MALMÖ HÖGSKOLA
Malmö University
School of Technology

Programming Using .NET, Advanced Course

Data Access using ADO.NET and SQL Server Express 2012 (LocalDB)

Assignment 4 – Animal Motel Version 4

Mandatory

Farid Naisan
University Lecturer
Department of Computer Sciences

Apu's Animal Motel - Version 4

1 Objectives

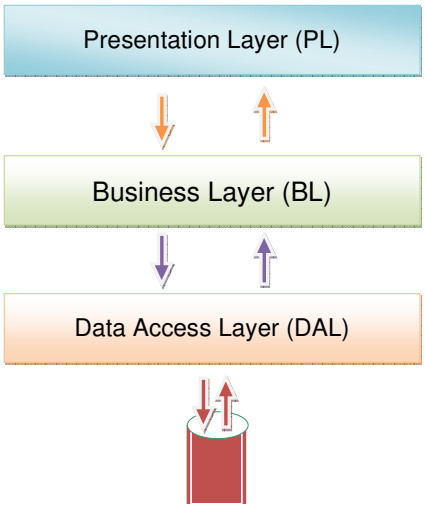
This assignment aims mainly to let you explore how .NET manages connection with databases. More specifically, this assignment provides training opportunities in the following areas:

1. Data access using ADO.NET.
2. Creation of a SQL database using SQL Server 2012 Express (LocalDB).
3. Design and implementation of a database using tables, attributes and SQL code.
4. Managing communication between your application and a database through a Data Access Layer (DAL).

2 Description

In this assignment, we will produce a final version of our Animal Motel project, based on the previous version released in the last assignment. In the last version, we worked with data persistence using object serialization. Object serialization and deserialization is an important feature in storing and transferring data to different devices. However, using databases for storing data is perhaps most optimal, especially due to the fact that the Database Management Systems (DBMS) that handle databases are powerful engines providing a lot of services. In addition to features and facilities, DBMS take care of security issues as well.

ADO.NET is Microsoft's technology for working with data sources of different types, databases, XML files and even objects. While letting the database design and other purely database-related issues be the concern of database courses, our focus in this course, and particularly in this assignment, is to learn and gain knowledge in how to communicate with databases from our applications in an object-oriented manner.



ADO.NET is an object-oriented set of libraries that allows you to interact with data sources. Commonly, the data source is a database, but it could also be a text file, an Excel spreadsheet, or an XML file. There are many different types of databases available. For example, there is the Microsoft SQL Server, Microsoft Access, Oracle, just to name a few.

We are going to use a disconnected approach using **DataSets**, **DataAdapters** and **CommandBuilders**. Although the connected model (working directly with the database) has its usage in some occasions, the disconnected layer is the most popular model as it allows you to work with a copy of the database in memory. You open the database read data and close the database. You work with your database in memory and when you want to update the database, you proceed in the revere order, open the database, save data and close the database.

In addition, applying a structured component-oriented model in laying out a maintainable solution for our

programming projects is also one of the main goals in this course. In our previous assignments, we took a first step in this direction and separated the presentation of data from the logics that processed data. We divided our software project into a Presentation Layer (PL, an EXE project) and a Business Logics Layer (BLL, a DLL-project).

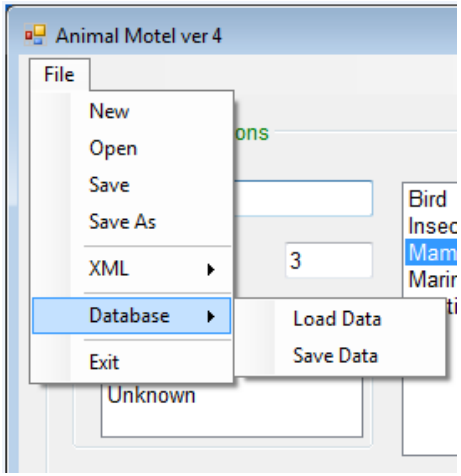
In this assignment, we separate these two layers from data logics and thus create a third layer, the Data Access Layer (DAL). This layer (a DLL project) is to be responsible for all data processing between a data source (database) and the BLL.

3 To Do

As mentioned earlier, in the last assignment we worked with serialization of data. In this assignment we will instead look at how to use ADO.Net and databases to achieve the same goal. This will be done without a need for extending or modifying our GUI more than a minimal amount. We´ll just be adding a submenu within our **File** menu for accessing the database we are going to create.

3.1 More precisely, the **AnimalManager** contains a collection that maintains data for animal object, all you need to do in this assignment is to save or update animal data from the collection, for instance, List <Animal> to database and load the database to the collection List<Animal>. For a passing grade, it is good enough that you save and load objects that are of the category Mammal (as in the last assignment).

3.2 Make sure that you take a back-up copy of your earlier version of the Animal Motel from the last assignment, so you can always go back to the last good version.



3.3 Secondly, you will need to have a Database Management System (DBMS) installed on your computer. Microsoft SQL Server 2012 Express **LocalDB** integrated inside VS 2012. When connecting, the necessary SQL Server infrastructure is automatically created and started, enabling the application to use the database without complex or time consuming configuration tasks. If you are using VS2010, you can either install the LocalDB (see the links on the module) or the SQL Server 2008. This version might already be installed on your computer with the installation of Visual Studio. Otherwise, you can download it from Microsoft's website:

http://www.microsoft.com/en-us/download/details.aspx?id=26729#overview

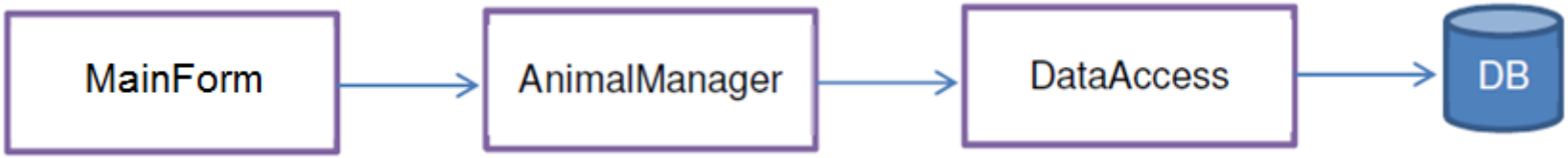
3.4 Insert a submenu item, **Database** (as in the figure), within the **File** menu for loading and saving data to/from a database that you are going to create using VS.

3.5 When the user selects the menu item **Load Data**, data is to be read from the database tables into objects in the application, and when the user selects the **Save Data** menu item, data from the animal objects are to be saved back into the database tables.

3.6 Create a DLL project for your DAL and add a reference to it in your application (PL).

4 Create your DAL project

4.1 Add a new Class Library Project to your solution. This will be your DAL-layer.



MainForm uses an object of the **AnimalManager** (just as in the previous assignment) which in turn creates and uses an object of the **DataAccess** class defined in the DAL. The DAL is the only layer that understands the language of the database. Using this pattern, you can easily change the database system or use another data source without need for many changes in the other layers – one good result of dividing your projects into components.

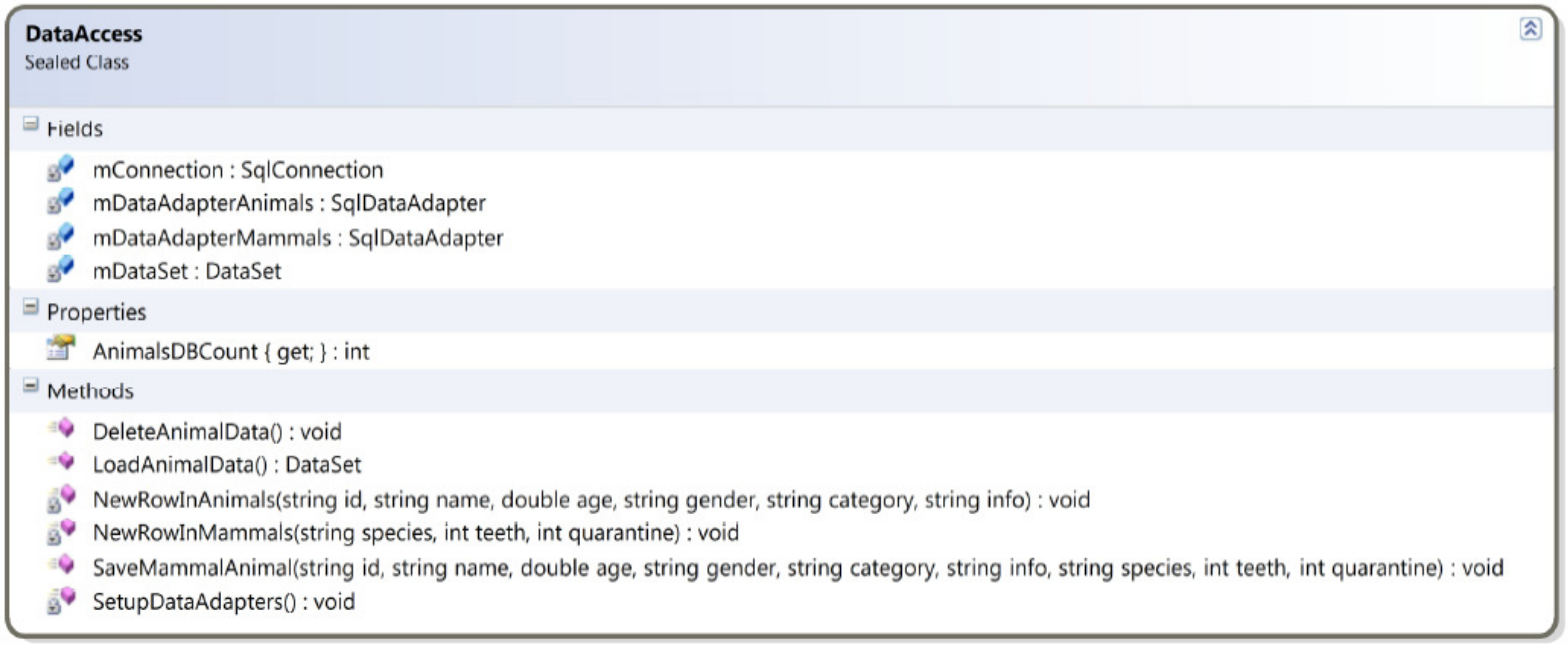
4.2 Add a database-file to your **AnimalMotel** project using Visual Studio or SQL Server Management Studio. Choose a Service Based Database (.**mdf**) and add it to your project. A service-based database is a database that is only accessed through a server. It uses an MDF data file, which is a SQL Server format. When you have a working program you will have the possibility to upload your database to a web host and try to access your database from there. You will find more information about the Web Host later in this document.

4.3 The database should now be visible in the Server Explorer window in VS. Create and design your tables using the table designer in VS. Refer to the grading criteria later in this document as to which requirements are set for achieving a passing and a higher grade. There is a video tutorial in the Module that shows how you can use VS to create a database, connect to a database, create tables and add test values.

4.4 The database should be loaded to a collection, for instance List<Animal>.and saved or updated with data from the collection, List<Animal>. Create a new class (**DataAccess.cs**) which will serve as the main class in your database layer. The class diagram below might give you an idea of the methods that you might need to write in the **DataAccess** class for providing data access services to the client objects (such as the **AnimalManager**).

4.5 You can use a **Dataset** or a **DataTable** to read data from the tables and a **DataSet** to save data back to the database. However, you may use simple method calls or other type of objects to establish communications between the **AnimalManager** and the **DataAccess** objects.

4.6 Delegate as much data handling logics and responsibilities to the **DataAccess** class as possible. Let the **AnimalManager** do the least work in loading and saving data from and to the database. Communicating with an object of the **DataAccess** class can, in essence, be done much in the same way as communicating with the Serialization class.



5 Requirements for a passing grade (ECT C, Swe G)

5.1 Create a database with at least one general data table that would work for all types of animals, specified in the Animal class (name, id, gender, category, age and extra info).

5.2 The data access logics should be programmed inside its own DAL assembly (DLL project), as explained above.

5.3 The DAL-project should not have any references to the other projects. Only the Animal-project should have a reference to the **DAL** library. As an optimization of the previous solution, you have to take the **AddAnimal** method from the **MainForm** and move it to the **AnimalManager** class. This is because the **MainForm** should not have any contact with the DAL-project. Remember that the method, when moved to the **AnimalManager** class should not access any GUI component on the **MainForm**.

5.4 Do not forget to handle possible exceptions, especially in conjunction with the input/out operations. Use a **try-catch** or the **using** statements to handle exceptions related to input/output processes.

5.5 .Compile, test and make sure that everything works. Your project should have some test data that can be loaded. **Include at least one data file that can be loaded for testing.**

5.6 Important: **DO NOT** hardcode any file path or the **connectionString**. The **connectionString** must be placed (as VS does) in the **app.config** file. It is also important to know that when using VS 2010, the SQLEXPRESS attribute must be present in the **connectionString**, as in theexample below:

```
connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\AnimalsDB.mdf;Integrated
Security=True;User Instance=True"
```

The connection string using LocalDB is different,

```
connectionString=" (localdb)\v11.0;AttachDbFileName=|DataDirectory|\AnimalsDB.mdf; ;Integrated
 Security=True;
```

5.7 Check that your database (mdf-file) is present in the BIN directory of your project; otherwise copy it there.

6 Requirements for a higher grade (Swe VG, or ECT A, B)

The following items are required only for a higher grade. You can skip this section, if you aim only at a passing grade C or Swe G, but if you are willing to qualify for higher grade, you are to accomplish the following two tasks in addition to the requirements listed above:

6.1 Write code to also include to save complete data for a species, data related to a category and data related for a special species. **Note**: it is sufficient that you select one species to load and save data for one specific species such as Dog, Cat or Sparrow. You may, for example, select the Mammal category and one of the object types (Dog, Cat, etc).. The animals should be loaded to its correct species, i.e. a dog should be saved as a Dog object.

7 Access to a Web Host

For the first time in this course we have the opportunity to give you access to a Web Host that you can upload to and access your database from. This is a new web server at MAH. Create an account using your ID and password. . Follow the instructions given at: https://ddwap.mah.se/dotnet/ to get started. Due to the fact that your teachers do not have administrative access to this server, they will unfortunately not be able to give any support or answer questions in this regard. There are help files on the mentioned site. Read these and use the forum in the module to share experiences and hints with your classmates. Using the web host is optional, but it is recommended you to get a deeper understanding of how you can work with data bases hosted at remote servers.

In addition, if it works for you on your computer, it is most likely that it will also work from other computers as well.

8 Testing and Submission

You must test your database before submitting your project. Test the exe-file a few times, load the database, add new data, save and exit the application. Start your application again, and check if the changes from the last time are still correct. Delete some data, save the database and exit again. Do this a few time and when you are sure about the performance of your application, upload the project to It's L as before.

9 Help, Hints and Tips

Most people think that they are working with the database when the database is visible in the project (the Solution Explorer), but that is not the case. Instead, when you execute your program, the database is by default copied to the bin/debug folder of your project and is accessed from there. You can see your projects bin/debug folders by choosing 'View All Files' (an icon in Solution Explorer). In the debug folder you will most likely see your database (after the first execution of your program). To see your table data you can double click on the database in VS to make it visible in Server Explorer and then you can see the tables and table data.

9.1 It is expected and understandable if you come up with many questions and that you will be in need some help. Using the SQL Server Express is not always free of headaches and more to that is that the problems are usually of local character; depending on the environment and configurations in the computer you are running the application. Therefore, it is very important that as soon as you face any problem, get active on the forum in the module and discuss the problem. There are going to be people who have long practical experience that they will share with you. Hopefully the LocalDB is easier to use. Most problems are related to the connectionString so try to see that you check this string first whenever you get connection problems.

9.2 If you have practical experience from working with databases, you may certainly use your solution. You don't have to consider all requirements given earlier if they don't fit into your solution. The important issue is that the database is not directly connected to the BLL or PL. Furthermore, this assignment has not proposed any improvements of the GUI, so if you wish to change the GUI and use other controls such **DataGridView**, just go ahead and do that.

Some hints (example code)

To get the connectionstring from the config-file:

```
private string connectionString = DBAccesLayer.Properties.Settings.Default.AnimalDBConnectionString;
```

You can use different adapters for the common animal data and specific animal data:

```
private SqlDataAdapter _dataAdapterAnimals;
private SqlDataAdapter _dataAdapterBird;

public void SetupDataAdapters()
{
    dataAdapterAnimals = new SqlDataAdapter();
    dataAdapterBird = new SqlDataAdapter();
```

```
}
```

To load an animal from the database using a dataset:

1. Open a sqlconnection

```
using (connection = new SqlConnection(connectionString))
{
     Connection.Open();
    //...
```

2. Prepare a sqlCommandString
3. Use the Fill method of the dataAdapterAnimals.

Alternatively, you can use a `SqlDataReader` to load data from the database.

For more help, see the lecture material on the module.

Good Luck!

Farid Naisan,
Course Responsible and Instructor