

Cross Platform Application Development Program Documentation

Contents

User Documentation	1
Scope & Functionality	1
Platform Testing	1
Windows 7	2
Ubuntu Linux	3
Known Bugs	3
Manual	5
Editing Items	6
Technical Documentation.....	8
GUI Design	8
Code Documentation.....	9

User Documentation

Scope & Functionality

For my cross platform application I chose to simulate a vending machine, with physical buttons replaced with UI buttons. You can add, edit and remove items using a dialog. The item list is stored in a comma separate values file in the program directory. Items are listed with their codes on the main vending screen and you can select an item code which is highlighted and the item and price displayed. When you add coins the messages on screen change.

Platform Testing

I tested the application on two platforms, Windows 7 and Linux. Usability of the interface on both platforms was as expected, bar font sizes. Here are my results:

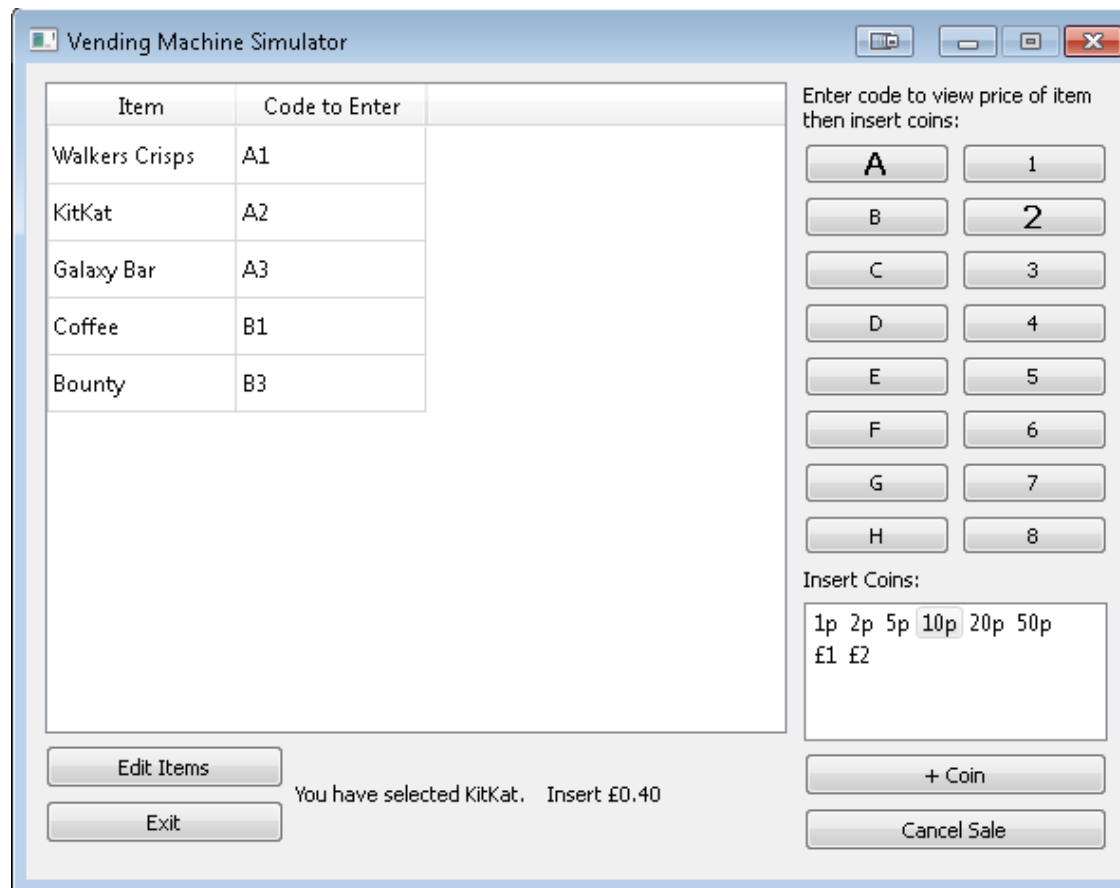


Fig 3. Main screen in action on Windows 7

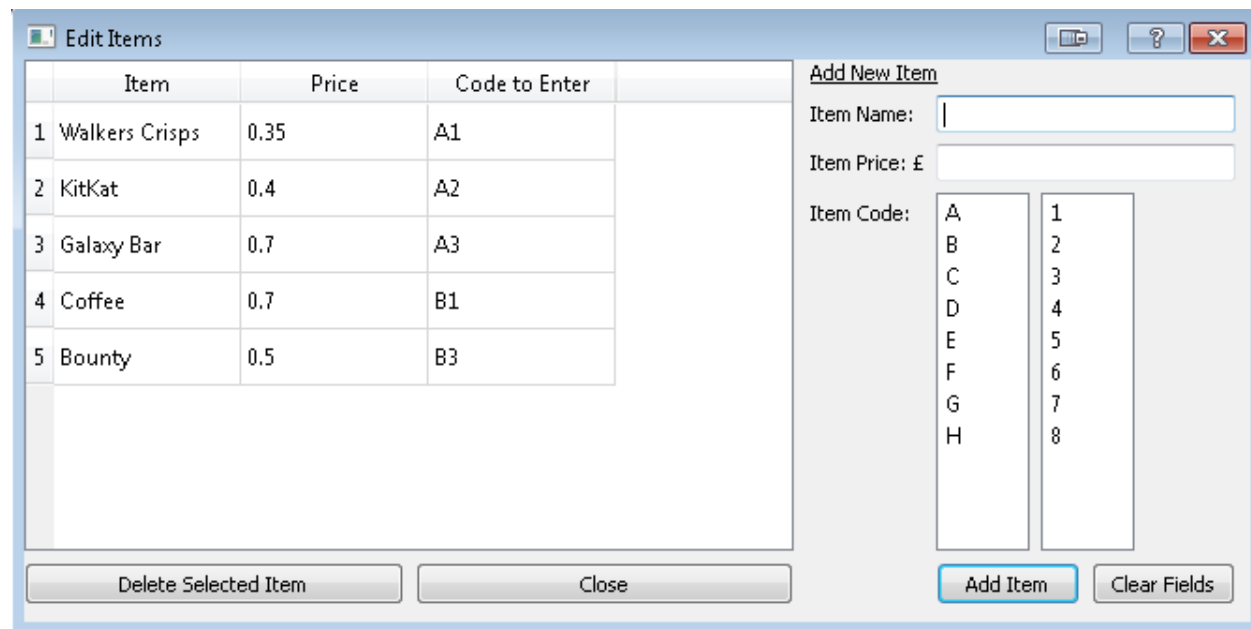


Fig 4. Edit screen in action on windows 7

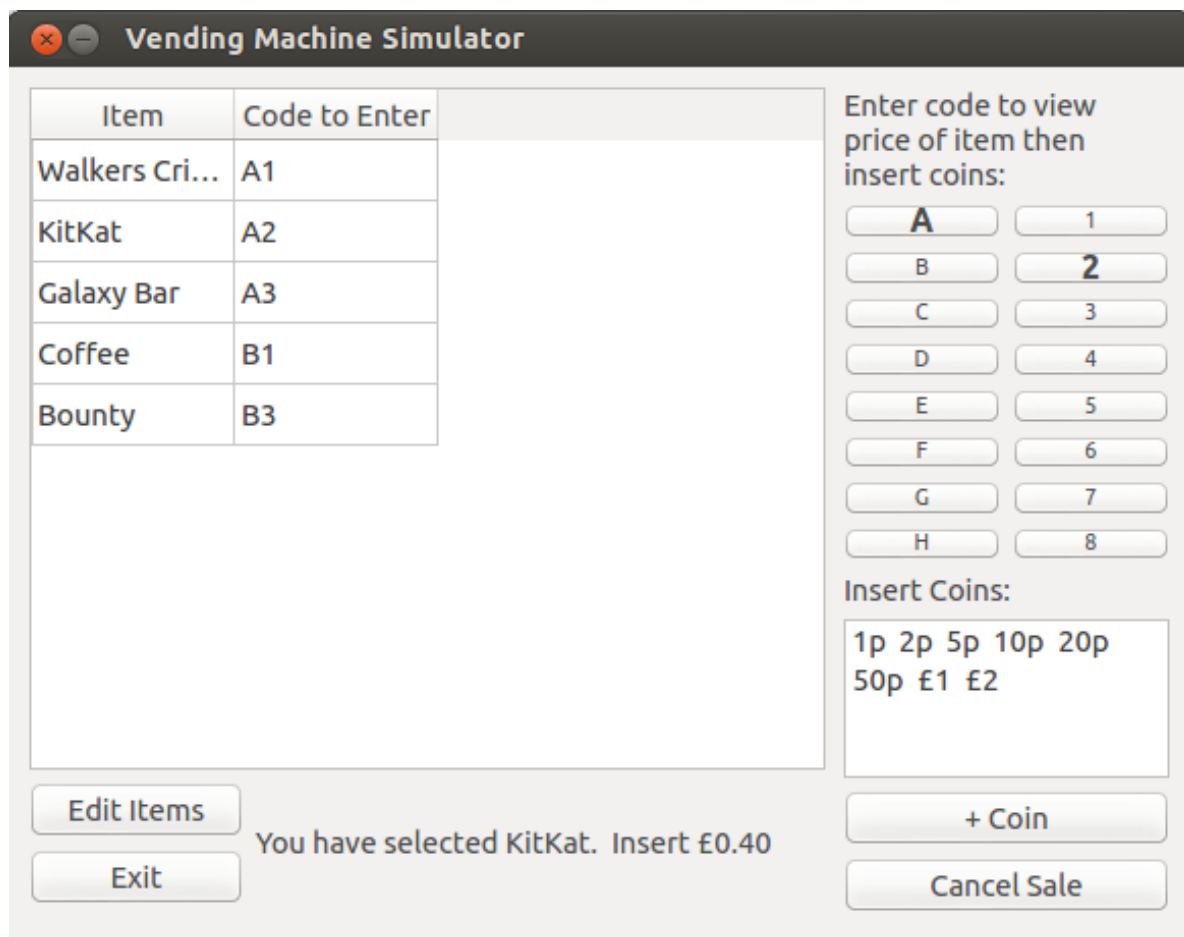


Fig 5. Vending interface on Ubuntu Linux

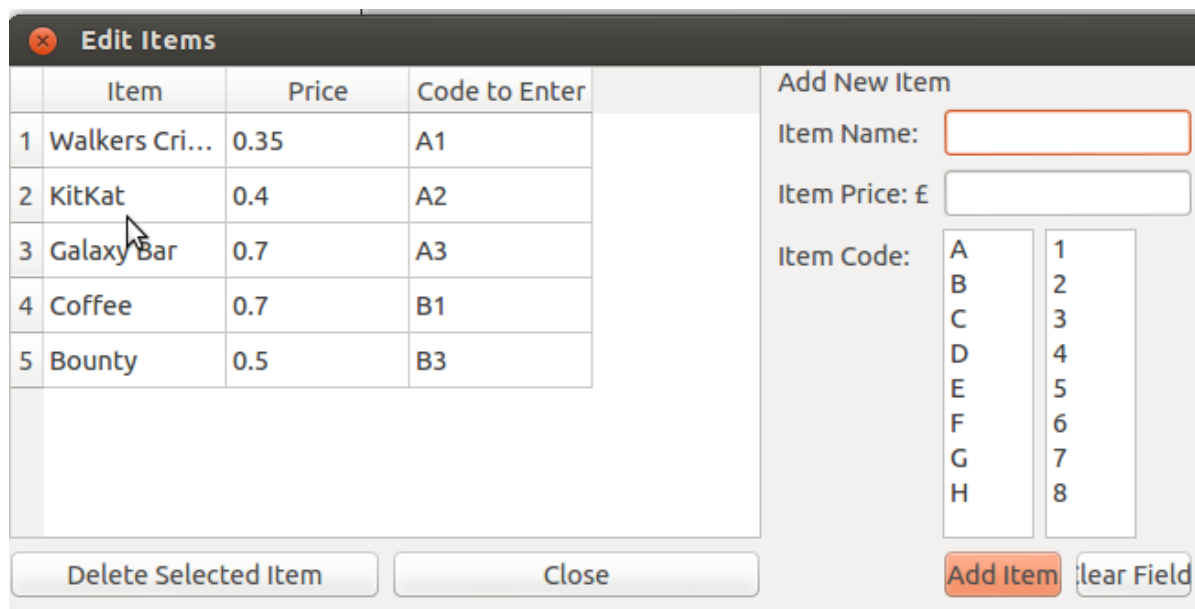


Fig 6. Edit items interface on Ubuntu. Buttons could be bigger; here the auto layout has failed, possibly to do with the size of the container grid layout.

Known Bugs

On Ubuntu Linux change correct amount of change isn't dealt correct, possible issue with the subtraction of amounts / number format used in Linux.

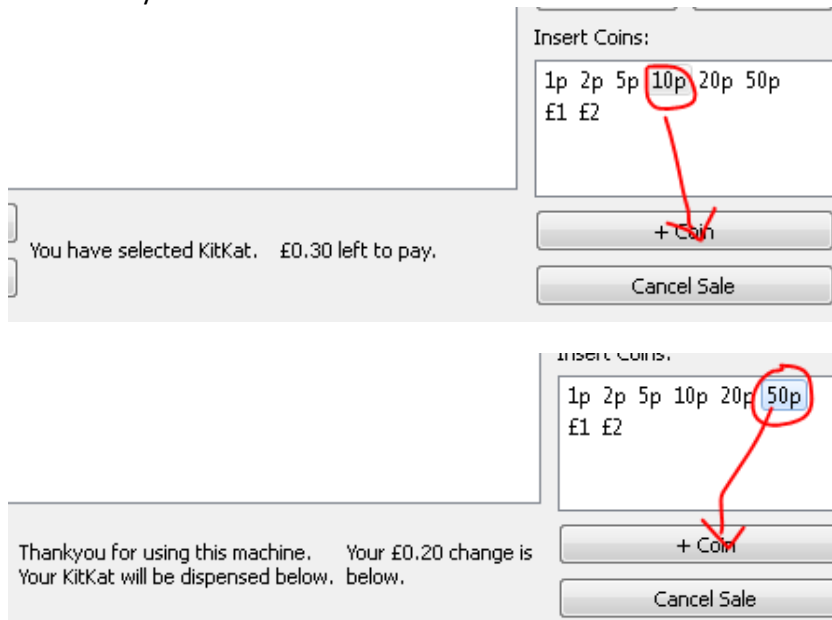


Fig 7 Adding coins on windows.

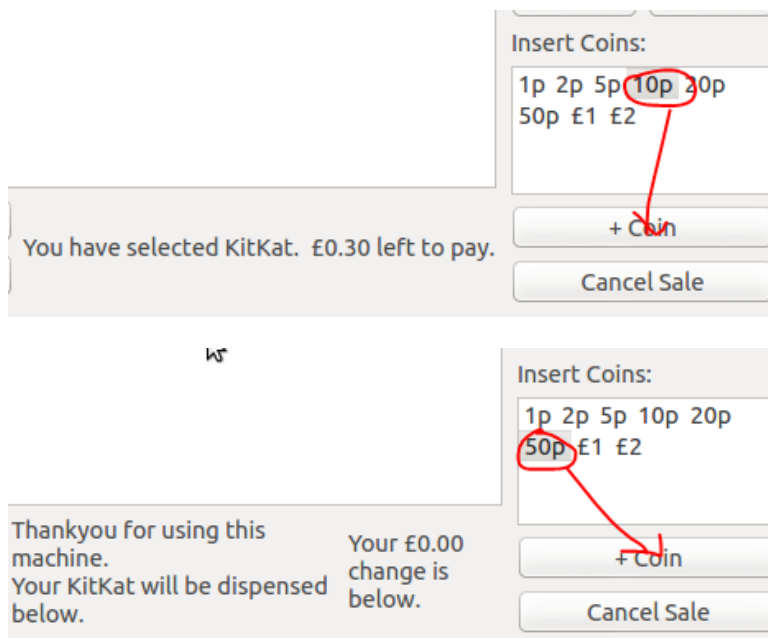


Fig 8. Adding the same coins on Linux. Notice the change bug.

However this only seems to work with decimal on decimal numbers, i.e. any coin under £1 as £5 was tried and paid with 6 x £2 and I got £1 change back.

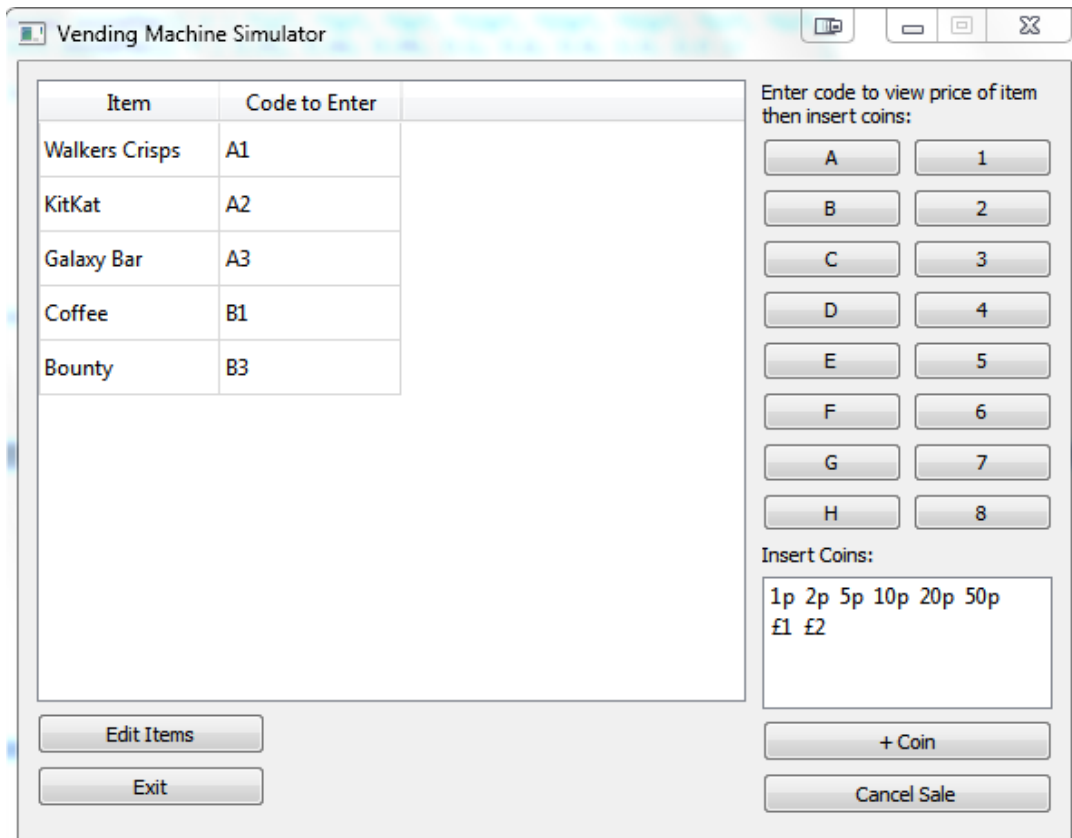
Here is the associated code:

```

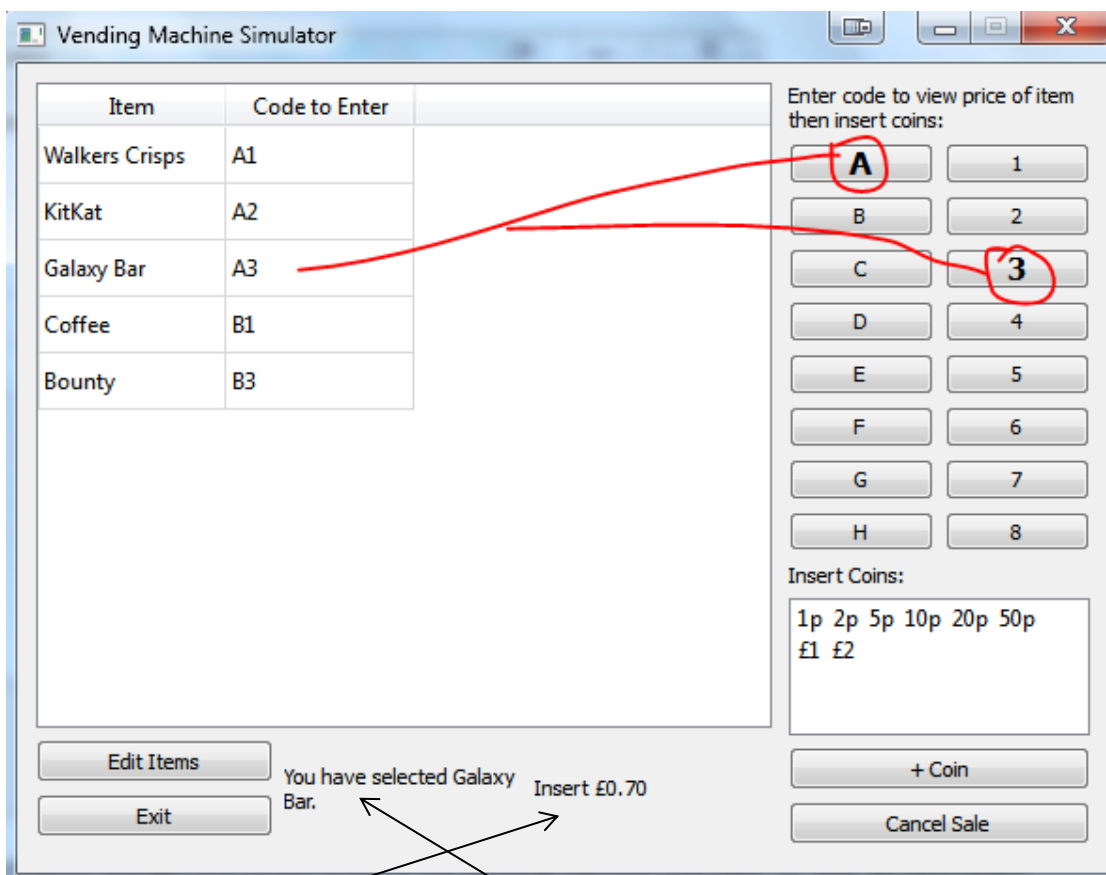
248     QString coinsTxt[] = { "1p", "2p", "5p", "10p", "20p", "50p", "£1", "£2" };
249     double coinsVal[] = { 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0 };
250
251     if(ui->lstCoins->currentItem() != NULL)
252     {
253         QString selectedCoin = ui->lstCoins->currentItem()->text();
254         for(int i = 0; i < 8; i++)
255         {
256             if(coinsTxt[i] == selectedCoin)
257             {
258                 totalToPay -= coinsVal[i];

```

Fig 9. vendingmachine.cpp, lines 246 - 258



When you start the application you get this screen which is the vending machine interface. To 'buy' an item select the corresponding code using the letter and number buttons. They will 'light up', becoming bold.



The number of coins needed and the item selected will be shown in the window.

Vending Machine Simulator

Item	Code to Enter
Walkers Crisps	A1
KitKat	A2
Galaxy Bar	A3
Coffee	B1
Bounty	B3

Enter code to view price of item then insert coins:

A 1
B 2
C 3
D 4
E 5
F 6
G 7
H 8

Insert Coins:
1p 2p 5p 10p 20p 50p
£1 £2

+ Coin
Cancel Sale

Edit Items
Exit

You have selected Galaxy Bar. £0.20 left to pay.

Insert coins by selecting coin value and clicking on '+ Coin' the amount will be updated accordingly. Cancel the sale by clicking Cancel Sale. Information will update accordingly.

Editing Items

Edit Items
Exit

Thankyou for using this machine. Your £0.00 change is below.

To add more items and edit item details including prices, click this button. This will bring up another dialog.

Edit Items

	Item	Price	Code to Enter
1	Walkers Crisps	0.35	A1
2	KitKat	0.4	A2
3	Galaxy Bar	0.7	A3
4	Coffee	0.7	B1
5	Bounty	0.5	B3

Add New Item

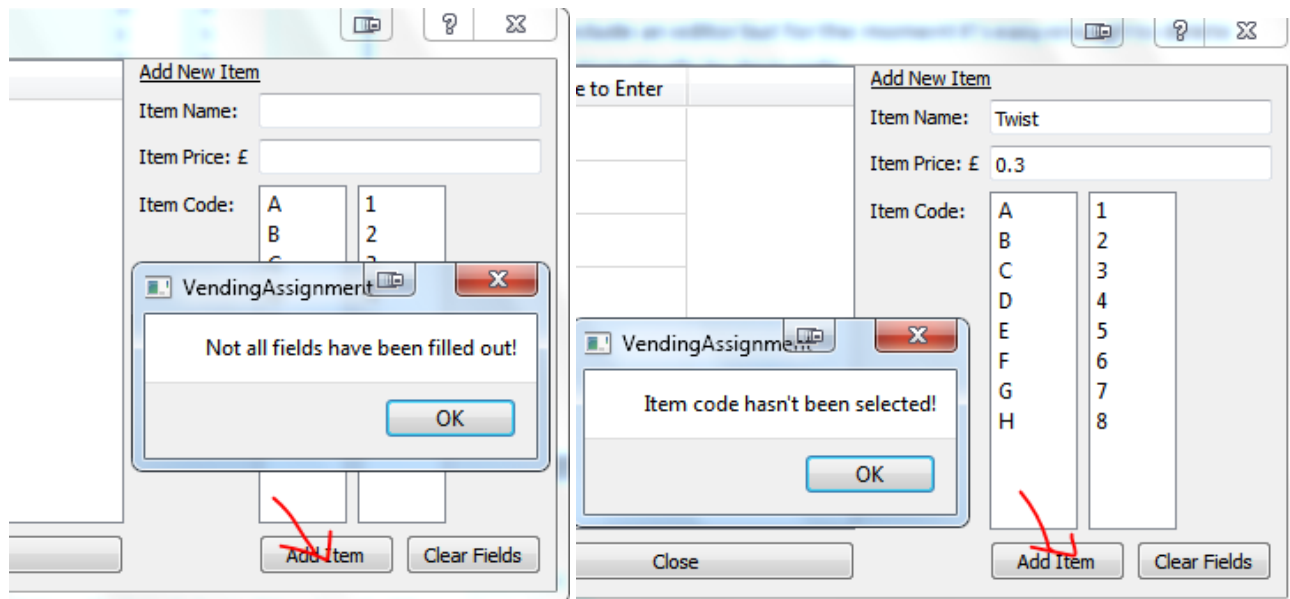
Item Name:

Item Price: £

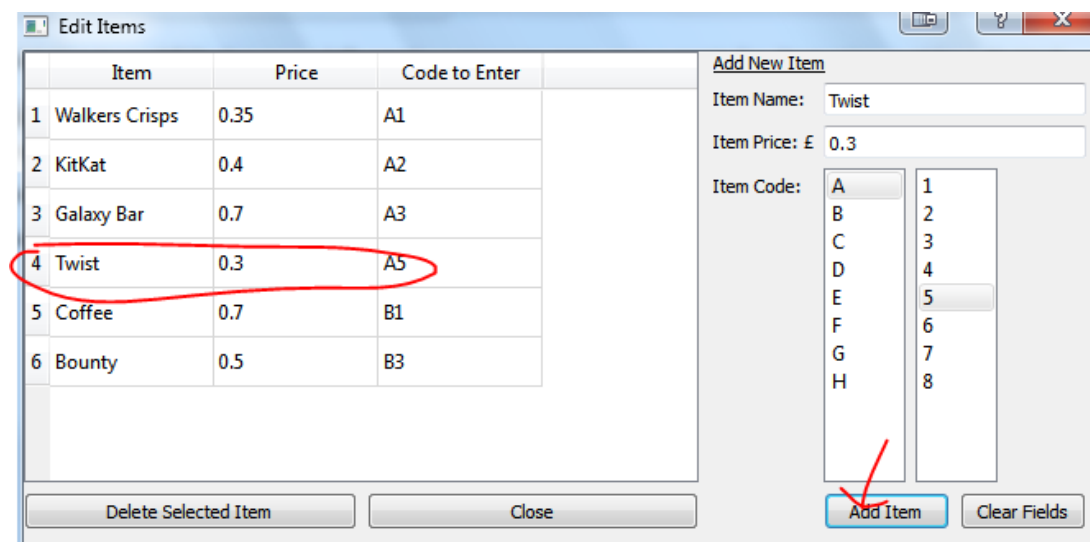
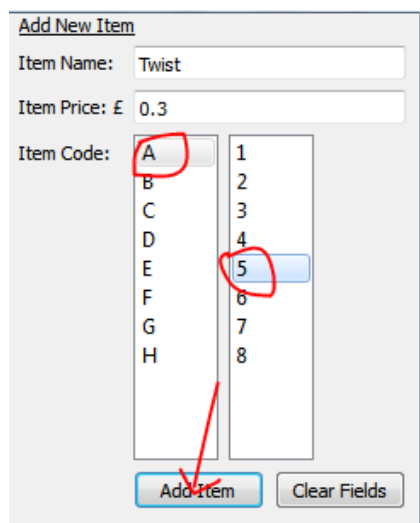
Item Code: A 1
B 2
C 3
D 4
E 5
F 6
G 7
H 8

Delete Selected Item
Close
Add Item
Clear Fields

Here you can delete items by clicking on them in the list and clicking 'Delete Selected Item' and add new items by filling out the form to the right and clicking Add Item. Addition of up to 64 items is possible, each with a separate code A – H and 1- 8. Future additions might include an editor but for the moment it's easy enough to delete something and re-add it as the list is sorted automatically by item code.



Validation is used if the boxes are left empty; a message pops up as a reminder. Item price must be a decimal.



GUI Design

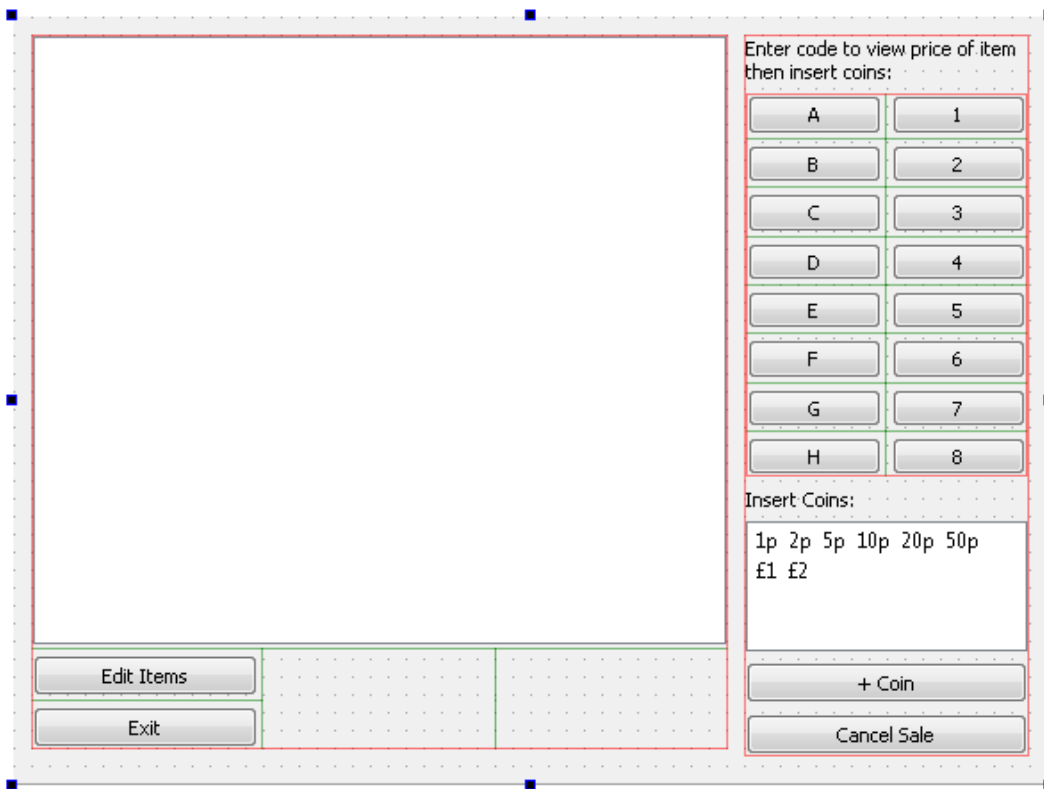


Fig 1. Vending interface on QT Creator UI designer.

This is the main screen. The vending machine push buttons A-H and 1-8 are inside a grid layout so that the application will manage the placement automatically and resize them per platform. The button panel and the other items on the right are in a vertical layout so they also snap to each other and stay one on top of another vertically.

The other screen areas to the left are arranged in a similar grid layout, with the two blank areas holding status text (one for price and one for item descriptions and messages). The grid allows the labels held within to size automatically to fit the (variable length) text. The main large box is a table view and lists items with their items and codes.

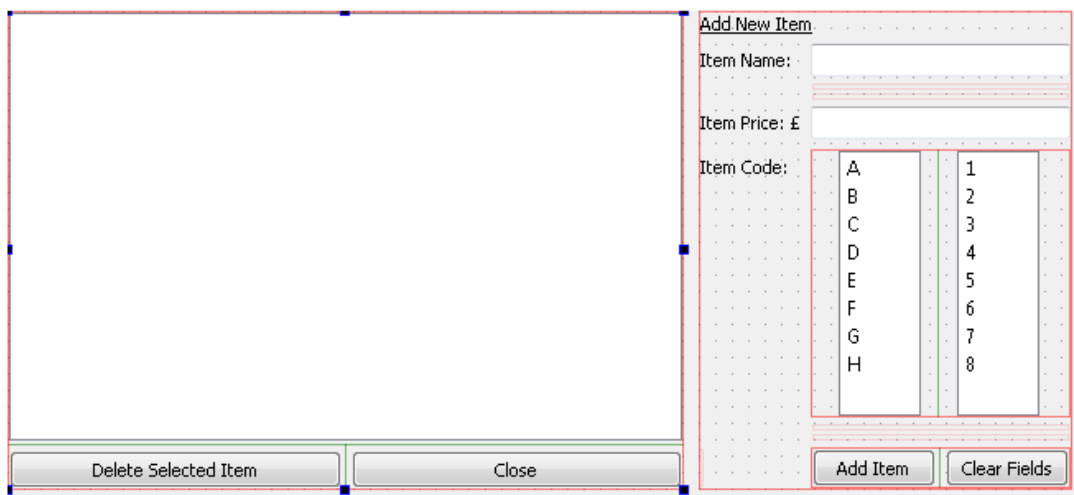








Fig 2. Item edit screen on QT Creator UI designer

This is the item editor screen, accessible by clicking 'Edit Items' on the main screen. A grid layout is used for the table view and the two buttons under it, to keep the buttons in line. On the right a vertical layout is used again for the form controls. The controls next to Item Code are in a grid layout of their own as two columns are needed here.

Code Documentation

I have documented the code with Doxygen and the resulting HTML files can be found in with the QT project.

Snapshots of the documentation below:

Class List	Class Index	Class Hierarchy	Class Members
Class Hierarchy			
This inheritance list is sorted roughly, but not completely, alphabetically:			
[detail level 1 2]			
 Item	Structure to define an Item , contains the attributes of an item. Used in a QList to form the central data store for the application		
 ItemLoader	Class to deal with Item objects, handles the conversion of disk files into items and central storage of them		
▼  QDialog			
 ItemDialog	Widget to allow the user to edit Items and add new Items to the vending machine		
▼  QWidget			
 VendingMachine	Main vending machine operation window		

Class List	Class Index	Class Hierarchy	Class Members
All	Functions	Variables	

Here is a list of all documented class members with links to the class documentation for each member:

- addItem() : **ItemLoader**
- deleteItemByItemCode() : **ItemLoader**
- getItemList() : **ItemDialog**
- getItemsAsList() : **ItemLoader**
- itemCode : **Item**
- ItemDialog() : **ItemDialog**
- itemName : **Item**
- itemPrice : **Item**
- readItemsFromFile() : **ItemLoader**
- saveItemsToFile() : **ItemLoader**
- setupTableData() : **VendingMachine**
- sortItemList() : **ItemLoader**
- VendingMachine() : **VendingMachine**
- ~VendingMachine() : **VendingMachine**