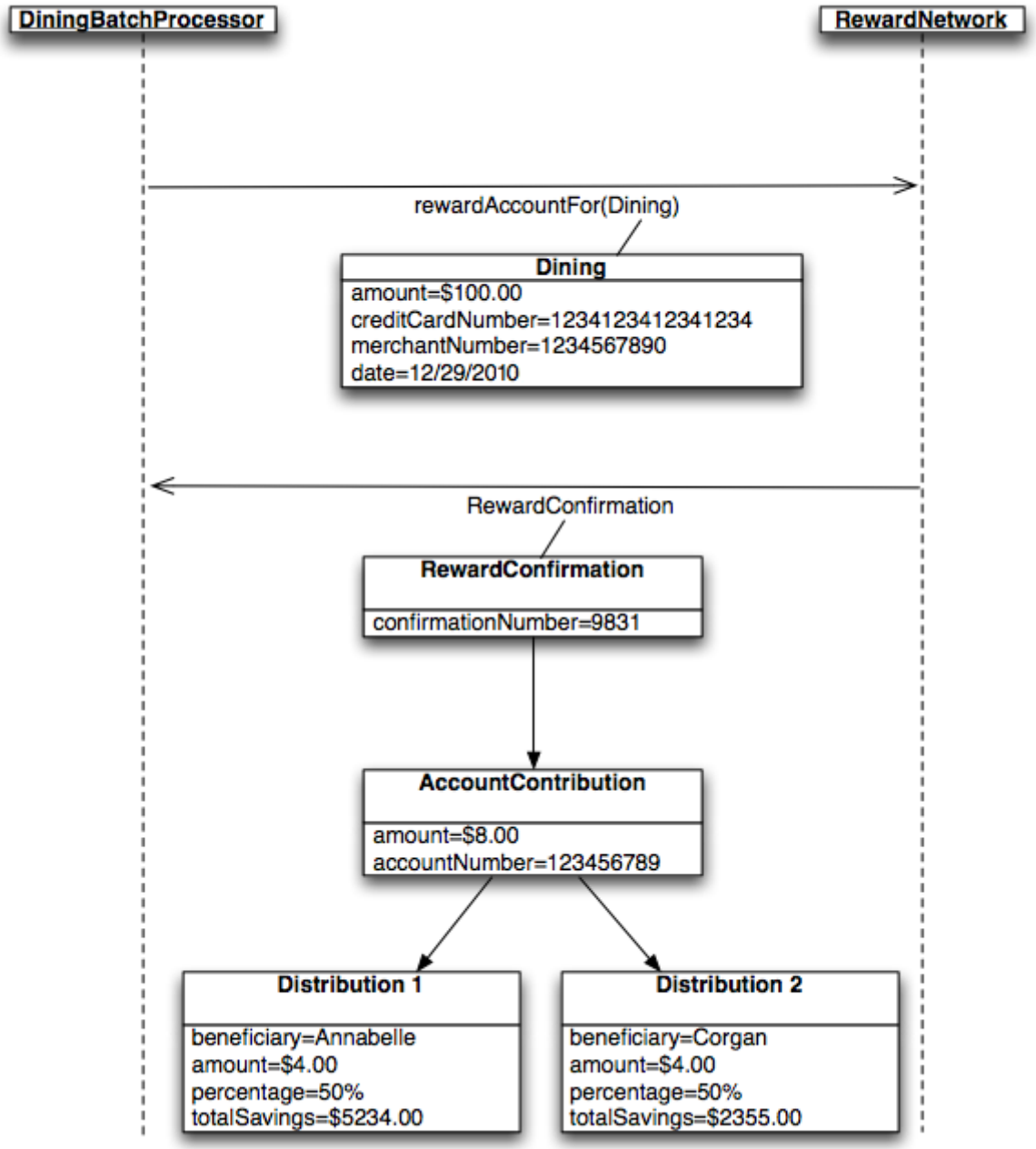

Figure 3. A client calling the `RewardNetwork` to reward an account for dining.

In this example, the account with credit card 1234123412341234 is rewarded for a $100.00 dining at restaurant 1234567890 that took place on 12/29/2010. The confirmed reward 9831 takes the form of an $8.00 account contribution distributed evenly among beneficiaries Annabelle and her brother Corgan.

I.3.1.2. Internal Application implementation

Internally, the `RewardNetwork` implementation delegates to domain objects to carry out a `rewardAccountFor(Dining)` transaction. Classes exist for the two central domain concepts of the application: `Account` and `Restaurant`. A `Restaurant` is responsible for calculating the benefit eligible to an account for a dining. An `Account` is responsible for distributing the benefit among its beneficiaries as a "contribution".

This flow is shown below:

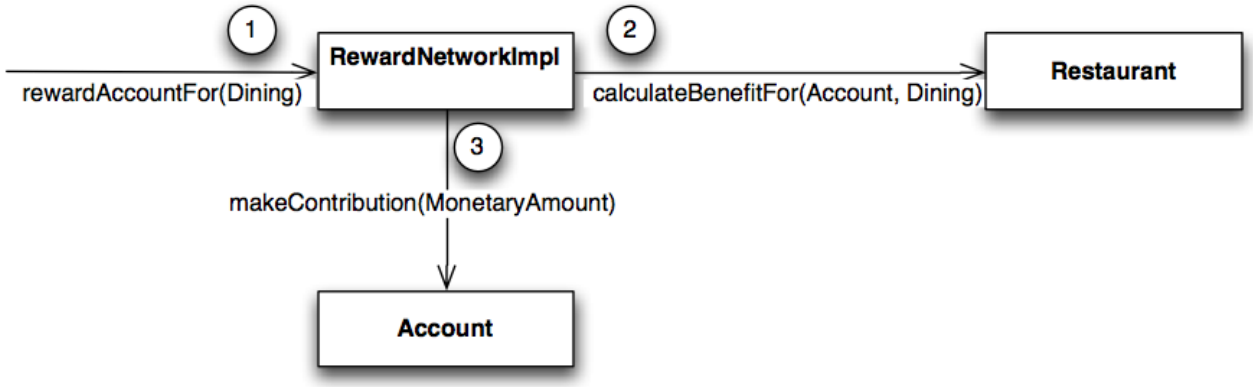

Figure 4. Objects working together to carry out the `rewardAccountFor(Dining)` use case.

The `RewardNetwork` asks the `Restaurant` to calculate how much benefit to award, then contributes that amount to the `Account`.

I.3.1.3. Supporting Reward Network Components

Account and restaurant information are stored in a persistent form inside a relational database. The `RewardNetwork` implementation delegates to supporting data access components called 'Repositories' to load `Account` and `Restaurant` objects from their relational representations.

- An `AccountRepository` is used to find an `Account` by its credit card number.
- A `RestaurantRepository` is used to find a `Restaurant` by its merchant number.
- A `RewardRepository` is used to track confirmed reward transactions for accounting purposes.

The full `rewardAccountFor(Dining)` sequence incorporating these repositories is shown in the first lab.

I.4. Reward Dining Database Schema

The Reward Dining applications use a database with this schema:

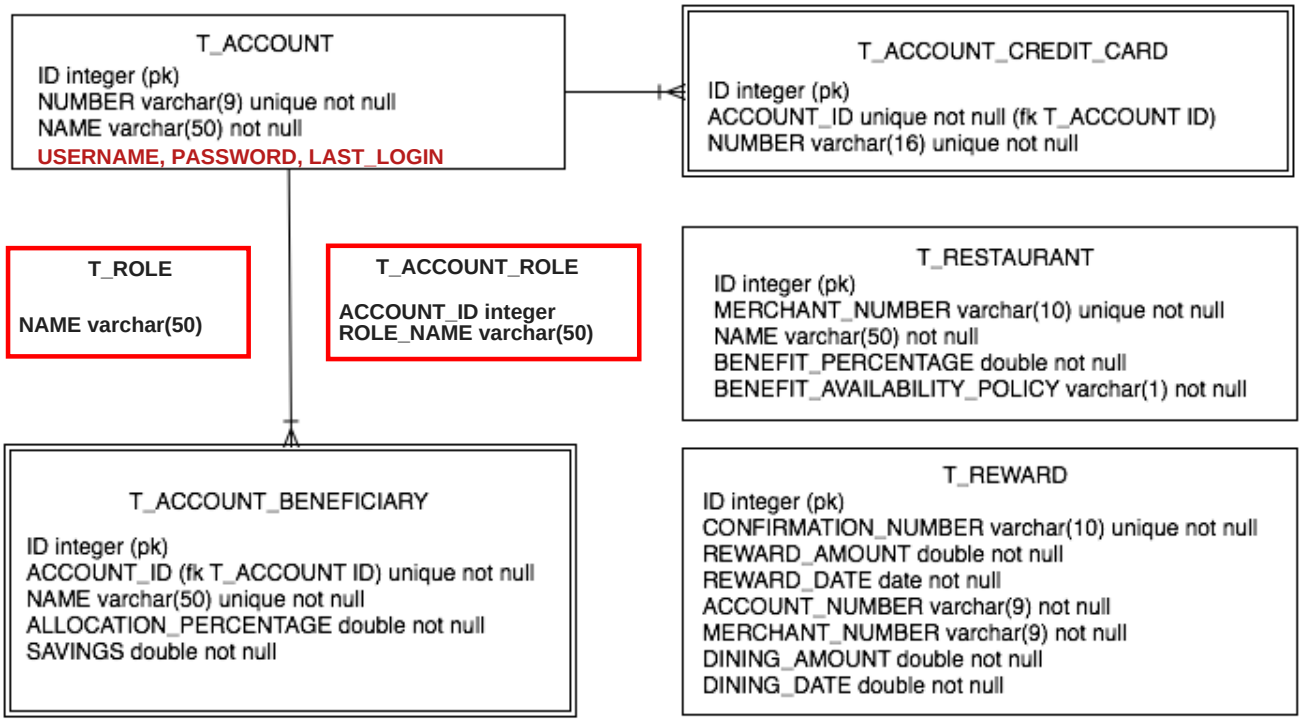

Figure 5. The Reward Dining database schema

In most of the labs a test database is provided for you. It is populated with test data by running scripts in `00-rewards-common/src/main/resources/db`. They are available as classpath resources.

There are two scripts `schema.sql` creates the necessary tables and `data.sql` adds test data.