Tool Rental POS

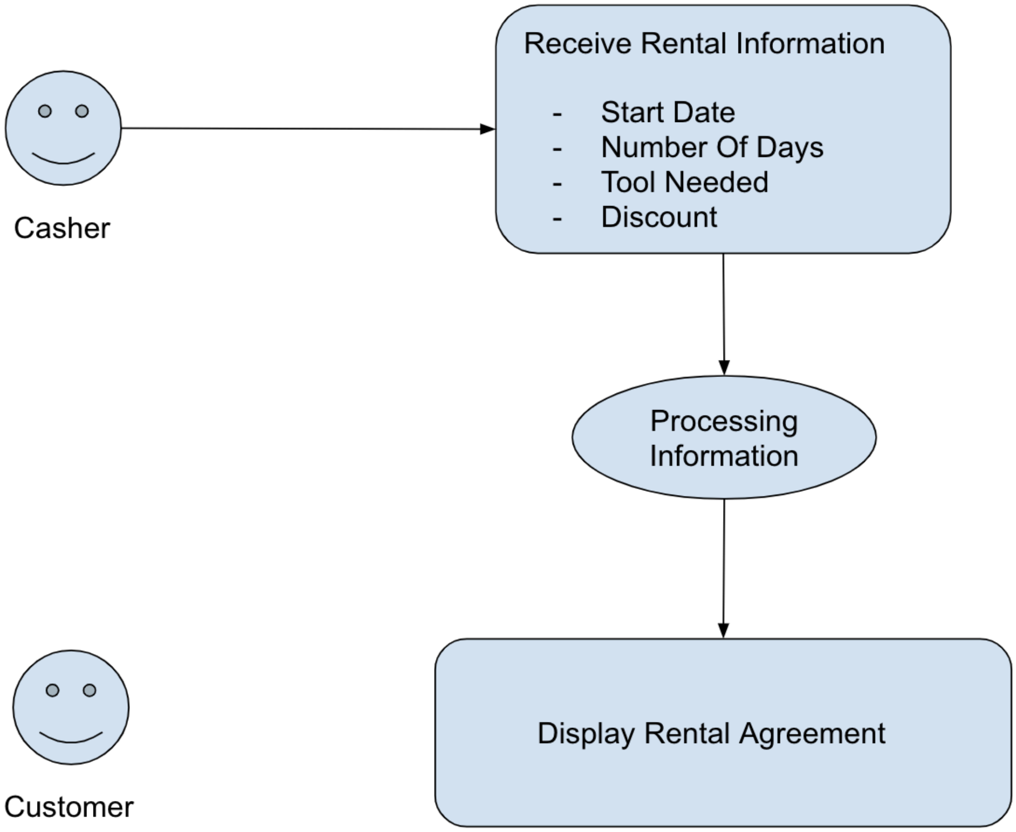
Introduction

This is a design document for a Tool Rental Point of Sale Application based on the requirement received from business.

- The application is a point-of-sale tool for a store, like Home Depot, that rents big tools.
- Customers rent a tool for a specified number of days.
- When a customer checks out a tool, a Rental Agreement is produced.
- The store charges a daily rental fee, whose amount is different for each tool type.
- Some tools are free of charge on weekends or holidays.
- Clerks may give customers a discount that is applied to the total daily charges to reduce the final charge.

Use Case Diagram

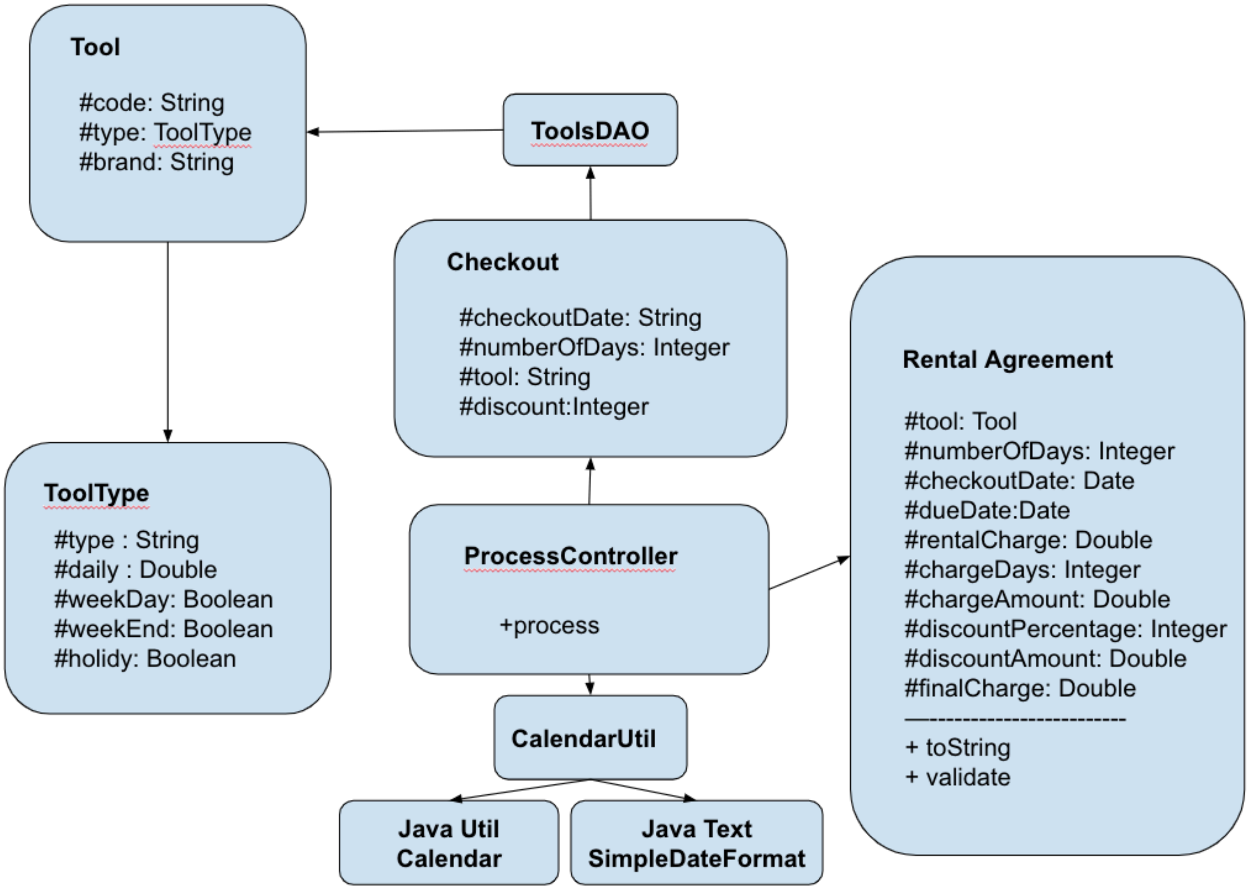
It's a simple application that works only by letting the clerk to enter the information needed , where we process the data and generate an agreement for the customer with the right amounts.



Classes and Components

This is the proposed design for this application which includes Models, Controllers, external components and Repositories to access data.

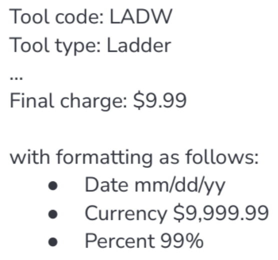
If any class / component has extra requirements , it will be listed below the diagram.



Models

- **Tool** : This is the object that has all the information about the tool , which includes the rates in ToolType object. All the information will be loaded by the ToolsDAO
- **Checkout**: This object has the information entered by the clerk
- **RentalAgreement** : This is the end result object after the data entered by the clerk has been processed and finalized.
    - All amounts rounded half up to cents.

- Printing the rental agreement will print out :



```
Tool code: LADW
Tool type: Ladder
...
Final charge: $9.99
```

with formatting as follows:

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

Controllers

- **ProcessController:** +process : This is the main class that starts the process of generating a rental agreement and accepts a Checkout Object.
    - The Checkout Object must be validated before processing
    - Rental day count is not 1 or greater
    - Discount percent is not in the range 0-100
    -

Repositories

- **ToolsDAO**: Currently this is a stub method with hardcoded data.

Helpers

- **CalendarUtil :** This is a helper class just deals with the dates and calculate the charges

External

- **util.java.Calendar**
- **SimpleDateFormat**

Data

Tools Data and Rates are coming form the ToolsDAO

Testing

An TTD is the preferred way to start your development . Your tests must cover most of the scenarios expected to produce valid rental agreements and some of the negative test cases to make sure no runtime exceptions happen.

Tests expected to test the system end to end but also for Individual methods inside the CalendarUtil

Here are **some** of the scenarios :

- Renting 0 days would not allow it to continue.
- Charge Holiday and Charge weekend
- No Charge Holiday
  - Labor Day
  - Independence Day
  - Independence Day Holiday out of range
- No Charge Weekend
- No Charge Holiday and No Charge Weekend
- Invalid Number of days
- Invalid Discount
- No Discount
- Test Calculations
  - Discount amount
  - Final charge after discount