# Programming – Assignment 1

The finished programme fully meets the specification detailed in the initial project outline. Several assumptions were made about the data that could be entered. Firstly, it was assumed that the initial balance could not exceed £10,000 positive or negative credit. It was also assumed that the maximum price for a bottle of wine was £500 and that only 500 bottles could be ordered at once. This is a sensible assumption for any small to medium sized wine merchant. As per the project specification, no upper limit was set on the length of the wine name string. It was also decided to not limit the length of the customer name string.

A variety of invalid test data was used to check that the programme handled such situations correctly. With regards to the customer name, an empty string was tested. In this case the programme terminated, as required in the project specification. Initial balances not within the allowed range, empty balances and balances not in the double format were all tested. In each case a JOption error message was displayed (seen in figures 5 and 6), and the user was allowed to re-enter the balance. An example of this can be seen in figure 14.

Table 1 shows a selection of the invalid data inputs tested for the wine name, price and quantity fields. Where applicable, a reference to the JOption error message that the data produced is shown. These scenarios represent just a small number of those tested. Figure 2 in the appendix shows an invalid input being entered into the GUI. In all test scenarios the programme operated as intended. It should be noted that the design decision was made to allow prices with more than 2 decimal places. In such a scenario the double is simply rounded to two decimal places before it is used to alter the customer’s balance.

A variety of valid data was also tested to ensure that the programming correctly displayed the transaction details and correctly updated the customer balance. Examples of these can be seen at the bottom of table 1. Figures 3 and 4 in the appendix shows the GUI after valid data has been entered. Both the sale function and the refund function were tested. A large range of values were tested, and results were checked with a calculator. The data shown in the table represents just a small sample of that tested. In each case the programme operated as intended, displaying the correct transaction cost and the new customer balance. When the customer is in credit it is displayed in the balance field (see figure 4). The text fields were also cleared as specified.

Table 1. Sample test data and the results produced

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario | Wine Name | Price | Quantity | Result |
| Empty wine name | None | £5.50 | 3 | JOption error message (Figure 10) |
| Negative price | Champagne | £-21.35 | 11 | JOption error message (Figure 9) |
| Negative quantity | Rose | £15.05 | -6 | JOption error message (figure 8) |
| Empty price | Chardonnay | None | 30 | JOption error message (figure 7) |
| Empty quantity | White | £10.00 | None | JOption error message (figure 13) |
| Price – wrong type | Red | Twenty | 5 | JOption error message (figure 7) |
| Quantity – wrong type | Prosecco | £56.90 | 6.35 AND xyz | JOption error messages (figure 13) |
| Out of range price | Pinot Noir | £700.50 | 10 | JOption error message (figure 12) |
| Out of range quantity | Own brand | £3.59 | 600 | JOption error message (figure 11) |
| Valid data – refund | Champagne | £3.97 | 1 | Cost: £3.18. Balance/GUI updated |
| Valid data – refund | Rose | £199.99 | 23 | Cost: £3679.82. Balance/GUI updated |
| Valid data – sale | Chardonnay | £7.49 | 425 | Cost: £3183.25. Balance/GUI updated |
| Valid data – sale | Prosecco | £39.00 | 5 | Cost: £195.00. Balance/GUI updated |

# Appendix 1 – Screen Dumps

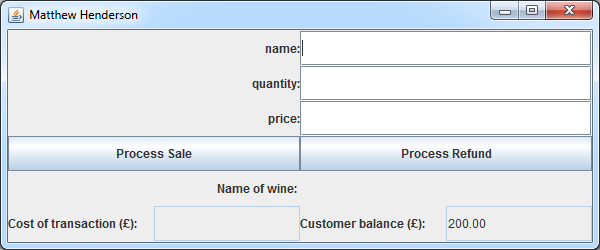


Figure 1. Programme correctly gets the customer’s name and initial balance from the input dialog boxes.

|  |  |
| --- | --- |
|  |  |

Figure 2. An invalid input and the corresponding error message that it generates.

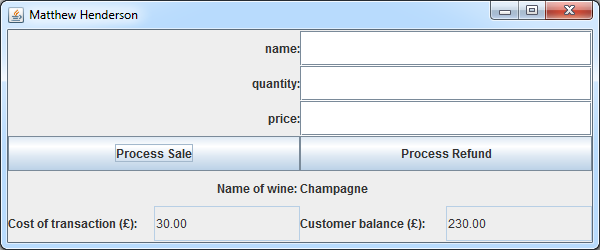


Figure 3. A correctly processed sale transaction.

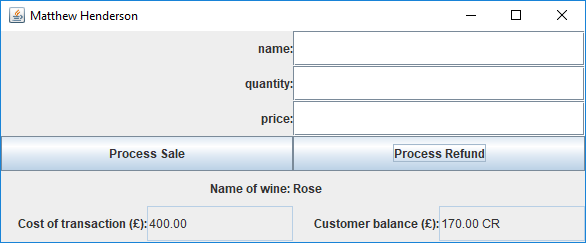


Figure 4. A correctly processed refund transaction showing the customer in credit

## Error Messages

|  |  |
| --- | --- |
| Figure 5. Error message for empty initial balances, or balances not entered as a valid double. | Figure 6. Error message for out of range balances |
| Figure 7. Error message for prices not entered as a valid double | Figure 8. Error message for negative quantities |
| Figure 9. Error message for negative prices | Figure 10. Error message for an empty name field |
| Figure 11. Error message for out of range quantities | Figure 12. Error message for out of range prices |
| Figure 13. Error message for quantities not entered as a valid integer | |
|  |  |

Figure 14. An incorrectly entered customer balance and the error message it displays