

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM  
KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**PHẦN MỀM HỌC TIẾNG ANH**

**GVHD: TS. NGUYỄN THÀNH SƠN**

**SVTH: NGUYỄN ĐỨC LINH    12110101**

**TRẦN THÀNH KHOA    13110082**

**TP. Hồ Chí Minh, ngày 9 tháng 11 năm 2016**

## Mục Lục

<b>CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG</b>	<b>3</b>
<b>I. Mô tả phần mềm:</b>	<b>3</b>
1. Tầm quan trọng của tiếng anh	3
2. Mô tả dữ liệu phần mềm.	3
<b>II. Quy trình xử lý</b>	<b>4</b>
<b>III. Mô hình Diagram.</b>	<b>5</b>
<b>CHƯƠNG 2: CẤU HÌNH VÀ CÀI ĐẶT HỆ THỐNG</b>	<b>6</b>
1. Tạo cơ sở dữ liệu và ràng buộc (Constraints).	6
2. Tạo các câu lệnh thêm, sửa, xóa trên Stored Procedure	9
2.1. Thêm, sửa, xóa User bằng quyền admin.	9
2.2. Thêm điểm	11
2.3. Thêm ngữ pháp.	11
2.4. Thêm từ vựng.	12
2.5. Sửa thông tin người dùng	12
2.6. Next, Back từ vựng, grammar	12
2.7. Kiểm tra login, thay đổi password	16
▪ Kiểm tra login	16
▪ Thay đổi password	16
2.8. Lấy dữ liệu ngữ pháp, từ vựng	16
▪ Lấy dữ liệu kiểm tra từ vựng	16
▪ Lấy dữ liệu kiểm tra ngữ pháp	17
▪ Lấy dữ liệu bài học từ vựng	17
▪ Lấy dữ liệu bài học ngữ pháp	17
3. Kết nối cơ sở dữ liệu	18
4. Phân quyền	21
4.1. Bảng cơ sở dữ liệu dùng để phân quyền:	21
4.2. Code phân quyền trên C#:	21
4.3. Phân quyền trên cơ sở dữ liệu:	22
5. Trigger, transaction	22
6. Function	24
7. View	27
8. Index	30

<b>CHƯƠNG 3: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM.....</b>	<b>31</b>
<b>1. Đăng nhập .....</b>	<b>31</b>
<b>2. Màn hình trang chủ: .....</b>	<b>31</b>
<b>3. Xem thông tin.....</b>	<b>32</b>
<b>4. Quản lý .....</b>	<b>33</b>
<b>5. Cấu hình. ....</b>	<b>35</b>

## CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG

### I. Mô tả phần mềm:

#### 1. Tầm quan trọng của tiếng anh

Học ngoại ngữ chính là quá trình biến “những âm thanh và con chữ vô nghĩa thành những biểu tượng có nghĩa” đối với chính mình. Thế nhưng những âm thanh và con chữ ấy lại quá phong phú với rất nhiều biến tấu tinh tế, hàm chứa cả một nền văn hóa nên lẽ dĩ nhiên chẳng có một phương pháp nào có thể thông hiểu một ngôn ngữ trong vòng 21 ngày. Điều này cũng nói lên rằng chỉ có ngữ cảnh và sự trải nghiệm qua một thời gian dài mới có thể giúp chúng ta nắm vững được một ngoại ngữ.

Học tiếng Anh cũng vậy. Vai trò của Tiếng Anh đối với những người làm việc trong lĩnh vực Công nghệ thông tin (CNTT) là điều không thể phủ nhận. Tuy nhiên việc học một ngoại ngữ, dù rất phổ thông như tiếng Anh, dường như vẫn luôn là một bức tường cao và dày rất khó vượt qua. Bên cạnh đó, nhu cầu học tập và làm việc trên máy vi tính ngày càng tăng. Vì vậy, một phần mềm hỗ trợ học Tiếng Anh là một nhu cầu thiết yếu.

Trong chương trình này, người dùng có thể rèn luyện kỹ năng nghe bằng tiếng Anh các vấn đề liên quan trong lĩnh vực công nghệ thông tin thông qua các bài đọc hiểu, các đoạn hội thoại do các giáo viên dày dặn kinh nghiệm của bộ môn Tiếng Anh và CNTT của trường đại học Oxford danh tiếng biên soạn.

#### 2. Mô tả dữ liệu phần mềm.

Để học tiếng anh hiệu quả, trước hết chúng ta nên có vốn từ vựng nhất định. Và việc học ngữ pháp thôi chắc chắn là không đủ để giúp học giỏi tiếng anh. Nhưng lại là nền tảng cho vốn kiến thức của bạn sau này. Bạn hãy coi ngữ pháp là một bài tập sinh lý mà các cầu thủ hay vận động viên vẫn làm: những bài tập đều đặn để chuẩn bị trước trận đấu quan trọng.

Vì tầm quan trọng của việc học từ vựng và ngữ pháp, cho nên phần mềm của chúng em sẽ tập trung đi sâu vào việc giúp người học có thể nâng cao vốn từ vựng cũng như nắm chắc nền tảng ngữ pháp tiếng anh. Phần mềm gồm các thành phần chính.

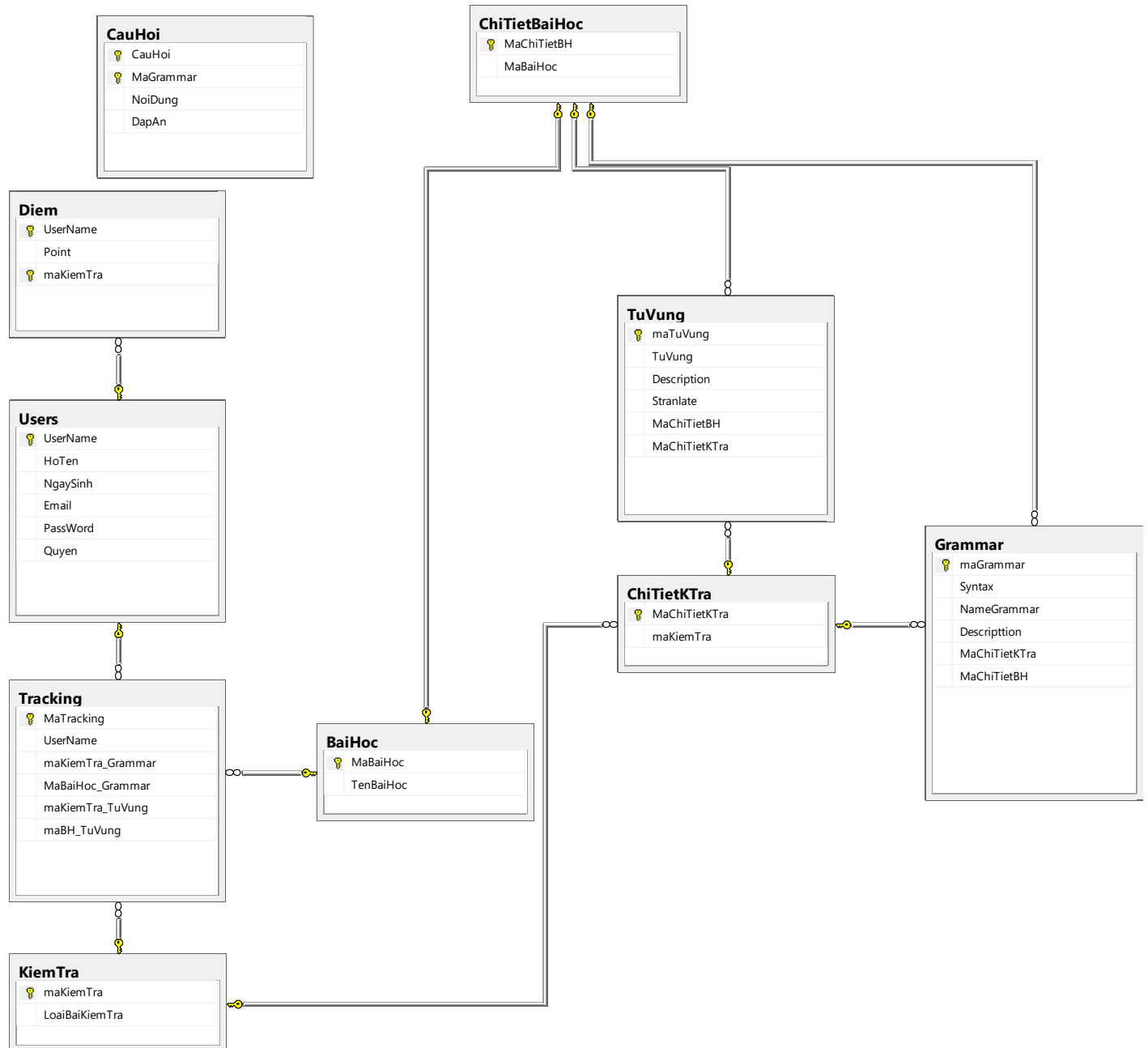
- **Quản lý thông tin người dùng và quyền đăng nhập hệ thống:** lưu thông tin người dùng khi người dùng tạo tài khoản trong phần mềm gồm: Mã User, Họ Tên, Ngày Sinh, Email, UserName, Password.
- **Quản lý Tracking:** lưu thông tin cần thiết của người dùng để biết người dùng đã học từ vựng và ngữ pháp tới phần nào.
- **Quản lý điểm:** lưu trữ thông tin điểm của người dùng bao gồm: Mã User, Mã Kiểm Tra, Điểm.

- **Quản lý bài học:** mỗi người dùng sẽ có một danh sách bài học cho riêng mình, mỗi bài học đều có phần từ vựng và ngữ pháp bao gồm:
  - **Quản lý bài học:** bao gồm các thông tin sau: Mã Bài Học, Tên Bài Học, Ngày Học, Mã User.
  - **Quản lý chi tiết bài học:** bao gồm các thông tin sau: Mã Chi Tiết Bài Học, Mã Bài Học.
- **Quản lý kiểm tra:** mỗi người dùng sẽ có một danh sách các bài kiểm tra cho riêng mình, mỗi bài kiểm tra đều có phần từ vựng và ngữ pháp bao gồm:
  - **Quản lý kiểm tra:** bao gồm các thông tin sau: Mã Kiểm Tra, Ngày Thi, Loại Bài Kiểm Tra, Mã User
  - **Quản lý chi tiết kiểm tra:** bao gồm các thông tin sau: Mã Chi Tiết Kiểm Tra, Mã Kiểm Tra.
- **Quản lý từ vựng:** lưu trữ các thông tin sau: Mã Từ Vựng, Từ Vựng, Description, Stralate, Mã Chi Tiết Bài Học, Mã Chi Tiết Kiểm Tra.
- **Quản lý grammar:** lưu các thông tin sau: Mã Grammar, Syntax, Name Grammar, Description, Mã Chi Tietes Kiểm Tra, Mã Chi Tiết Bài Học

## II. Quy trình xử lý

- Để sử dụng các chức năng của phần mềm, yêu cầu người dùng phải đăng nhập vào hệ thống. Admin có quyền cho người dùng vào hệ thống với quyền nào đó trên hệ thống. Việc thực hiện đăng nhập thông qua bảng **Đăng Nhập**.
- Khi người dùng học đến một số bài nhất định, Admin sẽ tạo ra đề kiểm tra để người dùng kiểm tra lại kiến thức thông qua bảng **Tạo Bài Kiểm Tra**.
- Khi người dùng học xong bài học, Admin phải tạo ra bài học tiếp theo để người dùng tiếp tục học trên hệ thống thông qua bảng **Tạo Bài Học**.
- Nếu một phần hay toàn bộ bài học không chính xác, người học có thể report lại lỗi để Admin kiểm tra và sửa lỗi thông qua bảng **Report**.
- Khi người dùng mới đăng ký thông tin để tạo account tại bảng **Đăng Ký**, thông tin đó sẽ được gửi về Admin để Admin xét duyệt.
- Người dùng có thể xem thông tin cá nhân và thông tin việc học của mình qua bảng **Thông tin User**.
- Khi người dùng muốn seach từ vựng, người dùng có thể seach thông qua bảng **Tra Từ Vựng**.

### III. Mô hình Diagram.



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

### 1. Tạo cơ sở dữ liệu và ràng buộc (Constraints).

Trong phần này áp dụng các ràng buộc như: unique, not null, ràng buộc khóa ngoại, ràng buộc khóa chính, đặt tên cho các ràng buộc.

```
CREATE TABLE [dbo].[BaiHoc](
    [MaBaiHoc] [int] NOT NULL,
    [TenBaiHoc] [nvarchar](30) NULL,
    [NgayHoc] [date] NULL,
    [maUser] [int] NOT NULL,
    CONSTRAINT [PK_BaiHoc] PRIMARY KEY
```

```
CREATE TABLE [dbo].[CauHoi](
    [CauHoi] [int] NOT NULL,
    [MaGrammar] [int] NOT NULL,
    [NoiDung] [nvarchar](500) NULL,
    [DapAn] [nvarchar](50) NULL,
    CONSTRAINT [PK_CauHoi] PRIMARY KEY
```

```
CREATE TABLE [dbo].[ChiTietBaiHoc](
    [MaChiTietBH] [int] NOT NULL,
    [MaBaiHoc] [int] NULL,
    CONSTRAINT [PK_ChiTietBaiHoc] PRIMARY KEY
```

```
CREATE TABLE [dbo].[ChiTietKTra](
    [MaChiTietKTra] [int] IDENTITY(1,1) NOT NULL,
    [maKiemTra] [int] NULL,
    CONSTRAINT [PK_ChiTietKTra] PRIMARY KEY
```

```
CREATE TABLE [dbo].[Diem](
    [UserName] [nvarchar](50) NOT NULL,
    [Point] [int] NULL,
    [maKiemTra] [int] NOT NULL,
    CONSTRAINT [PK_Diem_1] PRIMARY KEY
```

```
CREATE TABLE [dbo].[Grammar](
    [maGrammar] [int] IDENTITY(1,1) NOT NULL,
    [Syntax] [nvarchar](50) NULL,
    [NameGrammar] [nvarchar](50) NULL,
    [Description] [nvarchar](300) NULL,
    [MaChiTietKTra] [int] NULL,
    [MaChiTietBH] [int] NULL,
    CONSTRAINT [PK_Grammar] PRIMARY KEY
```

```
CREATE TABLE [dbo].[KiemTra](
    [maKiemTra] [int] IDENTITY(1,1) NOT NULL,
```

```
[LoaiBaiKiemTra] [nchar](10) NULL,  
CONSTRAINT [PK_KiemTra_1] PRIMARY KEY
```

```
CREATE TABLE [dbo].[Tracking](  
    [MaTracking] [int] IDENTITY(1,1) NOT NULL,  
    [UserName] [nvarchar](50) NULL,  
    [maKiemTra_Grammar] [int] NULL,  
    [MaBaiHoc_Grammar] [int] NULL,  
    [maKiemTra_TuVung] [int] NULL,  
    [maBH_TuVung] [int] NULL,  
    CONSTRAINT [PK_Tracking_1] PRIMARY KEY
```

```
CREATE TABLE [dbo].[TuVung](  
    [maTuVung] [int] IDENTITY(1,1) NOT NULL,  
    [TuVung] [nchar](50) NULL,  
    [Description] [nvarchar](200) NULL,  
    [Stranlate] [nvarchar](200) NULL,  
    [MaChiTietBH] [int] NULL,  
    [MaChiTietKTra] [int] NULL,  
    CONSTRAINT [PK_TuVung] PRIMARY KEY
```

```
CREATE TABLE [dbo].[Users](  
    [UserName] [nvarchar](50) NOT NULL,  
    [HoTen] [nvarchar](50) NULL,  
    [NgaySinh] [nchar](10) NULL,  
    [Email] [nvarchar](50) NULL,  
    [PassWord] [nvarchar](50) NULL,  
    [Quyen] [nchar](15) NULL,  
    CONSTRAINT [PK_User] PRIMARY KEY
```

❖ Ràng buộc khóa ngoại:

```
ALTER TABLE [dbo].[BaiHoc] WITH CHECK ADD CONSTRAINT  
[FK_BaiHoc_ChiTietBaiHoc] FOREIGN KEY([MaBaiHoc])  
REFERENCES [dbo].[ChiTietBaiHoc] ([MaChiTietBH])  
GO
```

```
ALTER TABLE [dbo].[BaiHoc] CHECK CONSTRAINT [FK_BaiHoc_ChiTietBaiHoc]  
GO
```

```
ALTER TABLE [dbo].[ChiTietKTra] WITH CHECK ADD CONSTRAINT  
[FK_ChiTietKTra_KiemTra] FOREIGN KEY([maKiemTra])  
REFERENCES [dbo].[KiemTra] ([maKiemTra])  
GO
```

```
ALTER TABLE [dbo].[ChiTietKTra] CHECK CONSTRAINT [FK_ChiTietKTra_KiemTra]  
GO
```

```
ALTER TABLE [dbo].[Diem] WITH CHECK ADD CONSTRAINT [FK_Diem_Users]  
FOREIGN KEY([UserName])  
REFERENCES [dbo].[Users] ([UserName])
```



```
GO
ALTER TABLE [dbo].[Diem] CHECK CONSTRAINT [FK_Diem_Users]
GO
ALTER TABLE [dbo].[Grammar] WITH CHECK ADD CONSTRAINT
[FK_Grammar_ChiTietBaiHoc] FOREIGN KEY([MaChiTietBH])
REFERENCES [dbo].[ChiTietBaiHoc] ([MaChiTietBH])
GO
ALTER TABLE [dbo].[Grammar] CHECK CONSTRAINT [FK_Grammar_ChiTietBaiHoc]
GO
ALTER TABLE [dbo].[Grammar] WITH CHECK ADD CONSTRAINT
[FK_Grammar_ChiTietKTra] FOREIGN KEY([MaChiTietKTra])
REFERENCES [dbo].[ChiTietKTra] ([MaChiTietKTra])
GO
ALTER TABLE [dbo].[Grammar] CHECK CONSTRAINT [FK_Grammar_ChiTietKTra]
GO
ALTER TABLE [dbo].[Tracking] WITH CHECK ADD CONSTRAINT
[FK_Tracking_BaiHoc] FOREIGN KEY([MaBaiHoc_Grammar])
REFERENCES [dbo].[BaiHoc] ([MaBaiHoc])
GO
ALTER TABLE [dbo].[Tracking] CHECK CONSTRAINT [FK_Tracking_BaiHoc]
GO
ALTER TABLE [dbo].[Tracking] WITH CHECK ADD CONSTRAINT
[FK_Tracking_KiemTra] FOREIGN KEY([maKiemTra_Grammar])
REFERENCES [dbo].[KiemTra] ([maKiemTra])
GO
ALTER TABLE [dbo].[Tracking] CHECK CONSTRAINT [FK_Tracking_KiemTra]
GO
ALTER TABLE [dbo].[Tracking] WITH CHECK ADD CONSTRAINT [FK_Tracking_Users]
FOREIGN KEY([UserName])
REFERENCES [dbo].[Users] ([UserName])
GO
ALTER TABLE [dbo].[Tracking] CHECK CONSTRAINT [FK_Tracking_Users]
GO
ALTER TABLE [dbo].[TuVung] WITH CHECK ADD CONSTRAINT
[FK_TuVung_ChiTietBaiHoc] FOREIGN KEY([MaChiTietBH])
REFERENCES [dbo].[ChiTietBaiHoc] ([MaChiTietBH])
GO
ALTER TABLE [dbo].[TuVung] CHECK CONSTRAINT [FK_TuVung_ChiTietBaiHoc]
GO
ALTER TABLE [dbo].[TuVung] WITH CHECK ADD CONSTRAINT
[FK_TuVung_ChiTietKTra] FOREIGN KEY([MaChiTietKTra])
REFERENCES [dbo].[ChiTietKTra] ([MaChiTietKTra])
GO
ALTER TABLE [dbo].[TuVung] CHECK CONSTRAINT [FK_TuVung_ChiTietKTra]
GO
```

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

### 2.1. Thêm, sửa, xóa User bằng quyền admin.

- Thêm

```
CREATE PROCEDURE [dbo].[sp_Insert_Username]
@HoTen nvarchar(50),
@NgàySinh date,
@EMAIL NVARCHAR(50),
@UserName nvarchar(50),
@PASS NVARCHAR(50),
@Quyền NVARCHAR(50)
```

AS

Begin

declare @sql nvarchar(max)

declare @count int

SET @count = (SELECT COUNT(\*) FROM Users WHERE UserName  
=@UserName);

IF(@count =0)

BEGIN

IF(@Quyền ='HV')

BEGIN

set @sql = 'CREATE LOGIN [' +@USERNAME+  
']WITH PASSWORD= "' +@PASS+"', DEFAULT\_DATABASE=[Toeic],  
CHECK\_EXPIRATION=ON, CHECK\_POLICY=ON; CREATE USER['  
+@USERNAME+'] FOR LOGIN[' +@USERNAME+'];ALTER ROLE  
[db\_owner] ADD MEMBER [' +@UserName+']';

exec(@sql);

print 'EXCE HV'

END

IF(@Quyền ='ADMIN')

BEGIN

set @sql = 'CREATE LOGIN [' +@USERNAME+  
']WITH PASSWORD= "' +@PASS+"', DEFAULT\_DATABASE=[Toeic],  
CHECK\_EXPIRATION=ON, CHECK\_POLICY=ON ;ALTER SERVER  
ROLE [securityadmin] ADD MEMBER [' +@USERNAME+']; CREATE  
USER[' +@USERNAME+'] FOR LOGIN[' +@USERNAME+'];ALTER  
ROLE [db\_owner] ADD MEMBER [' +@UserName+']';

exec(@sql);

print 'EXCE Admin'

END

ELSE

BEGIN

```
        set @sql = 'CREATE LOGIN [' + @USERNAME +
'WITH PASSWORD= "' + @PASS + "', DEFAULT_DATABASE=[Toeic],
CHECK_EXPIRATION=ON, CHECK_POLICY=ON ;ALTER SERVER
ROLE [securityadmin] ADD MEMBER [' + @USERNAME +']; CREATE
USER[' + @USERNAME +'] FOR LOGIN[' + @USERNAME +'];ALTER
ROLE [db_datareader] ADD MEMBER [' + @UserName +'];
        exec (@sql);
        print 'EXCE Admin'
    END
```

```
        INSERT INTO dbo.Users VALUES
(@UserName, @HoTen, @NgaySinh, @EMAIL, @PASS, @Quyen);
        INSERT INTO
dbo.Tracking(UserName, maKiemTra_Grammar, MaBaiHoc_Grammar, maKie
mTra_TuVung, maBH_TuVung) VALUES (@UserName, 1, 1, 1, 1);
```

END

End

- Sửa  
CREATE PROCEDURE [dbo].[AD\_UDAPTE\_NGUOIDUNG]  
@HoTen nvarchar(50),  
@NgaySinh date,  
@EMAIL NVARCHAR(50),  
@UserName nvarchar(50),  
@PASS NVARCHAR(50),  
@QUYEN NVARCHAR(50)  
  
AS  
Begin  
 UPDATE DBO.Users SET  
HoTen=@HoTen, NgaySinh=@NgaySinh, Email=@EMAIL, UserN  
ame=@UserName, PassWord=@PASS, Quyen=@QUYEN where  
UserName=@UserName;

End

- Xóa  
CREATE PROCEDURE [dbo].[delete\_user]  
@userName nvarchar(50)  
AS  
Begin

```
declare @sql1 nvarchar(max)
declare @sql2 nvarchar(max)
declare @sql3 nvarchar(max)
delete from dbo.Users where UserName =@userName;
set @sql1='DROP USER ' + @USERNAME+";
set @sql2='DROP LOGIN ' + @userName+";

exec(@sql1);
exec(@sql2);
```

End

- Lấy thông tin User

```
CREATE PROCEDURE [dbo].[LAYTHONGTIN_USER]
@UserName nvarchar(50)
AS
Begin
    SELECT * FROM dbo.laythongtin( @UserName);
End
```

## 2.2. Thêm điểm

```
CREATE PROCEDURE [dbo].[Insert_DIEM]
@MaUSER INT,
@Point INT,
@MaKTra INT
AS
Begin
    INSERT INTO dbo.Diem VALUES
(@MaUSER,@Point,@MaKTra);
End
```

## 2.3. Thêm ngữ pháp.

```
Create PROCEDURE [dbo].[Insert_Grammer]
@Syntax nvarchar(50),
@NameGrammar nvarchar(50),
@Description nvarchar(300),
@MaChiTietBH int,
@MaChiTietKTra int

AS
Begin
    INSERT INTO
dbo.Grammar(Syntax,NameGrammar,Description,MaChiTietBH,
```

```
MaChiTietKTra) VALUES
(@Syntax,@NameGrammar,@Description,@MaChiTietBH,@Ma
ChiTietKTra);
End
```

**2.4. Thêm từ vựng.**

```
CREATE PROCEDURE [dbo].[Insert_TuVung]
@TuVung nvarchar(50),
@Description nvarchar(200),
@Stranlate nvarchar(200),
@MaChiTietBH int,
@MaChiTietKTra int
AS
Begin
    INSERT INTO dbo.TuVung
(TuVung,TuVung.Description,Stranlate,MaChiTietBH,MaChiTiet
KTra) VALUES
(@TuVung,@Description,@Stranlate,@MaChiTietBH,@MaChiTi
etKTra);
End
```

**2.5. Sửa thông tin người dùng**

```
CREATE PROCEDURE [dbo].[UDAPTE_NGUOIDUNG]
@HoTen nvarchar(50),
@NgaySinh date,
@EMAIL NVARCHAR(50),
@UserName nvarchar(50)
AS
Begin
    UPDATE DBO.Users SET
HoTen=@HoTen,NgaySinh=@NgaySinh,Email=@EMAIL,UserN
ame=@UserName where UserName=@UserName;
End
```

**2.6. Next, Back từ vựng, grammar**

- Next từ vựng

```
CREATE PROCEDURE [dbo].[NEXT_TUVUNG]
@USERNAME VARCHAR(50)
Begin
    DECLARE @TEMP INT;
```

```
Declare CS_TuVung CURSOR FOR(SELECT* FROM
LAYMABH_TuVung_TRACKING(@USERNAME));
OPEN CS_TuVung
FETCH NEXT FROM CS_TuVung INTO @TEMP
CLOSE CS_TuVung
IF(@TEMP<12)
BEGIN
UPDATE DBO.Tracking SET maBH_TuVung= @TEMP+1
WHERE UserName=@USERNAME;
END
SELECT *
FROM
LAYMABH_TuVung_TRACKING(@USERNAME),TuVung,Chi
TietBaiHoc
WHERE TuVung.MaChiTietBH = ChiTietBaiHoc.MaChiTietBH
AND ChiTietBaiHoc.MaBaiHoc =
LAYMABH_TuVung_TRACKING.maBH_TuVung;
End
```

▪ Next kiểm tra từ vựng

```
CREATE PROCEDURE [dbo].[NEXT_KTra_TUVUNG]
@USERNAME VARCHAR(50)
AS
Begin
DECLARE @TEMP INT;
Declare CS_TuVung CURSOR FOR(SELECT* FROM
LAY_MA_KTra_TuVung_TRACKING(@USERNAME));
OPEN CS_TuVung
FETCH NEXT FROM CS_TuVung INTO @TEMP
CLOSE CS_TuVung
IF(@TEMP<12)
BEGIN
UPDATE DBO.Tracking SET maKiemTra_TuVung=
@TEMP+1 WHERE UserName=@USERNAME;
END
SELECT *
FROM
LAY_MA_KTra_TuVung_TRACKING(@USERNAME),TuVung
,ChiTietKTra
```

```
WHERE TuVung.MaChiTietKTra = ChiTietKTra.MaChiTietKTra
AND ChiTietKTra.maKiemTra =
LAY_MA_KTra_TuVung_TRACKING.maKiemTra_TuVung;
End
```

- Next ngữ pháp

```
CREATE PROCEDURE [dbo].[NEXT_GRAMMAR]
@USERNAME VARCHAR(50)
AS
Begin
    DECLARE @TEMP INT;
    Declare CS_MAGRAMMAR CURSOR FOR(SELECT*
FROM
LAYMABH_GRAMMAR_TRACKING(@USERNAME));
    OPEN CS_MAGRAMMAR
    FETCH NEXT FROM CS_MAGRAMMAR INTO @TEMP
    CLOSE CS_MAGRAMMAR
    if(@TEMP<9)
    BEGIN
        UPDATE DBO.Tracking
        SET MaBaiHoc_GRAMMAR= @TEMP+3
        WHERE UserName=@USERNAME;
        END
        SELECT *
FROM
LAYMABH_GRAMMAR_TRACKING(@USERNAME),Gramm
ar,ChiTietBaiHoc
WHERE Grammar.MaChiTietBH = ChiTietBaiHoc.MaChiTietBH
AND ChiTietBaiHoc.MaBaiHoc >=
LAYMABH_GRAMMAR_TRACKING.MABAIHOC_GRAMM
AR AND ChiTietBaiHoc.MaBaiHoc <
LAYMABH_GRAMMAR_TRACKING.MABAIHOC_GRAMM
AR +3;
End
```

- Back kiểm tra từ vựng

```
CREATE PROCEDURE [dbo].[BACK_TuVung_KTra]
@MA_TuVung INT,
@username varchar(50)
AS
Begin
```

```
UPDATE dbo.Tracking SET maKiemTra_TuVung
=@MA_TuVung -1 where UserName =@username;
SELECT *
FROM TuVung,ChiTietKTra
WHERE TuVung.MaChiTietKTra= ChiTietKTra.MaChiTietKTra
AND ChiTietKTra.maKiemTra= @MA_TuVung -1
End
```

▪ Back từ vựng

```
CREATE PROCEDURE [dbo].[BACK_TuVung]
@MA_TuVung INT,
@username varchar(50)
AS
Begin
UPDATE dbo.Tracking SET maBH_TuVung =@MA_TuVung
-1 where UserName =@username;
SELECT *
FROM TuVung,ChiTietBaiHoc
WHERE TuVung.MaChiTietBH =
ChiTietBaiHoc.MaChiTietBH AND ChiTietBaiHoc.MaBaiHoc=
@MA_TuVung -1
End
```

▪ Back ngữ pháp

```
CREATE PROCEDURE [dbo].[BACK_GRAMMAR]
@MA_GRAMMAR INT,
@username varchar(50)
AS
Begin
UPDATE dbo.Tracking SET MaBaiHoc_Grammar
=@MA_GRAMMAR -1 where UserName =@username;

SELECT *
FROM Grammar,ChiTietBaiHoc
WHERE Grammar.MaChiTietBH =
ChiTietBaiHoc.MaChiTietBH AND ChiTietBaiHoc.MaBaiHoc>
@MA_GRAMMAR -3 AND ChiTietBaiHoc.MaBaiHoc<=
@MA_GRAMMAR
End
```



## 2.7. Kiểm tra login, thay đổi password

### ▪ Kiểm tra login

```
CREATE PROCEDURE [dbo].[checklogin]
@username varchar(100),
@password varchar(100)
```

```
AS
```

```
Begin
```

```
    Select UserName,Password,Quyền from Users where
    UserName=@username and Password=@password
```

```
End
```

### ▪ Thay đổi password

```
CREATE PROCEDURE [dbo].[ChangePassWord]
```

```
@username varchar(50),
```

```
@password_old varchar(100),
```

```
@password_new varchar(100)
```

```
AS
```

```
Begin
```

```
    if((Select count( *) from Users where UserName=@username and
    PassWord =@password_old)=1)
```

```
        begin
```

```
            update Users set PassWord =@password_new where UserName
            =@username;
```

```
            select * from Users;
```

```
        end
```

```
End
```

## 2.8. Lấy dữ liệu ngữ pháp, từ vựng

### ▪ Lấy dữ liệu kiểm tra từ vựng

```
CREATE PROCEDURE [dbo].[GET_KTra_TuVung] @USERNAME
VARCHAR(50)
```

```
AS
```

```
Begin
```

```
    SELECT *
```

```
FROM
```

```
LAY_MA_KTra_TuVung_TRACKING(@USERNAME),TuVung,ChiTietBaiHoc
```

```
WHERE TuVung.MaChiTietBH = ChiTietBaiHoc.MaChiTietBH AND  
ChiTietBaiHoc.MaBaiHoc =  
LAY_MA_KTra_TuVung_TRACKING.maKiemTra_TuVung  
End
```

▪ **Lấy dữ liệu kiểm tra ngữ pháp**

```
CREATE PROCEDURE [dbo].[GET_KTra_GRAMMAR]  
@USERNAME VARCHAR(50)  
AS  
Begin  
    SELECT *  
FROM  
LAYMA_KTra_GRAMMAR_TRACKING(@USERNAME),Grammar,  
ChiTietBaiHoc  
WHERE Grammar.MaChiTietBH = ChiTietBaiHoc.MaChiTietBH AND  
ChiTietBaiHoc.MaBaiHoc =  
LAY_MA_KTra_GRAMMAR_TRACKING.maKiemTra_Grammar  
End
```

▪ **Lấy dữ liệu bài học từ vựng**

```
CREATE PROCEDURE [dbo].[GET_BH_TuVung] @USERNAME  
VARCHAR(50)  
AS  
Begin  
    SELECT *  
FROM  
LAYMABH_TuVung_TRACKING(@USERNAME),TuVung,ChiTietB  
aiHoc  
WHERE TuVung.MaChiTietBH = ChiTietBaiHoc.MaChiTietBH AND  
ChiTietBaiHoc.MaBaiHoc =  
LAYMABH_TuVung_TRACKING.maBH_TuVung  
End
```

▪ **Lấy dữ liệu bài học ngữ pháp**

```
CREATE PROCEDURE [dbo].[GET_BH_GRAMMER]  
@USERNAME VARCHAR(50)  
AS  
Begin  
    SELECT *
```

```
FROM
LAYMABH_GRAMMAR_TRACKING(@USERNAME),Grammar,Chi
TietBaiHoc
WHERE Grammar.MaChiTietBH = ChiTietBaiHoc.MaChiTietBH AND
ChiTietBaiHoc.MaBaiHoc >=
LAYMABH_GRAMMAR_TRACKING.MABAIHOC_GRAMMAR
AND ChiTietBaiHoc.MaBaiHoc <
LAYMABH_GRAMMAR_TRACKING.MABAIHOC_GRAMMAR +3
End
```

### 3. Kết nối cơ sở dữ liệu

Code kết nối cơ sở dữ liệu thông qua lớp DataAccesLayer.

```
namespace DAL
{
    public class DALayer
    {
        SqlConnection cnn=null;
        SqlCommand cmd=null;
        SqlDataAdapter adp=null;
        string str = null;

        public DALayer(string Source,string username,string password){
            str="Data Source="+Source+";Initial Catalog=Toeic;User
ID="+username+";Password="+password+"";
            cnn = new SqlConnection(str);
            cmd = cnn.CreateCommand();
        }

        public DataSet ExecuteQueryDataSet(string strSQL,CommandType
ct,params SqlParameter[] p)
        {
            if (cnn.State == ConnectionState.Open)
                cnn.Close();
            cnn.Open();

            DataSet ds = new DataSet();
            try
```

```
        {
            cmd.CommandText = strSQL;
            cmd.CommandType = ct;
            adp = new SqlDataAdapter(cmd);

            adp.Fill(ds);

        }
        catch (Exception e)
        {
            Console.WriteLine("co loi ket noi den datatbase!!!",
e.InnerException);

        }
        return ds;
    }
    public DataSet LayDuLieu_DK(string strSQL, CommandType
ct, params SqlParameter[] param)
    {
        if (cnn.State == ConnectionState.Open)
            cnn.Close();
        cnn.Open();
        cmd.Parameters.Clear();
        //cmd.CommandText = strSQL;
        //cmd.CommandType = ct;
        foreach (SqlParameter p in param)
        {
            cmd.Parameters.Add(p);

        }

        DataSet ds = new DataSet();
        try
        {
            cmd.CommandText = strSQL;
            cmd.CommandType = ct;
            adp = new SqlDataAdapter(cmd);

            adp.Fill(ds);
```

```
    }
    catch (Exception e)
    {
        Console.WriteLine("co loi LayDuLieu_DK",
e.InnerException);

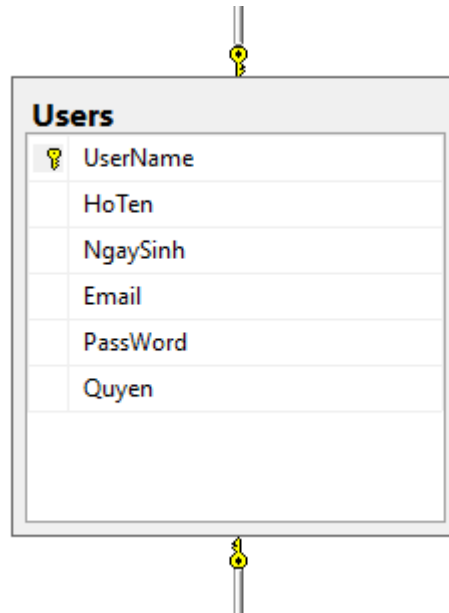
    }
    return ds;
}

public bool MyExecuteNonQuery(string strSQL, CommandType
ct, ref string error, params SqlParameter[] param)
{
    bool f = false;
    if (cnn.State == ConnectionState.Open)
        cnn.Close();
    cnn.Open();
    cmd.Parameters.Clear();
    cmd.CommandText = strSQL;
    cmd.CommandType = ct;
    foreach (SqlParameter p in param)
        cmd.Parameters.Add(p);
    try
    {
        cmd.ExecuteNonQuery();
        f = true;
    }
    catch (SqlException ex)
    {
        error = ex.Message;
    }
    finally
    {
        cnn.Close();
    }
    return f;
}
}
```

}

#### 4. Phân quyền

##### 4.1. Bảng cơ sở dữ liệu dùng để phân quyền:



UserName	HoTen	NgaySinh	Email	PassWord	Quyen
admin	Thanh Khoa	1994-03-01	tthanhkhoa@g...	123	ADMIN
duclinh	Nguyen Duc Linh	1994-01-01	linh@gmail.com	123	ADMIN
linh123	linh123	1900-01-01	a@gmail.com	1234	ADMIN
linhaaa	Duc Linh	05/11/1994	aaaa@gmail.com	123	ADMIN
tthanhkhoa	Tran Thanh Khoa	1994-03-01	tthanhkhoa@g...	123	HV

Đề tài này tạo ra 2 user: người dùng và admin

##### 4.2. Code phân quyền trên C#:

```
private void btnOK_Click(object sender, EventArgs e)
{
    thongtinnguoidung = new
BAL_USER(cbbDataSource.Text.ToString(), txtten.Text.ToString(),
txtmatkhau.Text.ToString());
    dt = new DataTable();
    dt.Clear();
    try
    {
```

```
        try
        {
            dt =
thongtinnguoidung.Checklogin(txtten.Text.ToString(),
txtmatkhau.Text.ToString()).Tables[0];
            temp = dt.Rows[0][2].ToString();
        }
        catch (Exception) {
            temp = "QT";
        }
        session = txtten.Text.ToString();
        this.Hide();

        MainForm mainform = new MainForm();
        mainform.username = session;
        mainform.DataSource =
cbbDataSource.Text.ToString();
        mainform.password = txtmatkhau.Text.ToString();
        mainform.Quyen = temp;

    }
    catch (System.Data.SqlClient.SqlException sqlException)
    {

        System.Windows.Forms.MessageBox.Show(sqlException.Message);
    }
}
```

#### 4.3. Phân quyền trên cơ sở dữ liệu:

### 5. Trigger, transaction

- 5.1. Tạo trigger đảm bảo rằng điểm nhập vào phải lớn hơn 0 và nhỏ hơn 10

```
Create TRIGGER [dbo].[CHECK_Diem] --Tên Trigger
ON [dbo].[Diem]
FOR UPDATE, INSERT
AS
BEGIN
    DECLARE @Diem AS INT
    SELECT @Diem=inserted.Point FROM inserted
    IF (@Diem>10 OR @Diem<0)
    BEGIN
        PRINT N'ĐIỂM KHÔNG HỢP LỆ'
        ROLLBACK TRANSACTION
    END
END
```

```
END
```

### 5.2. Tạo trigger đảm bảo rằng grammar không được trùng nhau

```
Create TRIGGER [dbo].[CHECK_Grammar] --Tên Trigger
ON [dbo].[Grammar]
FOR UPDATE, INSERT
AS
BEGIN
    DECLARE @GRAMMAR AS NCHAR(50), @TEMP AS INT
    SELECT @GRAMMAR=inserted.NameGrammar FROM inserted
    SELECT @TEMP=COUNT(*) FROM dbo.Grammar
    WHERE NameGrammar=@GRAMMAR
    IF (@TEMP>1)
    BEGIN
        PRINT N'GRAMMAR Đã Tồn Tại'
        ROLLBACK TRANSACTION
    END
END
```

### 5.3. Tạo trigger đảm bảo rằng từ vựng không được trùng nhau

```
Create TRIGGER [dbo].[CHECK_TuVung] --Tên Trigger
ON [dbo].[TuVung]
For UPDATE, INSERT
AS
BEGIN
    DECLARE @TUVUNG AS NCHAR(50), @TEMP AS INT
    SELECT @TUVUNG=inserted.TuVung FROM inserted
    SELECT @TEMP=COUNT(*) FROM dbo.TuVung
    WHERE TuVung=@TUVUNG
    IF (@TEMP>1)
    BEGIN
        PRINT N'Từ Vựng Đã Tồn Tại'
        ROLLBACK TRANSACTION
    END
END
```

### 5.4. Tạo trigger đảm bảo tài khoản không trùng nhau.

```
Create TRIGGER [dbo].[CHECK_USERS] --Tên Trigger
ON [dbo].[Users]
FOR UPDATE, INSERT
AS
BEGIN
    DECLARE @USERNAME AS NCHAR(50), @TEMP AS INT
    SELECT @USERNAME=inserted.UserName FROM inserted
    SELECT @TEMP=COUNT(*) FROM dbo.Users
    WHERE UserName=@USERNAME
    IF (@TEMP>1)
    BEGIN
        PRINT N'TÀI KHOẢN Đã Tồn Tại'
        ROLLBACK TRANSACTION
    END
END
```



## 6. Function

### 6.1. Sử dụng Function tạo bảng câu hỏi và đáp án.

Hàm trả về một bảng có 3 cột: Câu hỏi, Nội dung, Đáp án

```
CREATE function [dbo].[f_baitap](@grammar int)
returns table
as
return
( select CauHoi.CauHoi, CauHoi.NoiDung, CauHoi.DapAn
  from CauHoi
 where CauHoi.MaGrammar = @grammar)
```

### 6.2. Sử dụng function tạo bảng bài tập grammar

Hàm trả về một bảng có 4 cột: Grammar, Câu hỏi, Nội dung, Đáp án

```
Creat function [dbo].[f_BTgrammar](@maBaiHoc int)
returns table
as
return
( select Grammar.maGrammar, CauHoi, CauHoi.NoiDung, CauHoi.DapAn
  from f_CTbaiHoc(@maBaiHoc), Grammar, CauHoi
 where f_CTbaiHoc.MaChiTietBH = Grammar.MaChiTietBH and
        Grammar.maGrammar = CauHoi.MaGrammar)
```

### 6.3. Sử dụng function tạo bảng bài tập từ vựng

Hàm trả về một bảng có 2 cột: maTuVung, Từ Vựng

```
Create function [dbo].[f_BTtuvung](@maBaiHoc int)
returns table
as
return
( select TuVung.maTuVung, TuVung.TuVung
  from f_CTbaiHoc(@maBaiHoc), TuVung
 where f_CTbaiHoc.MaChiTietBH = TuVung.MaChiTietBH)
```

### 6.4. Sử dụng function tạo bảng chi tiết bài học

Hàm trả về một bảng có 1 cột: Mã Chi tiết Bài Học

```
Creat function [dbo].[f_CTbaiHoc](@maBaiHoc int)
returns table
as
return
( select MaChiTietBH
  from ChiTietBaiHoc
 where ChiTietBaiHoc.MaBaiHoc = @maBaiHoc)
```

- 6.5. Sử dụng function tạo bảng chi tiết kiểm tra  
Hàm trả về một bảng có 1 cột: Mã Chi Tiết Kiểm Tra

```

Creat function [dbo].[f_CTkiemtra](@maKiemTra int)
returns table
as
    return
        ( select MaChiTietKTra
          from ChiTietKTra
          where ChiTietKTra.maKiemTra = @maKiemTra)
```

- 6.6. Sử dụng function tạo bảng kiểm tra grammar  
Hàm trả về một bảng có 4 cột: Grammar, Câu hỏi, Nội dung, Đáp án

```

Creat function [dbo].[f_KTgrammar](@maKiemTra int)
returns table
as
    return
        ( select Grammar.maGrammar, CauHoi, CauHoi.NoiDung, CauHoi.DapAn
          from f_CTkiemtra(@maKiemTra), Grammar, CauHoi
          where f_CTkiemtra.MaChiTietKTra = Grammar.MaChiTietKTra and
          Grammar.maGrammar = CauHoi.MaGrammar)
```

- 6.7. Sử dụng function tạo bảng kiểm tra từ vựng  
Hàm trả về một bảng có 2 cột: Mã từ vựng, Từ vựng

```

Creat function [dbo].[f_KTtuvung](@maKiemTra int)
returns table
as
    return
        ( select TuVung.maTuVung, TuVung.TuVung
          from f_CTkiemtra(@maKiemTra), TuVung
          where f_CTkiemtra.MaChiTietKTra = TuVung.MaChiTietKTra)
```

- 6.8. Sử dụng function tạo bảng điểm kiểm tra.  
Hàm trả về một bảng có 3 cột: Mã user, Họ Tên, Điểm

```

Creat function [dbo].[f_PoinKT](@maKT int)
returns table
as
    return(
        select Users.maUser, Users.HoTen, Point
        from Diem, Users
        where Diem.maKiemTra = @maKT and Users.maUser = Diem.maUser
    )
```

- 6.9. Sử dụng function tạo bảng thống kê điểm user

Hàm trả về một bảng có 3 cột: Mã User, Họ Tên, Điểm

```

Creat function [dbo].[f_ThongKeDiem] (@maUser int)
returns table
as
return
(
select @maUser as MaUser, Users.HoTen, Diem.Point
from Users, Diem
where Users.maUser = Diem.maUser and Users.maUser = @maUser)

```

#### 6.10. Sử dụng function tạo bảng tra từ điển

Hàm trả về một bảng có 3 cột: Từ Vựng, Nghĩa từ vựng, Description

```

Creat function [dbo].[f_tratudien](@tu nvarchar(50))
returns table
as
return
(
select TuVung.TuVung, TuVung.Stranlate, TuVung.Description
from TuVung
where TuVung.TuVung like N'%' + @tu + '%')

```

#### 6.11. Sử dụng function tìm điểm cao nhất của user

```

Creat function [dbo].[f_MaxDiem](@maUser int)
returns int
as
begin
return
(
select Max(Diem.Point) as MaxDiem
from Diem
where Diem.maUser = @maUser)
end

```

#### 6.12. Sử dụng function tìm điểm cao nhất bài kiểm tra

```

Creat function [dbo].[f_MaxKT](@maKT int)
returns int
as
begin
return(
select MAX(Point)
from Diem, Users
where Diem.maKiemTra = @maKT and Users.maUser = Diem.maUser
)
end

```

#### 6.13. Sử dụng function tìm bài học tiếp theo cho user

```

Creat function [dbo].[f_NextBH](@maUser int)

```

```
returns nchar(30)
as
begin
    declare @CTBH int, @TenBH nchar(30)
    select @CTBH = BaiHoc.MaBaiHoc +1
    from Tracking, ChiTietBaiHoc, Grammar, BaiHoc
    where Tracking.maGrammar = Grammar.maGrammar and Grammar.MaChiTietBH
= ChiTietBaiHoc.MaChiTietBH
    and ChiTietBaiHoc.MaBaiHoc = BaiHoc.MaBaiHoc and BaiHoc.maUser =
@maUser
    select @TenBH = BaiHoc.TenBaiHoc
    from BaiHoc
    where BaiHoc.MaBaiHoc = @CTBH
    return @TenBH
end
```

## 7. View

- 7.1. Sử dụng View lọc ra các thuộc tính MaDT, SoLuong, NgayNhap áp dụng cho việc làm Báo cáo thống kê số lượng nhập và bán trong một khoảng thời gian nào đó.

```
CREATE VIEW [ChiTietNhap] AS
select MaDT, SoLuong, NgayNhap
from ChiTietPhieuNhap, PhieuNhap
where ChiTietPhieuNhap.MaPN=PhieuNhap.MaPN
```

Kết quả khi ta truy vấn từ View [ChiTietNhap]

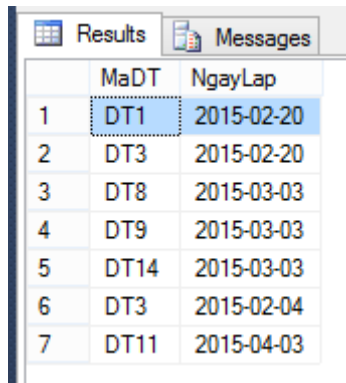
```
select * from dbo.ChiTietNhap
```

	MaDT	SoLuong	NgayNhap
1	DT1	3	2014-02-03
2	DT10	7	2014-02-03
3	DT15	4	2014-02-03
4	DT8	6	2014-02-03
5	DT4	5	2014-08-12
6	DT5	7	2014-08-12
7	DT9	5	2014-08-12
8	DT12	10	2014-12-12
9	DT14	7	2014-12-12
10	DT2	7	2014-12-12
11	DT11	7	2014-06-07
12	DT1	11	2015-01-02
13	DT16	7	2015-01-02
14	DT4	5	2015-01-02
15	DT6	10	2015-01-02
16	DT13	5	2015-02-03
17	DT14	2	2015-02-03
18	DT2	8	2015-02-03
19	DT3	10	2015-05-02
20	DT6	5	2015-05-02
21	DT7	10	2015-05-02
22	DT9	5	2015-05-02

```
CREATE VIEW [ChiTietBan] AS
select MaDT,NgayLap
from ChiTietHoaDon,HoaDon
where ChiTietHoaDon.MaHD=HoaDon.MaHD
```

Kết quả khi ta truy vấn từ View [ChiTietBan]

```
select * from dbo.ChiTietBan
```



	MaDT	NgayLap
1	DT1	2015-02-20
2	DT3	2015-02-20
3	DT8	2015-03-03
4	DT9	2015-03-03
5	DT14	2015-03-03
6	DT3	2015-02-04
7	DT11	2015-04-03

7.2. Sử dụng cho chức năng làm báo cáo doanh thu.

```
create View [BaoCaoDoanhThu] as
select ChiTietHoaDon.MaDT, ChiTietHoaDon.Gia,
ChiTietHoaDon.ThanhTien,
HoaDon.NgayLap
from ChiTietHoaDon, DienThoai, HoaDon
where ChiTietHoaDon.MaDT= DienThoai.MaDT and
ChiTietHoaDon.MaHD=HoaDon.MaHD
```

kết quả:



	Mã Điện Thoại	Giá Tiền	Thành Tiền	Ngày Lập
▶	DT1	1649000	1649000	2/20/2015
	DT3	479000	479000	2/20/2015
	DT7	300000	280000	2/20/2015
	DT8	300000	300000	3/3/2015
	DT9	2409000	2409000	3/3/2015
	DT14	4800000	4800000	3/3/2015
<	DT2	479000	479000	2/4/2015

Tổng Số Lượng: 8      Tổng Tiền:      Thoát

7.3. Sử dụng View cho chức năng Tìm Kiếm.

```
create view [view_DuLieuTimKiem] as
select Gia, TenDT, TenHSX
from DienThoai, HangSX
```

where DienThoai.MaHSX=HangSX.MaHSX  
kết quả truy vấn:

	Gia	TenDT	TenHSX
1	1649000	Samsung Galaxy V Plus	SangSum
2	5000000	Asus Zenfone 2 ZE551ML	Other
3	5240000	Xiaomi Mi4	Sony
4	450000	Mobiistar B60	Coolpad
5	3290000	HTC 8X	HTC
6	4800000	oppo neo5	Alcatell
7	560000	Nokia N160	Nokia
8	2030000	samsung galaxy J1	SangSum
9	3365000	Samsung Galaxy Grand Prime	SangSum
10	479000	Suntexk X5 Plus 2	Nokia
11	14160000	Apple iPhone 6	Apple
12	3790000	Lenovo a7000	lenovo
13	8990000	HTC One M8 Eye	HTC
14	3950000	KingZone N5	Huawei
15	300000	Philips	ZTE
16	2409000	LG Optimus L3 II E425	LG

## 8. Index

1. Index trên cột username  
CREATE INDEX indexUser ON Users(username)
2. Index trên cột Tuvung  
CREATE INDEX indexTuVung ON TuVung(TuVung)
3. Index trên cột Tranlaste  
CREATE INDEX indexTranlaste ON TuVung(Tranlaste)
4. Index trên cột Grammar  
CREATE INDEX indexGrammar ON Grammar(NameGrammar)
5. Index trên cột điểm  
CREATE UNIQUE INDEX indexDiem ON DIEM(username,point)

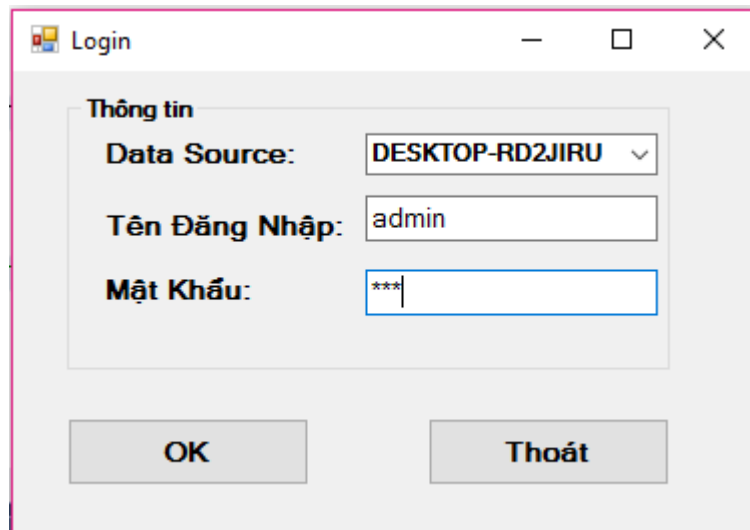
## CHƯƠNG 3: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

### 1. Đăng nhập

Người dùng nhập ip sever chứa cơ sở dữ liệu và sử dụng tài khoản của mình để đăng nhập vào hệ thống.

Phần mềm có 2 quyền như sau:

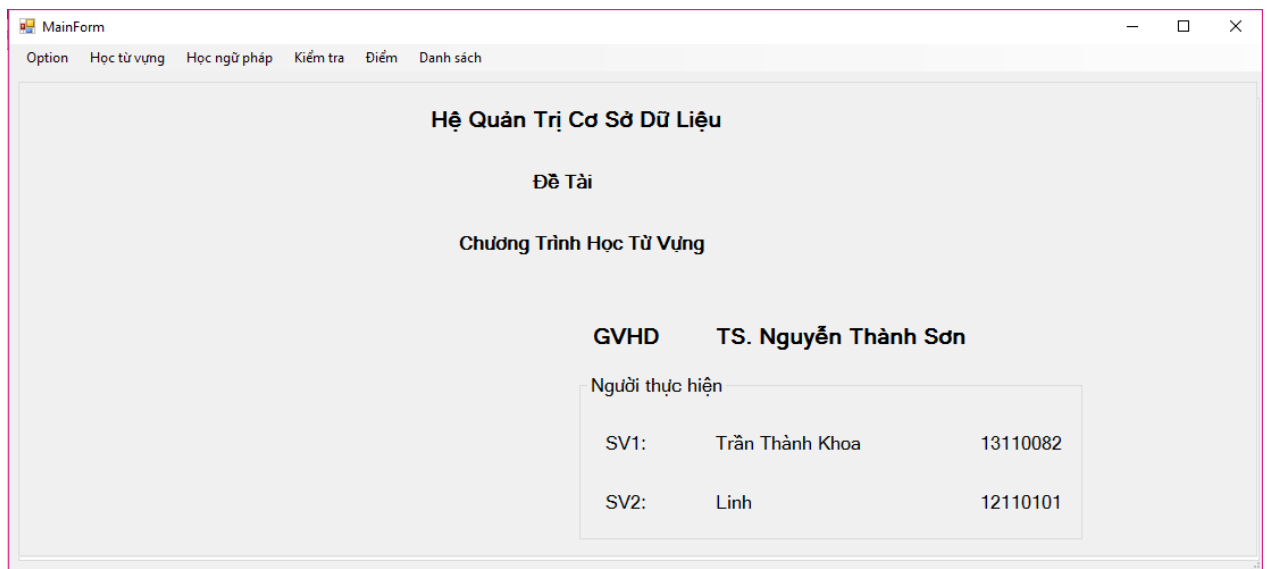
- Quyền admin.
- Quyền người dùng.



### 2. Màn hình trang chủ:

Sau khi đăng nhập thì trang chủ của mỗi quyền đăng nhập sẽ hiện lên.

- Nếu chọn quyền đăng nhập là admin thì sẽ ra giao diện như sau:



SV	Họ Tên	Số
SV1:	Trần Thành Khoa	13110082
SV2:	Linh	12110101



- Nếu đăng nhập là quyền nhân viên sẽ có giao diện sau:



MainForm

Option Học từ vựng Học ngữ pháp Kiểm tra Điểm Danh sách

Hệ Quản Trị Cơ Sở Dữ Liệu

Đề Tài

Chương Trình Học Từ Vựng

GVHD TS. Nguyễn Thành Sơn

Người thực hiện

SV1:	Trần Thành Khoa	13110082
SV2:	Linh	12110101

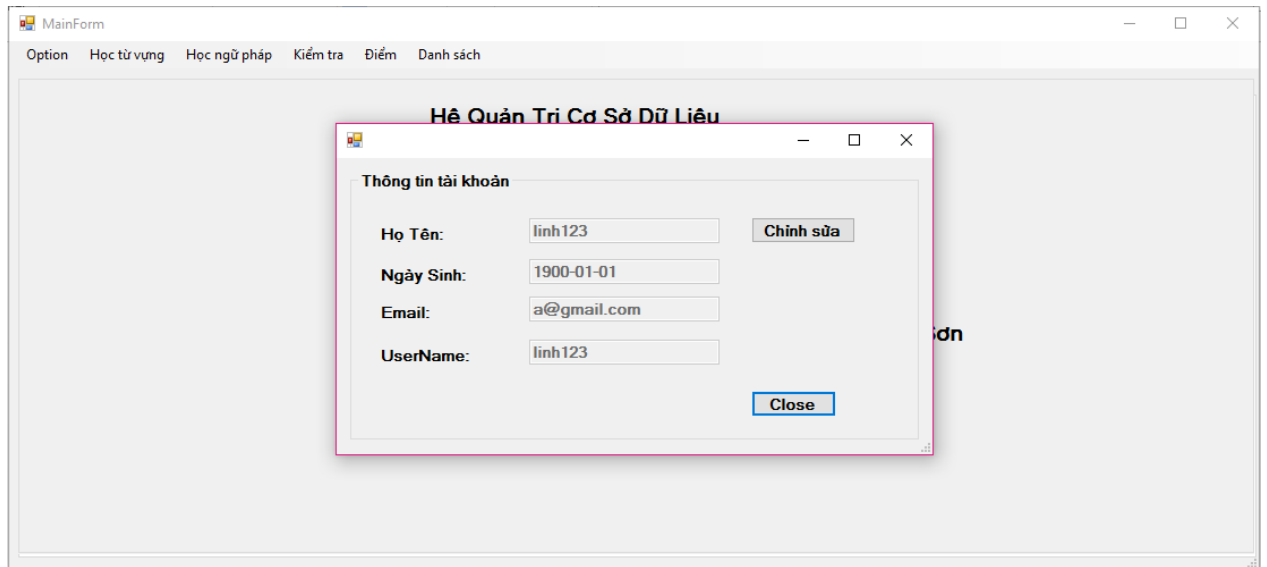
### 3. Xem thông tin

Trong phần mềm này có thể xem thông tin của các thành phần sau:

- Thông tin tài khoản.
- Thống kê điểm.
- Học từ vựng.
- Học ngữ pháp.
- Kiểm tra từ vựng.
- Kiểm tra ngữ pháp.
- Thông tin danh sách từ vựng.
- Thông tin danh sách ngữ pháp.
- Thông tin danh sách người học.

Nếu là quyền người dùng thì mục nào ẩn đi thì mục đó có nghĩa là họ không có quyền được xem.

**Ví dụ:** Sau khi đăng nhập với quyền người dùng, muốn xem mục thống kê điểm. Bạn vào **Option -> Thông tin tài khoản**. Lúc này màn hình xem danh mục thông tin tài khoản hiện lên.



Tương tự cho các mục khác.

Nếu là admin cũng thao tác giống như vậy.

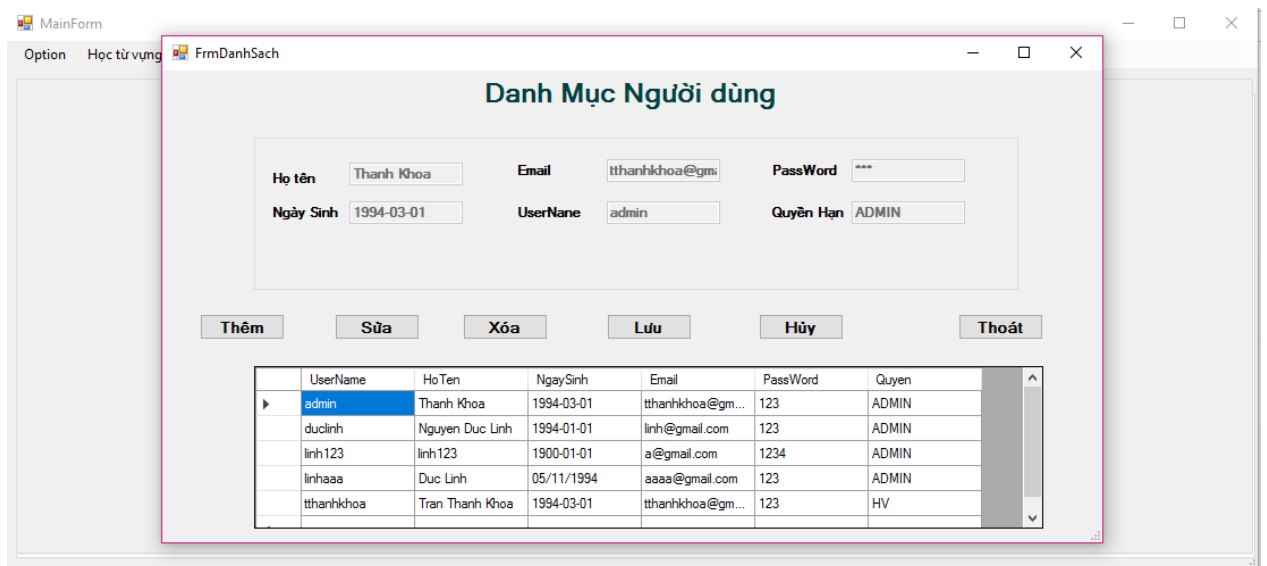
#### 4. Quản lý

Ở phần quản lý này cũng quản lý các thông tin như phần “Xem thông tin”. Phần này cho phép chúng ta thêm, cập nhật, xóa những thông tin mà ta muốn.

Ví dụ: ta cần quản lý danh mục người dùng.

Vào **Danh Sách** -> **Danh Sách Người Dùng**.

Giao diện quản lý danh mục người dùng hiện lên.



- Lúc này các nút thêm, sửa, xóa hiện lên.

- Nếu chọn thêm thì hệ thống sẽ cho phép ta điền những thông tin về người dùng vào. Đây là giao diện khi nhấn nút thêm.

**Danh Mục Người dùng**

Họ tên:  Email:  PassWord:   
Ngày Sinh:  UserNane:  Quyền Hạn:

Thêm Sửa Xóa Lưu Hủy Thoát

	UserName	Ho Ten	NgaySinh	Email	PassWord	Quyen
▶	admin	Thanh Khoa	1994-03-01	tthanhkhoa@gm...	123	ADMIN
	duclinh	Nguyen Duc Linh	1994-01-01	linh@gmail.com	123	ADMIN
	linh123	linh 123	1900-01-01	a@gmail.com	1234	ADMIN
	linhaaa	Duc Linh	05/11/1994	aaaa@gmail.com	123	ADMIN
	tthanhkhoa	Tran Thanh Khoa	1994-03-01	tthanhkhoa@gm...	123	HV

- Sau khi điền những thông tin về người dùng xong thì ta nhấn nút lưu nếu muốn lưu hoặc nhấn nút hủy nếu không muốn lưu lại.
- Khi nhấn nút lưu thì người dùng đó sẽ được hiển thị xuống phía dưới.
- Nếu muốn Sửa người dùng nào thì ta chọn người dùng bên dưới và click chuột vào nút Sửa. Lúc này, thì chúng ta chỉ có thể sửa trên các thuộc tính.

**Danh Mục Người dùng**

Họ tên:  Email:  PassWord:   
Ngày Sinh:  UserNane:  Quyền Hạn:

Thêm Sửa Xóa Lưu Hủy Thoát

	UserName	Ho Ten	NgaySinh	Email	PassWord	Quyen
▶	duclinh	Nguyen Duc Linh	1994-01-01	linh@gmail.com	123	ADMIN
	linh123	linh 123	1900-01-01	a@gmail.com	1234	ADMIN
	linhaaa	Duc Linh	05/11/1994	aaaa@gmail.com	123	ADMIN
	tthanhkhoa	Tran Thanh Khoa	1994-03-01	tthanhkhoa@gm...	123	HV

- Nếu muốn xóa thì ta chọn 1 người dùng nào đó và click chuột vào nút Xóa.

## 5. Cấu hình.

- **Nếu muốn đăng xuất khỏi hệ thống.**

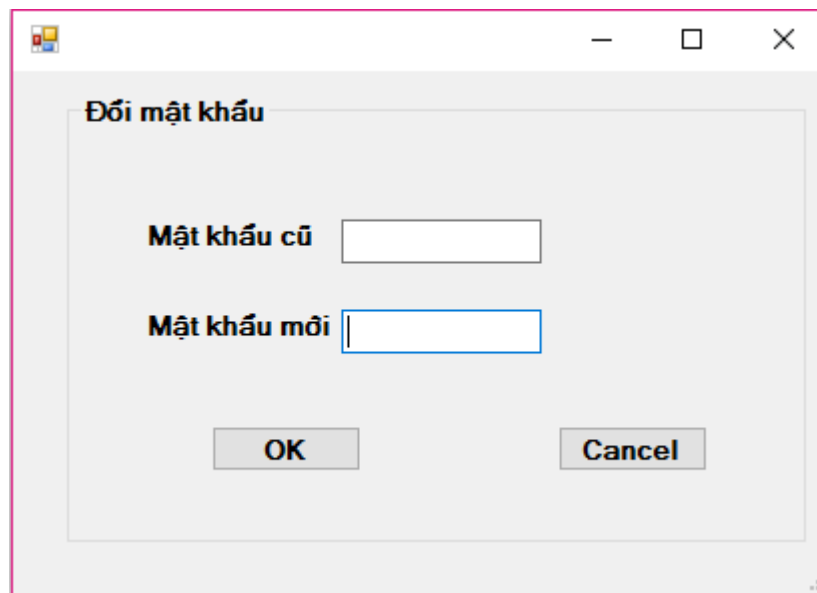
Vào **Option** -> chọn **Đăng Xuất**.

Bạn sẽ thoát khỏi hệ thống. Nếu muốn vào lại hệ thống phải đăng nhập lại.

- **Nếu muốn thay đổi mật khẩu.**

Vào **Option** -> chọn **Đổi Mật Khẩu**.

Giao diện đổi mật khẩu hiện lên.



- Bạn sẽ nhập những thông tin mà hệ thống yêu cầu.
- Click chuột vào **đổi mật khẩu**.
- Lúc này mật khẩu của bạn sẽ được thay đổi. lần sau khi đăng nhập vào thì bạn sẽ phải đăng nhập bằng mật khẩu mới này.