Rental Equipment Back Office

Architectural Documentation

By: Dru Devore

Table of Contents

Overview....1
High Level Architecture....2
Application Packages....2
com.rental.equipment....2
com.rental.equipment.data....3
com.rental.equipment.exception....3
com.rental.equipment.utils....3
Application Objects....3
Cart....3
RentalAgreement....3
Tool....4
Tools....5
ToolCharge....5
calculateChargeDays....5
ToolCharges....5
CheckoutException....5
Holiday....6
Application Data....6
Tools....6
ToolCharges....6
Simple Sequence Diagram....7

Overview

This document defines the architectural design of the Rental Equipment Back Office 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 that makes the rental information available to a user interface application or printed to the console. This application has no user interface or database though it is intended to be used with multiple external user interfaces and contains its own data source.

This is a Java application written using the company standards and guidelines. Following the design below will create a functional application that is easily extendable to include features like more tools, rental conditions, and holidays just to name a few.

High Level Architecture

The application uses the tools code or Tool, number of rental days, discount amount, and checkout date to create a Cart object. When the Cart object is complete the process will use the checkout function in the Cart to create a RentalAgreement object. The RentalAgreement calculates the remaining fields for the agreement and makes them available to the user interface. The user interface can either pull the individual fields from the RentalAgreement or print an agreement report to the console.

Though the Cart and RentalAgreement are the only 2 classes the user interfaces need to function properly there are 4 other classes that the user interfaces will find useful. The Tool class holds the individual tool data which would be useful to display to a user for selection with the Tools class holding an array of tools available to the user. ToolCharge contains the charge information about each tool type and ToolCharges contains an array of tool charges available to the user.

Besides these classes there are 2 other classes. One is an exception named CheckoutException which is used when there is an error with the data in the cart at checkout time. The other is the Holiday object which is a utility class used to calculate holidays when the ToolCharge object is calculating the number of days to charge for the tool.

Application Packages

This application is organized into a main package com.rental.equipment with 3 sub packages holding different functionality.

- com.rental.equipment
- com.rental.equipment.data
- com.rental.equipment.exception
- com.rental.equipment.utils

com.rental.equipment

The main package contains the Cart and RentalAgreement classes. The user interface application will create the Cart object and when ready it will call the checkout method that will create the RentalAgreement object. These are the only 2 objects the user interfaces need to preform the back office processing for renting equipment though there are 4 other classes that user interfaces will find useful for processing a rental.

com.rental.equipment.data

The data package contains the tool data classes Tool, Tools, ToolCharge, and ToolCharges. They are the other classes that the user interfaces will find useful to complete their tasks. They will assist the user interfaces in displaying the tool and rental pricing information to the users for selection.

com.rental.equipment.exception

The exception package holds the CheckoutException that reports an error with the checkout process if the Cart object is not populated properly.

com.rental.equipment.utils

The utils package holds the Holiday object that is used to calculate if the provided day is a holiday which is important when the ToolCharge object is calculating the number of days to charge for a rental.

Application Objects

This is the description of the classes in the application.

Cart

The cart object contains the properties:

- Tool
- rentalDays
- discountPercent
- checkoutDate

The Cart can be created either blank or with all the data require using either a tool code or a Tool object. This gives the user interfaces the flexibility to create the cart in any way needed to satisfy their requirements.

The cart also contains getters and setters for all the properties along with a checkout method to create and return the RentalAgreement. As mentioned above if the Cart properties are not populated properly when checkout is called it will throw a CheckoutException.

RentalAgreement

The creation of the RentalAgreement is only available to the Cart object in the checkout method. Once the RentalAgreement is created it can not be updated with new rental details. If the rental details are incorrect a new RentalAgreement will have to be created. The RentalAgreement constructor accepts a Cart object which contains the following properties that are made available through the RentalAgreement:

- Tool
- rentalDays

- discountPercent
- checkoutDate

In addition to this information it holds the following properties that are calculated at creation time:

- chargeDays – This is calculated by the ToolCharges class and is described below.
- dailyRentalCharge – This is retrieved from the ToolCharges class.
- discountAmount – This is calculated using the preDiscountCharge * the discountPercent converted to a decimal.
- dueDate – This is calculated by adding the rentalDays to the checkoutDate.
- finalCharge – This is calculated by subtracting the discountAmount from the preDiscountCharge.
- preDiscountCharge – This is calculated by using the dailyRentalCharge * chargeDays.

The RentalAgreement information can be either retrieved using the getter methods or can be printed to the console using the printReport() method. The report must be print all values in the RentalAgreement in the following structure:

Tool code: LADW

Tool type: Ladder

…

Final charge: $9.99

The values in the RentalAgreement report must have the following formats:

with formatting as follows:

- Date mm/dd/yy
- Currency $9,999.99
- Percent 99%

Tool

The Tool class holds the tool properties:

- code
- type
- brand

The properties are available through getters. There are no setters because once the Tool is created it cannot be changed. The Tool objects are only created through the Tools class.

Tools

The Tools class creates an array of Tool objects in its static initializeTools method. Besides the initializeTools method it has 3 other static methods. One returns the array of tools and the other 2 allow you to query a tool by either the code or a combination of the type and brand. The tools array must be checked to be not null before attempting to use it or retrieving it. If it is null the initializeTools method must be called before allowing access.

ToolCharge

The ToolCharge class holds the properties

- toolType
- dailyCharge
- weekdayCharge
- weekendCharge
- holidayCharge

It is used to provide the tool rental charge information to the user interface and the RentalAgreement. It is also used to calculate the number of days to charge the rental for based on the checkout date and number of days to rent using the calculateChargeDays(Calendar checkout, int rentDays) method. In addition to that it has getters for all the properties but like the Tools class no setters as the properties cannot be changed once initialized.

calculateChargeDays

The rules for figuring out the charge days is as follows.

- If weekdayCharge is set to false we don’t charge for weekday rentals.
- If wekendCharge is set to false we don’t charge for weekend rentals.
- If holidayCharge is set to false we don’t charge for holiday rentals.

The calculateChargeDays uses the rules above to calculate the days the tool should be charged.

ToolCharges

The ToolCharges class creates an array or ToolCharge objects during its static initializeToolCharges method. Besides the initializeToolCharges method there are 2 other methods. One to return the array of ToolCharge objects and the other to look up a ToolCharge based on the toolType. The tool charge array must be checked to be not null before attempting to use it or retrieving it. If it is null the initializeToolCharges method must be called before allowing access.

CheckoutException

The CheckoutException class is a custom exception that extends Exception and implements the CheckoutException(String message) constructor to create a new exception when needed.

CheckoutException is thrown when the checkout is attempted and one or more of the following conditions exist.

- There is a null tool object.
- Rental day count is less than 1.
- Discount percent is not in the range of 0-100.
- The checkout date is in the past.

Holiday

Holiday is a helper class that has 1 public static method isHoliday(Calendar day). It has only one purpose which is to answer the question if the day that is represented by the day is considered a holiday in the application.

The holidays observed by this application are as follows.

1 Independence Day – Is July 4th. If it falls on the weekend, it is observed on the closest weekday.

1.1 If Sat, then Friday before.

1.2 If Sunday, then Monday after.

2 Labor Day – First Monday in September.

Application Data

For the application to function properly there must be data loaded. As mentioned above the Tools and ToolCharges classes create the Tool and ToolCharge objects respectively using the static initialization methods. Below are tables for what data must be created in the initialization methods of each class.

Tools

Tool Code	**Tool Type**	**Brand**
CHNS	Chainsaw	Stihl
LADW	Ladder	Werner
JAKD	Jackhammer	DeWalt
JAKR	Jackhammer	Ridgid

ToolCharges

Tool Type	**Daily Charge**	**Weekday Charge**	**Weekend Charge**	**Holiday Charge**
Ladder	$1.99	Yes	Yes	No
Chainsaw	$1.49	Yes	No	Yes
Jackhammer	$2.99	Yes	No	No

Example Sequence Diagram