PHÁT TRIỂN ỨNG DUNG CSDL 1

Tháng 9/2014

LẬP TRÌNH ỨNG DỤNG VỚI NGÔN NGỮ C#

Bài tập lập trình giao diện kết nối dữ liệu với C#

Tóm tắt nội dung bài thực hành:

Hướng dẫn về môi trường lập trình, cấu trúc chương trình, sử dụng các kiểu dữ liệu cơ bản, các cấu trúc điều khiển của ngôn ngữ lập trinh c#.

Bộ môn **Hệ thống thông tin**

Khoa Công nghệ thông tin

ĐH Khoa hoc tư nhiên TP HCM

MỤC LỤC

1 Yêu cầu chung

- Sinh viên thực hiện 3 bài tập bên dưới trong cùng một project.

2 Lược đồ dữ liệu

Bảng	Cột	Kiểu dữ liệu	Mô tả
NguoiDung	maNguoiDung	Chuỗi	
	hoTen	Chuỗi	
	email	Chuỗi	
	gioiTinh	Chuỗi	
	maPhong		

Dữ liệu gồm các bảng sau:

2.1 NguoiDung

maNguoiDung

3 Các lớp dữ liệu

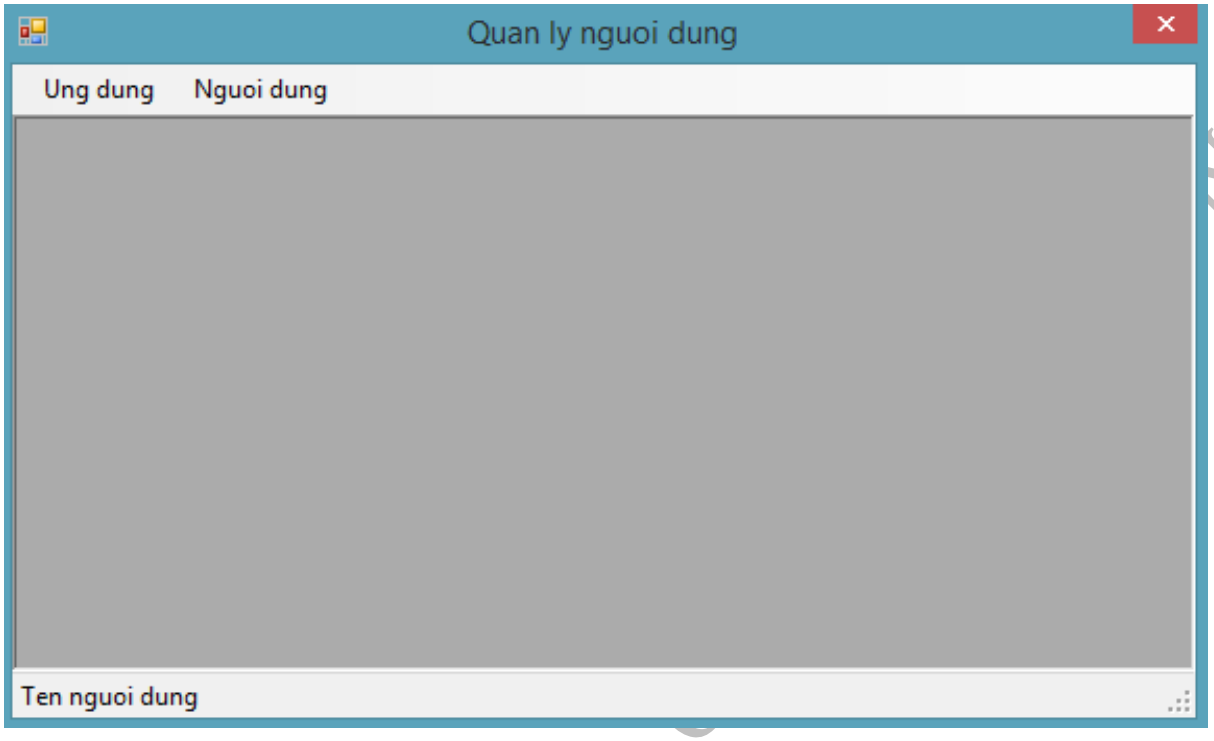
- Tạo thư mục Models bằng cách nhấn chuột phải vào project, chọn Add/New Folder.
- Tạo lớp NguoiDung trong thư mục Models của project.
- Lớp NguoiDung có đầy đủ thuộc tính của bảng NguoiDung trong dữ liệu.
- Tạo các properties để truy xuất các thuộc tính trong lớp người dùng.

4 Giao diện chính (frmMain)

4.1 Thiết kế giao diện

- Thiết kế giao diện sau bằng các control: MenuStrip và StatusStrip.
  - Menu "Ứng dụng" bao gồm item "Thoát"

- Menu “Người dùng” bao gồm item “Quản lý”
- Status Strip gồm 1 item: “Tên người dùng”



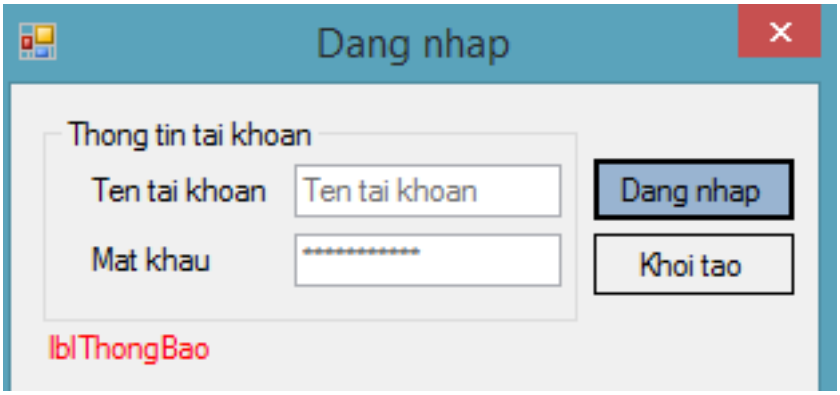
4.2 Xử lý

- Giao diện chính frmGiaoDienChinh:
  - Thêm thuộc tính private _nguoiDung với kiểu NguoiDung đã tạo trong lớp Models/NguoiDung và gán giá trị bằng null.
  - Khi giao diện mới nạp lên, mở giao diện Đăng nhập (frmDangNhap) bằng cách tạo một thể hiện của lớp frmDangNhap và gọi hàm ShowDialog().
- MenuStrip:
  - Khi người dùng nhấn vào item “Thoát”, chương trình sẽ kết thúc
  - Khi người dùng nhấn vào item “Người dùng/Quản lý”, giao diện quản lý người dùng (frmQLNguoiDung) sẽ được hiển thị.
- Status item: “Tên người dùng” sẽ hiển thị tên người dùng đang đăng nhập hiện tại.

5 Đăng nhập

5.1 Thiết kế giao diện đăng nhập

- Thiết kế giao diện như hình sau:



5.2 Thiết kế xử lý giao diện

- txtMatKhau:
  - sử dụng thuộc tính PasswordChar để textbox luôn hiển thị dấu '-' khi người dùng nhập mật khẩu.
  - Hiển thị 8 ký tự * màu xám khi chưa nhập dữ liệu.
  - Khi người dùng nhập dữ liệu thì xoá các ký tự màu xám và chữ người dùng nhập sẽ có màu đen.
- txtTenTaiKhoan:
  - hiển thị chữ "Tên tài khoản" với màu xám khi textbox không có dữ liệu.
  - Khi người dùng nhập dữ liệu thì xoá các ký tự màu xám và chữ người dùng nhập sẽ có màu đen.
- lblThongBao:
  - Sử dụng thuộc tính Font và ForeColor để in đậm và tô màu đỏ dòng thông báo.
  - Mặc định hoặc khi không có lỗi, người dùng sẽ không thấy label này.
- btnKhoiTao: đưa các textbox, label về lại trạng thái ban đầu khi giao diện được mở lên.
- btnDangNhap:
  - Kiểm tra tên tài khoản có nằm trong dữ liệu không. Nếu có thì tắt giao diện đăng nhập.
  - Nếu không có lỗi xảy ra, đổi lblThongBao sang màu xanh và hiện thông báo "Đăng nhập thành công".

5.3 Thêm hàm xử lý vào lớp NguoiDung

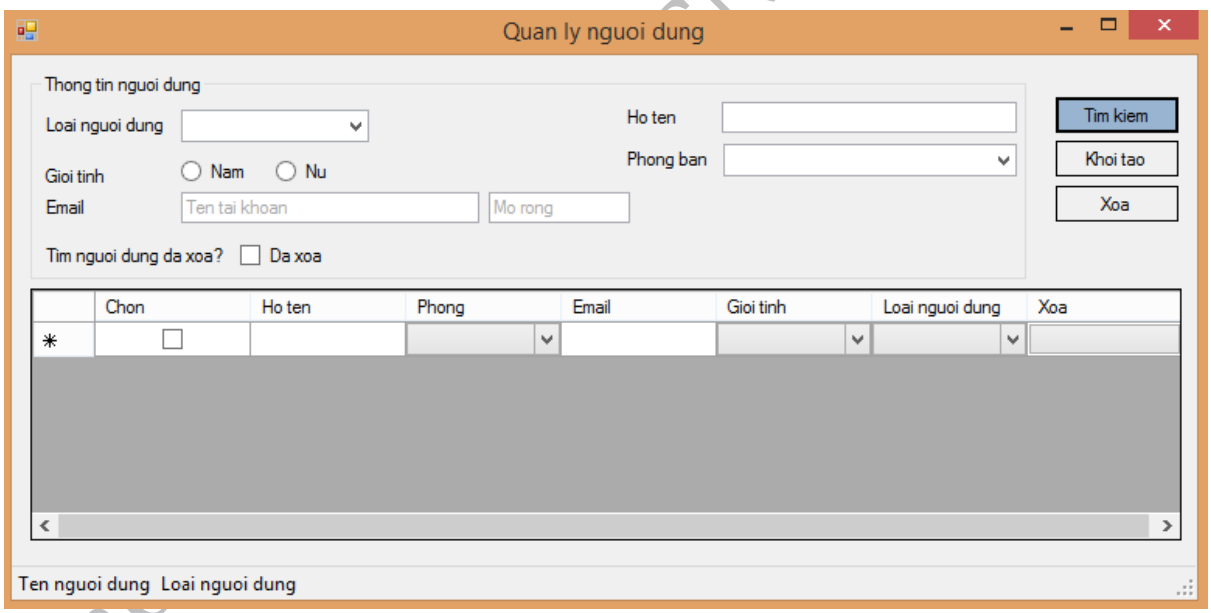
5.3.1 Hàm KiemTraDangNhap

- Tham số: tenTaiKhoan, matKhau
- Kết quả trả về: đối tượng NguoiDung được tìm thấy, còn nếu không thì trả về giá trị null.
- Đây là hàm static
- Hàm này được sử dụng trong lớp frmDangNhap để kiểm tra thông tin đăng nhập và sử dụng stored procedure uspLayThongTinTaiKhoan để lấy thông tin người dùng từ tên tài khoản và mật khẩu.

6 Quản lý người dùng

6.1 Thiết kế giao diện

- Thiết kế giao diện như hình sau:



- Các control sử dụng trong giao diện thiết kế thu chi: MenuStrip, StatusStrip,…

6.2 Thiết kế xử lý giao diện

- cboLoaiNguoiDung, cboPhongBan:
  - ➢ Nạp danh sách phòng ban vào 2 combobox này.
  - ➢ Mặc định chọn thành phần đầu tiên trong combobox.
- chkGioiTinh:

- Chỉ được phép chọn 1 trong 2 giá trị nam hoặc nữ

- txtEmail và txtEmailMoRong:
  - txtEmail:
    - mặc định hiển thị chữ "Tên email" bằng màu xám.
    - Xử lý tương tự nút txtTenTaiKhoan ở giao diện đăng nhập.
  - txtEmailMoRong:
    - mặc định hiển thi chữ "Mở rộng" bằng màu xám.
    - Xử lý tương tự nút txtTenTaiKhoan ở giao diện đăng nhập.
    - Thiết lập auto complete:
      - Sử dụng thuộc tính AutoCompleteCustomSource để nhập các phần mở rộng của email gồm: "gmail.com", "yahoo.com", "yahoo.com.vn".
      - Thiết lập AutoCompleteMode là "Suggest".
      - Thiết lập AutoCompleteSource là "CustomSource"
- txtHoTen:
  - Thiết lập auto complete tương tự như txtEmailMoRong với danh sách họ tên được lấy lên từ cơ sở dữ liệu và mode là "Append".
- grdDanhSach:
  - Tạo các cột: Chọn loại DataGridViewCheckBoxColumn; Họ tên, email loại DataGridViewTextBox; phòng, giới tính, loại người dùng loại DataGridViewComboBox; xoá loại DataGridViewButtonColumn với giá trị hiện thị là chữ "Xoá".
  - Không tự động phát sinh các cột bằng cách gán thuộc tính AutoGenerateColumns trong sự kiện Load của form.
  - Xử lý sự kiện CellClick:
    - Khi người dùng chọn 1 ô trên gridview, lấy các dữ liệu trong gridview thể hiện lại lên các ô trong giao diện.

        - ✧ Kiểm tra xem người dùng có đang nhấn vào nút Xoá trên gridview không bằng cách kiểm tra ColumnIndex trong tham số DataGridViewCellEventArgs của hàm xử lý sự kiện.
    - ➢ Xử lý sự kiện MouseHover, MouseLeave để tô màu vàng các dòng được con trỏ chuột quét qua.
- btnXoa: Lặp qua từng phần tử trong thuộc tính SelectedRows của gridview, gán mỗi phần tử bằng biến item. Sau đó, sử dụng hàm RemoveAt của thuộc tính Rows trong gridview và truyền vào item.Index để xoá phần tử tại vị trí được chọn.
- btnKhoiTao: khởi tạo giao diện về lại trạng thái ban đầu.
- btnTimKiem:
    - ➢ Tìm kiếm theo các điều kiện người dùng nhập vào giao diện. Nếu người dùng bỏ nhập vào phần nào thì sẽ không tìm kiếm theo điều kiện đó. Thực hiện chức năng này bằng cách kiểm tra phần nào trên giao diện đã nhập hay chưa và cộng chuỗi SQL cho điều kiện tương ứng.

6.3 Thêm hàm xử lý vào lớp NguoiDung

6.3.1 Hàm static TimKiemNguoiDung

- Tham số: loaiNguoiDung, hoTen, gioiTinh, phongBan, email, timDaXoa.
- Kết quả: danh sách các đối tượng NguoiDung thoả điều kiện tìm kiếm.
- Nếu điều kiện nào không được truy vấn thì tham số của điều kiện đó sẽ là null và câu truy vấn ở điều kiện where sẽ không có điều kiện của tham số này.

6.3.2 Hàm XoaNguoiDung

- Tham số: không có
- Kết quả: trả về giá trị true/false cho biết đã xoá thành công hay không.

6.3.3 Hàm static XoaDanhSachNguoiDung

- Tham số: List<NguoiDung> chứa danh sách các người dùng cần xoá
- Kết quả: trả về giá trị true/false để xác nhận đã xoá thành công hay không.

- Hàm này sẽ gọi lại hàm XoaNguoiDung của từng đối tượng NguoiDung để thực thi.