

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI**

Quản lý bãi xe

**GVHD: TS.NGUYỄN THÀNH SƠN**

**SVTH:**

**NGUYỄN KHẮC DƯƠNG 20110627**

**ĐẶNG TUẤN DUẪN 20110757**

**PHAN GIA HUY 19110369**

**LÊ HOÀNG HẢI ĐĂNG 20110265**

**TP. Hồ Chí Minh, ngày 17 tháng 11 năm**  **2022**

QUẢN LÝ BÃI XE

[**CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG** 1](#_Toc119601600)

[I. Mô tả quản lý bãi xe: 1](#_Toc119601601)

[**1.** **Tầm quan trọng của phần mềm quản lý bãi xe** 1](#_Toc119601602)

[**2.** **Mô tả dữ liệu của phần mềm** 1](#_Toc119601603)

[**3.** **Chức năng của phần mềm** 2](#_Toc119601604)

[II. Quy trình xử lý 3](#_Toc119601605)

[III. Mô hình Diagram 3](#_Toc119601606)

[**CHƯƠNG 2: CẤU HÌNH VÀ CÀI ĐẶT HỆ THỐNG** 4](#_Toc119601607)

[1. Tạo cơ sở dữ liệu và ràng buộc (constraint) 4](#_Toc119601608)

[2. Tạo các câu lệnh thêm, sửa, xóa trên Stored Procedure 6](#_Toc119601609)

[**2.1. Tạo tài khoản** 6](#_Toc119601610)

[**2.2. Update tài khoản** 6](#_Toc119601611)

[**2.3. Xóa tài khoản** 7](#_Toc119601612)

[**2.4. Tạo Bảng Giá** 7](#_Toc119601613)

[**2.5. Update bảng giá** 7](#_Toc119601614)

[**2.6. Xóa bảng giá** 7](#_Toc119601615)

[**2.7. Tạo Nhân Viên mới** 8](#_Toc119601616)

[**2.8. Update Nhân Viên** 8](#_Toc119601617)

[**2.9. Thêm Giữ xe** 9](#_Toc119601618)

[**2.10. Ra bãi xe** 9](#_Toc119601619)

[**2.11. Mất xe** 10](#_Toc119601620)

[**2.12. Hư xe** 10](#_Toc119601621)

[**2.13. Login** 10](#_Toc119601622)

[3. Function 11](#_Toc119601623)

[**3.1. Function Xem tài khoản** 11](#_Toc119601624)

[**3.2. Function Lấy mã nhân viên** 11](#_Toc119601625)

[**3.3. Function Tính tiền** 12](#_Toc119601626)

[4. Views 12](#_Toc119601627)

[**4.1. View Bảng giá** 12](#_Toc119601628)

[**4.2. View Giữ Xe (Home)** 12](#_Toc119601629)

[**4.3. View Xe Hư, Mất** 13](#_Toc119601630)

[5. Phân quyền 13](#_Toc119601631)

[6. Trigger 14](#_Toc119601632)

[**6.1. Trigger tăng chỗ trống** 14](#_Toc119601633)

[**6.2. Trigger giảm chỗ trống** 14](#_Toc119601634)

[**6.3. Trigger phân quyền** 15](#_Toc119601635)

[**6.4. Trigger xoá User** 15](#_Toc119601636)

[7. Transaction 16](#_Toc119601637)

[**CHƯƠNG 3: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM** 17](#_Toc119601638)

[1. Đăng nhập 17](#_Toc119601639)

[2. Form Trang chủ 19](#_Toc119601640)

[3. Form Tài khoản 25](#_Toc119601641)

[4. Form Bãi xe 28](#_Toc119601642)

[5. Form Nhân Viên 31](#_Toc119601643)

# **CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG**

## I. Mô tả quản lý bãi xe:

### **Tầm quan trọng của phần mềm quản lý bãi xe**

Bãi xe là thứ không thể thiếu của bất kì toà nhà, bệnh viện, trường học, công ty,… và để quản lý bãi xe không phải là điều đơn giản. Trước đây thì mọi người quản lý bãi xe bằng cách ghi nhớ, viết giấy gửi xe nhưng điều này không đảm bảo được hiệu quả vì với số lượng xe quá nhiều thì chúng ta sẽ có nhầm lẫn, nếu kẻ gian khai thác được điều này thì sẽ làm cho ta chịu nhiều thiệt thòi

Việc dung phần mề quản lý bãi xe sẽ khắc phục được những điểm đề cập trên, ngoài ra còn giúp cho người quản lý tiết kiệm chi phí, làm việc cũng sẽ nhanh chóng và hiệu quả hơn. Nhân viên cũng dễ dàng quản lý bãi xe hơn

1. **Mô tả dữ liệu của phần mềm**

- Nhân viên của bãi giữ xe gồm các thông tin : Id nhân viên, họ và tên, năm sinh, địa chỉ, giới tính, CMND, số điện thoại, email, chức vụ

- Tài khoản gồm các thông tin : Id, Username, PassWord, Id nhân viên, Quyền

- Khu vực gửi xe gồm các thông tin : Mã khu vực Tên khu vực, Số chỗ.

- Bảng giá gồm các thông tin : Loại xe, Giá cả, Mã khu vực.

- Giữ xe gồm các thông tin : Id giữ xe, Id nhân viên , ngày giữ, ngày trả, biển số xe, loại xe và Mã tình trạng.

- Tình trạng: Mã tình trạng, tình trạng.

1. **Chức năng của phần mềm**

- Kết nối cơ sở dữ liệu, lấy thông tin khách hàng, thẻ, xe, nhân viên,...

- Cho phép thêm, sửa, xóa, tìm kiếm thông tin data

- Chức năng chung của 2 đối tượng: đăng nhập, đăng xuất, xem thông tin,...

- Quản lý:

+ Quản lý đăng nhập vào hệ thông bằng tài khoản với quyền admin, lựa chọn từng chức năng phù hợp với yêu cầu: thêm, sửa, xóa, tìm kiếm, xem thông tin tài khoản nhân viên.

+ Khi thêm, sửa cần nhập đầy đủ thông tin về tài khoản sau đó dữ liệu sẽ được cập nhật vào cơ sở dữ liệu.

+ Nếu có sai sót thì hệ thống sẽ báo lỗi cho người dùng biết

- Nhân viên:

+ Nhân viên đăng nhập vào hệ thông bằng tài khoản với quyền nhân viên, lựa chọn từng chức năng phù hợp với yêu cầu: thêm, sửa, tìm kiếm, xe thông tin gửi xe, giá, chỗ trống trong bãi xe

+ Khi có khách gửi xe thì nhân viên sẽ lấy thông tin biển số và loại xe để hướng dẫn khách hàng đưa xe vào bãi

+ Khi thêm, sửa cần nhập đầy đủ thông tin về tài khoản sau đó dữ liệu sẽ được cập nhật vào cơ sở dữ liệu.

+ Nếu có sai sót thì hệ thống sẽ báo lỗi cho người dùng biết

+ Tính tiền cho khách khi ra bãi giữ xe

## II. Quy trình xử lý

Một bãi trông gửi xe có 2 cổng: Một cổng xe vào, một cổng xe ra. Bãi chia thành 4 khu dành cho 4 loại xe khác nhau: Xe máy, ô tô, xe tải và container. Khi khách đến gửi xe, người coi xe nhận dạng xe theo bảng phân loại, sau đó kiểm tra chỗ trống trong bãi. Nếu chỗ dành cho loại xe đó đã hết thì thông báo cho khách. Ngược lại thì ghi vé đưa cho khách và hưởng dẫn xe vào bãi, đồng thời ghi những thông tin trên vé vào hệ thống

Khi khách lấy xe, người coi xe kiểm tra vé xem vé là thật hay giả, đối chiếu vé với xe. Nếu vé giả hay không đúng xe thì không cho nhận xe. Ngược lại thì in phiếu thanh toán và thu tiền của khách, đồng thời cập nhật thông tin vào hệ thống

## III. Mô hình Diagram

Diagram

Description automatically generated

# **CHƯƠNG 2: CẤU HÌNH VÀ CÀI ĐẶT HỆ THỐNG**

## Tạo cơ sở dữ liệu và ràng buộc (constraint)

* Bảng Chức vụ

Go

CREATE TABLE dbo.ChucVu(

MaCV int PRIMARY KEY,

ChucVu nvarchar(50) Null

)

* Bảng Nhân viên

Go

CREATE TABLE dbo.NhanVien(

MaNV INT IDENTITY(1,1) PRIMARY KEY,

TenNV nvarchar(50) NULL,

NamSinh char(4) NULL,

DiaChi nvarchar(500) Null,

GioiTinh nvarchar(4) Null,

CMND nvarchar(15) Null,

Sdt nvarchar(12) Null,

Email nvarchar(100) Null,

ChucVu int references ChucVu(MaCV)

)

* Bảng Tài khoản

GO

CREATE TABLE dbo.TaiKhoan(

MaTk INT IDENTITY(1,1) PRIMARY KEY,

Username nvarchar(30) NULL,

MatKhau nvarchar(30) NULL,

MaNV int references NhanVien(MaNV) ON DELETE CASCADE ON UPDATE CASCADE,

Quyen int references ChucVu(MaCV) ON UPDATE CASCADE ON DELETE SET NULL

)

* Bảng khu vực

GO

CREATE TABLE dbo.KhuVuc(

MaKV int PRIMARY KEY,

TenKV nvarchar(30) NULL,

SoCho int NULL

)

* Bảng Tình trạng

GO

CREATE TABLE dbo.TinhTrang(

MaTT int PRIMARY KEY,

TinhTrang nvarchar(10) NULL

)

* Bảng Giá

GO

CREATE TABLE dbo.BangGia(

LoaiXe nvarchar(20) PRIMARY KEY,

Gia bigint NULL,

MaKV int references KhuVuc(MaKV)

)

* Bảng giữ xe

GO

CREATE TABLE dbo.GiuXe(

MaGX INT IDENTITY(1,1) PRIMARY KEY,

MaNV int references NhanVien(MaNV),

NgayVao datetime Null,

NgayRa datetime Null,

BienSoXe nvarchar(30) NULL,

LoaiXe int references KhuVuc(MaKV),

MaTT int references TinhTrang(MaTT),

CONSTRAINT c\_ngay CHECK(NgayRa>NgayVao),

)

## 2. Tạo các câu lệnh thêm, sửa, xóa trên Stored Procedure

### **2.1. Tạo tài khoản**

create proc sp\_addAccount(

@Username nvarchar(30),

@Password nvarchar(30),

@MaNV int,

@Quyen int)

as

BEGIN

INSERT INTO dbo.TaiKhoan(Username,MatKhau,MaNV,Quyen) values ( @Username,@Password,@MaNV,@Quyen)

END

### **2.2. Update tài khoản**

create proc sp\_updateAccount(

@Username nvarchar(30),

@Password nvarchar(30),

@Quyen int

)

as

begin

if (len(@Password) > 0)

UPDATE dbo.TaiKhoan set

MatKhau = @Password,

Quyen = @Quyen

WHERE Username = @Username

else

UPDATE dbo.TaiKhoan set

Quyen = @Quyen

WHERE Username = @Username

End

### **2.3. Xóa tài khoản**

create proc sp\_deleteAccount(

@Username nvarchar(30)

)

as

begin

delete from dbo.TaiKhoan where Username = @Username

END

### **2.4. Tạo Bảng Giá**

create proc sp\_addBangGia(

@LoaiXe nvarchar(20),

@Gia bigint,

@MaKV int)

as

BEGIN

INSERT INTO dbo.BangGia(LoaiXe,Gia,MaKV) values ( @LoaiXe, @Gia, @MaKV)

END

### **2.5. Update bảng giá**

create proc sp\_updateBangGia (

@LoaiXe nvarchar(20),

)

as

begin

UPDATE dbo.BangGia set

Gia = @Gia,

MaKV = @MaKV

WHERE LoaiXe = @LoaiXe

End

## 

### **2.6. Xóa bảng giá**

create proc sp\_deleteBangGia(

@LoaiXe nvarchar(20)

)

as

begin

delete from dbo.BangGia where LoaiXe = @LoaiXe

end

### **2.7. Tạo Nhân Viên mới**

create proc sp\_addNhanVien(

@MaNV int,

@TenNV NVARCHAR(50),

@NamSinh char(4),

@DiaChi nvarchar(500),

@GioiTinh nvarchar(4),

@CMND nvarchar(15),

@Sdt nvarchar(12),

@Email nvarchar(100),

@ChucVu int)

as

BEGIN

INSERT INTO dbo.NhanVien(MaNV,TenNV,NamSinh,DiaChi,

GioiTinh,CMND,Sdt,Email,ChucVu)

values(@MaNV,@TenNV,@NamSinh,@DiaChi,

@GioiTinh,@CMND,@Sdt,@Email,@ChucVu)

END

### **2.8. Update Nhân Viên**

create proc sp\_updateNhanVien(

@MaNV int,

@TenNV NVARCHAR(50),

@NamSinh char(4),

@DiaChi nvarchar(500),

@GioiTinh nvarchar(4),

@CMND nvarchar(15),

@Sdt nvarchar(12),

@Email nvarchar(100),

@ChucVu int)

as

BEGIN

UPDATE dbo.NhanVien SET

TenNV=@TenNV,

NamSinh=@NamSinh,

DiaChi=@DiaChi,

GioiTinh=@GioiTinh,

CMND=@CMND,

Sdt=@Sdt,

Email=@Email,

ChucVu=@ChucVu

WHERE MaNV=@MaNV

END

### **2.9. Thêm Giữ xe**

go

create proc sp\_addGiuXe

@MaNV int, @BienSoXe nvarchar(30), @LoaiXe int

as

BEGIN

INSERT INTO dbo.GiuXe(MaNV,NgayVao,NgayRa,BienSoXe,LoaiXe,MaTT) values (@MaNV, getdate(), null, @BienSoXe, @LoaiXe, 1)

END

### **2.10. Ra bãi xe**

go

create proc sp\_updateRaBaiXe

@MaGX int

as

BEGIN

update GiuXe

set MaTT = 3,

NgayRa=getdate()

where MaGX = @MaGX

END

### **2.11. Mất xe**

go

create proc sp\_updateMatXe

@MaGX int

as

BEGIN

update GiuXe

set MaTT = 2

where MaGX = @MaGX

END

### **2.12. Hư xe**

go

create proc sp\_updateHuXe

@MaGX int

as

BEGIN

update GiuXe

set MaTT = 4

where MaGX = @MaGX

END

### **2.13. Login**

GO

CREATE PROCEDURE sp\_login

@Username nvarchar(20),

@Password nvarchar(20),

@ChkQuyen int

as

begin

if exists (select \* from TaiKhoan where Username = @Username and MatKhau = @Password and @ChkQuyen = Quyen and Quyen=1)

select 1 as code

else if exists (select \* from TaiKhoan where Username = @Username and MatKhau = @Password and @ChkQuyen = Quyen and Quyen=2)

select 0 as code

else if exists(select \* from TaiKhoan where Username = @Username and MatKhau != @Password and @ChkQuyen != Quyen)

select 2 as code

else if exists(select \* from TaiKhoan where Username = @Username and MatKhau = @Password)

select 2 as code

else select 3 as code

end

## 3. Function

### **3.1. Function Xem tài khoản**

CREATE FUNCTION XemTaiKhoan (

@Username nvarchar(30)

)

RETURNS TABLE AS

return

(SELECT

TaiKhoan.Username as Username,

TaiKhoan.MaNV as MaNV,

NhanVien.TenNV as TenNV,

NhanVien.NamSinh as NamSinh,

NhanVien.DiaChi as DiaChi,

NhanVien.GioiTinh as GioiTinh,

NhanVien.CMND as CMND,

NhanVien.Sdt as Sdt,

NhanVien.Email as Email,

ChucVu.ChucVu as ChucVu

FROM dbo.TaiKhoan Inner join dbo.NhanVien

on dbo.TaiKhoan.MaNV = dbo.NhanVien.MaNV Inner join dbo.ChucVu

on dbo.TaiKhoan.Quyen = dbo.ChucVu.MaCV

WHERE @Username = '' or dbo.TaiKhoan.Username = @Username)

### **3.2. Function Lấy mã nhân viên**

CREATE FUNCTION getMaNhanVien (@Username nvarchar(30))

RETURNS int AS

BEGIN

DECLARE @maNV int;

SELECT @maNV = NhanVien.MaNV from NhanVien join TaiKhoan on NhanVien.MaNV = TaiKhoan.MaNV where TaiKhoan.Username = @Username;

RETURN @maNV;

END

### **3.3. Function Tính tiền**

Go

CREATE FUNCTION TinhTien (@MaGiuXe INT)

RETURNS bigint AS

BEGIN

DECLARE @tien bigint;

SELECT @tien= (datediff(DD,NgayVao,getdate())+1)\*BangGia.Gia

FROM GiuXe join BangGia On BangGia.MaKV = GiuXe.LoaiXe

WHERE MaGX=@MaGiuXe;

RETURN @tien;

END

## 4. Views

### **4.1. View Bảng giá**

go

create view BangGia\_info AS

SELECT LoaiXe, Gia, kv.TenKV , bg.MaKV

FROM BangGia bg join KhuVuc kv

on bg.MaKV = kv.MaKV

### **4.2. View Giữ Xe (Home)**

go

create view Home\_View AS

SELECT MaGX, NgayVao, BienSoXe, BangGia.LoaiXe, TinhTrang.TinhTrang

FROM dbo.GiuXe join BangGia

on GiuXe.LoaiXe = BangGia.MaKV

join TinhTrang

on GiuXe.MaTT = TinhTrang.MaTT

Where GiuXe.MaTT = 1

### **4.3. View Xe Hư, Mất**

go

create view XeHuMat\_View AS

SELECT MaGX, NgayVao, BienSoXe, BangGia.LoaiXe, TinhTrang.TinhTrang

FROM GiuXe join BangGia

on GiuXe.LoaiXe = BangGia.MaKV

join TinhTrang

on GiuXe.MaTT = TinhTrang.MaTT

Where GiuXe.MaTT = 2 or GiuXe.MaTT = 4

## 5. Phân quyền

Dùng storeprocedure để tạo phân quyền cho User

go

Create procedure sp\_PhanQuyenUser

@login nvarchar(30),

@pass nvarchar(30)

as

declare @sql nvarchar(max)

set @sql = 'use QLBaiXe;' +

'create login ' + @login +

' with password = ''' + @pass + '''; ' +

'create user '+ + @login + ' from login ' + @login + ';'

+'Grant select,insert,update on GiuXe to '+@login+';'

+'Grant select on BangGia to '+@login+';'

+'Grant select on NhanVien to '+@login+';'

+'Grant exec on sp\_addGiuXe to '+ @login+';'

+'Grant exec on sp\_updateRaBaiXe to '+ @login+';'

+'Grant exec on sp\_updateMatXe to '+ @login+';'

+'Grant exec on sp\_updateHuXe to '+ @login+';'

+'Grant exec on TinhTien to '+ @login+';'

+'Grant select on Home\_View to '+ @login+';'

+'Grant select on BangGia\_View to '+ @login+';'

exec (@sql)

Dùng storeprocedure để Xoá User

go

Create procedure sp\_XoaUser

@login nvarchar(30)

as

declare @sql nvarchar(max)

set @sql = 'use QLBaiXe;' +

+'drop user ' + @login + '; '

+'drop login ' + @login + '; '

exec (@sql)

## 6. Trigger

### **6.1. Trigger tăng chỗ trống**

go

Create trigger trg\_GiuXe on GiuXe after insert as

begin

update KhuVuc

set SoCho = SoCho - 1

From KhuVuc join inserted i

On KhuVuc.MaKV = i.LoaiXe

End

### **6.2. Trigger giảm chỗ trống**

Go

Create trigger trg\_TraXe on GiuXe after update as

begin

update KhuVuc

set SoCho = SoCho + 1

From KhuVuc join inserted i

On KhuVuc.MaKV = i.LoaiXe

End

### **6.3. Trigger phân quyền**

Go

Create trigger trg\_PhanQuyen on TaiKhoan after insert as

declare @Username nvarchar(30), @MatKhau nvarchar(30), @Quyen int

select @Username = Username from inserted

select @MatKhau = MatKhau from inserted

select @Quyen = Quyen from inserted

begin

if @Quyen = 1

begin

exec sp\_PhanQuyenUser @Username,@MatKhau

end

End

### **6.4. Trigger xoá User**

Go

Create trigger trg\_XoaUser on TaiKhoan after delete, update as

declare @Username nvarchar(30), @Quyen int

select @Username = Username from inserted

select @Quyen = Quyen from inserted

begin

if @Quyen = 1

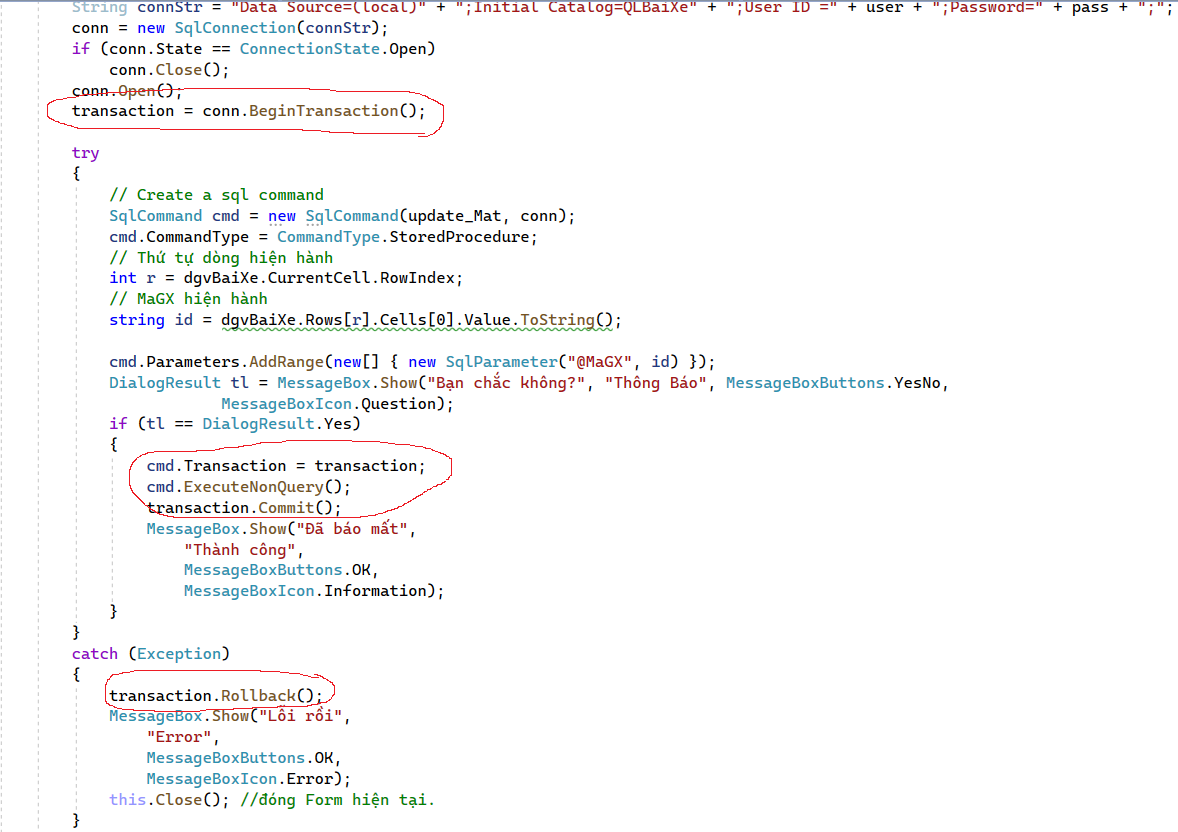
begin

exec sp\_XoaUser @Username

end

End

## 7. Transaction

Dùng transaction của SqlTransaction để gọi **BeginTransaction()**, **Commit** khi thực hiện đúng và **RollBack** khi thực hiện sai, có lỗi xảy ra

# **CHƯƠNG 3: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM**

## 1. Đăng nhập

Graphical user interface, application, website

Description automatically generated

* Chức năng Thoát: Khi người dùng nhấn nút Thoát sẽ hiện lên thông báo

Graphical user interface, application

Description automatically generated

Nếu người dùng ấn Yes thì ứng dụng sẽ đóng còn No sẽ huỷ hành động này

* Chức năng hiện mật khẩu: Khi ta nhấn vào checkbox **Hiển thị mật khẩu** hệ thống sẽ hiển thị thông tin đã nhập trong ô mật khẩu

Graphical user interface, application

Description automatically generated

* Khi người dùng đăng nhập và phần mềm sẽ kiểm tra username và password có tồn tại trong database hay không nếu có thì sẽ trở về trang chủ nếu không sẽ xuất hiện thông báo lỗi.

Graphical user interface, application, chat or text message

Description automatically generated

* Với người dùng là admin thì sẽ vào trang quản trị đã được quy định:

Graphical user interface

Description automatically generated with medium confidence

* Với người dùng là nhân viên thì sẽ vào trang home:

Graphical user interface, application

Description automatically generated

## 2. Form Trang chủ

- Đây là trang dành cho nhân viên, được dùng để quản lý bãi xe, nó sẽ hiện các thông tin về chỗ trống, giá, xe đang giữ và có thể thêmm, sửa tình trạng của xe đang giữ

Graphical user interface

Description automatically generated

- From tài khoản có những chức năng sau:

+ Thêm xe vào bãi giữ xe : Người dùng thêm thông tin bằng cách nhấp vào nút **Vào** khi đó bảng thông tin sẽ hiện ra cho người dùng nhập và ẩn các chức năng khác chỉ để lại **Lưu, Huỷ**

**Graphical user interface, table

Description automatically generated**

Sau khi nhập thông tin rồi nhấn nút **In** hệ thống sẽ hiện ra thông tin đã nhập để người dùng kiểm tra

Graphical user interface

Description automatically generated

Nếu ấn **Huỷ** thì sẽ đóng form và không thực hiện gì cả còn nếu ấn **In** sẽ hiện thông báo “Đã In” và thực hiện thêm dữ liệu vào hệ thống

Graphical user interface

Description automatically generated

+ Báo mất, hư hại khi có sự cố xảy ra: Người dùng chọn xe bị mất hoặc hư hại trên data gridviews và ấn **Báo mất** hoặc **Báo hư** **hại**

Graphical user interface, application

Description automatically generated

Khi người dùng chọn No thì sẽ không thực hiện gì cả, còn chọn Yes thì sẽ update tình trạng lên hệ thống

+ Ra bãi xe: Người dùng chọn xe bị đi ra trên data gridviews và ấn **Ra** hệ thống sẽ hiện ra thông tin của xe ra và tính tiền để người dùng kiểm tra

Graphical user interface

Description automatically generated

Nếu ấn **Huỷ** thì sẽ đóng form và không thực hiện gì cả còn nếu ấn **In** sẽ hiện thông báo “Đã In” và thực hiện update dữ liệu vào hệ thống

**Graphical user interface, application

Description automatically generated**

**+** Tìm kiếm: Người dùng chọn loại tìm kiếm, nhập thông tin tìm kiếm và nhấn “Enter” trên bàn phím, khi đó hệ thống sẽ tìm dữ liệu trên database và trả về trên datagridview

Graphical user interface

Description automatically generated

+ Load lại dữ liệu: Người dùng load lại dữ liệu bằng cách ấn nút **Reload** sau khi tìm kiếm

Graphical user interface, table

Description automatically generated

+ Đăng xuất: Người dùng thoát bằng cách ấn nút **Đăng xuất**

Graphical user interface, application

Description automatically generated

Khi ấn Yes thì sẽ trở về lại trang login và No để huỷ hành động này

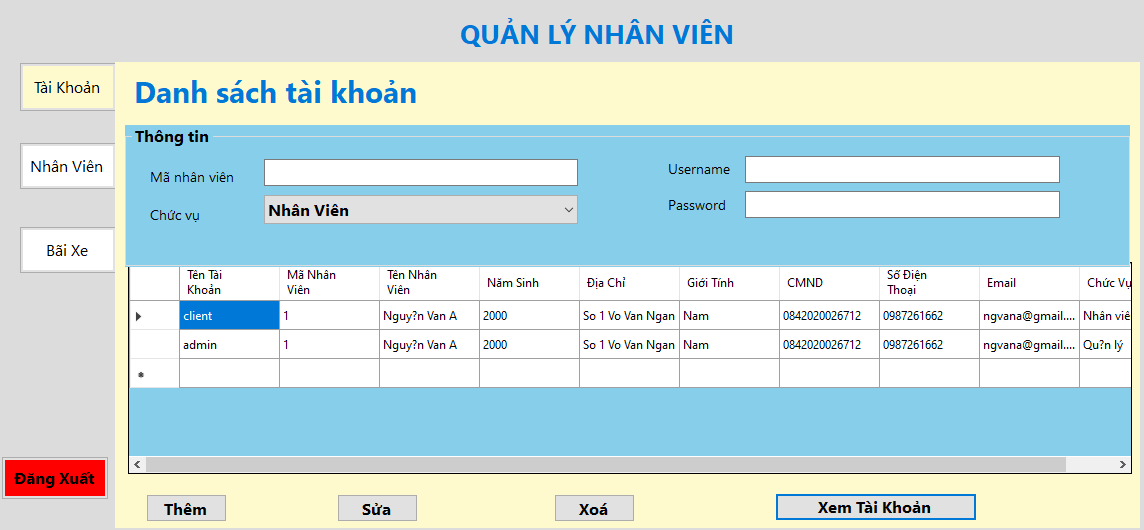
## 3. Form Tài khoản

- Đây là trang dành cho admin, được dùng để quản lý tài khoản như xem, thêm, xóa, sửa thông tin tài khoản của tất cả nhân viên.

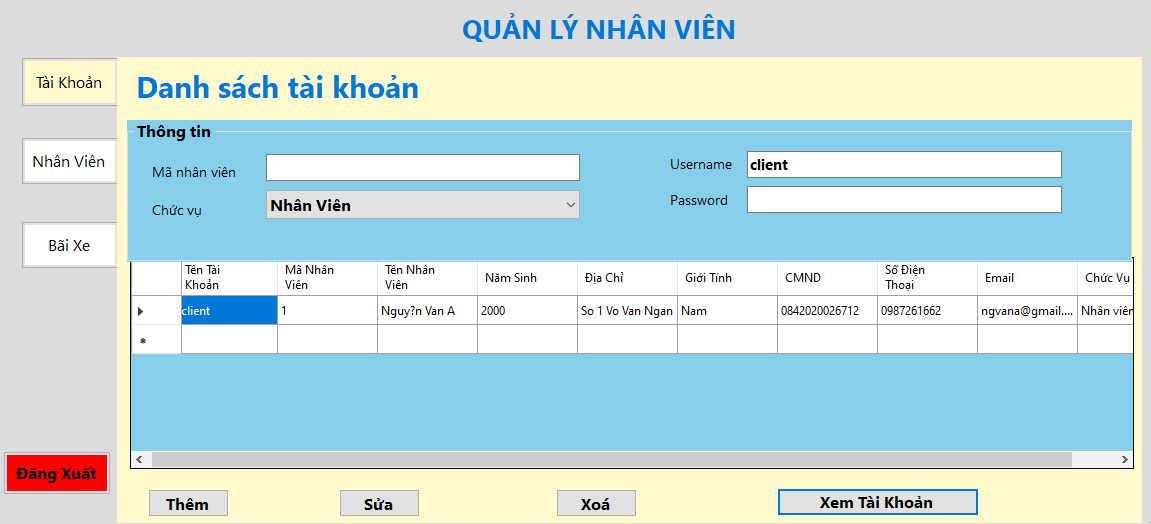


- From tài khoản có những chức năng sau:

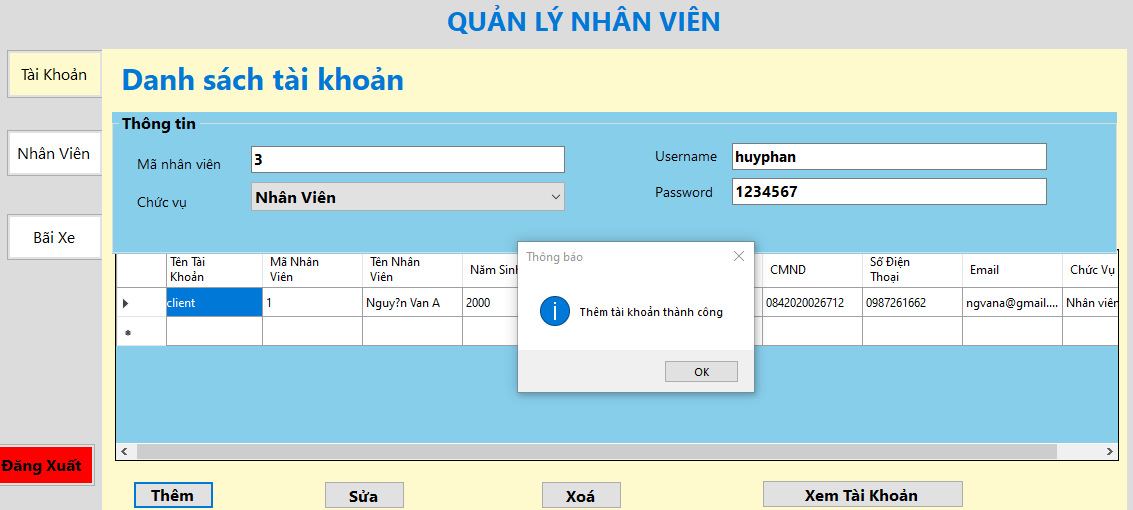
+ Xem tất cả thông tin tài khoản: Người dùng có thể xem tất cả thông tin tài khoản bằng cách nhấp vào nút **Xem Tài Khoản** với trường **Username** để trống

****

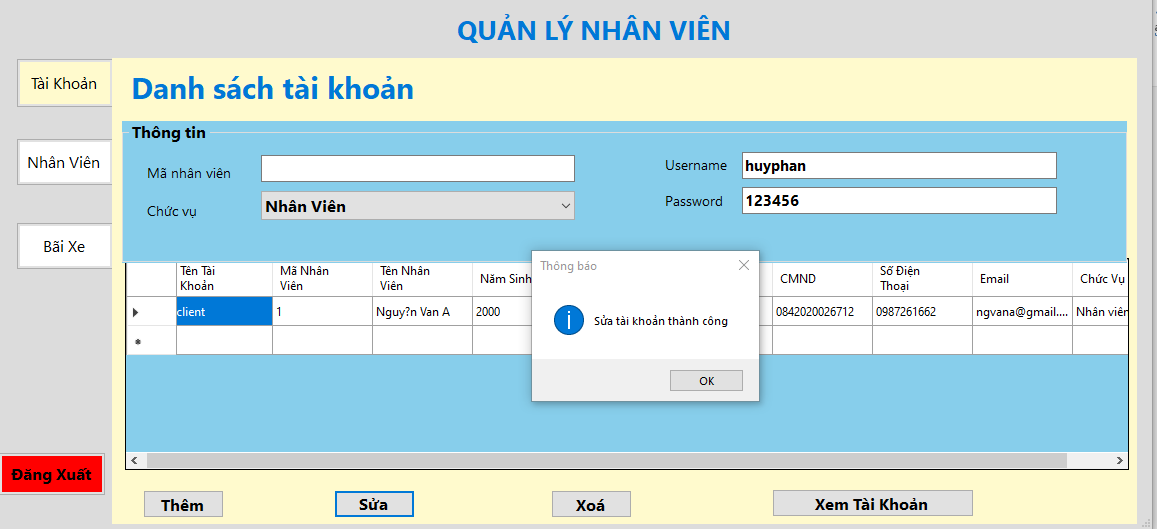
+ Xem thông tin của 1 tài khoản cụ thể: Người dùng có thể xem thông tin của 1 tài khoản cụ thể bằng cách điền tên tài khoản vào trường **Username** và ấn nút **Xem Tài Khoản**

****

+ Thêm tài khoản: Người dùng có thể thêm 1 tài khoản mới cho nhân viên bằng cách điền dầy đủ thông tin vào cái trường **Mã nhân viên, Username, Password,** và chọn chức vụ. Sau đó ấn nút **Thêm**

****

**+** Sửa thông tin tài khoản: Người dùng có thể sửa thông tin tài khoản bằng cách điền tên tài khoản muốn sửa vào trường **Username**, chọn chức vụ hoặc password mới cho tài khoản và ấn nút **Sửa.** Nếu trường **Password** để trống thì password của tài khoản sẽ không thay đổi.



+ Xóa tài khoản: Người dùng có thể xóa tài khoản bằng cách điền tên tài khoản cần xóa vào trường **Username** và ấn nút xóa.



## 4. Form Bãi xe

Đây là trang dành cho nhân viên quản lý bãi xe, được dùng để quản lý tài khoản bảng giá,tình trạng xe.

Graphical user interface

Description automatically generated

- Form bãi xe có những chức năng sau:

+ Xem thông tin bảng giá: Người dùng có thể xem tất cả thông tin bảng giá bằng cách chọn vào hàng muốn xem.

+ Thêm tài khoản: Nhân viên có thể thêm 1 bảng giá mới cho xe bằng cách điền dầy đủ thông tin vào các trường **Loại xe, Giá tiền và Khu vực**. Sau đó ấn nút **Thêm**

**Graphical user interface, application

Description automatically generated**

**+** Sửa thông tin bảng giá: Người dùng có thể sửa thông tin bảng giá bằng cách ấn nút **Sửa.** Sau đó chọn dòng muốn sửa , điền lại các thông tin cần sửa vào **Tiền Giữ Xe , Khu vực** rồi nhấn nút **Lưu** để **Lưu**

**Bấm nút Sửa**

Graphical user interface

Description automatically generated

**Sau khi sửa xong 🡪 Bấm Lưu**

**Graphical user interface, application

Description automatically generated**

+ Xóa bảng giá: Người dùng có thể xóa bảng giá bằng cách chọn vào dòng muốn xóa và ấn nút Xóa.

Graphical user interface

Description automatically generated

Chọn Có sẽ xoá dữ liệu trong database, Không thì sẽ huỷ hành động này

## 5. Form Nhân Viên

Đây là trang dành để quản lí nhân viên, được dùng để xem, thêm, xóa, sửa thông tin của nhân viên

Graphical user interface, application

Description automatically generated

- Form Nhân Viên có những chức năng sau:

+ Xem thông tin nhân viên: Người dùng có thể xem tất cả thông tin của nhân viên

+ Sửa thông tin nhân viên: Người dùng có thể sửa thông tin của nhân viên

+ Xóa nhân viên: người dùng có thể xóa 1 nhân viên bất kì

*Thêm nhân viên*: có thể thêm 1 nhân viên mới bằng cách nhấn nút **Thêm**, sau đó điền dầy đủ thông tin vào các trường **Tên Nhân Viên, Năm Sinh, Địa Chỉ, Giới Tính, CMND, Số Điện Thoại, Email, Chức Vụ** sau đó bấm nút **Lưu**

**Graphical user interface, table

Description automatically generated**

**Sau khi thêm xong 🡪 Bấm Lưu**

**Graphical user interface, table

Description automatically generated**

**Nhấn Quay Lại để thực hiện chức năng khác**

Sửa thông tin nhân viên : Người dùng có thể sửa thông tin nhân viên bằng cách ấn nút **Sửa.** Sau đó chọn dòng muốn sửa , điền lại các thông tin cần sửa vào các trường tương ứng,rồi nhấn nút **Lưu** để lưu lại kết quả

**Bấm nút Sửa**

Graphical user interface, application, table

Description automatically generated

**Sau khi sửa xong 🡪 Bấm Lưu**

**Graphical user interface, application, table

Description automatically generated**

*Xóa nhân viên :* Người dùng có thể xóa nhân viên bằng cách chọn vào dòng muốn xóa và ấn nút **Xóa**.

Graphical user interface, application

Description automatically generated

**Chọn Yes để xác nhận**