Database and Information Systems

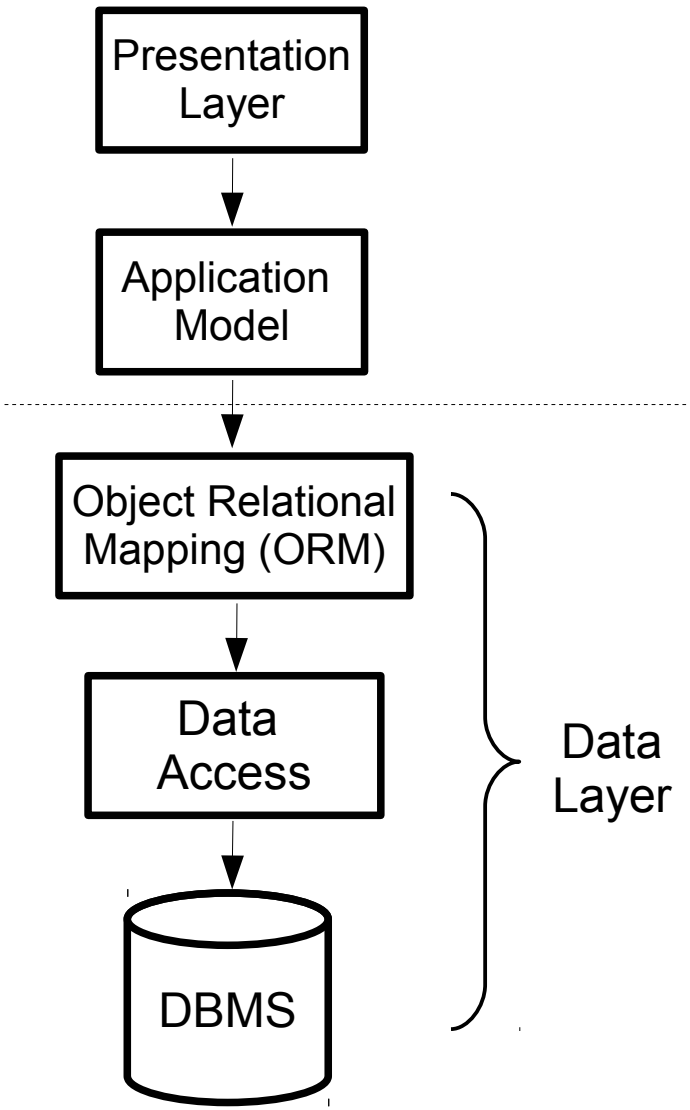
Michal Krátký

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB – Technical University of Ostrava

2020/2021

Content

Architecture of Information System



- Presentation layer: JSP, ASPX, ...
- Model: Java, C#, ...
- ORM: Hibernate, user defined ORM, ...
- Data access: JDBC, ADO.NET, ...
- DBMS: relational DBMS + stored procedure, triggers, ...

ASP.NET, Features

- We can use any programming language supported by the .NET platform.
- Source codes are compiled.
- Large class libraries are available (.NET classes).
- Name space: `System.Web.*`.
- Web Forms are used.
- We can develop web services.
- MS VisualStudio .NET 20xx

ASP.NET Structure

namespace: `System.Web.UI`

- class `Control` - encapsulation of common functions of UI:
  - `Controls`, `ID`, `Parent`, `EnableViewState`, `Visible`, `Context`, `ViewState`, ...
- class `Page` - it represents a web page
  - `Application`, `Request`, `Response`, `Server`, `Session`, `Cache`, `ErrorPage`, `IsPostBack`, `IsValid`, `Trace`, `Validators`, ...
- class `UserControl`
  - a base class for user-defined controls

ASP.NET Web Forms - Presentation Layer

- Visual components:
  - A mark-up language – HTML/WML.
  - A page of ASP.NET Web Forms – file.aspx.
  - Block: <% %>. This block is compiled.
- Program logic:
  - Object model
  - Event handling
  - Any programming language
  - It is included in the .aspx file (Code Inline) or our code is in a separated file (Code Behind), for example aspx.cs in the case of C#.

ASP.NET Controls

- Standard/Server Controls – `Label`, `CheckBox`, ...
- Data Controls – `SqlDataSource`, `ObjectDataSource`, `GridView`, `DetailView`, ...
- Validation Controls,
- Login Controls,
- Navigation Controls – `Menu`, `TreeView`, ...,
- WebParts Controls,
- HTML Controls - HTML marks,
- User Controls - user marks.

ASP.NET Server Controls

- Programmable objects on the server side, UI elements and so on.
- Objects create a page; they can have own output.
- Simple adaptation to user requirements.
- Features are set by declarations (by attributes) or by program (in code).

Server Controls

- Server controls (SC) are related with the page by elements with the attribute `runat="server"`.
- Each SC must be identified by the attribute `id`.
- This attribute is used for the manipulation with SC.
- Event handling – an event is specified by the attribute value.
  For example `asp:button` contains the attribute `Onclick` with the name of a method. We must implement this method.

Server Controls

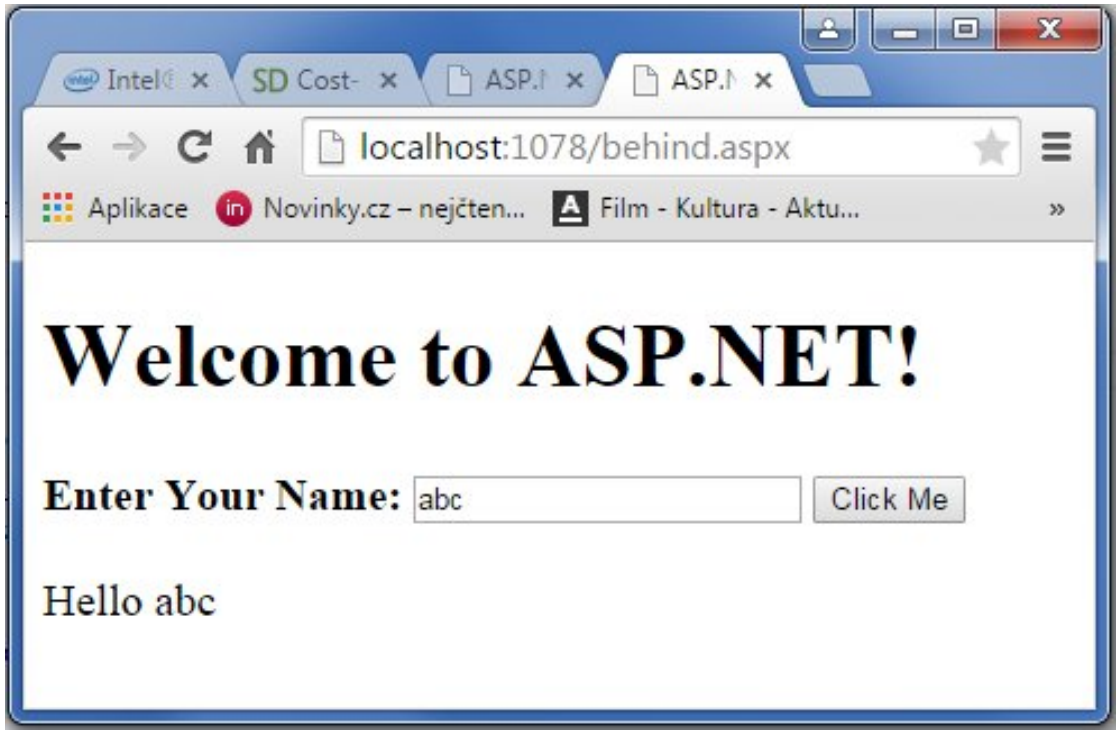
namespace: `System.Web.UI.WebControls`

AdRotator	BulletedList	Button
Calendar	CheckBox	CheckBoxList
DropDownList	FileUpload	HiddenField
HyperLink	Image	ImageButton
ImageMap	Label	LinkButton
ListBox	Literal	MultiView and View
Panel	PlaceHolder	RadioButton
RadioButtonList	Substitution	Table
TextBox	Wizard	Xml

Example - Server Controls, Notice

- The implementation is located in the aspx file, we talk about **code inline**.
- This technique is rather inappropriate since events are treat in the presentation layer. $\Rightarrow$ we use **code behind**; events are treat in separated files.

Example - Code Behind



Example - Code Behind, behind.aspx 1/3

Code Behind - each event handler is stored in a different file than the form.

```
<%@ page language="C#" CodeFile="behind.aspx.cs"
        Inherits="behind_aspx" %>
<html>
<head>
    <title>ASP.NET CodeBehind Pages</title>
</head>
<body>
    <form id="Form1" runat="server">
      <h1>Welcome to ASP.NET 2.0!</h1>
      <b>Enter Your Name:</b>
      <asp:TextBox ID="TextBox1" Runat="server" />
      <asp:Button ID="Button1" Text="Click␣Me"
        OnClick="Button1_Click" Runat="server" />
```

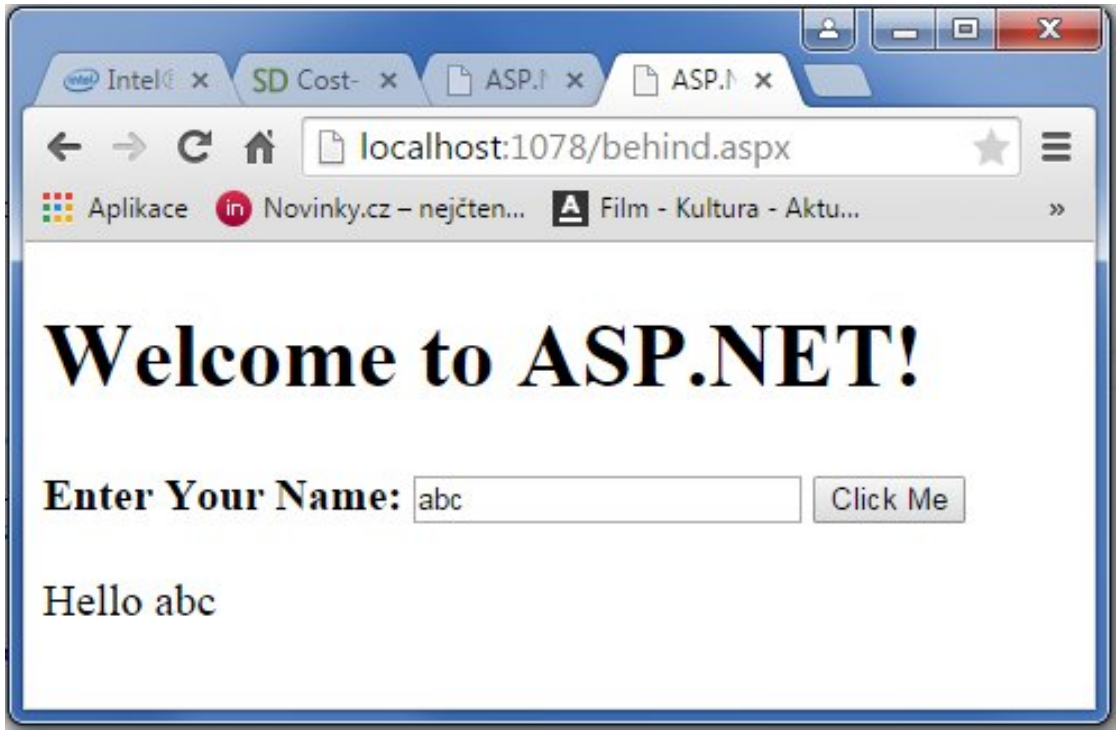
Example - Code Behind, behind.aspx 2/3

```
        <br />
        <br />
        <asp:Label ID="Label1" Text="Hello"
                   Runat="server" />
    </form>
</body>
</html>
```

Example - Code Behind, behind.aspx.cs 3/3

```
using System;

public partial class behind_aspx :
                                System.Web.UI.Page
{
  protected void Button1_Click(object sender,
                                      EventArgs e)
  {
    Label1.Text = "Hello " + TextBox1.Text;
  }
}
```

Example - Code Behind



Code Sharing among Pages

Code Directory

- Any user class (auxiliary classes, classes of object relational mapping, ...) used by ASP.NET forms must be put in the `App_Code` directory.
- The subdirectories of the directory `App_Code` must be registered in the file `Web.config`.

Global Assembly Cache

- .NET components must be registered in the file `Web.config`.
- In Visual Studio, it is processed by adding a component in the `References` directory of the project tree.

The registration is due to security reason.

Example - User Controls, usercontrols.aspx 1/2

We can create user defined tags.

```
<%@ Page Language="C#" %>
<%@ Register TagPrefix="tuo" TagName="message"
    Src="usercontrols.ascx" %>

<html>
<body style="font: 10pt verdana">
  <h3>A Simple User Control</h3>
  <tuo:message Text="Hello World!" Color="blue"
    runat="server" id="Message" />
</body>
</html>
```

Example - User Controls, usercontrols.ascx 2/2

```
<script language="C#" runat="server">
   public String Color;
   public String Text;
</script>

<span id="Message" style="color:<%=Color%>">
   <%=Text%></span>
```

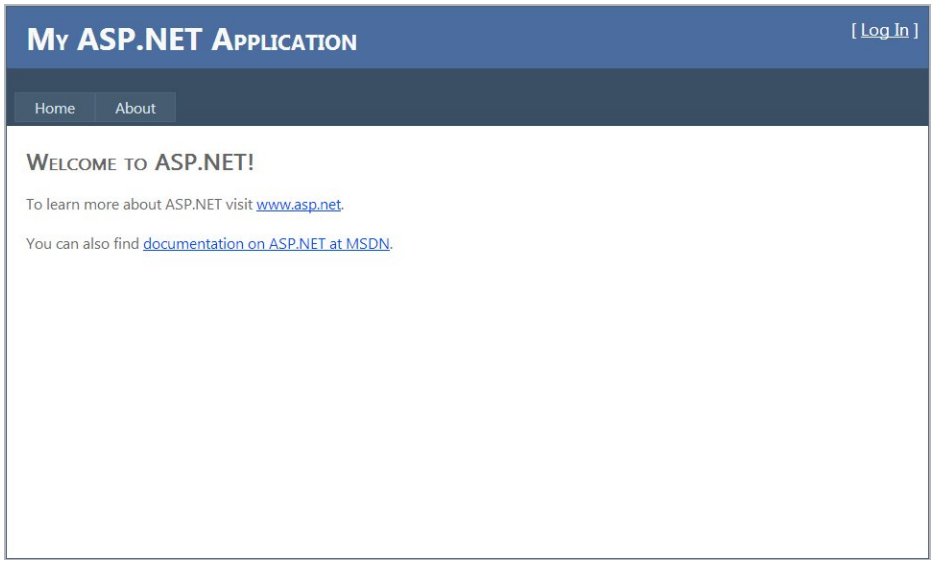
Comment: Definition of the user defined tag.

Master Page, Web.config

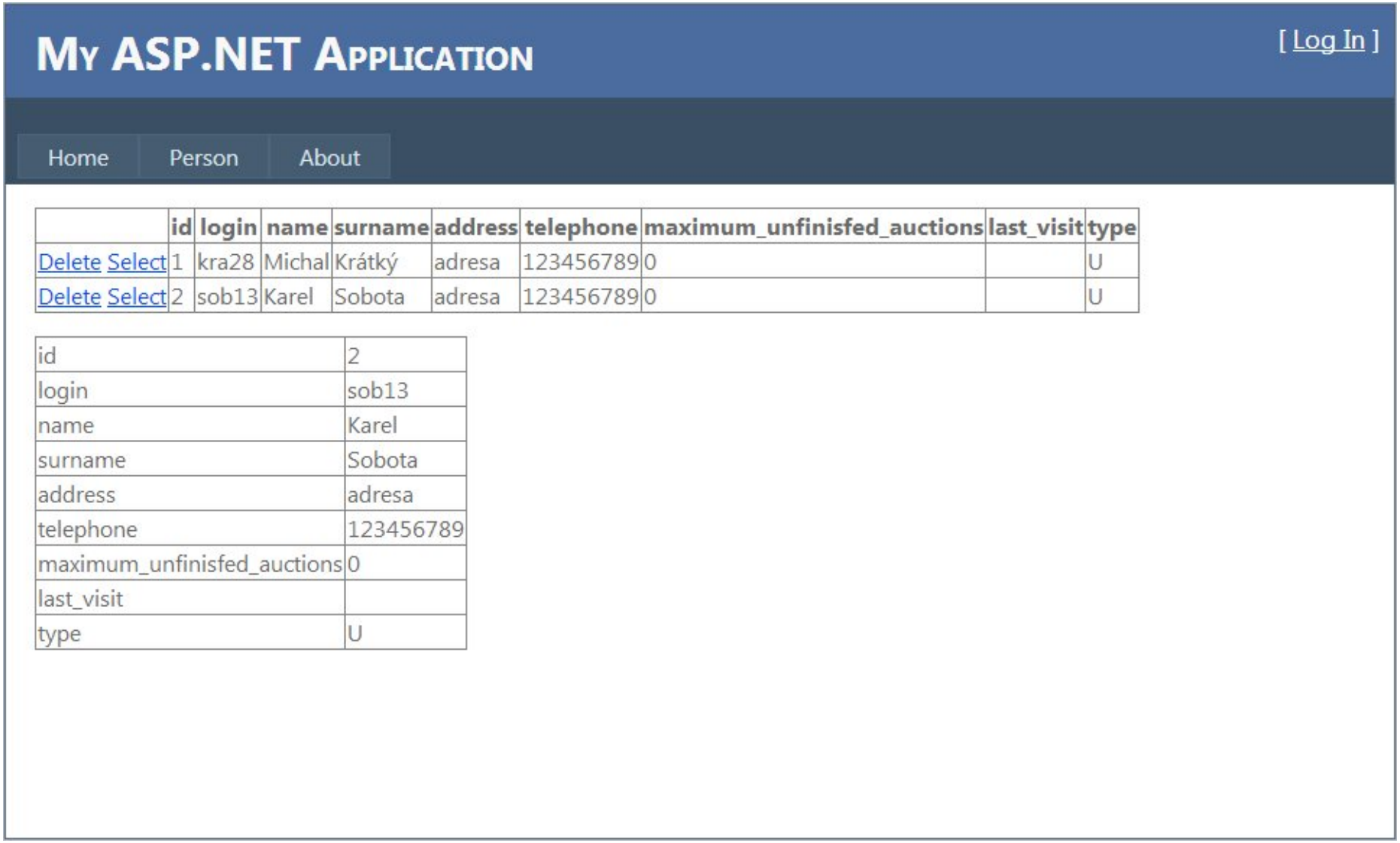
- There are two central points of each ASP.NET application:
  - **Master page**, e.g. Site.Master: forms of an application share common features, e.g. look&feels, menu, ...
  - **Web.config**: it includes a common configuration of the application: connection string, registration of .NET components, code directory, ...

Application Building in Visual Studio

- Menu: File/New Project
- Select: Visual C# a ASP.NET Web Application
- The generated applications includes:
  - Basic form: Deafult.aspx
  - Log In form: Account/Login.aspx
  - About form: About.aspx



Basic Application



	id	login	name	surname	address	telephone	maximum_unfinisfed_auctions	last_visit	type
Delete Select	1	kra28	Michal	Krátký	adresa	123456789	0		U
Delete Select	2	sob13	Karel	Sobota	adresa	123456789	0		U

id	2
login	sob13
name	Karel
surname	Sobota
address	adresa
telephone	123456789
maximum_unfinisfed_auctions	0
last_visit	
type	U

Form Building 1/2

- The right mouse button: **Add/New item** and choose `Web Form using Master Page`.
  - Set the form name and the master page `Site.Master`.
- Add `GridView` from Toolbox:

```
<asp:GridView ID="GridViewPerson" runat="server" ...
```

- Add `SqlDataSource` from Toolbox and edit it[1]:

```
<asp:SqlDataSource ID="sdsPerson" runat="server"
 SelectCommand="SELECT␣*␣FROM␣Person"
 ConnectionString=
 "server=dbsys.cs.vsb.cz\STUDENT;database=dais;user=dais;"
 + "password=tuo460DbEd;">
</asp:SqlDataSource>
```

[1] Write a communication with a DBMS in the presentation layer is inappropriate, we solve it using a data layer and ORM.

Form Building 2/2

- Add the data source in the GridView element:

```
<asp:GridView ID="GridViewPerson" runat="server"
    DataSourceID="sdsPerson"> ...
```

- Add in the MasterPage `Site.Master`:

```
<asp:MenuItem NavigateUrl="~/Form/Person.aspx"
  Text="Person"/>
```

  In the element: `<asp:Menu ...><Items>`

Setting Form to Master Page

- If a form is not created to a master page, we must set the master page in the form:
  - Add: `MasterPageFile=" /Site.master"` into: `<%@ %>`
  - Delete tags: html, body and so on.
  - Add: `<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">`

Connection string Configuration

- Add element <configuration> in Web.config:

```
<connectionStrings>
 <add name="ConnectionString"
          connectionString="server=dbsys.cs.vsb.cz\STUDENT;"
         + "database=dais;user=dais;password=tuo460DbEd;" />
</connectionStrings>
```

- Change the data source in Person.aspx:

```
<asp:SqlDataSource ID="sdsPerson" runat="server"
 SelectCommand="SELECT␣*␣FROM␣Person"
 ConnectionString=
 "<%␣$connectionStrings:ConnectionString␣%>">
</asp:SqlDataSource>
```

Record Detail 1/2

- Add links for the delete and select operations of a record:

```
<asp:GridView ID="GridViewPerson" runat="server"
 DataKeyNames="id" AllowPaging="True"
 DataSourceID="sdsPerson">
 <Columns>
  <asp:CommandField ShowSelectButton="True"
                    ShowDeleteButton="True" />
 </Columns>
</asp:GridView>
```

- Add a record detail, element `DetailView`:

```
<asp:DetailsView ID="DetailsViewPerson" runat="server"
 AutoGenerateRows="true" DataSourceID="sdsPersonDetail"
 DataKeyNames="id">
</asp:DetailsView>
```

Record Detail 2/2

- Add a data source returning the current record of the GridView:

```
<asp:SqlDataSource ID="sdsPersonDetail" runat="server"
 SelectCommand="Select␣*␣FROM␣Person␣WHERE␣id=@id"
 ConnectionString=
          "<%␣$connectionStrings:ConnectionString␣%>">
 <SelectParameters>
  <asp:ControlParameter PropertyName="SelectedValue"
       Type="Int32" Name="id"
       ControlID="GridViewPerson" DefaultValue="1">
         </asp:ControlParameter>
 </SelectParameters>
</asp:SqlDataSource>
```

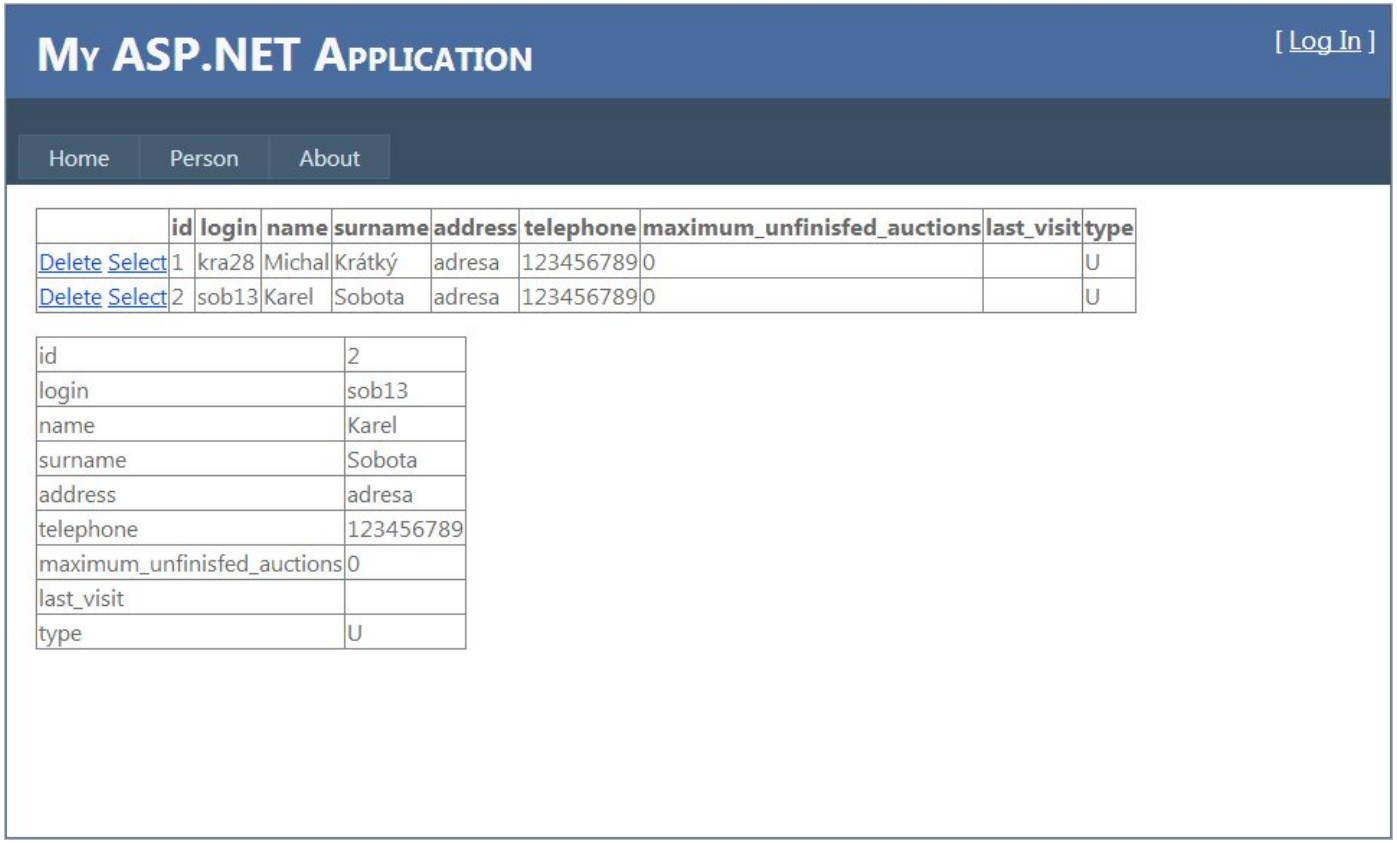
Complete Form/Person.aspx 1/2

```
<%@ Page Language="C#" MasterPageFile="~/Site.Master"
  AutoEventWireup="true" CodeBehind="Person.aspx.cs"
  Inherits="BidWebApp.Form.User" %>

<asp:Content ID="BodyContent" runat="server"
    ContentPlaceHolderID="MainContent">
  <p><asp:GridView ID="GridViewPerson" runat="server"
      DataKeyNames="id"
            AllowPaging="True" DataSourceID="sdsPerson">
      <Columns>
        <asp:CommandField ShowSelectButton="True"
                          ShowDeleteButton="True" />
      </Columns>
  </asp:GridView></p>

  <p><asp:DetailsView ID="DetailsViewPerson" runat="server"
     AutoGenerateRows="true" DataSourceID="sdsPersonDetail"
           DataKeyNames="id">
  </asp:DetailsView></p>
```

Complete Form/Person.aspx 2/2

```
<asp:SqlDataSource ID="sdsPerson" runat="server"
 SelectCommand="SELECT * FROM Person"
 ConnectionString="<% $connectionStrings:ConnectionString %>">
</asp:SqlDataSource>

<asp:SqlDataSource ID="sdsPersonDetail" runat="server"
 SelectCommand="Select * FROM Person WHERE id=@id"
 ConnectionString="<% $connectionStrings:ConnectionString %>">
 <SelectParameters>
  <asp:ControlParameter PropertyName="SelectedValue" Type="Int32"
          Name="id" ControlID="GridViewPerson" DefaultValue="1">
       </asp:ControlParameter></SelectParameters>
</asp:SqlDataSource>
</asp:Content>
```

Basic Application

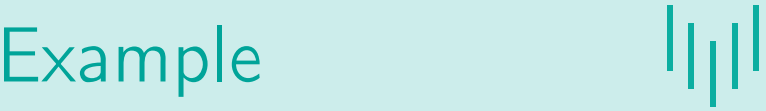


- Why this way of implementation **is not a good idea** (a mix of presentation, application, and data layers)?
  - It is a good practice to separate data presentation and model.

References

- Course: ASP.NET Application Development
- `http://www.asp.net`
- Tutorial – `http://www.asp.net/QuickStart/aspnet/Default.aspx`.
- `http://dotnet.jku.at/courses/dotnet/`

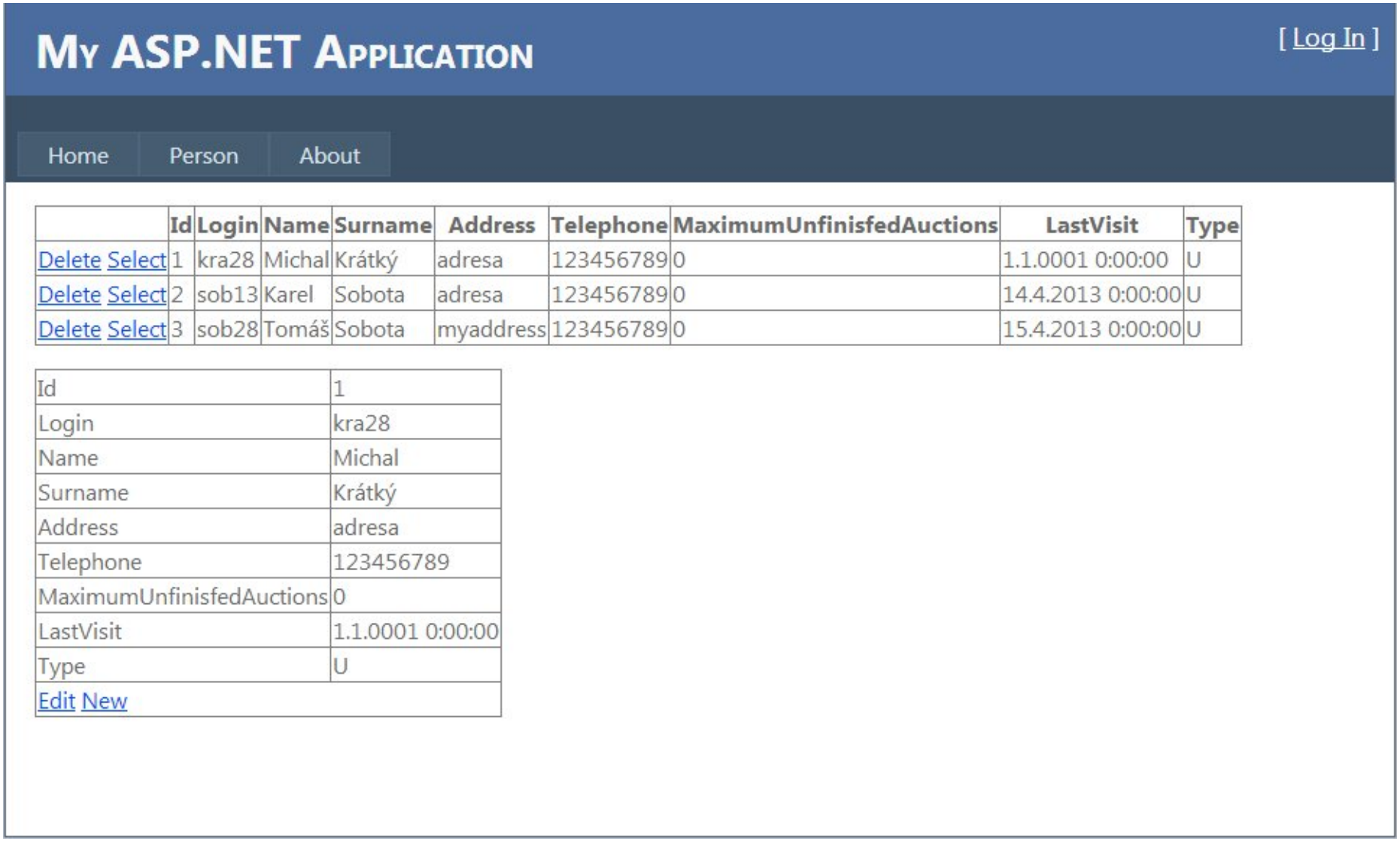
Integration of ORM into ASP.NET, Example



Let us have the following ORM:

- Database tables: User, Category, Auction.
- DTO classes: User, Category, Auction.
- DAO classes: UserTable, CategoryTable, AuctionTable.
- Auxiliary classes: Database

Example, ORM



My ASP.NET Application

[Log In]

Home | Person | About

	Id	Login	Name	Surname	Address	Telephone	MaximumUnfinisfedAuctions	LastVisit	Type
Delete Select	1	kra28	Michal	Krátký	adresa	123456789	0	1.1.0001 0:00:00	U
Delete Select	2	sob13	Karel	Sobota	adresa	123456789	0	14.4.2013 0:00:00	U
Delete Select	3	sob28	Tomáš	Sobota	myaddress	123456789	0	15.4.2013 0:00:00	U

Id	1
Login	kra28
Name	Michal
Surname	Krátký
Address	adresa
Telephone	123456789
MaximumUnfinisfedAuctions	0
LastVisit	1.1.0001 0:00:00
Type	U
Edit New	

Example – ORM, User.aspx 1/2

```
1  <%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="
2      CodeBehind="User.aspx.cs" Inherits="AuctionWebApp.Form.Person" %>
3  <asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
4
5  <p><asp:GridView ID="GridViewUser" runat="server" DataKeyNames="id"
6    AllowPaging="True" DataSourceID="odsUser">
7    <Columns>
8      <asp:CommandField ShowSelectButton="True" ShowDeleteButton="True" />
9    </Columns>
10 </asp:GridView></p>
11
12 <p><asp:DetailsView ID="DetailsViewUser" runat="server"
13   AutoGenerateRows="true" DataSourceID="odsUserDetail" DataKeyNames="id">
14   <Fields>
15     <asp:CommandField ShowEditButton="True" ShowInsertButton="True" />
16   </Fields>
17 </asp:DetailsView></p>
```

Notice: `GridView` and `DetailsView` are related to instances of `ObjectDataSource`.

Example – ORM, User.aspx 2/2

```
1     <asp:ObjectDataSource ID="odsUser" runat="server"
2         TypeName="Auction.Database.UserTable"
3       SelectMethod="Select"  DeleteMethod="Delete">
4       <DeleteParameters>
5         <asp:ControlParameter Type="Int32" Name="id" ControlID="GridViewUser">
6         </asp:ControlParameter>
7       </DeleteParameters>
8     </asp:ObjectDataSource>
9
10    <asp:ObjectDataSource ID="odsUserDetail" runat="server"
11      TypeName="Auction.Database.UserTable"
12      DataObjectTypeName="Auction.Database.User"
13      SelectMethod="Select" InsertMethod="Insert" UpdateMethod="Update">
14      <SelectParameters>
15        <asp:ControlParameter PropertyName="SelectedValue" Type="Int32" Name="id"
16        ControlID="GridViewPerson" DefaultValue="1"></asp:ControlParameter>
17      </SelectParameters>
18    </asp:ObjectDataSource>
19  </asp:Content>
```

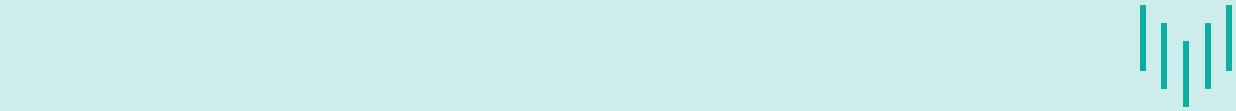
Notice: We must specify DTO `User`, DAO `UserTable` and its methods for the operations: Select, Update, Insert and Delete.

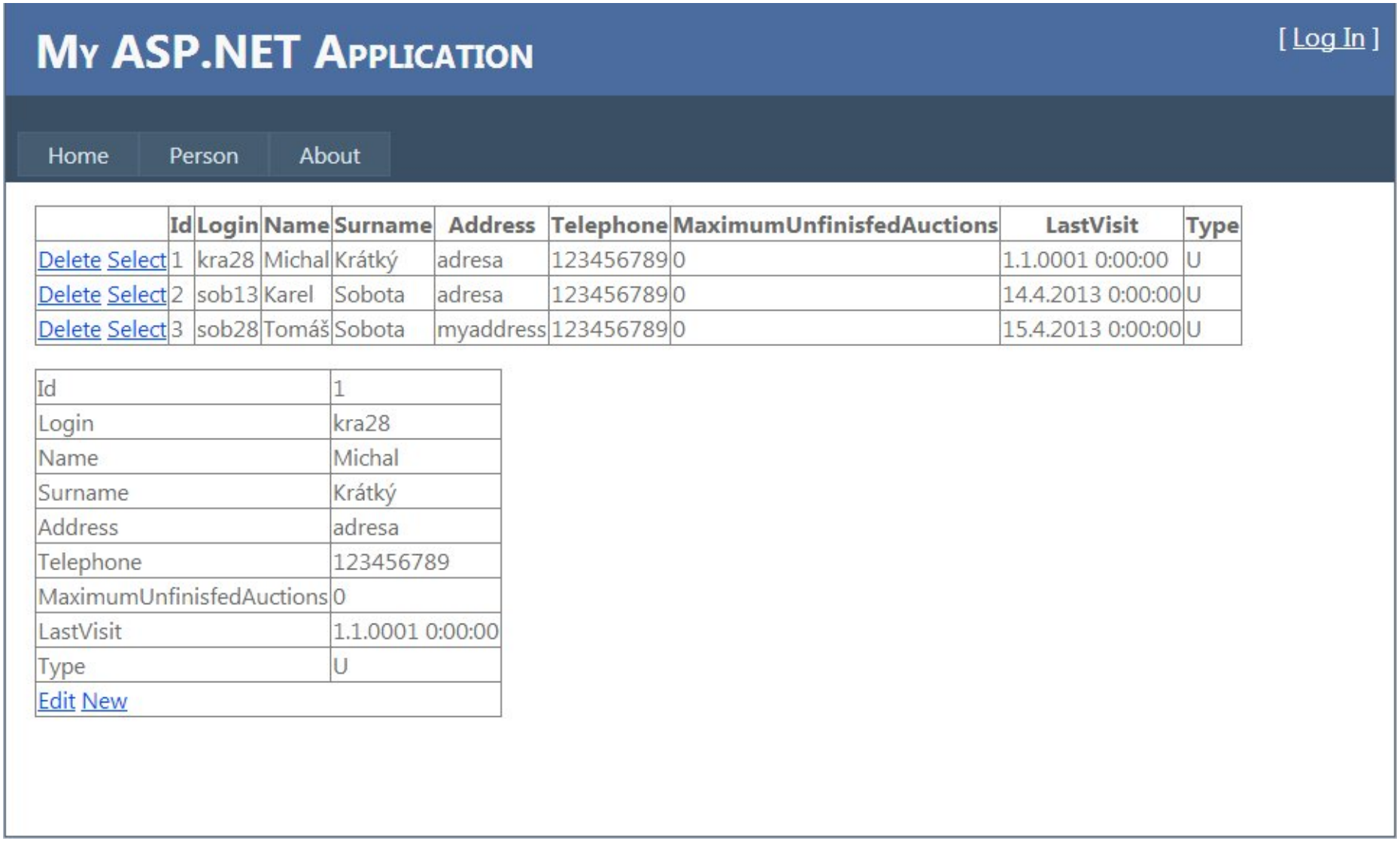
Example – ORM, Database.cs

```
1   ...
2   public class Database {
3     private SqlConnection mConnection;
4     SqlTransaction mSqlTransaction = null;
5     private String mLanguage = "en";
6
7     public Database() {
8       mConnection = new SqlConnection();
9     }
10
11    public bool Connect() {
12      bool ret = true;
13      if (mConnection.State != System.Data.ConnectionState.Open)  {
14        ret = Connect(WebConfigurationManager.ConnectionStrings[
15          "ConnectionString"].ConnectionString);
16      }
17      return ret;
18    }
```

Lines 14-15: The connection string is defined in the file Web.config

Example, ORM





MY ASP.NET APPLICATION

[Log In]

Home Person About

	Id	Login	Name	Surname	Address	Telephone	MaximumUnfinisfedAuctions	LastVisit	Type
Delete Select	1	kra28	Michal	Krátký	adresa	123456789	0	1.1.0001 0:00:00	U
Delete Select	2	sob13	Karel	Sobota	adresa	123456789	0	14.4.2013 0:00:00	U
Delete Select	3	sob28	Tomáš	Sobota	myaddress	123456789	0	15.4.2013 0:00:00	U

Id	1
Login	kra28
Name	Michal
Surname	Krátký
Address	adresa
Telephone	123456789
MaximumUnfinisfedAuctions	0
LastVisit	1.1.0001 0:00:00
Type	U
Edit New	