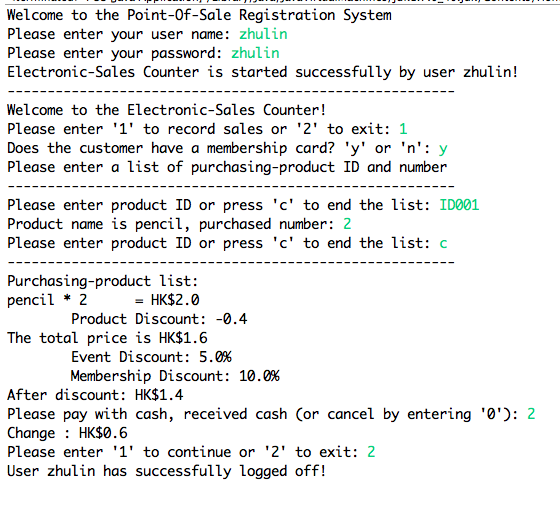
# POS JUnit Testing Report

## Application Overview

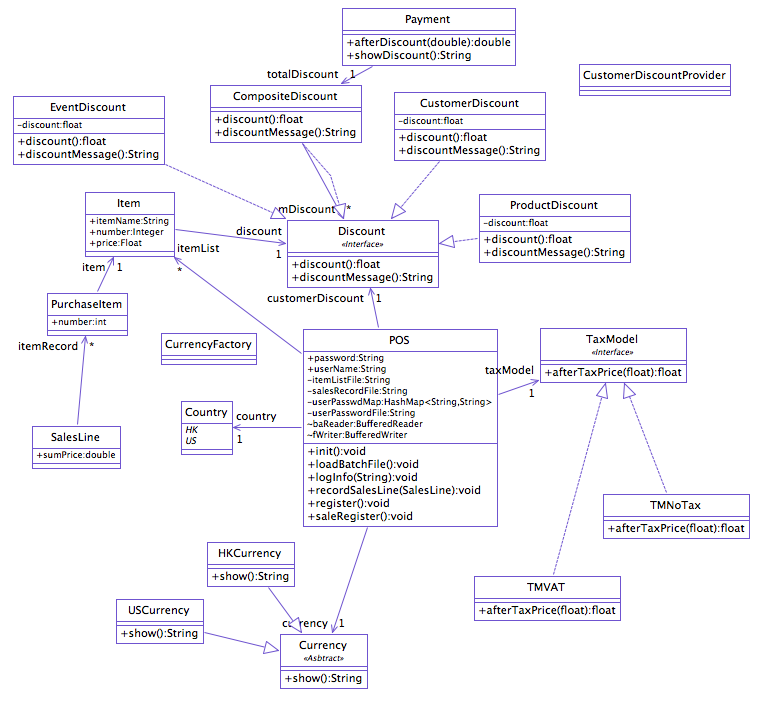
Point of Sale(POS) is an open-source project whose basic functionality is to help customers and to make payments to merchants in exchange of goods or services. After a customer has bought something, he should login to the system and make payments for what he has bought, at the same time the systems would perform some kinds of discount based on the information about the customer and record the purchase information. There are several basic parts about in this system:

* Customer log in/ log out, using username and password
* Record goods bought and compute the payment based on number and discount information
* Log the purchase information into log file



The structure of POS system is as follows, there is only one single package of the project called POS:

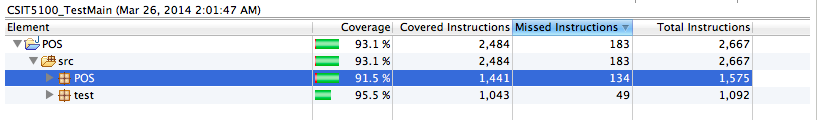
* POS.java contains the POS class, which includes the execute main function as the entry of the program. POS class is implemented as singleton class to ensure there is always a single POS instance in the execution.
* TaxModel.java is used to provide interface to define the model of tax.
* Currency.java contains the interface for showing the different kinds of currency the system would use.
* Discount.java provide the interface for defining different kinds of discount including event-based discount and user-based discount and item-based discount.
* CompositeDiscount.java contains the class which help to combine several kinds discount together the compute the final discount for those discounts.



## Testing Coverage

The statement coverage of this unit testing is 91.5%.

Actually it is not very difficult for us to achieve testing coverage which is higher than 90%. Instead of writing test code in POS package, I create another package which is used to test the functions in POS. The basic idea is that to test POS system I test every single methods in POS package as many as possible. An effect write another package besides POS to do the unite test is that I can only invoke the public methods in POS and reference the public class in POS. Those private classes and methods is beyond the scope for test package can cover. But using reflection we can test private members. I would cover this later.



## How to achiever high coverage

## Infeasible Program Statements

## Program Failures