TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

**BÁO CÁO THỰC TẬP NHẬN THỨC**



**TÊN ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ ĐĂNG KÝ THI CHỨNG CHỈ**

**Nhóm: 07**

**Sinh viên thực hiện:**

1. Trưởng nhóm: Phan Thị Phương Thảo Lớp : 45k21.2
2. Thành viên: Nguyễn Hữu Quốc Lớp : 45k21.2
3. Thành viên: Đỗ Chí Toàn Lớp : 45k21.2

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

**Đà Nẵng, 6/2021**

**DANH SÁCH THÀNH VIÊN THAM GIA**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Họ và tên** | **Lớp** | **Nhiệm vụ** |
| 1 | Phan Thị Phương Thảo | 45K21.2 | Trưởng nhóm |
| 2 | Nguyễn Hữu Quốc | 45K21.2 | Thành viên |
| 3 | Đỗ Chí Toàn | 45K21.2 | Thành viên |

**LỜI CẢM ƠN**

Để hoàn thành bài báo cáo này, nhóm chúng em xin tỏ lòng biết ơn sâu sắc đến khoa Thống Kê – Tin Học và đặc biệt là cô Cao Thị Nhâm đã tận tình hướng dẫn, chỉ dạy trong thời gian thực hiện môn học.

Trong thời gian vừa qua dù nhóm chung em đã rất cố gắng nhưng một phần cũng do kiến thức còn hạn chế nên bài báo cáo này khó tránh khỏi những sai sót . Nhóm chúng em rất mong nhận được ý kiến đóng góp của cô để bài báo cáo được hoàn thiện hơn.

Nhóm em xin trân thành cảm ơn !

**LỜI CAM ĐOAN**

Bài báo cáo thực tập với đề tài “ Xây dựng ứng dụng quản lý đăng ký thi chứng chỉ ” là một dự án được thực hiện dưới sự hướng dẫn của giảng viên Cao Thị Nhâm. Ngoài ra không có sự sao chép bài từ cá nhân khác. Nội dung báo cáo của đề tài là sản phẩm mà nhóm em đã cùng nhau nổ lực nghiên cứu trong thời gian vừa qua. Kết quả trình bày trong báo cáo hoàn toàn trung thực.

Nhóm chúng em xin chịu hoàn toàn trách nhiệm về lời cam đoan trên!

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nội dung công việc** | **TG bắt đầu** | **TG kết thúc** | **Kết quả** |
| 01 | Phan Thị Phương Thảo | Tạo cơ sở dữ liệu | 14/06/2021 | 16/06/2021 | Tốt |
|  |  | Tạo form đăng nhập và quản lý kì thi | 17/06/2021 | 18/06/2021 | Tốt |
|  |  | Code và hoàn thiện chức năng đăng nhập | 19/06/2021 | 22/06/2021 | Tốt |
|  |  | Code và hoàn thiện chức năng quản lý kì thi | 23/06/2021 | 27/07/2021 | Tốt |
|  |  | Viết báo cáo chương 3 | 28/06/2021 | 30/06/2021 | Tốt |
|  |  | Chỉnh sửa và hoàn thiện báo cáo | 01/07/2021 | 02/07/2021 | Tốt |
| 02 | Đỗ Chí Toàn | Tạo form sơ bộ cho dự án | 14/06/2021 | 16/06/2021 | Tốt |
|  |  | Tạo form quản lý thí sinh và quản lý đăng kí | 17/06/2021 | 18/06/2021 | Tốt |
|  |  | Viết báo cáo chương 1 | 18/06/2021 | 19/06/2021 | Tốt |
|  |  | Code và hoàn thiện chức năng quản lý thí sinh | 19/06/2021 | 22/06/2021 | Tốt |
|  |  | Code và hoàn thiện chức năng quản lý đăng kí | 23/06/2021 | 28/07/2021 | Tốt |
| 03 | Nguyễn Hữu Quốc | Tạo form đăng kí và thống kê | 17/06/2021 | 18/06/2021 | Tốt |
|  |  | Viết báo cáo chương 2 | 18/06/2021 | 19/06/2021 | Tốt |
|  |  | Code và hoàn thiện chức năng đăng kí | 19/06/2021 | 22/06/2021 | Tốt |
|  |  | Code và hoàn thiện chức năng thống kê | 23/06/2021 | 28/07/2021 | Tốt |
|  |  | Làm video giới thiệu | 01/07/2021 | 02/07/2021 | Tốt |

# MỤC LỤc

[MỤC LỤC ii](#_Toc74235359)

[DANH MỤC HÌNH ẢNH iv](#_Toc74235360)

[DANH MỤC BẢNG BIỂU v](#_Toc74235361)

[DANH MỤC CÁC TỪ VIẾT TẮT vi](#_Toc74235362)

[LỜI MỞ ĐẦU 1](#_Toc74643749)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc74643750)

[1.1. Tổng quan về .NET framework 2](#_Toc74643751)

[1.1.1.Khái niệm ngôn ngữ lập trình C# 2](#_Toc74643752)

[1.1.2. Đặc điểm ngôn ngữ lập trình C# 2](#_Toc74643753)

[1.2. Tổng quan về Windows Forms 2](#_Toc74643754)

[1.3. Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2](#_Toc74643755)

[CHƯƠNG 2. phân tích và thiết kế ứng dụng 3](#_Toc74643756)

[2.1. Mô tả bài toán 3](#_Toc74643757)

[2.2. Thiết kế các chức năng 3](#_Toc74643758)

[2.3. Thiết kế cơ sở dữ liệu 3](#_Toc74643759)

[2.3.1 Thiết kế cơ sở dữ liệu mức vật lý 3](#_Toc74643763)

[2.3.2. Tạo cơ sở dữ liệu 3](#_Toc74643764)

[2.3.2. Mô hình cơ sở dữ liệu (Database diagram) 3](#_Toc74643764)

[CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG 4](#_Toc74643760)

[3.1. Sơ đồ chức năng 4](#_Toc74643761)

[3.2. Thông tin chi tiết các form làm việc 4](#_Toc74643762)

[3.2.1. Form đăng nhập 4](#_Toc74643763)

[3.2.2. Form đăng kí 4](#_Toc74643763)

[3.2.3. Form quản lý người đăng ký 4](#_Toc74643764)

[3.2.4. Form quản lý kỳ thi 4](#_Toc74643763)

[3.2.5. Form quản lý thí sinh 4](#_Toc74643764)

[3.2.6. Form thống kê 4](#_Toc74643763)

[CHƯƠNG 4. KẾT QUẢ 5](#_Toc74643765)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc74643766)

[TÀI LIỆU THAM KHẢO 7](#_Toc74643767)

[PHỤ LỤC (nếu có) 8](#_Toc74643768)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình C# 2](#_Toc74235469)

[Hình 2.1 Database diagram 5](#_Toc74235470)

[Hình 3.1 Sơ đồ chức năng 5](#_Toc74235470)

[Hình 3.2 Form Đăng Nhập 5](#_Toc74235470)

[Hình 3.3 Form Đăng Kí 5](#_Toc74235470)

[Hình 3.4 Form Quản Lí Thí Sinh 5](#_Toc74235470)

[Hình 3.5 Form Quản Lí Kì Thi 5](#_Toc74235470)

[Hình 3.6 Form Quản Lí Đăng Kí 5](#_Toc74235470)

[Hình 3.7 Form Thống Kê 5](#_Toc74235470)

DANH MỤC BẢNG BIỂU

Bảng 2-1 Bảng KI\_THI. 4

Bảng 2-2 Bảng THI\_SINH. 5

Bảng 2-3 Bảng DANG\_KI. 6

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Ký hiệu | Chữ viết đầy đủ |
| 1 | CMT | Chứng minh thư |
| 2 | ĐT | Điện thoại |
| 3 | KT | Kì Thi |
| 4 | CSDL | Cơ sở dữ liệu |
| 5 | ĐK | Đăng Kí |

# LỜI MỞ ĐẦU

1. **Mục tiêu của đề tài**

* Nghiên cứu ngôn ngữ lập trình C# (Windows Forms)
* Nghiên cứu hệ quản trị CSDL Microsoft SQL Server 2016
* Xây dựng ứng dụng desktop để quản lý ứng dụng quản lý đăng ký thi chứng chỉ

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ngôn ngữ lập trình C# (Windows Forms)
* Nghiên cứu Microsoft SQL Server 2016
* Phân tích bài toán quản lý ứng dụng quản lý đăng ký thi chứng chỉ
* Phát triển ứng dụng để quản lý ứng dụng quản lý đăng ký thi chứng chỉ

1. **Kết cấu của đề tài**

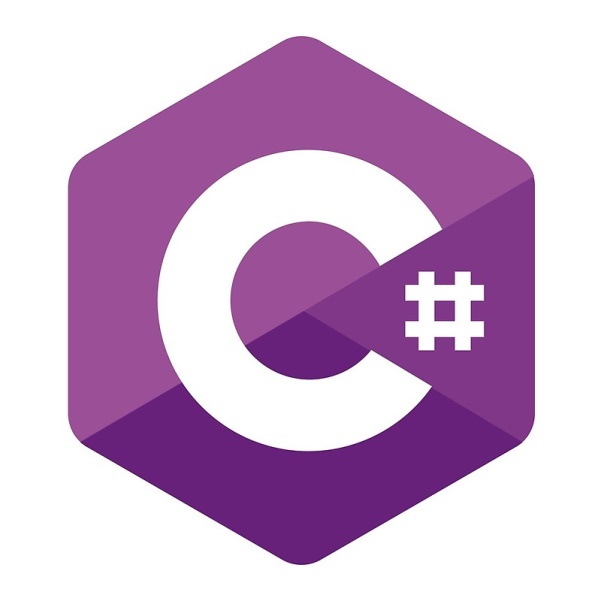
Đề tài được tổ chức gồm 4 chương nội dung và phần kết luận.

* **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

# CƠ SỞ LÝ THUYẾT

## Tổng quan về .NET framework

### Khái niệm ngôn ngữ lập trình C#



Hình .1 Ngôn ngữ lập trình C#

Ngôn ngữ lập trình C# (Hình 1.1) là một ngôn ngữ lập trình hướng đối tượng đa năng, mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

### Đặc điểm ngôn ngữ lập trình C#

* Mục đích sử dụng C#:

C# là một ngôn ngữ lập trình có mục đích chung để xây dựng các loại chương trình và ứng dụng khác nhau. C# có tính linh hoạt rất cao, tuy nhiên có ba lĩnh vực nó thường được sử dụng: Phát triển ứng dụng web, Ứng dụng Windows, Lập trình Games

* Ưu điểm của ngôn ngữ C#:

Lập trình C# là ngôn ngữ phổ biến nhất trong số các ngôn ngữ của Microsoft. Có những đặc điểm vượt trội như: Thời gian develop nhanh hơn, đường cong học tập thấp, khả năng mở rộng cao, sự hỗ trợ cộng đồng lớn,…

## Tổng quan về Windows Forms

Windows Form là một công nghệ của Microsoft, được sử dụng để phát triển rất nhiều ứng dụng nhờ giao diện kéo thả đơn giản, dễ code và rất tiện ích. Window Form rất thích hợp để làm các phần mềm quản lý, kế toán, thống kê và tạo ra các giao diện tuyệt đẹp và hiện đại

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

* Khái niệm : 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 những dữ liệu khác.

# phân tích và thiết kế ứng dụng

## Mô tả bài toán

* Thực trạng: một trung tâm khảo thí muốn xây dựng ứng dụng quản lý người đến đăng kí thi chứng chỉ (ngoại ngữ, tin học).
* Giải pháp: Xây dựng một phân mềm/ ứng dụng để quản lý người đến đăng kí thi chứng chỉ (ngoại ngữ, tin học) bao gồm các chức năng chính: Đăng kí, Quản lí đăng kí, Thí sinh, Kì thi, Thông kê.

## Thiết kế các chức năng

* Đăng nhập: Khi người dùng sử dụng hệ thống thì cần phải đăng nhập vào hệ thống
* Đăng kí: Hiển thị những kì thi còn trống chỗ cho thí sinh đăng kí. Sau khi đăng kí xong thì tăng SoLuongDK
* Quản lí đăng kí của thí sinh Thêm/sửa/xóa/hiển thị/tìm kiếm
* Thí sinh Sửa/ hiển thị/tìm kiếm
* Kì thi Thêm/sửa/xóa/hiển thị/tìm kiếm kì thi
* Thống kê Số lượng các kì thi, số lượng đăng kí thi trong khoảng thời gian xác định.

## Thiết kế cơ sở dữ liệu

### Thiết kế cơ sở dữ liệu mức vật lý

**KI\_THI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Ràng buộc | Chú thích |
| MaKT | Varchar | 50 | Primary key | Mã kì thi |
| PhongThi | Varchar | 5 | Not null | Phòng thi |
| NgayThi | Date |  | Not null | Ngày thi |
| GioThi | Time |  | Not null | Giờ thi |
| SoLuongToiDa | Int |  | Not null | Số lượng tối đa |
| SoLuongDK | Int |  | Not null | Số lượng đăng kí |

**THI\_SINH**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Ràng buộc | Chú thích |
| SoDT | Varchar | 15 | Primary key | Số điện thoại |
| HoTen | Nvarchar | 100 | Not null | Họ tên |
| SoCMT | Varchar | 10 | Not null | Số chứng minh thư |
| NgaySinh | Date |  | Not null | Ngày Sinh |

**DANG\_KI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Ràng buộc | Chú thích |
| SoDT | Varchar | 15 | Primary key | Số điện thoại |
| MaKT | Varchar | 50 | Primary key | Mã kì thi |
| NgayDK | Date |  | Not null | Ngày đăng kí |
| LePhi | Int |  | Not null | Lệ phí |

### Tạo cơ sở dữ liệu

* Tạo cơ sở dữ liệu

Create database QLDANGKYTHI

Use QLDANGKYTHI

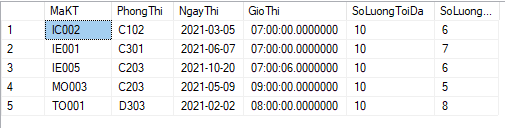
* Tạo bảng **KI\_THI**

Create table KI\_THI

( MaKT varchar(50) not null primary key, PhongThi varchar(5) not null,

NgayThi date not null, GioThi time not null, SoLuongToiDa int not null, SoLuongDK int not null )

Sau khi nhập dữ liệu, ta có kết quả của bảng ***KI\_THI*** như sau:



**Bảng 2-1 Bảng KI\_THI**

* Tạo bảng **THI\_SINH**

Create table THI\_SINH

(

SoDT varchar(15) not null primary key, HoTen nvarchar(100) not null,

SoCMT varchar(10) not null, NgaySinh date not null

)

Sau khi nhập dữ liệu, ta có kết quả của bảng ***THI\_SINH*** như sau:



**Bảng 2-2 Bảng THI\_SINH**

Tạo bảng DANG\_KI

Create table DANG\_KI

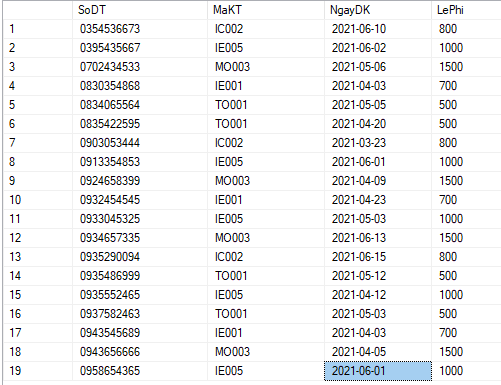
(

SoDT varchar(15) not null, MaKT varchar(50) not null, NgayDK date not null, LePhi int not null,

Primary key (sodt, makt)

)

Sau khi nhập dữ liệu, ta có kết quả của bảng ***DANG\_KI*** như sau:



**Bảng 2-3 Bảng DANG\_KI**

### Mô hình cơ sở dữ liệu (Database diagram)

**Hình 2.1: Database diagram**

# PHÁT TRIỂN ỨNG DỤNG

## Sơ đồ chức năng

Sơ đồ chức năng sau đây là toàn bộ mô hình mô phỏng các chức năng của hệ thống bao gồm những chức năng mà hệ thống có thể thực hiện được. Nhìn vào sơ đồ chúng ta có thể tóm lược được hoạt động của phần mềm quản lý này

**Đăng kí**

**Quản lí thí sinh**

**Phần mềm quản lí thi chứng chỉ**

**Quản lí kì thi**

**Quản lí đăng kí**

**Thống kê**

Hình 3-1: Sơ đồ chức năng

## Thông tin chi tiết các form làm việc

### Form Đăng nhập

### C:\Users\KimAnh\Downloads\dn.png

### Hình 3-2: Form đăng nhập

**3.2.1.1. Mô tả form**

Khi người dùng sử dụng hệ thống thì cần phải đăng nhập vào hệ thống. Mỗi một người dùng sẽ được cung cấp một tài khoản và mật khẩu để đăng nhập.

* Cách thực hiện:
* Nút đăng nhập: Kiểm tra tính hợp lệ
* + Nếu tài khoản mật khẩu đúng thì thông báo “ Đăng nhập thành công” vào giao diện chính
* + Nếu sai thì thông báo “ Đăng nhập thất bại” và yêu cầu nhập lại
  + Nút thoát: Kết thúc sử dụng hệ thống và gửi thông báo xác nhận

**3.2.1.2. Mô tả sự kiện**

* **Click ĐĂNG NHẬP:**

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

{

SqlConnection conn = new SqlConnection("Data Source=DESKTOP-GCVU3V9\\SQLEXPRESS;Initial Catalog=QLDANGKYTHI;Integrated Security=True");

try

{

conn.Open();

string tk = txtTK.Text;

string mk = txtMK.Text;

string sql = "select \* from nguoidung where TaiKhoan='" + tk + "' and MatKhau='" + mk + "'";

SqlCommand cmd = new SqlCommand(sql, conn);

SqlDataReader dta = cmd.ExecuteReader();

if(dta.Read()==true)

{

MessageBox.Show("Đăng nhập thành công");

frmMain KT = new frmMain();

KT.Show();

this.Hide();

}

else

{

MessageBox.Show("Đăng nhập thất bại");

}

}

catch

{

MessageBox.Show("Lỗi kết nối");

}

}

* **Click THOÁT:**

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

{

DialogResult dr;

dr = MessageBox.Show("Bạn có muốn thoát khỏi chương trình không ?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

Application.Exit();

}

### Form Đăng Kí

****

***Hình 3-3: Form Đăng Kí***

**3.2.2.1. Mô tả form**

Màn hình có các mục sau:

* Họ và tên: Người dùng nhập họ và tên của người đăng kí
* Số CMT: Nhập số CMT của người đăng kí
* Số ĐT: Nhập số ĐT của người đăng kí
* Ngày Sinh: Nhập ngày sinh của người đăng kí
* Mã kì thi đăng kí: Chọn mã kì thi đăng kí được hiển thị ở bảng. Nếu chọn kì thi đã đầy thí sinh, hệ thống sẽ gửi thông báo để người dùng chọn mã kì thi khác.
* Lệ Phí: Nhập lệ phí thi
* Ngày đăng kí: Chọn ngày đến đăng kí thi
* Nút đăng kí: Đăng kí kì thi và tăng số lượng đăng kí vào bảng CSDL
* Nút xóa: Xóa toàn bộ thông tin vừa nhập
* Note: Form cũng có chức nặng khi một thí sinh củ tới đăng kí thì chỉ cần nhập SĐT nếu đã tồn tại thì hệ thống sẽ thông báo là thí sinh từng đăng kí, lúc đó hệ thống sẽ tự động điền những thông tin của thí sinh này vào bảng và người dùng có thể sửa ngay lúc này

**3.2.2.2. Mô tả sự kiện**

* **Load Form**

private void frmDK\_Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("LỖI KẾT NỐI!");

}

string sQuerye = "select \* from KI\_THI";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dataGridViewKT.DataSource = KT;

dataGridViewKT.AutoSizeColumnsMode =

DataGridViewAutoSizeColumnsMode.Fill; con.Close();

txtMKT.DataSource = KT;

txtMKT.DisplayMember = "MaKT";

con.Close();

}

* **Click ĐĂNG KÍ**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("LỖI KẾT NỐI", "THÔNG BÁO", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sHoTen = txtHVT.Text;

string sSoDT = txtSDT.Text;

string sSoCMT = txtSCMT.Text;

string sMaKT = txtMKT.Text;

string sLePhi = txtLP.Text;

string sNgaySinh = NgaySinh.Value.ToString("yyyy-MM-d");

string sNgayDK = NgayDangKi.Value.ToString("yyyy-MM-d");

string sQuery2 = "insert into DANG\_KI values (@SoDT, @MaKT, @NgayDK, @LePhi)";

string sQuery = "insert into THI\_SINH values (@SoDT, @HoTen, @SoCMT, @NgaySinh)";

string Squery3 = "update KI\_THI set SoLuongDK = SoLuongDK + 1 where MaKT = @MaKT";

string sQuery4 = "update THI\_SINH set HoTen = @HoTen, SoCMT = @SoCMT, NgaySinh = @NgaySinh where SoDT = @SoDT";

SqlCommand cmd = new SqlCommand(sQuery, con);

SqlCommand cmd1 = new SqlCommand(sQuery2, con);

SqlCommand cmd2 = new SqlCommand(Squery3, con);

SqlCommand cmd3 = new SqlCommand(sQuery4, con);

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

cmd.Parameters.AddWithValue("@HoTen", sHoTen);

cmd.Parameters.AddWithValue("@SoCMT", sSoCMT);

cmd.Parameters.AddWithValue("@NgaySinh", sNgaySinh);

cmd1.Parameters.AddWithValue("@SoDT", sSoDT);

cmd1.Parameters.AddWithValue("@MaKT", sMaKT);

cmd1.Parameters.AddWithValue("@NgayDK", sNgayDK);

cmd1.Parameters.AddWithValue("@LePhi", sLePhi);

cmd2.Parameters.AddWithValue("@MaKT", sMaKT);

cmd3.Parameters.AddWithValue("@SoDT", sSoDT);

cmd3.Parameters.AddWithValue("@HoTen", sHoTen);

cmd3.Parameters.AddWithValue("@SoCMT", sSoCMT);

cmd3.Parameters.AddWithValue("@NgaySinh", sNgaySinh);

string sCheck = "Select \* from THI\_SINH where SoDT = '" + txtSDT.Text + "'";

SqlCommand cmdC = new SqlCommand(sCheck, con);

SqlDataReader dtr = cmdC.ExecuteReader();

if (dtr.Read() == true)

{

try

{

dtr.Close();

cmd1.ExecuteNonQuery();

cmd2.ExecuteNonQuery();

cmd3.ExecuteNonQuery();

MessageBox.Show("ĐĂNG KÍ THÀNH CÔNG!!!", "THÔNG BÁO", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtSCMT.Text = "";

txtHVT.Text = "";

txtLP.Text = "";

txtMKT.Text = "";

txtSDT.Text = "";

}

catch

{

MessageBox.Show("LỖI! KIỂM TRA LẠI THÔNG TIN", "THÔNG BÁO", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

try

{

dtr.Close(); cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

cmd2.ExecuteNonQuery();

MessageBox.Show("Đăng kí thành công!!!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtSCMT.Text = ""; txtHVT.Text = ""; txtLP.Text = ""; txtMKT.Text = ""; txtSDT.Text = "";

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi, vui lòng kiểm tra lại thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

string sQuerye = "select \* from KI\_THI";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dataGridViewKT.DataSource = KT; dataGridViewKT.AutoSizeColumnsMode =

DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* **Click Xóa**

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

{

txtSCMT.Text = "";

txtHVT.Text = "";

txtLP.Text = "";

txtMKT.Text = "";

txtSDT.Text = "";

}

* **TEXT CHANGED:**

private void txtSDT\_TextChanged(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("LỖI KẾT NỐI!", "THÔNG BÁO", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sCheck = "Select \* from THI\_SINH where SoDT = '" + txtSDT.Text + "'";

SqlCommand cmdC = new SqlCommand(sCheck, con);

SqlDataReader dtr = cmdC.ExecuteReader();

if (dtr.Read() == true)

{

txtHVT.Text = dtr["HoTen"].ToString();

txtSCMT.Text = dtr["SoCMT"].ToString();

NgaySinh.Text = dtr["NgaySinh"].ToString();

MessageBox.Show("Thí sinh đã đăng kí!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

con.Close();

}

* **Cell Content Double Click:**

private void dataGridViewKT\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

int i, y, z; i = dataGridViewKT.CurrentRow.Index;

y = Convert.ToInt32(dataGridViewKT.Rows[i].Cells[4].Value.ToString());

z = Convert.ToInt32(dataGridViewKT.Rows[i].Cells[5].Value.ToString());

if (y == z)

{

MessageBox.Show("Thí sinh đầy, xin mời chọn lại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

txtMKT.Text = dataGridViewKT.Rows[i].Cells[0].Value.ToString();

MessageBox.Show("Chọn thành công mã kì thi: " + txtMKT.Text + ".", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

### Form Quản Lí Thí Sinh



***Hình 3-4: Form Quản lí thí sinh***

**3.2.3.1. Mô tả form**

Màn hình có các mục sau:

* Tên thí sinh: Hiển thị tên thí sinh được chọn ở bảng để chỉnh sửa nếu sai sót
* Số CMT: Hiển thị CMT thí sinh được chọn ở bảng để chỉnh sửa nếu sai sót
* Ngày Sinh: Chỉnh sửa nếu có sai sót
* Số ĐT: Hiển thị số ĐT của thí sinh được chọn
* Nút chỉnh sửa: Lưu lại thí sinh sau khi chỉnh sửa
* Nút xóa: Xóa thông tin thí sinh đã đăng kí
* Nhập từ khóa: Nhập vào số điện thoại của thí sinh cần tìm kiếm.
* Nút tìm kiếm: Tìm ra đăng kí của thí sinh đúng với từ khóa ( Nhập đến đâu hiển thị đến đó )

**3.2.3.2. Mô tả sự kiện**

* **Load form:**

private void frmTS\_Load(object sender, EventArgs e)

{

dtgvTS.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuery = "select \* from THI\_SINH";

SqlCommand cmd = new SqlCommand(sQuery, con);

DataTable KT = new DataTable(); SqlDataReader dr = cmd.ExecuteReader(); KT.Load(dr);

dtgvTS.DataSource = KT;

con.Close();

}

* **Text Changed TÌM KIẾM:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuery = "select \* from THI\_SINH where SoDT like '%" + txttukhoa.Text + "%'";

SqlCommand cmd = new SqlCommand(sQuery, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

KT.Load(dr);

dtgvTS.DataSource = KT;

dtgvTS.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* + **Cell Content Click:**

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

{

int i;

i = dtgvTS.CurrentRow.Index;

txtSDT.Text = dtgvTS.Rows[i].Cells[0].Value.ToString();

txtTTS.Text = dtgvTS.Rows[i].Cells[1].Value.ToString();

txtSCMT.Text = dtgvTS.Rows[i].Cells[2].Value.ToString();

dtNS.Text = dtgvTS.Rows[i].Cells[3].Value.ToString();

txtSDT.Enabled = false;

* **Click CHỈNH SỬA:**

}

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Llỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sHoTen = txtTTS.Text;

string sSoDT = txtSDT.Text;

string sSoCMT = txtSCMT.Text;

string sNgaySinh = dtNS.Value.ToString("yyyy-MM-d");

string sQuery = "update THI\_SINH set NgaySinh = @NgaySinh , HoTen = @HoTen, SoCMT = @SoCMT where SoDT = @SoDT ";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@HoTen", sHoTen);

cmd.Parameters.AddWithValue("@SoCMT", sSoCMT);

cmd.Parameters.AddWithValue("@NgaySinh", sNgaySinh);

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

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công!!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(" Lỗi cập nhật, kiểm tra lại thông tin", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "SELECT \* from THI\_SINH";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader();

KT.Load(dr);

dtgvTS.DataSource = KT;

dtgvTS.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* **Click XÓA:**

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

{

DialogResult ret = MessageBox.Show("Bạn có muốn xóa thí sinh này?", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sSoDT = txtSDT.Text;

string sQuery = "delete THI\_SINH where SoDT= @SoDT delete THI\_SINH where SoDT = @SoDT";

SqlCommand cmd = new SqlCommand(sQuery, con);

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

try

{

cmd.ExecuteNonQuery();

txtSDT.Text = "";

txtTTS.Text = "";

txtSCMT.Text = "";

txtSDT.Enabled = true;

MessageBox.Show("Xóa thành công!!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa dữ liệu, kiểm tra lại thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "select \* from THI\_SINH";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dtgvTS.DataSource = KT;

dtgvTS.AutoSizeColumnsMode =

DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

### Form Quản Lí Kì Thi



***Hình 3-5: Form Quản lí thí sinh***

**3.2.4.1. Mô tả Form**

Màn hình có các mục sau:

* Mã kì thi: Hiển thị kì thi được chọn/ Nhập để thêm mới 1 kì thi khác
* Phòng thi: Hiển thị phòng thi của kì thi được chọn/ Nhập vào phòng thi của kì thi mới
* Ngày thi: - Hiển thị ngày thi của kì thi được chọn/ Nhập vào ngày thi của kì thi mới
* Giờ thi: Hiển thị giờ thi của kì thi được chọn/ Nhập vào giờ thi của kì thi mới
* Số lượng tối đa: Hiển thị số lượng tối đa của kì thi được chọn/ Nhập vào số lượng thí sinh tối đa của kì thi mới
* Số lượng đăng kí: Hiển thị số lượng ĐK của kì thi
  + - * + Nút thêm: tạo mới một kì thi đã nhập
        + Nút chỉnh sửa: Lưu kì thi sau khi chỉnh sửa
        + Nút Xóa: Xóa kì thi đã chọn
* Nhập mã KT: Nhập vào mã KT cần tìm kiếm
* Nút tìm kiếm: Tìm nội dung đúng với từ khóa (nhập đến đâu hiển thị đến đó)

**3.2.4.2. Mô tả sự kiện**

* **Load form:**

private void frmKT\_Load(object sender, EventArgs e)

{

dtgvKT.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuery = "select \* from KI\_THI";

SqlCommand cmd = new SqlCommand(sQuery, con);

DataTable KT = new DataTable(); SqlDataReader dr = cmd.ExecuteReader(); KT.Load(dr);

dtgvKT.DataSource = KT;

txtMKT.DataSource = KT;

txtMKT.DisplayMember = "MaKT";

con.Close();

}

* **Cell Content Click:**

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

{

int i;

i = dtgvKT.CurrentRow.Index; txtMKT.Text =

dtgvKT.Rows[i].Cells[0].Value.ToString(); txtPT.Text =

dtgvKT.Rows[i].Cells[1].Value.ToString(); dtGT.Text =

dtgvKT.Rows[i].Cells[3].Value.ToString(); txtSLTD.Text =

dtgvKT.Rows[i].Cells[4].Value.ToString(); dtNT.Text =

dtgvKT.Rows[i].Cells[2].Value.ToString(); txtSLDK.Text =

dtgvKT.Rows[i].Cells[5].Value.ToString(); txtMKT.Enabled = false;

}

* **Click TÌM KIẾM:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuery = "select \* from KI\_THI where MaKT like '%" + txttukhoa.Text + "%'";

SqlCommand cmd = new SqlCommand(sQuery, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

KT.Load(dr);

dtgvKT.DataSource = KT;

dtgvKT.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill; con.Close();

}

* **Click CHỈNH SỬA:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sMakt = txtMKT.Text; string sPhongThi = txtPT.Text;

string sNgayThi = NgayThi.Value.ToString("yyyy-MM-d");

string sGioThi= GioThi.Value.ToString("HH:mm:ss");

string sSoLuongToiDa = txtSLTD.Text;

string sSoLuongDK = txtSLDK.Text;

string sQuery = "update KI\_THI set PhongThi = @PhongThi , NgayThi = @NgayThi , GioThi = @GioThi , SoLuongToiDa = @SoLuongToiDa, SoLuongDK = @SoLuongDK where MaKT = @MaKT";

SqlCommand cmd = new SqlCommand(sQuery, con); cmd.Parameters.AddWithValue("@MaKT", sMakt);

cmd.Parameters.AddWithValue("@PhongThi", sPhongThi);

cmd.Parameters.AddWithValue("@NgayThi", sNgayThi);

cmd.Parameters.AddWithValue("@GioThi", sGioThi);

cmd.Parameters.AddWithValue("@SoluongToiDa", sSoLuongToiDa);

cmd.Parameters.AddWithValue("@SoLuongDK", sSoLuongDK);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMKT.Text = ""; txtPT.Text = ""; txtSLDK.Text = ""; txtSLDK.Text = ""; txtMKT.Enabled = true;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi trong quá trình cập nhật thông tin", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "select \* from KI\_THI"; SqlCommand cmde = new SqlCommand(sQuerye, con); DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dtgvKT.DataSource = KT; dtgvKT.AutoSizeColumnsMode =

DataGridViewAutoSizeColumnsMode.Fill; con.Close();

}

* **Click THÊM:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuery = "insert into KI\_THI values (@MaKT, @PhongThi, @NgayThi, @GioThi, @SoLuongToiDa ,0)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaKT", txtMKT.Text);

cmd.Parameters.AddWithValue("@PhongThi", txtPT.Text);

cmd.Parameters.AddWithValue("@NgayThi", NgayThi.Value.ToString("yyyy-MM-dd"));

cmd.Parameters.AddWithValue("@GioThi", GioThi.Value.ToString("HH:mm:ss"));

cmd.Parameters.AddWithValue("@SoLuongToiDa", txtSLTD.Text);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMKT.Text = ""; txtPT.Text = ""; txtSLDK.Text = ""; txtSLTD.Text = ""; txtMKT.Enabled = true;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi trong quá trình thêm, hãy kiểm tra lại thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "select \* from KI\_THI";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dtgvKT.DataSource = KT;

dtgvKT.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* **Click XÓA:**

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

{

DialogResult ret = MessageBox.Show("Bạn muốn xóa kì thi này?", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sMaKT = txtMKT.Text;

string sQuery = "delete DANG\_KI where MaKT = @MaKT delete KI\_THI where MaKT = @MaKT";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaKT", sMaKT);

try

{

cmd.ExecuteNonQuery();

txtMKT.Text = "";

txtPT.Text = "";

txtSLDK.Text = "";

txtSLTD.Text = "";

txtMKT.Enabled = true;

MessageBox.Show("Xóa thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi trong quá trình xóa dữ liệu, kiểm tra lại thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "select \* from KI\_THI";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dtgvKT.DataSource = KT; dtgvKT.AutoSizeColumnsMode =

DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

### Form Quản Lí Đăng Kí



***Hình 3-6: Form Quản lí thí sinh***

**3.2.5.1. Mô tả Form**

Màn hình có các mục sau:

* Số ĐT: Hiển thị số ĐT của thí sinh được chọn ở bảng
* Mã Kì Thi: Hiển thị mã kì thi của thí sinh được chọn ở bảng
* Ngày đăng kí: Hiển thị ngày đăng kí của thí sinh được chọn ở bảng
* Lệ phí: Hiển thị lệ phí của thí sinh được chọn ở bảng, được chỉnh sửa nếu có sai sót
* Nút chỉnh sửa: Lưu lại thí sinh sau khi chỉnh sửa
* Nút xóa: Xóa đăng kí của thí sinh trong CSDL
* Nhập Số ĐT: Nhập vào số điện thoại đăng kí của thí sinh cần tìm kiếm
* Nút tìm kiếm: Tìm ra đăng kí của thí sinh đúng với từ khóa (nhập đến đâu hiển thị đến đó)

**3.2.5.2. Mô tả sự kiện**

* **Load form:**

private void frmQLDK\_Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi kết nối DB");

}

string sQuery = "select \* from DANG\_KI";

SqlCommand cmd = new SqlCommand(sQuery, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

KT.Load(dr);

dtgvDK.DataSource = KT;

dtgvDK.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

txtMKT.DataSource = KT;

txtMKT.DisplayMember = "MaKT";

con.Close();

}

* **Click TÌM KIẾM:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuery = "select \* from DANG\_KI where SoDT like '%" + txttukhoa.Text + "%'";

SqlCommand cmd = new SqlCommand(sQuery, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

KT.Load(dr);

dtgvDK.DataSource = KT;

dtgvDK.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* **Click Xóa:**

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

{

DialogResult ret = MessageBox.Show("Bạn có muốn xóa kì thi này?", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sSoDT = txtSDT.Text;

string sQuery = "delete DANG\_KI where SoDT= @SoDT delete DANG\_KI where SoDT = @SoDT";

SqlCommand cmd = new SqlCommand(sQuery, con);

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

try

{

cmd.ExecuteNonQuery();

txtSDT.Text = "";

txtMKT.Text = "";

txtLP.Text = "";

txtSDT.Enabled = true;

MessageBox.Show("Xóa thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa dữ liệu, kiểm tra lại thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "select \* from DANG\_KI";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader(); KT.Load(dr);

dtgvDK.DataSource = KT;

dtgvDK.AutoSizeColumnsMode =

DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

}

* **Click CHỈNH SỬA:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sMaKT = txtMKT.Text;

string sSDT = txtSDT.Text;

string sLP = txtLP.Text;

string sNgayDK = dtngaydk.Value.ToString("yyyy-MM-d");

string sQuery = "update DANG\_KI set NgayDK = @NgayDK, LePhi = @LePhi where SoDT = @SoDT and MaKT = @MaKT";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@NgayDK", sNgayDK);

cmd.Parameters.AddWithValue("@MaKT", sMaKT);

cmd.Parameters.AddWithValue("@LePhi", sLP);

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

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi, kiểm tra lại thông tin.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string sQuerye = "select \* from DANG\_KI";

SqlCommand cmde = new SqlCommand(sQuerye, con);

DataTable KT = new DataTable();

SqlDataReader dr = cmde.ExecuteReader();

KT.Load(dr); dtgvDK.DataSource = KT;

dtgvDK.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* **Cell Click:**

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

{

int i;

i = dtgvDK.CurrentRow.Index;

txtSDT.Text =

dtgvDK.Rows[i].Cells[0].Value.ToString();

txtMKT.Text =

dtgvDK.Rows[i].Cells[1].Value.ToString();

dtngaydk.Text =

dtgvDK.Rows[i].Cells[2].Value.ToString();

txtLP.Text =

dtgvDK.Rows[i].Cells[3].Value.ToString();

txtSDT.Enabled = false;

txtMKT.Enabled = false;

}

### Form Thống Kê



***Hình 3-7: Form Thống Kê***

**3.2.6.1. Mô tả Form**

Màn hình có các mục sau:

* Từ ngày: Chọn thời gian bắt đầu
* Đến ngày: Chọn thời gian kết thúc
* Nút Số lượng đăng kí: Xem số lượng ĐK của thí sinh trong khoảng thời gian đã chọn
* Nút Số lượng kì thi: Xem số lượng kì thi trong khoảng thời gian đã chọn

**3.2.6.2. Mô tả sự kiện**

* **Click SỐ LƯỢNG KÌ THI:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("Xảy ra lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string TruyVan = "select LEFT(MaKT, 2) Ki\_Thi, count(MaKT) So\_Luong\_Ki\_Thi from KI\_THI where NgayThi between @from and @to group by LEFT(MaKT, 2)";

string sFrom = dtFrom.Value.ToString("yyyy-MM-dd");

string sTo = dtTo.Value.ToString("yyyy-MM-dd");

SqlCommand cmd = new SqlCommand(TruyVan, con);

cmd.Parameters.AddWithValue("@from", sFrom);

cmd.Parameters.AddWithValue("to", sTo);

DataTable TK = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

TK.Load(dr); dtgvTK.DataSource = TK;

dtgvTK.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close();

}

* **Click SỐ LƯỢNG ĐĂNG KÍ:**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch

{

MessageBox.Show("Xảy ra lỗi khi kết nối!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

string TruyVan = "select LEFT(MaKT, 2) Ki\_Thi, count(SoDT) So\_Luong\_Dang\_Ki, sum(LePhi) Tong\_Thu from DANG\_KI where NgayDK between @from and @to group by LEFT(MaKT, 2)";

string sFrom = dtFrom.Value.ToString("yyyy-MM-dd");

string sTo = dtTo.Value.ToString("yyyy-MM-dd");

SqlCommand cmd = new SqlCommand(TruyVan, con);

cmd.Parameters.AddWithValue("@from", sFrom);

cmd.Parameters.AddWithValue("to", sTo);

DataTable TK = new DataTable();

SqlDataReader dr = cmd.ExecuteReader();

TK.Load(dr); dtgvTK.DataSource = TK;

dtgvTK.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

con.Close( );

**CHƯƠNG 4: KẾT QUẢ**

* 1. **Kết quả đạt được:**
* Phân tích và thiết kế được ứng dụng quản lý đăng kí thi chứng chỉ và các chức năng đơn giản
* Thiết kế được CSDL, bảng, ràng buộc dữ liệu, thiết lập được quan hệ giữa các bảng dữ liệu, các ràng buộc toàn vẹn dữ liệu trong CSDL (khóa chính, khóa ngoại). Hoàn thành được cơ sở dữ liệu đơn giản.
* Thiết kế được các form và viết được code để chạy thành chương trình.
* Hoàn thiện được ứng dụng quản lý đăng kí thi chứng chỉ và các chức năng đơn giản.
* Biết tìm hiểu và sửa lỗi gặp phải trong quá trình làm bài.
* Hoàn thiện báo cáo theo yêu cầu nội dung đã đề ra
* Rèn luyện kỹ năng làm việc nhóm được tốt hơn.
  1. **Ưu điểm của hệ thống :**
* Hệ thống có thể thực hiện các chức năng: thêm, tìm kiếm, chỉnh sửa, xoá ở các form.
* Thống kê được doanh thu theo thời gian

**4.3 Nhược điểm của hệ thống**

* Với thời gian thực tập và khả năng còn hạn hẹp, ứng dụng quản lý đăng kí thi chứng chỉ còn đơn giản và thiếu sót.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* **Kết luận:**

Qua quá trình nghiên cứu cùng sự hướng dẫn tận tình của giảng viên Cao Thị Nhâm đã giúp nhóm em hoàn thành đề tài “***Ứng dụng quản lý đăng kí thi chứng chỉ***”. Các chức năng của ứng dụng được thiết kế thân thiện với người dùng, giúp người dùng dễ dàng thực hiện được các thao tác một cách thuận tiện. Tuy nhiên ứng dụng thiết kế còn đơn giản, chưa bắt mắt và còn nhiều hạn chế. Nhóm chúng em rất mong nhận được ý kiến của thầy cô để có thể xây dựng ứng dụng hoàn thiện hơn.

* **Hướng phát triển:**
* Tìm hiểu và cải thiện các chức năng chưa thực hiện được để tối ưu cho phần mềm
* Thiết kế giao diện phần mềm đẹp, rõ ràng hơn thân thiện với người dùng hơn.