Informatics Institute of Technology

Department of Computing

Bsc(Hons) Artificial Intelligence and Data Science

Module: CM1601 Programming Fundamentals

Semester 2

Module Coordinator: Mr. Iresh Bandara

Individual Coursework Part 2 Report

Student Details :

Name: Vanuja Thihansith Sooriyaarachchi

RGU ID: 2311130

IIT ID: 20222408

Executive Summary

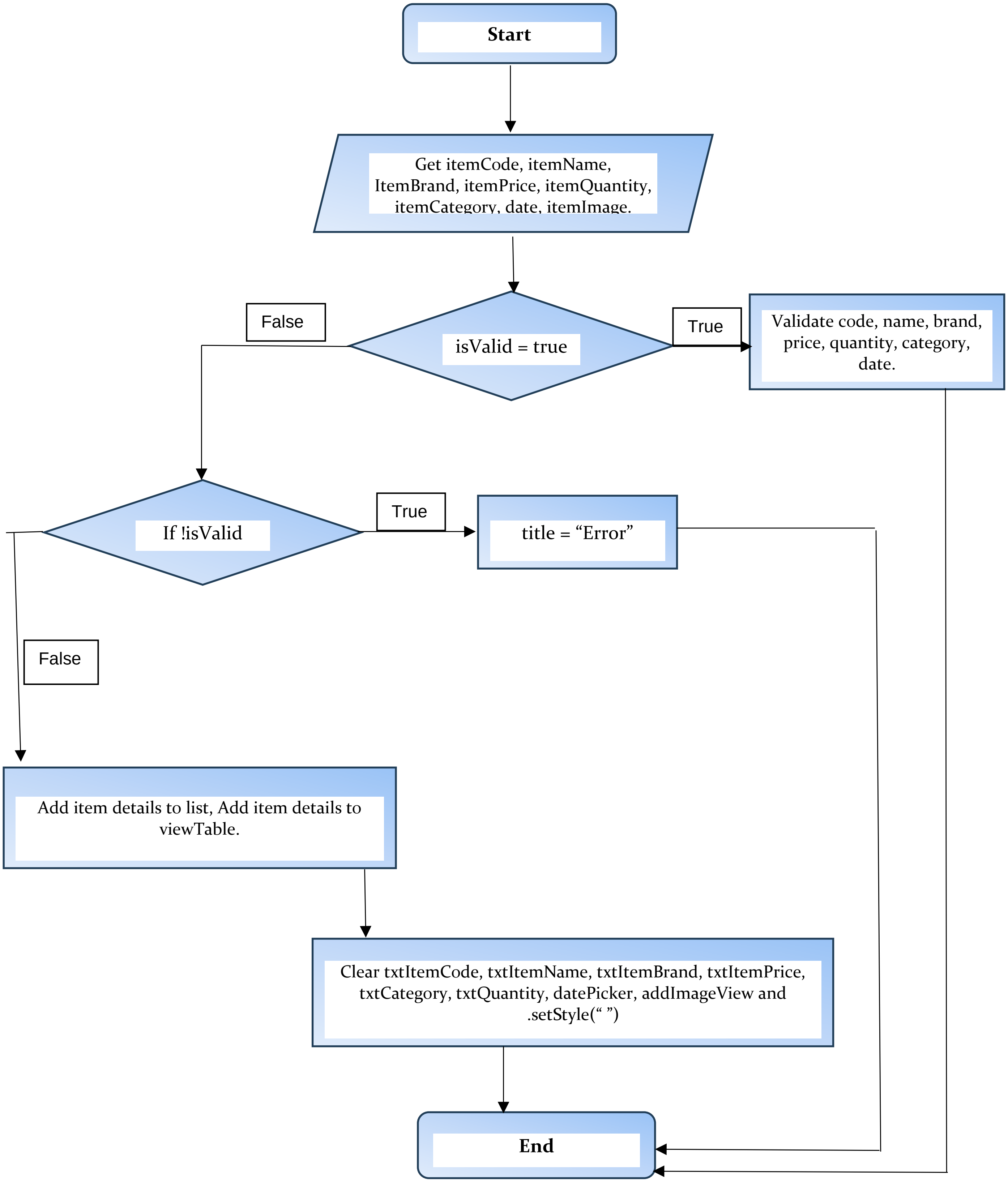
It is required to develop a GUI application for John’s Internet Café using Java and OOP concept. The application utilizes JavaFX for the user interface and allows users to add, update, and view items in a table format.

Overall, this code demonstrates a basic inventory management system with functionalities for adding, updating, and viewing items. It incorporates input validation, error handling, and data persistence through file I/O. However, additional features like item deletion, sorting, and filtering could further enhance the application's functionality and usability.

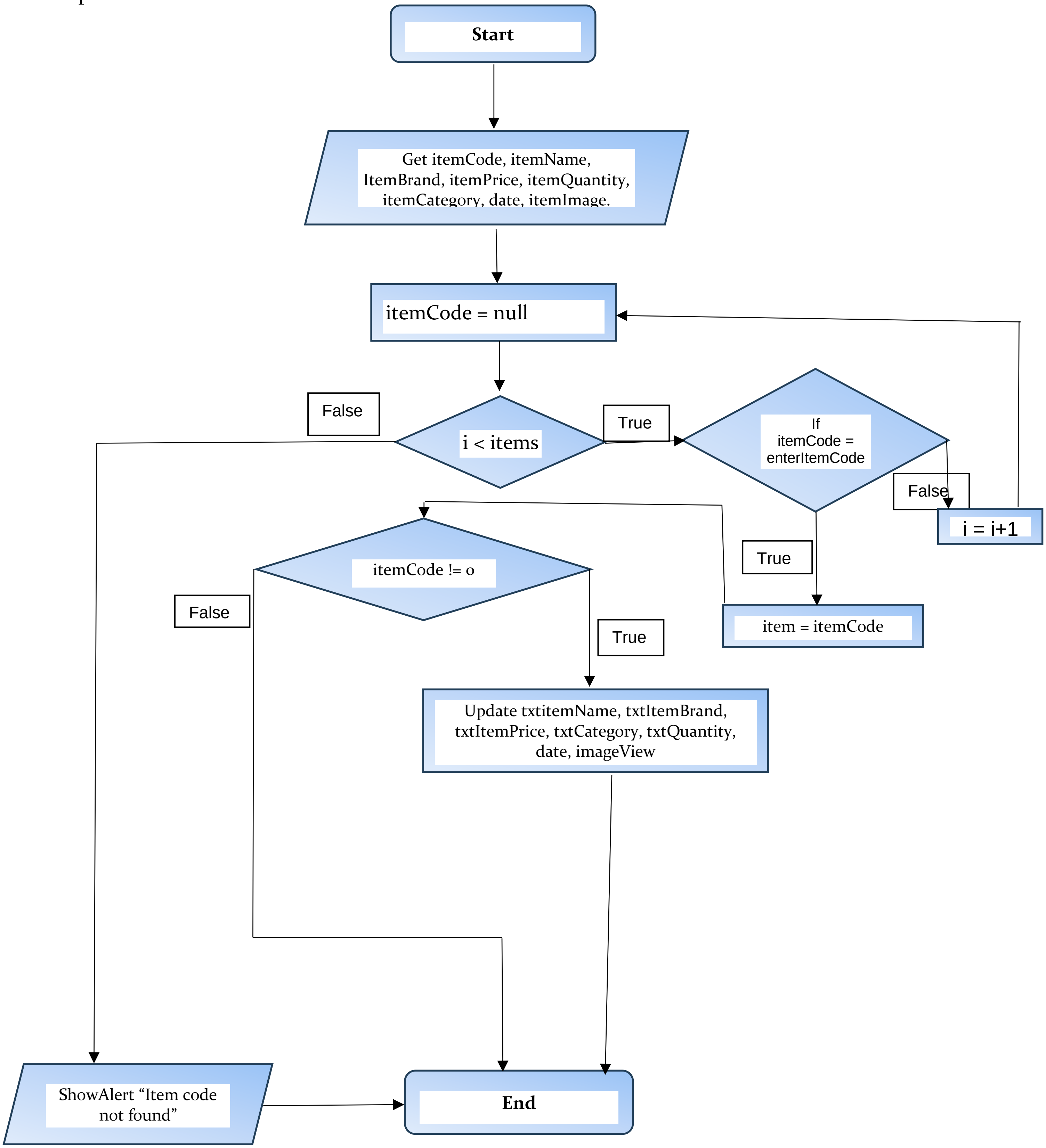
Contents

Flow Charts

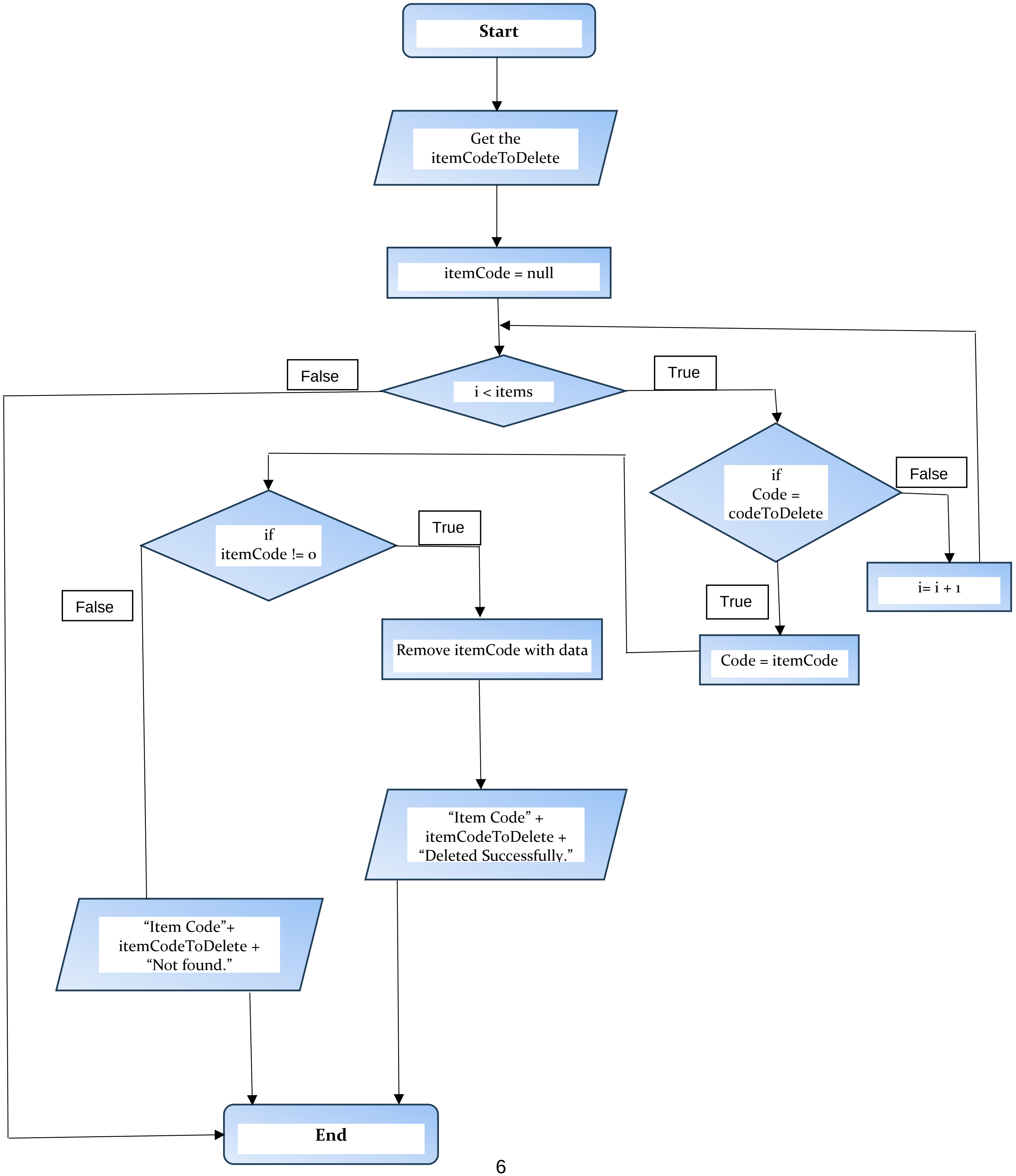
1. Add


Start
Get itemCode, itemName, ItemBrand, itemPrice, itemQuantity, itemCategory, date, itemImage.
False
isValid = true
True
Validate code, name, brand, price, quantity, category, date.
If !isValid
True
title = "Error"
False
Add item details to list, Add item details to viewTable.
Clear txtItemCode, txtItemName, txtItemBrand, txtItemPrice, txtCategory, txtQuantity, datePicker, addImageView and .setStyle(" ")
End

2. Update

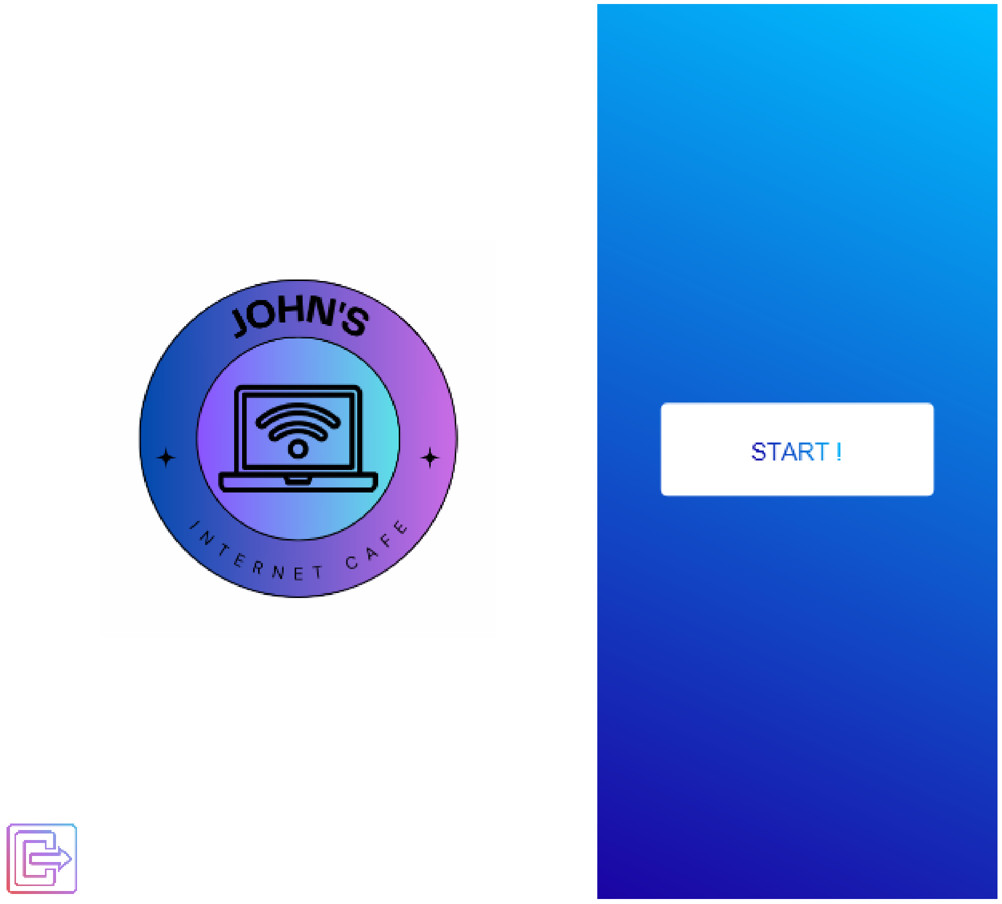

Start
Get itemCode, itemName, ItemBrand, itemPrice, itemQuantity, itemCategory, date, itemImage.
itemCode = null
False
i < items
True
If itemCode = enterItemCode
False
i = i+1
True
itemCode != 0
item = itemCode
False
True
Update txtitemName, txtItemBrand, txtItemPrice, txtCategory, txtQuantity, date, imageView
ShowAlert "Item code not found"
End

3. Delete


Start
Get the itemCodeToDelete
itemCode = null
False
i < items
True
if Code = codeToDelete
False
True
i= i + 1
Code = itemCode
if itemCode != 0
True
False
Remove itemCode with data
“Item Code” + itemCodeToDelete + “Deleted Successfully.”
“Item Code”+ itemCodeToDelete + “Not found.”
End

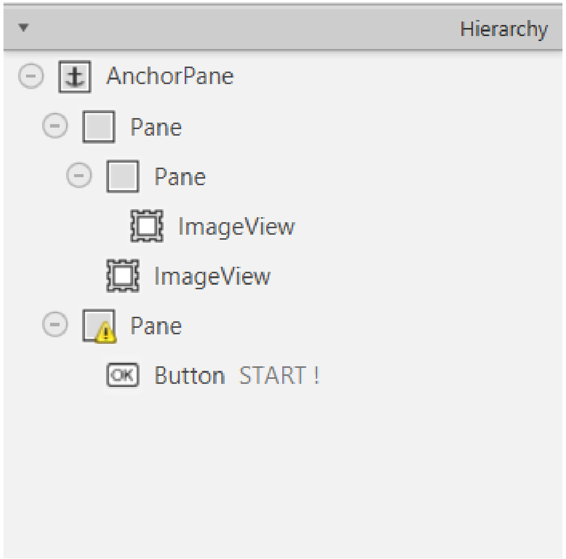
Introduction to function with code

1.Starting page



- ➢ User can see start page as above after launching the program.
- ➢ I have used a logo, icons and buttons on this start page.
- ➢ This start page has 2 buttons.
  - ▪ Start – Go to the user interface.
  - ▪ Exit – Exit from the program.

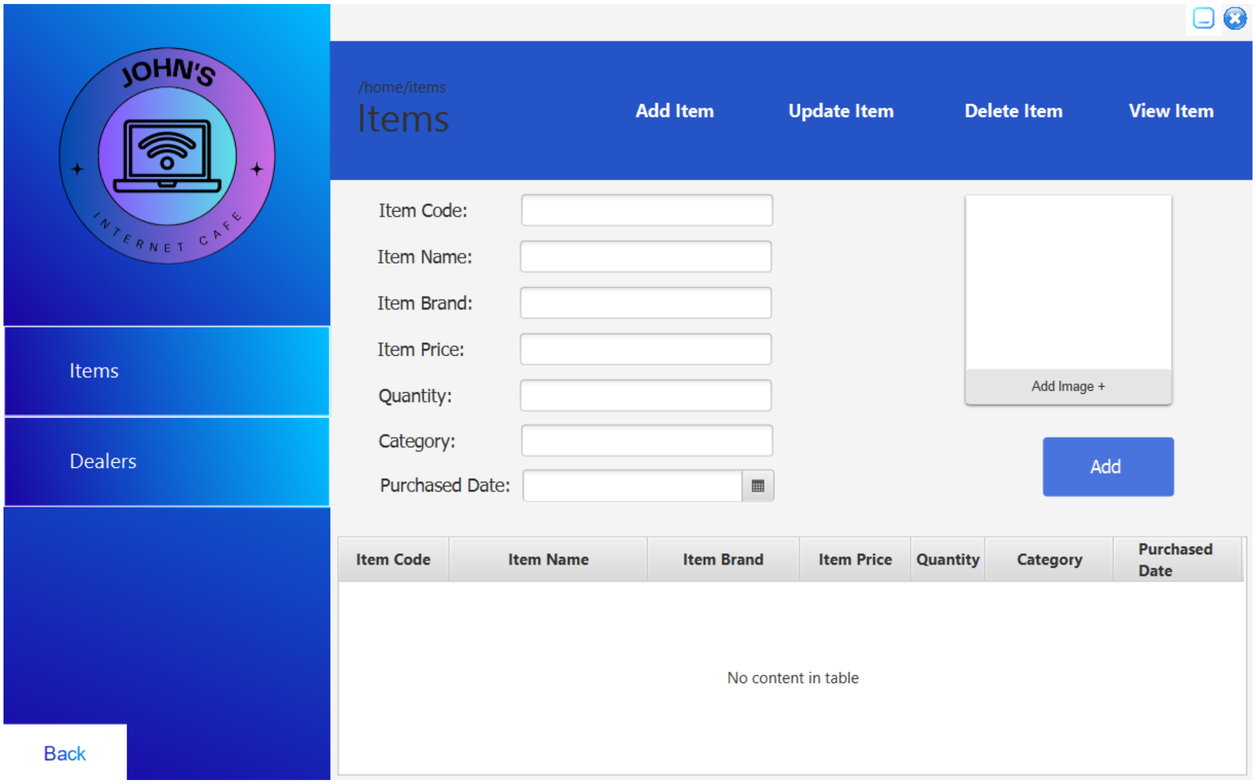
- SceneBuilder Hierarchy



Hierarchy

- AnchorPane
  - Pane
    - Pane
      - ImageView
    - ImageView
  - Pane
    - Button START !

2. Item section

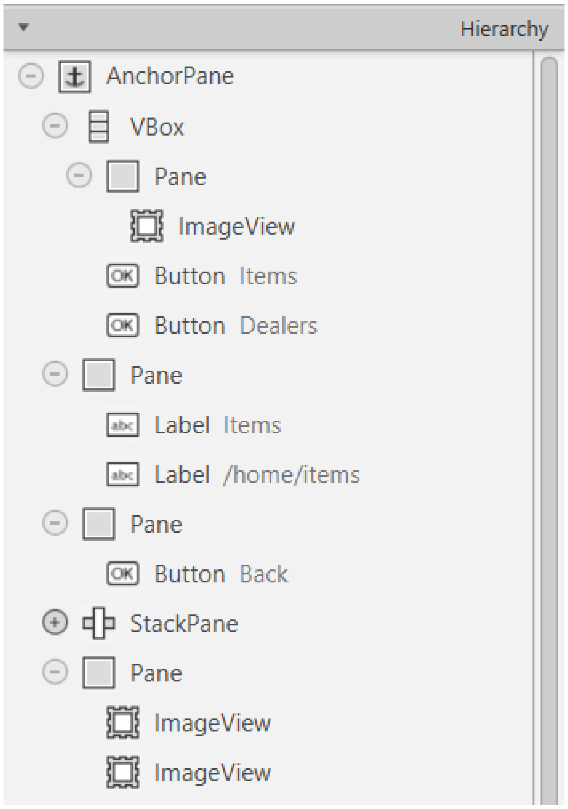


Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

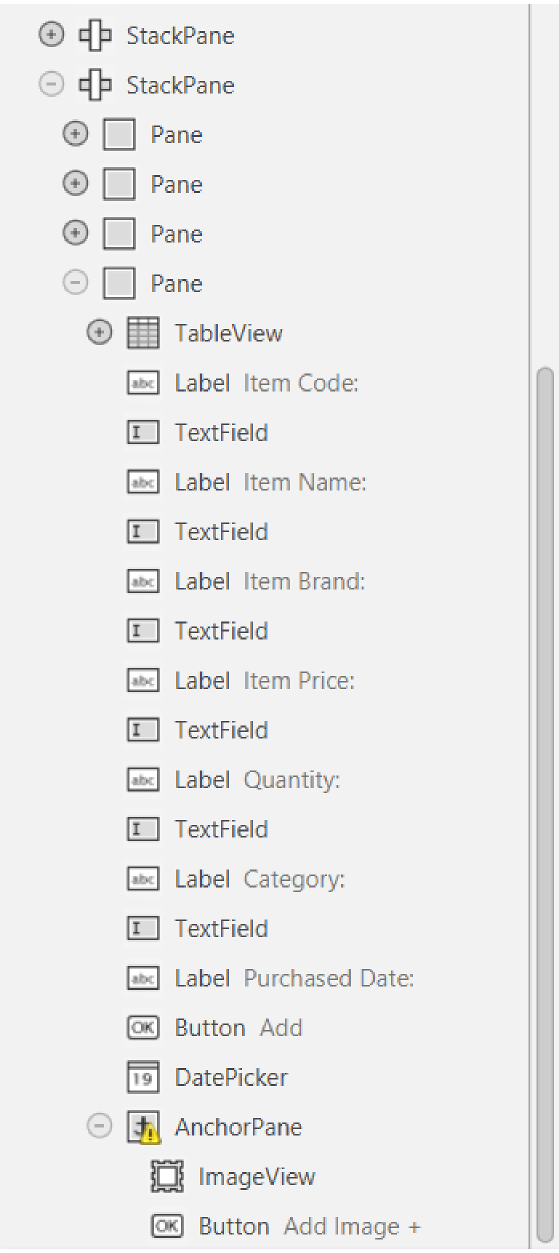
No content in table

- ➢ After clicking the Start button, the user enters the user interface. Users can see the Items section like this.
- ➢ Users can access the corresponding function from the buttons in the header.

SceneBuilder Hierarchy

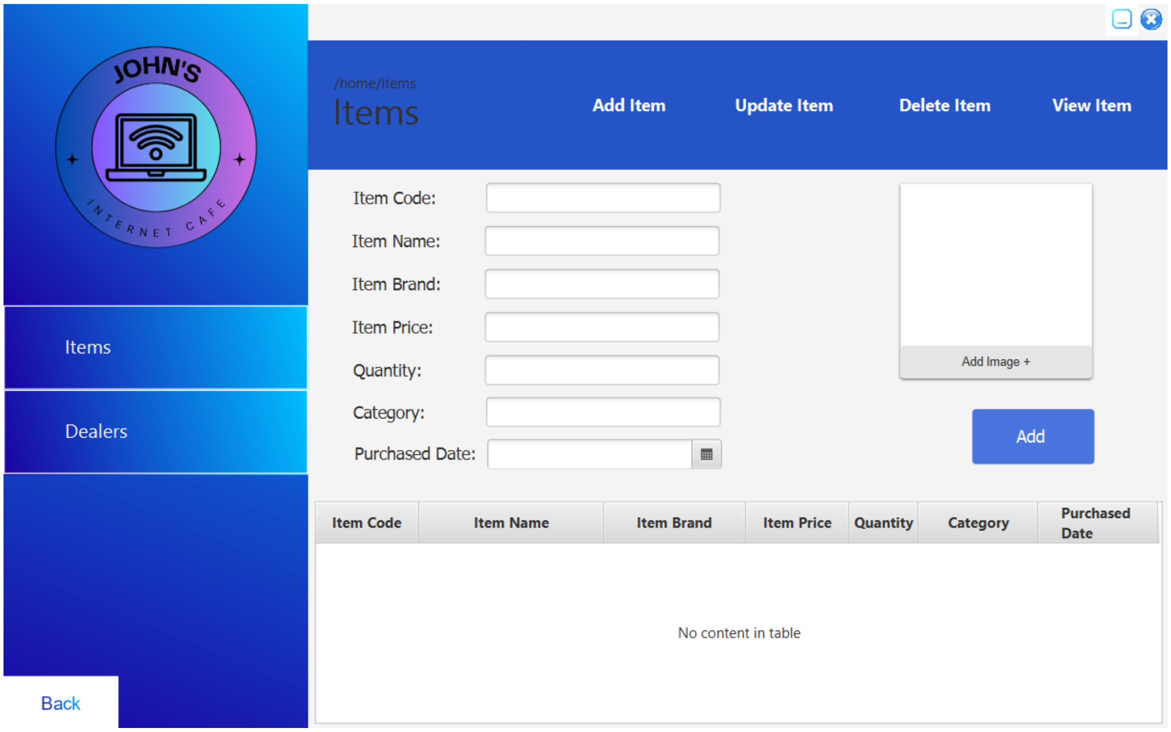


```
Hierarchy
AnchorPane
  VBox
    Pane
      ImageView
    Button  Items
    Button  Dealers
  Pane
    Label  Items
    Label  /home/items
  Pane
    Button  Back
  StackPane
  Pane
    ImageView
    ImageView
```



```
StackPane
StackPane
  Pane
  Pane
  Pane
  Pane
    TableView
    Label  Item Code:
    TextField
    Label  Item Name:
    TextField
    Label  Item Brand:
    TextField
    Label  Item Price:
    TextField
    Label  Quantity:
    TextField
    Label  Category:
    TextField
    Label  Purchased Date:
    Button  Add
    DatePicker
    AnchorPane
      ImageView
      Button  Add Image +
```

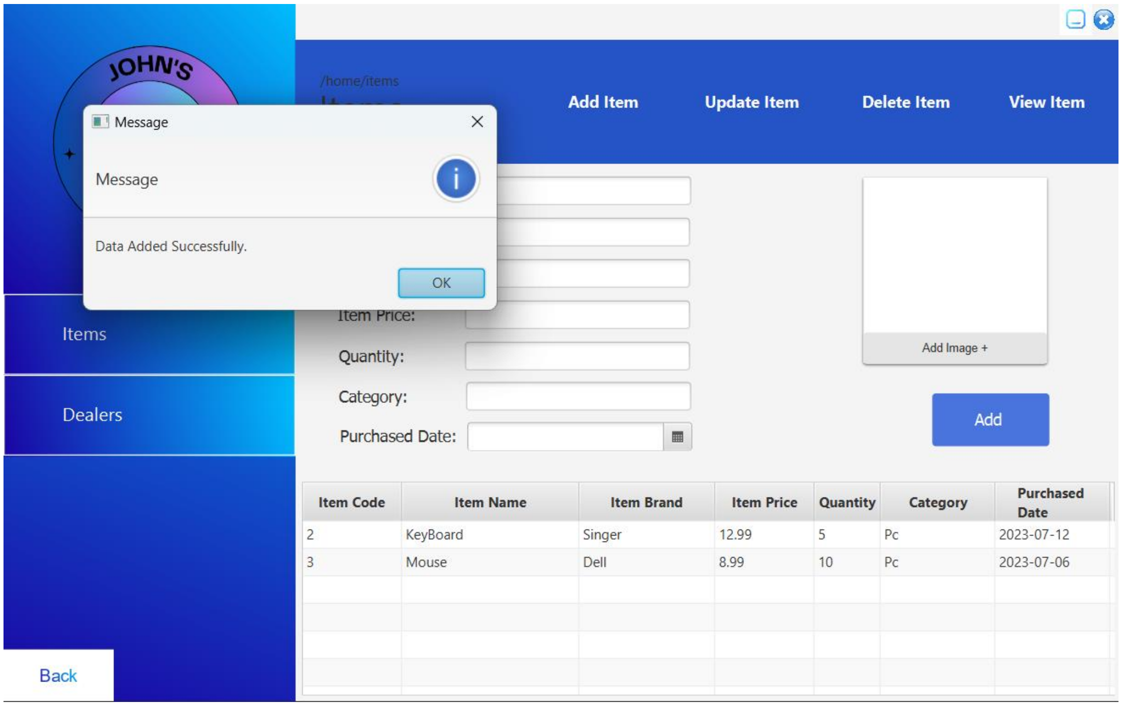
3.Add items



Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

No content in table

- ➢ When the user fills the text fields and clicks the "Add" button, the details will be added to the "saveAddItems.txt" file and shows the below table.
  - After adding the item, the system will display an alert.



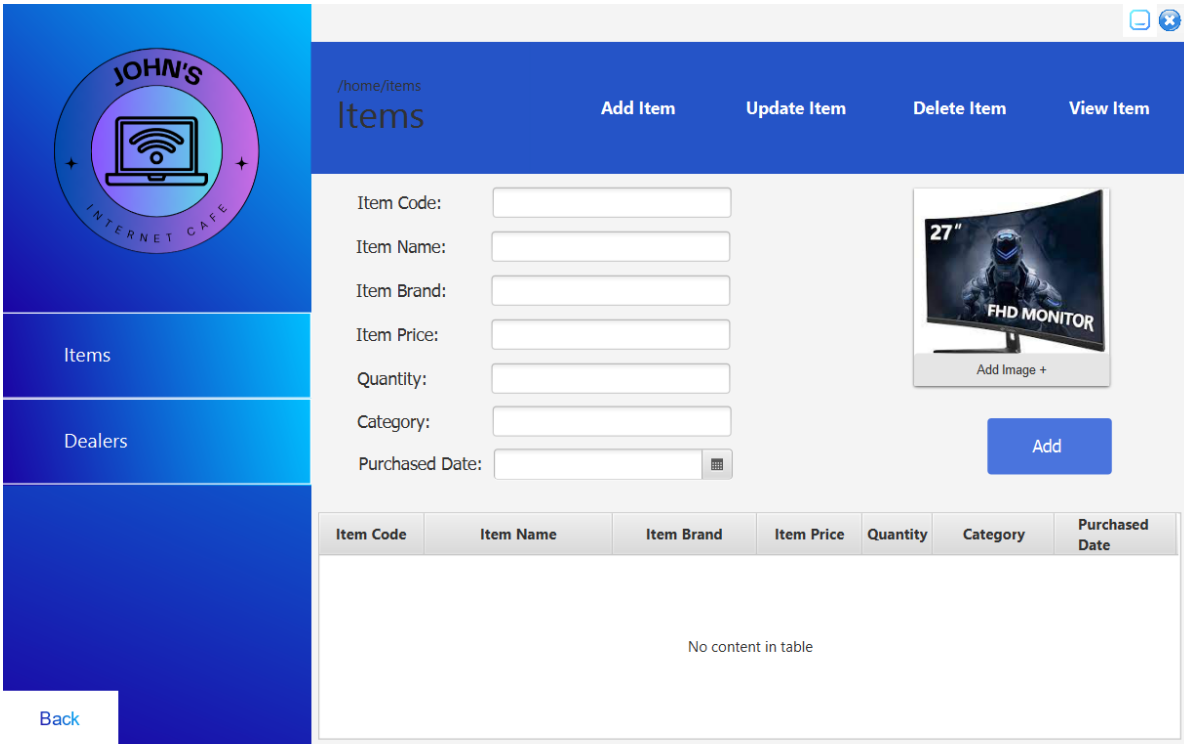
Message

Message

Data Added Successfully.

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date
2	KeyBoard	Singer	12.99	5	Pc	2023-07-12
3	Mouse	Dell	8.99	10	Pc	2023-07-06

- Users can add images using Add Image button.



Item Code:

Item Name:

Item Brand:

Item Price:

Quantity:

Category:

Purchased Date:

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

No content in table

- ❖ Add image:

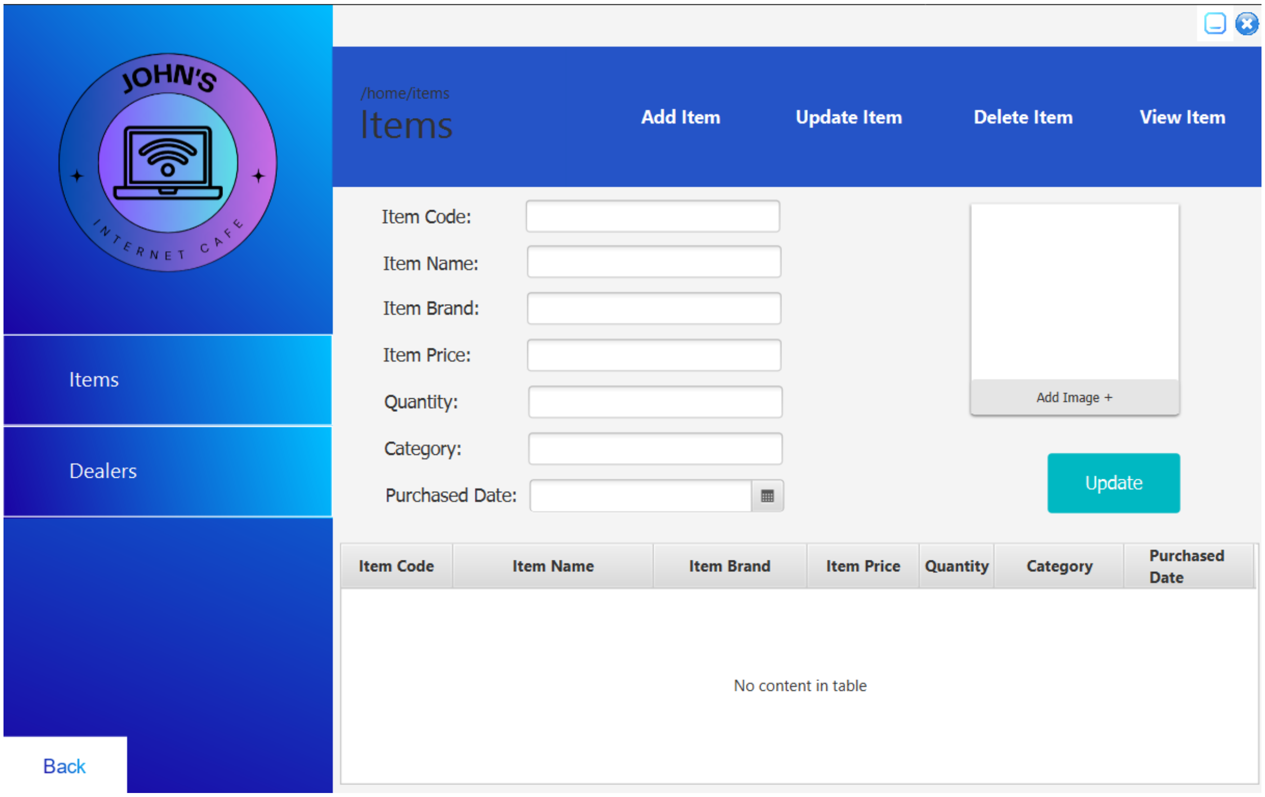
```
@FXML
private void handleAddImage(ActionEvent event) {
    FileChooser fileChooser = new FileChooser();
    fileChooser.setTitle("Open Image File");
    fileChooser.getExtensionFilters().add(new
FileChooser.ExtensionFilter("Image Files", "*.png", "*.jpg", "*.jpeg",
"*.gif"));

    File selectedFile =
fileChooser.showOpenDialog(btnAddImage.getScene().getWindow());

    if (selectedFile != null) {
        try {
            // Load the selected image file into an Image object
            Image image = new Image(selectedFile.toURI().toString());
            // Set the Image object to the ImageView
            addImageView.setImage(image);

            Items selectedItem =
tableView4.getSelectionModel().getSelectedItem();
            if (selectedItem != null) {
                // Set the image for the selected item
                selectedItem.setItemImage(image);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

4.Update items

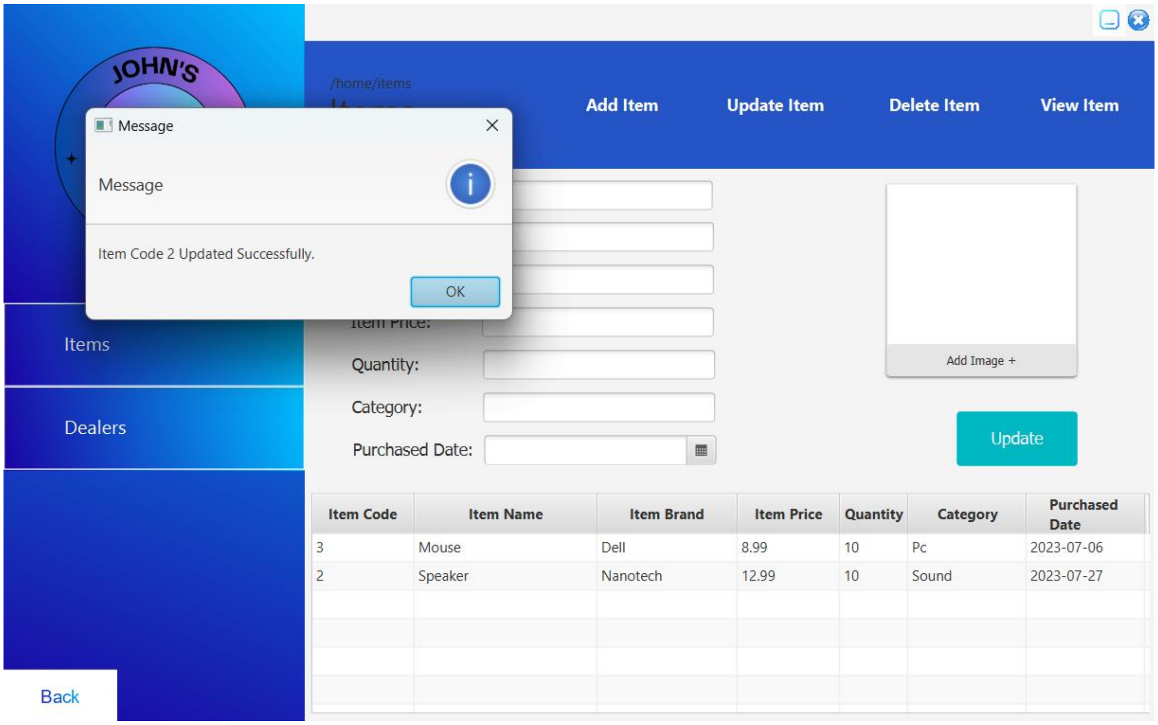


Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

No content in table

➢ In this section users can update item details.

- After updating the item data, the system will display an alert.



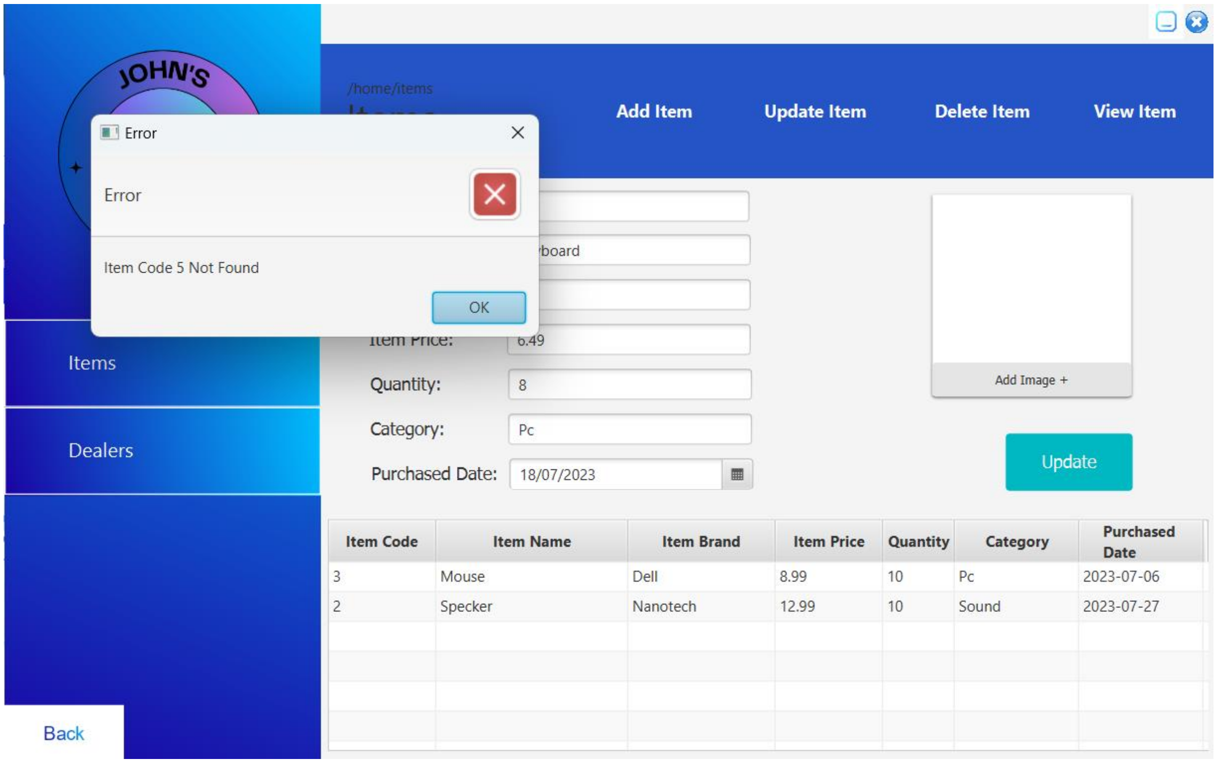
Message

Message

Item Code 2 Updated Successfully.

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date
3	Mouse	Dell	8.99	10	Pc	2023-07-06
2	Speaker	Nanotech	12.99	10	Sound	2023-07-27

- After updating the item data with wrong code, the system will display an error.



Error

Error

Item Code 5 Not Found

board

Item Price: 6.49

Quantity: 8

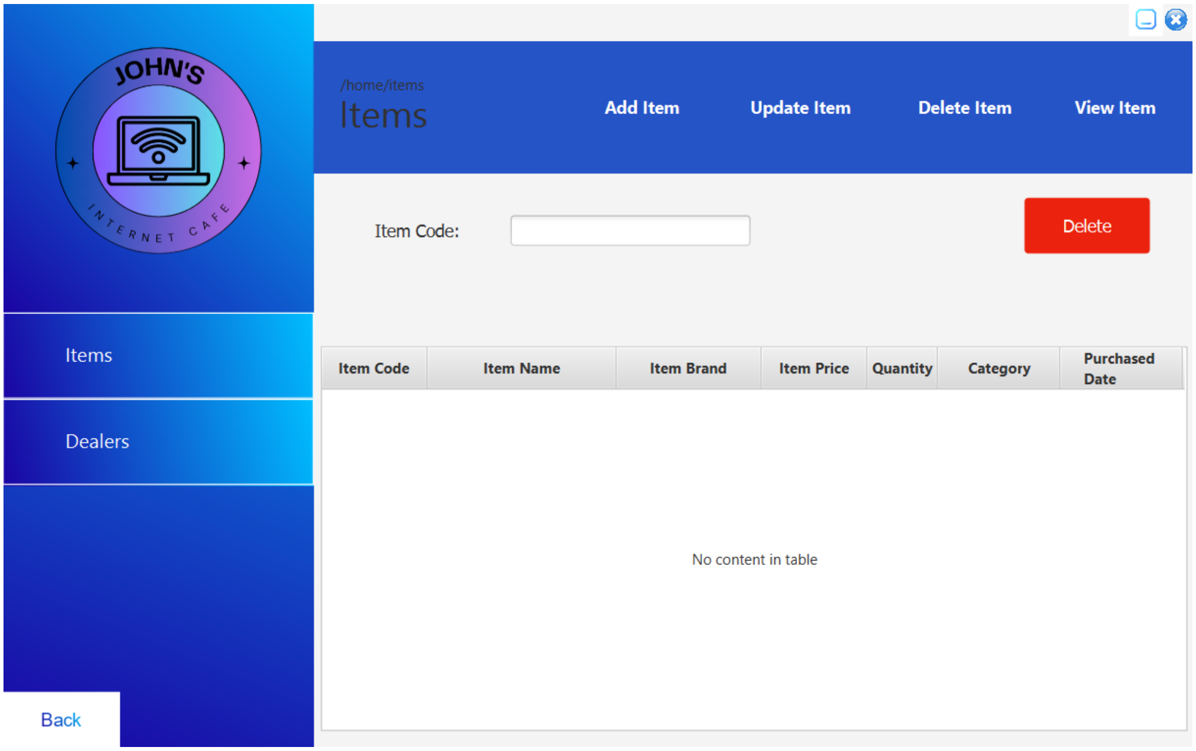
Category: Pc

Purchased Date: 18/07/2023

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date
3	Mouse	Dell	8.99	10	Pc	2023-07-06
2	Specker	Nanotech	12.99	10	Sound	2023-07-27

5.Delete items

- In this section users can delete item data using item code.

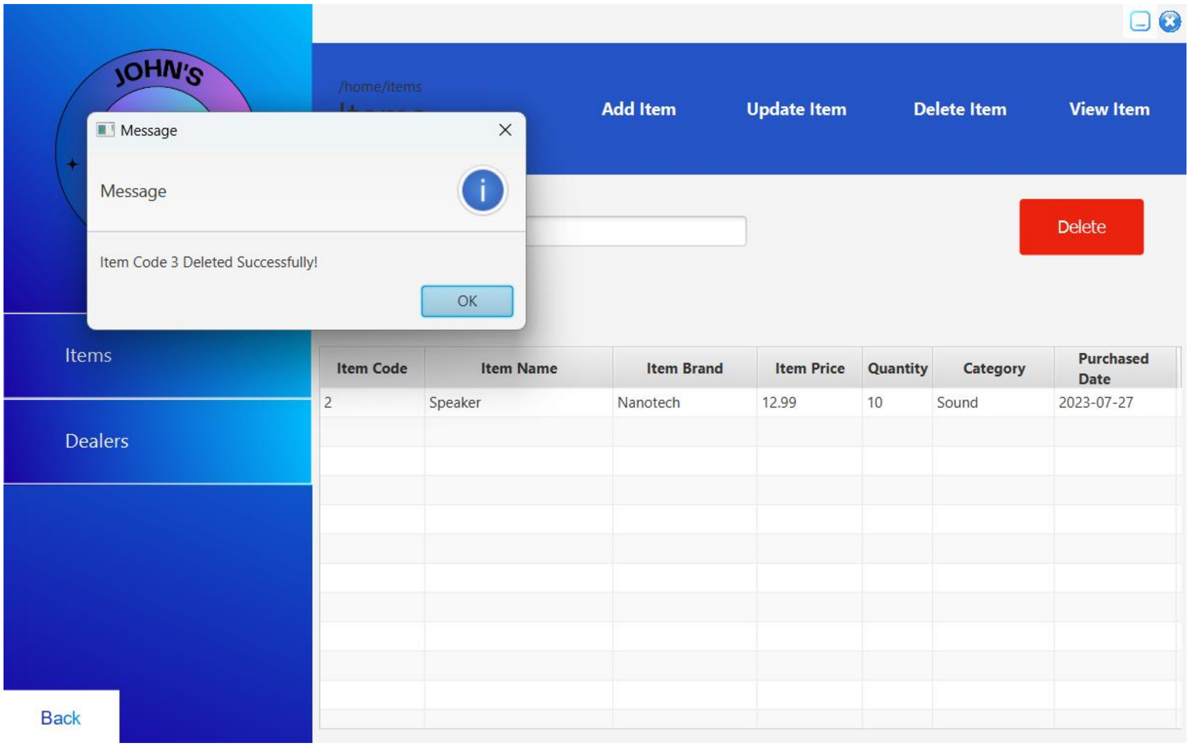


Item Code:

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

No content in table

- After deleting the item data, the system will display an alert.



Message

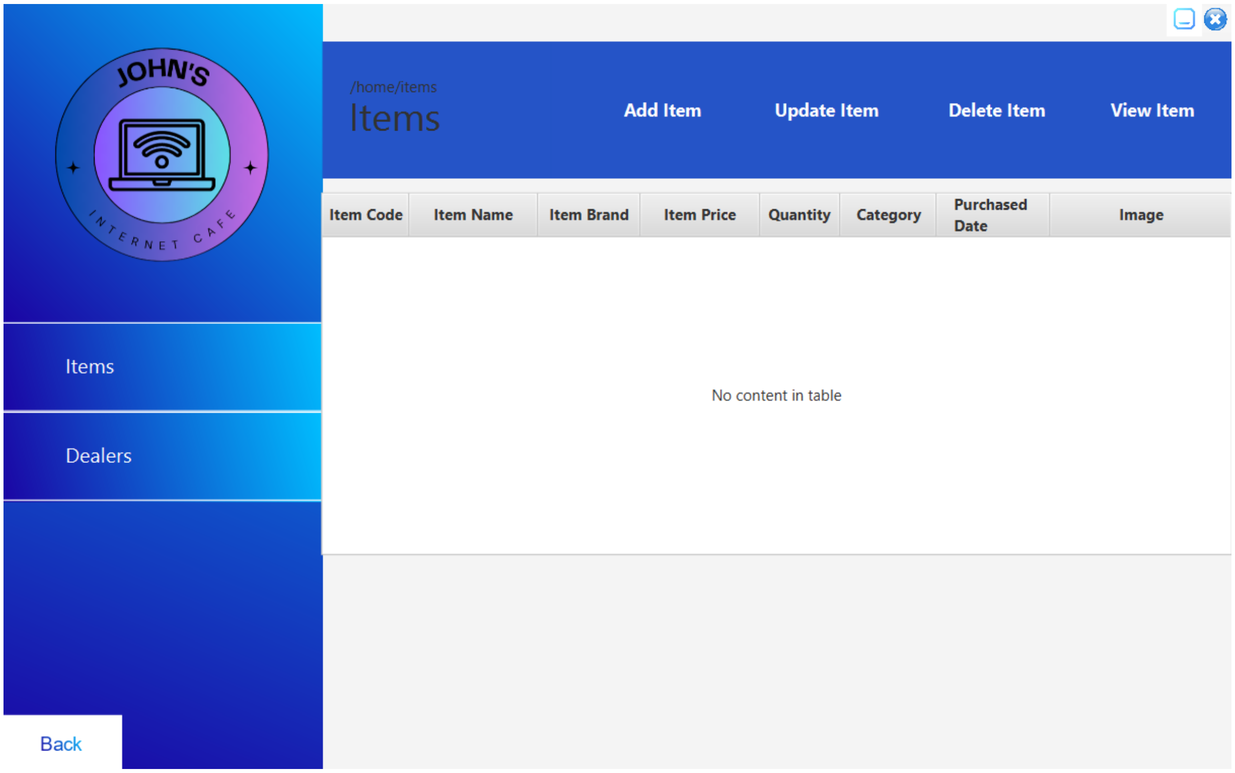
Message

Item Code 3 Deleted Successfully!

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date
2	Speaker	Nanotech	12.99	10	Sound	2023-07-27

6.View items

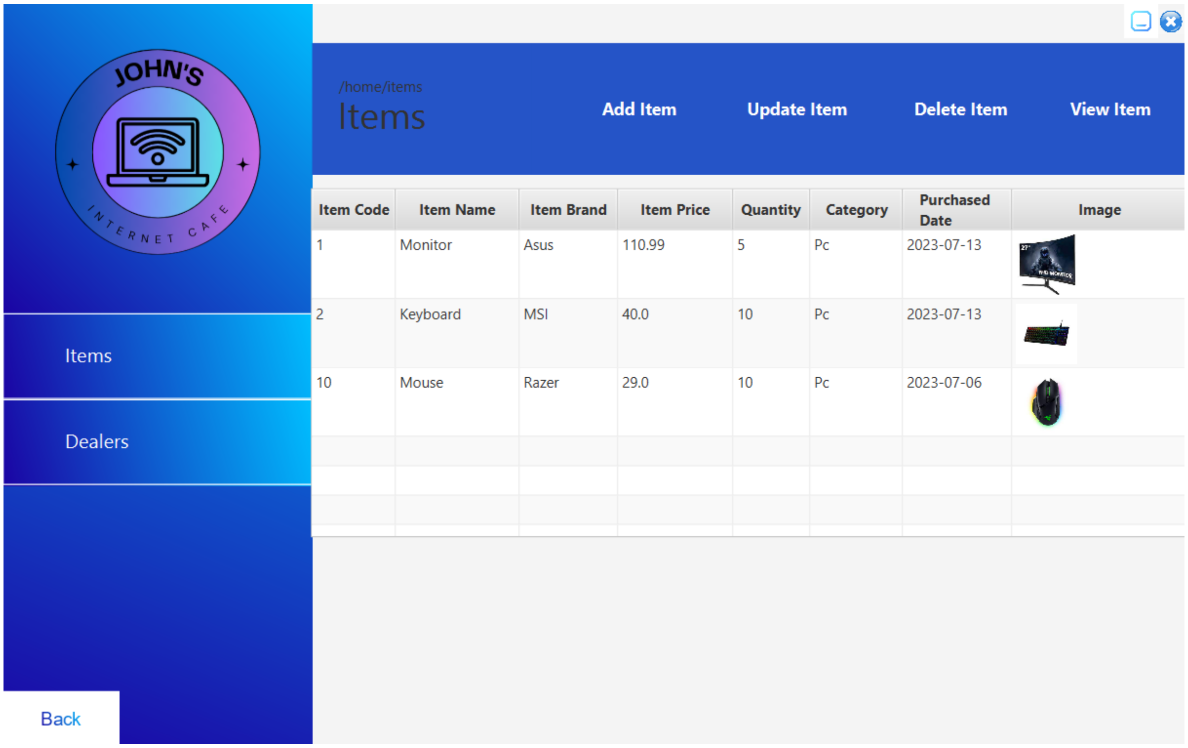
➢ In this section users can view all the item data when they are added.



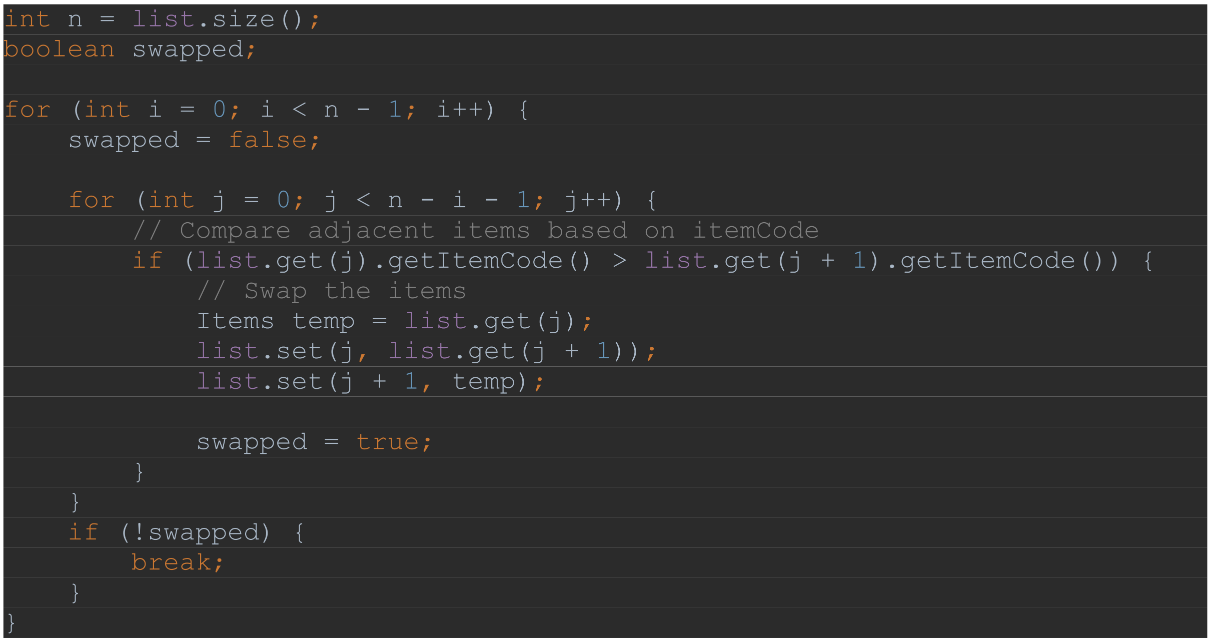
Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date	Image

No content in table

- The system sorts according to the item code automatically when user adds data.



Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date	Image
1	Monitor	Asus	110.99	5	Pc	2023-07-13	
2	Keyboard	MSI	40.0	10	Pc	2023-07-13	
10	Mouse	Razer	29.0	10	Pc	2023-07-06	



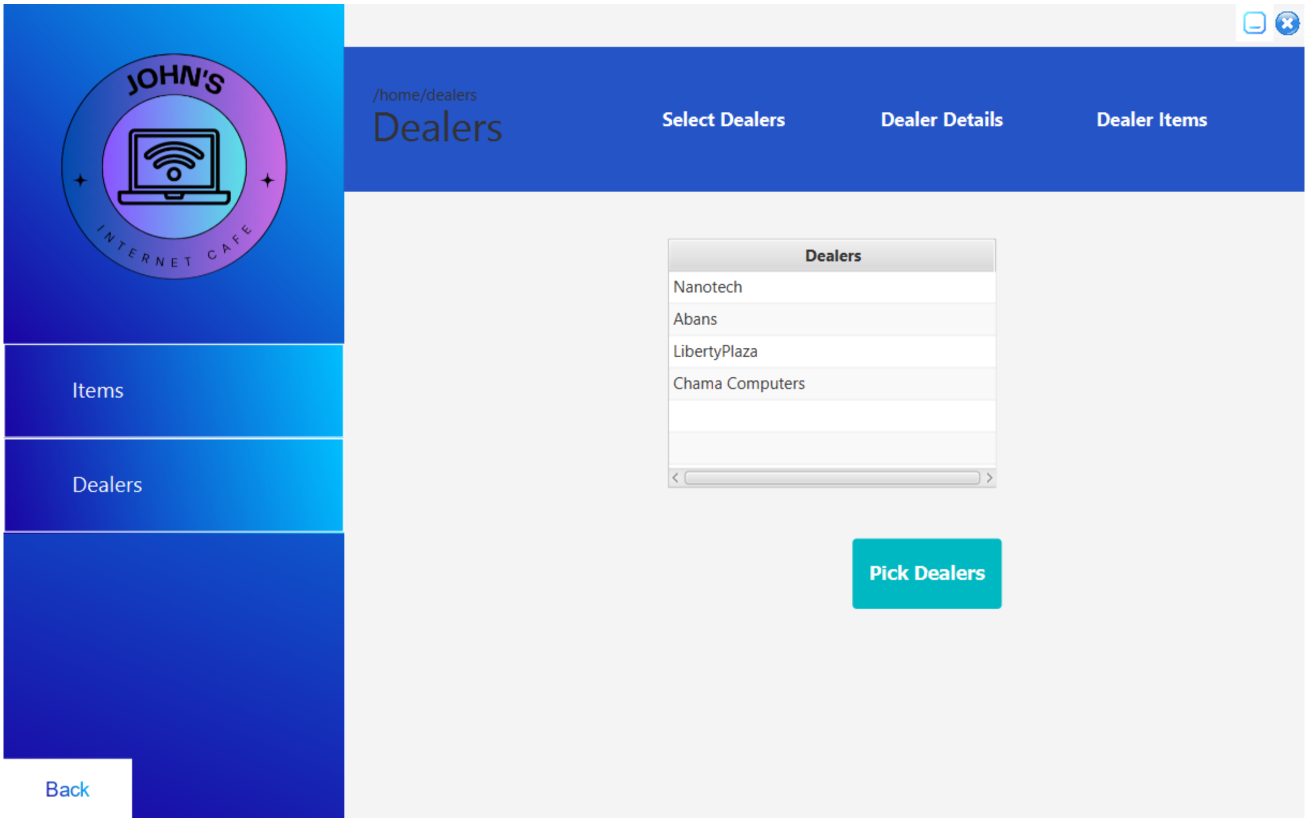
```
int n = list.size();
boolean swapped;

for (int i = 0; i < n - 1; i++) {
    swapped = false;

    for (int j = 0; j < n - i - 1; j++) {
        // Compare adjacent items based on itemCode
        if (list.get(j).getItemCode() > list.get(j + 1).getItemCode()) {
            // Swap the items
            Items temp = list.get(j);
            list.set(j, list.get(j + 1));
            list.set(j + 1, temp);

            swapped = true;
        }
    }
    if (!swapped) {
        break;
    }
}
```

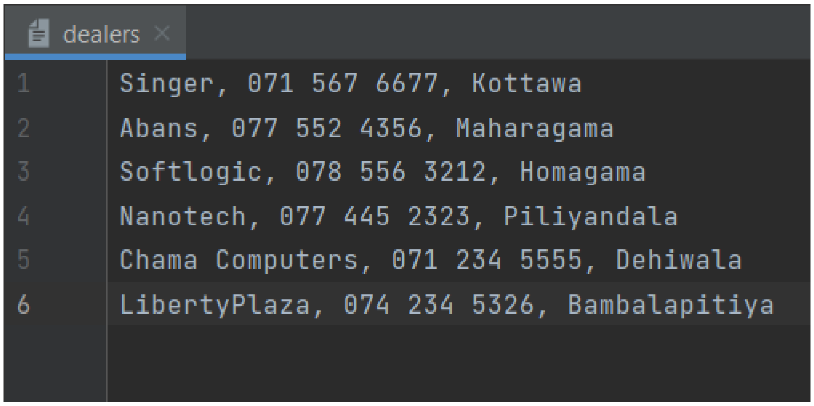
7.Select dealers



Dealers
Nanotech
Abans
LibertyPlaza
Chama Computers

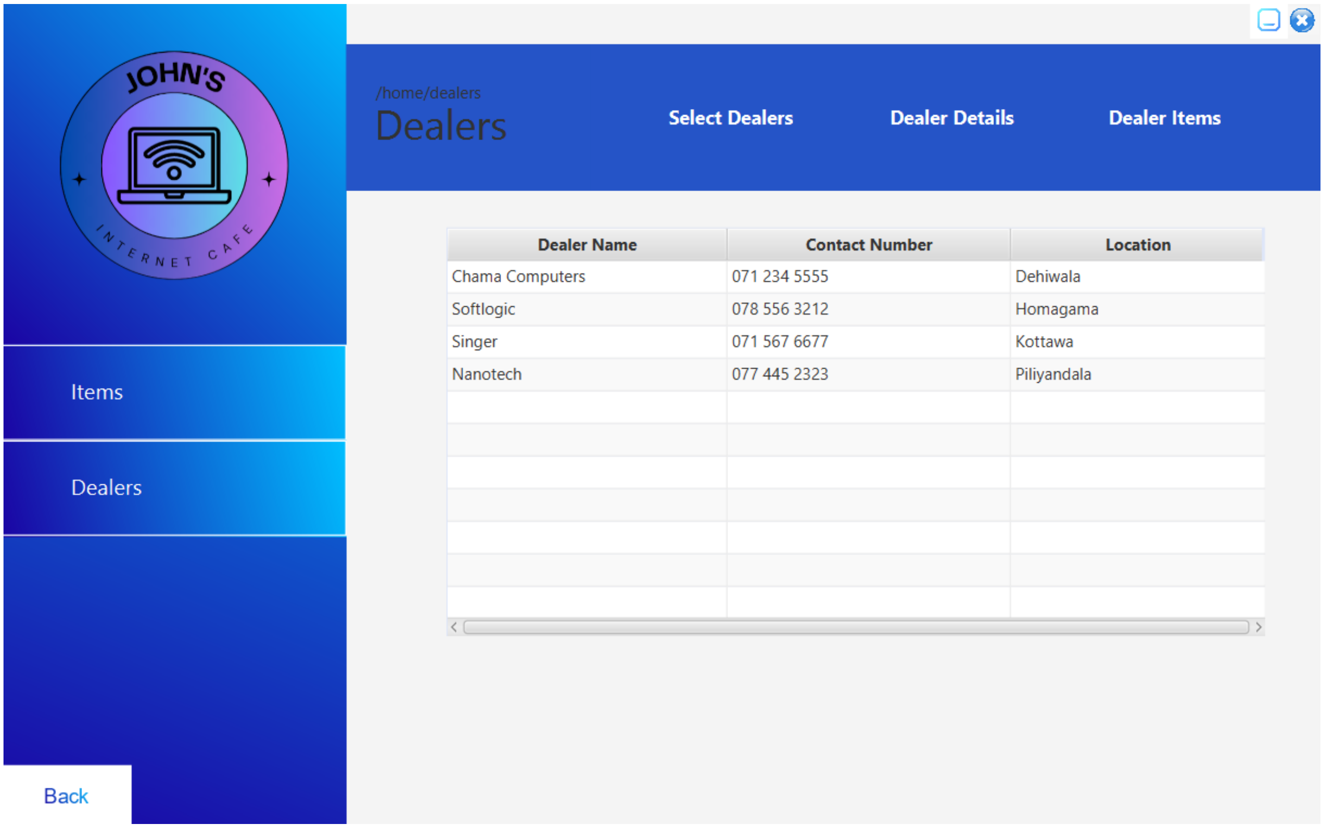
- ➢ In this section user can select 4 random dealers from text file.

- ❖ Dealers.txt



```
Singer, 071 567 6677, Kottawa
Abans, 077 552 4356, Maharagama
Softlogic, 078 556 3212, Homagama
Nanotech, 077 445 2323, Piliyandala
Chama Computers, 071 234 5555, Dehiwala
LibertyPlaza, 074 234 5326, Bambalapitiya
```

8. Dealer details



Dealer Name	Contact Number	Location
Chama Computers	071 234 5555	Dehiwala
Softlogic	078 556 3212	Homagama
Singer	071 567 6677	Kottawa
Nanotech	077 445 2323	Piliyandala

➢ In this section shows details of randomly selected dealers.

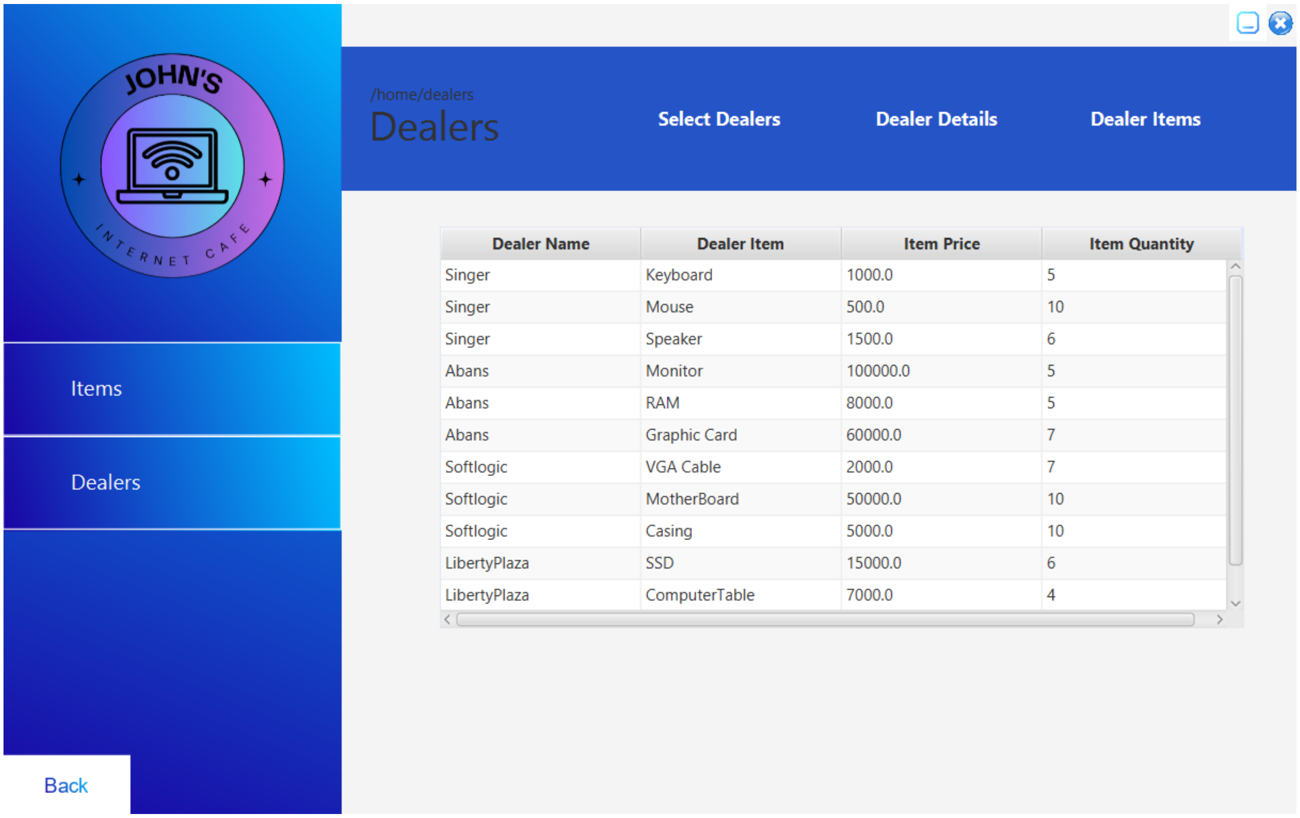
- Sorted according to the dealer location.

```
private void bubbleSort(List<Dealers> dealers) {
    int n = dealers.size();
    boolean swapped;

    for (int i = 0; i < n - 1; i++) {
        swapped = false;
        for (int j = 0; j < n - i - 1; j++) {
            // Compare dealer locations using the compareTo() method
            if (dealers.get(j).getDealerLocation().compareTo(dealers.get(j
+ 1).getDealerLocation()) > 0) {
                // Swap dealers at index j and j+1
                Dealers temp = dealers.get(j);
                dealers.set(j, dealers.get(j + 1));
                dealers.set(j + 1, temp);
                swapped = true;
            }
        }

        if (!swapped) {
            break;
        }
    }
}
```

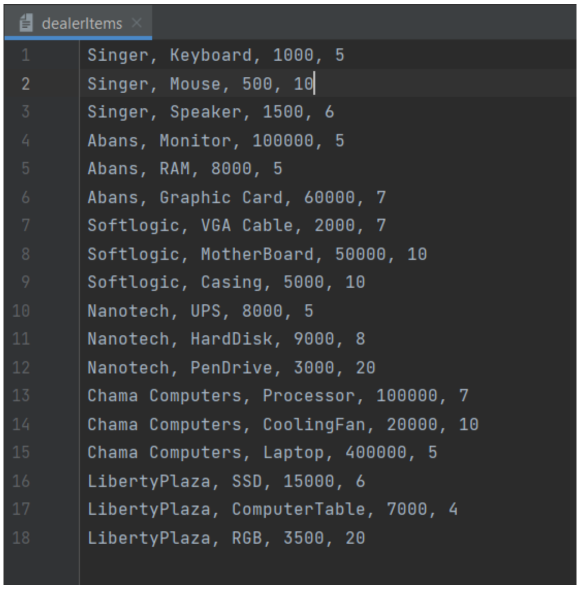
9. Dealer Items



Dealer Name	Dealer Item	Item Price	Item Quantity
Singer	Keyboard	1000.0	5
Singer	Mouse	500.0	10
Singer	Speaker	1500.0	6
Abans	Monitor	100000.0	5
Abans	RAM	8000.0	5
Abans	Graphic Card	60000.0	7
Softlogic	VGA Cable	2000.0	7
Softlogic	MotherBoard	50000.0	10
Softlogic	Casing	5000.0	10
LibertyPlaza	SSD	15000.0	6
LibertyPlaza	ComputerTable	7000.0	4

- ➢ This section shows items of randomly selected dealers.

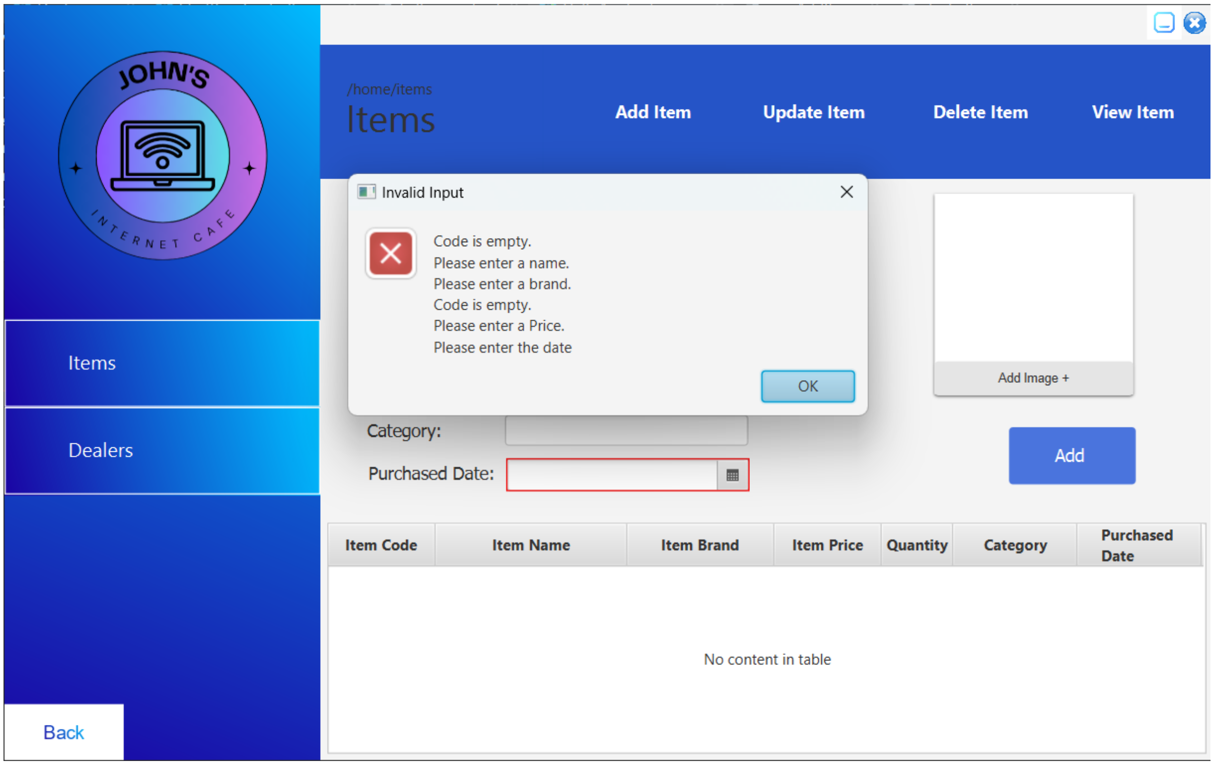
- ❖ DealerItems.txt



```
Singer, Keyboard, 1000, 5
Singer, Mouse, 500, 10
Singer, Speaker, 1500, 6
Abans, Monitor, 100000, 5
Abans, RAM, 8000, 5
Abans, Graphic Card, 60000, 7
Softlogic, VGA Cable, 2000, 7
Softlogic, MotherBoard, 50000, 10
Softlogic, Casing, 5000, 10
Nanotech, UPS, 8000, 5
Nanotech, HardDisk, 9000, 8
Nanotech, PenDrive, 3000, 20
Chama Computers, Processor, 100000, 7
Chama Computers, CoolingFan, 20000, 10
Chama Computers, Laptop, 400000, 5
LibertyPlaza, SSD, 15000, 6
LibertyPlaza, ComputerTable, 7000, 4
LibertyPlaza, RGB, 3500, 20
```

Validation

- All the fields must be filled.



Invalid Input

Code is empty.
Please enter a name.
Please enter a brand.
Code is empty.
Please enter a Price.
Please enter the date

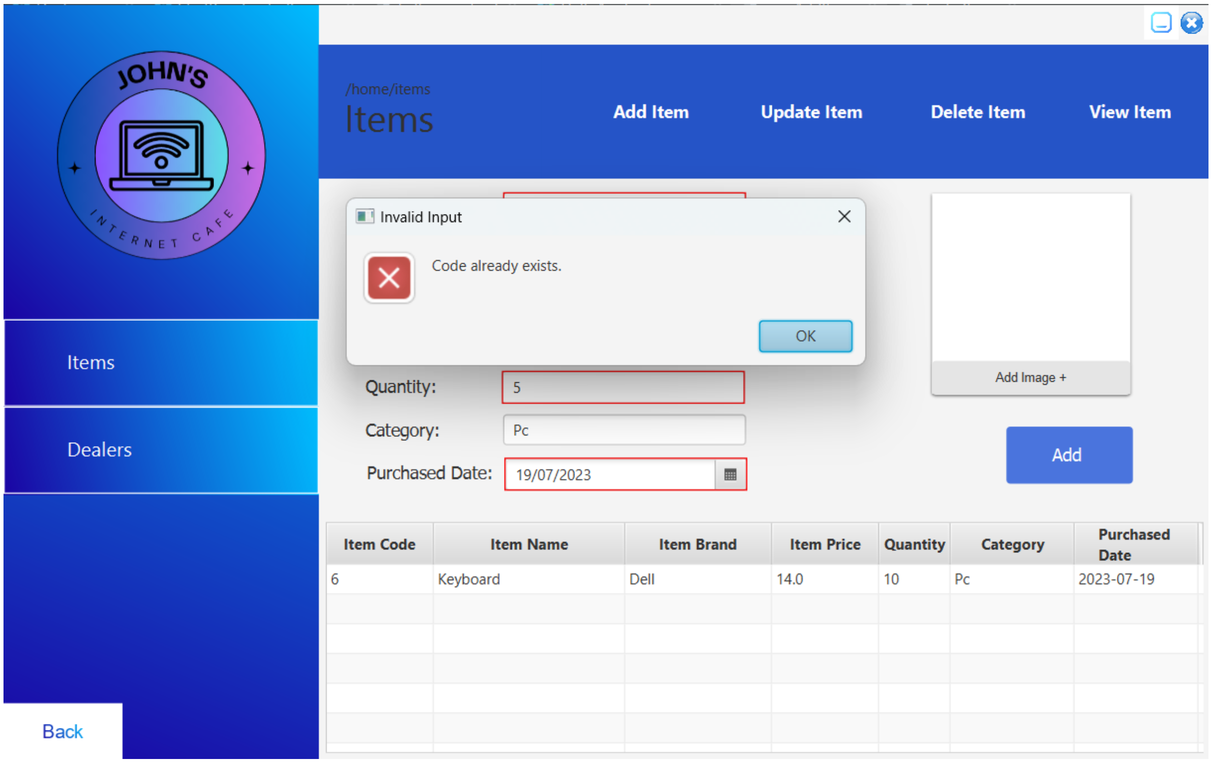
Category:

Purchased Date:

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

No content in table

- Only one item can exist with one item code.



Invalid Input

Code already exists.

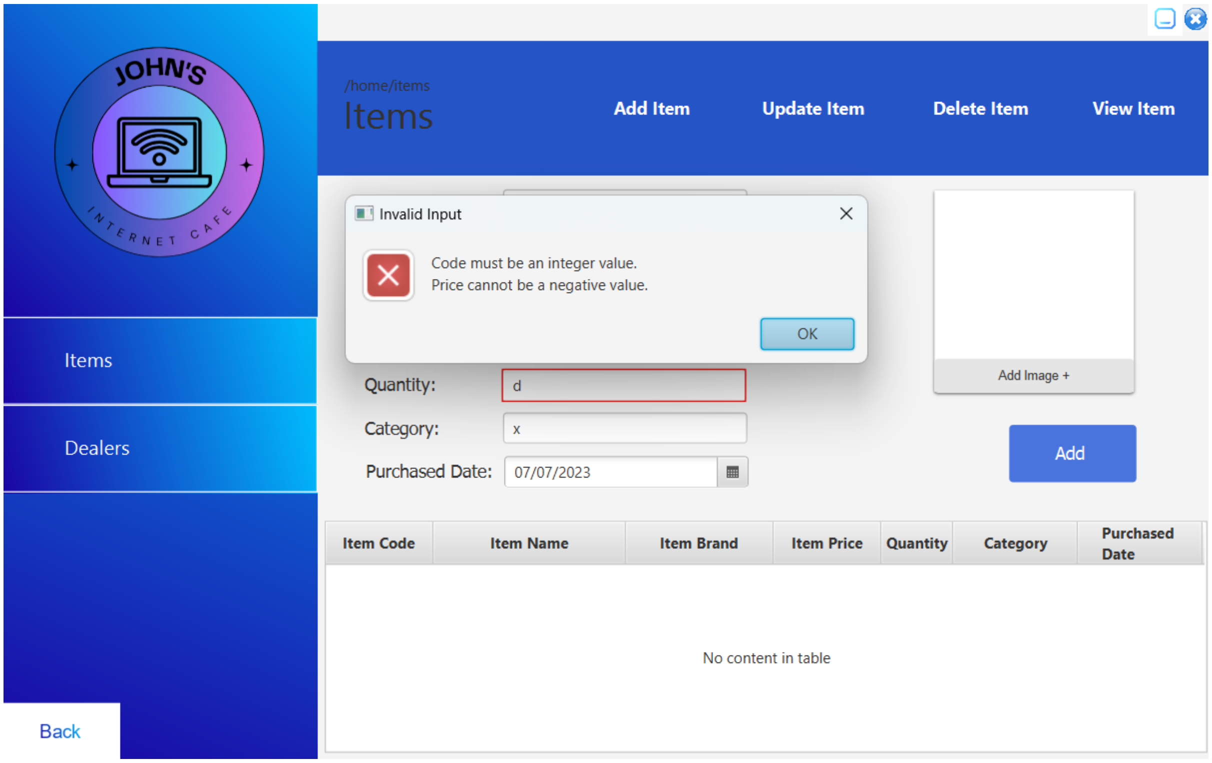
Quantity: 5

Category: Pc

Purchased Date: 19/07/2023

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date
6	Keyboard	Dell	14.0	10	Pc	2023-07-19

- Price and quantity can’t be a negative value or String.



Invalid Input

Code must be an integer value.
Price cannot be a negative value.

Quantity: d

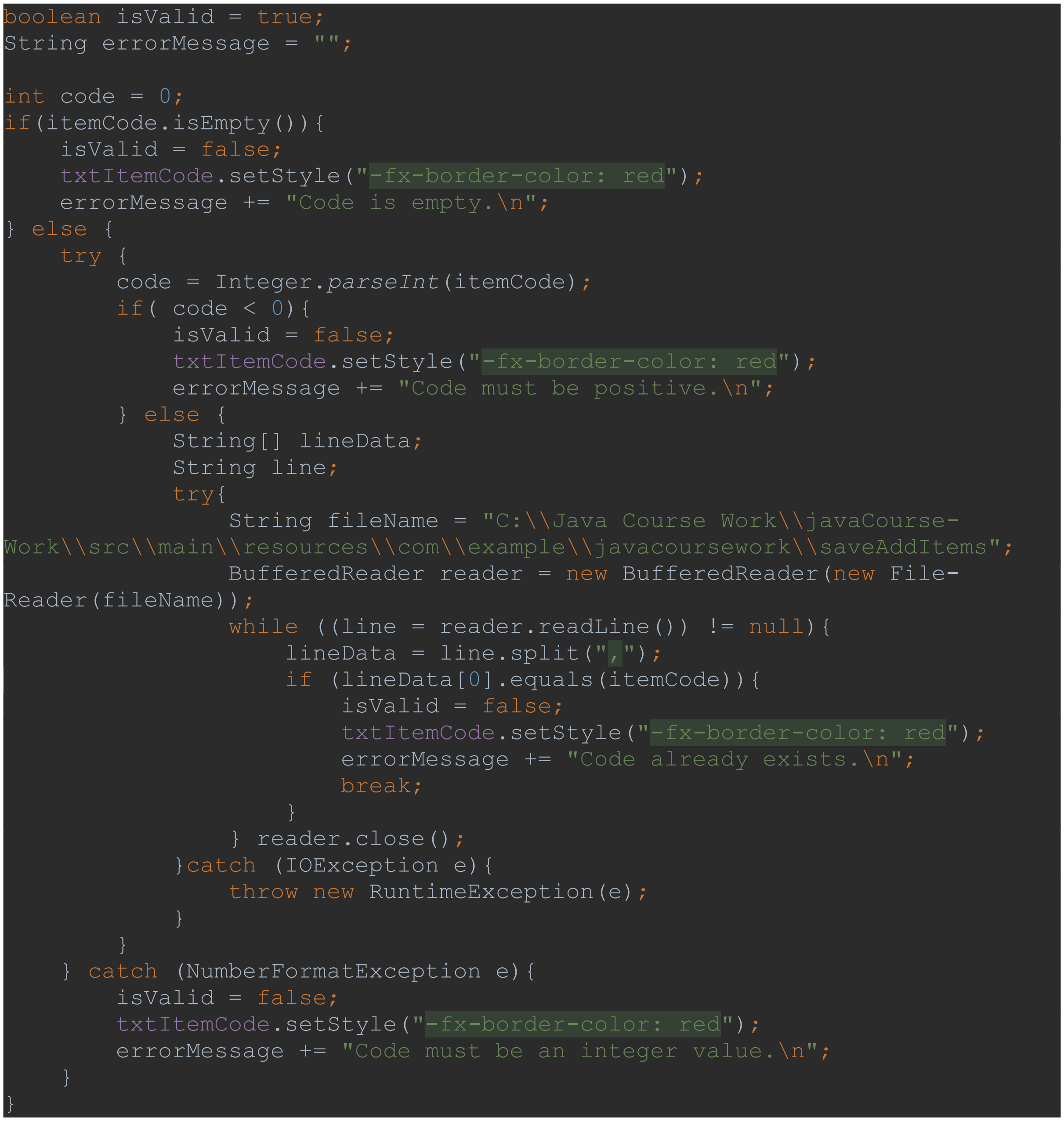
Category: x

Purchased Date: 07/07/2023

Item Code	Item Name	Item Brand	Item Price	Quantity	Category	Purchased Date

No content in table

- ❖ Validate item code:



```
boolean isValid = true;
String errorMessage = "";

int code = 0;
if(itemCode.isEmpty()){
    isValid = false;
    txtItemCode.setStyle("-fx-border-color: red");
    errorMessage += "Code is empty.\n";
} else {
    try {
        code = Integer.parseInt(itemCode);
        if( code < 0){
            isValid = false;
            txtItemCode.setStyle("-fx-border-color: red");
            errorMessage += "Code must be positive.\n";
        } else {
            String[] lineData;
            String line;
            try{
                String fileName = "C:\\Java Course Work\\javaCourse-Work\\src\\main\\resources\\com\\example\\javacoursework\\saveAddItems";
                BufferedReader reader = new BufferedReader(new FileReader(fileName));
                while ((line = reader.readLine()) != null){
                    lineData = line.split(",");
                    if (lineData[0].equals(itemCode)){
                        isValid = false;
                        txtItemCode.setStyle("-fx-border-color: red");
                        errorMessage += "Code already exists.\n";
                        break;
                    }
                } reader.close();
            }catch (IOException e){
                throw new RuntimeException(e);
            }
        }
    } catch (NumberFormatException e){
        isValid = false;
        txtItemCode.setStyle("-fx-border-color: red");
        errorMessage += "Code must be an integer value.\n";
    }
}
```

- ❖ Validate item name:

```
if (itemName.isEmpty()) {
    isValid = false;
    txtItemName.setStyle("-fx-border-color: red;");
    errorMessage += "Please enter a name.\n";
} else {
    // Check if the name contains only letters and spaces
    if (!itemName.matches("[a-zA-Z ]+")) {
        isValid = false;
        txtItemName.setStyle("-fx-border-color: red;");
        errorMessage += "Please enter a valid name (only letters and spaces
are allowed).\n";
    }
}
```

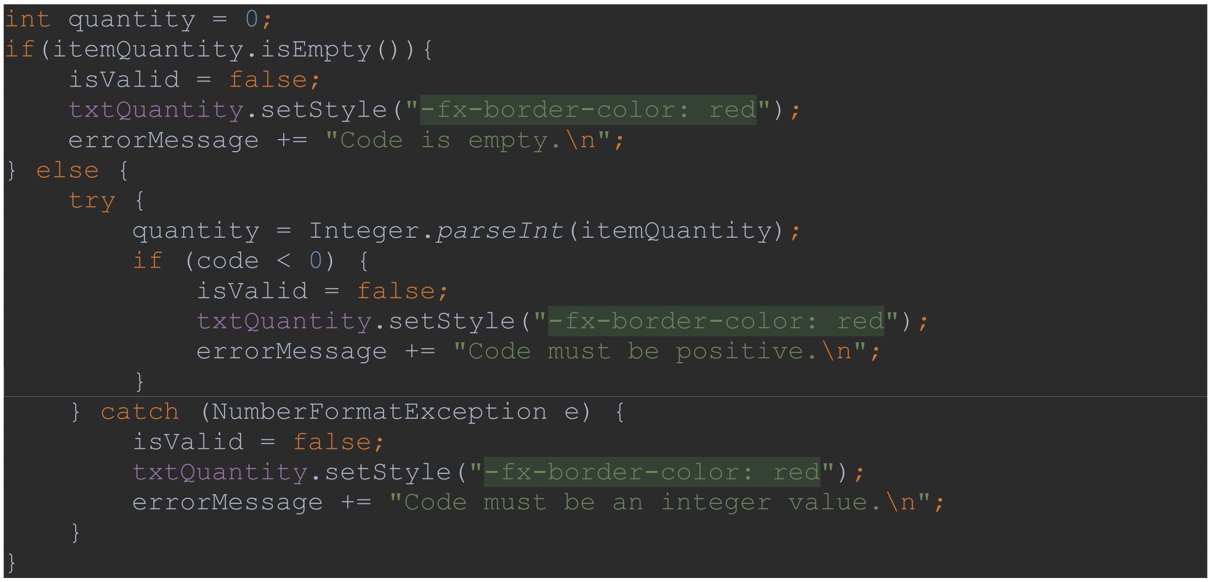
- ❖ Validate item brand:

```
if (itemBrand.isEmpty()) {
    isValid = false;
    txtItemBrand.setStyle("-fx-border-color: red;");
    errorMessage += "Please enter a brand.\n";
}
```

- ❖ Validate item category:

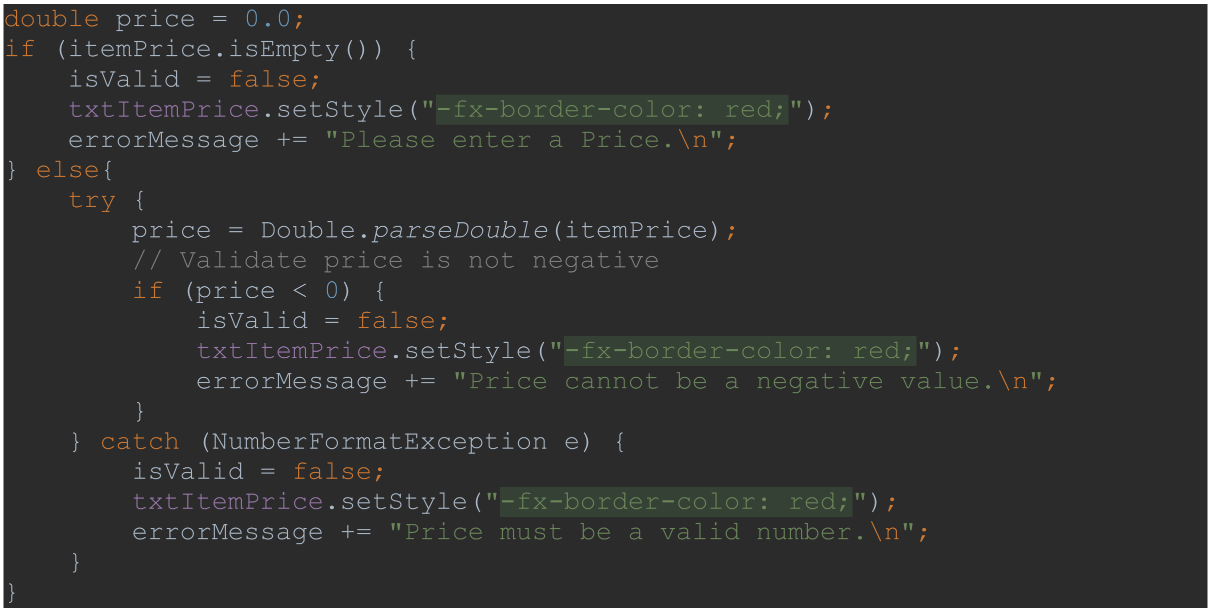
```
if (itemCategory.isEmpty()) {
    isValid = false;
    txtCategory.setStyle("-fx-border-color: red;");
    errorMessage += "Please enter a category.\n";
}
```

- ❖ Validate item quantity:



```
int quantity = 0;
if(itemQuantity.isEmpty()){
    isValid = false;
    txtQuantity.setStyle("-fx-border-color: red");
    errorMessage += "Code is empty.\n";
} else {
    try {
        quantity = Integer.parseInt(itemQuantity);
        if (code < 0) {
            isValid = false;
            txtQuantity.setStyle("-fx-border-color: red");
            errorMessage += "Code must be positive.\n";
        }
    } catch (NumberFormatException e) {
        isValid = false;
        txtQuantity.setStyle("-fx-border-color: red");
        errorMessage += "Code must be an integer value.\n";
    }
}
```

- ❖ Validate item price:



```
double price = 0.0;
if (itemPrice.isEmpty()) {
    isValid = false;
    txtItemPrice.setStyle("-fx-border-color: red;");
    errorMessage += "Please enter a Price.\n";
} else{
    try {
        price = Double.parseDouble(itemPrice);
        // Validate price is not negative
        if (price < 0) {
            isValid = false;
            txtItemPrice.setStyle("-fx-border-color: red;");
            errorMessage += "Price cannot be a negative value.\n";
        }
    } catch (NumberFormatException e) {
        isValid = false;
        txtItemPrice.setStyle("-fx-border-color: red;");
        errorMessage += "Price must be a valid number.\n";
    }
}
```

- ❖ Validate item purchased date:

```
if (date == null){
    isValid = false;
    datePicker.setStyle("-fx-border-color: red");
    errorMessage += "Please enter the date\n";
}
```

- ❖ Error checking and adding details to the driver list:

```
// If all fields are valid, proceed with adding the item
    if (isValid) {

        // Create the new Items object and add it to the itemsList
        Items newItem = new Items(Integer.parseInt(itemCode), itemName,
itemBrand, Float.parseFloat(itemPrice), Integer.parseInt(itemQuantity),
itemCategory, date, itemImage);
        itemsList.add(newItem);
        list.add(newItem);

        // Clear input fields after adding data
        txtItemCode.clear();
        txtItemName.clear();
        txtItemBrand.clear();
        txtCategory.clear();
        txtItemPrice.clear();
        txtQuantity.clear();
        datePicker.setValue(null);
        addImageView.setImage(null);

        txtItemCode.setStyle("");
        txtItemName.setStyle("");
        txtItemBrand.setStyle("");
        txtCategory.setStyle("");
        txtItemPrice.setStyle("");
        txtQuantity.setStyle("");
        datePicker.setStyle("");

        settingTheTable();
        tableView.setItems(list);

        saveDataToFile();
    } else {
        showAlert("Invalid Input", errorMessage);
    }
}
private void showAlert(String title, String message){
    Alert alert = new Alert(Alert.AlertType.ERROR);
    alert.setTitle(title);
    alert.setHeaderText(null);
    alert.setContentText(message);
    alert.showAndWait();
}
```

J-units, Test plan and test cases

1.Add items

Test plan for **Add Items**				
Test No:	Test Input	Expected Output	Actual Output	Pass/ Fail

1	Code = 01
	Name = Monitor
	Brand = Razer
	Price = 89.99
	Quantity = 10
	Category = Pc
	Date = 13/07/2023
	Image =
		Add data into the view table	Add data into the view table	Pass

2	Code = 01	Display warning: "Code already exists." and the color of the boarder turns red.	Code already exists.	Pass
3	Price = Eighty-nine	Display warning: "Price must be a valid number." and the color of the boarder turns red.	Price must be a valid number.	Pass
4	Quantity = Ten	Display warning: "Quantity must be an integer value." and the color of the boarder turns red.	Quantity must be an integer value.	Pass

5	Code = null
	Name = null
	Brand = null
	Price = null
	Quantity = null
	Category = null
	Date = null
		Display warning: "Code is empty,
		Please enter a name,
		Please enter a brand,
		Please enter a category,
		Quantity cannot be empty,
		Please enter a price,
		Please enter a date"
			"Code is empty,
			Please enter a name,
			Please enter a brand,
			Please enter a category,
			Quantity cannot be empty,
			Please enter a price,
			Please enter a date"
				Pass

2.Update Item

Test plan for **Update Items**				
Test No:	Test Input	Expected Output	Actual Output	Pass/ Fail

1	Code = 01
	Name = Keyboard
	Brand = Razer
	Price = 25
	Quantity = 10
	Category = Pc
	Date = 13/07/2023
	Image =
		Update data into the view table	Update data into the view table	Pass

2	Code = 05	Display warning: “Item code 05 (Input code) not found.” and the color of the boarder turns red.	Item code 05 not found.	Pass

3.Delete Item

Test plan for **Delete Items**				
Test No:	Test Input	Expected Output	Actual Output	Pass/ Fail
1	Code = 01	Display message: “Item code (entered code) deleted successfully.”	Item code 01 deleted successfully.	Pass
2	Code = 05	Display warning: “Item code 05 (Input code) not found.”	Item code 05 not found.	Pass

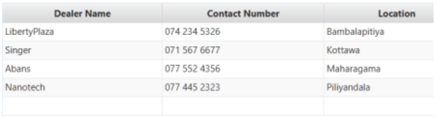
4.View Item

Test plan for **Dealer Details**				
Test No:	Test	Expected Output	Actual Output	Pass/ Fail
1	Click “View Item” button.	Displaying added item details	Displaying added item details	Pass

5.Select dealers

Test plan for **Select Dealers**				
Test No:	Test	Expected Output	Actual Output	Pass/ Fail
1	Click “Pick Dealer” button.	Display 4 random dealers.	Abans Nanotech Singer Softlogic	Pass
2	Click again “Pick Dealer” button.	Display 4 different dealers.	Softlogic Chama Computers Liberty Plaza Singer	Pass

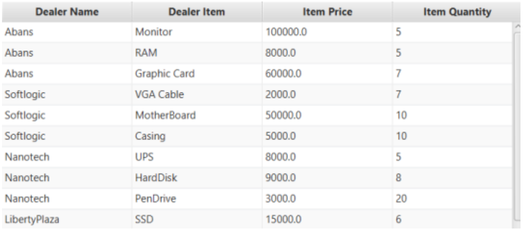
6.Dealer Details

Test plan for **Dealer Details**				
Test No:	Test	Expected Output	Actual Output	Pass/ Fail
1	Click “Dealer Details” button.	Display details (Name, Contact, Location) of selected dealers.		Pass

Dealer Name	Contact Number	Location
LibertyPlaza	074 234 5326	Bambalapitiya
Singer	071 567 6677	Kottawa
Abans	077 552 4356	Maharagama
Nanotech	077 445 2323	Piliyandala

7.Dealer Items

<table>
<tr><th colspan="5">Test plan for

Dealer Items</th></tr>
<tr><td>Test No:</td><td>Test</td><td>Expected Output</td><td>Actual Output</td><td>Pass/ Fail</td></tr>
<tr><td>1</td><td>Click “Dealer Items” button.</td><td>Display Items (Dealer Name, Dealer Item, Item Price, Item Quantity) of selected dealers.</td><td>
<table>
<tr><th>Dealer Name</th><th>Dealer Item</th><th>Item Price</th><th>Item Quantity</th></tr>
<tr><td>Abans</td><td>Monitor</td><td>100000.0</td><td>5</td></tr>
<tr><td>Abans</td><td>RAM</td><td>8000.0</td><td>5</td></tr>
<tr><td>Abans</td><td>Graphic Card</td><td>60000.0</td><td>7</td></tr>
<tr><td>Softlogic</td><td>VGA Cable</td><td>2000.0</td><td>7</td></tr>
<tr><td>Softlogic</td><td>MotherBoard</td><td>50000.0</td><td>10</td></tr>
<tr><td>Softlogic</td><td>Casing</td><td>5000.0</td><td>10</td></tr>
<tr><td>Nanotech</td><td>UPS</td><td>8000.0</td><td>5</td></tr>
<tr><td>Nanotech</td><td>HardDisk</td><td>9000.0</td><td>8</td></tr>
<tr><td>Nanotech</td><td>PenDrive</td><td>3000.0</td><td>20</td></tr>
<tr><td>LibertyPlaza</td><td>SSD</td><td>15000.0</td><td>6</td></tr>
</table></td><td>Pass</td></tr>
</table>

Robustness and the maintainability

Robustness:

- ✓ Input Validation: The code performs various input validations to ensure that the data entered by the user is valid and appropriate for the fields.

- ✓ Error Handling: The code includes error handling mechanisms, such as displaying error messages when input data is invalid.

Maintainability:

- ✓ Modular Design: The code is divided into several methods, each serving a specific purpose. This modular design improves maintainability by allowing developers to understand and modify individual parts of the code without affecting other sections.

- ✓ Code Reusability: The code uses reusable components. Reusable code reduces redundancy and simplifies maintenance.

- ✓ Clear Variable Naming: Variable names like itemCode, itemName, etc., have meaningful names, which makes the code more readable and easier to maintain.

- ✓ Comments: Properly documented code helps other developers understand the codebase more effectively.

Conclusions & Assumptions

Conclusion:

- ✓ The code implements a JavaFX application for managing item data in an inventory system.
- ✓ It provides basic input validation to ensure that the user enters valid and appropriate data for item details.

Assumptions:

- ✓ I decided the item code should be unique because the system identifies items using their code.
- ✓ Code, Price and Quantity must be an integer value.
- ✓ I'm assuming that the user selects the date using date picker.
- ✓ I'm assuming this system is only for John's personal use.

Reference

- www.youtube.com. (n.d.). OOP 06 - JAVA Constructor Overloading in Sinhala. [online] Available at: https://www.youtube.com/watch?v=OpAp5-603cg&list=RDCMUCHv_YIRCWBqJEBkk2I0JlPg&start_radio=1&rv=OpAp5-603cg&t=348 [Accessed 30 Jul. 2023].

- www.youtube.com. (n.d.). #1 JavaFX UI Design Dashboard Utility. [online] Available at: https://www.youtube.com/watch?v=cPF3qGTjYgk&t=629s [Accessed 30 Jul. 2023].

- www.youtube.com. (n.d.). JavaFX DatePicker 📅. [online] Available at: https://www.youtube.com/watch?v=3Ht-JMQh2JI&t=182s [Accessed 30 Jul. 2023].

- Canva (2023). Canva. [online] Canva. Available at: https://www.canva.com/.