Excite ATM

(Readme)

Table of Contents

Overview

This document provides required information about the ExciteATM application to build and run the application.

Requirements

As given in the PDF document. Refer the requirements PDF document.

Pre-requisites

ExciteATM is a standalone java application (jar), built using the following :

- **Java**
  java version "1.7.0_45"
  Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
  Java HotSpot(TM) 64-Bit Server VM (build 24.45-b08, mixed mode)

- **Eclipse**
  Eclipse Java EE IDE for Web Developers.
  Version: Luna Release (4.4.0)
  Build id: 20140612-0600

- **Maven**

  Apache Maven 3.2.2 (45f7c06d68e745d05611f7fd14efb6594181933e; 2014-06-17T23:51:42+10:00)
  Maven home: C:\work2\apache-maven-3.2.2
  Java version: 1.7.0_25, vendor: Oracle Corporation
  Java home: C:\Program Files\Java\jdk1.7.0_25\jre
  Default locale: en_IN, platform encoding: Cp1252
  OS name: "windows 8", version: "6.2", arch: "amd64", family: "windows"

- **git**
  git version 1.8.4.msysgit.0

  Application source URL : https://github.com/dvgj/atmexcite.git

- **junit**
  junit 3.8.1

Build/Design/Development Considerations

- Packages categorized into Contracts, Implementation, Utilities, Configurations.
- Maven for packaging.
- JUnit for testing.
- Contract definition using interfaces.
- Pluggable design, implementations plugged in through configuration.
- Factory pattern for locating and instantiating implementations.
- Singleton to create one instance of the ATM implementation.
- Simple AOP using Java Proxy, for auditing time taken by methods.
- XML persistence using JAXB to maintain the state.
- Quick command line for user input.
- Adequate documentation using Java Docs.

Application Demo

Packaging

Run the mvn package to build, run the unit test cases, and build the executable jar.

```
>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building excite-atm 1.0.0
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ excite-atm ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\James David\Documents\GitHub\atmexcite\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ excite-atm ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ excite-atm ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\James David\Documents\GitHub\atmexcite\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ excite-atm ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ excite-atm ---
[INFO] Surefire report directory: C:\Users\James David\Documents\GitHub\atmexcite\target\surefire-reports
```

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.excite.atm.impl.ATMImplCoreTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.396 sec
Running com.excite.atm.impl.ATMImplErrorsTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 sec

Results :

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ excite-atm ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.597 s
[INFO] Finished at: 2014-07-20T17:12:33+10:00
[INFO] Final Memory: 8M/245M
[INFO] ------------------------------------------------------------------------
>
```

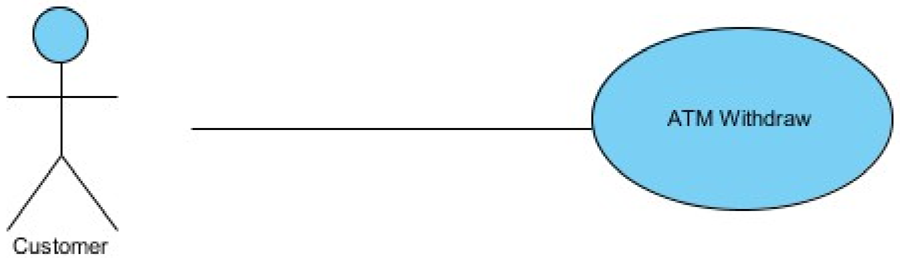
Running the application

Shell script run.sh will run the appropriate class from the jar 'java -cp target/excite-atm-1.0.0.jar com.excite.atm.demo.ATMDemo'. Opens a command line tool accepting the amount that needs to be withdrawn, and shows the balance notes for each denomination.

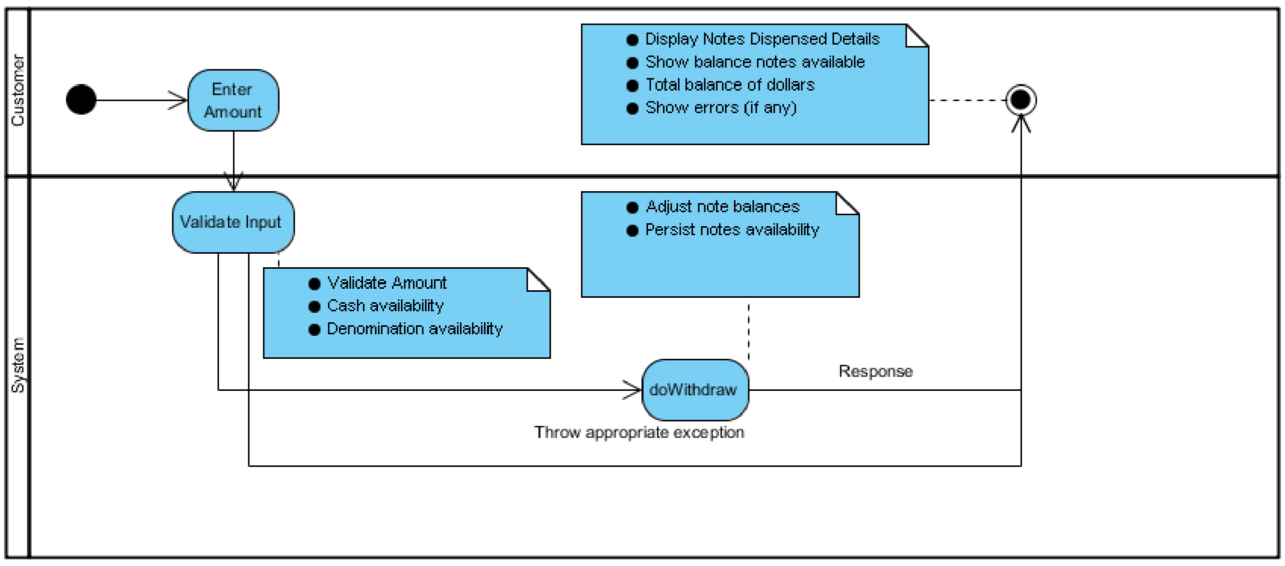
```
>run
Available commands :
amount -> just enter the amount to withdraw and show balance
quit -> to exit the application
$20 -> 100 notes available
$50 -> 100 notes available
Balance : 7000
>20
For $20 [1 notes of 20] [0 notes of 50] cash dispensed
$20 -> 99 notes available
$50 -> 100 notes available
Balance : 6980
>50
For $50 [0 notes of 20] [1 notes of 50] cash dispensed
$20 -> 99 notes available
$50 -> 99 notes available
Balance : 6930
>70
For $70 [1 notes of 20] [1 notes of 50] cash dispensed
$20 -> 98 notes available
$50 -> 98 notes available
Balance : 6860
>10
ERROR : Unable to dispense cash for specified amount, incorrect denominations, must be in 20s and/or 50s
>7000
ERROR : Insufficient cash at ATM
>133
ERROR : Denominations not available or incorrect denominations (must be in 20s and/or 50s)
>6860
For $6860 [98 notes of 20] [98 notes of 50] cash dispensed
$20 -> 0 notes available
$50 -> 0 notes available
Balance : 0
>
```

Application Design

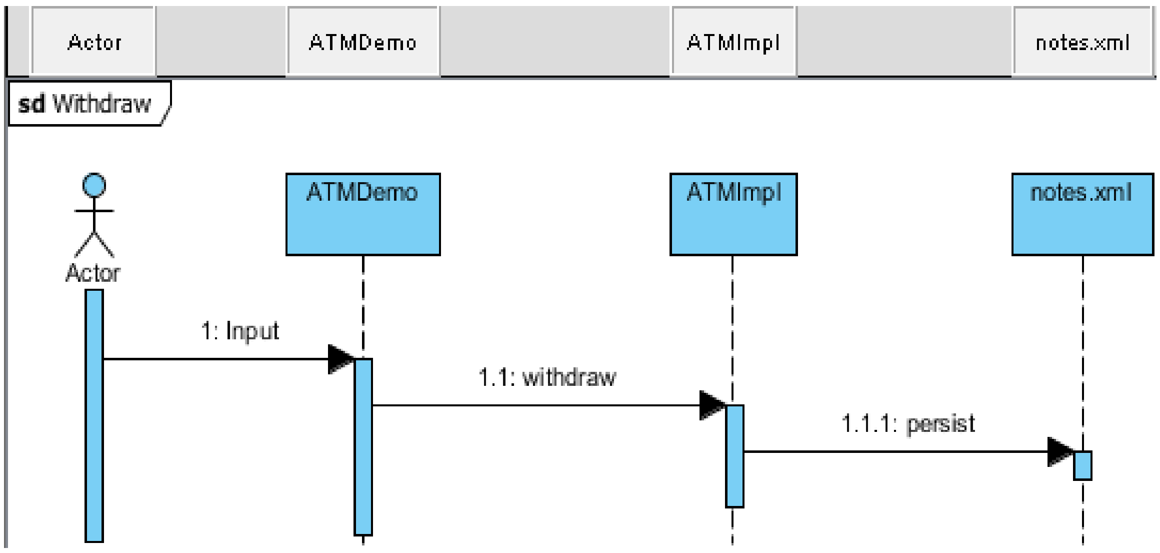
Use case – Withdraw



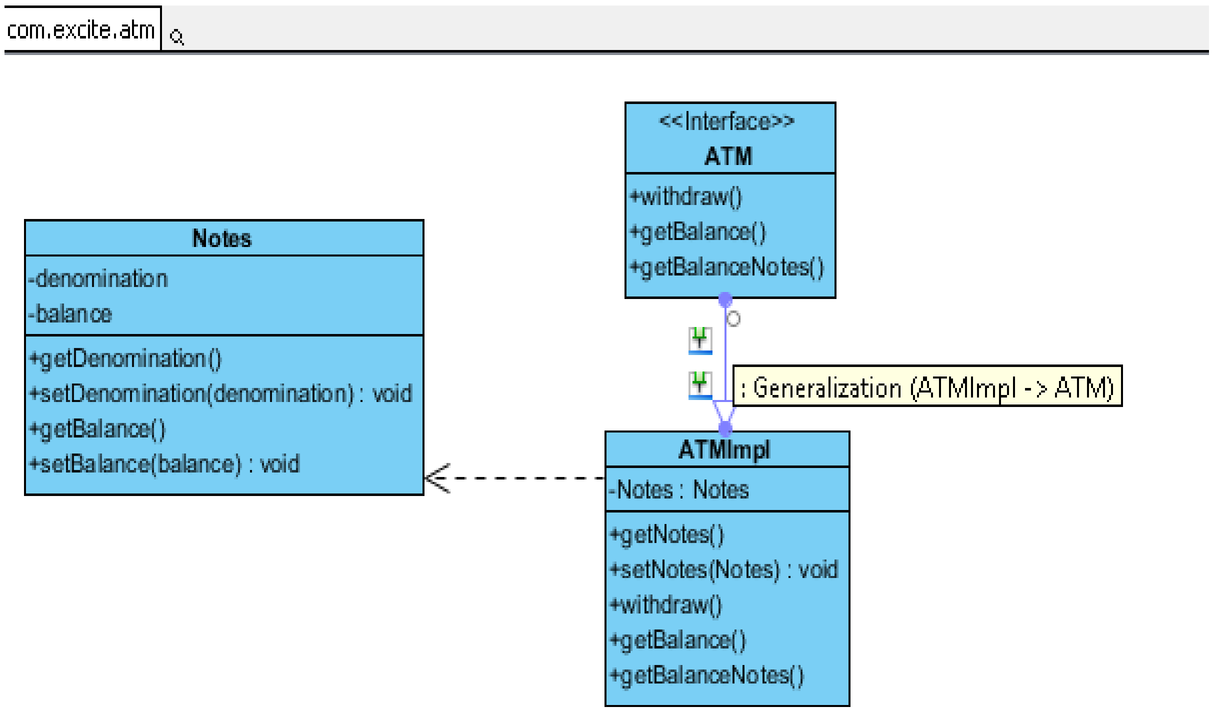
Activity Flow



Sequence



Classes and Interfaces



Files and Configurations

run.sh / run.bat

In Linux/Unix OS, execute run.sh to start the command line tool. During initialization, the application displays the available balances of each note. In the prompt, just enter an amount (to withdraw), and the notes are dispensed accordingly, and balance is shown.

javadoc.xml

Ant script for generating the javadoc.

notes.xml

Primary file the application is configured to use to initializing the system with available number of notes for configured denominations.

appl_config.properties

This is to configure various application properties viz.

- Implementation Properties file (Interface vs. Implementation).
- XML file to load the balances of each denomination.
- Configure the various denomination the application supports.

impl_config.properties

This property file is used to configure the implementation that an application must use for the interfaces provided as contract.

log.properties

Configuration details for application logging.

messages.properties

Resource bundle file for supporting multilingual messages for various error messages displayed by the system.

Application Package

More information about each class and methods are available in javadocs.

com.excite.atm.core

Contains the application core interfaces and classes viz.

- ATM Interface (that any implementation should abide by).
- Notes, a POJO defining the denomination and balance attributes.
- AppContext, expose the application properties via thread context.
- InvocationHandler, a simple AOP to intercept method invocations and log the time taken for each method.

com.excite.atm.demo

Contains the Demo class, to run the application.

- Initializes the implementation configured against the ATM interface
- Loads the denominations and balances from the configured XML
- Reads input from the user from command line and executes appropriate methods on the ATM application, displays back the appropriate responses to the user.

com.excite.atm.exceptions

Contains a generic ATMException to report 3 different errors. Messages are loaded from ResourceBundle.

com.excite.atm.impl

Core implementation package, providing implementation for ATM, AppContext interface and Constants used by the application

com.excite.atm.util

Contains the factory class for instantiating appropriate implementations, utility classes for reading and writing to XMLs using JAXB.