TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO THỰC TẬP NHẬN THỨC**

**TÊN ĐỀ TÀI**

**ỨNG DỤNG QUẢN LÝ CÁC GIAO DỊCH SỬA CHỮA THIẾT BỊ**

**Nhóm: 3**

**Sinh viên thực hiện:**

1. Trưởng nhóm: Nguyễn Thị Trúc Ly Lớp :45k21.1
2. Thành viên: Trần Nguyễn Quỳnh Hoanh Lớp : 45K21.1
3. Thành viên: Lê Thị Vân Lớp : 45K21.1

**Giảng viên hướng dẫn:** côCao Thị Nhâm

**Đà Nẵng, 6/2021**

**DANH SÁCH THÀNH VIÊN THAM GIA**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Họ và tên** | **Lớp** | **Nhiệm vụ** |
| 1 | Nguyễn Thị Trúc Ly | 45K21.1 | Trưởng nhóm |
| 2 | Trần Nguyễn Quỳnh Hoanh | 45K21.1 | Thành viên |
| 3 | Lê Thị Vân | 45K21.1 | Thành viên |

**LỜI CẢM ƠN**

*Thay mặt tất cả các thành viên của nhóm 3, chúng em xin cảm ơn cô Cao Thị Nhâm là giáo viên phụ trách thực tập đã tận tình chỉ dạy chúng em, trong suốt vài tuần qua. Mặc dù, tụi em không phải là học sinh xuất sắc, nhưng nhờ có sự hướng dẫn và quản lý từ cô chúng em nhận thấy được mình đã có sự phát triển trong quá trình làm việc cá nhân và nhóm. Thật sự, mặc dù tụi em có những lỗi sai và làm nhiệm vụ chưa hoàn thiện nhưng sự đáp lại từ cô khiến chúng em biết mình luôn được cô hướng dẫn hết mình. Ngoài ra, chúng em cảm ơn khoa và các giảng viên khác đã tạo ra chương trình thực tập hè này, để tụi em có cơ hội tiếp thu những nguồn kiến thức mới, nhận được nhiều kinh nghiệm làm dự án nhóm. Cuối cùng, em xin một lần nữa cảm ơn thầy cô rất nhiều ạ.*

**LỜI CAM ĐOAN**

Chúng em xin cam đoan dự án “ỨNG DỤNG QUẢN LÍ GIAO DỊCH SỬA CHỮA THIẾT BỊ” là kết quả nghiên cứu độc lập dưới sự hướng dẫn của giáo viên hướng dẫn: ThS.Cao Thị Nhâm. Ngoài ra, không có bất cứ sự sao chép nào của người khác.

MỤC LỤC:

[- MỤC LỤC 2](#_Toc76744356)

[- LỜI MỞ ĐẦU 1](#_Toc76744357)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 4](#_Toc76744358)

[1.1.1. Ngôn ngữ lập trình C# 4](#_Toc76744359)

[1.1.2. Nền tảng .NET 5](#_Toc76744360)

[1.2. Tổng quan về Windows Form 5](#_Toc76744361)

[1.3. Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server 5](#_Toc76744362)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ ỨNG DỤNG 7](#_Toc76744363)

[2.1. Mô tả bài toán 7](#_Toc76744364)

[2.1.1.Thực Trạng 7](#_Toc76744365)

[2.1.2 Giải pháp 7](#_Toc76744366)

[2.1.3. Các chức năng của hệ thống 7](#_Toc76744367)

[**2.1.4.** **Môi trường triển khai** 7](#_Toc76744368)

[2.2. Thiết kế cơ sở dữ liệu 8](#_Toc76744369)

[2.3. Tạo cơ sở dữ liệu. 11](#_Toc76744370)

[2.3.1. Tạo bảng KHACH\_HANG 11](#_Toc76744371)

[2.3.2. Tạo bảng NHAN\_VIEN 12](#_Toc76744372)

[2.3.3. Tạo bảng SUA\_CHUA 12](#_Toc76744373)

[2.3.4 Tạo bảng SUA\_CHUA\_CHI\_TIET 13](#_Toc76744374)

[2.3.5 Tạo bảng CONG\_VIEC 14](#_Toc76744375)

[2.4. Mô hình quan hệ. 15](#_Toc76744376)

[**2.5.** Thông tin chi tiết các form làm việc. 15](#_Toc76744377)

[**2.5.1.** **Form quản lý khách hàng** 15](#_Toc76744378)

[**2.5.2.** **Form quản lý nhân viên** 15](#_Toc76744379)

[**2.5.3.** **Form quản lý giao dịch sửa chữa (form hóa đơn)** 16](#_Toc76744380)

[**2.5.4.** **Form thống kê** 16](#_Toc76744381)

[CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG 17](#_Toc76744382)

[3.1. Sơ đồ phân rã chức năng. 17](#_Toc76744383)

[**3.2.1.** **Form NHAN\_VIEN (Hoanh)** 18](#_Toc76744384)

[**3.2.2.** **Form KHACH\_HANG (Vân)** 18](#_Toc76744385)

[**3.2.3.** **Form SUA\_CHUA (Ly)** 20](#_Toc76744386)

[**3.2.4.** **Form Thống Kê (Hoanh+Vân)** 21](#_Toc76744387)

[CHƯƠNG 4. KẾT QUẢ 22](#_Toc76744388)

[- PHỤ LỤC: Phần code của nhóm cho ứng dụng 22](#_Toc76744389)

[- KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 64](#_Toc76744390)

[- TÀI LIỆU THAM KHẢO 65](#_Toc76744391)

[- PHỤ LỤC (nếu có) 66](#_Toc76744392)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_heading=h.z337ya)

[Hình 4.1 Kiến trúc của mô hình RNN 5](#_heading=h.2grqrue)

DANH MỤC BẢNG BIỂU

Bảng 2.1. Bảng khách hàng

Bảng 2.2. Bảng nhân viên

Bảng 2.3. Bảng sửa chữa

DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**CSDL: cơ sở dữ liệu**

**SĐT: Số điện thoại**

# LỜI MỞ ĐẦU

1. **Mục tiêu của đề tài**

* Nghiên cứu ngôn ngữ lập trình C# (Windows Forms)
* Phát triển ứng dụng quản lý giao dịch sửa chữa thiết bị

1. **Nhiệm vụ của đề tài**

* Học ngôn ngữ lập trình c# trên nền tảng Windows forms.
* Xây dựng cơ sở dữ liệu cho ứng dụng quản lý giao dịch sửa chữa thiết bị.
* Xây dựng ứng dụng quản lý giao dịch sửa chữa thiết bị trên nền tảng .NET

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm 4 chương nội dung và phần kết luận.

* **Chương 1**: Cơ sở lý thuyết
* **Chương 2**: Phân tích và thiết kế ứng dụng
* **Chương 3**: Phát triển ứng dụng
* **Chương 4**: Kết quả
* Kết luận và hướng phát triển

# BẢNG PHÂN CÔNG CÔNG VIỆC

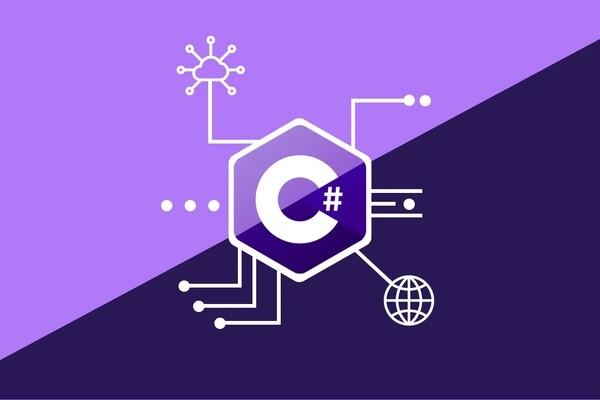
**Nhóm 3:**

**Tên đề tài: “Ứng dụng các quản lý giao dịch sửa chữa thiết bị”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Công việc hoàn thành** | **Chú thích** | **Phần trăm** |
| **1** | **Nguyễn Thị Trúc Ly** | - Tạo form quản lý giao dịch ( form sửa chữa)  -Tạo cơ sở dữ liệu SQL  - Code form hóa đơn sửa chữa  - Viết báo cáo  - Quay video thuyết trình | **Hoàn thành**  **đúng thời gian** | **36%** |
| **2** | **Lê Thị Vân** | -Tạo form quản lý khách hàng + code form thống kê  - Code form Quản lý khách hàng + code form thống kê  - Làm powerpoint | **Hoàn thành đúng thời gian** | **32%** |
| **3** | **Trần Nguyễn Quỳnh Hoanh** | -Tạo form quản lý nhân viên  - Tạo cơ sở dữ liệu SQL  - Code form Quản lý nhân viên + code form thống kê  - Làm powerpoint | **Hoàn thành đúng thời gian** | **32%** |

1. **CƠ SỞ LÝ THUYẾT**
   1. Tổng quan về .NET framework

### Ngôn ngữ lập trình C#



*Hình 1.1 Ngôn ngữ lập trình C#*

Ngôn ngữ lập trình C# (Hình 1.1) là ngôn ngữ  [lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) đa năng, mạnh mẽ được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), C# là phần khởi đầu cho kế hoạch [.NET](https://vi.wikipedia.org/wiki/.NET_Framework) của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo [ECMA](https://vi.wikipedia.org/w/index.php?title=ECMA&action=edit&redlink=1) là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên [C++](https://vi.wikipedia.org/wiki/C%2B%2B) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)).

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

### 1.1.2. Nền tảng .NET

- **.NET Framework** được Microsoft đưa ra chính thức từ năm 2002. .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP.NET(1-4) hoạt động dựa trên .NET Framework.

- Mono là phiên bản cộng đồng nhằm mang .NET đến những nền tảng ngoài Windows. Mono được phát triển chủ yếu nhằm xây dựng những ứng dụng với giao diện người dùng và được sử dụng rất rộng rãi: Unity Game, Xamarin…

- Cho đến năm 2013, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core. Từ đây, C# có thể được sử dụng để phát triển các loại ứng dụng đa nền tảng trên các hệ điều hành khác nhau (Windows, Linux, MacOS,…)

## Tổng quan về Windows Form

**Windows Form** là nơi cung cấp thành phần giao diện môi trường cho NET Framework, được xây dựng trên thư viện Windows API và sử dụng ngôn ngữ lập trình C#.

**Windows form cơ bản gồm:**

-Một Form là khung dùng hiển thị thông tin đến người dùng.

-Các Control được đặt trong form và được lập trình để đáp ứng sự kiện.

## Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server

**SQL server chính là** một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System \_ RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và và những dữ liệu khác.

* Các thành phần cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service, ... Tất cả hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.
* SQL Server không phải là một hệ quản trị độc lập mà nó chỉ là một thành phần với vai trò ngôn ngữ là công cụ giao tiếp giữa hệ cơ sở dữ liệu và người dùng. Chính vì thế nó được sử dụng trong các dịch vụ thiết kế web đẹp với chức năng giao tiếp với người dùng với vai trò sau:
* SQL là một ngôn ngữ đòi hỏi tính tương tác cao: Người dùng có thể dễ dàng trao đổi với các tiện ích thông qua các câu lệnh của SQL đến cơ sở dữ liệu và nhân kết quả từ cơ sở dữ liệu.
* SQL là một ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình có thể xây dựng các chương trình ứng dụng giao tiếp với cơ sở dữ liệu bằng cách nhúng các câu lệnh SQL vào trong ngôn ngữ lập trình.
* SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu: Người quản trị cơ sở dữ liệu có thể quản lý, định nghĩa và và điều khiển truy cập cơ sở dữ liệu thông qua SQL.
* SQL là một ngôn ngữ lập trình cho các hệ thống chủ khách: SQL được sử dụng như một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách chủ.
* SQL là ngôn ngữ lập trình cho các hệ thống chủ khách: SQL được sử dụng như một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách chủ web và máy chủ internet.
* SQL là ngôn ngữ cơ sở dữ liệu phân tán: Với vai trò giao tiếp với các hệ thống trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

1. **PHÂN TÍCH THIẾT KẾ ỨNG DỤNG**

## Mô tả bài toán

### 2.1.1.Thực Trạng

Công ty máy tính Kim Anh là công ty chuyên về mua bán máy tính cũ và sửa chữa các thiết bị di động như điện thoại, máy tính bảng. Hiện tại công ty đang cần xây dựng một ứng dụng để quản lý các giao dịch sửa chữa của các thiết bị.

### 2.1.2 Giải pháp

Xây dựng hệ thống quản lý các giao dịch sửa chữa. Giúp người dùng sử dụng hệ thống làm thuận tiện công tác: Quản lý hàng hóa, quản lý thông tin nhân viên, quản lí thông tin khách hàng, quản lý các hóa đơn…Từ đó giúp quản lý các giao dịch diễn ra một cách thông minh và dễ dàng hơn.

### 2.1.3. Các chức năng của hệ thống

* Hiển thị
* Tìm kiếm
* Thêm
* Cập nhật
* In (hóa đơn)
* Thống kê doanh thu
  + 1. **Môi trường triển khai**

• Hệ Quản Trị cơ sở dữ liệu SQL Server 2016 Management Studio

• Nền Visual studio 2019

• Ngôn ngữ C#

• Phần mềm GitHub.

## 2.2. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **KHACH\_HANG** | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Mô tả** |
| **MaKH** | Varchar(15) | Khóa chính | Dùng số đt để làm mã khách hàng |
| **TenKH** | Nvarchar(100) |  |  |
| **DiaChi** | Nvarchar(100) |  |  |
| **ThanThiet** | Int |  | Lưu điểm khách hàng thân thiết |

|  |  |  |  |
| --- | --- | --- | --- |
| **NHAN\_VIEN** | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Mô tả** |
| **MaNV** | Int | Khóa chính |  |
| **TenNV** | Nvarchar(100) |  |  |
| **SoDT** | Nvarchar(100) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SUA\_CHUA** | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Mô tả** |
| **MaHD** | Varchar(20) | Khóa chính |  |
| **MaKH** | Varchar(15) |  |  |
| **MaNV** | Int |  |  |
| **GioNhan** | Time(7) |  | Giờ nhận thiết bị |
| **NgayNhan** | Date |  | Ngày nhận thiết bị |
| **NayTra** | Date |  | Ngày trả thiết bị |
| **GioTra** | Time(7) |  | Giờ trả thiết bị |
| **TrangThai** | Nvarchar(50) |  | Đánh dấu trạng thái của giao dịch: Xong, Đang sửa, Hủy sửa |
| **TongTien** | Int |  | Tổng tiền thanh toán cả hóa đơn |

**SUA\_CHUA\_CHI\_TIET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Mô tả** |
| **MaHD** | Varchar(20) | Khóa chính |  |
| **MaCV** | Varchar(50) | Khóa chính |  |
| **SoLuong** | Int |  |  |
| **ThanhTien** | Numeric(15, 0) |  | Giá sửa chữa |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Mô tả** |
| **MaCV** | **Varchar(50)** | **Khóa chính** |  |
| **NoiDung** | **Nvarchar(500)** |  | Dịch vụ khách hàng muốn thực hiện |
| **ThanhTien** | **Numeric(15,0)** |  | Giá sửa chữa |

## 2.3. Tạo cơ sở dữ liệu.

**Create database QLGD\_suachua**

**Use QLGD\_suachua**

### 2.3.1. Tạo bảng KHACH\_HANG

**Create table KHACH\_HANG**

**(**

**MaKH Varchar(15) primary key not null,**

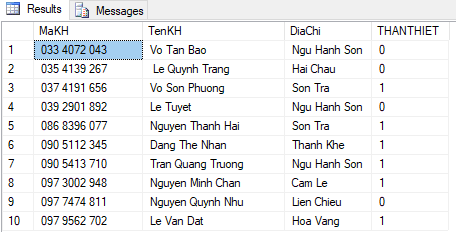
**TenKH NVARCHAR(1002) not null,**

**DiaChi Nvarchar(100) not null,**

**THANTHIET int not null**

**)**

Sau khi nhập dữ liệu và chạy ra kết quả ta được bảng **KHACH\_HANG** như sau:



*Bảng 2.1: Bảng khách hàng*

### 2.3.2. Tạo bảng NHAN\_VIEN

**Create table NHAN\_VIEN**

**(**

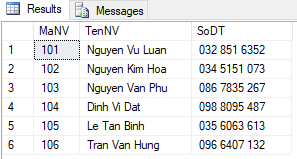
**MaNV INT not null Primary key,**

**TenNV Nvarchar(100) not null,**

**SoDT NVARCHAR(100) not null**

**)**

Sau khi nhập dữ liệu và chạy ra kết quả ta được bảng **NHAN\_VIEN** như sau:



*Bảng 2.2. Bảng nhân viên.*

### 2.3.3. Tạo bảng SUA\_CHUA

Create table SUA\_CHUA

(MaHD varchar (20) not null primary key,

MaKH varchar (15) not null,

MaNV int not null,

GioNhan time not null,

NgayNhan date not null,

NgayTra date ,

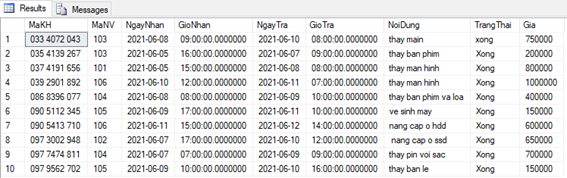
GioTra time ,

TrangThai Nvarchar (50) not null,

TongTien int not null

)

Sau khi nhập dữ liệu và chạy ra kết quả ta được bảng **SUA\_CHUA** như sau:



*2.3 Hình mô tả bảng sửa chữa*

### 2.3.4 Tạo bảng SUA\_CHUA\_CHI\_TIET

Create table SUA\_CHUA\_CHI\_TIET

( MaHD varchar (20) not null,

MaCV varchar (50) not null,

NoiDung Nvarchar (500) not null,

SoLuong int not null,

ThanhTien Numeric (15) not null,

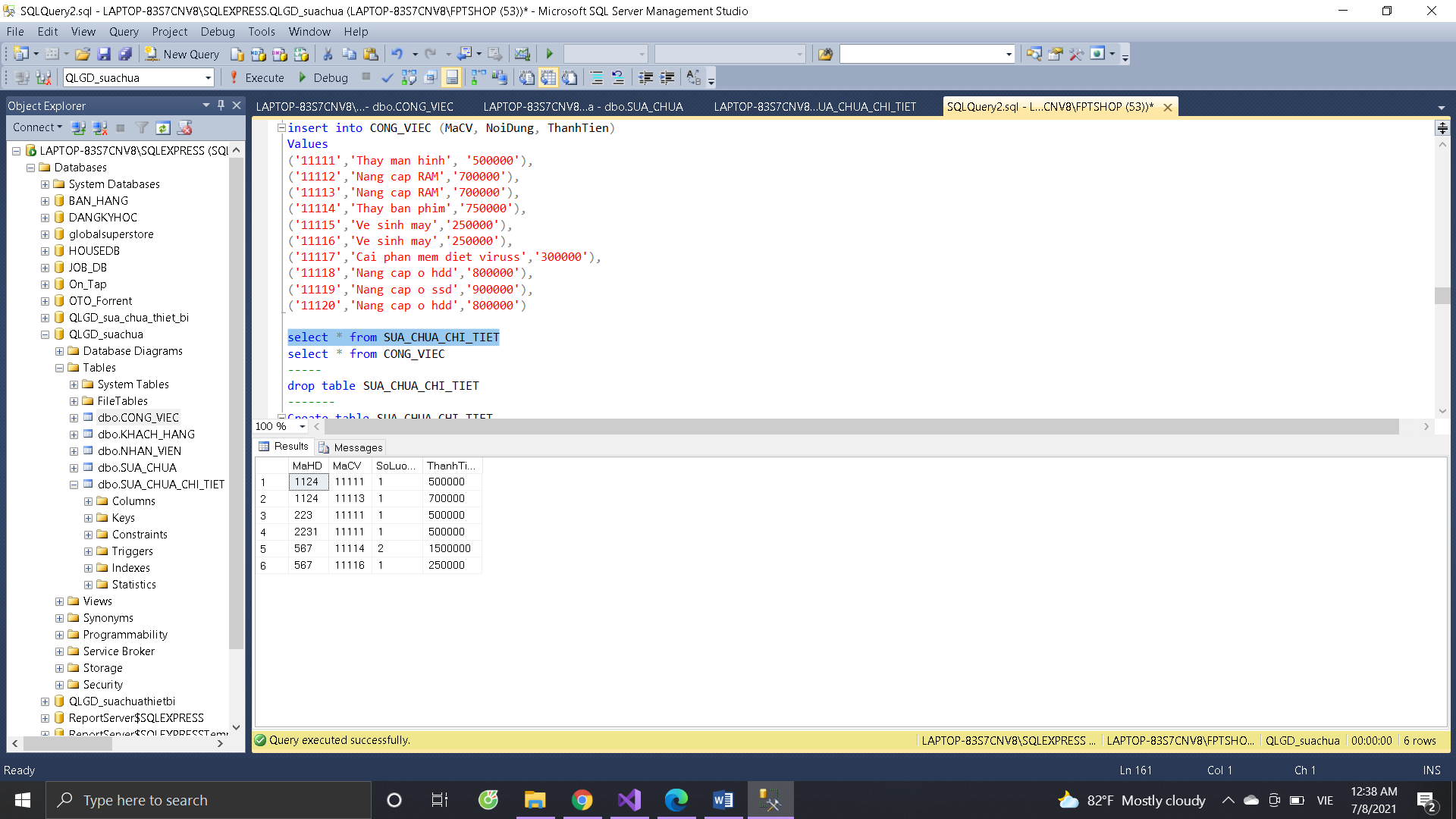
primary key (MaHD, MaCV),

foreign key (MaHD) references SUA\_CHUA(MaHD),

foreign key (MaCV) references CONG\_VIEC(MaCV)

)

Sau khi nhập dữ liệu và chạy ra kết quả ta được bảng **SUA\_CHUA\_CHI\_TIET** như sau:



*2.4 Hình mô tả bảng sửa chữa chi tiết*

### 2.3.5 Tạo bảng CONG\_VIEC

Create table CONG\_VIEC

(MaCV varchar (50) not null,

NoiDung Nvarchar (500) not null,

ThanhTien Numeric (15) not null,

primary key (MaCV)

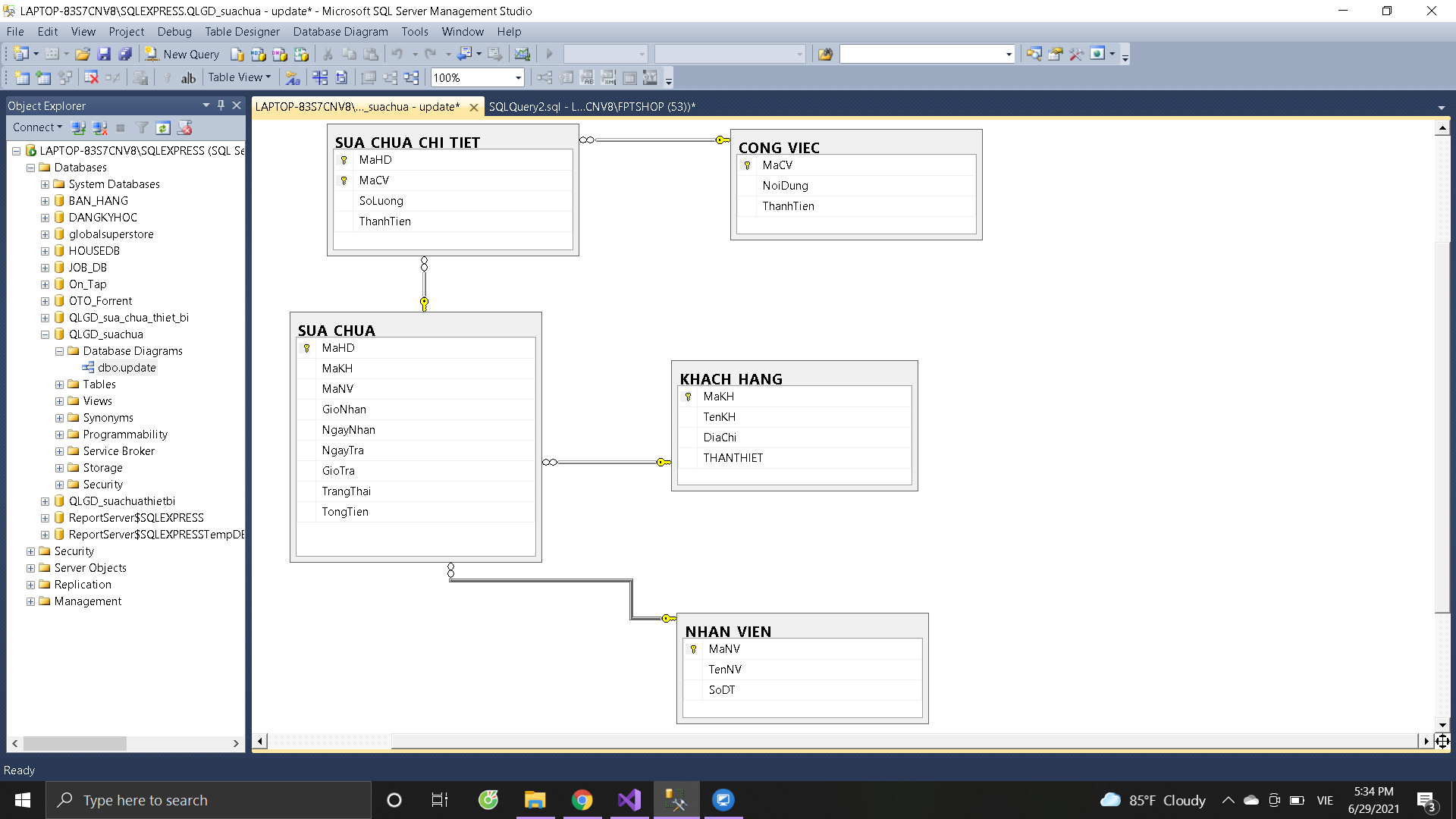
)

Sau khi nhập dữ liệu và chạy ra kết quả ta được bảng **CONG\_VIEC** như sau:



*2.5 Hình mô tả bảng công việc*

## 2.4. Mô hình quan hệ.

****

*Sơ đồ 1: Sơ đồ quan hệ*

* 1. Thông tin chi tiết các form làm việc.

Ứng dụng quản lý hoạt động giao dịch sửa chữa thiết bị mới mục đích giúp người dùng xử lý thông tin chính xác và tiết kiệm thời gian hơn. Thông qua các ứng dụng, người quản lý có thể kiểm soát các thông tin bao gồm: Quản lý khách hàng, quản lý nhân viên, quản lý giao dịch sửa chữa và thống kê doanh thu.

* + 1. **Form quản lý khách hàng**
* Hiển thị đầy đủ danh sách của khách hàng.
* Thực hiện các chức năng: Tìm kiếm, hiển thị, thêm mới, sửa, xóa thông tin của khách hàng.
  + 1. **Form quản lý nhân viên**
* Hiển thị đầy đủ danh sách của nhân viên.
* Thực hiện các chức năng: Tìm kiếm, hiển thị, thêm mới, sửa, xóa thông tin của nhân viên.
  + 1. **Form quản lý giao dịch sửa chữa (form hóa đơn)**
* Hiển thị các thông tin giao dịch có trong hóa đơn.
* Thực hiện các chức năng: Tìm kiếm, hiển thị, thêm mới, sửa, xóa thông tin và in hóa đơn.
  + 1. **Form thống kê**
* Thống kê doanh thu của công ty theo thời gian và tình trạng sản phẩm.

1. **PHÁT TRIỂN ỨNG DỤNG**

## Sơ đồ phân rã chức năng.

Quản lý giao dịch sửa chữa

Quản lý khách hàng

Xử lý đơn hàng

Nội bộ

Cập nhật thông tin khách hàng

Thống kê

Quản lý nhân viên

Thống kê doanh thu

Cập nhật thông tin nhân viên

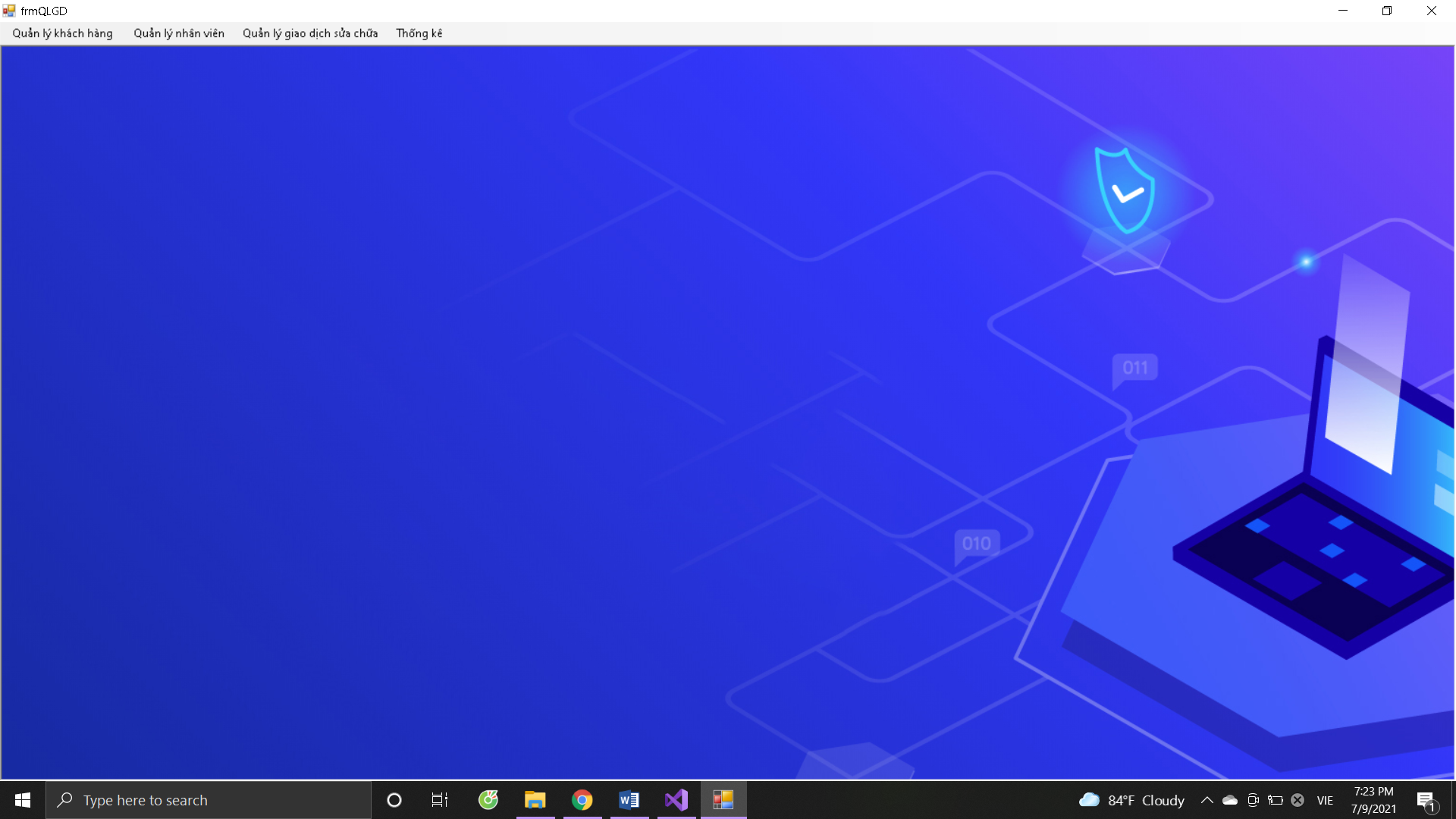
- Ứng dụng quản lý giao dịch sửa chữa của cửa hàng KIM ANH sẽ bao gồm 3 phần chính:

+ Xử lý đơn hàng: là giao diện đầu tiên khi khởi động ứng dụng, giúp dễ dàng kiểm soát thông tin các đơn hàng hơn.

+ Nội bộ: sẽ bao gồm 2 nội dung là quản lý nhân viên của cửa hàng và thống kê doanh thu.

+ Thông tin khách hàng: cập nhật, thêm, xóa, sửa các thông tin của khách hàng khi đến giao dịch.

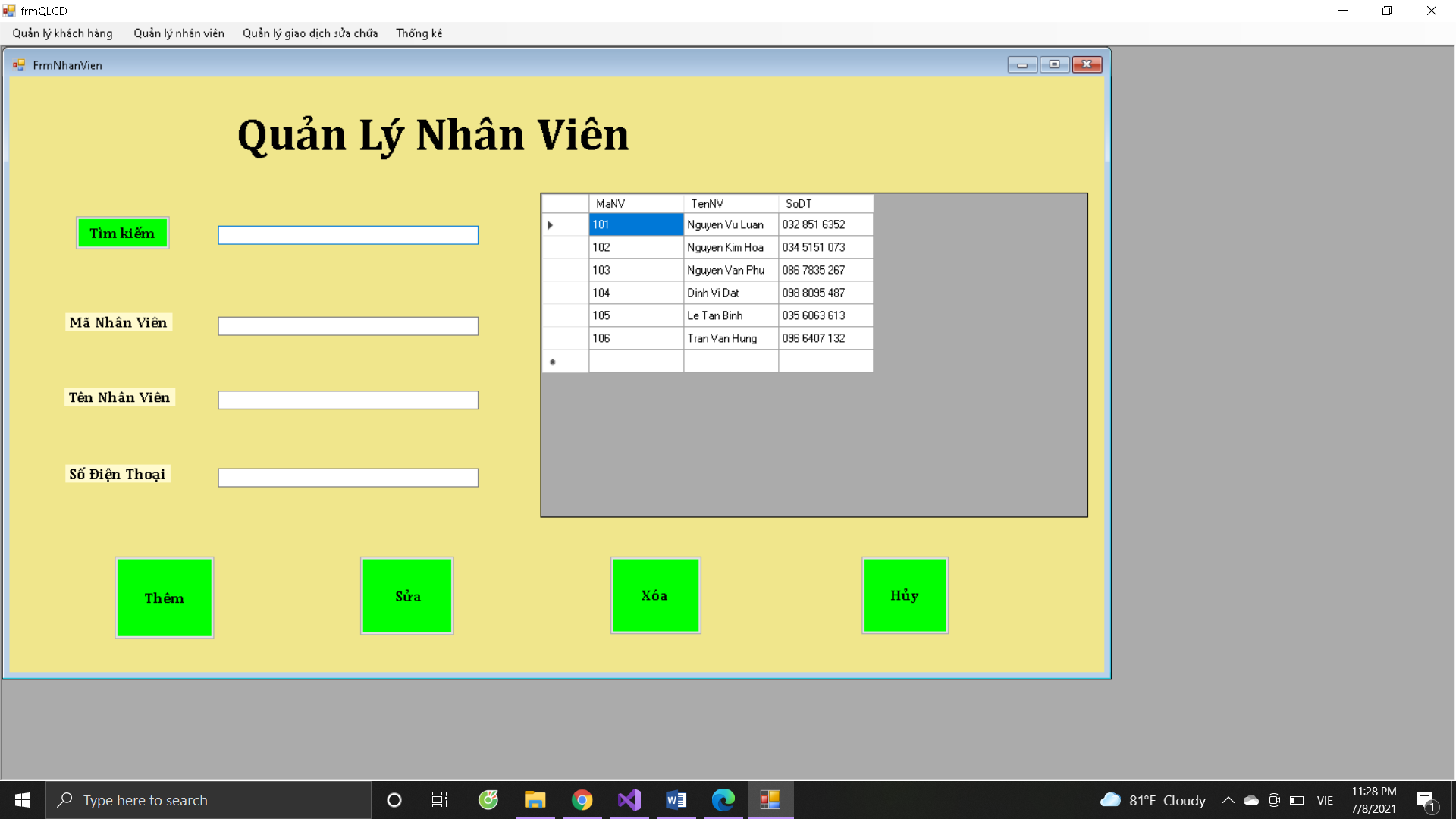
* 1. **Thông tin chi tiết các form làm việc**
     1. **Form TỔNG\_QLGD**



*Hình 3.2.1: Giao diện form tổng*

* + 1. **Form NHAN\_VIEN (Hoanh)**

Để kiểm tra thông tin và cập nhập thông tin của nhân viên -> Click **Quản lý nhân viên**

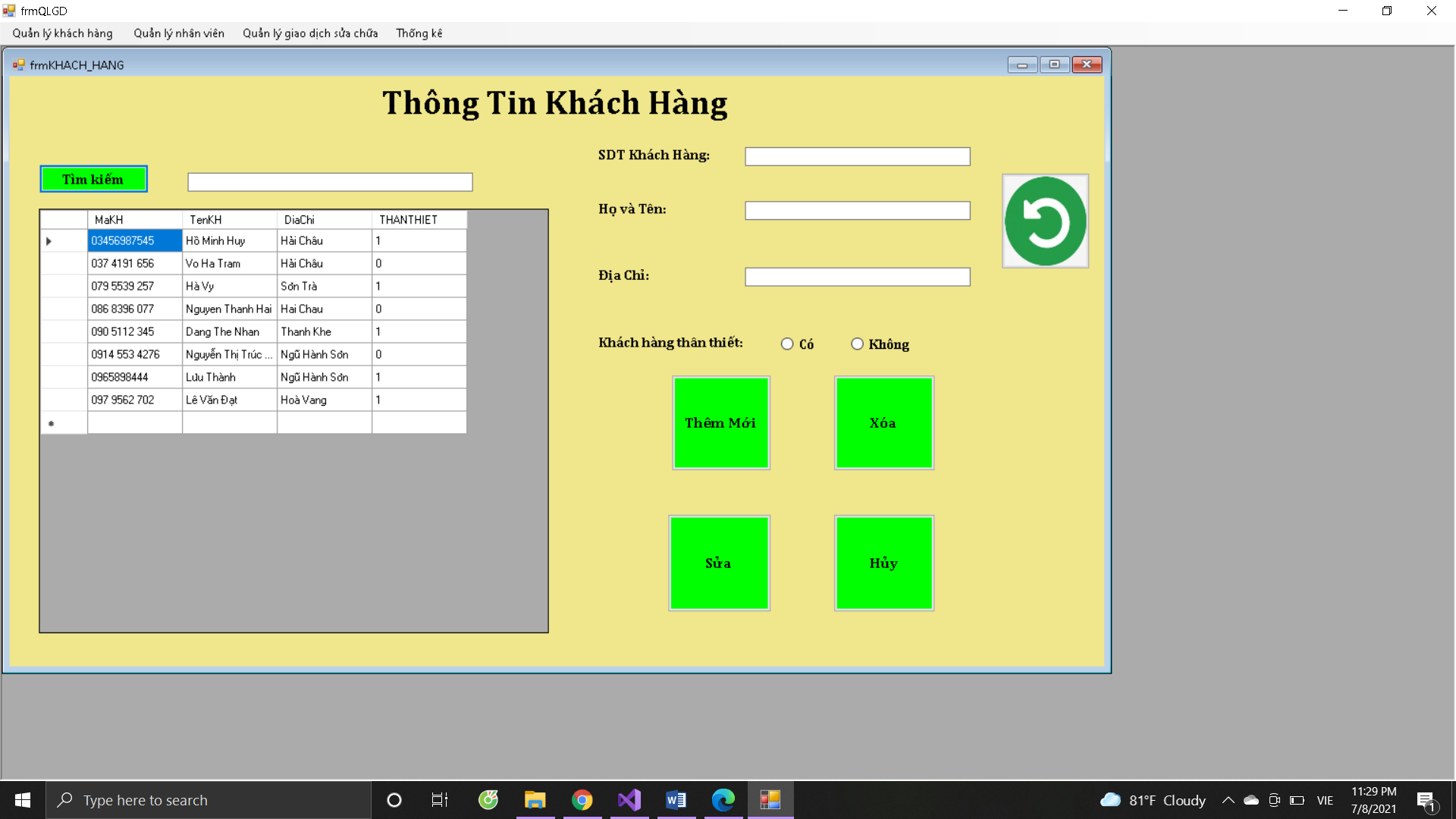


*Hình 3.2.2. Giao diện form Quản lý nhân viên*

Người dùng lần lượt cập nhật thông tin của nhân viên vào các mục sau “Mã nhân viên”, “Tên nhân viên”, “Số điện thoại”.

Đối với “Mã nhân viên” đây là mục bắt buộc tất cả nhân viên đều phải điền vào vì ”Mã nhân viên” có chức năng xác nhận nhân viên nào đã thực hiện cuộc giao dịch này, ai là người phụ trách xử lý,...tránh gây khó khăn trong truy cứu trách nhiệm.

* Để nhận biết ai là người nhận sử lý công việc này thì vào mục “Tìm kiếm”, sau đó gõ “Tên Nhân Viên”, thông tin sẽ được hiển thị ở màn hình bên dưới.
* “Thêm”: Thêm thông tin của nhân viên mới.
* “Cập nhật”: Cập nhật, sửa thông tin của nhân viên.
* “Xóa”: Xóa tất cả các thông tin của nhân viên đó.
* “Hủy”: Nhấn “Hủy/ESC” để thoát khỏi giao diện.
* Tất cả thông tin đều được hiển thị lên bảng dữ liệu.
  + 1. **Form KHACH\_HANG (Vân)**

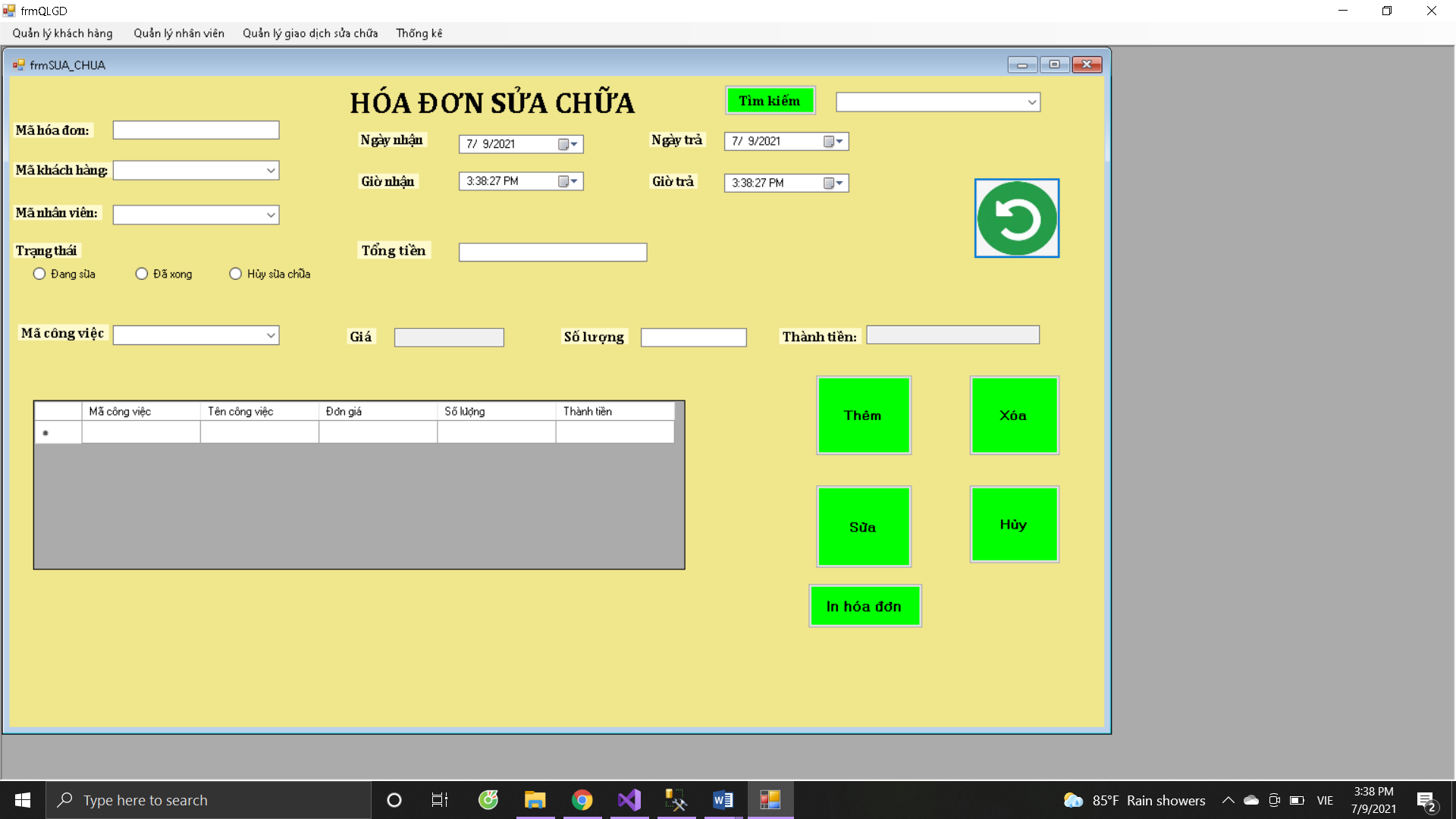
Tại mục “Quản lý khách hàng”, ta có thể xem thông tin từng khách hàng cũng như thêm, sửa, xóa và tìm kiếm khách hàng:

*Hình 3.2.3 Giao diện form khách hàng*

Để cập nhật và kiểm tra thông tin của khách hàng. Click vào mục “Quản lý khách hàng”.

Người dùng lần lượt cập nhật các thông tin cần thiết như: “SĐT Khách hàng” chính là Mã Khách Hàng; “Họ và tên”; “Địa chỉ”; “Khách hàng thân thiết”, đây là điểm số của khách hàng tích lũy được khi nhận được dịch vụ sửa chữa từ cửa hàng, 1 lần giao dịch khách hàng sẽ được 1 điểm, đủ số điểm tích lũy khách hàng sẽ nhận được nhiều ưu đãi từ công ty.

* “Tìm kiếm”: Nhập “Tên khách hàng” để tra nhanh thông tin của khách hàng.
* “Thêm”: Nhấn “Thêm/ENTER để cập nhật thông tin.
* “Xóa”: Xóa thông tin của khách hàng
* “Cập nhật”: Sửa thông tin của khách hàng.
* “Hủy”: Nhấn “Hủy/X” để thoát khỏi giao diện.
* Tất cả thông tin đều được hiển thị lên bảng dữ liệu datagridview.
  + 1. **Form HÓA ĐƠN SỬA CHỮA (Ly)**



*Hình 3.2.4: Giao diện form hóa đơn sữa chữa*

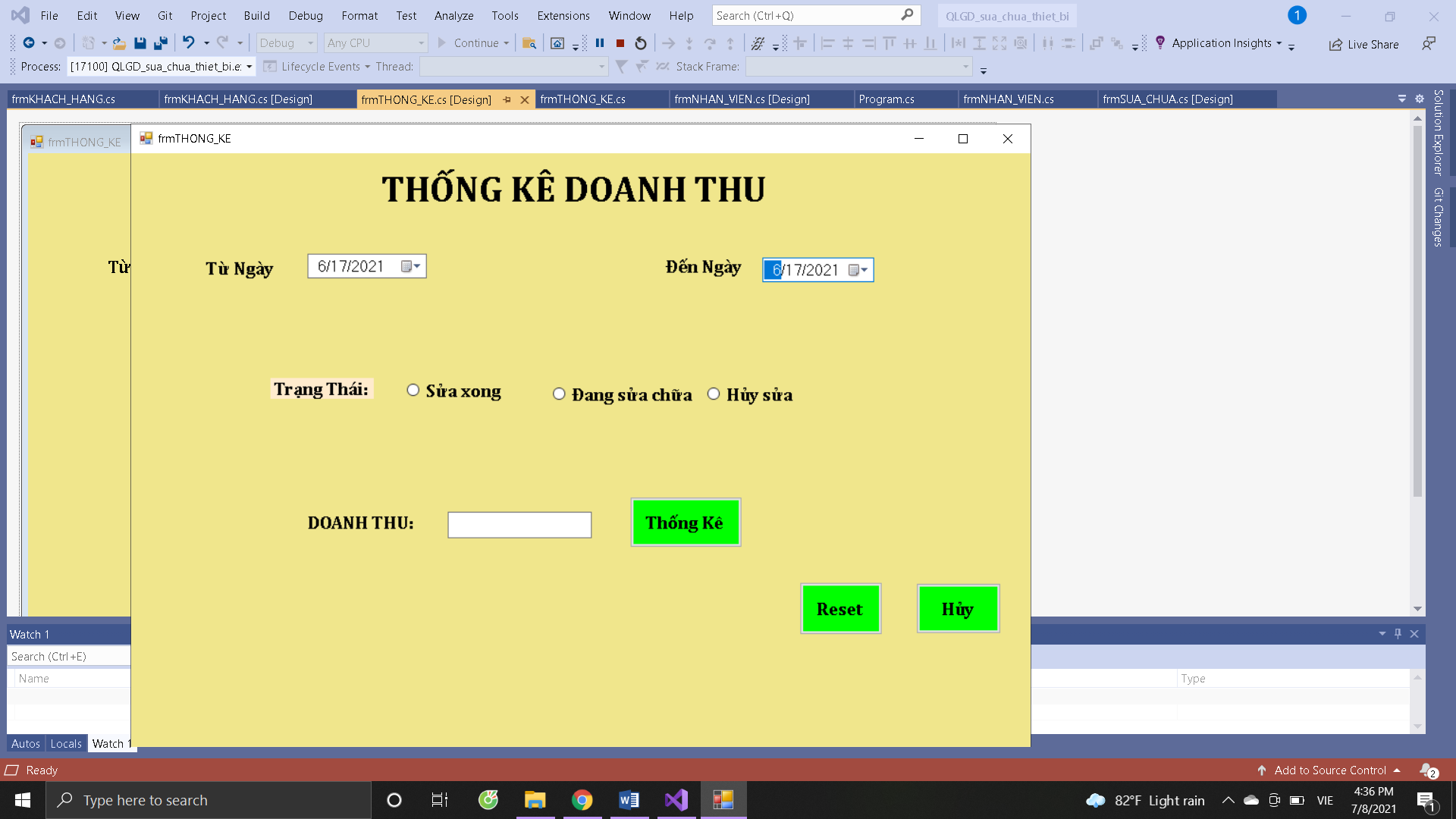
Form “quản lý giao dịch sửa chữa” là giao diện chứa các thông tin giao dịch của khách hàng với công ty Kim Anh. Theo đó, form có chức năng thực hiện các công việc như hiển thị thông tin giao dịch, cập nhật, sửa, xóa, in hóa đơn cho các đơn hàng.

Cụ thể như sau, trước trước khi tiếp nhận được số điện thoại khách hàng “Mã khách hàng”, trước tiên phải tìm được “Mã hóa đơn” để tìm lại đơn hàng mà khách hàng đã đặt sửa chữa. Click vào nút “Tìm kiếm” để tìm ra lịch sử giao dịch giữa “khách hàng” và “nhân viên” gồm thông tin giao dịch như “Tên khách hàng”, “Tên nhân viên”, “Nội dung công việc”, “giá tiền”, “số lượng”, “trạng thái”, “ngày nhận đơn”, “ngày trả”, ...

* Nếu đơn hàng đã được lưu sẵn trong máy, nhân viên chỉ cần nhấn vào mục tìm kiếm, tra “mã hóa đơn” mà khách hàng đã thực hiện giao dịch trước đó.
* Nếu thông tin chưa tồn tại, nhân viên có nhiệm vụ cập nhật đầy đủ thông tin cho form “Quản lý sửa chữa”. Nhập đầy đủ các thông tin bao gồm: “Mã hóa đơn”, “Mã khách hàng”, “Mã nhân viên”, “Mã khách hàng”,”Mã công việc”,”Trạng thái”,”Giờ nhận”, “Ngày nhận”, “Giờ trả”, “Ngày trả”,”Mã công việc”... sau đó nhập “số lượng” hệ thống tự động dựa vào “giá” để tính “tổng tiền” kháh hàng cần thanh toán.
* sau đó click vào nút **“THÊM”,** để thêm mới thông tin về đơn hàng vào trong dữ liệu.
* “Sửa”: Các thông tin sửa là hỗ trợ cho cập nhật thông tin của đơn hàng. Các thông tin bao gồm: “Trạng thái” của đơn hàng, “Ngày trả”, “Giờ trả”.
* “Xóa”: Gõ “SDT của khách hàng”, hoặc “Mã đơn hàng” để tìm kiếm đơn hàng và xóa.
* “Hủy”: Đóng form “quản lý sửa chữa”.
* “In hóa đơn”: Sau khi xử lý thông tin đơn hàng, tiến hành in hóa đơn giao cho khách hàng.
* Mọi thông tin được hiển thị lên bảng dữ liệu.

Sau đó, cập nhật thông tin của khách hang

* + 1. **Form Thống Kê (Hoanh+Vân)**



*Hình 3.2.5: Giao diện form Thống kê*

Tại mục Thống Kê, ta có thể dựa vào hóa đơn để thống kê doanh thu của công ty theo tháng hay 1 khoảng thời gian nhất định.

Form thực hiện tính toán doanh số của công ty trong khoảng thời gian nhất định -> Click “Thống kê doanh thu”

* Chọn “Ngày bắt đầu” và chọn “Ngày kết thúc”, click chọn trạng thái ‘sửa xong’, “đang sửa” hoặc “hủy sửa”.
* Sau đó click chọn “Thống Kê”. Thông tin hiển thị là “Tổng Tiền”.
* “Reset”: khi muốn nhập lại thông tin khác.
* “Hủy”: Đóng giao diện “Thống kê doanh thu”.

1. **KẾT QUẢ**

**Ưu điểm:**

+ Đề tài “ ứng dụng quản lý giao dịch sửa chữa thiết bị”, thực tế và được sử dụng nhiều trong công việc bên ngoài.

+ Thông qua báo cáo thực tập: giúp sinh viên nâng cao tính làm việc nhóm, hỗ trợ lẫn nhau để hoàn thành tốt công việc đúng thời hạn.

+ Hỗ trợ giúp sinh viên nâng cao, làm quen với ứng dụng c# và SQL.

+ Phần code được nhận sự hỗ trợ từ giáo viên hướng dẫn và các trang web nước ngoài, giúp công việc đơn giản hơn.

**Nhược điểm:**

+ Nhóm chúng em, vì khả năng code còn yếu nên còn một số lỗi vẫn chưa hoàn thiện tốt form.

+ Đây là bài tập khá khó, nên thường xuyên gặp lỗi khi chạy và nhập các form của từng thành viên, khiến công việc đôi lúc không đúng tiến độ.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã được hoàn thiện gần như 100%.

# TÀI LIỆU THAM KHẢO

1. Phan, D.-V., et al., *Liver cancer prediction in a viral hepatitis cohort: A deep learning approach.* International Journal of Cancer, 2020. **147**(10): p. 2871-2878.

2. Phan, D.V., et al., *Deep learning approaches for sleep disorder prediction in an asthma cohort.* J Asthma, 2020: p. 1-9.

Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC: Phần code của nhóm cho ứng dụng

***FORM NHÂN VIÊN:***

**Load form NHAN\_VIEN:**

private void frmNHAN\_VIEN\_Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sQuery = "select \* from NHAN\_VIEN";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

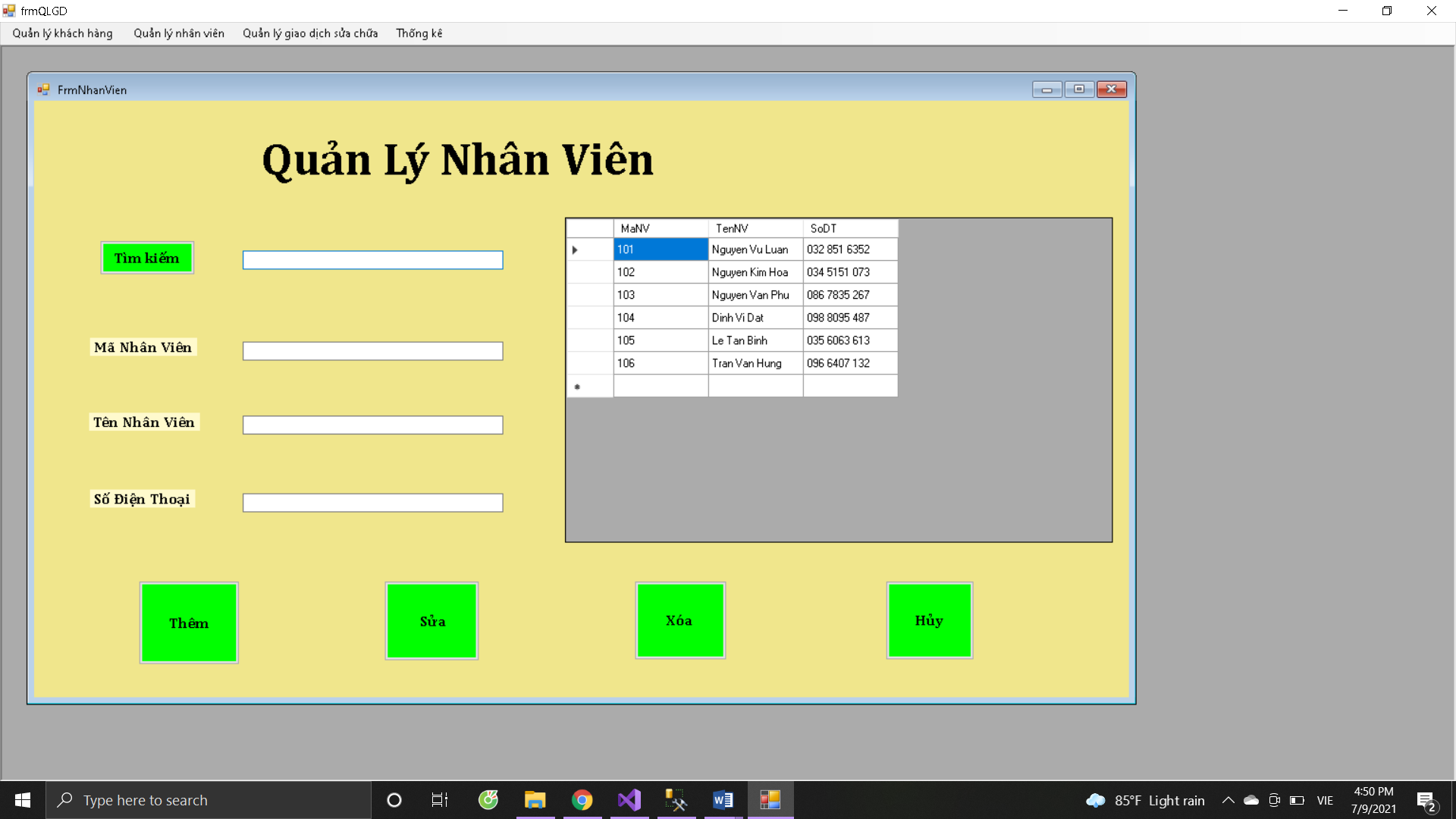
adapter.Fill(ds, "Nhanvien");

dataGridView1.DataSource = ds.Tables["Nhanvien"];

con.Close();

}

**Datagridview:**



private void dataGridView1\_cellclick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

DataGridViewRow row = this.dataGridView1.Rows[e.RowIndex];

txtmanv.Text = row.Cells[0].Value.ToString();

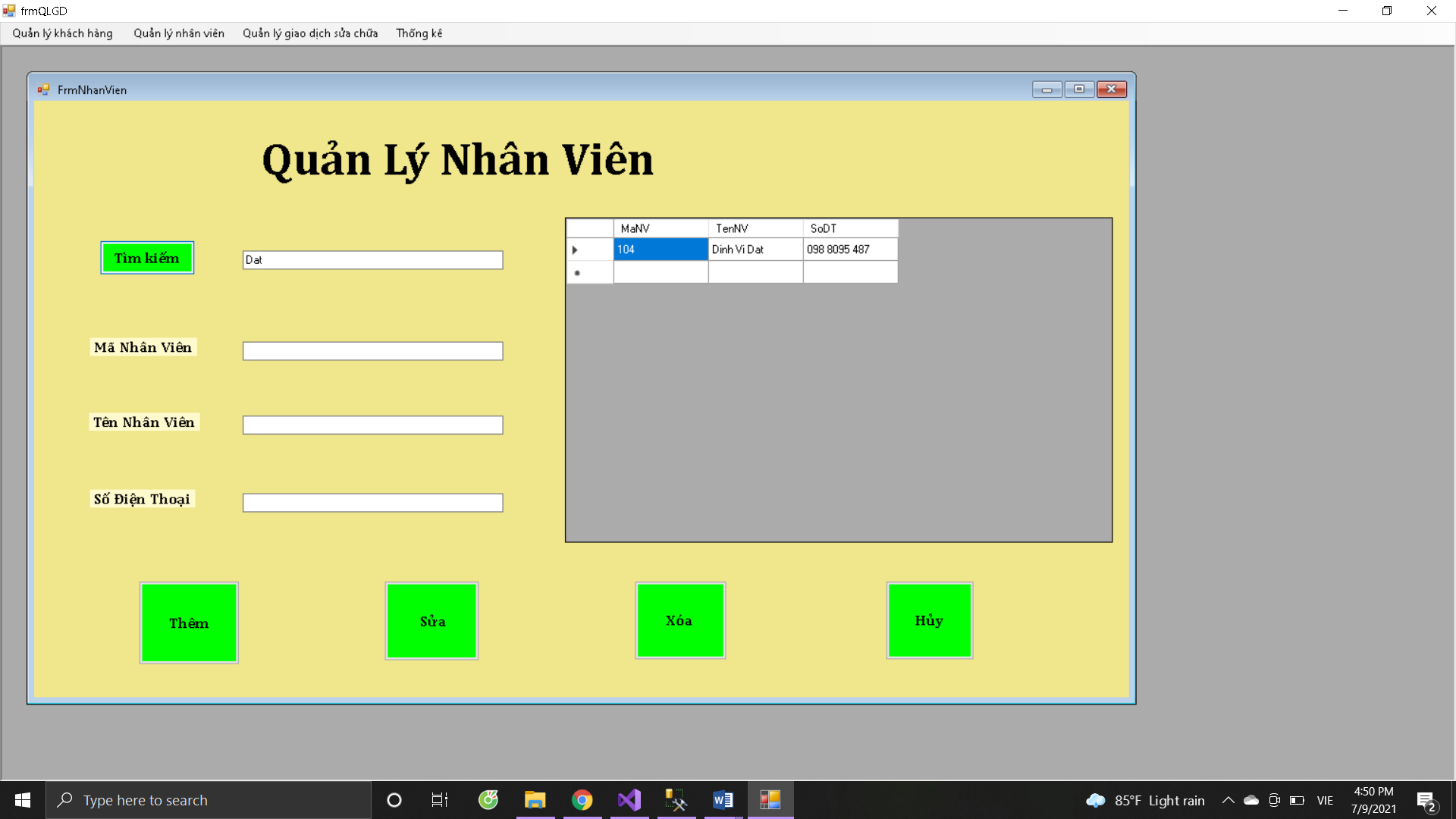
txttennv.Text = row.Cells[1].Value.ToString();

txtsodt.Text = row.Cells[2].Value.ToString();

}

}

**Click\_tìm kiếm.**



private void btntk\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sQuery = "select \* from NHAN\_VIEN where TenNV like N'%" + txttk.Text + "%'";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

try

{

adapter.Fill(ds, "Nhanvien");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Xảy ra lỗi trong quá trình tìm kiếm");

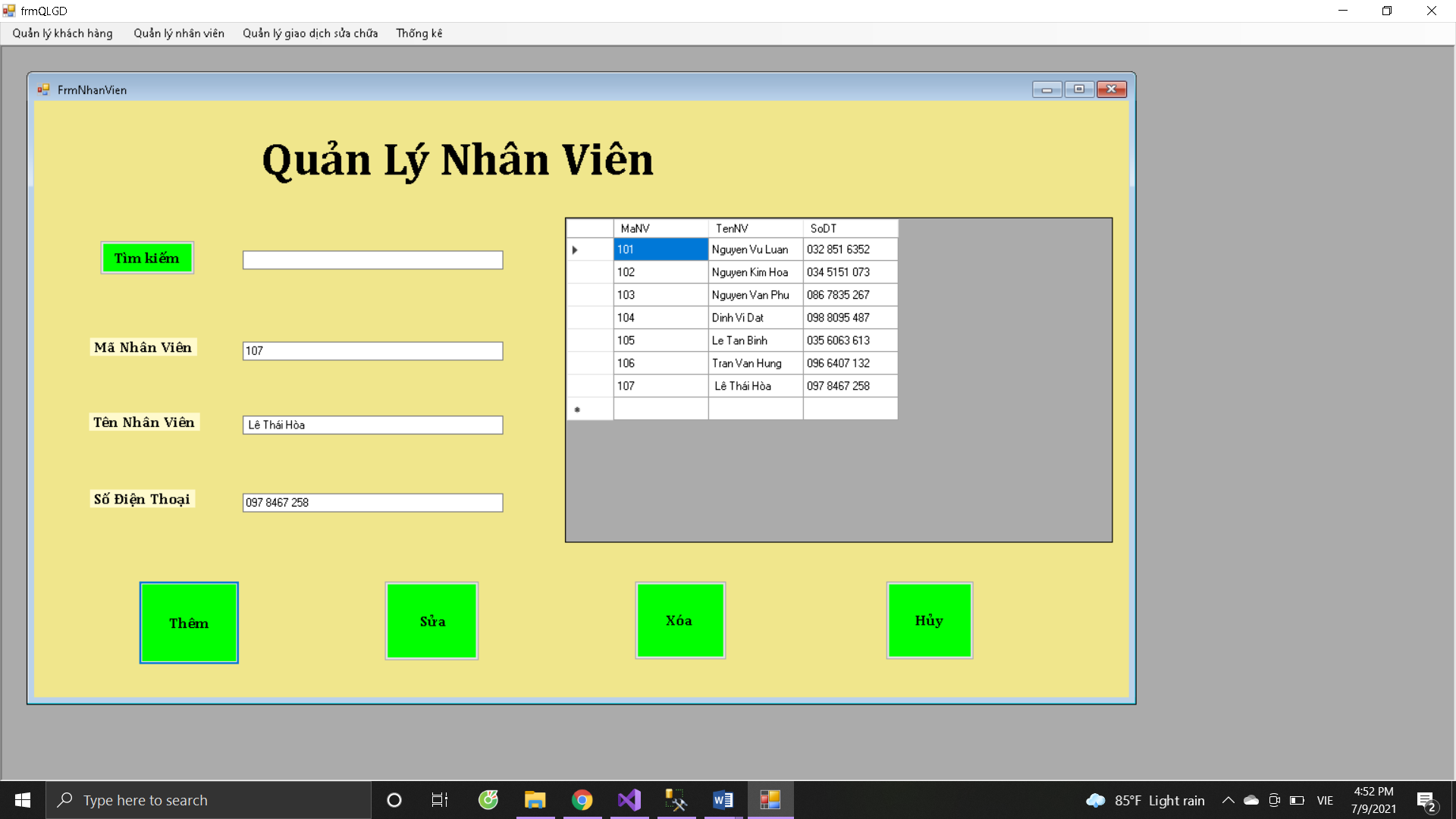
}

dataGridView1.DataSource = ds.Tables["Nhanvien"];

con.Close();

}

**Click\_Thêm Mới:**



private void btnthem\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sMaNV = txtmanv.Text;

string sTenNV = txttennv.Text;

string sSoDT = txtsodt.Text;

string sQuery = "insert into NHAN\_VIEN values(@MaNV, @TenNV, @SoDT)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaNV", sMaNV);

cmd.Parameters.AddWithValue("@TenNV", sTenNV);

cmd.Parameters.AddWithValue("@SoDT", sSoDT);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công!");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(),"Xảy ra lỗi trong quá trình thêm dữ liệu");

}

string sQuery1 = "select \* from NHAN\_VIEN";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

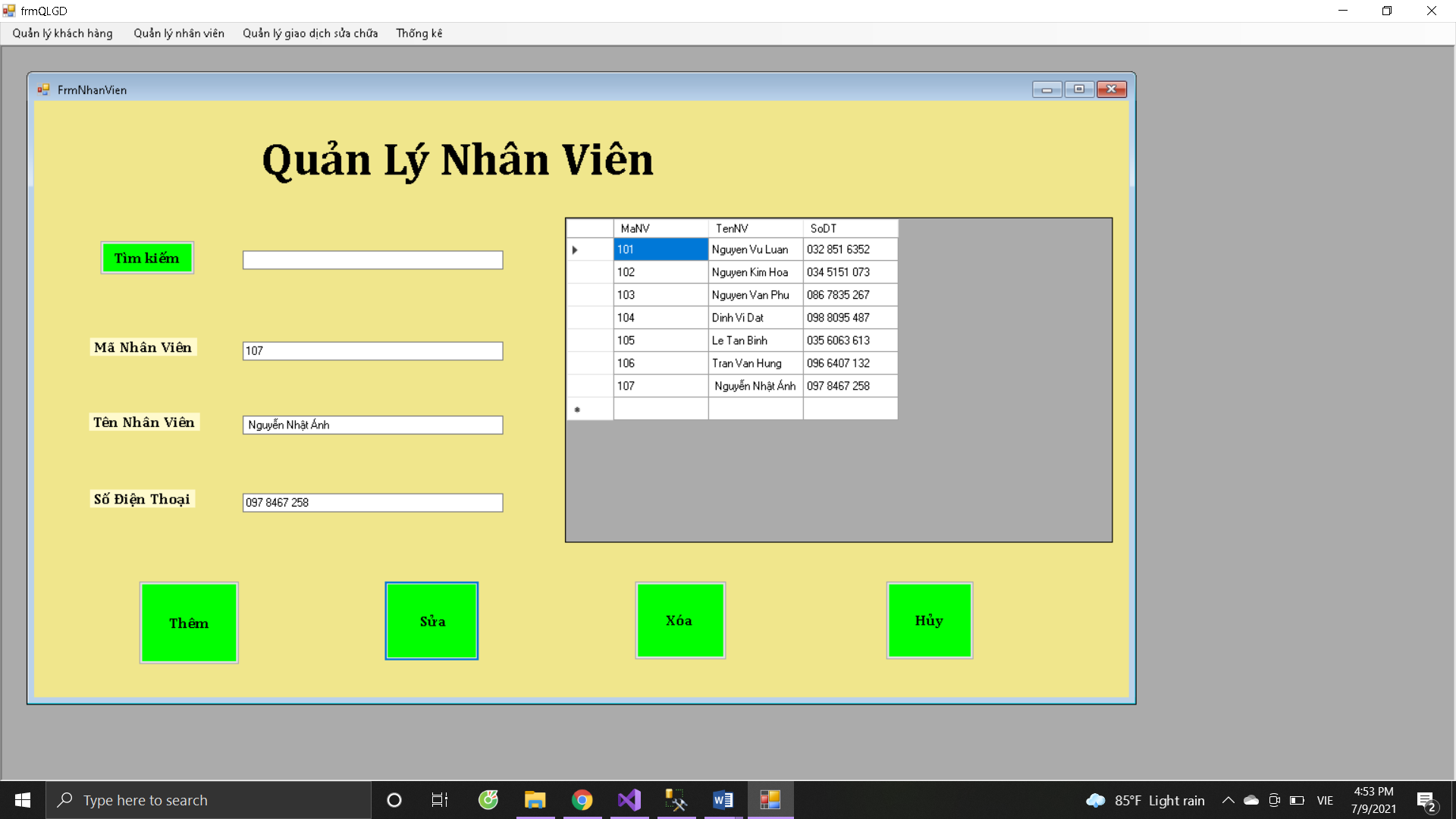
adapter.Fill(ds, "Nhanvien");

dataGridView1.DataSource = ds.Tables["Nhanvien”];

con.Close();

}

**Click Sửa.**



private void btnsua\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sMaNV = txtmanv.Text;

string sTenNV = txttennv.Text;

string sSoDT = txtsodt.Text;

string sQuery = "update NHAN\_VIEN set TenNV = @TenNV, SoDT = @SoDT where MaNV = @MaNV";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaNV", sMaNV);

cmd.Parameters.AddWithValue("@TenNV", sTenNV);

cmd.Parameters.AddWithValue("@SoDT", sSoDT);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thay đổi thành công!");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Xảy ra lỗi trong quá trình thay đổi dữ liệu");

}

string sQuery1 = "select \* from NHAN\_VIEN";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "Nhanvien");

dataGridView1.DataSource = ds.Tables["Nhanvien"];

con.Close();

}

**Click \_Xóa:**



private void btnxoa\_Click\_1(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string snhanvien = txtmanv.Text;

string sQuery = "delete SUA\_CHUA where MaNV = @nhanvien";

string sQuery1 = "delete NHAN\_VIEN where MaNV = @nhanvien";

SqlCommand cmd = new SqlCommand(sQuery, con);

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

cmd.Parameters.AddWithValue("@nhanvien", snhanvien);

cmd1.Parameters.AddWithValue("@nhanvien", snhanvien);

try

{

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

MessageBox.Show("Xóa thành công");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Xảy ra lỗi trong quá trình xóa dữ liệu");

}

string sQuery2 = "select \* from NHAN\_VIEN";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery2, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "Nhanvien");

dataGridView1.DataSource = ds.Tables["Nhanvien"];

con.Close();

}

**Click\_Hủy:**

This.close();

***FORM\_KHÁCH HÀNG:***

**Load form KHACH\_HANG**

private void frmKHACH\_HANG\_Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu!!");

}

string sQuery = "Select \* from KHACH\_HANG";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

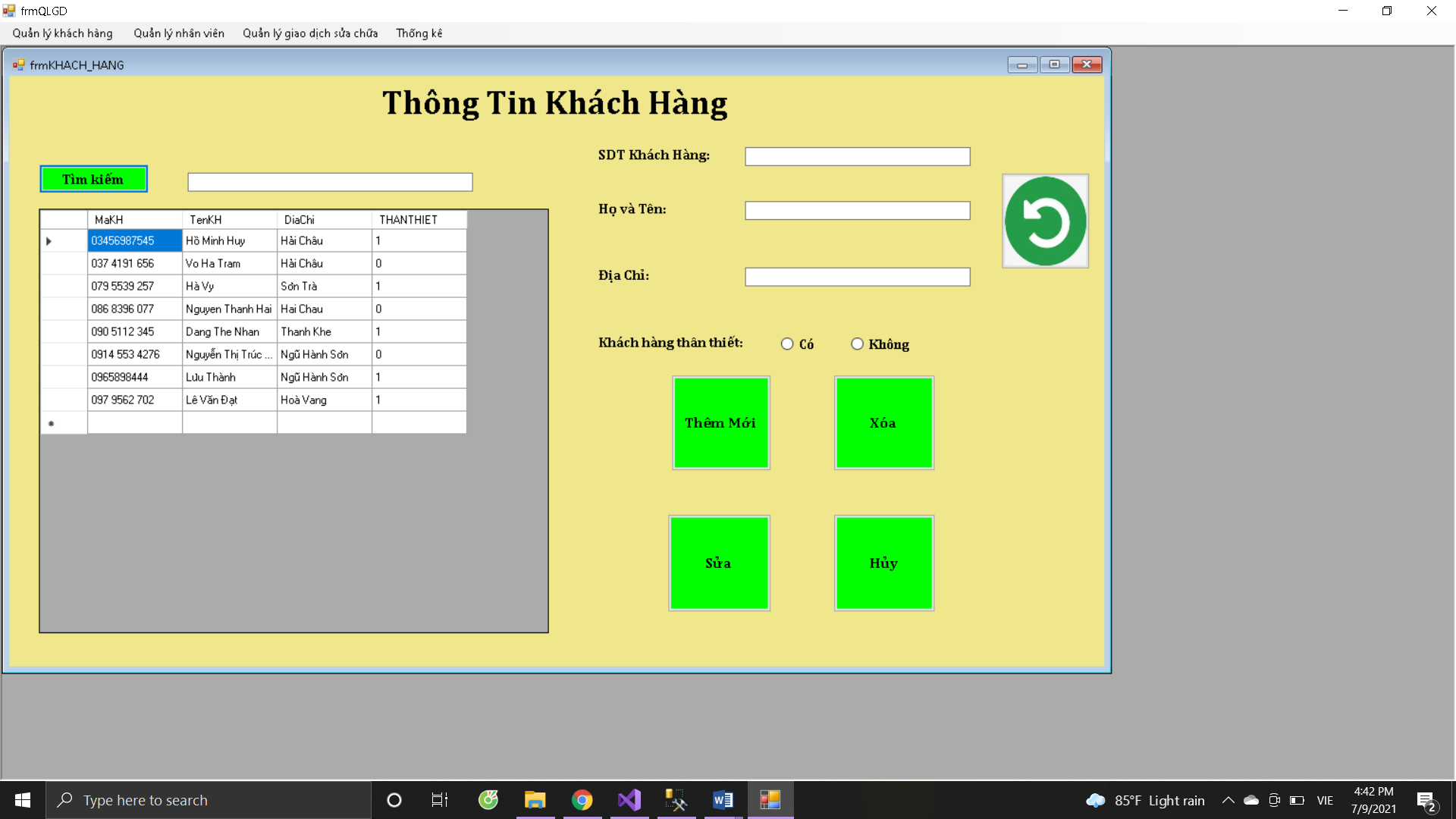
adapter.Fill(ds, "KhachHang");

DTkhachhang.DataSource = ds.Tables["KhachHang"];

con.Close();

}

**Datagridview**



private void DTkhachhang\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

txtSDT.Text = DTkhachhang.Rows[e.RowIndex].Cells["MaKH"].Value.ToString();

txtTen.Text = DTkhachhang.Rows[e.RowIndex].Cells["TenKH"].Value.ToString();

txtDC.Text = DTkhachhang.Rows[e.RowIndex].Cells["DiaChi"].Value.ToString();

int ithanthiet = Convert.ToInt16(DTkhachhang.Rows[e.RowIndex].Cells["THANTHIET"].Value);

if (ithanthiet == 0)

{

rbco.Checked = true;

}

else

{

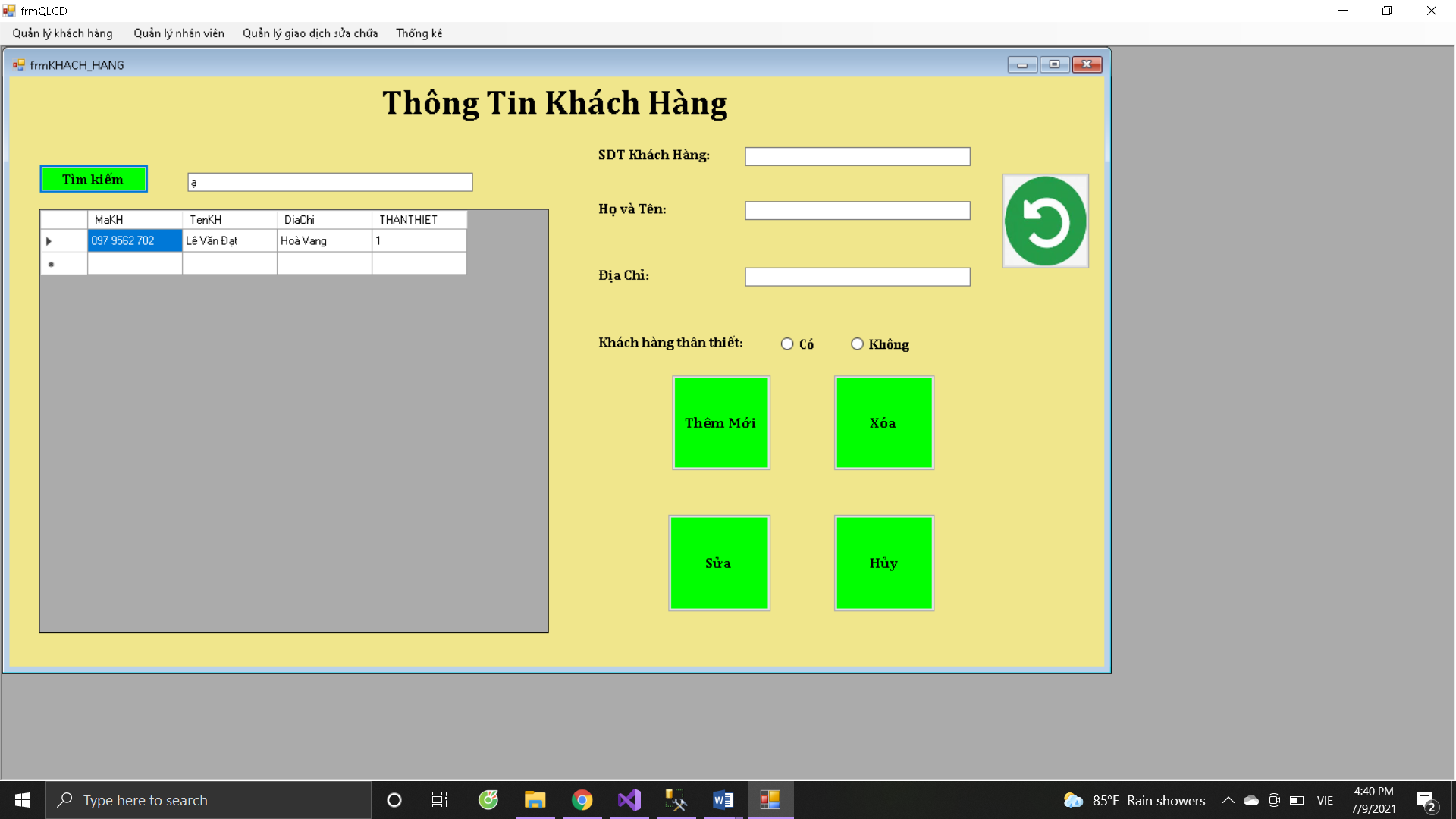
rbkhong.Checked = true;

}

txtSDT.Enabled = false;

}

**Click \_Tìm Kiếm**



private void btntimkiem\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu!!");

}

string sQuery = "Select \* from KHACH\_HANG where TenKH like N'%" + txttimkiem.Text + "%'";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

try

{

adapter.Fill(ds, "KhachHang");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

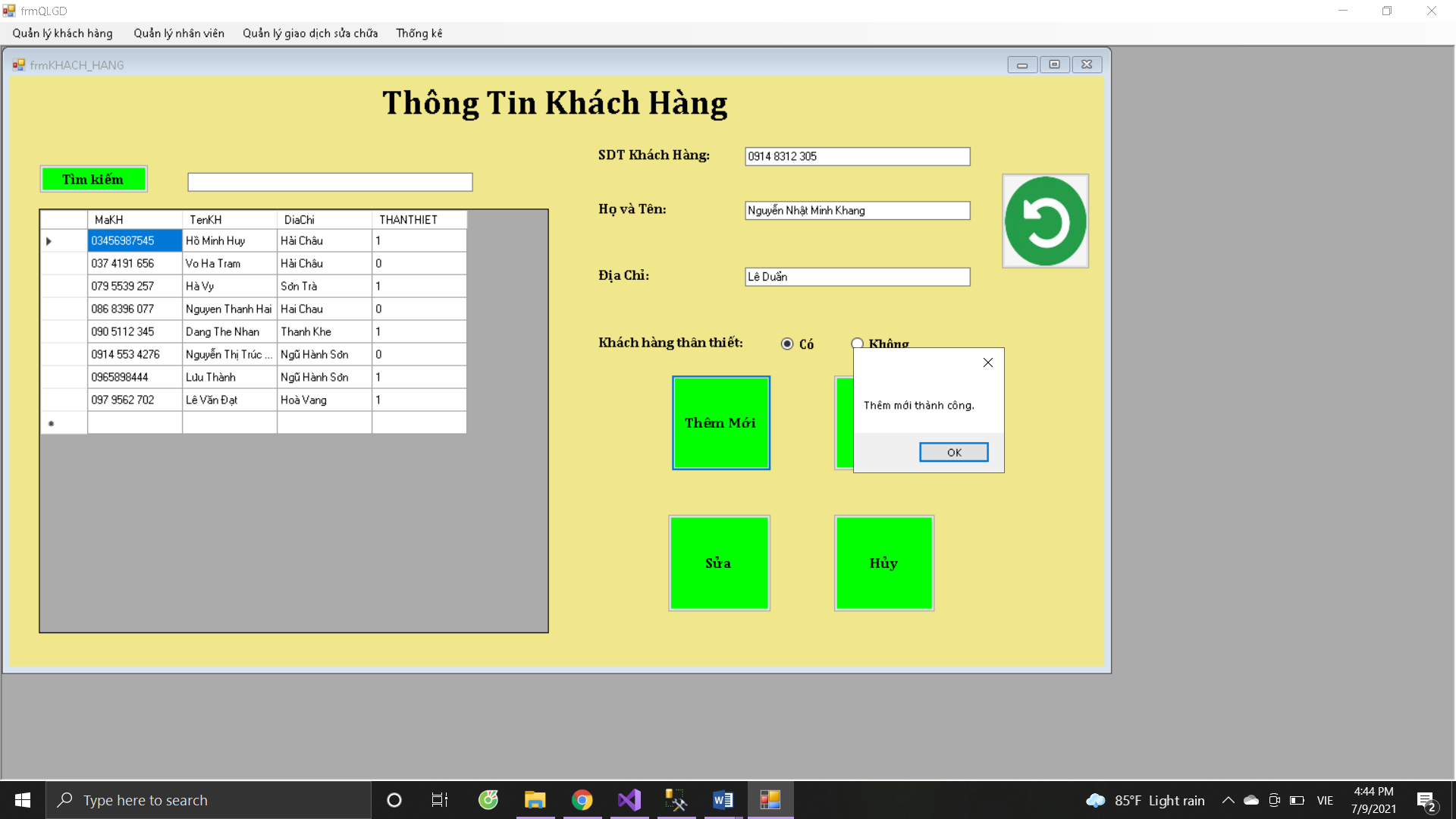
}

DTkhachhang.DataSource = ds.Tables["KhachHang"];

con.Close();

}

**Click\_Thêm Mới**



private void button2\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sSDT = txtSDT.Text;

string sTenKH = txtTen.Text;

string sDiachi = txtDC.Text;

int ithanthiet = 0;

if (rbco.Checked == true)

{

ithanthiet = 1;

}

string sQuery = "Insert into KHACH\_HANG values(@MaKH, @TenKH, @DiaChi,@THANTHIET)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaKH", sSDT);

cmd.Parameters.AddWithValue("@TenKH", sTenKH);

cmd.Parameters.AddWithValue("@DiaChi", sDiachi);

cmd.Parameters.AddWithValue("@THANTHIET", ithanthiet);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công.");

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình thêm mới dữ liệu.");

}

string sQuery2 = "Select \* from KHACH\_HANG";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery2, con);

DataSet ds = new DataSet();

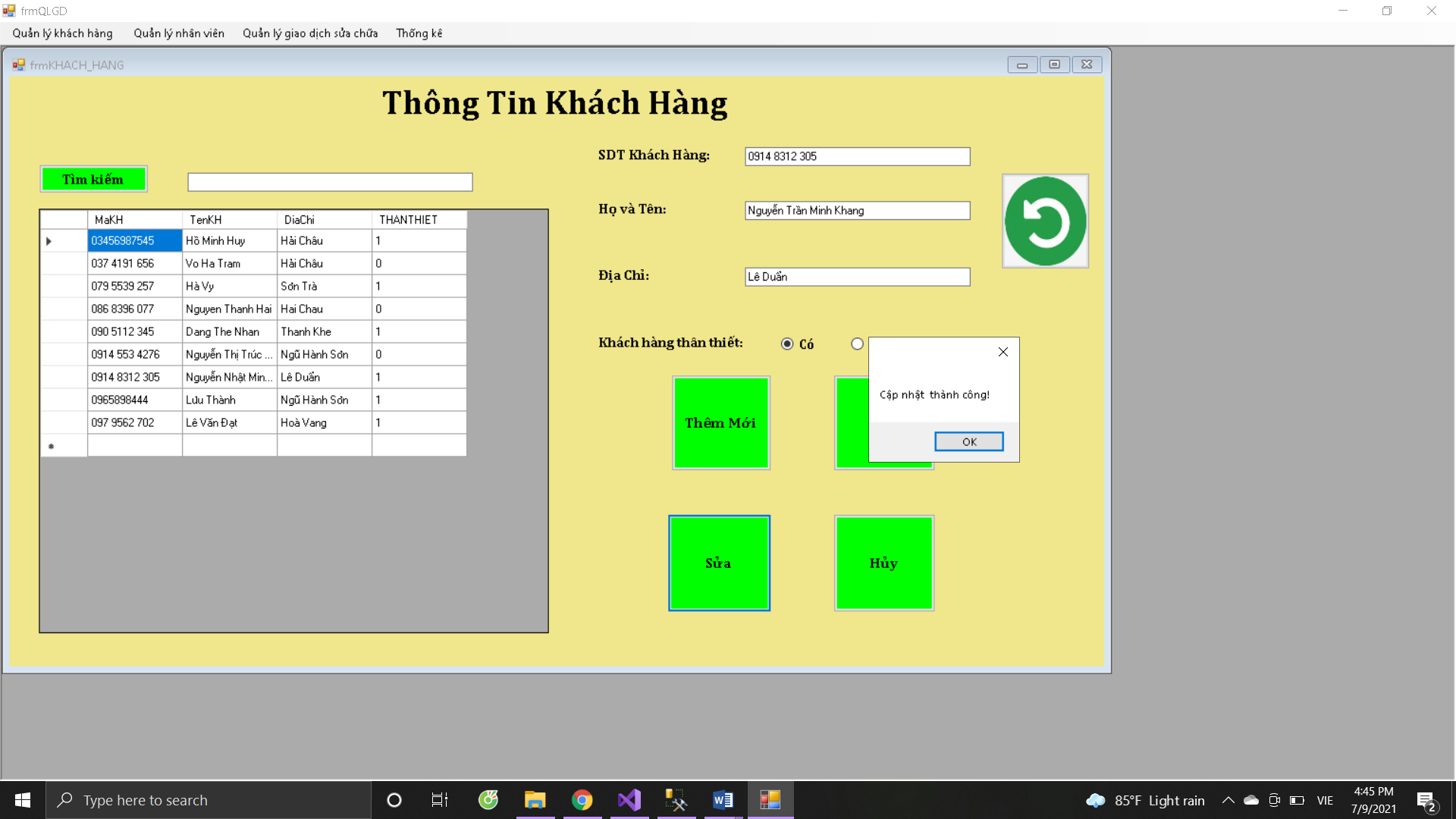
adapter.Fill(ds, "KhachHang");

DTkhachhang.DataSource = ds.Tables["KhachHang"];

con.Close();

}

**Click\_Sửa**



private void btnCapnhat\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sSDT = txtSDT.Text;

string sTen = txtTen.Text;

string sDC = txtDC.Text;

int ithanthiet = 0;

if (rbco.Checked == true)

{

ithanthiet = 1;

}

string sQuery = "update KHACH\_HANG set TenKH = @TenKH, DiaChi = @DiaChi, ThanThiet = @Thanthiet where MaKH = @MaKH";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaKH", sSDT);

cmd.Parameters.AddWithValue("@TenKH", sTen);

cmd.Parameters.AddWithValue("@DiaChi", sDC);

cmd.Parameters.AddWithValue("@THANTHIET", ithanthiet);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công!");

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình thay đổi dữ liệu!!");

}

string sQuery1 = "Select \* from KHACH\_HANG";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

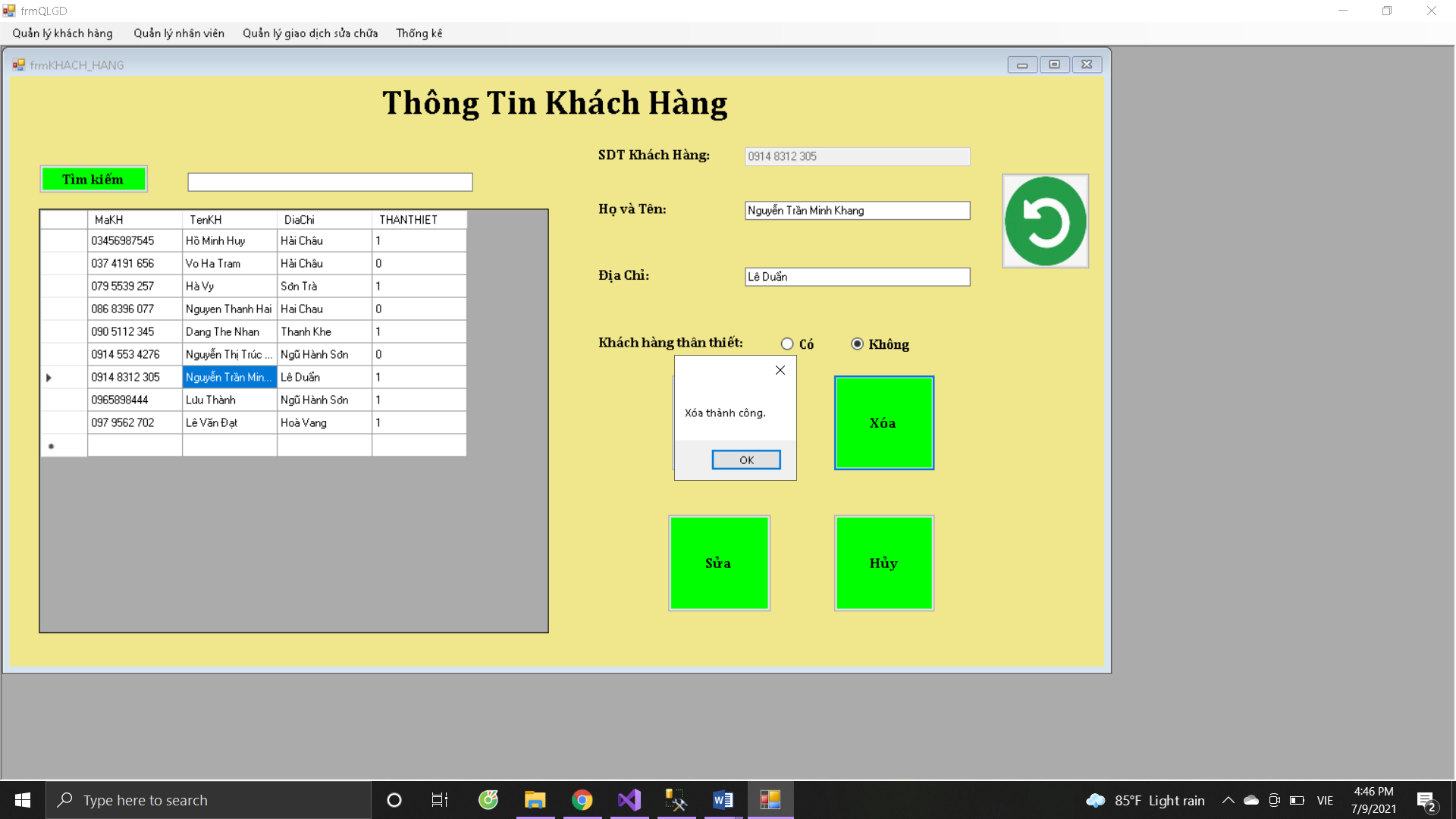
adapter.Fill(ds, "KhachHang");

DTkhachhang.DataSource = ds.Tables["KhachHang"];

con.Close();

}

**Click\_Xóa**



private void btnXoa\_Click(object sender, EventArgs e)

{

DialogResult ret = MessageBox.Show("Bạn thật sự muốn xóa??", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

try

{

string sSDT = txtSDT.Text;

string sQuery = "delete SUA\_CHUA where MaKH = @SDT";

string sQuery1 = "delete KHACH\_HANG where MaKH = @SDT";

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@SDT", sSDT);

cmd1.Parameters.AddWithValue("@SDT", sSDT);

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

MessageBox.Show("Xóa thành công.");

}

catch (Exception ex)

{

MessageBox.Show("Không thể xóa khách hàng này vì có liên kết thông tin khác", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

txtSDT.Text = "";

txtTen.Text = "";

txtDC.Text = "";

rbkhong.Checked = false;

rbco.Checked = false;

txtSDT.Enabled = true;

string sQuery2 = "Select \* from KHACH\_HANG";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery2, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "KhachHang");

DTkhachhang.DataSource = ds.Tables["KhachHang"];

con.Close();

}

}

**Mô tả sự kiện:**

**Load form thống kê.**

private void frmTHONG\_KE\_ Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu!!");

}

string sQuery = "Select TongTien from SUA\_CHUA";

SqlDataAdapter adapter = new SqlDataAdapter (sQuery, con);

DataSet ds = new DataSet ();

adapter.Fill (ds, "SUA\_CHUA");

con.Close();

}

**Click Thống kê.**



private void button1\_Click\_1(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

try

{

string sQuery1 = "select TrangThai, sum(TongTien)as TongTien from SUA\_CHUA " +

"where NgayNhan between @NgayNhan and @NgayTra" + txttk.SelectedText.ToString() + " group by TrangThai”;

String sNgayNhan = dtungay.Value.ToString("yyyy-MM-dd");

String sNgayTra = ddenngay.Value.ToString("yyyy-MM-dd");

string sTrangthai;

if (rddangsua.Checked == true)

{

sTrangthai = "Đang sửa";

}

if (rdsuaxong.Checked == true)

{

sTrangthai = "Sửa Xong";

}

else

{

sTrangthai = "Hủy sửa chữa";

}

SqlCommand cmd = new SqlCommand(sQuery1, con);

cmd.Parameters.AddWithValue("@NgayNhan", sNgayNhan);

cmd.Parameters.AddWithValue("@NgayTra", sNgayTra);

cmd.Parameters.AddWithValue("@TrangThai", sTrangthai);

DataTable dt = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

dt.Load(dr);

txttk.Text = dt.Rows[0][1].ToString();

}

catch (Exception ex)

{

MessageBox.Show("Thống kê không thành công!\n" + ex.Message, "Thông báo!");

}

finally

{

con.Close();

}

**Click nút Reset:**

private void button2\_Click(object sender, EventArgs e)

{

txttk.Text = "";

}

**Click nút Hủy:**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

***Mô tả sự kiện:***

**Load form SUA\_CHUA**

private void frmSUA\_CHUA\_Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sQuery1 = "select MaKH,TenKH from KHACH\_HANG";

SqlDataAdapter adapter1 = new SqlDataAdapter(sQuery1, con);

DataSet ds1 = new DataSet();

adapter1.Fill(ds1, "KH");

cbMaKH.DataSource = ds1.Tables["KH"];

cbMaKH.ValueMember = "MaKH";

cbMaKH.DisplayMember = "TenKH";

string sQuery2 = "select MaNV,TenNV from NHAN\_VIEN";

SqlDataAdapter adapter2 = new SqlDataAdapter(sQuery2, con);

DataSet ds2 = new DataSet();

adapter2.Fill(ds2, "NV");

cbMaNV.DataSource = ds2.Tables["NV"];

cbMaNV.ValueMember = "MaNV";

cbMaNV.DisplayMember = "TenNV";

string sQuery3 = "select MaCV,NoiDung from CONG\_VIEC";

SqlDataAdapter adapter3 = new SqlDataAdapter(sQuery3, con);

DataSet ds3 = new DataSet();

adapter3.Fill(ds3, "CV");

cbMaCV.DataSource = ds3.Tables["CV"];

cbMaCV.ValueMember = "MaCV";

cbMaCV.DisplayMember = "NoiDung";

string sQuery4 = "select MaHD from SUA\_CHUA";

SqlDataAdapter adapter4 = new SqlDataAdapter(sQuery4, con);

DataSet ds4 = new DataSet();

adapter4.Fill(ds4, "HD");

cbTimkiem.DataSource = ds4.Tables["HD"];

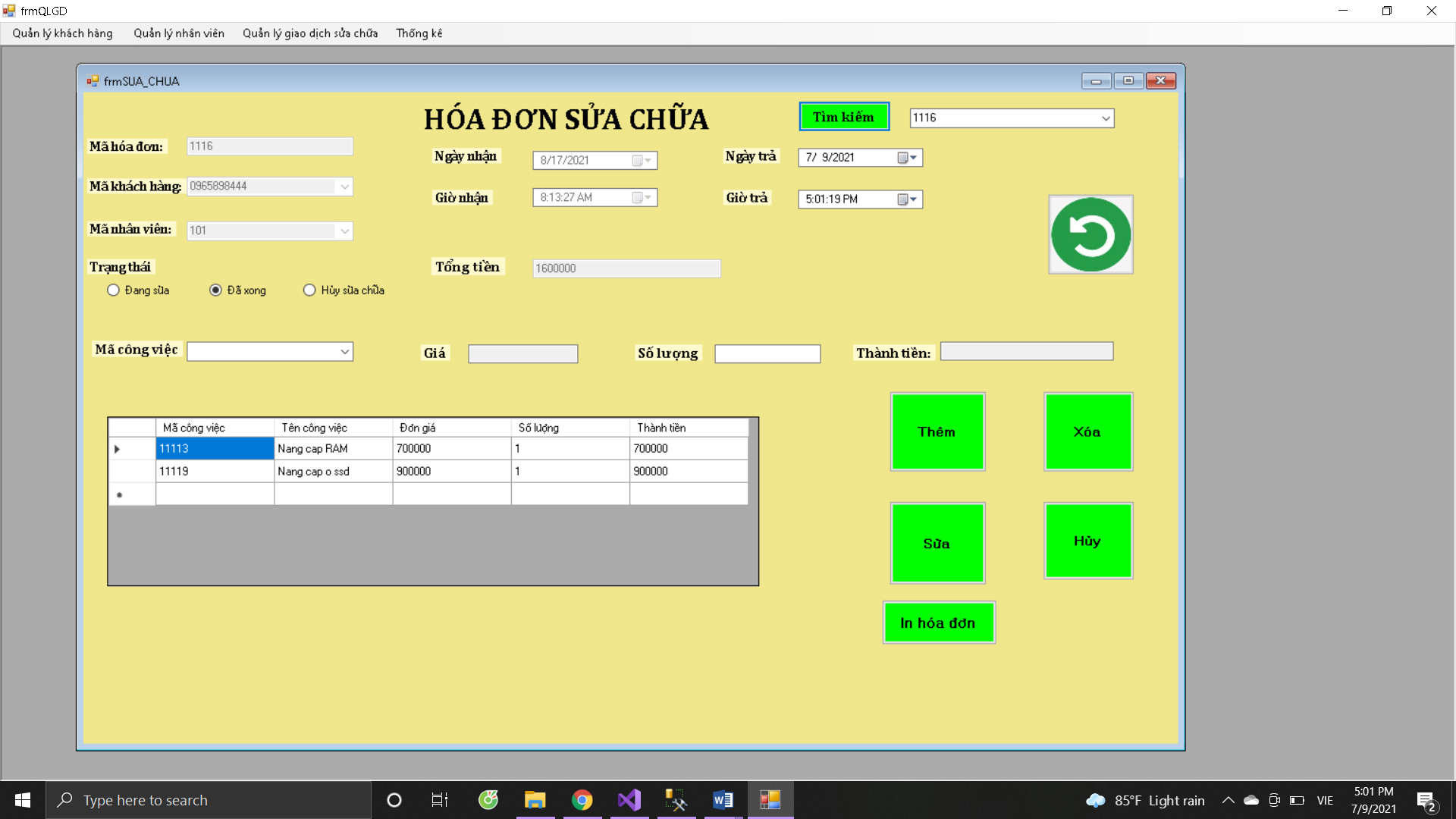
cbTimkiem.ValueMember = "MaHD";

cbTimkiem.DisplayMember = "MaHD";

con.Close();

}

**click Tìm kiếm:**



SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

string sQuery = "Select \* from SUA\_CHUA where MaHD=" + cbTimkiem.SelectedValue.ToString();

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "giaodich");

txtMaHD.Text = ds.Tables["giaodich"].Rows[0][0].ToString();

cbMaKH.Text = ds.Tables["giaodich"].Rows[0][1].ToString();

cbMaNV.Text = ds.Tables["giaodich"].Rows[0][2].ToString();

// txtMaHD.Text = ds.Tables["giaodich"].Rows[0][3].ToString();

try

{

dateTimePicker2.Text = ds.Tables["giaodich"].Rows[0][4].ToString();

dateTimePicker1.Text = ds.Tables["giaodich"].Rows[0][3].ToString();

dateTimePicker4.Text = ds.Tables["giaodich"].Rows[0][5].ToString();

dateTimePicker3.Text = ds.Tables["giaodich"].Rows[0][6].ToString();

}

catch (Exception ex) { }

if (ds.Tables["giaodich"].Rows[0][7].ToString() == "Đang sửa")

{

rbDangsua.Checked = true;

}

else if (ds.Tables["giaodich"].Rows[0][7].ToString() == "Xong")

{

rbDaxong.Checked = true;

}

else

{

rbHuy.Checked = true;

}

txtTongtien.Text = ds.Tables["giaodich"].Rows[0][8].ToString();

txtMaHD.Enabled = false;

string sQuery1 = "Select CONG\_VIEC.MaCV, NoiDung, Cong\_Viec.ThanhTien, SoLuong, SUA\_CHUA\_CHI\_TIET.ThanhTien from SUA\_CHUA\_CHI\_TIET join CONG\_VIEC on SUA\_CHUA\_CHI\_TIET.MaCV = CONG\_VIEC.MaCV where MaHD = " + cbTimkiem.SelectedValue.ToString();

SqlDataAdapter adapter1 = new SqlDataAdapter(sQuery1, con);

DataSet ds1 = new DataSet();

adapter1.Fill(ds1, "giaodichchitiet");

int iCount = ds1.Tables[0].Rows.Count;

for (int i=0; i < iCount; i++)

{

dgSCCT.Rows.Add(ds1.Tables[0].Rows[i][0], ds1.Tables[0].Rows[i][1], ds1.Tables[0].Rows[i][2], ds1.Tables[0].Rows[i][3], ds1.Tables[0].Rows[i][4]);

}

//để người dùng ko sửa được các mục khác ngày, giờ trả và trạng thái thì ở đây cần phải disable các item khác đi, ví dụ:

txtMaHD.Enabled = false;

cbMaKH.Enabled = false;

cbMaNV.Enabled = false;

dateTimePicker1.Enabled = false;

dateTimePicker2.Enabled = false;

txtTongtien.Enabled = false;

dgSCCT.Enabled = false;

//tương tự với các control khác

con.Close();

**Click Thêm mới:**



SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception Ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

//Them vao bang SUA\_CHUA

int iMaHD = Convert.ToInt32(txtMaHD.Text);

string sMaKH = cbMaKH.SelectedValue.ToString();

string sMaNV = cbMaNV.SelectedValue.ToString();

string sNgaynhan = dateTimePicker2.Value.ToString("yyyy-MM-dd");

string sGionhan = dateTimePicker1.Value.ToString("HH:mm:ss");

string sNgaytra = dateTimePicker4.Value.ToString("yyyy-MM-dd");

string sGiotra = dateTimePicker3.Value.ToString("HH:mm:ss");

String sTongtien = txtTongtien.Text;

string sTrangthai;

if (rbDangsua.Checked == true)

{

sTrangthai = "Đang sửa"; }

else if (rbDaxong.Checked == true)

{

sTrangthai = "Xong";

}

else

{

sTrangthai = "Hủy";

}

string sQuery = "insert into SUA\_CHUA(MaHD,MaKH,MaNV,Gionhan,Ngaynhan,TrangThai,TongTien) values " +

"(@MaHD,@MaKh,@MaNV,@GioNhan,@NgayNhan,@TrangThai,@TongTien)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHD", iMaHD);

cmd.Parameters.AddWithValue("@MaKH", sMaKH);

cmd.Parameters.AddWithValue("@MaNV", sMaNV);

cmd.Parameters.AddWithValue("@GioNhan", sGionhan);

cmd.Parameters.AddWithValue("@NgayNhan", sNgaynhan);

cmd.Parameters.AddWithValue("@TrangThai", sTrangthai);

cmd.Parameters.AddWithValue("@TongTien", sTongtien);

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

//Them giu lieu vao bang SUA\_CHUA\_CHI\_TIET

int iCount = dgSCCT.Rows.Count;

for (int i = 0; i < iCount-1; i++)

{

string sQuery1 = "insert into SUA\_CHUA\_CHI\_TIET(MaHD, MaCV, SoLuong, ThanhTien) values (@MaHD,@MaCV,@SoLuong,@ThanhTien)";

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

cmd1.Parameters.AddWithValue("@MaHD", iMaHD);

cmd1.Parameters.AddWithValue("@MaCV", dgSCCT.Rows[i].Cells[0].Value);

cmd1.Parameters.AddWithValue("@SoLuong", dgSCCT.Rows[i].Cells[3].Value);

cmd1.Parameters.AddWithValue("@ThanhTien", dgSCCT.Rows[i].Cells[4].Value);

try

{

cmd1.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

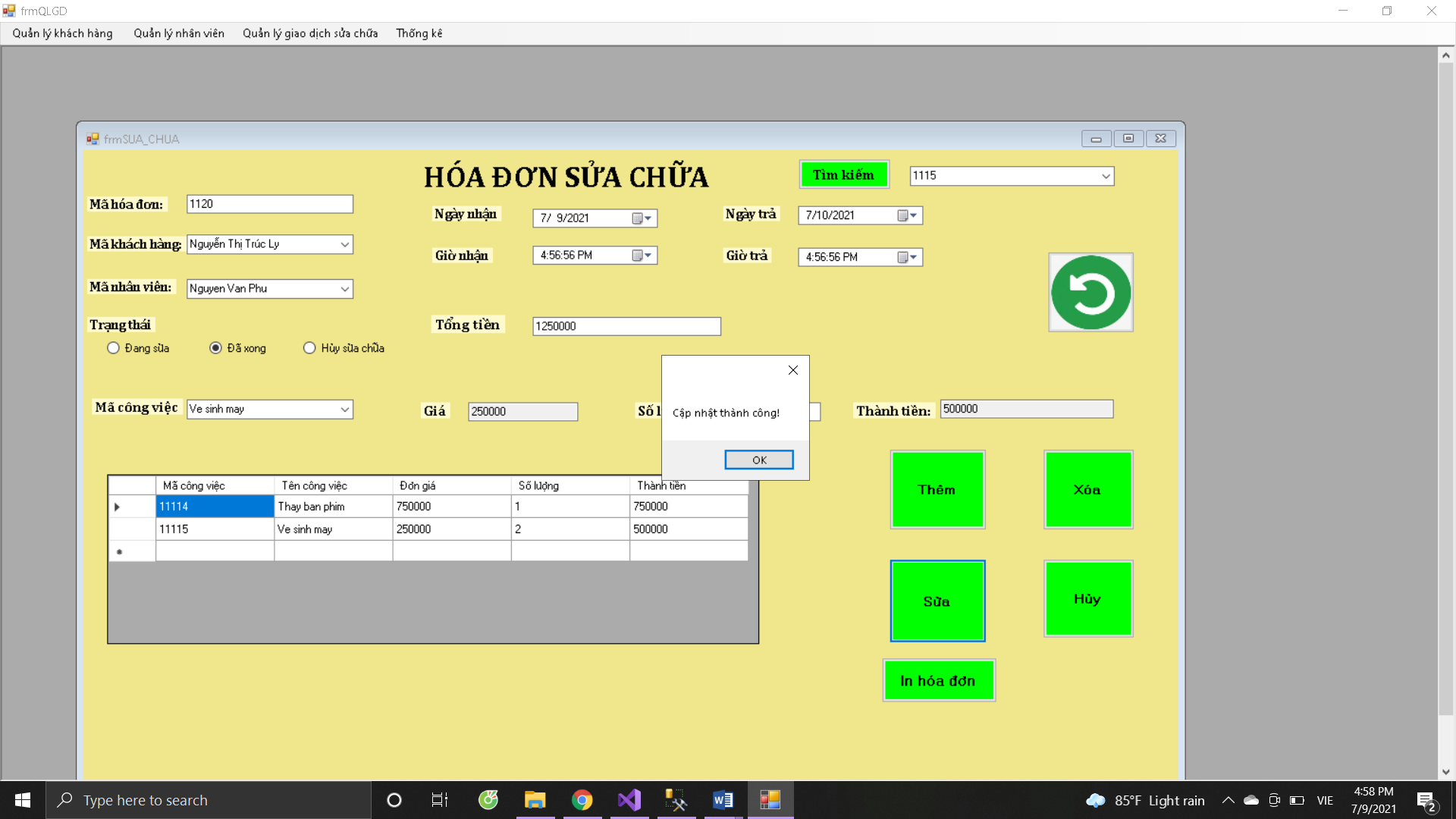
}

MessageBox.Show("Xử lý xong!");

con.Close();//Buoc 3

}

**Click Sửa:**



SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

//lấy ra ngày trả, giờ trả và trạng thái

string sMaHD = txtMaHD.Text;

string sNgaytra = dateTimePicker4.Value.ToString("yyyy-MM-dd");

string sGiotra = dateTimePicker3.Value.ToString("HH:mm:ss");

string sTrangthai;

if (rbDangsua.Checked == true)

{

sTrangthai = "Đang sửa";

}

else if (rbDaxong.Checked == true)

{

sTrangthai = "Xong";

}

else

{

sTrangthai = "Hủy";

}

string sQuery = "update SUA\_CHUA set ngaytra=@ngaytra, giotra = @giotra, trangthai = @trangthai where mahd = @mahd";

SqlCommand cmd = new SqlCommand(sQuery, con);

//còn chưa khai báo các biên @ngaytra, @giotra, @trangthai, @mahd cho lệnh

cmd.Parameters.AddWithValue("@maHD", sMaHD);

cmd.Parameters.AddWithValue("@ngaytra", sNgaytra);

cmd.Parameters.AddWithValue("@giotra", sGiotra);

cmd.Parameters.AddWithValue("@trangthai", sTrangthai);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công!");

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình cập nhật!");

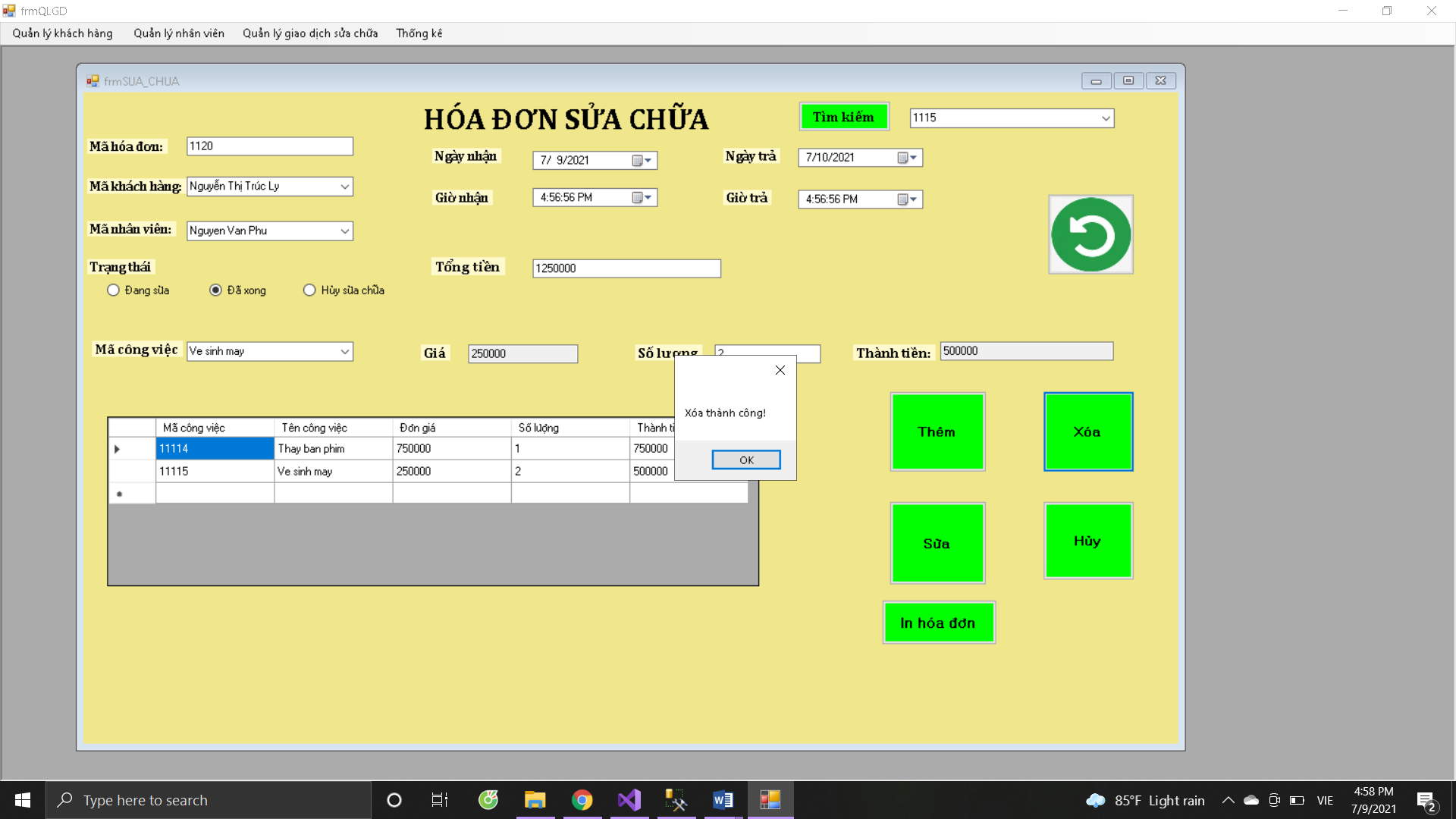
}

//cập nhật bảng hóa đơn

//update hoadon set ngaytra = @ngaytra, giotra = @giotra where mahd = @mahd

con.Close();

**Click Xóa:**



DialogResult ret = MessageBox.Show("Có chắn chắn xóa không?", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

//phần này xóa sửa chữa chi tiêt --> Dùng vòng lặp

int iCount = dgSCCT.Rows.Count;

for (int i = 0; i < iCount - 1; i++)

{

string sMaCV = dgSCCT.Rows[i].Cells[0].Value.ToString();

string sQuery1 = "delete SUA\_CHUA\_CHI\_TIET where Mahd=@MaHD and MaCV=@MaCV";

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

cmd1.Parameters.AddWithValue("@MaCV",dgSCCT.Rows[i].Cells[0].Value);

cmd1.Parameters.AddWithValue("@MaHD", txtMaHD.Text);

try

{

cmd1.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

string sMaHD = txtMaHD.Text;

string sQuery = "delete SUA\_CHUA where mahd=@MaDH ";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaDH", sMaHD);

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

MessageBox.Show("Xóa thành công!");

con.Close();

txtMaHD.Text = "";

cbMaCV.Text = "";

cbMaKH.Text = "";

cbMaNV.Text = "";

txtSL.Text = "";

txtThanhtien.Text = "";

txtTongtien.Text = "";

txtGia.Text = "";

dgSCCT.Rows.Clear();

}

**click Hủy:**

this.Close();

**Click In Hóa Đơn:**

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception Ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối dữ liệu");

}

//Them vao bang SUA\_CHUA

int iMaHD = Convert.ToInt32(txtMaHD.Text);

string sMaKH = cbMaKH.SelectedValue.ToString();

string sMaNV = cbMaNV.SelectedValue.ToString();

string sNgaynhan = dateTimePicker2.Value.ToString("yyyy-MM-dd");

string sGionhan = dateTimePicker1.Value.ToString("HH:mm:ss");

string sNgaytra = dateTimePicker4.Value.ToString("yyyy-MM-dd");

string sGiotra = dateTimePicker3.Value.ToString("HH:mm:ss");

String sTongtien = txtTongtien.Text;

string sTrangthai;

if (rbDangsua.Checked == true)

{

sTrangthai = "Đang sửa";

}

if (rbDaxong.Checked == true)

{

sTrangthai = "Xong";

}

else

{

sTrangthai = "Hủy";

}

string sQuery = "insert into SUA\_CHUA(MaHD,MaKH,MaNV,Gionhan,Ngaynhan,Ngaytra,Giotra,TrangThai,TongTien) values " +

"(@MaHD,@MaKh,@MaNV,@Gionhan,@NgayNhan,@NgayTra,@GioTra,@TrangThai,@TongTien)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHD", iMaHD);

cmd.Parameters.AddWithValue("@MaKH", sMaKH);

cmd.Parameters.AddWithValue("@Gionhan", sGionhan);

cmd.Parameters.AddWithValue("@Ngaynhan", sNgaynhan);

cmd.Parameters.AddWithValue("@Ngaytra", sNgaytra);

cmd.Parameters.AddWithValue("@Giotra", sGiotra);

cmd.Parameters.AddWithValue("@Tongtien", sTongtien);

cmd.Parameters.AddWithValue("@Trangthai", sTrangthai);

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

//Them giu lieu vao bang SUA\_CHUA\_CHI\_TIET

int iCount = dgSCCT.Rows.Count;

for (int i=0; i<iCount; i++)

{

string sQuery1 = "insert into SUA\_CHUA\_CHI\_TIET values (@MaHD,@MaCV,@Soluong,@Thanhtien)";

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

cmd1.Parameters.AddWithValue("@MaHD", iMaHD);

cmd1.Parameters.AddWithValue("@MaCV", dgSCCT.Rows[i].Cells[0].Value);

cmd1.Parameters.AddWithValue("@Soluong", dgSCCT.Rows[i].Cells[3].Value);

cmd1.Parameters.AddWithValue("@Thanhtien", dgSCCT.Rows[i].Cells[4].Value);

try

{

cmd1.ExecuteNonQuery();

//Messagebox.Show("Them moi thanh cong!");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

MessageBox.Show("Sử lý xong!");

con.Close();

**DataGridview frmSUA\_CHUA:**

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{ txtMaHD.Text= dgSCCT.Rows[e.RowIndex].Cells ["MaHD"].Value.ToString();

cbMaKH.Text = dgSCCT.Rows[e.RowIndex].Cells["MaKH"].Value.ToString();

cbMaNV.Text = dgSCCT.Rows[e.RowIndex].Cells["MaNV"].Value.ToString();

txtTongtien.Text= dgSCCT.Rows[e.RowIndex].Cells["Tongtien"].Value.ToString();

dateTimePicker2.Value =

Convert.ToDateTime(dgSCCT.Rows[e.RowIndex].Cells["Ngaynhan"].Value.ToString());

dateTimePicker1.Value =

Convert.ToDateTime(dgSCCT.Rows[e.RowIndex].Cells["Gionhan"].Value.ToString());

dateTimePicker4.Value =

Convert.ToDateTime(dgSCCT.Rows[e.RowIndex].Cells["Ngaytra"].Value.ToString());

dateTimePicker3.Value =

Convert.ToDateTime(dgSCCT.Rows[e.RowIndex].Cells["Giotra"].Value.ToString());

string sTrangthai =

dgSCCT.Rows[e.RowIndex].Cells["Trangthai"].Value.ToString();

if (sTrangthai == "Đang sửa")

{

rbDangsua.Checked = true;

}

if (sTrangthai == "Đã xong")

{

rbDaxong.Checked = true;

}

else

{

rbHuy.Checked = true;

}

txtThanhtien.Text =

dgSCCT.Rows[e.RowIndex].Cells["ThanhTien"].Value.ToString();

cbMaKH.Enabled = false;

cbMaNV.Enabled = false;

}