**VendingMachine Design Documentation**

**Document Author(s): Louis Warner**

**Date: 1/27/2016**

**Design Rationale**

The VendingMachine project uses a total of seven classes and one interface to implement the program. The figure on the following page shows the interface and classes and how they relate to each other in a UML class diagram.

The VendingMachine class is the main class for the project. It has a set of four possible states represented by the classes: NoMoneyInserted, MoneyInserted, StockMode, and OutOfOrder. The VendingMachine keeps a count of how much money is currently inside the machine (starting with a default value of $100.00. It also keeps a count of how much money the user has inserted into the machine. The VendingMachine has an array of Item objects that are loaded in by processing the ItemList.txt file. The VendingMachine also has a default password of ABC123 that is used to enter stock mode.

Each Item object has an item code, an item name, an item price, and an item quantity. As the user interacts with the machine, the quantity of the item within the VendingMachine will decrease (when item is purchased) or increase (when items are restocked). Initial and restocked item codes, names, prices, and quantities can be edited in the ItemList.txt file, although the code maintains the option to have these values edited by a user in the future with additional code. For now, the items in the machine will be determined by the ItemList.txt file.

The VMState interface contains all methods by which the user can currently interact with the VendingMachine. The user may view an item list or exit the program no matter which state the machine is in. All other methods vary a bit based on the current state of the VendingMachine.

The machine begins in the NoMoneyInserted state. In this state, the user may:

* Insert cash (setting the state to MoneyInserted)
* Attempt to change to stock mode (by entering the correct password)

In the MoneyInserted state the user may:

* Insert additional cash
* Return cash (setting the state to NoMoneyInserted)
* Attempt to purchase a product

In the StockMode state the user cannot insert cash or purchase products, but the user may:

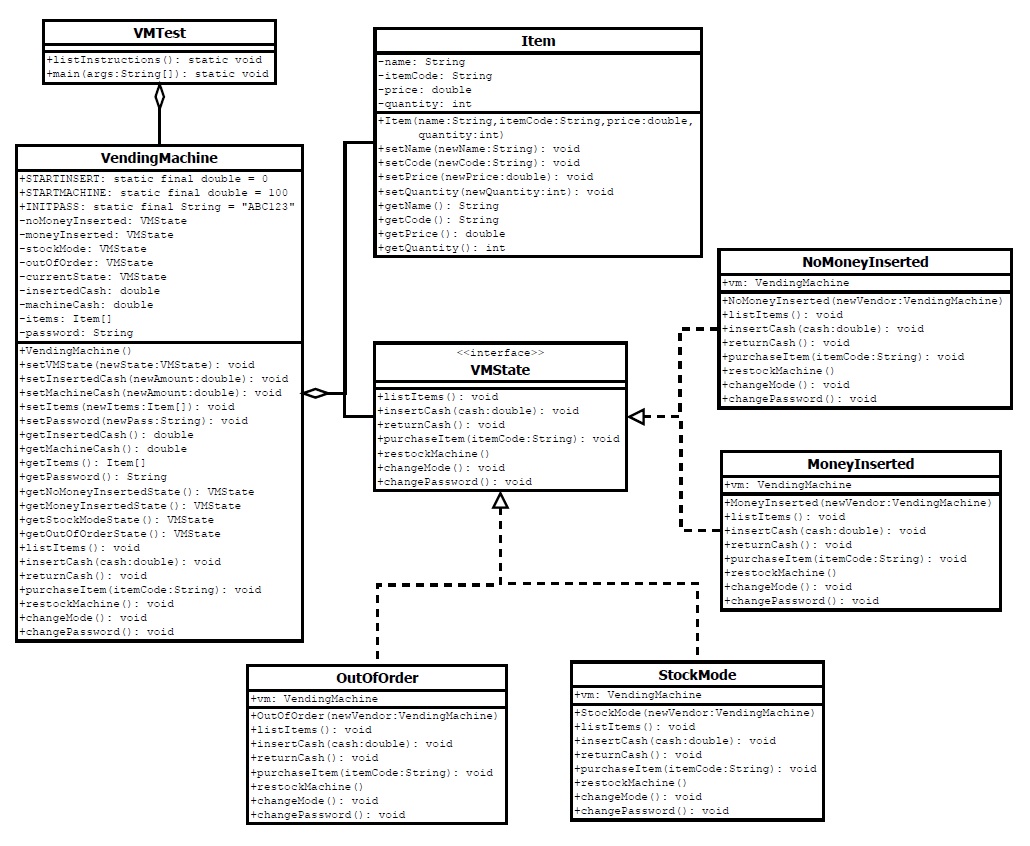
* Return cash (returning the profits from the machine to the user)
* Restock the machine to the default values from ItemList.txt
* Change the VendingMachine’s password
* Return to user mode

In the OutOfOrder state the only option the user has is to enter stock mode and restock the items. This is the only way the machine can escape the OutOfOrder state.

VMTest is the command line test program with the main method for starting the VendingMachine project. It will continually list the user’s options and loop until the program has exited. Once all files have been compiled it can be run using the command:

Java VMTest

Note: <UML Diagram in Figure on following page>



**Figure: Class Diagram for VendingMachine**

**Project Timeline**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Change Description** |
| **1/27/2016**  **1/28/2016**  **1/30/2016**  **1/31/2016** | Louis Warner  Louis Warner  Louis Warner  Louis Warner | * Began Writing Design & Created UML Markup * Created Context & Proposed States * Implemented Java Code & Began Testing * Finalized Documentation & Submitted Assessment |