**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Báo cáo bộ môn

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**QUẢN LÝ KÝ TÚC XÁ**



**Giáo viên hướng dẫn:**

GVCC, TS. NGUYỄN THÀNH SƠN

**Sinh viên thực hiện:**

**NHÓM 1**

|  |  |
| --- | --- |
| Hồ Huy Hoàng | **MSSV: 18110284** |
| Lâm Quốc Khánh | **MSSV: 18110304** |
| Nguyễn Trần Phúc | **MSSV: 18110339** |
| Nguyễn Huỳnh Minh Tiến | **MSSV: 18110377** |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**Ký tên**

MỤC LỤC

[MỤC LỤC 3](#_Toc59607517)

[1. Tổng quan về phần mềm 7](#_Toc59607518)

[**1.1. Mục tiêu** 7](#_Toc59607519)

[**1.2. Phạm vi** 7](#_Toc59607520)

[**1.3. Mô tả** 7](#_Toc59607521)

[**1.4. Thông tin phần mềm** 8](#_Toc59607522)

[**1.5. Chức năng của phần mềm** 8](#_Toc59607523)

[2. Lược đồ quan hệ 8](#_Toc59607524)

[**2.1. Mô hình ERD** 8](#_Toc59607525)

[***2.1.1.* Mô hình ERD** 8](#_Toc59607526)

[***2.1.2.* Lược đồ quan hệ** 9](#_Toc59607527)

[**2.2. Mô tả cơ sở dữ liệu** 10](#_Toc59607528)

[**2.2.1. Bảng PROVINCE** 10](#_Toc59607529)

[**2.2.2. Bảng DISTRICT** 10](#_Toc59607530)

[**2.2.3. Bảng COMMUNE** 10](#_Toc59607531)

[**2.2.4. Bảng ADDRESS** 10](#_Toc59607532)

[**2.2.5. Bảng SECTOR** 11](#_Toc59607533)

[**2.2.6. Bảng ROOM\_TYPE** 11](#_Toc59607534)

[**2.2.7. Bảng ROOM** 11](#_Toc59607535)

[**2.2.8. Bảng ROOM\_REGISTRATION** 11](#_Toc59607536)

[**2.2.9. Bảng INSURANCE** 12](#_Toc59607537)

[**2.2.10. Bảng UNIT** 12](#_Toc59607538)

[**2.2.11. Bảng SERVICE** 12](#_Toc59607539)

[**2.2.12. Bảng COLLEGE** 12](#_Toc59607540)

[**2.2.13. Bảng USER** 12](#_Toc59607541)

[**2.2.14. Bảng RELATIVE** 13](#_Toc59607542)

[**2.2.15. Bảng ADMIN** 13](#_Toc59607543)

[**2.2.16. Bảng EMPLOYEE** 13](#_Toc59607544)

[**2.2.17. Bảng STUDENT** 13](#_Toc59607545)

[**2.2.18. Bảng BILL** 14](#_Toc59607546)

[**2.2.19. Bảng BILL\_DETAIL** 14](#_Toc59607547)

[**2.2.20. Bảng PAYMENT** 14](#_Toc59607548)

[**2.3. Biểu đồ quan hệ giữa các bảng (Database diagram)** 15](#_Toc59607549)

[3. Danh sách các Views, Stored procedures, Functions, Triggers 16](#_Toc59607550)

[**3.1. Views** 16](#_Toc59607551)

[**3.1.1. Địa chỉ** 16](#_Toc59607552)

[*3.1.1.1. Địa chỉ* 16](#_Toc59607553)

[*3.1.1.2. Địa chỉ (View)* 17](#_Toc59607554)

[*3.1.1.3. Xã/phường* 17](#_Toc59607555)

[*3.1.1.4. Huyện/Quận* 17](#_Toc59607556)

[*3.1.1.5. Tỉnh/TP* 18](#_Toc59607557)

[**3.1.2. Hóa đơn -** Thông tin hóa đơn 18](#_Toc59607558)

[**3.1.3. Nhân viên** 18](#_Toc59607559)

[*3.1.3.1. Thông tin nhân viên* 18](#_Toc59607560)

[**3.1.4. Sinh viên** 19](#_Toc59607561)

[*3.1.4.1. Sinh viên (general)* 19](#_Toc59607562)

[*3.1.4.2. Thông tin chi tiết sinh viên* 19](#_Toc59607563)

[**3.1.5. Phòng** 20](#_Toc59607564)

[*3.1.5.1. Phiếu đăng ký phòng* 20](#_Toc59607565)

[*3.1.5.2. Loại phòng* 21](#_Toc59607566)

[**3.1.6. Dịch vụ** 21](#_Toc59607567)

[*3.1.6.1. Đơn vị dịch vụ* 21](#_Toc59607568)

[**3.2. Stored procedures** 21](#_Toc59607569)

[**3.2.1. Insert** 21](#_Toc59607570)

[*3.2.1.1. Thêm thông tin chi tiết nhân viên* 21](#_Toc59607571)

[*3.2.1.2. Thêm thông tin chi tiết sinh viên* 23](#_Toc59607572)

[*3.2.1.3. Thêm tài khoản sinh viên* 24](#_Toc59607573)

[*3.2.1.4. Thêm địa chỉ* 25](#_Toc59607574)

[*3.2.1.5. Thêm mã thẻ BHYT* 26](#_Toc59607575)

[*3.2.1.6. Thêm hóa đơn* 26](#_Toc59607576)

[*3.2.1.7. Thêm phiếu thanh toán* 27](#_Toc59607577)

[*3.2.1.8. Thêm phiếu đăng ký phòng* 28](#_Toc59607578)

[*3.2.1.9. Thông thông tin chi tiết hóa đơn* 29](#_Toc59607579)

[*3.2.1.10. Thêm dịch vụ* 30](#_Toc59607580)

[*3.2.1.11. Thêm đơn vị của dịch vụ* 31](#_Toc59607581)

[**3.2.2. Update** 32](#_Toc59607582)

[*3.2.2.1. Cập nhật thông tin địa chỉ* 32](#_Toc59607583)

[*3.2.2.2. Cập nhật thông tin phiếu đăng ký phòng* 32](#_Toc59607584)

[*3.2.2.3. Cập nhật thông tin lương nhân viên* 33](#_Toc59607585)

[*3.2.2.4. Cập nhật thông tin dịch vụ* 33](#_Toc59607586)

[*3.2.2.5. Cập nhật trạng thái của dịch vụ* 34](#_Toc59607587)

[*3.2.2.6. Cập nhật thông tin trường Đại học cho sinh viên* 34](#_Toc59607588)

[*3.2.2.7. Cập nhật thông tin nhân viên* 34](#_Toc59607589)

[**3.2.3. Đăng nhập** 36](#_Toc59607590)

[*3.2.3.1. Đăng nhập hệ thống qua Username, Password* 36](#_Toc59607591)

[*3.2.3.2. Đăng nhập hệ thống qua Username, Password, Role\_Name* 37](#_Toc59607592)

[*3.2.3.3. Thay đổi mật khẩu người dùng* 37](#_Toc59607593)

[**3.2.4. Lấy thông tin người dùng** 38](#_Toc59607594)

[*3.2.4.1.* ***Người dùng*** 38](#_Toc59607595)

[*3.2.4.1.1. Lấy danh sách tất cả người dùng* 38](#_Toc59607596)

[*3.2.4.1.2. Lấy thông tin chi tiết người dùng qua ID* 38](#_Toc59607597)

[*3.2.4.1.3. Lấy thông tin chi tiết người dùng qua Username* 39](#_Toc59607598)

[*3.2.4.1.4. Lấy danh sách tất cả admin* 39](#_Toc59607599)

[*3.2.4.1.5. Lấy thông tin chi tiết admin qua ID* 39](#_Toc59607600)

[*3.2.4.2.* ***Nhân viên*** 40](#_Toc59607601)

[*3.2.4.2.1. Lấy danh sách tất cả nhân viên* 40](#_Toc59607602)

[*3.2.4.2.2. Lấy thông tin nhân viên qua UserID* 40](#_Toc59607603)

[*3.2.4.2.3. Lấy thông tin chi tiết của nhân viên trên view qua UserID* 41](#_Toc59607604)

[*3.2.4.3.* ***Sinh viên*** 41](#_Toc59607605)

[*3.2.4.3.1. Lấy danh sách tất cả sinh viên bao gồm đầy đủ thông tin* 41](#_Toc59607606)

[*3.2.4.3.2. Lấy danh sách tất cả sinh viên hiện còn ở KTX* 41](#_Toc59607607)

[*3.2.4.3.3. Lấy danh sách sinh viên dựa trên tên trường Đại học* 42](#_Toc59607608)

[*3.2.4.3.4. Lấy danh sách sinh viên dựa trên họ tên* 42](#_Toc59607609)

[*3.2.4.3.5. Lấy danh sách sinh viên dựa trên giới tính* 43](#_Toc59607610)

[*3.2.4.3.6. Lấy danh sách sinh viên gần đúng dựa trên StudentID* 43](#_Toc59607611)

[*3.2.4.3.7. Lấy danh sách sinh viên dựa trên số CMND* 44](#_Toc59607612)

[*3.2.4.3.8. Lấy danh sách sinh viên dựa trên UserID* 44](#_Toc59607613)

[*3.2.4.3.9. Lấy danh sách sinh viên gần đúng dựa trên UserID* 45](#_Toc59607614)

[*3.2.4.3.10. Lấy danh sách sinh viên trên View dựa trên UserID* 45](#_Toc59607615)

[*3.2.4.3.11. Lấy danh sách sinh viên đã rời khỏi KTX* 45](#_Toc59607616)

[**3.2.5. Lấy thông tin địa chỉ** 46](#_Toc59607617)

[*3.2.5.1. Lấy danh sách các tỉnh/TP* 46](#_Toc59607618)

[*3.2.5.2. Lấy huyện/quận dựa vào tên tỉnh/TP* 46](#_Toc59607619)

[*3.2.5.3. Lấy tên xã/phường/thị trấn dựa theo tên huyện/quận và tỉnh/TP* 47](#_Toc59607620)

[**3.2.6. Lấy danh sách các trường Đại học** 47](#_Toc59607621)

[**3.2.7. Lấy thông tin phòng** 48](#_Toc59607622)

[*3.2.7.1. Lấy danh sách tất cả các phòng trong KTX* 48](#_Toc59607623)

[*3.2.7.2. Lấy danh sách tất cả các phòng trong KTX (View)* 48](#_Toc59607624)

[*3.2.7.3. Lấy danh sách các khu trong KTX* 49](#_Toc59607625)

[*3.2.7.4. Lấy danh sách các phòng thông qua mã khu* 49](#_Toc59607626)

[**3.2.8. Lấy thông tin hóa đơn** 49](#_Toc59607627)

[*3.2.8.1. Lấy danh sách các hóa đơn (View)* 49](#_Toc59607628)

[*3.2.8.2. Lấy danh sách hóa đơn theo một khoảng thời gian* 50](#_Toc59607629)

[*3.2.8.3. Lấy danh sách hóa đơn theo khu – phòng* 51](#_Toc59607630)

[**3.2.9. Lấy thông tin phiếu đăng ký phòng** 51](#_Toc59607631)

[*3.2.9.1. Lấy danh sách tất cả phiếu đăng ký phòng* 51](#_Toc59607632)

[*3.2.9.2. Lấy danh sách tất cả phiếu đăng ký phòng cho phòng cụ thể* 51](#_Toc59607633)

[*3.2.9.3. Lấy danh sách phiếu đăng ký phòng qua mã số sinh viên* 52](#_Toc59607634)

[*3.2.9.4. Lấy danh sách phiếu đăng ký phòng qua họ tên sinh viên* 52](#_Toc59607635)

[**3.2.10. Lấy thông tin dịch vụ** 53](#_Toc59607636)

[*3.2.10.1. Lấy danh sách dịch vụ* 53](#_Toc59607637)

[*3.2.10.2. Lấy danh sách đơn vị dịch vụ* 53](#_Toc59607638)

[*3.2.10.3. Lấy đơn vị dịch vụ qua tên dịch vụ* 53](#_Toc59607639)

[**3.2.11. Transaction** 54](#_Toc59607640)

[*3.2.11.1. Cập nhật thông tin sinh viên* 54](#_Toc59607641)

[*3.2.11.2. Thêm sinh viên* 56](#_Toc59607642)

[**3.3. Functions** 58](#_Toc59607643)

[**3.3.1. Table-valued functions** 58](#_Toc59607644)

[*3.3.1.1. Lấy danh sách sinh viên* 58](#_Toc59607645)

[*3.3.1.2. Lấy thông tin chi tiết hóa đơn* 59](#_Toc59607646)

[**3.3.2. Scalar-valued functions** 61](#_Toc59607647)

[*3.3.2.1. Tạo mã MD5* 61](#_Toc59607648)

[*3.3.2.2. Tạo mật khẩu mặc định* 61](#_Toc59607649)

[*3.3.2.3.* ***Tìm kiếm gần đúng*** 61](#_Toc59607650)

[*3.3.2.4. Đếm số lượng sinh viên ở trong phòng cụ thể* 62](#_Toc59607651)

[*3.3.2.5. Lấy mã tòa nhà dựa trên tên tòa nhà* 63](#_Toc59607652)

[*3.3.2.6. Lấy mã hóa đơn của một phòng theo tháng cụ thể* 63](#_Toc59607653)

[*3.3.2.7. Lấy mã xã/phường qua tên xã/phường* 64](#_Toc59607654)

[*3.3.2.8. Lấy mã huyện/quận qua tên huyện/quận* 64](#_Toc59607655)

[*3.3.2.9. Lấy mã tỉnh/TP qua tên tỉnh/TP* 65](#_Toc59607656)

[*3.3.2.10. Lấy chỉ số cũ của dịch vụ* 65](#_Toc59607657)

[**3.4. Triggers** 66](#_Toc59607658)

[**3.4.1. Người dùng** 66](#_Toc59607659)

[*3.4.1.1. Thay đổi mật khẩu mặc định* 66](#_Toc59607660)

[*3.4.1.2. Thay đổi mật khẩu mặc định cho sinh viên* 66](#_Toc59607661)

[*3.4.1.3. Thay đổi mật khẩu mặc định cho sinh viên* 66](#_Toc59607662)

[**3.4.2. Đăng ký phòng** 66](#_Toc59607663)

[*3.4.2.1. Khi tạo phiếu đăng ký phòng* 66](#_Toc59607664)

[*3.4.2.2. Khi hủy phiếu đăng ký phòng* 66](#_Toc59607665)

[**3.4.3. Hóa đơn** 66](#_Toc59607666)

[*3.4.3.1. Kiểm tra hóa đơn* 66](#_Toc59607667)

[*3.4.3.2. Thanh toán hóa đơn* 66](#_Toc59607668)

[4. Phân quyền trong phần mềm 68](#_Toc59607669)

[**1.1. Tạo user và roles** 68](#_Toc59607670)

[**1.2. Phân quyền đăng nhập hệ thống** 68](#_Toc59607671)

[5. Hướng dẫn sử dụng 68](#_Toc59607672)

[**1.3. Cài đặt chương trình** 68](#_Toc59607673)

[**1.4. Chi tiết** 68](#_Toc59607674)

[**1.4.1. Khởi động ứng dụng (Guest)** 68](#_Toc59607675)

[**1.4.2. Đăng nhập** 69](#_Toc59607676)

[**1.4.3. Màn hình dashboard** 69](#_Toc59607677)

[*1.4.3.1.* ***Admin*** 69](#_Toc59607678)

[*1.4.3.2.* ***Employee*** 70](#_Toc59607679)

[*1.4.3.3.* ***Student*** 70](#_Toc59607680)

[*1.4.3.4.* ***Guest*** 72](#_Toc59607681)

[**1.4.4. Chức năng** 72](#_Toc59607682)

[*1.4.4.1. Đổi mật khẩu* 72](#_Toc59607683)

[*1.4.4.2. Đăng xuất* 72](#_Toc59607684)

[*1.4.4.3. Xem và sửa thông tin cá nhân* 73](#_Toc59607685)

[*1.4.4.4. Thêm phiếu đăng ký phòng* 73](#_Toc59607686)

[*1.4.4.5. Thêm và thanh toán hóa đơn* 74](#_Toc59607687)

[*1.4.4.5.1. Thêm hóa đơn* 74](#_Toc59607688)

[*1.4.4.5.2. Thanh toán hóa đơn* 75](#_Toc59607689)

[*1.4.4.6. Thêm tài khoản sinh viên* 75](#_Toc59607690)

[*1.4.4.7. Thêm và cập nhật tài khoản nhân viên* 76](#_Toc59607691)

[*1.4.4.8. Xem danh sách nhân viên* 76](#_Toc59607692)

[*1.4.4.9. Xem danh sách sinh viên* 76](#_Toc59607693)

[*1.4.4.10. Xem danh sách các khu nhà* 76](#_Toc59607694)

[*1.4.4.11. Xem danh sách các loại phòng* 76](#_Toc59607695)

[*1.4.4.12. Xem danh sách các dịch vụ* 76](#_Toc59607696)

[*1.4.4.13. Tìm kiếm* 76](#_Toc59607697)

[*1.4.4.13.1. Sinh viên* 76](#_Toc59607698)

[*1.4.4.13.2. Phiếu đăng ký phòng* 76](#_Toc59607699)

[*1.4.4.13.3. Hóa đơn* 76](#_Toc59607700)

[*1.4.4.14. Hướng dẫn sử dụng* 76](#_Toc59607701)

[6. Kết luận 77](#_Toc59607702)

[**1.5. Các phần đã đạt được trong đồ án** 77](#_Toc59607703)

[**1.6. Các phần đặc sắc, đặc biệt trong đồ án** 77](#_Toc59607704)

[**1.7. Một số hạn chế** 77](#_Toc59607705)

[**1.8. Cam kết** 77](#_Toc59607706)

1. **Tổng quan về phần mềm**
   1. **Mục tiêu**

Hằng năm, số lượng sinh viên đăng kí ở tại các Kí túc xá để cho nhu cầu sinh hoạt đời sống là rất lớn, có thể lên đến hàng chục ngàn sinh viên. Thông tin của sinh viên và các dịch vụ cần được lưu một cách chính xác để tránh những sai sót, ảnh hướng đến chất lượng cuộc sống và đau đó cũng là kết quả học tập của sinh viên. Do đó, cần xây dựng một hệ thống quản lí kí túc xá để quản lí hiệu quả hơn.

Để giúp các Kí Túc Xá lưu trữ và quản lý các thông tin cần thiết một cách dễ dàng và khoa học. Số lượng thông tin mà kí túc xá xử lý là rất lớn và nhu cầu tin học hóa toàn bộ kí túc xá rất cần thiết, không những chỉ để lưu trữ dữ liệu một cách tập trung mà còn có thể dễ dàng thao tác và cập nhật.

* 1. **Phạm vi**

Đề tài được áp dụng trong Ký túc xá theo mô hình tự phục vụ. Các chức năng tập trung vào việc quản lý các thông tin của người dùng và dịch vụ của ký túc xá.

* 1. **Mô tả**

Kí túc xá thì có nhiều khu. Mỗi khu gồm có nhiều phòng. Mỗi phòng được xác định bởi mã khu, mã phòng, mã loại phòng. Loại phòng cho biết số lượng sinh viên tối đa có thể ở trong phòng đó, diện tích phòng ở và đơn giá phòng/tháng. Mỗi sinh viên chỉ có thể đăng kí ở 1 phòng tại 1 thời điểm xác định. Kí túc xá cho phép sinh viên ở nhiều trường khác nhau có thể vào ở, mỗi sinh viên thuộc 1 khoa trong một trường, mỗi trường có nhiều khoa. Mỗi khoa có mã khoa, tên khoa. Mỗi trường có tên trường, mã trường.

Thông tin sinh viên bao gồm: Mã số sinh viên, mã khoa, mã trường, diện sinh viên, họ tên, giới tính, ngày sinh, số CMND, mã thẻ bảo hiểm y tế, quê quán, email và số điện thoại liên hệ. Mỗi sinh viên có 1 ảnh chân dung, có thông tin liên hệ của người thân, bao gồm: số điện thoại, họ tên, địa chỉ, email. Diện sinh viên chứa các thông tin về mã diện, tên diện sinh viên, tỉ lệ miễn giảm.

Khi đăng ký ở ký túc xá, thông tin phiếu đăng ký bao gồm: mã phiếu đăng kí, mã số sinh viên, mã nhân viên đăng kí, mã khu, mã phòng, học kì, năm học, ngày giờ đăng kí, thời hạn, ngày bắt đầu ở. Mỗi nhân viên của kí túc xá bao gồm các thông tin: mã nhân viên, chứng minh nhân dân, họ tên, số điện thoại, email, địa chỉ, lương, mã người quản lí, ảnh chân dung.

KTX có nhiều dịch vụ. Thông tin mỗi dịch vụ gồm: mã dịch vụ, tên dịch vụ, giá và đơn vị tính. Mỗi phòng có thể sử dụng nhiều dịch vụ của KTX và chi trả hóa đơn dịch vụ theo tháng. Thông tin về hóa đơn cho từng tháng gồm: mã hóa đơn, ngày xuất hóa đơn, ngày thanh toán, mã nhân viên xuất hóa đơn, mã khu, mã phòng.

Mỗi người dùng sẽ có 1 tài khoản đăng nhập ứng dụng. Thông tin tài khoản gồm: tên đăng nhập, mật khẩu, loại tài khoản, mã người dùng.

* 1. **Thông tin phần mềm**

*Tên phần mềm:*Dormitory Management **–** Phần mềm Quản lý Ký túc xá.

*Ngôn ngữ lập trình:* C# (Window Form).

*Công nghệ sử dụng:* ADO.NET

*Phần mềm hỗ trợ:* SQL Server.

* 1. **Chức năng của phần mềm**
* Thêm, sửa, xóa các dịch vụ, thông tin hóa đơn và phiếu đăng ký phòng.
* Xem thông tin về các dịch vụ, loại phòng của kí túc xá.
* Thống kê mức sử dụng dịch vụ theo tháng, theo năm.
* Phân quyền:
  + Quản lý: có đầy đủ tất cả các quyền đã nêu ở trên.
  + Nhân viên: Có đầy đủ các quyền trên, ngoại trừ việc thêm, xóa, sửa nhân viên.
  + Người dùng: xem thông tin cá nhân, thông tin hóa đơn.
  + Khách truy cập: xem thông tin về dịch vụ ký túc xá.

1. **Lược đồ quan hệ**
   1. **Mô hình ERD**
      1. **Mô hình ERD**

Diagram

Description automatically generated

Chi tiết:

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

* + 1. **Lược đồ quan hệ**

**PROVINCE** (Province\_ID, Province\_Name, Province\_Type)

**DISTRICT** (District\_ID, District\_Name, District\_Type, Province\_ID)

**COMMUNE** (Commune\_ID, Commune\_Name, Commune\_Type, Priority, District\_ID)

**ADDRESS** (Address\_ID, Street, Commune\_ID, District\_ID, Province\_ID)

**SECTOR** (Sector\_ID, Sector\_Name)

**ROOM\_TYPE** (Room\_Type\_ID, Room\_Type\_Name, Price, Area, Capacity)

**ROOM** (Room\_ID, Sector\_ID, Room\_Type\_ID)

**ROOM\_REGISTRATION** (Room\_Registration\_ID, SSN, Room\_ID, Employee\_ID, Sector\_ID, Start\_Date, Semester, Academic\_Year, Duration, Status)

**INSURANCE** (Insurance\_ID, Base\_Practice, Registration\_Date, Duration)

**UNIT** (Unit\_ID, Unit\_Name)

**SERVICE**  (Service\_ID, Service\_Name, Unit\_ID, Price\_Per\_Unit, Status)

**COLLEGE** (College\_ID, College\_Code, College\_Name)

**USER** (User\_ID, Last\_Name, First\_Name, DoB, Gender, SSN, Address\_ID, Phone\_Number\_1, Phone\_Number\_2, Email, Img\_Path, User\_Name, Password, User\_Type, Status)

**RELATIVE** (User\_ID, Relative\_User\_ID, Relationship)

**ADMIN** (User\_ID)

**EMPLOYEE** (User\_ID, Start\_Date, Salary)

**STUDENT** (User\_ID, Student\_ID, Collage\_ID, Faculty, Major, Insurance\_ID, Status\_Registration\_Room)

**BILL** (Bill\_ID, Employee\_ID, Room\_ID, Created\_Time, Total, Sector\_ID, Month, Year, Status)

**BILL\_DETAIL** (Bill\_Detail\_ID, Bill\_ID, Service\_ID, Old\_Quantity, New\_Quantity, Total\_Cost)

**PAYMENT** (Payment\_ID, Bill\_ID, Employee\_ID, Paying\_Date, Amount)

* 1. **Mô tả cơ sở dữ liệu**
     1. ***Bảng PROVINCE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Province\_ID | VARCHAR(2) | Khóa chính, dùng để phân biệt mỗi tỉnh |
| 2 | Province\_Name | NVARCHAR(20) | Tên tỉnh |
| 3 | Province\_Type | VARCHAR(1) | Loại tỉnh |

* + 1. ***Bảng DISTRICT***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | District\_ID | VARCHAR(3) | Khóa chính, dùng để phân biệt mỗi huyện. |
| 2 | District\_Name | NVARCHAR(40) | Tên huyện |
| 3 | District\_Type | VARCHAR(1) | Loại huyện |
| 4 | Province\_ID | VARCHAR(2) | Mã tỉnh, là khóa ngoại đến Province\_ID của bảng PROVINCE |

* + 1. ***Bảng COMMUNE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Commune\_ID | VARCHAR(5) | Khóa chính, dùng để phân biệt các xã. |
| 2 | Commune\_Name | NVARCHAR(40) | Tên xã |
| 3 | Commune\_Type | VARCHAR(1) | Loại xã |
| 4 | Priority | VARCHAR(6) | Độ ưu tiên |
| 5 | District\_ID | VARCHAR(3) | Mã huyện, là khóa ngoại đến District\_ID của bảng DISTRICT |

* + 1. ***Bảng ADDRESS***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Address\_ID | BIGINT | Khóa chính, thứ tự của mã các địa chỉ |
| 2 | Street | NVARCHAR(50) | Tên đường |
| 3 | Commune\_ID | VARCHAR(5) | Tên xã, là khóa ngoại đến Commune\_ID của bảng COMMUNE |
| 4 | District\_ID | VARCHAR(3) | Mã huyện, là khóa ngoại đến District\_ID của bảng DISTRICT |
| 5 | Province\_ID | VARCHAR(2) | Mã tỉnh, là khóa ngoại đến Province\_ID của bảng PROVINCE |

* + 1. ***Bảng SECTOR***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Sector\_ID | VARCHAR(10) | Khóa chính, dùng để phân biệt tòa nhà |
| 2 | Sector\_Name | NVARCHAR(50) | Tên tòa nhà |

* + 1. ***Bảng ROOM\_TYPE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Room\_Type\_ID | INT | Khóa chính, mã loại phòng |
| 2 | Room\_Type\_Name | NVARCHAR(20) | Tên loại phòng |
| 3 | Price | DECIMAL(19, 4) | Giá phòng |
| 4 | Area | DECIMAL(8, 2) | Diện tích |
| 5 | Capacity | INT | Sức chứa (số lượng sinh viên tối đa) |

* + 1. ***Bảng ROOM***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Room\_ID | NVARCHAR(10) | Khóa chính, là mã phòng |
| 2 | Sector\_ID | VARCHAR(10) | Mã khu (mã tòa nhà), khóa ngoại |
| 3 | Room\_Type\_ID | INT | Mã loại phòng, khóa ngoại |

* + 1. ***Bảng ROOM\_REGISTRATION***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Room\_Registration\_ID | BIGINT | Khóa chính, mã phiếu đăng ký phòng |
| 2 | SSN | VARCHAR(12) | Số CMND/CCCD |
| 3 | Room\_ID | NVARCHAR(10) | Mã phòng, khóa ngoại |
| 4 | Employee\_ID | BIGINT | Mã nhân viên, khóa ngoại |
| 5 | Sector\_ID | VARCHAR(10) | Mã tòa nhà, khóa ngoại |
| 6 | Start\_Date | DATETIME | Ngày bắt đầu ở |
| 7 | Semester | INT | Học kỳ |
| 8 | Academic\_Year | INT | Năm học |
| 9 | Duration | NVARCHAR(20) | Thời hạn |
| 10 | Status | INT | Trạng thái |

* + 1. ***Bảng INSURANCE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Insurance\_ID | VARCHAR(15) | Khóa chính, là mã BHYT |
| 2 | Base\_Practice | N VARCHAR(50) | Nơi khám chữa bệnh ban đầu |
| 3 | Registration\_Date | DATE | Ngày đăng ký |
| 4 | Duration | INT | Thời hạn |

* + 1. ***Bảng UNIT***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Unit\_ID | INT | Khóa chính, mã đơn vị dịch vụ |
| 2 | Unit\_Name | NVARCHAR(50) | Tên đơn vị dịch vụ |

* + 1. ***Bảng SERVICE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Service\_ID | INT | Khóa chính, Mã dịch vụ |
| 2 | Service\_Name | NVARCHAR(50) | Tên dịch vụ |
| 3 | Unit\_ID | INT | Mã đơn vị dịch vụ, khóa ngoại |
| 4 | Price\_Per\_Unit | DECIMAL(19,4) | Giá dịch vụ |
| 5 | Status | BIT | Trạng thái |

* + 1. ***Bảng COLLEGE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | College\_ID | INT | Khóa chính, mã số trường Đại học |
| 2 | College\_Code | VARCHAR(5) | Mã trường (tên viết tắt khi tuyển sinh) |
| 3 | College\_Name | NVARCHAR(100) | Tên trường |

* + 1. ***Bảng USER***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | User\_ID | BIGINT | Khóa chỉnh, mã người dùng |
| 2 | Last\_Name | NVARCHAR(40) | Họ |
| 3 | First\_Name | NVARCHAR(20) | Tên |
| 4 | DoB | DATE | Ngày tháng năm sinh |
| 5 | Gender | NVARCHAR(5) | Giới tính |
| 6 | SSN | VARCHAR(12) | CMND/CCCD |
| 7 | Address\_ID | BIGINT | Mã địa chỉ |
| 8 | Phone\_Number\_1 | VARCHAR(15) | SĐT 1 |
| 9 | Phone\_Number\_2 | VARCHAR(15) | SĐT 2 |
| 10 | Email | VARCHAR(40) | Email |
| 11 | Img\_Path | VARCHAR(300) | Đường dẫn lưu ảnh đại diện |
| 12 | User\_Name | VARCHAR(16) | Tên người dùng |
| 13 | Password | VARCHAR(32) | Mật khẩu |
| 14 | User\_Type | VARCHAR(10) | Loại người dùng |
| 15 | Status | BIT | Trạng thái |

* + 1. ***Bảng RELATIVE***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** | |
| 1 | User\_ID | BIGINT | Mã người dùng, khóa ngoại | Khóa chính |
| 2 | Relative\_User\_ID | BIGINT | Mã người phụ thuộc |
| 3 | Relationship | NVARCHAR(20) | Mối quan hệ | |

* + 1. ***Bảng ADMIN***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | User\_ID | BIGINT | Mã người dùng |

* + 1. ***Bảng EMPLOYEE***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | User\_ID | BIGINT | Mã người dùng |
| 2 | Start\_Date | DATE | Ngày bắt đầu làm việc |
| 3 | Salary | DECIMAL(19,4) | Lương |

* + 1. ***Bảng STUDENT***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | User\_ID | BIGINT | Mã người dùng |
| 2 | Student\_ID | VARCHAR(15) | MSSV |
| 3 | College\_ID | INT | Mã ĐH, khóa ngoại |
| 4 | Faculty | NVARCHAR(50) | Khoa |
| 5 | Major | NVARCHAR(50) | Ngành học |
| 6 | Insurance\_ID | VARCHAR(15) | Mã BHYT, khóa ngoại |
| 7 | Status\_Registration\_Room | BIT | Trạng thái |

* + 1. ***Bảng BILL***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Bill\_ID | BIGINT | Mã hóa đơn, khóa chính |
| 2 | Employee\_ID | BIGINT | Mã nhân viên, khóa ngoại |
| 3 | Room\_ID | NVARCHAR(10) | Mã phòng, khóa ngoại |
| 4 | Created\_Time | DATETIME | Ngày lập hóa đơn |
| 5 | Total | DECIMAL(19,4) | Tổng tiền |
| 6 | Sector\_ID | VARCHAR(10) | Mã khu, khóa ngoại |
| 7 | Month | INT | Tháng |
| 8 | Year | INT | Năm |
| 9 | Status | BIT | Trạng thái |

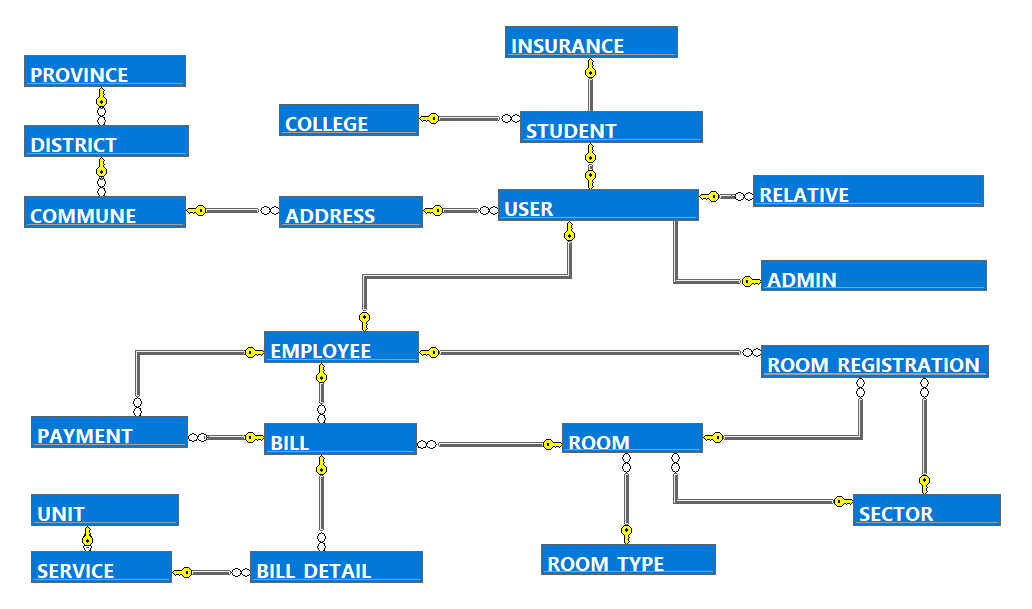
* + 1. ***Bảng BILL\_DETAIL***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Bill\_Detail\_ID | BIGINT | Mã chi tiết hóa đơn, khóa chính |
| 2 | Bill\_ID | BIGINT | Mã hóa đơn, khóa ngoại |
| 3 | Service\_ID | INT | Mã dịch vụ, khóa ngoại |
| 4 | Old\_Quantity | INT | Chỉ số cũ |
| 5 | New\_Quantity | INT | Chỉ số mới |
| 6 | Total\_Cost | DECIMAL(19,4) | Tổng hóa đơn |

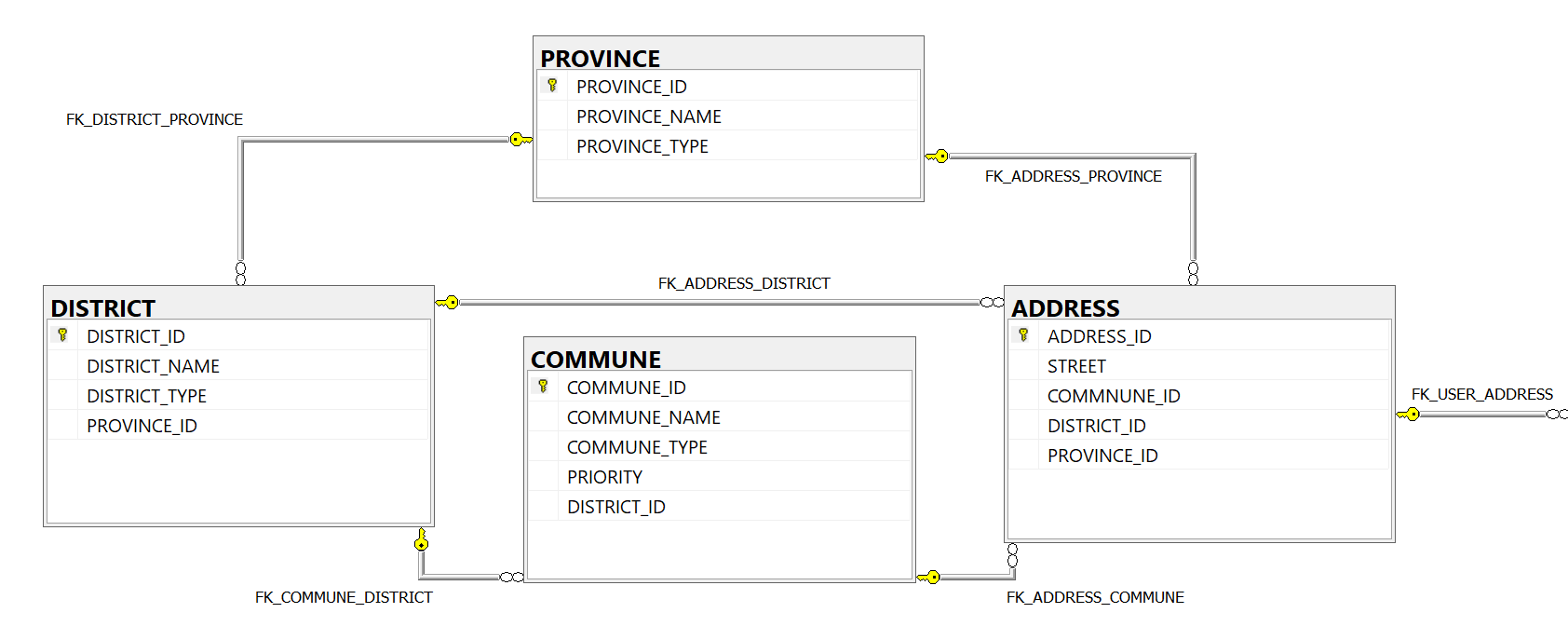
* + 1. ***Bảng PAYMENT***

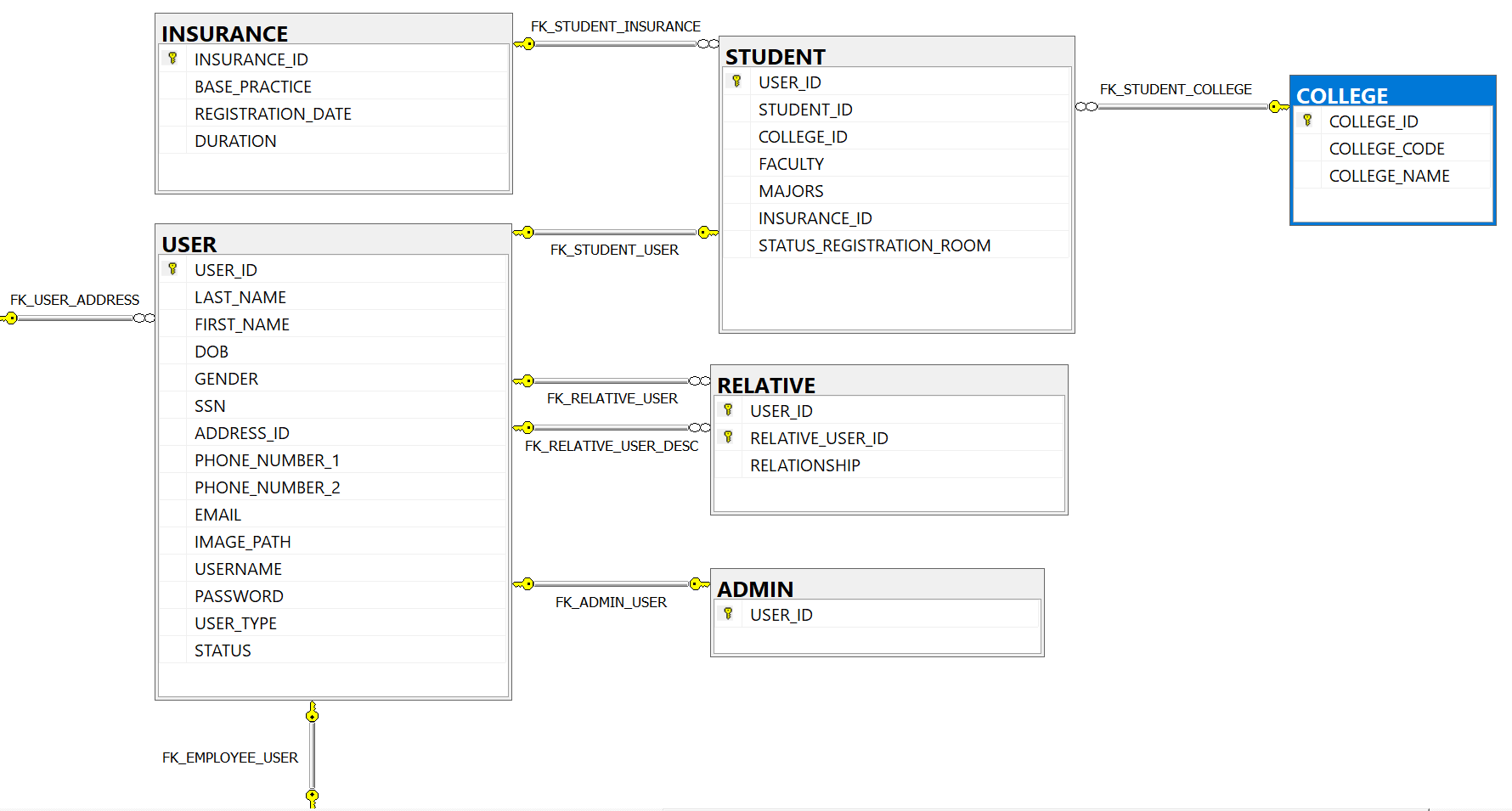
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** | |
| 1 | Payment\_ID | BIGINT | Mã phiếu thanh toán | Khóa chính của PAYMENT |
| 2 | Bill\_ID | BIGINT | Mã hóa đơn, khóa ngoại |
| 3 | Employee\_ID | BIGINT | Mã nhân viên. Khóa ngoại | |
| 4 | Paying\_Date | DATETIME | Ngày thanh toán | |
| 5 | Amount | DECIMAL(19,4) | Số tiền thanh toán | |

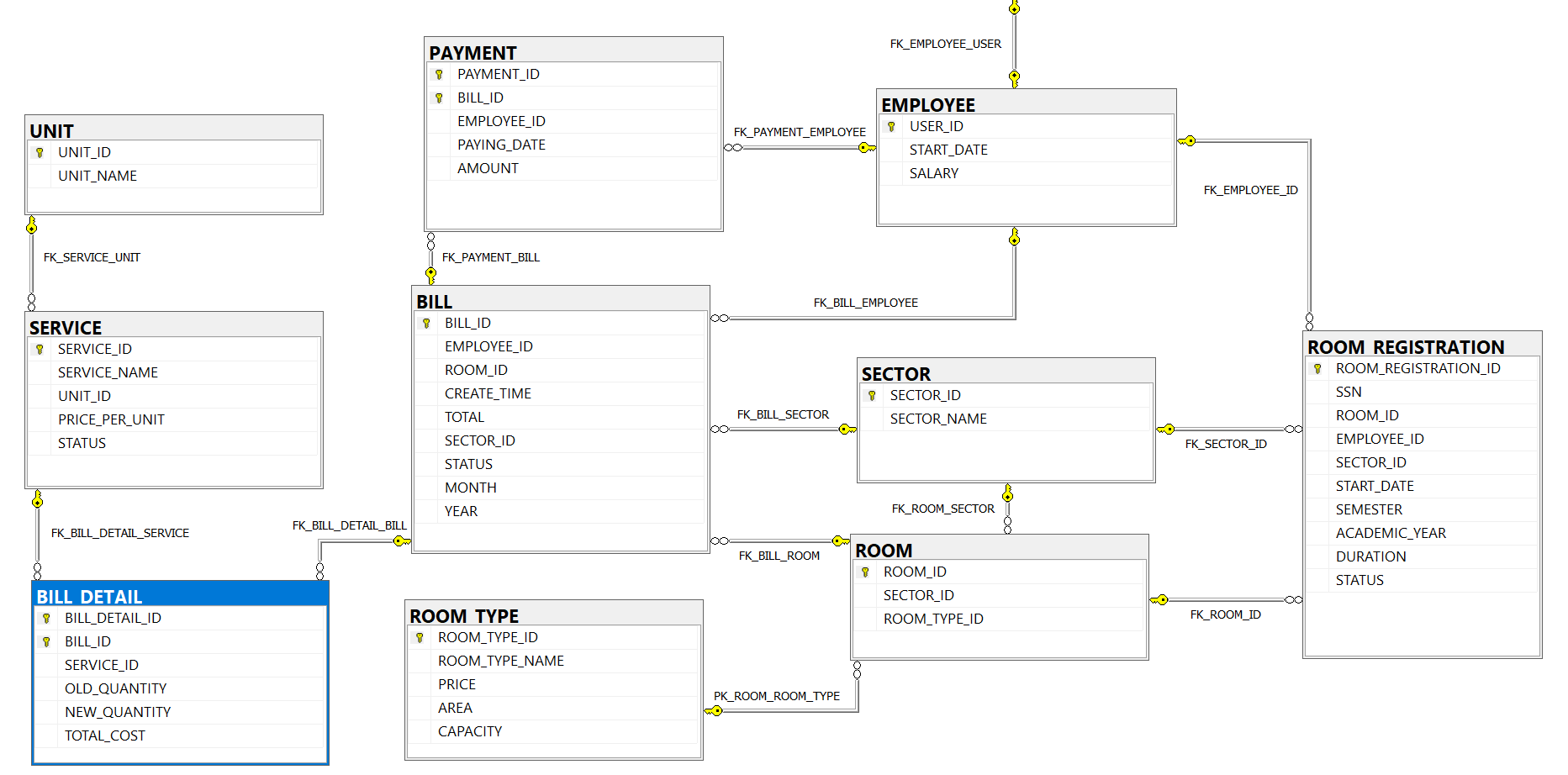
* 1. **Biểu đồ quan hệ giữa các bảng (Database diagram)**

****

**Chi tiết:**

****

****

****

1. **Danh sách các** **Views, Stored procedures,** **Functions,** **Triggers**
   1. **Views**
      1. ***Địa chỉ***
         1. *Địa chỉ*

|  |
| --- |
| -- Tạo view địa chỉ general  CREATE OR ALTER VIEW [V\_ADDRESSGENERAL]  AS      SELECT A.ADDRESS\_ID, A.STREET, P.PROVINCE\_NAME, D.DISTRICT\_NAME, C.COMMUNE\_NAME      FROM [dbo].[ADDRESS] AS A          INNER JOIN [dbo].[PROVINCE] AS P ON P.PROVINCE\_ID = A.PROVINCE\_ID          INNER JOIN [dbo].[DISTRICT] AS D ON D.DISTRICT\_ID = A.DISTRICT\_ID          INNER JOIN [dbo].[COMMUNE] AS C ON C.COMMUNE\_ID = A.COMMNUNE\_ID  GO |

* + - 1. *Địa chỉ (View)*

|  |
| --- |
| -- Tạo view địa chỉ  CREATE OR ALTER VIEW [V\_ADDRESS]  AS      SELECT A.ADDRESS\_ID, A.STREET, P.PROVINCE\_NAME, D.DISTRICT\_NAME, C.COMMUNE\_NAME      FROM [dbo].[ADDRESS] AS A          INNER JOIN [dbo].[V\_PROVINCE] AS P ON P.PROVINCE\_ID = A.PROVINCE\_ID          INNER JOIN [dbo].[V\_DISTRICT] AS D ON D.DISTRICT\_ID = A.DISTRICT\_ID          INNER JOIN [dbo].[V\_COMMUNE] AS C ON C.COMMUNE\_ID = A.COMMNUNE\_ID  GO |

* + - 1. *Xã/phường*

|  |
| --- |
| -- Tạo view xã, phường  CREATE OR ALTER VIEW [V\_COMMUNE]  AS      SELECT          C.COMMUNE\_ID,          CASE              WHEN C.COMMUNE\_TYPE = 'T' THEN CONCAT(N'Thị trấn ', C.COMMUNE\_NAME)              WHEN C.COMMUNE\_TYPE = 'W' THEN CONCAT(N'Phường ', C.COMMUNE\_NAME)              ELSE CONCAT(N'Xã ', C.COMMUNE\_NAME)          END AS COMMUNE\_NAME,          C.COMMUNE\_TYPE,          C.PRIORITY,          C.DISTRICT\_ID      FROM [dbo].[COMMUNE] AS C  GO |

* + - 1. *Huyện/Quận*

|  |
| --- |
| -- Tạo view huyện  CREATE OR ALTER VIEW [V\_DISTRICT]  AS      SELECT          D.DISTRICT\_ID,          CASE              WHEN D.DISTRICT\_TYPE = 'C' THEN CONCAT(N'Thành phố ', D.DISTRICT\_NAME)              WHEN D.DISTRICT\_TYPE = 'D' THEN CONCAT(N'Quận ', D.DISTRICT\_NAME)              WHEN D.DISTRICT\_TYPE = 'T' THEN CONCAT(N'Thị xã ', D.DISTRICT\_NAME)              ELSE CONCAT(N'Huyện ', D.DISTRICT\_NAME)          END AS DISTRICT\_NAME,          D.PROVINCE\_ID,          D.DISTRICT\_TYPE      FROM [dbo].[DISTRICT] AS D  GO |

* + - 1. *Tỉnh/TP*

|  |
| --- |
| -- Tạo view tỉnh  CREATE OR ALTER VIEW [V\_PROVINCE]  AS      SELECT          P.PROVINCE\_ID,          CASE              WHEN P.PROVINCE\_TYPE = 'C' THEN CONCAT(N'Thành phố ', P.PROVINCE\_NAME)              ELSE CONCAT(N'Tỉnh ', P.PROVINCE\_NAME)          END AS PROVINCE\_NAME ,          P.PROVINCE\_TYPE      FROM [dbo].[PROVINCE] AS P  GO |

* + 1. ***Hóa đơn*** ***-*** *Thông tin hóa đơn*

|  |
| --- |
| -- View bill  CREATE OR ALTER VIEW [V\_BILL]  AS      SELECT          B.BILL\_ID AS [Bill Id],          B.CREATE\_TIME AS [Create time],          B.EMPLOYEE\_ID AS [Employee Id],          CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS [Employee name],          SEC.SECTOR\_NAME AS [Sector],          R.ROOM\_ID AS [Room],          B.[MONTH] AS [Month],          B.[YEAR] AS [Year],          B.TOTAL AS [Total],          B.[STATUS] AS [Status]      FROM [dbo].[BILL] AS B          INNER JOIN [dbo].[SECTOR] AS SEC ON SEC.SECTOR\_ID = B.SECTOR\_ID          INNER JOIN [dbo].[USER] AS U ON U.USER\_ID = B.EMPLOYEE\_ID          INNER JOIN dbo.ROOM AS R ON R.ROOM\_ID = B.ROOM\_ID  GO |

* + 1. ***Nhân viên***
       1. *Thông tin nhân viên*

|  |
| --- |
| -- Tạo view nhân viên  CREATE OR ALTER VIEW [V\_EMPLOYEE]  AS      SELECT          U.[USER\_ID],          U.LAST\_NAME,          U.FIRST\_NAME,          CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS [FULL\_NAME],          U.DOB,          U.GENDER,          U.SSN,          A.STREET,          A.PROVINCE\_NAME,          A.DISTRICT\_NAME,          A.COMMUNE\_NAME,          U.PHONE\_NUMBER\_1,          U.PHONE\_NUMBER\_2,          U.EMAIL,          E.[START\_DATE],          E.SALARY      FROM [dbo].[USER] AS U          INNER JOIN [dbo].[V\_ADDRESSGENERAL] AS A ON A.ADDRESS\_ID = U.ADDRESS\_ID          INNER JOIN [dbo].[EMPLOYEE] AS E ON E.[USER\_ID] = U.[USER\_ID]  GO |

* + 1. ***Sinh viên***
       1. *Sinh viên (general)*

|  |
| --- |
| -- Tạo view sinh viên general  CREATE OR ALTER VIEW [V\_STUDENTGENERAL]  AS      SELECT          U.[USER\_ID],          CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS [FULL\_NAME],          U.DOB,          U.GENDER,          U.SSN,          U.PHONE\_NUMBER\_1,          U.EMAIL,          S.STUDENT\_ID,          C.COLLEGE\_NAME      FROM [dbo].[USER] AS U          INNER JOIN [dbo].[V\_ADDRESS] AS A ON A.ADDRESS\_ID = U.ADDRESS\_ID          INNER JOIN [dbo].[STUDENT] AS S ON S.[USER\_ID] = U.[USER\_ID]          INNER JOIN [dbo].[COLLEGE] AS C ON C.COLLEGE\_ID = S.COLLEGE\_ID  GO |

* + - 1. *Thông tin chi tiết sinh viên*

|  |
| --- |
| -- Tạo view sinh viên  CREATE OR ALTER VIEW [V\_STUDENT]  AS      SELECT          U.[USER\_ID],          U.LAST\_NAME,          U.FIRST\_NAME,          CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS [FULL\_NAME],          U.DOB,          U.GENDER,          U.SSN,          A.STREET,          A.PROVINCE\_NAME,          A.DISTRICT\_NAME,          A.COMMUNE\_NAME,          U.PHONE\_NUMBER\_1,          U.PHONE\_NUMBER\_2,          U.EMAIL,          S.STUDENT\_ID,          C.COLLEGE\_NAME,          S.FACULTY,          S.MAJORS      FROM [dbo].[USER] AS U          INNER JOIN [dbo].[V\_ADDRESSGENERAL] AS A ON A.ADDRESS\_ID = U.ADDRESS\_ID          INNER JOIN [dbo].[STUDENT] AS S ON S.[USER\_ID] = U.[USER\_ID]          INNER JOIN [dbo].[COLLEGE] AS C ON C.COLLEGE\_ID = S.COLLEGE\_ID  GO |

* + 1. ***Phòng***
       1. *Phiếu đăng ký phòng*

|  |
| --- |
| -- Tạo view room regestration  CREATE OR ALTER VIEW [V\_ROOM\_REGISTRATION]  AS      SELECT          RR.ROOM\_REGISTRATION\_ID AS [Id],          S.SECTOR\_NAME AS [Building],          R.ROOM\_ID AS [Room],          STU.STUDENT\_ID AS [Student Id],          STU.FULL\_NAME AS [Student Name],          E.USER\_ID AS [Employee Id],          E.FULL\_NAME AS [Employee Name],          RR.START\_DATE AS [Start date],          RR.DURATION AS [Duration]      FROM [dbo].[ROOM\_REGISTRATION] AS RR          INNER JOIN [dbo].[SECTOR] AS S ON S.SECTOR\_ID = RR.SECTOR\_ID          INNER JOIN [dbo].[ROOM] AS R ON R.ROOM\_ID = RR.ROOM\_ID          INNER JOIN (              SELECT U.SSN, S.STUDENT\_ID, CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS FULL\_NAME          FROM [dbo].[STUDENT] AS S              INNER JOIN [dbo].[USER] AS U ON U.[USER\_ID] = S.[USER\_ID]          ) AS STU ON STU.SSN = RR.SSN          INNER JOIN (              SELECT E.[USER\_ID], CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS FULL\_NAME          FROM [dbo].[EMPLOYEE] AS E              INNER JOIN [dbo].[USER] AS U ON U.[USER\_ID] = E.[USER\_ID]          ) AS E ON E.USER\_ID = RR.EMPLOYEE\_ID  GO |

* + - 1. *Loại phòng*

|  |
| --- |
| -- View Loại phòng  CREATE OR ALTER VIEW V\_GetRoomSectorType  AS      SELECT dbo.ROOM.ROOM\_ID, dbo.SECTOR.SECTOR\_NAME, dbo.ROOM\_TYPE.ROOM\_TYPE\_ID, dbo.ROOM\_TYPE.ROOM\_TYPE\_NAME, dbo.ROOM\_TYPE.PRICE, dbo.ROOM\_TYPE.AREA, dbo.ROOM\_TYPE.CAPACITY      FROM dbo.ROOM INNER JOIN dbo.SECTOR ON SECTOR.SECTOR\_ID = ROOM.SECTOR\_ID INNER JOIN dbo.ROOM\_TYPE ON ROOM\_TYPE.ROOM\_TYPE\_ID = ROOM.ROOM\_TYPE\_ID  GO |

* + 1. ***Dịch vụ***
       1. *Đơn vị dịch vụ*

|  |
| --- |
| -- View Service Unit  CREATE OR ALTER VIEW [V\_SERVICE\_UNIT]  AS      SELECT dbo.SERVICE.SERVICE\_ID, dbo.SERVICE.SERVICE\_NAME, dbo.SERVICE.PRICE\_PER\_UNIT, dbo.UNIT.UNIT\_NAME, dbo.SERVICE.STATUS      FROM dbo.SERVICE INNER JOIN          dbo.UNIT ON dbo.UNIT.UNIT\_ID = dbo.SERVICE.UNIT\_ID  GO |

* 1. **Stored procedures**
     1. ***Insert***
        1. *Thêm thông tin chi tiết nhân viên*

|  |
| --- |
| -- Dùng để thêm nhân viên  CREATE OR ALTER PROC USP\_AddEmployee      @LAST\_NAME NVARCHAR(40),      @FIRST\_NAME NVARCHAR(20),      @DOB DATE,      @GENDER NVARCHAR(5),      @SSN VARCHAR(12),      @PHONE\_NUMBER\_1 VARCHAR(15),      @PHONE\_NUMBER\_2 VARCHAR(15),      @EMAIL VARCHAR(40),      @IMAGE\_PATH VARCHAR(300),      @USER\_TYPE VARCHAR(10),      @PROVINCE\_NAME NVARCHAR(20),      @DISTRICT\_NAME NVARCHAR(40),      @COMMUNE\_NAME NVARCHAR(40),      @STREET NVARCHAR(50),      @STATR\_DATE DATE,      @SALARY DECIMAL(19,4)  AS  BEGIN      -- Tạo ra biến      DECLARE @PROVINCE\_ID VARCHAR(2)      DECLARE @DISTRICT\_ID VARCHAR(3)      DECLARE @COMMUNE\_ID VARCHAR(5)      DECLARE @ADDRESS\_ID BIGINT      DECLARE @USER\_ID BIGINT      -- Gán dữ liệu cho 3 biến      SET @PROVINCE\_ID = dbo.UFN\_GetProvinceIdByProvinceName(@PROVINCE\_NAME)      SET @DISTRICT\_ID          = dbo.UFN\_GetDistrictIdByDictrictName(@DISTRICT\_NAME, @PROVINCE\_ID)      SELECT @COMMUNE\_ID = COMMUNE\_ID      FROM dbo.COMMUNE      WHERE DISTRICT\_ID = @DISTRICT\_ID          AND COMMUNE\_NAME = @COMMUNE\_NAME      -- Thêm địa chỉ      INSERT INTO dbo.[ADDRESS] (STREET, COMMNUNE\_ID, DISTRICT\_ID, PROVINCE\_ID)      VALUES (@STREET, @COMMUNE\_ID, @DISTRICT\_ID, @PROVINCE\_ID)      -- Lấy ra mã địa chỉ      SET @ADDRESS\_ID = ( SELECT TOP 1 ADDRESS\_ID                          FROM dbo.[ADDRESS] ORDER BY ADDRESS\_ID DESC)      --Thêm vào bảng User      INSERT INTO dbo.[USER] (LAST\_NAME, FIRST\_NAME, DOB, GENDER, SSN,                              ADDRESS\_ID, PHONE\_NUMBER\_1, PHONE\_NUMBER\_2,                              EMAIL, IMAGE\_PATH, USERNAME, PASSWORD,                              USER\_TYPE, STATUS)      VALUES (@LAST\_NAME, @FIRST\_NAME, @DOB,@GENDER, @SSN,@ADDRESS\_ID,              @PHONE\_NUMBER\_1, @PHONE\_NUMBER\_2, @EMAIL, @IMAGE\_PATH,              @SSN, '', @USER\_TYPE, 1)      -- Lấy ra User vừa được thêm vào      SET @USER\_ID = (SELECT TOP 1 USER\_ID                      FROM dbo.[USER] ORDER BY USER\_ID DESC)      -- Thêm vào bảng nhân viên      INSERT INTO dbo.[EMPLOYEE] (USER\_ID, START\_DATE, SALARY)      VALUES(@USER\_ID, @STATR\_DATE, @SALARY)  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddUser(string lastName, string firstName, DateTime dob,   string gender, string ssn, string phoneNumber1, string phoneNumber2,  string email, string imagePath, string userType,  string provinceName, string districtName, string communeName,  string street, DateTime startDate, string salary)  {  string dobTemp = Convert.ToDateTime(dob).ToString("yyyyMMdd");  string startDateTemp = Convert.ToDateTime(startDate).ToString("yyyyMMdd");  string query = "EXEC dbo.USP\_AddEmployee @LAST\_NAME , @FIRST\_NAME , @DOB , @GENDER , @SSN " +  "@PHONE\_NUMBER\_1 , @PHONE\_NUMBER\_2 , @EMAIL , @IMAGE\_PATH , @USER\_TYPE , " +  "@PROVINCE\_NAME , @DISTRICT\_NAME , @COMMUNE\_NAME , @STREET , @STATR\_DATE , @SALARY";  object[] parameters = new object[]  {  lastName, firstName, dobTemp, gender, ssn,  phoneNumber1, phoneNumber2, email, imagePath, userType,  provinceName, districtName, communeName, street, startDateTemp, salary  };  int result = DataProvider.ExcuteNonQuery(query, parameters);  return result > 0;  } |

* + - 1. *Thêm thông tin chi tiết sinh viên*

|  |
| --- |
| --- THÊM USER - STUDENT  CREATE OR ALTER PROC USP\_INSERT\_USER\_STUDENT  (      @LAST\_NAME NVARCHAR(40),      @FIRST\_NAME NVARCHAR(20),      @DOB DATE,@GENDER NVARCHAR(5),      @SSN VARCHAR(12),      @PHONE\_NUMBER\_1 VARCHAR(15),      @PHONE\_NUMBER\_2 VARCHAR(15),      @EMAIL VARCHAR(40),      @IMAGE\_PATH VARCHAR(300),      @USER\_TYPE VARCHAR(10),      @STATUS BIT  )  AS  BEGIN      DECLARE @ADDRESS\_ID INT      SELECT @ADDRESS\_ID = (SELECT MAX(ADDRESS\_ID) FROM dbo.ADDRESS)      INSERT INTO dbo.[USER]      (          LAST\_NAME,          FIRST\_NAME,          DOB,          GENDER,          SSN,          ADDRESS\_ID,          PHONE\_NUMBER\_1,          PHONE\_NUMBER\_2,          EMAIL,          IMAGE\_PATH,          USERNAME,          PASSWORD,          USER\_TYPE,          STATUS      )      VALUES      (   @LAST\_NAME,         -- LAST\_NAME - nvarchar(40)          @FIRST\_NAME,        -- FIRST\_NAME - nvarchar(20)          @DOB,               -- DOB - date          @GENDER,            -- GENDER - nvarchar(5)          @SSN,               -- SSN - varchar(12)          @ADDRESS\_ID,        -- ADDRESS\_ID - bigint          @PHONE\_NUMBER\_1,    -- PHONE\_NUMBER\_1 - varchar(15)          @PHONE\_NUMBER\_2,    -- PHONE\_NUMBER\_2 - varchar(15)          @EMAIL,             -- EMAIL - varchar(40)          @IMAGE\_PATH,        -- IMAGE\_PATH - varchar(300)          @EMAIL,             -- USERNAME - varchar(16)          'student',          -- PASSWORD - varchar(32)          @USER\_TYPE,         -- USER\_TYPE - varchar(10)          @STATUS             -- STATUS - bit          )  END  GO |

* + - 1. *Thêm tài khoản sinh viên*

|  |
| --- |
| --- THÊM USER - STUDENT  CREATE OR ALTER PROC USP\_INSERT\_USER\_STUDENT  (      @LAST\_NAME NVARCHAR(40),      @FIRST\_NAME NVARCHAR(20),      @DOB DATE,@GENDER NVARCHAR(5),      @SSN VARCHAR(12),      @PHONE\_NUMBER\_1 VARCHAR(15),      @PHONE\_NUMBER\_2 VARCHAR(15),      @EMAIL VARCHAR(40),      @IMAGE\_PATH VARCHAR(300),      @USER\_TYPE VARCHAR(10),      @STATUS BIT  )  AS  BEGIN      DECLARE @ADDRESS\_ID INT      SELECT @ADDRESS\_ID = (SELECT MAX(ADDRESS\_ID) FROM dbo.ADDRESS)      INSERT INTO dbo.[USER]      (          LAST\_NAME,          FIRST\_NAME,          DOB,          GENDER,          SSN,          ADDRESS\_ID,          PHONE\_NUMBER\_1,          PHONE\_NUMBER\_2,          EMAIL,          IMAGE\_PATH,          USERNAME,          PASSWORD,          USER\_TYPE,          STATUS      )      VALUES      (   @LAST\_NAME,       -- LAST\_NAME - nvarchar(40)          @FIRST\_NAME,       -- FIRST\_NAME - nvarchar(20)          @DOB, -- DOB - date          @GENDER,       -- GENDER - nvarchar(5)          @SSN,        -- SSN - varchar(12)          @ADDRESS\_ID,         -- ADDRESS\_ID - bigint          @PHONE\_NUMBER\_1,        -- PHONE\_NUMBER\_1 - varchar(15)          @PHONE\_NUMBER\_2,        -- PHONE\_NUMBER\_2 - varchar(15)          @EMAIL,        -- EMAIL - varchar(40)          @IMAGE\_PATH,        -- IMAGE\_PATH - varchar(300)          @EMAIL,        -- USERNAME - varchar(16)          'student',        -- PASSWORD - varchar(32)          @USER\_TYPE ,        -- USER\_TYPE - varchar(10)          @STATUS       -- STATUS - bit          )  END  GO |

* + - 1. *Thêm địa chỉ*

|  |
| --- |
| -- Dùng để thêm địa chỉ  CREATE OR ALTER PROC [dbo].[USP\_INSERT\_ADDRESS]  (      @Street NVARCHAR(50),      @Commune\_Name NVARCHAR(50),      @District\_Name NVARCHAR(50),      @Province\_Name NVARCHAR(50)  )  AS  BEGIN      DECLARE @Commune\_ID VARCHAR(5),              @District\_ID VARCHAR(3),              @Province\_ID VARCHAR(2)      SELECT @Province\_ID = dbo.UFN\_GetProvinceIdByProvinceName(@Province\_Name)      SELECT @District\_ID = dbo.UFN\_GetDistrictIdByDictrictName(@District\_Name, @Province\_ID)      SELECT @Commune\_ID = dbo.UFN\_GetCommuneidByCommuneName(@Commune\_Name,@District\_ID)      INSERT INTO dbo.ADDRESS (STREET, COMMNUNE\_ID, DISTRICT\_ID, PROVINCE\_ID)      VALUES (@Street, @Commune\_ID, @District\_ID, @Province\_ID )  END  GO |

* + - 1. *Thêm mã thẻ BHYT*

|  |
| --- |
| -- Thêm bảo hiểm  CREATE OR ALTER PROC USP\_INSERT\_INSURANCE  (@Insurence\_ID VARCHAR(15))  AS  BEGIN      INSERT INTO dbo.INSURANCE      (          INSURANCE\_ID,          BASE\_PRACTICE,          REGISTRATION\_DATE,          DURATION      )      VALUES      (   @Insurence\_ID,          NULL,          NULL,          NULL          )  END  GO |

* + - 1. *Thêm hóa đơn*

|  |
| --- |
| -- Thêm Bill  CREATE OR ALTER PROC [dbo].[USP\_InsetBill]  (      @Employee\_ID BIGINT,      @Room\_Name NVARCHAR(10),      @Sector\_Name NVARCHAR(50),      @Create\_date DATETIME,      @Month INT,      @Year INT,      @Status BIT,      @total DECIMAL(19,4)  )  AS  BEGIN      DECLARE @Room\_ID NVARCHAR(10),              @Sector\_ID VARCHAR(10)      SELECT @Sector\_ID = dbo.SECTOR.SECTOR\_ID      FROM dbo.SECTOR      WHERE dbo.SECTOR.SECTOR\_NAME = @Sector\_Name      SELECT @Room\_ID = dbo.ROOM.ROOM\_ID      FROM dbo.ROOM      WHERE dbo.ROOM.SECTOR\_ID = @Sector\_ID          AND dbo.ROOM.ROOM\_ID = @Room\_Name      INSERT INTO dbo.BILL(EMPLOYEE\_ID, ROOM\_ID, CREATE\_TIME,                          TOTAL, STATUS, MONTH, YEAR, Sector\_ID)      VALUES (@Employee\_ID, @Room\_ID, @Create\_date,              @total, @Status, @Month, @Year, @Sector\_ID)  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddBill(long Employee\_ID, string Room\_Name,  string Sector\_Name, DateTime CreatDay, string Month,  string Year, string Status, string Total)  {  string CREATEDAY = Convert.ToDateTime(CreatDay).ToString("yyyyMMdd");  string query = string.Format("EXEC dbo.USP\_InsetBill " +  "@Employee\_ID = {0}," +  "@Room\_Name = N'{1}'," +  "@Sector\_Name = N'{2}'," +  "@Create\_date = '{3}'," +  "@Month = {4}," +  "@Year = {5}, " +  "@Status = {6}," +  "@total ={7} ",  Employee\_ID, Room\_Name, Sector\_Name,  CREATEDAY, Month, Year, Status, Total);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Thêm phiếu thanh toán*

|  |
| --- |
| -- Thanh toán  CREATE OR ALTER PROC [dbo].[USP\_INSERT\_PAYMENT]  (      @Employee\_ID INT,      @Paying\_Date DATETIME,      @Amount DECIMAL(19,4),      @Sector\_Name VARCHAR(10),      @Room\_ID NVARCHAR(10),      @Month INT,      @Year INT  )  AS  BEGIN      DECLARE @Bill\_ID INT      SET @Bill\_ID = dbo.UFN\_GetBillIdBySectornameRoomidMonthYear(@Sector\_Name,@Room\_ID,@Month,@Year)      INSERT INTO dbo.PAYMENT (BILL\_ID, EMPLOYEE\_ID, PAYING\_DATE, AMOUNT)      VALUES (@Bill\_ID, @Employee\_ID, @Paying\_Date, @Amount)  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddPayMent(string Employee\_ID, DateTime Paying\_Date,  Decimal Amount, string Sector\_Name, string Room\_ID, string Month, string Year)  {  string PAYINGDAY = Convert.ToDateTime(Paying\_Date).ToString("yyyyMMdd");  string query = string.Format("EXEC dbo.USP\_INSERT\_PAYMENT " +  "@Employee\_ID = {0}," +  "@Paying\_Date = '{1}', " +  "@Amount = {2}," +  "@Sector\_Name = '{3}'," +  "@Room\_ID = N'{4}', " +  "@Month = {5}," +  "@Year = {6}",  Employee\_ID, PAYINGDAY, Amount, Sector\_Name,  Room\_ID, Month, Year);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Thêm phiếu đăng ký phòng*

|  |
| --- |
| --USP\_INSERT\_ROOMREGISTRATION  CREATE OR ALTER PROC USP\_INSERT\_ROOMREGISTRATION  (      @EMPLOYEE\_ID BIGINT,      @SSN VARCHAR(12),      @SECTOR\_NAME NVARCHAR(50),      @ROOM\_ID NVARCHAR(10),      @START\_DAY DATETIME,      @SEMESTER INT,      @ACADEMIC\_YEAR INT,      @DURATION NVARCHAR(20),      @STATUS BIT  )  AS  BEGIN      DECLARE @SECTOR\_ID VARCHAR(10)      SELECT @SECTOR\_ID = dbo.UFN\_Get\_SectorID\_By\_SectorName(@SECTOR\_NAME)      INSERT INTO dbo.ROOM\_REGISTRATION      (          SSN,          ROOM\_ID,          EMPLOYEE\_ID,          SECTOR\_ID,          START\_DATE,          SEMESTER,          ACADEMIC\_YEAR,          DURATION,          STATUS      )      VALUES      (   @SSN,          @ROOM\_ID,          @EMPLOYEE\_ID,          @SECTOR\_ID,          @START\_DAY,          @SEMESTER,          @ACADEMIC\_YEAR,          @DURATION,          @STATUS          )  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddRoomRegistration(long Employee\_ID, string Ssn,  string Sector\_Name, string Room\_ID, DateTime Start\_Day, int Semester,  int Academic\_Year, string Duaration, string Status)  {  string START\_DAY = Convert.ToDateTime(Start\_Day).ToString("yyyyMMdd");  string query = string.Format("EXEC dbo.USP\_INSERT\_ROOMREGISTRATION " +  "@EMPLOYEE\_ID = {0}," +  "@SSN = {1}," +  "@SECTOR\_NAME = N'{2}', " +  "@ROOM\_ID = N'{3}'," +  "@START\_DAY = '{4}'," +  "@SEMESTER = {5}," +  "@ACADEMIC\_YEAR ={6}," +  "@DURATION = N'{7}', " +  "@STATUS = {8}",  Employee\_ID, Ssn, Sector\_Name, Room\_ID, START\_DAY,  Semester, Academic\_Year, Duaration, Status);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Thông thông tin chi tiết hóa đơn*

|  |
| --- |
| -- Thêm dịch vụ  vào chi tiết hóa đơn  CREATE OR ALTER PROC [dbo].[USP\_INSERT\_SERVICE\_BILL\_DETAIL]  (      @Service\_Name NVARCHAR(20),      @Quantity INT,      @Sector\_Name VARCHAR(10),      @Room\_ID NVARCHAR(10),      @Month INT,      @Year INT,      @Unit\_Name NVARCHAR(20),      @Total\_Cost DECIMAL(19,4))  AS  BEGIN      DECLARE @Bill\_ID BIGINT, @Service\_ID INT, @Old\_Quantity INT,@New\_Quantity INT, @Sector\_ID NVARCHAR(20)      SELECT @Bill\_ID = (SELECT MAX(dbo.BILL.BILL\_ID) FROM dbo.BILL)      SELECT @Sector\_ID = dbo.SECTOR.SECTOR\_ID FROM dbo.SECTOR WHERE dbo.SECTOR.SECTOR\_NAME = @Sector\_Name      SET @Month = @Month - 1      IF(@Month = 0)      BEGIN          SET @Month = 12          SET @Year = @Year -1      END      SELECT @Service\_ID = dbo.SERVICE.SERVICE\_ID FROM dbo.SERVICE WHERE dbo.SERVICE.SERVICE\_NAME = @Service\_Name      SET @Old\_Quantity = dbo.UFN\_GetOldQuantityForNewBill(@Sector\_ID,@Room\_ID,@Month,@Year,@Service\_Name)      SET @New\_Quantity = @Old\_Quantity + @Quantity      INSERT INTO dbo.BILL\_DETAIL(BILL\_ID, SERVICE\_ID, OLD\_QUANTITY, NEW\_QUANTITY, UNIT\_NAME,TOTAL\_COST)      VALUES      (   @Bill\_ID, -- BILL\_ID - bigint          @Service\_ID, -- SERVICE\_ID - int          @Old\_Quantity, -- OLD\_QUANTITY - int          @New\_Quantity,  -- NEW\_QUANTITY - int          @Unit\_Name,          @Total\_Cost      )  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddDetailBill(string Service\_Name, string Quantity,  string Sector\_Name, string Room\_ID, string Month, string Year,  string Unit\_Name, string Total\_Cost\_Per\_Service)  {  string query = string.Format("EXEC dbo.USP\_INSERT\_SERVICE\_BILL\_DETAIL " +  "@Service\_Name = N'{0}', " +  "@Quantity = {1}," +  "@Sector\_Name = N'{2}'," +  "@Room\_ID = N'{3}', " +  "@Month = {4}," +  "@Year = {5}," +  "@Unit\_Name = {6}," +  "@Total\_Cost = {7}",  Service\_Name, Quantity, Sector\_Name, Room\_ID, Month, Year,  Unit\_Name, Total\_Cost\_Per\_Service);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Thêm dịch vụ*

|  |
| --- |
| CREATE OR ALTER PROC USP\_InsertService  (      @Service\_Name NVARCHAR(20),      @Price\_Per\_Unit DECIMAL(19,4),      @Unit\_Name NVARCHAR(20)  )  AS  BEGIN      DECLARE @Unit\_ID INT      EXEC dbo.USP\_InsertUnit @Unit\_Name = @Unit\_Name      SELECT @Unit\_ID = ( SELECT MAX(dbo.UNIT.UNIT\_ID)                          FROM dbo.UNIT)      INSERT INTO dbo.SERVICE      (          SERVICE\_NAME,          UNIT\_ID,          PRICE\_PER\_UNIT,          STATUS      )      VALUES      (   @Service\_Name,          @Unit\_ID,          @Price\_Per\_Unit,          1          )  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddService(string Service\_Name, string Price\_Per\_Unit, string Unit\_Name)  {  string query = string.Format("EXEC dbo.USP\_InsertService " +  "@Service\_Name = N'{0}'," +  "@Price\_Per\_Unit = {1}," +  "@Unit\_Name = N'{2}'",  Service\_Name, Price\_Per\_Unit, Unit\_Name);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Thêm đơn vị của dịch vụ*

|  |
| --- |
| CREATE OR ALTER PROC USP\_InsertUnit  (@Unit\_Name NVARCHAR(20))  AS  BEGIN      INSERT INTO dbo.UNIT      (          UNIT\_NAME      )      VALUES      (       @Unit\_Name      )  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + 1. ***Update***
       1. *Cập nhật thông tin địa chỉ*

|  |
| --- |
| -- cập nhật địa chỉ  CREATE OR ALTER PROC USP\_UpdateAddRess  (      @AddRess\_ID INT,      @Street NVARCHAR(50),      @Commune\_Name NVARCHAR(50),      @District\_Name NVARCHAR(50),      @Province\_Name NVARCHAR(50)  )  AS  BEGIN      DECLARE @Commune\_ID VARCHAR(5),              @District\_ID VARCHAR(3),              @Province\_ID VARCHAR(2)      SELECT @Province\_ID = dbo.UFN\_GetProvinceIdByProvinceName(@Province\_Name)      SELECT @District\_ID = dbo.UFN\_GetDistrictIdByDictrictName(@District\_Name, @Province\_ID)      SELECT @Commune\_ID = dbo.UFN\_GetCommuneidByCommuneName(@Commune\_Name, @District\_ID)      UPDATE dbo.ADDRESS      SET          dbo.ADDRESS.STREET = @Street,          dbo.ADDRESS.COMMNUNE\_ID = @Commune\_ID,          dbo.ADDRESS.DISTRICT\_ID = @District\_ID,          dbo.ADDRESS.PROVINCE\_ID = @Province\_ID      WHERE dbo.ADDRESS.ADDRESS\_ID = @AddRess\_ID;  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Cập nhật thông tin phiếu đăng ký phòng*

|  |
| --- |
| CREATE OR ALTER PROC USP\_UpdateRoomRegistration  (      @Ssn VARCHAR(15)  )  AS  BEGIN      UPDATE dbo.ROOM\_REGISTRATION      SET dbo.ROOM\_REGISTRATION.STATUS = 0      WHERE dbo.ROOM\_REGISTRATION.SSN = @Ssn;  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Cập nhật thông tin lương nhân viên*

|  |
| --- |
| -- Cập nhật lương cho nhân viên  CREATE OR ALTER PROC USP\_UpdateSalary(      @USER\_ID INT,      @SALARY DECIMAL(19,4)      )  AS  BEGIN      UPDATE dbo.EMPLOYEE SET SALARY = @SALARY WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool UpdateSalary(long userId, string salary)  {  string query = string.Format("EXEC dbo.USP\_UpdateSalary @USER\_ID = {0}, @SALARY = {1} ", userId, salary);  DataTable result = DataProvider.ExcuteQuery(query);  return result.Rows.Count > 0;  } |

* + - 1. *Cập nhật thông tin dịch vụ*

|  |
| --- |
| -- cập nhật dịch vụ  CREATE OR ALTER PROC USP\_UpdateService  (      @Service\_Name NVARCHAR(50),      @Price DECIMAL (19,4)  )  AS  BEGIN      UPDATE dbo.SERVICE      SET dbo.SERVICE.PRICE\_PER\_UNIT = @Price      WHERE dbo.SERVICE.SERVICE\_NAME = @Service\_Name;  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool UpdateService(string Service\_Name, decimal Price)  {  string query = string.Format("EXEC dbo.USP\_UpdateService " +  "@Service\_Name = N'{0}'," +  "@Price = {1}",  Service\_Name, Price);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Cập nhật trạng thái của dịch vụ*

|  |
| --- |
| CREATE OR ALTER PROC USP\_UnableService  (      @Service\_Name NVARCHAR(20)  )  AS  BEGIN      UPDATE dbo.SERVICE      SET dbo.SERVICE.STATUS = 0      WHERE dbo.SERVICE.SERVICE\_NAME = @Service\_Name;  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool DeleteService(string Service\_Name)  {  string query = string.Format("EXEC dbo.USP\_UnableService @Service\_Name = N'{0}'", Service\_Name);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + - 1. *Cập nhật thông tin trường Đại học cho sinh viên*

|  |
| --- |
| -- Cập nhật trường đại học  CREATE OR ALTER PROC USP\_UpdateStudentCollege  (      @User\_ID INT,      @College\_Name NVARCHAR(50),      @Faculty NVARCHAR(50),      @Major NVARCHAR(50)  )  AS  BEGIN      DECLARE @College\_ID INT      SELECT @College\_ID = dbo.COLLEGE.COLLEGE\_ID  FROM dbo.COLLEGE WHERE dbo.COLLEGE.COLLEGE\_NAME = @College\_Name      UPDATE dbo.STUDENT      SET dbo.STUDENT.COLLEGE\_ID = @College\_ID,          dbo.STUDENT.FACULTY = @Faculty,          dbo.STUDENT.MAJORS = @Major      WHERE dbo.STUDENT.USER\_ID = @User\_ID;  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Cập nhật thông tin nhân viên*

|  |
| --- |
| -- Dùng để sửa nhân viên  CREATE OR ALTER PROC USP\_EditEmployee      @LAST\_NAME NVARCHAR(40),      @FIRST\_NAME NVARCHAR(20),      @DOB DATE,      @GENDER NVARCHAR(5),      @SSN VARCHAR(12),      @PHONE\_NUMBER\_1 VARCHAR(15),      @PHONE\_NUMBER\_2 VARCHAR(15),      @EMAIL VARCHAR(40),      @IMAGE\_PATH VARCHAR(300),      @USER\_TYPE VARCHAR(10),      @PROVINCE\_NAME NVARCHAR(20),      @DISTRICT\_NAME NVARCHAR(40),      @COMMUNE\_NAME NVARCHAR(40),      @STREET NVARCHAR(50),      @USER\_ID BIGINT  AS  BEGIN      -- Tạo ra biến      DECLARE @PROVINCE\_ID VARCHAR(2)      DECLARE @DISTRICT\_ID VARCHAR(3)      DECLARE @COMMUNE\_ID VARCHAR(5)      DECLARE @ADDRESS\_ID BIGINT      -- Gán dữ liệu cho 3 biến      SET @PROVINCE\_ID = dbo.UFN\_GetProvinceIdByProvinceName(@PROVINCE\_NAME)      SET @DISTRICT\_ID = dbo.UFN\_GetDistrictIdByDictrictName(@DISTRICT\_NAME, @PROVINCE\_ID)      SELECT @COMMUNE\_ID = COMMUNE\_ID      FROM dbo.COMMUNE      WHERE DISTRICT\_ID = @DISTRICT\_ID AND COMMUNE\_NAME = @COMMUNE\_NAME      -- Sửa địa chỉ      SELECT @ADDRESS\_ID = ADDRESS\_ID      FROM dbo.[USER] WHERE dbo.[USER].USER\_ID = @USER\_ID      UPDATE dbo.ADDRESS      SET STREET = @STREET,          PROVINCE\_ID = @PROVINCE\_ID,          DISTRICT\_ID = @DISTRICT\_ID,          COMMNUNE\_ID = @COMMUNE\_ID      WHERE ADDRESS\_ID = @ADDRESS\_ID      -- Sửa tài khoản      UPDATE dbo.[USER]      SET LAST\_NAME = @LAST\_NAME,          FIRST\_NAME = @FIRST\_NAME,          DOB = @DOB,          GENDER = @GENDER,          ADDRESS\_ID =@ADDRESS\_ID,          PHONE\_NUMBER\_1 = @PHONE\_NUMBER\_1,          PHONE\_NUMBER\_2 = @PHONE\_NUMBER\_2,          EMAIL = @EMAIL,          IMAGE\_PATH  = @IMAGE\_PATH,          SSN = @SSN      WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool EditUser(string lastName, string firstName, DateTime dob,  string gender, string ssn, string phoneNumber1, string phoneNumber2,  string email, string imagePath, string userType, string provinceName,  string districtName, string communeName, string street,long userId)  {  string dobTemp = Convert.ToDateTime(dob).ToString("yyyyMMdd");  string query = string.Format("EXEC dbo.USP\_EditEmployee " +  "@LAST\_NAME = N'{0}'," +  "@FIRST\_NAME = N'{1}'," +  "@DOB = '{2}'," +  "@GENDER = N'{3}'," +  "@SSN = '{4}'," +  "@PHONE\_NUMBER\_1 = '{5}'," +  "@PHONE\_NUMBER\_2 = '{6}', " +  "@EMAIL = '{7}', " +  "@IMAGE\_PATH = '{8}', " +  "@USER\_TYPE = '{9}', " +  "@PROVINCE\_NAME = N'{10}', " +  "@DISTRICT\_NAME = N'{11}', " +  "@COMMUNE\_NAME = N'{12}'," +  "@STREET = N'{13}', " +  "@USER\_ID = {14}",  lastName, firstName, dobTemp, gender, ssn,  phoneNumber1, phoneNumber2, email, imagePath, userType,  provinceName, districtName, communeName, street,userId);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + 1. ***Đăng nhập***
       1. *Đăng nhập hệ thống qua Username, Password*

|  |
| --- |
| -- Dùng để đăng nhập:  CREATE OR ALTER PROC USP\_Login      @USERNAME VARCHAR(16),      @PASSWORD VARCHAR(32)  AS  BEGIN      DECLARE @PASSWORD\_GENERATE VARCHAR(32)      SET @PASSWORD\_GENERATE = dbo.UFN\_GenerateMD5(@PASSWORD)      SELECT \* FROM dbo.[USER]      WHERE USERNAME = @USERNAME COLLATE                          SQL\_Latin1\_General\_CP1\_CS\_AS              AND PASSWORD = @PASSWORD\_GENERATE                          COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS  -- Phân biệt chữ hoa chữ thường  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool Login(string userName, string passWord)  {  string query = "EXEC dbo.USP\_Login @USERNAME , @PASSWORD";//, userName, passWord);  DataTable result = DataProvider.ExcuteQuery(query, new object[] { userName, passWord });  return result.Rows.Count > 0;  } |

* + - 1. *Đăng nhập hệ thống qua Username, Password, Role\_Name*

|  |
| --- |
| --Tạo login SQL  CREATE OR ALTER PROC USP\_CREATE\_LOGIN\_USER  (      @Role\_Name VARCHAR(50),      @Login\_Name VARCHAR(50),      @Password\_Login VARCHAR(50)  )  AS  BEGIN      DECLARE @Login\_UserName VARCHAR(50),              @QueryLogin VARCHAR(100),              @QueryUser VARCHAR(100)      SET @Login\_UserName = @Role\_Name + @Login\_Name      SET @QueryLogin ='CREATE LOGIN ' + @Login\_UserName + ' WITH PASSWORD = ' + QUOTENAME(@Password\_Login, '''')      SET @QueryUser = CONCAT('CREATE USER ', @Login\_UserName, ' FOR LOGIN ', @Login\_UserName);      EXEC (@QueryLogin)      EXEC (@QueryUser)  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Thay đổi mật khẩu người dùng*

|  |
| --- |
| -- Dùng để thay đổi mật khẩu  CREATE OR ALTER PROC USP\_ChangePassword      @USER\_ID BIGINT,      @NEWPASS VARCHAR(32)  AS  BEGIN      DECLARE @newPassword VARCHAR(32)      SET @newPassword = dbo.UFN\_GenerateMD5(@NEWPASS)      UPDATE dbo.[USER]      SET PASSWORD = @newPassword      WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool ChangePassword(int userId, string newpass)  {  string query = string.Format("EXEC dbo.USP\_ChangePassword " +  "@USER\_ID = {0}, " +  "@NEWPASS = '{1}'",  userId, newpass);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  } |

* + 1. ***Lấy thông tin người dùng***
       1. ***Người dùng***
          1. *Lấy danh sách tất cả người dùng*

|  |
| --- |
| -- Lấy danh sách user  CREATE OR ALTER PROC USP\_GetListUser  AS BEGIN      SELECT \* FROM dbo.[USER]  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<UserDTO> GetListUser()  {  List<UserDTO> userDTOs = new List<UserDTO>();  string query = string.Format("EXEC dbo.USP\_GetListUser");  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  UserDTO userDTO = new UserDTO(item);  userDTOs.Add(userDTO);  }  return userDTOs;  } |

* + - * 1. *Lấy thông tin chi tiết người dùng qua ID*

|  |
| --- |
| -- Dùng để lấy người dùng bằng id.  CREATE OR ALTER PROC USP\_GetUserById      @USER\_ID BIGINT  AS  BEGIN      SELECT \* FROM dbo.[USER] WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static UserDTO GetUserById(string id)  {  string query = string.Format("EXEC dbo.USP\_GetUserById @USER\_ID = {0}", id);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  return new UserDTO(item);  }  return null;  } |

* + - * 1. *Lấy thông tin chi tiết người dùng qua Username*

|  |
| --- |
| -- Dùng để lấy người dùng bằng username.  CREATE OR ALTER PROC USP\_GetUserByUsername      @USERNAME VARCHAR(16)  AS  BEGIN      SELECT \* FROM dbo.[USER] WHERE USERNAME = @USERNAME  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static UserDTO GetUserByUsername(string userName)  {  string query = string.Format("EXEC dbo.USP\_GetUserByUsername @USERNAME = '{0}'",userName);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  return new UserDTO(item);  }  return null;  } |

* + - * 1. *Lấy danh sách tất cả admin*

|  |
| --- |
| -- Dùng để lấy danh sách quản trị viên.  CREATE OR ALTER PROC USP\_GetListAdmin  AS  BEGIN      SELECT \* FROM dbo.[USER] WHERE USER\_TYPE = 'ADMIN'  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - * 1. *Lấy thông tin chi tiết admin qua ID*

|  |
| --- |
| -- Dùng để lấy quản trị viên bằng id.  CREATE OR ALTER PROC USP\_GetAdminById      @USER\_ID BIGINT  AS  BEGIN      SELECT \* FROM dbo.ADMIN WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static AdminDTO GetAdminById(long id)  {  string query = string.Format("EXEC dbo.USP\_GetAdminById @USER\_ID = {0}", id);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  return new AdminDTO(item);  }  return null;  } |

* + - 1. ***Nhân viên***
         1. *Lấy danh sách tất cả nhân viên*

|  |
| --- |
| -- Lấy danh sách nhân viên  CREATE OR ALTER PROC USP\_GetListEmployeeView  AS  BEGIN      SELECT \* FROM dbo.V\_EMPLOYEE  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetEmployees()  {  string query = string.Format("EXEC dbo.USP\_GetListEmployeeView");  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - * 1. *Lấy thông tin nhân viên qua UserID*

|  |
| --- |
| -- Dùng để lấy nhân viên bằng id.  CREATE OR ALTER PROC USP\_GetEmployeeById      @USER\_ID BIGINT  AS  BEGIN      SELECT \* FROM dbo.EMPLOYEE WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static EmployeeDTO GetEmployeeById(long id)  {  string query = string.Format("EXEC dbo.USP\_GetEmployeeById @USER\_ID = {0}", id);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  return new EmployeeDTO(item);  }  return null;  } |

* + - * 1. *Lấy thông tin chi tiết của nhân viên trên view qua UserID*

|  |
| --- |
| -- Lấy 1 nhân viên  CREATE OR ALTER PROC USP\_GetEmployeeView(      @USER\_ID INT      )  AS  BEGIN      SELECT \* FROM dbo.V\_EMPLOYEE WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetEmployeesByUserId(long userId)  {  string query = string.Format("EXEC dbo.USP\_GetEmployeeView @USER\_ID = {0}",userId);  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. ***Sinh viên***
         1. *Lấy danh sách tất cả sinh viên bao gồm đầy đủ thông tin*

|  |
| --- |
| -- Lấy danh sách sinh viên general  CREATE OR ALTER PROC USP\_GetListStudentGeneral  AS  BEGIN      SELECT  \* FROM dbo.UFN\_GetListStudent(2)  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> GetListStudentView()  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneral");  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentView = new StudentViewDTO(item);  studentViewDTOs.Add(studentView);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách tất cả sinh viên hiện còn ở KTX*

|  |
| --- |
| -- Lấy danh sách sinh viên general Alive  CREATE OR ALTER PROC USP\_GetListStudentGeneralALive  AS  BEGIN      SELECT  \* FROM dbo.UFN\_GetListStudent(1)  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> GetListStudentViewALive()  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralALive");  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentView = new StudentViewDTO(item);  studentViewDTOs.Add(studentView);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên dựa trên tên trường Đại học*

|  |
| --- |
| CREATE OR ALTER PROC USP\_GetListStudentGeneralByCollegeName(@COLLEGE\_NAME NVARCHAR(150))  AS BEGIN      SELECT \* FROM dbo.V\_STUDENTGENERAL WHERE dbo.[SearchLike](COLLEGE\_NAME)      LIKE N'%' + dbo.[SearchLike](@COLLEGE\_NAME) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> SearchStudentViewByCollege(string college)  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralByCollegeName @COLLEGE\_NAME = N'{0}'", college);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentViewDTO = new StudentViewDTO(item);  studentViewDTOs.Add(studentViewDTO);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên dựa trên họ tên*

|  |
| --- |
| -- Tìm kiếm gần đúng sinh viên bởi FullName  CREATE OR ALTER PROC USP\_GetListStudentGeneralByFullName(@FULL\_NAME NVARCHAR(100))  AS BEGIN      SELECT \* FROM dbo.V\_STUDENTGENERAL WHERE dbo.[SearchLike](FULL\_NAME)      LIKE N'%' + dbo.[SearchLike](@FULL\_NAME) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> SearchStudentViewByFullName(string fullName)  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralByFullName @FULL\_NAME = N'{0}'", fullName);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentViewDTO = new StudentViewDTO(item);  studentViewDTOs.Add(studentViewDTO);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên dựa trên giới tính*

|  |
| --- |
| -- Tìm kiếm gần đúng sinh viên bởi Gender  CREATE OR ALTER PROC USP\_GetListStudentGeneralByGender(@GENDER NVARCHAR(10))  AS BEGIN      SELECT \* FROM dbo.V\_STUDENTGENERAL WHERE dbo.[SearchLike](GENDER)      LIKE N'%' + dbo.[SearchLike](@GENDER) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> SearchStudentViewByGender(string gender)  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralByGender @GENDER = N'{0}'", gender);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentViewDTO = new StudentViewDTO(item);  studentViewDTOs.Add(studentViewDTO);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên gần đúng dựa trên StudentID*

|  |
| --- |
| -- Tìm kiếm gần đúng sinh viên bởi StudentId  CREATE OR ALTER PROC USP\_GetListStudentGeneralByStudentId(@STUDENT\_ID VARCHAR(15))  AS BEGIN      SELECT \* FROM dbo.V\_STUDENTGENERAL WHERE dbo.[SearchLike](STUDENT\_ID)      LIKE N'%' + dbo.[SearchLike](@STUDENT\_ID) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> SearchStudentViewByStudentId(string studentId)  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralByStudentId @STUDENT\_ID = '{0}'", studentId);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentViewDTO = new StudentViewDTO(item);  studentViewDTOs.Add(studentViewDTO);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên dựa trên số CMND*

|  |
| --- |
| -- Tìm kiếm gần đúng sinh viên bởi Ssn  CREATE OR ALTER PROC USP\_GetListStudentGeneralBySsn (@SSN VARCHAR(12))  AS BEGIN      SELECT \* FROM dbo.V\_STUDENTGENERAL WHERE dbo.[SearchLike](SSN)      LIKE N'%' + dbo.[SearchLike](@SSN) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> SearchStudentViewBySsn(string ssn)  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralBySsn @SSN = '{0}'", ssn);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentViewDTO = new StudentViewDTO(item);  studentViewDTOs.Add(studentViewDTO);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên dựa trên UserID*

|  |
| --- |
| -- Dùng để lấy sinh viên bằng id.  CREATE OR ALTER PROC USP\_GetStudentById      @USER\_ID BIGINT  AS  BEGIN      SELECT \* FROM dbo.STUDENT WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static StudentDTO GetStudentById(long id)  {  string query = string.Format("EXEC dbo.USP\_GetStudentById @USER\_ID = {0}", id);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  return new StudentDTO(item);  }  return null;  } |

* + - * 1. *Lấy danh sách sinh viên gần đúng dựa trên UserID*

|  |
| --- |
| -- Tìm kiếm gần đúng sinh viên bởi userId  CREATE OR ALTER PROC USP\_GetListStudentGeneralByUserId(@USER\_ID VARCHAR(10))  AS BEGIN      SELECT \* FROM dbo.V\_STUDENTGENERAL WHERE dbo.[SearchLike](USER\_ID)      LIKE N'%' + dbo.[SearchLike](@USER\_ID) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<StudentViewDTO> SearchStudentViewByUserId(string userId)  {  List<StudentViewDTO> studentViewDTOs = new List<StudentViewDTO>();  string query = string.Format("EXEC dbo.USP\_GetListStudentGeneralByUserId @USER\_ID = '{0}'", userId);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  StudentViewDTO studentViewDTO = new StudentViewDTO(item);  studentViewDTOs.Add(studentViewDTO);  }  return studentViewDTOs;  } |

* + - * 1. *Lấy danh sách sinh viên trên View dựa trên UserID*

|  |
| --- |
| -- Lấy sinh viên bằng mã  CREATE OR ALTER PROC USP\_GetListStudentView(      @USER\_ID INT      )  AS  BEGIN      SELECT \* FROM dbo.[V\_STUDENT] WHERE USER\_ID = @USER\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListStudentView(long user)  {  string query = string.Format("EXEC dbo.USP\_GetListStudentView @USER\_ID = {0}",user);  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - * 1. *Lấy danh sách sinh viên đã rời khỏi KTX*

|  |
| --- |
| -- Lấy danh sách sinh viên general Alive  CREATE OR ALTER PROC USP\_GetListStudentGeneralGoingOut  AS  BEGIN      SELECT \* FROM dbo.UFN\_GetListStudent(0)  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + 1. ***Lấy thông tin địa chỉ***
       1. *Lấy danh sách các tỉnh/TP*

|  |
| --- |
| -- Dùng để lấy Province.  CREATE OR ALTER PROC USP\_GetListProvince  AS  BEGIN      SELECT \* FROM dbo.PROVINCE  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<ProvinceDTO> GetListProvince()  {  List<ProvinceDTO> provinceDTOs = new List<ProvinceDTO>();  string query = string.Format("EXEC dbo.USP\_GetListProvince");  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  ProvinceDTO provinceDTO = new ProvinceDTO(item);  provinceDTOs.Add(provinceDTO);  }  return provinceDTOs;  } |

* + - 1. *Lấy huyện/quận dựa vào tên tỉnh/TP*

|  |
| --- |
| -- Dùng để lấy quận, huyện bằng tên tỉnh.  CREATE OR ALTER PROC USP\_GetListDistrictByProvinceName      @PROVINCE\_NAME NVARCHAR(20)  AS  BEGIN      DECLARE @PROVINCE\_ID VARCHAR(2)      SET @PROVINCE\_ID = dbo.UFN\_GetProvinceIdByProvinceName(@PROVINCE\_NAME)      SELECT \* FROM dbo.DISTRICT WHERE PROVINCE\_ID = @PROVINCE\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<DistrictDTO> GetListDistrictByProvinceName(string provinceName)  {  List<DistrictDTO> districtDTOs = new List<DistrictDTO>();  string query = string.Format("EXEC dbo.USP\_GetListDistrictByProvinceName " +  "@PROVINCE\_NAME = N'{0}'", provinceName);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  DistrictDTO districtDTO = new DistrictDTO(item);  districtDTOs.Add(districtDTO);  }  return districtDTOs;  } |

* + - 1. *Lấy tên xã/phường/thị trấn dựa theo tên huyện/quận và tỉnh/TP*

|  |
| --- |
| -- Dùng để lấy xã, phường bằng của huyện bằng tỉnh, huyện.  CREATE OR ALTER PROC USP\_GetListCommuneByProvinceAndDistrict      @PROVINCE\_NAME NVARCHAR(20),      @DISTRICT\_NAME NVARCHAR(40)  AS  BEGIN      DECLARE @PROVINCE\_ID VARCHAR(2)      DECLARE @DISTRICT\_ID VARCHAR(3)      SET @PROVINCE\_ID = dbo.UFN\_GetProvinceIdByProvinceName(@PROVINCE\_NAME)      SET @DISTRICT\_ID = dbo.UFN\_GetDistrictIdByDictrictName(@DISTRICT\_NAME, @PROVINCE\_ID)      SELECT \* FROM dbo.COMMUNE WHERE DISTRICT\_ID = @DISTRICT\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<CommuneDTO> GetLisCommuneByProvinceAndDistrict(string provinceName, string districtName)  {  List<CommuneDTO> communeDTOs = new List<CommuneDTO>();  string query = string.Format("EXEC dbo.USP\_GetListCommuneByProvinceAndDistrict " +  "@PROVINCE\_NAME = N'{0}', @DISTRICT\_NAME = N'{1}'", provinceName,districtName);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  CommuneDTO communeDTO = new CommuneDTO(item);  communeDTOs.Add(communeDTO);  }  return communeDTOs;  } |

* + 1. ***Lấy danh sách các trường Đại học***

|  |
| --- |
| -- Lấy danh sách College  CREATE OR ALTER PROC USP\_GetListCollege  AS  BEGIN      SELECT \*      FROM dbo.COLLEGE  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<CollegeDTO> GetListColloge()  {  string query = string.Format("EXEC dbo.USP\_GetListCollege");  DataTable dataTable = DataProvider.ExcuteQuery(query);  List<CollegeDTO> listCollege = new List<CollegeDTO>();  foreach (DataRow item in dataTable.Rows)  {  CollegeDTO temp = new CollegeDTO(item);  listCollege.Add(temp);  }  return listCollege;  } |

* + 1. ***Lấy thông tin phòng***
       1. *Lấy danh sách tất cả các phòng trong KTX*

|  |
| --- |
| -- Lấy danh sách phòng  CREATE OR ALTER PROC [dbo].[USP\_GetListRoom]  AS  BEGIN      SELECT \* FROM dbo.ROOM  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<RoomDTO> GetListRoom()  {  string query = string.Format("EXEC dbo.USP\_GetListRoom");  DataTable dataTable = DataProvider.ExcuteQuery(query);  List<RoomDTO> ListRoom = new List<RoomDTO>();  foreach (DataRow item in dataTable.Rows)  {  RoomDTO Room = new RoomDTO(item);  ListRoom.Add(Room);  }  return ListRoom;  } |

* + - 1. *Lấy danh sách tất cả các phòng trong KTX (View)*

|  |
| --- |
| -- Lấy danh sách roomview  CREATE OR ALTER PROC USP\_GetListRoomView  AS  BEGIN      SELECT \* FROM V\_GetRoomSectorType ORDER BY SECTOR\_NAME ASC  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListRoomView()  {  string query = string.Format("EXEC dbo.USP\_GetListRoomView");  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. *Lấy danh sách các khu trong KTX*

|  |
| --- |
| -- Lấy danh sách khu  CREATE OR ALTER PROC [dbo].[USP\_GetListSector]  AS  BEGIN      SELECT \* FROM dbo.SECTOR  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<SectorDTO> GetListSector()  {  string query = string.Format("EXEC dbo.USP\_GetListSector");  DataTable dataTable = DataProvider.ExcuteQuery(query);  List<SectorDTO> ListSector = new List<SectorDTO>();  foreach (DataRow item in dataTable.Rows)  {  SectorDTO Sector = new SectorDTO(item);  ListSector.Add(Sector);  }  return ListSector;  } |

* + - 1. *Lấy danh sách các phòng thông qua mã khu*

|  |
| --- |
| -- Lấy danh sách phòng bằng Sector\_Id  CREATE OR ALTER PROC [dbo].[USP\_GetListRoomBySectorID]      @Sector\_ID VARCHAR(10)  AS  BEGIN      SELECT \* FROM dbo.ROOM      WHERE dbo.ROOM.SECTOR\_ID = @Sector\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<RoomDTO> GetListRoomBySector(string Sector\_ID)  {  string query = string.Format("EXEC dbo.USP\_GetListRoomBySectorID @Sector\_ID = {0}", Sector\_ID);  DataTable dataTable = DataProvider.ExcuteQuery(query);  List<RoomDTO> ListRoom = new List<RoomDTO>();  foreach (DataRow item in dataTable.Rows)  {  RoomDTO Room = new RoomDTO(item);  ListRoom.Add(Room);  }  return ListRoom;  } |

* + 1. ***Lấy thông tin hóa đơn***
       1. *Lấy danh sách các hóa đơn (View)*

|  |
| --- |
| -- Lấy danh sách các hóa đơn  CREATE OR ALTER PROC USP\_GetListBillView  AS  BEGIN      SELECT \* FROM dbo.[V\_BILL]  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListBillView()  {  string query = string.Format("EXEC dbo.USP\_GetListBillView");  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. *Lấy danh sách hóa đơn theo một khoảng thời gian*

|  |
| --- |
| -- Lấy danh sách Bill View từ tháng này, năm ngày đến tháng kia năm kia  CREATE OR ALTER PROC USP\_GetListBillViewByDate(      @MONTH\_FROM INT,      @YEAR\_FROM INT,      @MONTH\_TO INT,      @YEAR\_TO INT      )  AS      BEGIN          IF(@YEAR\_FROM > @YEAR\_TO)              BEGIN                  SELECT \* FROM dbo.[V\_BILL] WHERE Year > @YEAR\_FROM AND YEAR < @YEAR\_TO              END          IF (@YEAR\_FROM = @YEAR\_TO)              BEGIN                  SELECT \* FROM dbo.[V\_BILL] WHERE Month > @MONTH\_FROM - 1 AND Month < @MONTH\_TO + 1 AND Year = @YEAR\_TO              END          IF (@YEAR\_FROM < @YEAR\_TO)              BEGIN                  SELECT \* FROM dbo.[V\_BILL] WHERE Month > @MONTH\_FROM - 1 AND Month < 13 AND Year = @YEAR\_FROM                  UNION                  SELECT \* FROM dbo.[V\_BILL] WHERE Month > 0 AND Month < @MONTH\_TO + 1 AND Year = @YEAR\_TO                  UNION                  SELECT \* FROM dbo.[V\_BILL] WHERE Month > 0 AND Month < 13 AND Year > @YEAR\_FROM AND Year < @YEAR\_TO              END      END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy danh sách hóa đơn theo khu – phòng*

|  |
| --- |
| -- Lấy danh sách Bill View bằng Sector và room  CREATE OR ALTER PROC USP\_GetListBillViewBySectorAndRoom(      @SECTOR\_NAME NVARCHAR(50),      @ROOM\_ID NVARCHAR(10)      )  AS BEGIN      SELECT \* FROM dbo.[V\_BILL] WHERE Sector = @SECTOR\_NAME AND Room = @ROOM\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListBillViewByDate(int monthFrom,  int yearFrom, int monthTo, int yearTo)  {  string query = string.Format("EXEC dbo.USP\_GetListBillViewByDate " +  "@MONTH\_FROM = {0}, @YEAR\_FROM = {1}, " +  "@MONTH\_TO = {2}, @YEAR\_TO = {3}",  monthFrom, yearFrom, monthTo, yearTo);  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + 1. ***Lấy thông tin phiếu đăng ký phòng***
       1. *Lấy danh sách tất cả phiếu đăng ký phòng*

|  |
| --- |
| -- Lấy danh sách Room\_Registration  CREATE OR ALTER PROC USP\_GetListRoomRegistration  AS BEGIN      SELECT \* FROM dbo.V\_ROOM\_REGISTRATION  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListRoomRegistration()  {  string query = string.Format("EXEC dbo.USP\_GetListRoomRegistration");  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. *Lấy danh sách tất cả phiếu đăng ký phòng cho phòng cụ thể*

|  |
| --- |
| -- Lấy danh sách Room\_Registration bằng Sector và room  CREATE OR ALTER PROC USP\_GetListRoomRegistrationBySectorAndRoom(      @SECTOR\_NAME NVARCHAR(50),      @ROOM\_ID NVARCHAR(10)      )  AS BEGIN      SELECT \* FROM dbo.V\_ROOM\_REGISTRATION      WHERE Building = @SECTOR\_NAME AND Room = @ROOM\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListRoomRegistrationByBuldingAndRoom(string bulding, string room)  {  string query = string.Format("EXEC dbo.USP\_GetListRoomRegistrationBySectorAndRoom @SECTOR\_NAME = N'{0}', @ROOM\_ID = N'{1}'", bulding,room);  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. *Lấy danh sách phiếu đăng ký phòng qua mã số sinh viên*

|  |
| --- |
| -- Lấy danh sách Room\_Registration bằng StudentId  CREATE OR ALTER PROC USP\_GetListRoomRegistrationByStudentId(@STUDENT\_ID VARCHAR(15))  AS BEGIN      SELECT \* FROM dbo.V\_ROOM\_REGISTRATION      WHERE dbo.[SearchLike]([Student Id])      LIKE N'%' + dbo.[SearchLike](@STUDENT\_ID) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListRoomRegistrationByStudentId(string studentId)  {  string query = string.Format("EXEC dbo.USP\_GetListRoomRegistrationByStudentId @STUDENT\_ID = '{0}'",studentId);  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. *Lấy danh sách phiếu đăng ký phòng qua họ tên sinh viên*

|  |
| --- |
| -- Lấy danh sách Room\_Registration bằng StudentName  CREATE OR ALTER PROC USP\_GetListRoomRegistrationByStudentName(@STUDENT\_NAME NVARCHAR(100))  AS BEGIN      SELECT \* FROM dbo.V\_ROOM\_REGISTRATION      WHERE dbo.[SearchLike]([Student Name])      LIKE N'%' + dbo.[SearchLike](@STUDENT\_NAME) + '%'  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetListRoomRegistrationByStudentName(string studentName)  {  string query = string.Format("EXEC dbo.USP\_GetListRoomRegistrationByStudentName @STUDENT\_NAME = N'{0}'", studentName);  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + 1. ***Lấy thông tin dịch vụ***
       1. *Lấy danh sách dịch vụ*

|  |
| --- |
| -- Lấy danh sách service  CREATE OR ALTER PROC [dbo].[USP\_GetServicesInfo]  AS  BEGIN      SELECT s.SERVICE\_ID, s.SERVICE\_NAME, s.PRICE\_PER\_UNIT, u.UNIT\_NAME      FROM DBO.SERVICE s INNER JOIN dbo.UNIT u ON s.UNIT\_ID = u.UNIT\_ID      WHERE s.STATUS = 1;  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static DataTable GetServicesInfo()  {  string query = string.Format("EXEC dbo.USP\_GetServicesInfo");  DataTable dataTable = DataProvider.ExcuteQuery(query);  return dataTable;  } |

* + - 1. *Lấy danh sách đơn vị dịch vụ*

|  |
| --- |
| -- Lấy danh sách đơn vị của dịch vụ  CREATE OR ALTER PROC [dbo].[GetUnitByServiceName]      @Service\_name NVARCHAR(60)  AS  BEGIN      SELECT dbo.UNIT.UNIT\_NAME, dbo.UNIT.UNIT\_ID      FROM dbo.SERVICE, dbo.UNIT      WHERE dbo.SERVICE.UNIT\_ID = dbo.UNIT.UNIT\_ID          AND SERVICE\_NAME = @Service\_name  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static UnitDTO GetUnitByServiceName(string ServiceName)  {  string query = string.Format("EXEC dbo.GetUnitByServiceName @Service\_name = N'{0}'", ServiceName);  DataTable dataTable = DataProvider.ExcuteQuery(query);  foreach (DataRow item in dataTable.Rows)  {  return new UnitDTO(item);  }  return null;  } |

* + - 1. *Lấy đơn vị dịch vụ qua tên dịch vụ*

|  |
| --- |
| -- Lấy danh sách đơn vị dịch vụ  CREATE OR ALTER PROC [dbo].[USP\_GetListServiceUnit]  AS  BEGIN      SELECT \*      FROM dbo.V\_SERVICE\_UNIT      WHERE STATUS = 1  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<ServiceDTO> GetListService()  {  string query = string.Format("EXEC dbo.USP\_GetListService");  DataTable dataTable = DataProvider.ExcuteQuery(query);  List<ServiceDTO> listService = new List<ServiceDTO>();  foreach (DataRow item in dataTable.Rows)  {  ServiceDTO temp = new ServiceDTO(item);  listService.Add(temp);  }  return listService;  } |

* + 1. ***Transaction***
       1. *Cập nhật thông tin sinh viên*

|  |
| --- |
| -- cập nhật sinh viên  CREATE OR ALTER PROC USP\_TRANSACTION\_UpdateStudent  (      @Ssn VARCHAR(12),      @Street NVARCHAR(50),      @Commune\_Name NVARCHAR(50),      @District\_Name NVARCHAR(50),      @Province\_Name NVARCHAR(50),      @College\_Name NVARCHAR(50),      @Faculty NVARCHAR(50),      @Major NVARCHAR(50),      @Phone\_Number\_1 VARCHAR(15),      @Phone\_Number\_2 VARCHAR(15),      @Email VARCHAR(40)  )  AS  BEGIN      DECLARE @User\_ID INT,              @Address\_ID INT      SELECT @User\_ID = dbo.[USER].USER\_ID      FROM dbo.[USER] WHERE dbo.[USER].SSN = @Ssn      SELECT @Address\_ID = dbo.[USER].ADDRESS\_ID      FROM dbo.[USER] WHERE dbo.[USER].SSN = @Ssn      BEGIN TRANSACTION          EXEC dbo.USP\_UpdateAddRess              @AddRess\_ID = @Address\_ID,              @Street = @Street,              @Commune\_Name = @Commune\_Name,              @District\_Name = @District\_Name,              @Province\_Name = @Province\_Name          EXEC dbo.USP\_UpdateStudentCollege              @User\_ID = @User\_ID,              @College\_Name = @College\_Name,              @Faculty = @Faculty,              @Major = @Major          UPDATE dbo.[USER]          SET dbo.[USER].PHONE\_NUMBER\_1 = NULL,              dbo.[USER].PHONE\_NUMBER\_2 = NULL,              dbo.[USER].EMAIL = NULL          WHERE dbo.[USER].USER\_ID = @User\_ID;          IF( @Phone\_Number\_1 IN                  (SELECT dbo.[USER].PHONE\_NUMBER\_1 FROM dbo.[USER])              OR              @Phone\_Number\_1 IN                  (SELECT dbo.[USER].PHONE\_NUMBER\_2 FROM dbo.[USER])          )          BEGIN              RAISERROR('Phone\_Number\_1 Is Exist',16,1)              ROLLBACK          END          ELSE IF (@Phone\_Number\_2 IN                      (SELECT dbo.[USER].PHONE\_NUMBER\_2 FROM dbo.[USER])                  OR                  @Phone\_Number\_2 IN                      (SELECT dbo.[USER].PHONE\_NUMBER\_1 FROM dbo.[USER])                  )          BEGIN              RAISERROR('Phone\_Number\_2 Is Exist',16,1)              ROLLBACK          END          ELSE IF(@Email IN (SELECT dbo.[USER].EMAIL FROM dbo.[USER]))          BEGIN              RAISERROR('Email Is Exist',16,1)              ROLLBACK          END          ELSE          BEGIN              UPDATE dbo.[USER]              SET dbo.[USER].PHONE\_NUMBER\_1 = @Phone\_Number\_1,                  dbo.[USER].PHONE\_NUMBER\_2 = @Phone\_Number\_2,                  dbo.[USER].EMAIL = @Email              WHERE dbo.[USER].USER\_ID = @User\_ID;              COMMIT          END  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool UpdateStudent(string Ssn, string Street, string Commune\_Name,  string District\_Name, string Province\_Name, string College\_Name,  string Faculty, string Major, string Phone\_Number\_1,  string Phone\_Number\_2, string Email)  {  string query = string.Format("EXEC dbo.USP\_TRANSACTION\_UpdateStudent " +  "@Ssn = '{0}'," +  "@Street = N'{1}'," +  "@Commune\_Name = N'{2}'," +  "@District\_Name = N'{3}'," +  "@Province\_Name = N'{4}'," +  "@College\_Name = N'{5}'," +  "@Faculty = N'{6}'," +  "@Major = N'{7}'," +  "@Phone\_Number\_1 = '{8}'," +  "@Phone\_Number\_2 = '{9}'," +  "@Email = '{10}'",  Ssn, Street, Commune\_Name, District\_Name,  Province\_Name, College\_Name, Faculty,  Major, Phone\_Number\_1, Phone\_Number\_2, Email);  int result = DataProvider.ExcuteNonQuery(query);  return result > 0;  }  } |

* + - 1. *Thêm sinh viên*

|  |
| --- |
| -- TRANSACTION -- THÊM 1 SINH VIÊN  CREATE OR ALTER PROC TRANS\_INSERT\_STUDENT  (      @Street NVARCHAR(50),      @Commune\_Name NVARCHAR(50),      @District\_Name NVARCHAR(50),      @Province\_Name NVARCHAR(50),      @Insurence\_ID VARCHAR(15),      @LAST\_NAME NVARCHAR(40),      @FIRST\_NAME NVARCHAR(20),      @DOB DATE,@GENDER NVARCHAR(5),      @SSN VARCHAR(12),      @PHONE\_NUMBER\_1 VARCHAR(15),      @PHONE\_NUMBER\_2 VARCHAR(15),      @EMAIL VARCHAR(40),      @IMAGE\_PATH VARCHAR(300),      @USER\_TYPE VARCHAR(10),      @STATUS BIT,      @STUDENT\_ID VARCHAR(15),      @COLLEGE\_NAME NVARCHAR(50),      @FACULTY NVARCHAR(50),      @MAJORS NVARCHAR(50)  )  AS  BEGIN      BEGIN TRANSACTION          CREATE TABLE TEMPT\_STUDENT          (              T\_SSN VARCHAR(12),              T\_INSURANCE\_ID VARCHAR(15),              T\_PHONENUMBER1 VARCHAR(15),              T\_PHONENUMBER2 VARCHAR(15),              T\_EMAIL VARCHAR(40)          )          INSERT INTO dbo.TEMPT\_STUDENT          (              T\_SSN,              T\_INSURANCE\_ID,              T\_PHONENUMBER1,              T\_PHONENUMBER2,              T\_EMAIL          )          SELECT dbo.[USER].SSN, dbo.STUDENT.INSURANCE\_ID, dbo.[USER].PHONE\_NUMBER\_1, dbo.[USER].PHONE\_NUMBER\_2, dbo.[USER].EMAIL          FROM dbo.[USER] INNER JOIN dbo.STUDENT              ON STUDENT.USER\_ID = [USER].USER\_ID          EXEC dbo.USP\_INSERT\_INSURANCE              @Insurence\_ID = @Insurence\_ID            EXEC dbo.USP\_INSERT\_ADDRESS              @Street = @Street,              @Commune\_Name = @Commune\_Name,              @District\_Name = @District\_Name,              @Province\_Name = @Province\_Name          EXEC dbo.USP\_INSERT\_USER\_STUDENT              @LAST\_NAME = @LAST\_NAME,            -- nvarchar(40)              @FIRST\_NAME = @FIRST\_NAME,          -- nvarchar(20)              @DOB = @DOB,                        -- date              @GENDER = @GENDER,                  -- nvarchar(5)              @SSN = @SSN,                        -- varchar(12)              @PHONE\_NUMBER\_1 = @PHONE\_NUMBER\_1,  -- varchar(15)              @PHONE\_NUMBER\_2 = @PHONE\_NUMBER\_2,  -- varchar(15)              @EMAIL = @EMAIL,                    -- varchar(40)              @IMAGE\_PATH = '',                   -- varchar(300)              @USER\_TYPE = @USER\_TYPE,            -- varchar(10)              @STATUS = @STATUS                   -- bit          EXEC dbo.USP\_INSERT\_STUDENT              @STUDENT\_ID = @STUDENT\_ID,          -- varchar(15)              @COLLEGE\_NAME = @COLLEGE\_NAME,      -- nvarchar(50)              @FACULTY = @FACULTY,                -- nvarchar(50)              @MAJORS = @MAJORS,                  -- nvarchar(50)              @INSURANCE\_ID = @Insurence\_ID,      -- varchar(15)              @STATUS\_REGISTRATION\_ROOM = 0       -- bit            IF( @SSN IN (SELECT dbo.TEMPT\_STUDENT.T\_SSN FROM dbo.TEMPT\_STUDENT))          BEGIN              RAISERROR('SSN Is exist',16,1)              ROLLBACK          END          ELSE          BEGIN              DROP TABLE dbo.TEMPT\_STUDENT              COMMIT          END  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static bool AddStudent(string Street, string Commune\_Name, string District\_Name,  string Province\_Name, string Insurance\_ID, string Last\_Name, string First\_Name,  DateTime DoB, string Gender, string Ssn, string Phone\_Number\_1,  string Phone\_Number\_2, string Email, string Image\_Path, string User\_Type,  string Status, string Student\_ID, string College\_Name,  string Faculty, string Majors)  {  string query = "EXEC dbo.TRANS\_INSERT\_STUDENT @Street , @Commune\_Name , " +  "@District\_Name , @Province\_Name, @Insurence\_ID , " +  "@LAST\_NAME , @FIRST\_NAME , @DOB , @GENDER , @SSN , " +  "@PHONE\_NUMBER\_1 , @PHONE\_NUMBER\_2, @EMAIL , @IMAGE\_PATH , " +  "@USER\_TYPE , @STATUS , @STUDENT\_ID, @COLLEGE\_NAME, " +  "@FACULTY , @MAJORS ";  var paras = new object[] {  Street,  Commune\_Name,  District\_Name,  Province\_Name,  Insurance\_ID,  Last\_Name,  First\_Name,  Convert.ToDateTime(DoB).ToString("yyyyMMdd"),  Gender,  Ssn,  Phone\_Number\_1,  Phone\_Number\_2,  Email,  Image\_Path ?? "",  User\_Type,  Status,  Student\_ID,  College\_Name,  Faculty,  Majors  };  int result = DataProvider.ExcuteNonQuery(query, paras);  return result > 0;  } |

* 1. **Functions**
     1. ***Table-valued functions***
        1. *Lấy danh sách sinh viên*

|  |
| --- |
| -- lấy danh sách sinh viên theo status  -- status = 0 : đã hết hợp đồng và rời khỏi KTX,  -- status = 1 : còn ở KTX,  -- status = 2 : tất cả  CREATE OR ALTER FUNCTION UFN\_GetListStudent (@STATUS INT)  RETURNS @Result TABLE (      [Id] BIGINT NOT NULL,      [Full name] NVARCHAR(51) NULL,      [Date of birth] DATE,      [Gender] VARCHAR(5),      [Ssn] VARCHAR(12),      [Phone number] VARCHAR(15),      [Email] VARCHAR(40),      [Student Id] VARCHAR(15) NOT NULL,      [College] NVARCHAR(100) NOT NULL  )  AS  BEGIN      INSERT @Result      SELECT          U.[USER\_ID] AS [Id],          CONCAT(U.LAST\_NAME, ' ', U.FIRST\_NAME) AS [Full name],          U.DOB AS [Date of birth],          U.GENDER AS [Gender],          U.SSN,          U.PHONE\_NUMBER\_1 AS [Phone number],          U.EMAIL AS [Email],          S.STUDENT\_ID AS [Student Id],          C.COLLEGE\_NAME AS [College]      FROM [dbo].[USER] AS U          INNER JOIN [dbo].[V\_ADDRESS] AS A ON A.ADDRESS\_ID = U.ADDRESS\_ID          INNER JOIN [dbo].[STUDENT] AS S ON S.[USER\_ID] = U.[USER\_ID]          INNER JOIN [dbo].[COLLEGE] AS C ON C.COLLEGE\_ID = S.COLLEGE\_ID      WHERE U.[STATUS] = @STATUS OR @STATUS = 2      RETURN  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy thông tin chi tiết hóa đơn*

|  |
| --- |
| -- Trả về giao diện Bill  CREATE OR ALTER FUNCTION UFN\_ReturnForViewBillDetail  (      @Sector\_Name VARCHAR(10),      @Room\_ID NVARCHAR(10),      @Month INT,      @Year INT  )  RETURNS @View\_BillDetail TABLE (      Service\_ID INT,      ServiceName NVARCHAR(50),      Unit\_Name NVARCHAR(50),      PricePerUnit DECIMAL(19,4),      Old\_Quantity INT,      New\_Quantity INT,      Total\_Cost DECIMAL(19,4),      Bill\_Detail\_ID INT,      Bill\_ID INT)  AS  BEGIN      DECLARE @Bill\_ID INT, @Service\_Name NVARCHAR(50)      SELECT @Bill\_ID = dbo.UFN\_GetBillIdBySectornameRoomidMonthYear(@Sector\_Name,@Room\_ID,@Month,@Year)      INSERT INTO @View\_BillDetail      SELECT          BD.SERVICE\_ID,          S.[SERVICE\_NAME],          BD.UNIT\_NAME,          S.PRICE\_PER\_UNIT,          BD.OLD\_QUANTITY,          BD.NEW\_QUANTITY,          BD.TOTAL\_COST,          BD.BILL\_DETAIL\_ID,          BD.BILL\_ID      FROM dbo.BILL\_DETAIL AS BD          INNER JOIN dbo.SERVICE AS S ON S.SERVICE\_ID = BD.SERVICE\_ID          INNER JOIN dbo.BILL AS B ON B.BILL\_ID = BD.BILL\_ID      WHERE BD.BILL\_ID = @Bill\_ID      RETURN  END  GO |

Triển khai trên C#:

|  |
| --- |
| public static List<BillDetailDTO> GetViewBillDetail(string Sector\_Name,  string Room\_ID, string Month, string Year)  {  string query = string.Format("SELECT \* FROM dbo.UFN\_ReturnForViewBillDetail(" +  "N'{0}', '{1}', {2}, {3})", Sector\_Name, Room\_ID, Month, Year);  DataTable dataTable = DataProvider.ExcuteQuery(query);  List<BillDetailDTO> ListService = new List<BillDetailDTO>();  foreach (DataRow item in dataTable.Rows)  {  BillDetailDTO temp = new BillDetailDTO(item);  ListService.Add(temp);  }  return ListService;  } |

* + 1. ***Scalar-valued functions***
       1. *Tạo mã MD5*

|  |
| --- |
| -- Generate mật khẩu:  CREATE OR ALTER FUNCTION UFN\_GenerateMD5 (@OldPass VARCHAR(32))      RETURNS VARCHAR(32)  AS  BEGIN      RETURN CONVERT(VARCHAR(32), HashBytes('MD5', @OldPass), 2)  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Tạo mật khẩu mặc định*

|  |
| --- |
| -- Hàm tạo mật khẩu mặt định  CREATE OR ALTER FUNCTION UFN\_NewPassword(@lastPassword VARCHAR(32), @preFix VARCHAR(4), @size INT)  RETURNS VARCHAR(10)  AS      BEGIN          IF (@lastPassword = '')              SET @lastPassword = @preFix + REPLICATE(0, @size - LEN(@preFix))          DECLARE @newPassword VARCHAR(32)          SET @lastPassword = LTRIM(RTRIM(@lastPassword))          SET @size = @size - LEN(@preFix)          SET @newPassword = @preFix + REPLICATE(0, @size - LEN(@preFix))          RETURN @newPassword      END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. ***Tìm kiếm gần đúng***

|  |
| --- |
| -- Tạo hàm tìm tên gần đúng  CREATE OR ALTER FUNCTION [dbo].[SearchLike] ( @strInput NVARCHAR(4000) )  RETURNS NVARCHAR(4000)  AS  BEGIN      IF @strInput IS NULL RETURN @strInput      IF @strInput = '' RETURN @strInput      DECLARE @RT NVARCHAR(4000)      DECLARE @SIGN\_CHARS NCHAR(136)      DECLARE @UNSIGN\_CHARS NCHAR (136)        SET @SIGN\_CHARS = N'ăâđêôơưàảãạáằẳẵặắầẩẫậấèẻẽẹéềểễệế ìỉĩịíòỏõọóồổỗộốờởỡợớùủũụúừửữựứỳỷỹỵý      ĂÂĐÊÔƠƯÀẢÃẠÁẰẲẴẶẮẦẨẪẬẤÈẺẼẸÉỀỂỄỆẾÌỈĨỊÍ ÒỎÕỌÓỒỔỖỘỐỜỞỠỢỚÙỦŨỤÚỪỬỮỰỨỲỶỸỴÝ' + NCHAR(272)+ NCHAR(208)      SET @UNSIGN\_CHARS = N'aadeoouaaaaaaaaaaaaaaaeeeeeeeeee iiiiiooooooooooooooouuuuuuuuuuyyyyy      AADEOOUAAAAAAAAAAAAAAAEEEEEEEEEEIIIII OOOOOOOOOOOOOOOUUUUUUUUUUYYYYYDD'      DECLARE @COUNTER int      DECLARE @COUNTER1 INT      SET @COUNTER = 1      WHILE (@COUNTER <=LEN(@strInput))          BEGIN              SET @COUNTER1 = 1              WHILE (@COUNTER1 <=LEN(@SIGN\_CHARS)+1)              BEGIN                  IF UNICODE(SUBSTRING(@SIGN\_CHARS, @COUNTER1,1)) = UNICODE(SUBSTRING(@strInput,@COUNTER ,1) )                      BEGIN                          IF @COUNTER=1                              SET @strInput = SUBSTRING(@UNSIGN\_CHARS, @COUNTER1,1) + SUBSTRING(@strInput, @COUNTER+1,LEN(@strInput)-1)                          ELSE                              SET @strInput = SUBSTRING(@strInput, 1, @COUNTER-1) +                                  SUBSTRING(@UNSIGN\_CHARS, @COUNTER1,1) + SUBSTRING(@strInput, @COUNTER+1,LEN(@strInput)- @COUNTER)                          BREAK                      END                  SET @COUNTER1 = @COUNTER1 +1              END              SET @COUNTER = @COUNTER +1          END        SET @strInput = replace(@strInput,' ','-')      RETURN @strInput  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Đếm số lượng sinh viên ở trong phòng cụ thể*

|  |
| --- |
| --Trả về số lượng sinh viên đã đăng lý vào phòng nào đó  CREATE OR ALTER FUNCTION UFN\_CountNumberOfStudentInRoom  (      @Sector\_ID VARCHAR(10),      @Room\_ID NVARCHAR(10)  )  RETURNS INT  AS BEGIN      DECLARE @Number INT      SELECT @Number = COUNT(dbo.ROOM\_REGISTRATION.SSN)      FROM dbo.ROOM\_REGISTRATION      WHERE ROOM\_REGISTRATION.SECTOR\_ID = @Sector\_ID          AND ROOM\_REGISTRATION.ROOM\_ID = @Room\_ID      RETURN @Number  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy mã tòa nhà dựa trên tên tòa nhà*

|  |
| --- |
| -- GetSectorIDBySectorName  CREATE OR ALTER FUNCTION UFN\_Get\_SectorID\_By\_SectorName      (@SECTOR\_NAME NVARCHAR(50))  RETURNS VARCHAR(10)  AS  BEGIN      DECLARE @SECTOR\_ID VARCHAR(10)      SELECT @SECTOR\_ID = dbo.SECTOR.SECTOR\_ID  FROM dbo.SECTOR WHERE dbo.SECTOR.SECTOR\_NAME = @SECTOR\_NAME      RETURN @SECTOR\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy mã hóa đơn của một phòng theo tháng cụ thể*

|  |
| --- |
| -- Lấy Bill ID -- GET\_BILL\_ID\_BY\_SECTORNAME\_ROOMID\_MONTH\_YEAR  CREATE OR ALTER FUNCTION UFN\_GetBillIdBySectornameRoomidMonthYear  (      @Sector\_Name VARCHAR(10),      @Room\_ID NVARCHAR(10),      @Month INT,      @Year INT  )  RETURNS INT  AS  BEGIN      DECLARE @Sector\_ID VARCHAR(10),              @Bill\_ID INT      SELECT @Sector\_ID = dbo.SECTOR.SECTOR\_ID      FROM dbo.SECTOR      WHERE dbo.SECTOR.SECTOR\_NAME = @Sector\_Name      SELECT @Bill\_ID = dbo.BILL.BILL\_ID      FROM dbo.BILL      WHERE dbo.BILL.Sector\_ID = @Sector\_ID          AND dbo.BILL.ROOM\_ID = @Room\_ID          AND dbo.BILL.MONTH = @Month          AND dbo.BILL.YEAR = @Year        RETURN @Bill\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy mã xã/phường qua tên xã/phường*

|  |
| --- |
| -- Lấy Commune\_ID từ Commune\_Name----------  CREATE OR ALTER FUNCTION UFN\_GetCommuneidByCommuneName  (      @Commune\_Name NVARCHAR(40),      @Distric\_ID VARCHAR(3)  )  RETURNS VARCHAR(5)  AS  BEGIN      DECLARE @COMMUNE\_ID VARCHAR(5)      SELECT @COMMUNE\_ID = COMMUNE\_ID      FROM dbo.COMMUNE      WHERE COMMUNE\_NAME = @Commune\_Name AND DISTRICT\_ID = @Distric\_ID      RETURN @COMMUNE\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy mã huyện/quận qua tên huyện/quận*

|  |
| --- |
| -- Hàm lấy mã huyện bằng tên huyện:  CREATE OR ALTER FUNCTION UFN\_GetDistrictIdByDictrictName(@DISTRICT\_NAME NVARCHAR(40),@PROVINCE\_ID VARCHAR(2))  RETURNS VARCHAR(3)  AS  BEGIN      DECLARE @DISTRICT\_ID VARCHAR(3)      SELECT @DISTRICT\_ID = DISTRICT\_ID      FROM dbo.DISTRICT      WHERE DISTRICT\_NAME = @DISTRICT\_NAME AND PROVINCE\_ID = @PROVINCE\_ID      RETURN @DISTRICT\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy mã tỉnh/TP qua tên tỉnh/TP*

|  |
| --- |
| -- Hàm lấy mã tỉnh bằng tên tỉnh:  CREATE OR ALTER FUNCTION UFN\_GetProvinceIdByProvinceName (@PROVINCE\_NAME NVARCHAR(20))      RETURNS VARCHAR(2)  AS  BEGIN      DECLARE @PROVINCE\_ID VARCHAR(2)      SELECT @PROVINCE\_ID = PROVINCE\_ID      FROM dbo.PROVINCE      WHERE PROVINCE\_NAME = @PROVINCE\_NAME      RETURN @PROVINCE\_ID  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Lấy chỉ số cũ của dịch vụ*

|  |
| --- |
| -- Lấy chỉ số mới của tháng trước đó của Bill  CREATE OR ALTER FUNCTION [dbo].[UFN\_GetOldQuantityForNewBill]  (      @Sector\_ID VARCHAR(10),      @Room\_ID NVARCHAR(10),      @Month INT, @Year INT,      @Service\_Name NVARCHAR(20)  )  RETURNS INT  AS  BEGIN      DECLARE @Name NVARCHAR(20),              @Quantity INT        SET @Name = @Service\_Name      SELECT @Quantity = dbo.BILL\_DETAIL.NEW\_QUANTITY      FROM dbo.BILL          INNER JOIN dbo.BILL\_DETAIL ON BILL\_DETAIL.BILL\_ID = BILL.BILL\_ID          INNER JOIN dbo.SERVICE ON SERVICE.SERVICE\_ID = BILL\_DETAIL.SERVICE\_ID      WHERE dbo.BILL.Sector\_ID = @Sector\_ID         AND dbo.BILL.ROOM\_ID = @Room\_ID         AND dbo.BILL.MONTH = @Month         AND dbo.BILL.YEAR = @Year         AND dbo.SERVICE.SERVICE\_NAME =  @Service\_Name        IF @Quantity IS NULL          SET @Quantity = 0        RETURN @Quantity  END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* 1. **Triggers**
     1. ***Người dùng***
        1. *Thay đổi mật khẩu mặc định*

|  |
| --- |
| -- Thay đổi mật khẩu mặc định  CREATE OR ALTER TRIGGER TRG\_DefaultPassword ON [dbo].[USER]  FOR INSERT  AS      BEGIN          DECLARE @lass\_USER\_ID BIGINT          DECLARE @newPassword VARCHAR(32)          SET @lass\_USER\_ID= ( SELECT TOP 1 USER\_ID                               FROM dbo.[USER] ORDER BY USER\_ID DESC)          SET @newPassword = dbo.UFN\_NewPassword('','dbms',32)          SET @newPassword = dbo.UFN\_GenerateMD5(@newPassword)          UPDATE dbo.[USER]          SET PASSWORD = @newPassword          WHERE USER\_ID = @lass\_USER\_ID AND USER\_TYPE = 'EMPLOYEE'      END  GO |

* + - 1. *Thay đổi mật khẩu mặc định cho sinh viên*

|  |
| --- |
| -- Thay đổi mật khẩu mặt định sinh viên  CREATE OR ALTER TRIGGER TRG\_DefaultPasswordStudent ON [dbo].[USER]  FOR INSERT  AS      BEGIN          DECLARE @lass\_USER\_ID BIGINT          DECLARE @newPassword VARCHAR(32)          SET @lass\_USER\_ID= ( SELECT TOP 1 USER\_ID                               FROM dbo.[USER] ORDER BY USER\_ID DESC)          SET @newPassword = dbo.UFN\_NewPassword('','mem',32)          SET @newPassword = dbo.UFN\_GenerateMD5(@newPassword)          UPDATE dbo.[USER]          SET PASSWORD = @newPassword WHERE USER\_ID = @lass\_USER\_ID          AND USER\_TYPE = 'STUDENT'      END  GO |

* + - 1. *Thay đổi mật khẩu mặc định cho sinh viên*

|  |
| --- |
| ALTER TRIGGER [dbo].[TRG\_LOCK\_USER\_STUDENT]  ON [dbo].[USER]  FOR UPDATE  AS  DECLARE @SSN VARCHAR(15), @USER\_ID BIGINT  SELECT @SSN = Inserted.SSN FROM Inserted  SELECT @USER\_ID = dbo.[USER].USER\_ID FROM dbo.[USER] WHERE dbo.[USER].SSN = @SSN  UPDATE dbo.STUDENT  SET dbo.STUDENT.STATUS\_REGISTRATION\_ROOM = 0  WHERE dbo.STUDENT.USER\_ID = @USER\_ID  UPDATE dbo.ROOM\_REGISTRATION  SET dbo.ROOM\_REGISTRATION.STATUS = 0  WHERE dbo.ROOM\_REGISTRATION.SSN = @SSN |

* + 1. ***Đăng ký phòng***
       1. *Khi tạo phiếu đăng ký phòng*

|  |
| --- |
| --TRG\_INSERT\_ROOM\_REGISTRATION  CREATE OR ALTER TRIGGER TRG\_INSERT\_ROOM\_REGISTRATION  ON dbo.ROOM\_REGISTRATION  FOR INSERT  AS      DECLARE @SSN VARCHAR(12),              @SECTOR\_ID VARCHAR(10),              @ROOM\_ID NVARCHAR(10),              @CAPACITY INT,              @CURRENT\_REGISTRATER INT      SELECT @SSN = Inserted.SSN FROM Inserted      SELECT @SECTOR\_ID =  Inserted.SECTOR\_ID FROM Inserted      SELECT @ROOM\_ID = Inserted.ROOM\_ID FROM Inserted      SELECT @CAPACITY = dbo.ROOM\_TYPE.CAPACITY      FROM dbo.ROOM INNER JOIN dbo.ROOM\_TYPE              ON ROOM\_TYPE.ROOM\_TYPE\_ID = ROOM.ROOM\_TYPE\_ID      WHERE dbo.ROOM.SECTOR\_ID = @SECTOR\_ID AND dbo.ROOM.ROOM\_ID = @ROOM\_ID      SELECT @CURRENT\_REGISTRATER = dbo.UFN\_CountNumberOfStudentInRoom(@SECTOR\_ID, @ROOM\_ID)      IF (@SSN IN (SELECT dbo.[USER].SSN                  FROM dbo.STUDENT INNER JOIN dbo.[USER]                      ON [USER].USER\_ID = STUDENT.USER\_ID                  WHERE dbo.STUDENT.STATUS\_REGISTRATION\_ROOM = 1))      BEGIN          RAISERROR(N'SSN Is Exist',16,1)          ROLLBACK      END      ELSE IF (@CURRENT\_REGISTRATER = @CAPACITY + 1)      BEGIN          RAISERROR(N'ROOM IS FULL',16,1)          ROLLBACK      END      ELSE      BEGIN          DECLARE @USER\_ID\_UPDATE VARCHAR(12)          SELECT @USER\_ID\_UPDATE = dbo.[USER].USER\_ID          FROM dbo.STUDENT INNER JOIN dbo.[USER]              ON [USER].USER\_ID = STUDENT.USER\_ID          WHERE dbo.[USER].SSN = @SSN          UPDATE dbo.STUDENT          SET STATUS\_REGISTRATION\_ROOM = 1          WHERE dbo.STUDENT.USER\_ID = @USER\_ID\_UPDATE      END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Khi hủy phiếu đăng ký phòng*

|  |
| --- |
| -- huy đăng kí phòng  CREATE OR ALTER TRIGGER TRG\_CANCEL\_ROOM\_REGISTRATION  ON dbo.ROOM\_REGISTRATION  FOR UPDATE  AS      DECLARE @User\_ID INT, @Tempt\_Ssn VARCHAR(12)      SELECT @Tempt\_Ssn = Inserted.SSN FROM Inserted      SELECT @User\_ID = dbo.[USER].USER\_ID  FROM dbo.[USER] WHERE dbo.[USER].SSN = @Tempt\_Ssn      UPDATE dbo.STUDENT      SET dbo.STUDENT.STATUS\_REGISTRATION\_ROOM =0      WHERE dbo.STUDENT.USER\_ID = @User\_ID  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + 1. ***Hóa đơn***
       1. *Kiểm tra hóa đơn*

|  |
| --- |
| -- TRG CHECKBILL  CREATE OR ALTER TRIGGER CheckBill  ON dbo.BILL  FOR INSERT  AS      DECLARE @Sector\_ID VARCHAR(10),              @Room\_ID NVARCHAR(10),              @Month INT,              @Year INT,              @Old\_Sector\_Name VARCHAR(10),              @Old\_Room\_ID NVARCHAR(10),              @Old\_Month INT,              @Old\_Year INT        SELECT @Sector\_ID = Inserted.Sector\_ID FROM Inserted      SELECT @Room\_ID = Inserted.ROOM\_ID FROM Inserted      SELECT @Month = Inserted.MONTH FROM Inserted      SELECT @Year = Inserted.YEAR FROM Inserted      IF (@Sector\_ID IN (SELECT dbo.TEMPT.T\_Sector\_ID FROM dbo.TEMPT)          AND @Room\_ID IN (SELECT dbo.TEMPT.T\_Room\_ID                          FROM dbo.TEMPT                          WHERE dbo.TEMPT.T\_Sector\_ID = @Sector\_ID)          AND @Month IN (SELECT dbo.TEMPT.T\_MONTH                          FROM dbo.TEMPT                          WHERE dbo.TEMPT.T\_Sector\_ID = @Sector\_ID                              AND dbo.TEMPT.T\_Room\_ID = @Room\_ID)          AND @Year IN (SELECT dbo.TEMPT.T\_YEAR                          FROM dbo.TEMPT                          WHERE dbo.TEMPT.T\_Sector\_ID = @Sector\_ID                              AND dbo.TEMPT.T\_Room\_ID = @Room\_ID                              AND dbo.TEMPT.T\_MONTH = @Month)          )      BEGIN          RAISERROR(N'Bill Is Exists',16,1)          ROLLBACK      END      ELSE          BEGIN                  INSERT INTO dbo.TEMPT                  (                      T\_Sector\_ID,                      T\_Room\_ID,                      T\_Month,                      T\_Year                  )                  VALUES                  (                      @Sector\_ID,                      @Room\_ID,                      @Month,                      @Year                  )          END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

* + - 1. *Thanh toán hóa đơn*

|  |
| --- |
| -- Cập nhật trạng thái tại bẳng hóa đơn bằng 1 (đã thanh toán)  -- nếu hóa đơn đó đã thanh toán thì thông báo "this bill has been paid"  CREATE OR ALTER TRIGGER TRG\_INSERT\_PAYMENT  ON dbo.PAYMENT  FOR INSERT  AS      DECLARE @Bill\_ID INT      SELECT @Bill\_ID = Inserted.BILL\_ID FROM Inserted      IF(@Bill\_ID IN (SELECT dbo.BILL.BILL\_ID                      FROM dbo.BILL                      WHERE dbo.BILL.STATUS = 1))      BEGIN          RAISERROR(N'This bill has been paid',16,1)          ROLLBACK      END      ELSE      BEGIN          UPDATE dbo.BILL          SET STATUS = 1 WHERE dbo.BILL.BILL\_ID = @Bill\_ID      END  GO |

Triển khai trên C#:

|  |
| --- |
|  |

1. **Phân quyền trong phần mềm**
   1. **Tạo user và roles**
      1. **Tạo role**

|  |
| --- |
| CREATE ROLE EMPLOYEE  CREATE ROLE ADMIN  CREATE ROLE STUDENT |

* + 1. **Tạo user**

|  |
| --- |
| CREATE OR ALTER PROC USP\_CREATE\_LOGIN\_USER  (  @Role\_Name VARCHAR(50),  @Login\_Name VARCHAR(50),  @Password\_Login VARCHAR(50)  )  AS  BEGIN  DECLARE @Login\_UserName VARCHAR(50),  @QueryLogin VARCHAR(100),  @QueryUser VARCHAR(100)  SET @Login\_UserName = @Login\_Name  SET @QueryLogin ='CREATE LOGIN ' + @Login\_UserName + ' WITH PASSWORD = ' + QUOTENAME(@Password\_Login, '''')  SET @QueryUser = CONCAT('CREATE USER ', @Login\_UserName, ' FOR LOGIN ', @Login\_UserName);  EXEC (@QueryLogin)  EXEC (@QueryUser)  EXEC sys.sp\_addrolemember @rolename = @Role\_Name,  @membername = @Login\_Name  END  GO |

* 1. **Phân quyền đăng nhập hệ thống**
     1. **Phân quyền admin**

|  |
| --- |
| --Grant Admin  --Login -- change password  GRANT EXECUTE ON dbo.USP\_Login TO ADMIN  GRANT EXECUTE ON dbo.USP\_ChangePassword TO ADMIN  --Room Registration  GRANT EXECUTE ON dbo.USP\_GetListSector TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListRoom TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListRoomBySectorID TO ADMIN  GRANT EXECUTE ON dbo.USP\_INSERT\_ROOMREGISTRATION TO ADMIN  --Bill  GRANT EXECUTE ON dbo.USP\_GetListServiceUnit TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListService TO ADMIN  GRANT EXECUTE ON dbo.GetUnitByServiceName TO ADMIN  GRANT EXECUTE ON dbo.USP\_InsetBill TO ADMIN  GRANT EXECUTE ON dbo.USP\_INSERT\_SERVICE\_BILL\_DETAIL TO ADMIN  GRANT SELECT ON dbo.UFN\_ReturnForViewBillDetail TO ADMIN  GRANT EXECUTE ON dbo.USP\_INSERT\_PAYMENT TO ADMIN  --Insert STUDENT  GRANT EXECUTE ON dbo.USP\_GetListProvince TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListDistrictByProvinceName TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListDistrictByProvinceName TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListCollege TO ADMIN  GRANT EXECUTE ON dbo.TRANS\_INSERT\_STUDENT TO ADMIN  GRANT EXECUTE ON USP\_CREATE\_LOGIN\_USER TO ADMIN  GRANT EXECUTE ON dbo.USP\_TRANSACTION\_UpdateStudent TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListCommuneByProvinceAndDistrict TO ADMIN  --Service  GRANT EXECUTE ON dbo.USP\_GetServicesInfo TO ADMIN  GRANT EXECUTE ON dbo.USP\_UpdateService TO ADMIN  GRANT EXECUTE ON dbo.USP\_InsertService TO ADMIN  GRANT EXECUTE ON dbo.USP\_UnableService TO ADMIN  --Room Type  GRANT EXECUTE ON dbo.USP\_GetListRoomView TO ADMIN  --Sector  --dbo.USP\_GetListSector  --Search  --Search STUDENT  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneral TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralALive TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralGoingOut TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByUserId TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByStudentId TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByFullName TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByGender TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralBySsn TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByCollegeName TO ADMIN  --Search ROOM-REGISTRATION  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistration TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistrationByStudentId TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistrationByStudentName TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistrationBySectorAndRoom TO ADMIN  --Search BILL  GRANT EXECUTE ON dbo.USP\_GetListBillView TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListBillViewBySectorAndRoom TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetListBillViewByDate TO ADMIN  ---Login  GRANT EXECUTE ON dbo.USP\_GetUserByUsername TO ADMIN  GRANT EXECUTE ON dbo.USP\_GetEmployeeById TO ADMIN  --View Info  GRANT EXECUTE ON dbo.USP\_GetEmployeeView TO ADMIN  --Update info  GRANT EXECUTE ON dbo.USP\_GetListUser TO ADMIN  GRANT EXECUTE ON dbo.USP\_EditEmployee TO ADMIN  ---LoadRoomRegistration  GRANT EXECUTE ON USP\_LoadRoomRegistrationByStudentID TO ADMIN  GRANT EXECUTE ON dbo.USP\_UpdateRoomRegistration TO ADMIN  --- GetStudentInfo  GRANT EXECUTE ON USP\_GetStudentInfo TO ADMIN  GRANT EXECUTE ON USP\_GetProvinceNameByProvinceID TO ADMIN  GRANT EXECUTE ON USP\_GetDistrictNameByDistrictID TO ADMIN  GRANT EXECUTE ON USP\_GetCommuneNameByCommuneID TO ADMIN  GRANT EXECUTE ON USP\_GetCollegeNameByCollegeID TO ADMIN  --- Lock User Student  GRANT EXECUTE ON USP\_LockUserStudent TO ADMIN  -- Add employee  GRANT EXECUTE ON dbo.USP\_AddEmployee TO ADMIN  --View List Employee  GRANT EXECUTE ON dbo.USP\_GetListEmployeeView TO ADMIN  --Update salary employee  GRANT EXECUTE ON dbo.USP\_UpdateSalary TO ADMIN  --Get admin by ID  GRANT EXECUTE ON dbo.USP\_GetAdminById TO ADMIN |

* + 1. **Phân quyền emloyee**

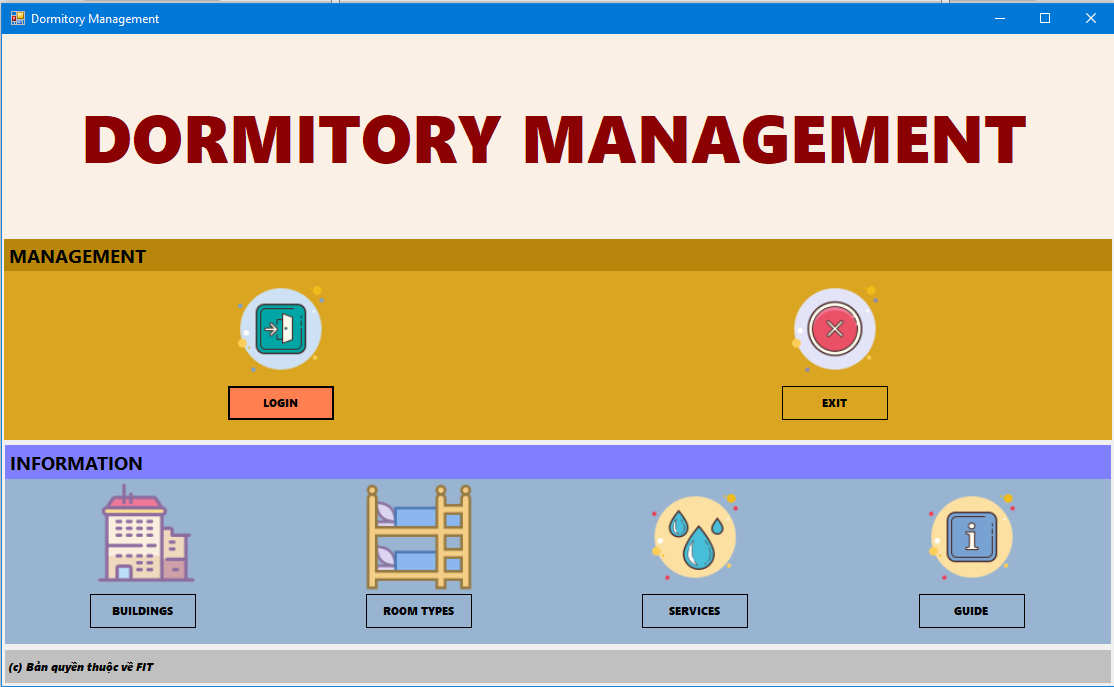
|  |
| --- |
| --Login -- change password  GRANT EXECUTE ON dbo.USP\_Login TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_ChangePassword TO EMPLOYEE  --Room Registration  GRANT EXECUTE ON dbo.USP\_GetListSector TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListRoom TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListRoomBySectorID TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_INSERT\_ROOMREGISTRATION TO EMPLOYEE  --Bill  GRANT EXECUTE ON dbo.USP\_GetListServiceUnit TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListService TO EMPLOYEE  GRANT EXECUTE ON dbo.GetUnitByServiceName TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_InsetBill TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_INSERT\_SERVICE\_BILL\_DETAIL TO EMPLOYEE  GRANT SELECT ON dbo.UFN\_ReturnForViewBillDetail TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_INSERT\_PAYMENT TO EMPLOYEE  --Insert STUDENT  GRANT EXECUTE ON dbo.USP\_GetListProvince TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListDistrictByProvinceName TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListDistrictByProvinceName TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListCollege TO EMPLOYEE  GRANT EXECUTE ON dbo.TRANS\_INSERT\_STUDENT TO EMPLOYEE  GRANT EXECUTE ON USP\_CREATE\_LOGIN\_USER TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_TRANSACTION\_UpdateStudent TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListCommuneByProvinceAndDistrict TO EMPLOYEE  --Service  GRANT EXECUTE ON dbo.USP\_GetServicesInfo TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_UpdateService TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_InsertService TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_UnableService TO EMPLOYEE  --Room Type  GRANT EXECUTE ON dbo.USP\_GetListRoomView TO EMPLOYEE  --Sector  --dbo.USP\_GetListSector  --Search  --Search STUDENT  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneral TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralALive TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralGoingOut TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByUserId TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByStudentId TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByFullName TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByGender TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralBySsn TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListStudentGeneralByCollegeName TO EMPLOYEE  --Search ROOM-REGISTRATION  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistration TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistrationByStudentId TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistrationByStudentName TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListRoomRegistrationBySectorAndRoom TO EMPLOYEE  --Search BILL  GRANT EXECUTE ON dbo.USP\_GetListBillView TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListBillViewBySectorAndRoom TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetListBillViewByDate TO EMPLOYEE  ---Login  GRANT EXECUTE ON dbo.USP\_GetUserByUsername TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_GetEmployeeById TO EMPLOYEE  --View Info  GRANT EXECUTE ON dbo.USP\_GetEmployeeView TO EMPLOYEE  --Update info  GRANT EXECUTE ON dbo.USP\_GetListUser TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_EditEmployee TO EMPLOYEE  ---LoadRoomRegistration  GRANT EXECUTE ON USP\_LoadRoomRegistrationByStudentID TO EMPLOYEE  GRANT EXECUTE ON dbo.USP\_UpdateRoomRegistration TO EMPLOYEE  --- GetStudentInfo  GRANT EXECUTE ON USP\_GetStudentInfo TO EMPLOYEE  GRANT EXECUTE ON USP\_GetProvinceNameByProvinceID TO EMPLOYEE  GRANT EXECUTE ON USP\_GetDistrictNameByDistrictID TO EMPLOYEE  GRANT EXECUTE ON USP\_GetCommuneNameByCommuneID TO EMPLOYEE  GRANT EXECUTE ON USP\_GetCollegeNameByCollegeID TO EMPLOYEE  --- Lock User Student  GRANT EXECUTE ON USP\_LockUserStudent TO EMPLOYEE |

* + 1. **Phân quyền student**

|  |
| --- |
| ---Grant STUDENT  -- Cchange Password  GRANT EXECUTE ON dbo.USP\_Login TO STUDENT  GRANT EXECUTE ON dbo.USP\_ChangePassword TO STUDENT  --View Info  GRANT EXECUTE ON dbo.USP\_GetListProvince TO STUDENT  GRANT EXECUTE ON dbo.USP\_GetListCollege TO STUDENT  GRANT EXECUTE ON dbo.USP\_GetListDistrictByProvinceName TO STUDENT  GRANT EXECUTE ON dbo.USP\_GetListCommuneByProvinceAndDistrict TO STUDENT  GRANT EXECUTE ON dbo.USP\_GetListStudentView TO STUDENT  --Update Info  GRANT EXECUTE ON dbo.USP\_TRANSACTION\_UpdateStudent TO STUDENT  --Sector  GRANT EXECUTE ON dbo.USP\_GetListSector TO STUDENT  --Room  GRANT EXECUTE ON dbo.USP\_GetListRoomView TO STUDENT  --Service  GRANT EXECUTE ON dbo.USP\_GetServicesInfo TO STUDENT  --  GRANT EXECUTE ON dbo.USP\_GetStudentById TO STUDENT |

1. **Hướng dẫn sử dụng**
   1. **Cài đặt chương trình**

* Bước 1: Restore Database
  + Chạy các file \*.sql theo thứ tự sau
    - 00\_DB\_Blank.sql
    - 01\_View.sql
    - 02\_Function.sql
    - 03\_Procedure.sql
    - 04\_ProcedureWithTrans.sql
    - 05\_Trigger.sql
    - 06\_RecentralizedRights.sql
    - 07\_InsertDataSample.sql
* Bước 2: Khởi chạy project
  + Mở file DormitoryManagement.sln để khởi động project bằng Visual Studio 2019
  + Chạy phần mềm: các tài khoản đăng nhập:
  1. **Chi tiết** 
     1. ***Khởi động ứng dụng (Guest)***

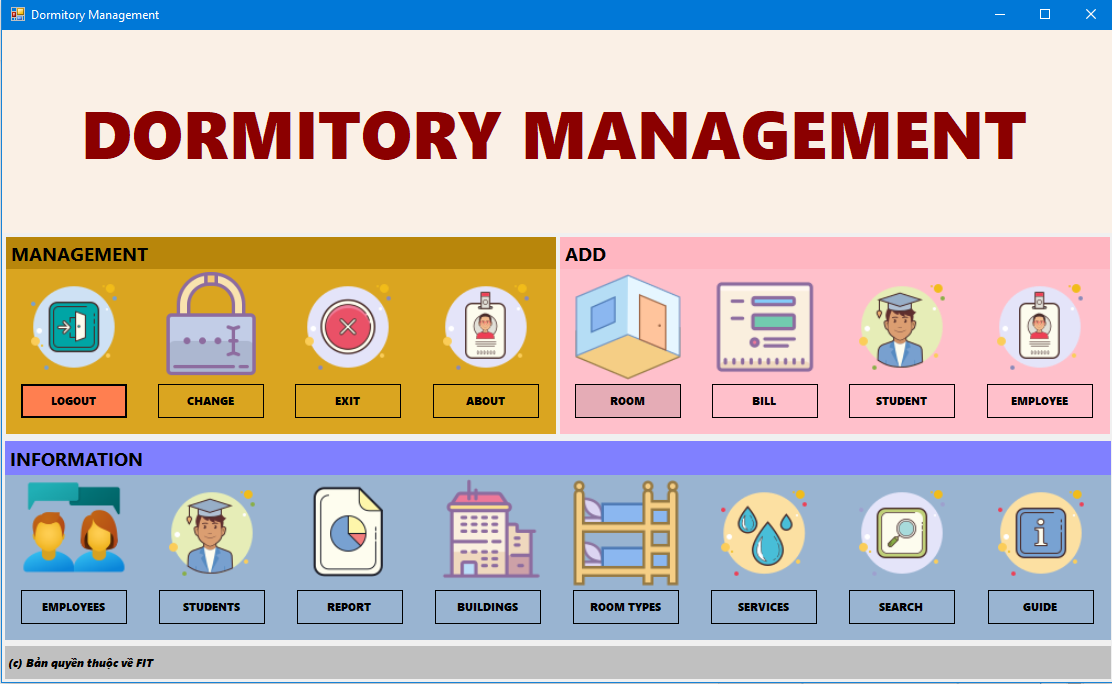


* + 1. ***Đăng nhập***

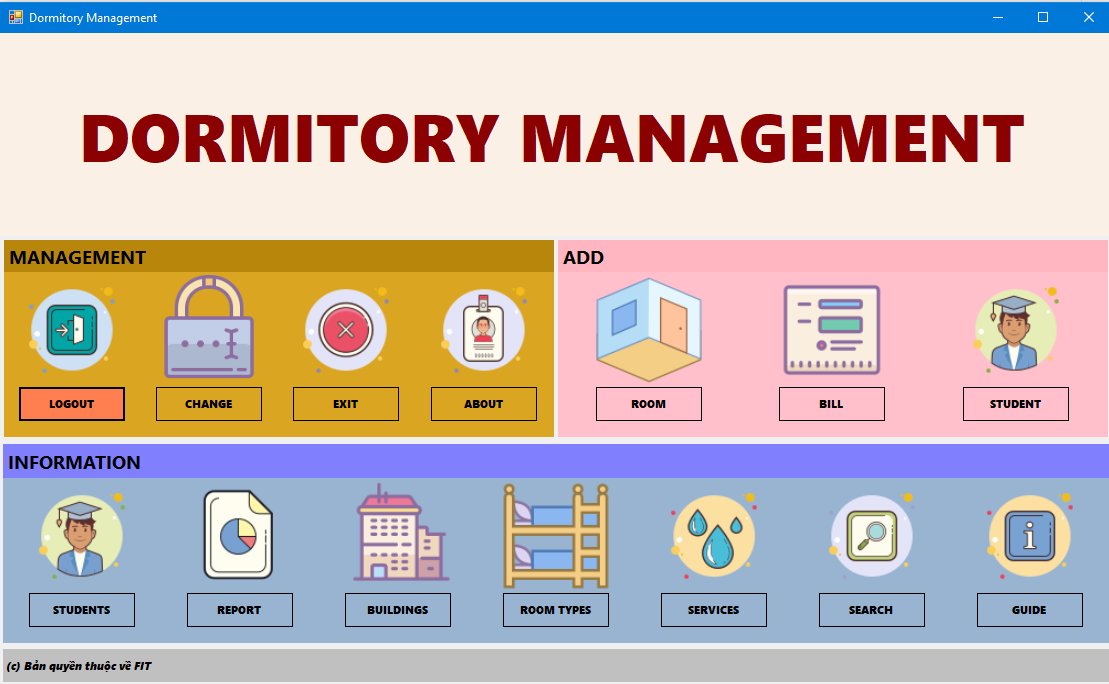


Nhập username và password cho tài khoản để đăng nhập. Tùy từng loại tài khoản (ứng với username, password đã nhập) mà sẽ hiển thị màn dashboard khác nhau.

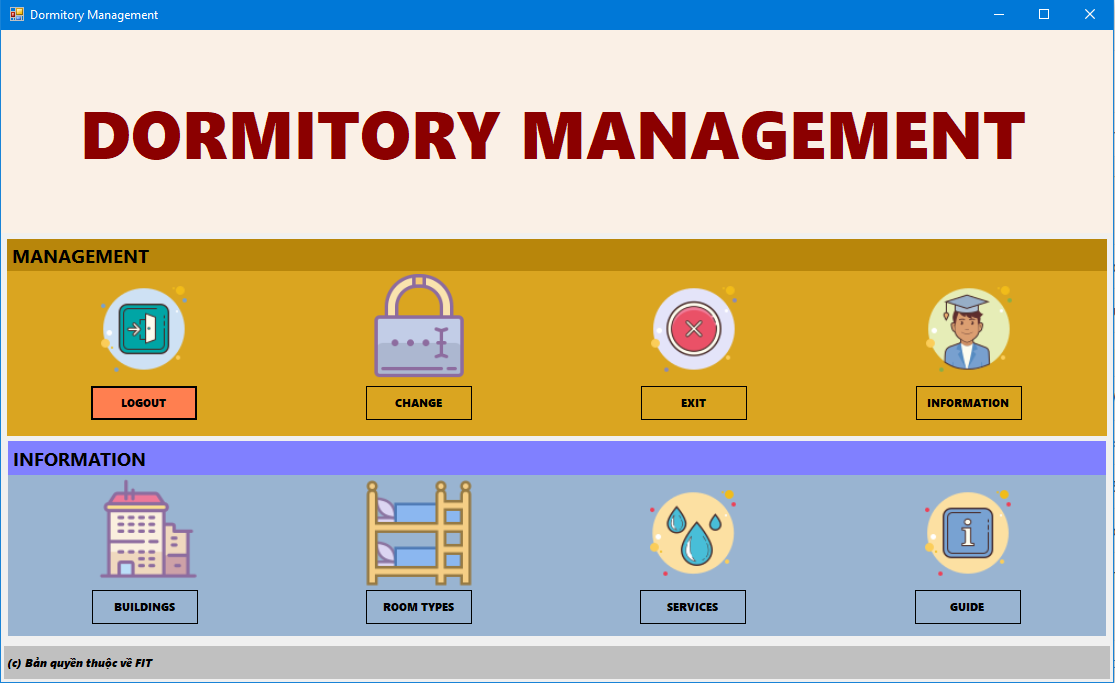
* + 1. ***Màn hình dashboard***
       1. ***Admin***



* + - 1. ***Employee***

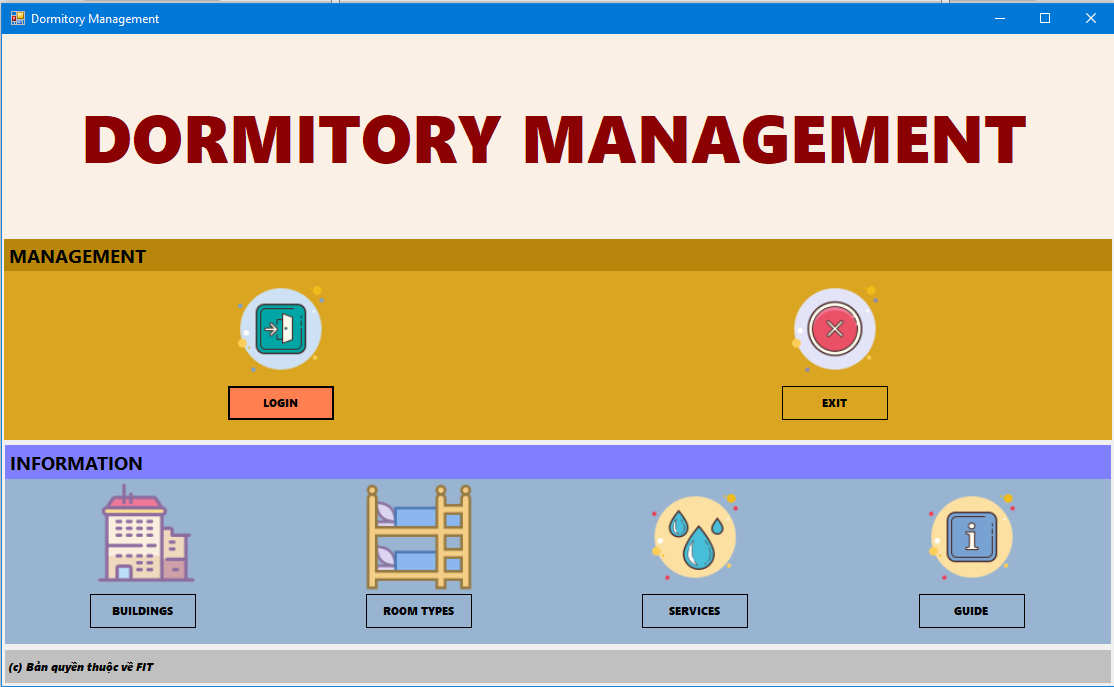


* + - 1. ***Student***

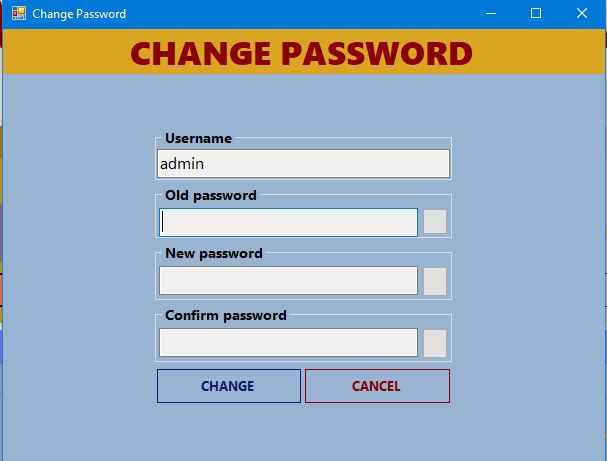


* + - 1. ***Guest***

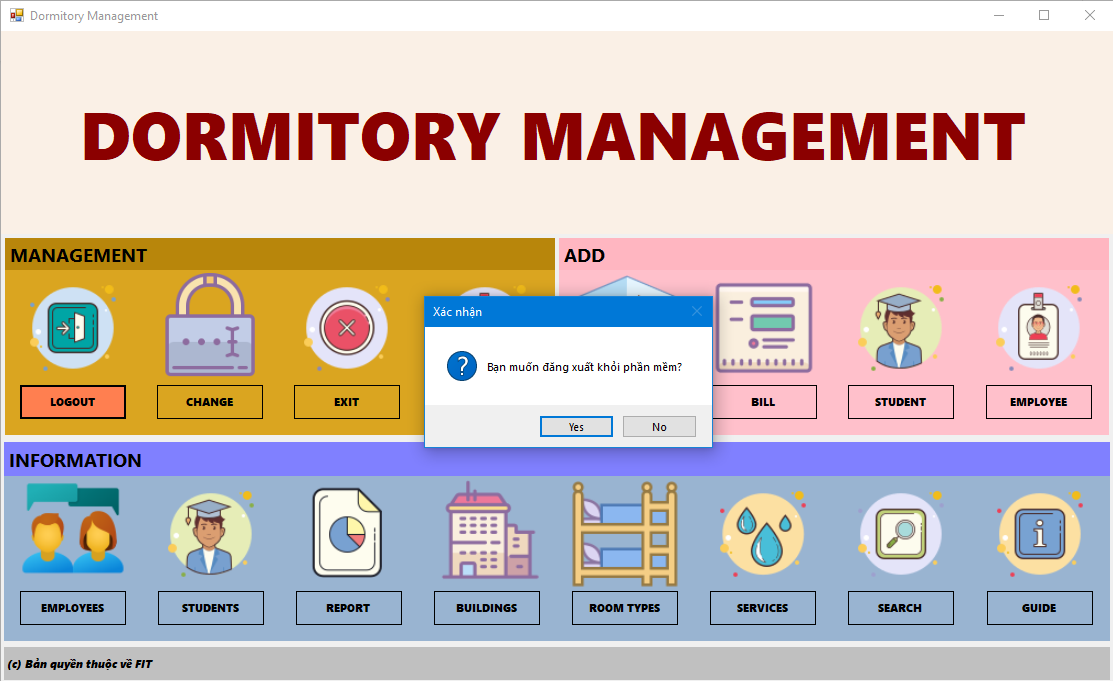
Không cần đăng nhập, màn hình như lúc khởi động ứng dụng.



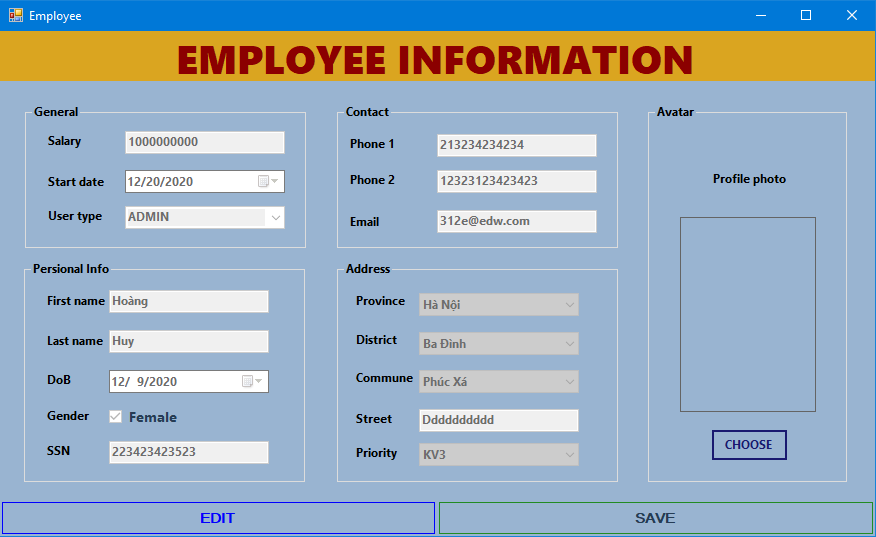
* + 1. ***Chức năng***
       1. *Đổi mật khẩu*



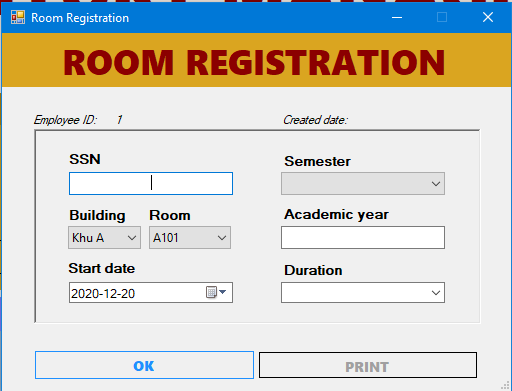
* + - 1. *Đăng xuất*



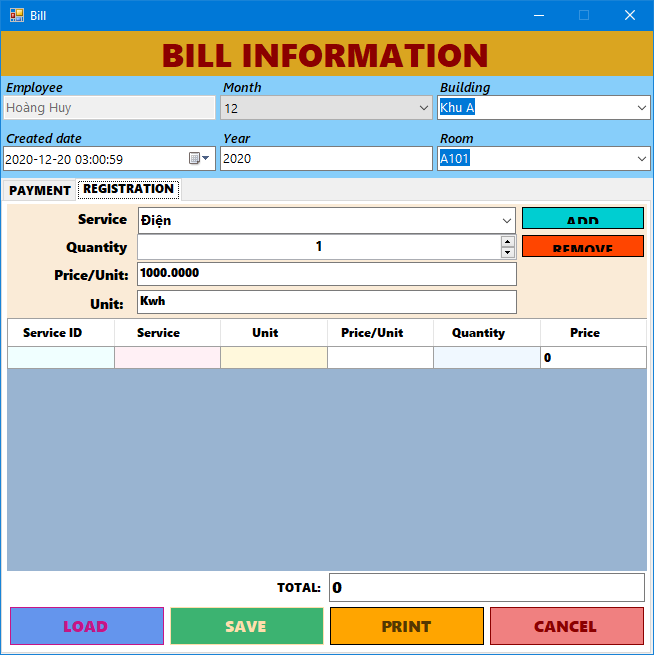
* + - 1. *Xem và sửa thông tin cá nhân*



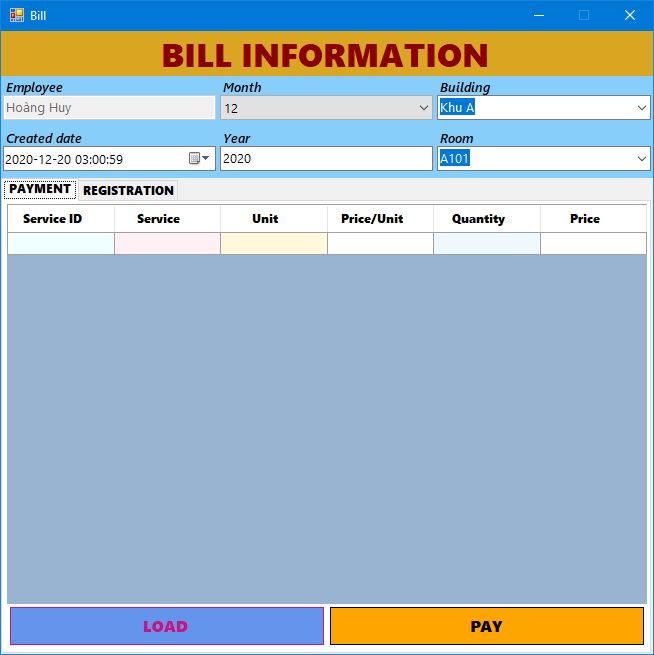
* + - 1. *Thêm phiếu đăng ký phòng*



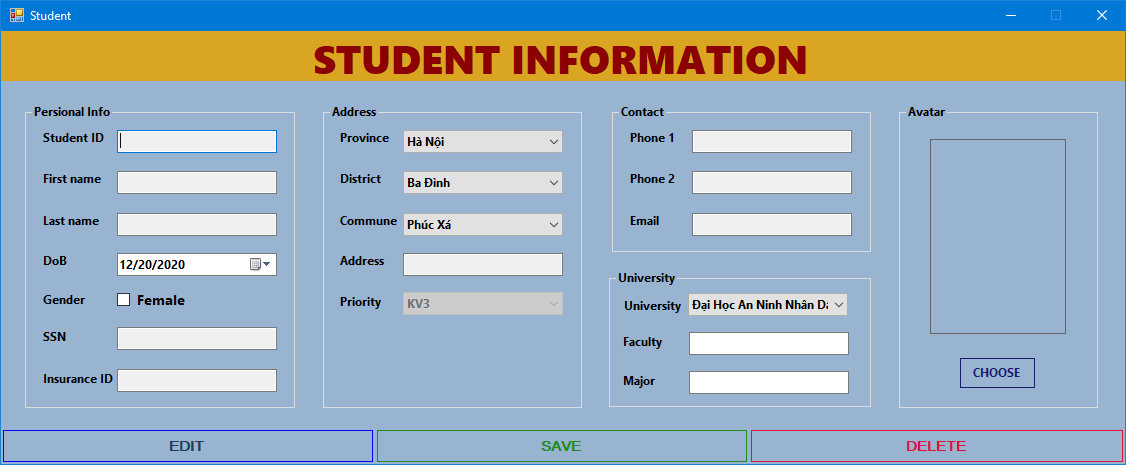
* + - 1. *Thêm và thanh toán hóa đơn*
         1. *Thêm hóa đơn*



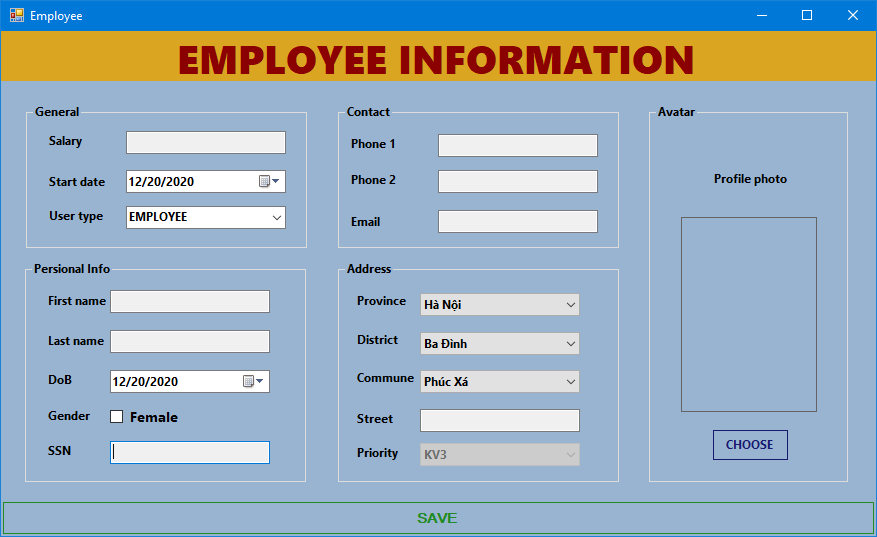
* + - * 1. *Thanh toán hóa đơn*



* + - 1. *Thêm tài khoản sinh viên*



* + - 1. *Thêm và cập nhật tài khoản nhân viên*



* + - 1. *Xem danh sách nhân viên*
      2. *Xem danh sách sinh viên*
      3. *Xem danh sách các khu nhà*
      4. *Xem danh sách các loại phòng*
      5. *Xem danh sách các dịch vụ*
      6. *Tìm kiếm*
         1. *Sinh viên*

*Theo trạng thái, mã người dùng, MSSV, họ tên, giới tính, CMND, trường Đại học*

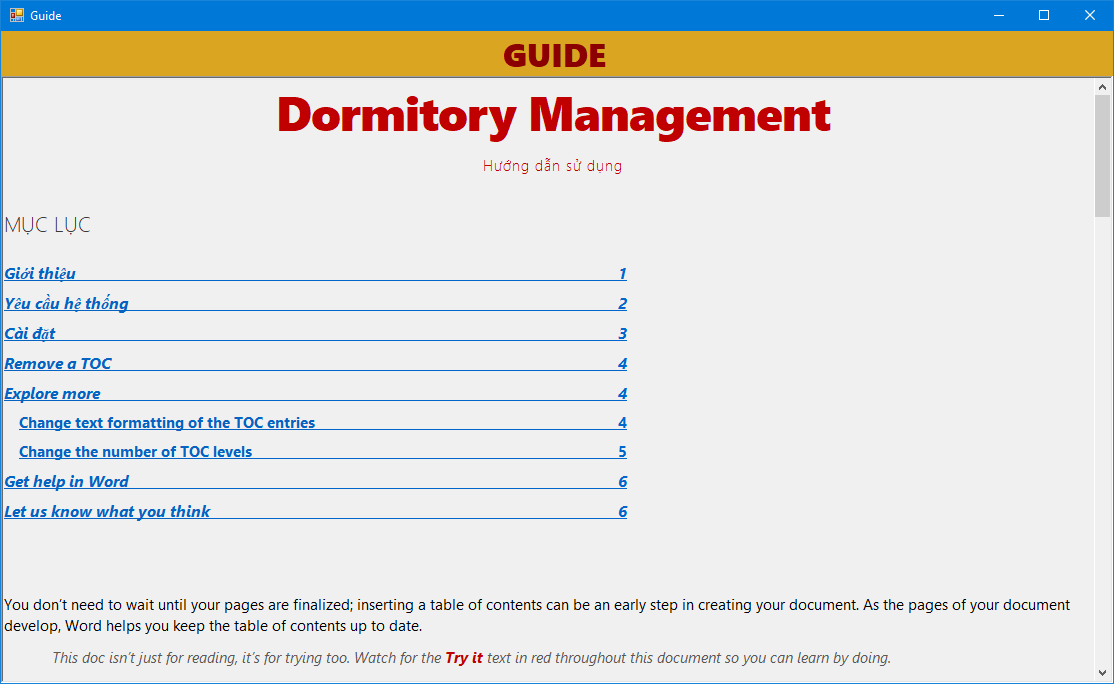
* + - * 1. *Phiếu đăng ký phòng*

*Theo khu, phòng, MSSV, Họ tên SV*

* + - * 1. *Hóa đơn*

*Theo khu, phòng, trạng thái, khoảng thời gian*

* + - 1. *Hướng dẫn sử dụng*



1. **Kết luận**
   1. **Các phần đã đạt được trong đồ án**
   2. **Các phần đặc sắc, đặc biệt trong đồ án**
   3. **Một số hạn chế**
   4. **Cam kết**

Chúng em xin cam đoan đồ án “Quản lý Kí Túc Xá” là thành quả của quá trình học tập và tìm hiểu dưới sự hướng dẫn của thầy Nguyễn Thành Sơn. Với lòng biết ơn sâu sắc và chân thành nhất, chúng em xin gửi đến thầy đã cùng với tri thức và tâm huyết của mình, truyền đạt vốn kiến thức quý báu cho chúng em trong suốt học kỳ này. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, bài báo cáo này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này. Mọi sự tham khảo đều có trích nguồn. Nếu có sự vi phạm nào chúng em xin hoàn toàn chịu trách nhiệm.