**BAK-APP User Manual**

**Version - 1.0**

Version Index

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Comment** | **Executed on** |
| 1.0 | Bhaskaran S | Initial Document | 14-10-2019 |
| 1.1 | Bhaskaran S | Updated Version | 19-10-2019 |

Index

1. Description
2. Dependencies
3. GUI
   1. Finding the Packing Information
   2. Adding the Inventory to dataSoure
   3. View the Inventory.
4. Logs
5. Executing Application through Eclipse and CLI mode
6. Sample Snapshots
7. Abbreviations

Snapshot

1. *Order Items window*
2. *Add Items window*
3. *View Items window*
4. *Execute Application via Command Line Argument Interface*
5. *Sample Execution on CLI*
6. *Sample Execution on CLI with invalid inventory.*
7. *Order Calculation*
8. *Add Items with sample input data*
9. *Added Items is written to datasource properties file*
10. *view inventory from datasource properties file*
11. Description

The BAK-APP application is the GUI and Command Line based small-scale application, which helps the user to find out the packaging information for the given input.

The purpose of this application is to make sure the given input is matched with the minimum number of the packs and to calculate the amount involved for it.

1. Dependencies

This application is based of Java and it requires the JRE environment and junit for test.

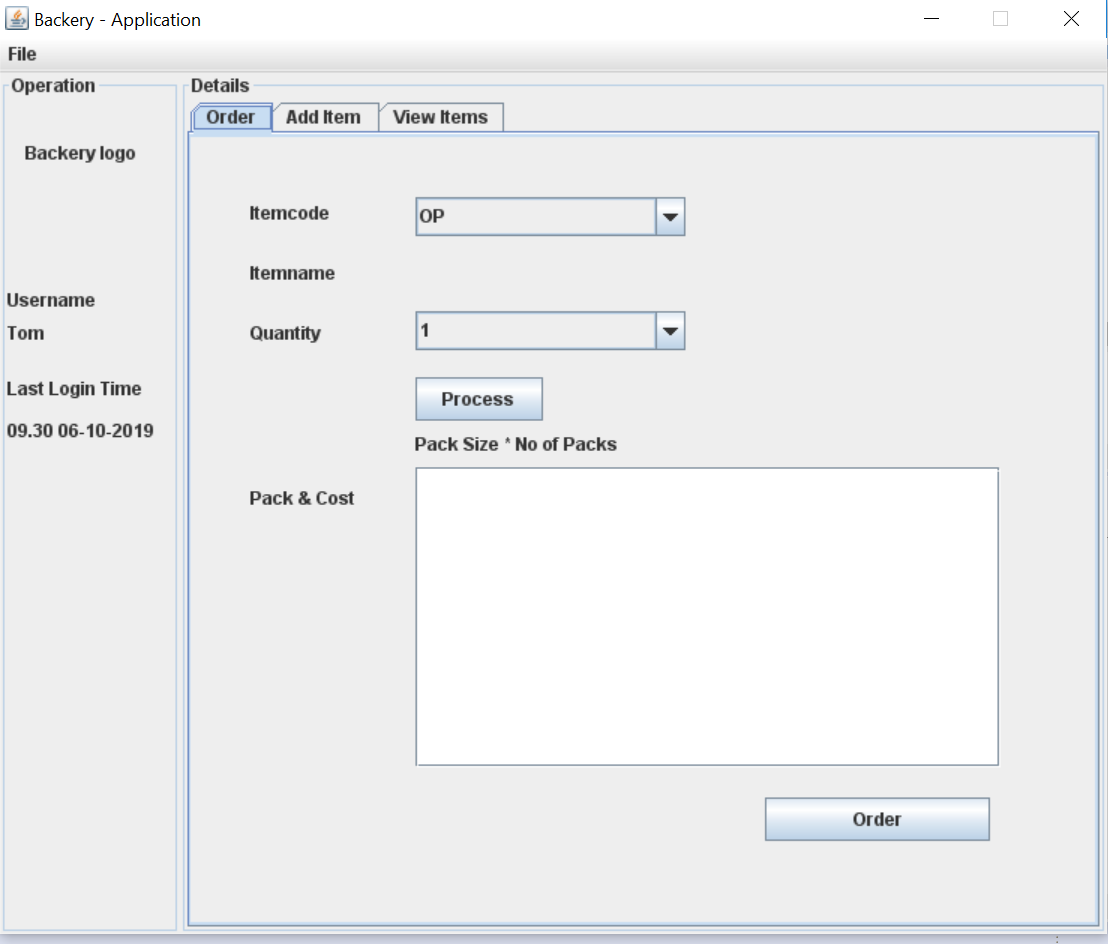
JDK/JRE 1.8 Version

Junit 4.11

Log4J 1.2.17

1. GUI

When this application is launched via the script, a GUI opens and show as like the following

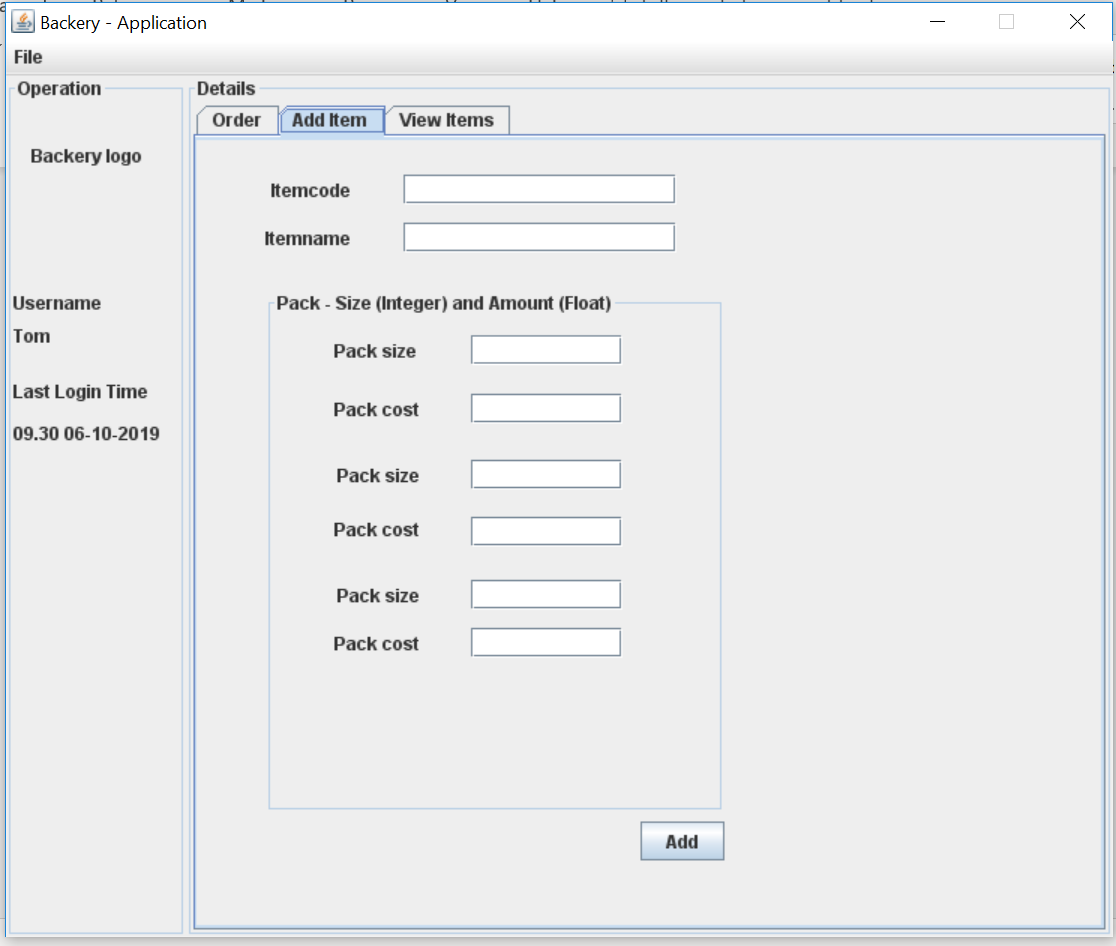


*Figure: 1 – Order Items window*

* 1. – Finding the Packaging information

1. Select the ItemCode from the dropdown.
2. Corresponding ItemName gets loaded in the below component.
3. Select the Quantity as per user requirement.
4. Click on the Process button.
5. The expected output is shown in the textArea.
   1. - Adding the items to the database
6. On the main tab, user can navigate to Add Items pane.
7. User can add Itemcode, Itemname and the pack information.
8. For now, adding 2 items is enabled.
9. GUI can validate the input given.
10. When a click on the Add , it writes the content to the database(datasource.properties) file.

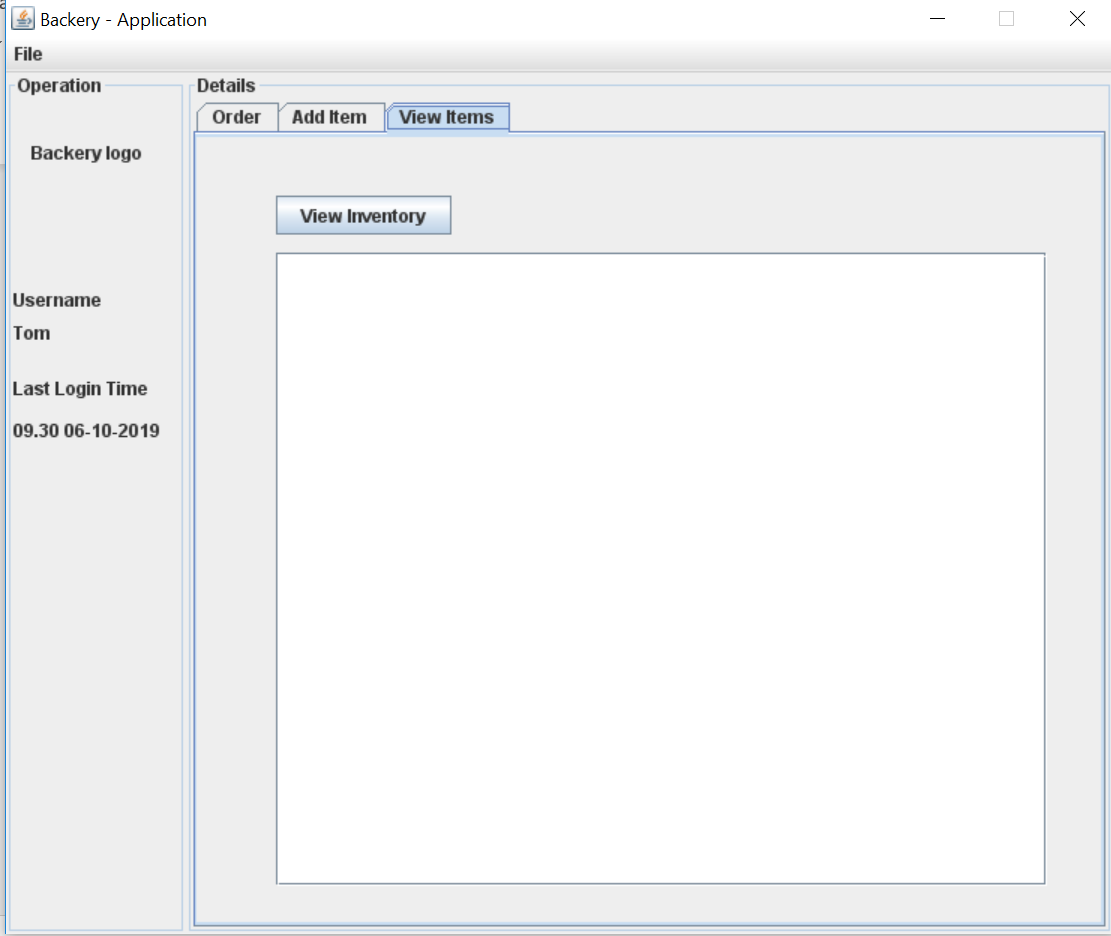
This is how the Add Item panel look like

*Figure: 2 – Add Items window*

* 1. – View the inventory

1. User can look at the existing items available in the inventory.

This is how the inventory panel look like



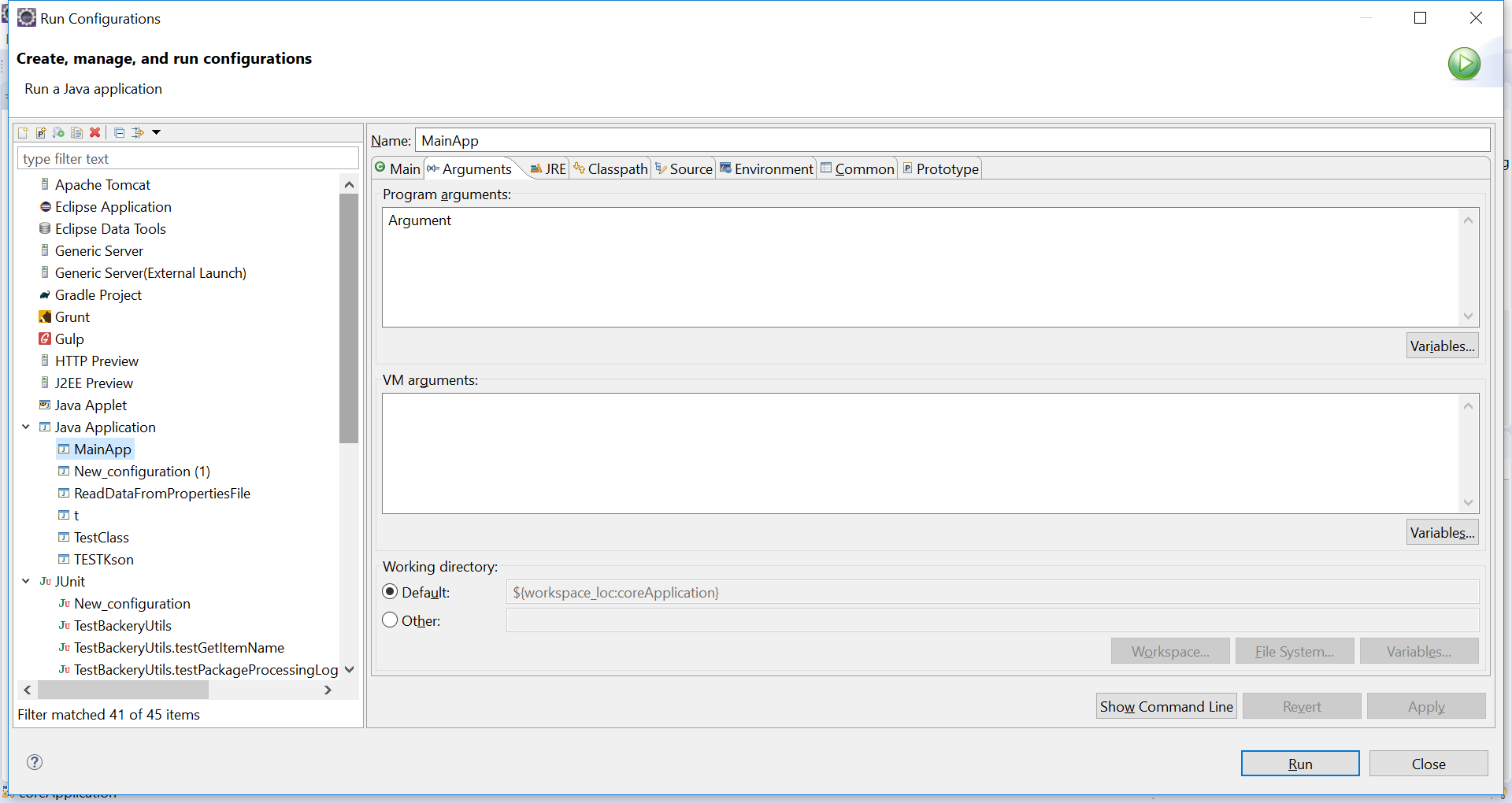
*Figure: 3 – View Items window*

1. Logs
2. This application writes down the logs on what is happening right from application launch to exit.
3. User can customize the logs which he/she wants, it can be any of **INFO**, **DEBUG**, **ERROR** and **WARN**.
4. The logs gets generated in the C:\ BakeryApplicationlogs\ Directory

1. Executing Application through Eclipse and CLI Mode
   1. – This application also supports the CLI Mode.

Steps to execute.

* 1. Execute the application with Command Line Arguments more than 0. (can be executed via Eclipse or by Command Line)
     1. Executing by Eclipse
     2. Right click on the Project from the Project Explorer window.
     3. Click Run As -> Run Configuration.
     4. Click Java Application on the left,
     5. Right Click.
     6. Select New Configuration
     7. Select the Project and the Main Class.
     8. On the Arguments Tab, Add any string or integer on the Program Arguments text area.



*Figure: 4 – Execute Application via Command Line Argument Interface*

* + 1. Click Apply and Run/Close.
  1. – Executing via Command Line Interface.

2.1.1 The following commands help user to execute this application via command line mode.

* + 1. **Note** : Java JRE/JDK should be installed first.
    2. To check whether Java is Installed or not, execute “java -version” in the CLI ( Terminal window ), if it responds to the output with version then it is available.

JAVA\_HOME=/usr/lib/jvm/jre1.8.0\_201

BAK-APP\_HOME=/opt/bak-app/

PROGRAM="org.backery.mainapp.MainApp"

BAK-APP\_EXECUTABLE="$JAVA\_HOME/bin/java -cp $BAK-APP\_HOME/lib/\*:. $PROGRAM &"

**Command In Detail.**

JAVA\_HOME=/usr/lib/jvm/jre1.8.0\_201

This command let system/application know where the JDK/JRE installed on the target machine.

BAK-APP\_HOME=/opt/bak-app/

This command let system/application know where bak-app home directory is.

PROGRAM="org.backery.mainapp.MainApp"

This command targets/points to the Backery App Main Java File which gets triggered when the application is started.

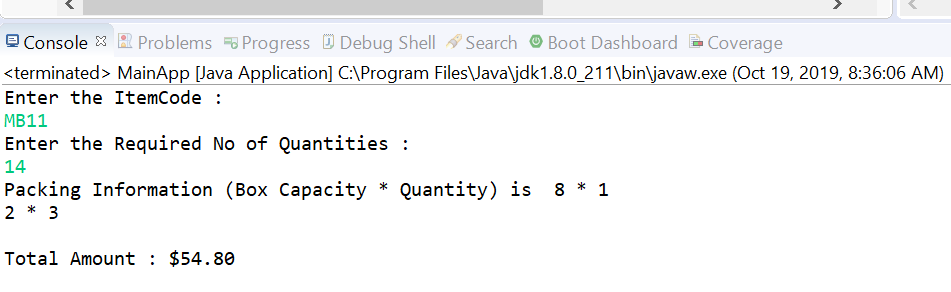
BAK-APP\_EXECUTABLE="$JAVA\_HOME/bin/java -cp $BAK-APP\_HOME/lib/\*:. $PROGRAM &"

This command takes the input from the above three system variables.

Basically from Java installed directory, execute the application which has the reference to the $PROGRAM with the classpath of lib directory (which contains the jars required to run ).

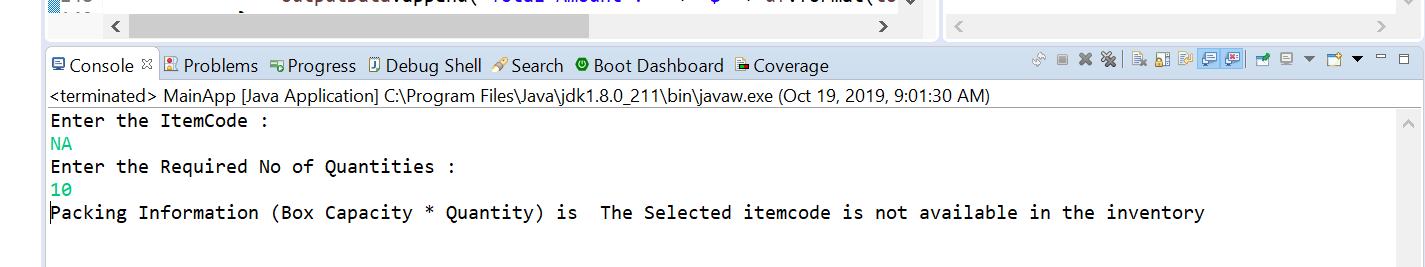
An Example of how the Command Line Execution look like

* 1. Enter the ItemCode
  2. Enter the Required Quantities.
  3. The output will be shown immediately.



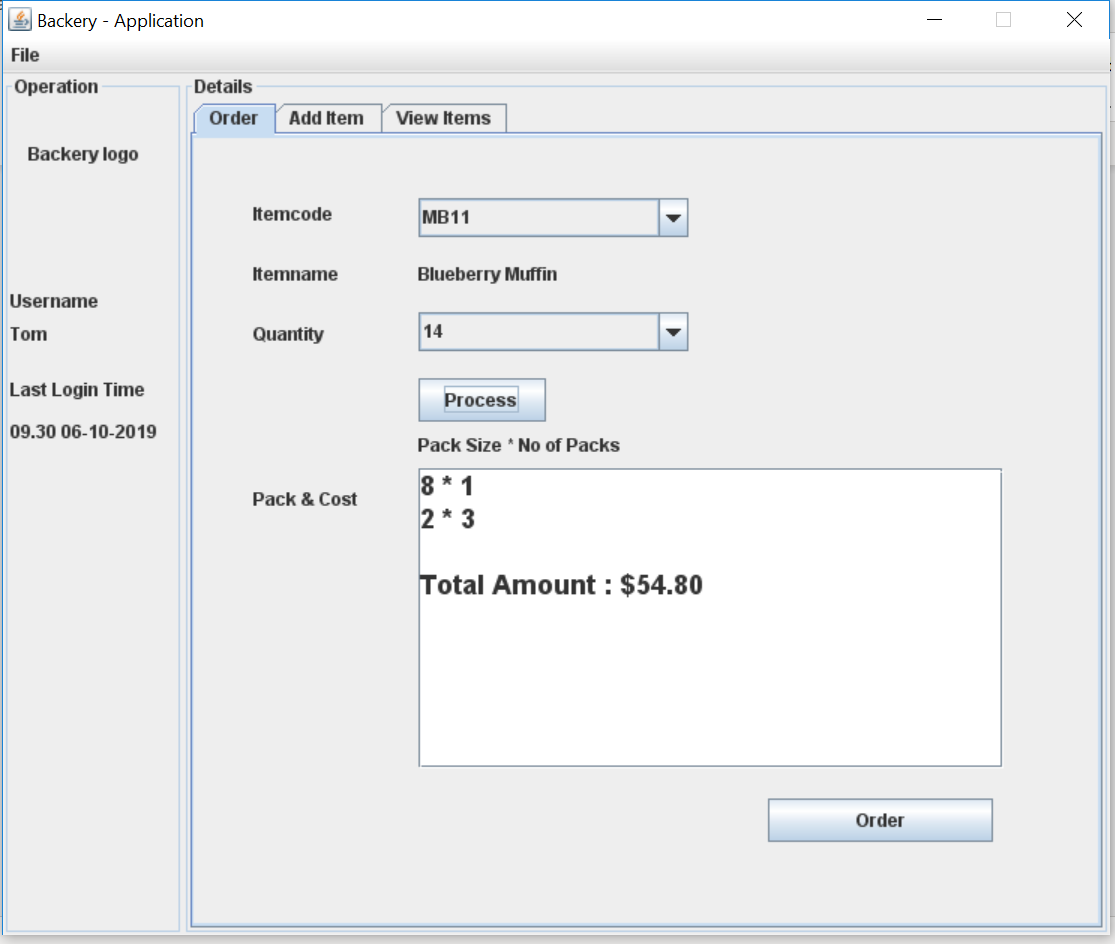
*Figure: 5 – Sample Execution on CLI*

* 1. When the given input does not available in the application, application will let user know the status by the message “The selected itemcode is not available in the inventory”

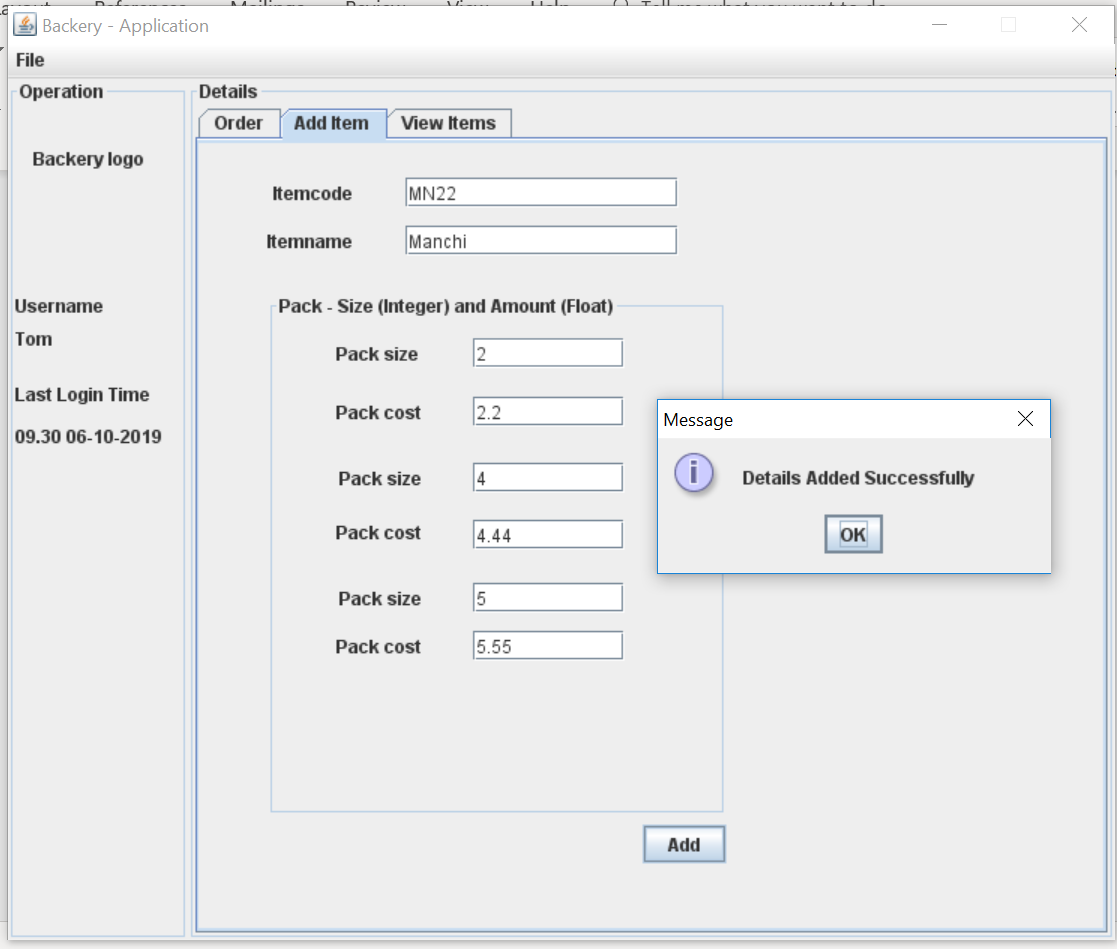


*Figure: 6 – Sample Execution on CLI with invalid inventory*

1. Sample Snapshot



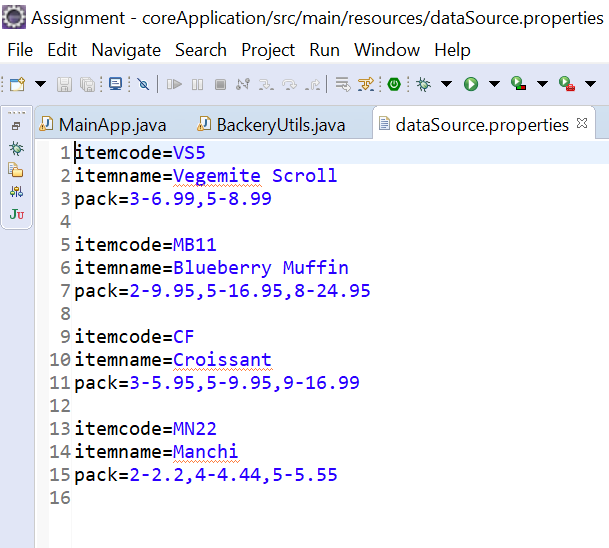
*Figure: 7 – Order Calculation.*



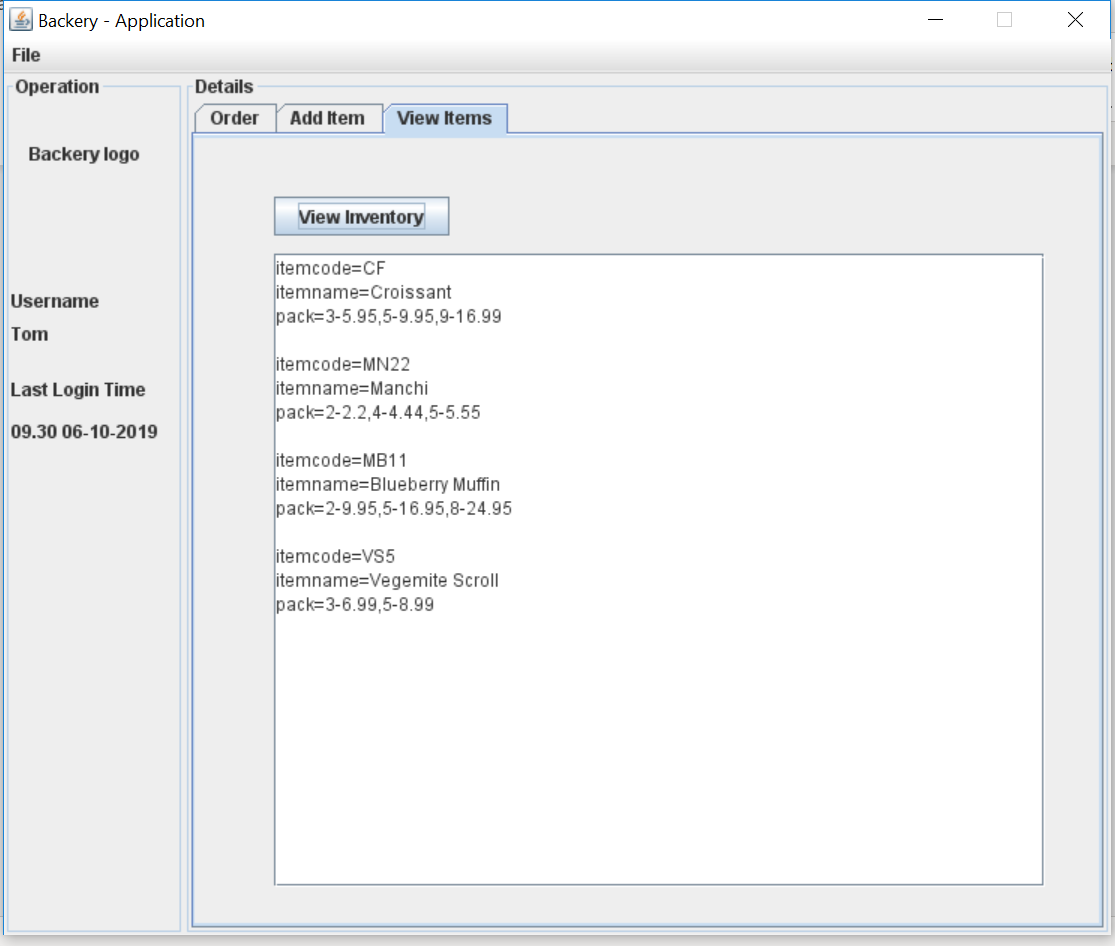
*Figure: 8 – Add Items with sample input data*

*Note : User can add minimum of two Pack Size and Pack Cost. Single entry for Pack Size and Pack Cost is not allowable.*

*Third entry is optional.*

*.*

*Figure: 9 – Added Items is written to datasource properties file.*



*Figure: 10 – view inventory from datasource properties file.*

1. Abbreviations

|  |  |
| --- | --- |
| JDK | Java Development Kit |
| JRE | Java Runtime Environment |