Assignment 2 – ATM in ASP.NET

# Purpose

* Create a basic ATM using ASP.NET
* Learn C# and the .NET Framework
* Create a database using MS SQL Server
* Interact with the database using LINQ to SQL

# Due Date/Course Weight

This assignment must be demonstrated by: **5:59:59pm Friday March 21st 2012**

* NOTE: Late assignments will not be accepted.

This assignment is wroth: **10% of your total course mark**

# Assessment

When you are ready to demo your results please print off the Assignment Assessment page provided on Blackboard. Provide your Name and Student Number.

# Summary of Tasks

Listed below are the sections of the assignment which will need to be completed:

1. Read every word in the assignment
2. Create a database to house the ATM bank information
3. Create tables in your new database to store ATM information
4. Create a new solution to house your ATM application
5. Create a LINQ to SQL Data Context to interact with your database
6. Write a program to interact with your database

# Create a Database

Use the Visual Studio’s Server Explorer to create a database named ATM2. Default properties will suffice – do not worry about advanced configuration.

# Create database Tables

Use the SQL Server Management Studio to create the following tables:

* People  
  Holds the information regarding the users of the database  
  + PersonId, Primary Key, int, not null, with Identity and Auto-increment constraints
  + FirstName, varchar(50), not null
  + LastName, varchar(50), not null
  + EmailAddress, varchar(255), not null
  + Password, varchar(10), not null
  + Social Insurance Number, varchar(11), not null
  + DateAccountCreated, datetime, not null
  + LastLoginDateTime, datetime, not null
* TransationType  
  Describes the transaction being done by the user  
  + TransactionTypeId, int, not null
  + Transaction, varchar(10), not null

Manually add the following information in the Transaction Type table using the ‘Show table data’ feature of the Server Explorer

**TransactionTypeId, Transaction**

1, Deposit,

2, Withdraw

* Transactions  
  Tracks the users actions in the system  
  + TransationId, primary key, int, not null, with identity and auto-increment constraints
  + PersonId, int, not null
  + TransactionTypeId, int, not null
  + AmountTransferred
  + DateOfTransaction, datetime, not null

# Create a new Project

Use Visual Studio to create a new project.

1. Open Visual Studio (2008 or 2010)
2. Click File -> New Web Site and the New Web Site dialog box will appear
3. Be sure that C# is selected from the Installed Templates menu
4. Select ASP.NET Empty Web Site
5. Name your project ATM2
6. Select your save Location
7. Click OK

Visual Studio will create an empty web site with a very basic ‘web.config’ file.

# Create LINQ to SQL Data Context

Use Visual Studio to create a LINQ to SQL Data Context using all the tables created in ‘Create Database Tables’ section of this assignment.

1. Right click on the ‘ATM2’ project file and select ‘New Folder’
2. Name this new folder ‘App\_Code’
3. Right click on the ‘App\_Code’ folder and select ‘Add New Item’
4. Be sure that C# is selected from the Installed Templates menu
5. Scroll down until you find ‘LINQ to SQL Classes’ and select that item.
6. Name the new object ‘ATM’ and click the ‘Add Button’

Visual Studio will create a new empty Data Context that can be manipulated using the Server Explorer.

1. Open the Server Explorer
2. Select the ATM2 database you created in the ‘Create A Database’ section.
3. Navigate to the ‘Tables’ folder of your ATM2 database.
4. Drag the tables created in the ‘Create database Tables’ section into the visual design area of your LINQ to SQL data context.
5. Save your project.

Upon saving the assignment Visual Studio will automatically generate all ADO.NET code needed for your application including but not limited to: Insert, Update and Delete. Use LINQ to query the new data context.

For Example:

ATMDataContext db = new ATMDataContext();

var transactions = (from transaction in db.Transactions select transaction);

# Write your application

Write an application to simulate the basic functionality of an Automated Teller Machine using ASP.NET. Your web site must include the following features:

**Create an App Theme**

* Your theme should include a common header, footer and menu(s).
* Use CSS to create this theme
* NOTE: Ignore ‘Skins’ for this application

**Create two master pages**

* Create a master page for your website that implements your theme created above. Be sure to include the common header, footer and menu(s).
  + Use the pages listed below to create your menu(s).
* Create a second master page for your website that will implement the same theme created above but does not include the menu(s). This login page will only implement your common header and footer but leave the rest of the site blank. Use this master page to implement your ‘login’ and ‘new user’ pages.

**Create the following web pages:**

* Default.aspx
  + Implement the second master page (without menus)
  + This web page will contain the screen necessary to log in to your ATM. Login using the users PersonId and Password
  + If the user does not have an account provide a link to ‘NewUser.aspx’
  + If login is successful bring the user to ‘Overview.aspx’
* NewUser.aspx
  + Implements the second master page (without menus)
  + This page provides a form necessary to create an account in the ATM
  + Upon successful creation the user is automatically logged into the ATM and taken to Overview.aspx
* Overview.aspx
  + Implements the first master page (with menus)
  + This page provides the user with a quick overview of their account including:
    - Amount of money in the account
    - A quick list of the last three transactions
* DepositMoney.aspx
  + Implements the first master page (with menus)
  + This page provides a form that allows the user to deposit money into the account
* WithdrawMoney.aspx
  + Implements the first master page (with menus)
  + This page provides a form that allows the user to withdraw money from the account
* Report.aspx
  + Implements the first master page (with menus)
  + A grid that shows all past users past transactions.
* CloseAccount.aspx
  + Implements the first master page (with menus)
  + This page provides a form that allows the user to close his/her account.
  + The user must verify account information (username and password) to be able to close the account.