namespace ThiagoG\_Assign2

{

public partial class Form1 : Form

{

List<Grocery> GroceryList = new List<Grocery>();

List<string> HeaderList = new List<string>();

public Form1()

{

InitializeComponent();

}

// -------Loading Data into Windows Forms-------------------------

private void LoadGroceryDataButton\_Click(object sender, EventArgs e)

{

try {

using (StreamReader reader = new StreamReader("productlist.csv"))

{

GroceryList.Clear();

GroceryListBox.Items.Clear();

if (!reader.EndOfStream) {

string headerLine = reader.ReadLine();

string[] headerArr = headerLine.Split(',');

foreach (string header in headerArr) {

HeaderList.Add(header);

}

}

while(!reader.EndOfStream)

{

string row = reader.ReadLine();

string[] rowArr = row.Split(',');

string itemName = rowArr[0];

string itemId = rowArr[1];

double.TryParse(rowArr[2], out double unitPrice);

int.TryParse(rowArr[3], out int initQty);

int.TryParse(rowArr[4], out int qtySold);

int.TryParse(rowArr[5], out int qtyRestock);

int.TryParse(rowArr[6], out int qtyMinRestock);

Grocery grocItem = new Grocery(itemName, itemId, unitPrice, initQty,qtySold, qtyRestock, qtyMinRestock);

GroceryList.Add(grocItem);

}

LoadItemToListBox(GroceryList);

StatusLabel.Text = $"{GroceryList.Count} Grocery Items were added from input file";

}

} catch (Exception err) {

MessageBox.Show($"Error: {err}");

}

}

private void LoadItemToListBox(List<Grocery> groceryList)

{

string[] headerArr = { "Available Quantity", "Total Sales" };

foreach (string header in headerArr) {

HeaderList.Add(header);

}

GroceryListBox.Items.Add(

$"{HeaderList[0],-20}" +

$"{HeaderList[1],-25}" +

$"{HeaderList[2],-20}" +

$"{HeaderList[3],-20}" +

$"{HeaderList[4],-20}" +

$"{HeaderList[5],-20}" +

$"{HeaderList[6],-20}" +

$"{HeaderList[7],-20}" +

$"{HeaderList[8],-20}"

);

foreach (Grocery grocery in groceryList)

{

GroceryListBox.Items.Add(grocery);

}

}

private bool UpdateListBoxValidations(TextBox textBox, out int textBoxInt, out int grocId)

{

grocId = GroceryListBox.SelectedIndex;

if (grocId < 1)

{

MessageBox.Show("Please selected a grocery item to increment sold qty");

textBoxInt = -1;

return false;

}

else

{

if (!int.TryParse(textBox.Text,out textBoxInt))

{

MessageBox.Show($"{textBox.Text} is not a valid number");

return false;

}

else if (textBoxInt <= 0)

{

MessageBox.Show("Quantity cannot be equal or lesser than 0");

return false;

}

else if (textBox.Name == "QtySoldTextBox" && textBoxInt > GroceryList[grocId - 1].AvailableQty)

{

MessageBox.Show("Cannot Sell more than the available quantity");

return false;

}

else if (textBox.Name == "QtyRestockTextBox" && textBoxInt < GroceryList[grocId - 1].QtyMinForRestock)

{

MessageBox.Show("Restock Quantity Should be greater than the Qty Min for Restock");

return false;

}

else

{

return true;

}

}

}

private void UpdateSoldButton\_Click(object sender, EventArgs e)

{

if (UpdateListBoxValidations(QtySoldTextBox, out int qtySold, out int grocId)) {

GroceryList[grocId - 1].QtySold += qtySold;

GroceryListBox.Items[grocId] = GroceryList[grocId - 1];

StatusLabel.Text = $"Sold Qty has been increased for the item with item code {GroceryList[grocId - 1].ItemCode}";

QtySoldTextBox.Text = "";

}

}

private void UpdateRestockButton\_Click(object sender, EventArgs e)

{

if (UpdateListBoxValidations(QtyRestockTextBox, out int qtyRestock, out int grocId))

{

GroceryList[grocId - 1].QtyRestocked += qtyRestock;

GroceryListBox.Items[grocId] = GroceryList[grocId - 1];

StatusLabel.Text = $"Incremented Restocked Qty for item with item code {GroceryList[grocId - 1].ItemCode}";

QtyRestockTextBox.Text = "";

}

}

private void DeleteButton\_Click(object sender, EventArgs e)

{

int grocId = GroceryListBox.SelectedIndex;

if (grocId < 0)

{

MessageBox.Show("Please selecte an Item to be Deleted");

} else if (grocId == 0)

{

MessageBox.Show("Cannot Delete Header");

}

else

{

StatusLabel.Text = $"Deleted Item with item code {GroceryList[grocId - 1].ItemCode}";

GroceryList.RemoveAt(grocId-1);

GroceryListBox.Items.RemoveAt(grocId);

}

}

private void SortItemButton\_Click(object sender, EventArgs e)

{

List<Grocery> sortedGroceryList = new List<Grocery>();

sortedGroceryList = (from grocery in GroceryList

orderby grocery.TotalSales descending ,grocery.ItemName

select grocery).ToList();

GroceryListBox.Items.Clear();

LoadItemToListBox(sortedGroceryList);

StatusLabel.Text = "Sorting Items by Total Sale";

}

private void GenerateOutputFile(string fileName,List<Grocery> list)

{

{

using (StreamWriter sw = new StreamWriter($"{fileName}.csv"))

{

string header = $"{HeaderList[0]},{HeaderList[1]},{HeaderList[2]},{HeaderList[3]},{HeaderList[4]},{HeaderList[5]},{HeaderList[6]},{HeaderList[7]},{HeaderList[8]}";

sw.WriteLine(header);

foreach (Grocery grocery in list)

{

sw.WriteLine($"{grocery.ItemCode},{grocery.ItemName},{grocery.UnityPrice},{grocery.InitialQty},{grocery.QtySold},{grocery.QtyRestocked},{grocery.QtyMinForRestock},{grocery.AvailableQty},{grocery.TotalSales}");

}

StatusLabel.Text = $"{list.Count} were saved into {fileName} output file";

}

}

}

private void SaveGroceryButton\_Click(object sender, EventArgs e)

{

GenerateOutputFile("updatedgrocery",GroceryList);

}

private void SaveSalesRepButton\_Click(object sender, EventArgs e)

{

List<Grocery> salesList = new List<Grocery>();

salesList = (from grocery in GroceryList where grocery.QtySold>0 select grocery).ToList();

GenerateOutputFile("grocerysales", salesList);

}

private void SaveRestockButton\_Click(object sender, EventArgs e)

{

List<Grocery> restocklist = new List<Grocery>();

restocklist = (from grocery in GroceryList where grocery.AvailableQty < grocery.QtyMinForRestock select grocery).ToList();

GenerateOutputFile("groceryrestocks", restocklist);

}