**NHẬN XÉT CỦA GIẢNG VIÊN**

, ngày tháng năm

**LỜI NÓI ĐẦU**

Khi xã hội ngày càng phát triển và dân số đông thì nhu cầu nhà ở ngày một tăng cao. Đặc biệt với nhóm đối tượng như sinh viên mới nhập học, sinh viên vừa ra trường, vợ chồng vừa kết hôn, người mới đi làm, công nhân … thì nhu cầu nhà ở của họ là hết sức cần thiết. Nhưng kinh tế là một trở ngại lớn khi hầu như họ không đủ khả năng để mua một căn hộ bởi giá rất đắt. Thấy được tiềm năng đó, người ta đã xây dựng các khu nhà ở cho thuê (nhà trọ) với mức giá rẻ, phù hợp hơn để đáp ứng thị trường.

Thế nhưng xây dựng là một chuyện, quản lý lại là một chuyện phải cân nhắc. Khi số lượng phòng nhiều và khách ở đông, thì việc quản lý thông tin phòng trọ, khách trọ, quản lý thu chi tiền điện nước và các dịch vụ phát sinh, quản lý các trang thiết bị của mỗi phòng, và đặc biệt là thống kê, cần được thực hiện một cách nhanh chóng, linh hoạt và chặt chẽ.

Quản lý bằng sổ sách vẫn khả thi, nhưng hiệu suất không cao nếu ta phải xử lý hàng xấp giấy tờ. Do đó, cần có một phần mềm để giải quyết vấn đề trên. Giúp ta lưu trữ thông tin an toàn, lâu dài và tự động hoá tối đa các thao tác so với thực tế như tính toán, thống kê.

Với mong muốn tin học hoá cách quản lý phòng trọ để hỗ trợ tối đa cho các chủ nhà trọ, và hướng đến mục đích là hoàn thành môn lập trình .NET Windows Form, em đã chọn đề tài “**Xây dựng** **phần mềm quản lý phòng trọ**”. Đề tài được chia thành các chương sau:

* Chương I : Tổng quan
* Chương II : Khảo sát sơ bộ
* Chương III : Phân tích hệ thống
* Chương IV : Thiết kế hệ thống
* Chương V : Kết luận

**ĐỀ MỤC**

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc26468356)

[1.1. MỤC TIÊU ĐỀ TÀI 1](#_Toc26468357)

[1.2. ỨNG DỤNG MÔ HÌNH LẬP TRÌNH 2](#_Toc26468358)

[CHƯƠNG 2. KHẢO SÁT SƠ BỘ 3](#_Toc26468359)

[2.1. KHẢO SÁT HỆ THỐNG 3](#_Toc26468360)

[2.1.1. QUẢN LÝ THÔNG TIN KHÁCH TRỌ 3](#_Toc26468361)

[2.1.2. QUẢN LÝ THÔNG TIN PHÒNG 3](#_Toc26468362)

[2.1.3. QUẢN LÝ THÔNG TIN CÁC TRANG — THIẾT BỊ 4](#_Toc26468363)

[2.1.4. QUẢN LÝ THÔNG TIN ĐIỆN NƯỚC, DỊCH VỤ PHÁT SINH VÀ HOÁ ĐƠN 4](#_Toc26468364)

[2.1.5. QUẢN LÝ BÁO CÁO – THỐNG KÊ 4](#_Toc26468365)

[2.1.6. QUẢN LÝ THÔNG TIN NGƯỜI SỬ DỤNG PHẦN MỀM 4](#_Toc26468366)

[2.2. KẾT CHƯƠNG 5](#_Toc26468367)

[CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG 6](#_Toc26468368)

[3.1. NGƯỜI DÙNG — ACTORS 6](#_Toc26468369)

[3.2. CÁC CHỨC NĂNG — USECASES 6](#_Toc26468370)

[3.2.1. ĐĂNG NHẬP HỆ THỐNG 7](#_Toc26468371)

[3.2.2. QUẢN LÝ THÔNG TIN PHÒNG 8](#_Toc26468372)

[3.2.3. QUẢN LÝ THÔNG TIN KHÁCH TRỌ 8](#_Toc26468373)

[3.2.4. QUẢN LÝ THÔNG TIN TRANG THIẾT BỊ 9](#_Toc26468374)

[3.2.5. QUẢN LÝ THÔNG TIN ĐIỆN NƯỚC, DỊCH VỤ VÀ THANH TOÁN 9](#_Toc26468375)

[3.2.6. LẬP BÁO CÁO – THỐNG KÊ 10](#_Toc26468376)

[3.2.7. QUẢN LÝ THÔNG TIN NGƯỜI SỬ DỤNG 10](#_Toc26468377)

[3.3. KẾT CHƯƠNG 11](#_Toc26468378)

[CHƯƠNG 4. THIẾT KẾ HỆ THỐNG 12](#_Toc26468379)

[4.1. THIẾT KẾ CƠ SỞ DỮ LIỆU 12](#_Toc26468380)

[4.1.1. MÔ HÌNH THỰC THỂ LIÊN KẾT — *ERD* 12](#_Toc26468381)

[4.1.2. MỘT SỐ RÀNG BUỘC TOÀN VẸN — RBTV 12](#_Toc26468382)

[4.2. THIẾT KẾ PHẦN MỀM — CODING 17](#_Toc26468383)

[4.2.1. TRÍCH DẪN NGUỒN TÀI LIỆU 17](#_Toc26468384)

[4.2.2. KIẾN TRÚC PHẦN MỀM 17](#_Toc26468385)

[4.3. CÁC FORM CHỨC NĂNG CỦA CHƯƠNG TRÌNH 20](#_Toc26468386)

[4.3.1. Đăng nhập 20](#_Toc26468387)

[4.3.2. Nhóm form chức năng quản lý thông tin phòng 22](#_Toc26468388)

[4.3.3. Nhóm form chức năng quản lý thông tin thiết bị 37](#_Toc26468389)

[4.3.4. Nhóm form chức năng quản lý thông tin dịch vụ - hoá đơn 43](#_Toc26468390)

[4.3.5. Nhóm form chức năng quản lý thông tin khách vi phạm 58](#_Toc26468391)

[4.3.6. Nhóm form chức năng quản lý thông tin khách trọ 63](#_Toc26468392)

[4.3.7. Nhóm form chức năng thống kê 72](#_Toc26468393)

[4.3.8. Nhóm form chức năng quản lý thông tin tài khoản 72](#_Toc26468394)

[CHƯƠNG 5. KẾT LUẬN 74](#_Toc26468395)

# TỔNG QUAN

## MỤC TIÊU ĐỀ TÀI

Mục tiêu của đề tài là phân tích, thiết kế và xây dựng hệ thống phần mềm giúp đơn giản hoá việc quản lý thông tin phòng trọ cho chủ trọ. Cụ thể giải quyết được các vấn đề đó là:

* **Quản lý thông tin Phòng**
  + Loại phòng, số phòng, số lượng khách trọ trong phòng
  + Trạng thái phòng (đang sửa chữa, ổn định, còn chỗ, đã đầy)
  + Thông tin khu và tầng
* **Quản lý thông tin Thiết bị**
  + Danh mục thiết bị và số lượng thiết bị khi nhập mới, khi phân bổ cho các phòng và khi tháo gỡ thiết bị
* **Quản lý thông tin Khách trọ**
  + Xử lý thông tin khách xin thuê vào (hợp đồng thuê) – ra khỏi khu trọ (trả phòng)
  + Xử lý khách trọ vi phạm nội quy
* **Quản lý thông tin khách trọ tạm vắng**
* **Quản lý thông tin người thân ở tạm trú**
* **Quản lý thông tin điện nước, các dịch vụ phát sinh và thanh toán**
  + Thông tin điện nước của mỗi phòng (chỉ số cũ – mới, tiền thanh toán theo tháng)
  + Danh mục dịch vụ phát sinh
  + Lập và in phiếu thu tiền phòng, điện nước, dịch vụ hàng tháng
* **Lập báo cáo – thống kê**
  + Những phòng còn nợ tiền điện nước và dịch vụ
  + Doanh thu của khu trọ theo phòng, tháng, năm
* **Quản lý thông tin người sử dụng phần mềm**
  + Thông tin tài khoản (tạo tài khoản), mật khẩu (đổi mật khẩu), quyền hạn (phân quyền)

## ỨNG DỤNG MÔ HÌNH LẬP TRÌNH

Phần mềm được xây dựng theo kiến trúc mô hình 3 lớp (**Three Layers Model**).

Mô hình giúp cho hệ thống được phân chia rõ ràng:

* **Presentation Layer** : *Phần giao diện tương tác với người sử dụng.*
* **Business Logic Layer**: *Tương tác với lớp giao diện và xử lý các ngiệp vụ logic, tương tác với lớp* ***Data*** *để truy vấn dữ liệu.*
* **Data Access Layer** : *Tương tác với lớp Logic và giao tiếp với hệ quản trị cơ sở dữ liệu.*
* **Data Transfer Object**: *Đối tượng luân chuyển dữ liệu, mục đích là gom dữ liệu theo đối tượng để dễ quản lý và sử dụng. Đây là những lớp được ánh xạ từ các bảng trong cơ sở dữ liệu. Thực chất không thuộc kiến trúc 3 lớp.*

Khi hệ thống được tổ chức như vậy, việc lập trình sẽ trở nên dễ dàng, tường minh và linh hoạt hơn rất nhiều so với cách lập trình truyền thống.

# KHẢO SÁT SƠ BỘ

## KHẢO SÁT HỆ THỐNG

Phần mềm quản lý khu trọ gồm 8 nhóm chức năng sau:

* Quản lý thông tin phòng
* Quản lý thông tin khách trọ
* Quản lý thông tin người thân tạm trú
* Quản lý thông tin điện, nước, dịch vụ phát sinh và thanh toán
* Quản lý thông tin trang - thiết bị
* Quản lý thông tin người sử dụng phần mềm
* Lập báo cáo - thống kê

### QUẢN LÝ THÔNG TIN KHÁCH TRỌ

* Mỗi khi đăng ký thuê trọ, khách sẽ ký hợp đồng với bên cho thuê, cung cấp các thông tin cá nhân như *họ tên, ngày sinh, giới tính, quê quán, số điện thoại*… cho chủ trọ để họ lưu lại và đăng ký tạm trú với chính quyền.
* Khi ở trọ, để khu trọ được ổn định, quy củ và trật tự thì khu trọ có những hình phạt khi khách vi phạm nội quy nào đó như *nhậu nhẹt say xỉn rồi gây rối, xích mích to tiếng với hàng xóm, mở nhạc inh ỏi cả xóm làng…*
* Khách trọ khi tạm vắng có thể được chủ trọ cập nhật trạng thái.

### QUẢN LÝ THÔNG TIN PHÒNG

Các thông tin về phòng trọ cần quản lý bao gồm *thông tin về loại phòng, dãy nhà, tầng, thông tin mỗi phòng như số lượng khách tối đa và đang ở.*

### QUẢN LÝ THÔNG TIN CÁC TRANG — THIẾT BỊ

Khi nhập thêm thông tin các thiết bị mới, các thiết bị sẽ được cập nhật số lượng, và bên cạnh đó là khi tiến hành phân bổ (bàn giao – lắp đặt) đến các phòng còn thiếu cũng như thu hồi thiết bị.

Do đó cần quản lý thông tin của các thiết bị như số lượng phân bổ ở mỗi phòng và số lượng tồn trong danh mục.

### QUẢN LÝ THÔNG TIN ĐIỆN NƯỚC, DỊCH VỤ PHÁT SINH VÀ HOÁ ĐƠN

Vào ngày cuối tháng, chủ trọ hoặc người quản lý sẽ tới từng phòng để ghi nhận chỉ số điện nước tại công tơ. Sau đó dựa vào chỉ số của tháng trước và đơn giá để tính thành tiền điện nước. Giá phòng sẽ dựa vào loại phòng.

Ngoài ra, khu trọ có thể cung cấp các dịch vụ như *Wifi, vệ sinh, trông giữ xe…* Bên thuê trọ có thể yêu cầu đăng ký dịch vụ với chủ trọ hoặc thôi sử dụng. Bên cho thuê sẽ dựa vào các dịch vụ mà phòng đã đăng ký để tổng hợp hoá đơn.

Sau cùng, bên cho thuê sẽ lập các phiếu báo tiền phòng rồi gửi cho mỗi phòng và yêu cầu đóng đúng hạn quy định.

### QUẢN LÝ BÁO CÁO – THỐNG KÊ

Định kỳ hoặc ngẫu nhiên, bên cho thuê sẽ tổng hợp thông tin, tạo các báo cáo – thống kê như:

* Tình hình số lượng (*phân bổ, còn tồn*) của từng thiết bị
* Những phòng còn nợ tiền điện nước và dịch vụ

### QUẢN LÝ THÔNG TIN NGƯỜI SỬ DỤNG PHẦN MỀM

Phần mềm sẽ được sử dụng bởi chủ trọ hoặc giao công việc cho người quản lý để xử lý các thông tin. Để có thể sử dụng thì chủ trọ hoặc người quản lý sẽ có một tài khoản để đăng nhập.

## KẾT CHƯƠNG

Qua phần khảo sát sơ bộ hệ thống (giả định – thông qua việc tham khảo các tài liệu), có thể rút ra những yêu cầu như trên. Để có một cái nhìn chi tiết hơn về các chức năng quản lý này, ta sẽ đi vào phần phân tích chuyên sâu ở chương tiếp theo.

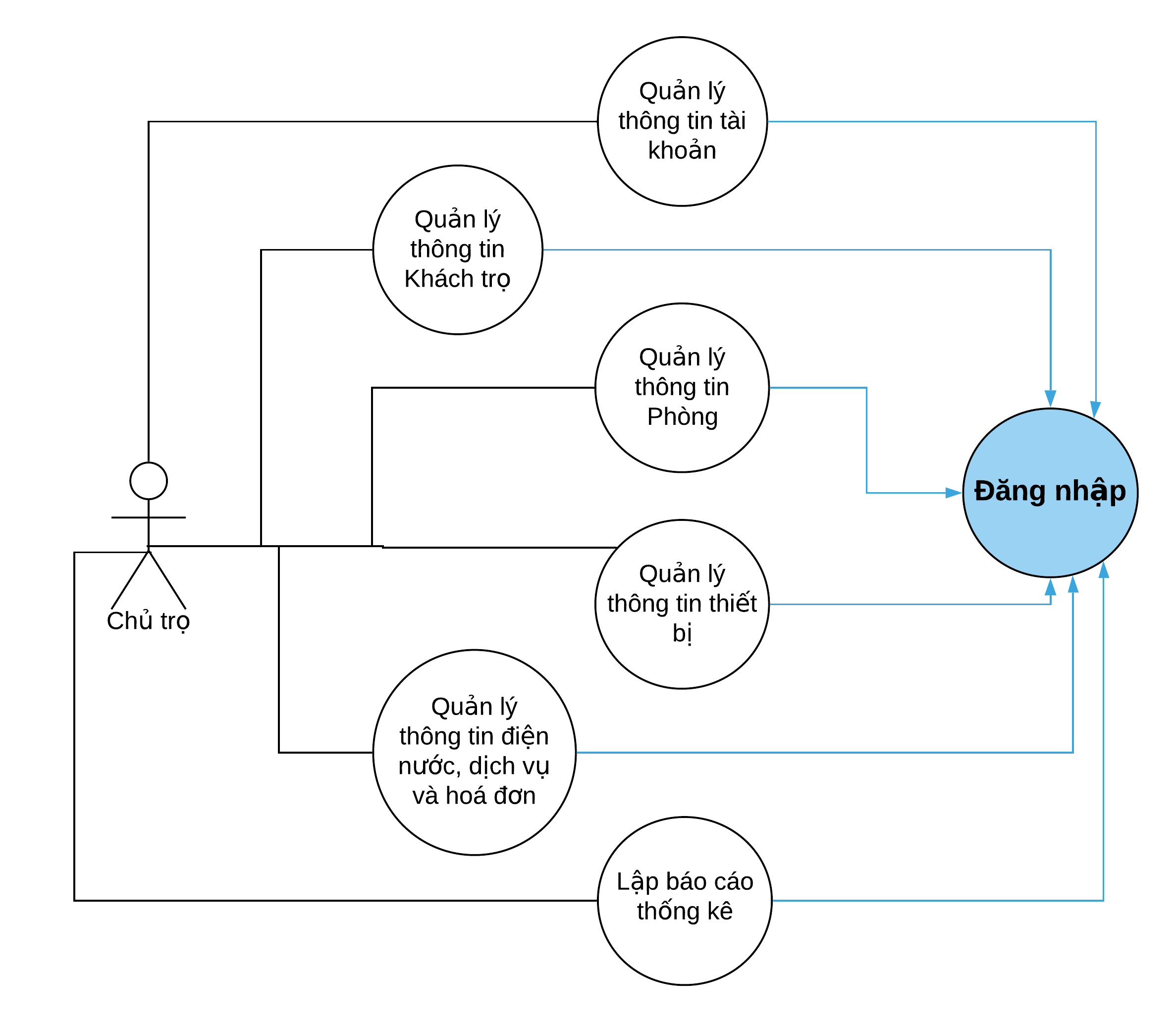
# PHÂN TÍCH HỆ THỐNG

Do giới hạn về thời gian và khả năng, kể từ chương phân tích hệ thống cho đến hết báo cáo thì một số nội dung sẽ được lược bỏ và tiếp tục phát triển sau này.

## NGƯỜI DÙNG — ACTORS

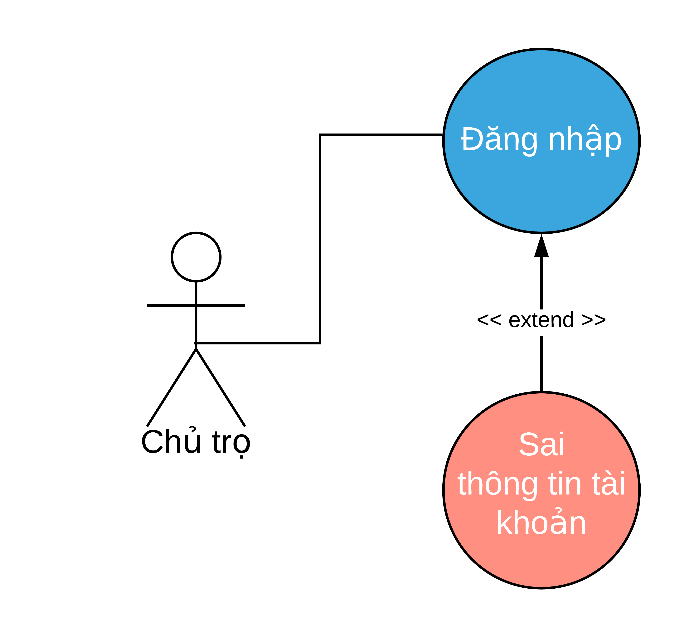
Qua việc khảo sát, ta thấy rằng hệ thống có một đối tượng chính trực tiếp sử dụng phần mềm là **Chủ trọ**.

## CÁC CHỨC NĂNG — USECASES

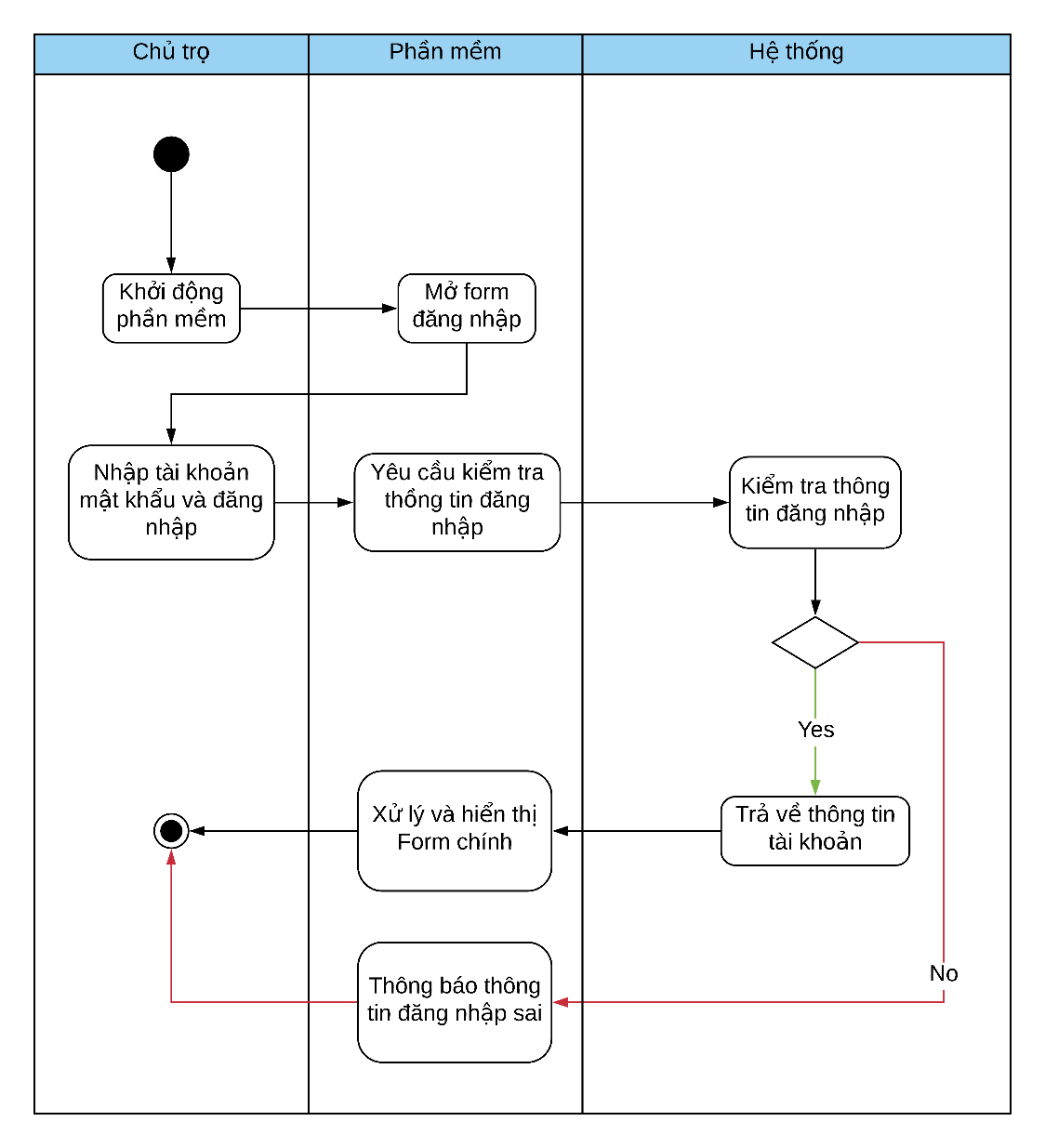


**Hình 1. Biểu đồ UseCase tổng quát**

### ĐĂNG NHẬP HỆ THỐNG

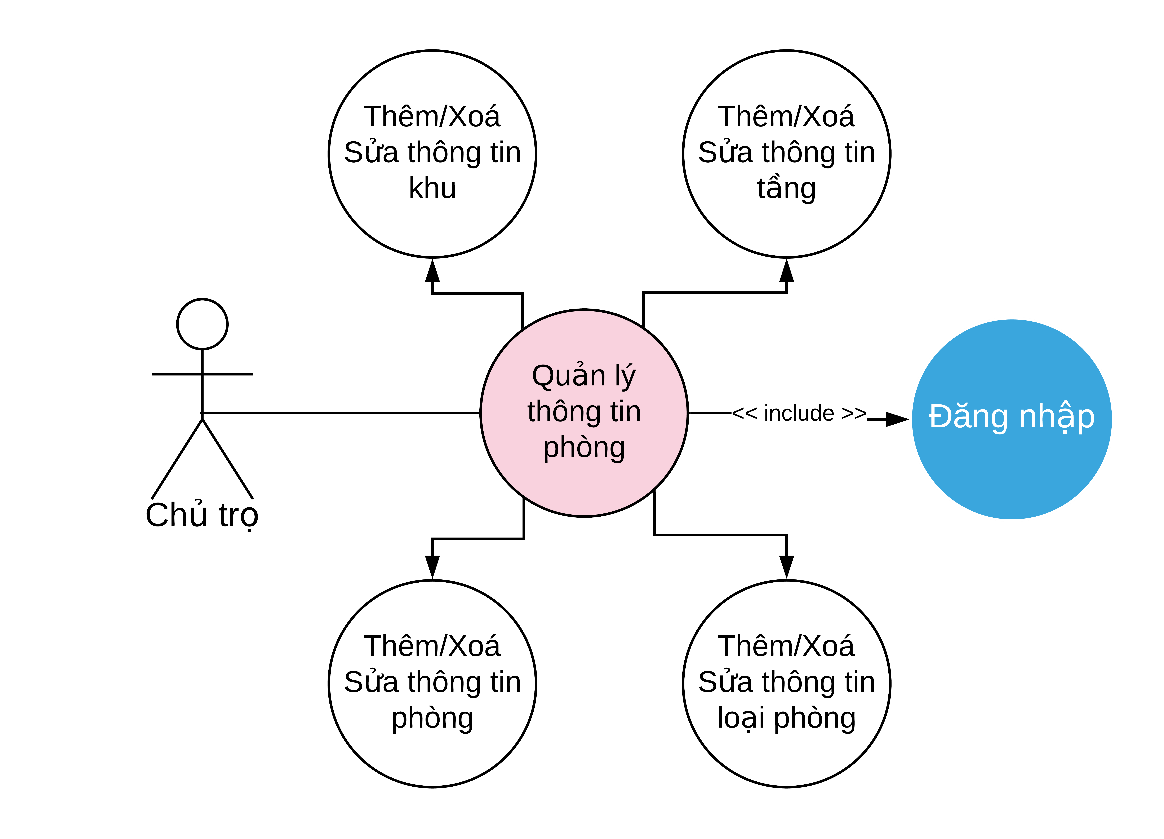


**Hình 2. Biểu đồ UseCase chức năng Đăng nhập**



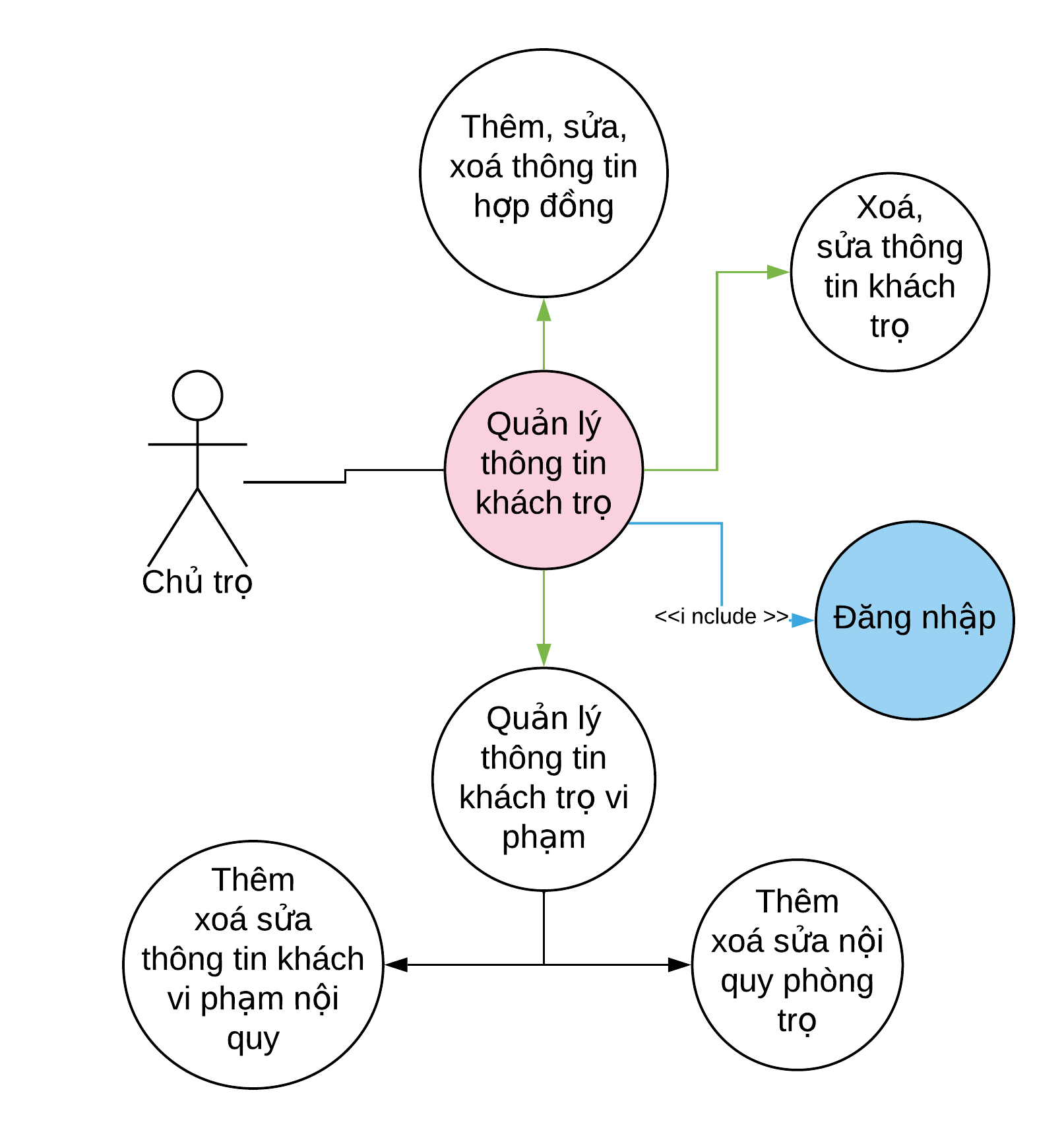
**Hình 3. Biểu đồ Hoạt động chức năng Đăng Nhập hệ thống**

### QUẢN LÝ THÔNG TIN PHÒNG



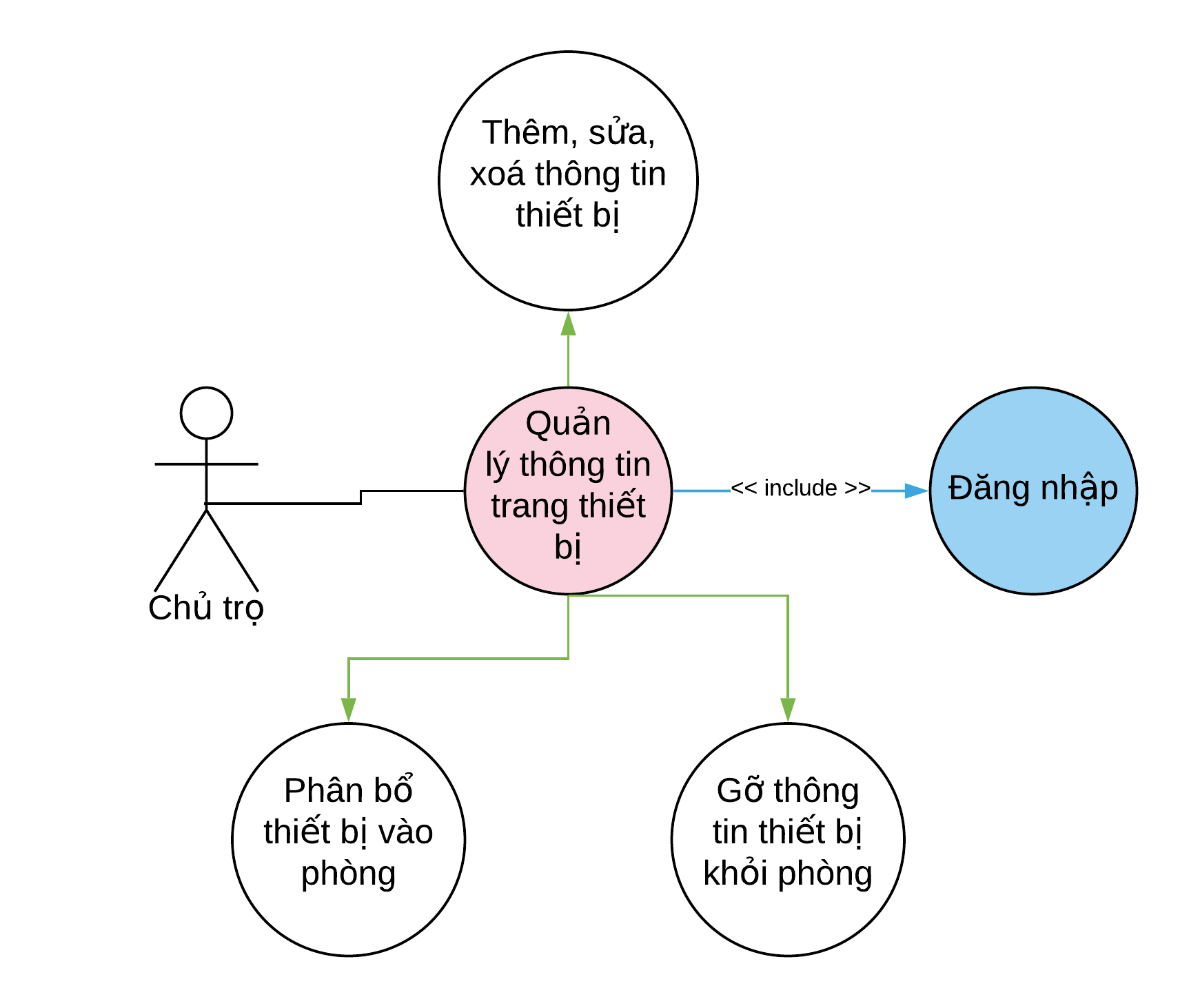
**Hình 4. Biểu đồ UseCase chức năng quản lý thông tin Phòng**

### QUẢN LÝ THÔNG TIN KHÁCH TRỌ



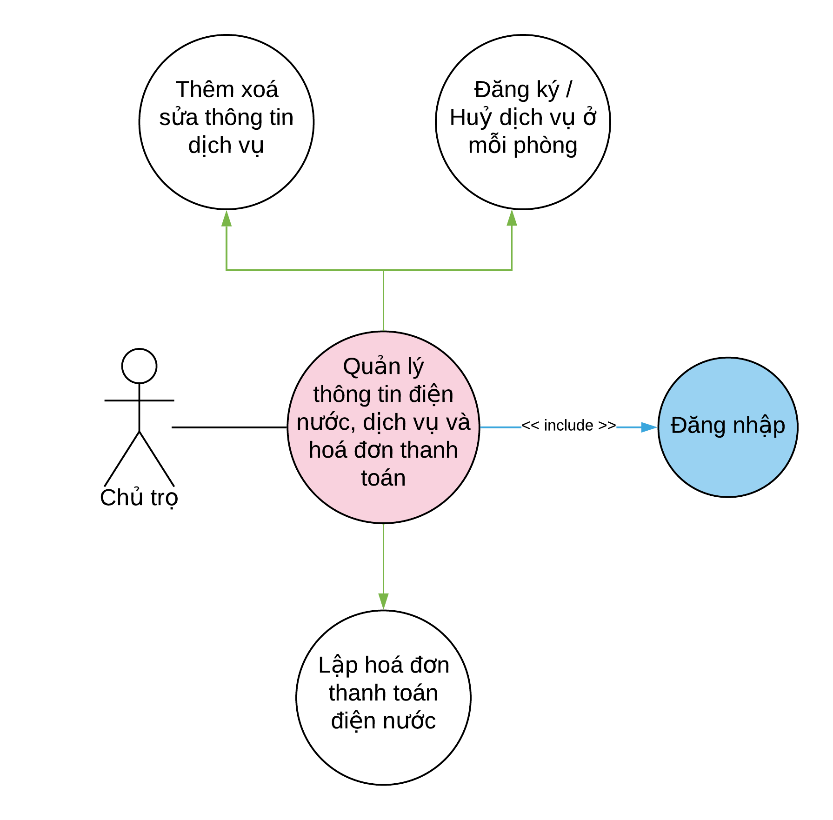
**Hình 5. Biểu đồ UseCase chức năng Quản lý thông tin Khách trọ**

### QUẢN LÝ THÔNG TIN TRANG THIẾT BỊ



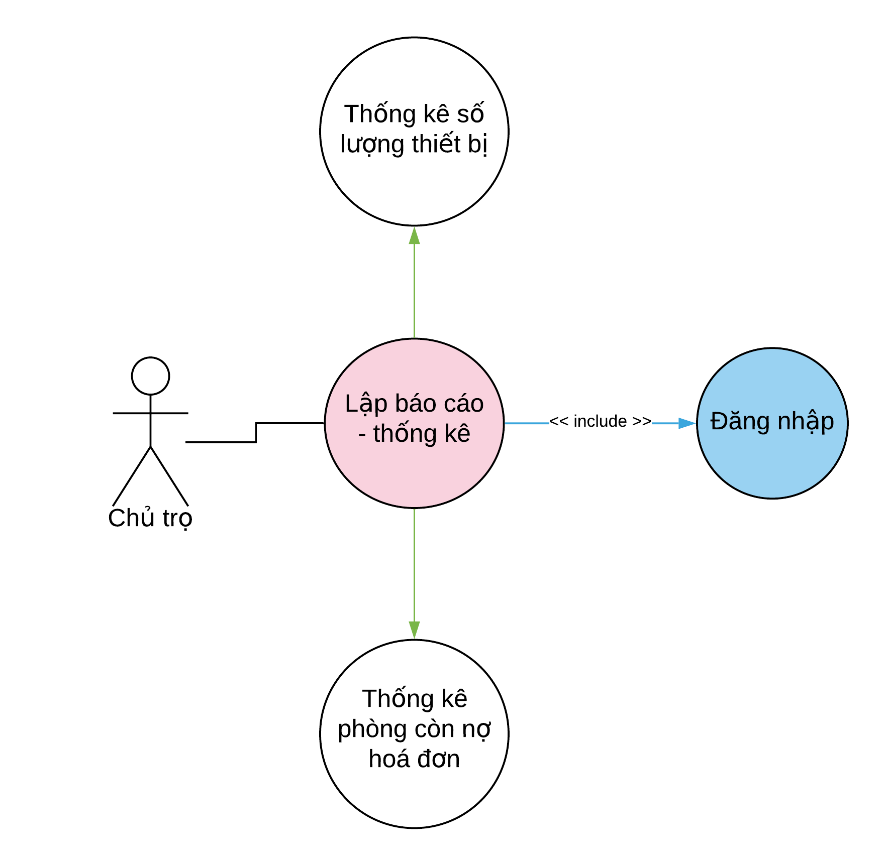
**Hình 6. Biểu đồ UseCase chức năng quản lý thông tin thiết bị**

### QUẢN LÝ THÔNG TIN ĐIỆN NƯỚC, DỊCH VỤ VÀ THANH TOÁN

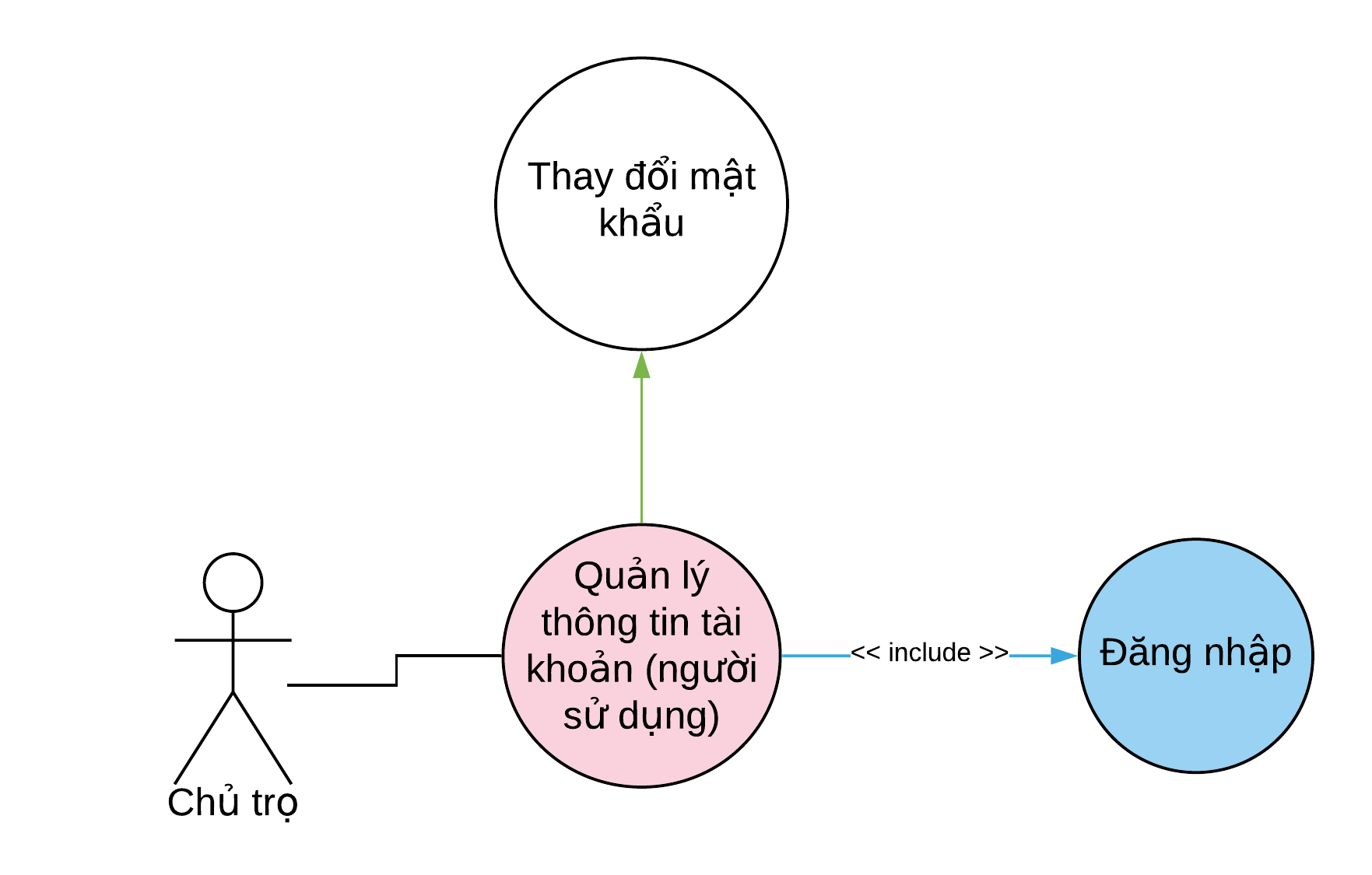


**Hình 7. Biểu đồ UseCase chức năng Quản lý thông tin điện nước, dịch vụ và Thanh toán**

### LẬP BÁO CÁO – THỐNG KÊ



### QUẢN LÝ THÔNG TIN NGƯỜI SỬ DỤNG



**Biểu đồ UseCase chức năng Lập báo cáo - thống kê**

## KẾT CHƯƠNG

Dựa vào việc khảo sát sơ bộ ở chương 2, ta đã tiến hành phân tích hệ thống bằng ngôn ngữ mô hình hoá UML (Unified Modeling Language). Những thông tin ở chương 3 này sẽ được áp dụng vào chương tiếp theo ngay sau đây.

# THIẾT KẾ HỆ THỐNG

## THIẾT KẾ CƠ SỞ DỮ LIỆU

Sử dụng ngôn ngữ SQL (*Structured Query Languge – Ngôn ngữ truy vấn có cấu trúc*), ta tiến hành thiết kế cơ sở dữ liệu như sau đây.

### MÔ HÌNH THỰC THỂ LIÊN KẾT — *ERD*

### MỘT SỐ RÀNG BUỘC TOÀN VẸN — RBTV

#### Bảng “Loại phòng”

* R1: Mỗi loại phòng có một mã riêng
  + Bối cảnh: Loại phòng

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | THÊM | XOÁ | SỬA |
| Loại phòng | +mã loại phòng | - | -(\*) |

* R2: Mỗi loại phòng có một tên riêng (VIP, Normal…)

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | THÊM | XOÁ | SỬA |
| Loại phòng | +tên loại phòng | - | +tên loại phòng |

* + Bối cảnh: Loại phòng

#### Bảng “Khu”

* R3: Mỗi khu có một mã riêng
  + Bối cảnh: Khu

|  |  |  |  |
| --- | --- | --- | --- |
| R3 | THÊM | XOÁ | SỬA |
| Khu | +mã khu | - | -(\*) |

* R4: Mỗi khu có một tên riêng (khu A, khu B…)

|  |  |  |  |
| --- | --- | --- | --- |
| R4 | THÊM | XOÁ | SỬA |
| Khu | +tên khu | - | +tên khu |

* + Bối cảnh: Khu

#### Bảng “Tầng”

* R5: Mỗi tầng có một mã riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R5 | THÊM | XOÁ | SỬA |
| Tầng | +mã tầng | - | -(\*) |

* + Bối cảnh: Tầng
* R6: Mỗi tầng thuộc vào khu nào đó
  + Bối cảnh: Quan hệ tầng - Khu

|  |  |  |  |
| --- | --- | --- | --- |
| R6 | THÊM | XOÁ | SỬA |
| Tầng | +mã khu | - | +mã khu |
| Khu | - | +mã khu | - (\*) |

#### Bảng “Phòng”

* R7: Mỗi phòng có một mã phòng riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R7 | THÊM | XOÁ | SỬA |
| Phòng | +mã phòng | - | - (\*) |

* + Bối cảnh: Phòng
* R8: Mỗi phòng có một tên phòng riêng (A100, A101…)

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | THÊM | XOÁ | SỬA |
| Phòng | +tên phòng | - | +tên phòng |

* + Bối cảnh: Phòng
* R9: Mỗi phòng phải thuộc một tầng nào đó

|  |  |  |  |
| --- | --- | --- | --- |
| R9 | THÊM | XOÁ | SỬA |
| Phòng | +mã tầng | - | +mã tầng |
| Tầng | - | +mã tầng | - (\*) |

* + Bối cảnh: Phòng
* R10: Mỗi phòng thuộc một loại phòng nào đó

|  |  |  |  |
| --- | --- | --- | --- |
| R10 | THÊM | XOÁ | SỬA |
| Phòng | +mã loại phòng | - | +mã loại phòng |
| Loại phòng | - | +mã loại phòng | - (\*) |

* + Bối cảnh: Phòng
* R11: Số lượng đang ở không vượt quá số lượng tối đa

|  |  |  |  |
| --- | --- | --- | --- |
| R11 | THÊM | XOÁ | SỬA |
| Phòng | +số lượng đang ở | - | +số lượng đang ở |

* + Bối cảnh: Phòng

#### Bảng “Hoá đơn”

* R12: Mỗi hoá đơn có một mã riêng biệt

|  |  |  |  |
| --- | --- | --- | --- |
| R12 | THÊM | XOÁ | SỬA |
| Hoá đơn | +mã hoá đơn | - | - (\*) |

* + Bối cảnh: Hoá đơn
* R13: Mỗi phòng chỉ có một hoá đơn vào mỗi tháng của năm

|  |  |  |  |
| --- | --- | --- | --- |
| R13 | THÊM | XOÁ | SỬA |
| Hoá đơn | +mã phòng, tháng, năm | - | + mã phòng, tháng, năm |

* + Bối cảnh: Hoá đơn
* R14: Mỗi phòng có duy nhất một thông tin hoá đơn về số điện và số nước của tháng đó
  + Bối cảnh: Hoá đơn điện nước

|  |  |  |  |
| --- | --- | --- | --- |
| R14 | THÊM | XOÁ | SỬA |
| Hoá đơn điện nước | +mã hoá đơn | - | - (\*) |

#### Bảng “Thiết bị”

* R15: Mỗi loại thiết bị có một mã riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R15 | THÊM | XOÁ | SỬA |
| Thiết bị | +mã thiết bị | - | - (\*) |

* + Bối cảnh: Thiết bị
* R16: Mỗi loại thiết bị có tên khác nhau

|  |  |  |  |
| --- | --- | --- | --- |
| R16 | THÊM | XOÁ | SỬA |
| Thiết bị | +tên thiết bị | - | + tên thiết bị |

* + Bối cảnh: Thiết bị

#### Bảng “Dịch vụ”

* R17: Mỗi dịch vụ có mã riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R17 | THÊM | XOÁ | SỬA |
| Dịch vụ | +mã dịch vụ | - | - (\*) |

* + Bối cảnh: Dịch vụ
* R18: Mỗi dịch vụ có tên riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R18 | THÊM | XOÁ | SỬA |
| Dịch vụ | +tên dịch vụ | - | +tên dịch vụ |

* + Bối cảnh: Dịch vụ

#### Bảng “Hợp đồng”

* R19: Mỗi hợp đồng có mã riêng
  + Bối cảnh: Hợp đồng

|  |  |  |  |
| --- | --- | --- | --- |
| R19 | THÊM | XOÁ | SỬA |
| Hợp đồng | +mã hợp đồng | - | - (\*) |

* R20: Mỗi hợp đồng thuộc phòng nào đó

|  |  |  |  |
| --- | --- | --- | --- |
| R20 | THÊM | XOÁ | SỬA |
| Hợp đồng | +mã hợp đồng | - | - (\*) |
| Phòng | - | +mã phòng | - (\*) |

* + Bối cảnh: Quan hệ Phòng – Hợp đồng

#### Bảng “Khách”

* R21: Mỗi khách có một mã riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R21 | THÊM | XOÁ | SỬA |
| Khách | +mã khách | - | - (\*) |

* + Bối cảnh: Khách
* R22: Mỗi khách có một số căn cước khác nhau

|  |  |  |  |
| --- | --- | --- | --- |
| R22 | THÊM | XOÁ | SỬA |
| Khách | +số căn cước | - | +số căn cước |

* + Bối cảnh: Khách

#### Bảng “Vi phạm”

* R23: Mỗi loại vi phạm có một mã riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R23 | THÊM | XOÁ | SỬA |
| Vi phạm | +mã loại vi phạm | - | - (\*) |

* + Bối cảnh: Vi phạm
* R24: Mỗi vi phạm có một tên (nội dung) riêng

|  |  |  |  |
| --- | --- | --- | --- |
| R24 | THÊM | XOÁ | SỬA |
| Vi phạm | +nội dung vi phạm | - | +nội dung vi phạm |

* + Bối cảnh: Vi phạm

#### Bảng “Khách – Vi phạm”

* R25: Mỗi khách có thể vi phạm nhiều nội quy, và có thể mỗi nội quy lại vi phạm nhiều lần

|  |  |  |  |
| --- | --- | --- | --- |
| R25 | THÊM | XOÁ | SỬA |
| Khách - Vi phạm | +mã vi phạm, mã khách, thời gian | - | - (\*) |

* + Bối cảnh: Quan hệ khách – vi phạm

## THIẾT KẾ PHẦN MỀM — CODING

### TRÍCH DẪN NGUỒN TÀI LIỆU

Chương trình sử dụng các thư viện đồ hoạ gồm:

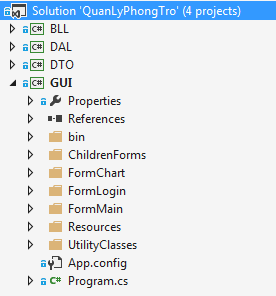
* [**DevExpress**](https://www.devexpress.com/products/net/controls/winforms/)
* [**Bunifu**](https://bunifuframework.com/products/bunifu-ui-winforms/)
* [**Material Skin**](https://github.com/IgnaceMaes/MaterialSkin)
* [**Metro UI**](https://github.com/dennismagno/metroframework-modern-ui)

Cách trình bày codes trong báo cáo này:

1. Copy codes của bạn vào clipboard
2. Vào trang sau: <http://www.planetb.ca/syntax-highlight-word>
3. Dán codes vào ô 🡪 chọn ngôn ngữ **C#** 🡪 **“Show Highlighted”**
4. Trình duyệt web sẽ mở trang mới chứa codes đã có định dạng, chỉ việc copy rồi dán vào báo cáo

### KIẾN TRÚC PHẦN MỀM

Được thiết kế theo mô hình 3 lớp, phần mềm được ráp bởi 4 projects như sau:



Để thuận tiện cho việc giao tiếp với hệ quản trị cơ sở dữ liệu, em đã viết một Class tên ***DataProvider.cs*** nhằm cung cấp các hàm tiện ích, giúp tối giản trong việc code.

* Đầu tiên, khởi tạo chuỗi kết nối và SqlConnection:

1. **private** **static** **string** myConnStr = @"Data Source=.\sqlexpress;
2. Initial Catalog=OCEAN\_HOSTEL;
3. Integrated Security=True";
5. **private** **static** SqlConnection connection = **new** SqlConnection(myConnStr);

* Hàm trả về các **SqlParameter** tự động:

1. **private** **static** SqlParameter[] GetParamsList(**string** query, **object**[] parameters)
2. {
3. List<SqlParameter> result = **new** List<SqlParameter>();
4. List<**string**> segments = query.Split(' ').ToList<**string**>();
6. **int** countParams = 0;
7. **foreach** (**string** segment **in** segments)
8. **if** (segment.Contains("@@")) //  Đây là một parameter trong chuỗi query
9. {
10. SqlParameter sp = **new** SqlParameter
11. {
12. ParameterName = segment,
13. Value = parameters[countParams++]
14. };
16. result.Add(sp);
17. }
19. **return** result.ToArray();
20. }

* Hàm ***ExecuteQuery()*** để chạy câu truy vấn trả về bảng dữ liệu:

1. **protected** **static** DataTable ExecuteQuery(**string** query, **object**[] parameters = **null**)
2. {
3. SqlCommand command = **new** SqlCommand(query, connection);
5. **if** (parameters != **null**)
6. command.Parameters.AddRange(GetParamsList(query, parameters));
8. SqlDataAdapter adapter = **new** SqlDataAdapter(command);
9. DataTable dt = **new** DataTable();
10. adapter.Fill(dt);
12. **return** dt;
13. }

* Hàm ***ExecuteNonQuery()*** để chạy câu truy vấn ***Insert, Delete, Update***:

1. **protected** **static** **int** ExecuteNonQuery(**string** query, **object**[] parameters = **null**)
2. {
3. **if** (connection.State == ConnectionState.Closed)
4. connection.Open();
6. SqlCommand command = **new** SqlCommand(query, connection);

9. **if** (parameters != **null**)
10. command.Parameters.AddRange(GetParamsList(query, parameters));
12. **int** result = command.ExecuteNonQuery();

15. **if** (connection.State == ConnectionState.Open)
16. connection.Close();
18. **return** result;
19. }

* Hàm ***ExecuteScalar()***  để trả về kết quả của câu truy vấn là giá trị đơn:

1. **protected** **static** **object** ExecuteScalar(**string** query, **object**[] parameters = **null**)
2. {
3. **if** (connection.State == ConnectionState.Closed)
4. connection.Open();
6. SqlCommand command = **new** SqlCommand(query, connection);

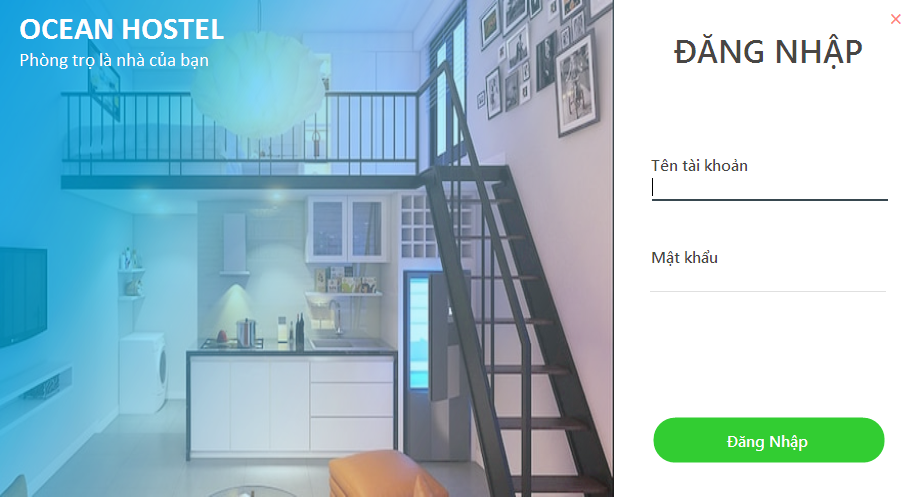
9. **if** (parameters != **null**)
10. command.Parameters.AddRange(GetParamsList(query, parameters));
12. **object** result = command.ExecuteScalar();

15. **if** (connection.State == ConnectionState.Open)
16. connection.Close();
18. **return** result;
19. }

Những khi cần thao tác với CSDL, chỉ việc gọi các hàm **ExecuteQuery(), ExecuteNonQuery(), ExecuteScalar()** tương ứng với mục đích muốn sử dụng. Truyền vào chuỗi truy vấn và các parameter nếu có.

## CÁC FORM CHỨC NĂNG CỦA CHƯƠNG TRÌNH

### Đăng nhập



**Hình 8. Form đăng nhập**

Các code xử lý chính:

1. **private** **bool** CheckBeforeLogin()     //  Kiểm tra TextBox trước khi truy vấn
2. {
3. **bool** state = **true**;

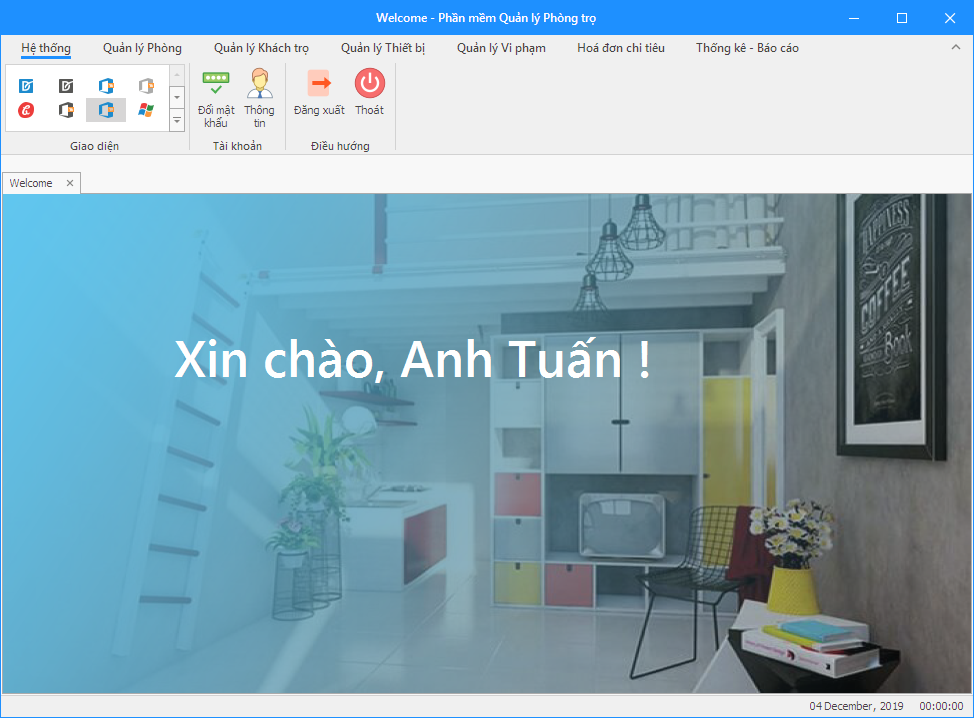
6. **if** (String.IsNullOrWhiteSpace(**this**.txt\_tenTaiKhoan.Text))
7. {
8. state = **false**;
10. XtraMessageBox.Show("Chưa nhập tên tài khoản !", "Thông báo",
11. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
13. **this**.txt\_tenTaiKhoan.Focus();
14. }
15. **else**
16. {
17. **if** (String.IsNullOrWhiteSpace(**this**.txt\_matKhau.Text))
18. {
19. state = **false**;
21. XtraMessageBox.Show("Chưa nhập mật khẩu !", "Thông báo",
22. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
24. **this**.txt\_matKhau.Focus();
25. }
26. }
28. **return** state;
29. }

Sự kiện nút đăng nhập Click:

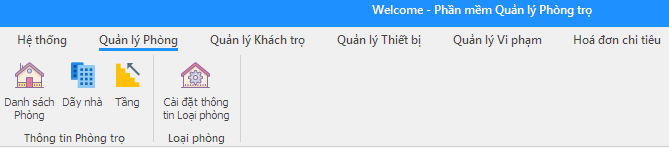
1. **private** **void** btn\_dangNhap\_Click(**object** sender, EventArgs e)
2. {
3. **if** (CheckBeforeLogin() == **false**)
4. **return**;

7. DTO\_NGUOIDUNG nguoiDung = **new** DTO\_NGUOIDUNG
8. {
9. TenDangNhap = txt\_tenTaiKhoan.Text,
10. MatKhau = txt\_matKhau.Text
11. };

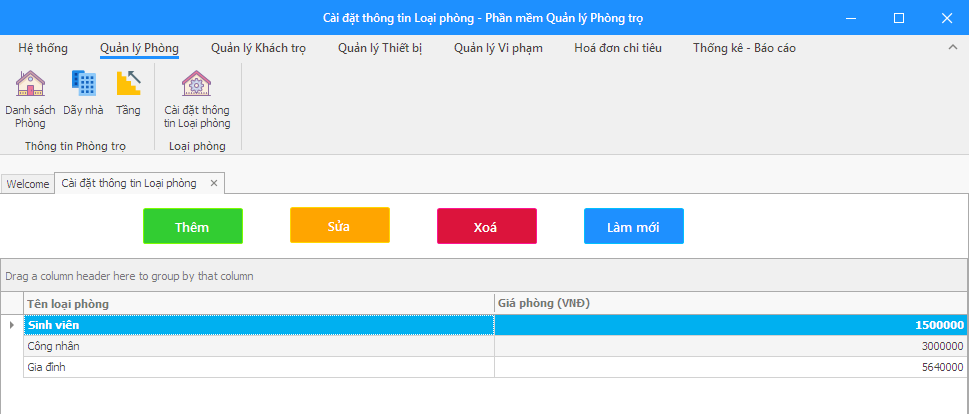
14. **try**
15. {
16. DataTable dt = BLL\_NGUOIDUNG.GetAccountInfo\_By\_Credentials(nguoiDung);
18. **if** (dt.Rows.Count > 0)
19. {
20. nguoiDung.Ho = dt.Rows[0]["HO"].ToString();
21. nguoiDung.Ten = dt.Rows[0]["TEN"].ToString();
23. fMain FormMain = **new** fMain(nguoiDung);
24. **this**.Hide();
25. FormMain.ShowDialog();
26. **this**.Close();
27. }
28. **else**
29. {
30. XtraMessageBox.Show("Thông tin đăng nhập không chính xác, vui lòng kiểm tra lại !",
31. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);
33. **this**.txt\_matKhau.Focus();
34. **this**.txt\_matKhau.SelectAll();
35. }
36. }
37. **catch**(Exception ex)
38. {
39. XtraMessageBox.Show(ex.Message,
40. "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
41. }
42. }

Sau khi đăng nhập thành công, giao diện form chính như sau:

### Nhóm form chức năng quản lý thông tin phòng



#### Form cài đặt thông tin loại phòng

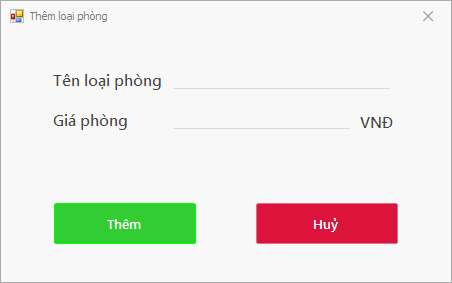


Code chính:

1. **public** **static** **bool** isAdded = **false**; //  Xác định liệu đã thêm hay chưa
2. **public** **static** **bool** isUpdated = **false**;
4. **private** **void** fLoaiPhong\_Load(**object** sender, EventArgs e)
5. {
6. LoadData();
7. }
9. **private** **void** LoadData()
10. {
11. **this**.gridControl\_loaiPhong.DataSource = BLL\_LOAIPHONG.LayDanhSachLoaiPhong();
13. **this**.gridView\_loaiPhong.Columns["TENLOAIPHG"].Caption = "Tên loại phòng";
14. **this**.gridView\_loaiPhong.Columns["GIAPHG"].Caption = "Giá phòng (VNĐ)";
15. **this**.gridView\_loaiPhong.Columns["MALOAIPHG"].Visible = **false**;
16. }
18. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
19. {
20. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.LoaiPhong.fAlter();
21. FormAlter.ShowDialog();
23. **if** (isAdded == **true**)
24. {
25. LoadData();
26. isAdded = **false**;
27. }
28. }
30. **private** **void** btn\_sua\_Click(**object** sender, EventArgs e)
31. {
32. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.LoaiPhong.fAlter(LayLoaiPhongHienTai());
33. FormAlter.ShowDialog();
35. **if** (isUpdated == **true**)
36. {
37. LoadData();
38. isUpdated = **false**;
39. }
40. }
42. **private** **void** btn\_lamMoi\_Click(**object** sender, EventArgs e)
43. {
44. LoadData();
45. }

48. **private** DTO\_LOAIPHONG LayLoaiPhongHienTai()
49. {
50. DataRow dr = **this**.gridView\_loaiPhong.GetFocusedDataRow();
52. DTO\_LOAIPHONG loaiPhong = **new** DTO\_LOAIPHONG
53. {
54. MaLoaiPhong = dr["MALOAIPHG"].ToString(),
55. TenLoaiPhong = dr["TENLOAIPHG"].ToString(),
56. GiaPhong = (**int**)dr["GIAPHG"]
57. };
59. **return** loaiPhong;
60. }
62. **private** **void** btn\_xoa\_Click(**object** sender, EventArgs e)
63. {
64. DTO\_LOAIPHONG loaiPhongHienTai = LayLoaiPhongHienTai();
66. **try**
67. {
68. **if** (BLL\_LOAIPHONG.LoaiPhongDuocThamChieu(loaiPhongHienTai) == **true**)
69. {
70. XtraMessageBox.Show("Thông tin loại phòng đang được sử dụng, không thể xoá !",
71. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
72. }
73. **else**
74. {
75. DialogResult dr = XtraMessageBox.Show(String.Format("Xác nhận xoá thông tin loại phòng: {0} ?", loaiPhongHienTai.TenLoaiPhong),
76. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
78. **if** (dr == DialogResult.Yes)
79. **if** (BLL\_LOAIPHONG.XoaLoaiPhong(loaiPhongHienTai) == **true**)
80. {
81. XtraMessageBox.Show("Xoá loại phòng thành công !",
82. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
84. LoadData();
85. }
86. }
87. }
88. **catch** (Exception ex)
89. {
90. XtraMessageBox.Show("Lỗi truy vấn !" + "\n\nNội dung lỗi: \n" + ex.Message,
91. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
92. }
93. }

Giao diện form chỉnh sửa / thêm thông tin loại phòng (**fAlter**):

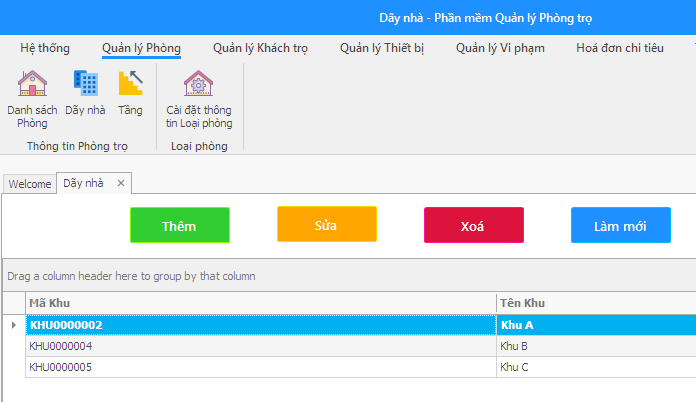


1. **private** DTO\_LOAIPHONG loaiPhong;
2. **private** **bool** isAdd = **true**;
4. **public** fAlter(DTO\_LOAIPHONG loaiPhong = **null**)
5. {
6. InitializeComponent();
8. **if** (loaiPhong != **null**)
9. {
10. isAdd = **false**;
11. **this**.loaiPhong = loaiPhong;
12. **this**.btn\_them.Text = "Lưu";
13. **this**.Text = "Sửa thông tin Loại phòng";
15. **this**.txt\_tenLoaiPhong.Text = loaiPhong.TenLoaiPhong;
16. **this**.txt\_giaPhong.Text = loaiPhong.GiaPhong.ToString();
17. }
18. }
20. **private** **void** btn\_huy\_Click(**object** sender, EventArgs e)
21. {
22. **this**.Close();
23. }
25. **private** **bool** CheckBeforeSave()
26. {
27. **bool** state = **true**;
29. **if** (String.IsNullOrWhiteSpace(**this**.txt\_tenLoaiPhong.Text))
30. {
31. XtraMessageBox.Show("Chưa nhập tên loại phòng !", "Thông báo",
32. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
34. **this**.txt\_tenLoaiPhong.Focus();
35. state = **false**;
36. }
37. **else**
38. **if** (String.IsNullOrWhiteSpace(**this**.txt\_giaPhong.Text))
39. {
40. XtraMessageBox.Show("Chưa nhập giá phòng !", "Thông báo",
41. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
43. **this**.txt\_giaPhong.Focus();
44. state = **false**;
45. }
47. **return** state;
48. }
50. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
51. {
52. **if** (CheckBeforeSave() == **false**)     //  Kiểm tra rỗng
53. **return**;

56. **try**
57. {
58. **if** (isAdd == **true**)     //  Thêm
59. {
60. **this**.loaiPhong = **new** DTO\_LOAIPHONG
61. {
62. TenLoaiPhong = **this**.txt\_tenLoaiPhong.Text,
63. GiaPhong = **int**.Parse(**this**.txt\_giaPhong.Text)
64. };
66. **if** (BLL\_LOAIPHONG.ThemLoaiPhong(loaiPhong) == **true**)
67. {
68. XtraMessageBox.Show("Thêm loại phòng thành công !", "Thông báo",
69. MessageBoxButtons.OK, MessageBoxIcon.Information);
71. ChildrenForms.QuanLyPhong.LoaiPhong.fLoaiPhong.isAdded = **true**;
72. **this**.Close();
73. }
74. }
75. **else**
76. {
77. **this**.loaiPhong.TenLoaiPhong = **this**.txt\_tenLoaiPhong.Text;
78. **this**.loaiPhong.GiaPhong = **int**.Parse(**this**.txt\_giaPhong.Text);

81. **if** (BLL\_LOAIPHONG.SuaLoaiPhong(loaiPhong) == **true**)
82. {
83. XtraMessageBox.Show("Sửa thông tin loại phòng thành công !", "Thông báo",
84. MessageBoxButtons.OK, MessageBoxIcon.Information);
86. ChildrenForms.QuanLyPhong.LoaiPhong.fLoaiPhong.isUpdated = **true**;
87. **this**.Close();
88. }
89. }
90. }
91. **catch**(Exception ex)
92. {
93. **if** (ex.Message.Contains("duplicate"))   //  Trùng tên loại phòng
94. {
95. XtraMessageBox.Show("Tên loại phòng đã tồn tại !", "Thông báo",
96. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
98. **this**.txt\_tenLoaiPhong.Focus();
99. **this**.txt\_tenLoaiPhong.SelectAll();
100. }
101. **else**
102. XtraMessageBox.Show("Lỗi truy vấn !" + "\n\nNội dung lỗi: \n" + ex.Message,
103. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
104. }
105. }

#### Form quản lý thông tin khu (dãy nhà)



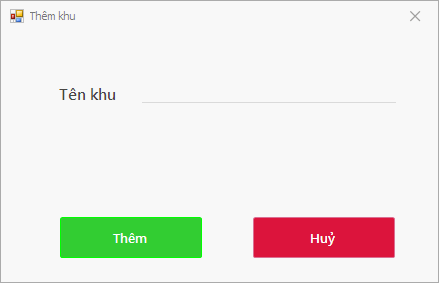
Codes chính:

1. **public** **static** **bool** isAdded = **false**; //  Xác định liệu đã thêm hay chưa
2. **public** **static** **bool** isUpdated = **false**;
4. **private** **void** LoadData()
5. {
6. **this**.gridControl\_khu.DataSource = BLL\_KHU.LayDanhSachKhu();
7. **this**.gridView\_khu.Columns["MAKHU"].Caption = "Mã Khu";
8. **this**.gridView\_khu.Columns["TENKHU"].Caption = "Tên Khu";
9. }
11. **private** **void** fDayNha\_Load(**object** sender, EventArgs e)
12. {
13. LoadData();
14. }

17. **private** DTO\_KHU LayThongTinKhuDangChon()
18. {
19. DataRow dr = gridView\_khu.GetFocusedDataRow();
21. DTO\_KHU khu = **new** DTO\_KHU
22. {
23. TenKhu = dr["TENKHU"].ToString(),
24. MaKhu = dr["MAKHU"].ToString()
25. };
27. **return** khu;
28. }
30. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
31. {
32. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.DayNha.fAlter();
33. FormAlter.ShowDialog();
35. **if** (isAdded == **true**)
36. {
37. LoadData();
38. isAdded = **false**;
39. }
40. }
42. **private** **void** btn\_sua\_Click(**object** sender, EventArgs e)
43. {
44. DTO\_KHU khuHienTai = LayThongTinKhuDangChon();
46. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.DayNha.fAlter(khuHienTai);
47. FormAlter.ShowDialog();

50. **if** (isUpdated == **true**)
51. {
52. LoadData();
53. isUpdated = **false**;
54. }
55. }
57. **private** **void** btn\_lamMoi\_Click(**object** sender, EventArgs e)
58. {
59. LoadData();
60. }
62. **private** **void** btn\_xoa\_Click(**object** sender, EventArgs e)
63. {
64. DTO\_KHU khuHienTai = LayThongTinKhuDangChon();
66. **try**
67. {
68. **if** (BLL\_KHU.KhuDaDuocThamChieu(khuHienTai) == **true**)
69. {
70. XtraMessageBox.Show("Khu đang chứa các tầng, không thể xoá !",
71. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);
72. }
73. **else**
74. {
75. DialogResult dr = XtraMessageBox.Show(String.Format("Xác nhận xoá thông tin khu: {0}?", khuHienTai.TenKhu),
76. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
78. **if** (dr == DialogResult.Yes)
79. **if** (BLL\_KHU.XoaKhu(khuHienTai) == **true**)
80. {
81. XtraMessageBox.Show("Xoá khu thành công !",
82. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
84. LoadData();
85. }
86. }
87. }
88. **catch** (Exception ex)
89. {
90. XtraMessageBox.Show("Lỗi khi xoá khu !\n\n" + "Nội dung lỗi:\n" + ex.Message,
91. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
92. }
93. }

Giao diện form chỉnh sửa / thêm thông tin dãy nhà:

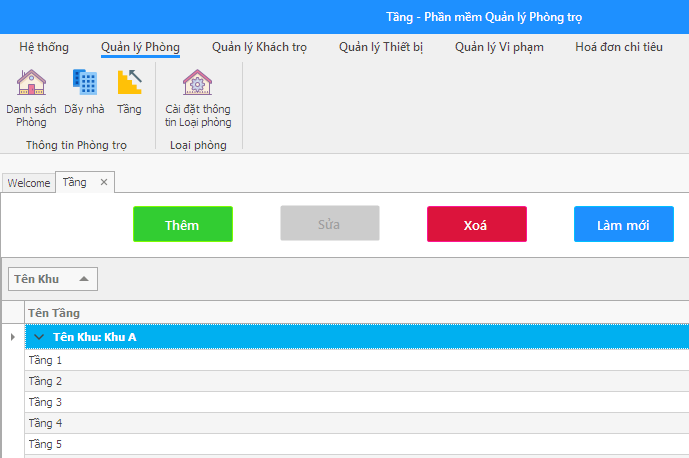


Codes chính:

1. **private** **bool** isAddNew = **true**;
2. **private** DTO\_KHU khu;
4. **public** fAlter(DTO\_KHU khu = **null**)
5. {
6. InitializeComponent();

9. **if** (khu != **null**)    //  Truyền vào thông tin khu từ form cha ==> Là Update
10. {
11. isAddNew = **false**;
13. **this**.khu = khu;
14. **this**.txt\_tenKhu.Text = khu.TenKhu;
15. **this**.btn\_them.Text = "Lưu";
16. **this**.Text = "Sửa thông tin Khu";
17. }
18. }
20. **private** **void** btn\_huy\_Click(**object** sender, EventArgs e)
21. {
22. **this**.Close();
23. }
25. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
26. {
27. **if** (isAddNew == **true**)       //  Là trường hợp Insert
28. {
29. **try**
30. {
31. **this**.khu = **new** DTO\_KHU
32. {
33. TenKhu = **this**.txt\_tenKhu.Text
34. };
36. **if** (BLL\_KHU.ThemKhu(**this**.khu) == **true**)
37. {
38. XtraMessageBox.Show("Thêm khu thành công !", "Thông báo",
39. MessageBoxButtons.OK, MessageBoxIcon.Information);
41. fDayNha.isAdded = **true**;
42. **this**.Close();
43. }
44. }
45. **catch** (Exception ex)
46. {
47. **if** (ex.Message.Contains("duplicate"))
48. {
49. XtraMessageBox.Show("Tên khu đã tồn tại !", "Thông báo",
50. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
52. **this**.txt\_tenKhu.Focus();
53. **this**.txt\_tenKhu.SelectAll();
54. }
55. **else**
56. XtraMessageBox.Show("Lỗi khi thêm khu !" + "\n\nNội dung lỗi: \n" + ex.Message,                      "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
57. }
58. }
59. **else**                        //  Trường hợp Update
60. {
61. **try**
62. {
63. **this**.khu.TenKhu = **this**.txt\_tenKhu.Text;
65. **if** (BLL\_KHU.CapNhatKhu(khu) == **true**)
66. {
67. XtraMessageBox.Show("Cập nhật thông tin khu thành công !", "Thông báo",
68. MessageBoxButtons.OK, MessageBoxIcon.Information);
70. fDayNha.isUpdated = **true**;
71. **this**.Close();
72. }
73. }
74. **catch** (Exception ex)
75. {
76. **if** (ex.Message.Contains("duplicate"))
77. {
78. XtraMessageBox.Show("Tên khu đã tồn tại !", "Thông báo",
79. MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
81. **this**.txt\_tenKhu.Focus();
82. **this**.txt\_tenKhu.SelectAll();
83. }
84. **else**
85. XtraMessageBox.Show("Lỗi khi cập nhật khu !" + "\n\nNội dung lỗi: \n" + ex.Message,
86. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
87. }
88. }
89. }

#### Form quản lý thông tin tầng



Codes chính:

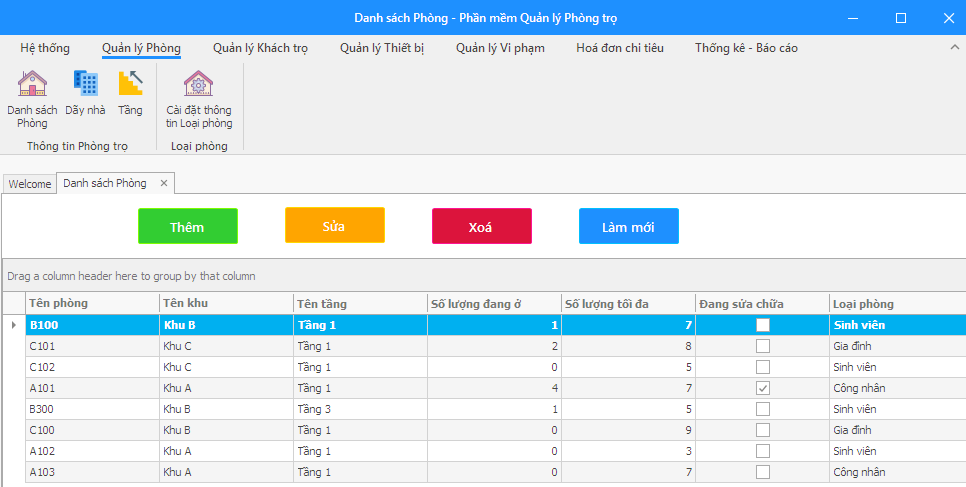
1. **public** **static** **bool** isAdded = **false**; //  Xác định liệu đã thêm hay chưa
2. **public** **static** **bool** isUpdated = **false**;
4. **public** fTang()
5. {
6. InitializeComponent();
7. }
9. **private** **void** LoadData()
10. {
11. DataTable dt = BLL\_TANG.LayDanhSachTang();
12. **if** (dt.Rows.Count == 0)
13. {
14. **this**.btn\_xoa.Enabled = **false**;
15. }
16. **else**
17. {
18. **this**.btn\_xoa.Enabled = **true**;
19. }
21. **this**.gridControl\_tang.DataSource = dt;
22. **this**.gridView\_tang.Columns["TENKHU"].Caption = "Tên Khu";
23. **this**.gridView\_tang.Columns["TENTANG"].Caption = "Tên Tầng";
25. **this**.gridView\_tang.Columns["MATANG"].Caption = "Mã Tầng";
26. **this**.gridView\_tang.Columns["MATANG"].Visible = **false**;
27. **this**.gridView\_tang.Columns["MAKHU"].Caption = "Mã Khu";
28. **this**.gridView\_tang.Columns["MAKHU"].Visible = **false**;
29. }
31. **private** **void** fTang\_Load(object sender, EventArgs e)
32. {
33. LoadData();
34. }
36. **private** **void** btn\_them\_Click(object sender, EventArgs e)
37. {
38. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.Tang.fAlter();
39. FormAlter.ShowDialog();

42. **if** (isAdded == **true**)
43. {
44. isAdded = **false**;
45. LoadData();
46. }
47. }
49. **private** **void** btn\_xoa\_Click(object sender, EventArgs e)
50. {
51. DTO\_TANG tangHienTai = LayThongTinTangHienTai();
53. **try**
54. {
55. DTO\_TANG tangCaoNhat = BLL\_TANG.LayTangCaoNhat\_TheoMaKhu(tangHienTai);
56. **if** (tangCaoNhat != null)
57. {
58. string message = String.Format("Bạn đang chọn khu {0}, tầng cao nhất là {1}.\n" +
59. "Chỉ có thể xoá thông tin từ tầng cao nhất trở xuống.\n" +
60. "Xác nhận xoá ?", tangCaoNhat.TenKhu, tangCaoNhat.TenTang);
62. DialogResult dr = XtraMessageBox.Show(message, "Thông báo",
63. MessageBoxButtons.YesNo, MessageBoxIcon.Question);

66. **if** (dr == DialogResult.Yes)
67. **if** (BLL\_TANG.TangDaDuocThamChieu(tangCaoNhat) == **true**)
68. {
69. XtraMessageBox.Show("Thông tin tầng đang được sử dụng, không thể xoá !\n" +
70. "Vui lòng xoá thông tin các PHÒNG của tầng này trước.",
71. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
72. }
73. **else**
74. {
75. message = String.Format("Xác nhận xoá thông tin: {0} ?", tangCaoNhat.TenTang);
76. dr = XtraMessageBox.Show(message, "Thông báo", MessageBoxButtons.YesNo);
78. **if** (dr == DialogResult.Yes)
79. {
80. **if** (BLL\_TANG.XoaTang(tangCaoNhat) == **true**)
81. {
82. XtraMessageBox.Show("Xoá thông tin tầng thành công !",
83. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
85. LoadData();
86. }
87. }
88. }
89. }
90. **else**
91. {
92. //   Không xảy ra - Chỉ khi có tầng trong khu thì mới có thể chọn
93. }
94. }
95. **catch** (Exception ex)
96. {
97. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
98. "Thông báo", MessageBoxButtons.OK,  MessageBoxIcon.Warning);
99. }
100. }


104. **private** DTO\_TANG LayThongTinTangHienTai()
105. {
106. DataRow dr = **this**.gridView\_tang.GetFocusedDataRow();
108. DTO\_TANG tang = **new** DTO\_TANG
109. {
110. MaKhu = dr["MAKHU"].ToString(),
111. MaTang = dr["MATANG"].ToString(),
112. TenTang = dr["TENTANG"].ToString(),
113. TenKhu = dr["TENKHU"].ToString()
114. };
116. **return** tang;
117. }
119. **private** **void** btn\_lamMoi\_Click(object sender, EventArgs e)
120. {
121. LoadData();
122. }

#### Form quản lý thông tin phòng

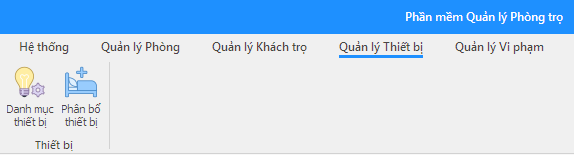


Codes chính:

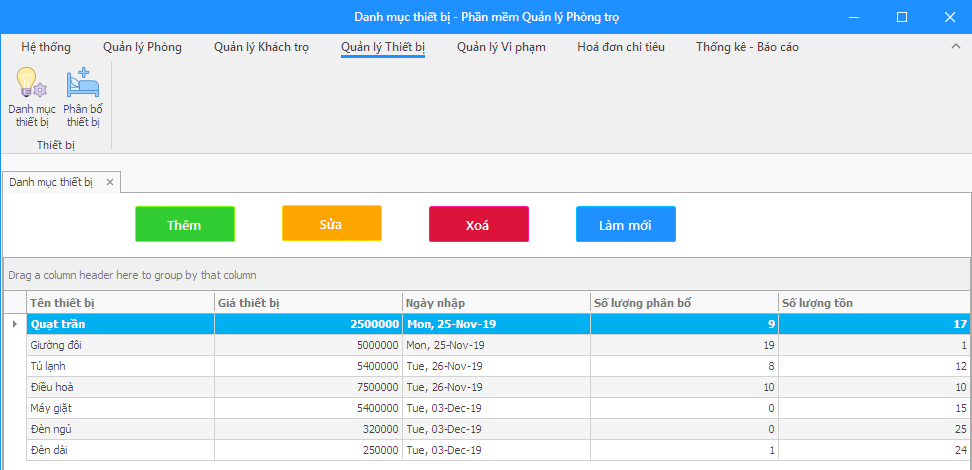
1. **public** **static** **bool** isAdded = **false**;
2. **public** **static** **bool** isUpdated = **false**;
4. **public** fPhong()
5. {
6. InitializeComponent();
7. }
9. **private** **void** fPhong\_Load(object sender, EventArgs e)
10. {
11. LoadData();
12. }
14. **private** **void** LoadData()
15. {
16. **this**.gridControl\_phong.DataSource = BLL\_PHONG.LayDanhSachPhong();
18. **this**.gridView\_phong.Columns["TENKHU"].Caption = "Tên khu";
19. **this**.gridView\_phong.Columns["TENTANG"].Caption = "Tên tầng";
20. **this**.gridView\_phong.Columns["TENPHG"].Caption = "Tên phòng";
21. **this**.gridView\_phong.Columns["TENPHG"].VisibleIndex = 0;
22. **this**.gridView\_phong.Columns["SOLG\_TOIDA"].Caption = "Số lượng tối đa";
23. **this**.gridView\_phong.Columns["SOLG\_DANGO"].Caption = "Số lượng đang ở";
24. **this**.gridView\_phong.Columns["DANGSUACHUA"].Caption = "Đang sửa chữa";
25. **this**.gridView\_phong.Columns["TENLOAIPHG"].Caption = "Loại phòng";
27. **this**.gridView\_phong.Columns["MAKHU"].Visible = **false**;
28. **this**.gridView\_phong.Columns["MATANG"].Visible = **false**;
29. **this**.gridView\_phong.Columns["MAPHG"].Visible = **false**;
30. **this**.gridView\_phong.Columns["MALOAIPHG"].Visible = **false**;
31. }
33. **private** **void** btn\_lamMoi\_Click(object sender, EventArgs e)
34. {
35. LoadData();
36. }
38. **private** **void** btn\_them\_Click(object sender, EventArgs e)
39. {
40. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.Phong.fAlter();
41. FormAlter.ShowDialog();
43. **if** (isAdded == **true**)
44. {
45. LoadData();
46. isAdded = **false**;
47. }
48. }
50. **private** **void** btn\_sua\_Click(object sender, EventArgs e)
51. {
52. DTO\_PHONG phongHienTai = LayPhongHienTai();
53. XtraForm FormAlter = **new** ChildrenForms.QuanLyPhong.Phong.fAlter(phongHienTai);
54. FormAlter.ShowDialog();
56. **if** (isUpdated == **true**)
57. {
58. LoadData();
59. isUpdated = **false**;
60. }
61. }

64. **private** DTO\_PHONG LayPhongHienTai()
65. {
66. DataRow dr = **this**.gridView\_phong.GetFocusedDataRow();
68. DTO\_PHONG phong = **new** DTO\_PHONG
69. {
70. MaPhong = dr["MAPHG"].ToString(),
71. TenPhong = dr["TENPHG"].ToString(),
72. SoLuongToiDa = (**int**)dr["SOLG\_TOIDA"],
73. SoLuongDangO = (**int**)dr["SOLG\_DANGO"],
74. MaLoaiPhong = dr["MALOAIPHG"].ToString(),
75. MaTang = dr["MATANG"].ToString(),
76. DangSuaChua = dr["DANGSUACHUA"].ToString(),
77. TenKhu = dr["TENKHU"].ToString(),
78. TenTang = dr["TENTANG"].ToString(),
79. TenLoaiPhong = dr["TENLOAIPHG"].ToString()
80. };
82. **return** phong;
83. }
85. **private** **void** btn\_xoa\_Click(object sender, EventArgs e)
86. {
87. **try**
88. {
89. DTO\_PHONG phong = LayPhongHienTai();
91. **if** (BLL\_PHONG.PhongDaDuocThamChieu(phong) == **true**)
92. XtraMessageBox.Show("Thông tin phòng đang được sử dụng, không thể xoá !",
93. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
94. **else**
95. {
96. DialogResult dr = XtraMessageBox.Show(String.Format("Xác nhận xoá thông tin: {0} ?", phong.TenPhong),
97. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
99. **if** (dr == DialogResult.Yes)
100. **if** (BLL\_PHONG.XoaPhong(phong) == **true**)
101. {
102. XtraMessageBox.Show("Xoá phòng thành công !", "Thông báo",
103. MessageBoxButtons.OK, MessageBoxIcon.Information);
105. LoadData();
106. }
107. }
108. }
109. **catch** (Exception ex)
110. {
111. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
112. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
113. }
114. }

### Nhóm form chức năng quản lý thông tin thiết bị



#### Form quản lý danh mục thiết bị

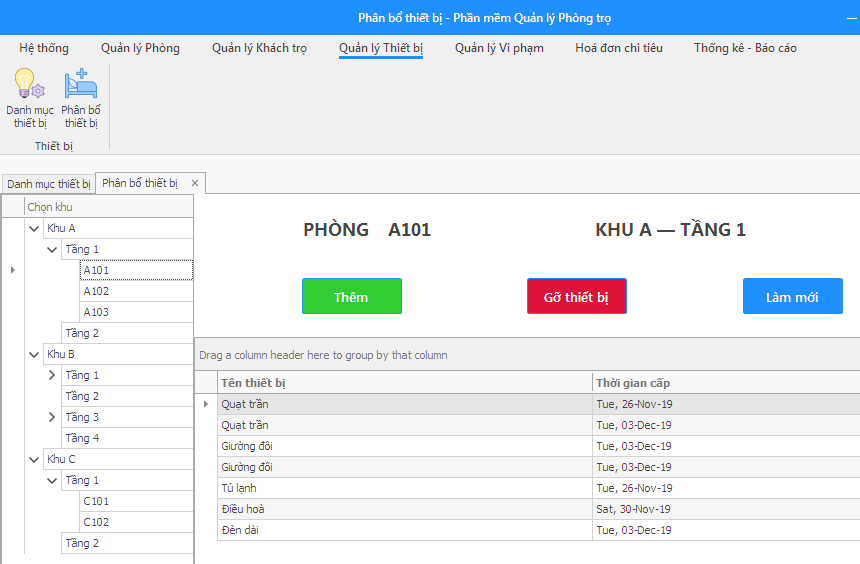


Codes chính:

1. **public** **static** **bool** isAdded = **false**;
2. **public** **static** **bool** isUpdated = **false**;
4. **public** fDanhMucThietBi()
5. {
6. InitializeComponent();
7. }
9. **private** **void** fDanhMucThietBi\_Load(object sender, EventArgs e)
10. {
11. LoadData();
12. }
14. **private** **void** LoadData()
15. {
16. DataTable dt = BLL\_THIETBI.LayDanhSachThietBi();
17. **if** (dt.Rows.Count == 0)
18. {
19. **this**.btn\_sua.Enabled = **false**;
20. **this**.btn\_xoa.Enabled = **false**;
21. }
22. **else**
23. {
24. **this**.btn\_sua.Enabled = **true**;
25. **this**.btn\_xoa.Enabled = **true**;
26. }
28. **this**.gridControl\_thietBi.DataSource = dt;
29. **this**.gridView\_thietBi.Columns["MATBI"].Visible = **false**;
31. **this**.gridView\_thietBi.Columns["TENTBI"].Caption = "Tên thiết bị";
32. **this**.gridView\_thietBi.Columns["GIATBI"].Caption = "Giá thiết bị";
33. **this**.gridView\_thietBi.Columns["NGAYNHAP"].Caption = "Ngày nhập";
34. **this**.gridView\_thietBi.Columns["SOLG\_TON"].Caption = "Số lượng tồn";
35. **this**.gridView\_thietBi.Columns["SOLG\_PHANBO"].Caption = "Số lượng phân bổ";
36. }
38. **private** **void** btn\_them\_Click(object sender, EventArgs e)
39. {
40. XtraForm FormAlter = **new** ChildrenForms.QuanLyThietBi.DanhMucThietBi.fAlter();
41. FormAlter.ShowDialog();
43. **if** (isAdded == **true**)
44. {
45. LoadData();
46. isAdded = **false**;
47. }
48. }
50. **private** **void** btn\_sua\_Click(object sender, EventArgs e)
51. {
52. DTO\_THIETBI thietBiHienTai = LayThietBiHienTai();
53. XtraForm FormAlter = **new** ChildrenForms.QuanLyThietBi.DanhMucThietBi.fAlter(thietBiHienTai);
54. FormAlter.ShowDialog();
56. **if** (isUpdated == **true**)
57. {
58. LoadData();
59. isUpdated = **false**;
60. }
61. }

64. **private** DTO\_THIETBI LayThietBiHienTai()
65. {
66. DataRow dr = **this**.gridView\_thietBi.GetFocusedDataRow();
68. DTO\_THIETBI thietBi = **new** DTO\_THIETBI
69. {
70. MaThietBi  = (string)dr["MATBI"],
71. TenThietBi = (string)dr["TENTBI"],
72. GiaThietBi = (**int**)dr["GIATBI"],
73. NgayNhap = (DateTime)dr["NGAYNHAP"],
74. SoLuongTon = (**int**)dr["SOLG\_TON"],
75. SoLuongPhanBo = (**int**)dr["SOLG\_PHANBO"]
76. };
78. **return** thietBi;
79. }
81. **private** **void** btn\_xoa\_Click(object sender, EventArgs e)
82. {
83. DTO\_THIETBI thietBiHienTai = LayThietBiHienTai();
85. **try**
86. {
87. **if** (BLL\_THIETBI.ThietBiDaDuocThamChieu(thietBiHienTai) == **true**)
88. {
89. XtraMessageBox.Show("Thông tin thiết bị đang được sử dụng, không thể xoá !",
90. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
91. }
92. **else**
93. {
94. DialogResult dr = XtraMessageBox.Show(String.Format("Xác nhận xoá thông tin thiết bị: {0} ?", thietBiHienTai.TenThietBi),
95. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
97. **if** (dr == DialogResult.Yes)
98. **if** (BLL\_THIETBI.XoaThietBi(thietBiHienTai) == **true**)
99. {
100. XtraMessageBox.Show("Xoá thông tin thiết bị thành công !", "Thông báo",
101. MessageBoxButtons.OK, MessageBoxIcon.Information);
103. LoadData();
104. }
105. }
106. }
107. **catch** (Exception ex)
108. {
109. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
110. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
111. }
112. }
114. **private** **void** btn\_lamMoi\_Click(object sender, EventArgs e)
115. {
116. LoadData();
117. }

#### Form phân bổ thiết bị tới các phòng



Codes chính:

1. **public** **static** **bool** isAdded = **false**;
2. **public** **static** **bool** isUpdated = **false**;
3. **private** string maPhong = "";

6. **public** fPhanBoThietBi()
7. {
8. InitializeComponent();
9. }
11. **private** **void** fPhanBoThietBi\_Load(object sender, EventArgs e)
12. {
13. LoadTreeView();
14. }
16. **private** **void** LoadTreeView()
17. {
18. string[] listKhu = BLL\_KHU.LayDanhSachTenKhu();
19. DTO\_KHU khu = **new** DTO\_KHU();
20. DTO\_TANG tang = **new** DTO\_TANG();
22. **int** countNodes = 0;
23. foreach (string str in listKhu)
24. {
25. **this**.treeList\_khuTang.Nodes.Add(**new** object[]{ str });
26. khu.TenKhu = str;
27. khu.MaKhu = BLL\_KHU.LayMaKhuTheoTenKhu(khu);
28. string[] listTang = BLL\_TANG.LayDanhSachTenTang\_TheoTenKhu(khu);   //  Lấy các tầng thuộc khu

31. **int** countLv1Nodes = 0;
32. foreach (string ss in listTang)
33. {
34. **this**.treeList\_khuTang.Nodes[countNodes].Nodes.Add(**new** object[] { ss });
36. tang.TenTang = ss;
37. tang.MaTang = BLL\_TANG.LayMaTang\_TheoTenTang\_MaKhu(tang, khu);
38. string[] listPhong = BLL\_PHONG.LayDanhSachTenPhongTheoMaTang(tang);

41. foreach (string s in listPhong)
42. **this**.treeList\_khuTang.Nodes[countNodes].Nodes[countLv1Nodes].Nodes.Add(**new** object[]{s});
44. countLv1Nodes++;
45. }

48. countNodes++;
49. }
50. }

53. **private** **void** LoadPhong()
54. {
55. TreeList tl = **this**.treeList\_khuTang;
56. TreeListNode node = tl.FocusedNode;
57. TreeListNode parentNode = node.ParentNode;
59. **if** (parentNode != null) //  Tầng
60. {
61. TreeListNode grandNode = parentNode.ParentNode;     //  Khu
63. **if** (grandNode != null)
64. {
65. **this**.btn\_them.Enabled = **true**;
66. **this**.btn\_xoa.Enabled = **true**;
67. **this**.btn\_lamMoi.Enabled = **true**;
69. string tenPhong = node.GetDisplayText(0);
70. string tenTang = parentNode.GetDisplayText(0);
71. string tenKhu = grandNode.GetDisplayText(0);

74. **this**.lb\_tenPhong.Text = tenPhong;
75. **this**.lb\_khuTang.Text = String.Format("{0} — {1}", tenKhu.ToUpper(), tenTang.ToUpper());

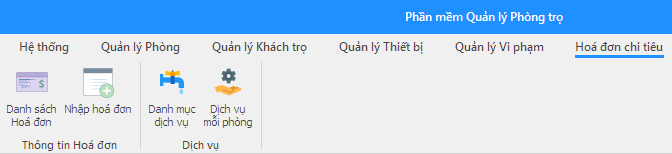
78. DTO\_PHONG phong = **new** DTO\_PHONG();
79. phong.TenPhong = tenPhong;
80. phong.MaPhong = BLL\_PHONG.LayMaPhongTheoTenPhong(phong);
81. **this**.maPhong = phong.MaPhong;

84. DataTable dt = BLL\_THIETBI\_PHONG.LayDanhSachThietBiPhongTheoMaPhong(phong);
85. **if** (dt.Rows.Count == 0)
86. {
87. **this**.btn\_xoa.Enabled = **false**;
88. }
89. **else**
90. {
91. **this**.btn\_xoa.Enabled = **true**;
92. }
94. **this**.gridControl\_thietBiPhong.DataSource = dt;
95. **this**.gridView\_thietBiPhong.Columns["MAPHG"].Visible = **false**;
96. **this**.gridView\_thietBiPhong.Columns["MATBI"].Visible = **false**;
98. **this**.gridView\_thietBiPhong.Columns["TENTBI"].VisibleIndex = 0;
99. **this**.gridView\_thietBiPhong.Columns["TENTBI"].Caption = "Tên thiết bị";
100. **this**.gridView\_thietBiPhong.Columns["THOIGIANCAP"].Caption = "Thời gian cấp";
101. }
102. **else**
103. {
104. **this**.btn\_them.Enabled = **false**;
105. **this**.btn\_xoa.Enabled = **false**;
106. **this**.btn\_lamMoi.Enabled = **false**;
108. **this**.gridControl\_thietBiPhong.DataSource = null;
109. **this**.lb\_tenPhong.Text = "?";
110. **this**.lb\_khuTang.Text = "KHU – TẦNG";
111. }
112. }
113. }
115. **private** **void** treeList\_khuTang\_FocusedNodeChanged(object sender, DevExpress.XtraTreeList.FocusedNodeChangedEventArgs e)
116. {
117. LoadPhong();
118. }

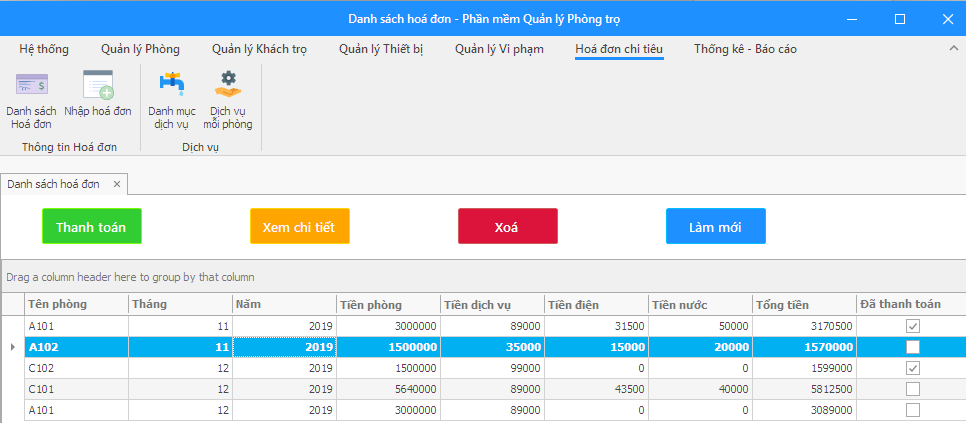
121. **private** **void** btn\_them\_Click(object sender, EventArgs e)
122. {
123. DTO\_THIETBI\_PHONG thietBiPhong = **new** DTO\_THIETBI\_PHONG();
124. thietBiPhong.MaPhong = **this**.maPhong;
126. XtraForm FormAlter = **new** ChildrenForms.QuanLyThietBi.PhanBoThietBi.fAlter(thietBiPhong);
127. FormAlter.ShowDialog();
129. **if** (isAdded == **true**)
130. {
131. LoadPhong();
132. isAdded = **false**;
133. }
134. }
136. **private** **void** btn\_lamMoi\_Click(object sender, EventArgs e)
137. {
138. LoadPhong();
139. }

142. **private** DTO\_THIETBI\_PHONG LayThietBiPhongHienTai()
143. {
144. DataRow dr = **this**.gridView\_thietBiPhong.GetFocusedDataRow();
146. DTO\_THIETBI\_PHONG thietBiPhong = **new** DTO\_THIETBI\_PHONG
147. {
148. MaThietBi = (string)dr["MATBI"],
149. TenThietBi = (string)dr["TENTBI"],
150. MaPhong = (string)dr["MAPHG"],
151. NgayCap = (DateTime)dr["THOIGIANCAP"]
152. };
154. **return** thietBiPhong;
155. }
157. **private** **void** btn\_xoa\_Click(object sender, EventArgs e)
158. {
159. DTO\_THIETBI\_PHONG thietBiPhong = LayThietBiPhongHienTai();
161. **try**
162. {
163. **if** (BLL\_THIETBI\_PHONG.XoaThietBiPhong(thietBiPhong) == **true**)
164. {
165. XtraMessageBox.Show("Gỡ thông tin thiết bị thành công !", "Thông báo",
166. MessageBoxButtons.OK, MessageBoxIcon.Information);
168. LoadPhong();
169. }
170. }
171. **catch** (Exception ex)
172. {
173. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
174. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
175. }
176. }

### Nhóm form chức năng quản lý thông tin dịch vụ - hoá đơn



#### Form danh sách hoá đơn



Codes chính:

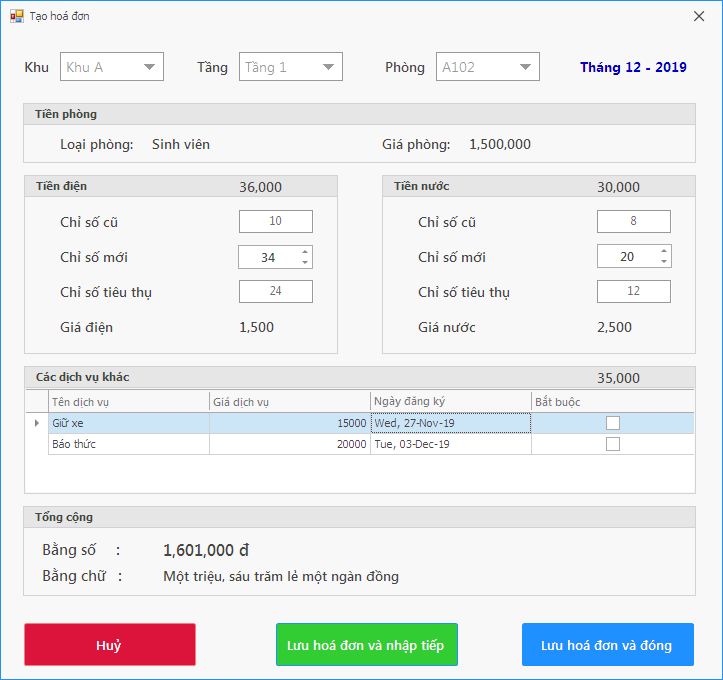
1. **private** DTO\_HOADON hoaDon;
3. **public** fDanhSachHoaDon()
4. {
5. InitializeComponent();
6. }
8. **private** **void** fDanhSachHoaDon\_Load(object sender, EventArgs e)
9. {
10. LoadData();
11. }
13. **private** **void** LoadData()
14. {
15. DataTable dt = BLL\_HOADON.LayDanhSachHoaDon();
17. **if** (dt.Rows.Count == 0)
18. {
19. **this**.btn\_thanhToan.Enabled = **false**;
20. **this**.btn\_xemChiTiet.Enabled = **false**;
21. **this**.btn\_xoa.Enabled = **false**;
22. }
23. **else**
24. {
25. **this**.btn\_thanhToan.Enabled = **true**;
26. **this**.btn\_xemChiTiet.Enabled = **true**;
27. **this**.btn\_xoa.Enabled = **true**;
28. }

31. **this**.gridControl\_hoaDon.DataSource = dt;
33. **this**.gridView\_hoaDon.Columns["MAHOADON"].Visible = **false**;
34. **this**.gridView\_hoaDon.Columns["MAPHG"].Visible = **false**;
36. **this**.gridView\_hoaDon.Columns["TENPHG"].Caption = "Tên phòng";
37. **this**.gridView\_hoaDon.Columns["THANG"].Caption = "Tháng";
38. **this**.gridView\_hoaDon.Columns["NAM"].Caption = "Năm";
40. **this**.gridView\_hoaDon.Columns["TIENPHONG"].Caption = "Tiền phòng";
41. **this**.gridView\_hoaDon.Columns["TIENDICHVU"].Caption = "Tiền dịch vụ";
42. **this**.gridView\_hoaDon.Columns["TIENDIEN"].Caption = "Tiền điện";
43. **this**.gridView\_hoaDon.Columns["TIENNUOC"].Caption = "Tiền nước";
44. **this**.gridView\_hoaDon.Columns["TONGTIEN"].Caption = "Tổng tiền";
45. **this**.gridView\_hoaDon.Columns["DATHANHTOAN"].Caption = "Đã thanh toán";
46. }
48. **private** **void** gridView\_hoaDon\_SelectionChanged(object sender, DevExpress.Data.SelectionChangedEventArgs e)
49. {
50. DataRow dr = **this**.gridView\_hoaDon.GetFocusedDataRow();
52. DTO\_HOADON hoaDonHienTai = **new** DTO\_HOADON
53. {
54. MaHoaDon = (string)dr["MAHOADON"],
55. MaPhong = (string)dr["MAPHG"],
56. TenPhong = (string)dr["TENPHG"],
57. Thang = (**int**)dr["THANG"],
58. Nam = (**int**)dr["NAM"],
59. TienPhong = (**int**)dr["TIENPHONG"],
60. TienDien = (**int**)dr["TIENDIEN"],
61. TienNuoc = (**int**)dr["TIENNUOC"],
62. TienDichVu = (**int**)dr["TIENDICHVU"],
63. TongTien = (**int**)dr["TONGTIEN"],
64. DaThanhToan = dr["DATHANHTOAN"].ToString()
65. };
67. **this**.hoaDon = hoaDonHienTai;

70. **if** (hoaDonHienTai.DaThanhToan == "True")
71. {
72. **this**.btn\_thanhToan.Enabled = **false**;
73. }
74. **else**
75. {
76. **this**.btn\_thanhToan.Enabled = **true**;
77. }
78. }
80. **private** **void** btn\_lamMoi\_Click(object sender, EventArgs e)
81. {
82. LoadData();
83. }
85. **private** **void** btn\_thanhToan\_Click(object sender, EventArgs e)
86. {
87. **try**
88. {
89. string message = String.Format(@"Xác nhận thanh toán hoá đơn
91. Phòng: {0}
92. Tháng: {1}
93. Năm: {2}
95. Tiền phòng: {3:#,##0}
96. Tiền điện: {4:#,##0}
97. Tiền nước: {5:#,##0}
98. Tiền dịch vụ: {6:#,##0}
100. Tổng tiền: {7:#,##0}
102. ",
103. **this**.hoaDon.TenPhong,
104. **this**.hoaDon.Thang,
105. **this**.hoaDon.Nam,
106. **this**.hoaDon.TienPhong,
107. **this**.hoaDon.TienDien,
108. **this**.hoaDon.TienNuoc,
109. **this**.hoaDon.TienDichVu,
110. **this**.hoaDon.TongTien);

113. DialogResult dr = XtraMessageBox.Show(message,
114. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
116. **if** (dr == DialogResult.Yes)
117. **if** (BLL\_HOADON.ThanhToanHoaDon(**this**.hoaDon) == **true**)
118. {
119. XtraMessageBox.Show("Thanh toán thành công !", "Thông báo",
120. MessageBoxButtons.OK, MessageBoxIcon.Information);
122. LoadData();
123. }
124. }
125. **catch** (Exception ex)
126. {
127. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
128. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
129. }
130. }
132. **private** **void** btn\_xemChiTiet\_Click(object sender, EventArgs e)
133. {
134. XtraForm FormChiTiet = **new** ChildrenForms.HoaDonChiTieu.DanhSachHoaDon.fChiTietHoaDon(**this**.hoaDon);
135. FormChiTiet.ShowDialog();
136. }
138. **private** **void** btn\_xoa\_Click(object sender, EventArgs e)
139. {
140. **try**
141. {
142. DialogResult dr = XtraMessageBox.Show("Thông tin quan trọng, xác nhận xoá ?",
143. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
145. **if** (dr == DialogResult.Yes)
146. **if** (BLL\_HOADON.XoaHoaDon(**this**.hoaDon) == **true**)
147. {
148. XtraMessageBox.Show("Xoá hoá đơn thành công !", "Thông báo",
149. MessageBoxButtons.OK, MessageBoxIcon.Information);
151. LoadData();
152. }
153. }
154. **catch** (Exception ex)
155. {
156. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
157. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
158. } }

#### Form tạo hoá đơn



Codes chính:

1. **int** tongTien = 0;
2. **int** tienPhong = 0;
3. **int** tienDichVu = 0;
4. **int** tienDien = 0;
5. **int** tienNuoc = 0;
6. **int** cbPhong\_lastSelectedIndex = -1;
8. **public** fThemHoaDon()
9. {
10. InitializeComponent();
11. }
13. **private** **void** btn\_huy\_Click(**object** sender, EventArgs e)
14. {
15. **this**.Close();
16. }
18. **private** **void** fThemHoaDon\_Load(**object** sender, EventArgs e)
19. {
20. **this**.cb\_khu.Items.AddRange(BLL\_KHU.LayDanhSachTenKhu());
21. SetMonthYear();
22. LoadGiaDienNuoc();
23. }
25. **private** **void** SetMonthYear()
26. {
27. **int** thang = DateTime.Today.Month;
28. **int** nam = DateTime.Today.Year;
30. **this**.lb\_thangNam.Text = String.Format("Tháng {0:d2} - {1}", thang, nam);
31. }
33. **private** **void** LoadGiaDienNuoc()
34. {
35. DataTable dt = BLL\_DICHVU.LayThongTinDienNuoc();
37. **this**.lb\_giaDien.Text = String.Format("{0:#,##0}", dt.Rows[0]["GIADICHVU"]);
38. **this**.lb\_giaNuoc.Text = String.Format("{0:#,##0}", dt.Rows[1]["GIADICHVU"]);
39. }
41. **private** **void** cb\_khu\_SelectedIndexChanged(**object** sender, EventArgs e)
42. {
43. DTO\_KHU khu = **new** DTO\_KHU();
44. khu.TenKhu = **this**.cb\_khu.Text;
46. **this**.cb\_tang.SelectedIndex = -1;
47. **this**.cb\_phong.SelectedIndex = -1;
48. **this**.cb\_tang.Items.Clear();
49. **this**.cb\_phong.Items.Clear();
51. **this**.cb\_tang.Items.AddRange(BLL\_TANG.LayDanhSachTenTang\_TheoTenKhu(khu));
52. }
54. **private** **void** cb\_tang\_SelectedIndexChanged(**object** sender, EventArgs e)
55. {
56. DTO\_KHU khu = **new** DTO\_KHU();
57. khu.TenKhu = **this**.cb\_khu.Text;
58. khu.MaKhu = BLL\_KHU.LayMaKhuTheoTenKhu(khu);
60. DTO\_TANG tang = **new** DTO\_TANG();
61. tang.TenTang = **this**.cb\_tang.Text;
62. tang.MaTang = BLL\_TANG.LayMaTang\_TheoTenTang\_MaKhu(tang, khu);
64. **this**.cb\_phong.SelectedIndex = -1;
65. **this**.cb\_phong.Items.Clear();
66. **this**.cb\_phong.Items.AddRange(BLL\_PHONG.LayDanhSachTenPhongTheoMaTang(tang));
67. }
69. **private** **void** cb\_phong\_SelectedIndexChanged(**object** sender, EventArgs e)
70. {
71. **if** (**this**.cb\_phong.SelectedIndex != **this**.cbPhong\_lastSelectedIndex)
72. {
73. **this**.cbPhong\_lastSelectedIndex = **this**.cb\_phong.SelectedIndex;
74. }
75. **else**
76. {
77. **return**;
78. }

81. ResetForm();
83. **int** thang = DateTime.Today.Month;
84. **int** nam = DateTime.Today.Year;

87. DTO\_PHONG phong = **new** DTO\_PHONG();
88. phong.TenPhong = **this**.cb\_phong.Text;
89. phong.MaPhong = BLL\_PHONG.LayMaPhongTheoTenPhong(phong);
91. DTO\_HOADON hoaDon = **new** DTO\_HOADON
92. {
93. MaPhong = phong.MaPhong,
94. Thang = thang,
95. Nam = nam
96. };


100. **try**
101. {
102. **if** (**this**.cb\_phong.SelectedIndex >= 0)
103. **if** (BLL\_HOADON.PhongDaGhiHoaDon(hoaDon) == **true**)
104. {
105. XtraMessageBox.Show("Phòng đã ghi hoá đơn tháng này !", "Thông báo",
106. MessageBoxButtons.OK, MessageBoxIcon.Warning);
108. **this**.cb\_phong.SelectedIndex = -1;
109. **this**.btn\_luuVaNhap.Enabled = **false**;
110. **this**.btn\_luuVaThoat.Enabled = **false**;
111. **this**.num\_soDienMoi.Enabled = **false**;
112. **this**.num\_soNuocMoi.Enabled = **false**;
113. }
114. **else**
115. {
116. **this**.btn\_luuVaNhap.Enabled = **true**;
117. **this**.btn\_luuVaThoat.Enabled = **true**;
118. **this**.num\_soDienMoi.Enabled = **true**;
119. **this**.num\_soNuocMoi.Enabled = **true**;

122. DataTable dt = BLL\_PHONG.LayThongTinLoaiPhongTheoTenPhong(phong);
123. **this**.lb\_loaiPhong.Text = (**string**)dt.Rows[0]["TENLOAIPHG"];
124. **int** giaPhong = (**int**)dt.Rows[0]["GIAPHG"];
125. **this**.lb\_giaPhong.Text = String.Format("{0:#,##0}", giaPhong);
126. **this**.lb\_loaiPhong.Visible = **true**;
127. **this**.lb\_giaPhong.Visible = **true**;
129. **this**.tienPhong = giaPhong;
130. SetTotal();

133. DTO\_HOADON\_DIENNUOC hdn = BLL\_HOADON\_DIENNUOC.LayThongTinThangTruoc(hoaDon);
134. **this**.txt\_soDienCu.Text = hdn.SoDien.ToString();
135. **this**.txt\_soNuocCu.Text = hdn.SoNuoc.ToString();

138. DataTable listServices = BLL\_DICHVU\_PHONG.LayDanhSachDichVuKhacTheoMaPhong(phong);
139. **this**.gridControl\_dichVuPhong.DataSource = listServices;
140. **this**.gridView\_dichVuPhong.Columns["MAPHG"].Visible = **false**;
141. **this**.gridView\_dichVuPhong.Columns["MADICHVU"].Visible = **false**;
142. **this**.gridView\_dichVuPhong.Columns["TENDICHVU"].Caption = "Tên dịch vụ";
143. **this**.gridView\_dichVuPhong.Columns["TENDICHVU"].VisibleIndex = 0;
144. **this**.gridView\_dichVuPhong.Columns["GIADICHVU"].Caption = "Giá dịch vụ";
145. **this**.gridView\_dichVuPhong.Columns["GIADICHVU"].VisibleIndex = 1;
146. **this**.gridView\_dichVuPhong.Columns["NGAYDANGKY"].Caption = "Ngày đăng ký";
147. **this**.gridView\_dichVuPhong.Columns["NGAYDANGKY"].VisibleIndex = 2;
148. **this**.gridView\_dichVuPhong.Columns["BATBUOC"].Caption = "Bắt buộc";
150. **if** (listServices.Rows.Count > 0)
151. {
152. **int** sum = 0;
154. **foreach** (DataRow dr **in** listServices.Rows)
155. sum += (**int**)dr["GIADICHVU"];
157. **this**.lb\_tongTienDichVu.Visible = **true**;
158. **this**.lb\_tongTienDichVu.Text = String.Format("{0:#,##0}", sum);
160. **this**.tienDichVu = sum;
161. SetTotal();
162. }
163. }
164. **else**
165. {
166. **this**.btn\_luuVaNhap.Enabled = **false**;
167. **this**.btn\_luuVaThoat.Enabled = **false**;
168. **this**.num\_soDienMoi.Enabled = **false**;
169. **this**.num\_soNuocMoi.Enabled = **false**;
170. }
171. }
172. **catch** (Exception ex)
173. {
174. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
175. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
176. }
177. }
179. **private** **void** num\_soDienMoi\_EditValueChanged(**object** sender, EventArgs e)
180. {
181. **int** soDienTieuThu = (**int**)**this**.num\_soDienMoi.Value - **int**.Parse(txt\_soDienCu.Text);
182. **this**.txt\_soDienTieuThu.Text = soDienTieuThu.ToString();
184. **int** tongTienDien = soDienTieuThu \* (**int**.Parse(Str.Remove(**this**.lb\_giaDien.Text, ',')));
185. **this**.tienDien = tongTienDien;
186. SetTotal();
188. **this**.lb\_tongTienDien.Visible = **true**;
189. **this**.lb\_tongTienDien.Text = String.Format("{0:#,##0}", tongTienDien);
190. }
192. **private** **void** num\_soNuocMoi\_EditValueChanged(**object** sender, EventArgs e)
193. {
194. **int** soNuocTieuThu = (**int**)**this**.num\_soNuocMoi.Value - **int**.Parse(txt\_soNuocCu.Text);
195. **this**.txt\_soNuocTieuThu.Text = soNuocTieuThu.ToString();
197. **int** tongTienNuoc = soNuocTieuThu \* (**int**.Parse(Str.Remove(**this**.lb\_giaNuoc.Text, ',')));
198. **this**.tienNuoc = tongTienNuoc;
199. SetTotal();
201. **this**.lb\_tongTienNuoc.Visible = **true**;
202. **this**.lb\_tongTienNuoc.Text = String.Format("{0:#,##0}", tongTienNuoc);
203. }

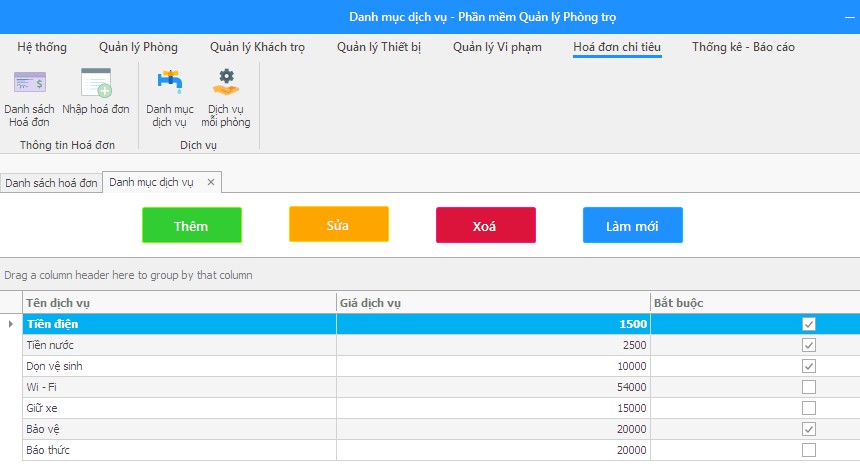
206. **private** **void** SetTotal()
207. {
208. **this**.tongTien = **this**.tienPhong + **this**.tienDichVu + **this**.tienDien + **this**.tienNuoc;
210. **this**.lb\_tongTien.Text = String.Format("{0:#,##0} đ", **this**.tongTien);
211. **this**.lb\_tienBangChu.Text = Reader.ReadNumber((**decimal**)**this**.tongTien);
212. }
214. **private** **void** ResetForm()
215. {
216. **this**.lb\_loaiPhong.Visible = **false**;
217. **this**.lb\_giaPhong.Visible = **false**;
218. **this**.lb\_tongTienDichVu.Visible = **false**;
219. **this**.lb\_tongTienNuoc.Visible = **false**;
220. **this**.lb\_tongTienDien.Visible = **false**;
222. **this**.num\_soDienMoi.Value = 0;
223. **this**.num\_soNuocMoi.Value = 0;
225. **this**.txt\_soDienCu.Text = "–";
226. **this**.txt\_soNuocCu.Text = "–";
227. **this**.txt\_soDienTieuThu.Text = "–";
228. **this**.txt\_soNuocTieuThu.Text = "–";
230. **this**.tienDichVu = 0;
231. **this**.tienPhong = 0;
232. **this**.tienDien = 0;
233. **this**.tienNuoc = 0;
234. SetTotal();
236. **this**.gridControl\_dichVuPhong.DataSource = **null**;
237. }

240. **private** **bool** LuuHoaDon()
241. {
242. **if** ((**int**)**this**.num\_soDienMoi.Value < (**int**.Parse(**this**.txt\_soDienCu.Text)))
243. {
244. XtraMessageBox.Show("Số điện mới không hợp lệ !", "Thông báo",
245. MessageBoxButtons.OK, MessageBoxIcon.Warning);
246. }
247. **else**
248. **if** ((**int**)**this**.num\_soNuocMoi.Value < (**int**.Parse(**this**.txt\_soNuocCu.Text)))
249. {
250. XtraMessageBox.Show("Số nước mới không hợp lệ !", "Thông báo",
251. MessageBoxButtons.OK, MessageBoxIcon.Warning);
252. }
253. **else**
254. {
255. DTO\_PHONG phong = **new** DTO\_PHONG();
256. phong.TenPhong = **this**.cb\_phong.Text;
257. phong.MaPhong = BLL\_PHONG.LayMaPhongTheoTenPhong(phong);
259. DTO\_HOADON hoaDon = **new** DTO\_HOADON
260. {
261. MaPhong = phong.MaPhong,
262. Thang = DateTime.Today.Month,
263. Nam = DateTime.Today.Year
264. };
266. DTO\_HOADON\_DIENNUOC hoaDonDienNuoc = **new** DTO\_HOADON\_DIENNUOC
267. {
268. SoDien = (**int**)**this**.num\_soDienMoi.Value,
269. SoNuoc = (**int**)**this**.num\_soNuocMoi.Value
270. };

273. **try**
274. {
275. **if** (BLL\_HOADON.ThemHoaDon(hoaDon, hoaDonDienNuoc) == **true**)
276. {
277. XtraMessageBox.Show("Ghi hoá đơn thành công !", "Thông báo",
278. MessageBoxButtons.OK, MessageBoxIcon.Information);
280. **return** **true**;
281. }
282. }
283. **catch** (Exception ex)
284. {
285. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
286. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
287. }
288. }

291. **return** **false**;
292. }
294. **private** **void** btn\_luuVaThoat\_Click(**object** sender, EventArgs e)
295. {
296. **if** (LuuHoaDon() == **true**)
297. **this**.Close();
298. }
300. **private** **void** btn\_luuVaNhap\_Click(**object** sender, EventArgs e)
301. {
302. **if** (LuuHoaDon() == **true**)
303. {
304. ResetForm();
306. **this**.cb\_khu.SelectedIndex = -1;
307. **this**.cb\_tang.SelectedIndex = -1;
308. **this**.cb\_phong.SelectedIndex = -1;
309. }
310. }

#### Form danh mục dịch vụ



Codes chính:

1. **public** **static** **bool** isAdded = **false**;
2. **public** **static** **bool** isUpdated = **false**;
4. **public** fDanhMucDichVu()
5. {
6. InitializeComponent();
7. }
9. **private** **void** fDanhMucDichVu\_Load(**object** sender, EventArgs e)
10. {
11. LoadData();
12. }
14. **private** **void** LoadData()
15. {
16. DataTable dt = BLL\_DICHVU.LayDanhSachDichVu();
17. **if** (dt.Rows.Count == 0)
18. {
19. **this**.btn\_sua.Enabled = **false**;
20. **this**.btn\_xoa.Enabled = **false**;
21. }
22. **else**
23. {
24. **this**.btn\_sua.Enabled = **true**;
25. **this**.btn\_xoa.Enabled = **true**;
26. }

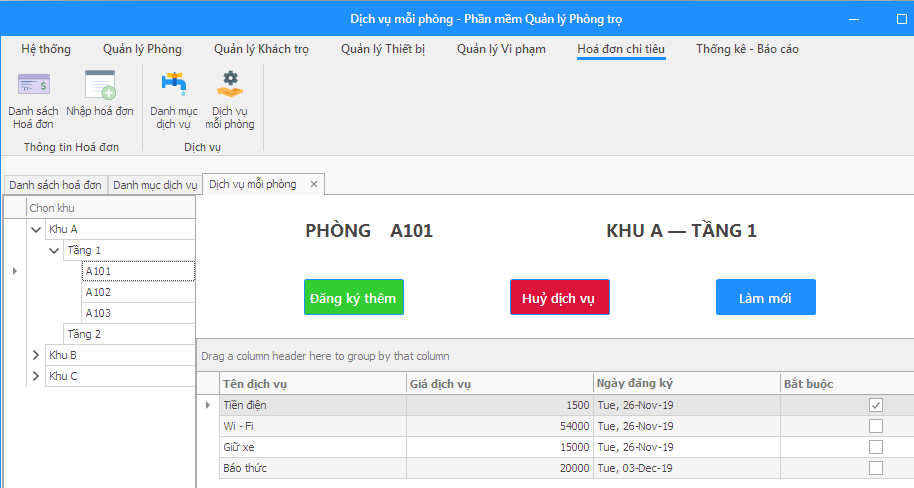
29. **this**.gridControl\_dichVu.DataSource = dt;
31. **this**.gridView\_dichVu.Columns["MADICHVU"].Visible = **false**;
32. **this**.gridView\_dichVu.Columns["TENDICHVU"].Caption = "Tên dịch vụ";
33. **this**.gridView\_dichVu.Columns["GIADICHVU"].Caption = "Giá dịch vụ";
34. **this**.gridView\_dichVu.Columns["BATBUOC"].Caption = "Bắt buộc";
35. }
37. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
38. {
39. XtraForm FormAlter = **new** ChildrenForms.DichVu.DanhMucDichVu.fAlter();
40. FormAlter.ShowDialog();
42. **if** (isAdded == **true**)
43. {
44. LoadData();
45. isAdded = **false**;
46. }
47. }
49. **private** **void** btn\_sua\_Click(**object** sender, EventArgs e)
50. {
51. DTO\_DICHVU dichVuHienTai = LayDichVuHienTai();
52. XtraForm FormAlter = **new** ChildrenForms.DichVu.DanhMucDichVu.fAlter(dichVuHienTai);
53. FormAlter.ShowDialog();
55. **if** (isUpdated == **true**)
56. {
57. LoadData();
58. isUpdated = **false**;
59. }
60. }
62. **private** **void** btn\_xoa\_Click(**object** sender, EventArgs e)
63. {
64. **try**
65. {
66. DTO\_DICHVU dichVuHienTai = LayDichVuHienTai();

69. **if** (dichVuHienTai.TenDichVu == "Tiền điện" ||
70. dichVuHienTai.TenDichVu == "Tiền nước")
71. {
72. XtraMessageBox.Show("Dịch vụ này là mặc định, không thể xoá !", "Thông báo",
73. MessageBoxButtons.OK, MessageBoxIcon.Warning);
75. **return**;
76. }
77. **else**
78. {
79. **if** (BLL\_DICHVU.DichVuDaDuocThamChieu(dichVuHienTai) == **true**)
80. {
81. XtraMessageBox.Show("Thông tin dịch vụ đang được sử dụng, không thể xoá !",
82. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
84. **return**;
85. }
86. **else**
87. {
88. DialogResult dr = XtraMessageBox.Show(String.Format("Xác nhận xoá dịch vụ: {0} ?", dichVuHienTai.TenDichVu),
89. "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

92. **if** (dr== DialogResult.Yes)
93. **if** (BLL\_DICHVU.XoaDichVu(dichVuHienTai) == **true**)
94. {
95. XtraMessageBox.Show("Xoá thông tin dịch vụ thành công !");
96. LoadData();
97. }
98. }
99. }
100. }
101. **catch** (Exception ex)
102. {
103. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
104. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
105. }
106. }
108. **private** **void** btn\_lamMoi\_Click(**object** sender, EventArgs e)
109. {
110. LoadData();
111. }

114. **private** DTO\_DICHVU LayDichVuHienTai()
115. {
116. DataRow dr = **this**.gridView\_dichVu.GetFocusedDataRow();
118. DTO\_DICHVU dichVu = **new** DTO\_DICHVU
119. {
120. MaDichVu = (**string**)dr["MADICHVU"],
121. TenDichVu = (**string**)dr["TENDICHVU"],
122. GiaDichVu = (**int**)dr["GIADICHVU"],
123. BatBuoc = dr["BATBUOC"].ToString()
124. };
126. **return** dichVu;
127. }

#### Form quản lý dịch vụ ở mỗi phòng



Codes chính:

1. **public** **static** **bool** isAdded = **false**;
2. **private** **string** maPhong;
4. **public** fDichVuMoiPhong()
5. {
6. InitializeComponent();
7. }
9. **private** **void** LoadTreeView()
10. {
11. **string**[] listKhu = BLL\_KHU.LayDanhSachTenKhu();
12. DTO\_KHU khu = **new** DTO\_KHU();
13. DTO\_TANG tang = **new** DTO\_TANG();
15. **int** countNodes = 0;
16. **foreach** (**string** str **in** listKhu)
17. {
18. **this**.treeList\_khuTang.Nodes.Add(**new** **object**[] { str });
19. khu.TenKhu = str;
20. khu.MaKhu = BLL\_KHU.LayMaKhuTheoTenKhu(khu);
21. **string**[] listTang = BLL\_TANG.LayDanhSachTenTang\_TheoTenKhu(khu);   //  Lấy các tầng thuộc khu

24. **int** countLv1Nodes = 0;
25. **foreach** (**string** ss **in** listTang)
26. {
27. **this**.treeList\_khuTang.Nodes[countNodes].Nodes.Add(**new** **object**[] { ss });
29. tang.TenTang = ss;
30. tang.MaTang = BLL\_TANG.LayMaTang\_TheoTenTang\_MaKhu(tang, khu);
31. **string**[] listPhong = BLL\_PHONG.LayDanhSachTenPhongTheoMaTang(tang);

34. **foreach** (**string** s **in** listPhong)
35. **this**.treeList\_khuTang.Nodes[countNodes].Nodes[countLv1Nodes].Nodes.Add(**new** **object**[] { s });
37. countLv1Nodes++;
38. }

41. countNodes++;
42. }
43. }
45. **private** **void** fDichVuMoiPhong\_Load(**object** sender, EventArgs e)
46. {
47. LoadTreeView();
48. }

51. **private** **void** LoadPhong()
52. {
53. TreeList tl = **this**.treeList\_khuTang;
54. TreeListNode node = tl.FocusedNode;
55. TreeListNode parentNode = node.ParentNode;
57. **if** (parentNode != **null**) //  Tầng
58. {
59. TreeListNode grandNode = parentNode.ParentNode;     //  Khu
61. **if** (grandNode != **null**)
62. {
63. **this**.btn\_dangKyThem.Enabled = **true**;
64. **this**.btn\_huyDichVu.Enabled = **true**;
65. **this**.btn\_lamMoi.Enabled = **true**;
67. **string** tenPhong = node.GetDisplayText(0);
68. **string** tenTang = parentNode.GetDisplayText(0);
69. **string** tenKhu = grandNode.GetDisplayText(0);

72. **this**.lb\_tenPhong.Text = tenPhong;
73. **this**.lb\_khuTang.Text = String.Format("{0} — {1}", tenKhu.ToUpper(), tenTang.ToUpper());

76. DTO\_PHONG phong = **new** DTO\_PHONG();
77. phong.TenPhong = tenPhong;
78. phong.MaPhong = BLL\_PHONG.LayMaPhongTheoTenPhong(phong);
79. **this**.maPhong = phong.MaPhong;
81. DataTable dt = BLL\_DICHVU\_PHONG.LayDanhSachDichVuTheoMaPhong(phong);
82. **if** (dt.Rows.Count == 0)
83. {
84. **this**.btn\_huyDichVu.Enabled = **false**;
85. }
86. **else**
87. {
88. **this**.btn\_huyDichVu.Enabled = **true**;
89. }
91. **this**.gridControl\_dichVuPhong.DataSource = dt;
93. **this**.gridView\_dichVuPhong.Columns["MAPHG"].Visible = **false**;
94. **this**.gridView\_dichVuPhong.Columns["MADICHVU"].Visible = **false**;
96. **this**.gridView\_dichVuPhong.Columns["TENDICHVU"].Caption = "Tên dịch vụ";
97. **this**.gridView\_dichVuPhong.Columns["TENDICHVU"].VisibleIndex = 0;
99. **this**.gridView\_dichVuPhong.Columns["GIADICHVU"].Caption = "Giá dịch vụ";
100. **this**.gridView\_dichVuPhong.Columns["GIADICHVU"].VisibleIndex = 1;
102. **this**.gridView\_dichVuPhong.Columns["NGAYDANGKY"].Caption = "Ngày đăng ký";
103. **this**.gridView\_dichVuPhong.Columns["NGAYDANGKY"].VisibleIndex = 2;
105. **this**.gridView\_dichVuPhong.Columns["BATBUOC"].Caption = "Bắt buộc";
106. }
107. **else**
108. {
109. **this**.btn\_dangKyThem.Enabled = **false**;
110. **this**.btn\_lamMoi.Enabled = **false**;
111. **this**.btn\_huyDichVu.Enabled = **false**;
113. **this**.lb\_tenPhong.Text = "?";
114. **this**.lb\_khuTang.Text = "KHU – TẦNG";
116. **this**.gridControl\_dichVuPhong.DataSource = **null**;
117. }
118. }
119. }

122. **private** **void** treeList\_khuTang\_FocusedNodeChanged(**object** sender, DevExpress.XtraTreeList.FocusedNodeChangedEventArgs e)
123. {
124. LoadPhong();
125. }
127. **private** **void** btn\_lamMoi\_Click(**object** sender, EventArgs e)
128. {
129. LoadPhong();
130. }
132. **private** **void** btn\_dangKyThem\_Click(**object** sender, EventArgs e)
133. {
134. DTO\_DICHVU\_PHONG dichVuPhongHienTai = LayDichVuPhongHienTai();
135. XtraForm FormAlter = **new** ChildrenForms.DichVu.DichVuMoiPhong.fAlter(dichVuPhongHienTai);
136. FormAlter.ShowDialog();

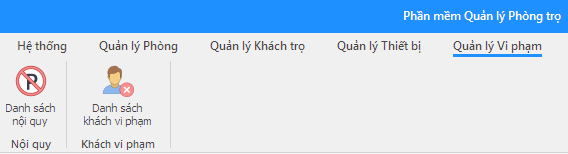
139. **if** (isAdded == **true**)
140. {
141. LoadPhong();
142. isAdded = **false**;
143. }
144. }

147. **private** DTO\_DICHVU\_PHONG LayDichVuPhongHienTai()
148. {
149. DataRow dr = **this**.gridView\_dichVuPhong.GetFocusedDataRow();
151. **if** (dr == **null**) //  GridView không có dòng
152. **return** **new** DTO\_DICHVU\_PHONG
153. {
154. MaPhong = **this**.maPhong,
155. MaDichVu = **null**
156. };

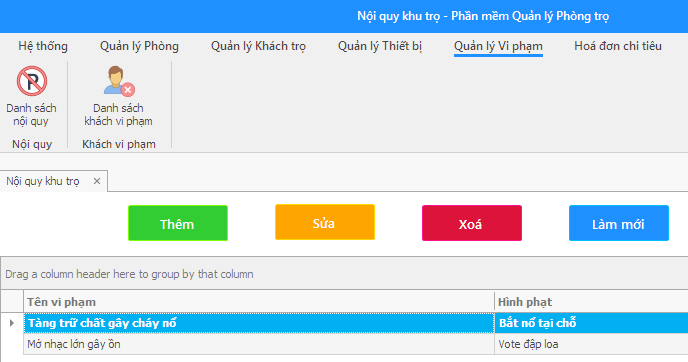
159. DTO\_DICHVU\_PHONG dichVuPhong = **new** DTO\_DICHVU\_PHONG //  GridView có ít nhất 1 dòng
160. {
161. MaDichVu = (**string**)dr["MADICHVU"],
162. MaPhong = (**string**)dr["MAPHG"],
163. NgayDangKy = (DateTime)dr["NGAYDANGKY"],
164. TenDichVu = (**string**)dr["TENDICHVU"]
165. };
167. **return** dichVuPhong;
168. }
170. **private** **void** btn\_huyDichVu\_Click(**object** sender, EventArgs e)
171. {
172. DTO\_DICHVU\_PHONG dichVuPhongHienTai = LayDichVuPhongHienTai();

175. **if** (dichVuPhongHienTai.MaDichVu == **null**)
176. {
177. XtraMessageBox.Show("Không có dữ liệu !", "Thông báo",
178. MessageBoxButtons.OK, MessageBoxIcon.Warning);
179. }
180. **else**
181. {
182. **if** (dichVuPhongHienTai.TenDichVu == "Tiền điện" ||
183. dichVuPhongHienTai.TenDichVu == "Tiền nước")
184. {
185. XtraMessageBox.Show("Dịch vụ mặc định, không thể huỷ !", "Thông báo",
186. MessageBoxButtons.OK, MessageBoxIcon.Warning);
188. **return**;
189. }
191. **string** question = String.Format("Xác nhận huỷ đăng ký dịch vụ: {0} ?", dichVuPhongHienTai.TenDichVu);
193. DialogResult dr = XtraMessageBox.Show(question, "Thông báo",
194. MessageBoxButtons.YesNo, MessageBoxIcon.Question);
196. **if** (dr == DialogResult.Yes)
197. {
198. **try**
199. {
200. **if** (BLL\_DICHVU\_PHONG.XoaDichVuPhong(dichVuPhongHienTai) == **true**)
201. {
202. XtraMessageBox.Show("Huỷ đăng ký dịch vụ thành công !",
203. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
205. LoadPhong();
206. }
207. }
208. **catch** (Exception ex)
209. {
210. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
211. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
212. }
213. }
214. }
215. }

### Nhóm form chức năng quản lý thông tin khách vi phạm



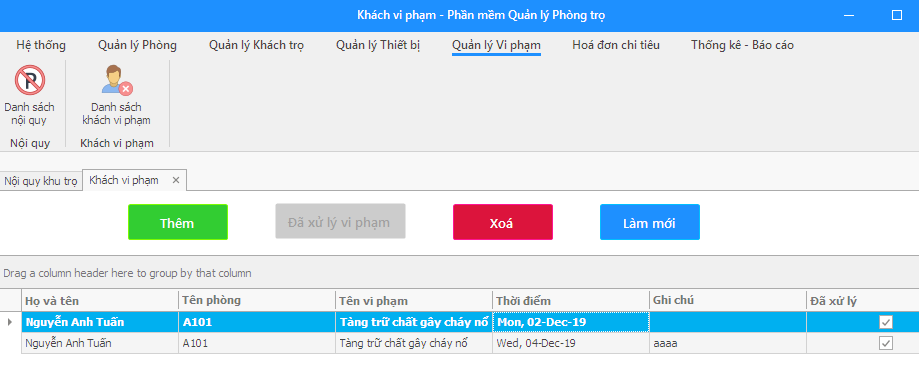
#### Form nội quy



Codes chính:

1. **public** **static** **bool** isAdded = **false**;
2. **public** **static** **bool** isUpdated = **false**;
4. **public** fViPham()
5. {
6. InitializeComponent();
7. }
9. **private** **void** fNoiQuy\_Load(**object** sender, EventArgs e)
10. {
11. LoadData();
12. }
14. **private** **void** LoadData()
15. {
16. DataTable dt = BLL\_VIPHAM.LayDanhSachNoiQuy();
18. **if** (dt.Rows.Count == 0)
19. {
20. **this**.btn\_sua.Enabled = **false**;
21. **this**.btn\_xoa.Enabled = **false**;
22. }
23. **else**
24. {
25. **this**.btn\_sua.Enabled = **true**;
26. **this**.btn\_xoa.Enabled = **true**;
27. }
29. **this**.gridControl\_noiQuy.DataSource = dt;
31. **this**.gridView\_noiQuy.Columns["MAVIPHAM"].Visible = **false**;
32. **this**.gridView\_noiQuy.Columns["NOIDUNG"].Caption = "Tên vi phạm";
33. **this**.gridView\_noiQuy.Columns["HINHPHAT"].Caption = "Hình phạt";
34. }
36. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
37. {
38. XtraForm FormAlter = **new** ChildrenForms.QuanLyViPham.ViPham.fAlter();
39. FormAlter.ShowDialog();
41. **if** (isAdded == **true**)
42. {
43. LoadData();
44. isAdded = **false**;
45. }
46. }
48. **private** **void** btn\_sua\_Click(**object** sender, EventArgs e)
49. {
50. XtraForm FormAlter = **new** ChildrenForms.QuanLyViPham.ViPham.fAlter(LayViPhamHienTai());
51. FormAlter.ShowDialog();
53. **if** (isUpdated == **true**)
54. {
55. LoadData();
56. isUpdated = **false**;
57. }
58. }
60. **private** DTO\_VIPHAM LayViPhamHienTai()
61. {
62. DataRow dr = **this**.gridView\_noiQuy.GetFocusedDataRow();
64. DTO\_VIPHAM viPham = **new** DTO\_VIPHAM
65. {
66. MaViPham = (**string**)dr["MAVIPHAM"],
67. NoiDung = (**string**)dr["NOIDUNG"],
68. HinhPhat = (**string**)dr["HINHPHAT"]
69. };
71. **return** viPham;
72. }
74. **private** **void** btn\_xoa\_Click(**object** sender, EventArgs e)
75. {
76. **try**
77. {
78. DTO\_VIPHAM viPhamHienTai = LayViPhamHienTai();
80. **string** message = String.Format("Xác nhận xoá vi phạm: \n{0} ?", viPhamHienTai.NoiDung);
81. DialogResult dr = XtraMessageBox.Show(message, "Thông báo",
82. MessageBoxButtons.YesNo, MessageBoxIcon.Question);
84. **if** (dr == DialogResult.Yes)
85. **if** (BLL\_VIPHAM.XoaViPham(viPhamHienTai) == **true**)
86. {
87. XtraMessageBox.Show("Xoá vi phạm thành công !", "Thông báo",
88. MessageBoxButtons.OK, MessageBoxIcon.Information);
90. LoadData();
91. }
92. }
93. **catch** (Exception ex)
94. {
96. }
97. }

#### Form quản lý thông tin khách trọ vi phạm



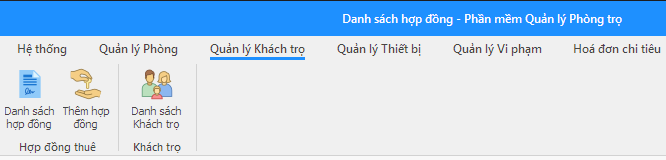
Codes chính:

1. **public** **static** **bool** isAdded = **false**;
3. **public** fKhachViPham()
4. {
5. InitializeComponent();
6. }
8. **private** **void** fKhachViPham\_Load(**object** sender, EventArgs e)
9. {
10. LoadData();
11. }
13. **private** **void** LoadData()
14. {
15. DataTable dt = BLL\_KHACH\_VIPHAM.LayDanhSachKhachViPham();
17. **if** (dt.Rows.Count == 0)
18. {
19. **this**.btn\_xuLy.Enabled = **false**;
20. **this**.btn\_xoa.Enabled = **false**;
21. }
22. **else**
23. {
24. **this**.btn\_xuLy.Enabled = **true**;
25. **this**.btn\_xoa.Enabled = **true**;
26. }
28. **this**.gridControl\_khachViPham.DataSource = dt;
29. **this**.gridView\_khachViPham.Columns["MAKHACH"].Visible = **false**;
30. **this**.gridView\_khachViPham.Columns["MAVIPHAM"].Visible = **false**;
32. **this**.gridView\_khachViPham.Columns["TENPHG"].Caption = "Tên phòng";
33. **this**.gridView\_khachViPham.Columns["NOIDUNG"].Caption = "Tên vi phạm";
34. **this**.gridView\_khachViPham.Columns["HOTEN"].Caption = "Họ và tên";
35. **this**.gridView\_khachViPham.Columns["THOIGIANVIPHAM"].Caption = "Thời điểm";
36. **this**.gridView\_khachViPham.Columns["DAGIAIQUYET"].Caption = "Đã xử lý";
37. **this**.gridView\_khachViPham.Columns["GHICHU"].Caption = "Ghi chú";
39. }
41. **private** **void** btn\_them\_Click(**object** sender, EventArgs e)
42. {
43. XtraForm FormAlter = **new** ChildrenForms.QuanLyViPham.KhachViPham.fAlter();
44. FormAlter.ShowDialog();
46. **if** (isAdded == **true**)
47. {
48. LoadData();
49. isAdded = **false**;
50. }
51. }
53. **private** **void** btn\_xoa\_Click(**object** sender, EventArgs e)
54. {
55. DataRow dr = **this**.gridView\_khachViPham.GetFocusedDataRow();
57. DTO\_KHACH\_VIPHAM khachViPham = **new** DTO\_KHACH\_VIPHAM
58. {
59. MaViPham = (**string**)dr["MAVIPHAM"],
60. MaKhach = (**string**)dr["MAKHACH"],
61. ThoiGianViPham = (DateTime)dr["THOIGIANVIPHAM"]
62. };

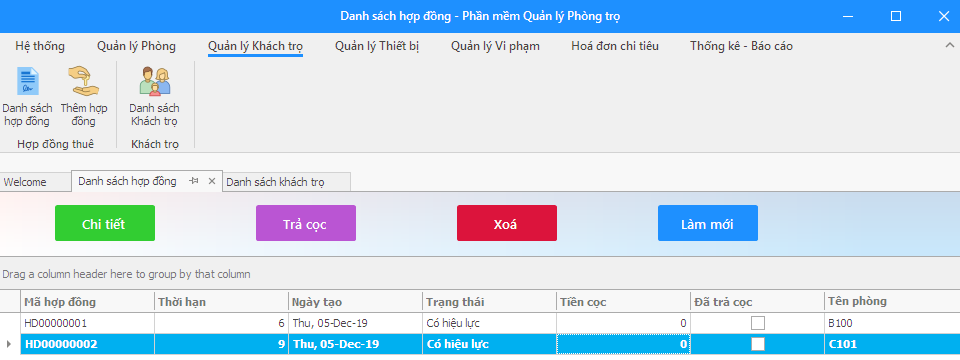
65. **try**
66. {
67. **if** (BLL\_KHACH\_VIPHAM.XoaKhachViPham(khachViPham) == **true**)
68. {
69. XtraMessageBox.Show("Xoá thông tin khách vi phạm thành công !",
70. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
72. LoadData();
73. }
74. }
75. **catch** (Exception ex)
76. {
77. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
78. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
79. }
81. }
83. **private** **void** gridView\_khachViPham\_SelectionChanged(**object** sender, DevExpress.Data.SelectionChangedEventArgs e)
84. {
85. DataRow dr = **this**.gridView\_khachViPham.GetFocusedDataRow();
87. **if** (dr["DAGIAIQUYET"].ToString() == "True")
88. {
89. **this**.btn\_xuLy.Enabled = **false**;
90. }
91. **else**
92. {
93. **this**.btn\_xuLy.Enabled = **true**;
94. }
95. }
97. **private** **void** btn\_xuLy\_Click(**object** sender, EventArgs e)
98. {
99. DataRow dr = **this**.gridView\_khachViPham.GetFocusedDataRow();
101. DTO\_KHACH\_VIPHAM khachViPham = **new** DTO\_KHACH\_VIPHAM
102. {
103. MaViPham = (**string**)dr["MAVIPHAM"],
104. MaKhach = (**string**)dr["MAKHACH"],
105. ThoiGianViPham = (DateTime)dr["THOIGIANVIPHAM"]
106. };

109. **try**
110. {
111. **if** (BLL\_KHACH\_VIPHAM.XuLyViPham(khachViPham) == **true**)
112. {
113. XtraMessageBox.Show("Cập nhật thông tin vi phạm thành công !",
114. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
116. LoadData();
117. }
118. }
119. **catch** (Exception ex)
120. {
121. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
122. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
123. }
124. }

### Nhóm form chức năng quản lý thông tin khách trọ



#### Form danh sách hợp đồng



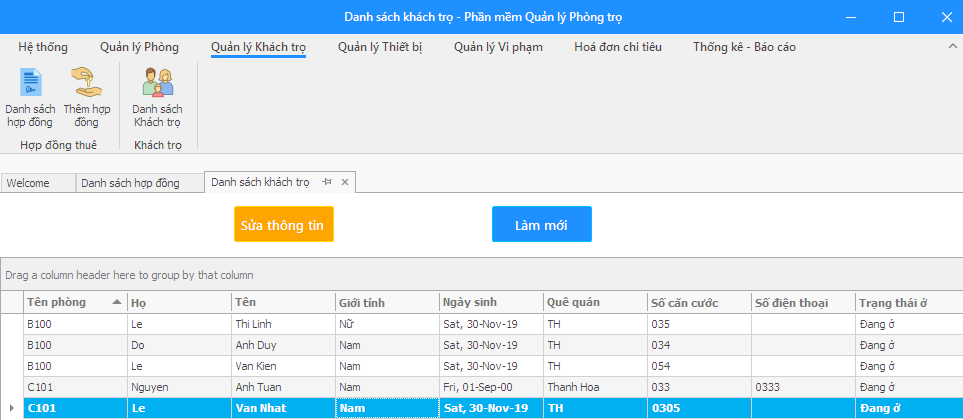
Codes chính:

1. **private** **void** fHopDong\_Load(**object** sender, EventArgs e)
2. {
3. LoadData();
4. }
6. **private** **void** LoadData()
7. {
8. DataTable dt = BLL\_HOPDONG.LayDanhSachHopDong();
9. **if** (dt.Rows.Count == 0)
10. {
11. **this**.btn\_chiTiet.Enabled = **false**;
12. **this**.btn\_traCoc.Enabled = **false**;
13. **this**.btn\_xoa.Enabled = **false**;
14. }
15. **else**
16. {
17. **this**.btn\_chiTiet.Enabled = **true**;
18. **this**.btn\_traCoc.Enabled = **true**;
19. **this**.btn\_xoa.Enabled = **true**;
20. }

23. **this**.gridControl\_hopDong.DataSource = dt;
24. **this**.gridView\_hopDong.Columns["MAPHG"].Visible = **false**;
26. **this**.gridView\_hopDong.Columns["MAHOPDONG"].Caption = "Mã hợp đồng";
27. **this**.gridView\_hopDong.Columns["MAHOPDONG"].VisibleIndex = 0;
28. **this**.gridView\_hopDong.Columns["TENPHG"].Caption = "Tên phòng";
29. **this**.gridView\_hopDong.Columns["THOIHAN"].Caption = "Thời hạn";
30. **this**.gridView\_hopDong.Columns["THOIGIANTAO"].Caption = "Ngày tạo";
31. **this**.gridView\_hopDong.Columns["TRANGTHAI"].Caption = "Trạng thái";
32. **this**.gridView\_hopDong.Columns["TIENCOC"].Caption = "Tiền cọc";
33. **this**.gridView\_hopDong.Columns["DATRACOC"].Caption = "Đã trả cọc";
34. }
36. **private** **void** btn\_lamMoi\_Click(**object** sender, EventArgs e)
37. {
38. LoadData();
39. }
41. **private** **void** btn\_traCoc\_Click(**object** sender, EventArgs e)
42. {
43. DTO\_HOPDONG hopDong = **new** DTO\_HOPDONG();
44. hopDong.MaHopDong = (**string**)(**this**.gridView\_hopDong.GetFocusedDataRow())["MAHOPDONG"];
46. **try**
47. {
48. DialogResult dr = XtraMessageBox.Show("Xác nhận hoàn trả tiền cọc cho khách trọ ?\n" +
49. "Hành động không thể khôi phục.", "Thông báo",
50. MessageBoxButtons.YesNo, MessageBoxIcon.Question);
52. **if** (dr == DialogResult.Yes)
53. {
54. **if** (BLL\_HOPDONG.TraCocHopDong(hopDong) == **true**)
55. {
56. XtraMessageBox.Show("Đã cập nhật thông tin trả cọc !",
57. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
59. LoadData();
60. }
61. }
62. }
63. **catch** (Exception ex)
64. {
65. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
66. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
67. }
68. }
70. **private** **void** gridView\_hopDong\_SelectionChanged(**object** sender, DevExpress.Data.SelectionChangedEventArgs e)
71. {
72. DTO\_HOPDONG hopDong = **new** DTO\_HOPDONG();
73. hopDong.TrangThai = (**string**)(**this**.gridView\_hopDong.GetFocusedDataRow())["TRANGTHAI"];
74. hopDong.DaTraCoc = (**this**.gridView\_hopDong.GetFocusedDataRow())["DATRACOC"].ToString();

77. **if** (hopDong.DaTraCoc == "True")
78. {
79. **this**.btn\_traCoc.Enabled = **false**;
80. }
81. **else**
82. {
83. **this**.btn\_traCoc.Enabled = **true**;
84. }
85. }
87. **private** **void** btn\_xoa\_Click(**object** sender, EventArgs e)
88. {
89. DTO\_HOPDONG hopDong = **new** DTO\_HOPDONG();
90. hopDong.MaHopDong = (**string**)(**this**.gridView\_hopDong.GetFocusedDataRow())["MAHOPDONG"];
92. **try**
93. {
94. DialogResult dr = XtraMessageBox.Show("Xác nhận hợp đồng hết hạn ?\n" +
95. "Hành động không thể khôi phục.", "Thông báo",
96. MessageBoxButtons.YesNo, MessageBoxIcon.Question);
98. **if** (dr == DialogResult.Yes)
99. {
100. **if** (BLL\_HOPDONG.HetHanHopDong(hopDong) == **true**)
101. {
102. XtraMessageBox.Show("Đã cập nhật thông tin hợp đồng !",
103. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
105. LoadData();
106. }
107. }
108. }
109. **catch** (Exception ex)
110. {
111. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:\n" + ex.Message,
112. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
113. }
114. }

#### Form danh sách khách



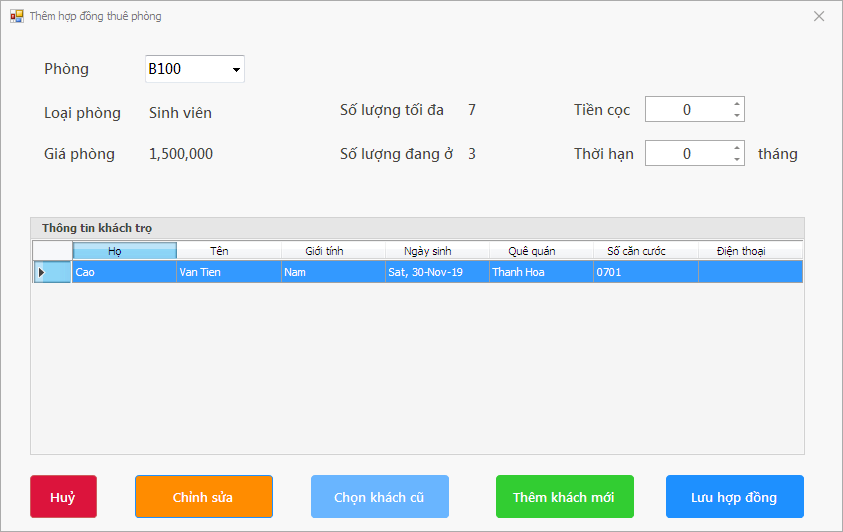
Codes chính:

1. **private** **void** fKhach\_Load(**object** sender, EventArgs e)
2. {
3. LoadData();
4. }
6. **private** **void** LoadData()
7. {
8. DataTable dt = BLL\_KHACH.LayDanhSachKhach();
10. **if** (dt.Rows.Count == 0)
11. {
12. **this**.btn\_sua.Enabled = **false**;
13. }
14. **else**
15. {
16. **this**.btn\_sua.Enabled = **true**;
17. }

20. **this**.gridControl\_khach.DataSource = dt;
22. **this**.gridView\_khach.Columns["MAKHACH"].Visible = **false**;
23. **this**.gridView\_khach.Columns["MAPHG"].Visible = **false**;
25. **this**.gridView\_khach.Columns["HO"].Caption = "Họ";
26. **this**.gridView\_khach.Columns["TEN"].Caption = "Tên";
27. **this**.gridView\_khach.Columns["GIOITINH"].Caption = "Giới tính";
28. **this**.gridView\_khach.Columns["NGAYSINH"].Caption = "Ngày sinh";
29. **this**.gridView\_khach.Columns["QUEQUAN"].Caption = "Quê quán";
30. **this**.gridView\_khach.Columns["SOCANCUOC"].Caption = "Số căn cước";
31. **this**.gridView\_khach.Columns["SODIENTHOAI"].Caption = "Số điện thoại";
32. **this**.gridView\_khach.Columns["TRANGTHAI"].Caption = "Trạng thái ở";
33. **this**.gridView\_khach.Columns["TENPHG"].Caption = "Tên phòng";
34. **this**.gridView\_khach.Columns["TENPHG"].VisibleIndex = 0;
35. }
37. **private** **void** btn\_lamMoi\_Click(**object** sender, EventArgs e)
38. {
39. LoadData();
40. }
42. **private** **void** btn\_sua\_Click(**object** sender, EventArgs e)
43. {
44. DataRow dr = **this**.gridView\_khach.GetFocusedDataRow();
46. DTO\_KHACH khach = **new** DTO\_KHACH
47. {
48. MaKhach = (**string**)dr["MAKHACH"],
49. Ho = (**string**)dr["HO"],
50. Ten = (**string**)dr["TEN"],
51. GioiTinh = (**string**)dr["GIOITINH"],
52. NgaySinh = (DateTime)dr["NGAYSINH"],
53. QueQuan = (**string**)dr["QUEQUAN"],
54. SoCanCuoc = (**string**)dr["SOCANCUOC"],
55. SoDienThoai = (**string**)dr["SODIENTHOAI"],
56. TrangThai = (**string**)dr["TRANGTHAI"]
57. };

60. XtraForm FormAlter = **new** ChildrenForms.KhachTro.DanhSachKhach.fAlter(khach);
61. FormAlter.ShowDialog();
63. **if** (isUpdated == **true**)
64. {
65. LoadData();
66. isUpdated = **false**;
67. }
68. }

#### Form thêm hợp đồng



Codes chính:

1. **public** **static** **bool** isUpdated = **false**;
2. **public** **static** **bool** isAdded = **false**;
3. **public** **static** DTO\_KHACH khachTro;
5. **public** fThemHopDong()
6. {
7. InitializeComponent();
8. khachTro = **new** DTO\_KHACH();
9. }
11. **private** **void** fThemHopDong\_Load(**object** sender, EventArgs e)
12. {
13. LoadPhong();
14. }

17. **private** **void** LoadPhong()
18. {
19. **this**.cb\_phong.DataSource = BLL\_PHONG.LayDanhSachPhongConTrong();
20. **this**.cb\_phong.DisplayMember = "TENPHG";
21. **this**.cb\_phong.ValueMember = "MAPHG";
22. BindData();
23. }
25. **private** **void** BindData()
26. {
27. **this**.lb\_giaPhong.DataBindings.Add("Text", **this**.cb\_phong.DataSource, "GIAPHG", **true**, DataSourceUpdateMode.Never, "", "#,##0");
28. **this**.lb\_loaiPhong.DataBindings.Add("Text", **this**.cb\_phong.DataSource, "TENLOAIPHG");
29. **this**.lb\_soLuongToiDa.DataBindings.Add("Text", **this**.cb\_phong.DataSource, "SOLG\_TOIDA");
30. **this**.lb\_soLuongDangO.DataBindings.Add("Text", **this**.cb\_phong.DataSource, "SOLG\_DANGO");
31. }
33. **private** **void** UnBindData()
34. {
35. **this**.lb\_giaPhong.DataBindings.Clear();
36. **this**.lb\_loaiPhong.DataBindings.Clear();
37. **this**.lb\_soLuongToiDa.DataBindings.Clear();
38. **this**.lb\_soLuongDangO.DataBindings.Clear();
39. }
41. **private** **void** dgv\_thongTinKhach\_RowsRemoved(**object** sender, DataGridViewRowsRemovedEventArgs e)
42. {
43. **if** (**this**.dgv\_thongTinKhach.RowCount == 0)   //  Nếu xoá hết dòng
44. {
45. **this**.btn\_tao.Enabled = **false**;
46. **this**.btn\_chinhSua.Enabled = **false**;
47. }
48. **else**
49. {
50. **this**.btn\_tao.Enabled = **true**;
51. **this**.btn\_chinhSua.Enabled = **true**;
52. }
54. LoadBalance();
55. }

58. **private** List<DTO\_KHACH> LayDanhSachKhach()
59. {
60. List<DTO\_KHACH> listKhach = **new** List<DTO\_KHACH>();

63. **foreach** (DataGridViewRow dr **in** **this**.dgv\_thongTinKhach.Rows)
64. listKhach.Add(**new** DTO\_KHACH
65. {
66. MaKhach = (**string**)dr.Cells["MAKHACH"].Value,
67. Ho = (**string**)dr.Cells["HO"].Value,
68. Ten = (**string**)dr.Cells["TEN"].Value,
69. GioiTinh = (**string**)dr.Cells["GIOITINH"].Value,
70. NgaySinh = DateTime.Parse(dr.Cells["NGAYSINH"].Value.ToString()),
71. QueQuan = (**string**)dr.Cells["QUEQUAN"].Value,
72. SoCanCuoc = (**string**)dr.Cells["CANCUOC"].Value,
73. SoDienThoai = (**string**)dr.Cells["DIENTHOAI"].Value
74. });
76. **return** listKhach;
77. }

80. **private** **void** btn\_chinhSua\_Click(**object** sender, EventArgs e)
81. {
82. DataGridViewRow dr = **this**.dgv\_thongTinKhach.CurrentRow;
83. khachTro.Ho = (**string**)dr.Cells["HO"].Value;
84. khachTro.Ten = (**string**)dr.Cells["TEN"].Value;
85. khachTro.GioiTinh = (**string**)dr.Cells["GIOITINH"].Value;
86. khachTro.NgaySinh = DateTime.Parse(dr.Cells["NGAYSINH"].Value.ToString());
87. khachTro.QueQuan = (**string**)dr.Cells["QUEQUAN"].Value;
88. khachTro.SoCanCuoc = (**string**)dr.Cells["CANCUOC"].Value;
89. khachTro.SoDienThoai = (**string**)dr.Cells["DIENTHOAI"].Value;

92. List<DTO\_KHACH> listKhach = LayDanhSachKhach();
93. **foreach** (DTO\_KHACH kh **in** listKhach)
94. **if** (kh.SoCanCuoc == khachTro.SoCanCuoc)
95. {
96. listKhach.Remove(kh);
97. **break**;
98. }
100. XtraForm FormThemKhach = **new** ChildrenForms.KhachTro.HopDong.ThemHopDong.fThemThongTinKhach(listKhach, **false**, khachTro);
101. FormThemKhach.ShowDialog();

104. **if** (isUpdated == **true**)
105. {
106. dr.Cells["HO"].Value = khachTro.Ho;
107. dr.Cells["TEN"].Value = khachTro.Ten;
108. dr.Cells["GIOITINH"].Value = khachTro.GioiTinh;
109. dr.Cells["NGAYSINH"].Value = khachTro.NgaySinh.ToShortDateString();
110. dr.Cells["QUEQUAN"].Value = khachTro.QueQuan;
111. dr.Cells["CANCUOC"].Value = khachTro.SoCanCuoc;
112. dr.Cells["DIENTHOAI"].Value = khachTro.SoDienThoai;
114. isUpdated = **false**;
115. }
116. }
118. **private** **void** btn\_themKhachMoi\_Click(**object** sender, EventArgs e)
119. {
120. XtraForm FormThemKhach = **new** ChildrenForms.KhachTro.HopDong.ThemHopDong.fThemThongTinKhach(LayDanhSachKhach(), **true**);
121. FormThemKhach.ShowDialog();
123. **if** (isAdded == **true**)
124. {
125. **this**.dgv\_thongTinKhach.Rows.Add
126. (
127. khachTro.MaKhach,
128. khachTro.Ho,
129. khachTro.Ten,
130. khachTro.GioiTinh,
131. khachTro.NgaySinh.ToShortDateString(),
132. khachTro.QueQuan,
133. khachTro.SoCanCuoc,
134. khachTro.SoDienThoai
135. );

138. isAdded = **false**;
139. **this**.btn\_tao.Enabled = **true**;
140. }
141. }
143. **private** **void** btn\_chonKhachCu\_Click(**object** sender, EventArgs e)
144. {
145. List<DTO\_KHACH> listKhach = LayDanhSachKhach();
146. XtraForm FormChonKhachCu = **new** ChildrenForms.KhachTro.HopDong.ThemHopDong.fChonKhachCu(listKhach);
147. FormChonKhachCu.ShowDialog();

150. **if** (isAdded == **true**)
151. {
152. **this**.dgv\_thongTinKhach.Rows.Add
153. (
154. khachTro.MaKhach,
155. khachTro.Ho,
156. khachTro.Ten,
157. khachTro.GioiTinh,
158. khachTro.NgaySinh.ToShortDateString(),
159. khachTro.QueQuan,
160. khachTro.SoCanCuoc,
161. khachTro.SoDienThoai
162. );
164. isAdded = **false**;
165. }
166. }
168. **private** **void** btn\_tao\_Click(**object** sender, EventArgs e)
169. {
170. **if** (**this**.num\_thoiHan.Value <= 0)
171. {
172. XtraMessageBox.Show("Thời hạn không hợp lệ !", "Thông báo",
173. MessageBoxButtons.OK, MessageBoxIcon.Warning);
175. **this**.num\_thoiHan.Focus();
176. **return**;
177. }

180. DTO\_HOPDONG hopDong = **new** DTO\_HOPDONG()
181. {
182. MaHopDong = (**string**)BLL\_HOPDONG.LayMaHopDongTiepTheo(),
183. MaPhong = **this**.cb\_phong.SelectedValue.ToString(),
184. TienCoc = (**int**)**this**.num\_tienCoc.Value,
185. ThoiHan = (**int**)**this**.num\_thoiHan.Value
186. };

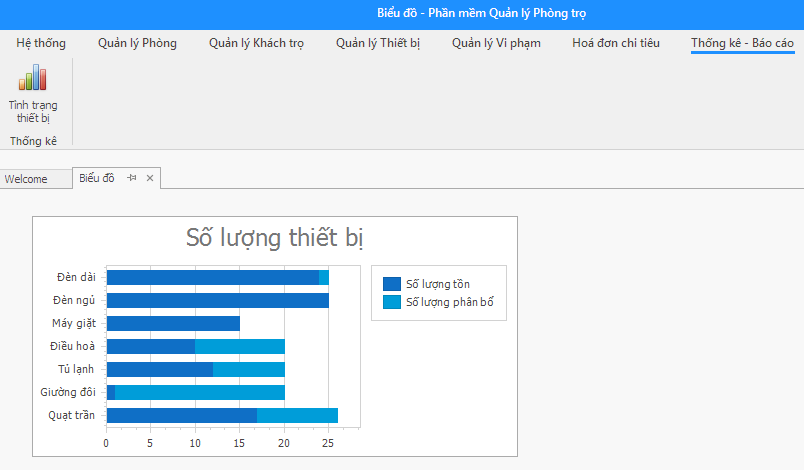
189. **try**         //  INSERT TO HOPDONG
190. {
191. **if** (BLL\_HOPDONG.ThemHopDong(hopDong) == **true**)
192. {
193. XtraMessageBox.Show("Thêm hợp đồng thành công !", "Thông báo",
194. MessageBoxButtons.OK, MessageBoxIcon.Information);

197. List<DTO\_KHACH> listKhach = LayDanhSachKhach();
198. **foreach** (DTO\_KHACH kh **in** listKhach)
199. {
200. **if** (kh.MaKhach == **null**)     //  Khách mới chưa có mã
201. {
202. kh.MaKhach = BLL\_KHACH.LayMaKhachTiepTheo();
203. BLL\_KHACH.ThemKhach(kh);
204. }

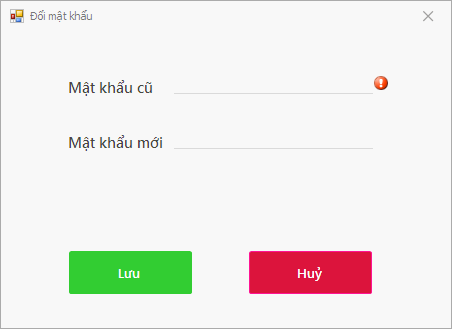
207. DTO\_KHACH\_HOPDONG khd = **new** DTO\_KHACH\_HOPDONG
208. {
209. MaKhach = kh.MaKhach,
210. MaHopDong = hopDong.MaHopDong
211. };
213. BLL\_KHACH\_HOPDONG.ThemKhachHopDong(khd);
214. }
216. **this**.Close();
217. }
218. }
219. **catch** (Exception ex)
220. {
221. XtraMessageBox.Show("Lỗi truy vấn !\n\n" + "Nội dung lỗi:" + ex.Message,
222. "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
223. }
224. }
226. **private** **void** btn\_huy\_Click(**object** sender, EventArgs e)
227. {
228. **this**.Close();
229. }
231. **private** **void** dgv\_thongTinKhach\_SelectionChanged(**object** sender, EventArgs e)
232. {
233. DataGridViewRow dgvr = **this**.dgv\_thongTinKhach.CurrentRow;
235. **if** (dgvr != **null**)
236. {
237. **if** (dgvr.Cells["MAKHACH"].Value != **null**)    //  Khách cũ
238. **this**.btn\_chinhSua.Enabled = **false**;
239. **else**
240. **this**.btn\_chinhSua.Enabled = **true**;       //  Khách mới
241. }
242. }
244. **private** **void** cb\_phong\_SelectedIndexChanged(**object** sender, EventArgs e)
245. {
247. }
249. **private** **void** dgv\_thongTinKhach\_RowsAdded(**object** sender, DataGridViewRowsAddedEventArgs e)
250. {
251. **this**.btn\_tao.Enabled = **true**;
252. LoadBalance();
253. }

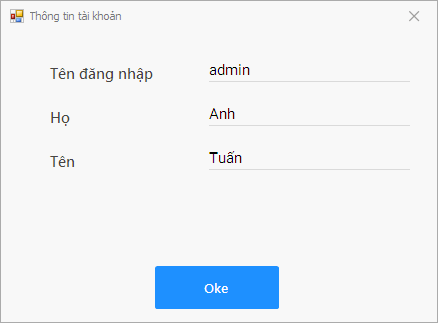
256. **private** **void** LoadBalance()
257. {
258. **int** balance = **int**.Parse(lb\_soLuongToiDa.Text) - **int**.Parse(lb\_soLuongDangO.Text);
260. **if** (**this**.dgv\_thongTinKhach.RowCount == balance)
261. {
262. **this**.btn\_themKhachMoi.Enabled = **false**;
263. **this**.btn\_chonKhachCu.Enabled = **false**;
264. }
265. **else**
266. {
267. **this**.btn\_themKhachMoi.Enabled = **true**;
268. **this**.btn\_chonKhachCu.Enabled = **true**;
269. }
270. }

### Nhóm form chức năng thống kê



### Nhóm form chức năng quản lý thông tin tài khoản





# KẾT LUẬN

Do thời gian có hạn nên hiện tại em chỉ có thể try hard được bấy nhiêu thôi. Nhưng trong tương lai, trong các môn học tiếp theo em sẽ vận dụng các kiến thức để hoàn thiện đề tài của mình.

Em cảm ơn cô và các bạn rất nhiều khi đã tin tưởng và làm cho em thấy rằng mình có được vị trí nào đó…