Exercise 3 Implementing the Shop with .NET

3.1 Overview

3.1.1 Goals:

After completing this exercise you should be able to:

- Give an overview of the component technologies in .NET
- Use the C# language to implement simple components.
- Create ASP.NET pages.
- Use the session object to store session specific data.
- Use the .NET database support to write code that uses a database.
- Compare the amount and type of work needed to accomplish the same task in the different technologies we have discussed during the course.

3.1.2 Requirements

Visual Studio 2010 should be installed on the local machine. You will need to use an IIS (Microsoft Internet Information Server), either a locally installed or the one at MDH.

3.1.3 Prerequisites

You should have read the .NET chapter in the book and know C# or a comparable programming language, since all programming will be done in C#, which is similar to Java and C++.

3.1.4 Examination

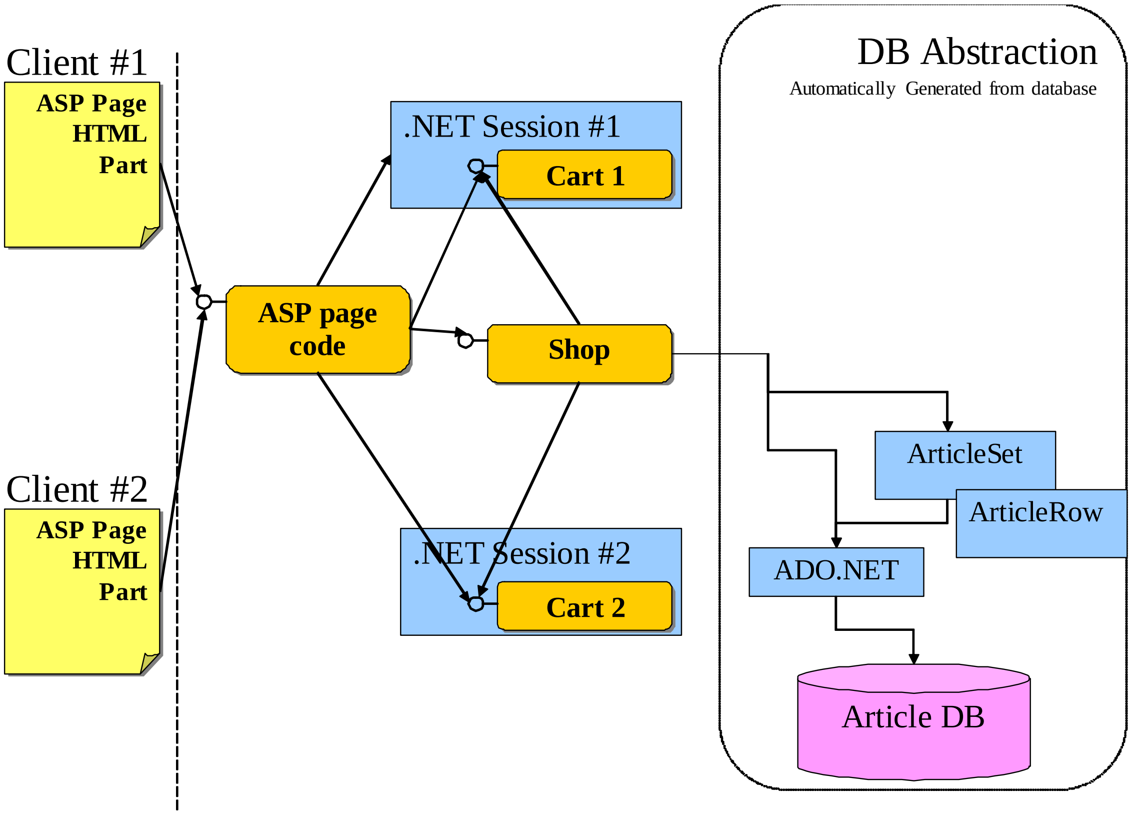
You should be able to explain all parts of your produced software.

The application should be zipped and submitted in Blackboard (see http://blackboard.mdh.se for instructions).

You are allowed to be maximum two persons in each group. Both must know – and be able to answer questions about – all parts of the produced software.

Since your final project builds upon .NET technologies, you will not have step by step instructions for this exercise.

3.2 Functional Description



You have to work out the details yourself, but here is a short scenario description of how it could work:

1. The web page is backed up by an object in the “ASP Page Code”.
2. This object locates / creates the shop object
3. The page object asks the shop for a count of articles.
4. The shop initialises the article structure from the database and prepares to deliver by index.
5. The page object asks for the article by index and receives a string (representing a specific article).
6. The user selects one or more articles.
7. The user presses the “Add” button and the page object will ask the current session for the cart, and add the selected items one by one (the cart only contains strings). A more detailed description follows:

   a. The ASP page object checks the session object (provided by .NET framework)

   b. If a shopping cart for this session exists, this is returned.

   c. If not, a new one will be created and put into the session and is then returned.

   (This procedure is necessary because the page object is created and destroyed at each request and is thus stateless.)

8. Each item is added to the shopping cart. (You typically have a list box that is bound to the cart, so the rightmost list box contains the selected items.)

9. The user may now press “Add” again to add more of the same item(s), or change the selection and then press “Add”. In addition, the user should be able to select items in the shopping cart and press “Remove”.
10. The user presses the “Buy” button
11. The page object gets the cart from the session object and does the purchase by passing the cart to the buy method of the shop object. The shop reads out the purchased objects and reduces the counts in the data base.

For simplicity we assume that the shop always has enough fruits and there is no way to cancel bought fruits (since this functionality would not add any new technologies to the exercise).

The techniques used in this exercise will be used in the project as well so it might be wise to keep an eye on how you wish to implement your repository.

Also the above description is just *one* idea of how the shop should work. If you like to, please go ahead and implement more cool features.