MSc Computer Science

School of Computing, Engineering and Digital technologies

Object-oriented Programming

ICA Element 2: Development Report

Name: Yaswanth Sai Chinthakayala

Student ID: W9640628

Table of Contents

1. Discussion of GUI Application

1.1 Overview

In this step, GUI application needs to be developed and modified from task 3 in element 1.

- The task 3 is a console-based application, to transform it into GUI, most of the code remain same and some of the code sections need to change.
- The arraylist data should be loaded into JTable and displayed.
- Adding items, buying items from the shop system needs to be enabled.

Table 1: Use of components from previous element

Console component	**Functionality**	**Change for GUI Version**
ASCStockItem.java	Data class	No change
stockItemsList	ArrayList for ASCStockItems	No change
loadData ()	Loads input data to arraylist	No change
saveData ()	Saves the arraylist data to a file	No change
buyItem ()	Buy an item from stock	Needs to be modified according to JTable
addStock ()	Add an item to the stock	Needs to be modified according to JTable

1.2 GUI Application

Graphic User Interface design needs to be created in the design tab of NetBeans. It shows how our application will be displayed to the users in the end. The designed GUI is shown in the figure below.

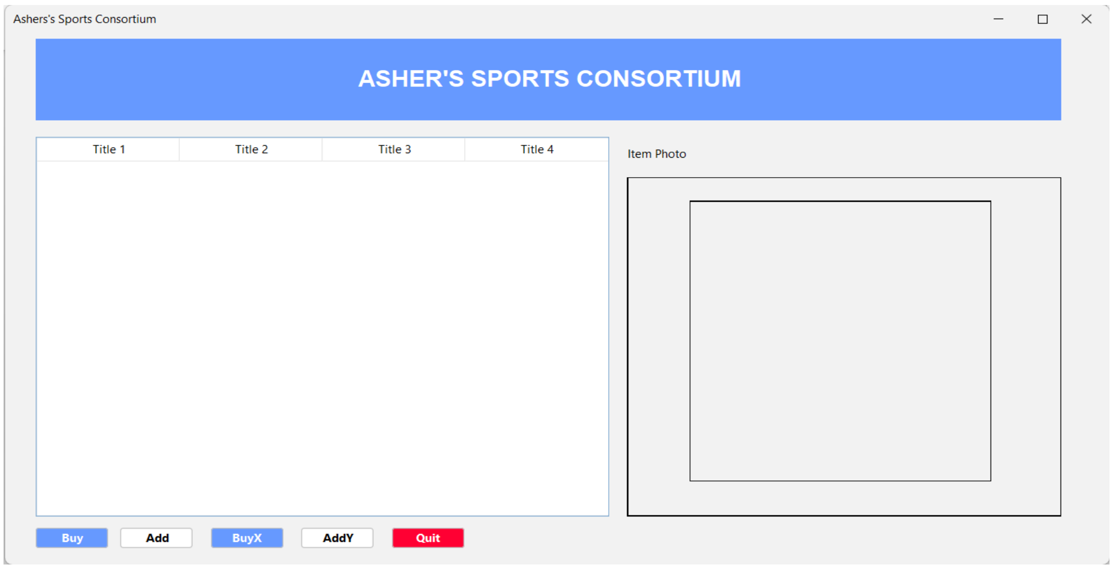


Figure 1: Blueprint of GUI

The elements present in the GUI are:

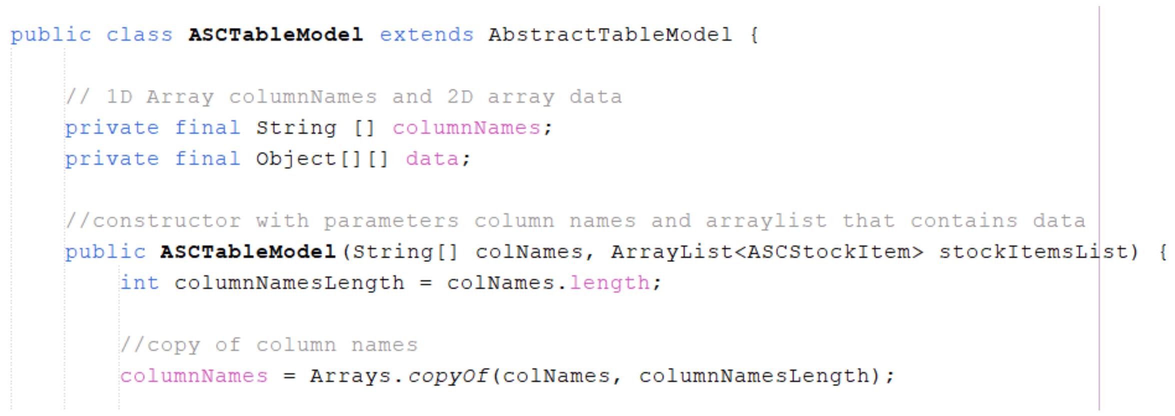
Table 2: GUI Elements

Swing Items	**Names**
JTable	ascStockItem
JButtons	buyButton, addButton, buyXButton, quitButton
Jpanel	photoPanel
JLabels	PhotoLabel, itemLabel

First, a JFrame class is created in the package and then on top of it all the components are placed. A JTable is created next, then all the jButtons are laid down. Next, a panel named photoPanel is built, In the panel two labels are used. PhotoLabel is used to display the image of the selected item and itemLabel is used to show the name of the item that got chosen.

1.3 Abstract Table Model

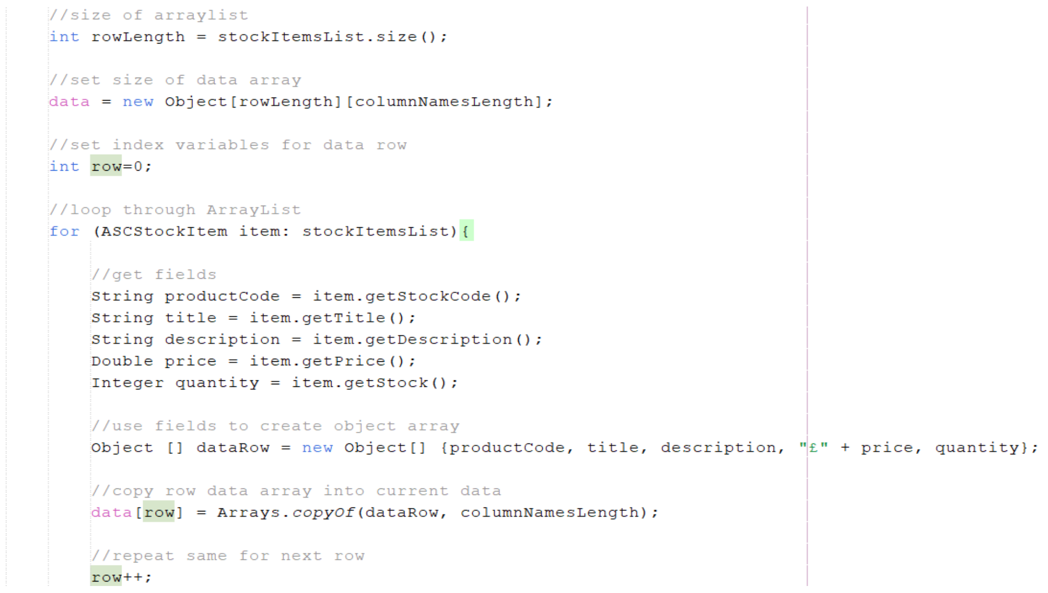
In the previous tasks, arraylist is used for storing and displaying items. But the JTable cannot display the items in arraylist directly. AbstractTableModel is needed for interpreting the values of arraylist into the JTable. The abstract class has many default methods that TableModel have, and all the listeners are maintained in this model. The abstract table model only needs three methods to be overridden to work.



```java
public class ASCTableModel extends AbstractTableModel {

    // 1D Array columnNames and 2D array data
    private final String [] columnNames;
    private final Object[][] data;

    //constructor with parameters column names and arraylist that contains data
    public ASCTableModel(String[] colNames, ArrayList<ASCStockItem> stockItemsList) {
        int columnNamesLength = colNames.length;

        //copy of column names
        columnNames = Arrays.copyOf(colNames, columnNamesLength);
```

Figure 2: ASC Table Model

The model that is used here is ascTableModel which extends to AbstractTableModel abstract class. The code is shown in the figure above. The constructor also needs the column names for the table and arraylist as parameters.

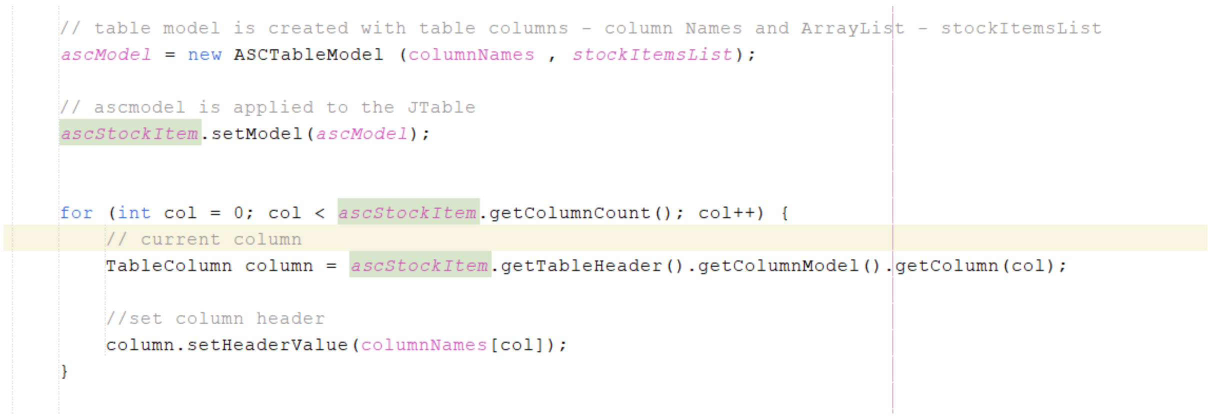


```java
        //size of arraylist
        int rowLength = stockItemsList.size();

        //set size of data array
        data = new Object[rowLength][columnNamesLength];

        //set index variables for data row
        int row=0;

        //loop through ArrayList
        for (ASCStockItem item: stockItemsList){

            //get fields
            String productCode = item.getStockCode();
            String title = item.getTitle();
            String description = item.getDescription();
            Double price = item.getPrice();
            Integer quantity = item.getStock();

            //use fields to create object array
            Object [] dataRow = new Object[] {productCode, title, description, "£" + price, quantity};

            //copy row data array into current data
            data[row] = Arrays.copyOf(dataRow, columnNamesLength);

            //repeat same for next row
            row++;
```

Figure 3: Updating rows for Table model.

The data is updated row wise by looping through the arraylist. Rows will be incremented until there are no items left.

1.4 Main Class Code



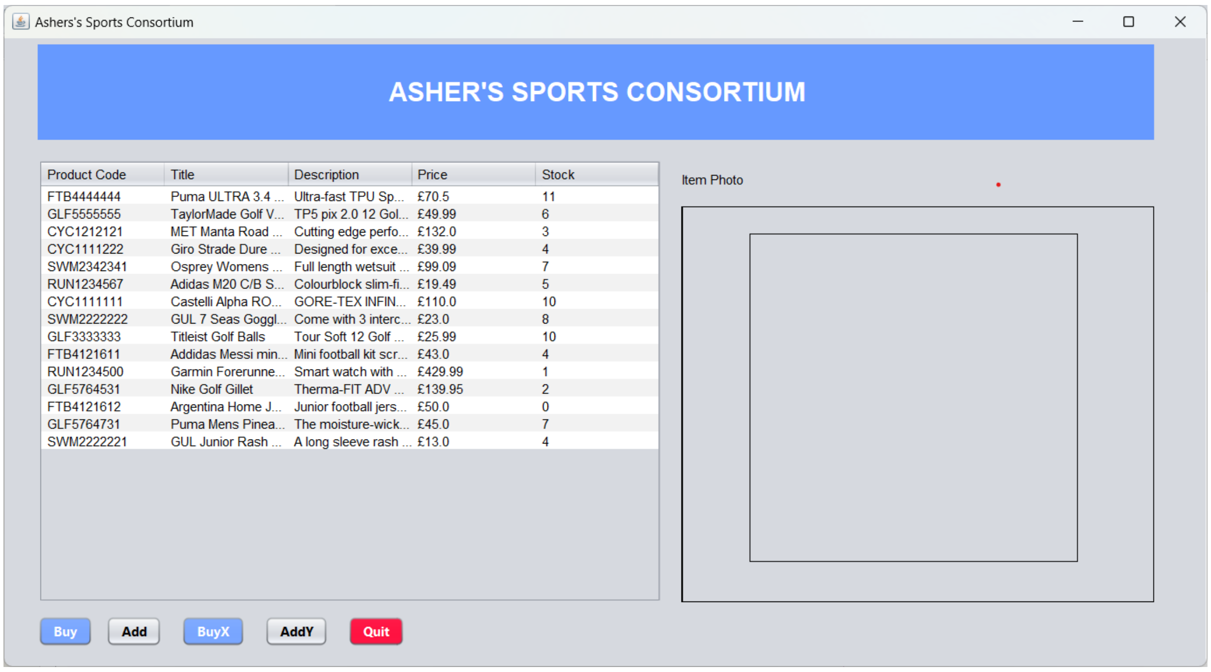
```
// table model is created with table columns - column Names and ArrayList - stockItemsList
ascModel = new ASCTableModel (columnNames , stockItemsList);

// ascmodel is applied to the JTable
ascStockItem.setModel(ascModel);


for (int col = 0; col < ascStockItem.getColumnCount(); col++) {
    // current column
    TableColumn column = ascStockItem.getTableHeader().getColumnModel().getColumn(col);

    //set column header
    column.setHeaderValue(columnNames[col]);
}
```

Figure 4: Setting abstract model to JTable

In the main class, a table model named ascModel is created. This model has set to the JTable ascStockItem that is designed in GUI. By looping through the column array the table headers are being set.

1.5 Testing

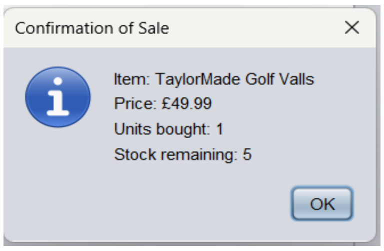
- The JTable is populated with data of AsherSportsConsortium3.csv.



Product Code	Title	Description	Price	Stock
FTB4444444	Puma ULTRA 3.4 ...	Ultra-fast TPU Sp...	£70.5	11
GLF5555555	TaylorMade Golf V...	TP5 pix 2.0 12 Gol...	£49.99	6
CYC1212121	MET Manta Road ...	Cutting edge perfo...	£132.0	3
CYC1111222	Giro Strade Dure ...	Designed for exce...	£39.99	4
SWM2342341	Osprey Womens ...	Full length wetsuit ...	£99.09	7
RUN1234567	Adidas M20 C/B S...	Colourblock slim-fi...	£19.49	5
CYC1111111	Castelli Alpha RO...	GORE-TEX INFIN...	£110.0	10
SWM2222222	GUL 7 Seas Goggl...	Come with 3 interc...	£23.0	8
GLF3333333	Titleist Golf Balls	Tour Soft 12 Golf ...	£25.99	10
FTB4121611	Addidas Messi min...	Mini football kit scr...	£43.0	4
RUN1234500	Garmin Forerunne...	Smart watch with ...	£429.99	1
GLF5764531	Nike Golf Gillet	Therma-FIT ADV ...	£139.95	2
FTB4121612	Argentina Home J...	Junior football jers...	£50.0	0
GLF5764731	Puma Mens Pinea...	The moisture-wick...	£45.0	7
SWM2222221	GUL Junior Rash ...	A long sleeve rash ...	£13.0	4

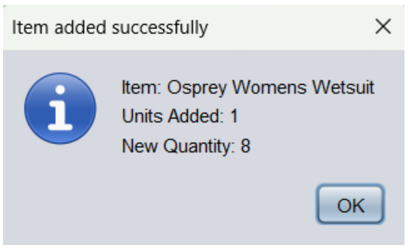
Figure 5: JTable with data

- The button functionality for Add, Buy and Quit button is also added. Screenshots are attached below by performing all the operations.


Confirmation of Sale

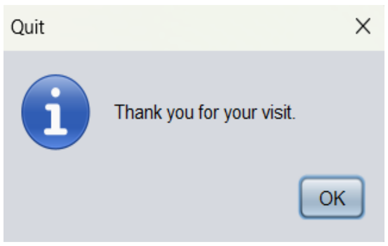
Item: TaylorMade Golf Valls
Price: £49.99
Units bought: 1
Stock remaining: 5

Figure 6: Sale Confirmation


Item added successfully

Item: Osprey Womens Wetsuit
Units Added: 1
New Quantity: 8

Figure 7: Added Stock Confirmation


Quit

Thank you for your visit.

Figure 8: Quit Confirmation

2. Discussion of Enhanced Application

2.1 Overview

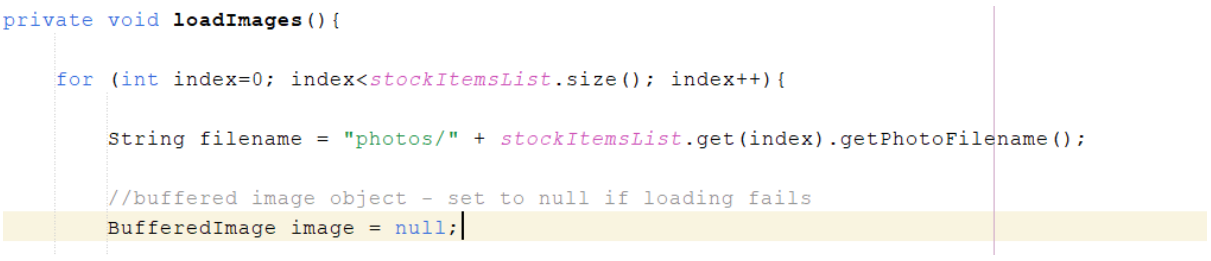
After developing the GUI, enhancements need to be done to include more features to the application. The overview of the improvements is:

- Images needs to be displayed in the photoLabel.
- Low stock Items should be notified if there are less than 5 units left for an item.
- With the help of buyX button and addY buttons, a feature needs to be enabled where multiple items can be bought and added to stock respectively.

2.2 Displaying an Image

Loading Images:

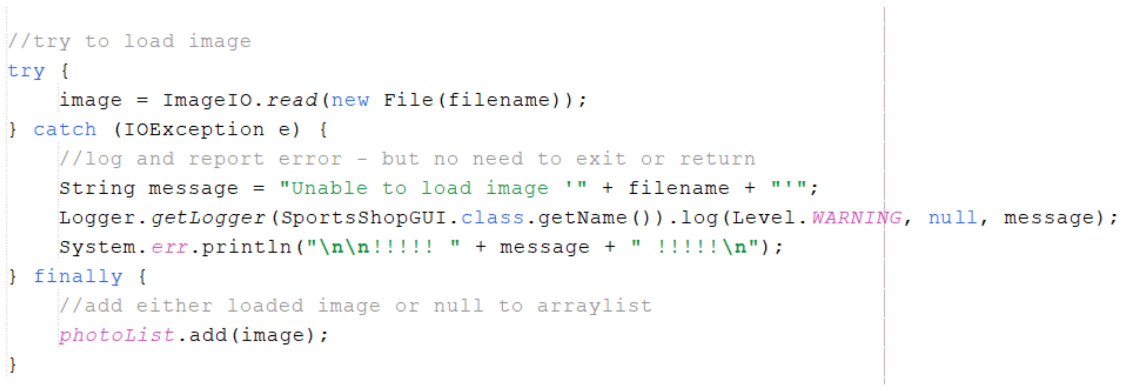
A method named loadData() is created, which preloads the images. The preloading of images helps the images to loader faster in the application. Initially, all the photos are placed in folder and that location is used.



```
private void loadImages(){

    for (int index=0; index<stockItemsList.size(); index++){

        String filename = "photos/" + stockItemsList.get(index).getPhotoFilename();

        //buffered image object - set to null if loading fails
        BufferedImage image = null;
```

Figure 9: Load image method

The method is started by looping through the length of arrayList. Each file name is same as the product Id with jpg extension. The photo path is loaded into filename and then loaded into an arrayList.



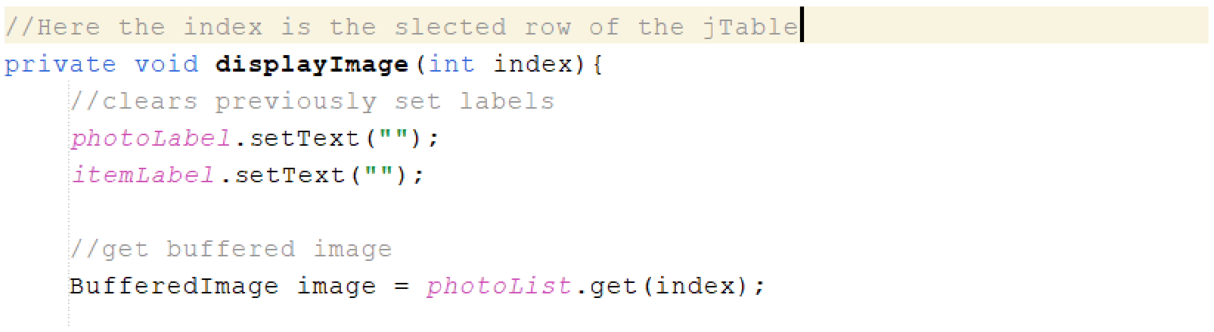
```
//try to load image
try {
    image = ImageIO.read(new File(filename));
} catch (IOException e) {
    //log and report error - but no need to exit or return
    String message = "Unable to load image '" + filename + "'";
    Logger.getLogger(SportsShopGUI.class.getName()).log(Level.WARNING, null, message);
    System.err.println("\n\n!!!!! " + message + " !!!!!\n");
} finally {
    //add either loaded image or null to arraylist
    photoList.add(image);
}
```

Figure 10: Adding photos to photoList

The first block will try to read the images and for any exceptions caught, simply the logger reports the error. Finally, all the images will be loaded to photoList array.

Display Method:

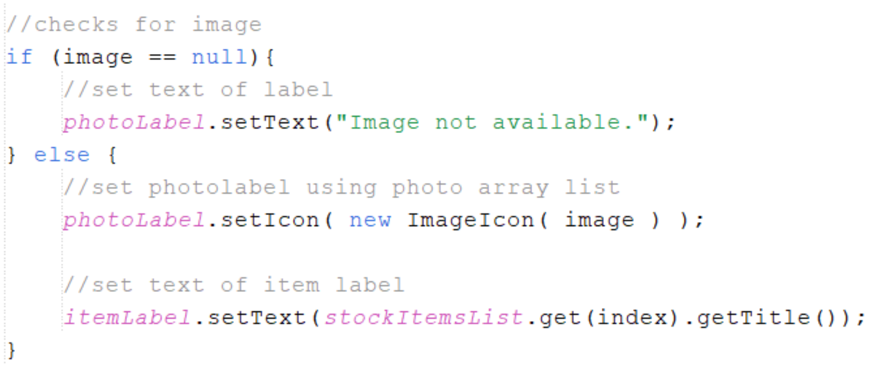
The display image method is called in listener event. The parameter that passed is the current selected row in the JTable. The photo and item labels are cleared first with setText. Then the image is selected from photoList.



```
//Here the index is the slected row of the jTable
private void displayImage(int index){
    //clears previously set labels
    photoLabel.setText("");
    itemLabel.setText("");

    //get buffered image
    BufferedImage image = photoList.get(index);
```

Figure 11: Displaying Image

The image and text labels are set accordingly If the image is not null, the same will be displayed or else image is not available is displayed.



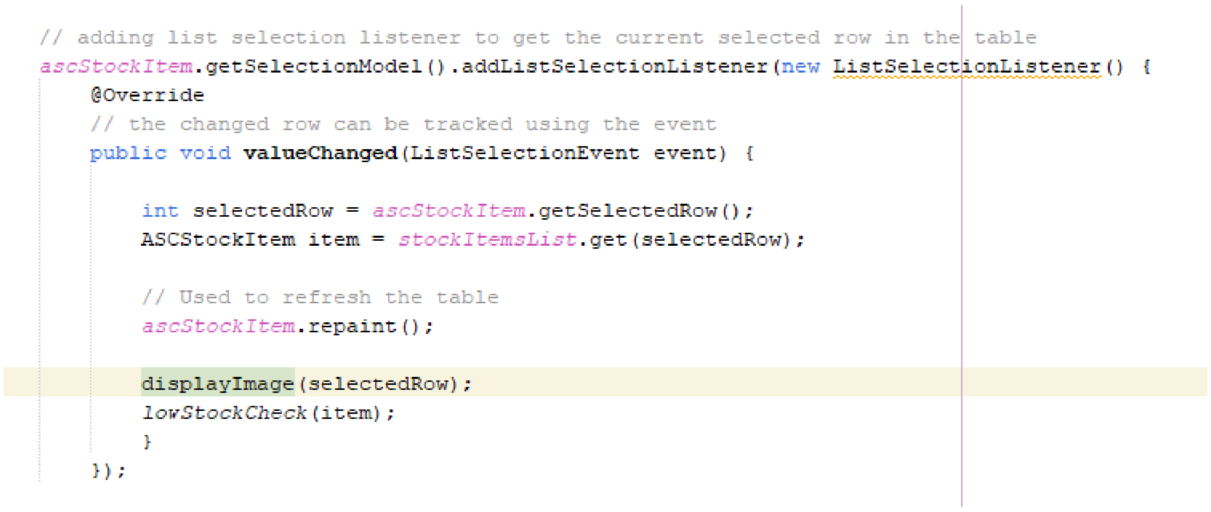
```java
//checks for image
if (image == null){
    //set text of label
    photoLabel.setText("Image not available.");
} else {
    //set photolabel using photo array list
    photoLabel.setIcon( new ImageIcon( image ) );

    //set text of item label
    itemLabel.setText(stockItemsList.get(index).getTitle());
}
```

Figure 12: Adding text to photo and Item Labels

List Event Listener

To display images of any selected row, firstly the event called “list selection listener” should be implemented. This listener event is used when the value of a table or list is modified.

As shown in the figure, getSelectionModel method is used to add this listener to the table. If the selection gets changed the row can be selected and then with the help of the help of displayImage () method, the image is displayed.

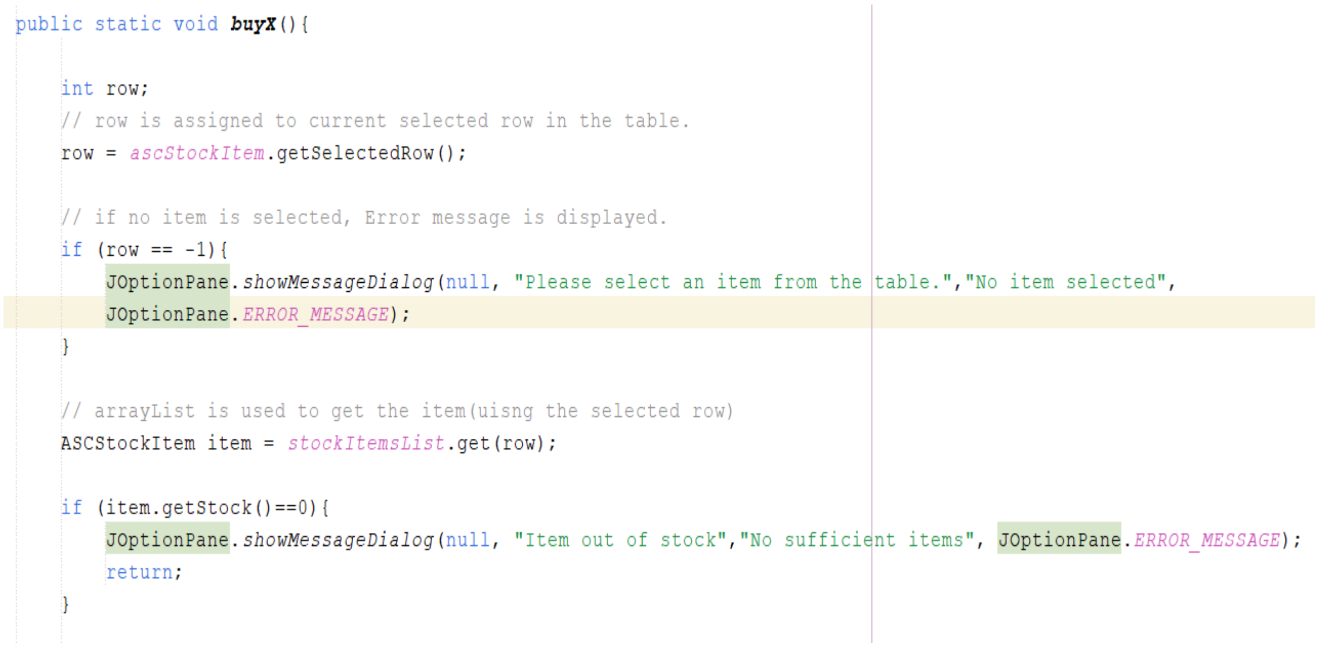


```java
// adding list selection listener to get the current selected row in the table
ascStockItem.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
    @Override
    // the changed row can be tracked using the event
    public void valueChanged(ListSelectionEvent event) {

        int selectedRow = ascStockItem.getSelectedRow();
        ASCStockItem item = stockItemsList.get(selectedRow);

        // Used to refresh the table
        ascStockItem.repaint();

        displayImage(selectedRow);
        lowStockCheck(item);
        }
    });
```

Figure 13: List selection listener

2.3 Adding and buying multiple items.

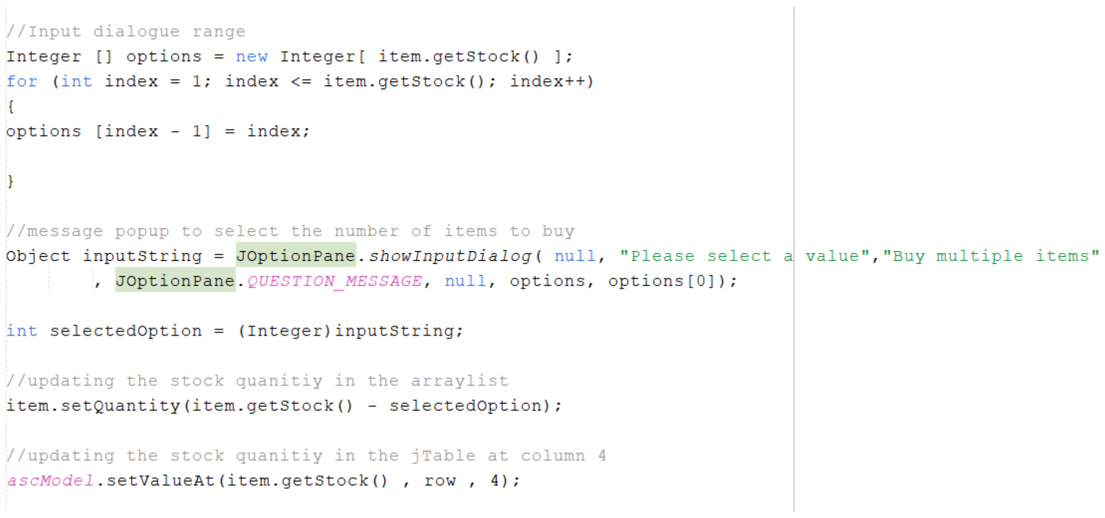
The features like adding and buying multiple items can be achieved by methods buyX and addY. Most of the code is same for both adding and buying items. The only change is the stock is decremented in buyX() and the stock is added in addY().



```
public static void buyX(){

    int row;
    // row is assigned to current selected row in the table.
    row = ascStockItem.getSelectedRow();

    // if no item is selected, Error message is displayed.
    if (row == -1){
        JOptionPane.showMessageDialog(null, "Please select an item from the table.","No item selected",
        JOptionPane.ERROR_MESSAGE);
    }

    // arrayList is used to get the item(uisng the selected row)
    ASCStockItem item = stockItemsList.get(row);

    if (item.getStock()==0){
        JOptionPane.showMessageDialog(null, "Item out of stock","No sufficient items", JOptionPane.ERROR_MESSAGE);
        return;
    }
```

Figure 14: BuyX method

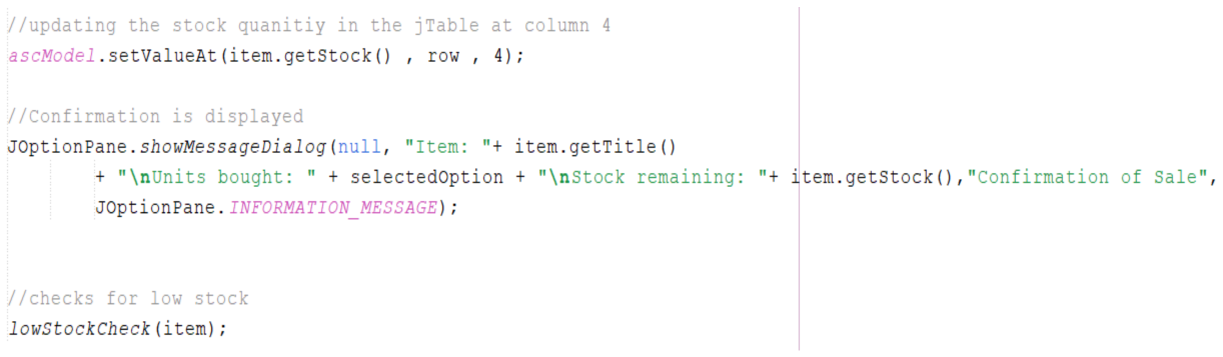
If the buyX button is pressed without selecting an item, an error message is displayed as shown in the above code. Next, if an item which has zero stock is bought popup notification is displayed “Item out of Stock”.



```
//Input dialogue range
Integer [] options = new Integer[ item.getStock() ];
for (int index = 1; index <= item.getStock(); index++)
{
options [index - 1] = index;

}

//message popup to select the number of items to buy
Object inputString = JOptionPane.showInputDialog( null, "Please select a value","Buy multiple items"
        , JOptionPane.QUESTION_MESSAGE, null, options, options[0]);

int selectedOption = (Integer)inputString;

//updating the stock quanitiy in the arraylist
item.setQuantity(item.getStock() - selectedOption);

//updating the stock quanitiy in the jTable at column 4
ascModel.setValueAt(item.getStock() , row , 4);
```

Figure 15: Updating arrayList and JTable

In the above figure, Initially the range of values is limited to length of arrayList. Next, popup is displayed using the JOptionPane and the option is selected. The selected option is decremented, and the value is set to table and arrayList.



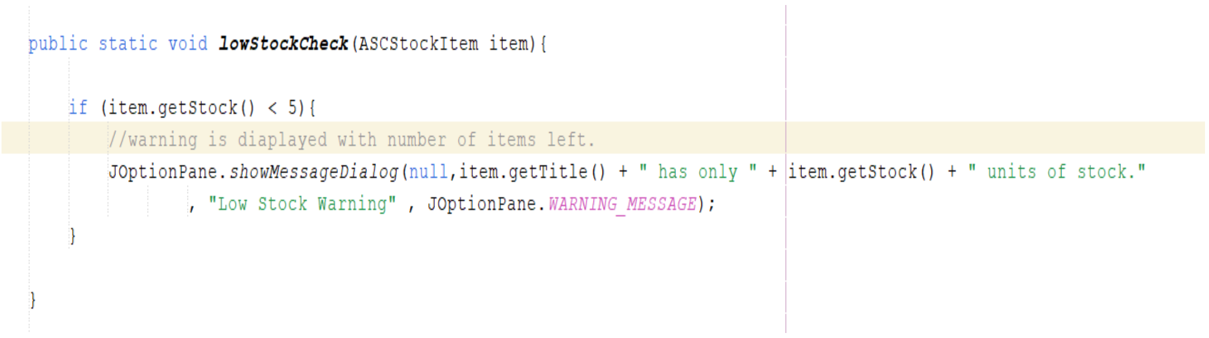
```
//updating the stock quanitiy in the jTable at column 4
ascModel.setValueAt(item.getStock() , row , 4);

//Confirmation is displayed
JOptionPane.showMessageDialog(null, "Item: "+ item.getTitle()
        + "\nUnits bought: " + selectedOption + "\nStock remaining: "+ item.getStock(),"Confirmation of Sale",
        JOptionPane.INFORMATION_MESSAGE);


//checks for low stock
lowStockCheck(item);
```

Figure 16: Sale Confirmation

In the end, an information message is displayed about the confirmation of sale and low stock check is done.

Low Stock Check:

The stock is checked by using getStock() getter function. If the item stock falls less than 5 then low stock warning is displayed.

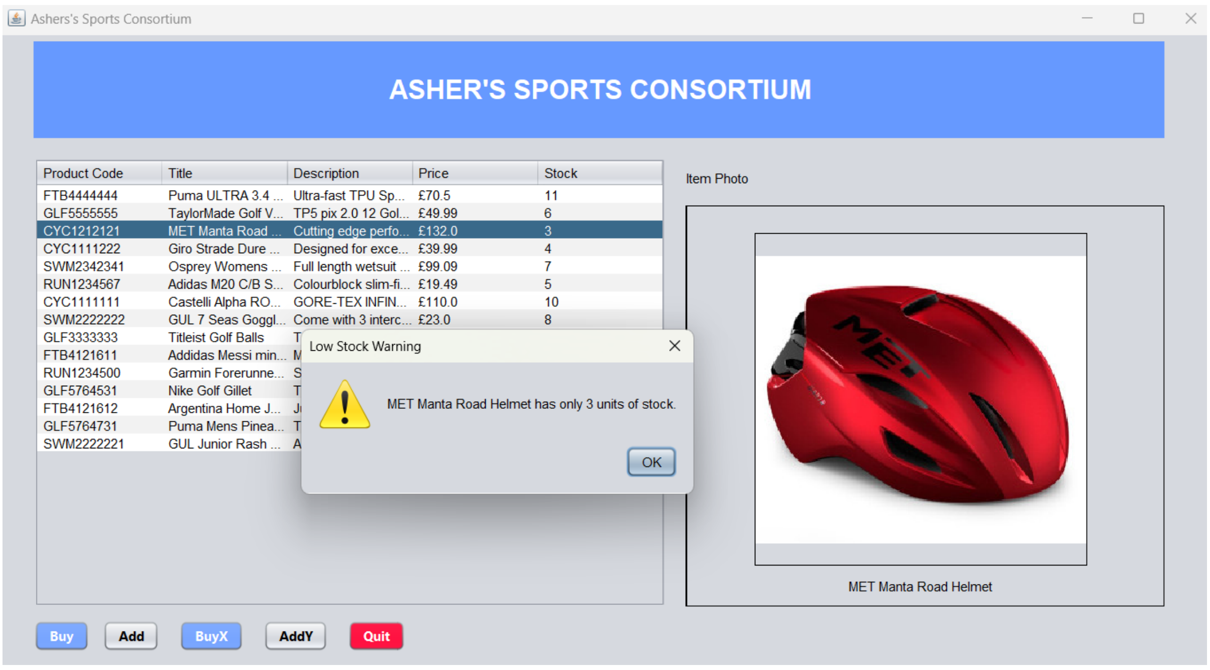


```
public static void lowStockCheck(ASCStockItem item){

    if (item.getStock() < 5){
        //warning is diaplayed with number of items left.
        JOptionPane.showMessageDialog(null,item.getTitle() + " has only " + item.getStock() + " units of stock."
                , "Low Stock Warning" , JOptionPane.WARNING_MESSAGE);
    }


}
```

Figure 17: Low stock check

2.4 Testing

- The image is displayed when a row is selected, if that item stock is less, warning message will also be displayed.



Product Code	Title	Description	Price	Stock
FTB4444444	Puma ULTRA 3.4 ...	Ultra-fast TPU Sp...	£70.5	11
GLF5555555	TaylorMade Golf V...	TP5 pix 2.0 12 Gol...	£49.99	6
CYC1212121	MET Manta Road ...	Cutting edge perfo...	£132.0	3
CYC1111222	Giro Strade Dure ...	Designed for exce...	£39.99	4
SWM2342341	Osprey Womens ...	Full length wetsuit ...	£99.09	7
RUN1234567	Adidas M20 C/B S...	Colourblock slim-fi...	£19.49	5
CYC1111111	Castelli Alpha RO...	GORE-TEX INFIN...	£110.0	10
SWM2222222	GUL 7 Seas Goggl...	Come with 3 interc...	£23.0	8
GLF3333333	Titleist Golf Balls	T		
FTB4121611	Addidas Messi min...	M		
RUN1234500	Garmin Forerunne...	S		
GLF5764531	Nike Golf Gillet	T		
FTB4121612	Argentina Home J...	J		
GLF5764731	Puma Mens Pinea...	T		
SWM2222221	GUL Junior Rash ...	A		

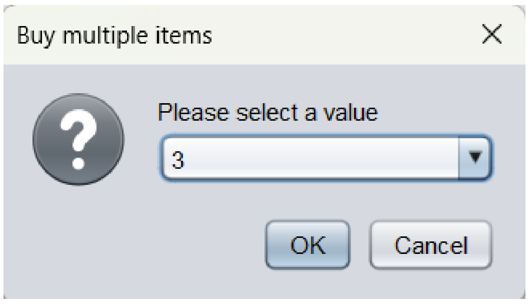
Low Stock Warning

MET Manta Road Helmet has only 3 units of stock.

OK

Figure 18: Image and warning are displayed.

- Using buyX button displays a popup message with number of items to buy. After buying information message is displayed.

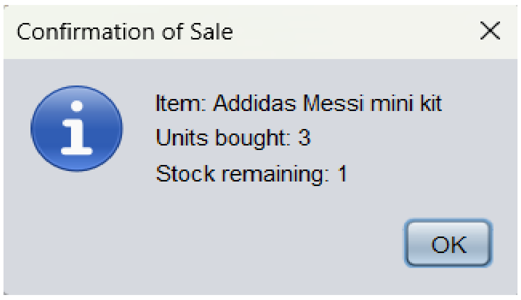

Buy multiple items

Please select a value

3

OK Cancel

Figure 19: Buy multiple items.

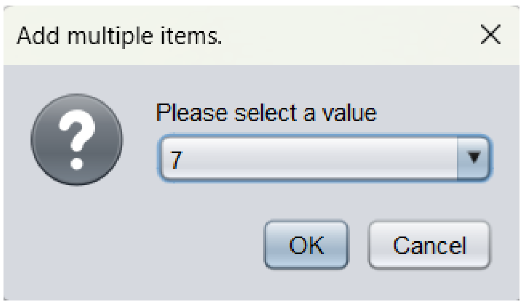

Confirmation of Sale

Item: Addidas Messi mini kit
Units bought: 3
Stock remaining: 1

OK

Figure 20: Sale Confirmation

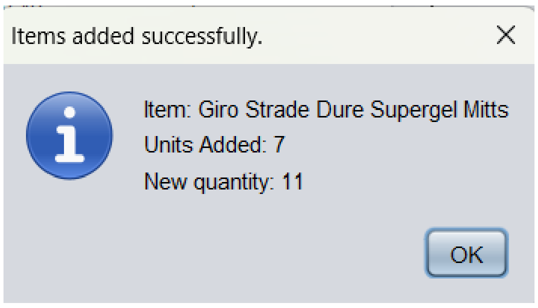
- Adding stock


Add multiple items.

Please select a value

7

Figure 21: Add multiple items.


Items added successfully.

Item: Giro Strade Dure Supergel Mitts
Units Added: 7
New quantity: 11

Figure 22: Added Stock Confirmation

3. Discussion of merged application

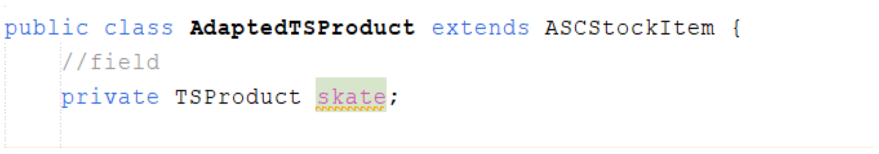
3.1 Overview

- Asher Sport's Consortium needs to be integrated with Teesside Skates which is a basic application that only have basic features.
- Program needs to be modified in such a way that it works for both asc and Teesside Skate products.

3.2 Adapter Pattern

A design pattern needs to be used for solving the problem of integrating the two different classes. Adapter pattern will be used in this merge. This pattern is a structural pattern which combines two incompatible classes.

Primarily, the TSProduct.java class needs to be refactor copied to the ASCSystem project. Then, Intermediary class is created, it is AdaptedTSProduct which extends to ASCStockITem as shown in the below figure.



```
public class AdaptedTSProduct extends ASCStockItem {
    //field
    private TSProduct skate;
```

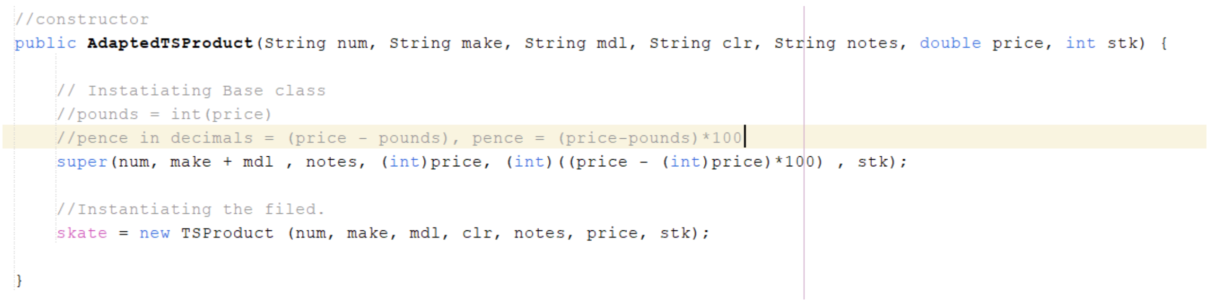
Figure 23: AdaptedTSProduct class

For the constructor all the parameters of TSProducts.java must be present in this as well. The base class needs to be initialized first. The parameters can be matched in this case, if all the parameters cannot be matched, override methods also can be used.

Table 3: ASC Parameters in terms of TS Parameters

ASCStockItem parameters match	**TSProduct parameters**
Product Id	num
Title	Make + mdl
Description	notes
Price in Pounds	(int) price
Price in pence	(Int) (price – (int)price) * 100

The super class can be initiated with the values as shown in the above table. The implementation can be seen in below figure. The tricky part of code is to convert the price (float) into pounds(int) and pence(int). The pounds will be equal to integer part of price. For pence, the pounds can be subtracted and multiplied by 100.



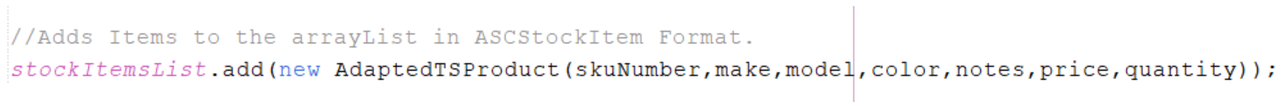
```
//constructor
public AdaptedTSProduct(String num, String make, String mdl, String clr, String notes, double price, int stk) {

    // Instatiating Base class
    //pounds = int(price)
    //pence in decimals = (price - pounds), pence = (price-pounds)*100
    super(num, make + mdl , notes, (int)price, (int)((price - (int)price)*100) , stk);

    //Instantiating the filed.
    skate = new TSProduct (num, make, mdl, clr, notes, price, stk);

}
```

Figure 24: Initialization of base class

The skate product is similar to TSProduct. All parameters which are in ascSystem are defined. Therefore, overriding methods is not necessary. AdapterTSProduct is ready to apply.

Main class code:

The loading of data uses two classes, one for ASC data and other for TS data, the only change that needed is when trying to load items into the arraylist, AdaptedTSProduct should be used.



```
//Adds Items to the arrayList in ASCStockItem Format.
stockItemsList.add(new AdaptedTSProduct(skuNumber,make,model,color,notes,price,quantity));
```

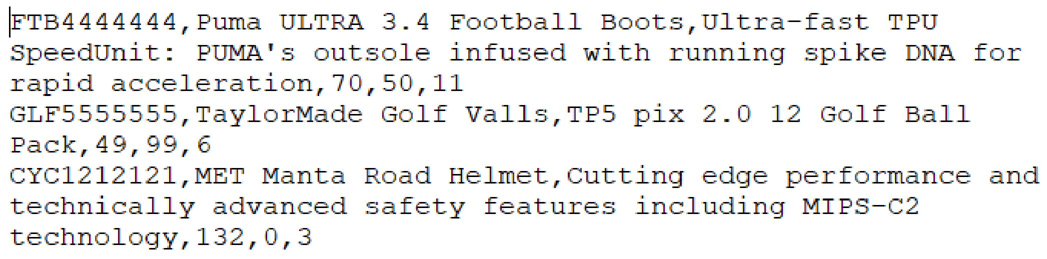
Figure 25: Adding items to arrayList using Adapted Class

3.3 Testing

- The ts_products.txt and AsherSportsConsortium3.csv data is loaded into the table and images should be displayed.

Input data in files:

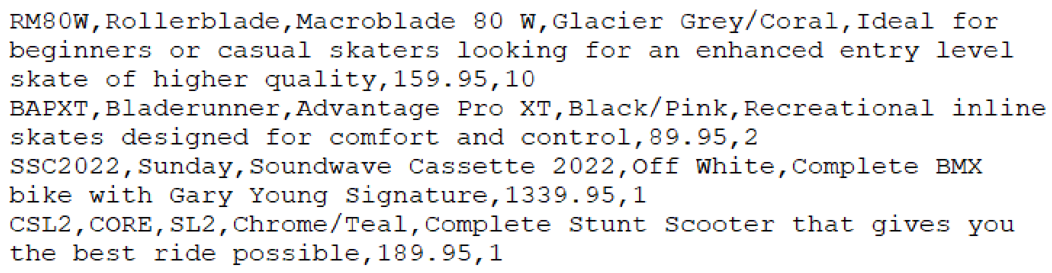
ASC Data



```
FTB4444444,Puma ULTRA 3.4 Football Boots,Ultra-fast TPU
SpeedUnit: PUMA's outsole infused with running spike DNA for
rapid acceleration,70,50,11
GLF5555555,TaylorMade Golf Valls,TP5 pix 2.0 12 Golf Ball
Pack,49,99,6
CYC1212121,MET Manta Road Helmet,Cutting edge performance and
technically advanced safety features including MIPS-C2
technology,132,0,3
```

Figure 26: ASC Data

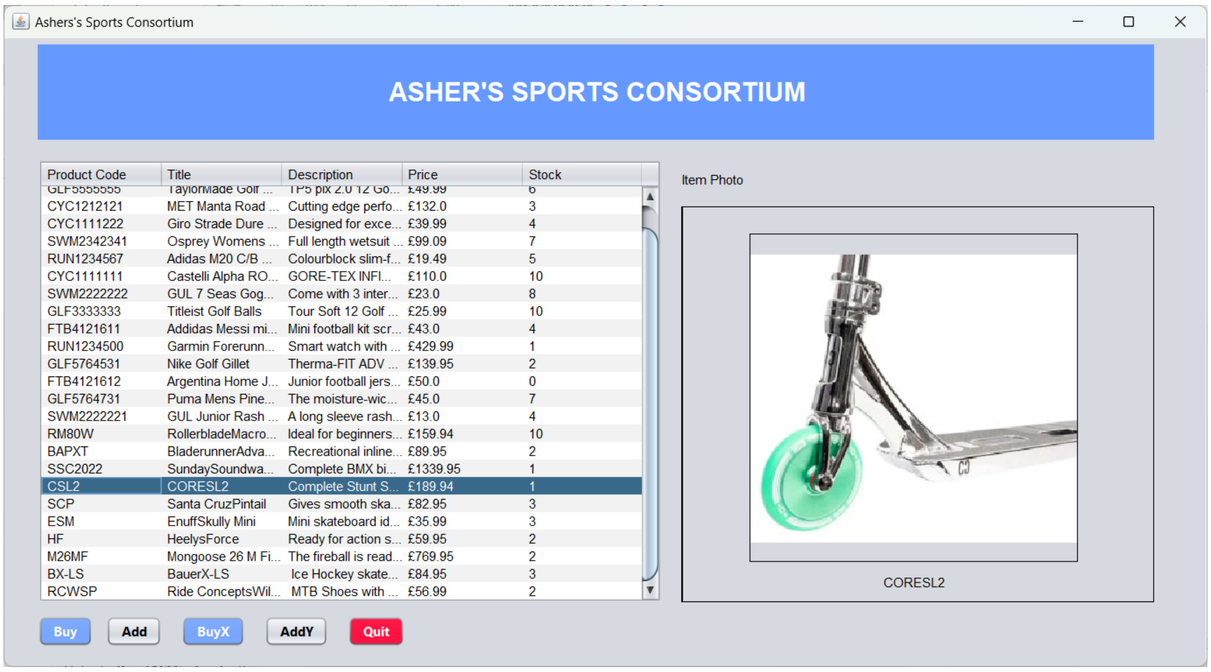
TS Data



```
RM80W,Rollerblade,Macroblade 80 W,Glacier Grey/Coral,Ideal for
beginners or casual skaters looking for an enhanced entry level
skate of higher quality,159.95,10
BAPXT,Bladerunner,Advantage Pro XT,Black/Pink,Recreational inline
skates designed for comfort and control,89.95,2
SSC2022,Sunday,Soundwave Cassette 2022,Off White,Complete BMX
bike with Gary Young Signature,1339.95,1
CSL2,CORE,SL2,Chrome/Teal,Complete Stunt Scooter that gives you
the best ride possible,189.95,1
```

Figure 27: TS Data

The combined data files loaded in GUI.



Product Code	Title	Description	Price	Stock
GLF5555555	TaylorMade Golf ...	TP5 pix 2.0 12 Go...	£49.99	6
CYC1212121	MET Manta Road ...	Cutting edge perfo...	£132.0	3
CYC1111222	Giro Strade Dure ...	Designed for exce...	£39.99	4
SWM2342341	Osprey Womens ...	Full length wetsuit ...	£99.09	7
RUN1234567	Adidas M20 C/B ...	Colourblock slim-f...	£19.49	5
CYC1111111	Castelli Alpha RO...	GORE-TEX INFI...	£110.0	10
SWM2222222	GUL 7 Seas Gog...	Come with 3 inter...	£23.0	8
GLF3333333	Titleist Golf Balls	Tour Soft 12 Golf ...	£25.99	10
FTB4121611	Addidas Messi mi...	Mini football kit scr...	£43.0	4
RUN1234500	Garmin Forerunn...	Smart watch with ...	£429.99	1
GLF5764531	Nike Golf Gillet	Therma-FIT ADV ...	£139.95	2
FTB4121612	Argentina Home J...	Junior football jers...	£50.0	0
GLF5764731	Puma Mens Pine...	The moisture-wic...	£45.0	7
SWM2222221	GUL Junior Rash ...	A long sleeve rash...	£13.0	4
RM80W	RollerbladeMacro...	Ideal for beginners...	£159.94	10
BAPXT	BladerunnerAdva...	Recreational inline...	£89.95	2
SSC2022	SundaySoundwa...	Complete BMX bi...	£1339.95	1
CSL2	CORESL2	Complete Stunt S...	£189.94	1
SCP	Santa CruzPintail	Gives smooth ska...	£82.95	3
ESM	EnuffSkully Mini	Mini skateboard id...	£35.99	3
HF	HeelysForce	Ready for action s...	£59.95	2
M26MF	Mongoose 26 M Fi...	The fireball is read...	£769.95	2
BX-LS	BauerX-LS	Ice Hockey skate...	£84.95	3
RCWSP	Ride ConceptsWil...	MTB Shoes with ...	£56.99	2

Figure 28: Merged Application

- The output file format does not change, only new entries from TS Products are added.

```
FTB4121611,Addidas Messi mini kit,Mini football kit screen printed with Messi Logo and made form 100% recycled materials,43,0,1
RUN1234500,Garmin Forerunner 265,Smart watch with built-in maps and a full-colour AMOLED touchscreen display,429,99,1
GLF5764531,Nike Golf Gillet,Therma-FIT ADV has advanced heat-regulating design that'll keep you warm on the golf course,139,95,2
FTB4121612,Argentina Home Jersey,Junior football jersey of 2022 World Cup winners,50,0,0
GLF5764731,Puma Mens Pineapples Golf Polo Shirt,The moisture-wicking fabric is perfect for a sunny day on the course; while the all-over
pineapple print will have you looking like a pro,45,0,7
SWM2222221,GUL Junior Rash Vest,A long sleeve rash vest which is lightweight and features a flat lock seams to prevent chafing,13,0,4
```

Figure 29: Output file

4. Discussion of Improvement

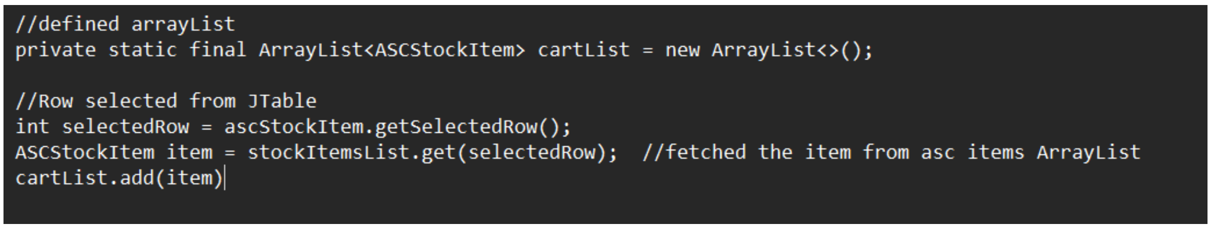
4.1 Overview

More features can be added to Asher Sports Consortium. The first improvement that comes to mind when thinking of improvements is cart function which enables users to add various items to cart and they can buy all the items at once.

4.2 Adding to Cart

A button will be created and placed at the bottom of JTable with name addToCart. When an item is selected, the item can be added to cart as many times as needed. Various items can also be selected and added to cart.

All the items in the cart can be bought at once. The hardest part is there should be track of all the items that are adding into the cart. An approach that can be used is arrayList named cartList. When the items are added to cart, they are automatically added to cartList.



```
//defined arrayList
private static final ArrayList<ASCStockItem> cartList = new ArrayList<>();

//Row selected from JTable
int selectedRow = ascStockItem.getSelectedRow();
ASCStockItem item = stockItemsList.get(selectedRow);  //fetched the item from asc items ArrayList
cartList.add(item)
```

Figure 30: Rough Code

After the cart is selected, when the items are bought the cartList will be used to update both arrayList of StockItemsList as well as JTable.

5. References:

Oracle Help Center. (n.d.). *JDK 20 Documentation*. [online] Available at: https://docs.oracle.com/javase/8/docs/api/javax/swing/table/AbstractTableModel.html. [Accessed 1 May 2023].

netbeans.apache.org. (n.d.). *Handling Images in a Java GUI Application*. [online] Available at: https://netbeans.apache.org/kb/docs/java/gui-image-display [Accessed 1 May 2023].

docs.oracle.com. (n.d.). *How to Write a List Selection Listener* [online] Available at: https://docs.oracle.com/javase/tutorial/uiswing/events/listselectionlistener.html [Accessed 1 May 2023].

www.tabnine.com. (n.d.). *ListSelectionListener java code examples | Tabnine*. [online] Available at: https://www.tabnine.com/code/java/classes/javax.swing.event.ListSelectionListener [Accessed 1 May 2023].