# CashMan3001

## Prerequisites

CashMan3001 requires Java 1.8 and Maven 3.5 are installed and configured.

The JAVA\_HOME variable must be correctly set to the Java 1.8 installation and both the Java and Maven Bin directories must be in the PATH environmental variables.

The Github repository and this document can be found in

https://github.com/lukejdavis/CashMan3001

## Building, Executing and Testing

While CashMan3001 was developed in Eclipse only Maven is required to buid, execute and test CashMan3001.

Open a command window and CD to the CashMan3001 directory

Run the following commands to build. Tests are skipped at the moment as only the rest interface tests were implemented and will be tested separately.  
mvn compile  
mvn install -Dmaven.test.skip=true

To execute CashMan3001 run the following command  
mvn spring-boot:run

The application can be manually tested using the test page below. CashMan3001 is presently only set up to work on the local host.  
<http://localhost:8080/>

To run the automated test scripts the following command should be executed in a separate command window while CashMan3001 is actively running. CD to the CashMan3001 directory first.  
mvn test

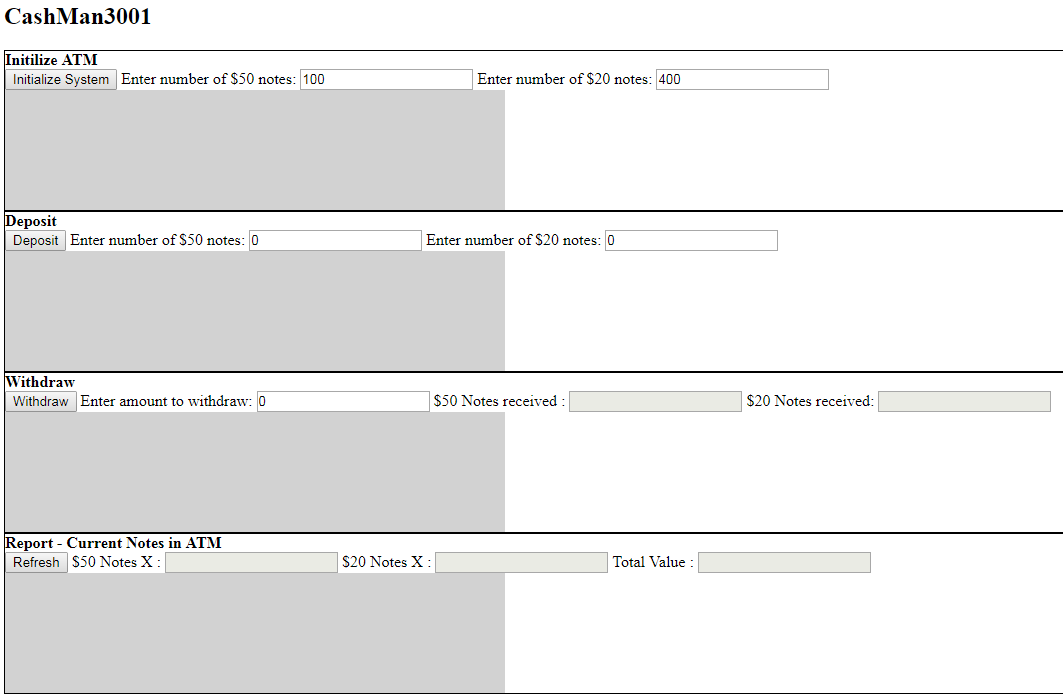
## Using the Test HTML Page

From the image below the Page is broken up into four sections. Initialize ATM allows the initialization of the ATM where the initial notes can be set. This can only be set once, a message will be returned if attempted again. The other sections cannot be used and will return an error if used prior to initialization.

Deposits allow the addition of new notes into the ATM. Simply enter the number of Fifty and Twenty dollar notes to add and hit the deposit button.

Withdraw allows the withdrawal of money from the ATM. It accepts only the amount to be withdrawn and returns the combination of notes received. An error will be displayed if more notes then are currently available are attempted to be withdrawn.

The report section is continuously updated after each transaction and displays the number of notes in the ATM and the current dollar value. It can be refreshed upon hitting the Refresh button.



## Libraries Used

The only external libraries used were spring-boot-starter-web, spring-boot-starter-test and spring-web which are the Spring Boot libraries and Spring Libraries. No other libraries were imported into the project. Some Java 1.8 libraries were used such as the JSON and JUnit libraries but these come installed with Java 1.8

## Notes

This project took approximately 5 hours to produce all up. I was able to complete all of the core requirements and some of the optional set. The rest interfaces will support all legal denominations of notes. A lot of options were chosen for simplicity, such as in memory management of the ATM rather than using hibernate as configuration time would have significantly added to the development time.