```csharp
1  using System;
2  using System.ComponentModel;
3  using System.Windows.Forms;
4
5  namespace ContactUsWF_App
6  {
7      public partial class Form1 : Form
8      {
9          //Supplying objects to a binding source
10         private BindingList<ContactUs> entries = new BindingList<ContactUs>();
11
12         public Form1()
13         {
14             InitializeComponent();
15
16             nameText.DataBindings.Clear();// Clear collection before any bindings
17             nameText.DataBindings.Add("Text", entriesSource, "Name", true,     ⮡
                   DataSourceUpdateMode.OnPropertyChanged);
18             phoneNumber.DataBindings.Clear();// Clear collection before any      ⮡
                   bindings
19             phoneNumber.DataBindings.Add("Text", entriesSource, "PhoneNumber",  ⮡
                   true, DataSourceUpdateMode.OnPropertyChanged);
20             emailText.DataBindings.Clear();// Clear collection before any        ⮡
                   bindings
21             emailText.DataBindings.Add("Text", entriesSource, "Email", true,    ⮡
                   DataSourceUpdateMode.OnPropertyChanged);
22             subjectText.DataBindings.Clear();// Clear collection before any      ⮡
                   bindings
23             subjectText.DataBindings.Add("Text", entriesSource, "Subject", true,⮡
                   DataSourceUpdateMode.OnPropertyChanged);
24             countryText.DataBindings.Clear();// Clear collection before any      ⮡
                   bindings
25             countryText.DataBindings.Add("Text", entriesSource, "Country", true,⮡
                   DataSourceUpdateMode.OnPropertyChanged);
26             descriptionText.DataBindings.Clear();// Clear collection before any  ⮡
                   bindings
27             descriptionText.DataBindings.Add("Text", entriesSource,             ⮡
                   "Description", true, DataSourceUpdateMode.OnPropertyChanged);
28             //dueDatePicker.DataBindings.Add("Value", entriesSource, "DueDate",  ⮡
                   true, DataSourceUpdateMode.OnPropertyChanged);
29
30             entriesSource.DataSource = entries;
31
32             CreateNewItem();
33         }
34         //Controls
35
36         private void CreateNewItem()
37         {
38             ContactUs newEntry = (ContactUs)entriesSource.AddNew();
39             newEntry.Name = "New name";
40             newEntry.PhoneNumber = "New phone Number";
```

```
41                newEntry.Email = "New email";
42                newEntry.Subject = "New subject";
43                newEntry.Description = "New message";
44                newEntry.Country = "New country";
45                entriesSource.ResetCurrentItem();
46                //newEntry.DueDate = DateTime.Now;
47            }
48
49        //Handling changes
50        private void entriesSource_ListChanged(object sender,              ⇀
              ListChangedEventArgs e)
51        {
52            switch (e.ListChangedType)
53            {
54                case ListChangedType.ItemAdded:
55                    MakeListViewItemForNewEntry(e.NewIndex);
56                    break;
57                case ListChangedType.ItemDeleted:
58                    RemoveListViewItem(e.NewIndex);
59                    break;
60                case ListChangedType.ItemChanged:
61                    UpdateListViewItem(e.NewIndex);
62                    break;
63            }
64        }
65
66        //Adding new list items when new to-do entries appear
67        private void MakeListViewItemForNewEntry(int newItemIndex)
68        {
69            ListViewItem item = new ListViewItem(); //New ListViewItem create 1    ⇀
              column by default
70            item.SubItems.Add(""); // We added 5 more columns by repeating        ⇀
              item.SubItems.Add(""); because we have 6 columns to view
71            item.SubItems.Add("");
72            item.SubItems.Add("");
73            item.SubItems.Add("");
74            item.SubItems.Add("");
75            entriesListView.Items.Insert(newItemIndex, item);
76        }
77
78        //Making list view items reflect changes
79        private void UpdateListViewItem(int itemIndex)
80        {
81            ListViewItem item = entriesListView.Items[itemIndex];
82            ContactUs entry = entries[itemIndex];
83
84            item.SubItems[0].Text = entry.Name;
85            item.SubItems[1].Text = entry.PhoneNumber;
86            item.SubItems[2].Text = entry.Email;
87            item.SubItems[3].Text = entry.Subject;
88            item.SubItems[4].Text = entry.Description;
89            item.SubItems[5].Text = entry.Country;
```

```
90
91                 //item.SubItems[1].Text = entry.DueDate.ToShortDateString();
92             }
93
94         //Removing list view items for deleted entries
95         private void RemoveListViewItem(int deletedItemIndex)
96         {
97             entriesListView.Items.RemoveAt(deletedItemIndex);
98         }
99
100         private void newItemButton_Click(object sender, EventArgs e)
101         {
102             CreateNewItem();
103         }
104
105         private void deleteItemButton_Click(object sender, EventArgs e)
106         {
107             if (entriesListView.SelectedIndices.Count != 0)
108             {
109                 int entryIndex = entriesListView.SelectedIndices[0];
110                 entriesSource.RemoveAt(entryIndex);
111             }
112         }
113
114         private void entriesListView_SelectedIndexChanged(object sender,
             EventArgs e)
115         {
116             if (entriesListView.SelectedIndices.Count != 0)
117             {
118                 int entryIndex = entriesListView.SelectedIndices[0];
119                 entriesSource.Position = entryIndex;
120             }
121         }
122     }
123 }
124
```