Program Cover Sheet

|  |
| --- |
| Name: Austin Rippee |
| Assignment: Assignment 7 |
| List any parts of the assignment that do not work/were not completed:  On a click of a selected prepped item or selected raw ingredient, and based on my MouseDown event, it goes through the process of doing a dragdrop event instead of just simply clicking the item so the message “Item has already been added” continues to pop up. With removing this MouseDown feature, the program works well with simply clicking a prepped item and displaying the raw ingredients. Just when I have the PreppedSelected MouseDown event or the RawItem MouseDown event, that is when I have the issue of where the program believes it is doing a drag-drop and I couldn’t find a way to decipher between a single click or a drag. This probably would have worked if I did some error checking or event checking but also couldn’t think of any proper way to implement this.  The other thing was I couldn’t implement multi select properly. I was able to select multiple items from the prepped items and the raw ingredients and able to drag both but couldn’t copy both into the listbox target, only the second item that was selected. |

|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.

'------------------------------------------------------------

'- File Name : frmChezSouSad.vb -

'- Part of Project: Main -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- File Purpose: -

'- The user will be prompted with a selection of a dish, then

'- based on that, they will be prompted with prepped items, then

'- based on that, prompted with raw ingredients made. This will

'- ultimately lead to creating their own dish.

'------------------------------------------------------------

'- Program Purpose: -

'- -

'- This program allows for the user to create their own dish

'- filled with prepped items that are made from raw materials

'- that are listed within.

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'- gdicDishes - Creates dictionary for all dishes made from the prepped items

'- gdicPreppedItems - Creates dictionary for all prepped items made from from the raw items

'- gdicRawItems - Creates dictionary for all raw items

'- isDoDragDropActive - Unused boolean to check if the drag drop feature is enabled

'------------------------------------------------------------

Public Class frmChezSouSad

'Creates dictionary for all raw items

Public gdicRawItems As New Dictionary(Of String, String)

'Creates dictionary for all prepped items made from from the raw items

Public gdicPreppedItems As New Dictionary(Of String, Dictionary(Of String, String))

'Creates dictionary for all dishes made from the prepped items

Public gdicDishes As New Dictionary(Of String, Dictionary(Of String, String))

'Creates temporary boolean to check if a dragdrop event is enabled (not being used, just had it for an idea)

Public blnIsDoDragDropActive As Boolean = False

'------------------------------------------------------------

'- Subprogram Name: frm\_Load -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user loads the

'– form. Items will be automatically added to the various

'- list boxes of dishes, prepped items, and raw ingredients.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub frm\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

frmSplash.ShowDialog()

'Load all raw items into the dictionary

AddRawIngredient("Basket")

AddRawIngredient("Beef Patty")

AddRawIngredient("Bun")

AddRawIngredient("Butter")

AddRawIngredient("Chicken")

AddRawIngredient("Coffee Beans")

AddRawIngredient("Creamer")

AddRawIngredient("Flour")

AddRawIngredient("Glass")

AddRawIngredient("Grapes")

AddRawIngredient("Ketchup")

AddRawIngredient("Mayonaise")

AddRawIngredient("Milk")

AddRawIngredient("Mustard")

AddRawIngredient("Oil")

AddRawIngredient("Onions")

AddRawIngredient("Pickles")

AddRawIngredient("Plate")

AddRawIngredient("Sugar")

AddRawIngredient("Syrup")

AddRawIngredient("Water")

'Load all prepped items

AddPreppedItem("Cereal Bowl", "Sugar")

AddPreppedItem("Cereal Bowl", "Milk")

AddPreppedItem("Coffee", "Creamer")

AddPreppedItem("Coffee", "Coffee Beans")

AddPreppedItem("Chicken salad", "")

AddPreppedItem("Cookie", "")

AddPreppedItem("Fries", "")

AddPreppedItem("Hamburger", "")

AddPreppedItem("Soft drink", "")

AddPreppedItem("Cookie", "Sugar")

AddPreppedItem("Cookie", "Flour")

AddPreppedItem("Cookie", "Butter")

AddPreppedItem("Soft drink", "Sugar")

AddPreppedItem("Soft drink", "Water")

AddPreppedItem("Soft drink", "Glass")

AddPreppedItem("Soft drink", "Syrup")

AddPreppedItem("Hamburger", "Beef patty")

AddPreppedItem("Hamburger", "Bun")

AddPreppedItem("Hamburger", "Ketchup")

AddPreppedItem("Hamburger", "Mayonaise")

AddPreppedItem("Hamburger", "Mustard")

AddPreppedItem("Hamburger", "Onions")

AddPreppedItem("Hamburger", "Pickles")

AddPreppedItem("Fries", "Oil")

AddPreppedItem("Fries", "Ketchup")

AddPreppedItem("Fries", "Basket")

AddPreppedItem("Chicken salad", "Plate")

AddPreppedItem("Chicken salad", "Chicken")

'Load all prepped items into each dish

AddDish("Hamburger platter", "Hamburger")

AddDish("Chicken salad platter", "")

AddDish("Breakfast", "Cereal Bowl")

AddDish("Breakfast", "Coffee")

'Add all raw items from the dictionary to the list box

For Each rawKey In gdicRawItems.Keys

lstRaw.Items.Add(gdicRawItems.Item(rawKey))

Next

'Add all prepped items from the dictionary to the list box

For Each rawingredient In gdicPreppedItems.Keys

lstPrep.Items.Add(rawingredient)

Next

'Add all dishes from the dictionary to the list box

DisplayDishes()

'enable listbox drag/drop

lstPrep.AllowDrop = True

lstPreppedSelected.AllowDrop = True

lstRaw.AllowDrop = True

lstRawSelected.AllowDrop = True

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnAddRaw\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- AddRaw button. This sub adds a new raw ingredient to the

'– list but will not be added twice and won't be added if

'- nothing has been added to the text box.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub btnAddRaw\_Click(sender As Object, e As EventArgs) Handles btnAddRaw.Click

'Checks if nothing has been entered

If txtRaw.Text = "" Then

MsgBox("No value entered, try again")

Else

'Checks if the key is already entered

If gdicRawItems.ContainsKey(txtRaw.Text) Then

MsgBox("Item already added, try something else")

Else

Try

'adds to dictionary

AddRawIngredient(txtRaw.Text)

Catch

MsgBox(Err.ToString)

End Try

'Clears the list box

lstRaw.Items.Clear()

'adds to the listbox

For Each rawKey In gdicRawItems.Keys

lstRaw.Items.Add(gdicRawItems.Item(rawKey))

Next

'sorts the listbox

lstRaw.Sorted = True

'sets the textbox to contain nothing

txtRaw.Text = ""

End If

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnAddPrep\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- AddPrep button. This sub adds a new prepped item to the

'– list but will not be added twice and won't be added if

'- nothing has been added to the text box. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub btnAddPrep\_Click(sender As Object, e As EventArgs) Handles btnAddPrep.Click

'Checks if nothing has been entered

If txtPrep.Text = "" Then

MsgBox("No value entered, try again")

Else

'Checks if the key is already entered

If gdicPreppedItems.ContainsKey(txtPrep.Text) Then

MsgBox("Item already added, try something else")

Else

Try

'adds to dictionary

AddPreppedItem(txtPrep.Text, "")

Catch

MsgBox(Err.ToString)

End Try

'Clears the list box

lstPrep.Items.Clear()

'adds to list box

For Each rawingredient In gdicPreppedItems.Keys

lstPrep.Items.Add(rawingredient)

Next

'sorts list box

lstPrep.Sorted = True

'sets text to nothing in textbox

txtPrep.Text = ""

End If

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnAddDish\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- AddDish button. This sub adds a new dish to the list but

'- will not be added twice and won't be added if nothing has

'- been added to the text box. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub btnAddDish\_Click(sender As Object, e As EventArgs) Handles btnAddDish.Click

'Checks if nothing has been entered

If txtDish.Text = "" Then

MsgBox("No value entered, try again")

Else

'Checks if item is already entered

If gdicDishes.ContainsKey(txtDish.Text) Then

MsgBox("Item already added, try something else")

Else

Try

'adds to dictionary

AddDish(txtDish.Text, "")

Catch

MsgBox(Err.ToString)

End Try

'Clears the list box

lstDish.Items.Clear()

'adds to listbox

DisplayDishes()

'sorts list box

lstDish.Sorted = True

'sets text to nothing in textbox

txtDish.Text = ""

End If

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnLeftRaw\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- LeftRaw button. This adds the selected item from the

'- raw ingrediants list to the selected raw ingredients.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- selectedItems - Keeps track of the selected raw items -

'------------------------------------------------------------

Private Sub btnLeftRaw\_Click(sender As Object, e As EventArgs) Handles btnLeftRaw.Click

If lstRaw.SelectedIndex = -1 Then

MsgBox("Nothing Selected, try again")

Else

Dim selectedItems = lstRaw.SelectedItems.Cast(Of String)

For Each item In selectedItems

'Adds the newly selected prepped item to the dish

AddPreppedItem(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem), item)

Next

'Clears list boxes

lstRawSelected.Items.Clear()

'Displays the raw items in the list box

DisplayRawItems(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem))

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnRightRaw\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- RightRaw button. This removes the raw ingredient from the

'- selected raw ingrediants. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub btnRightRaw\_Click(sender As Object, e As EventArgs) Handles btnRightRaw.Click

'Checks if nothing is selected

If lstRawSelected.SelectedIndex = -1 Then

MsgBox("Nothing Selected, try again")

Else

'removes from dictionary

gdicPreppedItems.Remove(lstRawSelected.GetItemText(lstRawSelected.SelectedItem))

'clears listbox

lstRawSelected.Items.Clear()

'displays the raw ingredients

DisplayRawItems(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem))

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnLeftPrep\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- LeftPrep button. This adds the selected item from the

'- prepped items list to the selected prepped items. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- selectedItems - Keeps track of the selected raw items -

'------------------------------------------------------------

Private Sub btnLeftPrep\_Click(sender As Object, e As EventArgs) Handles btnLeftPrep.Click

If lstPrep.SelectedIndex = -1 Then

MsgBox("Nothing Selected, try again")

Else

Dim selectedItems = lstPrep.SelectedItems.Cast(Of String)

For Each item In selectedItems

'Adds the newly selected prepped item to the dish

AddDish(lstDish.GetItemText(lstDish.SelectedItem), item)

Next

'Clears list boxes

lstPreppedSelected.Items.Clear()

lstRawSelected.Items.Clear()

'Displays the prepped items in the list box

DisplayPreppedItems(lstDish.GetItemText(lstDish.SelectedItem))

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: btnRightPrep\_Click -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- RightPrep button. This removes the prepped item from the

'- selected prepped items. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strMyDish - Loads dish value as string

'- strMyPrep - Loads prepped item value as string

'------------------------------------------------------------

Private Sub btnRightPrep\_Click(sender As Object, e As EventArgs) Handles btnRightPrep.Click

'Checks if nothing is selected

If lstPreppedSelected.SelectedIndex = -1 Then

MsgBox("Nothing Selected, try again")

Else

Dim strMyDish As String = lstDish.GetItemText(lstDish.SelectedItem)

Dim strMyPrep As String = lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem)

'removes from dictionary and listbox

gdicDishes.Item(strMyDish).Remove(strMyPrep)

'Clears the listbox

lstPreppedSelected.Items.Clear()

'display items

DisplayPreppedItems(lstDish.GetItemText(lstDish.SelectedItem))

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstDish\_SelectedIndexChanged -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks upon a

'- different index of the lstDish. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub lstDish\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles lstDish.SelectedIndexChanged

If lstDish.SelectedIndex <> -1 Then

'Displays the prepped items

DisplayPreppedItems(lstDish.GetItemText(lstDish.SelectedItem))

Else

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPreppedSelected\_SelectedIndexChanged -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks upon a

'- different index of the lstPreppedSelected. –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub lstPreppedSelected\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles lstPreppedSelected.SelectedIndexChanged

If lstPreppedSelected.SelectedIndex <> -1 Then

'Displays the raw ingredients

DisplayRawItems(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem))

Else

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: AddRawIngredient -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user adds a raw ingredient

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- item - value as string

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Public Sub AddRawIngredient(ByVal item As String)

'Adds the raw ingredient to the dictionary

gdicRawItems.Add(item, item)

End Sub

'------------------------------------------------------------

'- Subprogram Name: AddPreppedItem -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user wantes to add a prepped item

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- preppeditem - String as name

'- rawingredient - ingredient associated with prepped item

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- Found - keeps track of whats been added -

'------------------------------------------------------------

Public Sub AddPreppedItem(ByVal preppedItem As String, ByVal rawingredient As String)

Dim Found As Integer = 0

'loop through prepped items

For Each rawKey In gdicPreppedItems.Keys

If rawKey = preppedItem Then

Found = 1

Exit For

End If

Next

'checks whether an item has been found

If Found = 1 Then

Try

'add raw ingredient to prepped item

gdicPreppedItems.Item(preppedItem).Add(rawingredient, rawingredient)

Catch

MsgBox("Item has already been added")

End Try

Else

'add new item

gdicPreppedItems.Add(preppedItem, New Dictionary(Of String, String))

If rawingredient <> "" Then

'add prepped item to dish

gdicPreppedItems.Item(preppedItem).Add(rawingredient, rawingredient)

End If

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: AddDish -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user wants to add a new dish

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- dish - value name as string

'- preppedItem - prepped item value to add

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- Found - keeps track of what's been added -

'------------------------------------------------------------

Public Sub AddDish(ByVal dish As String, ByVal preppedItem As String)

Dim Found As Integer = 0

'loop through dishes

For Each dishKey In gdicDishes.Keys

If dishKey = dish Then

Found = 1

Exit For

End If

Next

'checks whether an item has been found

If Found = 1 Then

'add prepped item to dish

Try

gdicDishes.Item(dish).Add(preppedItem, preppedItem)

Catch

MsgBox("Item has already been added")

End Try

Else

'add new dish

gdicDishes.Add(dish, New Dictionary(Of String, String))

If preppedItem <> "" Then

'add prepped item to dish

gdicDishes.Item(dish).Add(preppedItem, preppedItem)

End If

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: DisplayPreppedItems -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user wants to display

'- the prepped items based on selected dish

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- dish - string as value name

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub DisplayPreppedItems(ByVal dish As String)

'checks if an item is selected

If lstDish.SelectedIndex <> -1 Then

'clears list boxes

lstPreppedSelected.Items.Clear()

lstRawSelected.Items.Clear()

'populates list box based on selected dish

For Each item In gdicDishes(lstDish.SelectedItem.ToString())

lstPreppedSelected.Items.Add(item.Key)

Next

Else

MsgBox("You have not selected a dish")

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: DisplayRawItems -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user wants to

'- display raw items based on selected prepped item

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- preppedItem - value name as String

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub DisplayRawItems(ByVal preppedItem As String)

'clears list box

lstRawSelected.Items.Clear()

'loops through raw ingredients based on prepped item

For Each item In gdicPreppedItems(lstPreppedSelected.SelectedItem.ToString())

lstRawSelected.Items.Add(item.Key)

Next

End Sub

'------------------------------------------------------------

'- Subprogram Name: DisplayDishes -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user wants to

'- display the available dishes

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Private Sub DisplayDishes()

'clears list boxes

lstDish.Items.Clear()

lstPreppedSelected.Items.Clear()

lstRawSelected.Items.Clear()

'loops through all dishes and populates list box

For Each dish In gdicDishes.Keys

lstDish.Items.Add(dish)

Next

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPrep\_MouseDown -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user pressed down

'- on a mouse from lstPrep

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- data - sets data object

'- effect - adds effect of copying

'- lst - listbox sets as a directcast

'------------------------------------------------------------

Private Sub lstPrep\_MouseDown(sender As Object, e As MouseEventArgs) Handles lstPrep.MouseDown

'initializes the listbox, data object and effects

Dim lst As ListBox = DirectCast(sender, ListBox)

Dim data As New DataObject()

Dim effect As DragDropEffects = DragDropEffects.Copy

'sets data to the selected item

data.SetData(lstPrep.GetItemText(lstPrep.SelectedItem))

effect = lst.DoDragDrop(data, effect)

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPreppedSelected\_DragEnter -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user enters a listbox

'- while something is being dragged into it

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None)

'------------------------------------------------------------

Private Sub lstPreppedSelected\_DragEnter(sender As Object, e As DragEventArgs) Handles lstPreppedSelected.DragEnter

e.Effect = DragDropEffects.Copy

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPreppedSelected\_DragDrop -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user performs a

'- drag-drop event on lstPreppedSelected

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strMyString - temporary string to hold the string value being copied -

'------------------------------------------------------------

Private Sub lstPreppedSelected\_DragDrop(sender As Object, e As DragEventArgs) Handles lstPreppedSelected.DragDrop

'creates a temporary string to hold the string value

Dim strMyString As String = e.Data.GetData(DataFormats.Text).ToString

'Adds the newly selected prepped item to the dish

AddDish(lstDish.GetItemText(lstDish.SelectedItem), strMyString)

'Clears list boxes

lstPreppedSelected.Items.Clear()

lstRawSelected.Items.Clear()

'Displays the prepped items in the list box

DisplayPreppedItems(lstDish.GetItemText(lstDish.SelectedItem))

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstRaw\_MouseDown -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user pressed down

'- on a mouse from lstRaw

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- data - sets data object

'- effect - adds effect of copying

'- lst - listbox sets as a directcast -

'------------------------------------------------------------

Private Sub lstRaw\_MouseDown(sender As Object, e As MouseEventArgs) Handles lstRaw.MouseDown

'initializes the listbox, data object and effects

Dim lst As ListBox = DirectCast(sender, ListBox)

Dim data As New DataObject()

Dim effect As DragDropEffects = DragDropEffects.Copy

'sets data to the selected item

data.SetData(lstRaw.GetItemText(lstRaw.SelectedItem))

effect = lst.DoDragDrop(data, effect)

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstRawSelected\_DragEnter -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user enters a listbox

'- while something is being dragged into it

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None)

'------------------------------------------------------------

Private Sub lstRawSelected\_DragEnter(sender As Object, e As DragEventArgs) Handles lstRawSelected.DragEnter

e.Effect = DragDropEffects.Copy

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstRawSelected\_DragDrop -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user performs a

'- drag-drop event on lstRawSelected

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strMyString - temporary string to hold the string value being copied -

'------------------------------------------------------------

Private Sub lstRawSelected\_DragDrop(sender As Object, e As DragEventArgs) Handles lstRawSelected.DragDrop

'creates a temporary string to hold the string value

Dim strMyString As String = e.Data.GetData(DataFormats.Text).ToString

'Adds the newly selected prepped item to the dish

AddPreppedItem(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem), strMyString)

'Clears list boxes

lstRawSelected.Items.Clear()

'Displays the raw items in the list box

DisplayRawItems(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem))

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPreppedSelected\_MouseDown -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user pressed down

'- on a mouse from lstPreppedSelected

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- data - sets data object

'- effect - adds effect of copying

'- lst - listbox sets as a directcast -

'------------------------------------------------------------

Private Sub lstPreppedSelected\_MouseDown(sender As Object, e As MouseEventArgs) Handles lstPreppedSelected.MouseDown

If blnIsDoDragDropActive = True Then

Else

'initializes the listbox, data object and effects

Dim lst As ListBox = DirectCast(sender, ListBox)

Dim data As New DataObject()

Dim effect As DragDropEffects = DragDropEffects.Copy

'sets data to the selected item

data.SetData(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem))

effect = lst.DoDragDrop(data, effect)

End If

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPrep\_DragEnter -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user enters a listbox

'- while something is being dragged into it

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None)

'------------------------------------------------------------

Private Sub lstPrep\_DragEnter(sender As Object, e As DragEventArgs) Handles lstPrep.DragEnter

e.Effect = DragDropEffects.Copy

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstPrep\_DragDrop -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user performs a

'- drag-drop event on lstPrep

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strMyDish - temporary string to hold name of dish selected

'- strMyString - temporary string to hold the string value being copied

'------------------------------------------------------------

Private Sub lstPrep\_DragDrop(sender As Object, e As DragEventArgs) Handles lstPrep.DragDrop

blnIsDoDragDropActive = True

'creates a temporary string to hold the string value

Dim strMyString As String = e.Data.GetData(DataFormats.Text).ToString

'sets the dish

Dim strMyDish As String = lstDish.GetItemText(lstDish.SelectedItem)

'removes from dictionary and listbox

gdicDishes.Item(strMyDish).Remove(strMyString)

'clears and repopulates listbox

lstPreppedSelected.Items.Clear()

DisplayPreppedItems(lstDish.GetItemText(lstDish.SelectedItem))

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstRawSelected\_MouseDown -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user pressed down

'- on a mouse from lstRawSelected

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- data - sets data object

'- effect - adds effect of copying

'- lst - listbox sets as a directcast -

'------------------------------------------------------------

Private Sub lstRawSelected\_MouseDown(sender As Object, e As MouseEventArgs) Handles lstRawSelected.MouseDown

'initializes the listbox, data object and effects

Dim lst As ListBox = DirectCast(sender, ListBox)

Dim data As New DataObject()

Dim effect As DragDropEffects = DragDropEffects.Copy

'sets data to the selected item

data.SetData(lstRawSelected.GetItemText(lstRawSelected.SelectedItem))

effect = lst.DoDragDrop(data, effect)

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstRaw\_DragEnter -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user enters a listbox

'- while something is being dragged into it

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None)

'------------------------------------------------------------

Private Sub lstRaw\_DragEnter(sender As Object, e As DragEventArgs) Handles lstRaw.DragEnter

e.Effect = DragDropEffects.Copy

End Sub

'------------------------------------------------------------

'- Subprogram Name: lstRaw\_DragDrop -

'------------------------------------------------------------

'- Written By: Austin Rippee -

'- Written On: March 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user performs a

'- drag-drop event on lstRaw

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strMyPrep - temporary string to hold name of prepped selected

'- strMyString - temporary string to hold the string value being copied

'------------------------------------------------------------

Private Sub lstRaw\_DragDrop(sender As Object, e As DragEventArgs) Handles lstRaw.DragDrop

'creates a temporary string to hold the string value

Dim strMyString As String = e.Data.GetData(DataFormats.Text).ToString

'sets the dish

Dim strMyPrep As String = lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem)

'removes from dictionary and listbox

gdicPreppedItems.Item(strMyPrep).Remove(strMyString)

'clears and repopulates the listbox

lstRawSelected.Items.Clear()

DisplayRawItems(lstPreppedSelected.GetItemText(lstPreppedSelected.SelectedItem))

End Sub

End Class

A picture containing graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated