**SG Technologies**

**Developer Manual**

**Nov. 19, 2015**

**1. Introduction**

The purpose of this manual is to explain the choices made by the programmers of this system to give insight into how the system works, as well as allow developers to make adjustments. This manual is intended for those who wish to make alterations to the source code to adjust the system to fit varying needs. It is organized in four main sections: **Introduction, System Summary, Getting Started,** and the **Developer’s Guide.**

**2. System Summary**

The system is made up of the following:

*Java files:*

Employee.java, EmployeeManagement.java, HandleReturns.java, Inventory.java, Item.java, Management.java, POR.java, POS.java, POH.java, PointOfSale.java, POSSystem.java, Rental.java, ReturnItem.java, Sale.java

*Databases:*

CouponNumber, employeeDatabase,employee log file, itemDatabase, rentalDatabase, saleinvoiceRecord, userDatabase

*System tests:*

EmployeeManagementTest.java, EmployeeTest.java, HandReturn.java, POSSystemTest.java, POSTest.java

The system was designed using the Singleton and Abstract Factory design patterns. It was programmed in Java and can be run in any compatible environment.

The system can handle and record sales or rental transactions, as well as processing item returns for refunds. It allows users to authorize payments, enter item ID via manual input or barcodes, and can also handle sales when offline. It is also equipped with administrative functions for system start-up and shut-down.

**3. Getting Started**

**3.1 System Requirements**

The system can be run by any modern computer, using a Java IDE (integrated development environment).

**3.2 Installation**

Users can install the software by downloading from Coursesite or Github. The application can be run in any Java integrated development environment software such as Eclipses and Dr. Java.

**3.3 Third-Party Components**

The system was designed without third-party components. The tax calculator was created for the system.

**4. Developer’s Guide**

**4.1 POS**

Global Variables:totalPrice (representing the price), tax (for the legality of transactions), returnSale (for verifying sales), couponNumber (database for coupons), tempFile (database for sales data)

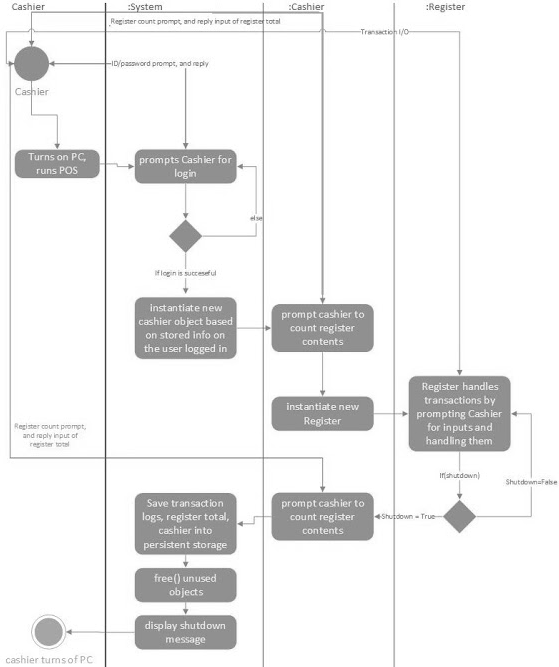
Methods: addItems (adds items to the transaction), startNew (creates a new transaction), updateTotal (keeps a running total of the cost), payment (a boolean for verifying transactions), enterItem (to enter an item in the sale), removeItem (removes items), taxCalculator (applies correct taxes to sale).

**4.1.1 New Use Case**

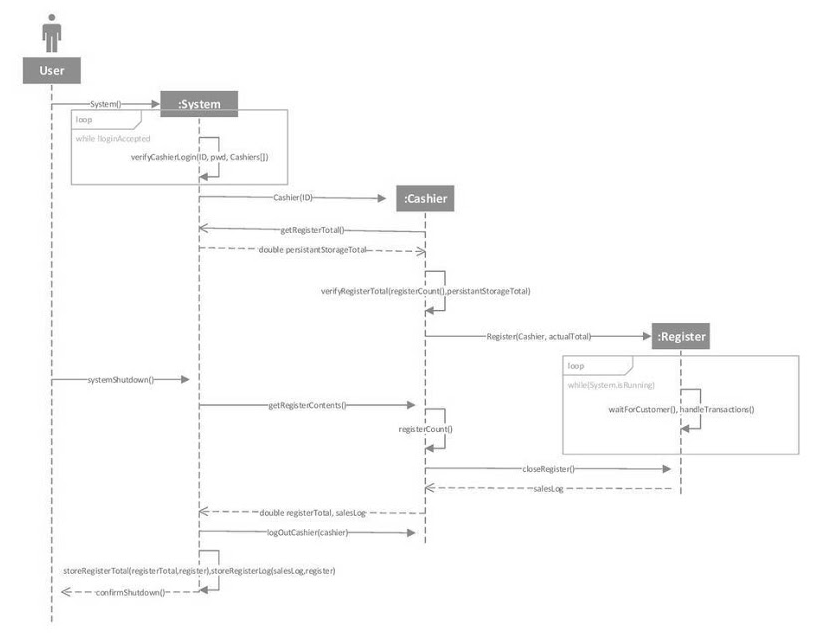
**4.2 Start Up/Shut Down**

**4.2.1 Design/Implementation of Start Up/Shut Down**

The start-up and shut-down functions of our system were designed to be as straightforward as possible and use the following sequence diagram:



**4.2.1.2 Class Diagrams**



**4.2.1.3 Data View**

**4.2.2 Customize or Extend Start Up/Shut Down**

**4.2.2.1 Components to be Modified**

There are no third party components to this section or parts that need to be modified.

**4.3 User Management**

**4.3.1 Design/Implementation**

**4.3.1.1 Architectural View**

**4.3.1.2 Class Diagrams**

**4.3.1.3 Data View**

**4.3.2 Customize or Extend User Management**

**4.3.2.1 Components to be Modified**

Adding a “senior cashier” would not be hard with the abstract factory design pattern we used.

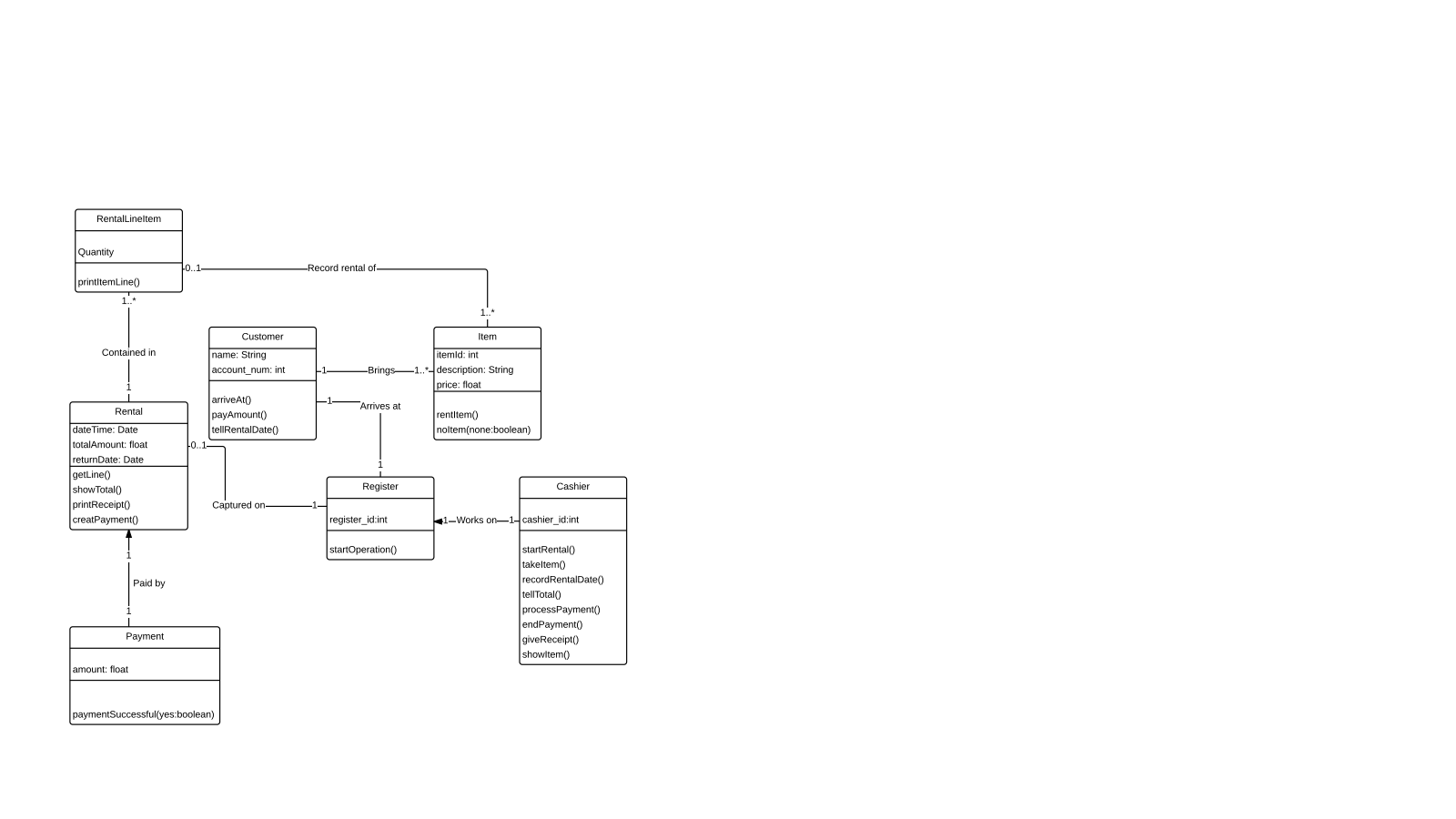
**4.4 Process Sale**

**4.4.1 Design/Implementation**

Sale was designed to be the standard for our POS system. The design was intended to be as simple as possible while incorporating all of the functionality required.

**4.4.1.1 Architectural View**

**4.4.1.2 Class Diagrams**



**4.4.1.3 Data View**

**4.4.2 Customize or Extend Process Sale**

**4.4.2.1 Components to be Modified**

There are no third party components to this section or parts that need to be modified.

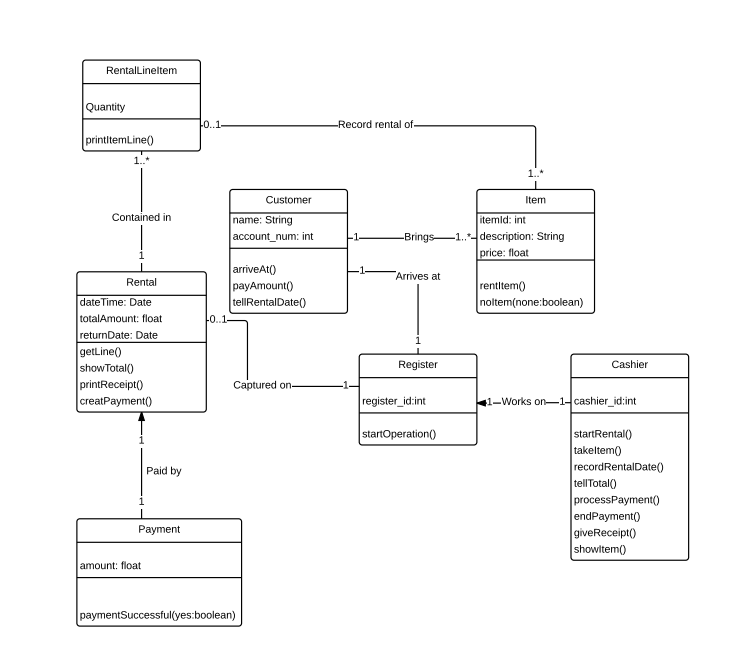
**4.5 Process Rental**

**4.5.1 Design/Implementation**

Rental was designed to be treated the same way as Sale, except for the return of the item at the end. Additionally, rental requires several different features not necessary in Sale, such as a date for returns.

**4.5.1.1 Architectural View**

**4.5.1.2 Class Diagrams**



**4.5.1.3 Data View**

**4.5.2 Customize or Extend Process Rental**

**4.5.2.1 Components to be Modified**

There are no third party components to this section or parts that need to be modified.

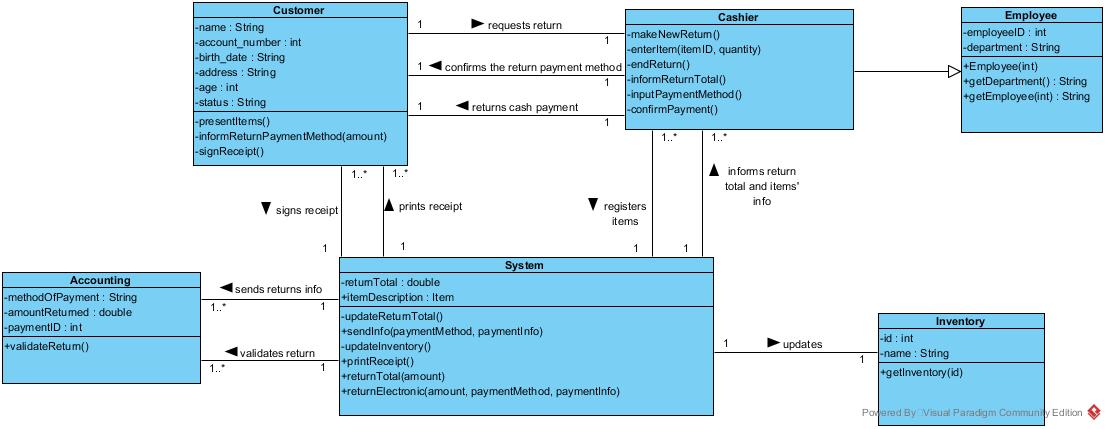
**4.5.2.2 Changing Rental Rules**

**4.6 Process Return**

**4.6.1 Design/Implementation**

**4.6.1.1 Architectural View**

**4.6.1.2 Class Diagrams**



**4.6.1.3 Data View**

**4.6.2 Customize or Extend Process Return**

**4.6.2.1 Components to be Modified**

There are no third party components to this section or parts that need to be modified.