

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



Báo cáo Lab 4

Mã hóa dữ liệu phía client

Môn học: Bảo mật cơ sở dữ liệu

CSC15002_22MMT

Sinh viên:

Nguyễn Hồ Đăng Duy
Phạm Quang Duy

Giảng viên hướng dẫn:

Nguyễn Đình Thúc
Nguyễn Thị Hường
Lê Trọng Anh Tú

Mục lục

1	Thông tin	2
1.1	Thông tin sinh viên	2
1.2	Phân công nhiệm vụ	2
2	Câu b	3
2.1	SP_INS_PUBLIC_ENCRYPT_NHANVIEN	3
2.2	SP_SEL_PUBLIC_ENCRYPT_NHANVIEN	4
3	Câu d	5
3.1	Các thuật toán mã hóa và giải mã	5
3.1.1	Thuật toán băm SHA1	5
3.1.2	Thuật toán mã hóa/giải mã RSA	5
3.1.3	Mã hóa với public key XML	6
3.2	Màn hình Quản lý đăng nhập	7
3.3	Màn hình Thêm nhân viên	9
3.4	Màn hình Quản lý lớp học	10
3.5	Màn hình Thông tin nhân viên	12
3.6	Màn hình Danh sách sinh viên	14
3.7	Màn hình Thêm sinh viên mới	17
3.8	Màn hình Xóa sinh viên	19
3.9	Màn hình Điều chỉnh thông tin sinh viên	21
3.10	Màn hình Chi tiết điểm	23
3.11	Màn hình Thêm / chỉnh sửa điểm	25
4	Câu e	27
4.1	Cài Đặt SQL Server Profiler	27
4.2	Cấu Hình SQL Profiler	27
4.3	Theo dõi truy vấn	28
4.4	Thực hiện với form thêm điểm	29

1 Thông tin

1.1 Thông tin sinh viên

Nhóm gồm có 2 thành viên:

- 22127085 - Nguyễn Hồ Đăng Duy - 22127085@student.hcmus.edu.vn
- 22127088 - Phạm Quang Duy - 22127088@student.hcmus.edu.vn

1.2 Phân công nhiệm vụ

Sinh viên	Các câu đã làm	Tiến độ
Nguyễn Hồ Đăng Duy	Các giao diện và procedure Đăng nhập, Quản lý lớp học, Danh sách sinh viên của lớp, Thêm sinh viên mới, Xóa sinh viên, Điều chỉnh thông tin sinh viên	100%
Phạm Quang Duy	Các hàm mã hóa và giải mã. Các giao diện và procedure Thêm nhân viên, Thông tin nhân viên, Thêm / chỉnh sửa điểm	100%

Bảng 1: Bảng phân công nhiệm vụ

2 Câu b

2.1 SP_INS_PUBLIC_ENCRYPT_NHANVIEN

```

1 IF EXISTS (SELECT 1 FROM sys.procedures WHERE name = '
    SP_INS_PUBLIC_ENCRYPT_NHANVIEN')
2 BEGIN
3     DROP PROCEDURE SP_INS_PUBLIC_ENCRYPT_NHANVIEN;
4 END
5 GO
6
7 CREATE PROCEDURE SP_INS_PUBLIC_ENCRYPT_NHANVIEN
8     @MANV NVARCHAR(20),
9     @HOTEN NVARCHAR(100),
10    @EMAIL NVARCHAR(100),
11    @LUONGCB VARBINARY(MAX),      -- RSA encrypted, base64 string
12    @TENDN NVARCHAR(100),
13    @MK VARBINARY(MAX),          -- SHA1 hashed password (hex string)
14    @PUBKEY VARCHAR(MAX),
15    @ErrorMessage NVARCHAR(200) OUTPUT
16 AS
17 BEGIN
18     SET NOCOUNT ON;
19
20     BEGIN TRY
21         IF EXISTS (SELECT 1 FROM NHANVIEN WHERE MANV = @MANV)
22             BEGIN
23                 SET @ErrorMessage = N'Ma nhan vien da ton tai.';
24                 RETURN;
25             END
26     DECLARE @SQL NVARCHAR(MAX);
27
28     INSERT INTO NHANVIEN (MANV, HOTEN, EMAIL, LUONG, TENDN, MATKHAU, PUBKEY)
29     VALUES (@MANV, @HOTEN, @EMAIL, @LUONGCB, @TENDN, @MK, @PUBKEY);
30
31     SET @ErrorMessage = N''; -- OK
32 END TRY
33 BEGIN CATCH
34     SET @ErrorMessage = ERROR_MESSAGE();
35 END CATCH
36 END
37 GO

```

2.2 SP_SEL_PUBLIC_ENCRYPT_NHANVIEN

```
1 IF OBJECT_ID('SP_SEL_PUBLIC_ENCRYPT_NHANVIEN', 'P') IS NOT NULL
2     DROP PROCEDURE SP_SEL_PUBLIC_ENCRYPT_NHANVIEN;
3 GO
4
5 CREATE PROCEDURE SP_SEL_PUBLIC_ENCRYPT_NHANVIEN
6     @TENDN NVARCHAR(100),
7     @MK VARBINARY(MAX)
8 AS
9 BEGIN
10     SET NOCOUNT ON;
11
12     SELECT MANV, HOTEN, EMAIL, LUONG
13     FROM NHANVIEN
14     WHERE MANV = @TENDN AND MATKHAU = @MK;
15 END
16 GO
```

3 Câu d

Chương trình được viết bằng C# trên Visual Studio Developer 2022

3.1 Các thuật toán mã hóa và giải mã

3.1.1 Thuật toán băm SHA1

Dùng để băm mật khẩu nhân viên (nhân viên đăng nhập, thêm mới).

```
1 public static byte[] HashPasswordSHA1(string input)
2 {
3     using (SHA1 sha1 = SHA1.Create())
4     {
5         return sha1.ComputeHash(Encoding.UTF8.GetBytes(input));
6     }
7 }
```

3.1.2 Thuật toán mã hóa/giải mã RSA

Mã hóa RSA: Dùng để mã hóa lương cơ bản nhân viên và điểm sinh viên, sử dụng khóa công khai (public key).

```
1 public static (byte[] encryptedData, string publicKeyXml, string privateKeyXml)
2 EncryptDataRSA(string plainText, byte[] hashedPassword, string manv)
3 {
4     // key = manv + hashedPassword
5     string keyContainerName = Convert.ToBase64String(Encoding.UTF8.GetBytes(manv
6     + "_" + Convert.ToBase64String(hashedPassword)));
7
8     CspParameters cspParams = new CspParameters
9     {
10         KeyContainerName = keyContainerName
11     };
12     using (var rsa = new RSACryptoServiceProvider(2048, cspParams))
13     {
14         rsa.PersistKeyInCsp = false;
15
16         string publicKeyXml = rsa.ToXmlString(false);
17         string privateKeyXml = rsa.ToXmlString(true);
18
19         byte[] dataToEncrypt = Encoding.UTF8.GetBytes(plainText);
20         byte[] encryptedData = rsa.Encrypt(dataToEncrypt, false);
21
22         return (encryptedData, publicKeyXml, privateKeyXml);
23     }
24 }
```

Giải mã RSA: Dùng để giải mã lượng hoặc điểm, sử dụng khóa riêng (private key) đã lưu riêng theo MANV.

```
1 public static string DecryptDataRSA(byte[] encryptedData, string privateKeyXml)
2 {
3     using (var rsa = new RSACryptoServiceProvider(2048))
4     {
5         rsa.PersistKeyInCsp = false;
6         rsa.FromXmlString(privateKeyXml);
7
8         byte[] decryptedBytes = rsa.Decrypt(encryptedData, false);
9         return Encoding.UTF8.GetString(decryptedBytes);
10    }
11 }
```

3.1.3 Mã hóa với public key XML

Được dùng khi nhân viên nhập điểm sinh viên:

```
1 public static byte[] EncryptWithPublicKey(string plainText, string publicKeyXml)
2 {
3     using (var rsa = new RSACryptoServiceProvider(2048))
4     {
5         rsa.PersistKeyInCsp = false;
6         rsa.FromXmlString(publicKeyXml);
7         byte[] dataToEncrypt = Encoding.UTF8.GetBytes(plainText);
8         return rsa.Encrypt(dataToEncrypt, false);
9     }
10 }
```

3.2 Màn hình Quản lý đăng nhập

Hình 1: Giao diện đăng nhập

- Người dùng được yêu cầu nhập vào **Mã nhân viên** và **Mật khẩu** đã được lưu trong cơ sở dữ liệu.
- Khi nhấn vào nút **Xác nhận** thì chương trình sẽ chạy procedure **SP_LOGIN_NHANVIEN** để kiểm tra. Nếu MANV và MK đúng thì chuyển sang giao diện Quản lý lớp học. Nếu không thì sẽ hiện MessageBox "Đăng nhập không thành công! Kiểm tra lại tài khoản hoặc mật khẩu."
- Nút **Hủy** để xóa hết các thông tin người dùng vừa nhập.
- Nút **Thêm nhân viên** để chuyển sang giao diện thêm một nhân viên mới.

Store procedure **SP_LOGIN_NHANVIEN** nhận vào 2 giá trị là **@MANV** và **@MK**, trong đó **@MK** đã được hash bằng SHA1 bằng C#. Nếu kiểm tra đúng thì sẽ trả về thông tin của nhân viên đó, còn không thì sẽ trả về NULL.

```

1 IF OBJECT_ID('SP_LOGIN_NHANVIEN', 'P') IS NOT NULL
2     DROP PROCEDURE SP_LOGIN_NHANVIEN;
3 GO
4
5 CREATE PROCEDURE SP_LOGIN_NHANVIEN

```

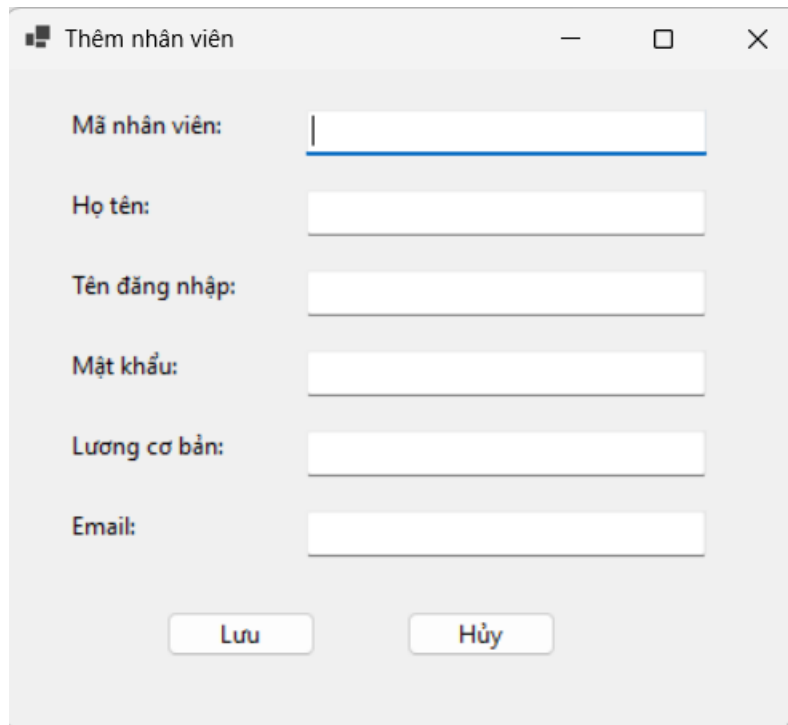


```

6      @MANV VARCHAR(20),
7      @MK VARBINARY(MAX) -- SHA1 ma hoa base64 tu C#
8 AS
9 BEGIN
10     SET NOCOUNT ON;
11
12     -- Convert base64 string -> VARBINARY
13     DECLARE @MK_BIN VARBINARY(MAX) =
14         CONVERT(VARBINARY(MAX), CAST(N'' AS XML).value('xs:base64Binary(sql:
15         variable("@MK"))', 'VARBINARY(MAX)'));
16
17     IF EXISTS (
18         SELECT 1 FROM NHANVIEN
19         WHERE MANV = @MANV AND MATKHAU = @MK_BIN
20     )
21     BEGIN
22         SELECT MANV, HOTEN, EMAIL
23         FROM NHANVIEN
24         WHERE MANV = @MANV
25     END
26     ELSE
27     BEGIN
28         SELECT NULL AS MANV, NULL AS HOTEN, NULL AS EMAIL
29     END
30 END
31 GO

```

3.3 Màn hình Thêm nhân viên

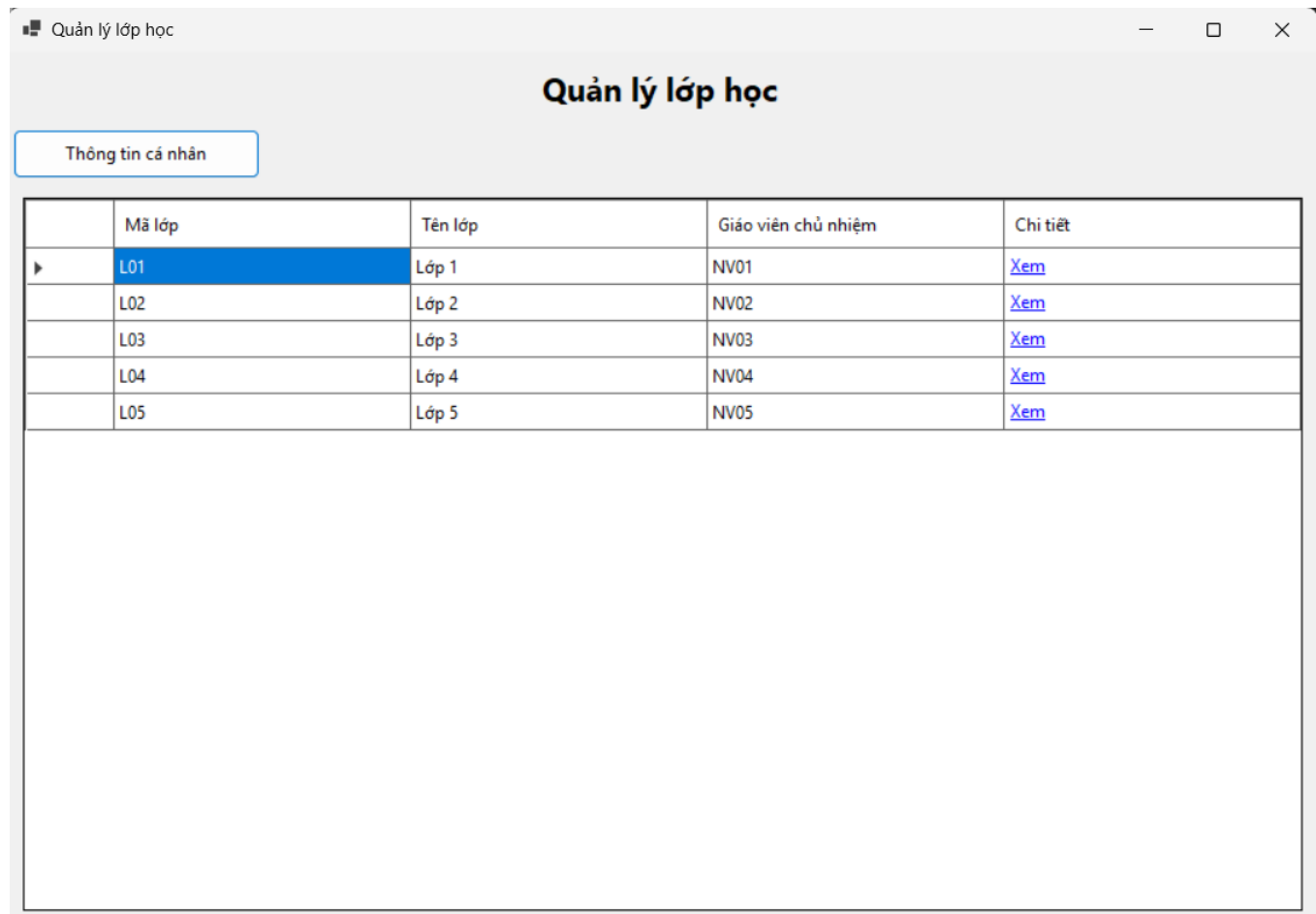


Hình 2: Giao diện Thêm nhân viên

- Người dùng được yêu cầu nhập vào các thông tin cần thiết của một nhân viên. Nếu thiếu thông tin thì sẽ không thể lưu.
- Giá trị **Mật khẩu** sẽ được hash bằng SHA1, **Lương cơ bản** sẽ được mã hóa bằng RSA trước khi lưu xuống database.
- Khi nhấn vào nút **Lưu** thì chương trình sẽ chạy procedure **SP_INS_PUBLIC_ENCRYPT_NHANVIEN** để lưu tất cả thông tin xuống database

Store procedure **SP_INS_PUBLIC_ENCRYPT_NHANVIEN** và thuật toán mã hóa đã được đề cập ở phần trên.

3.4 Màn hình Quản lý lớp học



Hình 3: Giao diện quản lý lớp học

- Sau khi đăng nhập thành công, chương trình sẽ gọi procedure **SP_GET_CLASSES** để xem toàn bộ danh sách lớp trong cơ sở dữ liệu.
- Khi nhấn vào nút **Xem**, chương trình sẽ chuyển qua giao diện Danh sách sinh viên của lớp.
- Nếu người dùng nhấn vào nút **Thông tin cá nhân** thì chương trình sẽ hiển thị thông tin của người vừa đăng nhập.

Store procedure SP_GET_CLASSES dùng để trả về toàn bộ danh sách lớp trong cơ sở dữ liệu.

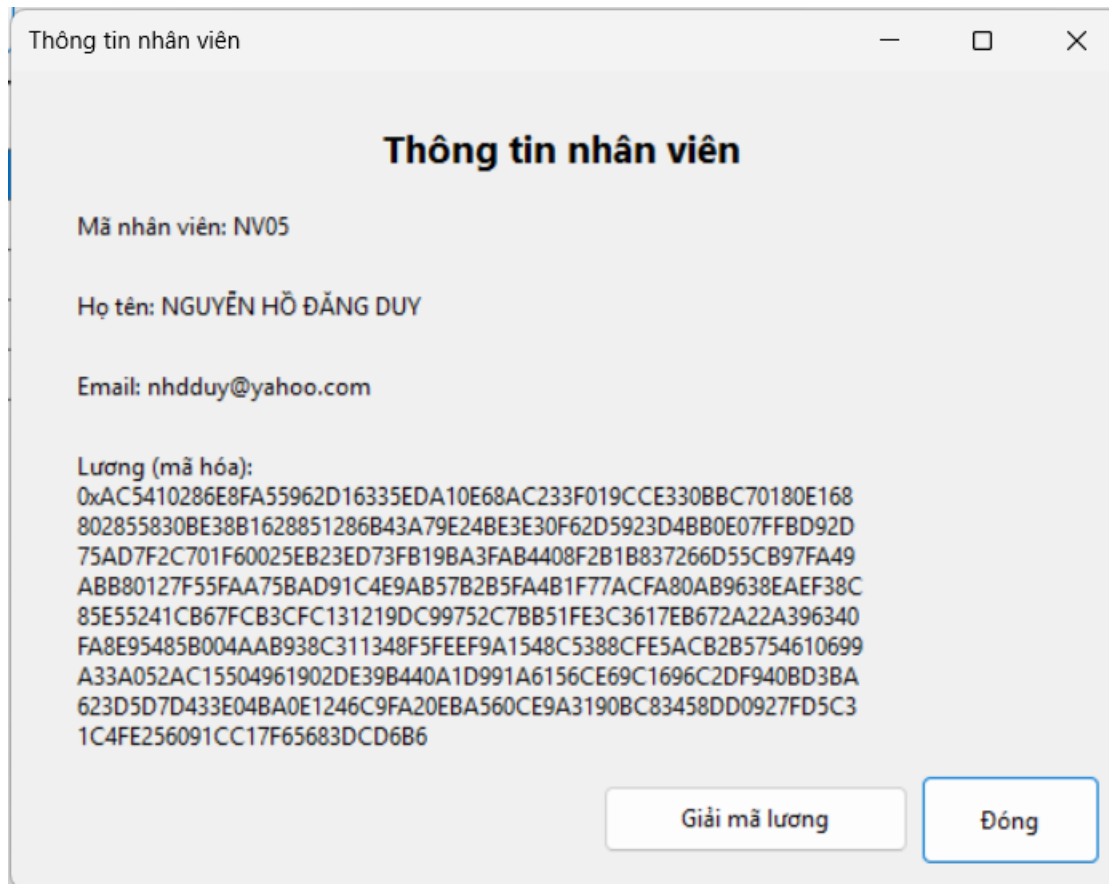
```

1 IF OBJECT_ID('SP_GET_CLASSES', 'P') IS NOT NULL
2 BEGIN
3     DROP PROCEDURE SP_GET_CLASSES;
4 END;
5 GO
6
7 CREATE PROCEDURE SP_GET_CLASSES
8 AS

```

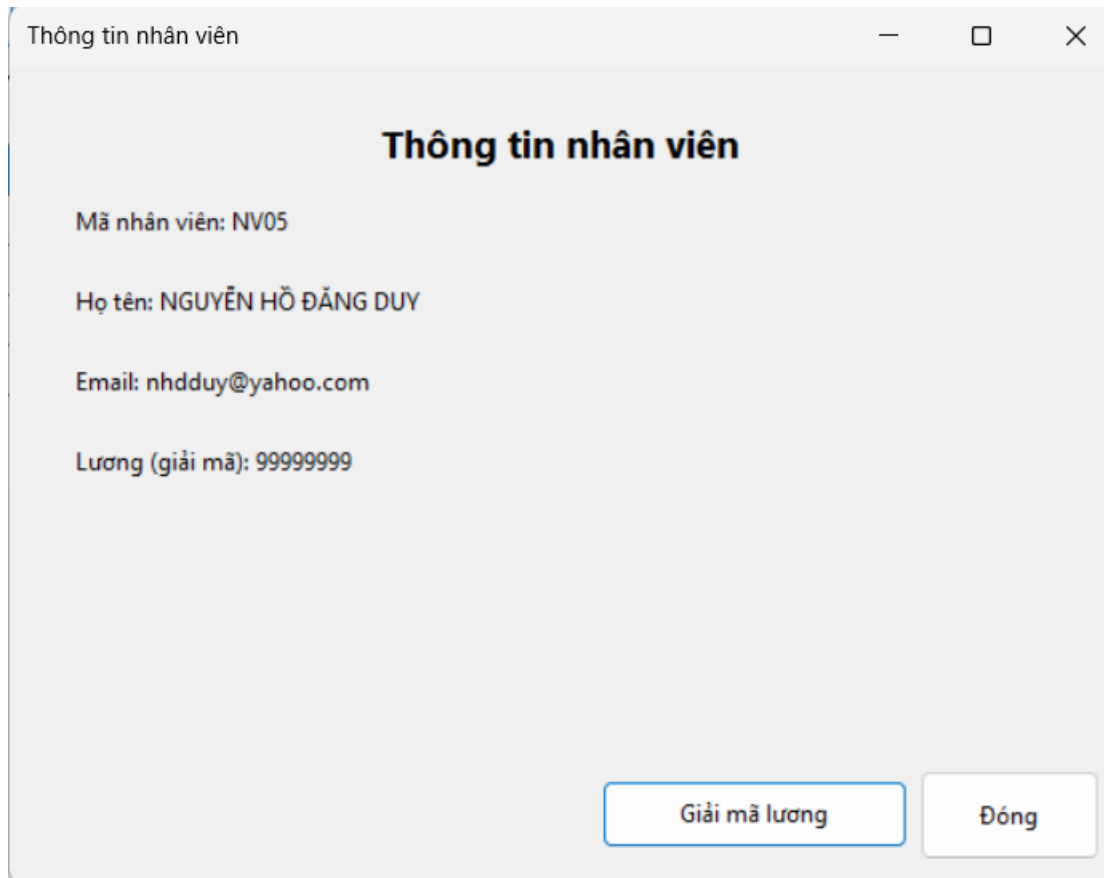
```
9 BEGIN
10     SELECT MALOP, TENLOP, MANV
11     FROM LOP;
12 END
13 GO
```

3.5 Màn hình Thông tin nhân viên



Hình 4: Giao diện Thông tin nhân viên

- Sau khi nhấn vào nút **Thông tin cá nhân** ở giao diện **Quản lý lớp học**, chương trình sẽ hiển thị thông tin chi tiết của nhân viên đã đăng nhập.
- Chương trình sẽ gọi procedure **SP_SEL_PUBLIC_ENCRYPT_NHANVIEN** để hiển thị thông tin của nhân viên (lương ở dạng mã hóa).
- Nếu người dùng muốn xem lương cụ thể thì nhấn **Giải mã lương**, màn hình sẽ hiển thị lương đã giải mã.



Hình 5: Lương đã được giải mã

Store procedure SP_SEL_PUBLIC_ENCRYPT_NHANVIEN và các hàm giải mã đã được đề cập ở phần trên.

Thông tin chi tiết lớp

Thông tin chi tiết lớp L01

Tìm kiếm sinh viên theo mã sinh viên

Tìm

MASV	HOTEN	NGAYSINH	DIACHI	MALOP
SV010	Bùi Bảo Chi	1/7/2003	Quận 11	L01
SV017	Lê Văn Quân	9/3/2004	Bình Tân	L01
SV019	Ngô Văn Quân	4/12/2003	Quận 11	L01
SV026	Phạm Quang Khoa	10/8/2003	Quận 11	L01
SV027	Hoàng Hoài Nam	12/28/2002	Quận 10	L01
SV032	Phạm Minh Nam	11/16/2002	Tân Bình	L01
SV034	Trần Bảo Uyên	10/5/2004	Nhà Bè	L01
SV039	Lê Hoài Nam	5/18/2003	Tân Bình	L01
SV045	Trần Thị Hà	6/26/2002	Bình Chánh	L01
SV047	Đặng Nhật Mai	11/19/2004	Nhà Bè	L01
SV051	Phạm Nhật Hà	9/13/2004	Quận 3	L01
SV052	Hoàng Thị Trang	9/26/2002	Bình Tân	L01
SV056	Đặng Minh Khoa	1/19/2004	Quận 5	L01
SV061	Ngô Đăng Khoa	2/11/2003	Gò Vấp	L01
SV070	Đặng Bảo An	2/8/2004	Quận 4	L01
SV087	Hồ Hữu An	5/20/2003	Gò Vấp	L01
SV090	Đỗ Hoài An	12/16/2002	Quận 2	L01

Thay đổi thông tin

Thêm sinh viên

Xóa sinh viên

Chỉnh sửa điểm

Hình 6: Giao diện Danh sách sinh viên lớp quản lý

- Chương trình sẽ gọi procedure **SP_GET_STUDENTS_BY_CLASS** để có thể hiển thị danh sách sinh viên của từng lớp.
- Nếu lớp đó do nhân viên đã đăng nhập quản lý, màn hình sẽ hiện 4 chức năng ở dưới để có thể thay đổi thông tin của sinh viên trong lớp. Khi nhấn vào thì chương trình sẽ chuyển sang các giao diện tương ứng.
- Nếu nhân viên đã đăng nhập không quản lý lớp đó thì chỉ có thể xem danh sách lớp.
- Nếu người dùng nhập mã sinh viên vào thanh tìm kiếm và nhấn **Tìm**, chương trình sẽ gọi procedure **SP_FIND_STUDENT_IN_CLASS**. Nếu mã sinh viên có tồn tại trong lớp thì sẽ trả về thông tin sinh viên đó, nếu không thì sẽ hiện MessageBox "Không tìm thấy sinh viên nào."

Thông tin chi tiết lớp

Thông tin chi tiết lớp L02

Tim kiếm sinh viên theo mã sinh viên

MASV	HOTEN	NGAYSINH	DIACHI	MALOP
SV002	Nguyễn Hữu Bình	12/12/2004	Quận 6	L02
SV007	Phạm Thị Nam	2/27/2002	Quận 10	L02
SV008	Đỗ Hoài Chi	1/21/2003	Tân Phú	L02
SV016	Bùi Hữu Phương	9/14/2002	Quận 10	L02
SV022	Ngô Bảo Sơn	1/23/2004	Quận 12	L02
SV025	Đỗ Văn Linh	9/20/2003	Quận 10	L02
SV035	Hoàng Hữu Uyên	10/21/2002	Quận 11	L02
SV036	Hồ Nhật Bình	12/10/2003	Quận 8	L02
SV038	Lê Thị Nam	11/21/2003	Nhà Bè	L02
SV040	Trần Hoài Trang	8/17/2002	Thủ Đức	L02
SV044	Đặng Thị Uyên	1/22/2003	Quận 6	L02
SV046	Đặng Thanh Hà	1/28/2003	Thủ Đức	L02
SV049	Đặng Hữu Bình	5/22/2003	Quận 2	L02
SV050	Phạm Bảo Chi	10/4/2002	Quận 11	L02
SV055	Nguyễn Hoài Linh	4/22/2002	Bình Chánh	L02
SV072	Hoàng Đăng Sơn	12/11/2002	Tân Phú	L02
SV073	Bùi Hữu Sơn	3/3/2004	Phú Nhuận	L02

Hình 7: Giao diện Danh sách sinh viên lớp không quản lý

Thông tin chi tiết lớp

Thông tin chi tiết lớp L01

SV017

MASV	HOTEN	NGAYSINH	DIACHI	MALOP
SV017	Lê Văn Quân	9/3/2004	Bình Tân	L01

Thay đổi thông tin Thêm sinh viên Xóa sinh viên Chỉnh sửa điểm

Hình 8: Kết quả tìm kiếm sinh viên bằng mã sinh viên

Store procedure SP_GET_STUDENTS_BY_CLASS nhận vào tham số @MALOP để có thể in ra thông tin chi tiết của các sinh viên trong lớp.

```
1 IF OBJECT_ID('SP_GET_STUDENTS_BY_CLASS', 'P') IS NOT NULL
2     DROP PROCEDURE SP_GET_STUDENTS_BY_CLASS;
3 GO
4
5 CREATE PROCEDURE SP_GET_STUDENTS_BY_CLASS
6     @MALOP NVARCHAR(20)
7 AS
8 BEGIN
9     SET NOCOUNT ON;
10    SELECT MASV, HOTEN, NGAYSINH, DIACHI, MALOP
11    FROM SINHVIEN
12    WHERE MALOP = @MALOP;
13 END
14 GO
```

Store procedure SP_FIND_STUDENT_IN_CLASSES nhận vào biến @MASV và @MALOP để tìm kiếm sinh viên trong lớp tương ứng.

```
1 IF OBJECT_ID('SP_FIND_STUDENT_IN_CLASS', 'P') IS NOT NULL
2     DROP PROCEDURE SP_FIND_STUDENT_IN_CLASS;
3 GO
4
5 CREATE PROCEDURE SP_FIND_STUDENT_IN_CLASS
6     @MASV NVARCHAR(20),
7     @MALOP NVARCHAR(20)
8 AS
9 BEGIN
10    SET NOCOUNT ON;
11    SELECT MASV, HOTEN, NGAYSINH, DIACHI, MALOP
12    FROM SINHVIEN
13    WHERE MASV = @MASV AND MALOP = @MALOP;
14 END
15 GO
```

3.7 Màn hình Thêm sinh viên mới

Hình 9: Giao diện thêm sinh viên

- Sau khi bật form thì chương trình sẽ chạy procedure **SP_GENERATE_NEW_STUDENT_ID** để tạo ID sinh viên mới nhất. Bằng cách lấy số của sinh viên cuối cùng và cộng thêm 1 hoặc nếu chưa tồn tại sinh viên nào thì ID sẽ là SV001.
- Sau khi điền đầy đủ thông tin, khi nhấn vào nút **Xác nhận** thì chương trình sẽ chạy procedure **SP_INSERT_STUDENT** với các tham số được thêm vào tương ứng dưới database (Mật khẩu sẽ được hash trước khi lưu xuống database).
- Nếu kiểm tra sự hợp lệ của tên đăng nhập hay mã lớp, Stored procedures sẽ thực hiện thêm sinh viên vào database.
- Nút **Hủy** để xóa hết các thông tin người dùng vừa nhập.

Store procedure **SP_INSERT_STUDENT** nhận vào các giá trị là @MASV, @HOTEN, @NGAYSINH, @DIACHI, @TENDN, @MATKHAU, @MALOP và trả ra @ErrorMessage nếu có lỗi (@MATKHAU đã được hash SHA1 bằng c#).

```

1 IF OBJECT_ID('SP_GENERATE_NEW_STUDENT_ID', 'P') IS NOT NULL
2     DROP PROCEDURE SP_GENERATE_NEW_STUDENT_ID;
3 GO
4
5 CREATE PROCEDURE SP_GENERATE_NEW_STUDENT_ID
6 AS
7 BEGIN
8     DECLARE @NewID INT;
9     DECLARE @MaxID NVARCHAR(20);
10    SELECT @MaxID = MAX(MASV) FROM SINHVIENT;

```

```

11     IF @MaxID IS NULL
12         SET @NewID = 1;
13     ELSE
14         SET @NewID = CAST(SUBSTRING(@MaxID, PATINDEX('%[0-9]%', @MaxID), LEN(
@MaxID)) AS INT) + 1;
15     SELECT 'SV' + RIGHT('000' + CAST(@NewID AS NVARCHAR(10)), 3) AS NewMASV;
16 END
17 GO
18
19 IF OBJECT_ID('SP_INSERT_STUDENT', 'P') IS NOT NULL
20     DROP PROCEDURE SP_INSERT_STUDENT;
21 GO
22
23 CREATE PROCEDURE SP_INSERT_STUDENT
24     @MASV VARCHAR(20),
25     @HOTEN NVARCHAR(100),
26     @NGAYSINH DATETIME,
27     @DIACHI NVARCHAR(200),
28     @MALOP VARCHAR(20),
29     @TENDN NVARCHAR(100),
30     @MATKHAU VARBINARY(MAX),
31     @ErrorMessage NVARCHAR(200) OUTPUT
32 AS
33 BEGIN
34     BEGIN TRY
35         SET NOCOUNT ON;
36
37         IF EXISTS (SELECT 1 FROM SINHVIEN WHERE TENDN = @TENDN)
38             BEGIN
39                 SET @ErrorMessage = 'Ten dang nhap da ton tai!';
40                 RETURN;
41             END
42
43         IF NOT EXISTS (SELECT 1 FROM LOP WHERE MALOP = @MALOP)
44             BEGIN
45                 SET @ErrorMessage = 'Ma lop khong hop le!';
46                 RETURN;
47             END
48
49         INSERT INTO SINHVIEN (MASV, HOTEN, NGAYSINH, DIACHI, MALOP, TENDN,
MATKHAU)
50             VALUES (@MASV, @HOTEN, @NGAYSINH, @DIACHI, @MALOP, @TENDN, @MATKHAU);
51
52         SET @ErrorMessage = '';
53     END TRY
54     BEGIN CATCH
55         SET @ErrorMessage = ERROR_MESSAGE();
56     END CATCH
57 END;
58 GO

```

3.8 Màn hình Xóa sinh viên

Hình 10: Giao diện xóa sinh viên

- Sau khi bật form thì chỉ cần nhập mã sinh viên thì chương trình sẽ chạy procedure **SP_FIND_STUDENT_IN_CLASS** để tìm sinh viên và hiển thị thông tin sinh viên đó.
- Sau khi đã hiện đầy đủ thông tin, khi nhấn vào nút **Xác nhận** thì chương trình sẽ hỏi xác nhận lại rồi chạy procedure **SP_DELETE_STUDENT** với các tham số được thêm vào tương ứng dưới database.
- Nếu kiểm tra sự hợp lệ của input, Stored procedures sẽ thực hiện xóa sinh viên vào database.
- Nút **Hủy** để xóa hết các thông tin người dùng vừa nhập.

Store procedure **SP_DELETE_STUDENT** nhận vào các giá trị là @MASV, @MALOP và trả ra @ErrorMessage nếu có lỗi.

```

1 IF OBJECT_ID('SP_DELETE_STUDENT', 'P') IS NOT NULL
2     DROP PROCEDURE SP_DELETE_STUDENT;
3 GO
4
5 CREATE PROCEDURE SP_DELETE_STUDENT
6     @MASV NVARCHAR(20),
7     @MALOP NVARCHAR(20),
8     @ErrorMessage NVARCHAR(200) OUTPUT
9 AS
10 BEGIN
11     SET NOCOUNT ON;
12     IF NOT EXISTS(SELECT 1 FROM SINHVIENT WHERE MASV = @MASV AND MALOP = @MALOP)
13     BEGIN
14         SET @ErrorMessage = 'Sinh vien khong ton tai trong lop.';
15         RETURN;

```

```
16     END
17
18     DELETE FROM BANGDIEM WHERE MASV = @MASV;
19     DELETE FROM SINHVIEN
20     WHERE MASV = @MASV AND MALOP = @MALOP;
21
22     SET @ErrorMessage = '';
23 END
24 GO
```

3.9 Màn hình Điều chỉnh thông tin sinh viên

Hình 11: Giao diện sửa thông tin sinh viên

- Sau khi bật form thì chỉ cần nhập mã sinh viên thì chương trình sẽ chạy procedure **SP_FIND_STUDENT_IN_CLASS** để tìm sinh viên và hiển thị thông tin sinh viên đó.
- Sau khi đã hiện đầy đủ thông tin, người dùng có thể chỉnh sửa thông tin so với thông tin hiện tại đang được hiển thị. Khi nhấn vào nút **Xác nhận** thì chương trình sẽ chạy procedure **SP_UPDATE_STUDENT_INFO** với các tham số được thêm vào tương ứng dưới database.
- Nếu kiểm tra sự hợp lệ của input, Stored procedures sẽ thực hiện chỉnh sửa thông tin sinh viên vào database.
- Nút **Hủy** để xóa hết các thông tin người dùng vừa nhập.

Store procedure **SP_UPDATE_STUDENT_INFO** nhận vào 6 giá trị là @MASV, @HOTEN, @NGAYSINH, @DIACHI, @OLD_MALOP, @NEW_MALOP và trả ra @ErrorMessage nếu có lỗi.

```

1 IF OBJECT_ID('SP_UPDATE_STUDENT_INFO', 'P') IS NOT NULL
2     DROP PROCEDURE SP_UPDATE_STUDENT_INFO;
3 GO
4
5 CREATE PROCEDURE SP_UPDATE_STUDENT_INFO
6     @MASV NVARCHAR(20),
7     @HOTEN NVARCHAR(100),
8     @NGAYSINH DATETIME,
9     @DIACHI NVARCHAR(200),
10    @OLD_MALOP NVARCHAR(20),
11    @NEW_MALOP NVARCHAR(20),

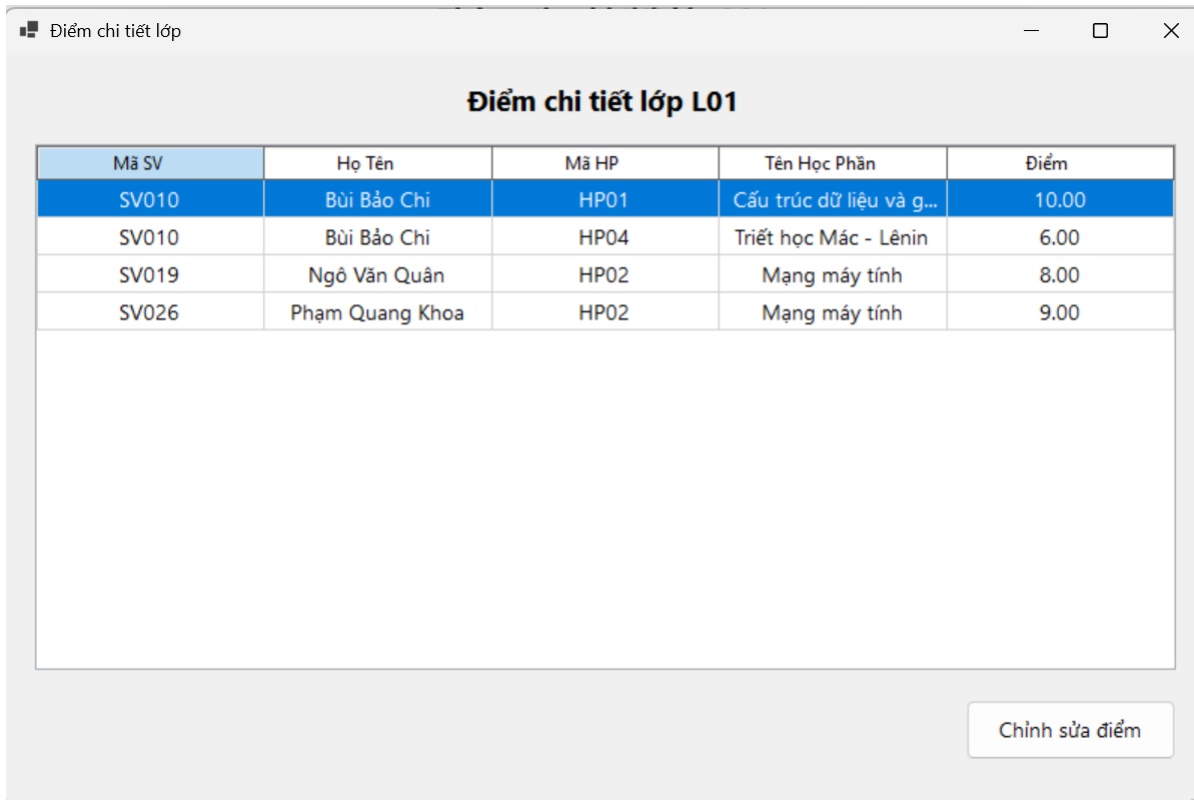
```

```

12      @ErrorMessage NVARCHAR(200) OUTPUT
13 AS
14 BEGIN
15     SET NOCOUNT ON;
16     IF NOT EXISTS(SELECT 1 FROM SINHVIEN WHERE MASV = @MASV AND MALOP =
@OLD_MALOP)
17     BEGIN
18         SET @ErrorMessage = 'Sinh vien khong ton tai trong lop hien tai';
19         RETURN;
20     END
21
22     IF @NEW_MALOP <> @OLD_MALOP
23     BEGIN
24         IF NOT EXISTS(SELECT 1 FROM LOP WHERE MALOP = @NEW_MALOP)
25         BEGIN
26             SET @ErrorMessage = 'Ma lop moi khong ton tai.';
27             RETURN;
28         END
29     END
30
31     UPDATE SINHVIEN
32     SET HOTEN = @HOTEN,
33         NGAYSINH = @NGAYSINH,
34         DIACHI = @DIACHI,
35         MALOP = @NEW_MALOP
36     WHERE MASV = @MASV AND MALOP = @OLD_MALOP;
37
38     SET @ErrorMessage = '';
39 END
40 GO

```

3.10 Màn hình Chi tiết điểm



Mã SV	Họ Tên	Mã HP	Tên Học Phần	Điểm
SV010	Bùi Bảo Chi	HP01	Cấu trúc dữ liệu và g...	10.00
SV010	Bùi Bảo Chi	HP04	Triết học Mác - Lênin	6.00
SV019	Ngô Văn Quân	HP02	Mạng máy tính	8.00
SV026	Phạm Quang Khoa	HP02	Mạng máy tính	9.00

Chỉnh sửa điểm

Hình 12: Giao diện chi tiết điểm của lớp

- Khi truy cập vào form thì chương trình sẽ chạy procedure **SP_GET_SCORES_BY_CLASS** với các tham số là mã lớp hiện tại và public key. Danh sách toàn bộ điểm của các sinh viên trong lớp đó sẽ hiện lên.
- Khi nhấn vào **Chỉnh sửa điểm**, chương trình sẽ mở ra giao diện để nhân viên thêm / chỉnh sửa điểm cho sinh viên cụ thể.

Store procedure **SP_GET_SCORES_BY_CLASS** nhận vào các giá trị là @MALOP, @PUBKEY và trả ra thông tin sinh viên cùng điểm ở dạng mã hóa (VARBINARY)

```

1 IF OBJECT_ID('SP_GET_SCORES_BY_CLASS', 'P') IS NOT NULL
2     DROP PROCEDURE SP_GET_SCORES_BY_CLASS;
3 GO
4
5 CREATE PROCEDURE SP_GET_SCORES_BY_CLASS
6     @MALOP VARCHAR(20),
7     @PUBKEY VARCHAR(20),
8     @ErrorMessage NVARCHAR(200) OUTPUT
9 AS
10 BEGIN
11     SET NOCOUNT ON;
12
13     IF NOT EXISTS (SELECT 1 FROM LOP WHERE MALOP = @MALOP)

```



```

14 BEGIN
15     SET @ErrorMessage = 'Ma lop khong ton tai!';
16     RETURN;
17 END
18
19 IF NOT EXISTS (SELECT 1 FROM NHANVIEN WHERE MANV = @PUBKEY)
20 BEGIN
21     SET @ErrorMessage = 'Ma nhan vien khong hop le!';
22     RETURN;
23 END
24
25 SELECT
26     S.MASV,
27     S.HOTEN,
28     B.MAHP,
29     H.TENHP,
30     B.DIEMTHI AS DIEM_ENCRYPTED
31 FROM BANGDIEM B
32 JOIN SINHVIENT S ON B.MASV = S.MASV
33 JOIN HOCPHAN H ON B.MAHP = H.MAHP
34 WHERE S.MALOP = @MALOP;
35
36 SET @ErrorMessage = '';
37 END;
38 GO

```

Điểm sẽ được giải mã ở phía C# để có thể hiển thị trên giao diện

```

1 foreach (DataRow row in dt.Rows)
2 {
3     try
4     {
5         byte[] encryptedBytes = row["DIEM_ENCRYPTED"] as byte[];
6         if (encryptedBytes != null)
7         {
8             string privateKey = EncryptionHelper.LoadPrivateKeyFromFile(
employeeId);
9             string decryptedText = EncryptionHelper.DecryptDataRSA(
encryptedBytes, privateKey);
10
11             if (decimal.TryParse(decryptedText, out decimal diem))
12             {
13                 row["DIEM"] = diem;
14             }
15             else
16             {
17                 row["DIEM"] = DBNull.Value;
18             }
19         }
20     }
21     catch (Exception ex)
22     {
23         row["DIEM"] = DBNull.Value;
24     }
25 }

```

3.11 Màn hình Thêm / chỉnh sửa điểm

Hình 13: Giao diện sửa điểm sinh viên

- Sau khi chọn chỉnh sửa và đã nhập đầy đủ thông tin để sửa điểm trong form điểm chi tiết, người dùng nhấn vào nút **Xác nhận** thì chương trình sẽ chạy procedure **SP_INS_DIEM** với các tham số được thêm vào tương ứng dưới database.
- Với sinh viên đã có điểm ở học phần đó, điểm sẽ được cập nhật mới. Còn với sinh viên chưa có điểm ở học phần đó, điểm sẽ được thêm vào.
- Nếu kiểm tra sự hợp lệ của input, Stored procedures sẽ thực hiện thêm / cập nhật điểm sinh viên vào database.
- Nút **Hủy** để xóa hết các thông tin người dùng vừa nhập.

Store procedure **SP_INS_DIEM** nhận vào các giá trị là @MASV, @MAHP, @DIEMTHI và trả ra @ErrorMessage nếu có lỗi. Điểm sẽ được mã hóa bằng C# trước khi lưu xuống database.

```

1 CREATE PROCEDURE SP_INS_DIEM
2     @MASV VARCHAR(20),
3     @MAHP VARCHAR(20),
4     @DIEMTHI VARBINARY(MAX),
5     @ErrorMessage NVARCHAR(200) OUTPUT
6 AS
7 BEGIN
8     BEGIN TRY
9         IF EXISTS (SELECT 1 FROM BANGDIEM WHERE MASV = @MASV AND MAHP = @MAHP)
10            BEGIN
11                UPDATE BANGDIEM

```

```

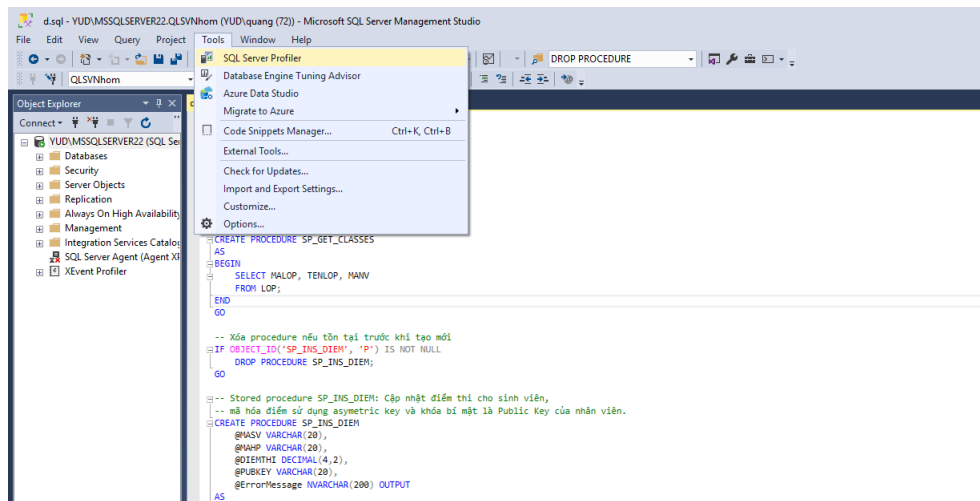
12         SET DIEMTHI = @DIEMTHI
13         WHERE MASV = @MASV AND MAHP = @MAHP;
14     END
15     ELSE
16     BEGIN
17         INSERT INTO BANGDIEM (MASV, MAHP, DIEMTHI)
18         VALUES (@MASV, @MAHP, @DIEMTHI);
19     END
20     SET @ErrorMessage = '';
21 END TRY
22 BEGIN CATCH
23     SET @ErrorMessage = ERROR_MESSAGE();
24 END CATCH
25 END;

```

4 Câu e

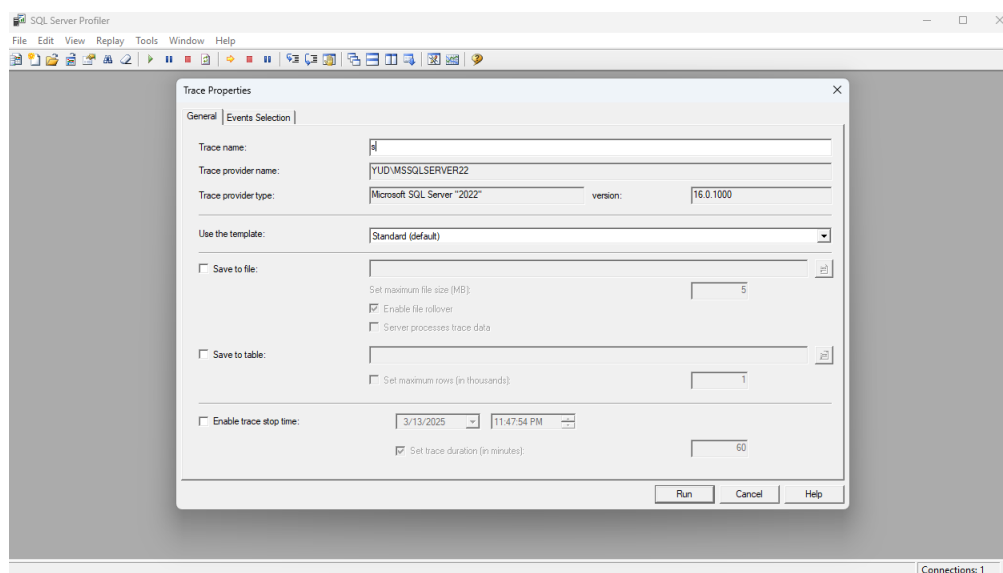
4.1 Cài Đặt SQL Server Profiler

1. Mở **SQL Server Management Studio (SSMS)**.
2. Chuyển đến **Tools** → **SQL Server Profiler**.
3. Kết nối đến SQL Server.



Hình 14: Kết nối SQL Profiler

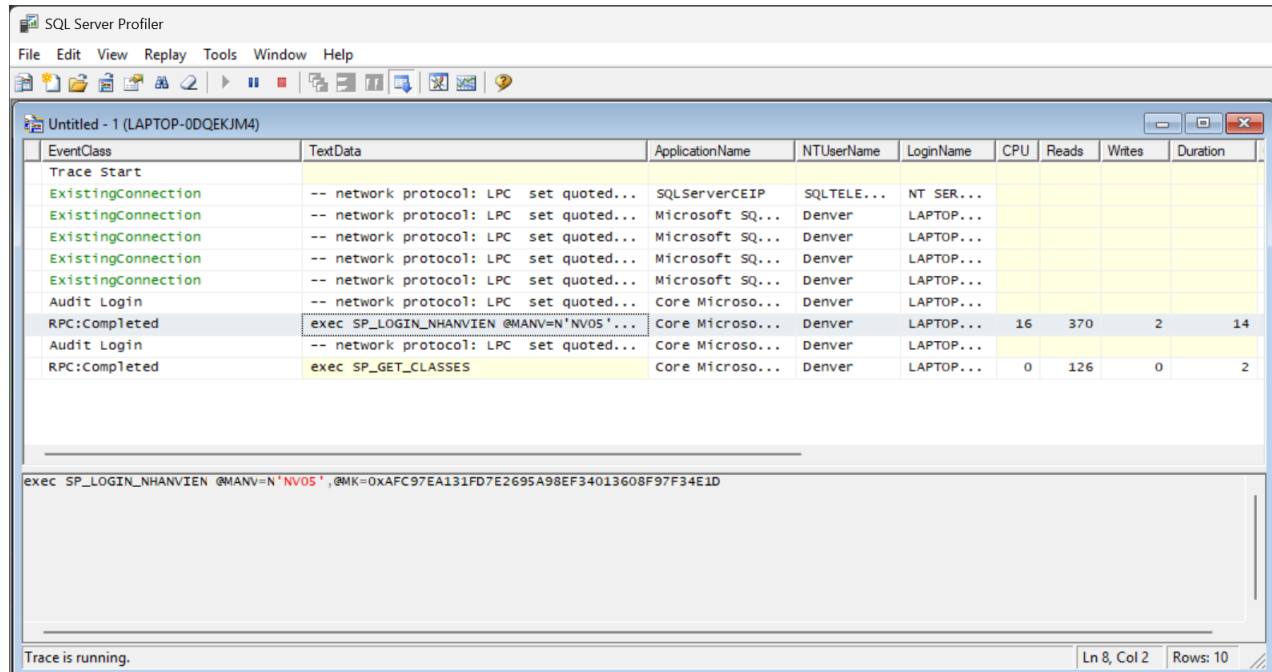
4.2 Cấu Hình SQL Profiler



Hình 15: Cấu hình Tracer

4.3 Theo dõi truy vấn

Khi thực hiện thao tác trên app, có tương tác với database, mọi thao tác sẽ được lưu lại trong file tracer đó. Ví dụ dưới là cho việc login của nhân viên.



EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration
Trace Start								
ExistingConnection	-- network protocol: LPC set quoted...	SQLServerCEIP	SQLTELE...	NT SER...				
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	Denver	LAPTOP...				
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	Denver	LAPTOP...				
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	Denver	LAPTOP...				
Audit Login	-- network protocol: LPC set quoted...	Core Microso...	Denver	LAPTOP...				
RPC:Completed	exec SP_LOGIN_NHANVIEN @MANV=N'NV05'...	Core Microso...	Denver	LAPTOP...	16	370	2	14
Audit Login	-- network protocol: LPC set quoted...	Core Microso...	Denver	LAPTOP...				
RPC:Completed	exec SP_GET_CLASSES	Core Microso...	Denver	LAPTOP...	0	126	0	2

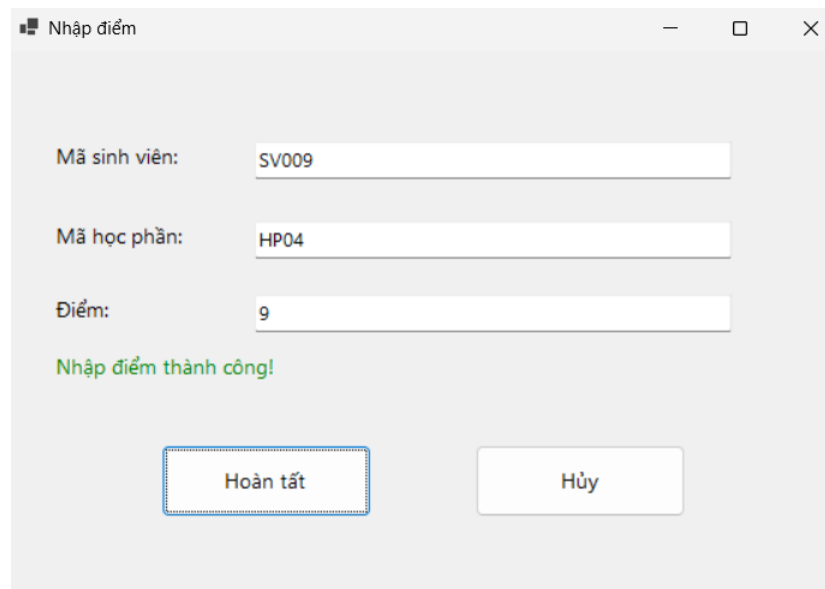
exec SP_LOGIN_NHANVIEN @MANV=N'NV05',@MK=0xAFC97EA131FD7E2695A98EF34013608F97F34E1D

Trace is running. Ln 8, Col 2 Rows: 10

Hình 16: Tracing

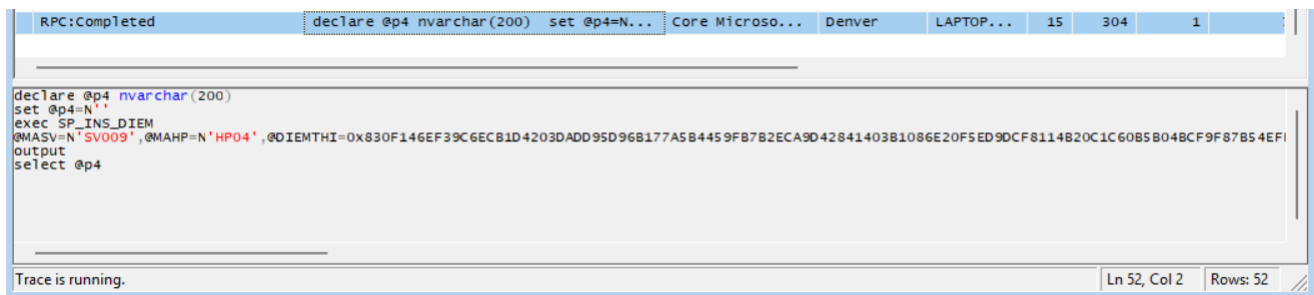
4.4 Thực hiện với form thêm điểm

Khi bật form, nhập dữ liệu và xác nhận, đoạn SQL Script sẽ được thực thi



Hình 17: Tracing

Sau đó đoạn SQL script đó cũng sẽ được lưu vào tracer. Tuy nhiên, điểm thi đã được mã hóa trong client trước khi gửi về server nên SQL Profiler không thể biết được điểm thi. Từ đó giúp hệ thống được bảo mật hơn.



Hình 18: Tracing