



Bahria University
Discovering Knowledge

BAHRIA UNIVERSITY KARACHI CAMPUS

DEPARTMENT OF COMPUTER ENGINEERING

Computer Programming

CSL-113

Project Title

Store Management System

By

Dawood Shehzad

02-136221-034

Bilal Ahmed

02-136221-028

Ali Saim

02-136221-011

List of Contents

Project Title	0
Introduction.....	2
Acknowledgement	2
Abstract	2
Problem & Solution	2
Overview of Project.....	3
Workflow	4
Output.....	5
Conclusion	6
References.....	6

Introduction

The programming language used for this program is C++ and the compilers used are Microsoft Visual Studio and Visual Studio Code. This program can be used to shop for fruits, snacks and vegetables which are available in the store. It can also be easily modified to add more categories and to edit item prices.

Customer data is saved onto text files so that it doesn't perish after the program has terminated.

Acknowledgement

We would like to extend our gratitude and appreciation for the Course and Lab Instructors for providing us with knowledge and intellect which resulted in the development of this program. It also wouldn't have been easily possible without the help of our friends who used their experience and knowledge to work-out solutions for different bugs and problems. Not forgetting all those random people that provided solutions for problems and uploaded code snippets and on the internet to help others.

Abstract

This store management system is a comprehensive retail management system made for a lower level general store. It provides a computerized solution to basic shop operations like maintaining customer records and data, keeping inventory, introducing deals and discounts. A Point-of-sale system is also included to accommodate store operations. It can not rival big commercial solutions, our target is to provide a small, inexpensive, customized solution with basic functionalities.

2

Problem & Solution

The client is in need of a cheap and easy to use solution for their store. The listed requirements state that there should be an option for the customers to Sign In to their accounts, if it's a new customer then they can be referred to the Sign Up section, there should be an Admin login section as well that can be used to manage customer data and other stuff.

After a successful signup, the user should be brought to the store home screen, this screen has all the categories/sections currently included in the software, for starters there should be fruit, snacks, game and vegetable sections. The user can select any section by entering the numeric value of that section.

All sections should contain a "Total Amount" line which has the total bill printed in front of it.

After the user is done with the shopping, they can head to the Checkout section where they are able to see the amount of items they purchased along with the total bill, there is also an option to enter a discount code which if correct, grants 20% off of the whole order, and these discount codes should be manageable through the admin panel.

Overview of Project

The first run of this program takes you to the login Screen when you can Sign In or Sign Up for a new account, it will ask for your full name, the username you want to signup with, and the account password. The program stores this information in separate files so we can retrieve them after the program terminates.

Now you have an account, you will be redirected to the Sign in screen, where you can use the set username and password combination to log into your account, the password you type is automatically converted to asterisks '*' and is then printed on the screen to give an impression of input, the Backspace key can be used to delete if a wrong character is typed by accident, the Enter key can be used to lock your input and save the username and password to a text file named 'customer_passwords.txt'. Similarly, customer names are stored in a different text file named 'customer_names.txt'. Once an account has been created, you are automatically redirected to the Sign In screen where you can use those credentials to log into your account.

If it's a staffer running the program, and their intention is to access the admin panel to manage customer data, they can choose the Admin Sign In module on the Login Screen, they can either use the hard-coded username & password, which is 'admin' & 'admin', or they can enter the specified password set for the admins which is present in a text file named 'admin.txt' (admin password should be set manually).

After login in, they will be able to see the Admin Panel where they can delete the passwords, names and usernames of all the customers, there is another option which lets you add more discount codes that can be used when checking out.

After the Log Ins, there appears the store homepage screen with all the item categories available, the user can select a category using numeric values, the first 3 categories (Fruits, Vegetables, Snacks) have the same functionality, the fourth category (Games) is just a casual add-on.

3

Lets say the user chooses the Fruits category, it will print the available item list along with the item prices, the user can select an item using numeric values, the program will then ask the user about the quantity they want to buy of that item, what ever that is, is multiplied with the price of that item, and the result is automatically added to the 'Total Amount' statement under the item list. This process repeats until the user decides to go back to the home page by entering '0', from there, a different category can be selected or the check-out section can be accessed by entering another '0'.

In the Checkout section, the program shows the Total Bill of that customer, along with the amount of items bought from the store, Lastly, there are 2 options to choose from, Discount Code and Checkout.

After Checking out, you are shown a message of appreciation, and the credits appear, before the program finally ends

Workflow

1. Login Section

- Customer Sign-Up
- Customer Sign-In
- Admin Sign-In
- Exit

2. Admin Panel

- Add New Customer
- Delete Customer Data
- Manage Discount Codes
- Exit

3. Store Home

- Fruits Section
- Vegetables Section
- Snacks Area
- Game Section
- Exit

4. Fruits/Veg/Snacks Sections

- Select Item
- Choose Item Quantity
- Repeat above 2 steps
- Exit

5. Game Section

- Word Scramble
- Rainbow Writing
- Exit

6. Checkout Section

- Total Amount / Items
- Checkout
- Discount Code Input
- Credits
- End

Output

```

1  #include <iostream> // Headers //
2  #include <string>
3  #include <windows.h>
4  #include <fstream>
5  #include <conio.h>
6  using namespace std; //
7
8  void buffer(); // Function Declarations //
9  void textColor(int);
10 string getPass();
11 void login();
12 void adminSignin();
13 void adminPanel();
14 void signIn();
15 void signUp();
16 void storeSection();
17 void fruitSec(int*);
18 void vegSec(int*);
19 void snackSec(int*);
20 void games(int*);
21 void wordScramble();
22 void wordColor();
23 void rainbowWrite(string input);
24 void checkOut();
25 bool checkCode();
26 void credits(); //
27
28 // Global Variables //
29 int Total = 0, totalItems = 0; // Initializing Total valuyes to 0
30 int* Totalptr = &Total; // To be able to update Total bill using pointers
31 const string clearLine = "\r"; // 'r' makes the cursor retreat to the line's start
32

```

| Store Database Management System Program |

```
----- Sign-in / Sign-up Screen -----
```

```
1 - Customer Sign-in
2 - Customer Sign-up
3 - Admin Sign-in
0 - Exit Program
```

```

32 int main() // Main
33 {
34     cout << "\t\t\t\t\t ----- \n";
35     cout << "\t\t\t\t\t | Store Database Management System Program | \n";
36     cout << "\t\t\t\t\t ----- \n";
37
38     login(); // First
39
40     storeSection(); // Runs after Login is successful
41
42     credits(); // Runs after Checkout
43     return 0;
44 }//
45
46 void buffer() // Loading Animation Effect
47 {
48     cout << ".";
49     Sleep(700);
50     cout << ".";
51     Sleep(700);
52     cout << ".";
53     Sleep(700);
54     cout << endl;
55 }//
56
57 void textColor(int color) // Console Color Manipulation
58 {
59     HANDLE console_color;
60     console_color = GetStdHandle(STD_OUTPUT_HANDLE);
61     SetConsoleTextAttribute(console_color, color + 8);
62 }//
63
64 void login() // Login
65 {
66     int login;
67     textColor(3);
68     system("cls");
69     cout << "\n\t\t\t\t\t Sign-in / Sign-up Screen ----- \n\n";
70
71     cout << "\t\t\t\t\t 1 - Customer Sign-in \n";
72     cout << "\t\t\t\t\t 2 - Customer Sign-up \n";
73     cout << "\t\t\t\t\t 3 - Admin Sign-in \n";
74     cout << "\t\t\t\t\t 0 - Exit Program \n";
75
76     cin >> login;
77     system("cls");
78
79     switch (login)
80     {
81     case 1:
82         signIn();
83         break;
84     case 2:
85         signUp();
86         break;
87     case 3:
88         adminsignin();
89         break;
90     case 0:
91     {
92         textColor(4); // Red
93         cout << "\nExiting Program";
94         buffer();
95         textColor(3); // Cyan
96     }
97     }
98 }//

```

```

149
150 void signUp()    // Customer Sign-Up
151 {
152     cin.ignore();
153     string name, user, pass;
154
155     ofstream names("customer_names.txt", ios::app);
156     ofstream write("customer_passwords.txt", ios::app);
157
158     system("cls");
159     cout << "\n\t ----- Customer Signup ----- \n\n";
160     cout << "\t\t Please enter your name: ";
161     getline(cin, name);
162
163     cout << "\t\t Set Username: ";
164     cin >> user;
165     cout << "\t\t Set Password: ";
166     pass = getPass();
167
168     cout << "\nCreating Account";
169     buffer();
170
171     names << name << endl; // Write in the customer
172     write << user << " ";
173     write << pass << endl;
174
175     textColor(2);
176     cout << "\nAccount created successfully!\n\n";
177     Sleep(500);
178
179     names.close();
180     write.close();
181     textColor(3);
182     signIn();
183 }//
184
185
186 void signIn()    // Customer Sign-In
187 {
188     ifstream names("customer_names.txt");
189     ifstream read("customer_passwords.txt");
190
191     struct customerDetails
192     {
193         string name, user, pass;
194         string user1, pass1;
195     } Customer;
196
197     Login:
198     system("cls");
199     cout << "\n\t\t ----- Customer Login ----- \n\n";
200
201     cout << "\t\t Enter username: ";
202     cin >> Customer.user;
203     cout << "\t\t Enter password: ";
204     Customer.pass = getPass();
205
206     nextCustomer:
207     getline(names, Customer.name); // Reads First Line
208     read >> Customer.user1;        // Reads 'first' word of the second line
209     read >> Customer.pass1;        // Reads 'second' word of the second line
210
211     if (Customer.user != Customer.user1)
212         goto nextCustomer;
213
214     if (Customer.user == Customer.user1 && Customer.pass == Customer.pass1)
215     {
216         cout << endl << "Signing In";
217         buffer();
218         textColor(2); // Lime
219
220         cout << "\n\t Login Successful!";
221         cout << "\n\t Welcome " << Customer.name << "!";
222         Sleep(2000);
223         system("cls");
224     }
225     else
226     {
227         textColor(4);
228         cout << "\nInvalid Credentials!\n";
229         goto Login;
230     }
231
232     read.close(); // Close file
233     names.close(); // Close file
234 }//

```

```

----- Customer Signup -----

Please Enter your Full Name: Full Name
Set Username: username
Set Password: *****

Creating Account...

```

6

```

----- Customer Login -----

Enter username: username
Enter password: *****

Signing In...

Login Successful!
Welcome Full Name!

```

```
----- Admin Login -----
Enter '0' to go back

Username: admin
Password: *****
Logging In...
Login successful!
```

```

Admin Panel
-----
1 - Add New Customer
2 - Purge Customer Data
3 - Add Discount Codes

0 - Exit

```



```

273 // ***** Store Home Page with categories ***** //
274 void storeSection()
275 {
276     int select;
277     system("cls");
278     textColor(3);
279
280     cout << "\n----- Store Sections ----- \n\n";
281
282     cout << "\t 1 - Fruits          \n";
283     cout << "\t 2 - Vegetables       \n";
284     cout << "\t 3 - Snacks              \n";
285     cout << "\t 4 - Games                \n";
286     cout << "\t 0 - Exit Program / Check-Out \n";
287
288     textColor(6);
289     cout << "\n\nNet Total = Rs. " << Total << endl;
290     textColor(3);
291
292     cin >> select;
293     system("cls");
294
295     switch (select)
296     {
297     case 1:
298         fruitSec(Totalptr);
299         break;
300     case 2:
301         vegSec(Totalptr);
302         break;
303     case 3:
304         snackSec(Totalptr);
305         break;
306     case 4:
307         games(Totalptr);
308         break;
309     case 0:
310     {
311         textColor(4);
312         cout << "\nChecking Out";
313         buffer();
314         textColor(3);
315         system("cls");
316         checkOut();
317     }
318     default:
319         cout << "wrong inp\n";
320     }
321 }
322

```

```

----- Store Sections -----

1 - Fruits
2 - Vegetables
3 - Snacks
4 - Games
0 - Exit Program / Check-Out

```

Net Total = Rs. 0

```

323 // ***** Fruits Section ***** //
324 void fruitSec(int* Total)
325 {
326     // Setting item prices and variables (q_ = quantity of)
327     int app = 280, ban = 200, man = 350, gra = 130;
328     int q_a = 0, q_b = 0, q_m = 0, q_g = 0;
329     int sumFruit, numFruit, choice;
330
331     Fruits:
332     system("cls");
333     cout << "\t          Fruits Area\n";
334     cout << "\t _____ \n\n";
335
336     cout << "\t 1 - Banana      |   Rs 200 / Dozen \n";
337     cout << "\t 2 - Grapes       |   Rs 130 / KG   \n";
338     cout << "\t 3 - Apples        |   Rs 280 / KG   \n";
339     cout << "\t 4 - Mangoes       |   Rs 350 / KG   \n";
340
341     cout << "\n\t 0 - Exit          " << endl;
342     textColor(6);
343     cout << "\n\nNet Total = Rs. " << *Total << " \n\n";
344     textColor(3);
345     cin >> choice;
346
347     switch (choice)
348     {
349     case 1:
350         cout << "How many Dozens of Bananas?: ";
351         cin >> q_b;
352         break;
353     case 2:
354         cout << "How many KGs of Grapes?: ";
355         cin >> q_g;
356         break;
357     case 3:
358         cout << "How many KGs of Apples?: ";
359         cin >> q_a;
360         break;
361     case 4:
362         cout << "How many KGs of Mangoes?: ";
363         cin >> q_m;
364         break;
365     case 0:
366         storeSection();
367         return;
368         break;
369     default:
370         cout << "Invalid Input! Please enter again \n";
371         Sleep(1000);
372         goto Fruits;
373         break;
374     }
375
376     numFruit = q_a + q_b + q_m + q_g;
377     sumFruit = q_a * app + q_b * ban + q_m * man + q_g * gra; // Calculating sum
378     q_a = 0, q_b = 0, q_m = 0, q_g = 0; // Flushing values after storing in sum
379
380     // Adding the sum of vegetables bought to the global 'Total' variable
381     *Total = *Total + sumFruit;
382     totallItems += numFruit;
383     goto Fruits;
384     // Going back to the label 'Fruits'
385 }
386

```

Fruits Area

```

1 - Banana      |   Rs 200 / Dozen
2 - Grapes       |   Rs 130 / KG
3 - Apples        |   Rs 280 / KG
4 - Mangoes       |   Rs 350 / KG

```

0 - Exit

Net Total = Rs. 0

1
How many Dozens of Bananas?: 2

Net Total = Rs. 400

```
void vegSec(int* Total)
```

```
1 - Potato      |      Rs 55 / KG
2 - Tomato      |      Rs 85 / KG
3 - Carrot      |      Rs 65 / KG
4 - Onion       |      Rs 70 / KG

0 - Exit
```

4
How many KGs of Onions?: 1

```
// ***** Snacks Area ***** //
```

```
void snackSec(int* Total)
```

```

1 - Lays      | Rs 80
2 - Samosa    | Rs 25
3 - Chaat     | Rs 85 / Plate
4 - Biscuits  | Rs 20 / Packet

0 - Go Back

```

```
1
How many Packets of Lays?: 1
```

9

```

510
511 // ***** Games Area ***** //
512 void games(int* Total)
513 {
514     int choice;
515     system("cls");
516     cout << "\t\t\t\t\t Gaming Area\t\t\t\t\t \n";
517     cout << "\t\t\t\t\t _____\t\t\t\t\t \n\n";
518
519     cout << "\t\t\t 1 - Word Scramble | Free\t\t\t\t\t \n";
520     cout << "\t\t\t 2 - Rainbow Writing | Free\t\t\t\t\t \n\n";
521     cout << "\t\t\t 0 - Go Back\t\t\t\t\t \n\n";
522     textcolor(6);
523     cout << "\n\nNet Total = Rs. " << *Total << "\n\n";
524     textcolor(3);
525     cin >> choice;
526
527     switch (choice)
528     {
529     case 1:
530         wordScramble();
531         break;
532     case 2:
533         wordColor();
534         break;
535     case 0:
536         textcolor(4);
537         cout << "\nExiting Section";
538         buffer();
539         storeSection();
540     }
541 }
542 // ***** Letter Randomizer ***** //
543 void wordScramble()
544 {
545     char choice;
546     string word;
547
548     wordInput:
549     system("cls");
550     cout << "\t\t\t\t\t Playing Word Scramble\t\t\t\t\t \n";
551     cout << "\t\t\t\t\t _____\t\t\t\t\t \n\n";
552
553     cout << "\t\t\t Enter a word: ";
554     cin >> word;
555
556     cout << "\n Scrambling";
557     buffer();
558
559     for (int i = 0; i < word.length(); i++)
560     {
561         for (int j = i + 1; j < word.length(); j++)
562         {
563             if (word[i] >= word[j])
564             {
565                 // Basic Bubble-sort technique
566                 swap(word[i], word[j]);
567                 swap(word[i + 1], word[j - 1]);
568             }
569         }
570     }
571
572     cout << " Scrambling Complete!\t\t\t\t\t \n\n ";
573
574     // Printing the scrambled output
575     textcolor(2);
576     for (int i = 0; i < word.length(); i++)
577         cout << word[i]; //
578
579     textcolor(3);
580     cout << "\n\n Scramble a new word? (y / n)\t\t\t\t\t \n";
581     cin >> choice;
582
583     if (choice == 'y')
584         goto wordInput;
585     else
586         games(Totalptr);
587 }
588

```

Gaming Area

```

1 - Word Scramble | Free
2 - Rainbow Writing | Free

0 - Go Back

```

Net Total = Rs. 500

10

Playing Word Scramble

Enter a word: Scramble

Scrambling...

Scrambling Complete!

Sabelmcr

Scramble a new word? (y / n)

n

Playing Rainbow Write

Enter a word to be colored: Rainbow

```
----- Color List -----
```

```
0 - Gray
1 - Dark Blue
2 - Lime
3 - Cyan
4 - Red
5 - Magenta
6 - Yellow
7 - White
8 - Rainbow
```

Choose a color you want the output in:

```
Choose a color you want the output in: 4
Coloring the output...
```

Rainbow

```
Choose a different color? (y / n)
n
```

```

655
656 void rainbowWrite(string input)
657 {
658     cout << endl;
659
660     for (int i = 0; i < 4; i++)          // 4 x 8 Colors
661     {
662         for (int P = 0; P < 8; P++)    // 8 Colors, one by one
663         {
664             //SetConsoleTextAttribute(console_color, P + 8);
665             textColor(P);
666
667             cout << input;
668             Sleep(300);
669             cout << clearLine;
670         }
671     }
672     buffer();
673     system("cls");
674 }//
675

```

```

677
678 // ***** Read Password input, display it as ' * ' ***** //
679 string getPass()
680 {
681     int i;
682     char pass[25];
683     string password;
684
685     for (i = 0; i < 25; i++)
686     {
687         pass[i] = _getch();          // _getch(); takes live input from keyboard
688
689         if (pass[i] == 13)           // '13' is the ASCII code of Enter key
690             break;                  // Loop breaks when enter is pressed
691         else if (pass[i] == 8){      // '8' is the ASCII code of Backspace key
692             cout << "\b \b";        // Aesterik is removed and the cursor retreats
693             i--;                    // Array is decremented
694             i--;                    // To delete the backspaced character
695         }
696         else
697             cout << " * ";          // If the input is not 'Enter', print *
698     }
699
700     pass[i] = '\0';                 // Terminating string after storing password with null
701     password = pass;                // pasting the character array to string variable
702
703     return password;
704 }//
705

```

```
702 // ***** Checkout Section *****  
703  
704 void checkOut()  
705 {  
706     string code;  
707     int choice, discount, flag = 0;  
708     system("cls");  
709     cout << "\t\t\t\t\t Check Out Section\n";  
710     cout << "\t\t\t\t\t _____\n\n";  
711  
712     textColor(6);  
713     cout << "\t No. of Items bought = " << totalItems << "\n\n";  
714     cout << "\t Total Bill = Rs. " << Total << "\n\n";  
715     textColor(3);  
716  
717     cout << "\t\t 1 - Enter Discount Code\n";  
718     cout << "\t\t 2 - Check Out\n";  
719     cin >> choice;  
720  
721     if (choice == 1)  
722         flag = checkCode();  
723     else {  
724         cout << "\n\t Please pay Rs. " << Total << " on the counter \n\n";  
725         system("pause");  
726         textColor(7); // White  
727         cout << "\n\t Thankyou for giving us business, Please come again! \n\n";  
728     }  
729  
730     // If the discount code was valid  
731     if (flag == 1)  
732     {  
733         discount = Total * 20 / 100;  
734         Total = Total - discount;  
735  
736         cout << "\n\t You have availed a discount of Rs. " << discount;  
737         cout << "\n\t Please pay Rs. " << Total << " on the counter \n\n";  
738         system("pause");  
739         textColor(7); // White  
740         cout << "\n\t Thankyou for giving us business, Please come again! \n\n";  
741     }  
742     Sleep(1500);  
743 }  
744
```

Check Out Section

No. of Items bought = 4

Total Bill = Rs. 500

- 1 - Enter Discount Code
- 2 - Check Out

13

```
745 // ***** Discount Code Checker ***** //
```

```
746
```

```
747 bool checkCode()
```

```
748 {
```

```
749     string code, discount;
```

```
750     ifstream read("discount_codes.txt");
```

```
751     system("cls");
```

```
752
```

```
753     cout << "\n\t          Discount Code          \n";
```

```
754     cout << " \t _____ \n\n";
```

```
755
```

```
756     cout << "\n\t Enter Discount code: ";
```

```
757     cin >> code;
```

```
758     cout << "\nChecking";
```

```
759     buffer();
```

```
760
```

```
761 nextCode:
```

```
762     getline(read, discount);
```

```
763
```

```
764     if (code == discount)
```

```
765     {
```

```
766         textColor(2);
```

```
767         cout << "\n\t Congratulations! you are eligible for a 20% discount on your bill" << endl;
```

```
768         textColor(3);
```

```
769         return true;
```

```
770     }
```

```
771     // If The code doesn't match & the file comes to an end
```

```
772     else if (read.eof() == 1) {
```

```
773
```

```
774         cout << "\t Discount code Invalid!";
```

```
775         Sleep(1000);
```

```
776         return false;
```

```
777     }
```

```
778     else
```

```
779         goto nextCode;
```

```
780 }//
```

```
781
```

Discount Code

Enter Discount code: free

Checking...

Congratulations! you are eligibl

You have availed a discount of R
Please pay Rs. 400 on the counte

Thankyou for giving us business

Discount Code

Enter Discount code: free20

Checking...

Congratulations! you are eligible for a 20% discount on your bill

You have availed a discount of Rs. 100
Please pay Rs. 400 on the counter

Thankyou for giving us business, Please come again!

```

781
782 // ***** CREDITS ***** //
783 void credits()
784 {
785     system("cls");
786     textcolor(3);
787     cout << "\n\t\t ----- Credits ----- \n";
788     cout << "\t\t          Project by          \n\n\n";
789
790     textcolor(7);
791     cout << "\t\t Dawood shahzad ----- 02-136221-034" << " \n";
792     textcolor(6);
793     cout << "\t\t Ali Saim ----- 02-136221-011" << " \n";
794     textcolor(2);
795     cout << "\t\t Bilal Ahmed ----- 02-136221-028" << " \n\n";
796     textcolor(3);
797
798     cout << "\n\n\t\t ----- Thankyou ----- \n\n\t\t\t";
799     system("pause");
800 }

```

```

----- Credits -----
          Project by

Dawood shahzad ----- 02-136221-034
Ali Saim ----- 02-136221-011
Bilal Ahmed ----- 02-136221-028

----- Thankyou -----

```

Name

- admin.txt
- customer_names.txt
- customer_passwords.txt
- discount_codes.txt

admin.txt - Notepad

File Edit Format View Help

admin
admin123

discount_codes.txt - Notepad

File Edit Format View Help

free20

customer_names.txt - Notepad

File Edit Format View Help

Dawood Shahzad
Ali Saim
Bilal Ahmed

customer_passwords.txt - Notepad

File Edit Format View Help

dawood.savage savage123
saimography saim321
lalib 9090

Conclusion

By implementing this Store management system, we are able to help many small businesses and stores get more flexibility for their users.

References

[Stack Overflow](#)

[W3Schools](#)

[Theghost753](#)

:----- *Thankyou* -----: