INFS3204/7204 Practical 4 – LINQ 2 SQL

Note that: for all pracs in this course, you need to submit your code in *a single zip file* to the blackboard. Pracs without submitting your codes *before your own prac session expires* will be marked 0. If you submit the codes before your own prac session expires and do not attend the prac session, you have to email your tutor to ask him to mark it offline.

The goal of this practical is to explore the LINQ 2 SQL. All practicals will have to be developed with Microsoft Visual Studio 2010 using C# as the programming language. No other languages will be accepted. This practical contributes to 5% of your overall grade. You must **submit your codes before your scheduled lab session expires in week 8**, in order to get it marked. **You are not allowed to change your session.**

Preparation

You are assumed to have completed Practical 3. This practical extends what you have already done in Practical 3. For this practical, you are only allowed to use **LINQ 2 SQL** for database communication. For those students who are unfamiliar with the LINQ 2 SQL, please watch the tutorial video (http://itee.uq.edu.au/~shenht/P4video.mp4) before trying to do this practical. You are free to choose Web Form or MVC for this Practical.

You need to make modifications to the Practical 3 (All other requirements are the same as in practical 3. You can use the same web services as in practical 3 **except** those that use traditional ADO.NET).

This practical is divided into three tasks.

- **Task 1: Creating a Login page (1 Mark)**
- **Task 2: Creating a Sign-Up page (1 Mark)**
- **Task 3: Implementing LINQDatabaseService (3 Marks)**

Task 1: Creating a Login page (1 Mark)

For this task, you first need to modify the database that you have created for practical 3 and add additional table(s) if needed to support multiple users to use the application. **Please note** that each user should be able to search among his/her own contacts, not all the existing contacts in the Contact table (You have to design an appropriate database schema to achieve this).

- **Login page (0.5 mark)**:

  Create a Login page with the following requirements:

  - Username (Textbox, required field)
  - Password (Password-Textbox, required field)
  - Login button
  - Sign-Up link

  When user clicks on the Login button, the **Login** web method from your **UserAccount** web service should be invoked. Create a new web service called **UserAccount** and add the following web method to it:

- **Login web method (0.5 mark):** This web method should receive the username and password from the Login page and check if they are valid. If username doesn't exist in the database or if the username and password don't match with each other, an error message should be displayed to user. Otherwise, the user will be redirected to the **Add Contact page.**

Task 2: Creating a Sign-Up page (1 Mark)

- **Sign-Up page (0.5 mark)**:

Create a Sign-Up page with the following requirements:

- Username (Textbox, required field)
- Password (Password-Textbox, required field)
- Confirm-Password (Password-Textbox, required field)
- Sign-Up button

When user clicks on the Sign-Up button, the **CreateAccount** web method from your **UserAccount** web service should be invoked. In the **UserAccount** web service, add the following web method to it:

- **CreateAccount web method (0.5 Mark)**: It has to receive the username and password from the Sign-Up page. An error message should be displayed to user if username has already existed in the database, or password does not match confirm-password. Otherwise, username and password will be saved in the database, and user will be redirected to the **Login page**.

Task 3: Implementing LINQDatabaseService (3 Marks)

For this task you need to create a new Web Servide called **LINQDatabaseService**, which has the following web methods (as in practical 3 but using LINQ 2 SQL).

- **FullnameExists (0.5 Mark)**: checks if a contact with the received fullname (firstname + lastname) already exists in the database
- **PhoneExists (0.5 Mark)**: checks if the phone number already exists in the database
- **SaveContact (0.5 Mark)**: adds the new contact information into the database
- **GetContact (0.5 Mark)**: gets the contact information from the database
- **UpdateContact (0.5 Mark)**: updates the contact information in the database
- **RemindBirthday (0.5 Mark):** gets current date (**please note** that only day and month properties of the current date are useful for this practical), and returns those contacts' information (firstname + lastname + email) whose birthday is today.

You need keep the following web pages **as in practical 3,** including other necessary web services to make the application work properly:

- **Add Contact page**
- **Search Contact page** (including the **Edit** functionality)
- **Birthday Reminder page**