-PROJECT REPORT-

CS 3101

Implementation of the Drink Dispenser System using C#

Members of The Group

L.Vipeshna	- S 14593
J.M.Divainson	- S 14539
Balakrishnan Thamayanthi	- S 14589
Piriyanka Periyathamby	- S 14569

Abstract

The project is to create an application that should provide service to the user, collect the user's orders and generate a bill for each order and made a payment.
The main objective for developing this project is to buy customers favorite drinks their own. This project provides some features to manage in every manner.

The main objective of the project on the Drink Dispenser system is to manage the details of drink items, customer orders, conform orders, and Payments.

The project is totally built at the customer end and thus anyone is guaranteed access.
The purpose of the project is to reduce the manual work, reduce workers expenses and managing the drink items, payment, customer order and confirm order.

Table of Content

1.Introduction

Drink Dispenser System is a simple desktop application developed using C# Language. The Project is based on the concept of generating the receipt of the total bill of a customer. Talking about the system, the user just has to select among the drinks items, enter the quantity and click on the order button to view the total price. The user can view the total receipt of their items which displays the number of their drinks items with the total amount

The design of the project is very simple so that the user won't find any difficulties while working on it. This project does not use any external file as a database/to store records. Drink Dispenser System helps the management to order drinks items and generating bills.

To run this project you must have installed Visual Studio IDE on your PC. Drink Dispenser System in C# is free to download with source code.

3.Requirements and specifications

3.1. Functional Requirements

These are the following functional requirements of the Drink Dispenser System and invoicing in C#

- **Display the Menu**: This will show you what drinks are available with their rates.
- **Select Drink item/s**: Drink Items are selected feel free to order.
- **Your Order**: This will show you which drink you are ordering, the number of drinks you order, and the total cash amount.
- **Order**: provide total bill amount and the message to the customer to tell insert the cash amount.
- **Confirm order** : Here the bill is generated and the balance is displayed if you enter the cash payment amount and confirm it.
- **Reset** : to set again a system.

3.2. Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Processor: Intel Pentium 4 or Higher
RAM: 64 MB or Higher
Hard Disk: 130 MB

3.3. Software Requirements

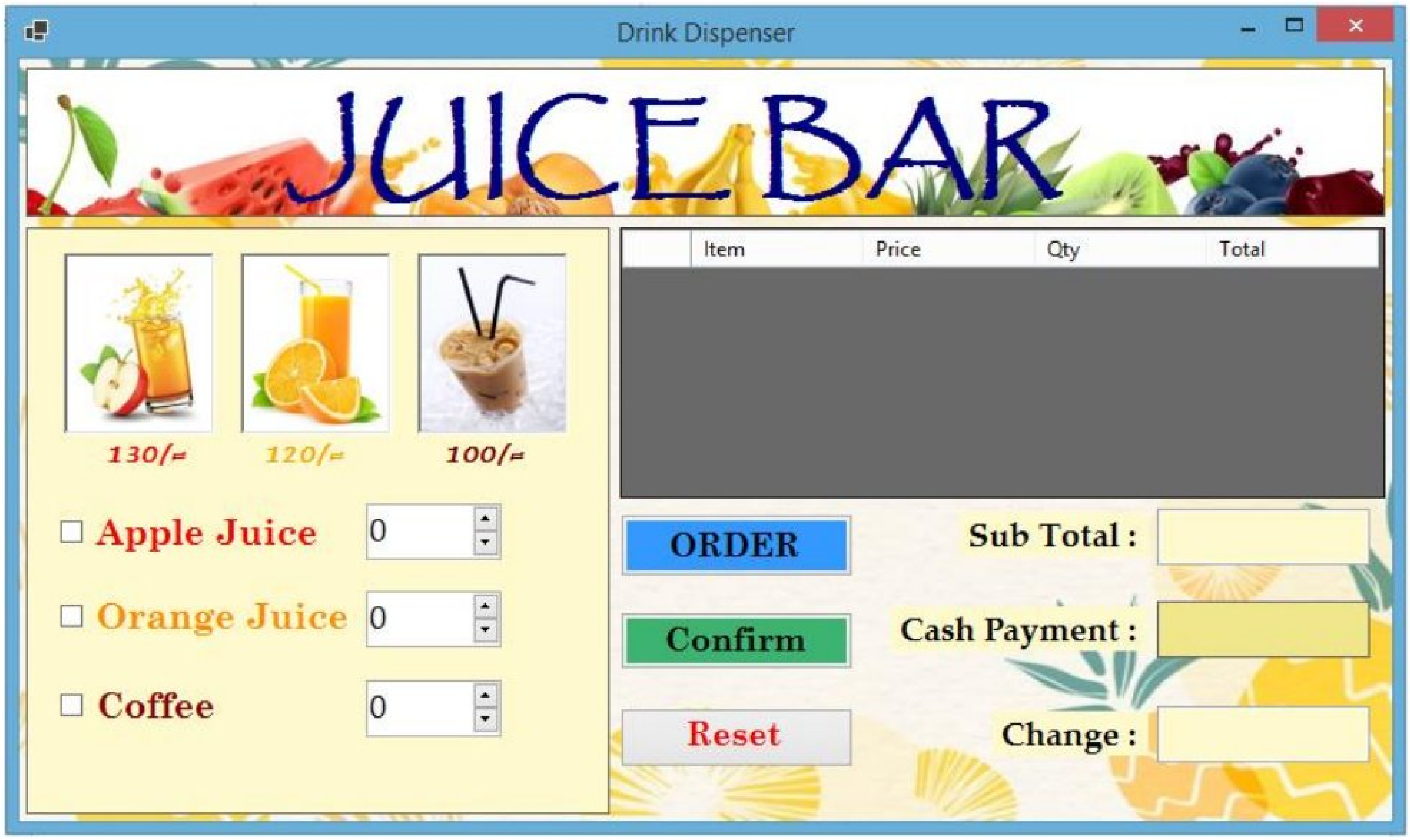
Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

To edit this project you must have installed Visual Studio IDE on your PC. Drink Dispenser System in C# is free to download with source code.

Operating System: Windows XP or Higher

4. Design of Solution

4.1.User Interface

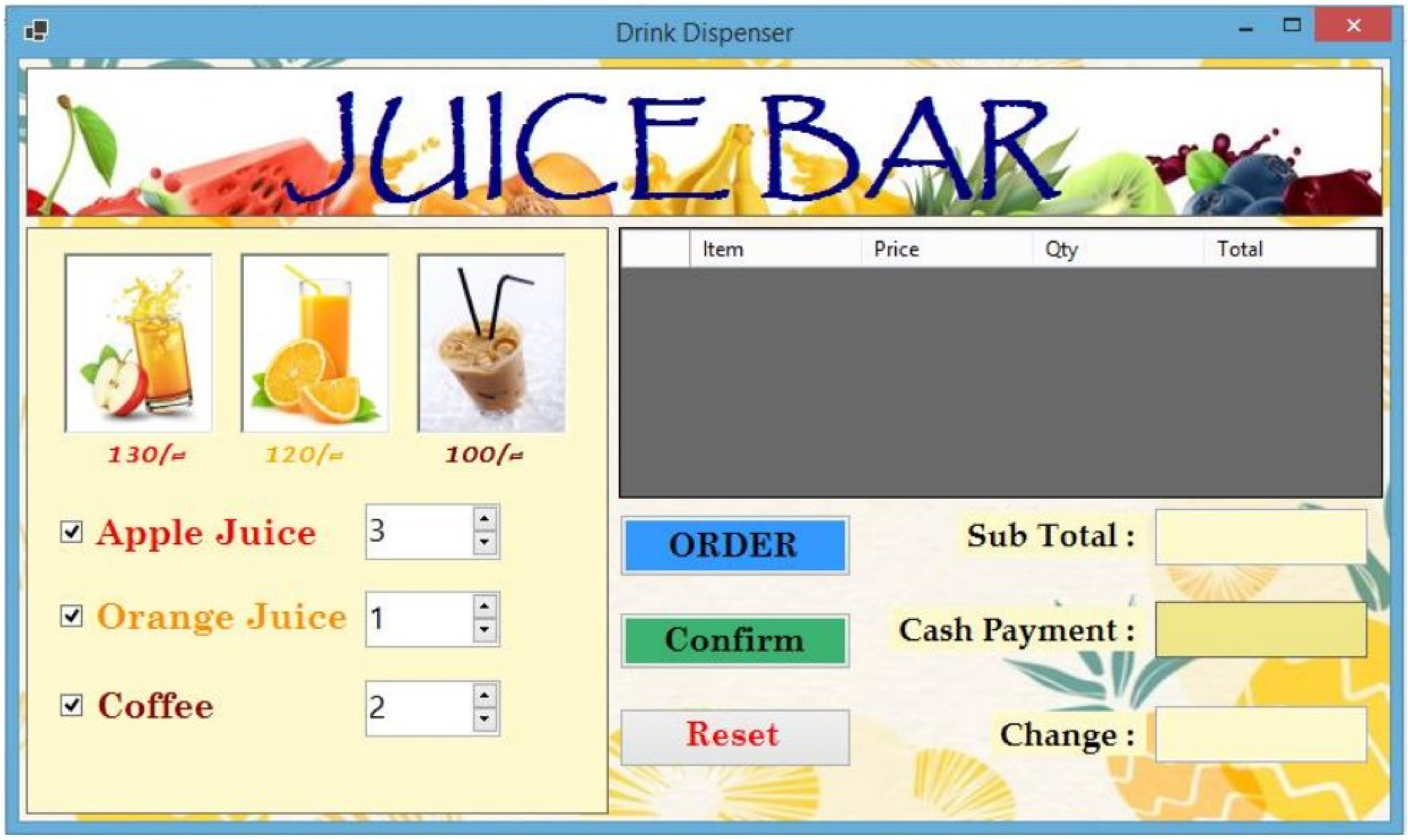


□ Apple Juice 0

□ Orange Juice 0

□ Coffee 0

1. Order your favorite drink and click on the check box to set the number of drinks you need. Then click the "ORDER" button.

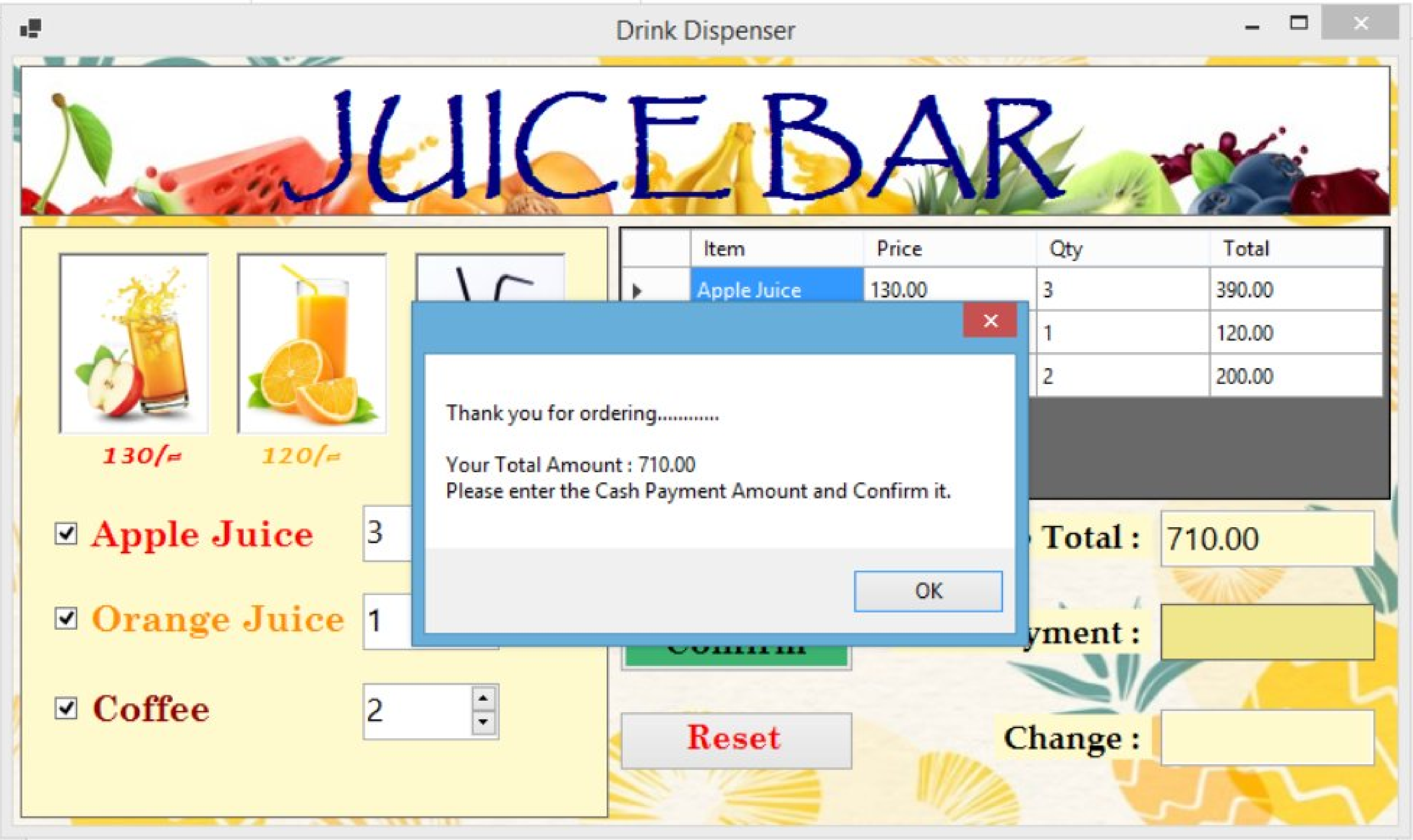


☑ Apple Juice 3

☑ Orange Juice 1

☑ Coffee 2

2. A message will appear to the customer showing the total amount of the bill and asking them to enter the cash payment amount.



☑ Apple Juice 3

☑ Orange Juice 1

☑ Coffee 2

Item	Price	Qty	Total
Apple Juice	130.00	3	390.00
		1	120.00
		2	200.00

Thank you for ordering...........

Your Total Amount : 710.00
Please enter the Cash Payment Amount and Confirm it.

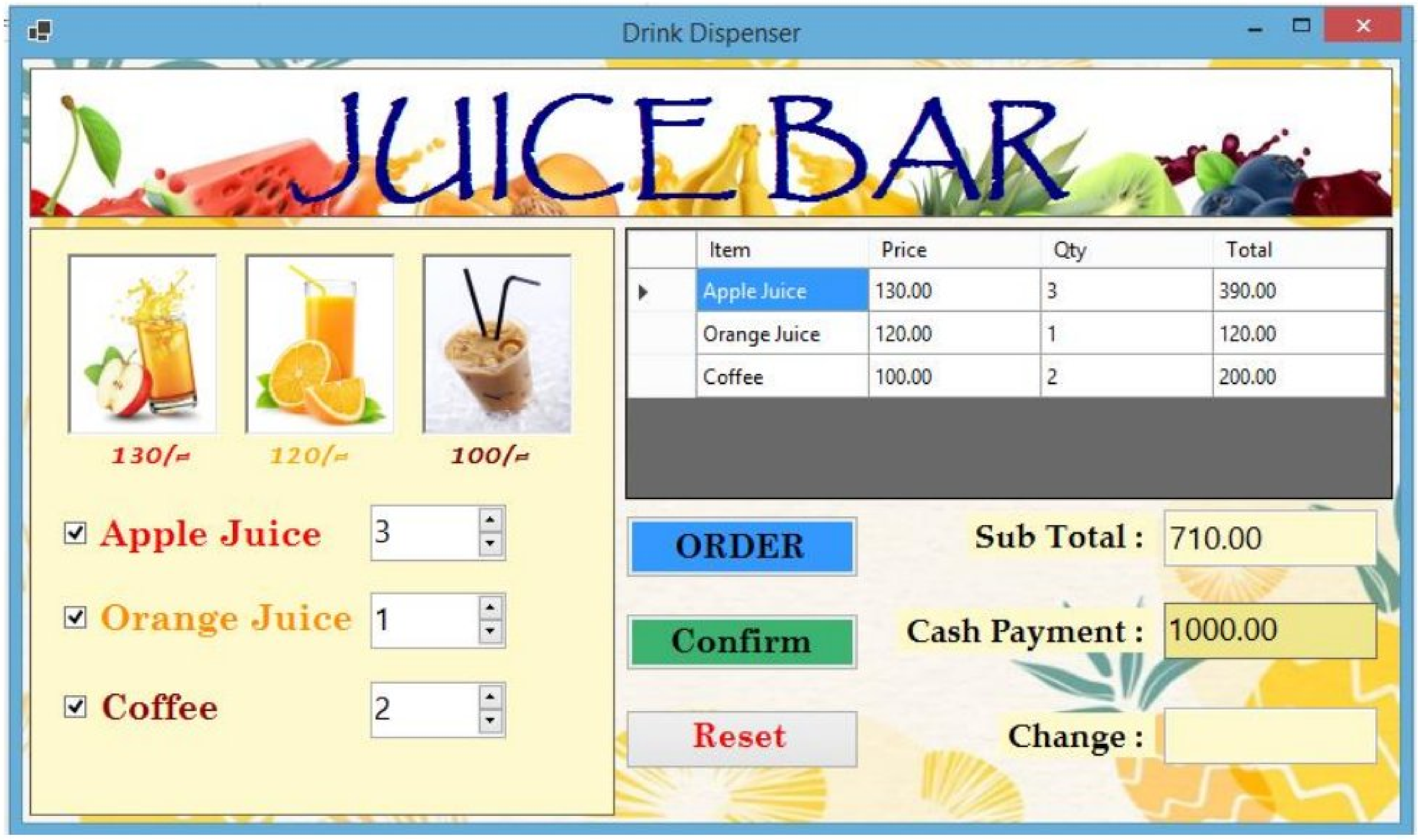
OK

Total : 710.00

yment :

Change :

3. Enter the cash payment amount then click the “Confirm” button.



☑ Apple Juice 3

☑ Orange Juice 1

☑ Coffee 2

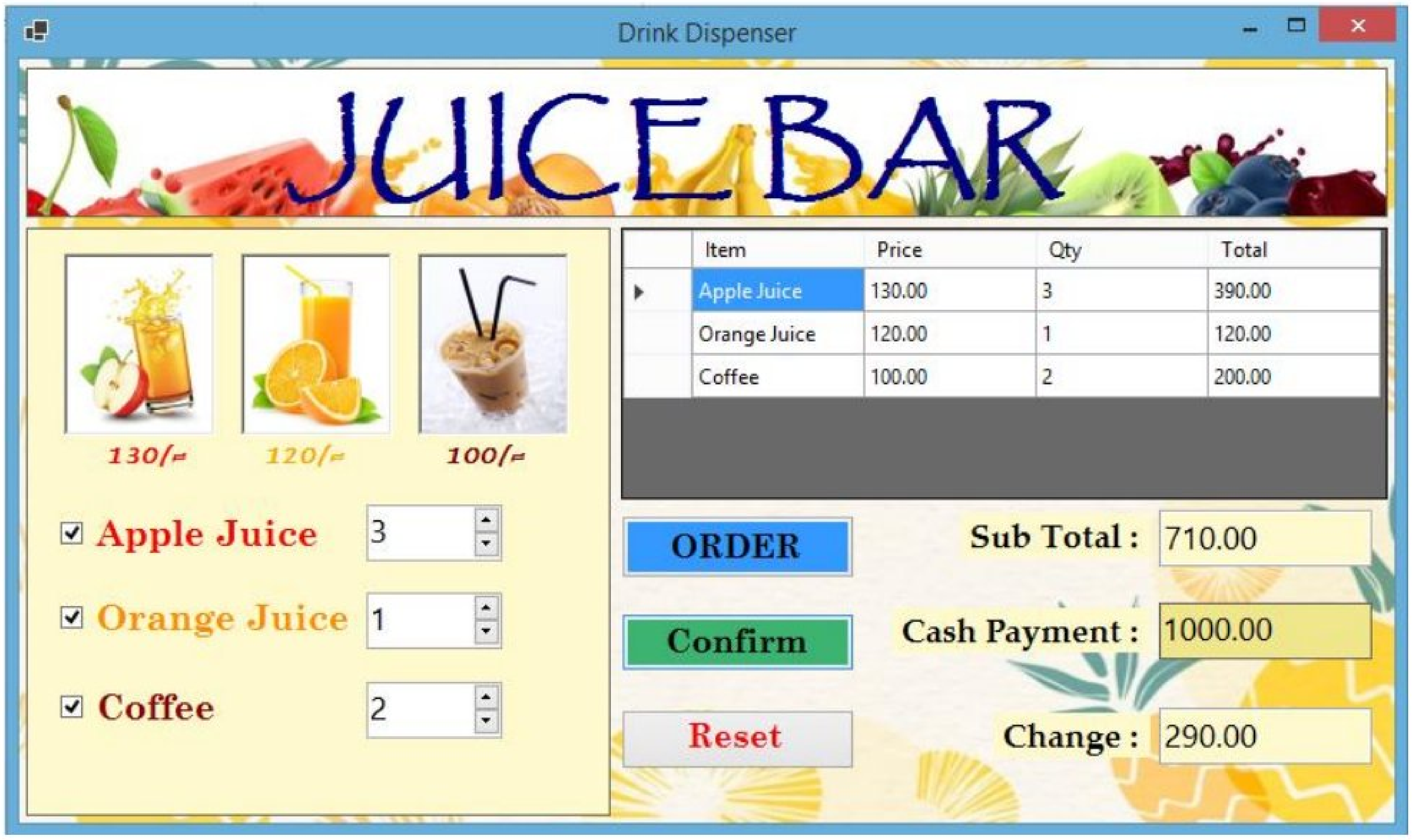
Item	Price	Qty	Total
Apple Juice	130.00	3	390.00
Orange Juice	120.00	1	120.00
Coffee	100.00	2	200.00

Sub Total : 710.00

Cash Payment : 1000.00

Change :

4. Then finally your order bill Total amount, Cash Payment amount and balance (change) amount will be displayed.



☑ Apple Juice 3

☑ Orange Juice 1

☑ Coffee 2

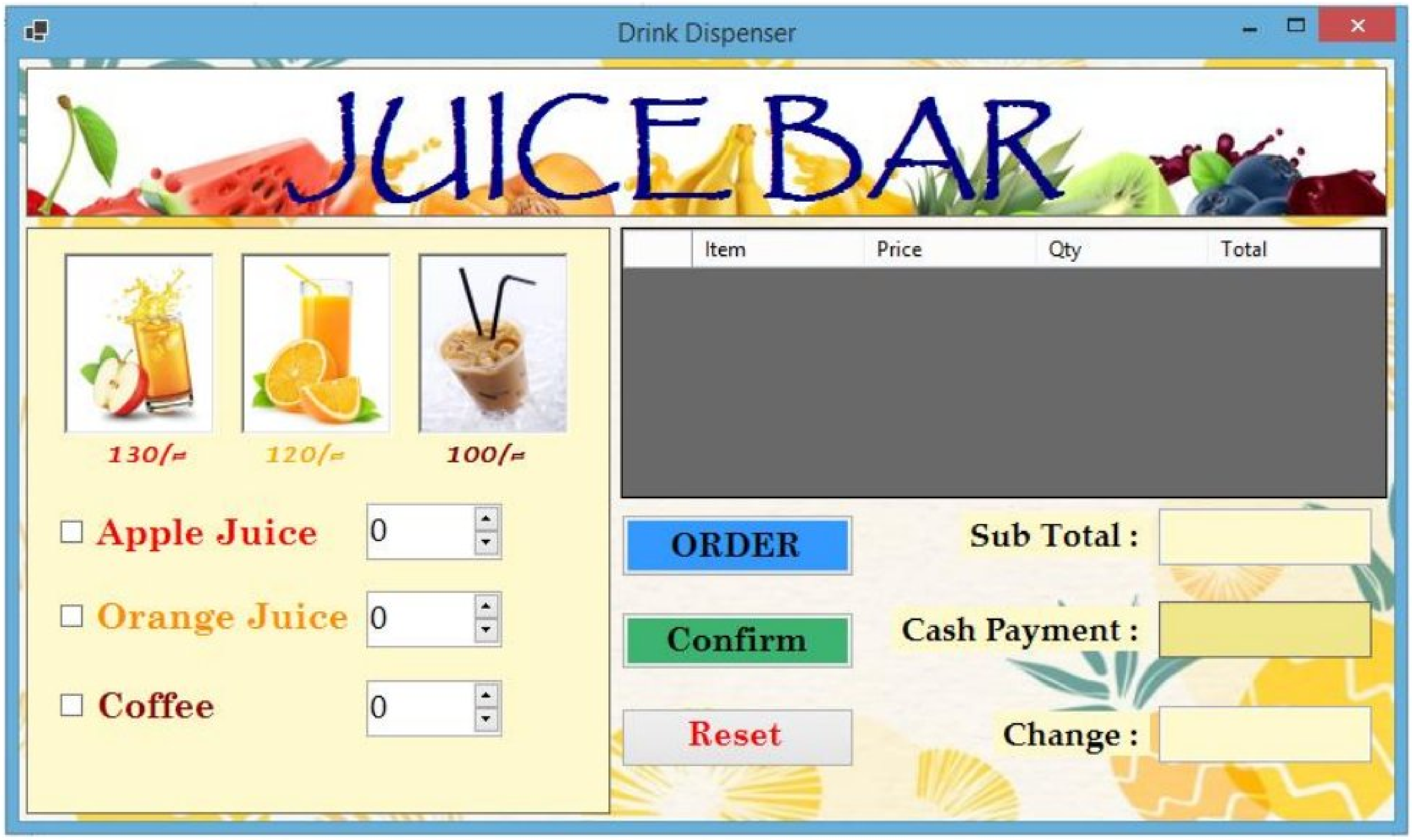
Item	Price	Qty	Total
Apple Juice	130.00	3	390.00
Orange Juice	120.00	1	120.00
Coffee	100.00	2	200.00

Sub Total : 710.00

Cash Payment : 1000.00

Change : 290.00

5. If you click on "Reset", the orders and all the amounts will be reset



□ Apple Juice 0

□ Orange Juice 0

□ Coffee 0

Item	Price	Qty	Total

Sub Total :

Cash Payment :

Change :

4.2. C# Code

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Drink_Dispenser
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        //######################## ORDER ############################
        string name;
        double price;
        double total;
        private void btnOrder_Click(object sender, EventArgs e)
        {
            if (checkBoxAp.Checked)
            {
                name = "Apple Juice";
                int qty = int.Parse(numericUpDown1.Value.ToString());
                price = 130.00;
                total = qty * price;
                this.dataGridView1.Rows.Add(name, price.ToString("#.00"),
qty, total.ToString("#.00"));
            }

            if (checkBoxOr.Checked)
            {
                name = "Orange Juice";
                int qty = int.Parse(numericUpDown2.Value.ToString());
                price = 120.00;
                total = qty * price;
                this.dataGridView1.Rows.Add(name, price.ToString("#.00"),
qty, total.ToString("#.00"));
            }

            if (checkBoxCo.Checked)
            {
                name = "Coffee";
                int qty = int.Parse(numericUpDown3.Value.ToString());
                price = 100.00;
                total = qty * price;
                this.dataGridView1.Rows.Add(name, price.ToString("#.00"),
qty, total.ToString("#.00"));
            }

            double sum = 0;
            for (int row = 0; row < dataGridView1.Rows.Count; row++)
            {
                sum = sum +
Convert.ToDouble(dataGridView1.Rows[row].Cells[3].Value);
            }

            textBoxTotal.Text = sum.ToString("#.00");
            MessageBox.Show("Thank you for ordering............ \n\n"
+"Your Total Amount : "+sum.ToString("#.00")
                + "\nPlease enter the Cash Payment Amount and Confirm
it.");
        }

        //######################### CONFIRM ORDER ###################
        private void btnConfirm_Click(object sender, EventArgs e)
        {
            double pym_amount = double.Parse(textBoxCash.Text);
            double total = double.Parse(textBoxTotal.Text);
            double change = pym_amount - total;
            textBoxCash.Text = pym_amount.ToString("#.00");
            textBoxChange.Text = change.ToString("#.00");
        }

        //######################### RESET ############################
        private void Resetcheckbox()
        {
            Action<Control.ControlCollection> func = null;

            func = (controls) =>
            {
                foreach (Control control in controls)
                    if (control is CheckBox)
                        (control as CheckBox).Checked = false;
                    else
                        func(control.Controls);
            };
            func(Controls);
        }

        private void Resetnumericupdown()
        {
            Action<Control.ControlCollection> func = null;

            func = (controls) =>
            {
                foreach (Control control in controls)
                    if (control is NumericUpDown)
                        (control as NumericUpDown).Value = 0;
                    else
                        func(control.Controls);
            };
            func(Controls);
        }
        private void btnReset_Click(object sender, EventArgs e)
        {
            Resetcheckbox();
            Resetnumericupdown();
            dataGridView1.DataSource = null;
            dataGridView1.Rows.Clear();
            textBoxTotal.Text = String.Empty;
            textBoxCash.Text = String.Empty;
            textBoxChange.Text = String.Empty;

        }
    }
}
```

5.References

- https://www.w3schools.com/cs/index.php
- https://code.visualstudio.com/docs/languages/csharp
- http://csharp.net-informations.com/
- http://csharp.net-informations.com/datagridview/csharp-datagridview-checkbox.htm