Pusat Pengajian Pengkomputeran
School of Computing
Universiti Utara Malaysia

STIA1113 PROGRAMMING I

FIRST SEMESTER SESSION 2023/2024 (A231)

GROUP PROJECT

PROJECT TITLE:
Computer Shop Management System Report

Prepared by:
Muhamad Hamiezi (301785)
Andyderis P.A.S (296530)

Prepared for:
Dr. Baharudin Bin Osman

COMPUTER SHOP MANAGEMENT SYSTEM

INTRODUCTION OF THE PROGRAM

What is the program is all about? The Computer Shop Management System is a multiple-function tool to provide assistance in managing computer shop operations. Comes with features that are user-friendly with its own functionalities such as menu interaction, sales, restocking, stock, add, and deletion, this system could help workers to work more efficiently compared to traditional managing systems. With a structure based on selection statement, repetition statement, method, and array to keep the information, the program aims to provide simplified daily tasks, efficient financial management, and overall east operation manager to manage the shop.

Why do you need a system? Below are a few reasons why having the system is a must.

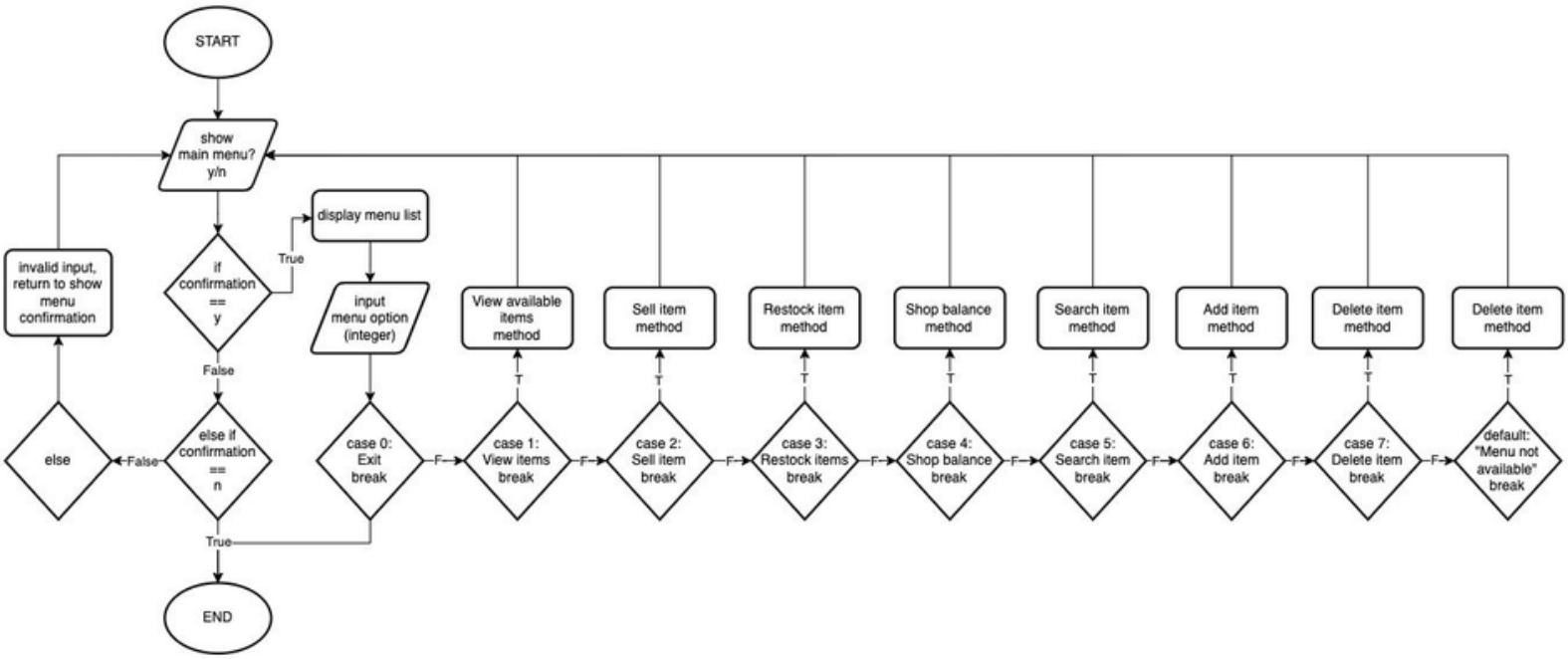
1. Efficiency, reducing the manual workload on workers. Having a non-human operational system able to process faster and more accurate results in managing inventory, sales, and financial transactions. enhances the overall efficiency of operations within a computer shop.
2. Accuracy, having systems to help maintain data accuracy by minimizing human errors.

What are the functions? Below are all seven function that comes with the program.

1. Show available item
2. Sell item
3. Restock item
4. Show shop balance
5. Search shop item
6. Add new item
7. Delete item

COMPUTER SHOP MANAGEMENT SYSTEM

THE CODE



THE CODE-APP PAGE

```
 1 | import java.util.Scanner;
 2 |
 3 | public class App {
 4 |
 5 |     public static void main(String[] args) {
 6 |
 7 |         System.out.println(" ");
 8 |         System.out.println("************************************************");
 9 |         System.out.println("*                                              *");
10 |         System.out.println("*  Welcome to Computer Shop Management System  *");
11 |         System.out.println("*                                              *");
12 |         System.out.println("************************************************");
13 |         System.out.println(" ");
14 |
15 |         switchMenu();
16 |     }
17 |
18 |     public static void switchMenu() {
19 |
20 |             System.out.println("\nDo you want to show Main Menu? (y/n) ");
21 |
22 |             Scanner scanner = new Scanner(System.in);
23 |             // Get user confirmation
24 |             char confirmation = scanner.next().charAt(0);
25 |
26 |
```

COMPUTER SHOP MANAGEMENT SYSTEM

```
27 |         if (confirmation == 'y' || confirmation == 'Y') {
28 |
29 |             int menu = displayMenu();
30 |             do {
31 |
32 |                 // Now you can use the 'menu' variable as needed
33 |                 switch (menu) {
34 |                     case 0:
35 |                         System.out.println("Exiting the program. Goodbye! \n");
36 |                         break;
37 |                     case 1:
38 |                         MenuList.viewItems();
39 |                         switchMenu();
40 |                         break;
41 |                     case 2:
42 |                         MenuList.sellItem(scanner);
43 |                         switchMenu();
44 |                         break;
45 |                     case 3:
46 |                         MenuList.restockItem(scanner);
47 |                         switchMenu();
48 |                         break;
49 |                     case 4:
50 |                         MenuList.viewShopBalance();
51 |                         switchMenu();
52 |                         break;
53 |                     case 5:
54 |                         MenuList.searchItem(scanner);
55 |                         switchMenu();
56 |                         break;
57 |                     case 6:
58 |                         MenuList.addItem(scanner);
59 |                         switchMenu();
60 |                         break;
61 |                     case 7:
62 |                         MenuList.deleteItem(scanner);
63 |                         switchMenu();
64 |                         break;
65 |                     default:
66 |                         System.out.println("Menu not available. Please choose menu from given list! \n");
67 |                         switchMenu();
68 |                     } break;
69 |             } while (menu != 0);
70 |
```

COMPUTER SHOP MANAGEMENT SYSTEM

```
71 |        } else if (confirmation == 'n' || confirmation == 'N') {
72 |            System.out.println("Menu not displayed. Exiting the program. Goodbye! \n");
73 |
74 |        } else {
75 |            System.out.println("Invalid input! \n");
76 |            switchMenu();
77 |        }
78 |
79 |        // Close the scanner
80 |        //scanner.close();
81 |    }
82 |
83 |    public static int displayMenu() {
84 |
85 |            // Menu Option Display
86 |            System.out.println(" ");
87 |            System.out.println("*************************************************");
88 |            System.out.println("*                MAIN MENU                  *");
89 |            System.out.println("*          0. Exit                        *");
90 |            System.out.println("*          1. Show Available Items         *");
91 |            System.out.println("*          2. Sell Item                    *");
92 |            System.out.println("*          3. Restock Item                 *");
93 |            System.out.println("*          4. Show Shop Balance            *");
94 |            System.out.println("*          5. Search Item                  *");
95 |            System.out.println("*          6. Add New Item                 *");
96 |            System.out.println("*          7. Delete Item                  *");
97 |            System.out.println("*************************************************");
98 |            System.out.print("Enter your menu: ");
99 |
100 |            Scanner menu = new Scanner(System.in);
101 |
102 |            // Validate that the user entered correct input
103 |            while (!menu.hasNextInt()) {
104 |                System.out.println("Invalid input please enter an integer! \n");
105 |                menu.next();
106 |            }
107 |            System.out.print("\n");
108 |            return menu.nextInt();
109 |    }
110 | }
```

COMPUTER SHOP MANAGEMENT SYSTEM

THE CODE - MENU LIST

```
 1 | import java.util.Arrays;
 2 | import java.util.Scanner;
 3 |
 4 | public class MenuList {
 5 |     // Assuming these arrays and variable are defined in the App class
 6 |     private static String[] items = { "mouse", "keyboard", "monitor", "desktop" };
 7 |     private static double[] itemPrices = { 20.0, 30.0, 150.0, 500.0 };
 8 |     private static int[] itemStocks = { 50, 30, 20, 10 };
 9 |     private static double shopBalance = 10000.0;
10 |
11 |     // Method for menu 1
12 |     public static void viewItems() {
13 |         System.out.println("==============================");
14 |         System.out.println("Menu 1 - Show Available Items");
15 |         System.out.println("==============================");
16 |         System.out.println("\nItems available in the shop: \n");
17 |         for (int i = 0; i < items.length; i++) {
18 |             String itemName = items[i];
19 |             double itemPrice = itemPrices[i];
20 |             int itemStock = itemStocks[i];
21 |             System.out.println(itemName + " - RM " + itemPrice + " | Stock: " + itemStock);
22 |         }
23 |     }
24 |
25 |     // Method for menu 2
26 |     public static void sellItem(Scanner scanner) {
27 |         System.out.println("==================");
28 |         System.out.println("Menu 2 - Sell Item");
29 |         System.out.println("==================");
30 |
31 |         for (String itemName : items) {
32 |             System.out.println(itemName);
33 |         }
34 |
35 |         System.out.print("\nEnter the item name to sell: ");
36 |         String itemName = scanner.next();
37 |         int itemIndex = Arrays.asList(items).indexOf(itemName);
38 |
39 |         if (itemIndex != -1 && itemStocks[itemIndex] > 0) {
40 |             double itemPrice = itemPrices[itemIndex];
41 |             itemStocks[itemIndex]--;
42 |             shopBalance += itemPrice;
43 |             System.out.println("Sold " + itemName + " for RM" + itemPrice);
44 |         } else {
45 |             System.out.println("Item not available or out of stock.");
```

COMPUTER SHOP MANAGEMENT SYSTEM

```
46:         }
47:     }
48:     //Method for menu 3
49:     public static void restockItem(Scanner scanner) {
50:         System.out.println("======================");
51:         System.out.println("Menu 3 - Restock Item");
52:         System.out.println("======================");
53:         System.out.print("\nEnter the item name to restock: ");
54:         String itemName = scanner.next();
55:         int itemIndex = Arrays.asList(items).indexOf(itemName);
56:
57:         if (itemIndex != -1) {
58:             System.out.print("Enter the quantity to restock: ");
59:             int quantity = scanner.nextInt();
60:             itemStocks[itemIndex] += quantity;
61:             double totalCost = itemPrices[itemIndex] * quantity;
62:             shopBalance -= totalCost;
63:             System.out.println("Restocked " + quantity + " " + itemName + " for RM" + totalCost);
64:         } else {
65:             System.out.println("Invalid item name.");
66:         }
67:     }
68:     //Method for menu 4
69:     public static void viewShopBalance() {
70:         System.out.println("============================");
71:         System.out.println("Menu 4 - View Shop Balance");
72:         System.out.println("============================");
73:         System.out.println("\nShop Balance: RM" + shopBalance);
74:     }
75:     //Method for menu 5
76:     public static void searchItem(Scanner scanner) {
77:         System.out.println("====================");
78:         System.out.println("Menu 5 - Search Item");
79:         System.out.println("====================");
80:         System.out.print("\nEnter the item name to search: ");
81:         String itemName = scanner.next();
82:
83:         int itemIndex = Arrays.asList(items).indexOf(itemName);
84:
85:         if (itemIndex != -1) {
86:             double itemPrice = itemPrices[itemIndex];
87:             int itemStock = itemStocks[itemIndex];
88:             System.out.println("Item found: " + itemName + " - Price: $" + itemPrice + " | Stock: " + itemStock);
89:         } else {
90:             System.out.println("Item not found.");
91:         }
```

COMPUTER SHOP MANAGEMENT SYSTEM

```
92:     }
93:     //Method for menu 6
94:     public static void addItem(Scanner scanner) {
95:         System.out.println("==================");
96:         System.out.println("Menu 6 - Add Item");
97:         System.out.println("==================");
98:
99:         System.out.print("Enter the new item name: ");
100:         String newItemName = scanner.next();
101:
102:         int newItemIndex = Arrays.asList(items).indexOf(newItemName);
103:
104:         if (newItemIndex != -1) {
105:             System.out.println("Item already exists. Please use the update item option.");
106:             return;
107:         }
108:
109:         System.out.print("Enter the price for " + newItemName + ": RM");
110:         double newItemPrice = scanner.nextDouble();
111:
112:         System.out.print("Enter the initial stock for " + newItemName + ": ");
113:         int newItemStock = scanner.nextInt();
114:
115:         // Assuming items, itemPrices, and itemStocks are defined in MenuList class
116:         items = Arrays.copyOf(items, items.length + 1);
117:         itemPrices = Arrays.copyOf(itemPrices, itemPrices.length + 1);
118:         itemStocks = Arrays.copyOf(itemStocks, itemStocks.length + 1);
119:
120:         items[items.length - 1] = newItemName;
121:         itemPrices[itemPrices.length - 1] = newItemPrice;
122:         itemStocks[itemStocks.length - 1] = newItemStock;
123:
124:         System.out.println(
125:                 "Added new item: " + newItemName + " | Price: RM" + newItemPrice + " | Initial Stock: " +
newItemStock);
126:     }
127:     //Method for menu 7
128:     public static void deleteItem(Scanner scanner) {
129:         System.out.println("====================");
130:         System.out.println("Menu 7 - Delete Item");
131:         System.out.println("====================");
132:         System.out.print("\nEnter the item name to delete: ");
133:         String itemName = scanner.next();
134:
135:         int itemIndex = Arrays.asList(items).indexOf(itemName);
136:
137:         if (itemIndex != -1) {
138:             // Remove the item from arrays
```

COMPUTER SHOP MANAGEMENT SYSTEM

```
            items = removeElement(items, itemIndex);
            itemPrices = removeElement(itemPrices, itemIndex);
            itemStocks = removeElement(itemStocks, itemIndex);

            System.out.println(itemName + " has been deleted.");
        } else {
            System.out.println("Item not found!!! \n");
            deleteItem(scanner);
        }
    }

    private static String[] removeElement(String[] array, int index) {
        String[] newArray = new String[array.length - 1];
        System.arraycopy(array, 0, newArray, 0, index);
        System.arraycopy(array, index + 1, newArray, index, array.length - index - 1);
        return newArray;
    }

    private static double[] removeElement(double[] array, int index) {
        double[] newArray = new double[array.length - 1];
        System.arraycopy(array, 0, newArray, 0, index);
        System.arraycopy(array, index + 1, newArray, index, array.length - index - 1);
        return newArray;
    }

    private static int[] removeElement(int[] array, int index) {
        int[] newArray = new int[array.length - 1];
        System.arraycopy(array, 0, newArray, 0, index);
        System.arraycopy(array, index + 1, newArray, index, array.length - index - 1);
        return newArray;
    }
}
```

COMPUTER SHOP MANAGEMENT SYSTEM

THE SAMPLE RUN - MANUAL

Welcome page

```
**********************************************
*                                            *
*  Welcome to Computer Shop Management System  *
*                                            *
**********************************************


Do you want to show Main Menu? (y/n)
```

Main menu - input (Y)

```
**********************************************
*                  MAIN MENU                 *
*           0. Exit                          *
*           1. Show Available Items          *
*           2. Sell Item                     *
*           3. Restock Item                  *
*           4. Show Shop Balance             *
*           5. Search Item                   *
*           6. Add New Item                  *
*           7. Delete Item                  *
**********************************************
Enter your menu:
```

1. Show Available Items - input (1)

```
==============================
Menu 1 - Show Available Items
==============================

Items available in the shop:

mouse - RM 20.0 | Stock: 50
keyboard - RM 30.0 | Stock: 30
monitor - RM 150.0 | Stock: 20
desktop - RM 500.0 | Stock: 10

Do you want to show Main Menu? (y/n)
```

COMPUTER SHOP MANAGEMENT SYSTEM

2. Sell Item - Input (2) (User required to put item name eg:mouse)

```
==================
Menu 2 - Sell Item
==================
mouse
keyboard
monitor
desktop

Enter the item name to sell:
```

2. Item successfully sold - input (user item choice)

```
Enter the item name to sell: mouse
Sold mouse for RM20.0
```

2. Items unavailable/unsuccessfully sold - input (user error input)

```
Enter the item name to sell: headphone
Item not available or out of stock.
```

3. Restock Item - Input (3) (User required to put item name, quantity eg:keyboard,1)

```
Enter your menu: 3

====================
Menu 3 - Restock Item
====================

Enter the item name to restock: keyboard
Enter the quantity to restock: 1
```

3. Restock Item Successfully (left)/ unsuccessfully (right) - input (user item choice unavailable)

```
Enter your menu: 3

====================
Menu 3 - Restock Item
====================

Enter the item name to restock: keyboard
Enter the quantity to restock: 1
Restocked 1 keyboard for RM30.0
```

```
Enter your menu: 3

====================
Menu 3 - Restock Item
====================

Enter the item name to restock: WIFI
Invalid item name.
```

COMPUTER SHOP MANAGEMENT SYSTEM

4. Show Shop Balance- Input (4) (show shop balance base on declared balance earlier)

```
Enter your menu: 4

=========================
Menu 4 - View Shop Balance
=========================

Shop Balance: RM10000.0
```

5. Search Item - Input (5) (User required to put item name base on base on available item)

```
Enter your menu: 5

====================
Menu 5 - Search Item
====================

Enter the item name to search:
```

5. Search Item Successfully - input (user item choice)

```
Enter your menu: 5

====================
Menu 5 - Search Item
====================

Enter the item name to search: desktop
Item found: desktop - Price: $500.0 | Stock: 10
```

5. Unavailable unsuccessfully - input (user item choice)

```
Enter your menu: 5

====================
Menu 5 - Search Item
====================

Enter the item name to search: john
Item not found.
```

COMPUTER SHOP MANAGEMENT SYSTEM

6. Add new item- Input (6) (let user to add new item to the system, user required to put item name, price, and initial stock)

```
Enter your menu: 6

=================
Menu 6 - Add Item
=================
Enter the new item name: cooler
Enter the price for cooler: RM25
Enter the initial stock for cooler: 5
```

6. Item Added Successfully - input (user item choice) (item added available item and search list)

```
Enter your menu: 6

=================
Menu 6 - Add Item
=================
Enter the new item name: cooler
Enter the price for cooler: RM25
Enter the initial stock for cooler: 5
Added new item: cooler | Price: RM25.0 | Initial Stock: 5
```

```
Items available in the shop:

mouse - RM 20.0 | Stock: 50
keyboard - RM 30.0 | Stock: 30
monitor - RM 150.0 | Stock: 20
desktop - RM 500.0 | Stock: 10
cooler - RM 25.0 | Stock: 5
```

COMPUTER SHOP MANAGEMENT SYSTEM

7. Delete item- Input (7) (let user delete item in available item, user required to put item name from available item list)

```
Enter your menu: 7

====================
Menu 7 - Delete Item
====================

Enter the item name to delete:
```

7. Item deleted Successfully - input (user item choice)

```
Enter your menu: 7

====================
Menu 7 - Delete Item
====================

Enter the item name to delete: cooler
cooler has been deleted.
```

7. Item delete unsuccessfully - input (user item choice not from listed)

```
Enter your menu: 7

====================
Menu 7 - Delete Item
====================

Enter the item name to delete: fan
Item not found.
```

8. Exiting the program - input (0) (let user exit the program)

```
Enter your menu: 0

Exiting the program. Goodbye!
```

COMPUTER SHOP MANAGEMENT SYSTEM

END