TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**PROJECT 2**

NGÀNH CÔNG NGHỆ THÔNG TIN

**THIẾT KẾ CHƯƠNG TRÌNH LẤY THÔNG TIN CỦA MÁY TÍNH GCI**

Sinh viên thực hiện: **Nguyễn Văn Tú**

Lớp CN-CNTT 01 - K58

Giáo viên hướng dẫn: Th.S. **Nguyễn Đức Tiến**

HÀ NỘI 10 -2015

PHIẾU GIAO NHIỆM VỤ MÔN HỌC

**1. Thông tin về sinh viên**

Họ và tên sinh viên: Nguyễn Văn Tú

Điện thoại liên lạc: 0977870028 Email: computer.nvt@gmail.com

Lớp: CN-CNTT 01 - K58 Hệ đào tạo: Đại học chính quy

**2. Sản phẩm mục tiêu của môn học**

Tìm hiểu về WMI, sử dụng công cụ WMI Code Creator để tạo mã sử dụng WMI bằng C#,VBScript,VB.NET (Trong Project sử dụng ngôn ngữ C#) để lấy các thông tin của máy tính. Áp dụng vào thiết kế phần mềm xem thông tin của máy tính GCI.

**3. Các tính năng**

***Các tính năng bắt buộc:***

* Lập trình/Tìm hiểu để hoàn thành sản phẩm: Có
* Sử dụng một kho repository để lưu trữ các phiên bản của project: Có

https://github.com/ComputerNguyen/GitHub.git

***Các tính năng tùy chọn:***

|  |  |
| --- | --- |
| Tính Năng Tùy Chọn | Áp dụng |
| Sử dụng các đối tượng trong ToolBox như:Text Box, Label, LinkLabel, GroupBox, ComboBox,… | **Có** |
| Lấy thông tin về máy tính :CPU,GPU,… | **Có** |
| Hiển thị thông tin trên GUI | **Có** |
| Kết nối Database SQL Server | **Không** |
| Giao diện Ribbon | **Không** |
| Quản lí mã nguồn | **Có** |
| In dữ liệu ra file | **Có** |
| Setup | **Có** |

***Tính năng khác nếu có:*** Kiểm tra các thông số máy tính.Lưu trữ thông tin máy tính.

MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ MÔN HỌC 2](#_Toc433153285)

[MỤC LỤC 3](#_Toc433153286)

[DANH MỤC HÌNH VẼ 4](#_Toc433153287)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 5](#_Toc433153288)

[1. NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG 6](#_Toc433153289)

[1.1. Các sản phẩm tương tự 6](#_Toc433153290)

[1.2. Các công nghệ kỹ thuật liên quan 7](#_Toc433153291)

[1.2.1. Microsoft Visual Studio 7](#_Toc433153292)

[1.2.2. GitHub 8](#_Toc433153293)

[1.2.3. WMI (Windows Management Instrumentation) 9](#_Toc433153294)

[2. PHÂN TÍCH THIẾT KẾ 10](#_Toc433153295)

[2.1. Mô hình kết nối 10](#_Toc433153296)

[2.2. Mô hình giao diện 11](#_Toc433153297)

[2.3. Các thành phần chức năng 12](#_Toc433153298)

[2.3.1. Các hàm, cấu trúc 12](#_Toc433153299)

[2.3.2. Một số thông tin khác 14](#_Toc433153300)

[3. SẢN PHẨM KẾT QUẢ 16](#_Toc433153301)

[3.1. Giao diện 16](#_Toc433153302)

[3.2. Lưu ý 21](#_Toc433153303)

[TÀI LIỆU THAM KHẢO 22](#_Toc433153304)

DANH MỤC HÌNH VẼ

[Hình 1. Get Hardware Information 6](#_Toc433151210)

[Hình 2. CPU Z 6](#_Toc433151211)

[Hình 3. Microsoft Visual Studio 7](#_Toc433151212)

[Hình 4. GitHub 8](#_Toc433151213)

[Hình 5. WMI 9](#_Toc433151214)

[Hình 6. Mô hình kết nối 10](#_Toc433151215)

[Hình 7. Mô hình giao diện 11](#_Toc433151216)

[Hình 8. Giao diện Tab CPU 16](#_Toc433151217)

[Hình 9. Giao diện Tab Mainboard 16](#_Toc433151218)

[Hình 10. Giao diện Tab Memory 17](#_Toc433151219)

[Hình 11. Giao diện Tab Graphics 17](#_Toc433151220)

[Hình 12. Giao diện Tab Graphics 18](#_Toc433151221)

[Hình 13. Giao diện Tab Graphics 18](#_Toc433151222)

[Hình 14.giao diện Tab Battery 19](#_Toc433151223)

[Hình 15. Giao diện Tab About 19](#_Toc433151224)

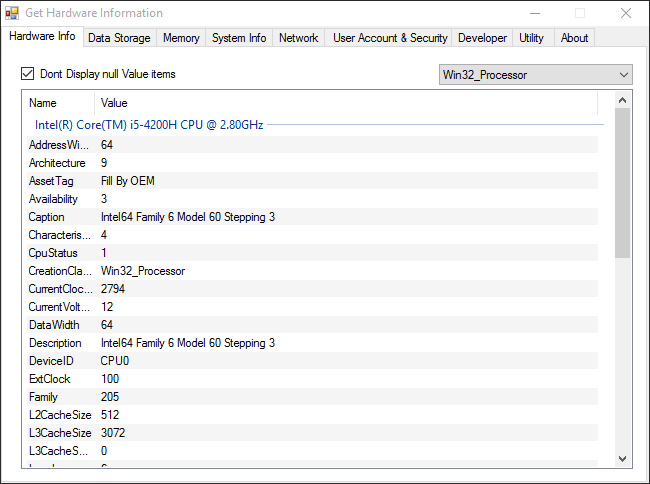
[Hình 16. Giao diện Save Info 20](#_Toc433151225)

DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Từ viết tắt** | **Ý nghĩa** |
| 1 | WMI | Windows Management Instrumentation |
| 2 | GCI | Get Computer Information |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |

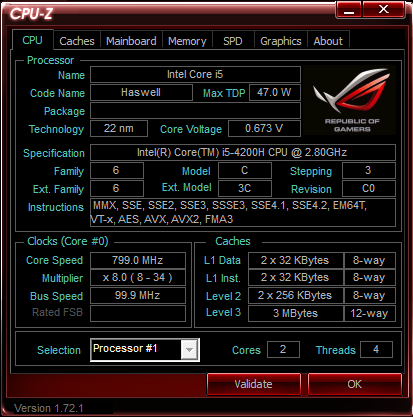
1. NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG
   1. Các sản phẩm tương tự

Get Hardware Information

**

Hình . Get Hardware Information

CPU Z

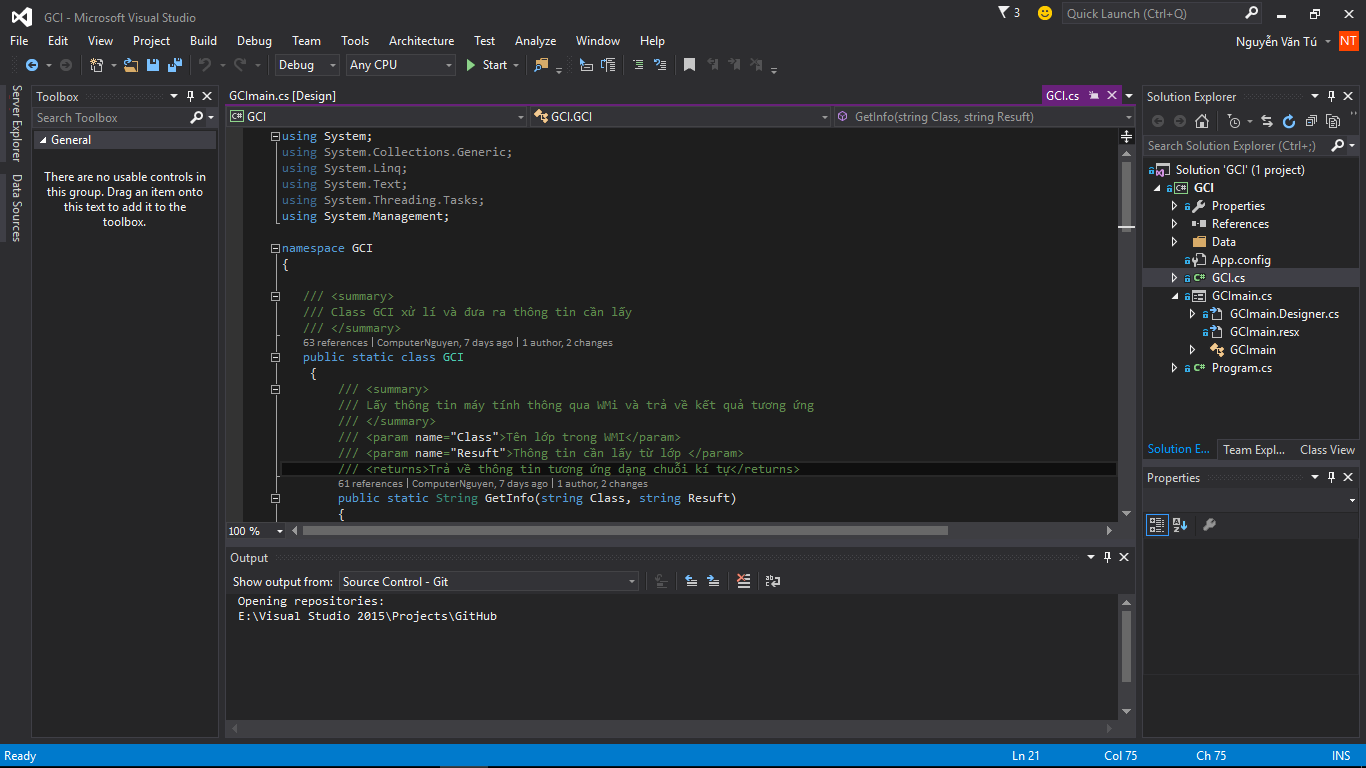


Hình . CPU Z

* 1. Các công nghệ kỹ thuật liên quan
     1. Microsoft Visual Studio

Là một công cụ đa năng, đặc biệt là trong việc tạo lập và phát triển phần mềm.

MVS hỗ trợ rất nhiều ngôn ngữ lập trình (C++,C#,VB…), công cụ (Spy++, Error lookup…), khả năng mở rộng cao với các extension và còn là một công cụ hỗ trợ debug rất mạnh.



Hình . Microsoft Visual Studio

Trong đồ án này, Microsoft Visual Studio được sử dụng để tạo giao diện và phần thực thi của chương trình. Microsoft Visual Studio Installer Projects tạo file setup.

* + 1. GitHub

Github [http://github.com](http://github.com/), còn được gọi là social network dành cho developer đi vào hoạt động tháng 2 năm 2008, là một dịch vụ sử dụng hệ thống quản lý phân tán GIT giúp người dùng lưu trữ source code cho các dự án.

GitHub cung cấp dịch vụ thương mại và cả tài khoản miễn phí cho các dự án nguồn mở. Theo khảo sát của người sử dụng Git vào năm 2009, Github hiện đang là server Git lưu trữ source code phổ biến nhất hiện nay.

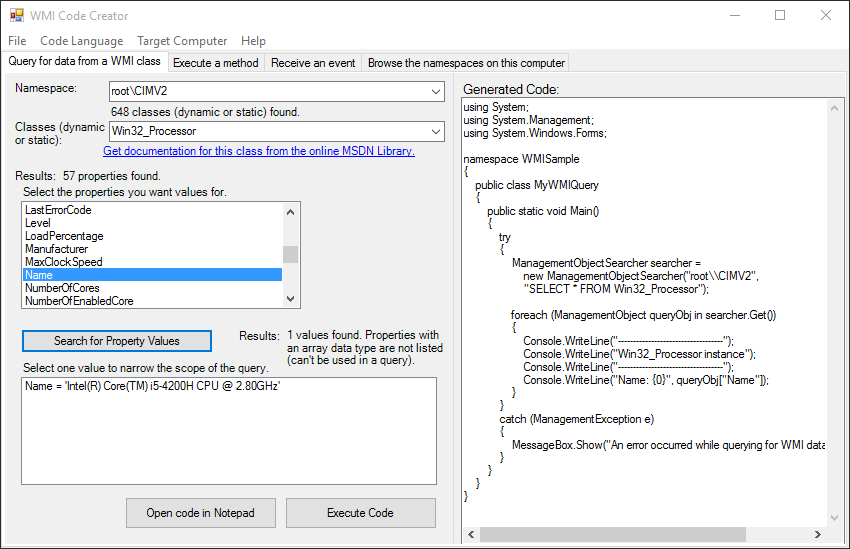


Hình . GitHub

Trong đồ án này, GitHub được dùng để lưu trữ mã nguồn và báo cáo môn học.

* + 1. WMI (Windows Management Instrumentation)

Windows Management Instrumentation là phần mở rộng của Windows Driver Model cung cấp một lớp giao tiếp giữa các thành phần hạ tầng với hệ điều hành. Nó cho phép các script có thể lấy thông tin, hoặc điều khiển các tác vụ hệ thống.



Hình . WMI

Trong đồ án này, WMI là công nghẹ được sử dụng với chức năng như là thư viện để sinh code.

1. PHÂN TÍCH THIẾT KẾ
   1. Mô hình kết nối

Sử dụng WMI để giao tiếp với máy tính lấy thông tin và xử lí hiển thị trên giao diện.

**Computer**

**WMI**

**Giao diện**

**Chương trình**

**Xử lí**

Hình . Mô hình kết nối

* 1. Mô hình giao diện

Graphics

Memory

[Icon] Get Computer Information

X

-

[]

GroupBox bao gồm các Label,TextBox,ComboBox,…

Chứa các thông tin hiển thị

CPU

MainBoard

Battery

About

Hình . Mô hình giao diện

* 1. Các thành phần chức năng
     1. Các hàm, cấu trúc

ManagementObjectSearcher("root\\CIMV2", "SELECT \* FROM " + Class);

Là phương thức khởi tạo của lớp ManagementObjectSearcher (lớp này là một trong những điểm thường xuyên nhất được sử dụng nhập để lấy thông tin quản lý máy tính).Khi khởi tạo, thì đối tượng mà phương thức yêu cầu sẽ là đầu vào để WMI thực hiện truy vấn.

public static String GetInfo(string Class, string Resuft)

{

ManagementObjectSearcher searcher = new ManagementObjectSearcher("root\\CIMV2", "SELECT \* FROM " + Class);

foreach (ManagementObject wmi in searcher.Get())

{ //Bẫy lỗi nếu không đúng cú pháp hoặc không tìm kiếm được trả về giá trị tên Resuft + ": Unknown"

try

{

return wmi.GetPropertyValue(Resuft).ToString();

}

catch { }

}

return Resuft + ": Unknown";

}

Khi gọi phương thức GetInfo() thì ManagementObjectSearcher sẽ thực hiện truy vấn trả lại thông tin của đối tượng cần truy vấn ở đây Class sẽ đại diện cho một lớp thông tin trong WMI, và Resuft là thuộc tính trong lớp đó mà muốn truy vấn thông tin.Phương thức sẽ trả về thông tin được truy vấn được ở dạng chuỗi.

public static string GetRamSize()

{

ManagementScope oMs = new ManagementScope();

ObjectQuery oQuery = new ObjectQuery("SELECT Capacity FROM Win32\_PhysicalMemory");

ManagementObjectSearcher oSearcher = new ManagementObjectSearcher(oMs, oQuery);

ManagementObjectCollection oCollection = oSearcher.Get();

long MemSize = 0;

long mCap = 0;

//

foreach (ManagementObject obj in oCollection)

{

mCap = Convert.ToInt64(obj["Capacity"]);

MemSize += mCap;

}

MemSize = (MemSize / 1024) / 1024;

return MemSize.ToString() + "MB";

}

Phương thức GetRamSize() cho phép ManagementObjectCollection thực hiện truy vấn đến lớp Win32\_PhysicalMemory và lấy thông tin của thuộc tính Capacity . Sau đó trả về tổng dung lượng Ram dưới dạng chuỗi.

public static string GetNoRamSlots()

{

int MemSlots = 0;

ManagementScope oMs = new ManagementScope();

ObjectQuery oQuery2 = new ObjectQuery("SELECT MemoryDevices FROM Win32\_PhysicalMemoryArray");

ManagementObjectSearcher oSearcher2 = new ManagementObjectSearcher(oMs, oQuery2);

ManagementObjectCollection oCollection2 = oSearcher2.Get();

foreach (ManagementObject obj in oCollection2)

{

MemSlots = Convert.ToInt32(obj["MemoryDevices"]);

}

return MemSlots.ToString();

}

Tương tự như phương thức GetRamSize() thì phương thức GetNoRamSlots() trả về thông tin của thuộc tính MemoryDevices trong lớp Win32\_PhysicalMemoryArray.

public GCImain()

{

//

//Hàm InitializeComponent() dùng để khởi tạo các đối tượng có trên form như textbox,combobox...

//

InitializeComponent();

}

Hàm GCImain() là hàm khởi tạo các đối tượng trên Form, cập nhật thay đổi trên giao diện Design.

public void GCImain\_Load(object sender, EventArgs e)

Đây là hàm lấy thông tin CPU,Mainboard,Memory,Battery,OS và hiển thị . Trong hàm này các textbox nhận kết quả khi gọi đến phương thức GetInfo ở trên theo mỗi thuộc tính cần thiết.

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

Đây là hàm lấy và hiển thị thông tin của GPU. Trong đó lựa chọn lớp cần truy vấn thông tin DisplayControllerConfiguration và DisplayConfiguration dựa vào combobox sau đó lấy các thông tin của các thuộc tính vào các textbox.

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

Hàm tạo sự kiện cho link .

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

{

if (saveFileDialog1.ShowDialog()==DialogResult.OK)

{

File.WriteAllText(saveFileDialog1.FileName,saveCPU+saveMain+saveMem+saveGra+saveBat+saveOS);

}

}

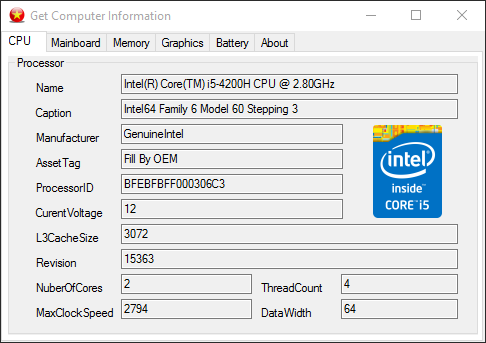
Hàm để lưu dữ liệu được xem ra file.

* + 1. Một số thông tin khác

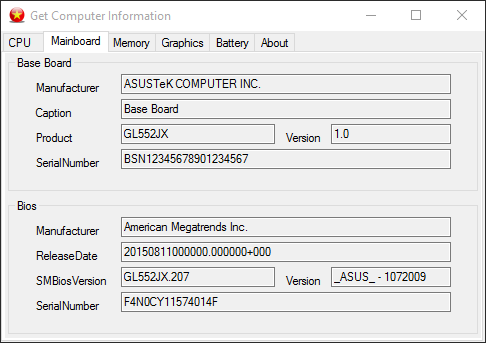
Chức năng của các lớp WMI:

* Win32\_Processor là lớp chứa các thông tin về thiết bị chạy trên hệ điều hành Windows. Nó bao gồm nhiều các thuộc tính cho phép ta lấy thông tin về các thiết bị như CPU,Cache,…
* Win32\_BaseBoard là lớp đại diện WMI cho phép lấy thông tin về bo mạch chủ hay hệ thống
* Win32\_BIOS là lớp WMI chứa các thuộc tính của các dịch vụ hệ thống vào ra
* Win32\_DiskDrive là lớp WMI chứa các thuộc tính về ổ đĩa vật lí trên máy tính hệ điều hành Windows.
* Win32\_PhysicalMemory là lớp WMI chứa các thuộc tính về bộ nhớ vật lí trên máy tính chạy hệ điều hành Windows.
* Win32\_PhysicalMemoryArray là lớp WMI chứa các thuộc tính về hệ thống máy tính của bộ nhớ vật lí, bao gồm các thiết bị lưu trữ, dung lượng bộ nhớ có sẵn,…
* Win32\_Battery là lớp WMI chứa các thuộc tính về pin trên máy tính hệ điều hành Windows như dung lượng, hãng sản xuất, thời gian sử dụng…
* Win32\_OperatingSystem là lớp WMI chứa các thông tin về hệ điều hành Windows được cài đặt.
* Win32\_DisplayControllerConfiguration là lớp WMI chứa các thuộc tính về thông tin cấu hình của hệ thống điều khiển hiển thị của máy tính hệ điều hành Windows.
* Win32\_DisplayConfiguration là lớp WMI chứa các thuộc tính về thông tin cấu hình của hệ thống hiển thị của máy tính hệ điều hành Windows

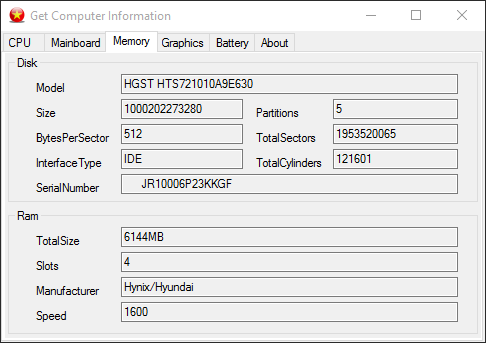
1. SẢN PHẨM KẾT QUẢ
   1. Giao diện



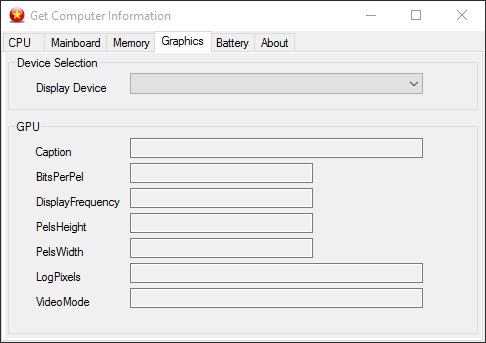
Hình . Giao diện Tab CPU



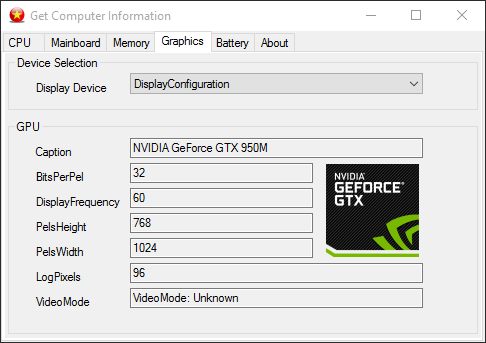
Hình . Giao diện Tab Mainboard



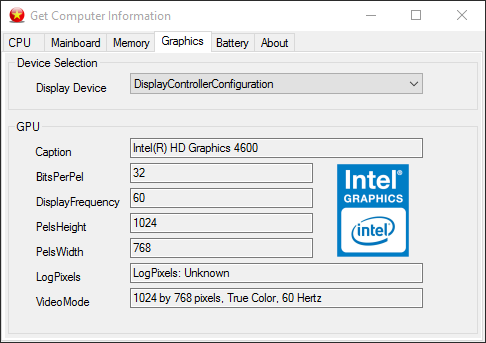
Hình . Giao diện Tab Memory



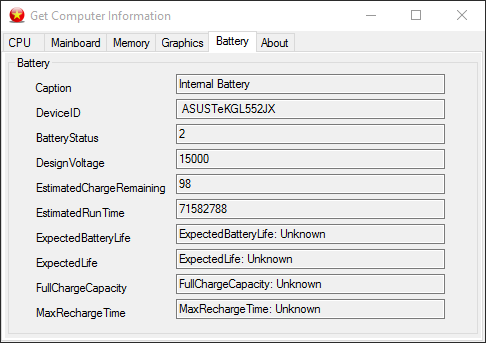
Hình . Giao diện Tab Graphics



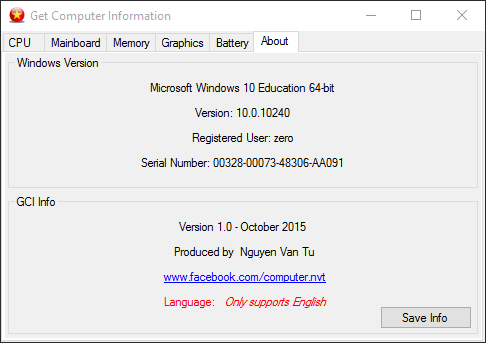
Hình . Giao diện Tab Graphics



Hình . Giao diện Tab Graphics

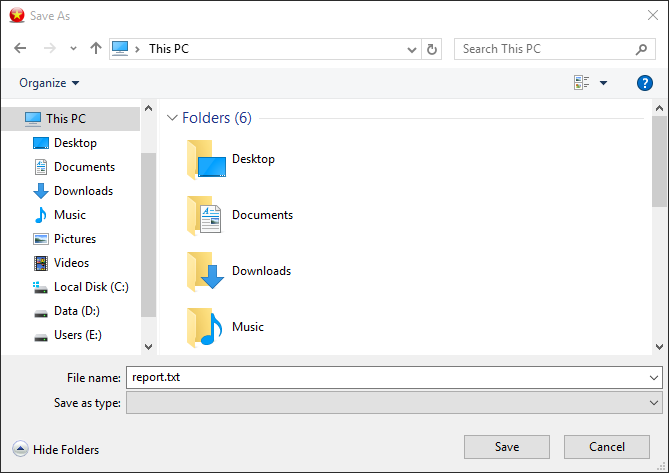


Hình .giao diện Tab Battery



Hình . Giao diện Tab About

Khi click vào nút Save Info ta được giao diện lưu file:



Hình . Giao diện Save Info

* 1. Lưu ý

Chương trình có thể sẽ cần quyền admin để hoạt động.

Yêu cầu .Net FrameWork 4.5

TÀI LIỆU THAM KHẢO

* *Tham khảo các project mã nguồn mở:*
* [*http://www.codescratcher.com/windows-forms/get-computer-hardware-information-using-c/*](http://www.codescratcher.com/windows-forms/get-computer-hardware-information-using-c/)
* [*http://www.codeproject.com/Articles/17973/How-To-Get-Hardware-Information-CPU-ID-MainBoard-I*](http://www.codeproject.com/Articles/17973/How-To-Get-Hardware-Information-CPU-ID-MainBoard-I)
* [*http://www.codeproject.com/Articles/568476/Creating-an-MSI-Setup-Package-for-Csharp-Windows*](http://www.codeproject.com/Articles/568476/Creating-an-MSI-Setup-Package-for-Csharp-Windows)
* *Video hướng dẫn lập trình Winform:*
* [*https://www.youtube.com/watch?v=165SCHk4Jic*](https://www.youtube.com/watch?v=165SCHk4Jic)