Lab Answer Key: Module 8: Accessing Remote Data

Lab: Retrieving and Modifying Grade Data in the Cloud

Exercise 1: Creating a WCF Data Service for the SchoolGrades Database

Task 1: Create the Grades.Web project

1. Start the MSL-TMG1 virtual machine if it is not already running.
2. Start the 20483B-SEA-DEV11 virtual machine.
3. Log on to Windows 8 as **Student** with the password **Pa$$w0rd**. If necessary, click **Switch User** to display the list of users.
4. Switch to the Windows 8 Start window and then type Explorer.
5. In the **Apps** list, click **File Explorer**.
6. In File Explorer, navigate to the **E:\Mod08\Labfiles\Databases** folder, and then double-click **SetupSchoolGradesDB.cmd**.
7. Close File Explorer.
8. Switch to the Windows 8 **Start** window.
9. Click **Visual Studio 2012**.
10. In Microsoft® Visual Studio®, on the **File** menu, point to **Open**, and then click **Project/Solution**.
11. In the **Open Project** dialog box, browse to **E:\Mod08\Labfiles\Starter\Exercise 1**, click **GradesPrototype.sln**, and then click **Open**.
12. In Solution Explorer, right-click the **GradesPrototype** solution, point to **Add**, and then click **New Project**.

13. In the **Add New Project** dialog box, in the left pane expand **Visual C#**, and then click **Web**.

14. In the templates list, click **ASP.NET Empty Web Application**.

15. In the **Name** box, type **Grades.Web**, and then click **OK**.

16. In Solution Explorer, right-click the **Grades.Web** project, and then click **Properties**.

17. On the **Web** tab, in the **Start Action** section, click **Don’t open a page. Wait for a request from an external application**.

18. In the **Servers** section, ensure that **Use Local IIS Web server** is selected.

19. In the **Project Url** box, type **http://localhost:1650/**, and then click **Create Virtual Directory**.

20. In the **Microsoft Visual Studio** dialog box, click **OK**.

21. In Solution Explorer, right-click the **GradesPrototype** solution, and then click **Set StartUp Projects**.

22. In the **Solution 'GradesPrototype' Property Pages** dialog box, click **Multiple startup projects**.

23. In the **Action** column for **Grades.Web** and **GradesPrototype**, click **Start**, and then click **OK**.

24. On the **File** menu, click **Save All**.

Task 2: Add a data service to the Grades.Web project

1. In Solution Explorer, right-click the **Grades.Web** project, point to **Add**, and then click **New Folder**.

2. Delete the existing folder name, type **Services**, and then press Enter.

3. Right-click **Services**, point to **Add**, and then click **New Item**.

4. In the templates list, click **WCF Data Service**.

5. In the **Name** box, type **GradesWebDataService**, and then click **Add**.

6. Right-click **Grades.Web**, and then click **Add Reference**.

7. In the **Reference Manager – Grades.Web** dialog box, expand **Solution**, and then select **Grades.DataModel**.

8. Click **Browse**.

9. In the **Select the files to reference** dialog box, browse to the **E:\Mod08\Labfiles\Starter\Exercise 1\packages\EntityFramework.5.0.0\lib\net45** folder, click **EntityFramework.dll**, and then click **Add**.

10. In the **Reference Manager – Grades.Web** dialog box, click **OK**.

11. In Solution Explorer, expand the **GradesPrototype** project, and then double-click **App.config**.

12. In the code editor, copy the following XML to the clipboard:

```
<connectionStrings>
    <add name="GradesDBEntities"
connectionString="metadata=res://*/GradesModel.csdl|res://*/Gra
desModel.ssdl|res://*/
GradesModel.msl;provider=System.Data.SqlClient;provider
connection string=&amp;quot;data
source=(localdb)\v11.0;initial
catalog=SchoolGradesDB;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework
&amp;quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

13. In Solution Explorer, in the **Grades.Web** project, double-click **Web.config**.

14. Click at the end of the opening **<configuration>** element, press Enter, and then

paste the contents of the clipboard.

Task 3: Specify the GradesDBEntities data context for the data service

1. In Solution Explorer, in Grades.Web, expand **Services**, and then double-click **GradesWebDataService.svc**.
2. In the code editor, click at the end of the **using System.Web;** code, press Enter, and then type the following code:

```
using Grades.DataModel;
```

3. On the **View** menu, click **Task List**.
4. In the **Task List** window, in the **Categories** list, click **Comments**.
5. Double-click the **TODO: put your data source class name here** task.
6. In the code editor, delete the code **/* TODO: put your data source class name here */**, and then type the following code:

```
SchoolGradesDBEntities
```

7. In the **Task List** window, double-click the **TODO: set rules to indicate which entity sets and service operations are visible, updatable, etc.** task.
8. In the code editor, click at the end of the comment line, press Enter, and then type the following code:

```
// Configure all entity sets to permit read and write access.
config.SetEntitySetAccessRule("Grades", EntitySetRights.All);

config.SetEntitySetAccessRule("Teachers", EntitySetRights.All);
config.SetEntitySetAccessRule("Students", EntitySetRights.All);
config.SetEntitySetAccessRule("Subjects", EntitySetRights.All);
```

```
config.SetEntitySetAccessRule("Subjects", EntitySetRights.All);
config.SetEntitySetAccessRule("Users", EntitySetRights.All);
```

Task 4: Add an operation to retrieve all of the students in a specified class

1. Click after the closing brace for the **InitializeService** method, press Enter twice, and then type the following code:

```
// Find all students in a specified class.
[WebGet]
public IEnumerable<Student> StudentsInClass(string className)
{
    var students = from Student s in
this.CurrentDataSource.Students
                   where String.Equals(s.Teacher.Class,
className)
                   select s;
    return students;
}
```

2. In the **Task List** window, double-click the **TODO: set rules to indicate which entity sets and service operations are visible, updatable, etc.** task.
3. In the code editor, click at the end of the comment line, press Enter, and then type the following code:

```
// Configure the StudentsInClass operation as read-only.
config.SetServiceOperationAccessRule("StudentsInClass",
ServiceOperationRights.AllRead);
```

Task 5: Build and test the data service

1. On the **Build** menu, click **Build Solution**.
2. In Solution Explorer, in the **Grades.Web** project, in the **Services** folder, right-click **GradesWebDataService.svc**, and then click **View in Browser (Internet Explorer)**.
3. In Internet Explorer, if the message **Intranet settings are turned off by default**, click **Don't show this message again**.
4. Verify that Internet Explorer® displays an XML description of the entities that the data service exposes.
5. Close Internet Explorer.
6. In Visual Studio, on the **File** menu, click **Close Solution**.

Results: After completing this exercise, you should have added a WCF Data Service to the application to provide remote access to the **SchoolGrades** database.

Exercise 2: Integrating the Data Service into the Application

Task 1: Add a service reference for the WCF Data Service to the GradesPrototype application

1. In Visual Studio, on the **File** menu, point to **Open**, and then click **Project/Solution**.
2. In the **Open Project** dialog box, browse to **E:\Mod08\Labfiles\Starter\Exercise 2**, click **GradesPrototype.sln**, and then click **Open**.
3. In Solution Explorer, right-click the **GradesPrototype** solution, and then click **Set StartUp Projects**.

4. In the **Solution 'GradesPrototype' Property Pages** dialog box, click **Multiple startup projects**.

5. In the **Action** column for **Grades.Web** and **GradesPrototype**, click **Start**, and then click **OK**.

6. On the **Build** menu, click **RebuildSolution**.

7. In Solution Explorer, expand **GradesPrototype**, expand **References**, right-click **Grades.DataModel,** and then click **Remove**.

8. Right-click **References**, and then click **Add Service Reference**.

9. In the **Add Service Reference** dialog box, in the **Address** box, type **http://localhost:1650**, and then click **Discover**.

10. In the **Namespace** box, type **Grades.DataModel**, and then click **OK**.

11. In the Solution Explorer toolbar, click **Show All Files**.

12. In Solution Explorer, in the **GradesPrototype** project, in the **Service References** folder, expand **Grades.DataModel**, expand **Reference.datasvcmap**, and then double-click **Reference.cs**.

13. In the code editor, modify the **namespace GradesPrototype.Grades.DataModel** code to look like the following code:

```
namespace Grades.DataModel
```

14. In Solution Explorer, right-click the **GradesPrototype** project, point to **Add**, and then click **New Folder**.

15. Delete the existing name, type **DataModel**, and then press Enter.

16. In Solution Explorer, expand the **Grades.DataModel** project, right-click **Classes.cs**, and then click **Copy**.

17. In GradesPrototype, right-click **DataModel**, and then click **Paste**.

18. In Grades.DataModel, right-click **customGrade.cs**, and then click **Copy**.

19. In GradesPrototype, right-click **DataModel**, and then click **Paste**.
20. In Grades.DataModel, right-click **customTeacher.cs**, and then click **Copy**.
21. In GradesPrototype, right-click **DataModel**, and then click **Paste**.

Task 2: Modify the code that accesses the EDM to use the WCF Data Service

1. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2a: Specify the URL of the GradesWebDataService** task.
2. In the code editor, below the comment, click inside the parentheses, and then type the following code:

```
new
Uri("http://localhost:1650/Services/GradesWebDataService.svc")
```

3. Add the following code to the **SessionContext** class, after the **Save** method:

```
static SessionContext()
{
    DBContext.MergeOption =
System.Data.Services.Client.MergeOption.PreserveChanges;
}
```

4. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2b: Load User and Grades data with Students** task.
5. In the code editor, at the end of the comment, press Enter, and then type the following code:

```
SessionContext.DBContext.LoadProperty(student, "User");
SessionContext.DBContext.LoadProperty(student, "Grades");
```

6. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2c: Load User and Students data with Teachers** task.

7. In the code editor, below the comment, click at the end of the **SessionContext.DBContext.Teachers** code, and then type the following code:

```
.Expand("User, Students")
```

8. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2d: Load User and Grades data with Students** task.

9. In the code editor, below the comment, click at the end of the **SessionContext.DBContext.Students** code, and then type the following code:

```
.Expand("User, Grades")
```

10. In Solution Explorer, in the **GradesPrototype** project, expand **DataModel**, and then double-click **customTeacher.cs**.

11. Click at the end of the **usingSystem.Threading.Tasks;** code, press Enter, and then type the following code:

```
using GradesPrototype.Services;
```

12. In the code editor, locate the **TODO: Exercise 2: Task 2e: Refer to the Students collection in the SessionContext.DBContext object** comment in the customTeacher.cs file. There are two comments with this text. This is because the comment is located in the customTeacher.cs file that you copied from the Grades.DataModel project. Make sure that you modify the customTeacher.cs file in the GradesPrototype project.

13. In the line below the comment, delete the word **Students**, and then type the following code:

```
SessionContext.DBContext.Students
```

14. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2f: Reference the SessionContext.DBContext.Students collection**.

15. In the code editor, below the comment, change **SessionContext.DBContext.Students.Local** to the following code:

```
SessionContext.DBContext.Students.Expand("User, Grades")
```

16. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2g: Use the AddToGrades method to add a new grade**.

17. In the code editor, below the comment, change **SessionContext.DBContext.Grades.Add(newGrade);** to the following code:

```
SessionContext.DBContext.AddToGrades(newGrade);
```

18. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2h: Load Subject data with Grades**.

19. In the code editor, below the comment, change **SessionContext.DBContext.Grades** to the following code:

```
SessionContext.DBContext.Grades.Expand("Subject")
```

20. In the **Task List** window, double-click the **TODO: Exercise 2: Task 2i: Use the AddToStudents method to add a new student**.

21. In the code editor, below the comment, change **SessionContext.DBContext.Students.Add(newStudent);** to the following code:

```
SessionContext.DBContext.AddToStudents(newStudent);
```

Task 3: Modify the code that saves changes back to the database to use the WCF Data Service

1. In the **Task List** window, double-click the **TODO: Exercise 2: Task 3a: Specify that the selected student has been changed** task.
2. In the code editor, click in the blank space below the comment, and then type the following code:

```
SessionContext.DBContext.UpdateObject(student);
```

3. In the **Task List** window, double-click the **TODO: Exercise 2: Task 3b: Specify that the current student has been changed** task.
4. In the code editor, click in the blank space below the comment, and then type the following code:

```
SessionContext.DBContext.UpdateObject(SessionContext.CurrentStudent);
```

5. In the **Task List** window, double-click the **TODO: Exercise 2: Task 3c: Specify that the current user has been changed** task.
6. Click in the blank space below the comment, and then type the following code:

```
SessionContext.DBContext.UpdateObject(currentUser);
```

Task 4: Build and test the application to verify that the application still functions correctly

1. On the **Build** menu, click **Build Solution**.
2. On the **Debug** menu, click **Start Without Debugging**.
3. In the **Username** box, type **vallee**.
4. In the **Password** box, type **password99**, and then click **Log on**.
5. In the students list, click **Eric Gruber**, and then click **Remove Student**.
6. In the **Confirm** dialog box, click **Yes**.
7. Verify that **Eric Gruber** is removed from the student list.
8. Click **Enroll Student**.
9. In the **Assign Student** dialog box, click **Jon Orton**.
10. In the **Confirm** dialog box, click **Yes**.
11. In the **Assign Student** dialog box, click **Close**, and then verify that **Jon Orton** is added to the student list.
12. Click **Change Password**.
13. In the **Change Password Dialog** dialog box, in the **Old Password** box, type **password99**.
14. In the **New Password** box, type **password88**.
15. In the **Confirm** box, type **password88**, and then click **OK**.
16. In the **Password** dialog box, click **OK**, and then click **Log off**.
17. Click **Log on**, verify that the **Logon Failed** dialog box appears, and then click **OK**.
18. In the **Password** box, type **password88**, and then click **Log on**.
19. Verify that the student list is displayed.

20. Click **Log off**, and then in the **Username** box, type **grubere**.
21. In the **Password** box, type **password**, and then click **Log on**.
22. Verify that the student profile for Eric Gruber appears, and then click **Log off**.
23. Close the application.
24. In Visual Studio, on the **File** menu, click **Close Solution**.

Results: After completing this exercise, you should have updated the Grades Prototype application to use the WCF Data Service.

Exercise 3: Retrieving Student Photographs Over the Web (If Time Permits)

Task 1: Create the ImageNameConverter value converter class

1. In Visual Studio, on the **File** menu, point to **Open**, and then click **Project/Solution**.
2. In the **Open Project** dialog box, browse to **E:\Mod08\Labfiles\Starter\Exercise 3**, click **GradesPrototype.sln**, and then click **Open**.
3. In Solution Explorer, right-click the **GradesPrototype** solution, and then click **Set StartUp Projects**.
4. In the **Solution 'GradesPrototype' Property Pages** dialog box, click **Multiple startup projects**.
5. In the **Action** column for **Grades.Web** and **GradesPrototype**, click **Start**, and then click **OK**.
6. On the **Build** menu, click **RebuildSolution**.
7. In the **Task List** window, double-click the **TODO: Exercise 3: Task 1: Create the ImageNameConverter value converter to convert the image name of a**

student photograph into the URL of the image on the Web server task.

8. Click at the end of the **// Converter class for transforming an image name for a photograph into a URL** comment, press Enter, and then type the following code:

```
public class ImageNameConverter : IValueConverter
{
}
```

9. In the **ImageNameConverter** class, type the following code:

```
const string webFolder =
"http://localhost:1650/Images/Portraits/";
```

10. Right-click the **IValueConverter** keyword, point to **Implement Interface**, and then click **Implement Interface**.
11. In the **Convert** method, delete the existing statement that throws a **NotImplementedException**, and then type the following code:

```
string fileName = value as string;
if (fileName != null)
{
    return string.Format("{0}{1}", webFolder, fileName);
}
else
{
    return string.Empty;
}
```

12. On the **Build** menu, click **Build Solution**.

Task 2: Add an Image control to the StudentsPage view and bind it to the ImageName property

1. In Solution Explorer, in the **GradesPrototype** project, expand **Views**, and then double-click **StudentsPage.xaml**.

2. In the XAML editor, in the **UserControl** element at the top of the markup, click after the **xmlns:d="http://schemas.microsoft.com/expression/blend/2008"** line, press Enter, and then type the following markup:

```
xmlns:local="clr-namespace:GradesPrototype.Views"
```

3. Locate the **TODO: Exercise 3: Task 2a. Add an instance of the ImageNameConverter class as a resource to the view** comment, click at the end of the comment, press Enter, and then type the following markup:

```
<UserControl.Resources>
    <local:ImageNameConverter x:Key="ImageNameConverter"/>
</UserControl.Resources>
```

4. Locate the **TODO: Exercise 3: Task 2b. Add an Image control to display the photo of the student** comment, click at the end of the comment, press Enter, and then type the following markup:

```
<Image Height="100" />
```

5. Locate the **Exercise 3: Task 2c. Bind the Image control to the ImageName property and use the ImageNameConverter to convert the image name into a URL** comment.

6. In the line above the comment, modify the **<Image Height="100" />** markup to look like the following markup:

```
<Image Height="100" Source="{Binding ImageName, Converter=
{StaticResource
ImageNameConverter}}" />
```


Task 3: Add an Image control to the StudentProfile view and bind it to the ImageName property

1. In Solution Explorer, double-click **StudentProfile.xaml**.
2. Locate the **TODO: Exercise 3: Task 3a. Add an instance of the ImageNameConverter class as a resource to the view** comment.
3. Click at the end of the comment, press Enter, and then type the following markup:

```
<app:ImageNameConverter x:Key="ImageNameConverter"/>
```

4. Locate the **TODO: Exercise 3: Task 3b. Add an Image control to display the photo of the student and bind the Image control to the ImageName property and use the ImageNameConverter to convert the image name into a URL** comment.
5. Click at the end of the comment, press Enter, and then type the following markup:

```
<Image Height="150" Source="{Binding ImageName, Converter=
{StaticResource
ImageNameConverter}}" />
```

Task 4: Add an Image control to the AssignStudentDialog control and bind it to the ImageName property

1. In Solution Explorer, expand **Controls**, and then double-click **AssignStudentDialog.xaml**.

2. In the XAML editor, at the top of the markup, click at the end of the **xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"** line, press Enter, and then type the following markup:

```
xmlns:local="clr-namespace:GradesPrototype.Views"
```

3. Locate the **TODO: Exercise 3: Task 4a. Add an instance of the ImageNameConverter class as a resource to the view** comment.

4. Click at the end of the comment, press Enter, and then type the following markup:

```
<Window.Resources>

    <local:ImageNameConverter x:Key="ImageNameConverter"/>
</Window.Resources>
```

5. Locate the **TODO: Exercise 3: Task 4b. Add an Image control to display the photo of the student and bind the control to the ImageName property and use the ImageNameConverter to convert the image name into a URL** comment.

6. Click at the end of the comment, press Enter, and then type the following markup:

```
<Image Height="100" Source="{Binding ImageName, Converter=
{StaticResource
ImageNameConverter}}" />
```

Task 5: Build and test the application, verifying that student's photographs appear in the list of students for the teacher

1. On the **Build** menu, click **Build Solution**.
2. On the **Debug** menu, click **Start Without Debugging**.
3. In the **Username** box, type **vallee**.
4. In the **Password** box, type **password88**, and then click **Log on**.
5. Verify that the students list now includes images.
6. Click **George Li**, and then verify that the student profile appears with an image.
7. Click **Remove Student**.
8. In the **Confirm** dialog box, click **Yes**.
9. Click **Enroll Student**.
10. In the **Assign Student** dialog box, the unassigned students each have an image.
11. Click **George Li**.
12. In the **Confirm** dialog box, click **Yes**.
13. In the **Assign Student** dialog box, click **Close**.
14. Verify that George Li is added to the students list with an image.
15. Close the application.
16. On the **File** menu, click **Close Solution**.

Results: After completing this exercise, the students list, student profile, and unassigned student dialog box will display the images of students that were

retrieved across the web.