

Web Database Programming

Created: 2011-01-21

Last update: 2015-12-20

Contents

Introduction	2
Use EasyDataSet as Data Source.....	2
Bind-data to single field	2
Data Query	2
Data Search	2
Data Navigation	2
Modify and Save Data	2
MySQL Database Credential.....	2
Bind Data to Html Table.....	2
Web Page Security	3
User Accounts in database.....	3
Create Logon Page	4
Use the logon page to protect other pages.....	11
Page for changing password	16
User Levels	25
Use user levels	25
Use web page user level	26
Access information only related to logged on user	27
Field Editors	42
Data Repeater	42
Change Query Filters at Runtime	42
Create New Records.....	42
Data Streaming	42
Fetch Data of One-to-Many Relation.....	42
Change Data-binding.....	42

Feedbacks	42
-----------------	----

Introduction

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Use EasyDataSet as Data Source

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Bind-data to single field

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Data Query

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Data Search

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Data Navigation

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Modify and Save Data

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

MySql Database Credential

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Bind Data to Html Table

See <http://www.limnor.com/support/webDatabaseProgramming1.pdf>

Web Page Security

If you publish your web files to a web server then the people from the whole world can modify your database. In some cases you want restrict certain pages to be accessible to certain levels of users.

This chapter describes how you can implement such web page security.

User Accounts in database

Assume user account information is saved in database tables. Assume a user account at least contains log in alias and password fields.

- Log in alias field – This is not the user real name like first name or last name. It is a unique user alias. It cannot have spaces in it. The user uses this alias to log in.
- Password field – This field saves a hash of the user password. For MySQL database it must be at least 41 characters long. Since the real password is not saved in the database, if the user forgets his/her password then the only way to recover the account is to let an Admin to erase the password and the user may create a new password.
- User level field – This is an optional field. If you want to implement multiple level permissions then you may use this field to indicate the user level. Level 0 has all the permissions. You may define levels larger than 0 to restrict permissions.

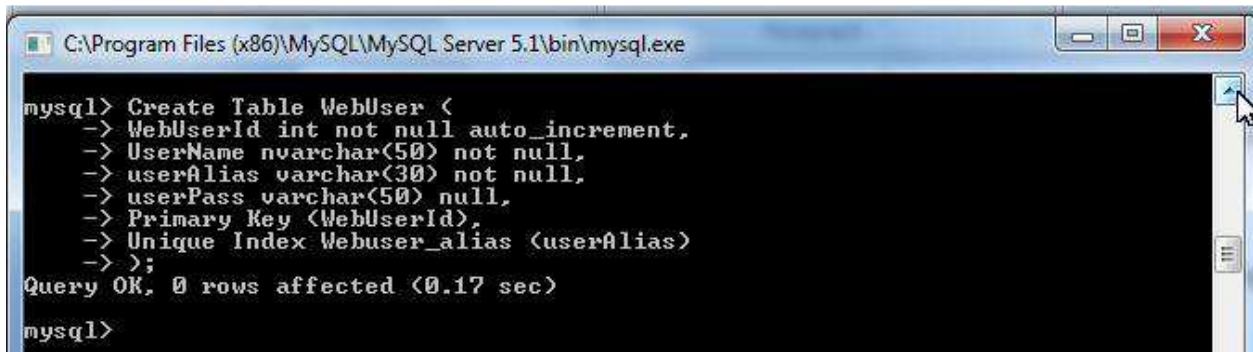
These fields can be in any table. But a unique index should be added on the user alias field.

Later we will add such fields to the student table.

For now, we create a WebUser table to include these fields. Following is a sample SQL script for creating the table.

```
Create Table WebUser (  
WebUserId int not null auto_increment,  
UserName nvarchar(50) not null,  
userAlias varchar(30) not null,  
userPass varchar(50) null,  
Primary Key (WebUserId),  
Unique Index Webuser_alias (userAlias)  
);
```

We create the table in our test MySQL database.



```
C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe

mysql> Create Table WebUser (
-> WebUserId int not null auto_increment,
-> UserName nvarchar(50) not null,
-> userAlias varchar(30) not null,
-> userPass varchar(50) null,
-> Primary Key (WebUserId),
-> Unique Index Webuser_alias (userAlias)
-> );
Query OK, 0 rows affected (0.17 sec)

mysql>
```

Note: do not enter password directly into the table. For example, DO NOT use the following command:

Update WebUser Set 'userPass'='test' WHERE WebUserID=1

Use the following command instead:

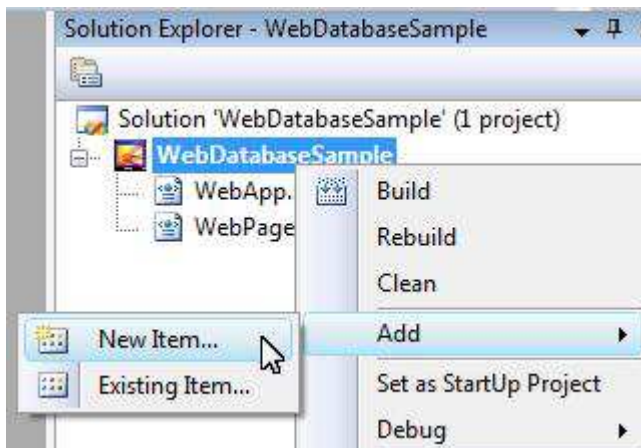
Update WebUser Set 'userPass'=PASSWORD('test') WHERE WebUserID=1

Using MySQL PASSWORD function, only a hash of the password is saved. The real password is not saved in the database. Thus, no one can open the database and view all passwords for all accounts.

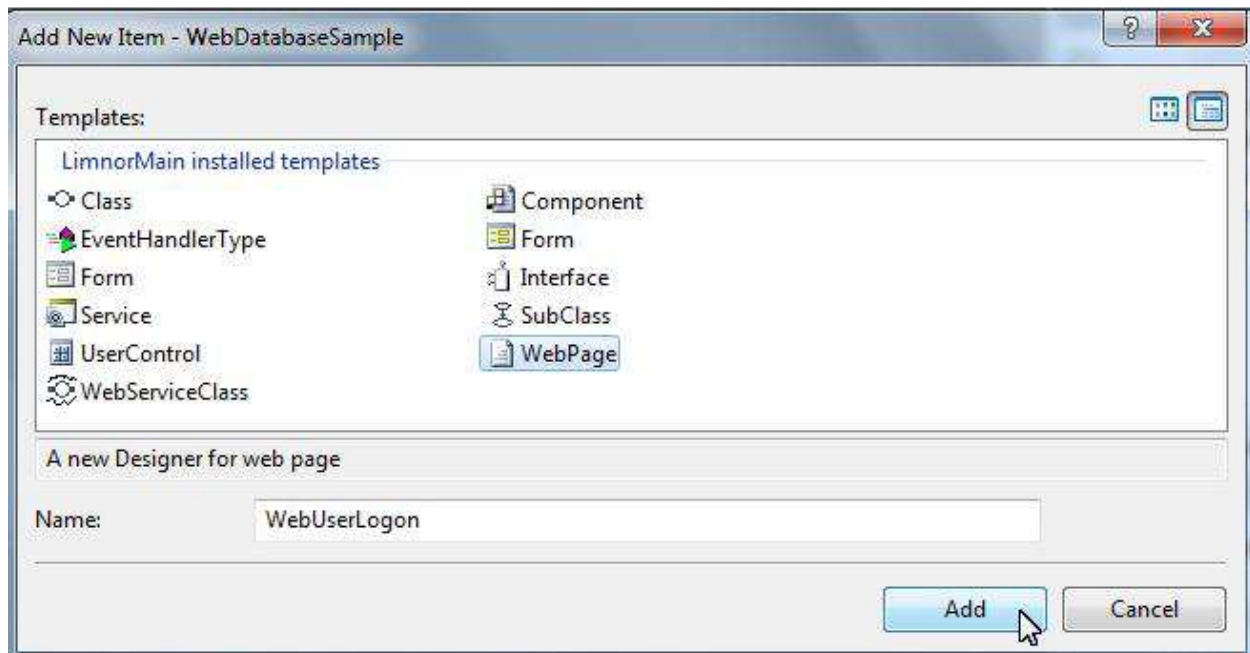
Create Logon Page

As the first step of web page security we need to create a logon page so that our web application knows whether the user should be restricted.

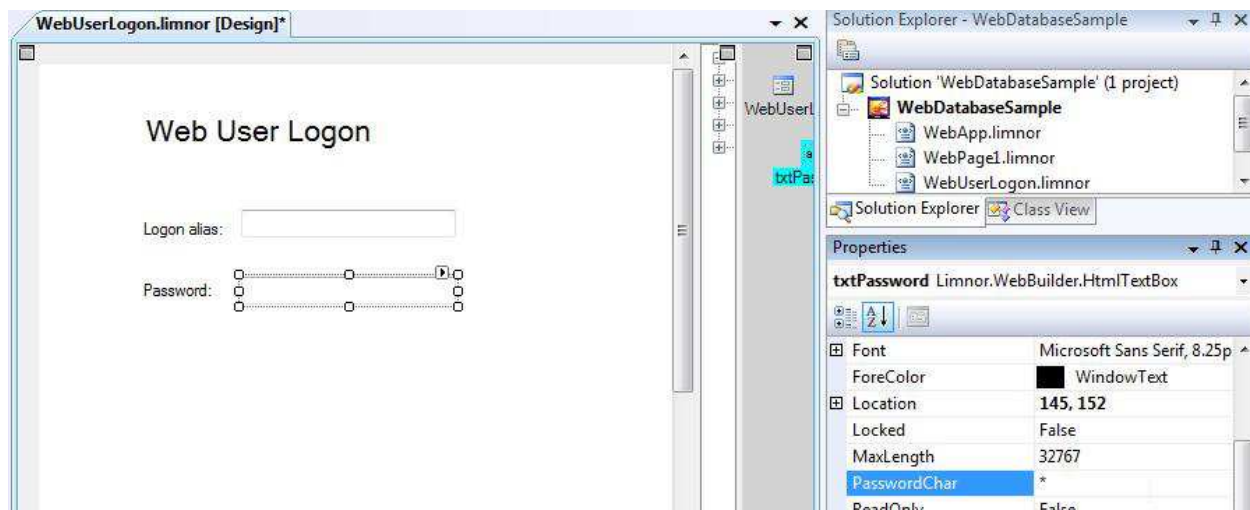
Right-click the web project; choose “Add” and “New Item...”



Select WebPage; name it WebUserLogon; click Add:



Add two text boxes to the web page for log in alias and password. For the password text set its PasswordChar to *

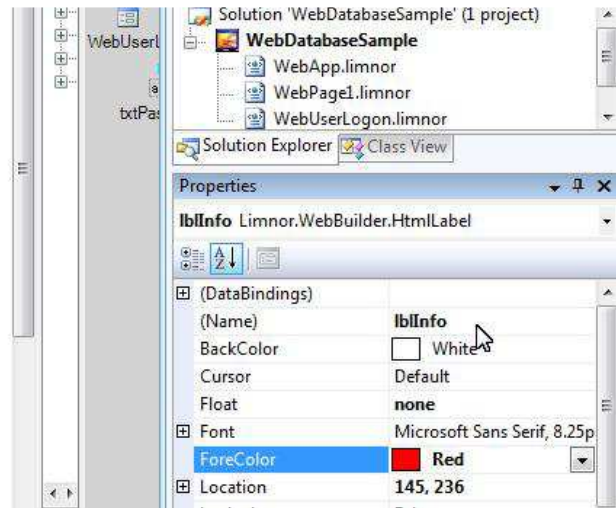


Add a button for doing the log in. Add a label for showing log in failure information. Name the label lblInfo and set its ForeColor to red:

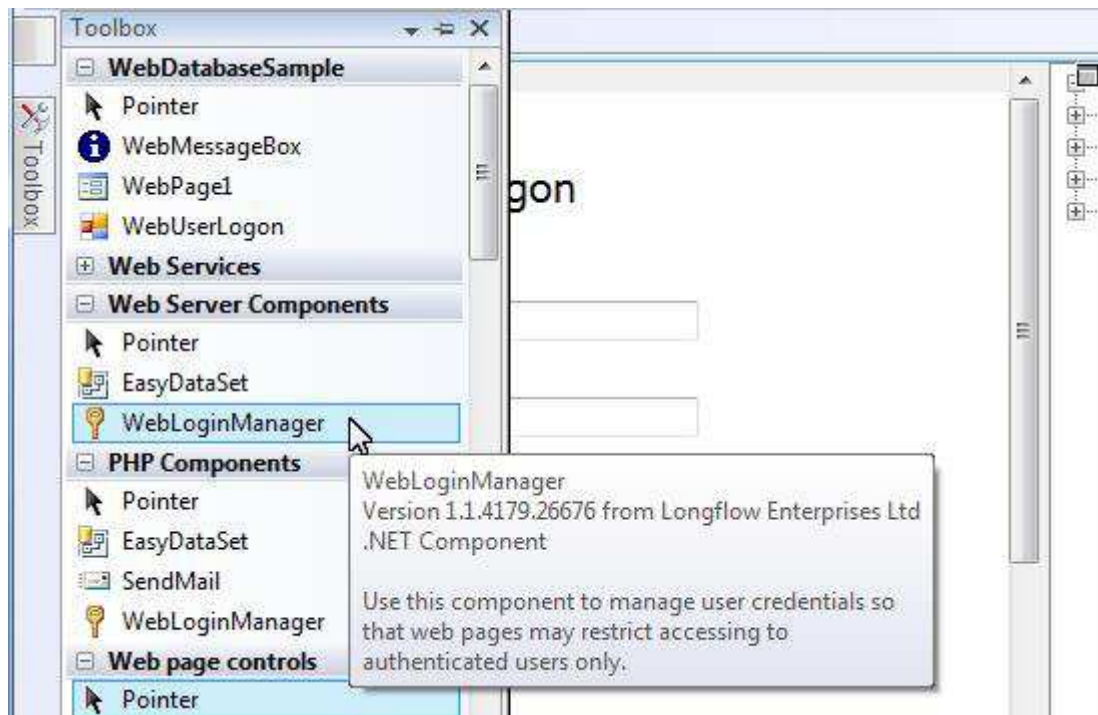
Web User Logon

Logon alias:

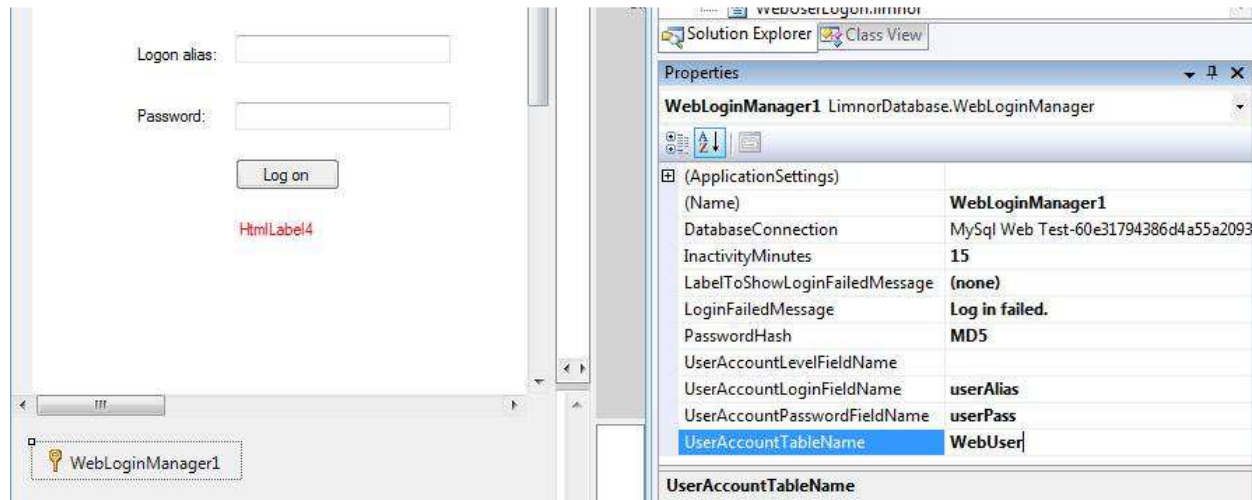
Password:



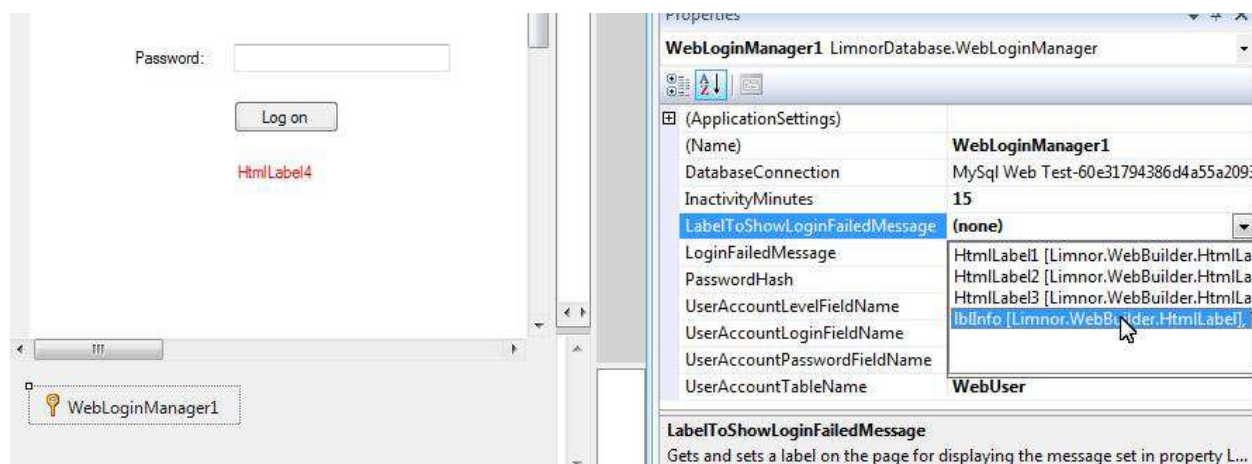
Add a WebLoginManager to the page:



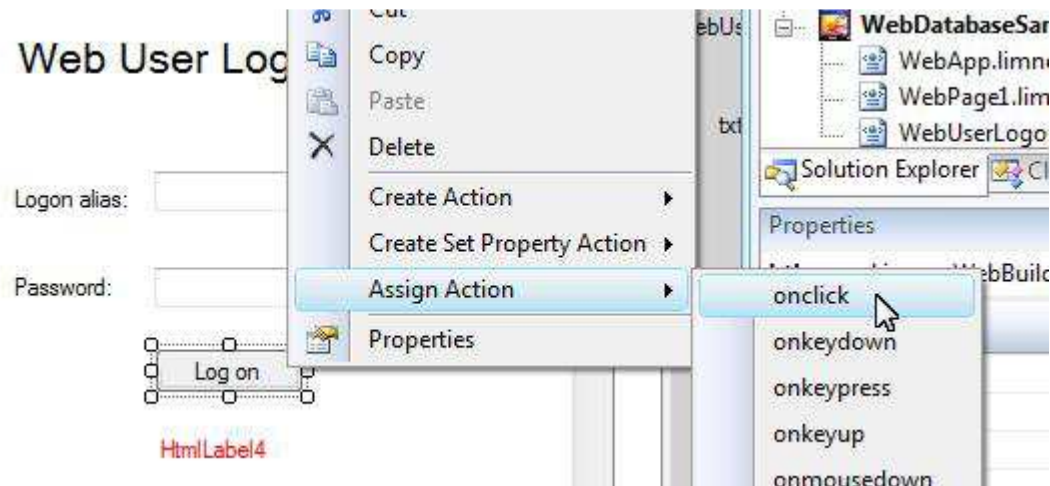
Set its properties:



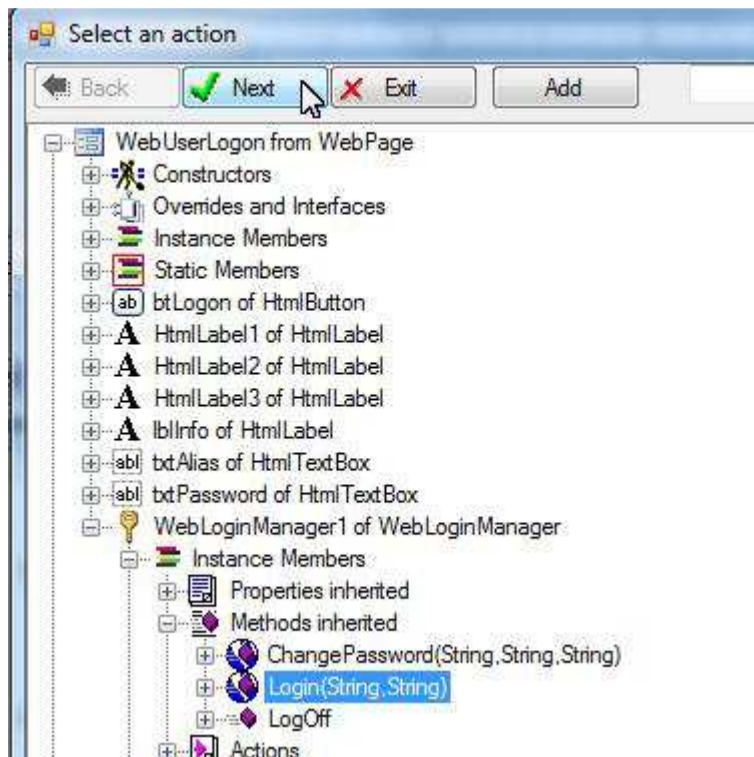
- DatabaseConnection – This is the MySQL database connection we used previously.
- InactivityMinutes – This is an integer representing in how many minutes the log in will expire if there are not mouse and keyboard activities.
- LoginFailedMessage – Give a message to be displayed when log in failed.
- PasswordHash – This property is only used for .Net web applications. For PHP web applications currently only MySQL database is supported and MySQL has built-in password hash.
- UserAccountLevelFieldName – this is the field name for user level. At this time we are not using it. So, leave it blank. We will use it later.
- UserAccountLoginFieldName – For this sample, the field name is userAlias
- UserAccountPasswordFieldName – for this sample, the field name is userPass
- UserAccountTableName – for this sample, the table name is WebUser
- LabelToShowLoginFailedMessage – this is a label on the web page for showing the failure message. We choose lblInfo:



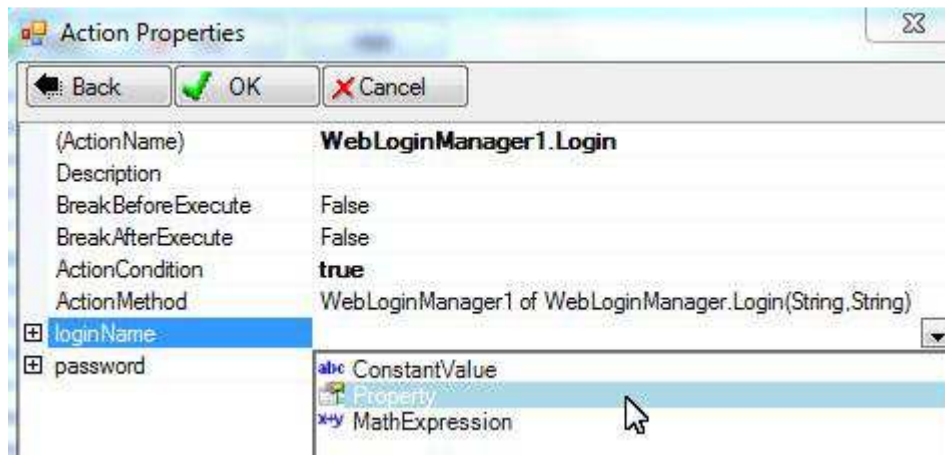
Right-click the “Log on” button; choose “Assign Action”; choose “onclick” event:



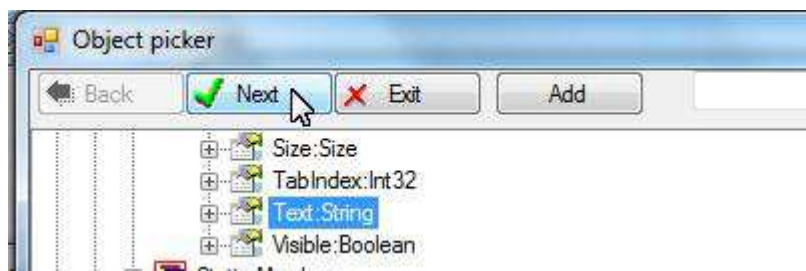
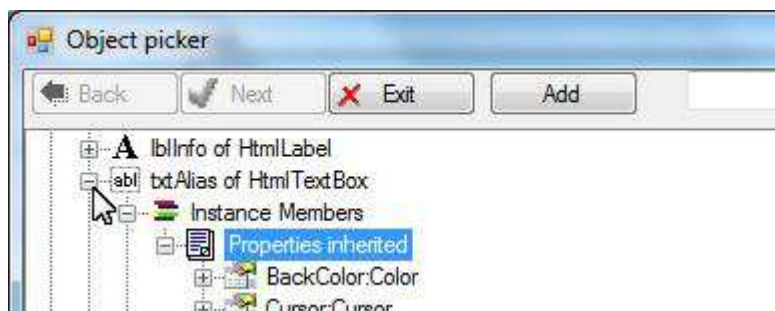
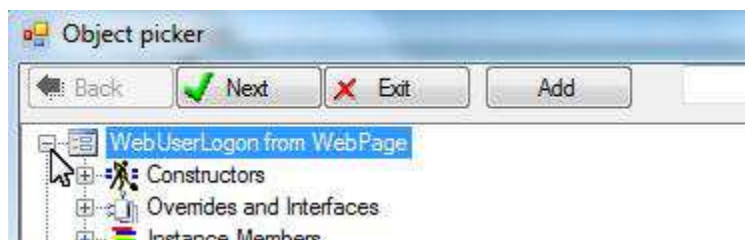
Select the Login method and click Next



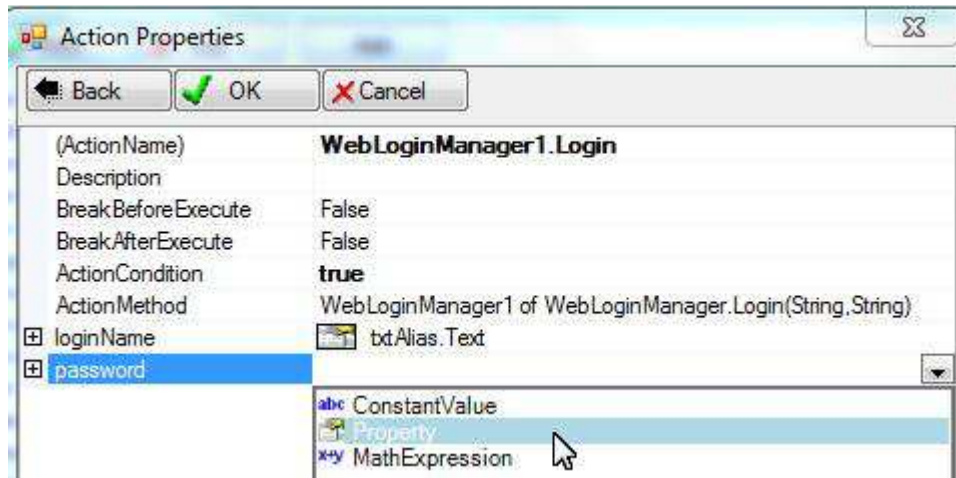
For "loginName"; choose "Property"



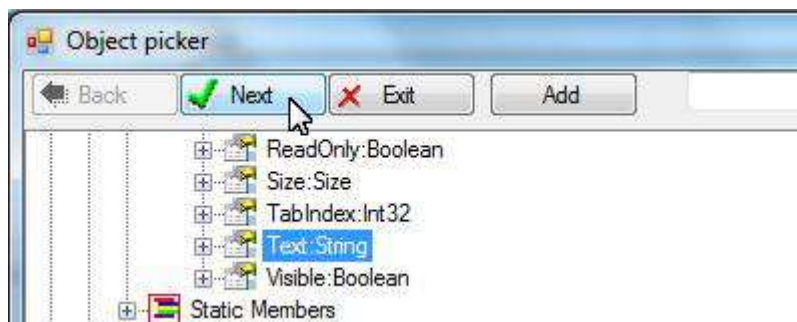
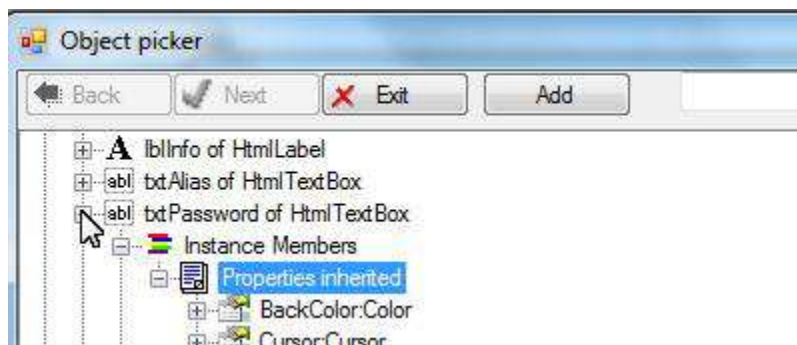
Select the Text property of the text box for the user alias:



For "password"; select "Property":



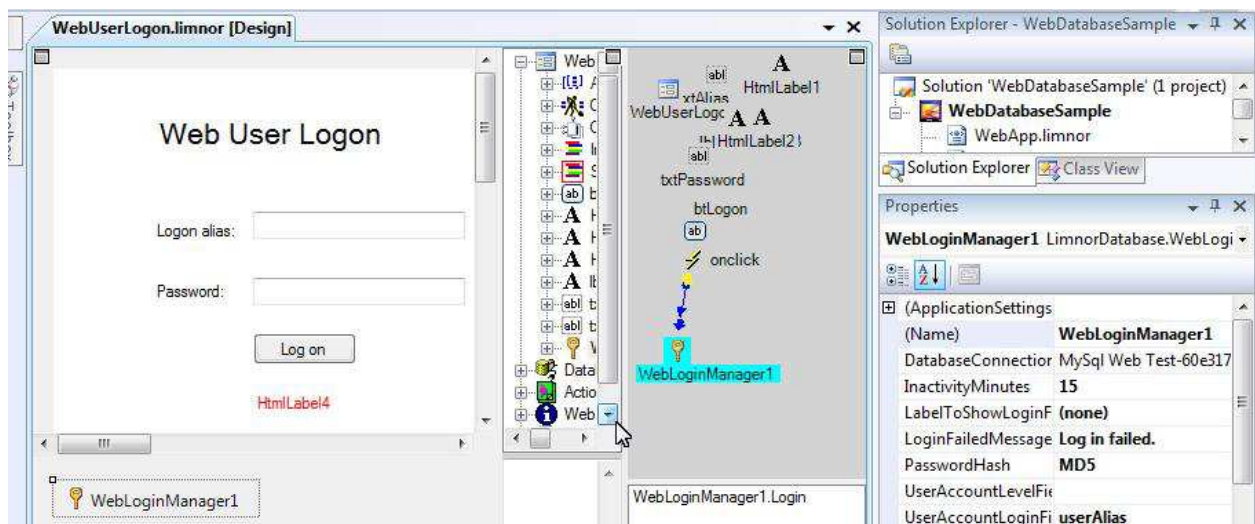
Select the Text property of the text box for the password:



Click OK to finish creating this login action and assigning it to the button:



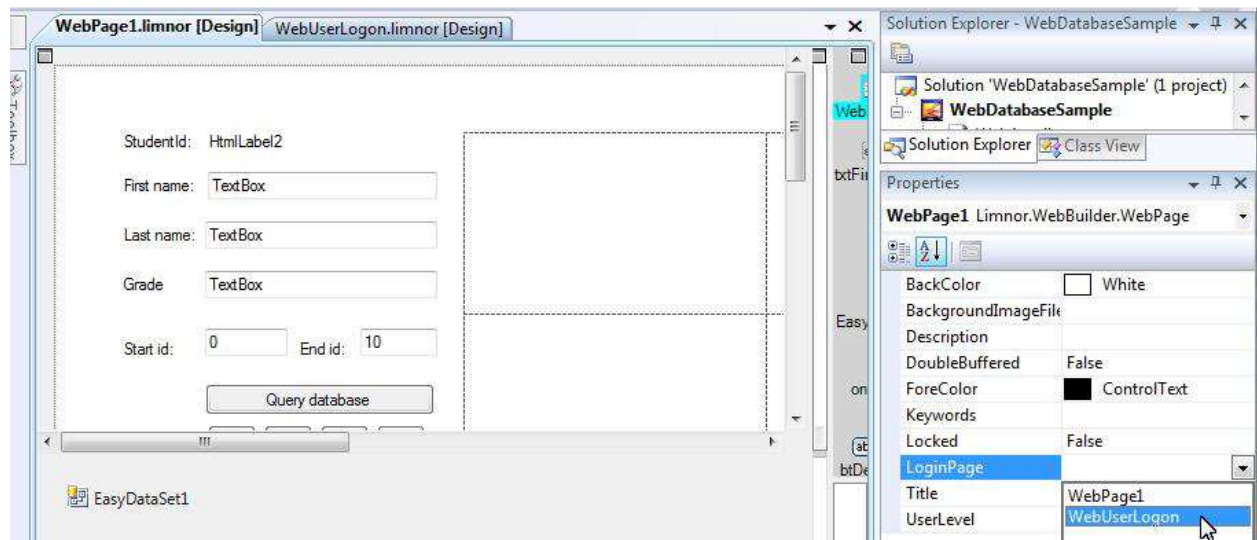
We are done creating this logon page:



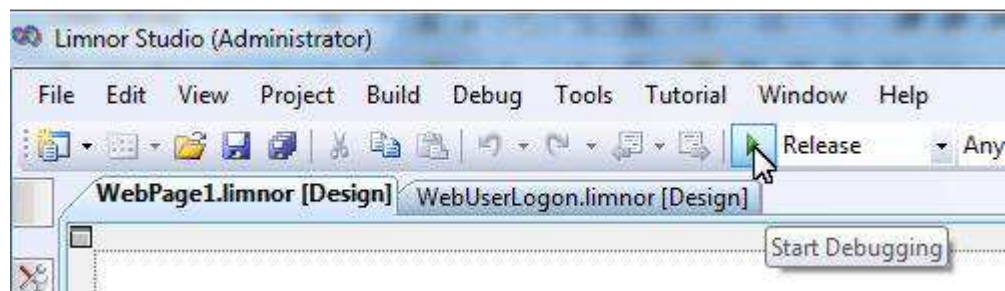
Use the logon page to protect other pages

A web logon page can be used by other web pages for protecting those web pages. One logon page can be used to protect many web pages as long as those web pages have the same user restriction requirements.

Suppose we want to use it to protect the web page for entering student data. Set the LogonPage property to the logon page we just created:



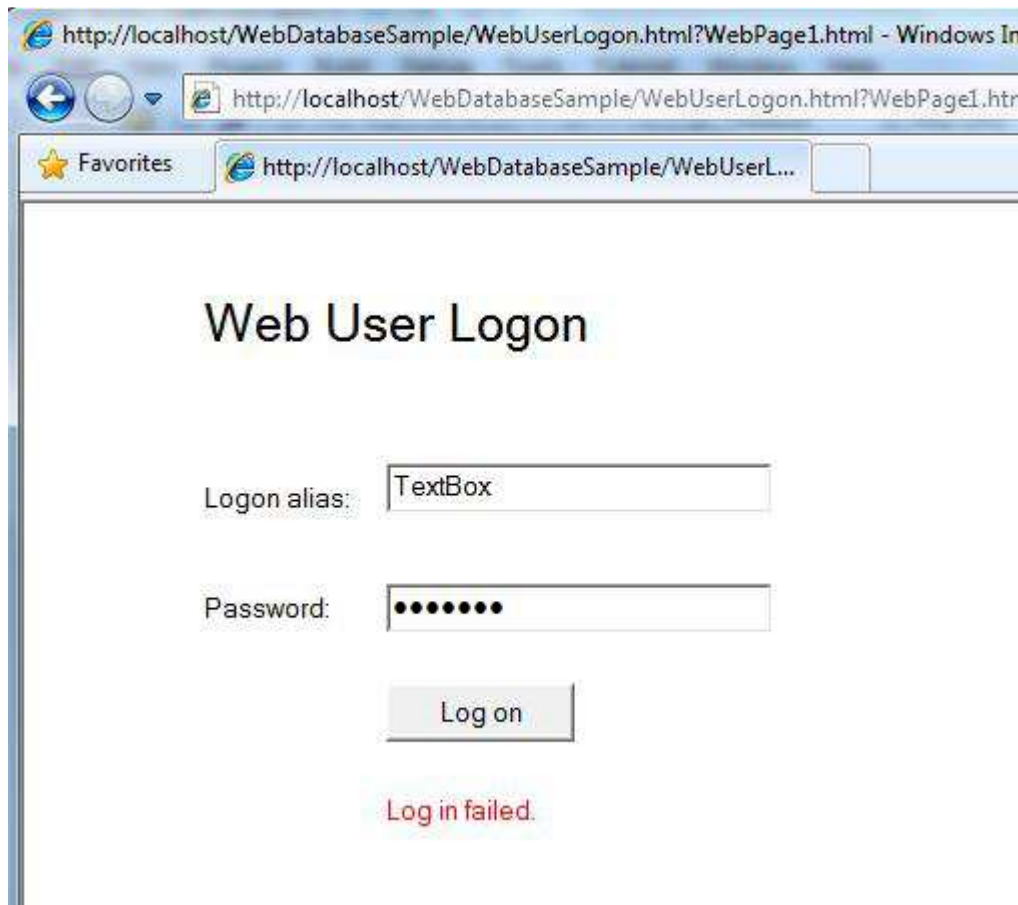
Test the web app:



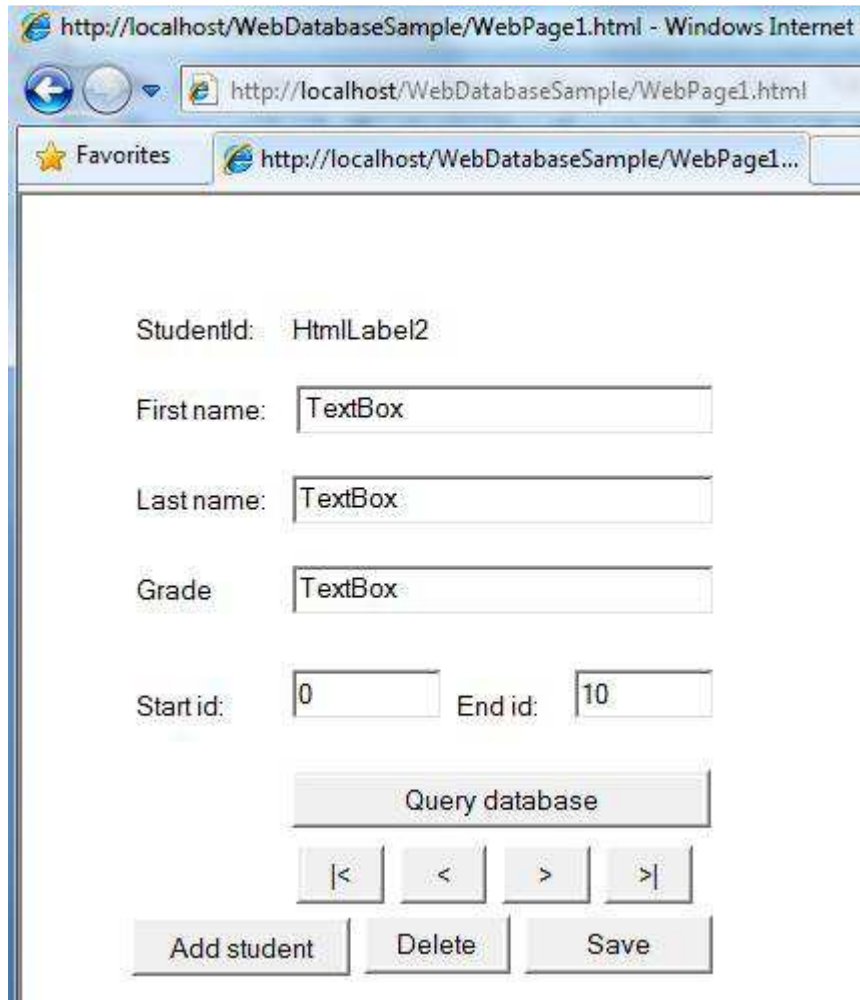
Instead of showing the web page for entering data, the logon page appears:



Click Log on, we will see login failure message:



Enter correct logon alias and password. Click “Log on” again. The web page for data entry appears:



The screenshot shows a web browser window with the address bar displaying `http://localhost/WebDatabaseSample/WebPage1.html`. The browser's Favorites bar is visible below the address bar. The main content area of the browser displays a web form with the following elements:

- A label `StudentId: HtmlLabel2`.
- A text input field labeled `First name:`.
- A text input field labeled `Last name:`.
- A text input field labeled `Grade`.
- Two text input fields for pagination: `Start id:` with the value `0` and `End id:` with the value `10`.
- A large button labeled `Query database`.
- Four small navigation buttons: `|<`, `<`, `>`, and `>|`.
- Three buttons at the bottom: `Add student`, `Delete`, and `Save`.

Click “Query database”. The data appears. We see that after logon the page works normally.

StudentId: 1

First name: Mike

Last name: Jonse

Grade: 5

Start id: 0 End id: 10

Query database

|< < > >|

Add student Delete Save

StudentId	Firstname	Lastname	BirthDate	Grade
1	Mike	Jonse	{null}	5
2	David	Jackson	{null}	6

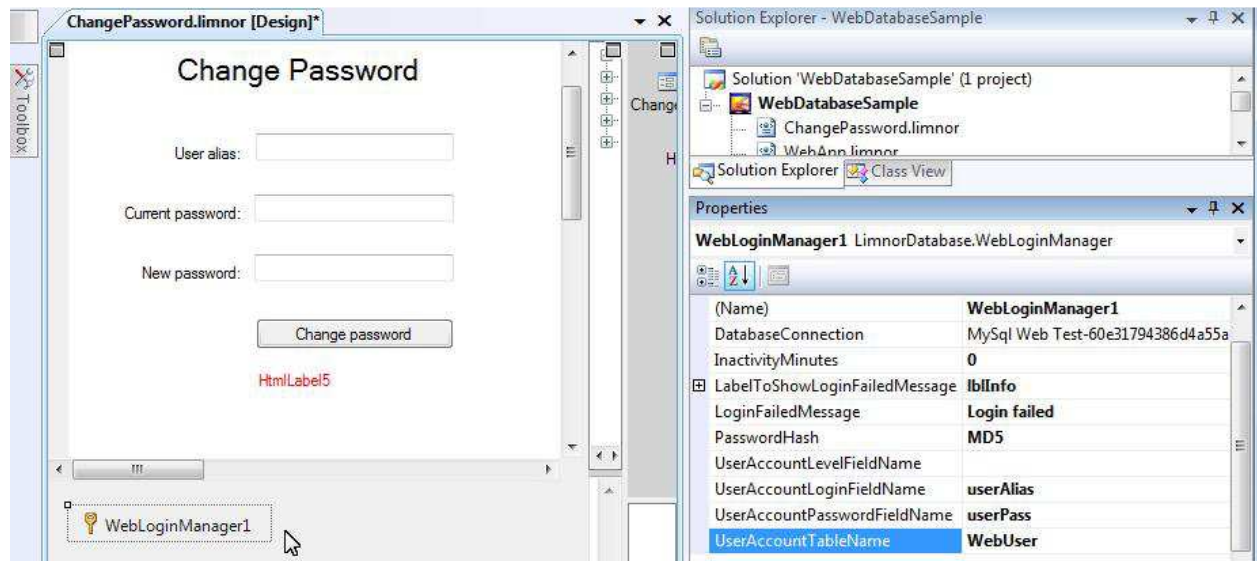
Page for changing password

The component WebLoginManager may also be used to change log in password.

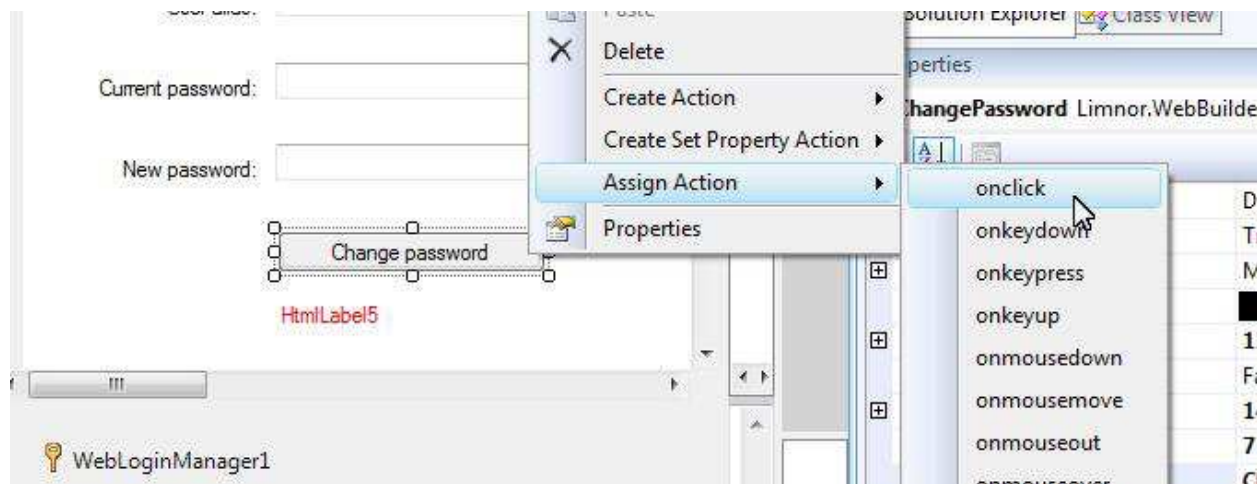
Add a new web page. Name the new page to "ChangePassword".

Add 3 text boxes for entering user alias, current password and new password, and a button for executing the change password action.

Also add a WebLoginManager to the new web page. Set all the properties:

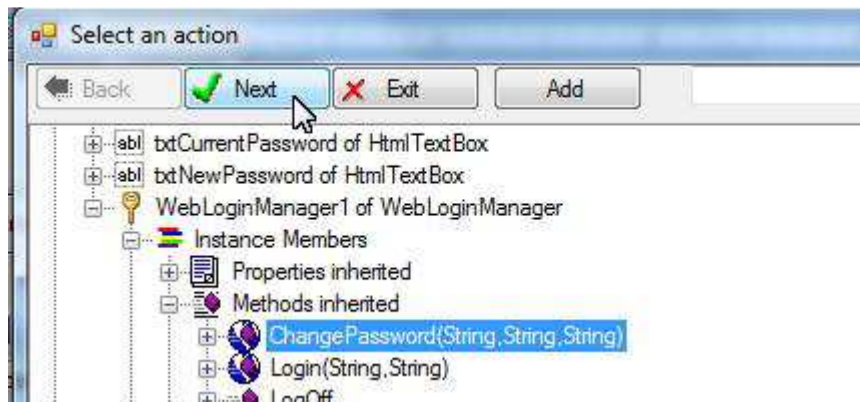


To make the button to trig a change password action, right-click the button, choose “Assign Action”; choose “onclick” event:

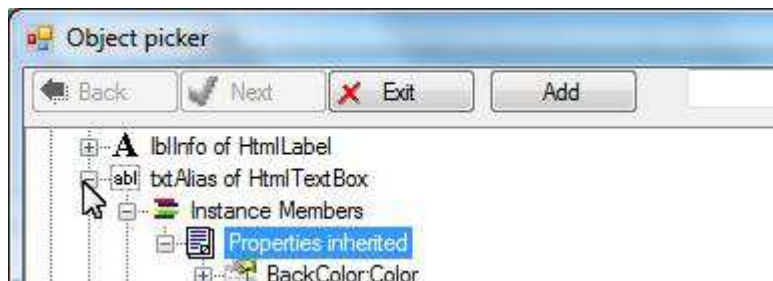
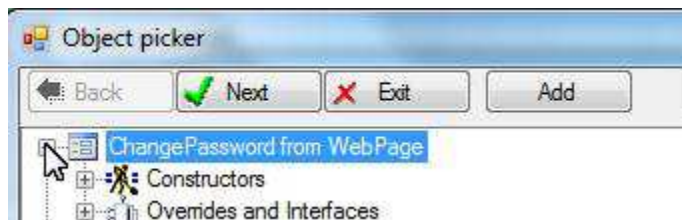
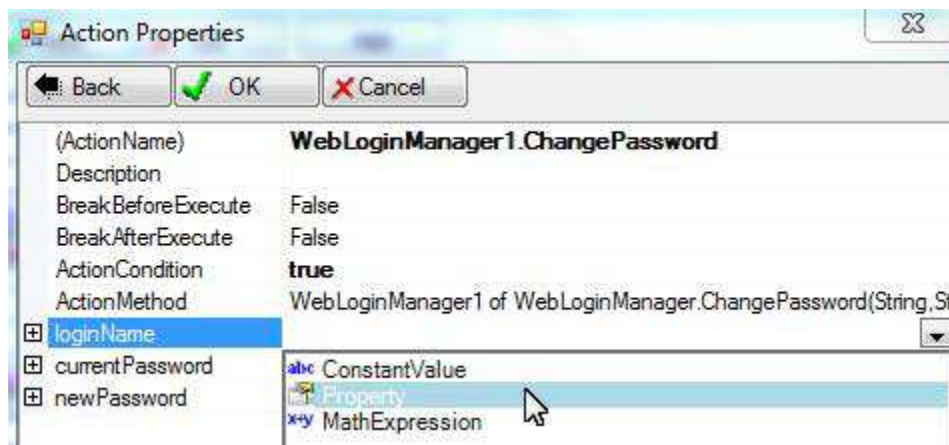


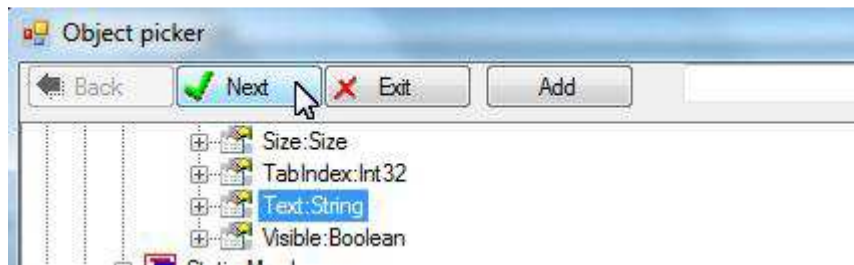
Select the ChangePassword method and click Next:



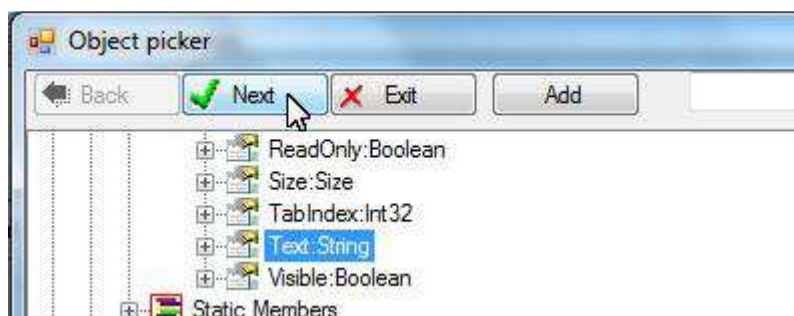
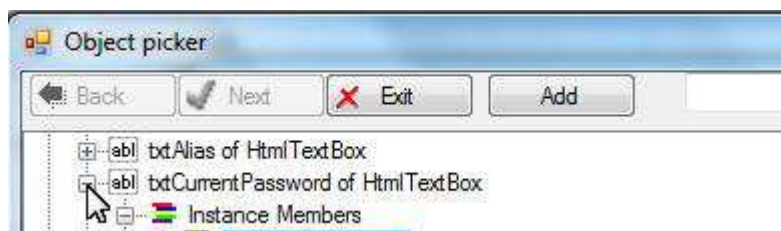
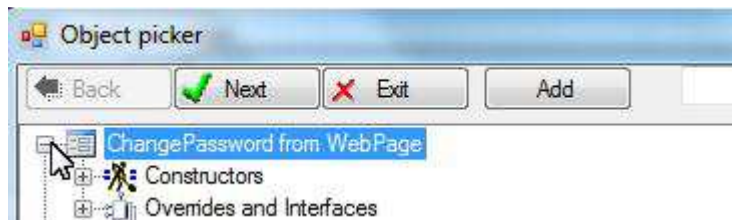
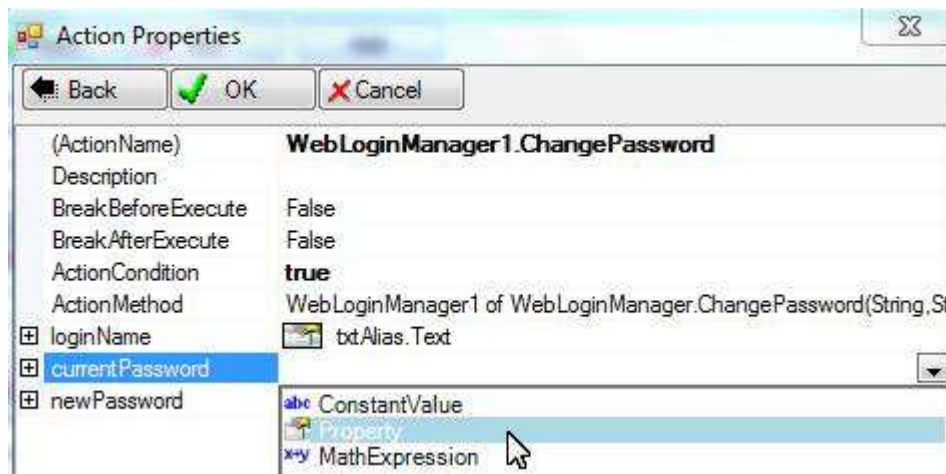


Set the loginName to the Text property of the text box for user alias:

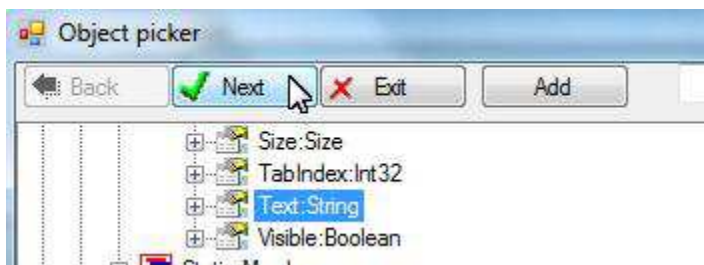
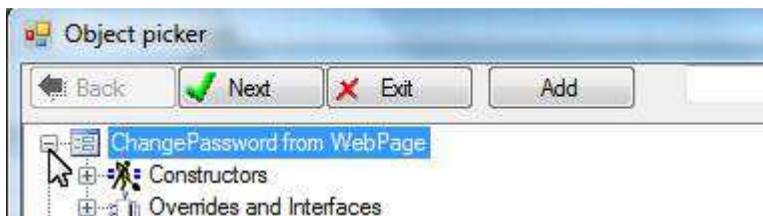
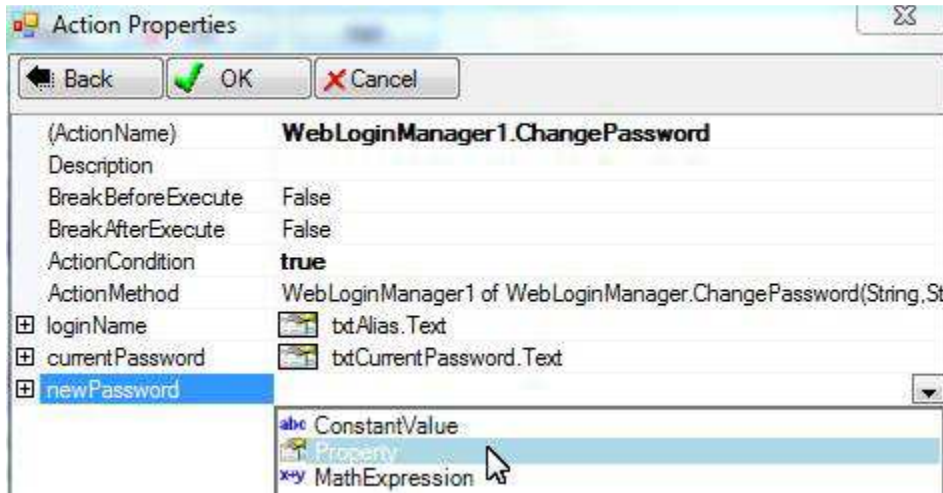




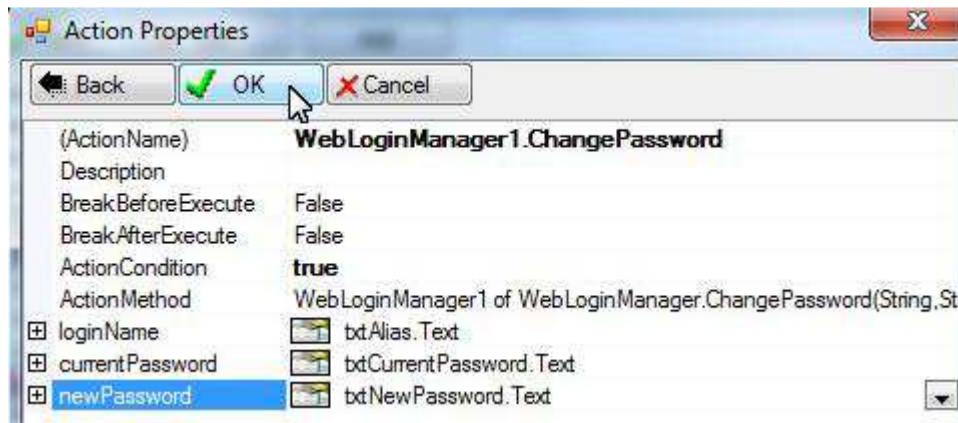
Set "currentPassword" to the Text property of the text box for current password:



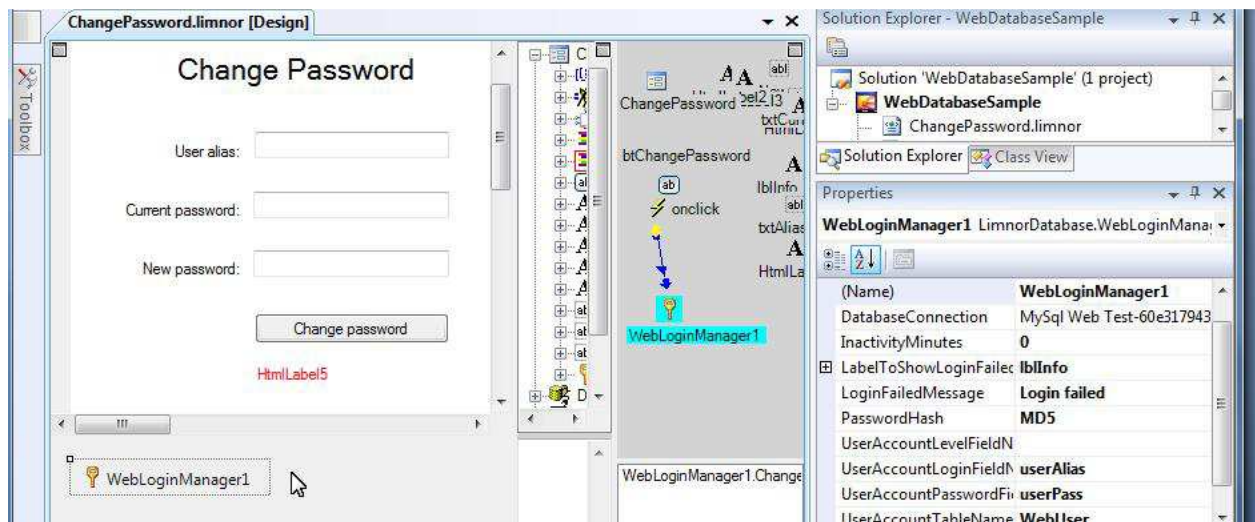
Set the “newPassword” to the Text property of the text box for new password:



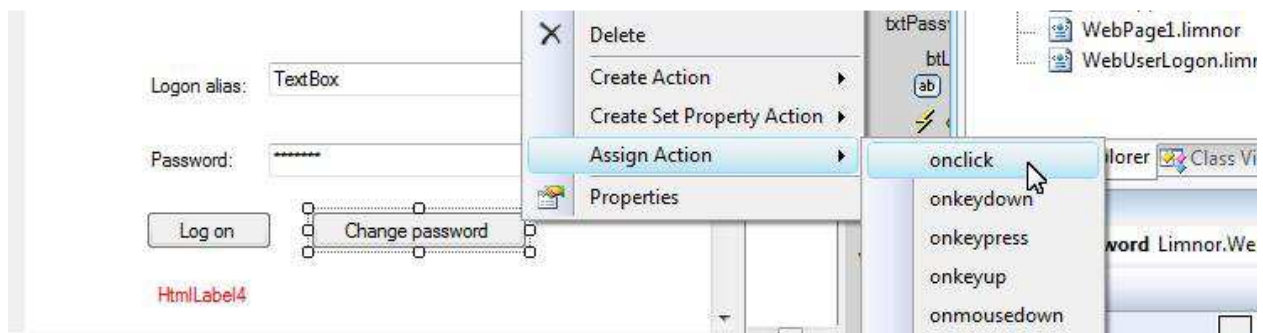
Click OK to finish creating this action and assigning it to the button:



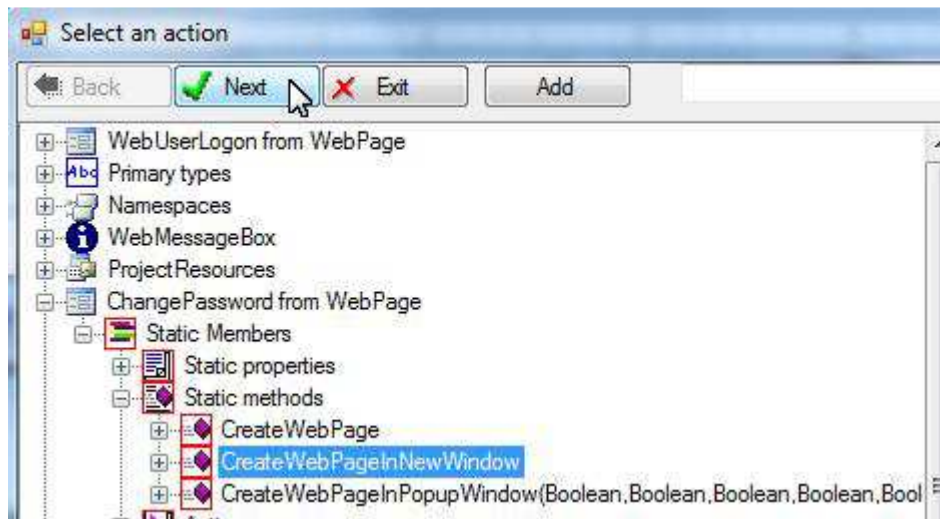
This is our Change Password web page:



Let's add a button on the Logon page to launch Change Password page:



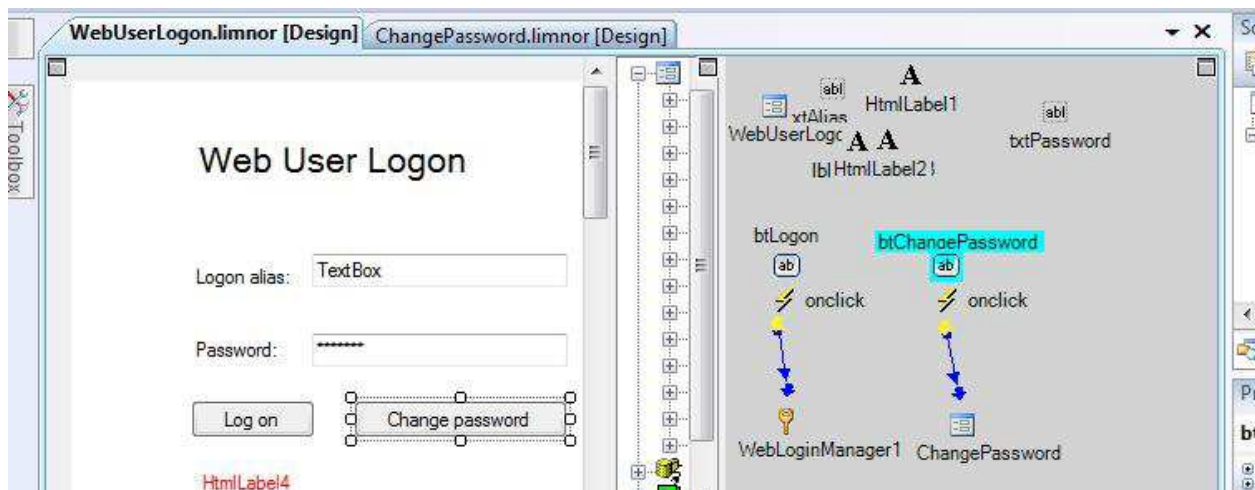
Change "CreateWebPageInNewWindow" under the ChangePassword class and click Next



Click OK to finish creating this action and assigning it to the button:



This is the new Logon page:



Run this web application. The logon page appears:

http://localhost/WebDatabaseSample/WebUserLogon.html?WebPage1.html - Windo

http://localhost/WebDatabaseSample/WebUserLogon.html?1

★ Favorites http://localhost/WebDatabaseSample/WebUserL...

Web User Logon

Logon alias:

Password:

HtmlLabel4

Click "Change password". The Change Password page appears. Enter the alias, current password and new password. If the current password is empty then keep the current password text box empty:



The screenshot shows a Windows Internet Explorer browser window. The address bar displays the URL `http://localhost/WebDatabaseSample/ChangePassword.html`. The browser's Favorites bar is visible below the address bar. The main content area of the browser displays a web page titled "Change Password". The page contains three input fields: "User alias:" with the value "a1", "Current password:" which is empty, and "New password:" which contains three dots. Below these fields is a button labeled "Change password". A mouse cursor is hovering over the button. At the bottom of the page, the text "HtmlLabel5" is visible in red.

Click "Change password". The message says "Password changed":

http://localhost/WebDatabaseSample/ChangePassword.html - Windows Internet E

http://localhost/WebDatabaseSample/ChangePassword.htm

Favorites http://localhost/WebDatabaseSample/ChangePa...

Change Password

User alias:

Current password:

New password:

Password changed

User Levels

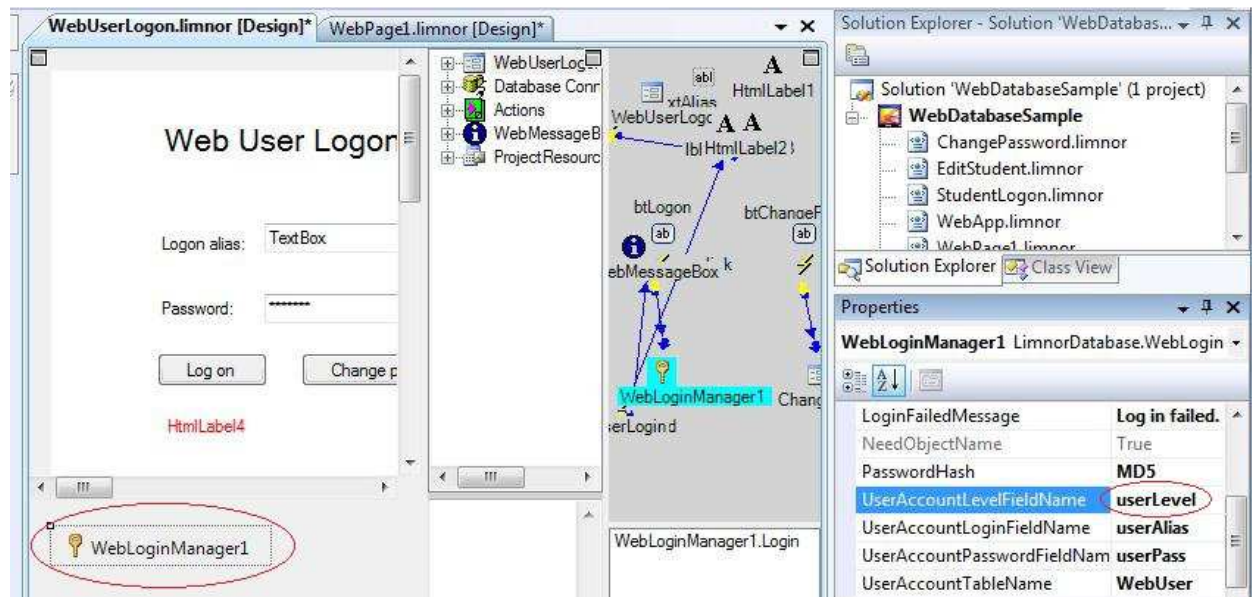
Each user may be given an access level. Each web page may be set a security level to restrict that it may only be accessed by the users with the desired access level.

Use user levels

In your user database, a user record should have a user level field. For example, a user table may be defined as following:

```
Create Table WebUser (  
  WebUserId int not null auto_increment,  
  UserName nvarchar(50) not null,  
  userAlias varchar(30) not null,  
  userLevel int null,  
  userPass varchar(50) null,  
  Primary Key (WebUserId),  
  Unique Index Webuser_alias (userAlias)  
);
```

Specify the field name for the user level for the log in manager in the log in page:



A user with level 0 may access any web pages. A user has limited web page access if user level is larger than 0.

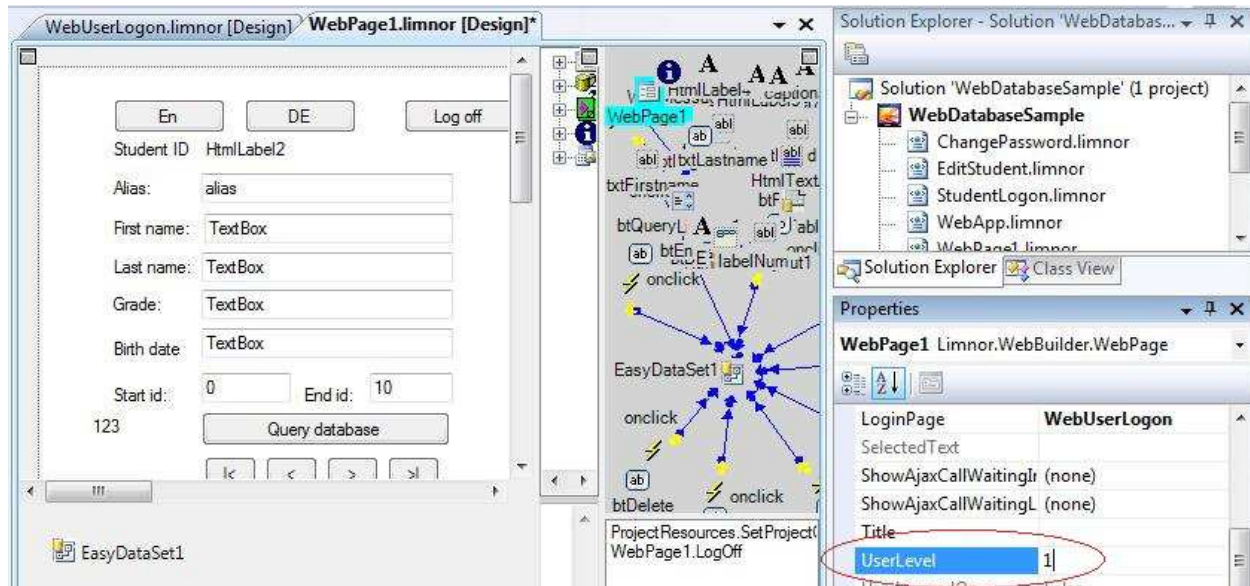
If a user logs in and visits a web page requiring a user level less than the user's level then the web page will be re-directed to the log in web page and display a message contained in property "LoginPermissionFailedMessage":



If this property is empty then such a message will be displayed: **You do not have permission to visit this web page.**

Use web page user level

Each web page has a UserLevel property. Set it to 0 or a positive integer to indicate the desired user level. For example, if a web page is set to 3 then only users with access level 0, 1, 2, and 3 may access the web page. A user with access level 4 or larger cannot access the web page.



With above setting, WebPage1 may only be accessed by users with level 0 and 1.

Access information only related to logged on user

Suppose we want to let students edit their information. We need to make sure that a student may only edit his/her information.

We may add a login alias field and a password field to the Student table to enable the log in of students.

This SQL script can be used to add a login alias field and a password field to the Student table:

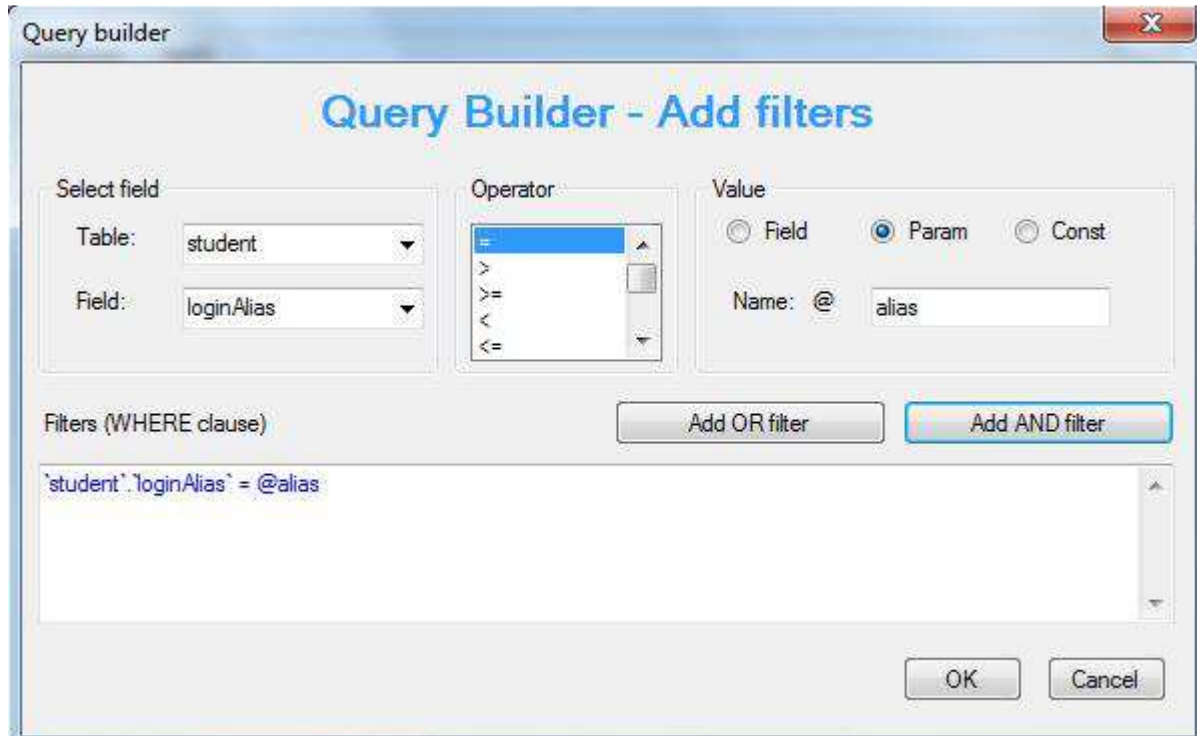
```
Alter Table Student add loginAlias varchar(20) not null, add userPass varchar(50) null;
```

After entering unique loginAlias values for all the existing records, use the following script to add unique index to the loginAlias table

```
Alter Table Student add unique (loginAlias);
```

Now we have prepared the Student table for enabling log on service.

Create a new Student Edit web page just as like WebPage1, but without the navigation buttons. Also, we use just one query parameter to search for the loginAlias.



The image shows a 'Query builder' window titled 'Query Builder - Add filters'. It contains three main sections: 'Select field', 'Operator', and 'Value'. In 'Select field', 'Table' is set to 'student' and 'Field' is set to 'loginAlias'. In 'Operator', the '=' operator is selected from a list. In 'Value', the 'Param' radio button is selected, and the 'Name' is '@alias'. Below these sections are buttons for 'Add OR filter' and 'Add AND filter'. At the bottom, there is a text area showing the filter expression: 'student.loginAlias = @alias'. At the very bottom are 'OK' and 'Cancel' buttons.

Query builder

Query Builder - Add filters

Select field

Table: student

Field: loginAlias

Operator

=

>

>=

<

<=

Value

☐ Field ☒ Param ☐ Const

Name: @ alias

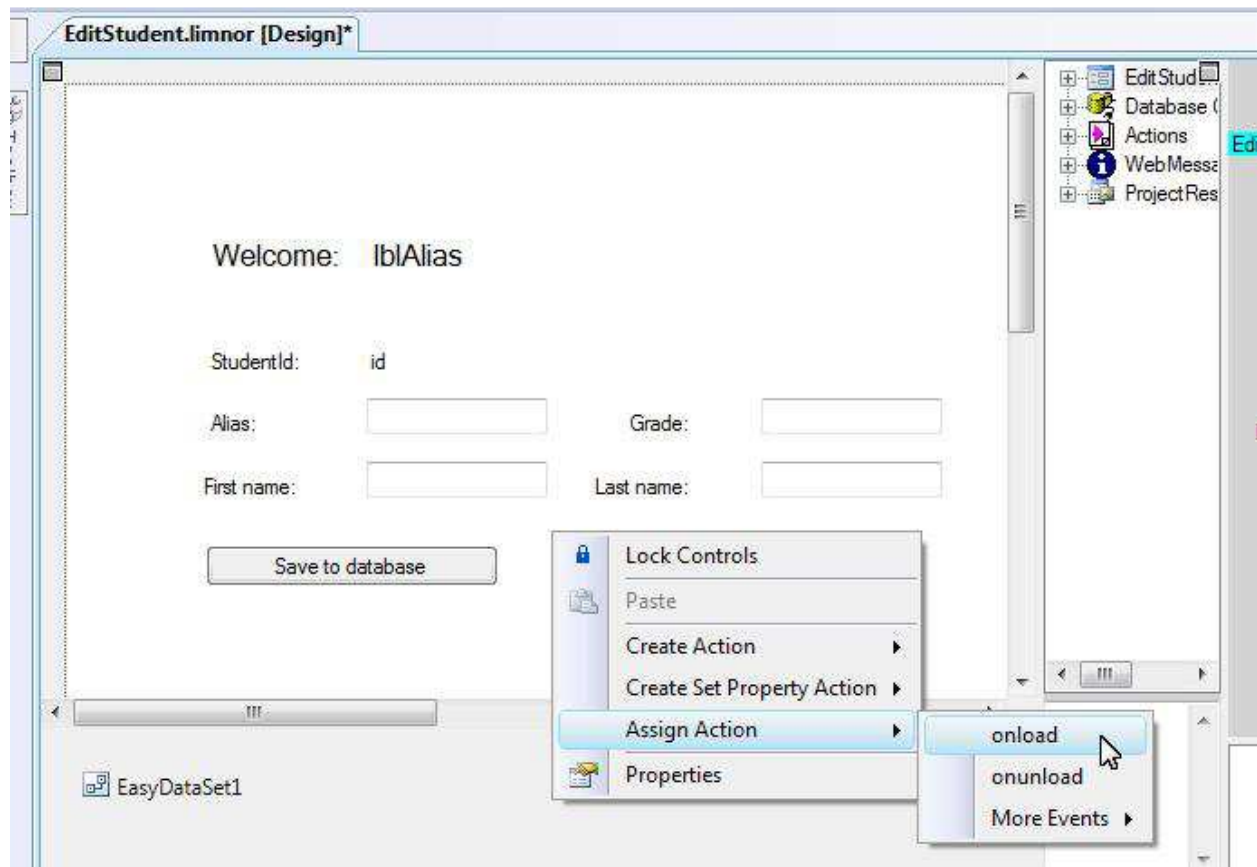
Filters (WHERE clause)

Add OR filter Add AND filter

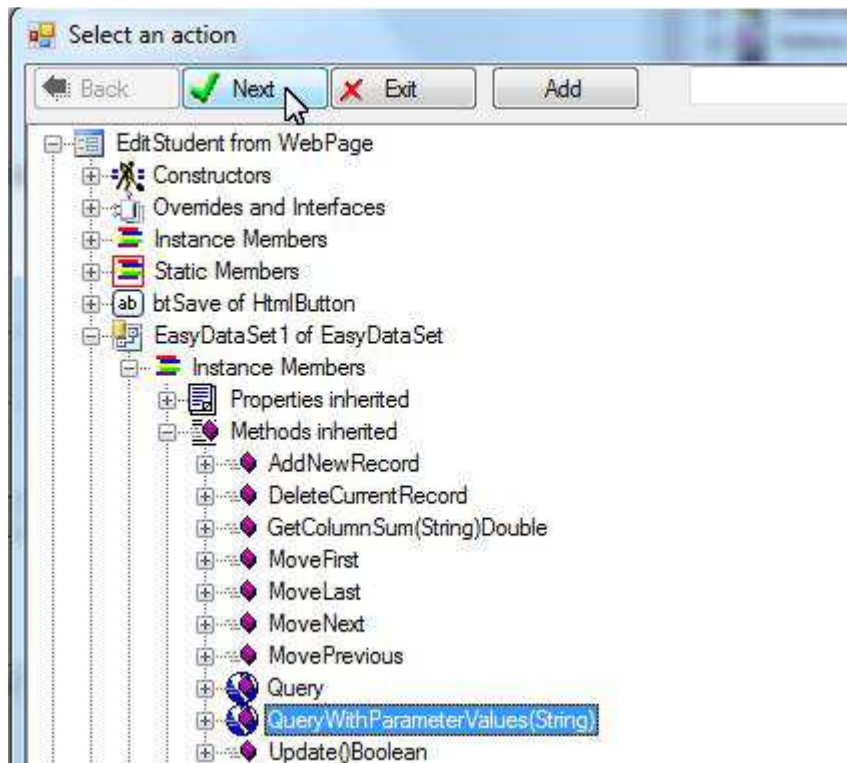
'student'.loginAlias = @alias

OK Cancel

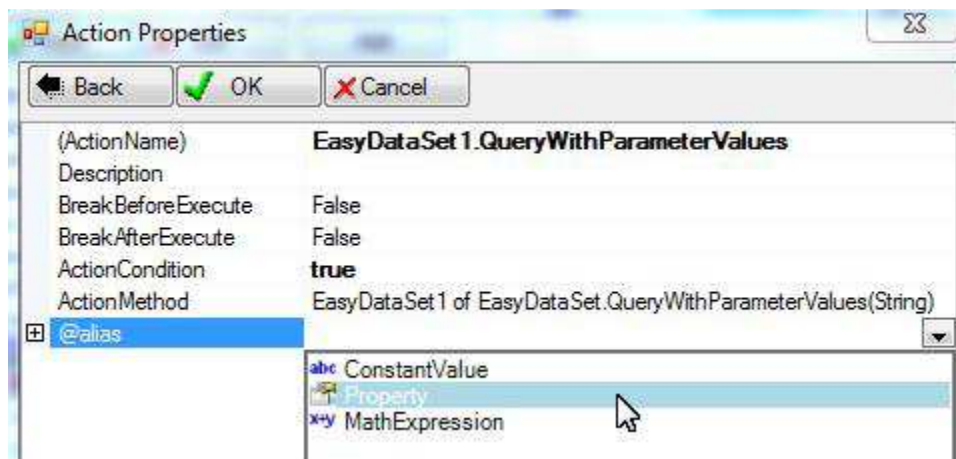
For this web page we will fetch data when the page is loaded. Right-click on the page, choose "Assign Action", choose "onload" event:

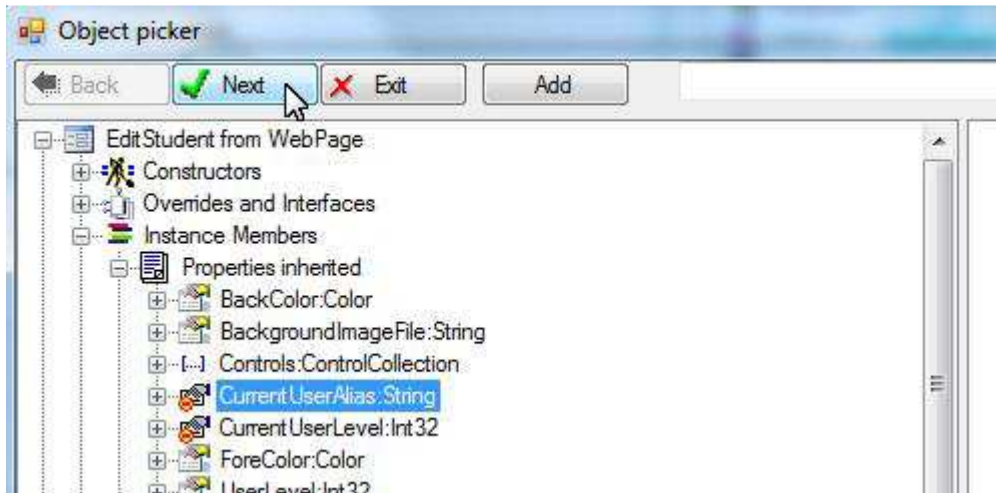


Select QuertWithParameterValues and click Next:

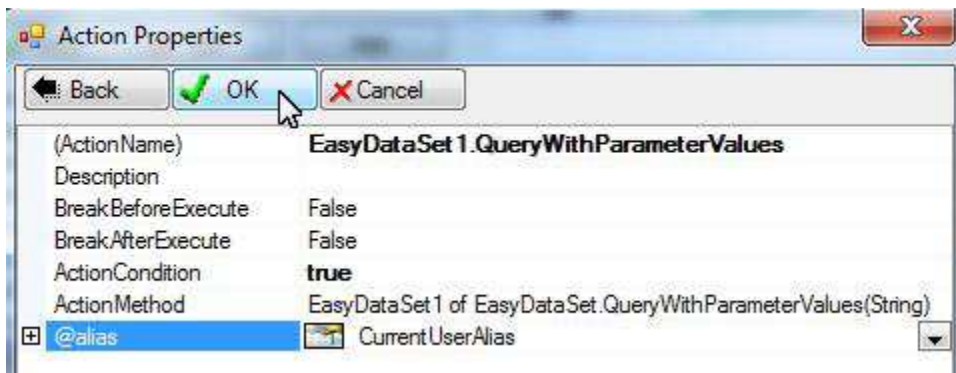


For “@alias”, select the CurrentUserAlias of the page:

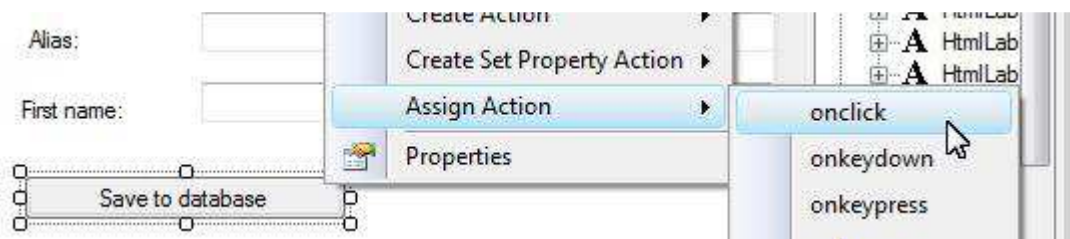




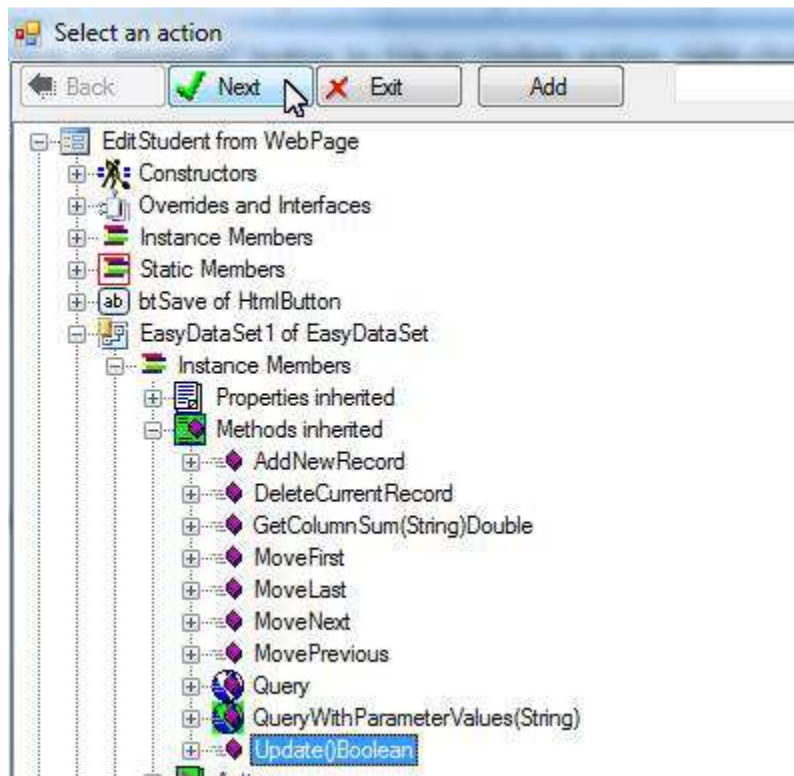
Click OK to finish creating this action and assigning it to the “onload” event:



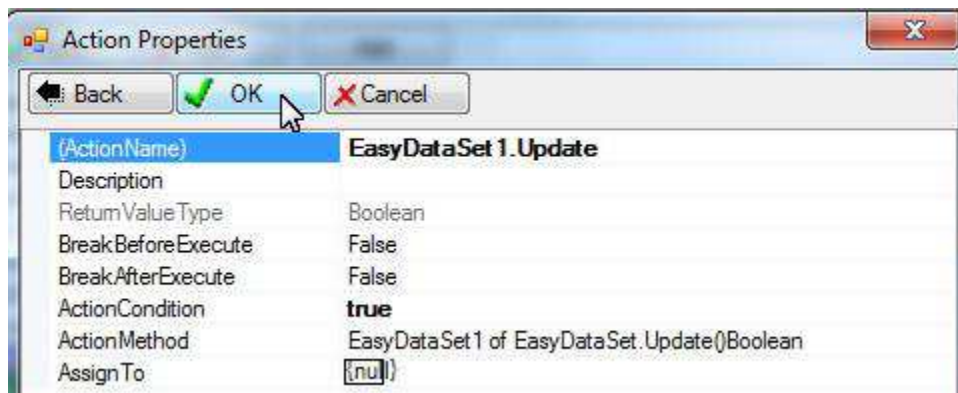
To let the “Save to database” button trig an Update action, right-click the button and chose “Assign Action”. Choose “onclick” event:



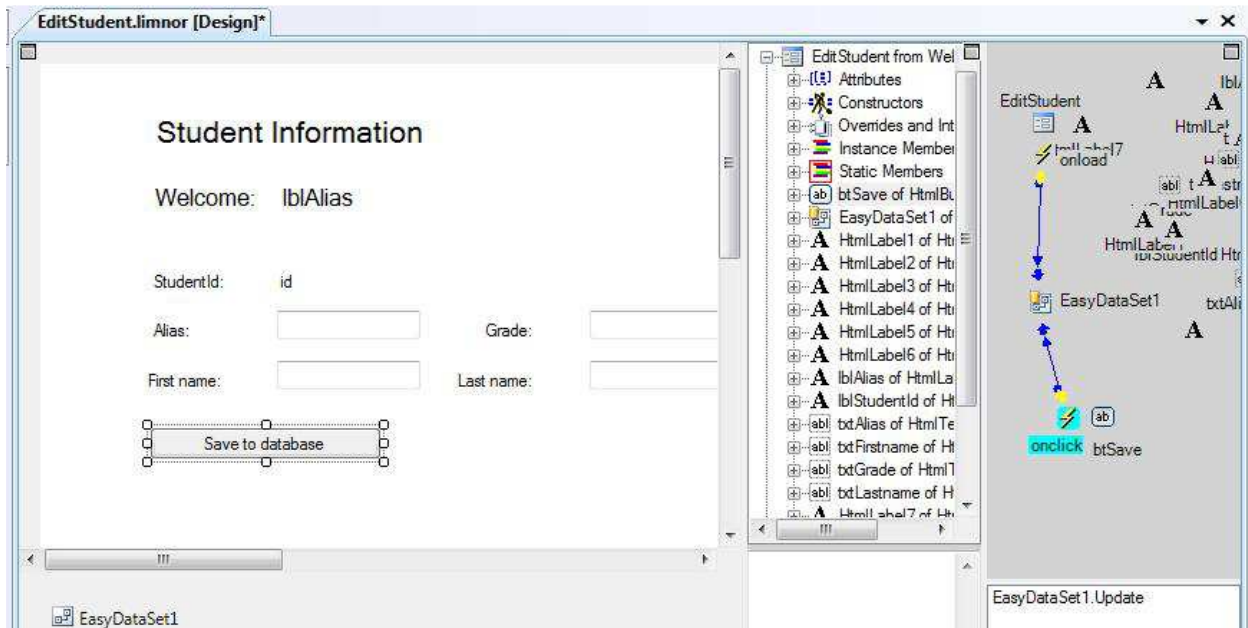
Choose “Update” method and click Next:



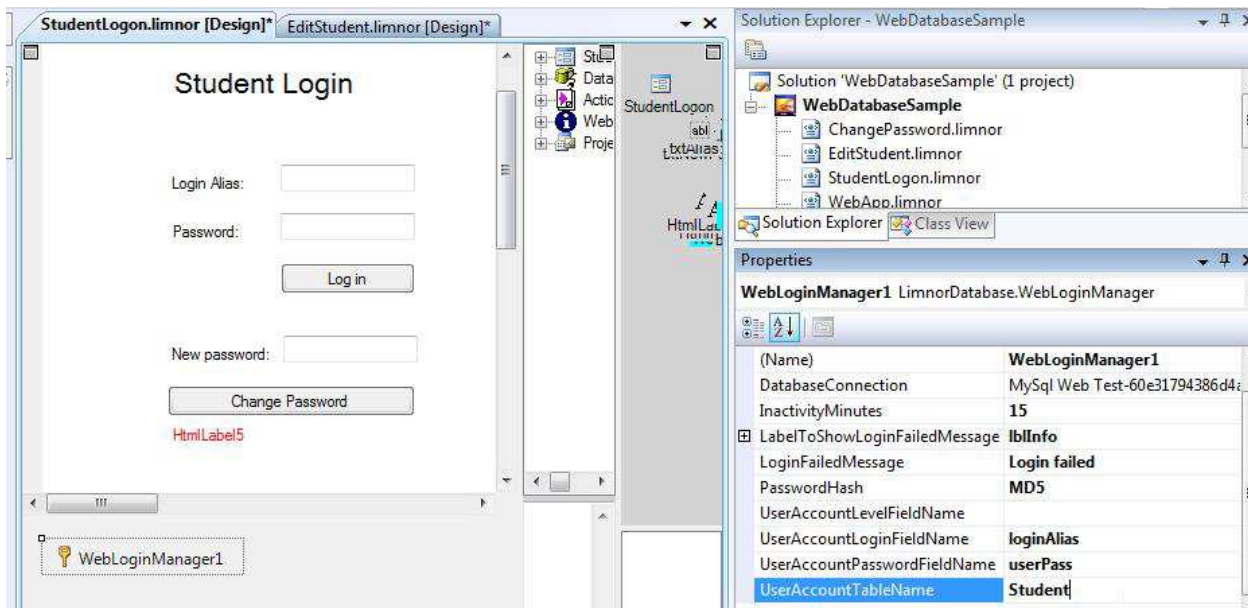
Click OK to finish creating this action and assigning it to the button:



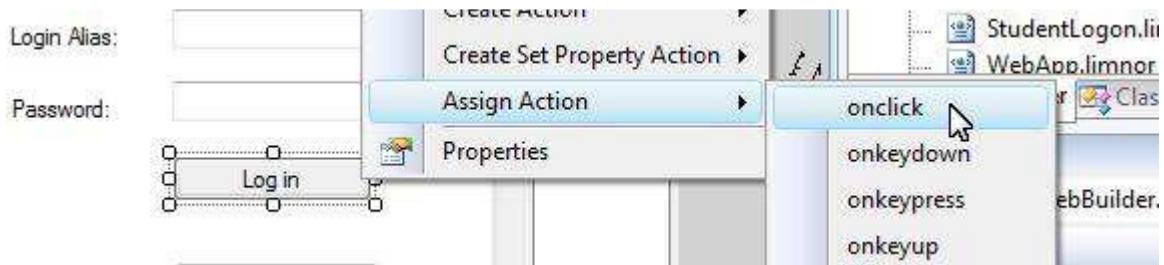
This is the student information page:



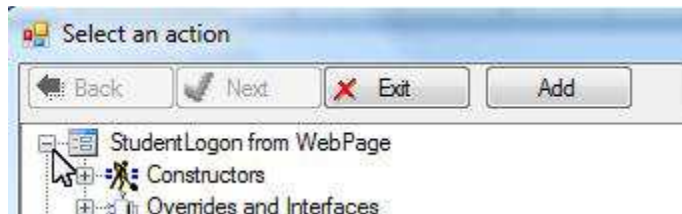
We'll create a student login page just like the web user login page, but the WebLoginManager will have configurations pointing to the Student table:



To setup login, right-click the "Log in" button, choose "Assign Action". Choose "onclick" event:



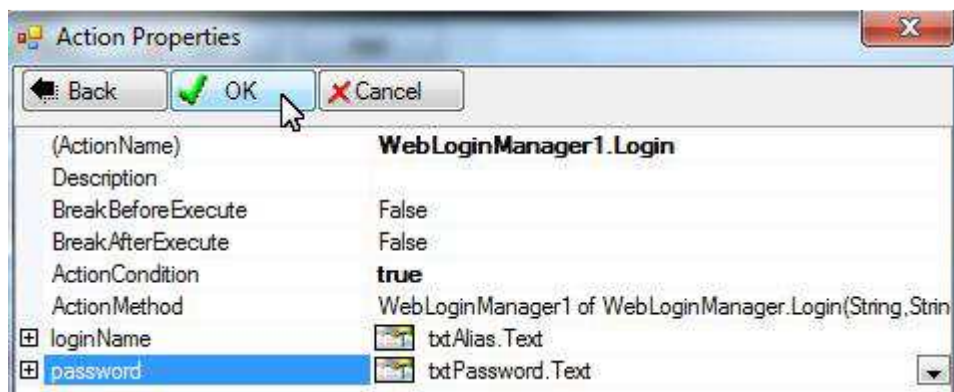
Select the Login method and click Next:



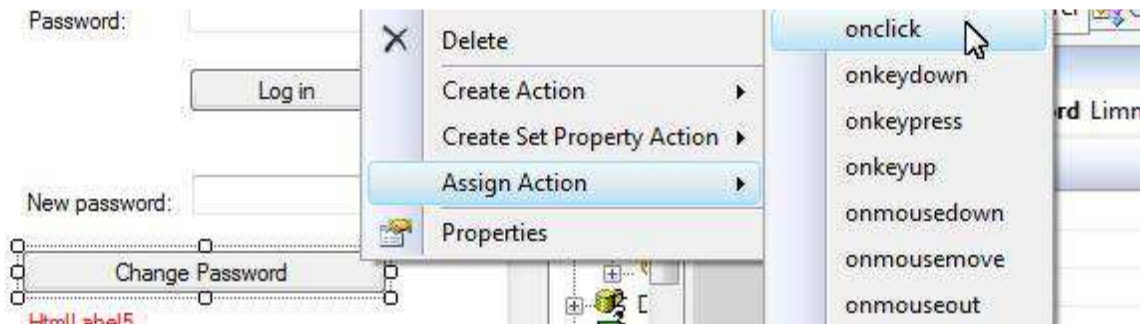
Set "loginName" to the Text property of the text box for the login alias.

Set "password" to the Text property of the text box for the password.

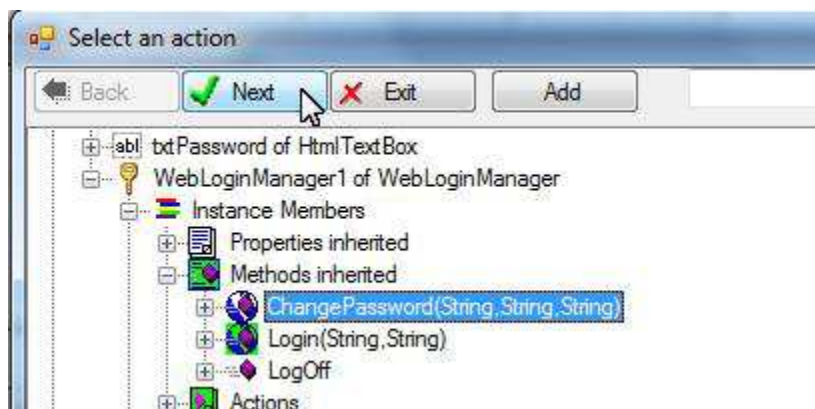
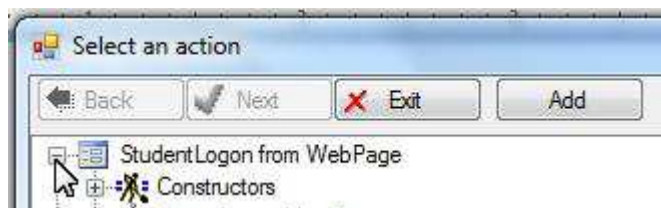
Click OK to finish creating this action and assigning it to the button:



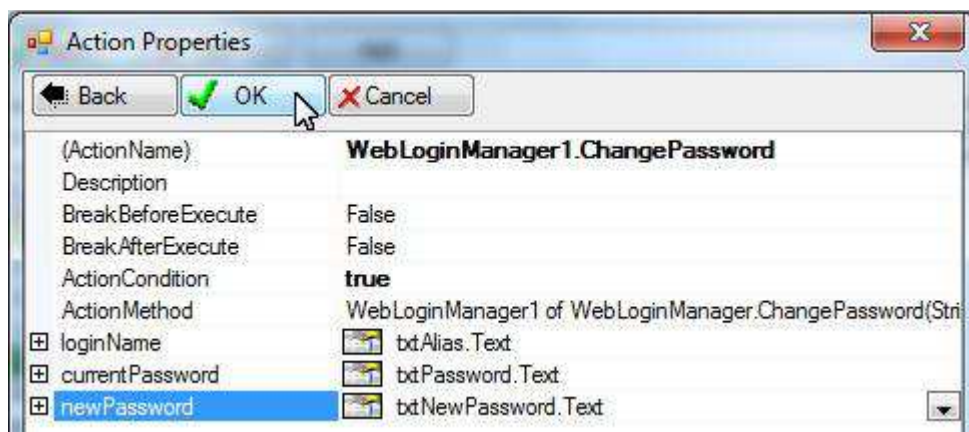
Right-click the "Change password" button and choose "Assign Action". Choose "onclick" event:



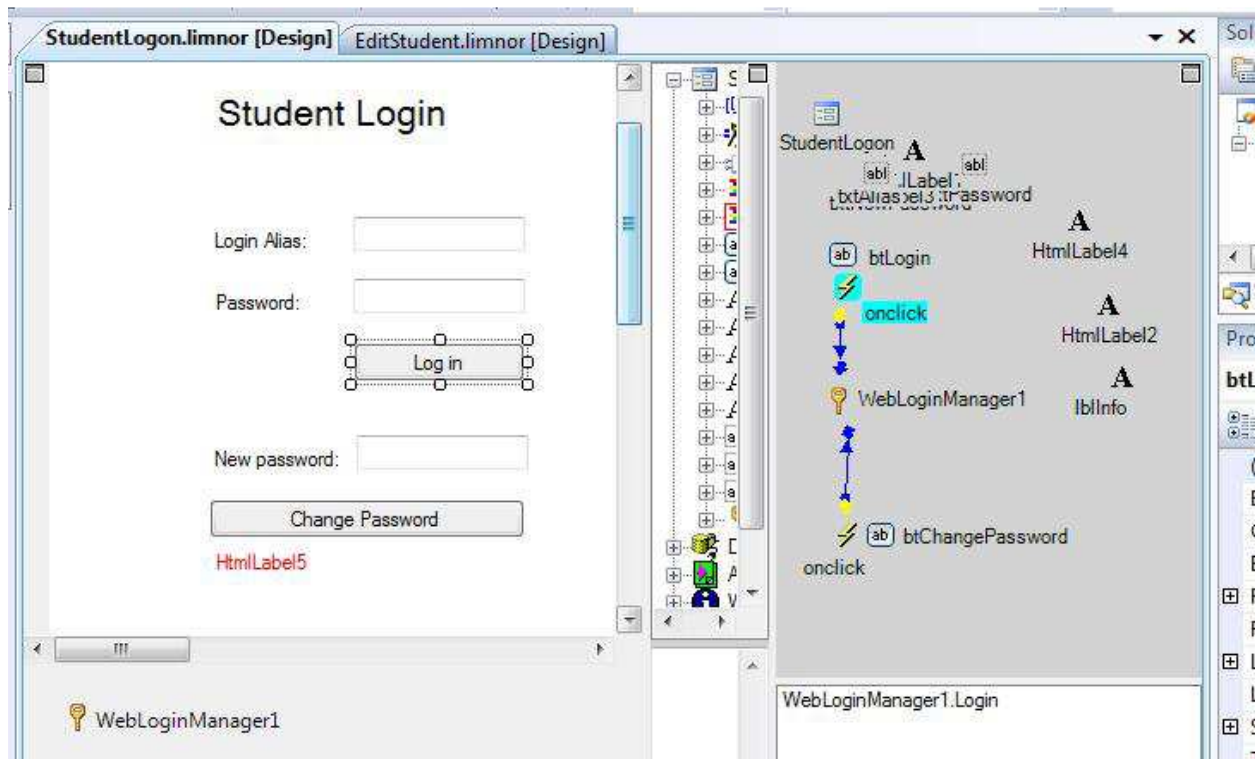
Choose "ChangePassword" and click Next:



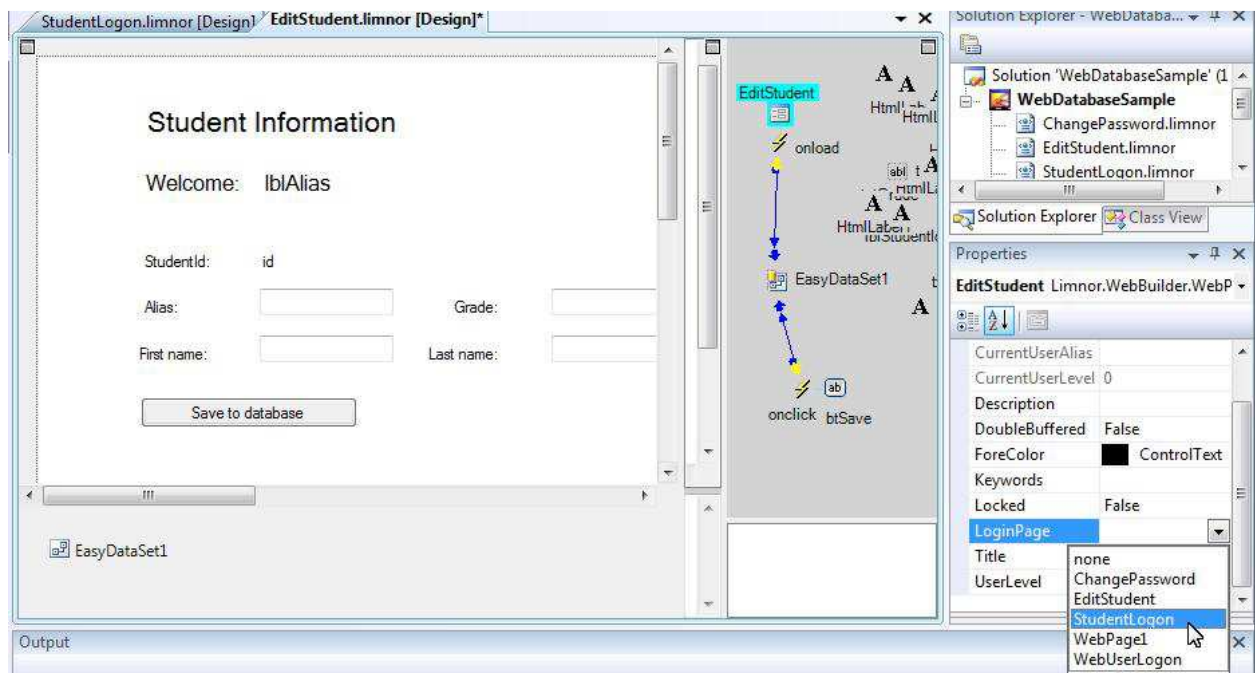
Set the action parameters to the Text properties of the text boxes on the page. Click OK to finish creating this action and assigning it to the button:



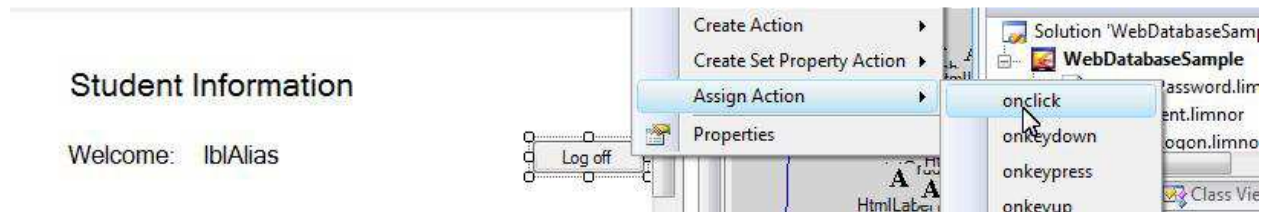
This is our Student Login page:



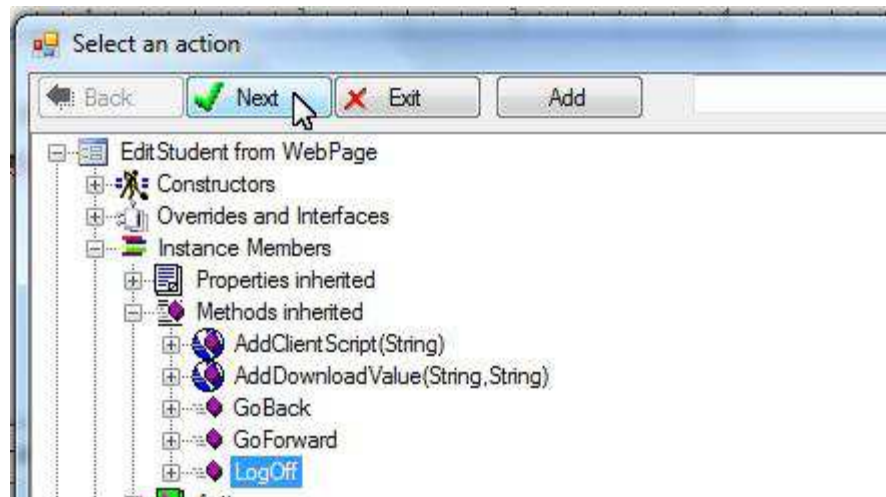
Set the Student Editor page to be protected by the Student Login page:



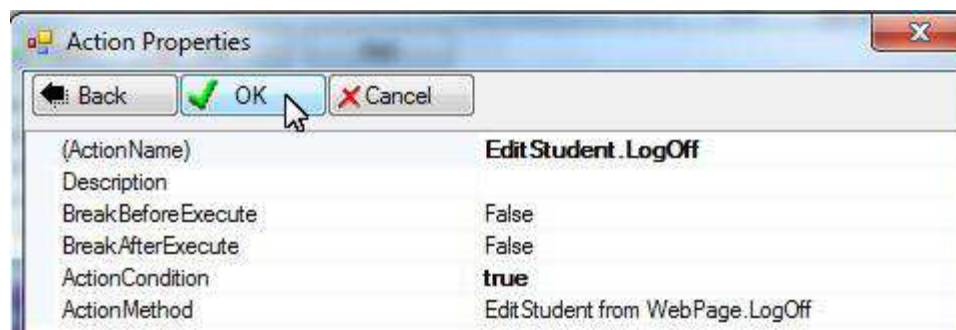
One more thing we can do is to add a Log Off button. Right-click the button; choose "Assign Action"; choose "onclick" event:



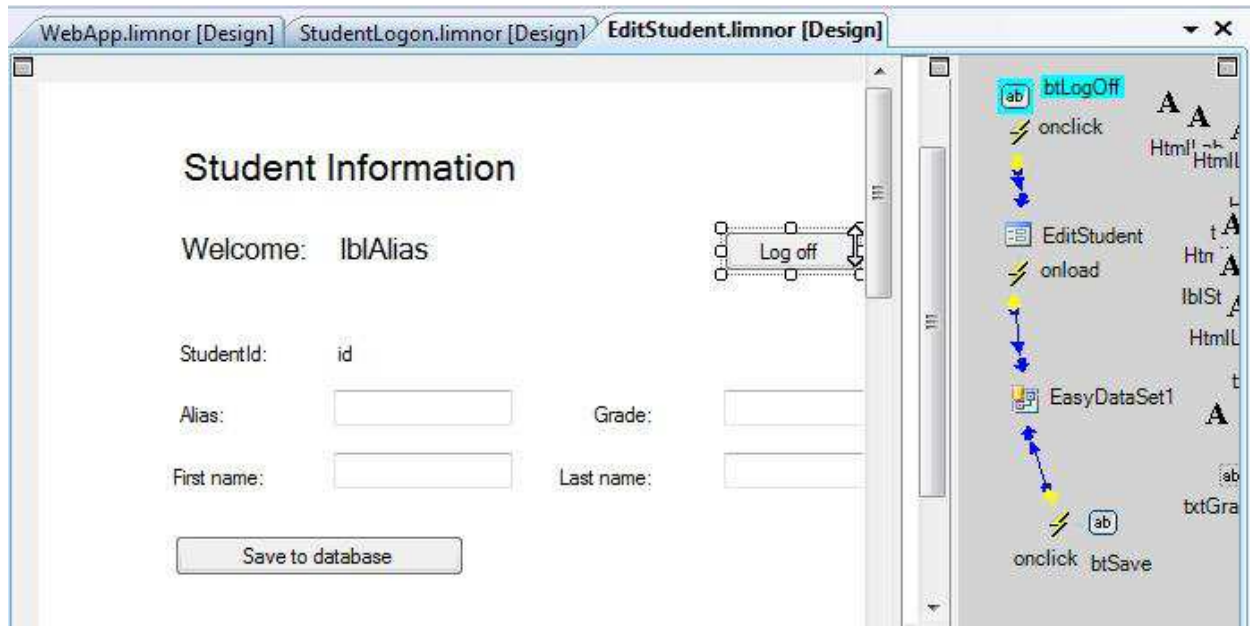
Choose the LogOff method and click Next:



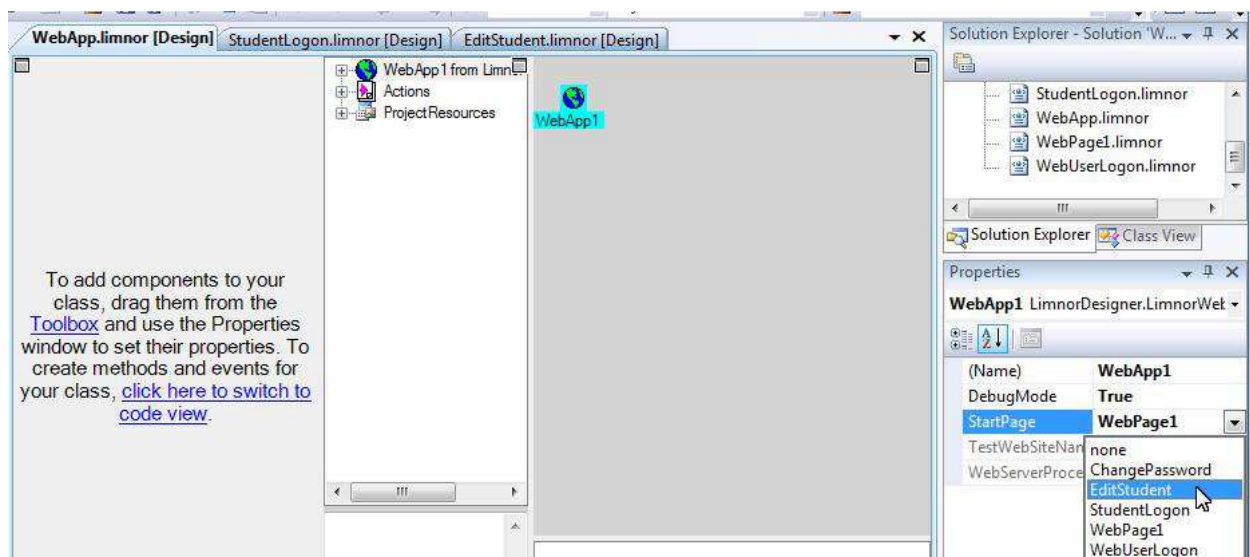
Click OK to finish creating this action and assigning it to the button:



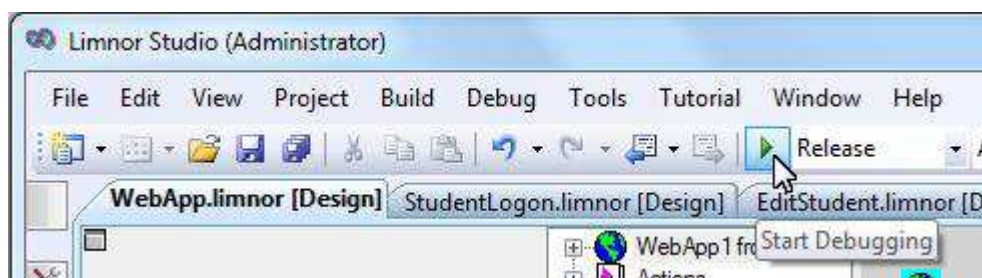
This is the student information page:



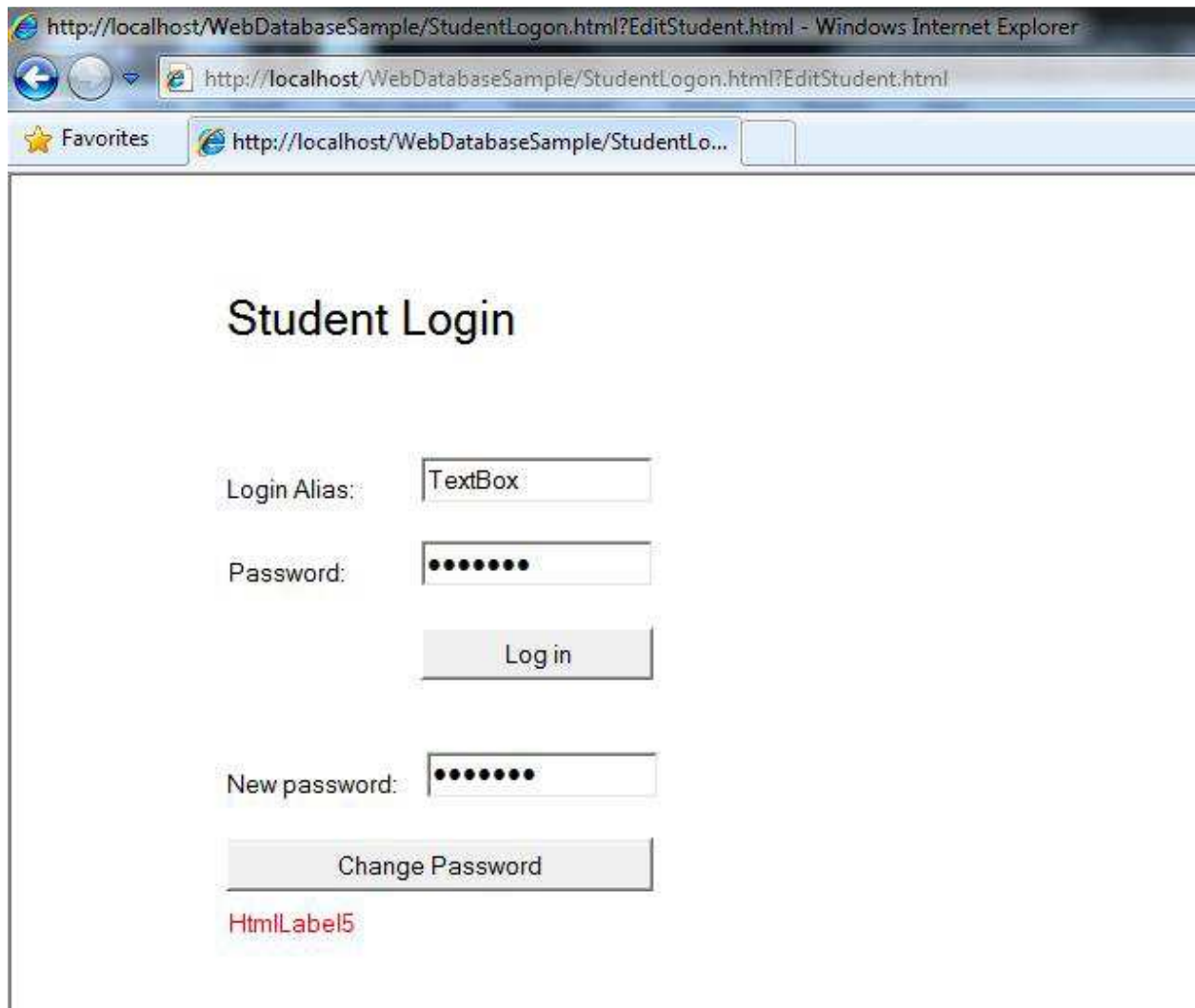
Set the Student Editor page as the StartPage for our testing:



Start testing:



The student login page appears:



The screenshot shows a Windows Internet Explorer browser window. The address bar displays the URL: `http://localhost/WebDatabaseSample/StudentLogon.html?EditStudent.html`. The page content is titled "Student Login". It features two main sections: a login section and a password change section. The login section includes a "Login Alias:" label, a text box containing the text "TextBox", a "Password:" label, a password box with seven dots, a "Log in" button, and an "HtmlLabel5" label in red text. The password change section includes a "New password:" label, a password box with seven dots, and a "Change Password" button.

Suppose we know user “mi” does not have a password. We leave the password blank and enter a new password in the “New password” text box. Click “Change Password”:

Student Login

Login Alias:

Password:

New password:

HtmlLabel5

The message says "Password changed":

Login Alias:

Password:

New password:

Password changed

We may use the new password to login:

Login Alias:

Password:

New password:

Password changed

The student edit page appears with the corresponding student information loaded:

http://localhost/WebDatabaseSample/EditStudent.html - Windows Internet Explorer

http://localhost/WebDatabaseSamp

http://localhost/WebDatabaseSample/EditStude...

Student Information

Welcome: mi

StudentId: 1

Alias: Grade:

First name: Last name:

The student may edit the information and click “Save to database” button to save the modifications to the database.

Field Editors

See <http://www.limnor.com/support/webDatabaseProgramming3.pdf>

Data Repeater

In chapter “Bind-data to single field”, we saw that a form can be designed with data-bound controls. Data from database are automatically displayed on the controls. The user may modify data on the controls and the data modifications can be saved back to database.

In such arrangement, one record is displayed on one web page.

Data Repeater allows you to design the form with data-bound controls, but the same design can be repeated on one web page. Thus many records can be displayed on one web page.

For details, see <http://www.limnor.com/support/WebDataRepeater.pdf>

Change Query Filters at Runtime

See <http://www.limnor.com/support/webDatabaseProgramming4.pdf>

Create New Records

See <http://www.limnor.com/support/webDatabaseProgramming5.pdf>

Data Streaming

See <http://www.limnor.com/support/webDatabaseProgramming5.pdf>

Fetch Data of One-to-Many Relation

See <http://www.limnor.com/support/webDatabaseProgramming5.pdf>

Change Data-binding

See <http://www.limnor.com/support/webDatabaseProgramming5.pdf>

Feedbacks

Please send your feedbacks to support@limnor.com, thanks!