TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**Project 2**

**XÂY DỰNG CHƯƠNG TRÌNH   
THEO DÕI Ổ CỨNG BẰNG S.M.A.R.T**

Sinh viên thực hiện: **Vũ Tuấn Anh**

Lớp CN-CNTT K57

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

HÀ NỘI 06 -2014

# PHIẾU GIAO NHIỆM VỤ MÔN HỌC

**1. Thông tin về sinh viên**

Họ và tên sinh viên: Vũ Tuấn Anh

Điện thoại liên lạc: 01676614597 Email:tuananh.vga@gmail.com

Lớp: CN-CNTT K57 Hệ đào tạo: Đại học chính quy

**2. Sản phẩm mục tiêu của môn học**

Xây dựng chương trình lấy thông tin về trạng thái của ổ đĩa cứng bằng công nghệ S.M.A.R.T

**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/tuananhvga/SMART-Monitor>

**Các tính năng tùy chọn:**

|  |  |
| --- | --- |
| Tính năng | Có/Không |
| Sử dụng các đối tượng trong toolbox như button, text box, combo box, list box | Có |
| Lấy thông tin về trạng thái của ổ cứng | Có |
| Dự đoán khả năng ổ cứng có thể gặp vấn đề | Có |
| Hiển thị thông tin trên GUI | Có |
| Đặt lịch trình lấy dữ liệu theo thời gian | Không |
| Cảnh báo khi ổ cứng gặp vấn đề | Không |
| Sử dụng Word/Excel để vẽ giao diện chương trình, 5 form | Không |
| Kết nối database SQLServer/MySQL/PostgreSQL/MongoDB | Có |
| Giao diện Ribbon | Không |

Tính năng khác nếu có: Sử dụng hệ cơ sở dữ liệu MongoDB để lưu trữ thông tin về ổ cứng trong những lần theo dõi trước đó, từ đó có thể xem lại và xác định thời gian ổ cứng bắt đầu gặp vấn đề

[PHIẾU GIAO NHIỆM VỤ MÔN HỌC 2](#_Toc405232556)

[DANH MỤC HÌNH VẼ 4](#_Toc405232557)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 4](#_Toc405232558)

[PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG 5](#_Toc405232559)

[1.1. Các sản phẩm tương tự 5](#_Toc405232560)

[1.2. Các công nghệ kỹ thuật liên quan 6](#_Toc405232561)

[1.2.1. Microsoft Visual Studio 6](#_Toc405232562)

[1.2.2. GitHub 6](#_Toc405232563)

[1.2.3. WMI (Windows Management Instrumentation) 6](#_Toc405232564)

[1.2.4. S.M.A.R.T (Self-Monitoring, Analysis and Reporting Technology) 6](#_Toc405232565)

[1.2.5. MongoDB 7](#_Toc405232566)

[PHẦN 2:PHÂN TÍCH THIẾT KẾ 8](#_Toc405232567)

[2.1. Mô hình mạng 8](#_Toc405232568)

[2.2. Cơ sở dữ liệu 8](#_Toc405232569)

[2.3. Mô hình giao diện 9](#_Toc405232570)

[2.4. Các thành phần chức năng 9](#_Toc405232571)

[PHẦN 3:SẢN PHẨM KẾT QUẢ 10](#_Toc405232572)

[3.1. Lưu ý 10](#_Toc405232573)

**DANH MỤC HÌNH VẼ**

[Hình 1 Mô hình ứng dụng 8](#_Toc405232493)

[Hình 2 Cấu trúc cơ sở dữ liệu 8](#_Toc405232494)

[Hình 3 Giao diện chính 9](#_Toc405232495)

[Hình 4 Giao diện chương trình 10](#_Toc405232496)

[Hình 5 Giao diện mục logging 10](#_Toc405232497)

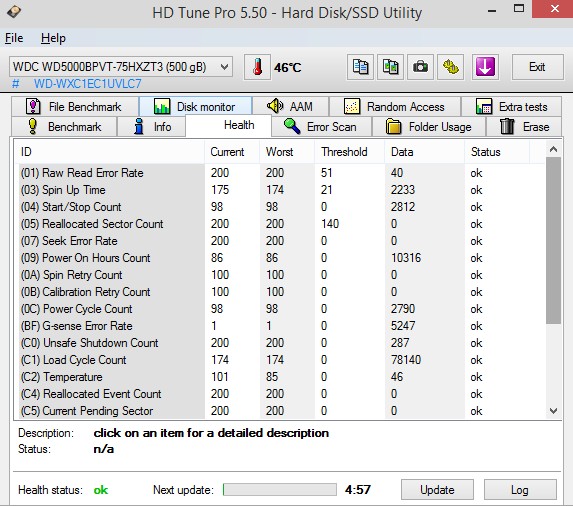
**DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Từ viết tắt** | **Ý nghĩa** |
| 1 | WMI | Windows Management Instrumentation |
| 2 | SMART | Self-Monitoring, Analysis and Reporting Technology |
| 3 | HDD | Hard Disk Drive |
| 4 | JSON | Javascript Object Notation |
| 5 | BSON | Binary JSON |

# PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG

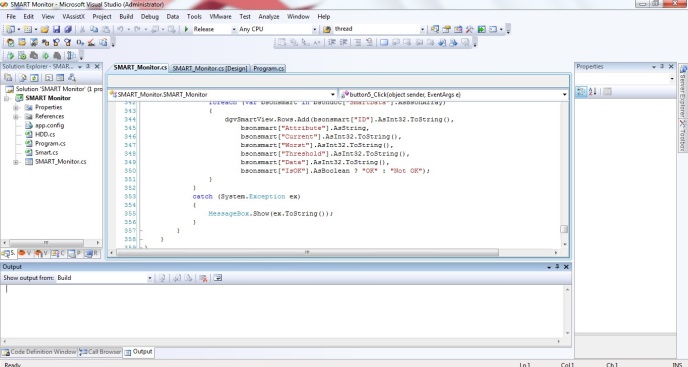
## Các sản phẩm tương tự

HD Tune

**

## Các công nghệ kỹ thuật liên quan

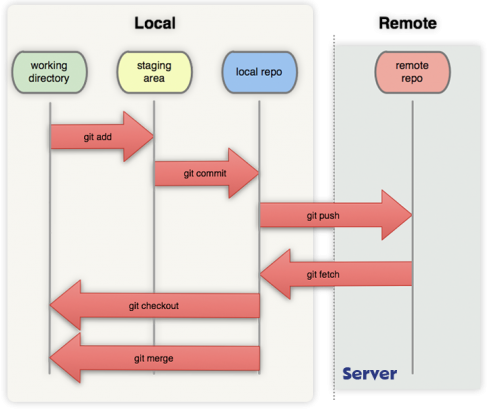
### Microsoft Visual Studio

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

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

### 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

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

### 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,

Trong đồ án này, WMI là công nghẹ được sử dụng để tương tác với tính năng S.M.A.R.T của ổ cứng.

### S.M.A.R.T (Self-Monitoring, Analysis and Reporting Technology)

SMART là công nghệ tự động giám sát tình trạng hoạt động của ổ đĩa cứng để thông qua BIOS, các phần mềm có thể thông báo cho người dùng biết trước các hư hỏng có thể xảy ra để xử lý (sao lưu dữ liệu, thay ổ cứng…).

### MongoDB

MongoDB là hệ CSDL không có quan hệ (NoSQL) sử dụng các cặp key-value thay cho các bảng trong CSDL quan hệ.

MongoDB lưu trữ dữ liệu dưới dạng BSON (dạng nhị phân của JSON).

Trong đồ án này, MongoDB được sử dụng để lưu trữ và trích xuất dữ liệu về SMART ở các thời điểm khác nhau với mục đích theo dõi tình trạng ổ cứng.

# PHẦN 2:PHÂN TÍCH THIẾT KẾ

## Mô hình mạng

Sử dụng WMI để giao tiếp với công nghệ S.M.A.R.T của ổ cứng, sử dụng hệ CSDL MongoDB để lưu trữ dữ liệu.

Program

SMART

Database

WMI

MongoDB

Hình 1 Mô hình ứng dụng

## Cơ sở dữ liệu

BsonCollection

SMART

BsonDocument

Serial : số seri HDD

Timeget : Thời điểm lấy dữ liệu

BsonArray SmartData

BsonDocument

ID

Atrribute  
Current  
Worst  
Threshold  
Data  
IsOK

…

Hình 2 Cấu trúc cơ sở dữ liệu

Thông tin về mỗi Atribbute là một phần tử BsonDocument trong BsonArray.

## Mô hình giao diện

SMART Monitor

X

-

[]

DataGridView hiển thị các thông số về ổ cứng dưới dạng bảng, mỗi hàng là một thuộc tính

Main | Option (TabControl)

Danh sách HDD (Combobox)

Thông tin cơ bản  
Serial  
Model

Trạng thái

v.v

Update

Exit

Hình 3 Giao diện chính

## Các thành phần chức năng

Dictionary<int,HDD> dDrives;

Đối tượng kiểu từ điển, đại diện cho các ổ cứng trong máy. Mỗi một ổ cứng được đánh số bằng ID, đại diện bằng một instance của class HDD.

void UpdateSMARTInfo();

Cập nhật thông tin về ổ cứng, lưu thông tin vào biến dDrives.

void ShowInfo();

Hiển thị thông tin lên giao diện chương trình.

void WriteData();

Hàm ghi dữ liệu từ dDrives vào CSDL, sử dụng Driver MongoDB của C#.

Hàm được gọi khi nhấn nút Save Data. (tab Option).

void ReadData();

Hàm đọc dữ liệu từ CSDL ghi vào dDrives, được gọi khi nhấn nút Load (tab Option).

ManagementObjectSearcher("SELECT \* from Win32\_DiskDrive");

Trả về đối tượng giúp liệt kê các ổ đĩa trong máy, là phương thức khởi tạo của class ManagementObjectSearcher, có một tham số duy nhất là câu truy vấn (dạng SQL) dùng để truy vấn dữ liệu trong WMI.

ObjectQuery("SELECT \* from MSStorageDriver\_FailurePredictStatus");

ObjectQuery("SELECT \* from MSStorageDriver\_FailurePredictData");

ObjectQuery("SELECT \* from MSStorageDriver\_FailurePredictThresholds");

Trả về đối tượng hỗ trợ cho việc lấy thông tin về ổ cứng.

Câu truy vấn đầu tiên dùng để lấy thông tin dự đoán xem ổ cứng có hoạt động tốt hay không.

Câu truy vấn thứ hai dùng để lấy dữ liệu về ổ cứng, giá trị của các thuộc tính mà SMART ghi nhận được.

Câu truy vẫn thứ ba lấy giá trị Threshold của các thuộc tính.

MongoClient(connectionString);

Trả về đối tượng đại diện cho một client kết nối đến MongoDB, tham số duy nhất là một chuỗi kết nối bao gồm giao thức, IP, port … Ví dụ mongodb://localhost để tự kết nối tới server local.

MongoClient().GetServer().GetDatabase(“name”);

Trả về đối tượng đại diện cho database trên server, tham số của phương thức này là tên của database.

BsonDocument.Add(“key”,value);

Thêm một cặp key/value vào một đối tượng BsonDocument.

Database.GetCollection(“name”);

Lấy đối tượng đại diện cho một Collection trong CSDL. Tham số là tên của Collection.

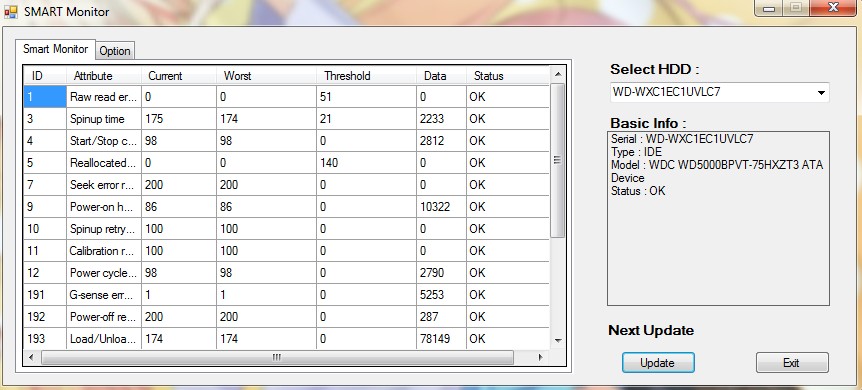
Collection.Insert(BsonDocument);

Thêm một đối tượng BsonDocument vào Collection.

Collection.FindAll().ElementAt(index);

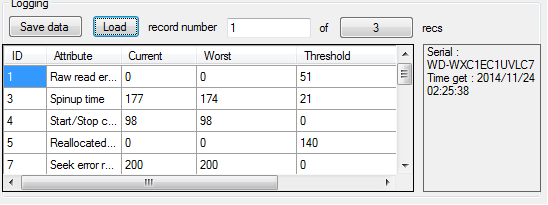
Lấy dữ liệu từ CSDL tại vị trí index đưa vào một đối tượng BsonDocument và trả về đối tượng đó.

# PHẦN 3:SẢN PHẨM KẾT QUẢ

**

Hình Giao diện chương trình

* Bấm nút update để cập nhật thông tin.
* Chọn HDD muốn xem thông tin ở phần Select HDD.



Hình Giao diện mục logging

Để xem một bản ghi, ta sẽ bấm vào nút có số 3 (mà ban đầu là nút get) để lấy số lượng bản ghi, sau đó nhập số thứ tự bản ghi muốn xem và ấn Load.

* Để lưu trữ lại dữ liệu vừa cập nhật, ta chỉ cần bấm nút Save Data.

## Lưu ý

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

Yêu cầu .Net FrameWork 3.5**.**