Impaq's Point of Sale Manual 1.0

Agustin Cabra – February 2015

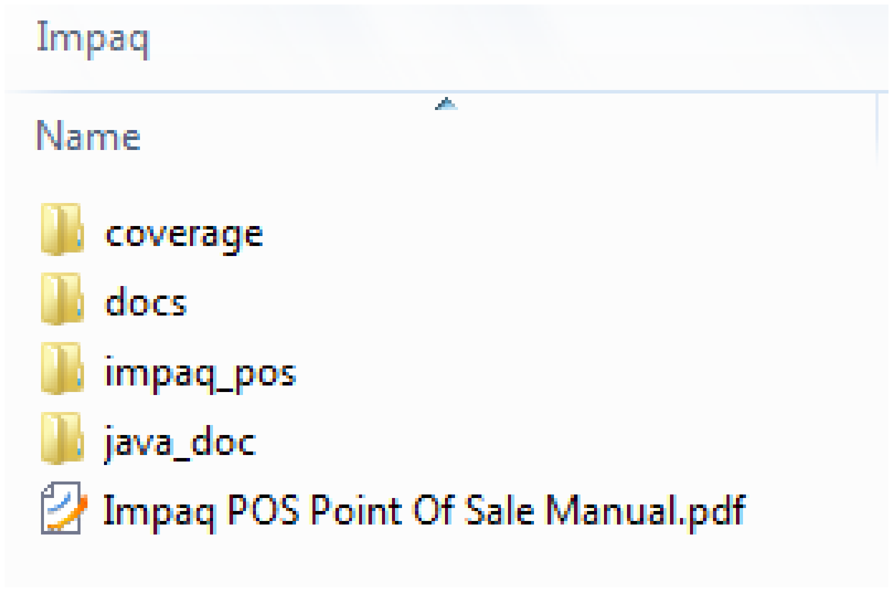
Contents

1. Folder Contents 3
2. Installation on eclipse 4
3. Final User View 8
4. The ‘data’ configuration folder 13
   a. The devices.txt file 13
   b. The products.txt file 14
   c. The config.properties file 15
   d. Setting a tax value 16
5. Technical details 17
   a. Javadoc 17
   b. Code coverage 18
   c. UML Class Diagram 19
   d. Unit Testing 20
6. Conclusions 21

1. Folder Contents

The zip compressed folder contains the following items:

- coverage folder containing the export of the last test run to the project
- docs contains source of Class Diagram and a PDF file exported
- impaq_pos contains the JavaProject information
- java_doc references the documentation generated by javadoc after using the annotations through the code
- This manual.

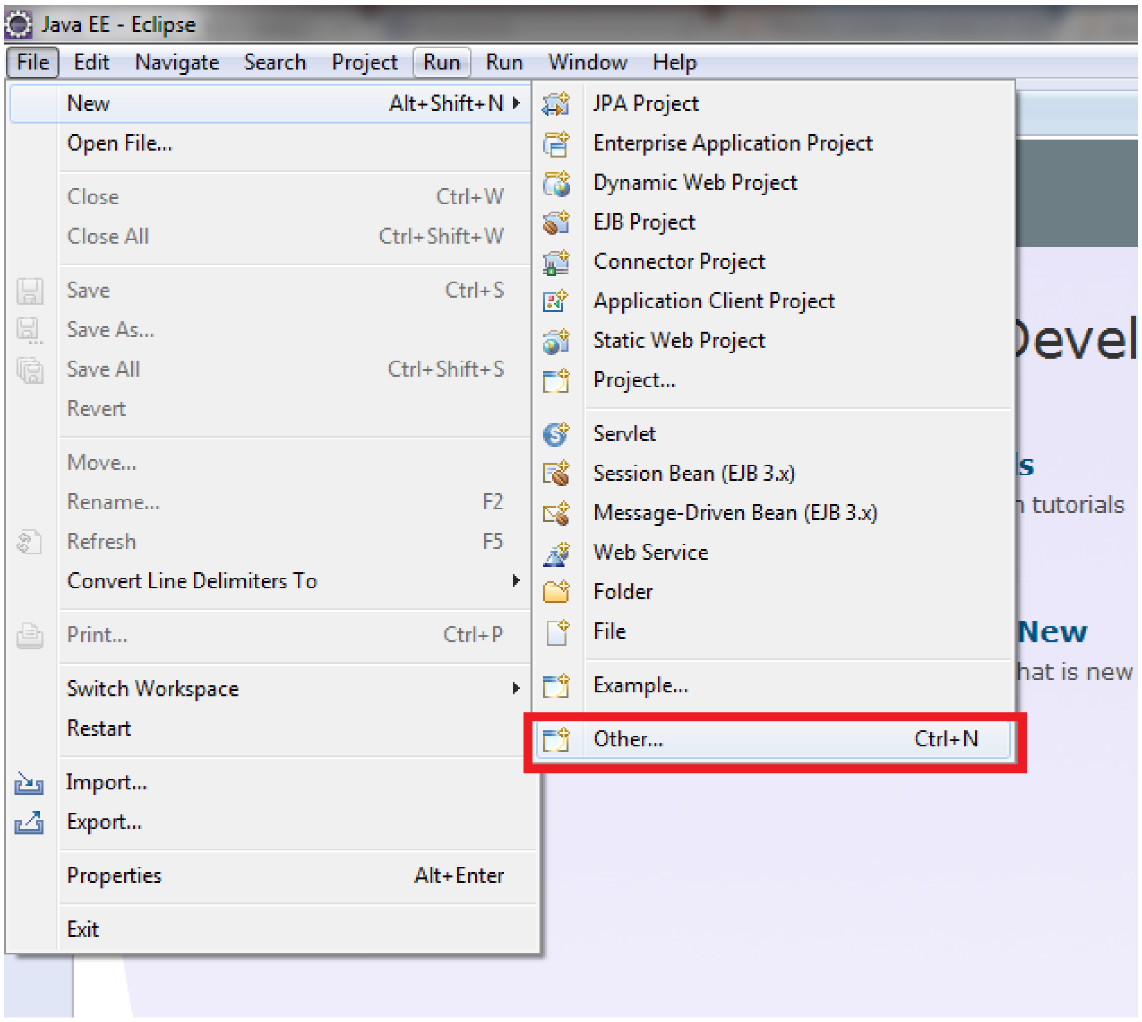

Impaq

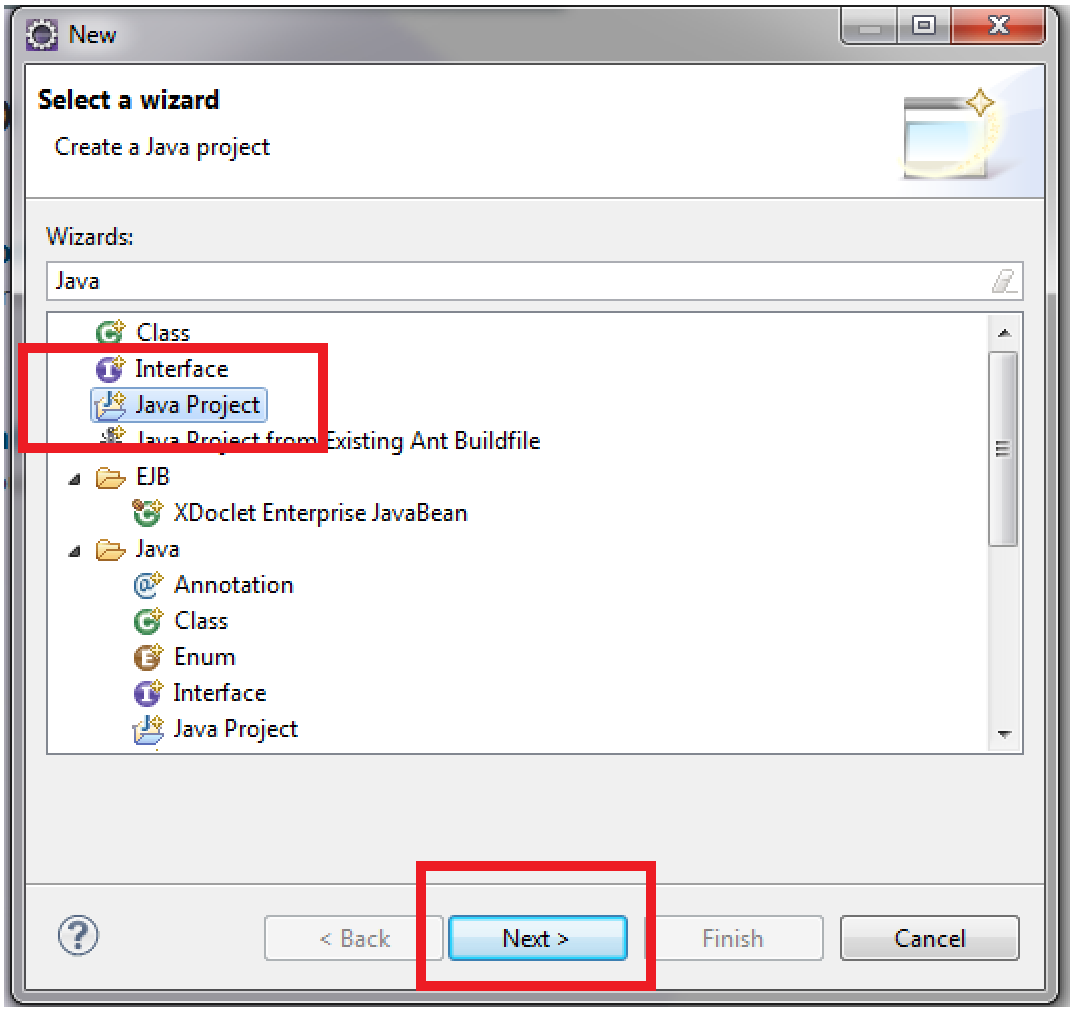
Name

coverage
docs
impaq_pos
java_doc
Impaq POS Point Of Sale Manual.pdf

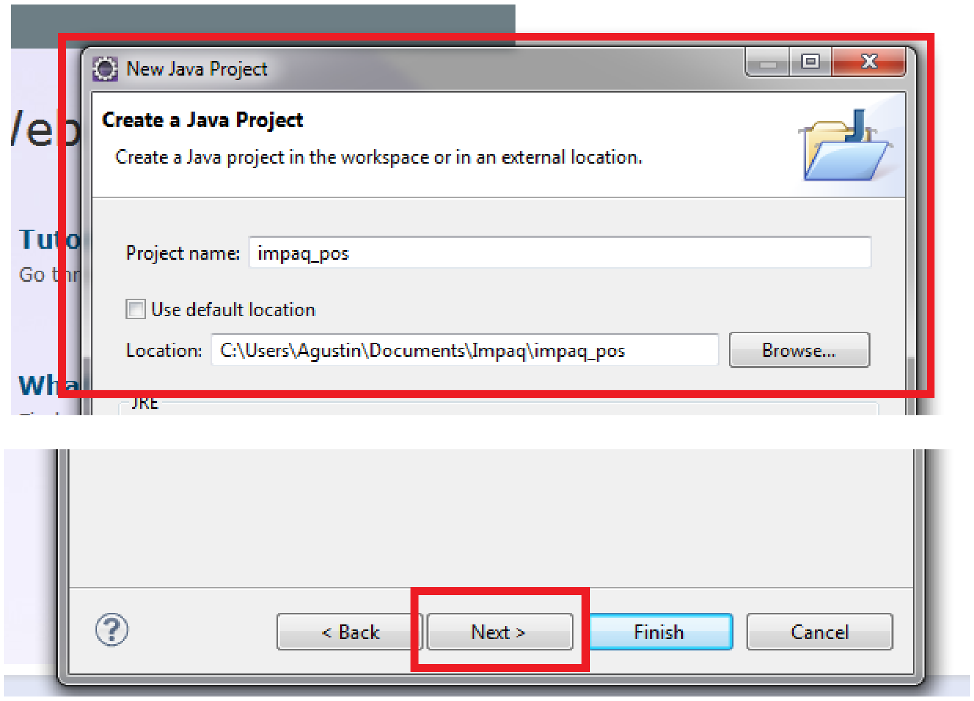
2. Installation on eclipse

a. Open eclipse and select new Project (Java Project)




New
Select a wizard
Create a Java project
Wizards:
Java
Class
Interface
Java Project
Java Project from Existing Ant Buildfile
EJB
XDoclet Enterprise JavaBean
Java
Annotation
Class
Enum
Interface
Java Project
< Back
Next >
Finish
Cancel

b. Select the folder impaq_pos as the Location



Create a Java Project

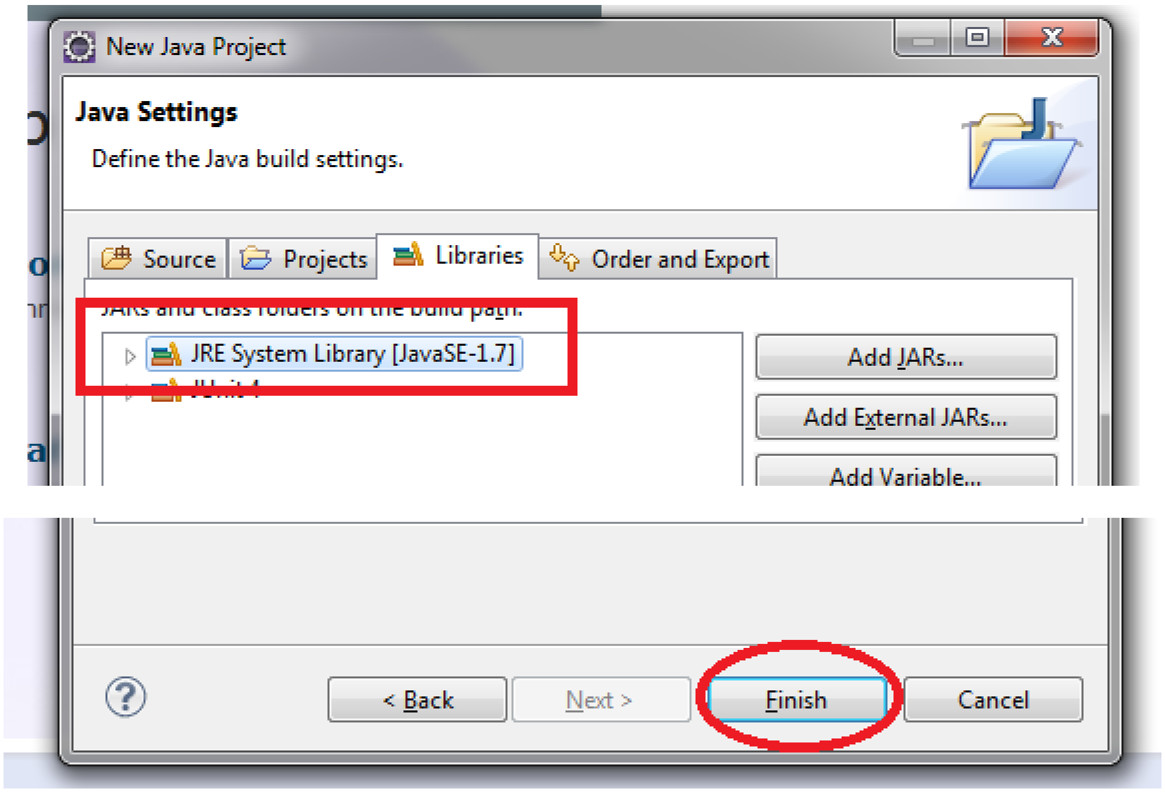
Create a Java project in the workspace or in an external location.

Project name: impaq_pos

Use default location

Location: C:\Users\Agustin\Documents\Impaq\impaq_pos

c. Verify the java version 1.7 to run the project



Java Settings

Define the Java build settings.

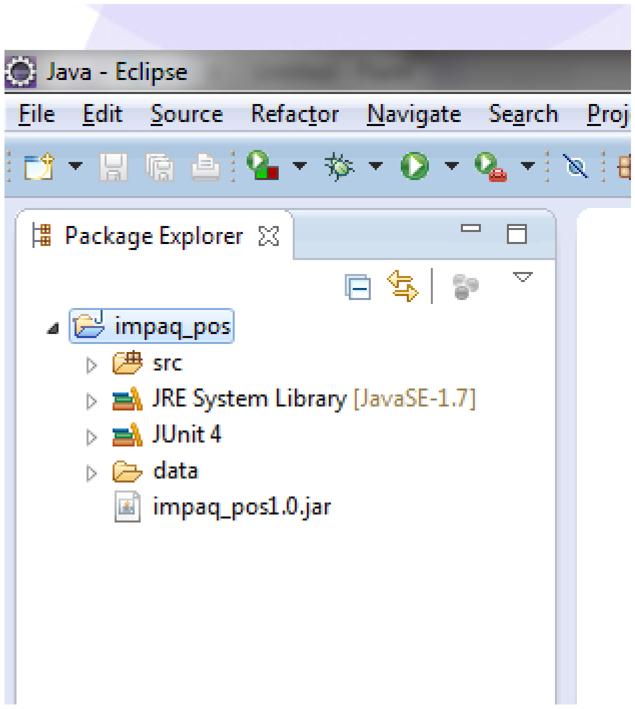
JARs and class folders on the build path:

JRE System Library [JavaSE-1.7]

JUnit 4

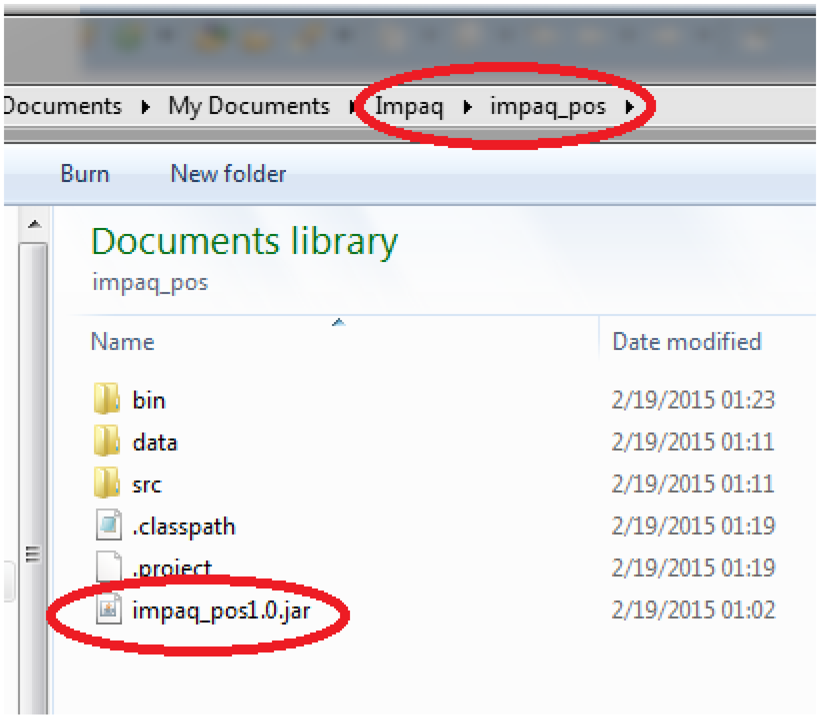
Note: If older version of java is configured some features as Unit Testing or coverage code may not be available. (http://java.com for more info)

d. The project should be now available


Java - Eclipse
File Edit Source Refactor Navigate Search Proj
Package Explorer
impaq_pos
src
JRE System Library [JavaSE-1.7]
JUnit 4
data
impaq_pos1.0.jar

3. Final User View

a. Under the folder impaq_pos there is the runnable version double click it.



Documents ▸ My Documents ▸ Impaq ▸ impaq_pos ▸

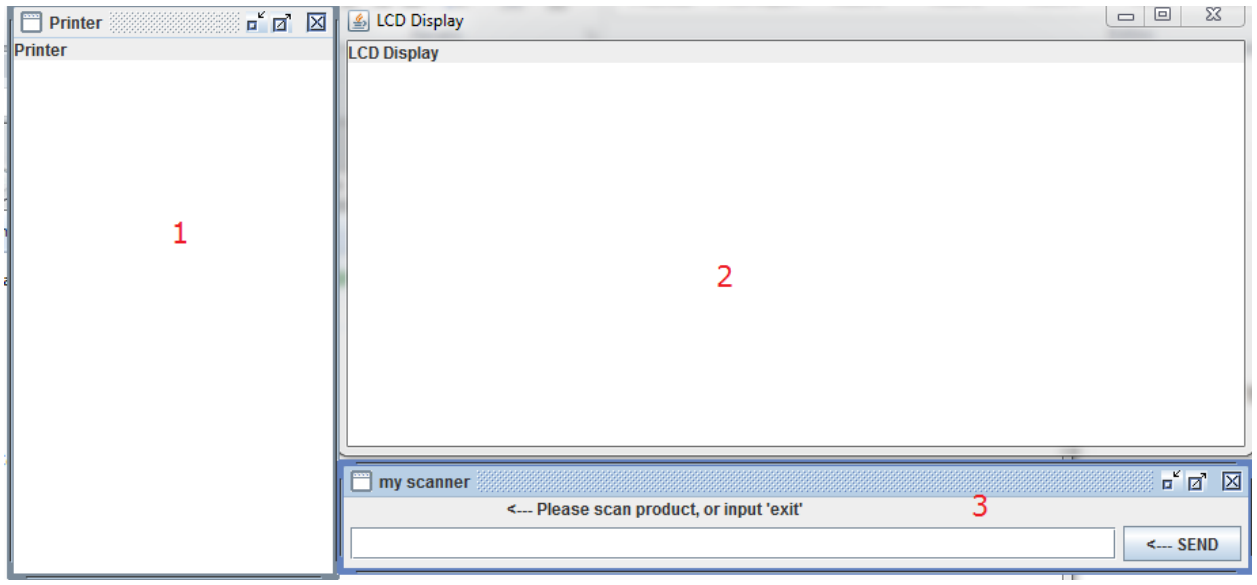
Burn New folder

Documents library
impaq_pos

Name	Date modified
bin	2/19/2015 01:23
data	2/19/2015 01:11
src	2/19/2015 01:11
.classpath	2/19/2015 01:19
.project	2/19/2015 01:19
impaq_pos1.0.jar	2/19/2015 01:02

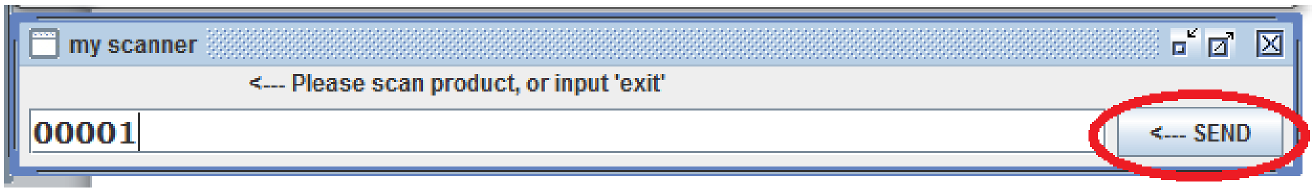
Note: this jar must be executed in the same folder as "data" thus it requires all those configuration files to start.

b. The three default views



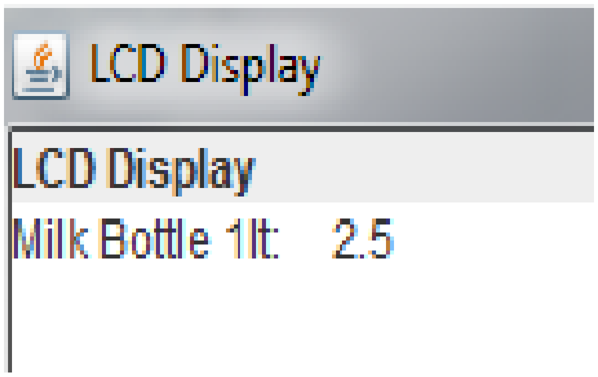
1. Represents the printer (Invoice details will be shown there)
2. Represents LCD Display (Details of scanning a product will appear there)
3. Barcode Input Scanner (Responds to ‘exit’ or to barcodes as defined in the config.properties file. The current configuration has by default 16 products with 5-digit barcode from (00001 to 00016).

c. Scanning a product: type in the barcode and hit the 'Enter key' or the 'send' button.



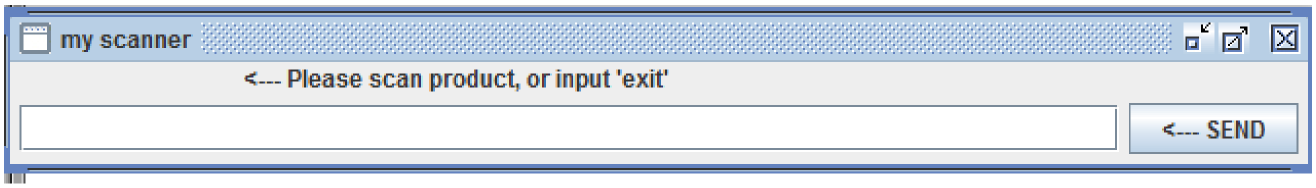
```
00001
```

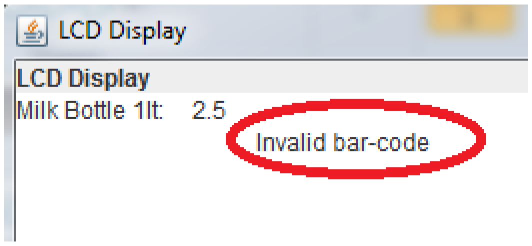
According to specification if the barcode is valid on the display will show its price and name.



```
LCD Display
Milk Bottle 1lt:    2.5
```

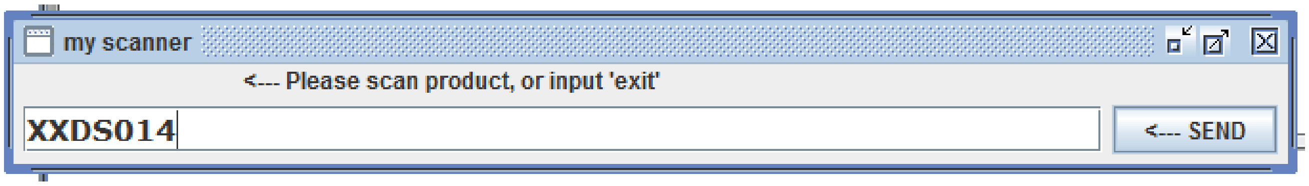
If the input is sent empty 'invalid code' message appears on the Display

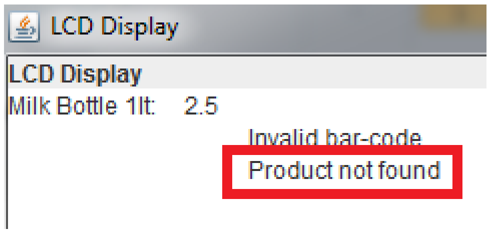




```
LCD Display
Milk Bottle 1lt:    2.5
                 Invalid bar-code
```

And if the barcode sent is not found as a configured product

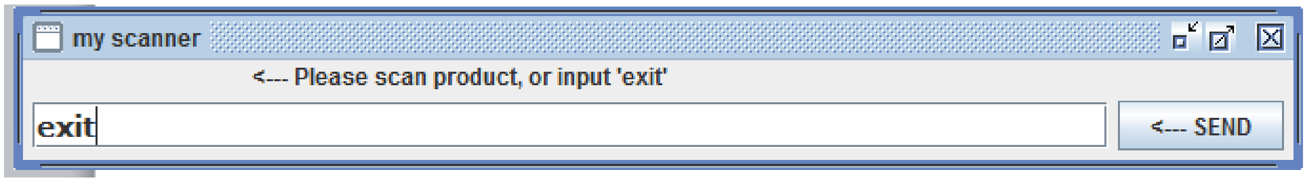


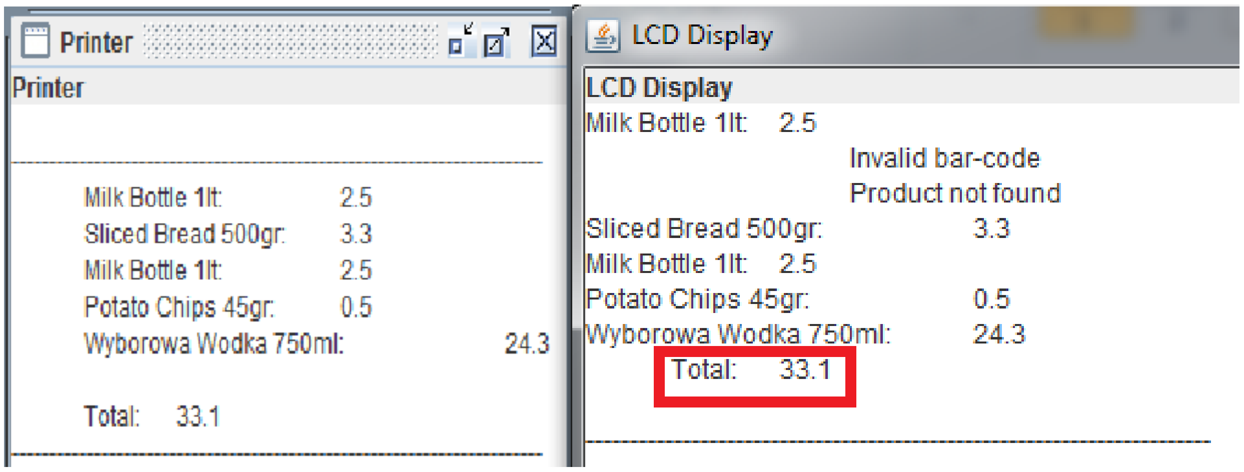


LCD Display

```
Milk Bottle 1lt:    2.5
                            Invalid bar-code
                            Product not found
```

And finally after sending the key-word 'exit', the total is printed on Display, and the invoice is printed on the 'printer' window.





Printer

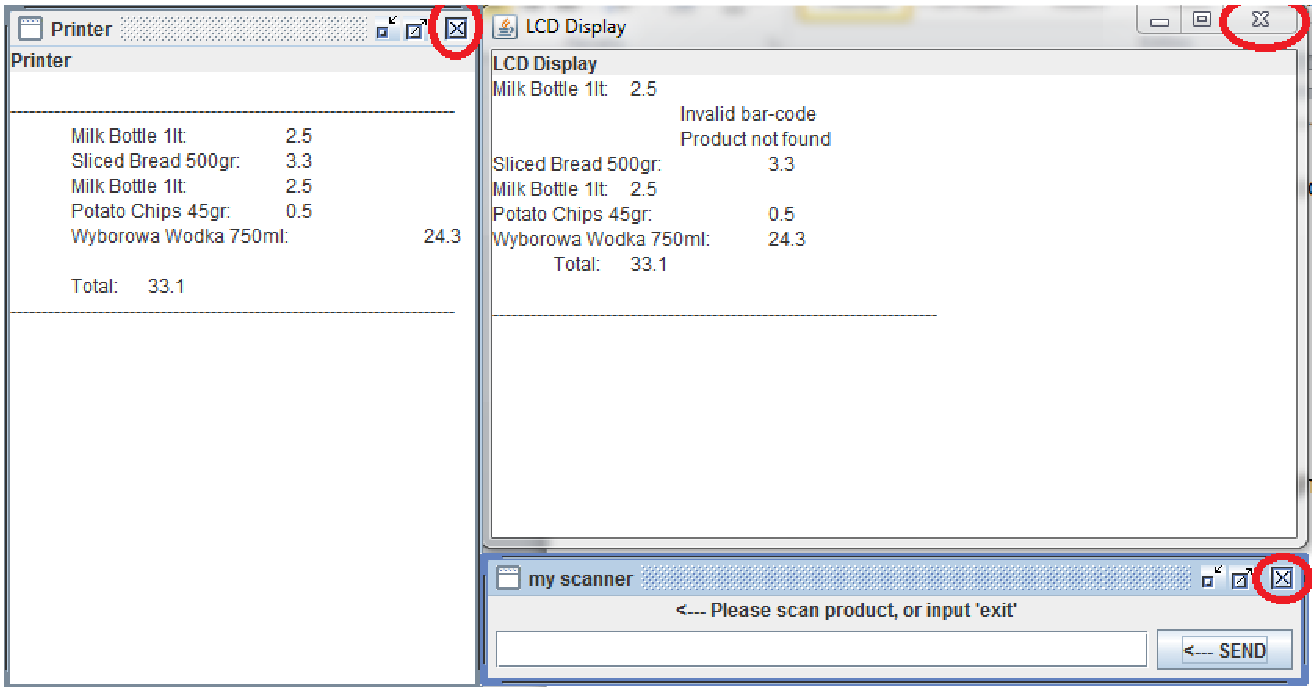
```
-------------------------------------------------
     Milk Bottle 1lt:          2.5
     Sliced Bread 500gr:       3.3
     Milk Bottle 1lt:          2.5
     Potato Chips 45gr:        0.5
     Wyborowa Wodka 750ml:              24.3

     Total:    33.1
-------------------------------------------------
```

LCD Display

```
Milk Bottle 1lt:    2.5
                          Invalid bar-code
                          Product not found
Sliced Bread 500gr:                3.3
Milk Bottle 1lt:    2.5
Potato Chips 45gr:                 0.5
Wyborowa Wodka 750ml:              24.3
      Total:    33.1

-------------------------------------------------------
```

At this point the systems starts a new order list and the process can continue as explained.

d. To Exit the system click on any of the 'x' symbols on the top right corner of the windows.



Printer

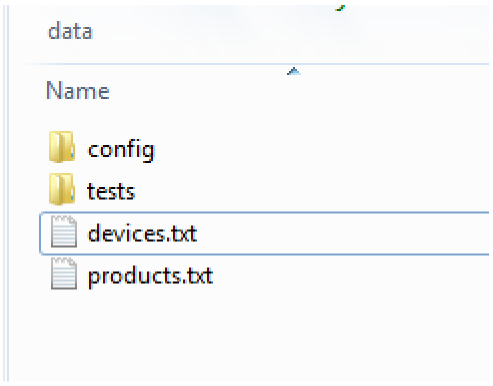
```
------------------------------------------------------------------
        Milk Bottle 1lt:            2.5
        Sliced Bread 500gr:         3.3
        Milk Bottle 1lt:            2.5
        Potato Chips 45gr:          0.5
        Wyborowa Wodka 750ml:                   24.3

        Total:     33.1
------------------------------------------------------------------
```

LCD Display

```
Milk Bottle 1lt:    2.5
                          Invalid bar-code
                          Product not found
Sliced Bread 500gr:             3.3
Milk Bottle 1lt:    2.5
Potato Chips 45gr:              0.5
Wyborowa Wodka 750ml:           24.3
         Total:     33.1

------------------------------------------------------------------
```

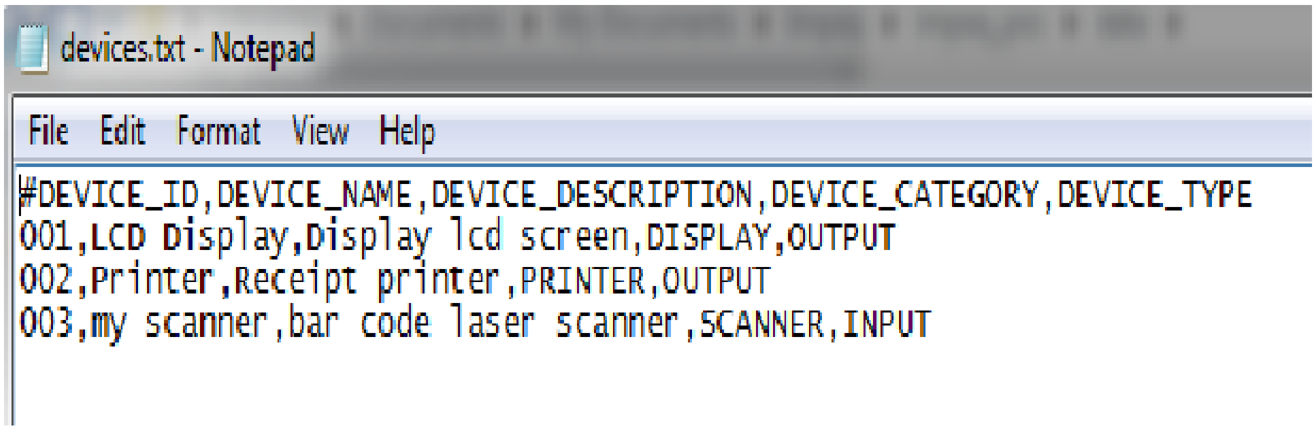
<--- Please scan product, or input 'exit'

4. The 'data' configuration folder



a. The devices.txt file

This file contains the mock of the devices that will be loaded into the system, currently are available three categories (DISPLAY, SCANNER and PRINTER) in two types (INPUT and OUTPUT).

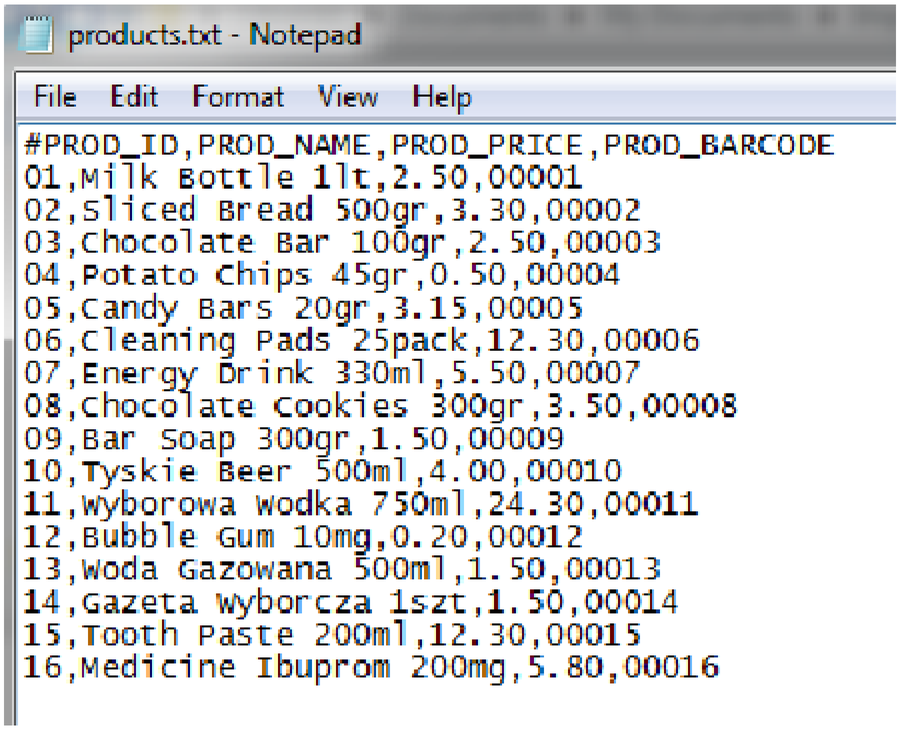


```
#DEVICE_ID,DEVICE_NAME,DEVICE_DESCRIPTION,DEVICE_CATEGORY,DEVICE_TYPE
001,LCD Display,Display lcd screen,DISPLAY,OUTPUT
002,Printer,Receipt printer,PRINTER,OUTPUT
003,my scanner,bar code laser scanner,SCANNER,INPUT
```

The header of this file contains the specification on how each column is taken as property of the device.

b. The products.txt file

This file contains the list of products available to use in the system (please note that currently there is no **inventory track** in the system so is assumed that the inventory is always available from the products defined in the file, this functionality may be available in future releases).



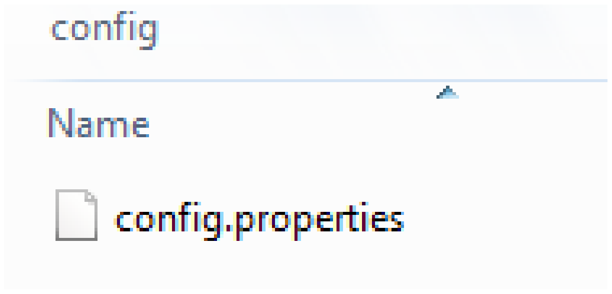
```
#PROD_ID,PROD_NAME,PROD_PRICE,PROD_BARCODE
01,Milk Bottle 1lt,2.50,00001
02,Sliced Bread 500gr,3.30,00002
03,Chocolate Bar 100gr,2.50,00003
04,Potato Chips 45gr,0.50,00004
05,Candy Bars 20gr,3.15,00005
06,Cleaning Pads 25pack,12.30,00006
07,Energy Drink 330ml,5.50,00007
08,Chocolate Cookies 300gr,3.50,00008
09,Bar Soap 300gr,1.50,00009
10,Tyskie Beer 500ml,4.00,00010
11,Wyborowa Wodka 750ml,24.30,00011
12,Bubble Gum 10mg,0.20,00012
13,Woda Gazowana 500ml,1.50,00013
14,Gazeta Wyborcza 1szt,1.50,00014
15,Tooth Paste 200ml,12.30,00015
16,Medicine Ibuprom 200mg,5.80,00016
```

This file also contains a coma separated value format, where every column can be described by its corresponding tag on the header.

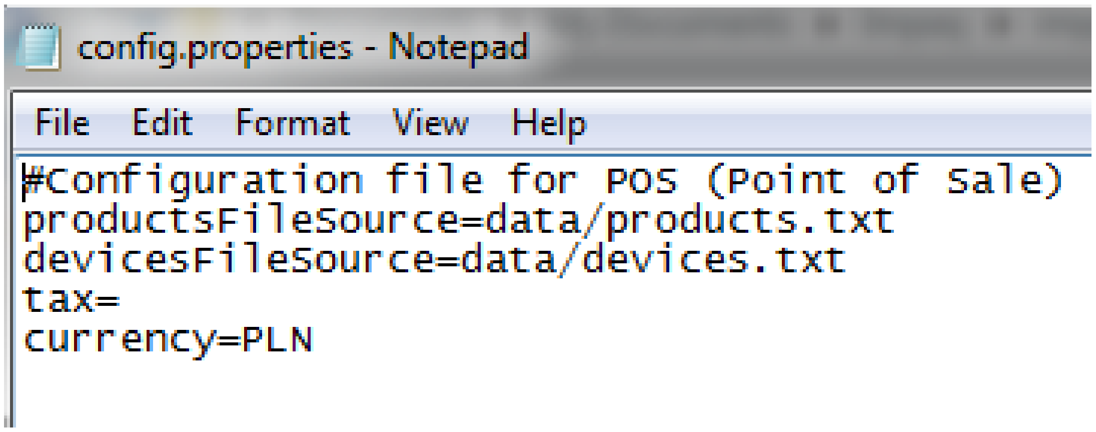
Note: The use of ‘Prod_id’ and ‘barcode’ is thinking on future releases when multiple catalogs can be loaded so a product may acquire diferent price according to the catalog barcode assigned.

c. The config.properties file

This file is contained on the config folder and contains the basic information for the system to run, (before modifying it make sure all the implications are clear)



All the configurations are stored here, it resolves the name files for the products and for devices, and also has a two additional parameters, 'tax' and 'currency', thinking on future releases, where the tax can be calculated and the currency shown depends on it.

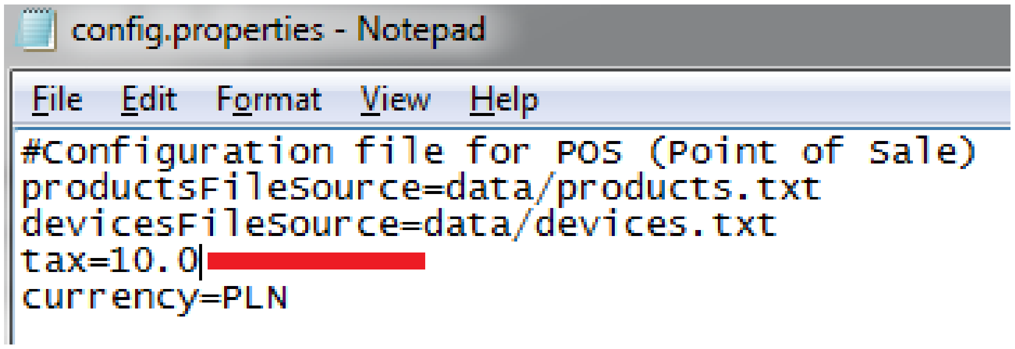


```
#Configuration file for POS (Point of Sale)
productsFileSource=data/products.txt
devicesFileSource=data/devices.txt
tax=
currency=PLN
```

If the tax property is stored as a number greater or equal to zero, the system will print additional details calculating subtotal, tax collected and total.

d. Setting a tax value

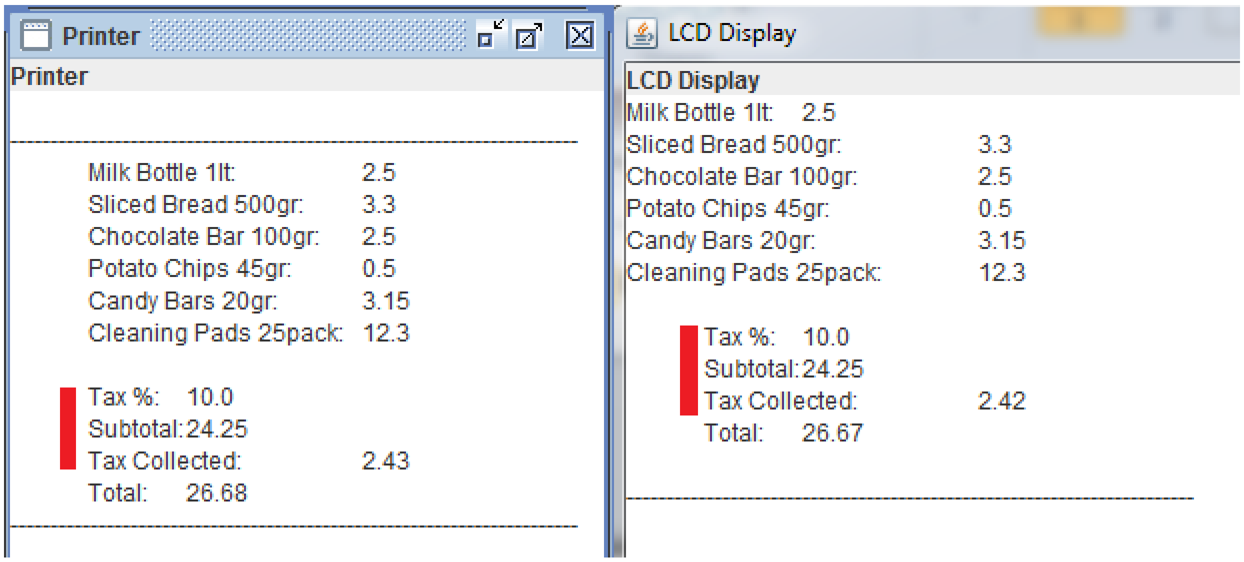
In the following example the system prints the details based on a tax modification of 10.0.



```
#Configuration file for POS (Point of Sale)
productsFileSource=data/products.txt
devicesFileSource=data/devices.txt
tax=10.0
currency=PLN
```

For the configuration changes to take place a restart of the system is required.

In the following example the new values can be seen.



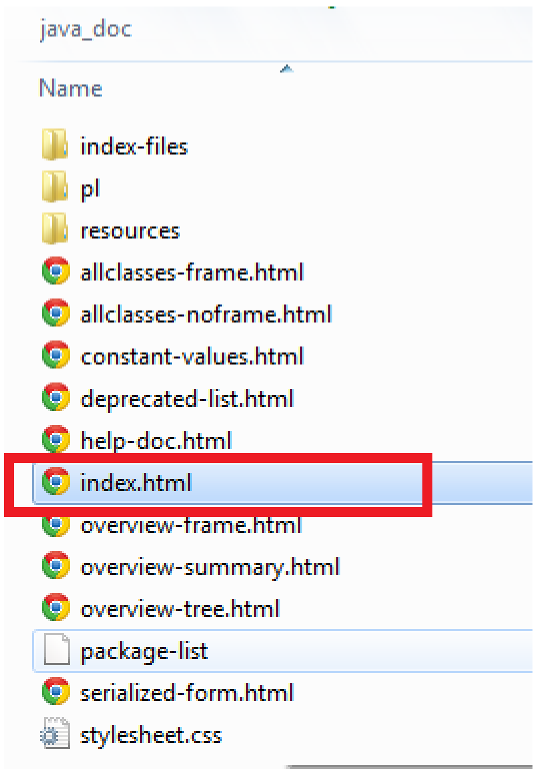
```
Printer

------------------------------------------------------------
        Milk Bottle 1lt:        2.5
        Sliced Bread 500gr:     3.3
        Chocolate Bar 100gr:    2.5
        Potato Chips 45gr:      0.5
        Candy Bars 20gr:        3.15
        Cleaning Pads 25pack:   12.3

        Tax %:    10.0
        Subtotal: 24.25
        Tax Collected:          2.43
        Total:    26.68
------------------------------------------------------------
```

```
LCD Display
Milk Bottle 1lt:   2.5
Sliced Bread 500gr:              3.3
Chocolate Bar 100gr:             2.5
Potato Chips 45gr:               0.5
Candy Bars 20gr:                 3.15
Cleaning Pads 25pack:            12.3

        Tax %:    10.0
        Subtotal: 24.25
        Tax Collected:           2.42
        Total:    26.67

------------------------------------------------------------
```

5. Technical details

a. Javadoc

The source code used javadoc annotations on comments for classes and methods, in order to provide a full documentation of the code. This information may be accessed by opening the index.html file under the folder 'java_doc' folder.

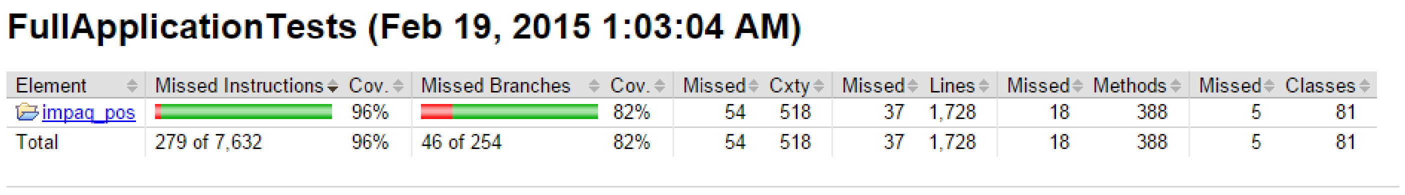

java_doc

Name

index-files
pl
resources
allclasses-frame.html
allclasses-noframe.html
constant-values.html
deprecated-list.html
help-doc.html
index.html
overview-frame.html
overview-summary.html
overview-tree.html
package-list
serialized-form.html
stylesheet.css

b. Code coverage

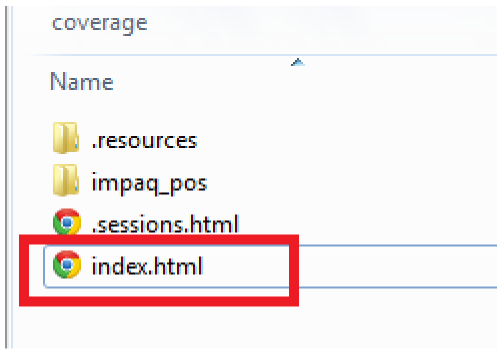
The coverage test results on last run, provided a 96% coverage, where the missing 4% corresponds to exception handling code where basically no relevant information may be required to return.



FullApplicationTests (Feb 19, 2015 1:03:04 AM)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
impaq_pos	🟥🟩	96%	🟥🟩	82%	54	518	37	1,728	18	388	5	81
Total	279 of 7,632	96%	46 of 254	82%	54	518	37	1,728	18	388	5	81

The results can be seen on opening the index.html file under the folder coverage.



coverage

Name

.resources

impaq_pos

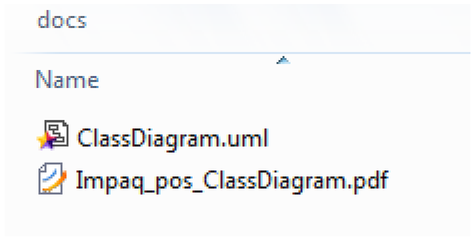
.sessions.html

index.html

The coverage documentation was executed using the ECLEmma eclipse plugin (http://www.eclemma.org/) available for free on the eclipse market place.

c. UML Class Diagram

Under the docs folder the source files for StarUML (http://staruml.sourceforge.net/v1/download.php) for class diagram and an exported PDF versions are available, the image was not included in this document to keep an adequate resolution.



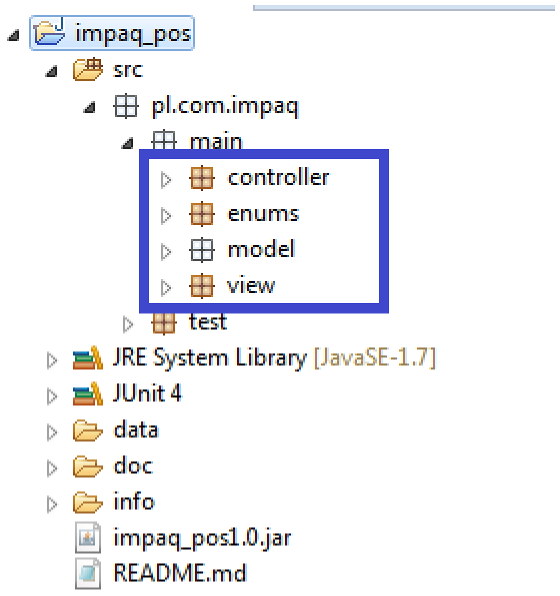
docs

Name

ClassDiagram.uml

Impaq_pos_ClassDiagram.pdf

The system was designed based on MVC design pattern; It presents a clear path to add new functionalities to the system. The package presentation reflects this.

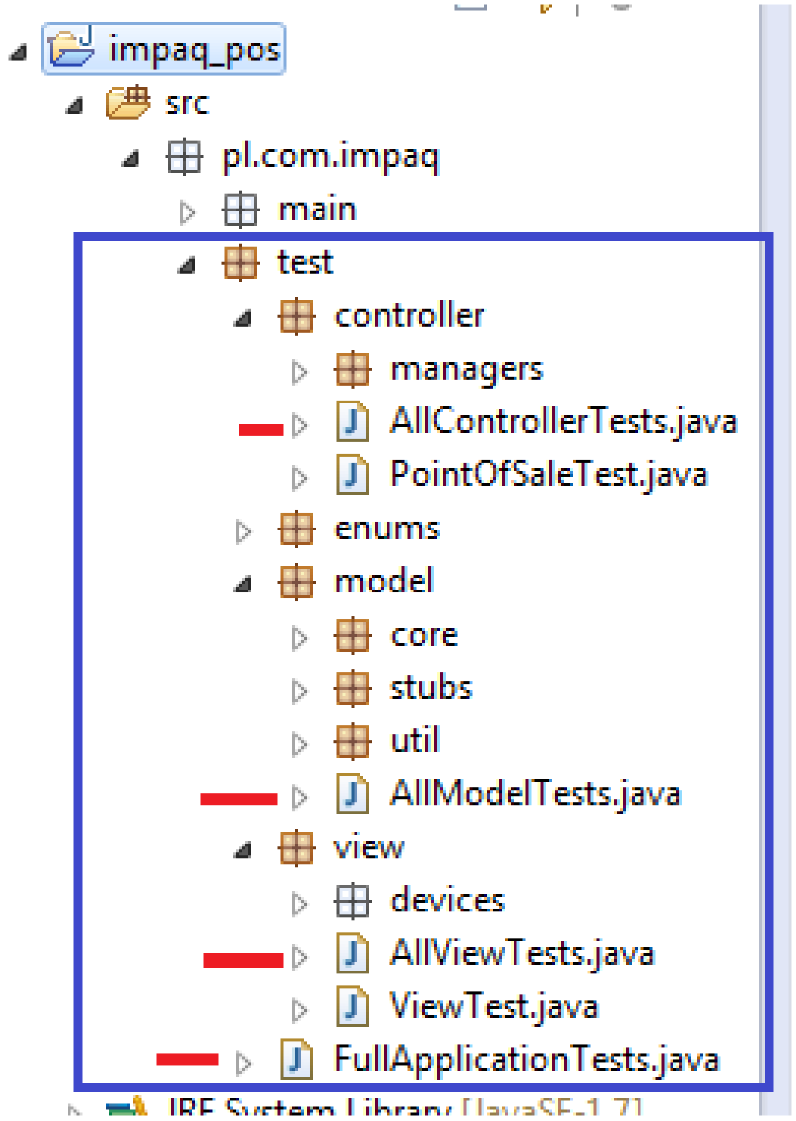


```
impaq_pos
  src
    pl.com.impaq
      main
        controller
        enums
        model
        view
      test
  JRE System Library [JavaSE-1.7]
  JUnit 4
  data
  doc
  info
  impaq_pos1.0.jar
  README.md
```

The singleton pattern was also used to keep only one instance of PointOfSale, the View and the PropertiesReader class.

d. Unit Testing

The unit testing was developed using JUnit4, in the source there are two major packages under 'pl.com.impaq' main and test, under the test package all the JUnit test files and Test Suites are available, there are 4 major Suites to test the code as shown in the following figure.



The FullApplicationTest.java file contains all the class specifications and is the one that complies with the 96% percent coverage specified on the previous subjects.

6. Conclusions

The system specifications where short, and the huge demand and extension of the possible PointOfSale system, made it a very challenging task to be performed, any questions or comments on this manual are welcome.

cabra.agustin@gmail.com
https://github.com/acabra85