```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Media;

namespace WpfApp2
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        Dictionary<string, int> drinks = new Dictionary<string, int>();
        Dictionary<string, int> orders = new Dictionary<string, int>();
        string takeout = "";
        public MainWindow()
        {
            InitializeComponent();

            //新增所有飲料品項
            AddNewDrink(drinks);

            //顯示飲料品項菜單
            DisplayDrinkMenu(drinks);
        }

        private void DisplayDrinkMenu(Dictionary<string, int> myDrinks)
        {
            foreach (var drink in myDrinks)
            {
                var sp = new StackPanel
                {
                    Orientation = Orientation.Horizontal
                };

                var cb = new CheckBox
                {
                    Content = $"{drink.Key} : {drink.Value}元",
                    Width = 200,
                    FontFamily = new FontFamily("Consolas"),
                    FontSize = 18,
                    Foreground = Brushes.Blue,
                    Margin = new Thickness(5)
                };

                var sl = new Slider
                {
                    Width = 100,
                    Value = 0,
                    Minimum = 0,
                    Maximum = 10,
                    VerticalAlignment = VerticalAlignment.Center,
                    IsSnapToTickEnabled = true
                };

                var lb = new Label
                {
                    Width = 50,
                    Content = "0",
                    FontFamily = new FontFamily("Consolas"),
                    FontSize = 18,
                    Foreground = Brushes.Red
                };

                sp.Children.Add(cb);
                sp.Children.Add(sl);
                sp.Children.Add(lb);

                //資料繫結
                Binding myBinding = new Binding("Value");
                myBinding.Source = sl;
                lb.SetBinding(ContentProperty, myBinding);

                stackpanel_DrinkMenu.Children.Add(sp);
            }
        }

        private void AddNewDrink(Dictionary<string, int> myDrinks)
        {
            //myDrinks.Add("紅茶大杯", 60);
            //myDrinks.Add("紅茶小杯", 40);
            //myDrinks.Add("綠茶大杯", 60);
            //myDrinks.Add("綠茶小杯", 40);
            //myDrinks.Add("咖啡大杯", 80);
            //myDrinks.Add("咖啡小杯", 50);
            //myDrinks.Add("可樂大杯", 40);
            //myDrinks.Add("可樂小杯", 20);

            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "CSV檔案|*.csv|文字檔案|*.txt|全部檔案|*.*";
            if (openFileDialog.ShowDialog() == true)
            {
                string filename = openFileDialog.FileName;
                //string alltext = File.ReadAllText(filename);
                string[] lines = File.ReadAllLines(filename);
                foreach (var line in lines)
                {
                    string[] tokens = line.Split(',');
                    string drinkName = tokens[0];
                    int price = Convert.ToInt32(tokens[1]);
                    myDrinks.Add(drinkName, price);
                }
            }
        }

        private void OrderButton_Click(object sender, RoutedEventArgs e)
        {
            // 將訂購的飲料加入訂單
            PlaceOrder(orders);

            // 顯示訂單內容
            DisplayOrder(orders);

            // 生成唯一的檔案名稱，例如使用日期和時間
            string timestamp = DateTime.Now.ToString("yyyyMMddHHmm");
            string folderPath = "D:\\Study_Kai\\五專三上\\zhuang-c-sharp\\WpfApp2"; // 請將 "YourFolderPath" 替換為目標資料夾的路徑
            string orderFileName = Path.Combine(folderPath, "Order_" + timestamp + ".txt");

            // 儲存訂單到新的 txt 檔案
            SaveOrderToFile(orders, orderFileName);
        }

        private void SaveOrderToFile(Dictionary<string, int> myOrders, string fileName)
        {
            try
            {
                using (StreamWriter writer = new StreamWriter(fileName))
                {
                    writer.WriteLine("訂單內容：");
                    foreach (var item in myOrders)
                    {
                        string drinkName = item.Key;
                        int quantity = myOrders[drinkName];
                        writer.WriteLine($"{drinkName} X {quantity}杯");
                    }
                    writer.Close();
                }
                MessageBox.Show("訂單已成功儲存到 " + fileName, "儲存成功");
            }
            catch (Exception ex)
            {
                MessageBox.Show("儲存訂單時發生錯誤：" + ex.Message, "錯誤");
            }
        }



        private void DisplayOrder(Dictionary<string, int> myOrders)
        {
            displayTextBlock.Inlines.Clear();
            Run titleString = new Run
            {
                Text = "您所訂購的飲品為 ",
                FontSize = 16,
                Foreground = Brushes.Blue
            };

            Run takeoutString = new Run
            {
                Text = $"{takeout}",
                FontSize = 16,
                FontWeight = FontWeights.Bold
            };

            displayTextBlock.Inlines.Add(titleString);
            displayTextBlock.Inlines.Add(takeoutString);
            displayTextBlock.Inlines.Add(new Run() { Text = " ，訂購明細如下: \n", FontSize = 16 });

            double total = 0.0;
            double sellPrice = 0.0;
            //string displayString = $"本次訂購為{takeout}，清單如下：\n";
            string discountString = "";

            int i = 1;
            foreach (var item in myOrders)
            {
                string drinkName = item.Key;
                int quantity = myOrders[drinkName];
                int price = drinks[drinkName];
                total += price * quantity;
                displayTextBlock.Inlines.Add(new Run() { Text = $"飲料品項{i}： {drinkName} X {quantity}杯，每杯{price}元，總共{price * quantity}元\n" });
                i++;
            }

            if (total >= 500)
            {
                discountString = "訂購滿500元以上者打8折";
                sellPrice = total * 0.8;
            }
            else if (total >= 300)
            {
                discountString = "訂購滿300元以上者打85折";
                sellPrice = total * 0.85;
            }
            else if (total >= 200)
            {
                discountString = "訂購滿200元以上者打9折";
                sellPrice = total * 0.9;
            }
            else
            {
                discountString = "訂購未滿200元以上者不打折";
                sellPrice = total;
            }

            Italic summaryString = new Italic(new Run
            {
                Text = $"本次訂購總共{myOrders.Count}項，{discountString}，售價{sellPrice}元",
                FontSize = 16,
                FontWeight = FontWeights.Bold,
                Foreground = Brushes.Red
            });
            displayTextBlock.Inlines.Add(summaryString);
        }

        private void PlaceOrder(Dictionary<string, int> myOrders)
        {
            myOrders.Clear();
            for (int i = 0; i < stackpanel_DrinkMenu.Children.Count; i++)
            {
                var sp = stackpanel_DrinkMenu.Children[i] as StackPanel;
                var cb = sp.Children[0] as CheckBox;
                var sl = sp.Children[1] as Slider;
                string drinkName = cb.Content.ToString().Substring(0, 4);
                int quantity = Convert.ToInt32(sl.Value);

                if (cb.IsChecked == true && quantity != 0)
                {
                    myOrders.Add(drinkName, quantity);
                }
            }
        }

        private void RadioButton_Checked(object sender, RoutedEventArgs e)
        {
            var rb = sender as RadioButton;
            if (rb.IsChecked == true) takeout = rb.Content.ToString();
            //MessageBox.Show(takeout);
        }
    }
}
```