Rental Equipment Back Office

Architectural Documentation

By: Dru Devore

# Overview

This document describes the architectural design of the Rental Equipment application following the Technical Assessment functional specification provided.

This application is the back office of a rental equipment application designed accept information entered into a user interface application by a user. Once the user enters the rental information in the user interface application the equipment rental data is transferred to this back office application for processing. This application then creates a rental agreement object that makes the rental information available to a front-end application or printed to the console.

The back office application has no front-end application or database required to operate.

# High Level Application

The application uses the toolCode, number of rental days, discount amount, and checkout date to create a Cart object and then a RentalAgreement at checkout. This application is the back-end of multiple user interfaces and must be flexible enough to handle different processing methods.

There are 2 main components in the application.

1. Cart – The object that holds the tool details, rentalDays, discount, and checkoutDate. In addition to holding the main data of the application it also contains the checkout methods that create the RentalAgreement Object.
2. RentalAgreement – This is the object that holds the Cart for access to the tool details, rentalDays, discount, and checkoutDate. It also calculates the rest of the information for the rental as well as provides a report method that prints the data to the console.

## Cart

There are 3 methods of creating the cart to get to checkout and creating the RentalAgreement. Though there are 3 methods of getting to creating to RentalAgreement all 3 follow the same basic steps.

1. Create a Cart object with the toolCode, rentalDays, discount, and checkoutDate.
2. Ask the Cart object to checkout which then creates a RentalAgreement object to be used for reporting.
3. The RentalAgreement is then used to either pull the agreement details using getter methods or printing the agreement details to the console.

The following are the 3 methods of creating the Cart and checking out.

### Creating an Empty Cart

This method is simply creating an empty cart and then filling in the cart details using the setter method. Once the cart is created the checkout() method can be called to create the RentalAgreement. If the required information in the cart is not populated or is incorrect the checkout method will throw a CheckoutException. This method is used in the case where the user interface is utilizing this application as the user is walking through the rental process.

### Creating a Completed Cart

This method is creating a fully populated cart and allowing the user of the user interface to make changes to the cart before creating the RentalAgreement. Once the user is ready to finalize the cart they call the same checkout() method as when creating the empty cart. As with the creating an empty cart if the information entered in the cart is incomplete or incorrect there will be a CheckoutException thrown. This method is also used when the user interface allows the user to walk through the rental process but will allows it to create a complete Cart with one call and make corrections as needed. There is 1 constructor available for creating a completed cart. It creates the cart with the toolCode, rentalDays, discount, and checkoutDate.

### Creating the Cart and RentalAgreement at Once

The last method is to create the Cart and RentalAgreement at the same time. This uses a static Cart method checkout(String toolCode, int rentalDays, int discount, Calendar checkoutDate) that creates the cart and then creates a RentalAgreement object. As with the other 2 methods attempting to checkout without the correct information will throw a CheckoutException. This is used with user interfaces that want to create everything at once and for batch processing.

## RentalAgreement

The creation of the RentalAgreement is only available to be called to by the Cart object in the checkout method. Once the RentalAgreement is created it can not be updated with new rental details. The RentalAgreement accepts the Cart object which contains the Tool, rentalDays, discount, and checkoutDate. All the other information available in the RentalAgreement is calculated when requested. The RentalAgreement information can be either retrieved using the getter methods or can be printed to the console using the printReport() method.

# Application Rules

There are a set of rules that must be followed to make sure that the application processes information correctly.

* Application
* The application must calculate the number of days to charge for based on
  + If the tool is charged for renting on the weekends.
  + If the tool is charged for renting
* When calculating the number of days The application must check for holidays

# Java Classes