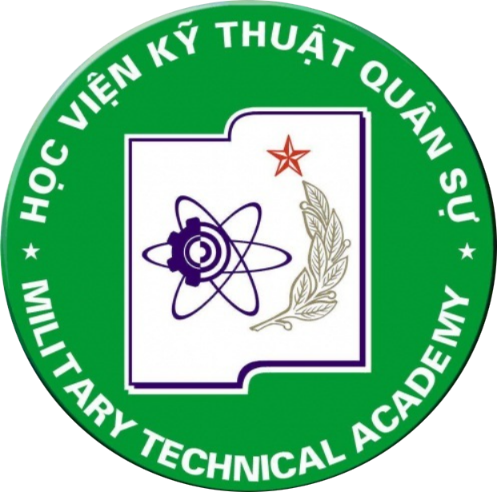


**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**KHOA CÔNG NGHỆ THÔNG TIN**

\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*



**BÁO CÁO**

**MÔN THỰC TẬP CƠ SỞ DỮ LIỆU**

***Đề tài: Quản lý phòng thí nghiệm***

**Giảng viên : Chu Thị Hường**

**Sinh viên thực hiện : Hà Mạnh Duy (BĐATTT)**

**Chu Thị Linh (BĐATTT)**

**Lê Thị Minh Châu (ANHTTT)**

**Lớp : BDATTT54 + ANHTTT54**

Hà Nội, Tháng 02/2021

MỤC LỤC

[CHƯƠNG I. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN 4](#_Toc73433800)

[1. Mô tả bài toán: 4](#_Toc73433801)

[2. Quy trình xử lý hệ thống 5](#_Toc73433802)

[3. Quy trình nghiệp vụ 5](#_Toc73433803)

[4. Một số biểu mẫu 7](#_Toc73433804)

[CHƯƠNG II. THIẾT KẾ CƠ SỞ DỮ LIỆU 8](#_Toc73433805)

[1. Xây dựng mô hình thực thể liên kết ER cho hệ thống đã mô tả. 8](#_Toc73433806)

[2. Xây dựng mô hình quan hệ từ mô hình ER 12](#_Toc73433807)

[3. Đặc tả các bảng 18](#_Toc73433808)

[CHƯƠNG III. KHAI THÁC DỮ LIỆU VÀ LẬP TRÌNH ỨNG DỤNG 23](#_Toc73433809)

[1. Mô tả tổng quan chương trình ứng dụng. 23](#_Toc73433810)

[2. Đăng nhập 24](#_Toc73433811)

[3. Đổi mật khẩu 27](#_Toc73433812)

[4. Quản lý thông tin tài khoản 28](#_Toc73433813)

[5. Quản lý nhân sự: Nhân viên 31](#_Toc73433814)

[6. Quản lý nhân sự: Giáo viên 35](#_Toc73433815)

[7. Quản lý hệ thống Phòng thí nghiệm 40](#_Toc73433816)

[8. Quản lý hệ thống Trang bị 45](#_Toc73433817)

[9. Quản lý các Bài thí nghiệm 57](#_Toc73433818)

[10. Quản lý việc đăng ký sử dụng Phòng thí nghiệm 65](#_Toc73433819)

[11. Quản lý các ca sử dụng Phòng thí nghiệm 75](#_Toc73433820)

[12. Thống kê theo Phòng thí nghiệm 81](#_Toc73433821)

[13. Thống kê theo Loại trang bị 85](#_Toc73433822)

[**Tổng kết và đánh giá. 89**](#_Toc73433823)

**DANH MỤC KÝ HIỆU CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **CHỮ VIẾT ĐẦY ĐỦ** | **CHỮ VIẾT TẮT** |
| Phòng thí nghiệm | PTN |
| Trang thiết bị | TTB |
| Cơ sở dữ liệu | CSDL |
| Bài thí nghiệm | BTN |
| Ca sử dụng | CSD |
| Kỹ thuật viên | KTV |
| Tên đăng nhập | TDN |
| Mật khẩu | MK |

**CHƯƠNG I. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN**

**1. Mô tả bài toán:**

Các phòng thí nghiệm của *Học viện Kĩ Thuật Quân sự* cần quản lý việc sử dụng phòng thí nghiệm cho các bài thí nghiệm, thực hành cho một môn học (phòng thí nghiệm có thể phục vụ cho các bài thí nghiệm, bài thực hành của các lớp học; việc nghiên cứu của các nhóm nghiên cứu khoa học; hay các học viên nghiên cứu cho bài luận án .., tuy nghiên trong phạm vi bài toán này chỉ giải quyết việc mảng thí nghiệm, thực hành của các lớp học). Sau đây là phần mô tả theo các nghiệp vụ hàng ngày của phòng thí nghiệm:

**Phòng thí nghiệm** có các **nhân viên** để quản lý hoạt động chung của phòng. Các thông tin cần lưu trữ của nhân viên quản lý: *mã nhân viên, họ tên, chức vụ, hình ảnh, giới tính, số điện thoại, ngày sinh.*

Phòng thí nghiệm được trang bị các **loại trang thiết bị** cần thiết để phục vụ cho học tập và nghiên cứu. Mỗi trang thiết bị có: *mã loại trang thiết bị, tên loại trang thiết bị.*

Mỗi loại trang thiết bị có nhiều **trang thiết bị** để bảo đảm về số lượng và chất lượng. Mỗi trang thiết bị được quản lý bởi: *mã trang thiết bị, tên trang thiết bị* (tên trang thiết bị có thể trùng với tên loại trang thiết bị)*, nhà sản xuất, trình trạng, giá tiền (*đồng*).*

Các trang thiết bị được trang bị để phục vụ cho các bài thí nghiệm hoặc bài thực hành, gọi chung là **bài thí nghiệm**. Mỗi bài thí nghiệm có các thông tin: *mã bài thí nghiệm, tên bài thí nghiệm và chi tiết bài thí nghiệm****.***

Các **giáo viên** sẽ đảm nhiệm hướng dẫn bài thí nghiệm. Với thông tin: *Mã giáo viên, tên giáo viên, hình ảnh, giới tính, số điện thoại, ngày sinh.*

* 1. ***Đăng kí sử dụng phòng thí nghiệm:***

Đầu mỗi học kỳ, dựa trên kế hoạch giảng dạy, giáo viên phải đăng kí với phòng thí nghiệm để các lớp học được sử dụng phòng thí nghiệm theo yêu cầu môn học.

Nội dung đăng kí với phòng thí nghiệm gồm các thông tin: năm học, học kỳ, lớp học, quân số lớp.

Trên cơ sở đề nghị của các giáo viên; nhân viên quản lý kiểm tra tình trạng của các trang thiết bị hiện có để chuẩn bị đầy đủ về chất lượng và số lượng trang thiết bị.

* 1. ***Ghi chép ca sử dụng phòng thí nghiệm:***

Mỗi lần mà lớp học sử dụng phòng thí nghiệm dưới sự hướng dẫn của giáo viên thì phải ghi chép thông tin ca sử dụng phòng thí nghiệm.

Việc ghi chép phải đầy đủ các thông tin: *giáo viên hướng dẫn, nhân viên quản lý, lớp học, quân số thí nghiệm, ngày thí nghiệm, ca trong ngày, bài thí nghiệm, chi tiết sử dụng trang thiết bị,* …

**2. Quy trình xử lý hệ thống**

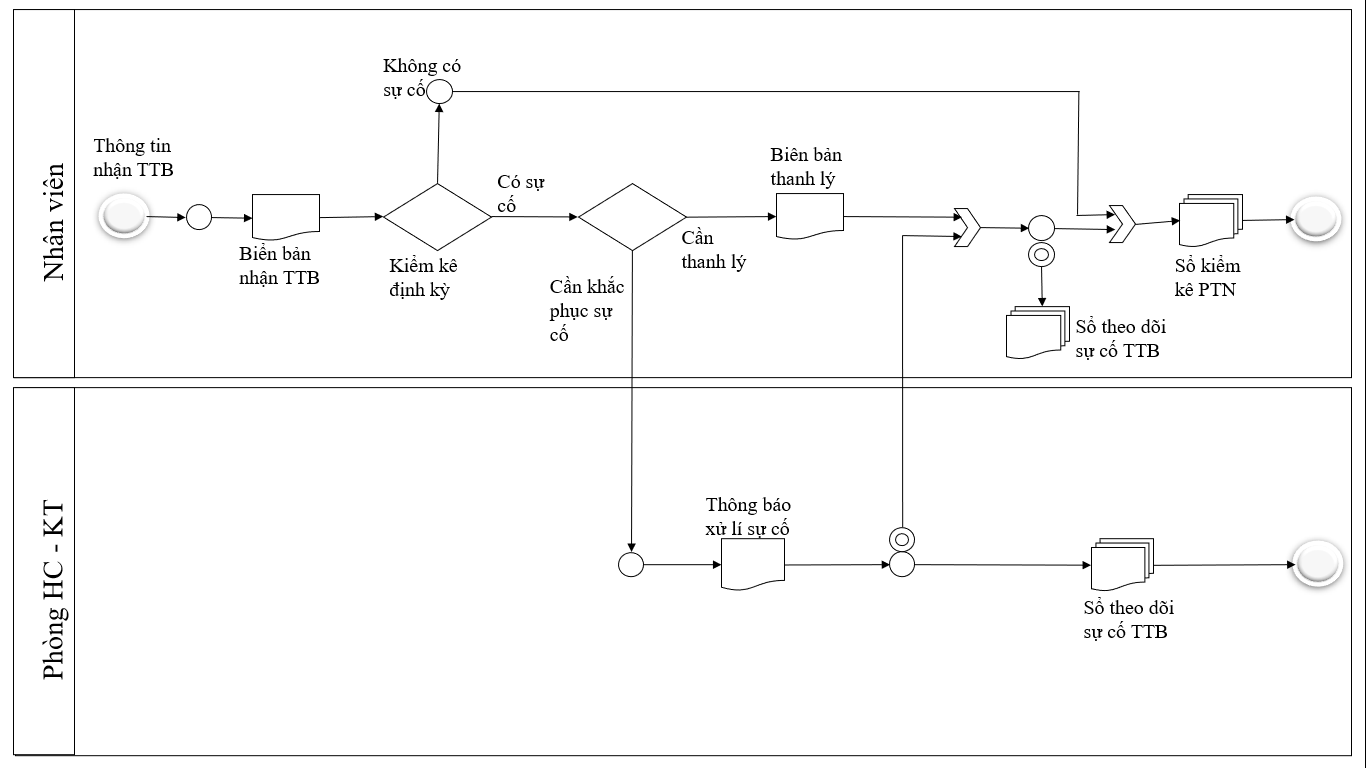
− Tiến trình 1: Quản lý theo dõi các phòng thí nghiệm và trang thiết bị.

− Tiến trình 2: Theo dõi đăng kí lịch thực hành.

− Tiến trình 3: Theo dõi sử dụng PTN.

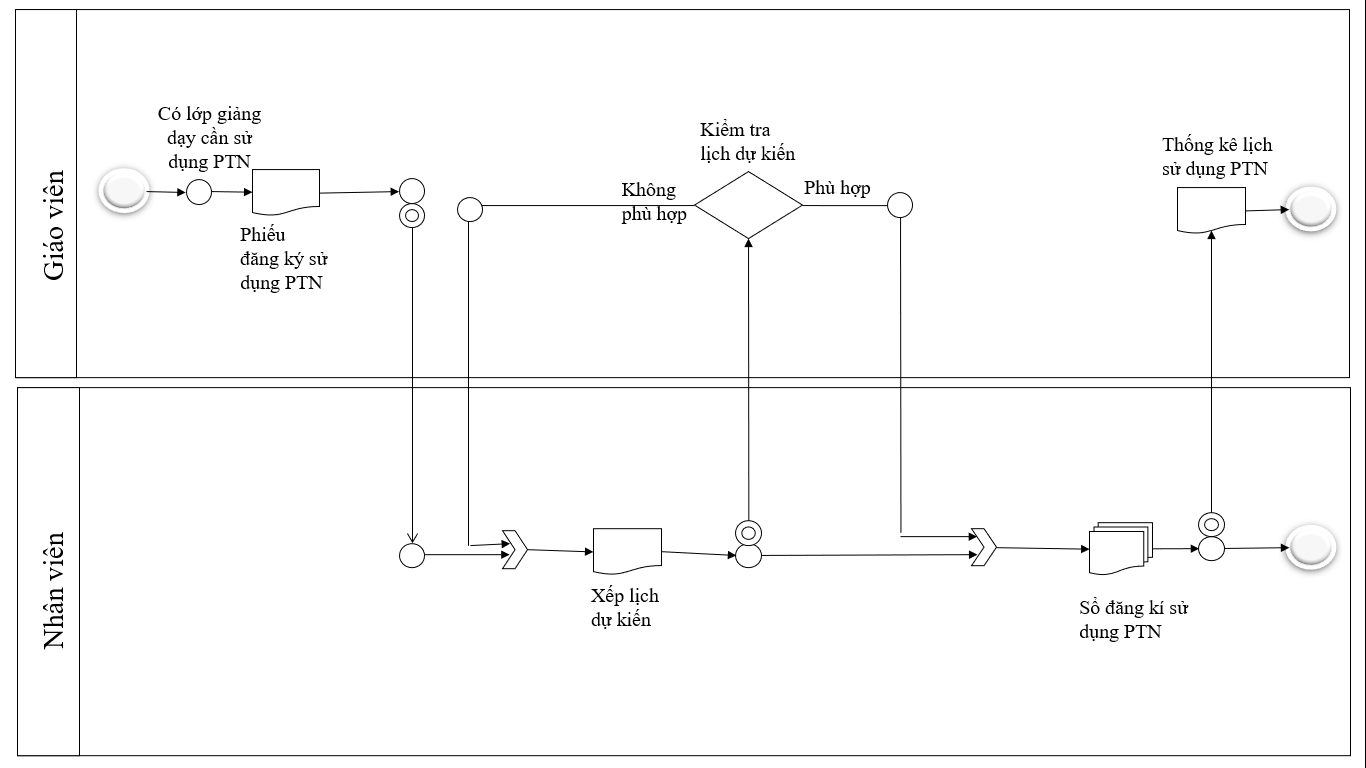
**3. Quy trình nghiệp vụ**

*3.1. Quản lý theo dõi các phòng thí nghiệm và trang thiết bị.*



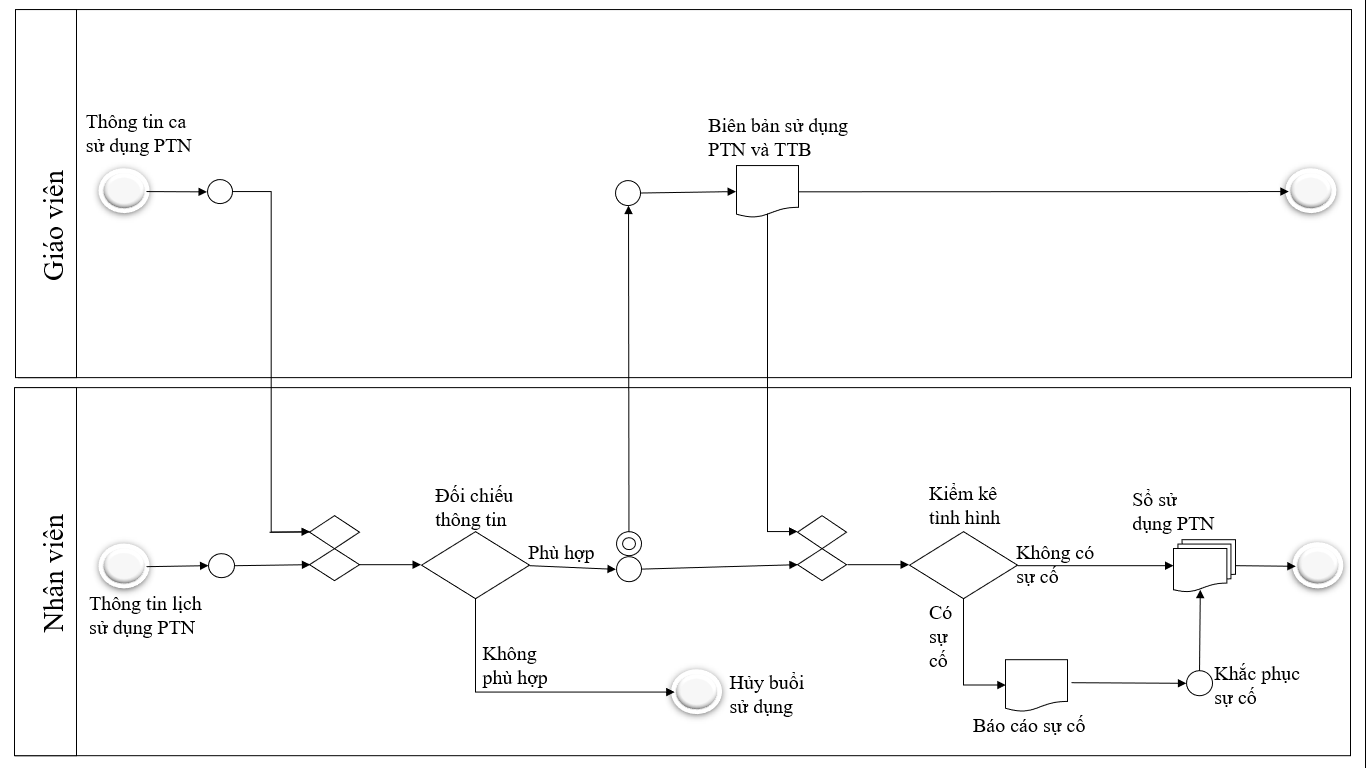
* Khi nhập TTB về hoặc khi được tặng TTB thì PTN có nhiệm vụ lấy thông tin về TTB để hoàn thiện biên bản nhận TTB và lưu trữ vào sổ theo dõi TTB.
* TTB trong PTN được kiểm kê theo định kỳ để đảm bảo về số lượng và chất lượng. Nếu kiểm kê TTB đầy đủ về số lượng và chất lượng thì ghi chép vào sổ kiểm kê PTN.
* Nếu TTB chưa đảm bảo về số lượng thì cần mua bổ sung. Nếu TTB chưa đảm bảo về chất lượng thì cần sửa chưa, bảo dưỡng.

*3.2. Đăng ký lịch thực hành.*



* Đầu mỗi học kỳ, mỗi giáo viên có kế hoạch giảng dạy các lớp mà nội dung bài học cần sử dụng PTN sẽ phải đăng ký với phòng thí nghiệm để sắp xếp lịch.
* Giáo viên đăng ký lịch thí nghiệm dự kiến theo các bài thí nghiệm.
* Nhân viên phòng thí nghiệm cần kiểm tra xem lịch sự kiến mà các giáo viên đăng kí có bị trùng nhau hay không. Nếu lịch trùng nhau, thì phải thay đổi lịch hoặc thay đổi phòng hoặc hủy lịch đó đi và bổ sung vào lịch mới.
* Khi chưa tiến hành thí nghiệm, giáo viên có thể thay đổi lịch đăng ký để phụ hợp với yêu cầu giảng dạy.

*3.3. Theo dõi sử dụng PTN*



* Khi đến ca sử dụng PTN, giáo viên và lớp có mặt ở PTN. Nhân viên PTN kiểm tra thông tin ca sử dụng của giáo viên có hợp lệ với nội dung trong sổ đăng ký sử dụng PTN hay không.
* Nếu không phù hợp thì hoãn hoặc hủy buổi sử dụng. Nếu phù hợp thì nhân viên và giáo viên ký xác nhận bàn giao PTN và trang thiết bị.
* Sau ca sử dụng PTN, giáo viên báo cáo tình hình sử dụng PTN và trang thiết bị để nhân viên kiểm tra lại.
* Sau khi xác nhận về tình hình ca sử dụng PTN thì nhân viên ghi chép vào sổ sử dụng PTN, có xác nhận của giáo viên.

4. Một số biểu mẫu

* 1. *Biểu mẫu 1*

|  |  |  |
| --- | --- | --- |
|  | Phiếu đăng ký sử dụng Phòng Thí Nghiệm Vật lý đại cương 2 | |
| Họ tên giáo viên: ............................................... | | Bài thí nghiệm: .......................................... |
| Lớp: ................................................................... | | Quân số thí nghiệm: .................................. |
| Năm học:............................. Học kỳ: ................. | | Kíp: ............................................................ |

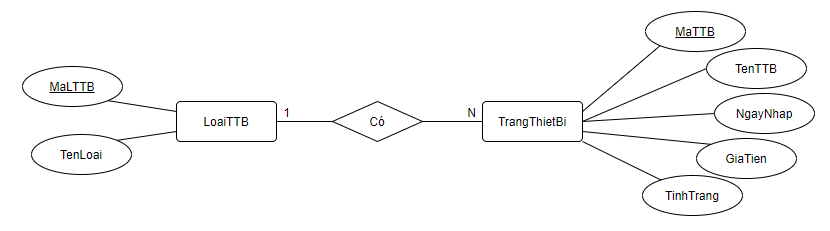
* 1. *Biểu mẫu 2*

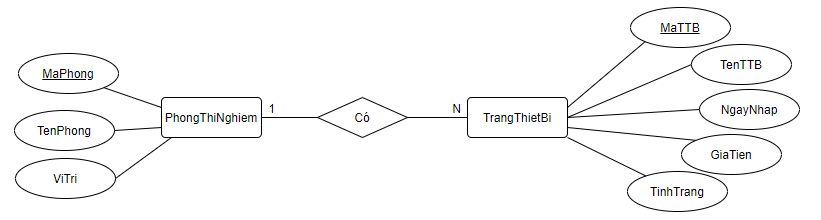
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lịch dự kiến Phòng thí nghiệm Vật Lý đại cương 2** | | | | | | | | |
| **Thời gian** | | | **Lớp** | **Số SV** | **Tình trạng trang thiết bị** | **GV hướng dẫn** | **Ghi chú** |
| **Ngày** | **Kíp** | **Ca** |
| Thứ 2  22/03/2021 | Sáng | 1 | BĐ54 | 20 | Đảm bảo | TS Hà Mạnh Duy |  |
| 2 |  |  |  |  |  |
| Chiều | 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

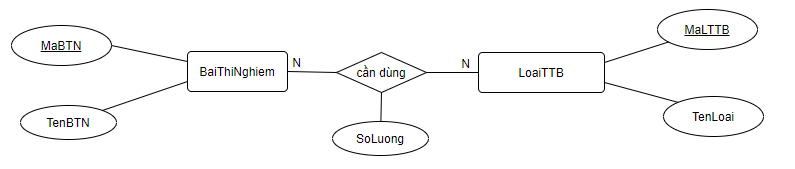
CHƯƠNG II. THIẾT KẾ CƠ SỞ DỮ LIỆU

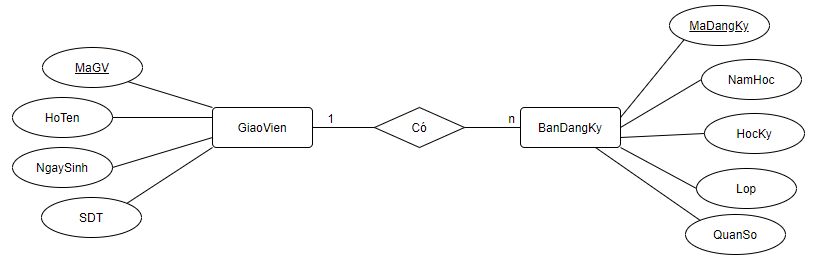
1. Xây dựng mô hình thực thể liên kết ER cho hệ thống đã mô tả.

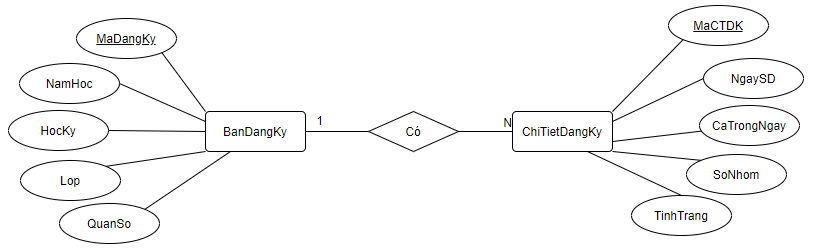
* 1. *Xác định các tập thực thể (thuộc tính khóa được gạch chân)*
* PhongThiNghiem (MaPhong, TenPhong, ViTri).
* LoaiTTB (MaLTTB, TenLoai).
* TrangThietBi (MaTTB, TenTTB, NgayNhap, GiaTien, TinhTrang).
* BaiThiNghiem (MaBTN, TenBTN).
* NhanVien (MaNV, HoTen, NgaySinh, SDT).
* GiaoVien (MaGV, HoTen, NgaySinh, SDT).
* BanDangKy (MaDangKy, Lop, QuanSo, NamHoc, HocKy).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang). (Tình trạng là đã thực hiện, chưa thực hiện hoặc đã hủy)
  1. *Các mối liên kết:*
* LoaiTTB có TrangThietBi.
* PhongThiNghiem có TrangThietBi.
* BaiThiNghiem cần dùng LoaiTTB.
* GiaoVien có BanDangKy.
* BanDangKy có ChiTietDangKy.
* NhanVien quản lý ChiTietDangKy.
* ChiTietDangKy thuộc PhongThiNghiem.
* ChiTietDangKy thuộc BaiThiNghiem.
* ChiTietDangKy sử dụng TrangThietBi.
  1. *Xác định mối quan hệ của các thực thể*

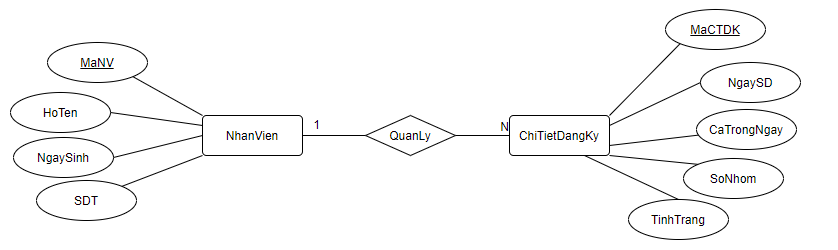


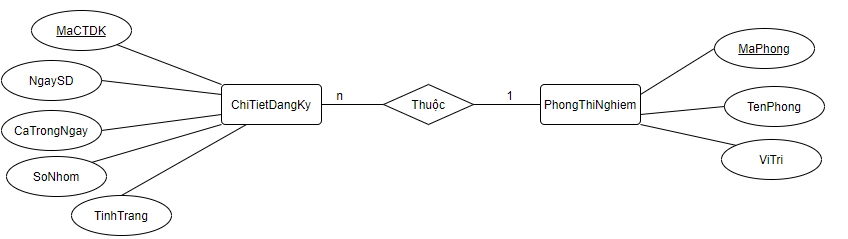


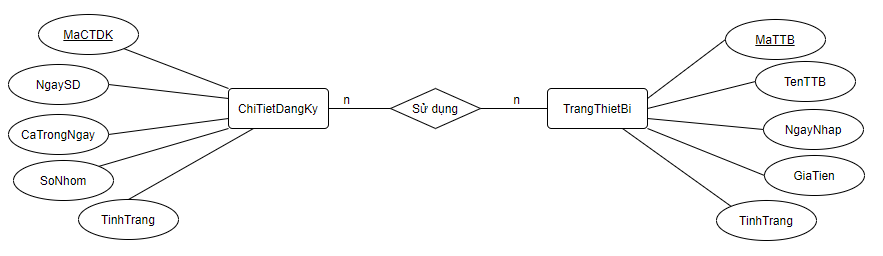
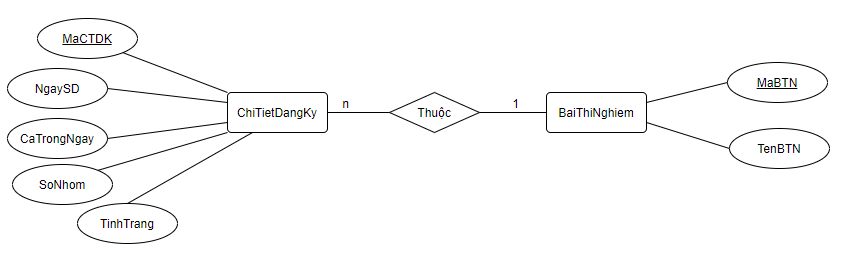




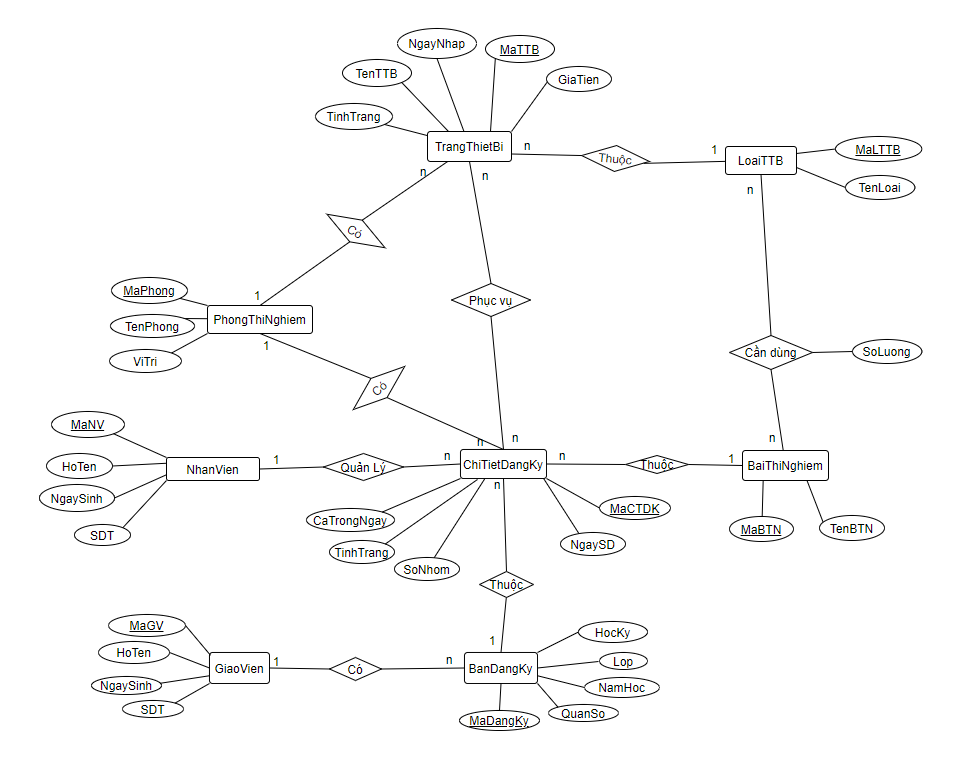








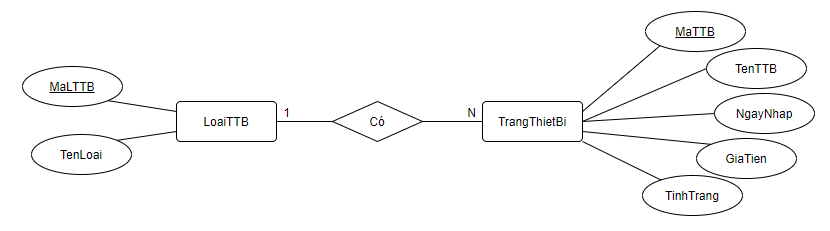
* 1. *Mô hình thực thể liên kết ER*



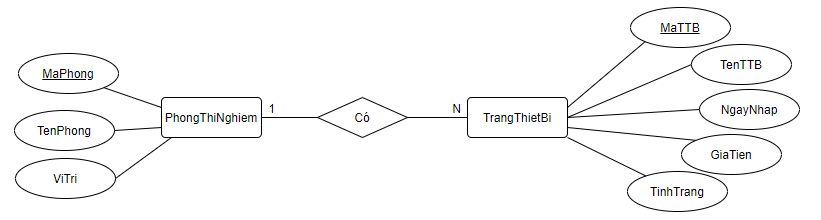
2. Xây dựng mô hình quan hệ từ mô hình ER

2.1. *Chuyển từ mô hình ER sang mô hình quan hệ:*

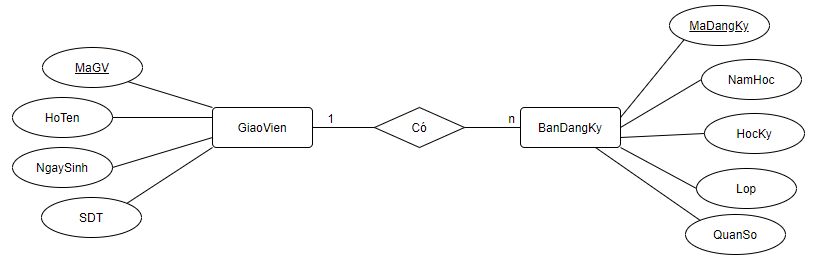
* *Xử lý liên kết 1 – 1:* Chuyển khóa chính của LĐQH này sang làm khóa ngoại của LĐQH kia
* *Xử lý liên kết 1 – n :* Chuyển khóa chính của bên 1 sang làm khóa ngoại của bên n:



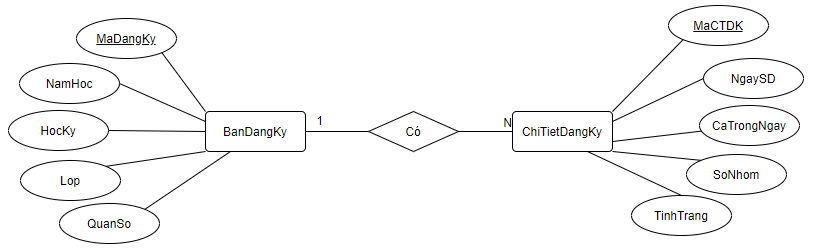
* LoaiTTB (MaLTTB, TenLoai).
* TrangThietBi (MaTTB, TenTTB, NgayNhap, GiaTien, TinhTrang, MaLTTB).



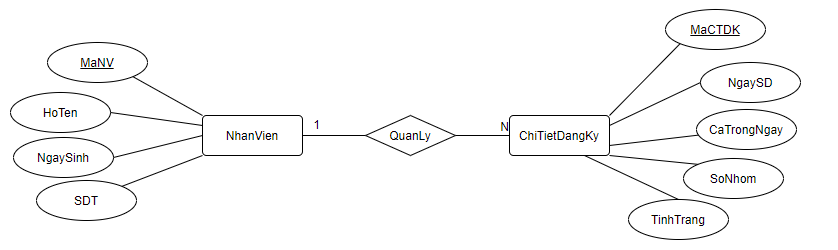
* PhongThiNghiem (MaPhong, TenPhong, ViTri).
* TrangThietBi (MaTTB, TenTTB, NgayNhap, GiaTien, TinhTrang, MaPhong).



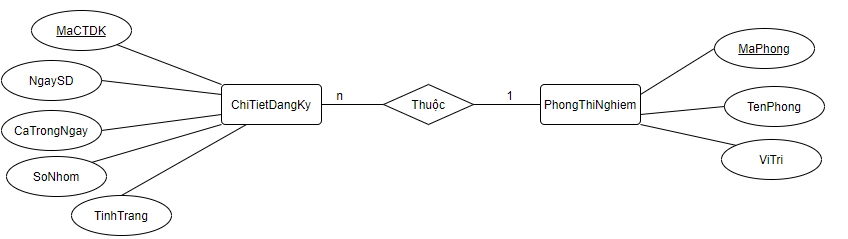
* GiaoVien (MaGV, HoTen, NgaySinh, SDT).
* BanDangKy (MaDangKy, Lop, QuanSo, NamHoc, HocKy, MaGV).



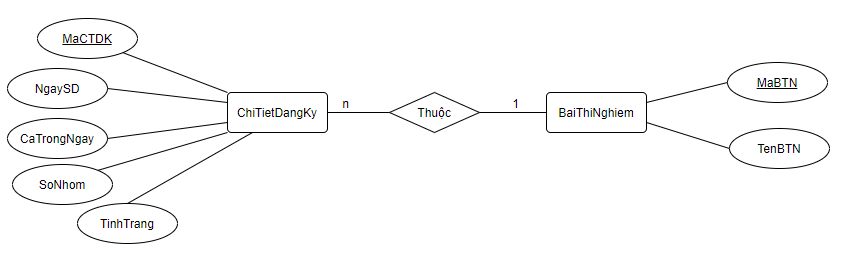
* BanDangKy (MaDangKy, Lop, QuanSo, NamHoc, HocKy).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaDangKy).



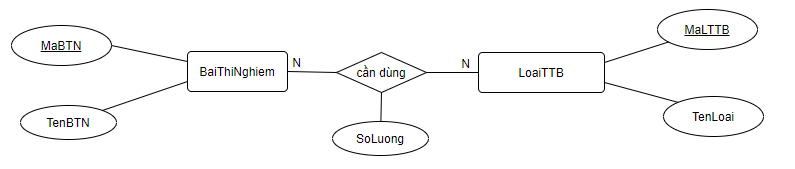
* NhanVien (MaNV, HoTen, NgaySinh, SDT).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaNV).



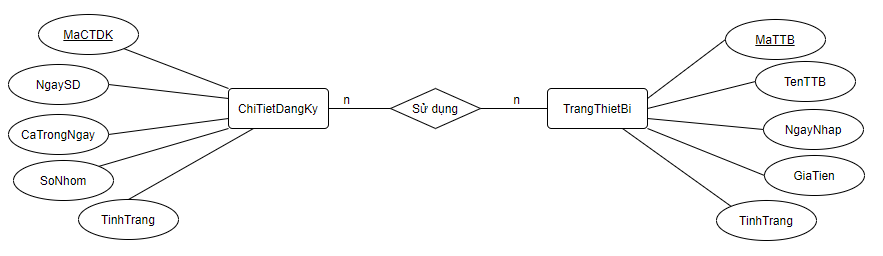
* PhongThiNghiem (MaPhong, TenPhong, ViTri).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaPhong).



* BaiThiNghiem (MaBTN, TenBTN).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaBTN).
* *Xử lý liên kết n – n:* Chuyển khóa chính của 2 LĐQH thành khóa ngoại của 1 LĐQH mới.



* LoaiTTB (MaLTTB, TenLoai).
* BaiThiNghiem (MaBTN, TenBTN).
* ChiTietTTBCanDung (MaChiTiet, SoLuong, MaLTTB, MaBTN)



* TrangThietBi (MaTTB, TenTTB, NgayNhap, GiaTien, TinhTrang).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang).
* ChiTietSuDung (MaCTSD, MaTTB, MaCTDK).

*2.2. Lược đồ cơ sở dữ liệu quan hệ:*

* PhongThiNghiem (MaPhong, TenPhong, ViTri).
* LoaiTTB (MaLTTB, TenLoai).
* TrangThietBi (MaTTB, TenTTB, NgayNhap, GiaTien, TinhTrang, MaLTTB, MaPhong).
* BaiThiNghiem (MaBTN, TenBTN).
* NhanVien (MaNV, HoTen, NgaySinh, SDT).
* GiaoVien (MaGV, HoTen, NgaySinh, SDT).
* BanDangKy (MaDangKy, Lop, QuanSo, NamHoc, HocKy, MaGV).
* ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaDangKy, MaNV, MaPhong, MaBTN).
* ChiTietTTBCanDung (MaChiTiet, SoLuong, MaLTTB, MaBTN)
* ChiTietSuDung (MaCTSD, MaTTB, MaCTDK).

*2.3. Đánh giá dạng chuẩn:*

1. PhongThiNghiem (MaPhong, TenPhong, ViTri).

* MaPhong -> TenPhong, ViTri
* Dạng chuẩn: 3NF

1. LoaiTTB (MaLTTB, TenLoai).

* MaLTTB -> TenLoai
* Dạng chuẩn: 3NF

1. TrangThietBi (MaTTB, TenTTB, NgayNhap, GiaTien, TinhTrang, MaLTTB, MaPhong)

* MaTTB -> TenTTB, NgayNhap, GiaTien, TinhTrang, MaLTTB, MaPhong
* Dạng chuẩn: 3NF

1. BaiThiNghiem (MaBTN, TenBTN).

* MaBTN -> TenBTN
* Dạng chuẩn: 3NF

1. NhanVien (MaNV, HoTen, NgaySinh, SDT).

* MaNV -> HoTen, NgaySinh, SDT
* Dạng chuẩn: 3NF

1. GiaoVien (MaGV, HoTen, NgaySinh, SDT).

* MaGV -> HoTen, NgaySinh, SDT
* Dạng chuẩn: 3NF

1. BanDangKy (MaDangKy, QuanSo, NamHoc, HocKy, Lop, MaGV).

* MaDangKy -> QuanSo, NamHoc, HocKy, Lop, MaGV
* Dạng chuẩn: 3NF

1. ChiTietDangKy (MaCTDK, NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaDangKy, MaNV, MaPhong, MaBTN).

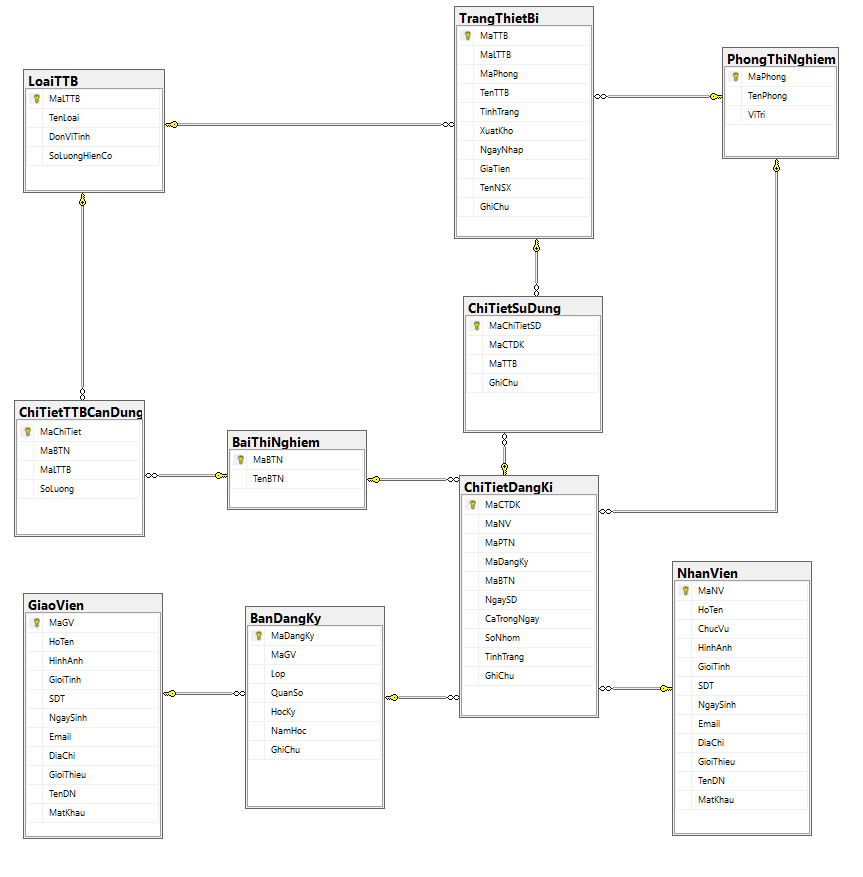
* MaCTDK -> NgaySD, CaTrongNgay, SoNhom, TinhTrang, MaDangKy, MaNV, MaPhong, MaBTN
* Dạng chuẩn: 3NF

1. ChiTietTTBCanDung (MaChiTiet, SoLuong, MaBTN, MaLTTB)

* MaChiTiet -> SoLuong, MaBTN, MaLTTB
* Dạng chuẩn: 3NF

1. ChiTietSuDung (MaChiTiet, MaCTDK, MaTTB)

* MaChiTiet -> MaCTDK, MaTTB
* Dạng chuẩn: 3NF

*2.4. Mô hình quan hệ:* 

3. Đặc tả các bảng

* ***Tổng gồm 10 bảng:***

PhongThiNghiem, LoaiTTB, TrangThietBi, BaiThiNghiem, NhanVien, GiaoVien, BanDangKy, ChiTietDangKy, ChiTietTTBCanDung, ChiTietSuDung.

|  |
| --- |
| ***PhongThiNghiem*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaPhong | varchar(10) |  | Primary key | Mã phòng thí nghiệm |
| TenPhong | nvarchar(50) |  |  | Tên phòng thí nghiệm |
| ViTri | nvarchar(50) |  |  | Vị trí |

|  |
| --- |
| ***LoaiTTB*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaLTTB | varchar(10) |  | Primary key | Mã loại trang thiết bị |
| TenLoai | nvarchar(50) |  |  | Tên loại trang thiết bị |
| DonViTinh | nvarchar(50) |  |  | Đơn vị tính |

|  |
| --- |
| ***TrangThietBi*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaTTB | varchar(10) |  | Primary key | Mã trang thiết bị |
| MaLTTB | varchar(10) |  | Foreign key | Mã loại trang thiết bị |
| MaPhong | varchar(10) |  | Foreign key | Mã phòng thí nghiệm |
| TenTTB | nvarchar(100) | 🗸 |  | Tên trang thiết bị |
| NgayNhap | date | 🗸 |  | Ngày nhập |
| GiaTien | int | 🗸 |  | Giá tiền (đồng) |
| TinhTrang | nvarchar(50) |  |  | Tình trạng hiện tại |
| XuatKho | bit |  |  | = 1: Đã xuất kho  = 0: Chưa xuất kho |
| GhiChu | nvarchar(100) | 🗸 |  | Ghi chú |

|  |
| --- |
| ***BaiThiNghiem*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaBTN | varchar(10) |  | Primary key | Mã bài thí nghiệm |
| TenBTN | nvarchar(100) |  |  | Tên bài thí nghiệm |

|  |
| --- |
| ***NhanVien*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaNV | varchar(10) |  | Primary key | Mã nhân viên |
| ChucVu | nvarchar(20) |  |  | chức vụ |
| HoTen | nvarchar(50) |  |  | Họ tên |
| NgaySinh | date | 🗸 |  | Ngày sinh |
| SDT | varchar(11) | 🗸 |  | Số điện thoại |
| TenDN | varchar(20) | 🗸 |  | Tên đăng nhập |
| MatKhau | varchar(20) | 🗸 |  | Mật khẩu |

|  |
| --- |
| ***GiaoVien*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaGV | varchar(10) |  | Primary key | Mã giáo viên |
| Hoten | nvarchar(50) |  |  | Họ tên giáo viên |
| NgaySinh | date | 🗸 |  | Ngày sinh |
| SDT | varchar(11) | 🗸 |  | Số điện thoại |
| TenDN | varchar(20) | 🗸 |  | Tên đăng nhập |
| MatKhau | varchar(20) | 🗸 |  | Mật khẩu |

|  |
| --- |
| ***BanDangKy*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaDangKy | varchar(10) |  | Primary key | Mã đăng ký |
| MaGV | varchar(10) |  | Foreign key | Mã giáo viên |
| Lop | nvarchar(50) |  |  | Lớp |
| QuanSo | int |  |  | Quân số thí nghiệm |
| Kip | nvarchar(50) |  |  | Kíp học |
| HocKy | int |  |  | Học kỳ |
| NamHoc | nvarchar(50) |  |  | Năm học |
| GhiChu | nvarchar(50) | 🗸 |  | Ghi Chú |

|  |
| --- |
| ***ChiTietTTBCanDung*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaChiTiet | varchar(10) |  | Primary key | Mã chi tiết trang thiết bị cần dùng |
| MaBTN | varchar(10) |  | Foreign key | Mã bài thí nghiệm |
| MaLTTB | varchar(10) |  | Foreign key | Mã loại trang thiết bị |
| SoLuong | int |  |  | Số lượng cần dùng |

|  |
| --- |
| ***ChiTietDangKy*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaCSD | varchar(10) |  | Primary key | Mã ca sử dụng |
| MaNV | varchar(10) |  | Foreign key | Mã nhân viên |
| MaPhong | varchar(10) |  | Foreign key | Mã phòng thí nghiệm |
| MaDangKy | varchar(10) |  | Foreign key | Mã bản đăng ký |
| MaBTN | varchar(10) |  | Foreign key | Mã bài thí nghiệm |
| NgaySD | date |  |  | Ngày sử dụng |
| CaTrongNgay | Int |  |  | Ca trong ngày |
| SoNhom | Int |  |  | Số nhóm |
| TinhTrang | nvarchar(20) |  |  | Tình trạng đã sử dụng hay chưa, hủy hay chưa |

|  |
| --- |
| ***ChiTietSuDung*** |
| *Column name* | *Data Type* | *Null* | *Index* | *Chú thích* |
| MaChiTietSD | varchar(10) |  | Primary key | Mã chi tiết sử dụng |
| MaCSD | varchar(10) |  | Foreign key | Mã ca sử dụng |
| MaTTB | varchar(10) |  | Foreign key | Mã trang thiết bị |
| GhiChu | nvarchar(100) | 🗸 |  | Ghi chú |

CHƯƠNG III. KHAI THÁC DỮ LIỆU VÀ LẬP TRÌNH ỨNG DỤNG

1. Mô tả tổng quan chương trình ứng dụng.

* + - * 1. Chương trình ứng dụng phân quyền cho 3 đối tượng:
* Chức vụ: Trưởng phòng – Quyền: admin
* Chức vụ: Kỹ thuật viên – Quyền: nhân viên
* Chức cụ: Giáo viên – Quyền: giáo viên
  1. Đăng nhập vào chương trình bằng tài khoản:

Quyền admin:

* TDN: admin
* MK: admin1

Quyền giáo viên:

* TDN: buithiphuongthao
* MK: 12345678

Quyền nhân viên:

* TDN: chuthilinh
* MK: 12345678

Hoặc các tài khoản khác trong CSDL – Bảng TaiKhoan

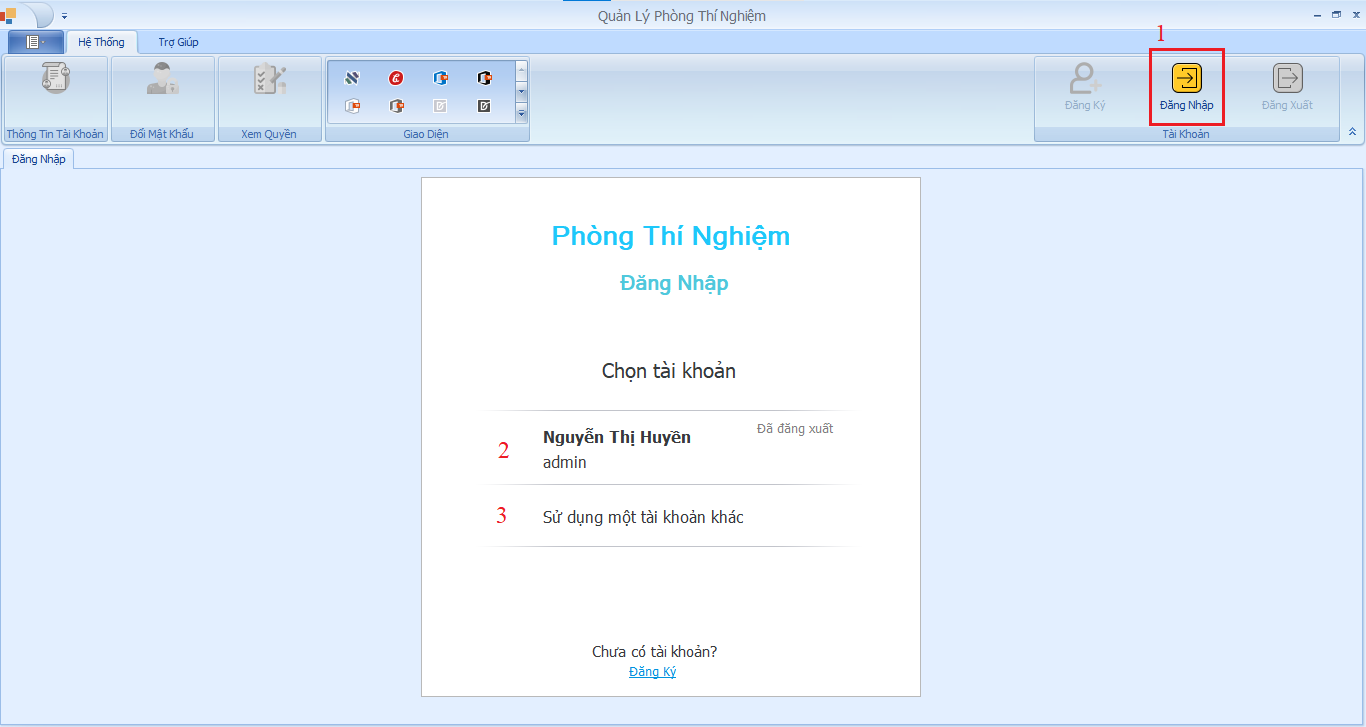
1. Đăng nhập
2. Chức năng hệ thống:

Đăng nhập vào hệ thống bằng 2 cách:

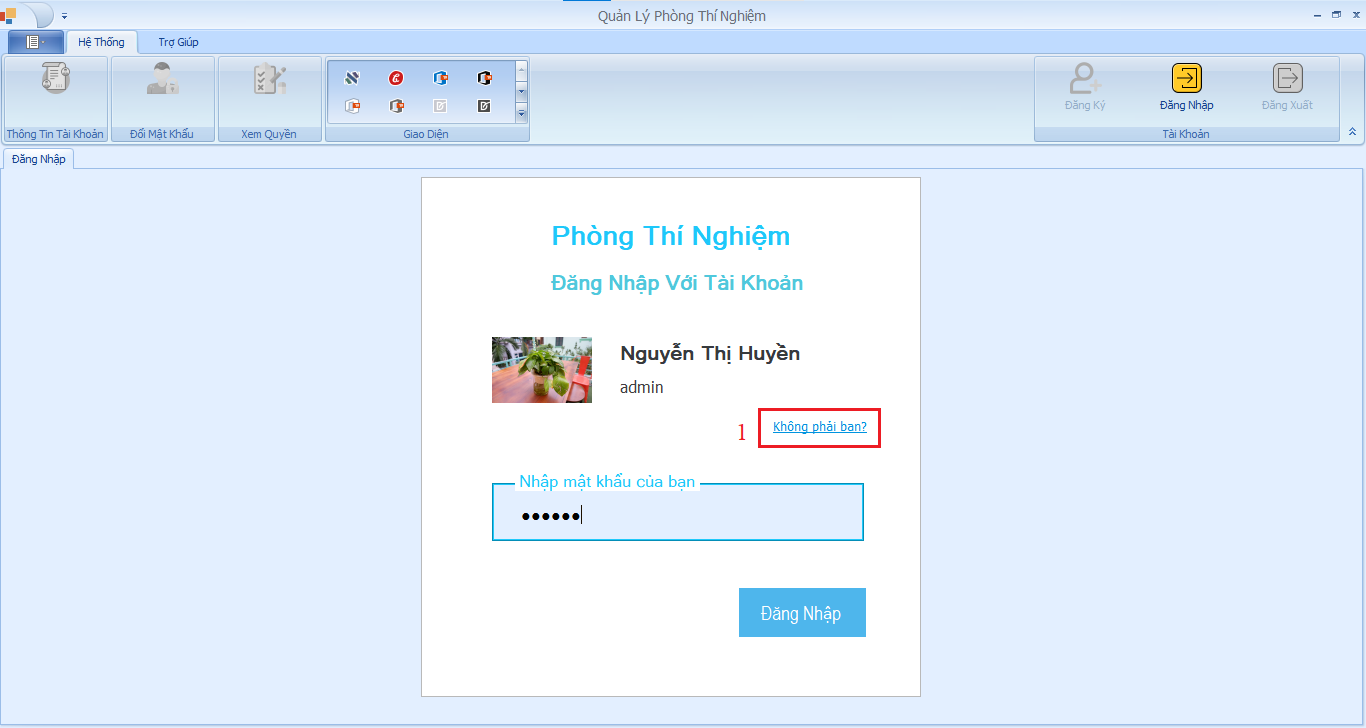
* Sử dụng tài khoản đăng nhập gần đây nhất.
* Sử dụng một tài khoản khác.

Khi đăng nhập thành công sẽ lưu thông tin tài khoản: Mã tài khoản, quyền, ....

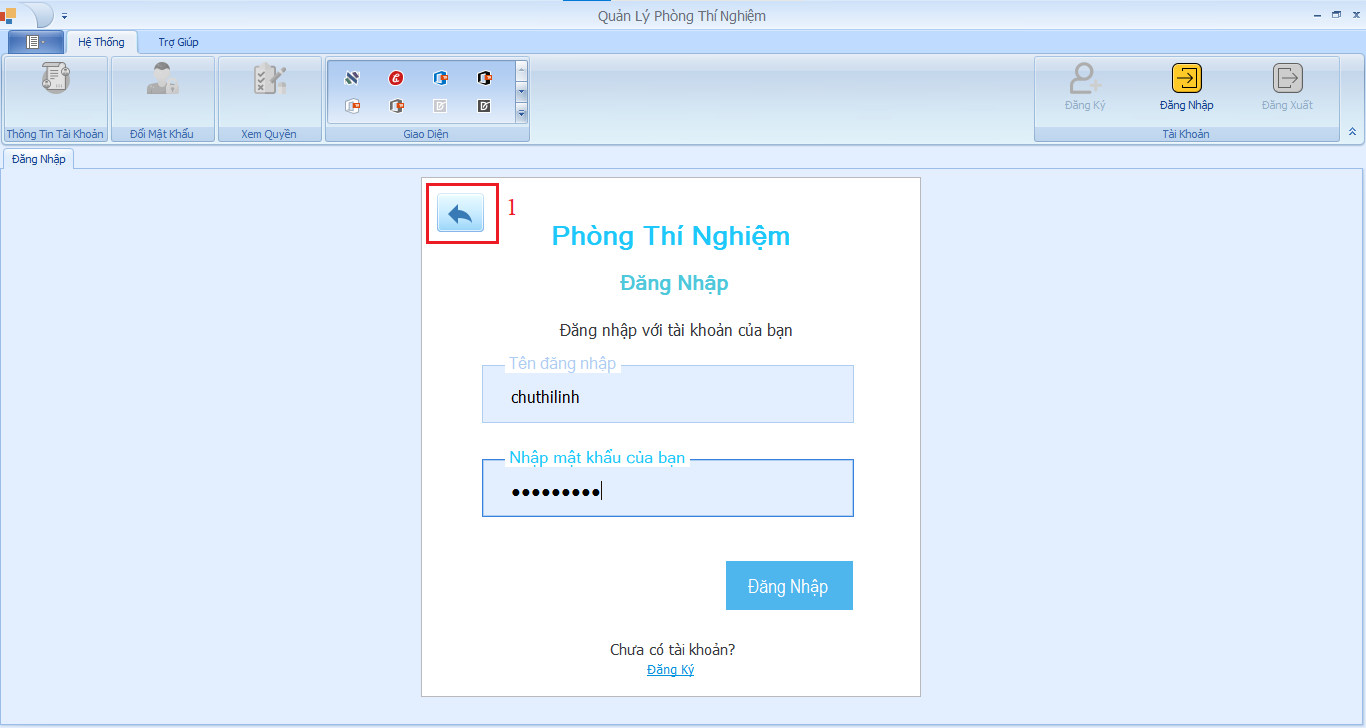
1. Giao diện hệ thống:



* Đây là giao diện chính sau khi nhấn vào Button Đăng nhập (1).
* Nhấn vào khu vực (2) để đăng nhập với tài khoản đăng nhập gần đây.
* Hoặc nhấn vào khu vực (3) để đăng nhập bằng tài khoản khác.



* Giao diện khi đăng nhập bằng tài khoản gần đây.
* Nhấn vào linkLabel “không phải bạn” (1) để quay lại giao diện chính.



* Giao diện khi đăng nhập bằng tài khoản khác.
* Có thể nhân vào button ở (1) để quay lại giao diện chính.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetTaiKhoanGanDay]

CREATE PROC [dbo].[spGetTaiKhoanGanDay]

-- Lấy thông tin: Mã tài khoản, Họ tên, Hình ảnh, Tên đăng nhập, Quyền

-- của tài khoản đăng nhập gần đây

AS

BEGIN

DECLARE @MaTK varchar(10)

IF EXISTS (SELECT MaTK FROM dbo.TaiKhoanGanDay)

BEGIN

SELECT @MaTK = MaTK FROM dbo.TaiKhoanGanDay

-- Lấy mã tài khoản vừa đăng nhập gần đây

IF EXISTS (SELECT MaNV FROM dbo.NhanVien WHERE MaNV = @MaTK)

-- Nếu tài khoản gần đây của nhân viên

-- Lấy thông tin của nhân viên có Mã NV = @MaTK

SELECT

NV.MaNV,

NV.HoTen,

NV.HinhAnh,

NV.TenDN,

NV.ChucVu AS Quyen

FROM dbo.NhanVien NV

WHERE NV.MaNV = @MaTK

ELSE

-- Nếu tài khoản gần đây của giáo viên

-- Lấy thông tin của giáo viên có Mã GV = @MaTK

SELECT

GV.MaGV,

GV.HoTen,

GV.HinhAnh,

GV.TenDN, N'Giáo viên' AS Quyen

FROM dbo.GiaoVien GV

WHERE GV.MaGV = @MaTK

END

END

GO

* Stored procedure [dbo].[spKiemTraDangNhap]

CREATE PROC [dbo].[spKiemTraDangNhap]

-- Kiểm tra xem có tồn tại tài khoản

-- với tên đăng nhập và mật khẩu truyền vào hay không

@TenDN varchar(20),

@MatKhau varchar(20)

AS

BEGIN

DECLARE @MaTK varchar(10) = NULL

-- Nếu có tài khoản nhân viên khớp với thông tin đăng nhập

IF EXISTS

(

SELECT NV.MaNV

FROM dbo.NhanVien NV

WHERE TenDN = @TenDN AND MatKhau = @MatKhau

)

BEGIN

-- Lấy mã của nhân viên đó

SELECT @MaTK = NV.MaNV

FROM dbo.NhanVien NV

WHERE TenDN = @TenDN AND MatKhau = @MatKhau

-- Lấy các thông tin tài khoản nhân viên:

-- Mã tài khoản, Họ tên, Hình ảnh, Tên đăng nhập, Chức vụ

SELECT

NV.MaNV AS MaTK,

NV.HoTen,

NV.HinhAnh,

NV.TenDN,

NV.ChucVu AS Quyen

FROM dbo.NhanVien NV

WHERE NV.MaNV = @MaTK

-- Đăng nhập thành công thì cập nhật bảng TaiKhoanGanDay

UPDATE [dbo].[TaiKhoanGanDay] SET MaTK = @MaTK

END

-- Nếu có tài khoản giáo viên khớp với thông tin đăng nhập

ELSE IF EXISTS

(

SELECT GV.MaGV

FROM dbo.GiaoVien GV

WHERE TenDN = @TenDN AND MatKhau = @MatKhau

)

BEGIN

-- Lấy mã của giáo viên đó

SELECT @MaTK = GV.MaGV

FROM dbo.GiaoVien GV

WHERE TenDN = @TenDN AND MatKhau = @MatKhau

-- Lấy các thông tin tài khoản giáo viên

SELECT

GV.MaGV AS MaTK,

GV.HoTen,

GV.HinhAnh,

GV.TenDN,

N'Giáo viên' AS Quyen

FROM dbo.GiaoVien GV

WHERE GV.MaGV = @MaTK

-- Đăng nhập thành công thì cập nhật bảng TaiKhoanGanDay

UPDATE [dbo].[TaiKhoanGanDay] SET MaTK = @MaTK

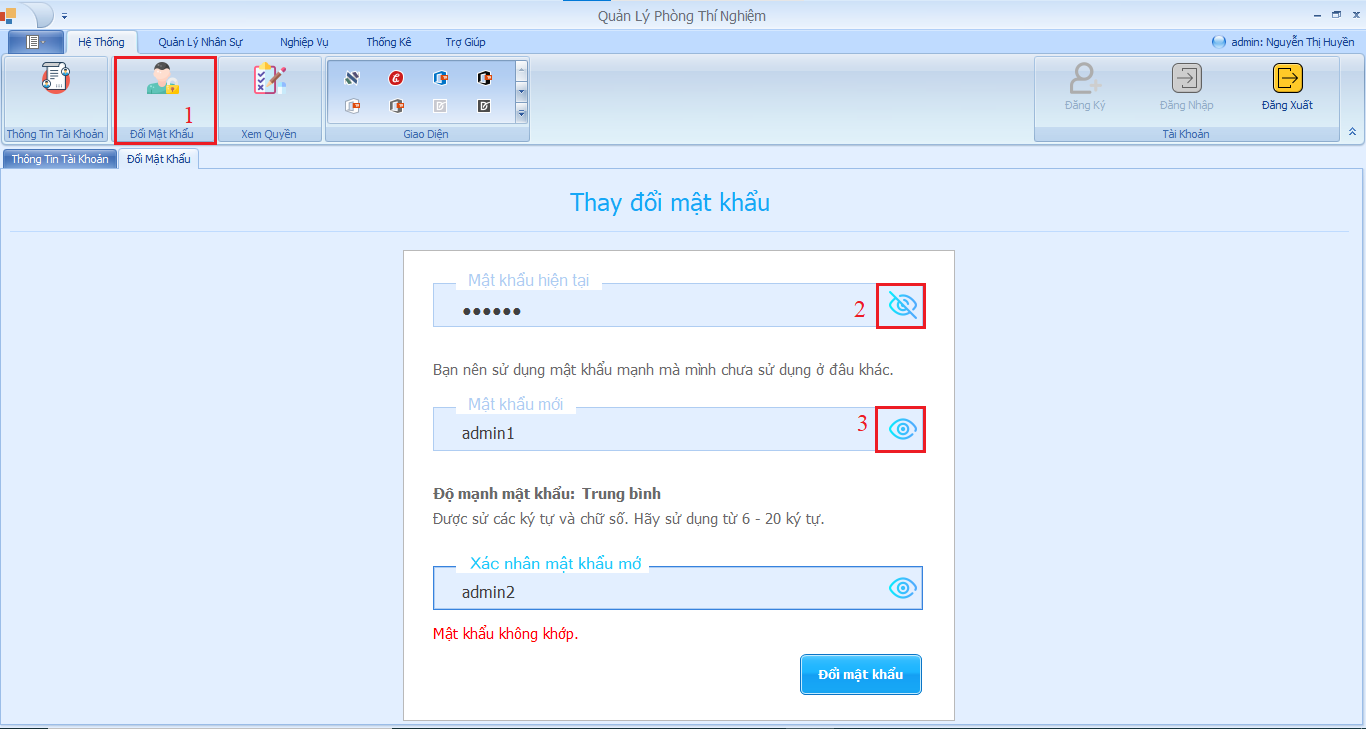
END

END

1. Đổi mật khẩu
2. Chức năng hệ thống:

Giúp người dùng có thể đổi mật khẩu của tài khoản mình.

1. Giao diện hệ thống:



* Giao diện khi nhấn vào button Đổi mật khẩu (1).
* MK mới phải đảm bảo đủ mạnh và chỉ chưa các chữ cái và chữ số.
* Xác nhận MK nhập lại khớp với MK mới.
* Có thể ẩn hoặc hiện mật khẩu bằng cách nhấn vào các Button ở (2), (3)

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spDoiMatKhau]

CREATE PROC [dbo].[spDoiMatKhau]

-- Kiểm tra mật khẩu cũ có khớp với tên đăng nhập hay không

-- Nếu khớp với tên đăng nhập thì đổi mật khẩu

@TenDN varchar(20),

@MatKhauCu varchar(20),

@MatKhauMoi varchar(20)

AS

BEGIN

DECLARE @MaTK varchar(10) = NULL

-- Kiểm tra có tồn tại tài khoản giáo viên ứng với TDN và MK cũ hay không

IF EXISTS (SELECT MaGV FROM dbo.GiaoVien WHERE TenDN = @TenDN AND MatKhau = @MatKhauCu)

BEGIN

SELECT @MaTK = MaGV FROM dbo.GiaoVien WHERE TenDN = @TenDN AND MatKhau = @MatKhauCu

-- Nếu MK cũ khớp với TDN thì Update mật khẩu

UPDATE [dbo].[GiaoVien]

SET MatKhau = @MatKhauMoi

WHERE MaGV = @MaTK

END

-- Kiểm tra có tồn tại tài khoản nhân viên ứng với TDN và MK cũ hay không

ELSE IF EXISTS (SELECT MaNV FROM dbo.NhanVien WHERE TenDN = @TenDN AND MatKhau = @MatKhauCu)

BEGIN

SELECT @MaTK = MaNV FROM dbo.NhanVien WHERE TenDN = @TenDN AND MatKhau = @MatKhauCu

-- Nếu MK cũ khớp với TDN thì Update mật khẩu

UPDATE [dbo].[NhanVien]

SET MatKhau = @MatKhauMoi

WHERE MaNV = @MaTK

END

-- Nếu MaTK = NULL thì MK cũ ko khớp.

-- Nếu MaTk != NULL thì đổi MK thành công

SELECT @MaTK

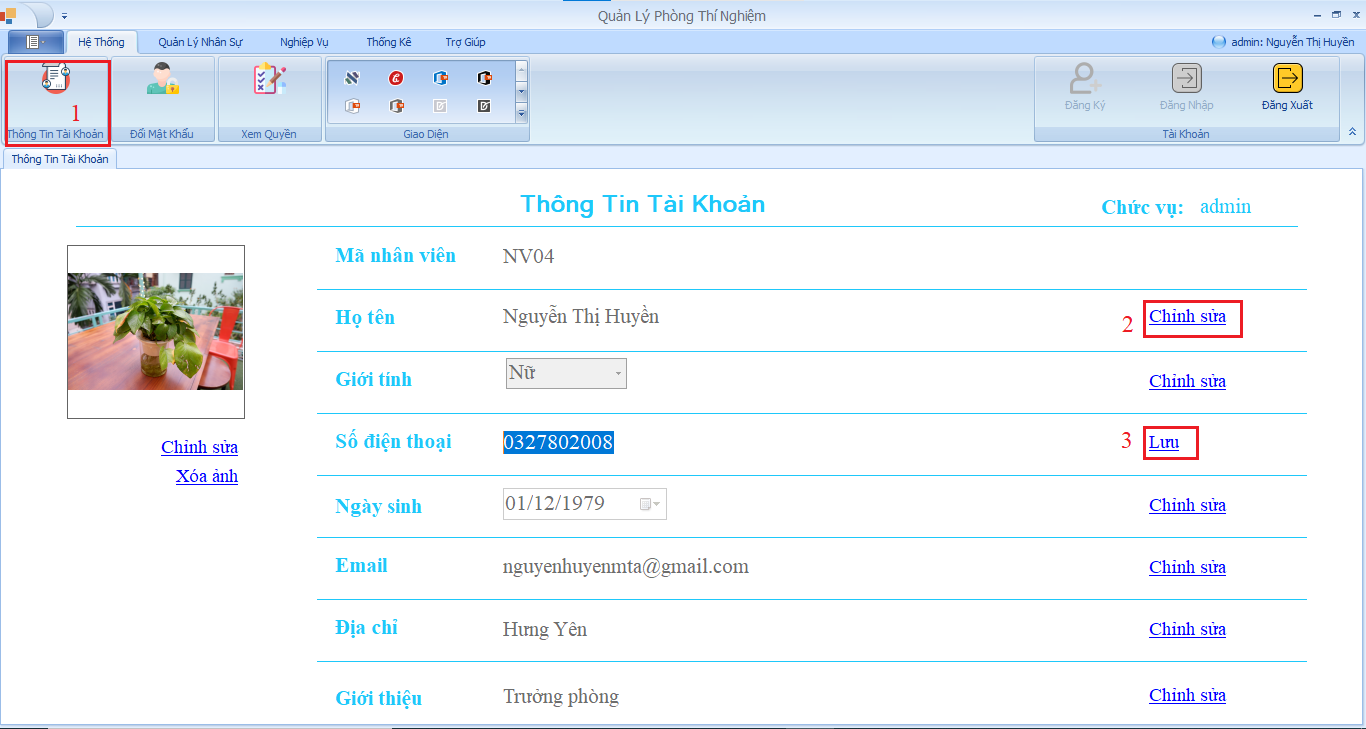
END

GO

1. Quản lý thông tin tài khoản
2. Chức năng hệ thống:

Khi đăng nhập thành công, người dùng có thể xem và sửa thông tin tài khoản của mình.

1. Giao diên hệ thống:



* Giao diện chương trình sau khi nhấn Button Thông tin tài khoản (1).
* Có thể nhấn các linkLabel Chỉnh sửa (2) để thay đổi thông tin tài khoản.
* Nhấn linkLabel Lưu (3) để lưu thay đổi.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetGiaoVienByMaTK]

CREATE PROC [dbo].[spGetGiaoVienByMaTK]

-- Lấy thông tin tài khoản giáo viên khi đăng nhập bằng tài khoản giáo viên

@MaTK varchar(10)

AS

BEGIN

SELECT

GV.HinhAnh,

GV.MaGV,

GV.HoTen,

GV.GioiTinh,

GV.SDT,

GV.NgaySinh,

GV.Email,

GV.DiaChi,

GV.GioiThieu

FROM dbo.GiaoVien GV

WHERE GV.MaGV = @MaTK

END

GO

* Stored procedure [dbo].[spGetNhanVienByMaTK]

CREATE PROC [dbo].[spGetNhanVienByMaTK]

-- Lấy thông tin tài khoản nhân viên viên khi đăng nhập bằng tài khoản nhân viên

@MaTK varchar(10)

AS

BEGIN

SELECT

NV.HinhAnh,

NV.MaNV,

NV.HoTen,

NV.ChucVu,

NV.GioiTinh,

NV.SDT,

NV.NgaySinh,

NV.Email,

NV.DiaChi,

NV.GioiThieu

FROM dbo.NhanVien NV

WHERE NV.MaNV = @MaTK

END

GO

* Stored procedure [dbo].[spUpdateNhanVien]

CREATE PROC [dbo].[spUpdateNhanVien]

-- Sửa thông tin của Nhân viên

@MaNV varchar(10),

@ChucVu nvarchar(20),

@HinhAnh image,

@HoTen nvarchar(50),

@GioiTinh nvarchar(10),

@SDT varchar(10),

@NgaySinh date,

@Email varchar(50),

@DiaChi nvarchar(100),

@GioiThieu nvarchar(100)

AS

BEGIN

UPDATE dbo.NhanVien SET

ChucVu = @ChucVu,

HinhAnh = @HinhAnh,

HoTen = @HoTen,

GioiTinh = @GioiTinh,

SDT = @SDT,

NgaySinh = @NgaySinh,

Email = @Email,

DiaChi = @DiaChi,

GioiThieu = @GioiThieu

WHERE MaNV = @MaNV

END

GO

* Stored procedure [dbo].[spUpdateGiaoVien]

CREATE PROC [dbo].[spUpdateGiaoVien]

-- Sửa thông tin của giáo viên

@MaGV varchar(10),

@HinhAnh image,

@HoTen nvarchar(50),

@GioiTinh nvarchar(10),

@SDT varchar(10),

@NgaySinh date,

@Email varchar(50),

@DiaChi nvarchar(100),

@GioiThieu nvarchar(100)

AS

BEGIN

UPDATE dbo.GiaoVien SET

HinhAnh = @HinhAnh,

HoTen = @HoTen,

GioiTinh = @GioiTinh,

SDT = @SDT,

NgaySinh = @NgaySinh,

Email = @Email,

DiaChi = @DiaChi,

GioiThieu = @GioiThieu

WHERE MaGV = @MaGV

END

GO

1. Quản lý nhân sự: Nhân viên
2. Chức năng hệ thống:

*Với admin:*

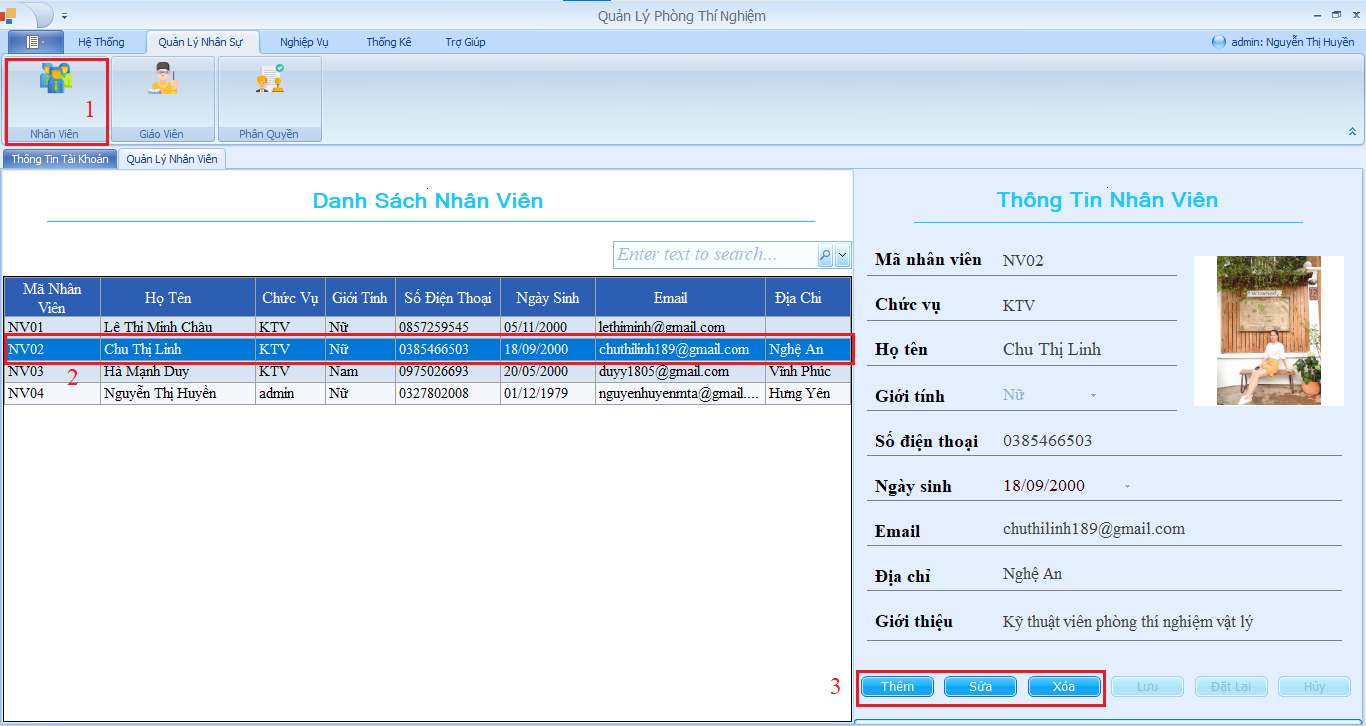
* Xem danh sách nhân viên, thông tin chi tiết của các nhân viên.
* Tìm kiếm danh sách nhân viên theo thông tin bất kì.
* Thêm, sửa thông tin nhân viên.
* Xóa nhân viên khi nhân viên đó chưa quản lý bất kỳ Ca sử dụng nào.

*Với Nhân viên và Giáo viên:*

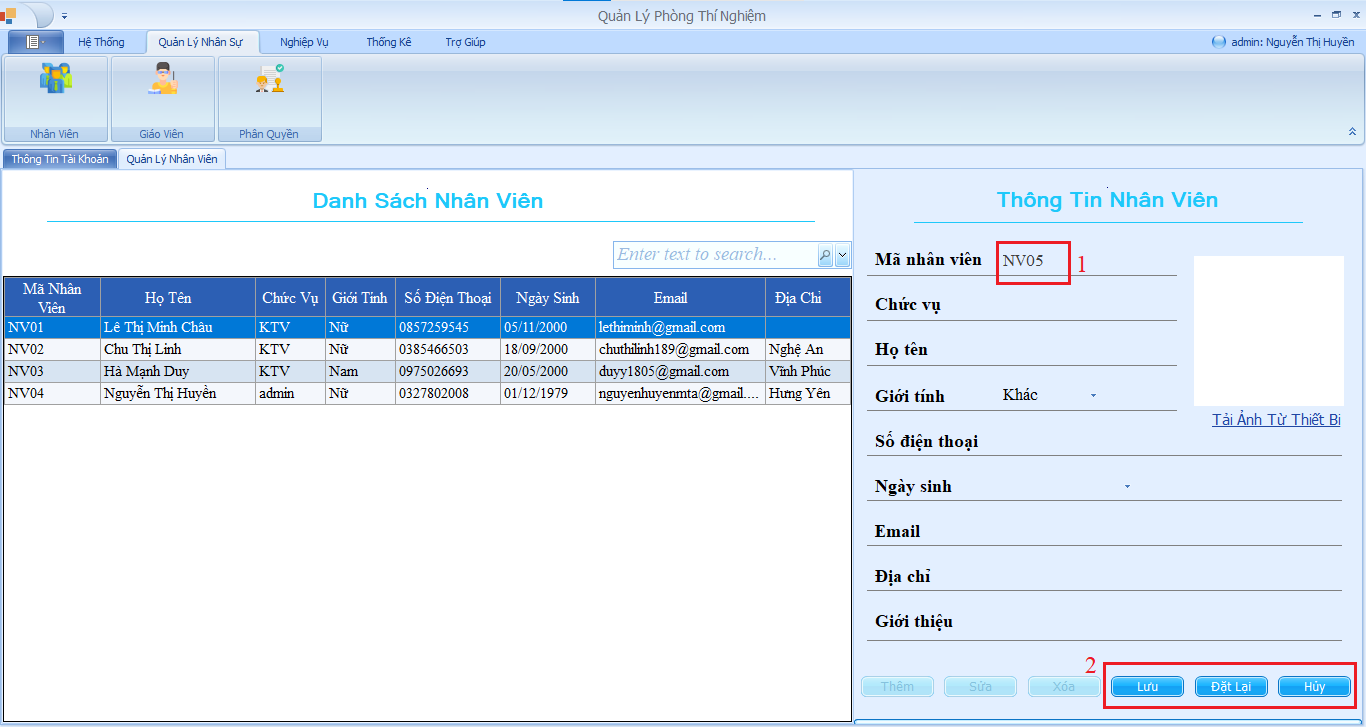
* Xem danh sách nhân viên, thông tin chi tiết của các nhân viên.
* Tìm kiếm danh sách nhân viên theo thông tin bất kì.

1. Giao diện hệ thống:

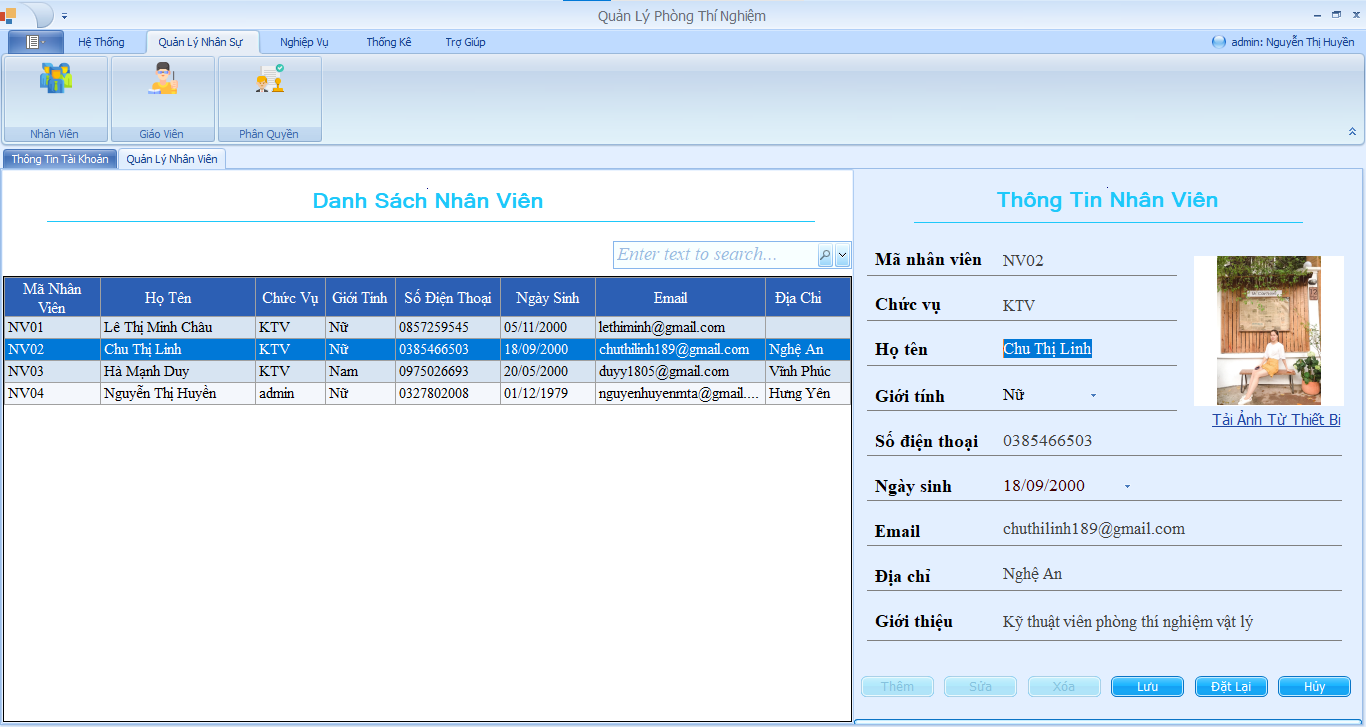
* Đăng nhập với quyền: Admin



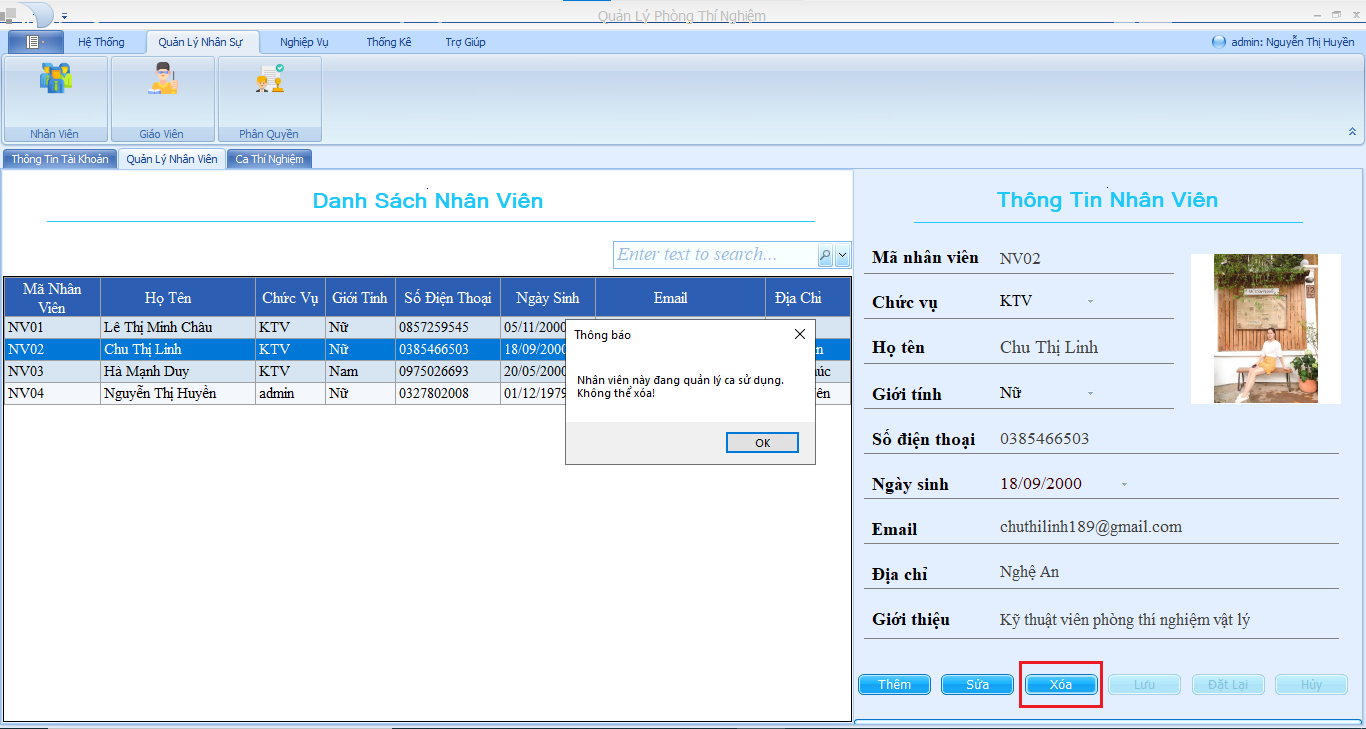
* Giao diện khi nhấn vào Button Nhân viên (1) thuộc Page Quản lý nhân sự.
* Có thể xem thông tin chi tiết của nhân viên khi Click vào Row chứa nhân viên đó (2).
* Thêm, sửa, xóa thông tin nhân viên bằng cách nhấn vào các Button ở (3)



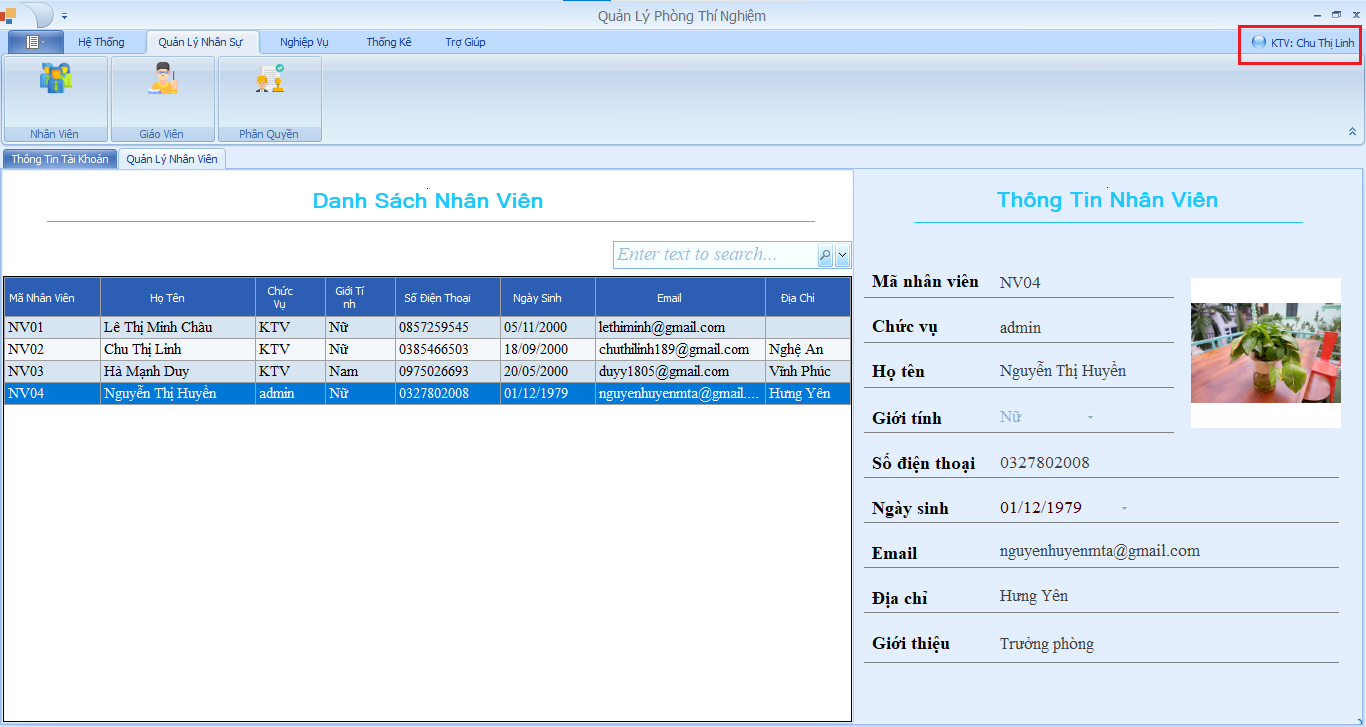
* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã nhân viên dạng “NVxx” như ở (1). Trong đó xx tăng tự động.
* Sau khi nhập các thông tin thì nhấn Button Lưu (2) để thêm nhân viên.
* Nhấn Button Đặt lại (2) để điền lại thông tin nhân viên.
* Khi không muốn thêm nhân viên nữa thì nhấn Button Hủy (2).



* Giao diện khi nhấn vào Button Sửa.



* Khi nhấn vào Button Xóa, sẽ kiểm tra xem Nhân viên đã quản lý Ca sử dụng nào hay chưa. Nếu có thì sẽ không xóa được.
* Đăng nhập với quyền: KTV hoặc Giáo viên



* Khi đăng nhập với quyền KTV hoặc Giáo viên thì chỉ xem được danh sách nhân viên và thông tin chi tiết của nhân viên. Không thể thêm, sửa, xóa nhân viên.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spUpdateNhanVien]
* Stored procedure [dbo].[spGetNhanVien]

CREATE PROC [dbo].[spGetNhanVien]

-- Lấy danh sách nhân viên

AS

BEGIN

SELECT

NV.MaNV,

NV.HoTen,

NV.ChucVu,

NV.GioiTinh,

NV.SDT,

NV.NgaySinh,

NV.Email,

NV.DiaChi

FROM dbo.NhanVien NV

END

GO

* Stored procedure [dbo].[spGetNhanVienByMaNV]

CREATE PROC [dbo].[spGetNhanVienByMaNV]

-- Lấy thông tin chi tiết của nhân viên từ Mã nhân viên

@MaNV varchar(10)

AS

BEGIN

SELECT

NV.HinhAnh,

NV.MaNV,

NV.HoTen,

NV.ChucVu,

NV.GioiTinh,

NV.SDT,

NV.NgaySinh,

NV.Email,

NV.DiaChi,

NV.GioiThieu

FROM dbo.NhanVien NV

WHERE NV.MaNV = @MaNV

END

GO

* Stored procedure [dbo].[spGetMaNhanVien]

CREATE PROC [dbo].[spGetMaNhanVien]

-- Tạo mã nhân viên có dạng NVxx

-- XX là số tự động tăng

AS

BEGIN

DECLARE @MaNV varchar(10)

DECLARE @MaNVMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã có nhân viên thì xx bằng mã NV lớn nhất + 1

IF EXISTS( SELECT MaNV FROM dbo.NhanVien)

BEGIN

SELECT @MaNVMax = MAX(MaNV) FROM dbo.NhanVien

SET @Max = CONVERT(int, SUBSTRING(@MaNVMax, 3, 2)) + 1

IF @Max < 10 SET @MaNV = 'NV' + '0' + CONVERT(varchar(1), @Max)

ELSE SET @MaNV = 'NV' + CONVERT(varchar(2), @Max)

END

-- Nếu chưa có nhân viên nào thì bắt đầu bằng 'NV01'

ELSE SET @MaNV = 'NV01'

SELECT @MaNV

END

GO

* Stored procedure [dbo].[spInsertNhanVien]

CREATE PROC [dbo].[spInsertNhanVien]

-- Thêm nhân viên

@MaNV varchar(10),

@ChucVu nvarchar(20),

@HinhAnh image,

@HoTen nvarchar(50),

@GioiTinh nvarchar(10),

@SDT varchar(10),

@NgaySinh date,

@Email varchar(50),

@DiaChi nvarchar(200),

@GioiThieu ntext

AS

BEGIN

INSERT INTO dbo.NhanVien (MaNV, ChucVu, HinhAnh, HoTen, GioiTinh, SDT, NgaySinh, Email, DiaChi, GioiThieu)

VALUES ( @MaNV, @ChucVu, @HinhAnh, @HoTen, @GioiTinh, @SDT, @NgaySinh, @Email, @DiaChi, @GioiThieu )

END

GO

* Stored procedure [dbo].[spDeleteNhanVien]

CREATE PROC [dbo].[spDeleteNhanVien]

-- Xóa nhân viên có Mã nhân viên được truyền vào

@MaNV varchar(10)

AS

BEGIN

DELETE FROM dbo.NhanVien WHERE MaNV = @MaNV

END

* Stored procedure [dbo].[spCountNVQuanLyCSD]

CREATE PROC [dbo].[spCountNVQuanLyCSD]

-- Đếm số ca sử dụng mà một nhân viên quản lý

-- Nếu NV chưa quản lý CSD nào thì mới được phép xóa NV đó

@MaNV varchar(10)

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.ChiTietDangKi CTDK

WHERE CTDK.MaNV = @MaNV

END

GO

1. Quản lý nhân sự: Giáo viên
2. Chức năng hệ thống:

*Với admin:*

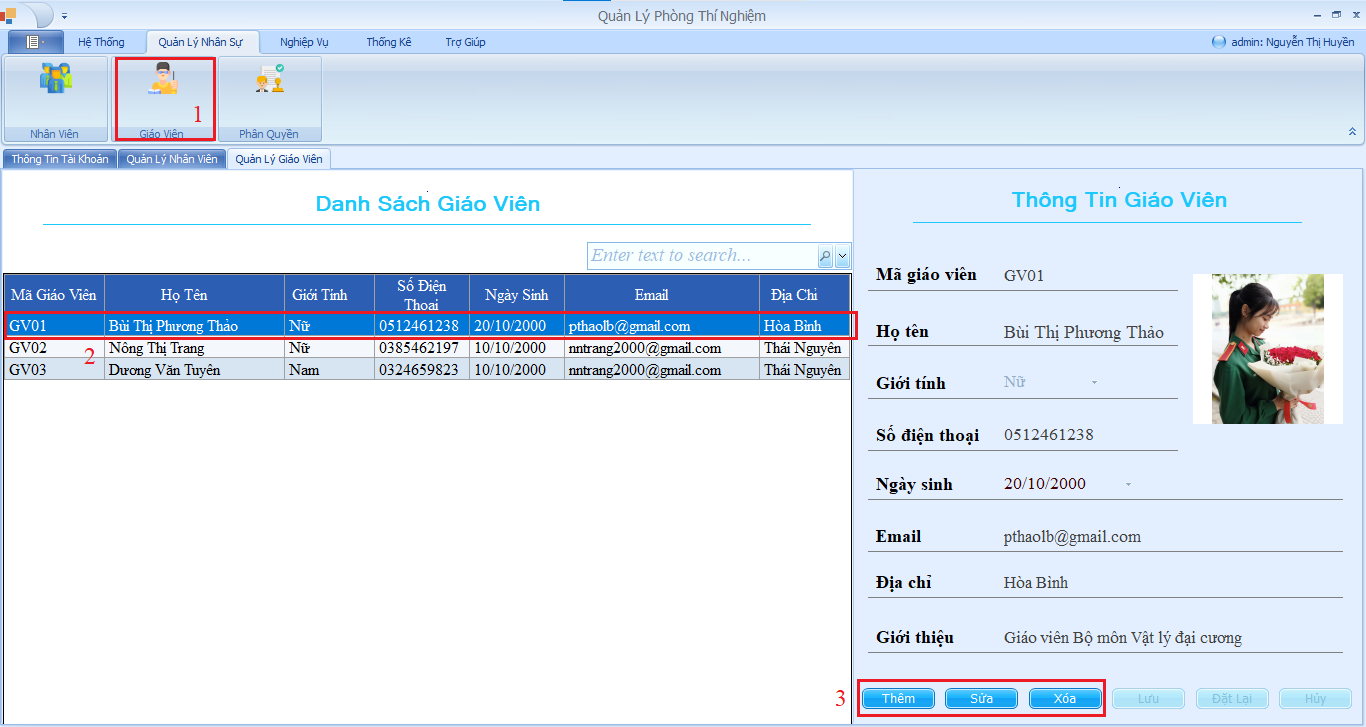
* Xem danh sách giáo viên, thông tin chi tiết của các giáo viên.
* Tìm kiếm danh sách giáo viên theo thông tin bất kì.
* Thêm, sửa thông tin giáo viên.
* Xóa giáo viên khi giáo viên đó không có bất kỳ Bản đăng ký sử dụng nào.

*Với Nhân viên và Giáo viên:*

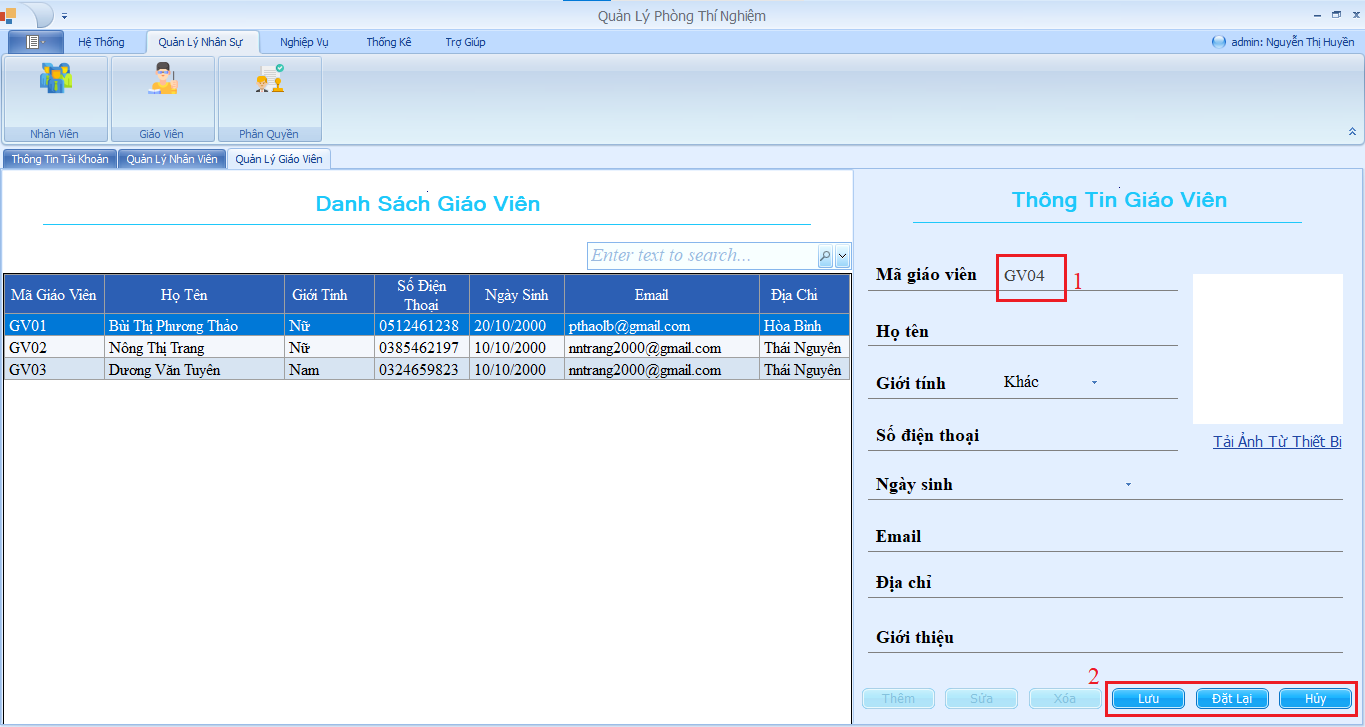
* Xem danh sách giáo viên, thông tin chi tiết của các giáo viên.
* Tìm kiếm danh sách giáo viên theo thông tin bất kì.

1. Giao diện hệ thống:

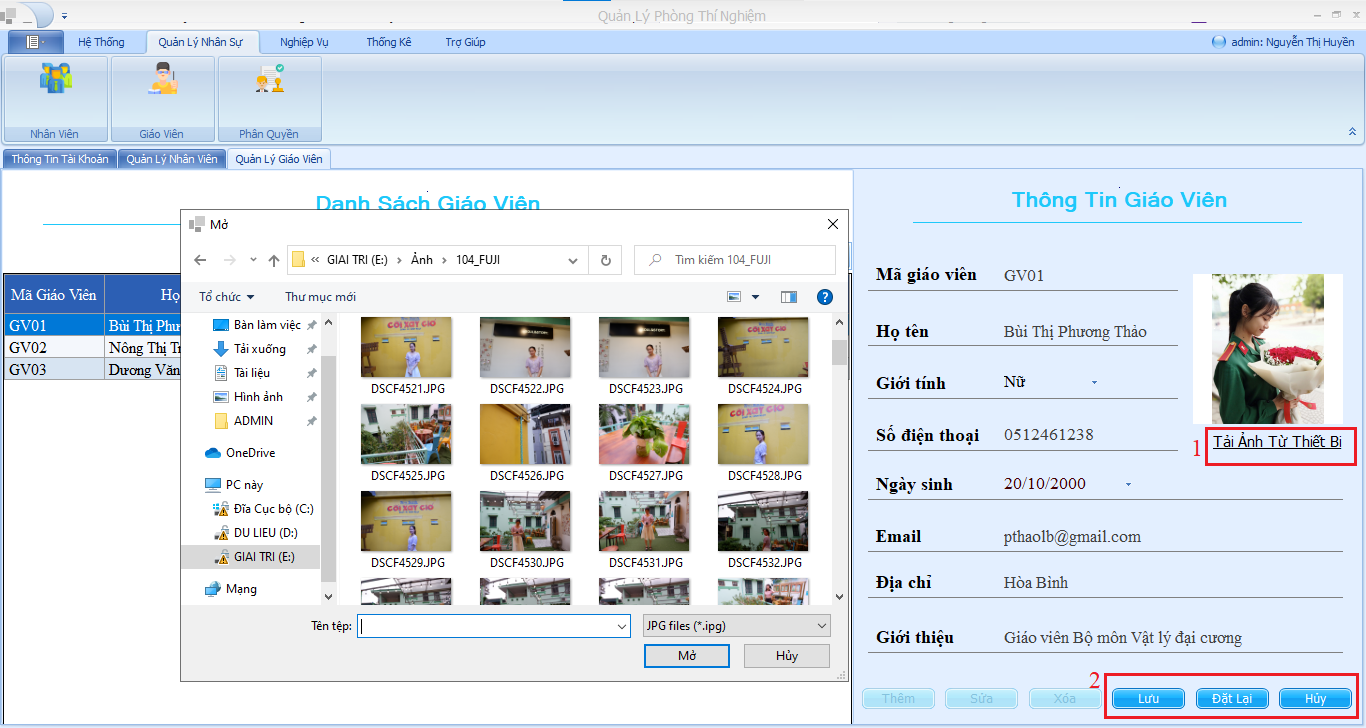
* Đăng nhập với quyền: Admin



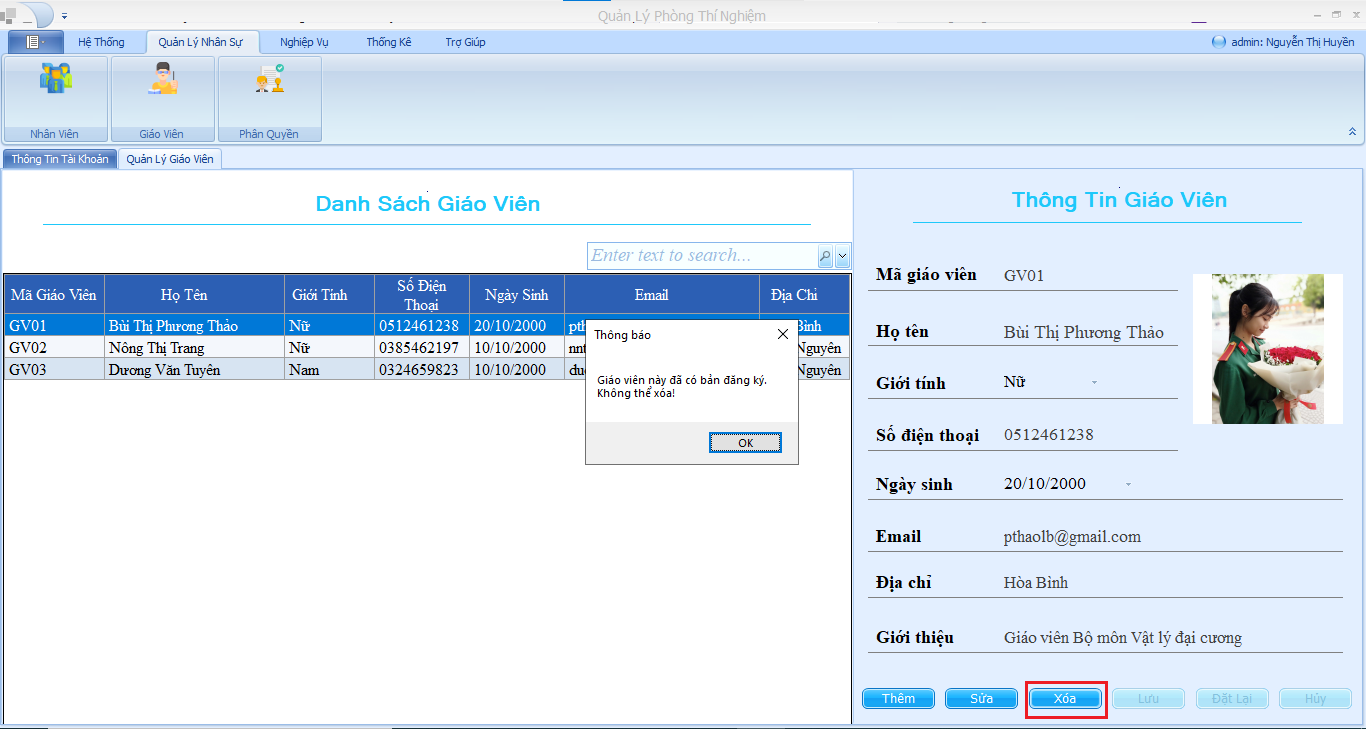
* Giao diện khi nhấn vào Button Giáo viên (1) thuộc Page Quản lý nhân sự.
* Có thể xem thông tin chi tiết của Giáo viên khi Click vào Row chứa giá viên đó (2).
* Thêm, sửa, xóa thông tin giáo viên bằng cách nhấn vào các Button ở (3)



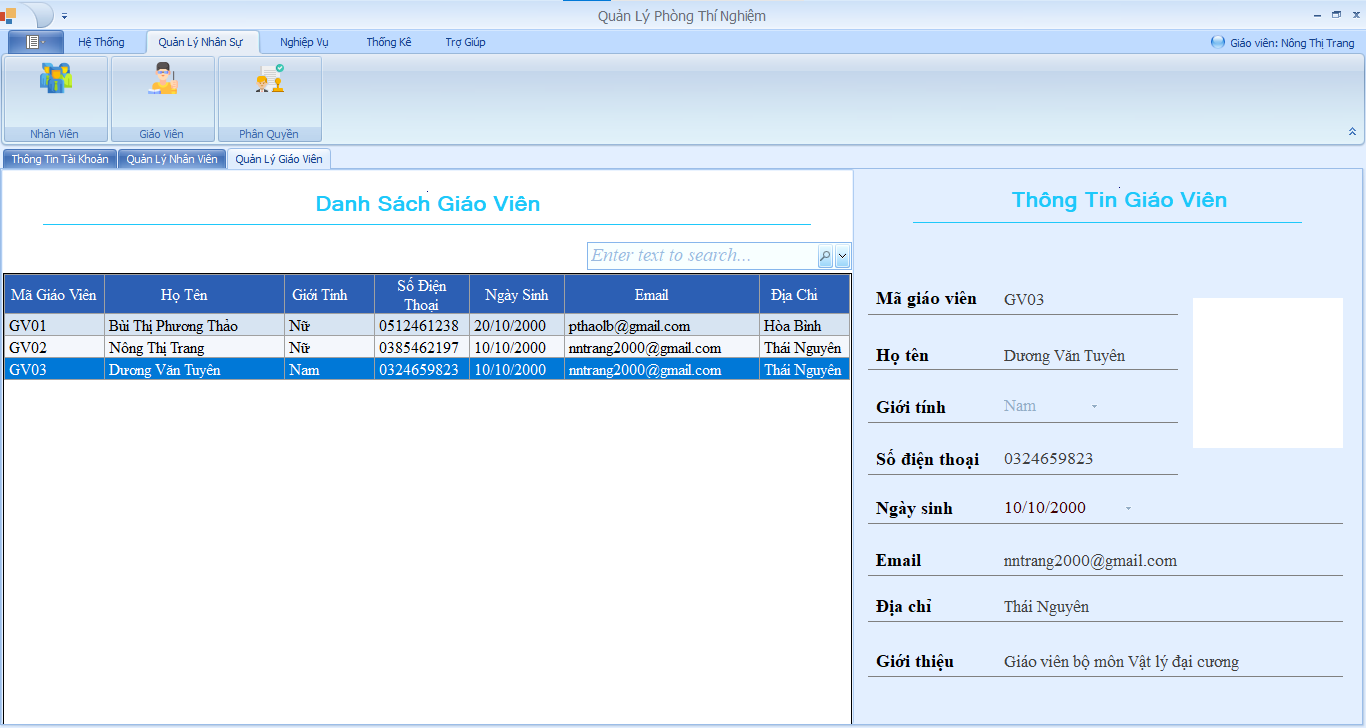
* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã giáo viên dạng “GVxx” như ở (1). Trong đó xx tăng tự động.
* Sau khi nhập các thông tin thì nhấn Button Lưu (2) để thêm giáo viên.
* Nhấn Button Đặt lại (2) để điền lại các thông tin giáo viên.
* Khi không muốn thêm giáo viên nữa thì nhấn Button Hủy (2).



* Giao diện khi nhấn vào Button Sửa và nhấn vào linkLabel Tải ảnh từ thiết bị (1).
* Nhấn Button Lưu (2) để lưu các thông tin đã chỉnh sửa.



* Khi nhấn vào Button Xóa, sẽ kiểm tra xem Giáo viên đã có Bản đăng ký nào hay chưa. Nếu có thì sẽ không xóa được.
* Đăng nhập với quyền: KTV hoặc Giáo viên



* Khi đăng nhập với quyền KTV hoặc Giáo viên thì chỉ xem được danh sách giáo viên và thông tin chi tiết của giáo viên. Không thể thêm, sửa, xóa giáo viên.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spUpdateGiaoVien]
* Stored procedure [dbo].[spGetGiaoVien]

CREATE PROC [dbo].[spGetGiaoVien]

-- Lấy danh sách giáo viên

AS

BEGIN

SELECT

GV.MaGV,

GV.HoTen,

GV.GioiTinh,

GV.SDT,

GV.NgaySinh,

GV.Email,

GV.DiaChi

FROM dbo.GiaoVien GV

END

GO

* Stored procedure [dbo].[spGetGiaoVienByMaGV]

CREATE PROC [dbo].[spGetGiaoVienByMaGV]

-- lấy thông tin chi tiết của Giáo Viên Từ Mã Giáo Viên

@MaGV varchar(10)

AS

BEGIN

SELECT

GV.HinhAnh,

GV.MaGV,

GV.HoTen,

GV.GioiTinh,

GV.SDT,

GV.NgaySinh,

GV.Email,

GV.DiaChi,

GV.GioiThieu

FROM dbo.GiaoVien GV

WHERE GV.MaGV = @MaGV

END

GO

* Stored procedure [dbo].[spGetMaGiaoVien]

CREATE PROC [dbo].[spGetMaGiaoVien]

-- Tạo Mã giáo viên có dạng GVxx

-- Trong đó XX là số tự động tăng

AS

BEGIN

DECLARE @MaGV varchar(10)

DECLARE @MaGVMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã có giáo viên thì xx bằng mã GV lớn nhất + 1

IF EXISTS( SELECT MaGV FROM dbo.GiaoVien)

BEGIN

SELECT @MaGVMax = MAX(MaGV) FROM dbo.GiaoVien

SET @Max = CONVERT(int, SUBSTRING(@MaGVMax, 3, 2)) + 1

IF @Max < 10 SET @MaGV = 'GV' + '0' + CONVERT(varchar(1), @Max)

ELSE SET @MaGV = 'GV' + CONVERT(varchar(2), @Max)

END

-- Nếu chưa có giáo viên nào thì bắt đầu bằng 'GV01'

ELSE SET @MaGV = 'GV01'

SELECT @MaGV

END

GO

* Stored procedure [dbo].[spInsertGiaoVien]

CREATE PROC [dbo].[spInsertGiaoVien]

-- Thêm Giáo Viên

@MaGV varchar(10),

@HinhAnh image,

@HoTen nvarchar(50),

@GioiTinh nvarchar(10),

@SDT varchar(10),

@NgaySinh date,

@Email varchar(50),

@DiaChi nvarchar(200),

@GioiThieu ntext

AS

BEGIN

INSERT INTO dbo.GiaoVien (MaGV, HinhAnh, HoTen, GioiTinh, SDT, NgaySinh, Email, DiaChi, GioiThieu)

VALUES ( @MaGV, @HinhAnh, @HoTen, @GioiTinh, @SDT, @NgaySinh, @Email, @DiaChi, @GioiThieu )

END

GO

* Stored procedure [dbo].[spDeleteGiaoVien]

CREATE PROC [dbo].[spDeleteGiaoVien]

-- Xóa giáo viên có mã GV được truyền vào

@MaGV varchar(10)

AS

BEGIN

DELETE FROM dbo.GiaoVien WHERE MaGV = @MaGV

END

GO

* Stored procedure [dbo].[spCountGVDangKySD]

CREATE PROC [dbo].[spCountGVDangKySD]

-- Đếm số bản đăng ký của 1 giáo viên

-- Nếu GV không có Bản đăng ký nào thì mới được phép xóa

@MaGV varchar(10)

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.BanDangKy BDK

WHERE BDK.MaGV = @MaGV

END

GO

1. Quản lý hệ thống Phòng thí nghiệm
2. Chức năng hệ thống:

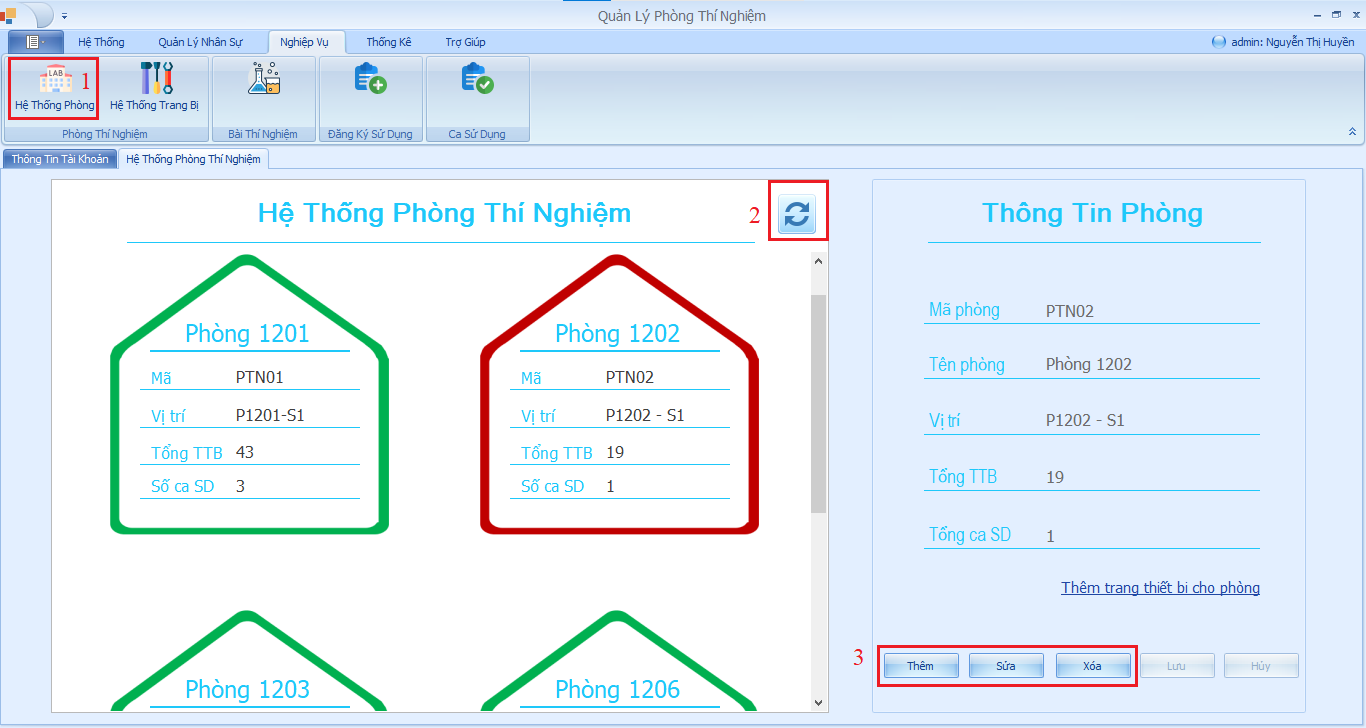
*Với admin:*

* Xem danh sách, thông tin các phòng thí nghiệm.
* Thêm, sửa thông tin Phòng thí nghiệm.
* Xóa Phòng thí nghiệm nếu trong phòng không có Trang thiết bị nào.
* Khi xóa Phòng thí nghiệm thì xóa tất cả các Ca sử dụng thuộc phòng đó, và xóa các Chi tiết sử dụng của các Ca sử dụng đó (dùng trigger).

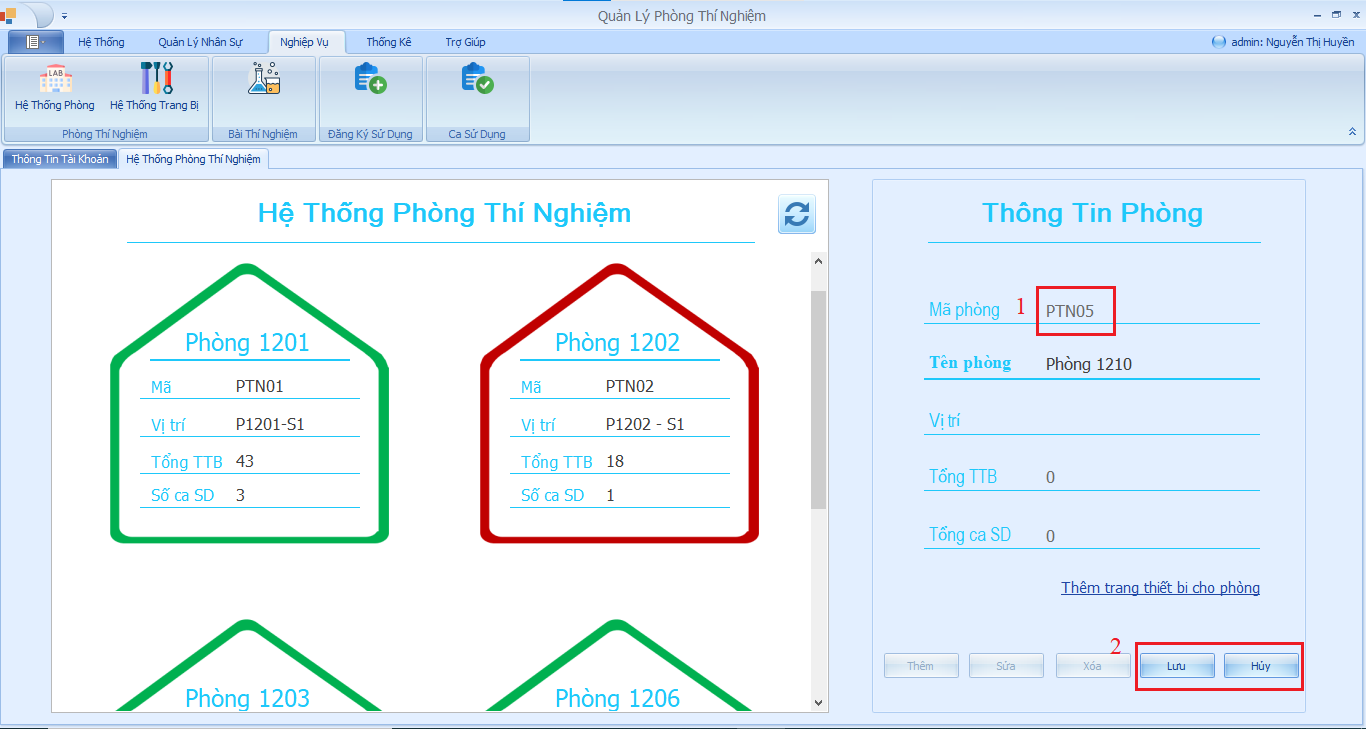
*Với Nhân viên và Giáo viên:*

* Xem danh sách, thông tin các phòng thí nghiệm.

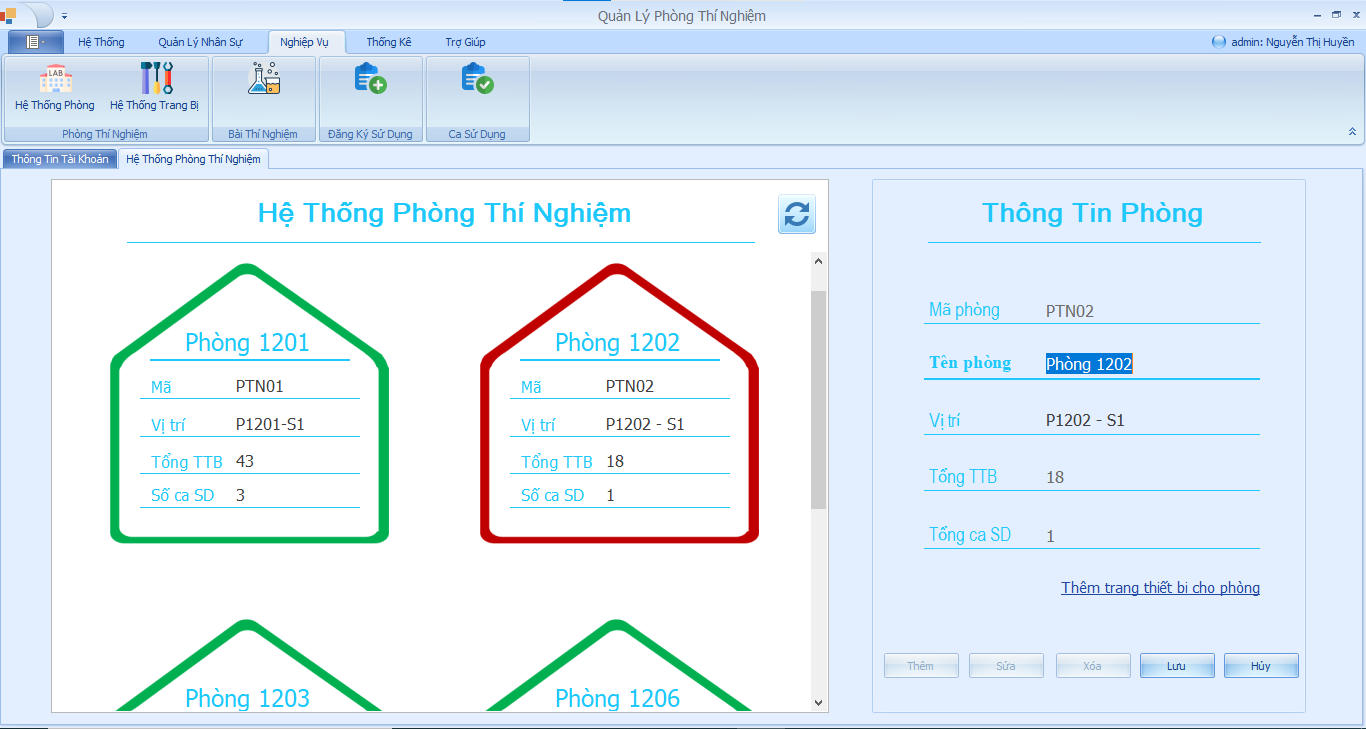
1. Giao diện hệ thống:



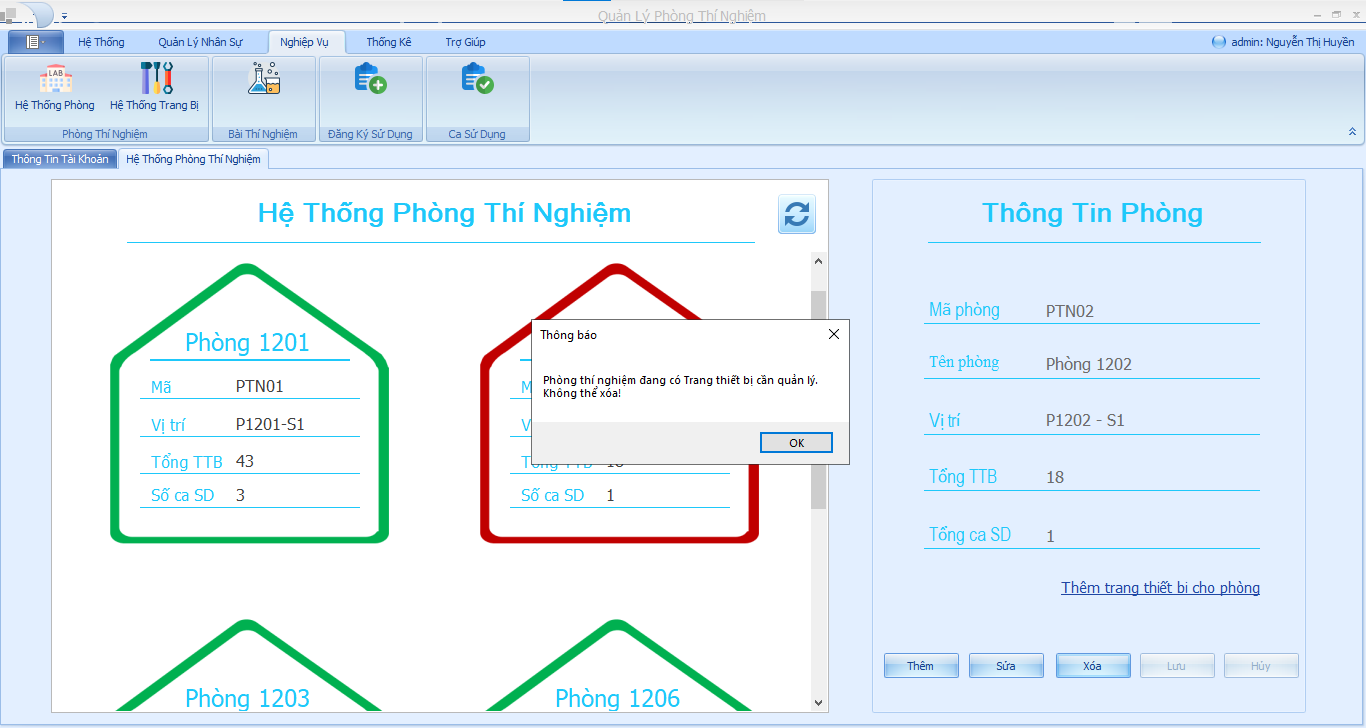
* Giao diện khi nhấn vào Button Hệ thống phòng (1) thuộc Page Nghiệp vụ.
* Nhấn vào Button (2) để tải lại thông tin Hệ thống phòng thí nghiệm.
* Có thể Click vào biểu tượng của Phòng thí nghiệm và Click vào các Button ở (3) để thêm, sửa hoặc xóa thông tin phòng.



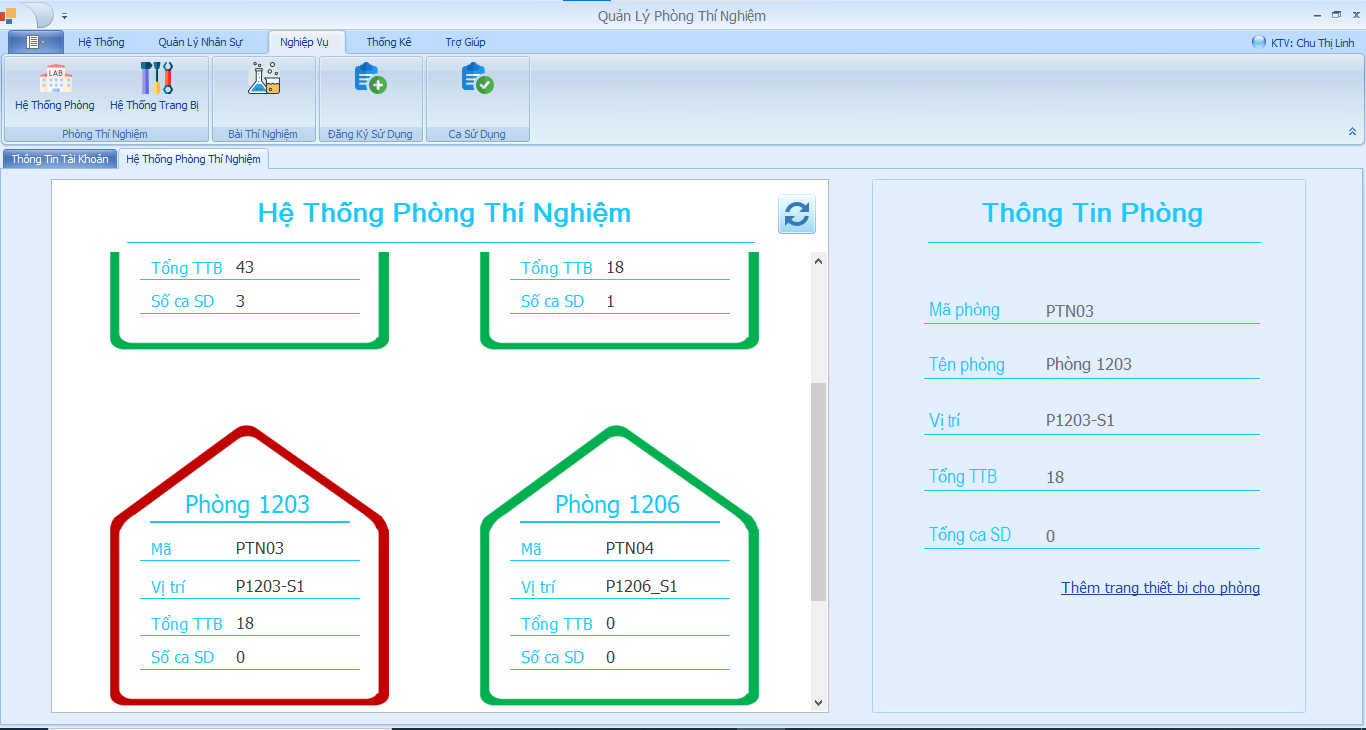
* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã phòng thí nghiệm dạng “PTNxx” như ở (1). Trong đó xx tăng tự động.
* Sau khi nhập các thông tin Tên phòng và Vị trí thì nhấn Button Lưu (2) để thêm phòng thí nghiệm.
* Khi không muốn thêm phòng thí nghiệm nữa thì nhấn Button Hủy (2).



* Giao diện khi nhấn vào Button Sửa.
* Chỉ sửa các thông tin Tên Phòng và Vị trí.
* Nhấn Button Lưu để lưu các thông tin đã chỉnh sửa.



* Khi nhấn vào Button Xóa, sẽ kiểm tra xem Phòng thí nghiệm có TTB hay không. Nếu có TTB thì không xóa được. Nếu không có TTB thì xóa Phòng đó và các ChiTietDangKy, ChiTietSuDung thuộc phòng đó. (dùng trigger)
* Đăng nhập với quyền: KTV hoặc Giáo viên



* Khi đăng nhập với quyền KTV hoặc Giáo viên thì chỉ xem được danh sách Phòng thí nghiệm. Không thể thêm, sửa, xóa Phòng thí nghiệm.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetPhongThiNghiem]

CREATE PROC [dbo].[spGetPhongThiNghiem]

-- Lấy danh sách phòng thí nghiệm

AS

BEGIN

SELECT

PTN.MaPhong,

PTN.TenPhong,

PTN.ViTri,

(

SELECT COUNT(\*)

FROM dbo.TrangThietBi TTB

WHERE TTB.MaPhong = PTN.MaPhong

) AS TongTTB,

(

SELECT COUNT(\*)

FROM dbo.ChiTietDangKi CTDK

WHERE CTDK.MaPTN = PTN.MaPhong

) AS TongCSD

FROM dbo.PhongThiNghiem PTN

END

GO

* Stored procedure [dbo].[spGetMaPhongThiNghiem]

CREATE PROC [dbo].[spGetMaPhongThiNghiem]

-- Tạo mã PTN có dạng 'PTNxx'

-- Trong đố XX là số tăng tự động

AS

BEGIN

DECLARE @MaPTN varchar(10)

DECLARE @MaPTNMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã có PTN thì xx bằng mã TPN lớn nhất + 1

IF EXISTS( SELECT MAPhong FROM dbo.PhongThiNghiem)

BEGIN

SELECT @MaPTNMax = MAX(MaPhong) FROM dbo.PhongThiNghiem

SET @Max = CONVERT(int, SUBSTRING(@MaPTNMax, 4, 2)) + 1

IF @Max < 10 SET @MaPTN = 'PTN' + '0' + CONVERT(varchar(1), @Max)

ELSE SET @MaPTN = 'PTN' + CONVERT(varchar(2), @Max)

END

-- Nếu chưa có PTN nào thì bắt đầu bằng 'PTN01'

ELSE SET @MaPTN = 'PTN01'

SELECT @MaPTN

END

GO

* Stored procedure [dbo].[spInsertPhongThiNghiem]

CREATE PROC [dbo].[spInsertPhongThiNghiem]

--Thêm phòng thí nghiệm

@MaPhong varchar(10),

@TenPhong nvarchar(50),

@ViTri nvarchar(50)

AS

BEGIN

INSERT INTO dbo.PhongThiNghiem (MaPhong, TenPhong, ViTri)

VALUES ( @MaPhong, @TenPhong, @ViTri)

END

GO

* Stored procedure [dbo].[spUpdatePhongThiNghiem]

CREATE PROC [dbo].[spUpdatePhongThiNghiem]

-- Sửa thông tin Phòng thí nghiệm

@MaPhong varchar(10),

@TenPhong nvarchar(50),

@ViTri nvarchar(50)

AS

BEGIN

UPDATE dbo.PhongThiNghiem SET

TenPhong = @TenPhong,

ViTri = @ViTri

WHERE MaPhong = @MaPhong

END

GO

* Stored procedure [dbo].[spDeletePhongThiNghiem]

CREATE PROC [dbo].[spDeletePhongThiNghiem]

-- Xóa phòng thí nghiệm Có Mã PTN truyền vào

@MaPhong varchar(10)

AS

BEGIN

DELETE FROM dbo.PhongThiNghiem WHERE MaPhong = @MaPhong

END

GO

CREATE TRIGGER [dbo].[tgDeletePhongThiNghiem]

ON [dbo].[PhongThiNghiem]

INSTEAD OF DELETE

-- Trigger xóa PTN:

-- Xóa chi tiết đăng ký thuộc Phòng trước

-- Xóa Phòng sau

AS

BEGIN

DECLARE @MaPhong varchar(20)

SELECT @MaPhong = MaPhong FROM deleted

DELETE FROM dbo.ChiTietDangKi WHERE MaPTN = @MaPhong

DELETE FROM dbo.PhongThiNghiem WHERE MaPhong = @MaPhong

END

GO

* Trigger [dbo].[tgDeletePhongThiNghiem]

CREATE TRIGGER [dbo].[tgDeletePhongThiNghiem]

ON [dbo].[PhongThiNghiem]

INSTEAD OF DELETE

-- Trigger xóa PTN:

-- Xóa chi tiết đăng ký thuộc Phòng trước

-- Xóa Phòng sau

AS

BEGIN

DELETE FROM dbo.ChiTietDangKi WHERE MaPTN IN (SELECT MaPhong FROM deleted)

DELETE FROM dbo.PhongThiNghiem WHERE MaPhong IN (SELECT MaPhong FROM deleted)

END

GO

* Trigger [dbo].[tgDeleteChiTietDangKi]

CREATE TRIGGER [dbo].[tgDeleteChiTietDangKi]

ON [dbo].[ChiTietDangKi]

INSTEAD OF DELETE

-- Trigger xóa ChiTietDangKy:

-- Xóa chi tiết sử dụng của ChiTietDangKy trước

-- Xóa ChiTietDangKy sau

AS

BEGIN

DELETE FROM dbo.ChiTietTTBCanDung WHERE MaBTN IN (SELECT MaBTN FROM deleted)

DELETE FROM dbo.BaiThiNghiem WHERE MaBTN IN (SELECT MaBTN FROM deleted)

END

GO

1. Quản lý hệ thống Trang bị
2. Chức năng hệ thống:

*Với admin và Nhân viên:*

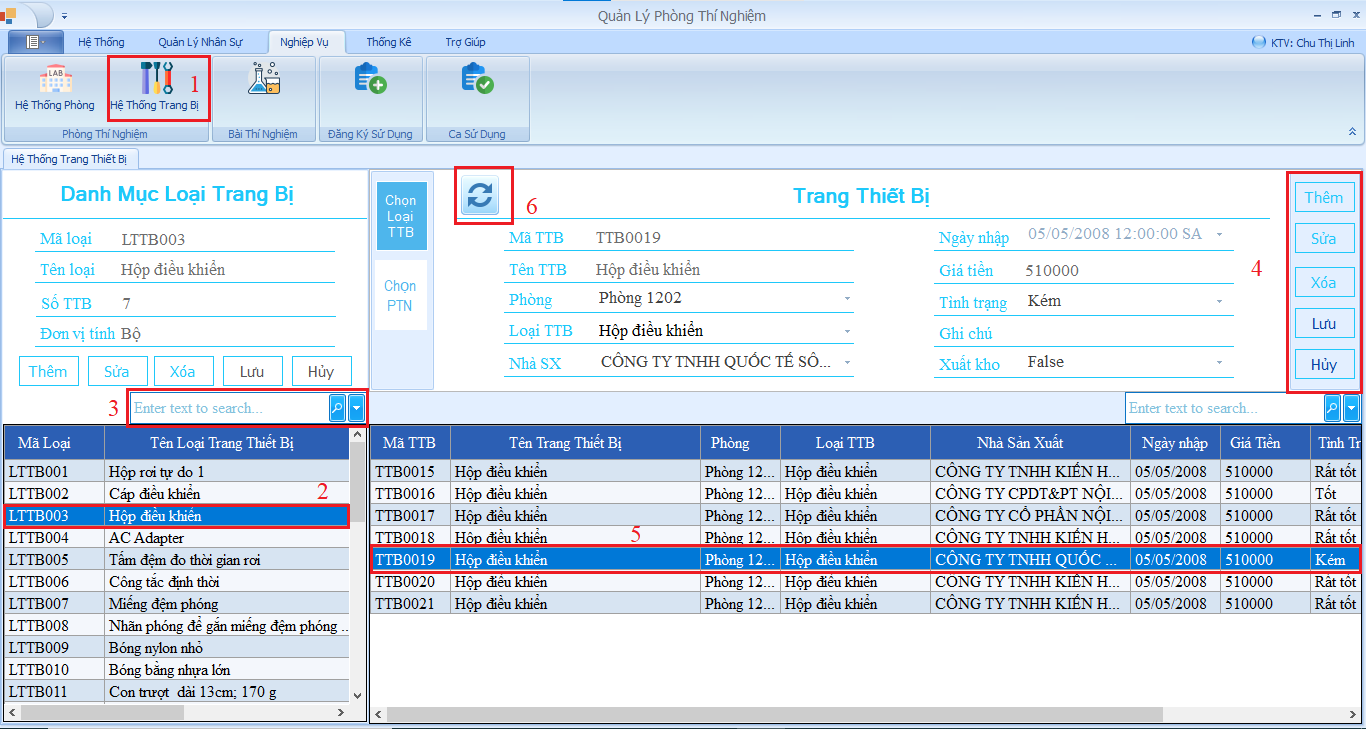
* Xem danh sách, thông tin các Loại trang thiết bị, xem Loại TTB đó có những TTB nào.
* Thêm, sửa thông tin Loại trang thiết bị.
* Xóa Loại trang thiết bị khỏi hệ thống khi Loại trang thiết bị đó không có Trang thiết bị nào trong kho và không có Bài thí nghiệm nào cần sử dụng Loại TTB đó.
* Xem danh sách, thông tin các Trang thiết bị (xem theo tất cả, xem theo Loại trang thiết bị, xem theo Phòng, hoặc xem theo Loại trang thiết bị của Phòng).
* Thêm, sửa thông tin Trang thiết bị.
* Xóa Trang thiết bị khỏi hệ thống khi Trang thiết bị đó chưa được sử dụng lần nào. (Nếu đã được sử dụng rồi thì có thể đánh dấu là Đã xuất kho).

*Với Giáo viên:*

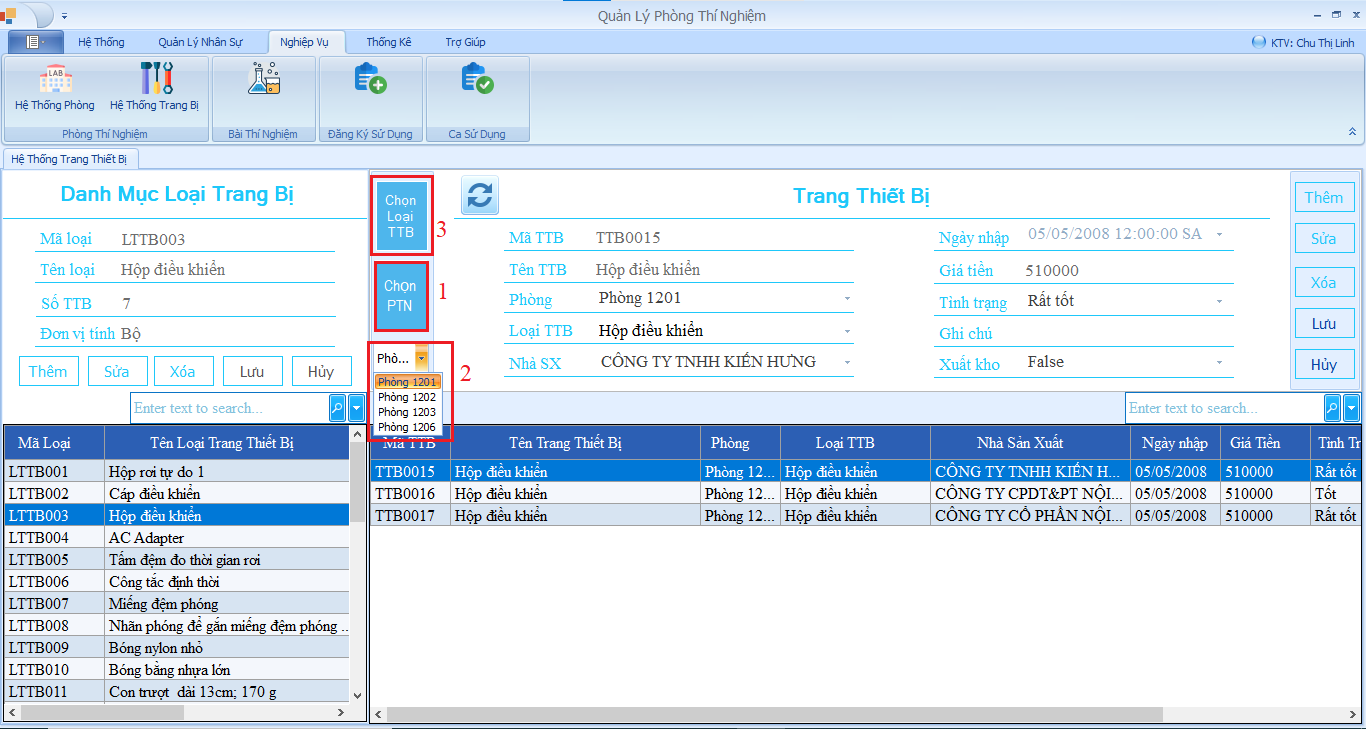
* Xem danh sách, thông tin các Loại trang thiết bị, xem Loại TTB đó có những TTB nào.
* Xem danh sách, thông tin các Trang thiết bị (xem theo tất cả, xem theo Loại trang thiết bị, xem theo Phòng, hoặc xem theo Loại trang thiết bị của Phòng).

1. Giao diện hệ thống:

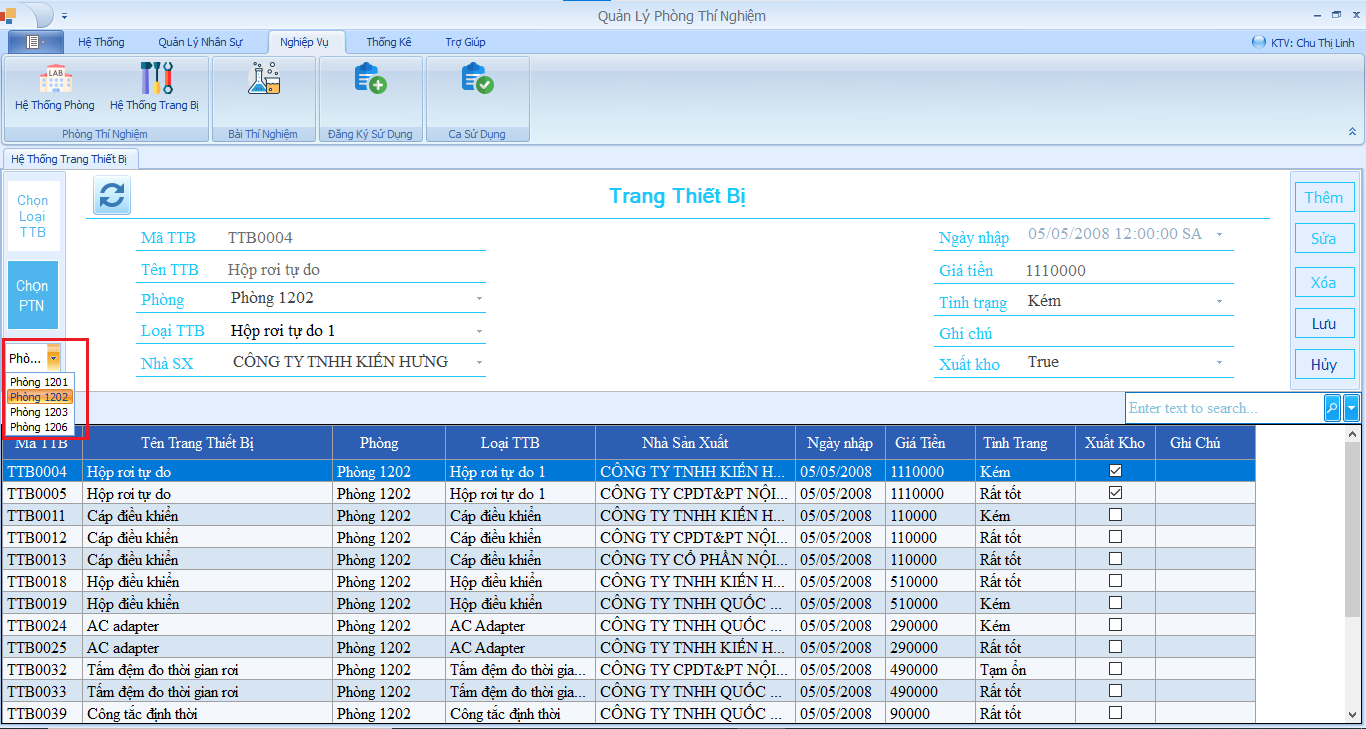
* Đăng nhập với quyền: Admin hoặc KTV



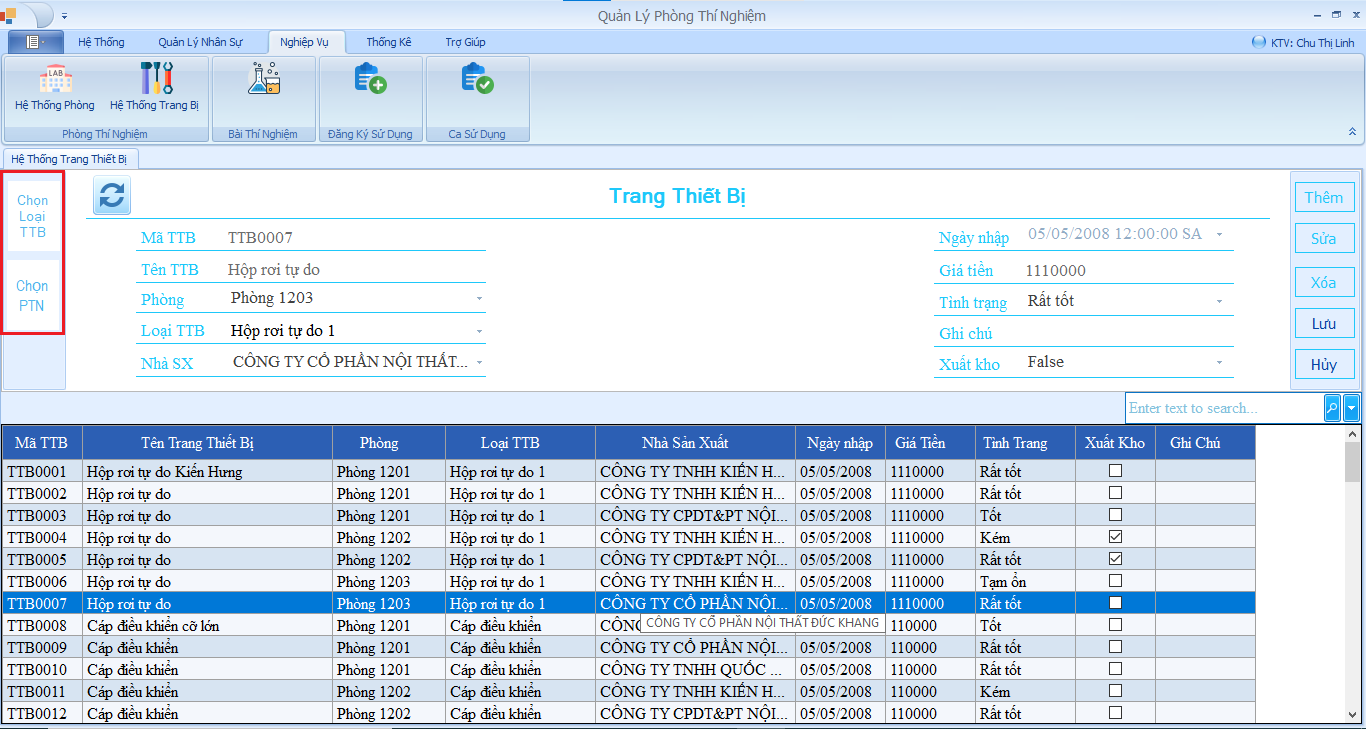
* Giao diện khi nhấn vào Button Hệ thống trang bị (1) thuộc Page Nghiệp vụ.
* Nhấn vào Row chứa Loại TTB (2) để chỉnh sửa thông tin Loại trang thiết bị đó và xem các Trang thiết bị thuộc loại đó.
* Nhấn vào ô Tìm kiếm (3) để xem danh sách Loại trang thiết bị theo từ khóa.
* Click vào các Button ở (4) để thêm, sửa hoặc xóa thông tin Trang thiết bị.
* Nhấn vào ô Tìm kiếm (3) để xem danh sách Trang thiết bị theo từ khóa.
* Nhấn vào Button (6) để tải lại thông tin hệ thống Trang bị.



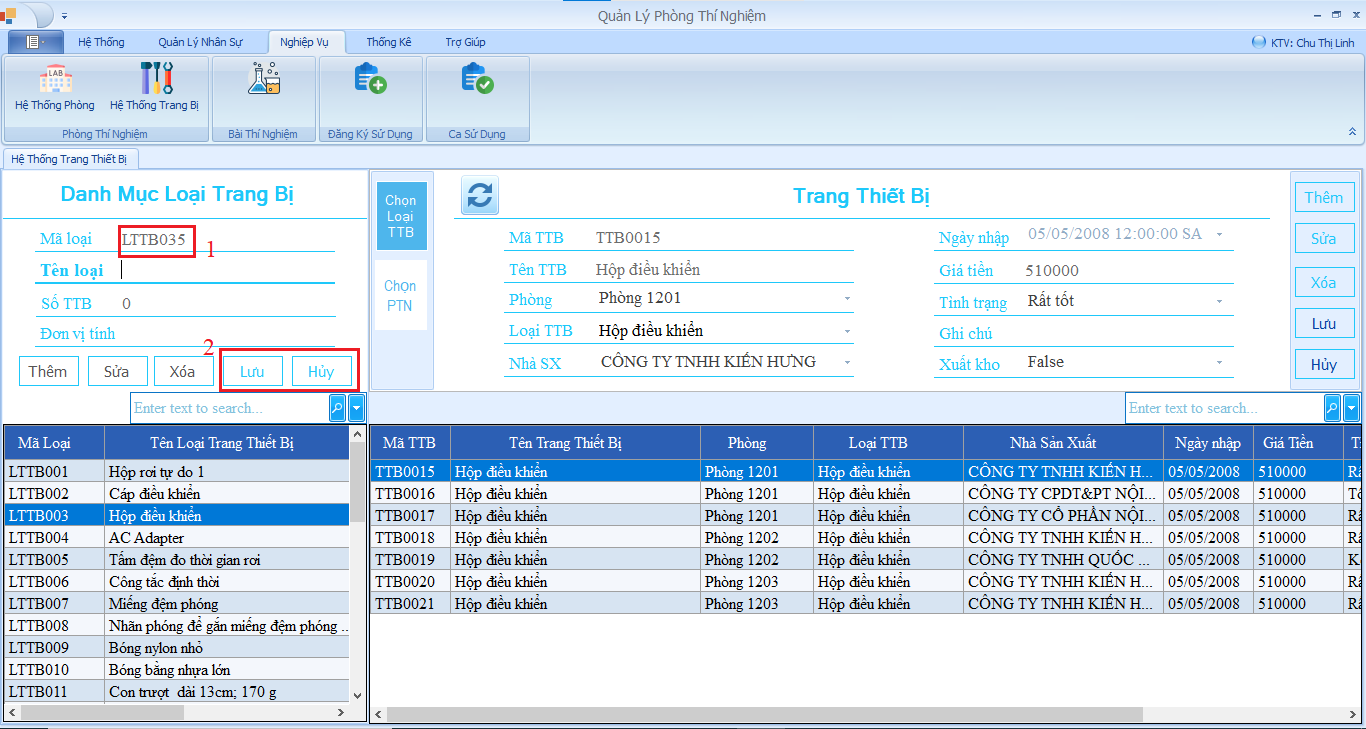
* Nhấn vào Button Chọn Phòng thí nghiệm (1) và chọn Phòng thí nghiệm cần theo dõi ở (2) để xem danh sách Trang thiết bị thuộc Loại TTB và Phòng thí nghiệm.
* Có thể nhấn Button Chọn loại TTB (3) để chọn Loại TTB hoặc không.



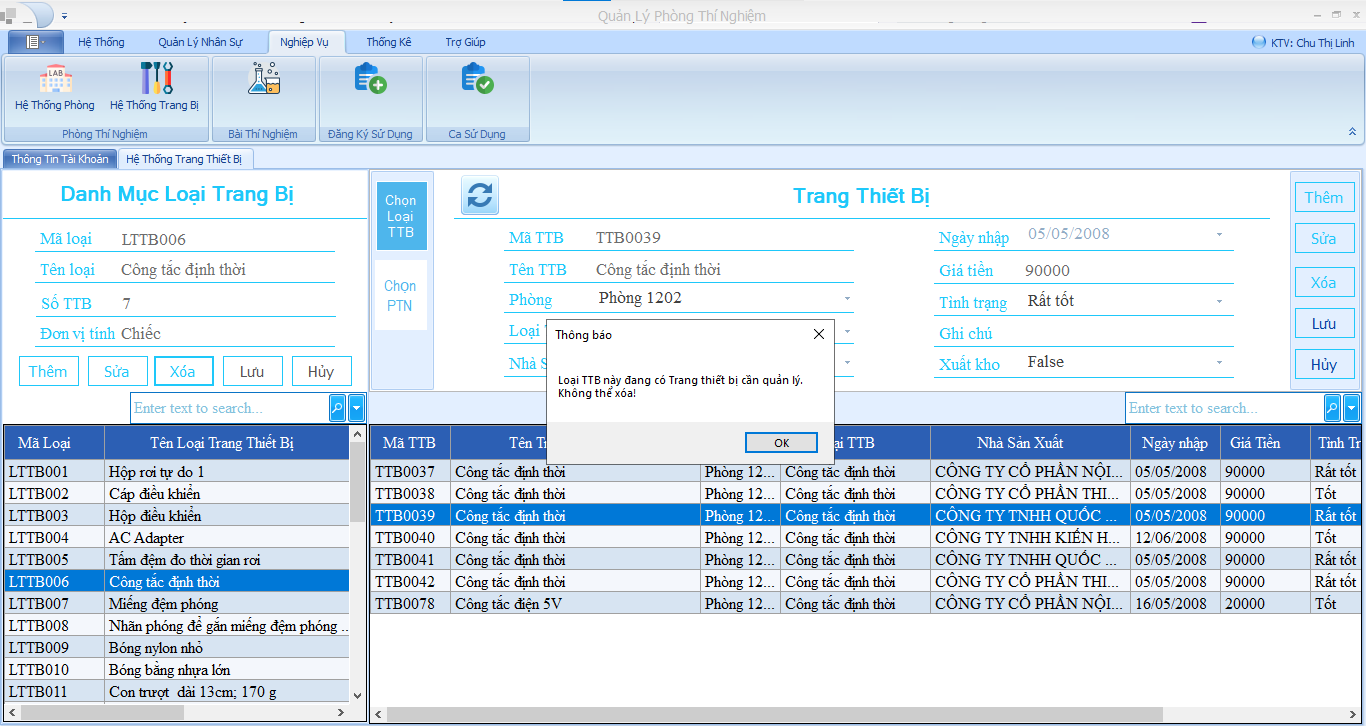
* Khi không chọn Loại TTB và chọn Phòng ta được danh sách Trang thiết bị thuộc Phòng.



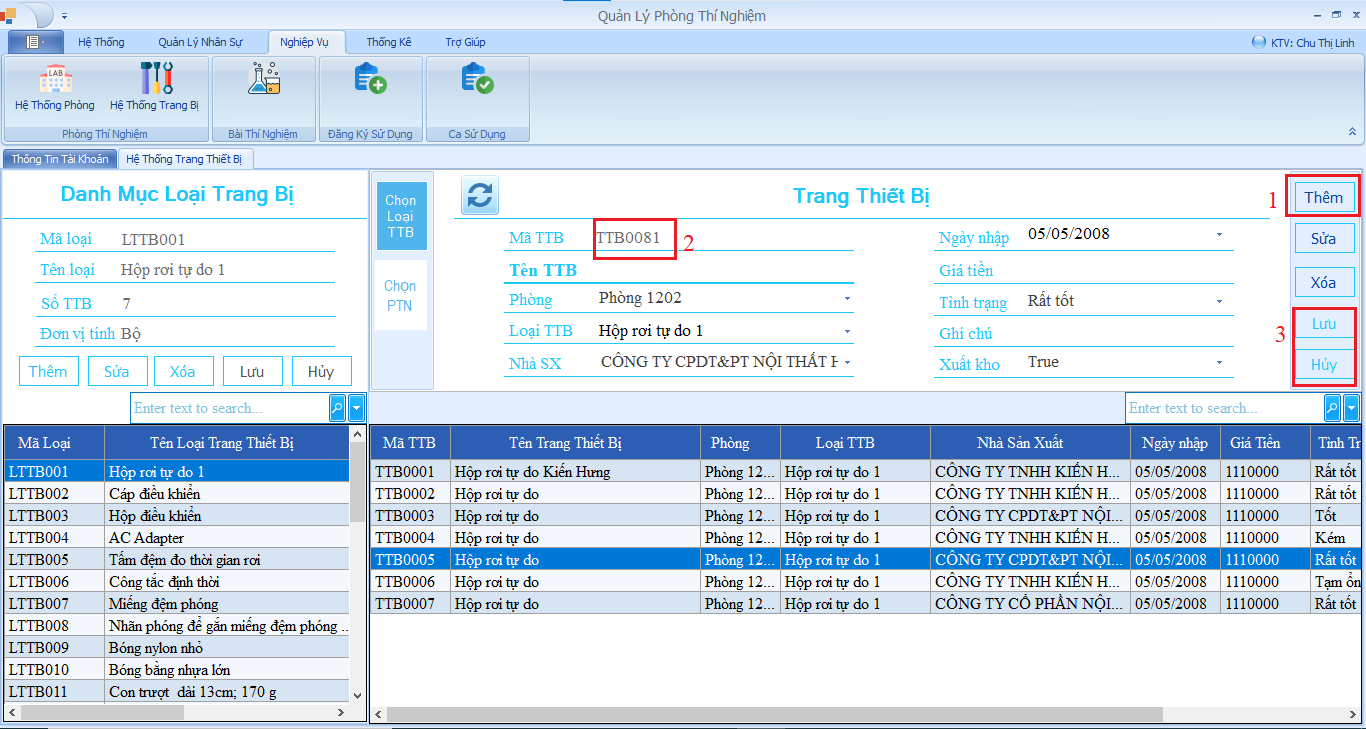
* Khi không chọn Loại TTB và không chọn Phòng ta được danh sách tất cả Trang thiết bị trong hệ thống.



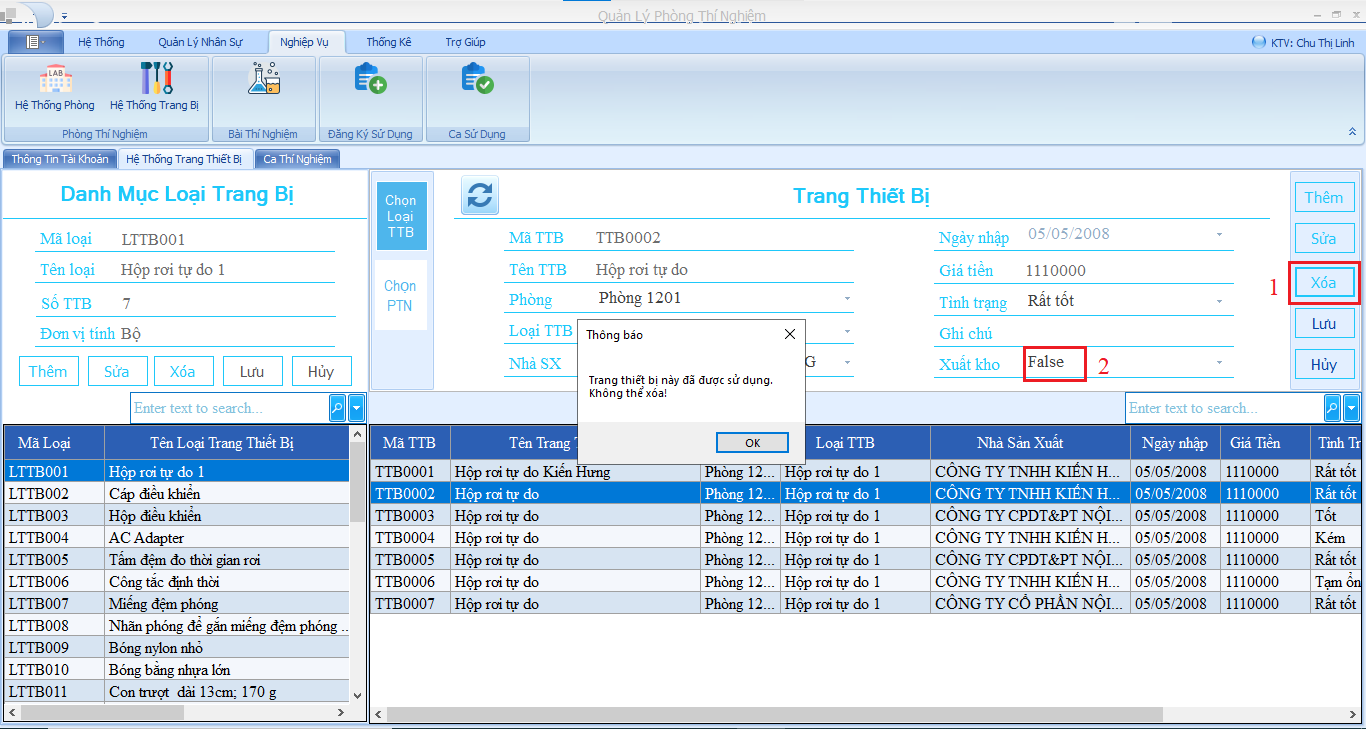
* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã Loại trang thiết bị dạng “LTTBxxx” như ở (1). Trong đó xxx tăng tự động.
* Sau khi nhập các thông tin Tên loại và Đơn vị tính thì nhấn Button Lưu (2) để thêm.
* Khi không muốn thêm nữa thì nhấn Button Hủy (2).



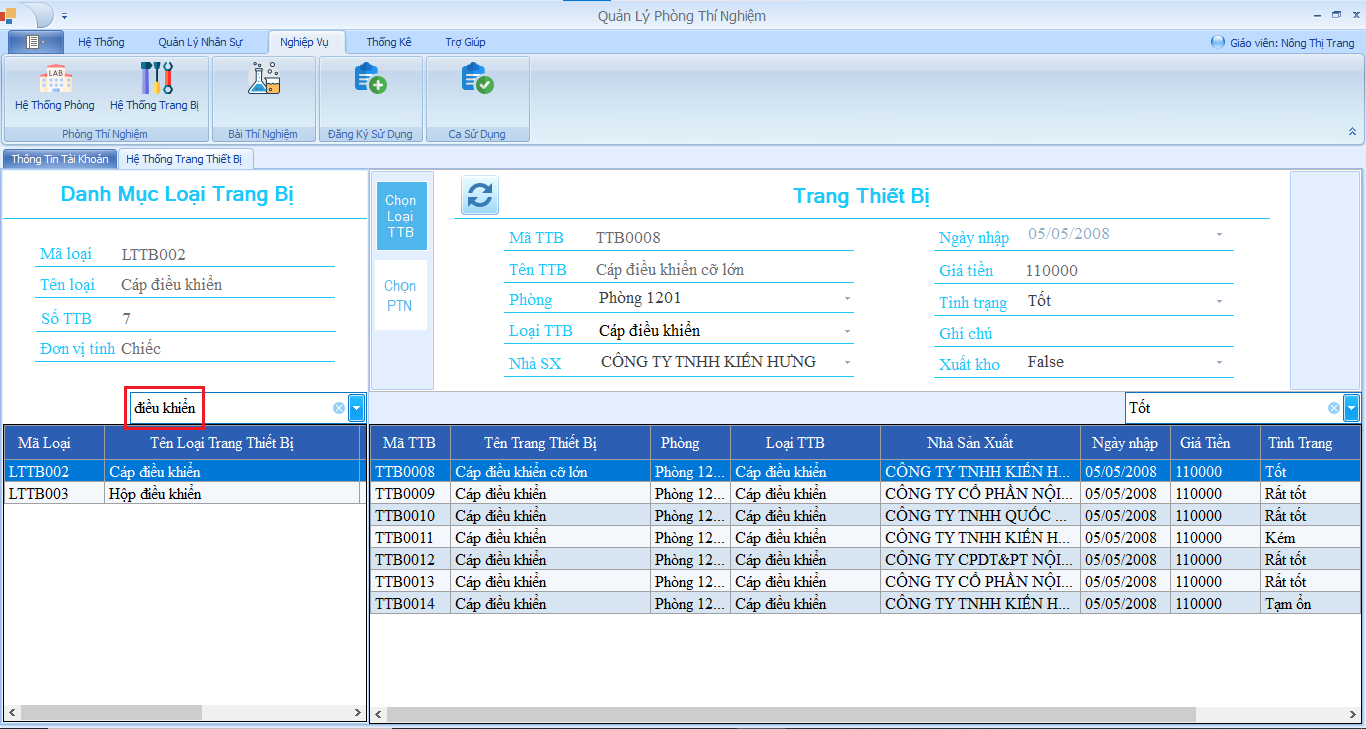
* Khi nhấn vào Button Xóa Loại TTB, sẽ kiểm tra xem Loại trang thiết bị đó có Trang thiết bị nào trong kho hoặc có Bài thí nghiệm nào cần sử dụng Loại TTB đó hay không. Nếu có thì không xóa được. Nếu không thì xóa Loại TTB đó.



* Giao diện sau khi nhấn vào Button Thêm TTB (1).
* Chương trình sẽ tự động cấp 1 Mã trang thiết bị dạng “TTBxxxx” như ở (1). Trong đó xxxx tăng tự động.
* Sau khi nhập các thông tin thì nhấn Button Lưu (2) để thêm.
* Khi không muốn thêm nữa thì nhấn Button Hủy (2).



* Khi nhấn vào Button Xóa TTB, sẽ kiểm tra xem trang thiết bị đó đã từng được sử dụng hay chưa. Nếu đã từng được sử dụng thì không xóa được.
* Đăng nhập với quyền: Giáo viên



* Không thể thêm, sửa, xóa Loại trang thiết bị và Trang thiết bị
* Có thể xem, tìm kiếm Loại TTB, Trang thiết bị theo Key.
* Ngoài ra, các chức năng khác (tìm kiếm, xem danh sách TTB thuộc Loại TTB, xem danh sách TTB thuộc Phòng thí nghiệm, …) tương tự như khi đăng nhập với quyền Admin hoặc Nhân viên.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetLoaiTrangthietBi]

CREATE PROC [dbo].[spGetLoaiTrangThietBi]

-- Lấy danh sách Loại trang thiết bị

AS

BEGIN

SELECT

LTTB.MaLTTB,

LTTB.TenLoai,

(

SELECT COUNT(ttb.MaTTB)

FROM dbo.TrangThietBi TTB

WHERE TTB.MaLTTB = LTTB.MaLTTB

) AS SoLuong,

LTTB.DonViTinh

FROM dbo.LoaiTTB LTTB

END

GO

* Stored procedure [dbo].[spGetMaLoaiTTB]

CREATE PROC [dbo].[spGetMaLoaiTTB]

-- Tạo mã Loại TTB có dạng 'LTTBxxx'

-- Trong đố XXX là số tăng tự động

AS

BEGIN

DECLARE @MaLoaiTTB varchar(10)

DECLARE @MaLoaiTTBMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã có Loại TTB thì xxx bằng mã LTTB lớn nhất + 1

IF EXISTS( SELECT MaLTTB FROM dbo.LoaiTTB)

BEGIN

SELECT @MaLoaiTTBMax = MAX(MaLTTB) FROM dbo.LoaiTTB

SET @Max = CONVERT(int, SUBSTRING(@MaLoaiTTBMax, 5, 3)) + 1

IF @Max < 10

SET @MaLoaiTTB = 'LTTB' + '00' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100)

SET @MaLoaiTTB = 'LTTB' + '0' + CONVERT(varchar(2), @Max)

ELSE

SET @MaLoaiTTB = 'LTTB' + '' + CONVERT(varchar(3), @Max)

END

-- Nếu chưa có Loại TTB nào thì bắt đầu bằng 'LTTB001'

ELSE SET @MaLoaiTTB = 'LTTB001'

SELECT @MaLoaiTTB

END

GO

* Stored procedure [dbo].[spInsertLoaiTrangThietBi]

CREATE PROC [dbo].[spInsertLoaiTrangThietBi]

-- Thêm Loại Trang thiết bị

@MaLTTB varchar(10),

@TenLoai nvarchar(50),

@DonViTinh nvarchar(50)

AS

BEGIN

INSERT INTO dbo.LoaiTTB (MaLTTB, TenLoai, DonViTinh)

VALUES ( @MaLTTB, @TenLoai, @DonViTinh)

END

GO

* Stored procedure [dbo].[spUpdateLoaiTrangthietBi]

CREATE PROC [dbo].[spUpdateLoaiTrangThietBi]

-- Sửa thông tin Loại trang thiết bị

@MaLTTB varchar(10),

@TenLoai nvarchar(50),

@DonViTinh nvarchar(50)

AS

BEGIN

UPDATE dbo.LoaiTTB SET

TenLoai = @TenLoai,

DonViTinh = @DonViTinh

WHERE MaLTTB = @MaLTTB

END

GO

* Stored procedure [dbo].[spDeleteLoaiTrangThietBi]

CREATE PROC [dbo].[spDeleteLoaiTrangThietBi]

-- Xóa Loại Trang thiết bị

@MaLTTB varchar(10)

AS

BEGIN

DELETE FROM dbo.LoaiTTB WHERE MaLTTB = @MaLTTB

END

GO

* Stored procedure [dbo].[spGetLoaiTrangThietBiByKey]

CREATE PROC [dbo].[spGetLoaiTrangThietBiByKey]

-- Tìm kiếm danh sách Loại Trang thiết bị theo Key truyền vào

@Key nvarchar(50)

AS

BEGIN

SELECT

LTTB.MaLTTB,

LTTB.TenLoai,

(

SELECT COUNT(TTB.MaTTB)

FROM dbo.TrangThietBi TTB

WHERE TTB.MaLTTB = LTTB.MaLTTB

) AS [SoLuong],

LTTB.DonViTinh

FROM dbo.LoaiTTB LTTB

WHERE LTTB.MaLTTB like ('%' + @Key + '%')

OR LTTB.TenLoai like ('%' + @Key + '%')

OR LTTB.DonViTinh like ('%' + @Key + '%')

END

GO

* Stored procedure [dbo].[spCountBTNCanDungLTTB]

CREATE PROC [dbo].[spCountBTNCanDungLTTB]

-- Đếm số lượng BTN cần dùng tới Loại TTB có mã truyền vào

-- Nếu không có BTN nào cần sử dụng Loại TTB này thì mới được phép xóa Loại TTB này

@MaLTTB varchar(10)

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.BaiThiNghiem BTN

INNER JOIN dbo.ChiTietTTBCanDung TTBCD ON TTBCD.MaBTN = BTN.MaBTN

WHERE TTBCD.MaLTTB = @MaLTTB

END

GO

* Stored procedure [dbo].[spGetTrangThietBi]

CREATE PROC [dbo].[spGetTrangThietBi]

-- Lấy danh sách Trang thiết bị

AS

BEGIN

SELECT

TTB.MaTTB,

TTB.TenTTB,

PTN.TenPhong,

LTTB.TenLoai,

TTB.TenNSX,

TTB.NgayNhap,

TTB.GiaTien,

TTB.TinhTrang,

TTB.GhiChu,

TTB.XuatKho

FROM dbo.TrangThietBi TTB

INNER JOIN dbo.PhongThiNghiem PTN ON TTB.MaPhong = PTN.MaPhong

INNER JOIN dbo.LoaiTTB LTTB ON TTB.MaLTTB = LTTB.MaLTTB

END

GO

* Stored procedure [dbo].[spGetTrangThietBiByLttbPtn]

CREATE PROC [dbo].[spGetTrangThietBiByLttbPtn]

-- Lấy danh sách Trang thiết bị thuộc Loại TTb và thuộc Phòng

@TenLoai nvarchar(50),

@TenPhong nvarchar(50)

AS

BEGIN

DECLARE

@MaLTTB varchar(10) = '',

@MaPhong varchar(10) = ''

-- Nếu @TenLoai truyền vào khác '' thì tìm theo Mã LTTB của Tên Loại TTB đó

IF (@TenLoai != '')

SELECT @MaLTTB = MaLTTB

FROM dbo.LoaiTTB

WHERE TenLoai = @TenLoai

-- Nếu @TenPhòng truyền vào khác '' thì tìm theo Mã LTTB của Tên Phòng đó

IF (@TenPhong != '')

SELECT @MaPhong = MaPhong

FROM dbo.PhongThiNghiem

WHERE TenPhong = @TenPhong

SELECT

TTB.MaTTB,

TTB.TenTTB,

PTN.TenPhong,

LTTB.TenLoai,

TTB.TenNSX,

TTB.NgayNhap,

TTB.GiaTien,

TTB.TinhTrang,

TTB.GhiChu,

TTB.XuatKho

FROM dbo.TrangThietBi TTB

INNER JOIN dbo.PhongThiNghiem PTN ON ttb.MaPhong = PTN.MaPhong

INNER JOIN dbo.LoaiTTB LTTB ON TTB.MaLTTB = LTTB.MaLTTB

WHERE LTTB.MaLTTB like ('%' + @MaLTTB + '%')

AND PTN.MaPhong like ('%' + @MaPhong + '%')

-- Nếu @MaLTTB = '' thì lấy theo tất cả các Loại TTB

-- Nếu @MaPhong = '' thì lấy theo tất cả các Phòng

END

GO

* Stored procedure [dbo].[spGetMaTrangThietBi]

CREATE PROC [dbo].[spGetMaTrangThietBi]

-- Tạo mã Trang thiết bị có dạng 'TTBxxxx'

-- Trong đố XXXX là số tăng tự động

AS

BEGIN

DECLARE @MaTTB varchar(10)

DECLARE @MaTTBMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã có Trang thiết bị thì xxxx bằng Trang Thiết bị lớn nhất + 1

IF EXISTS( SELECT MaTTB FROM dbo.TrangThietBi)

BEGIN

SELECT @MaTTBMax = MAX(MaTTB) FROM dbo.TrangThietBi

SET @Max = CONVERT(int, SUBSTRING(@MaTTBMax, 4, 4)) + 1

IF @Max < 10

SET @MaTTB = 'TTB' + '000' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100)

SET @MaTTB = 'TTB' + '00' + CONVERT(varchar(2), @Max)

ELSE IF (@Max < 100)

SET @MaTTB = 'TTB' + '0' + CONVERT(varchar(3), @Max)

ELSE

SET @MaTTB = 'TTB' + '' + CONVERT(varchar(4), @Max)

END

-- Nếu chưa có Loại TTB nào thì bắt đầu bằng 'TTB0001'

ELSE SET @MaTTB = 'TTB0001'

SELECT @MaTTB

END

GO

* Stored procedure [dbo].[spInsertTrangthietBi]

CREATE PROC [dbo].[spInsertTrangThietBi]

-- Thêm Trang thiết bị

@MaTTB varchar(10),

@TenLoai nvarchar(50),

@TenNSX nvarchar(50),

@TenPhong nvarchar(50),

@TenTTB nvarchar(50),

@NgayNhap date,

@GiaTien int,

@TinhTrang nvarchar(50),

@GhiChu nvarchar(100),

@XuatKho bit

AS

BEGIN

DECLARE

@MaLTTB varchar(10),

@MaPhong varchar(10)

SELECT @MaLTTB = LTTB.MaLTTB

FROM dbo.LoaiTTB LTTB

WHERE LTTB.TenLoai = @TenLoai

SELECT @MaPhong = PTN.MaPhong

FROM dbo.PhongThiNghiem PTN

WHERE PTN.TenPhong = @TenPhong

INSERT INTO dbo.TrangThietBi (MaTTB, MaLTTB, TenNSX, MaPhong, TenTTB, NgayNhap, GiaTien, TinhTrang, GhiChu, XuatKho)

VALUES (@MaTTB, @MaLTTB, @TenNSX, @MaPhong, @TenTTB, @NgayNhap, @GiaTien, @TinhTrang, @GhiChu, @XuatKho)

END

GO

* Stored procedure [dbo].[spUpdateTrangThietBi]

CREATE PROC [dbo].[spUpdateTrangThietBi]

-- Sửa thông tin Trang thiết bị

@MaTTB varchar(10),

@TenLoai nvarchar(50),

@TenNSX nvarchar(50),

@TenPhong nvarchar(50),

@TenTTB nvarchar(50),

@NgayNhap date,

@GiaTien int,

@TinhTrang nvarchar(50),

@GhiChu nvarchar(100),

@XuatKho bit

AS

BEGIN

DECLARE

@MaLTTB varchar(10),

@MaPhong varchar(10)

SELECT @MaLTTB = LTTB.MaLTTB

FROM dbo.LoaiTTB LTTB

WHERE LTTB.TenLoai = @TenLoai

SELECT @MaPhong = PTN.MaPhong

FROM dbo.PhongThiNghiem PTN

WHERE PTN.TenPhong = @TenPhong

UPDATE dbo.TrangThietBi SET

MaLTTB = @MaLTTB,

TenNSX = @TenNSX,

MaPhong = @MaPhong,

TenTTB = @TenTTB,

NgayNhap = @NgayNhap,

GiaTien = @GiaTien,

TinhTrang = @TinhTrang,

GhiChu = @GhiChu,

XuatKho = @XuatKho

WHERE MaTTB = @MaTTB

END

GO

* Stored procedure [dbo].[spDeleteTrangthietBi]

CREATE PROC [dbo].[spDeleteTrangThietBi]

-- Xóa Trang thiết bị

@MaTTB varchar(10)

AS

BEGIN

DELETE FROM dbo.TrangThietBi WHERE MaTTB = @MaTTB

END

GO

* Stored procedure [dbo].[spGetTrangthietBiByKey]

CREATE PROC [dbo].[spGetTrangThietBiByKey]

-- Tìm kiến danh sách Trang Thiết Bị theo Key

@TenLoai nvarchar(50),

@TenPhong nvarchar(50),

@Key nvarchar(50)

AS

BEGIN

SELECT

TTB.MaTTB,

TTB.TenTTB,

PTN.TenPhong,

LTTB.TenLoai,

TTB.TenNSX,

TTB.NgayNhap,

TTB.GiaTien,

TTB.TinhTrang,

TTB.GhiChu,

TTB.XuatKho

FROM dbo.TrangThietBi TTB

INNER JOIN dbo.PhongThiNghiem PTN ON TTB.MaPhong = PTN.MaPhong

INNER JOIN dbo.LoaiTTB LTTB ON TTB.MaLTTB = LTTB.MaLTTB

WHERE

LTTB.TenLoai like ('%' + @TenLoai + '%')

AND PTN.TenPhong like ('%' + @TenPhong + '%')

AND

(

TTB.MaTTB like ('%' + @Key + '%')

OR TTB.TenTTB like ('%' + @Key + '%')

OR PTN.TenPhong like ('%' + @Key + '%')

OR LTTB.TenLoai like ('%' + @Key + '%')

OR TTB.TenNSX like ('%' + @Key + '%')

OR TTB.NgayNhap like ('%' + @Key + '%')

OR TTB.GiaTien like ('%' + @Key + '%')

OR TTB.TinhTrang like ('%' + @Key + '%')

OR TTB.GhiChu like ('%' + @Key + '%')

)

END

GO

* Stored procedure [dbo].[spCountSoLuotSDTTB]

CREATE PROC [dbo].[spCountSoLuotSDTTB]

-- Đếm số số lượt sử dụng của TTB có mã truyền vào

-- Nếu TTB chưa từng được SD thì mới được xóa TTB này

@MaTTB varchar(10)

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.ChiTietSuDung CTSD

WHERE CTSD.MaTTB = @MaTTB

END

GO

1. Quản lý các Bài thí nghiệm
2. Chức năng hệ thống:

*Với admin hoặc Giáo viên:*

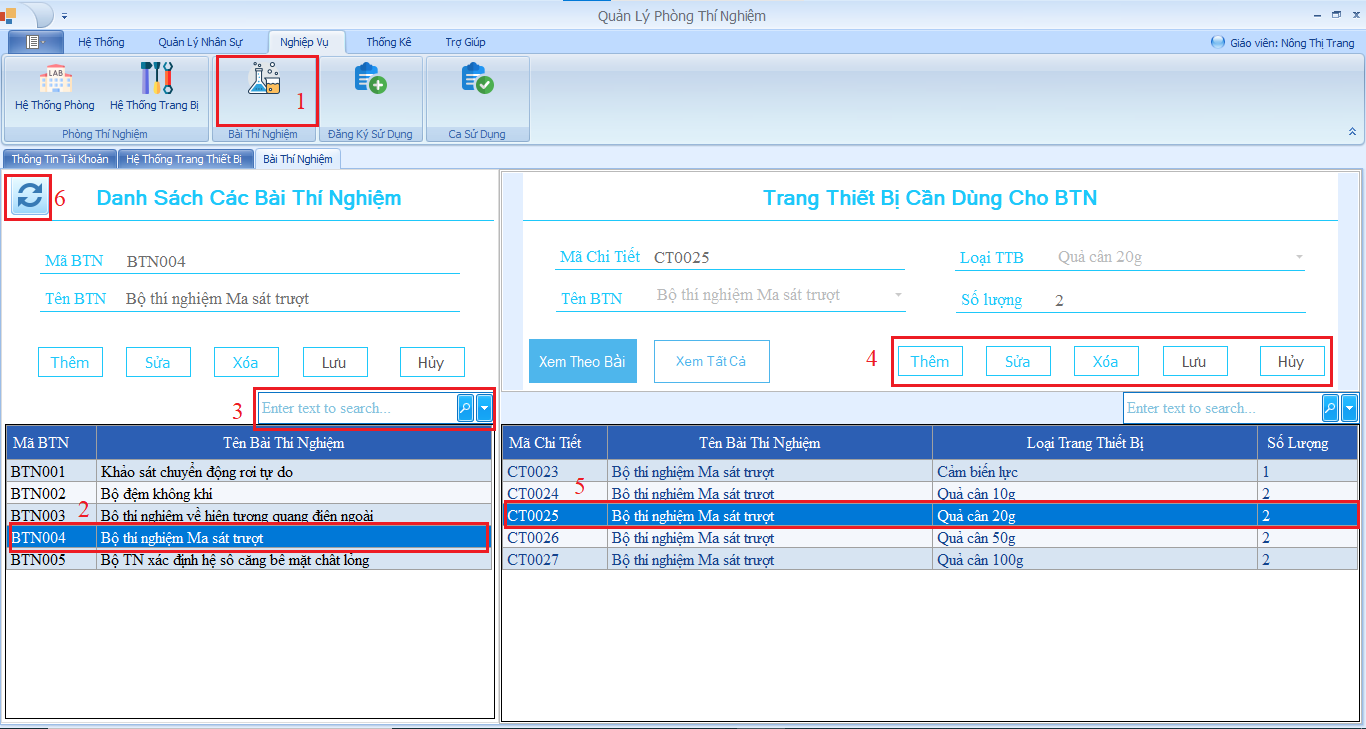
* Xem danh sách các Bài thí nghiệm và danh sách Trang thiết bị cần dùng cho Bài thí nghiệm đó.
* Thêm, sửa thông tin Bài thí nghiệm.
* Xóa Bài thí nghiệm khỏi hệ thống khi nó chưa được đăng ký thí nghiệm lần nào.
* Khi xóa Bài thí nghiệm thì sẽ xóa Chi tiết TTB cần dùng (dùng trigger).
* Thêm, sửa thông tin danh sách Trang thiết bị cần dùng cho Bài thí nghiệm.

*Với Nhân viên:*

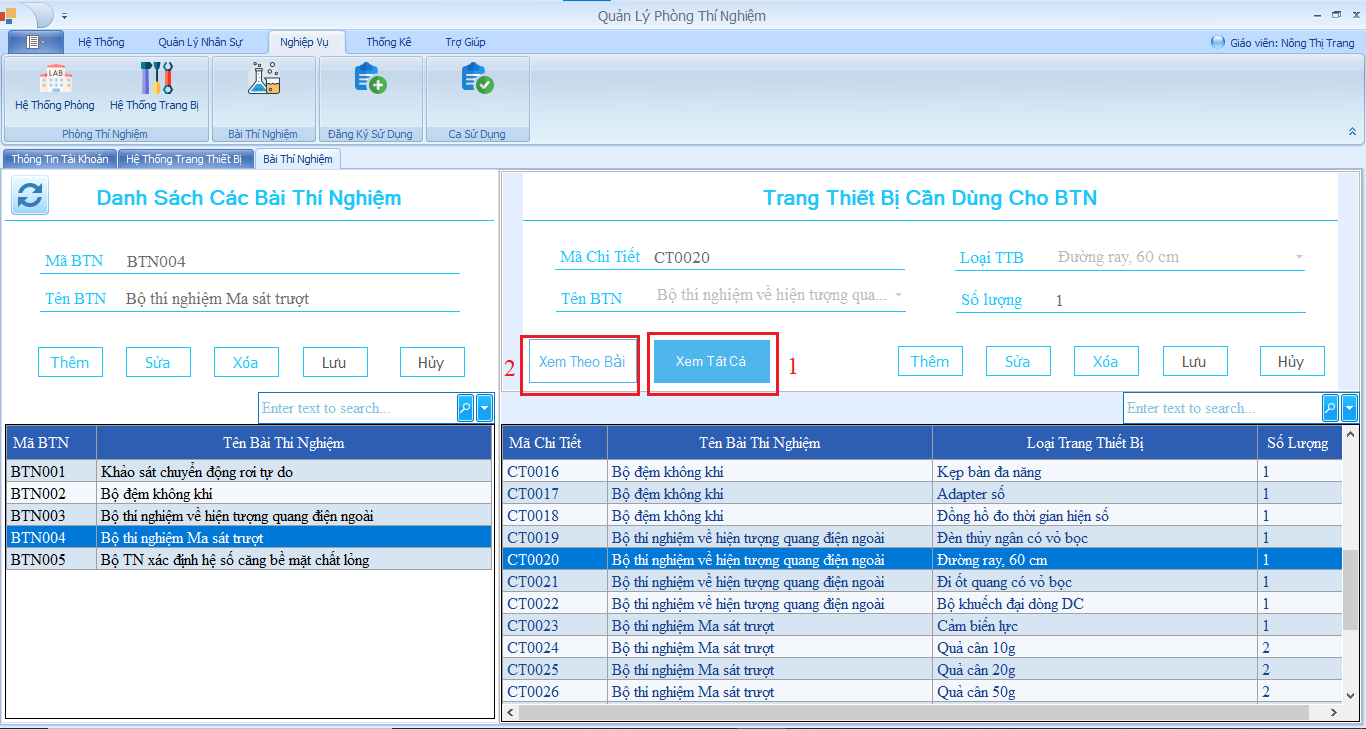
* Xem danh sách các Bài thí nghiệm và danh sách Trang thiết bị cần dùng cho Bài thí nghiệm đó.

1. Giao diện hệ thống:

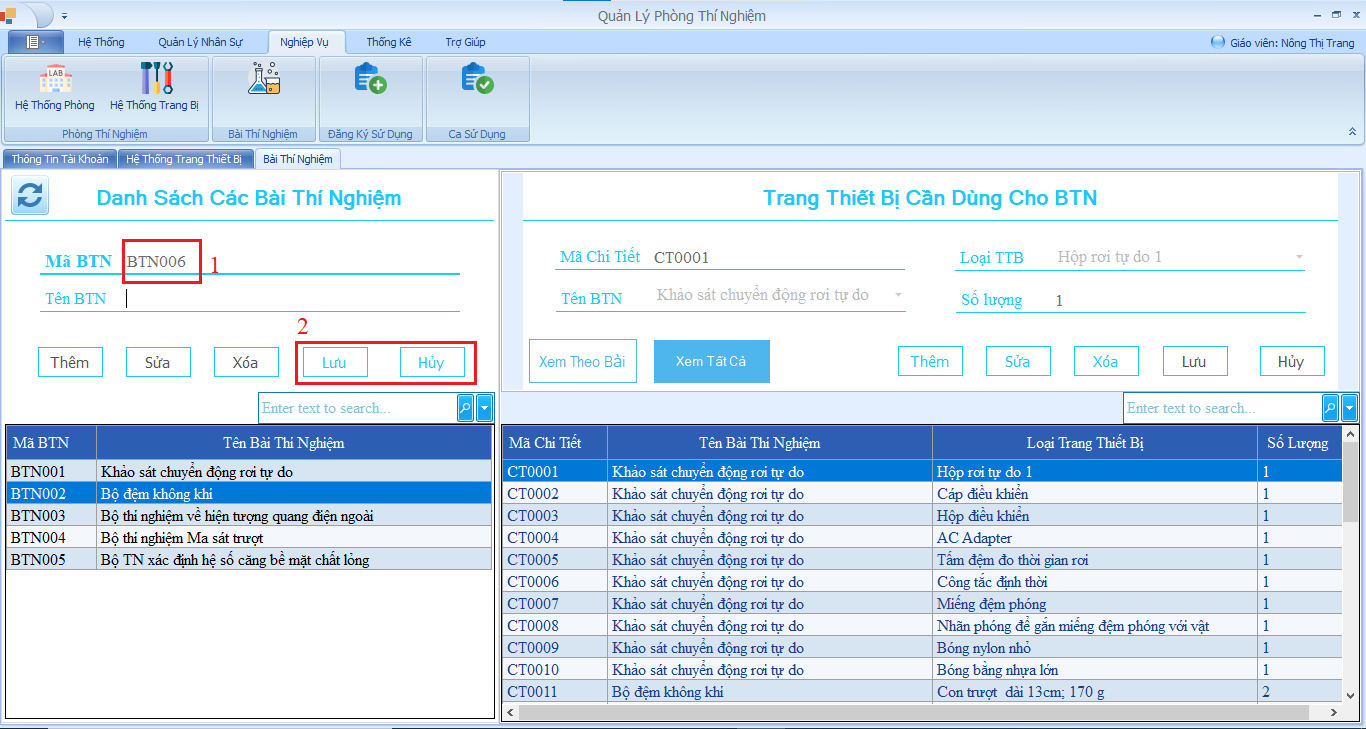
* Đăng nhập với quyền: Admin hoặc Giáo viên



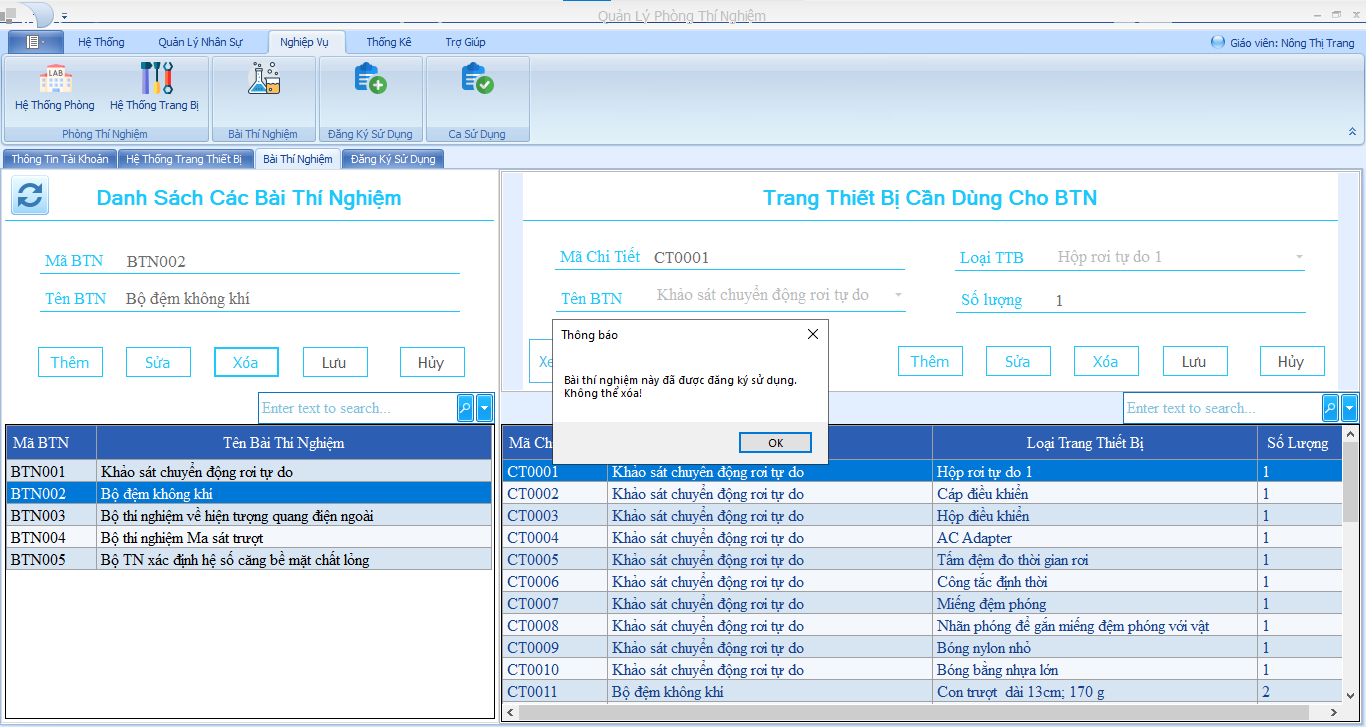
* Giao diện khi nhấn vào Button Hệ Bài thí nghiệm (1) thuộc Page Nghiệp vụ.
* Nhấn vào Row chưá Bài thí nghiệm (2) để xem thông tin Bài thí nghiệm đó và xem các Trang thiết bị cần dùng cho bài đó.
* Nhấn vào ô Tìm kiếm (3) để xem danh sách Bài thí nghiệm theo từ khóa.
* Click vào các Button ở (4) để thêm, sửa hoặc xóa thông tin Trang thiết bị cần dùng cho BTN.
* Nhấn vào ô Tìm kiếm (5) để xem danh sách Trang thiết bị cần dùng theo từ khóa.
* Nhấn vào Button (6) để tải lại thông tin hệ thống Bài thí nghiệm.



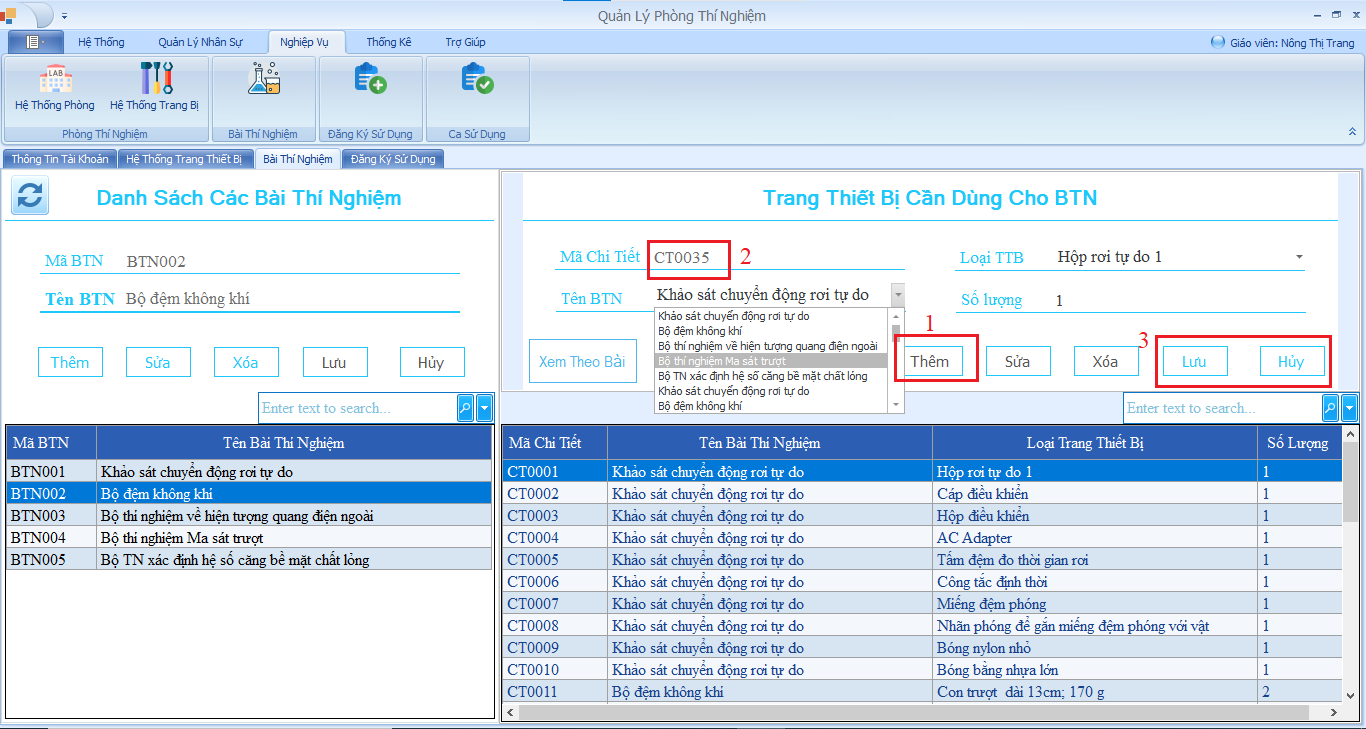
* Nhấn (1) để xem toàn bộ các Trang thiết bị cần dùng cho các Bài thí nghiệm.
* Nhấn (2) để xem các Trang thiết bị cần dùng cho 1 bài.



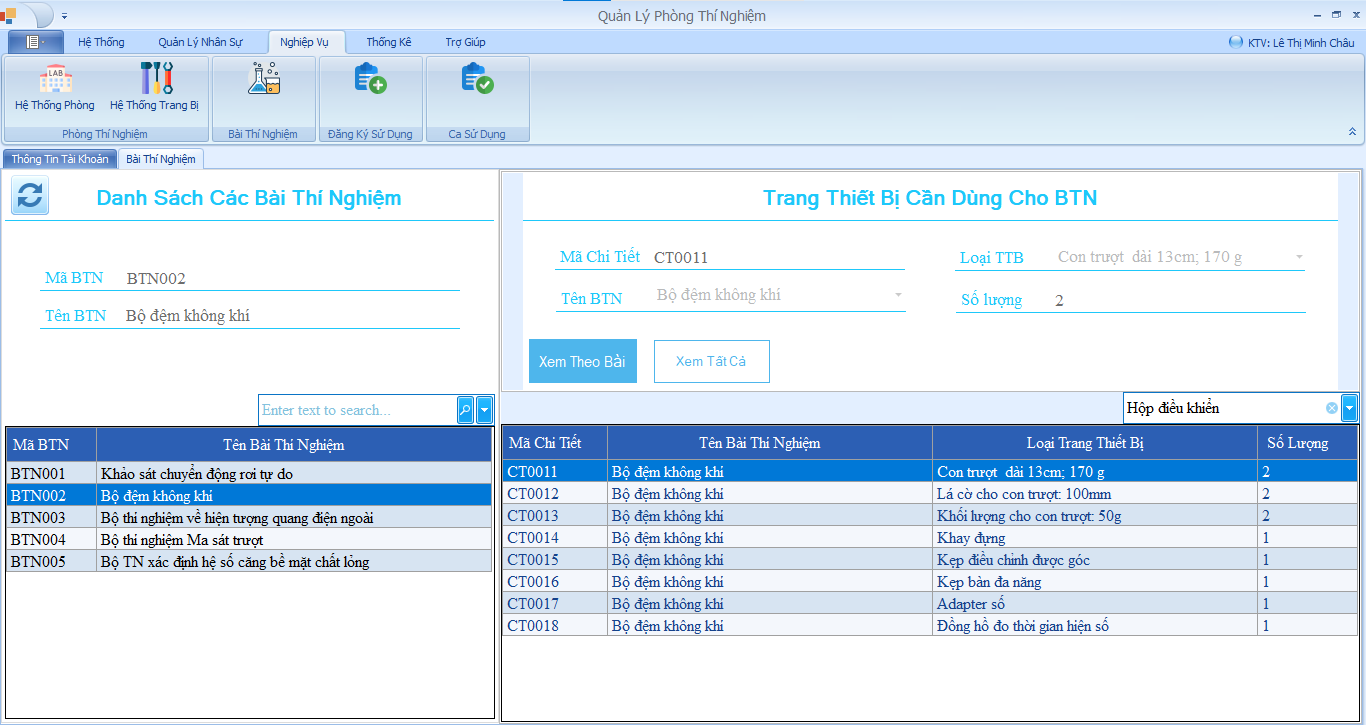
* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã bài thí nghiệm bị dạng “BTNxxx” như ở (1). Trong đó xxx tăng tự động.
* Sau khi nhập các thông tin Tên bài thì nhấn Button Lưu (2) để thêm.
* Khi không muốn thêm nữa thì nhấn Button Hủy (2).



* Khi nhấn vào Button Xóa Bài thí nghiệm, sẽ kiểm tra xem Bài thí nghiệm đó đã từng được đăng kí làm thí nghiệm hay chưa. Nếu có thì không xóa được.



* Giao diện sau khi nhấn vào Button Thêm (1).
* Chương trình sẽ tự động cấp 1 Mã chi tiết TTB cần dùng dạng “CTxxxx” như ở (1). Trong đó xxxx tăng tự động.
* Sau khi nhập các thông tin thì nhấn Button Lưu (2) để thêm.
* Khi không muốn thêm nữa thì nhấn Button Hủy (2).
* Đăng nhập với quyền: Nhân viên



* Không thể thêm, sửa, xóa Bài thí nghiệm và Chi tiết TTB cần dùng cho BTN.
* Có thể xem, tìm kiếm Bài thí nghiệm, Chi tiết TTB cần dùng theo Key.
* Ngoài ra, các chức năng khác (tìm kiếm, xem danh sách TTB cần dùng cho 1 bài thí nghiệm, xem danh sách TTB cần dùng cho tất cả các Bài thí nghiệm, …) tương tự như khi đăng nhập với quyền Admin hoặc Giáo viên.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetBaiThiNghiem]

CREATE PROC [dbo].[spGetBaiThiNghiem]

-- Lấy danh sách Bài thí nghiệm

AS

BEGIN

SELECT

BTN.MaBTN,

BTN.TenBTN

FROM dbo.BaiThiNghiem BTN

END

GO

* Stored procedure [dbo].[spGetMaBaiThiNghiem]

CREATE PROC [dbo].[spGetMaBaiThiNghiem]

-- Tạo mã BTN có dạng 'BTNxxx'

-- Trong đố XXX là số tăng tự động

AS

BEGIN

DECLARE @MaBTN varchar(10)

DECLARE @MaBTNMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã có BTN thì xxx bằng mã TPN lớn nhất + 1

IF EXISTS( SELECT MaBTN FROM dbo.BaiThiNghiem)

BEGIN

SELECT @MaBTNMax = MAX(MaBTN) FROM dbo.BaiThiNghiem

SET @Max = CONVERT(int, SUBSTRING(@MaBTNMax, 4, 3)) + 1

IF @Max < 10

SET @MaBTN = 'BTN' + '00' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100)

SET @MaBTN = 'BTN' + '0' + CONVERT(varchar(2), @Max)

ELSE IF (@Max < 1000)

SET @MaBTN = 'BTN' + '' + CONVERT(varchar(3), @Max)

END

-- Nếu chưa có BTN nào thì bắt đầu bằng 'BTN001'

ELSE SET @MaBTN = 'BTN001'

SELECT @MaBTN

END

GO

* Stored procedure [dbo].[spInsertBaiThiNghiem]

CREATE PROC [dbo].[spInsertBaiThiNghiem]

--Thêm bài thí nghiệm

@MaBTN varchar(10),

@TenBTN nvarchar(50)

AS

BEGIN

INSERT INTO dbo.BaiThiNghiem (MaBTN, TenBTN)

VALUES ( @MaBTN, @TenBTN)

END

GO

* Stored procedure [dbo].[spUpdateBaiThiNghiem]

CREATE PROC [dbo].[spUpdateBaiThiNghiem]

-- Sửa thông tin Bài thí nghiệm

@MaBTN varchar(10),

@TenBTN nvarchar(50)

AS

BEGIN

UPDATE dbo.BaiThiNghiem SET

TenBTN = @TenBTN

WHERE MaBTN = @MaBTN

END

GO

* Stored procedure [dbo].[spDeleteBaiThiNghiem]

CREATE PROC [dbo].[spDeleteBaiThiNghiem]

-- Xóa Bài thí nghiệm Có Mã BTN truyền vào

@MaBTN varchar(10)

AS

BEGIN

DELETE FROM dbo.BaiThiNghiem WHERE MaBTN = @MaBTN

END

GO

* Trigger [dbo].[tgDeleteBaiThiNghiem]

CREATE TRIGGER [dbo].[tgDeleteBaiThiNghiem]

ON [dbo].[BaiThiNghiem]

INSTEAD OF DELETE

-- Trigger xóa Bài thí nghiệm:

-- Xóa chi tiết trang thiết bị cần dùng trước

-- Xóa bài thí nghiệm sau

AS

BEGIN

DELETE FROM dbo.ChiTietTTBCanDung WHERE MaBTN IN (SELECT MaBTN FROM deleted)

DELETE FROM dbo.BaiThiNghiem WHERE MaBTN IN (SELECT MaBTN FROM deleted)

END

GO

* Stored procedure [dbo].[spCountLuotDangKyBTN]

CREATE PROC [dbo].[spCountLuotDangKyBTN]

-- Đếm số số lượt đang ký của BTN có mã truyền vào

-- Nếu BTN chưa từng được đăng ký thì mới được xóa BTN này

@MaBTN varchar(10)

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.ChiTietDangKi CTDK

WHERE CTDK.MaBTN = @MaBTN

END

GO

* Stored procedure [dbo].[spGetBaiThiNghiemByKey]

CREATE PROC [dbo].[spGetBaiThiNghiemByKey]

-- Tìm kiếm danh sách Bài thí nghiệm theo Key truyền vào

@Key nvarchar(50)

AS

BEGIN

SELECT

btn.MaBTN,

btn.TenBTN

FROM dbo.BaiThiNghiem btn

WHERE btn.MaBTN LIKE ('%' + @Key + '%')

OR btn.TenBTN LIKE ('%' + @Key + '%')

END

GO

* Stored procedure [dbo].[spGetTrangThietBiCanDung]

CREATE PROC [dbo].[spGetTrangThietBiCanDung]

-- Lấy toàn bộ danh sách Trang thiết bị cần dùng

AS

BEGIN

SELECT

TTBCD.MaChiTiet,

BTN.TenBTN,

LTTB.TenLoai,

TTBCD.SoLuong

FROM dbo.ChiTietTTBCanDung TTBCD

INNER JOIN dbo.BaiThiNghiem BTN ON TTBCD.MaBTN = BTN.MaBTN

INNER JOIN dbo.LoaiTTB LTTB ON TTBCD.MaLTTB = LTTB.MaLTTB

END

GO

* Stored procedure [dbo].[spGetTrangThietBiCanDungByMaBTN]

CREATE PROC [dbo].[spGetTrangThietBiCanDungByMaBTN]

-- Lấy danh sách Trang thiết bị cần dùng cho bài thí nghiệm có mã là @MaBTN

@MaBTN varchar(10)

AS

BEGIN

SELECT

TTBCD.MaChiTiet,

BTN.TenBTN,

LTTB.TenLoai,

TTBCD.SoLuong

FROM dbo.ChiTietTTBCanDung TTBCD

INNER JOIN dbo.BaiThiNghiem BTN ON TTBCD.MaBTN = BTN.MaBTN

INNER JOIN dbo.LoaiTTB LTTB ON TTBCD.MaLTTB = LTTB.MaLTTB

WHERE TTBCD.MaBTN = @MaBTN

END

GO

* Stored procedure [dbo].[spGetMaTrangBiCanDung]

CREATE PROC [dbo].[spGetMaTrangBiCanDung]

-- Tạo mã Trang thiết bị cần dùng có dạng 'CTxxxx'

-- Trong đố XXXX là số tăng tự động

AS

BEGIN

DECLARE @MaChiTiet varchar(10)

DECLARE @MaChiTietMax varchar(10)

DECLARE @Max varchar(10)

IF EXISTS( SELECT MaChiTiet FROM dbo.ChiTietTTBCanDung)

BEGIN

SELECT @MaChiTietMax = MAX(MaChiTiet) FROM dbo.ChiTietTTBCanDung

SET @Max = CONVERT(int, SUBSTRING(@MaChiTietMax, 3, 4)) + 1

IF @Max < 10

SET @MaChiTiet = 'CT' + '000' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100)

SET @MaChiTiet = 'CT' + '00' + CONVERT(varchar(2), @Max)

ELSE IF (@Max < 1000)

SET @MaChiTiet = 'CT' + '0' + CONVERT(varchar(3), @Max)

ELSE

SET @MaChiTiet = 'CT' + '' + CONVERT(varchar(4), @Max)

END

ELSE SET @MaChiTiet = 'CT0001'

SELECT @MaChiTiet

END

GO

* Stored procedure [dbo].[spInsertTrangthietBiCanDung]

CREATE PROC [dbo].[spInsertTrangBiCanDung]

-- Thêm Trang thiết bị cần dùng

@MaChiTiet varchar(10),

@TenBTN nvarchar(50),

@TenLoai nvarchar(50),

@SoLuong int

AS

BEGIN

DECLARE

@MaBTN varchar(10),

@MaLTTB varchar(10)

SELECT @MaLTTB = LTTB.MaLTTB

FROM dbo.LoaiTTB LTTB

WHERE LTTB.TenLoai = @TenLoai

SELECT @MaBTN = BTN.MaBTN

FROM dbo.BaiThiNghiem BTN

WHERE BTN.TenBTN = @TenBTN

INSERT INTO dbo.ChiTietTTBCanDung (MaChiTiet, MaBTN, MaLTTB, SoLuong)

VALUES (@MaChiTiet, @MaBTN, @MaLTTB, @SoLuong)

END

GO

* Stored procedure [dbo].[spUpdateTrangThietBiCanDung]

CREATE PROC [dbo].[spUpdateTrangThietBiCanDung]

-- Sửa thông tin Trang thiết bị cần dùng

@MaChiTiet varchar(10),

@TenBTN nvarchar(50),

@TenLoai nvarchar(50),

@SoLuong int

AS

BEGIN

DECLARE

@MaBTN varchar(10),

@MaLTTB varchar(10)

SELECT @MaLTTB = LTTB.MaLTTB

FROM dbo.LoaiTTB LTTB

WHERE LTTB.TenLoai = @TenLoai

SELECT @MaBTN = BTN.MaBTN

FROM dbo.BaiThiNghiem BTN

WHERE BTN.TenBTN = @TenBTN

UPDATE dbo.ChiTietTTBCanDung SET

MaBTN = @MaBTN,

MaLTTB = @MaLTTB,

SoLuong = @SoLuong

WHERE MaChiTiet = @MaChiTiet

END

GO

* Stored procedure [dbo].[spDeleteTrangthietBiCanDung]

CREATE PROC [dbo].[spDeleteTrangThietBiCanDung]

-- Xóa Trang thiết bị cần dùng

@MaChiTiet varchar(10)

AS

BEGIN

DELETE FROM dbo.ChiTietTTBCanDung WHERE MaChiTiet = @MaChiTiet

END

GO

* Stored procedure [dbo].[spGetTrangThietBiCanDungByKey]

CREATE PROC [dbo].[spGetTrangThietBiCanDungByKey]

-- Tìm kiến danh sách Trang Thiết Bị theo Key

@MaBTN varchar(10),

@Key nvarchar(50)

AS

BEGIN

SELECT

TTBCD.MaChiTiet,

BTN.TenBTN,

LTTB.TenLoai,

TTBCD.SoLuong

FROM dbo.ChiTietTTBCanDung TTBCD

INNER JOIN dbo.BaiThiNghiem BTN ON TTBCD.MaBTN = BTN.MaBTN

INNER JOIN dbo.LoaiTTB LTTB ON TTBCD.MaLTTB = LTTB.MaLTTB

WHERE

BTN.MaBTN LIKE ('%' + @MaBTN + '%')

AND

(

TTBCD.MaChiTiet LIKE ('%' + @Key + '%')

OR BTN.TenBTN LIKE ('%' + @Key + '%')

OR LTTB.TenLoai LIKE ('%' + @Key + '%')

OR TTBCD.SoLuong LIKE ('%' + @Key + '%')

)

END

GO

1. Quản lý việc đăng ký sử dụng Phòng thí nghiệm
2. Chức năng hệ thống:

*Với Giáo viên:*

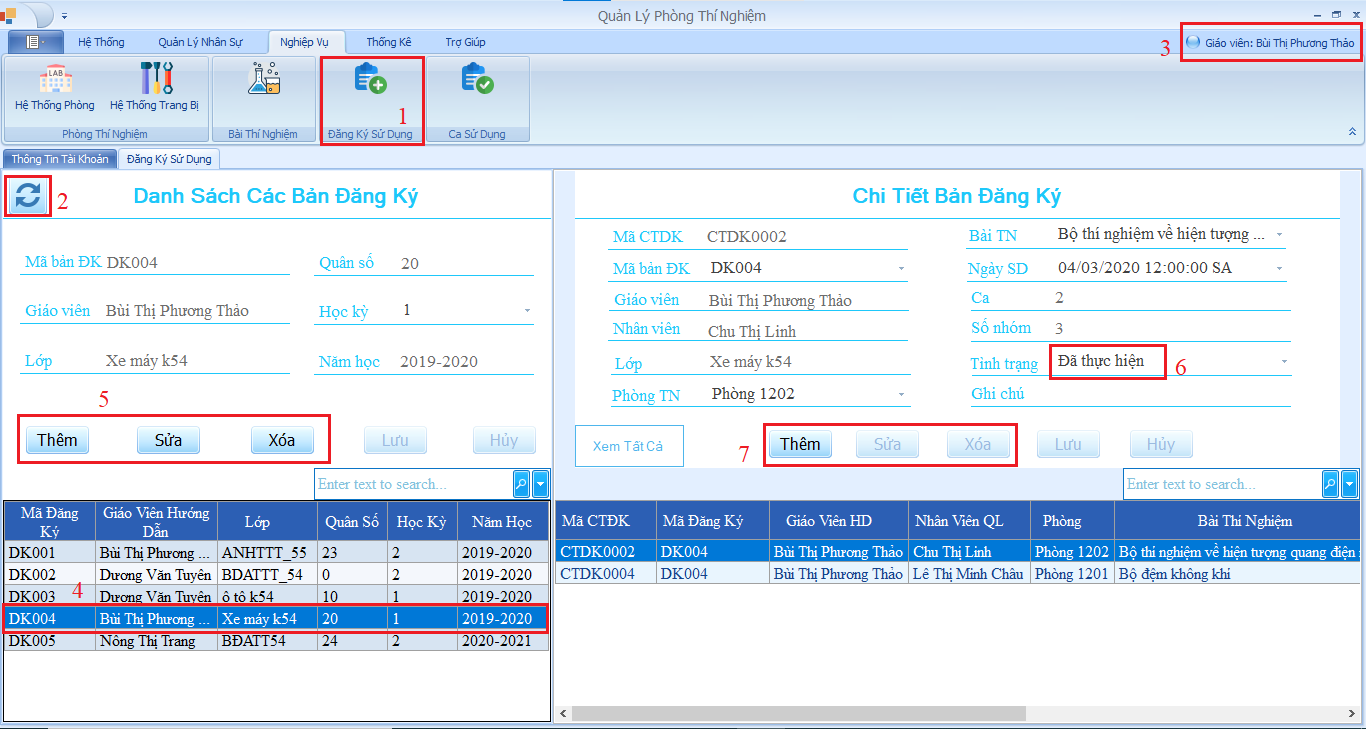
* Xem danh sách, thông tin các Bản đăng ký và Chi tiết thuộc Bản đăng ký đó.
* Thêm, sửa thông tin Bản đăng ký của mình.
* Xóa Bản đăng ký của mình khi không có Chi tiết đăng ký nào.
* Thêm thông tin Chi tiết bản đăng ký cho Bản đăng ký của mình.
* Sửa thông tin Chi tiết bản đăng ký cho Bản đăng ký của mình khi tình trạng là “Chưa thực hiện”.
* Xóa thông tin Chi tiết bản đăng ký cho Bản đăng ký của mình khi tình trạng là “Đã hủy”.

*Với Admin và Nhân viên:*

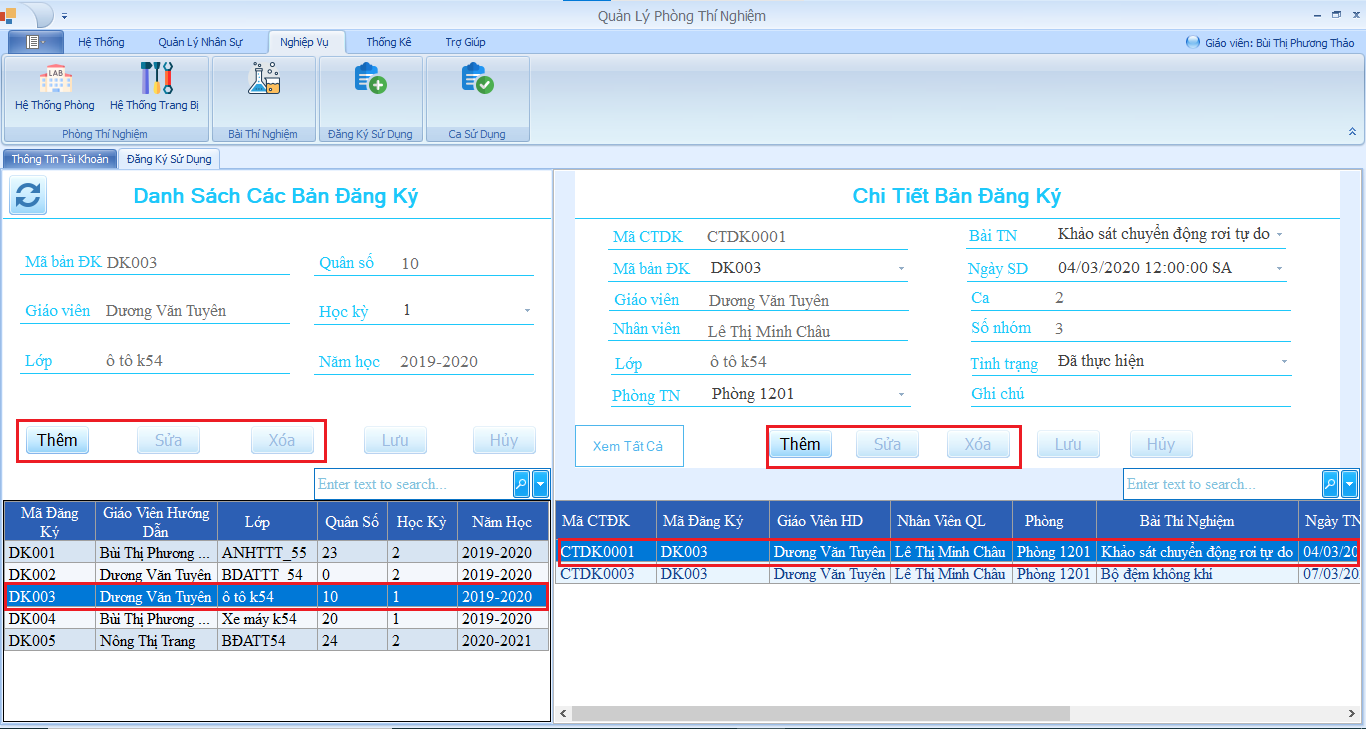
* Xem danh sách, thông tin các Bản đăng ký và Chi tiết thuộc Bản đăng ký đó.
* Thêm thông tin Chi tiết bản đăng ký mà mình quản lý cho Bản đăng ký.
* Sửa thông tin Chi tiết thuộc Bản đăng ký mà mình quản lý khi tình trạng là “Chưa thực hiện”.
* Xóa thông tin Chi tiết bản đăng ký mà mình quản lý khi tình trạng là “Đã hủy”.

1. Giao diện hệ thống:

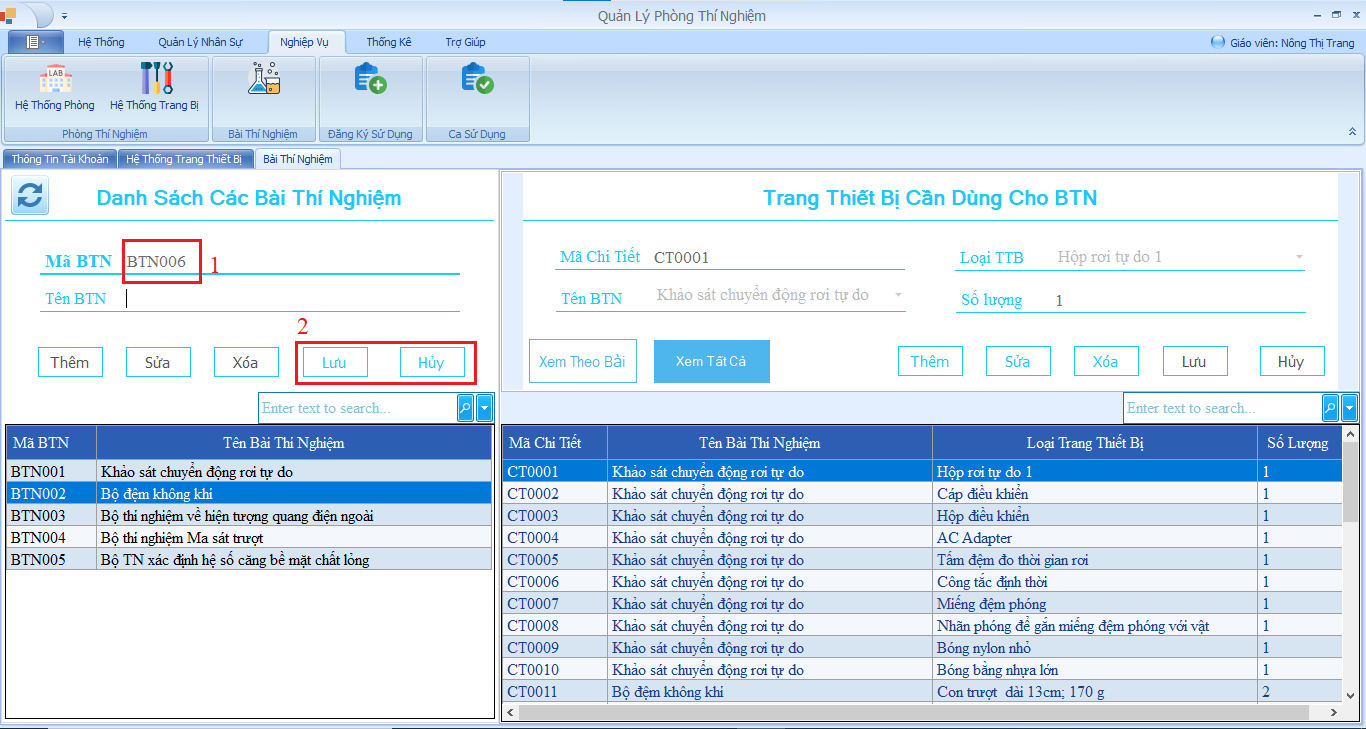
* Đăng nhập với quyền: Giáo viên



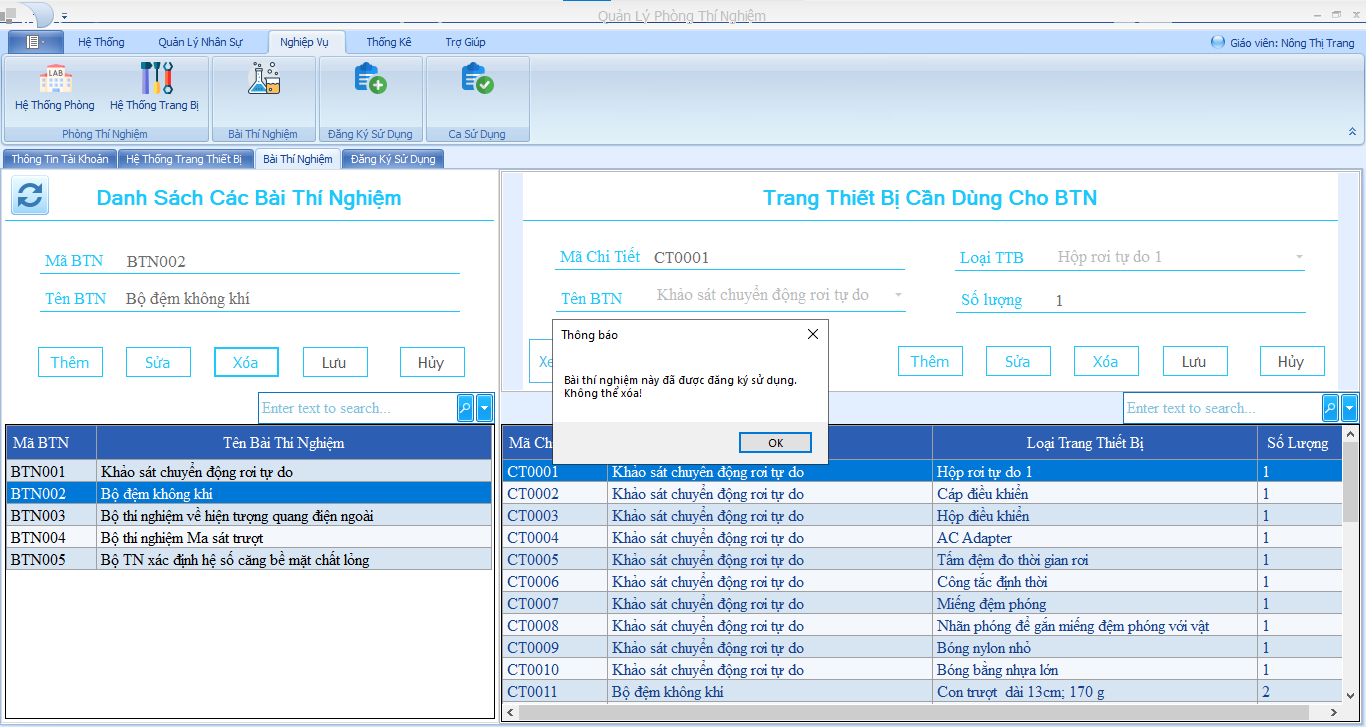
* Giao diện khi nhấn vào Button Đăng Ký Sử Dụng (1) thuộc Page Nghiệp vụ.
* Nhấn Button (2) để tải lại thông tin Đăng Ký Sử Dụng.
* Nhấn vào Row chưá Bản đăng ký (4) để xem thông tin Bản đăng ký đó và xem các Chi tiết bản đăng ký.
* Click vào các Button ở (5) để thêm, sửa hoặc xóa thông tin Bản đăng ký cần dùng cho BTN. Giáo viên (3) chỉ được sửa, xóa những Bản đăng ký của mình (4).
* Khi Chi tiết bản đăng ký là đã thực hiện (6) thì không thể sửa hay xóa (7).



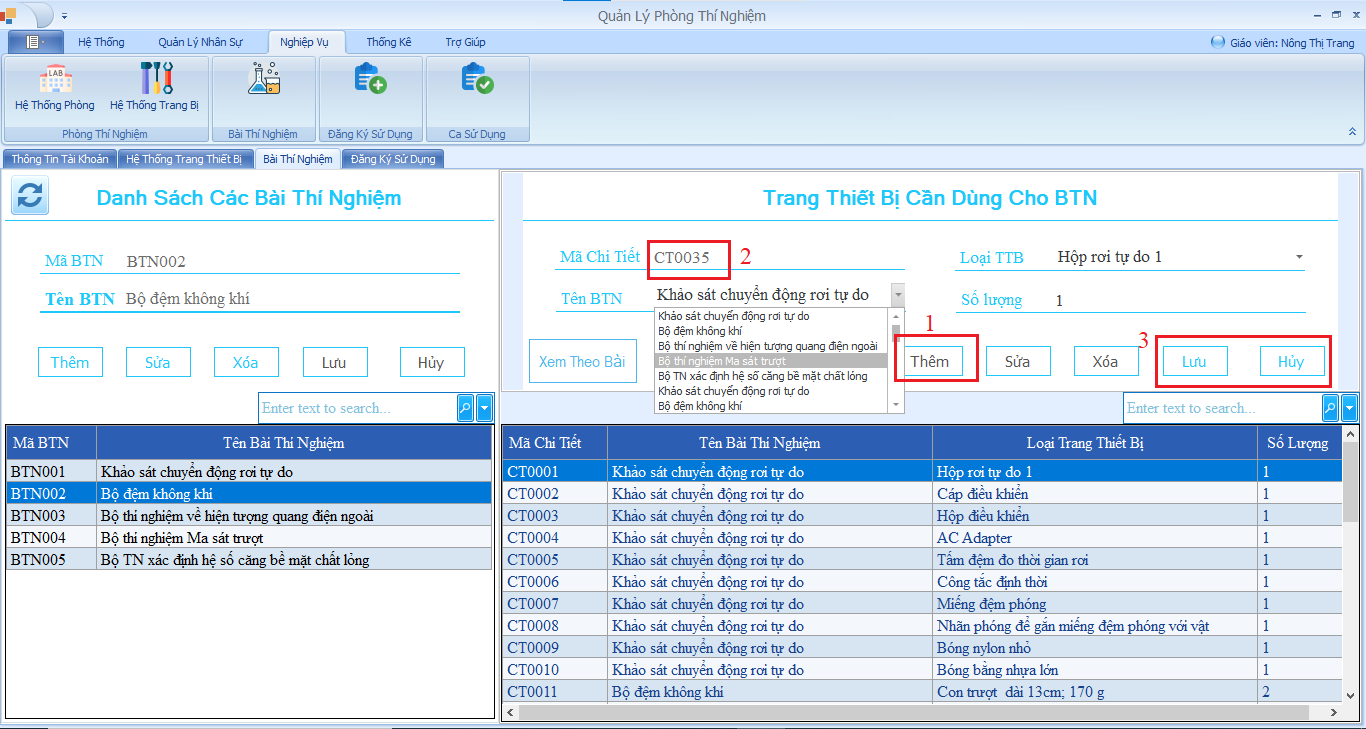
* Giáo viên không thể sửa, xóa những Bản đăng ký, Chi tiết bản đăng ký không phải của mình.



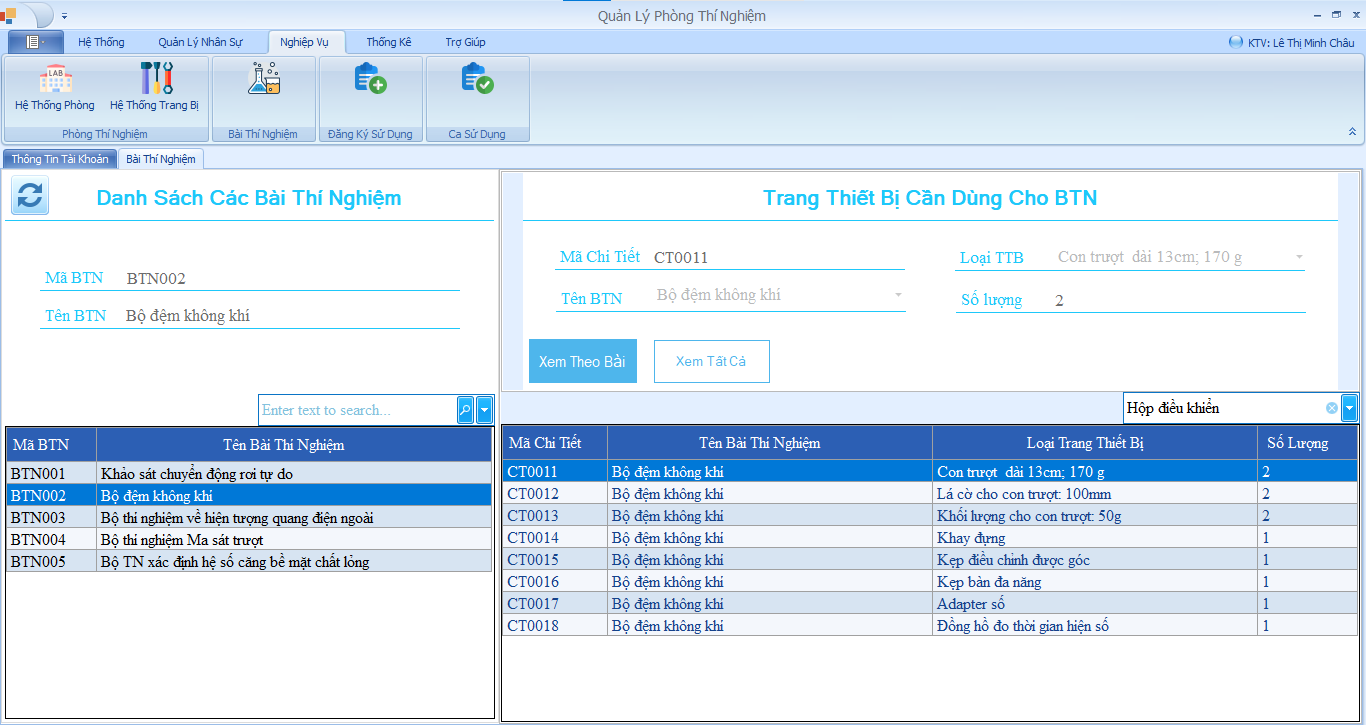
* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã bài thí nghiệm bị dạng “BTNxxx” như ở (1). Trong đó xxx tăng tự động.
* Sau khi nhập các thông tin Tên bài thì nhấn Button Lưu (2) để thêm.
* Khi không muốn thêm nữa thì nhấn Button Hủy (2).



* Khi nhấn vào Button Xóa Bài thí nghiệm, sẽ kiểm tra xem Bài thí nghiệm đó đã từng được đăng kí làm thí nghiệm hay chưa. Nếu có thì không xóa được.



* Giao diện sau khi nhấn vào Button Thêm (1).
* Chương trình sẽ tự động cấp 1 Mã chi tiết TTB cần dùng dạng “CTxxxx” như ở (1). Trong đó xxxx tăng tự động.
* Sau khi nhập các thông tin thì nhấn Button Lưu (2) để thêm.
* Khi không muốn thêm nữa thì nhấn Button Hủy (2).
* Đăng nhập với quyền: Admin hoặc Nhân viên



* Không thể thêm, sửa, xóa Bài thí nghiệm và Chi tiết TTB cần dùng cho BTN.
* Có thể xem, tìm kiếm Bài thí nghiệm, Chi tiết TTB cần dùng theo Key.
* Ngoài ra, các chức năng khác (tìm kiếm, xem danh sách TTB cần dùng cho 1 bài thí nghiệm, xem danh sách TTB cần dùng cho tất cả các Bài thí nghiệm, …) tương tự như khi đăng nhập với quyền Giáo viên.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetBanDangKy]

-- Lấy danh sách các Bản đăng ký

CREATE PROC [dbo].[spGetBanDangKy]

AS

BEGIN

SELECT

BDK.MaDangKy,

GV.HoTen,

BDK.Lop,

BDK.QuanSo,

BDK.HocKy,

BDK.NamHoc

FROM dbo.BanDangKy BDK

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

END

GO

* Stored procedure [dbo].[spGetMaBanDangKy]

-- Tạo mã Bản đăng ký có dạng 'DKxxx'

-- Trong đố XXX là số tăng tự động

CREATE PROC [dbo].[spGetMaBanDangKy]

AS

BEGIN

DECLARE @MaDangKy varchar(10)

DECLARE @MaDangKyMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã tồn tại Bản đăng ký thì lấy MaBDK bằng MaBDK lớn nhất cộng 1

IF EXISTS( SELECT MaDangKy FROM dbo.BanDangKy)

BEGIN

SELECT @MaDangKyMax = MAX(MaDangKy) FROM dbo.BanDangKy

SET @Max = CONVERT(int, SUBSTRING(@MaDangKyMax, 3, 3)) + 1

IF @Max < 10 SET @MaDangKy = 'DK' + '00' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100) SET @MaDangKy = 'DK' + '0' + CONVERT(varchar(2), @Max)

ELSE IF (@Max < 1000) SET @MaDangKy = 'DK' + '' + CONVERT(varchar(3), @Max)

END

-- Nếu chưa tồn tại bản đăng ký nào thì lấy mã bản đăng ký là DK001

ELSE SET @MaDangKy = 'DK001'

SELECT @MaDangKy

END

GO

* Stored procedure [dbo].[spInsertBanDangKy]

--Thêm Bản đăng ký

CREATE PROC [dbo].[spInsertBanDangKy]

@MaDangKy varchar(10),

@HoTen nvarchar(50),

@Lop nvarchar(50),

@QuanSo int,

@HocKy int,

@NamHoc nvarchar(50)

AS

BEGIN

DECLARE @MaGV varchar(10)

SELECT @MaGV = MaGV FROM dbo.GiaoVien GV WHERE GV.HoTen = @HoTen

INSERT INTO dbo.BanDangKy (MaDangKy, MaGV, Lop, QuanSo, HocKy, NamHoc)

VALUES ( @MaDangKy, @MaGV, @Lop, @QuanSo, @HocKy, @NamHoc)

END

GO

* Stored procedure [dbo].[spUpdateBanDangKy]

-- Sửa thông tin Bản đăng ký

CREATE PROC [dbo].[spUpdateBanDangKy]

@MaDangKy varchar(10),

@HoTen nvarchar(50),

@Lop nvarchar(50),

@QuanSo int,

@HocKy int,

@NamHoc nvarchar(50)

AS

BEGIN

DECLARE @MaGV varchar(10)

SELECT @MaGV = MaGV FROM dbo.GiaoVien GV WHERE GV.HoTen = @HoTen

UPDATE dbo.BanDangKy SET

MaGV = @MaGV,

Lop = @Lop,

QuanSo = @QuanSo,

HocKy = @HocKy,

NamHoc = @NamHoc

WHERE MaDangKy = @MaDangKy

END

GO

* Stored procedure [dbo].[spDeleteBanDangKy]

-- Xóa Bản đăng ký có Mã đăng ký truyền vào

CREATE PROC [dbo].[spDeleteBanDangKy]

@MaDangKy varchar(10)

AS

BEGIN

DELETE FROM dbo.BanDangKy WHERE MaDangKy = @MaDangKy

END

GO

* Stored procedure [dbo].[spCountChiTietCuaBDK]

CREATE PROC [dbo].[spCountChiTietCuaBDK]

-- Đếm số lượng chi tiết đăng ký của 1 bản đăng ký

-- Nếu bản đăng ký chưa có chi tiết đăng ký nào thì mới được phép xóa

@MaDangKy varchar(10)

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.ChiTietDangKi CTDK

WHERE CTDK.MaDangKy = @MaDangKy

END

GO

* Stored procedure [dbo].[spGetBanDangKyByKey]

-- Tìm kiếm danh sách Bản đăng ký theo Key truyền vào

CREATE PROC [dbo].[spGetBanDangKyByKey]

@Key nvarchar(50)

AS

BEGIN

SELECT

BDK.MaDangKy,

GV.HoTen,

BDK.Lop,

BDK.QuanSo,

BDK.HocKy,

BDK.NamHoc

FROM dbo.BanDangKy BDK

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

WHERE BDK.MaDangKy LIKE ('%' + @Key + '%')

OR GV.HoTen LIKE ('%' + @Key + '%')

OR BDK.Lop LIKE ('%' + @Key + '%')

OR BDK.QuanSo LIKE ('%' + @Key + '%')

OR BDK.HocKy LIKE ('%' + @Key + '%')

OR BDK.NamHoc LIKE ('%' + @Key + '%')

END

GO

* Stored procedure [dbo].[spGetChiTietDangKy]

-- Lấy Chi tiết Bản đăng ký

CREATE PROC [dbo].[spGetChiTietDangKy]

AS

BEGIN

SELECT

CTDK.MaCTDK,

BDK.MaDangKy,

GV.HoTen AS HoTenGV,

NV.HoTen AS HoTenNV,

PTN.TenPhong,

BTN.TenBTN,

CTDK.NgaySD,

BDK.Lop,

CTDK.CaTrongNgay,

CTDK.SoNhom,

CTDK.TinhTrang,

CTDK.GhiChu

FROM dbo.ChiTietDangKi CTDK

INNER JOIN dbo.PhongThiNghiem PTN ON CTDK.MaPTN = PTN.MaPhong

INNER JOIN dbo.BanDangKy BDK ON BDK.MaDangKy = CTDK.MaDangKy

INNER JOIN dbo.BaiThiNghiem BTN ON CTDK.MaBTN = BTN.MaBTN

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

INNER JOIN dbo.NhanVien NV ON CTDK.MaNV = NV.MaNV

END

GO

* Stored procedure [dbo].[spGetChiTietDangKyByMaDangKy]

-- Lấy Chi tiết đăng ký của Bản đăng ký có mã truyền vào

CREATE PROC [dbo].[spGetChiTietDangKyByMaDangKy]

@MaDangKy varchar(10)

AS

BEGIN

SELECT

CTDK.MaCTDK,

BDK.MaDangKy,

GV.HoTen AS HoTenGV,

NV.HoTen AS HoTenNV,

PTN.TenPhong,

BTN.TenBTN,

CTDK.NgaySD,

BDK.Lop,

CTDK.CaTrongNgay,

CTDK.SoNhom,

CTDK.TinhTrang,

CTDK.GhiChu

FROM dbo.ChiTietDangKi CTDK

INNER JOIN dbo.PhongThiNghiem PTN ON CTDK.MaPTN = PTN.MaPhong

INNER JOIN dbo.BanDangKy BDK ON BDK.MaDangKy = CTDK.MaDangKy

INNER JOIN dbo.BaiThiNghiem BTN ON CTDK.MaBTN = BTN.MaBTN

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

INNER JOIN dbo.NhanVien NV ON CTDK.MaNV = NV.MaNV

WHERE CTDK.MaDangKy = @MaDangKy

END

GO

* Stored procedure [dbo].[spInsertChiTietDangKy]

-- Thêm Chi tiết đăng ký

CREATE PROC [dbo].[spInsertChiTietDangKy]

@MaCTDK varchar(10),

@HoTenNV nvarchar(50),

@TenPhong nvarchar(50),

@MaDangKy nvarchar(50),

@TenBTN nvarchar(50),

@NgaySD date,

@CaTrongNgay int,

@SoNhom int,

@TinhTrang nvarchar(50),

@GhiChu text

AS

BEGIN

DECLARE

@MaPhong varchar(10),

@MaBTN varchar(10),

@MaNV varchar(10)

SELECT @MaPhong = PTN.MaPhong

FROM dbo.PhongThiNghiem PTN

WHERE PTN.TenPhong = @TenPhong

SELECT @MaBTN = BTN.MaBTN

FROM dbo.BaiThiNghiem BTN

WHERE BTN.TenBTN = @TenBTN

SELECT @MaNV = NV.MaNV

FROM dbo.NhanVien NV

WHERE NV.HoTen = @HoTenNV

INSERT INTO dbo.ChiTietDangKi (MaCTDK, MaNV, MaPTN, MaDangKy, MaBTN, NgaySD, CaTrongNgay, SoNhom, TinhTrang, GhiChu)

VALUES (@MaCTDK, @MaNV, @MaPhong, @MaDangKy, @MaBTN, @NgaySD, @CaTrongNgay, @SoNhom, @TinhTrang, @GhiChu)

END

GO

* Stored procedure [dbo].[spGetMaCTDK]

-- Tạo mã Chi tiết đăng ksy có dạng 'CTDKxxxx'

-- Trong đố XXXX là số tăng tự động

CREATE PROC [dbo].[spGetMaCTDK]

AS

BEGIN

DECLARE @MaCTDK varchar(10)

DECLARE @MaCTDKMax varchar(10)

DECLARE @Max varchar(10)

-- Nếu đã tồn tại Chi tiết đăng ký thì lấy MaCTDK bằng MaCTDK lớn nhất cộng 1

IF EXISTS( SELECT MaCTDK FROM dbo.ChiTietDangKi)

BEGIN

SELECT @MaCTDKMax = MAX(MaCTDK) FROM dbo.ChiTietDangKi

SET @Max = CONVERT(int, SUBSTRING(@MaCTDKMax, 5, 4)) + 1

IF @Max < 10 SET @MaCTDK = 'CTDK' + '000' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100) SET @MaCTDK = 'CTDK' + '00' + CONVERT(varchar(2), @Max)

ELSE IF (@Max < 1000) SET @MaCTDK = 'CTDK' + '0' + CONVERT(varchar(3), @Max)

ELSE SET @MaCTDK = 'CTDK' + '' + CONVERT(varchar(4), @Max)

END

-- Nếu chưa tồn tại Chi tiết đăng ký nào thì lấy mã bản đăng ký là CTDK0001

ELSE SET @MaCTDK = 'CTDK0001'

SELECT @MaCTDK

END

GO

* Stored procedure [dbo].[spLoadMaDangKy]

CREATE PROC [dbo].[spLoadMaDangKy]

-- Lấy danh sách toàn bộ Mã bản đăng ký

AS

BEGIN

SELECT BDK.MaDangKy

FROM dbo.BanDangKy BDK

END

GO

* Stored procedure [dbo].[spLoadMaDangKyByMaGV]

CREATE PROC [dbo].[spLoadMaDangKyByMaGV]

-- Lấy danh sách Mã bản đăng ký của 1 Giáo viên

@HoTenGV nvarchar(50)

AS

BEGIN

SELECT BDK.MaDangKy

FROM dbo.BanDangKy BDK

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

WHERE GV.HoTen = @HoTenGV

END

GO

* Stored procedure [dbo].[spUpdateChiTietDangKy]

-- Sửa thông tin Chi tiết đăng ký

CREATE PROC [dbo].[spUpdateChiTietDangKy]

@MaCTDK varchar(10),

@HoTenNV varchar(10),

@TenPhong nvarchar(50),

@MaDangKy nvarchar(50),

@TenBTN nvarchar(50),

@NgaySD date,

@CaTrongNgay int,

@SoNhom int,

@TinhTrang nvarchar(50),

@GhiChu text

AS

BEGIN

DECLARE

@MaPhong varchar(10),

@MaBTN varchar(10),

@MaNV varchar(10)

SELECT @MaPhong = PTN.MaPhong

FROM dbo.PhongThiNghiem PTN

WHERE PTN.TenPhong = @TenPhong

SELECT @MaBTN = BTN.MaBTN

FROM dbo.BaiThiNghiem BTN

WHERE BTN.TenBTN = @TenBTN

SELECT @MaNV = NV.MaNV

FROM dbo.NhanVien NV

WHERE NV.HoTen = @HoTenNV

UPDATE dbo.ChiTietDangKi SET

MaNV = @MaNV

,MaPTN = @MaPhong

,MaDangKy = @MaDangKy

,MaBTN = @MaBTN

,NgaySD = @NgaySD

,CaTrongNgay = @CaTrongNgay

,SoNhom = @SoNhom

,TinhTrang = @TinhTrang

,GhiChu = @GhiChu

WHERE MaCTDK = @MaCTDK

END

GO

* Stored procedure [dbo].[spDeleteChiTietDangKy]

-- Xóa Chi tiết đăng ký

CREATE PROC [dbo].[spDeleteChiTietDangKy]

@MaCTDK varchar(10)

AS

BEGIN

DELETE FROM dbo.ChiTietDangKi WHERE MaCTDK = @MaCTDK

END

GO

* Stored procedure [dbo].[spGetChiTietDangKyByKey]

-- Tìm kiến danh sách Chi tiết đăng ký theo Key

CREATE PROC [dbo].[spGetChiTietDangKyByKey]

@MaDangKy varchar(10),

@Key nvarchar(50)

AS

BEGIN

SELECT

CTDK.MaCTDK,

BDK.MaDangKy,

GV.HoTen AS HoTenGV,

NV.HoTen AS HoTenNV,

PTN.TenPhong,

BTN.TenBTN,

CTDK.NgaySD,

BDK.Lop,

CTDK.CaTrongNgay,

CTDK.SoNhom,

CTDK.TinhTrang,

CTDK.GhiChu

FROM dbo.ChiTietDangKi CTDK

INNER JOIN dbo.PhongThiNghiem PTN ON CTDK.MaPTN = PTN.MaPhong

INNER JOIN dbo.BanDangKy BDK ON BDK.MaDangKy = CTDK.MaDangKy

INNER JOIN dbo.BaiThiNghiem BTN ON CTDK.MaBTN = BTN.MaBTN

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

INNER JOIN dbo.NhanVien NV ON CTDK.MaNV = NV.MaNV

WHERE

CTDK.MaDangKy LIKE ('%' + @MaDangKy + '%')

AND

(

GV.HoTen LIKE ('%' + @Key + '%')

OR NV.HoTen LIKE ('%' + @Key + '%')

OR BDK.Lop LIKE ('%' + @Key + '%')

OR PTN.TenPhong LIKE ('%' + @Key + '%')

OR BTN.TenBTN LIKE ('%' + @Key + '%')

OR CTDK.NgaySD LIKE ('%' + @Key + '%')

OR CTDK.TinhTrang LIKE ('%' + @Key + '%')

OR CTDK.GhiChu LIKE ('%' + @Key + '%')

)

END

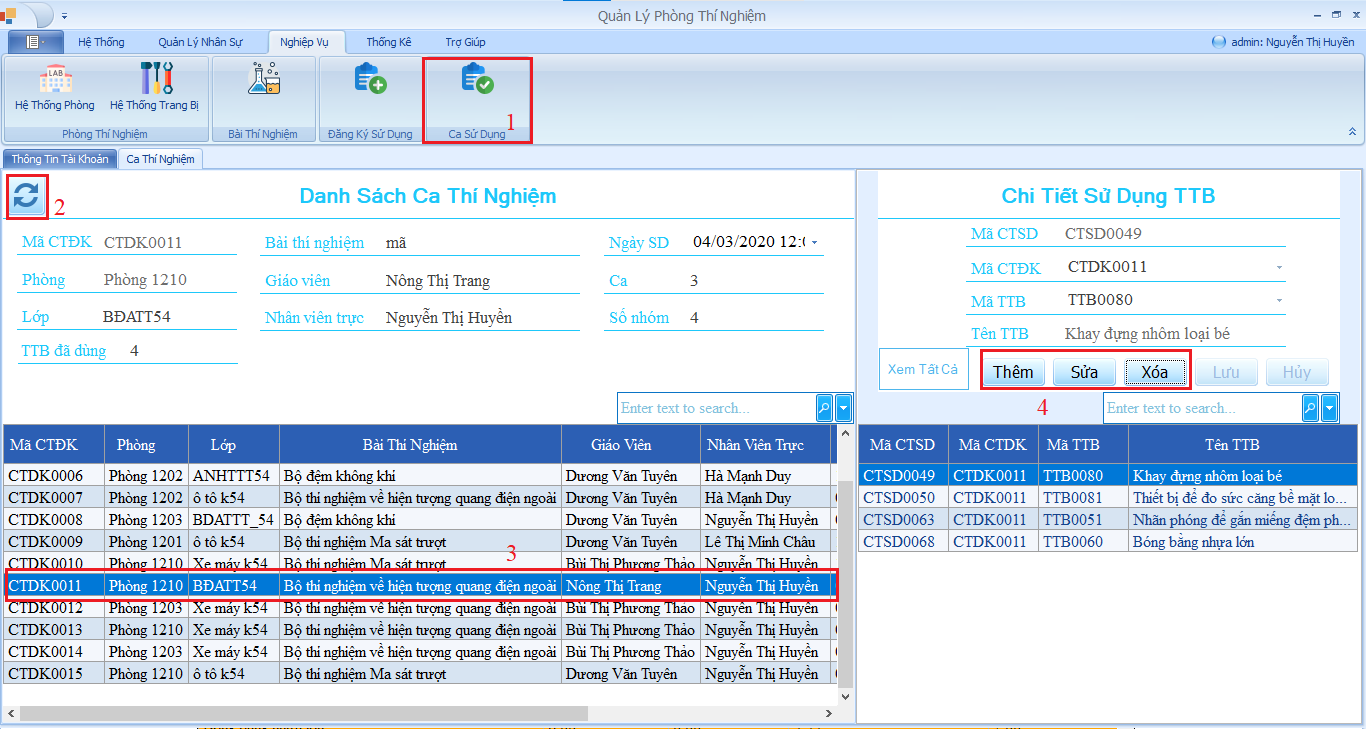
GO

1. Quản lý các ca sử dụng Phòng thí nghiệm
2. Chức năng hệ thống:

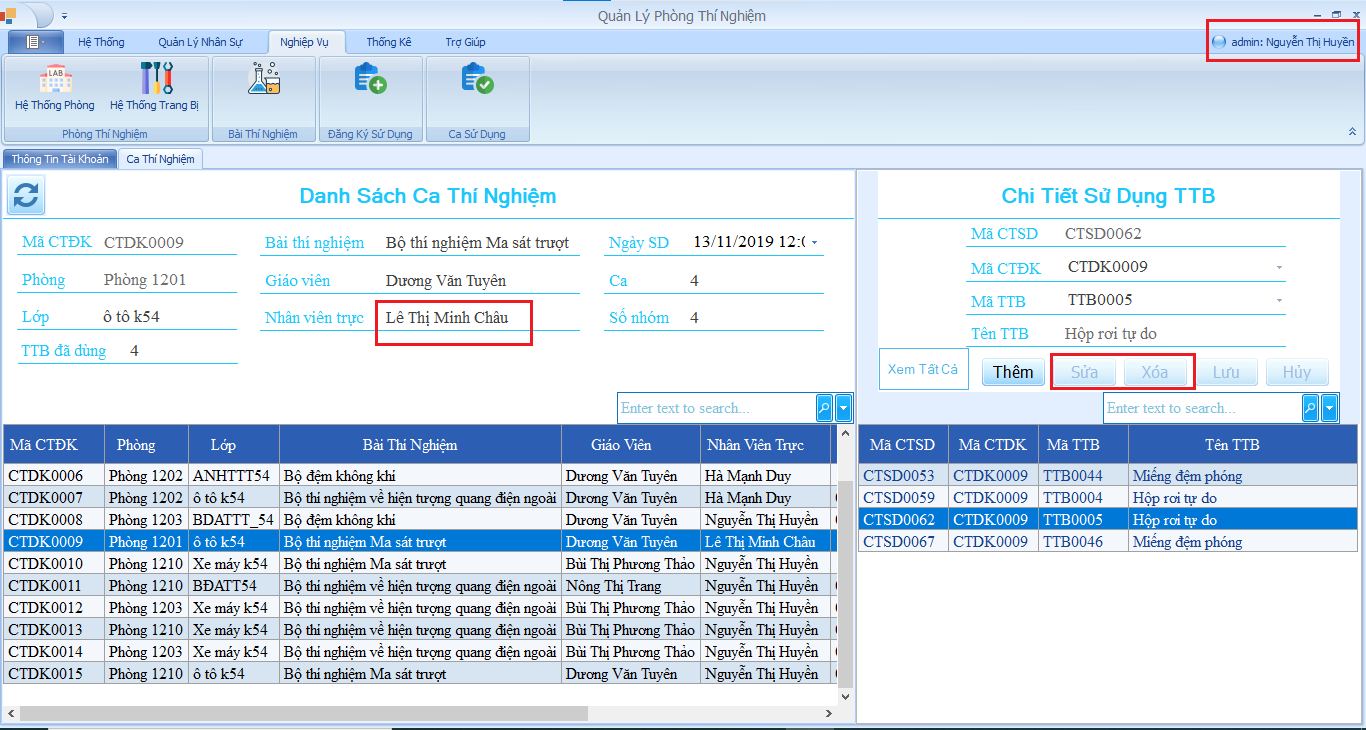
Đối với tất cả các quyền:

* Người dùng có thể xem danh sách các Ca sử dụng (là những Chi tiết đăng ký có tình trạng Đã thực hiện và Chưa thực hiện) và Chi tiết sử dụng TTB (Những TTB được sử dụng trong ca đó)
* Admin, nhân viên được sửa, xóa những Chi tiết sử dụng của Ca sử dụng mà mình trực.
* Giáo viên được sửa, xóa những Chi tiết sử dụng của Ca sử dụng mà mình hướng dẫn thí nghiệm.
* Khi thêm Chi tiết sử dụng của một Ca sử dụng có tình trạng là “Chưa thực hiện” thì sẽ chuyển tình trạng của ca đó thành “Đã thực hiện” (dùng trigger).

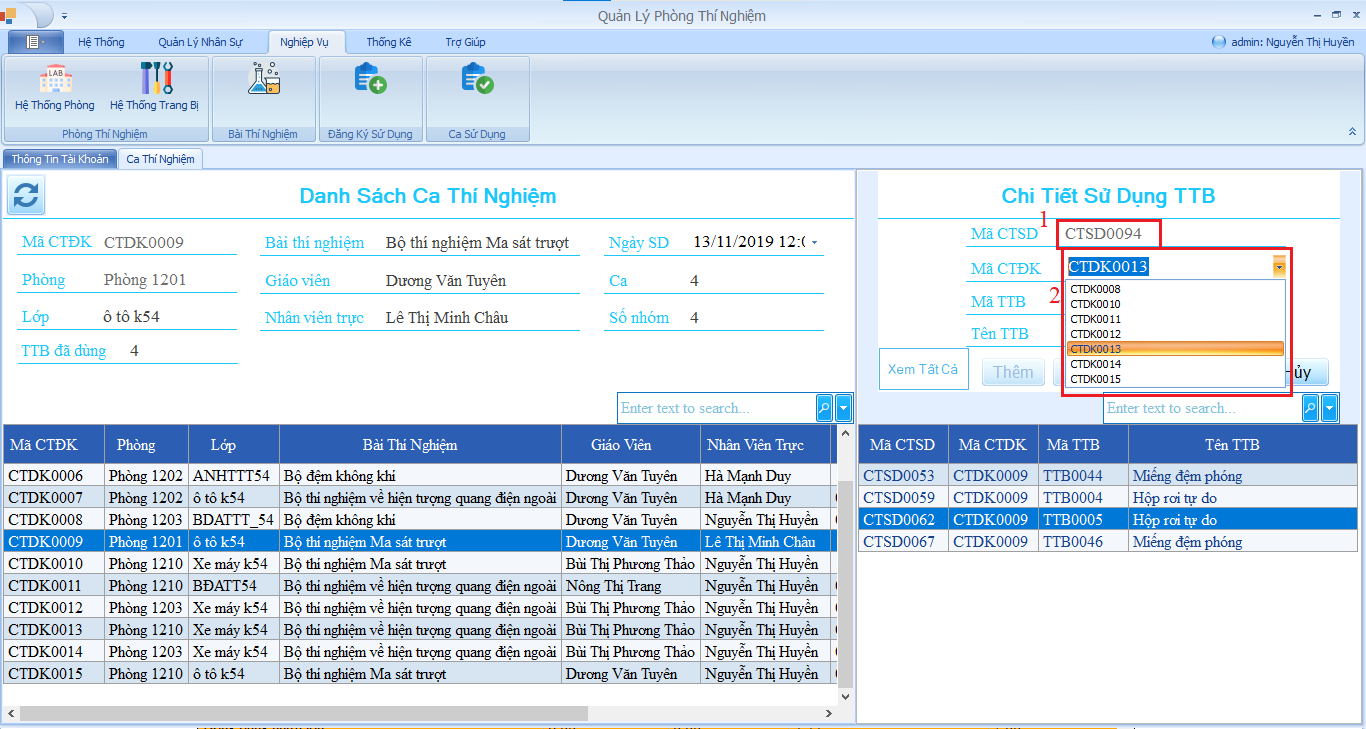
1. Giao diện hệ thống:



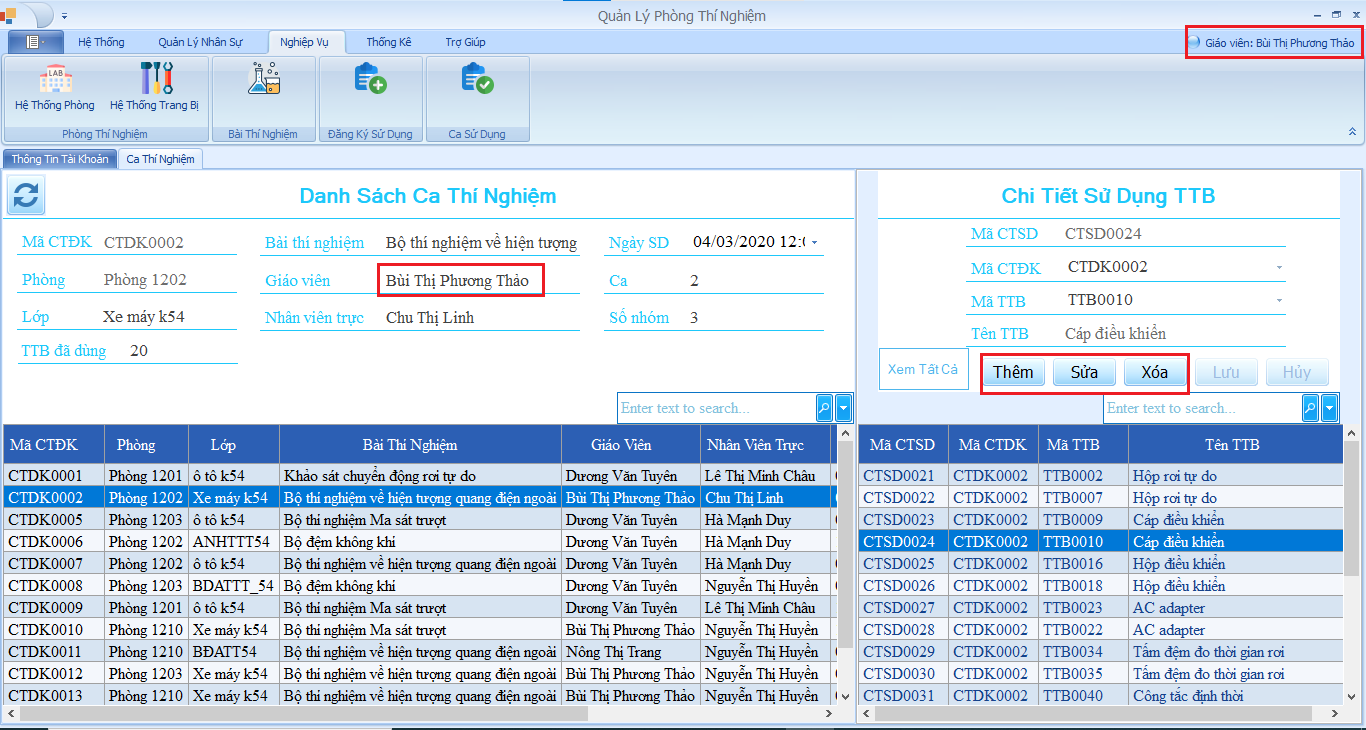
* Giao diện khi nhấn vào Button Ca Sử Dụng (1) thuộc Page Nghiệp vụ.
* Nhấn Button (2) để tải lại thông tin Ca Sử Dụng.
* Nhấn vào Row chưá Ca Sử Dụng (3) để xem thông tin Ca sử dụng đó và xem các Chi tiết sử dụng TTB của ca đó.
* Với admin hoặc nhân viên: có thể sửa, xóa Chi tiết sử dụng của Ca sử dụng mà mình trực.



* Nhân viên không thể sửa, xóa Chi tiết sử dụng của Ca sử dụng mà không phải mình trực.



* Giao diện sau khi nhấn vào Button Thêm.
* Chương trình sẽ tự động cấp 1 Mã chi tiết sử dụng bị dạng “CTSDxxxx” như ở (1). Trong đó xxxx tăng tự động.
* Có thể thêm Chi tiết sử dụng cho những Ca sử dụng mà mình trực hoặc hướng dẫn.



* Giáo viên có thể thêm, sửa, xóa Chi tiết sử dụng của Ca sử dụng mà mình hướng dẫn.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spGetCaSuDung]

-- Lấy danh sách Ca sử dụng

CREATE PROC [dbo].[spGetCaSuDung]

AS

BEGIN

SELECT

CTDK.MaCTDK,

PTN.TenPhong,

BDK.Lop,

BTN.TenBTN,

GV.HoTen AS HoTenGV,

NV.HoTen AS HoTenNV,

CTDK.NgaySD,

CTDK.CaTrongNgay,

CTDK.SoNhom,

(

SELECT COUNT(\*)

FROM dbo.ChiTietSuDung CTSD

WHERE CTSD.MaCTDK = CTDK.MaCTDK

) AS TTBDaDung

FROM dbo.ChiTietDangKi CTDK

INNER JOIN dbo.PhongThiNghiem PTN ON CTDK.MaPTN = PTN.MaPhong

INNER JOIN dbo.BanDangKy BDK ON BDK.MaDangKy = CTDK.MaDangKy

INNER JOIN dbo.BaiThiNghiem BTN ON CTDK.MaBTN = BTN.MaBTN

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

INNER JOIN dbo.NhanVien NV ON CTDK.MaNV = NV.MaNV

WHERE CTDK.TinhTrang = N'Đã thực hiện' OR CTDK.TinhTrang = N'Chưa thực hiện'

END

GO

* Stored procedure [dbo].[spGetCaSuDungByKey]

-- Lấy danh sách Ca sử dụng theo Key

CREATE PROC [dbo].[spGetCaSuDungByKey]

@Key nvarchar(50)

AS

BEGIN

SELECT

CTDK.MaCTDK,

PTN.TenPhong,

BDK.Lop,

BTN.TenBTN,

GV.HoTen AS HoTenGV,

NV.HoTen AS HoTenNV,

CTDK.NgaySD,

CTDK.CaTrongNgay,

CTDK.SoNhom,

(SELECT COUNT(\*) FROM dbo.ChiTietSuDung CTSD WHERE CTSD.MaCTDK = CTDK.MaCTDK) AS TTBDaDung

FROM dbo.ChiTietDangKi CTDK

INNER JOIN dbo.PhongThiNghiem PTN ON CTDK.MaPTN = PTN.MaPhong

INNER JOIN dbo.BanDangKy BDK ON BDK.MaDangKy = CTDK.MaDangKy

INNER JOIN dbo.BaiThiNghiem BTN ON CTDK.MaBTN = BTN.MaBTN

INNER JOIN dbo.GiaoVien GV ON BDK.MaGV = GV.MaGV

INNER JOIN dbo.NhanVien NV ON CTDK.MaNV = NV.MaNV

WHERE CTDK.TinhTrang = N'Đã thực hiện' OR CTDK.TinhTrang = N'Chưa thực hiện'

HAVING PTN.TenPhong LIKE ('%' + @Key + '%')

OR BDK.Lop LIKE ('%' + @Key + '%')

OR BTN.TenBTN LIKE ('%' + @Key + '%')

OR GV.HoTen LIKE ('%' + @Key + '%')

OR NV.HoTen LIKE ('%' + @Key + '%')

OR CTDK.NgaySD LIKE ('%' + @Key + '%')

END

GO

* Stored procedure [dbo].[spGetChiTietSuDung]

-- Lấy Chi tiết sử dụng

CREATE PROC [dbo].[spGetChiTietSuDung]

AS

BEGIN

SELECT

CTSD.MaChiTietSD,

CTSD.MaCTDK,

CTSD.MaTTB,

TTB.TenTTB

FROM dbo.ChiTietSuDung CTSD

INNER JOIN dbo.TrangThietBi TTB ON CTSD.MaTTB = TTB.MaTTB

END

GO

* Stored procedure [dbo].[spGetTrangThietBiCanDungByMaBTN]

-- Lấy Chi tiết sử dụng của Ca sử dụng có Mã truyền vào

CREATE PROC [dbo].[spGetChiTietSuDungByMaCTDK]

@MaCTDK varchar(10)

AS

BEGIN

SELECT

CTSD.MaChiTietSD,

CTSD.MaCTDK,

CTSD.MaTTB,

TTB.TenTTB

FROM dbo.ChiTietSuDung CTSD

INNER JOIN dbo.TrangThietBi TTB ON CTSD.MaTTB = TTB.MaTTB

WHERE CTSD.MaCTDK = @MaCTDK

END

GO

* Stored procedure [dbo].[spGetMaChiTietSD]

-- Tạo mã Chi tiết sử dụng có dạng 'CTSDxxxx'

-- Trong đố XXXX là số tăng tự động

CREATE PROC [dbo].[spGetMaChiTietSD]

AS

BEGIN

DECLARE @MaChiTietSD varchar(10)

DECLARE @MaChiTietSDMax varchar(10)

DECLARE @Max varchar(10)

IF EXISTS( SELECT MaChiTietSD FROM dbo.ChiTietSuDung)

BEGIN

SELECT @MaChiTietSDMax = MAX(MaChiTietSD) FROM dbo.ChiTietSuDung

SET @Max = CONVERT(int, SUBSTRING(@MaChiTietSDMax, 5, 4)) + 1

IF @Max < 10

SET @MaChiTietSD = 'CTSD' + '000' + CONVERT(varchar(1), @Max)

ELSE IF (@Max < 100)

SET @MaChiTietSD = 'CTSD' + '00' + CONVERT(varchar(2), @Max)

ELSE IF (@Max < 1000)

SET @MaChiTietSD = 'CTSD' + '0' + CONVERT(varchar(3), @Max)

ELSE

SET @MaChiTietSD = 'CTSD' + '' + CONVERT(varchar(4), @Max)

END

ELSE SET @MaChiTietSD = 'CTSD0001'

SELECT @MaChiTietSD

END

GO

* Stored procedure [dbo].[spInsertChiTietSuDung]

-- Thêm Chi tiết sử dụng

CREATE PROC [dbo].[spInsertChiTietSuDung]

@MaChiTietSD varchar(10),

@MaCTDK varchar(10),

@MaTTB varchar(10)

AS

BEGIN

INSERT INTO dbo.ChiTietSuDung(MaChiTietSD, MaCTDK, MaTTB)

VALUES (@MaChiTietSD, @MaCTDK, @MaTTB)

END

GO

* Trigger [dbo].[tgInsertChiTietSuDung]

CREATE TRIGGER [dbo].[tgInsertChiTietSuDung]

ON [dbo].[ChiTietSuDung]

INSTEAD OF INSERT

-- Trigger thêm Chi tiết sử dụng:

-- Nếu Ca sử dụng đang ở tình trạng Chưa thực hiện thì sửa thành tình trạng Đã thực hiện

-- Sau đó thêm Chi tiết sử dụng

AS

BEGIN

DECLARE

@MaChiTietSD varchar(20),

@MaCTDK varchar(20),

@MaTTB varchar(20)

SELECT @MaChiTietSD = MaChiTietSD FROM inserted

SELECT @MaCTDK = MaCTDK FROM inserted

SELECT @MaTTB = MaTTB FROM inserted

UPDATE dbo.ChiTietDangKi

SET TinhTrang = N'Đã thực hiện'

WHERE MaCTDK = @MaCTDK

INSERT INTO dbo.ChiTietSuDung(MaChiTietSD, MaCTDK, MaTTB)

VALUES (@MaChiTietSD, @MaCTDK, @MaTTB)

END

GO

* Stored procedure [dbo].[spUpdateChiTietSuDung]

-- Sửa thông tin Chi tiết sử dụng

CREATE PROC [dbo].[spUpdateChiTietSuDung]

@MaChiTietSD varchar(10),

@MaCTDK varchar(10),

@MaTTB varchar(10)

AS

BEGIN

UPDATE dbo.ChiTietSuDung SET

MaTTB = @MaTTB

WHERE MaChiTietSD = @MaChiTietSD

END

GO

* Stored procedure [dbo].[spDeleteChiTietSuDung]

-- Xóa Chi tiết sử dụng

CREATE PROC [dbo].[spDeleteChiTietSuDung]

@MaChiTietSD varchar(10)

AS

BEGIN

DELETE FROM dbo.ChiTietSuDung WHERE MaChiTietSD = @MaChiTietSD

END

GO

* Stored procedure [dbo].[spGetChiTietSuDungByKey]

-- Tìm kiến danh sách Chi tiết sử dụng theo Key

CREATE PROC [dbo].[spGetChiTietSuDungByKey]

@MaCTDK varchar(10),

@Key nvarchar(50)

AS

BEGIN

SELECT

CTSD.MaChiTietSD,

CTSD.MaCTDK,

CTSD.MaTTB,

TTB.TenTTB

FROM dbo.ChiTietSuDung CTSD

INNER JOIN dbo.TrangThietBi TTB ON CTSD.MaTTB = TTB.MaTTB

WHERE

CTSD.MaCTDK LIKE ('%' + @MaCTDK + '%')

AND

(

CTSD.MaChiTietSD LIKE ('%' + @Key + '%')

OR CTSD.MaCTDK LIKE ('%' + @Key + '%')

OR CTSD.MaTTB LIKE ('%' + @Key + '%')

OR TTB.TenTTB LIKE ('%' + @Key + '%')

)

END

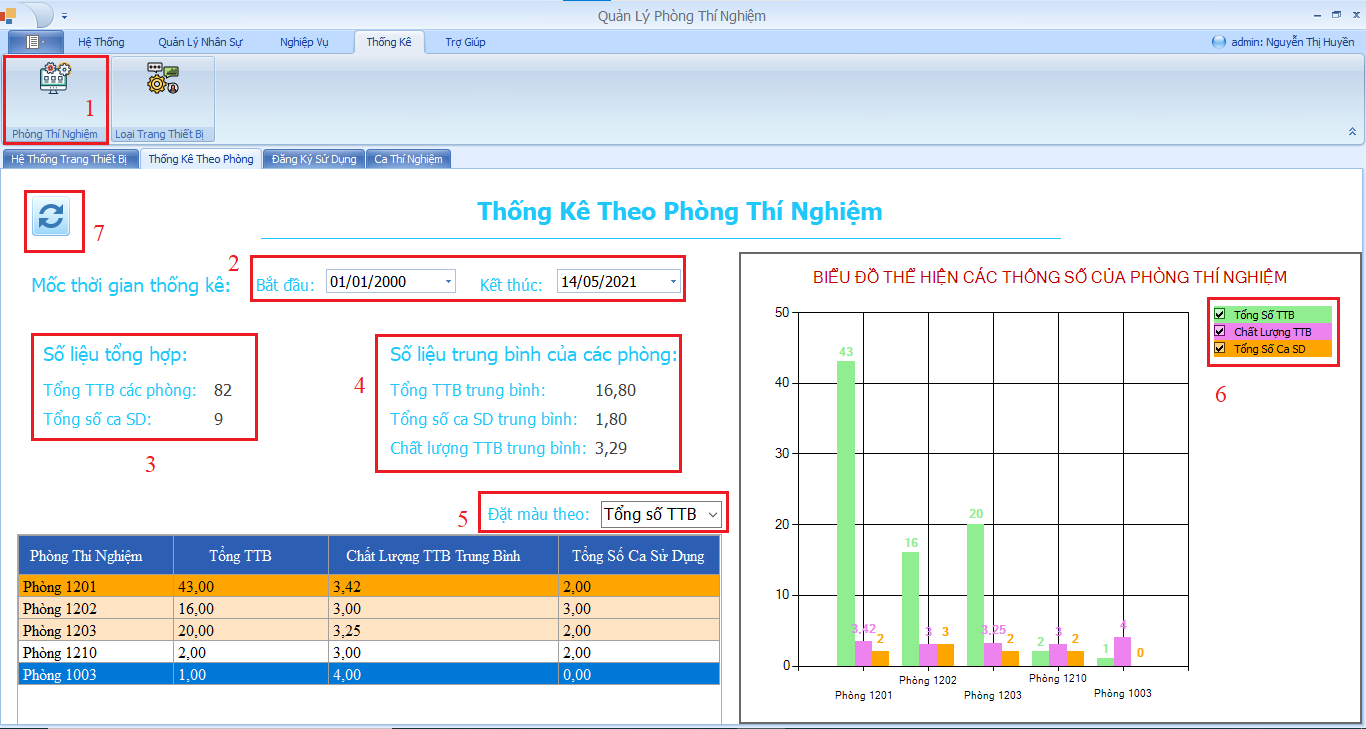
GO

1. Thống kê theo Phòng thí nghiệm
2. Chức năng hệ thống:

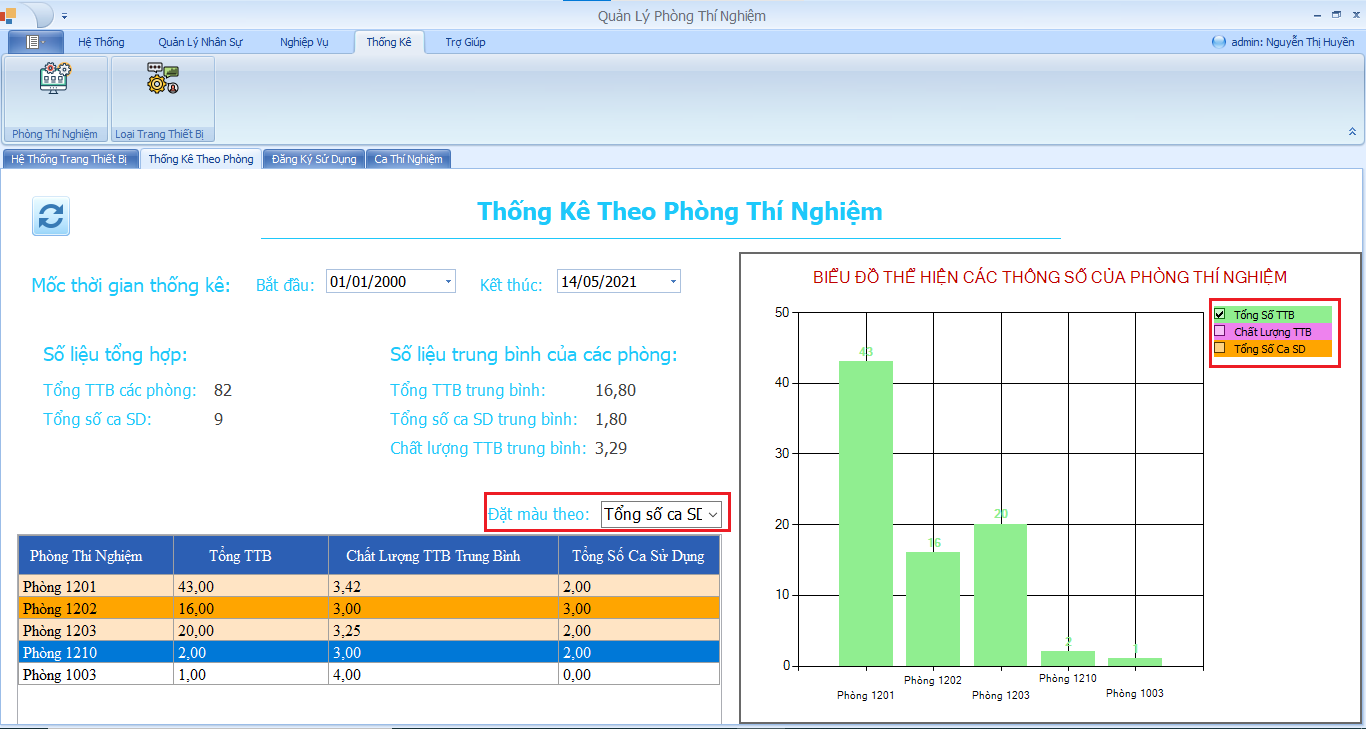
Cho phép tất cả các quyền:

* Xem số liệu trung bình của các Phòng thí nghiệm: Số TTB trung bình, Số TTB hỏng trung bình, Chất lượng TTB trung bình, Số ca sử dụng PTN trung bình.
* Thống kê số liệu của mỗi Phòng thí nghiệm: Tổng số TTB, Tổng số TTB hỏng, Chất lượng TTB trung bình, Số ca sử dụng Phòng.

1. Giao diện hệ thống:



* Giao diện khi nhấn vào Button Phòng Thí Nghiệm (1) thuộc Page Thống Kê.
* Chọn mốc thời gian ở (2) để thống kê theo mốc thời gian đó.
* Xem số liệu tổng hợp của các Phòng ở (3).
* Xem số liệu trung bình của các Phòng thí nghiệm ở (4).
* Thiết lập màu các hàng theo tiêu chí ở combobox (5):
  + Loại TTB có số liệu thống kê được lớn hơn 4/3 so với mức trung bình: màu cam.
  + Loại TTB có số liệu thống kê được bé hơn 2/3 so với mức trung bình: màu trắng.
  + Loại TTB có số liệu thống kê được xấp xỉ mức trung bình: màu da.
* Xem biểu đồ thể hiện các số liệu đã thống kê được. Có thể chọn tiêu chí để thống kê ở (6)
* Nhấn button (7) để tải lại thông tin thống kê.



* Khi đặt màu các hàng theo chỉ số Tổng số ca SD.
* Biểu đồ khi chỉ chọn thống kê theo Tổng số TTB của mỗi Phòng.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spCountTongTTB]

CREATE PROC [dbo].[spCountTongTTB]

-- Tính tổng TTB hiện có ở các phòng

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.TrangThietBi

WHERE XuatKho = 0

END

GO

* Stored procedure [dbo].[spCountTongCSD]

CREATE PROC [dbo].[spCountTongCSD]

-- Tính tổng ca sử dụng được thực hiện ở các phòng trong khoảng thời gian

@NgayBatDau date,

@NgayKetThuc date

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.ChiTietDangKi

WHERE TinhTrang = N'Đã thực hiện' AND NgaySD BETWEEN @NgayBatDau AND @NgayKetThuc

END

GO

* Funtion [dbo].[fcThongKe\_ChatLuongTTBFromPhong]

-- Hàm tính chất lượng TTB trung bình của các TTB thuộc 1 phòng

-- Nếu @MaPhong truyền vào là '' thì tính chất lượng TTB trung bình của các Phòng

-- Tính chất lượng trung bình trên thang điểm 4

CREATE FUNCTION [dbo].[fcThongKe\_ChatLuongTTBFromPhong]

(@MaPhong AS varchar(10))

RETURNS decimal(10,2)

AS

BEGIN

DECLARE

@TongTTB int,

@TongChatLuong float = 0,

@ChatLuongTB float = 0,

@TinhTrang nvarchar(50)

-- Sử dụng CURSOR để duyệt tới TinhTrang của các Trang thiết bị

-- Quy đổi thành điểm Chất lượng (0->4) theo TinhTrang tương ứng của TTB đó

-- Tính tổng điểm Chất lượng của các TrangThietBi thuộc Phòng thí nghiệm:

DECLARE curTTB CURSOR FOR

SELECT TinhTrang FROM dbo.TrangThietBi

WHERE MaPhong LIKE ('%' + @MaPhong + '%') AND XuatKho = 0

OPEN curTTB

FETCH NEXT FROM curTTB into @TinhTrang

WHILE (@@FETCH\_STATUS = 0)

BEGIN

IF (@TinhTrang = N'Rất tốt') SET @TongChatLuong += 4

ELSE IF (@TinhTrang = N'Tốt') SET @TongChatLuong += 3

ELSE IF (@TinhTrang = N'Tạm ổn') SET @TongChatLuong += 2

ELSE IF (@TinhTrang = N'Kém') SET @TongChatLuong += 1

ELSE IF (@TinhTrang = N'Hỏng') SET @TongChatLuong += 0

FETCH NEXT FROM curTTB into @TinhTrang

END

CLOSE curTTB

DEALLOCATE curTTB

-- Đếm số TTB thuộc loại TTB

SELECT @TongTTB = COUNT(\*)

FROM dbo.TrangThietBi TTB

WHERE MaPhong LIKE ('%' + @MaPhong + '%')

AND XuatKho = 0

-- Tính chất lượng trung bình

IF(@TongTTB > 0) SET @ChatLuongTB = @TongChatLuong / @TongTTB

RETURN @ChatLuongTB

END

GO

* Stored procedure [dbo].[spThongKe\_PhongThiNghiem]

-- Thống kê theo từng Phòng Thí Nghiệm theo Khoảng thời gian NgayBatDau đến NgayKetThuc

CREATE PROC [dbo].[spThongKe\_PhongThiNghiem]

@NgayBatDau date,

@NgayKetThuc date

AS

BEGIN

SELECT

PTN.TenPhong,

-- Tổng TTB thuộc Phòng Thí Nghiệm

(

SELECT CONVERT(decimal(10,2), COUNT(TTB.MaTTB))

FROM dbo.TrangThietBi TTB

WHERE TTB.MaPhong = PTN.MaPhong AND XuatKho = 0

) AS TongTTB,

-- Chất lượng trung bình của các TTB thuộc Loại TTB

(SELECT dbo.fcThongKe\_ChatLuongTTBFromPhong(PTN.MaPhong)) AS ChatLuongTTB,

-- Tổng số Ca sử dụng Phòng thí nghiệm trong Khoảng thời gian thống kê

(

SELECT CONVERT(decimal(10,2), COUNT(CTDK.MaCTDK))

FROM dbo.ChiTietDangKi CTDK

WHERE

CTDK.TinhTrang = N'Đã thực hiện'

AND CTDK.MaPTN = PTN.MaPhong

AND CTDK.NgaySD BETWEEN @NgayBatDau AND @NgayKetThuc

) AS TongCaSuDung

FROM dbo.PhongThiNghiem PTN

END

GO

* Stored procedure [dbo].[spThongKe\_PhongThiNghiemTB]

-- Tính các số liệu trung bình của các Phòng thí nghiệm

-- Trong khoảng thời gian từ Ngày bắt đầu đến ngày kết thúc. Gồm:

-- TenPhong: 'Trung bình'

-- TongTTB: Số lượng TTB trung bình mỗi Phòng hiện có

-- ChatLuongTTB: Chất lượng TTB trung bình của mỗi Phòng

-- TongCaSuDung: Số ca sử dụng trung bình của mỗi Phòng trong thời gian thống kê

CREATE PROC [dbo].[spThongKe\_PhongThiNghiemTB]

@NgayBatDau date,

@NgayKetThuc date

AS

BEGIN

DECLARE

@TongSoPhong decimal,

@TongTTBCacPhong decimal,

@TongCaSuDung decimal

-- Đếm số Phòng thí nghiệm

SELECT @TongSoPhong = COUNT(PTN.MaPhong) FROM dbo.PhongThiNghiem PTN

-- Đếm tổng số TTB các Phòng

SELECT @TongTTBCacPhong = COUNT(TTB.MaTTB) FROM dbo.TrangThietBi TTB

-- Đếm tổng số lượt sử dụng TTB trong khoảng thời gian

SELECT @TongCaSuDung = COUNT(CTDK.MaCTDK)

FROM dbo.ChiTietDangKi CTDK

WHERE

CTDK.TinhTrang = N'Đã thực hiện'

AND CTDK.NgaySD BETWEEN @NgayBatDau AND @NgayKetThuc

SELECT

N'Trung bình' AS TenPhong,

CONVERT(decimal(10,2), @TongTTBCacPhong/@TongSoPhong) AS TongTTB,

(SELECT dbo.fcThongKe\_ChatLuongTTBFromPhong('')) AS ChatLuongTTB,

CONVERT(decimal(10,2), @TongCaSuDung/@TongSoPhong) AS TongCaSuDung

END

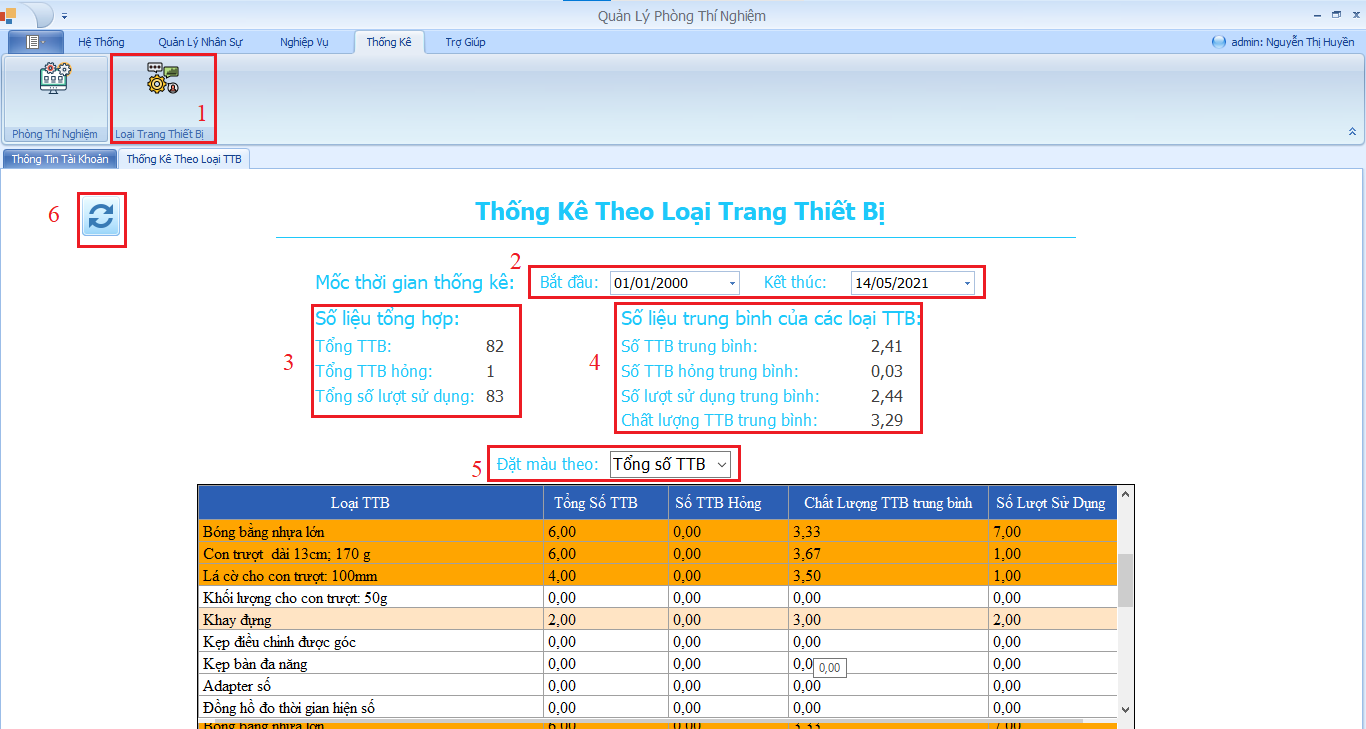
GO

1. Thống kê theo Loại trang bị
2. Chức năng hệ thống:

Cho phép tất cả các quyền:

* Xem số liệu trung bình của các Loại TTB: Tổng số TTB trung bình, Tổng số TTB hỏng trung bình, Chất lượng TTB trung bình, Số lượt sử dụng TTB trung bình.
* Xem số liệu của mỗi Loại TTB: Tổng số TTB, Tổng số TTB hỏng, Chất lượng TTB trung bình, Số lượt sử dụng TTB.

1. Giao diện hệ thống:



* Giao diện khi nhấn vào Button Loại Trang Thiết Bị (1) thuộc Page Thống kê.
* Chọn mốc thời gian ở (2) để thống kê theo mốc thời gian đó.
* Xem số liệu tổng hợp của các Loại TTB ở (3): Phần này tiếp tục hoàn thiện sau.
* Xem số liệu trung bình của các Loại TTB ở (4).
* Thiết lập màu các hàng theo dữ liệu ở combobox (5):
  + Loại TTB có số liệu thống kê được lớn hơn 4/3 so với mức trung bình: màu cam.
  + Loại TTB có số liệu thống kê được bé hơn 2/3 so với mức trung bình: màu trắng.
  + Loại TTB có số liệu thống kê được xấp xỉ mức trung bình: màu da.
* Nhấn vào (6) để tải lại dữ liệu thống kê.

1. Các lệnh SQL phục vụ:

* Stored procedure [dbo].[spCountTongTTB]
* Stored procedure [dbo].[spCountTongTTBHong]

CREATE PROC [dbo].[spCountTongTTBHong]

-- Tính tổng TTB đã hỏng hiện có ở các phòng

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.TrangThietBi

WHERE XuatKho = 0 AND TinhTrang = N'Hỏng'

END

GO

* Stored procedure [dbo].[spCountTongLuotSD]

CREATE PROC [dbo].[spCountTongLuotSD]

-- Tính tổng số lượt sử dụng TTB trong thời gian

@NgayBatDau date,

@NgayKetThuc date

AS

BEGIN

SELECT COUNT(\*)

FROM dbo.ChiTietSuDung CTSD

INNER JOIN dbo.ChiTietDangKi CTDK ON CTSD.MaCTDK = CTDK.MaCTDK

INNER JOIN dbo.TrangThietBi TTB ON CTSD.MaTTB = TTB.MaTTB

WHERE CTDK.NgaySD BETWEEN @NgayBatDau AND @NgayKetThuc

AND TTB.XuatKho = 0

END

GO

* Funtion [dbo].[fcTinh\_ChatLuongTTBFromLoaiTTB]

-- Hàm tính chất lượng TTB trung bình của Loại TTB có MaLTTB

-- Nếu @MaLTTB truyền vào là '' thì tính chất lượng TTB trung bình của các Loại

-- Tính chất lượng trung bình trên thang điểm 4

CREATE FUNCTION [dbo].[fcTinh\_ChatLuongTTBFromLoaiTTB]

(@MaLTTB AS varchar(10))

RETURNS decimal(10,2)

AS

BEGIN

DECLARE

@TongTTB int,

@TongChatLuong float = 0,

@ChatLuongTB float = 0,

@TinhTrang nvarchar(50)

-- Sử dụng CURSOR để duyệt tới TinhTrang của các Trang thiết bị

-- Quy đổi thành điểm Chất lượng (0->4) theo TinhTrang tương ứng của TTB đó

-- Tính tổng điểm Chất lượng của các TrangThietBi thuộc Loại TTB:

DECLARE curTTB CURSOR FOR

SELECT TinhTrang

FROM dbo.TrangThietBi

WHERE MaLTTB LIKE ('%' + @MaLTTB + '%')

AND XuatKho = 0

OPEN curTTB

FETCH NEXT FROM curTTB into @TinhTrang

WHILE (@@FETCH\_STATUS = 0)

BEGIN

IF (@TinhTrang = N'Rất tốt') SET @TongChatLuong += 4

ELSE IF (@TinhTrang = N'Tốt') SET @TongChatLuong += 3

ELSE IF (@TinhTrang = N'Tạm ổn') SET @TongChatLuong += 2

ELSE IF (@TinhTrang = N'Kém') SET @TongChatLuong += 1

ELSE IF (@TinhTrang = N'Hỏng') SET @TongChatLuong += 0

FETCH NEXT FROM curTTB into @TinhTrang

END

CLOSE curTTB

DEALLOCATE curTTB

-- Đếm số TTB thuộc loại TTB

SELECT @TongTTB = COUNT(\*)

FROM dbo.TrangThietBi TTB

WHERE TTB.MaLTTB LIKE ('%' + @MaLTTB + '%') AND XuatKho = 0

-- Tính chất lượng trung bình

IF(@TongTTB > 0) SET @ChatLuongTB = @TongChatLuong / @TongTTB

RETURN @ChatLuongTB

END

GO

* Stored procedure [dbo].[ spThongKe\_LoaiTrangThietBi]

-- Thống kê theo mỗi Loại Trang Thiết Bị

CREATE PROC [dbo].[spThongKe\_LoaiTrangThietBi]

@NgayBatDau date,

@NgayKetThuc date

AS

BEGIN

SELECT

LTTB.TenLoai,

-- Tổng TTB thuộc Loại TTB hiện có trong kho

(

SELECT CONVERT(decimal(10,2), COUNT(TTB.MaTTB))

FROM dbo.TrangThietBi TTB

WHERE TTB.MaLTTB = LTTB.MaLTTB AND TTB.XuatKho = 0

) AS TongTTB,

-- Tổng TTB đã hỏng thuộc Loại TTB hiện có trong kho

(

SELECT CONVERT(decimal(10,2), COUNT(TTB.MaTTB))

FROM dbo.TrangThietBi TTB

WHERE TTB.MaLTTB = LTTB.MaLTTB

AND TTB.TinhTrang = N'Hỏng'

AND TTB.XuatKho = 0

) AS SoTTBHong,

--Chất lượng trung bình của các TTB thuộc Loại TTB

(SELECT [dbo].[fcTinh\_ChatLuongTTBFromLoaiTTB](LTTB.MaLTTB)) AS ChatLuongTTB,

-- Tổng số lwuojt sử dụng các TTB trong kho thuộc Loại TTB

(

SELECT CONVERT(decimal(10,2), COUNT(CTSD.MaChiTietSD))

FROM dbo.ChiTietSuDung CTSD

INNER JOIN dbo.TrangThietBi TTB ON CTSD.MaTTB = TTB.MaTTB

INNER JOIN dbo.ChiTietDangKi CTDK ON CTSD.MaCTDK = CTDK.MaCTDK

WHERE

TTB.MaLTTB = LTTB.MaLTTB

AND TTB.XuatKho = 0

AND CTDK.NgaySD BETWEEN @NgayBatDau AND @NgayKetThuc

) AS TongLuotSuDung

FROM dbo.LoaiTTB LTTB

END

GO

* Stored procedure [dbo].[spThongKe\_LoaiTrangThietBiTB]

-- Tính các số liệu trung bình của các Loại Trang Thiết bi

-- Trong khoảng thời gian từ Ngày bắt đầu đến ngày kết thúc. Gồm:

-- TenLoai: 'Trung bình'

-- TongTTB: Số lượng TTB trong kho trung bình

-- SoTTBHong: Số lượng TTB đã hỏng trong kho trung bình

-- ChatLuongTTB: Chất lượng TTB trung bình

-- TongLuotSD: Số lượt TTB được sử dụng trung bình của mỗi loại trong khoảng thời gian

CREATE PROC [dbo].[spThongKe\_LoaiTrangThietBiTB]

@NgayBatDau date,

@NgayKetThuc date

AS

BEGIN

DECLARE

@TongSoLoaiTTB decimal,

@TongTTBCacLoai decimal,

@TongTTBHong decimal,

@TongLuotSuDung decimal

-- Đếm số LoạiTTB

SELECT @TongSoLoaiTTB = COUNT(LTTB.MaLTTB)

FROM dbo.LoaiTTB LTTB

-- Đếm số TTB

SELECT @TongTTBCacLoai = COUNT(TTB.MaTTB)

FROM dbo.TrangThietBi TTB

WHERE TTB.XuatKho = 0

-- Đếm số TTB đã hỏng ở trong kho

SELECT @TongTTBHong = COUNT(TTB.MaTTB)

FROM dbo.TrangThietBi TTB

WHERE TTB.TinhTrang = N'Hỏng'

AND TTB.XuatKho = 0

-- Đếm tổng số lượt sử dụng TTB trong khoảng thời gian

SELECT @TongLuotSuDung = COUNT(CTSD.MaChiTietSD)

FROM dbo.ChiTietSuDung CTSD

INNER JOIN dbo.ChiTietDangKi CTDK ON CTSD.MaCTDK = CTDK.MaCTDK

INNER JOIN dbo.TrangThietBi TTB ON CTSD.MaTTB = TTB.MaTTB

WHERE CTDK.NgaySD BETWEEN @NgayBatDau AND @NgayKetThuc

AND TTB.XuatKho = 0

SELECT

N'Trung bình' AS TenLoai,

CONVERT(decimal(10,2), @TongTTBCacLoai/@TongSoLoaiTTB) AS TongTTB,

CONVERT(decimal(10,2), @TongTTBHong/@TongSoLoaiTTB) AS SoTTBHong,

(SELECT [dbo].[fcTinh\_ChatLuongTTBFromLoaiTTB]('')) AS ChatLuongTTB,

CONVERT(decimal(10,2), @TongLuotSuDung/@TongSoLoaiTTB) AS TongLuotSuDung

END

GO

Tổng kết và đánh giá.

* + 1. Ưu điểm
* Chương trình có giao diện thân thiện, thuận tiện cho người dùng.
* Có các chức năng linh hoạt phù hợp với yêu cầu thực tế.
* Chương trình xây dựng theo mô hình 3 lớp MVC dễ mở rộng phát triển.
* Có sử sụng kết hợp Stored Procedure, Funtion, Cursor và các Trigger đảm bảo toàn vẹn dữ liệu.
  + 1. Khuyết điểm
* Một số chức năng còn chưa hoạt động ổn định.
* Chương trình khá năng do sử dụng DevExpress, lúc truy vấn vào CSDL thông qua các Stored Procedure tốc độ còn chậm, có thể gây khó chịu cho người dùng.
  + 1. Hướng phát triển
* Tiếp tục tìm hiểu về hệ quản trị CSDL bảo mật tốt.