Technical Manual

1. Front Screen
2. Sign UP Screen
3. Customer Window
4. Retailer Windows
5. Manager
6. Admin
7. Front Screen : -

* User variable is for the type of the user .. , database class is to connect the database with the software , containing all function of the database. Password is to store the password of the User He enterd. SignUp class is for the user who dint register yet.
* private : int User;
* private : database^ user;
* String^ password;
* private : SignUp^ ss;
  1. Four picture box are representing all the type of user , customer, retailer, manager, admin; these four function are handling the clicking event .. The user chooses his/her type by clicking on them.
* private: System::Void pictureBox1\_Click(System::Object^ sender, System::EventArgs^ e)
* private: System::Void pictureBox2\_Click(System::Object^ sender, System::EventArgs^ e)
* private: System::Void pictureBox3\_Click(System::Object^ sender, System::EventArgs^ e)
* private: System::Void pictureBox4\_Click(System::Object^ sender, System::EventArgs^ e)
  1. The Log\_button Clicking Event is Coded In The Function Given Below.. After Clicking On Login Button the Programme check for the user name and password in the database .. if matches it opens the next windows otherwise pop an error….
* private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

1. Sign Up Screen :-
   1. The Sign Up Button coding of the Front Screen lies in the fuction given below in the form1.
      1. private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)
   2. The Four Text Field of the sign up screen are for the value corresponding to each label in the Sign Up form. All the Coding of the Submit Button lies in function describe below in the SignUp form.
      1. private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)
      2. The String^ user\_type is for the customer,reatailer.
2. Customer Window:-
   1. There are tree tabs in The Main Window Buy Items, Previous Bills, Profile.
      1. Buy Items Tab:-
         1. The items list box presented on the left side of the tab is for showing the items presenet in the database/ which the user or retailer can select to buy/sell. All the coding lies in the function.

private: System::Void listBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)

* + - 1. The Quantity Meter is set in the bottom which is to select the quantity of the item selected in the item list box. As soon as the customer clicks on the Set Quantity button the item is added to the table1. Which is the bill being generated at the run time.

The coding of the set quantity button lies in the function describe below

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

* + - 1. The Table1 present in Right Side of the layout is for the Bill Customer Wants to Generates. Customer Add and Remove the Items from the table1. To remove the item from the table user clicks on the button Remove Item present at the right bottom part of the tab. Coding of remove button is present in the function given below.

private: System::Void Remove\_item\_Click(System::Object^ sender, System::EventArgs^ e)

* + - 1. The Place Order Button is to Pass the Bill.. As Soon As the user Click on It the Bill is sent to the Manager Of the shop.

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)

* + - 1. The Right Bottom Table Represents the Total Distinct Item and Total Value of the Bill.
    1. Previous Bill:-
       1. The Left Table of the Tab Shows The Previous Bill Dealt by the Customer, with the id corresponding to them. All the coding to get the bills from the database lies in the function given below.

private: System::Void Panel\_Main\_Selected(System::Object^ sender, System::Windows::Forms::TabControlEventArgs^ e)

* + - 1. The coding for selected Bill to be shown in the table2 present at the right side of the bill is described in the function given below.

private: System::Void Bills\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)

* + 1. Profile tab : - The Four Labels are to show the name of the user and user\_id of him. The Log out Button is coded in the function described below.

private: System::Void logout\_Click(System::Object^ sender, System::EventArgs^ e)

The Change Password Button Coding Is given in the Function described below.

private: System::Void button2\_Click\_1(System::Object^ sender, System::EventArgs^ e)

4. Retailer Window: - The Profile Tab of the Retailer is same as the Customer. The Only Difference is to Set the Price of the Item in Place Order Tab. And Modified Total\_Table of the Retailer in the Previous Bill Tab. All the Function of Retailer Are Same (Customer Buy Item -> Retailer Place Order and Customer Previous Bill -> Retailer->Previous Bill and Customer Profile tab -> Retailer Profile->Tab)

Just Few Features are added like price\_per\_unit numeric up down, which is for the retailer to set the price of the item at which he wants to sell the item.

5/ Manager Window:

1. Status of Objects available in Stock

private: System::Void Manager\_Load(System::Object^ sender, System::EventArgs^ e)

When manger form will loaded this function will draw the chart for items that are available in stock and use the query

MySqlDataReader^ data = dat->fill\_chart();

This query will receive the items from stock table and items min max values from item table then min ,max ,quantity will be converted into percentage and series MinMax is representing the min-max values in percentage and an another series Quantity is representing original stock available.

Chart used here is Range Bar Chart that will help manager which object should be purchased and when purchased that will lead to MAXIMUM PROFIT!

1. Retailer orders

Here we have used one list box , one data grid view and two buttons to accept or reject the orders.

private: System::Void fill\_itemsinlist()

this function is used for filling the user name which have placed any order. Same Users name will appear that many times he have placed order and if any order is rejected or accepted then then user name corresponding to that order will be removed from list box.

orderlist = new int[sizeoflist];

this integer array will store the order id when list box is filled and will be updated if any order get accepted or rejected.

fill\_list();

this query is used to fill the list box and it will get the order id and user name then all the order id’s will be stored in orderlist array.

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)

these button will update the review status of the orders placed in list box and again refresh the list box according to accepted or rejected orders

dat->changereview(orderlist[selectedIndex],"accepted");

corresponding query that will change the review status of the order whether it is accepted or rejected

private: System::Void listBox1\_SelectedIndexChanged\_3(System::Object^ sender, System::EventArgs^ e)

this function is invoked when the item/user selected in list box is changed then corresponding orders that are associated with that order id will be displayed in grid box

MySqlDataReader^ data = dat->find\_griduserid(curItem);

This query will get the user id from the given user name

MySqlDataReader^ data1 = dat1->get\_orders(user\_id);

Get the order id from database corresponding to related user id

MySqlDataReader^data2=dat2>order\_detais(System::Convert::ToInt32(orderlist[selectedindex])); this query will fetch the order details from the database and take an argument as order id that is stored in orderlist array

Label2 will display the total cost corresponding to selected order id!

(3) Changes

This tab have two functionality

To change the maximum and minimum values

private: System::Void comboBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)

this will set the all ready filled maximum and minimum values for the items selected in the combobox

private:System :: Void fillcombo()

this function will the all items value in combo box from item table in database.

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)

set the minimum and maximum value for selected item in combobox

To add new items in item table of database

There are simply two textbox and three numeric dropdown that will take the the item detail and also insure whether the text filled in text box is not blank

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e)

this will simply add the item in database and immediately update the items in combobox as well.

private: System::Void logout\_Click(System::Object^ sender, System::EventArgs^ e)

The Change Password Button Coding Is given in the Function described below.