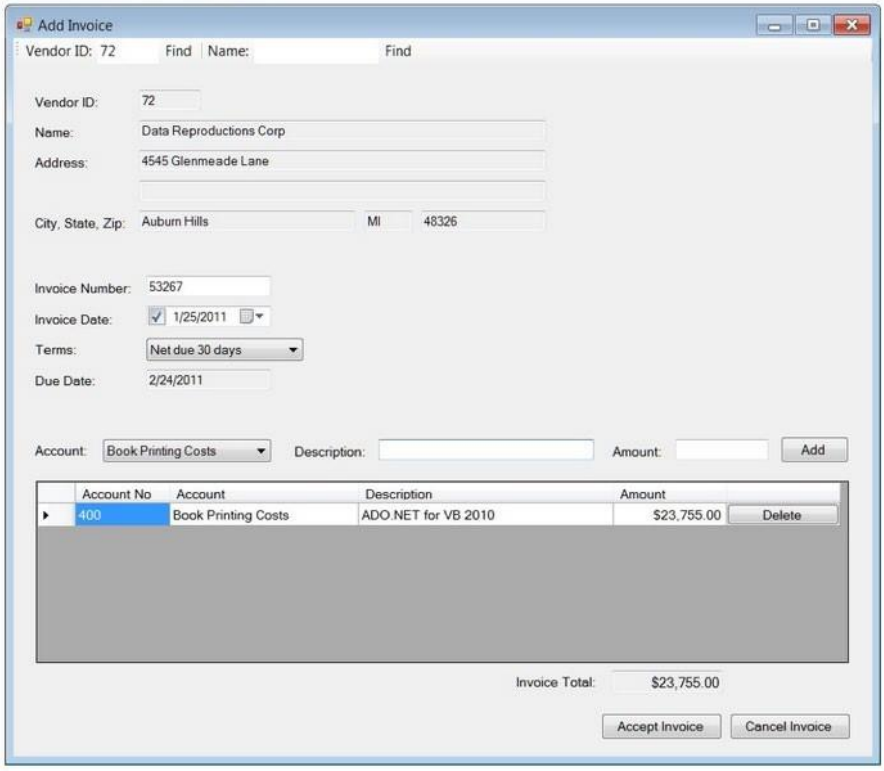
Build the Invoice Entry application

For this exercise, you'll build the application that's presented in figure 5-14 in your textbook, and shown below, from scratch. That will give you a good feel for the strengths and limitations of using data sources and datasets. As you develop this application, you can of course refer to the documentation that's presented in figures 515 through 5-17 in the textbook. But even with that, it could take you two hours or more to build this application.

The Add Invoice form



The Select Vendor form

Figure 5-14

Development guidelines

- Use the Dataset Designer to create the dataset class for this application as shown in figure 5-9 in the textbook, and use the Server Explorer to add tables to the dataset as shown in figure 5-10.
- Like any form, you should build the **Add Invoice** form from the top down with the data for one table at a time: (1) the controls that present the vendor data; (2) the controls that get the data for a new invoice; (3) the controls that get the data for a line item; and (4) the **DataGridView** control that presents the line items that have been entered for the current invoice.
- For each portion of the form, you can: (1) use the *Server Explorer* to add the required data table to the dataset; (2) drag the table or columns from the *Data Sources* window onto the form; (3) use the *Dataset Designer* to add the required queries; (4) use the *Form Designer* to set the required properties and make the required adjustments to the form; and (5) add the event handlers and modify the generated code as needed.
- Since the **Select Vendor** form just provides another way to get the data for a specific vendor, you can build this form and write the related code any time after you get the `FillByVendorID` portion of the **Add Invoice** form working right.
- As you build the application, you don't need to set all the properties right or add all the code for error handling. You just need to set the properties and add the code that affects the way the application works. Once you get the application working right, you can add those finishing touches.

Development notes

- If you have any trouble figuring out how to do something, you can open the book application and see how it does it. In fact, you may want to open this application in a separate instance of *Visual Studio*. Then, you can switch back and forth between your application and the book application whenever you want to.
- If you make a mistake as you're building the application, you can often undo it by clicking the Undo button or pressing $CCCCCCCC + ZZ$. That's often better than trying to fix your mistake.
- To display the check box in a **DateTimePicker** control like the **Invoice Date** control, set the `ShowCheckBox` property to True. This box is automatically checked when a user selects a date, so you can use the Checked property of the control to determine whether the user has selected a date.
- When you create the **InvoiceLineItems** data table, be sure to configure it so the table isn't refreshed after an insert operation. Then, you can use the technique

described in figure 5-8 to edit the `Select` statement for the main query so it includes the Description column from the **GLAccounts** table. When you accept this query, be sure to click the No button in the dialog box that's displayed so that the `Insert`, `Update`, and `Delete` statements that are generated from the main query aren't modified based on the change you made.