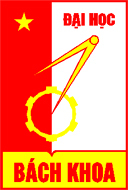
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**KHOA CNTT-TT**

-----🙞🙜🕮🙞🙜-----

****

**ĐỒ ÁN**

**Project 3**

**“Xây dựng chương trình giám sát thư mục”**

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

**Sinh viên thực hiện: Đặng Trọng Nghĩa**

**Lớp: CN-CNTT1 K55**

**PHIẾU GIAO NHIỆM VỤ MÔN HỌC**

1. **Thông tin sinh viên**

* Họ và tên sinh viên: Đặng Trọng Nghĩa
* Điện thoại liên lạc: 01644305022
* Lớp: CN-CNTT1 K55
* Email: [nghiadangtrongn92@gmail.com](mailto:nghiadangtrongn92@gmail.com)
* Hệ đào tạo: Đại học chính quy

1. **Sản phẩm mục tiêu của môn học**

* Xây dựng trương trình theo dõi các thư mục(file) trên máy tính
* Xác định người dùng nào đã thao tác trên thư mục(file).

1. **Các tính năng**

* Thống kê và theo dõi các ổ đĩa đang tồn tại trong máy(không tính ổ ảo)
* Ứng dụng chạy ở chế độ service
* Đọc thông tin từ file XML
* Theo dõi các thư mục (file) trên ở đĩa
* Kết nối cở sở dữ liệu SQL
* Đăng ký Event log khi bắt đầu chạy ứng dụng

**MỤC LỤC**

**Phiếu giao nhiệm vụ môn học…………………………………………………………1**

**MỤC LỤC ………………………………………………………………………………..2**

**PHẦN 1: NỀN TẢNG CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG ……………….....3**

* 1. Các sản phẩm tương tự ……………………………………………………………3
  2. Các công nghệ kỹ thuật liên quan ............................................................................5
     1. Microsoft Visual Studio …………………………………………………….5
     2. GitHub………………………………………………………………………6

**Phần 2: Phân tích thiết kế…………………………………………………………7**

2.1 Mô hình ứng dụng………………………………………………………………….7

2.2 Cơ sở dữ liệu……………………………………………………………………….7

2.3 Các hàm chức năng………………………………………………………………...8

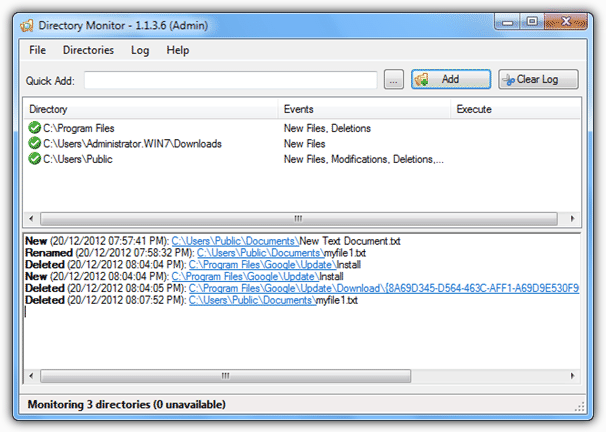
**Phần 3: Sản Phẩm và kết quả ……………………………………………………11**

**PHẦN 1: NỀN TẢNG VÀ CÔNG CỤ PHÁT TRIỂN ỨNG DỤNG**

* 1. **Các sản phẩm tương tự**

Một số những phần mềm tương tự

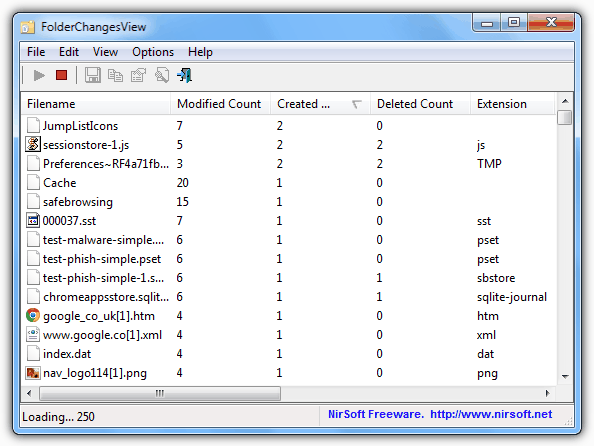
* Directory



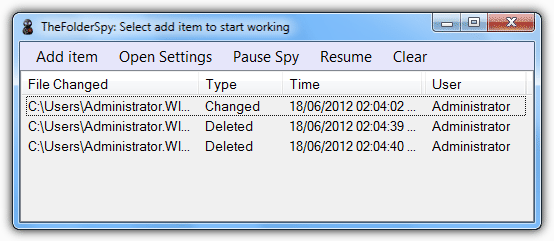
* File and Folder Watcher



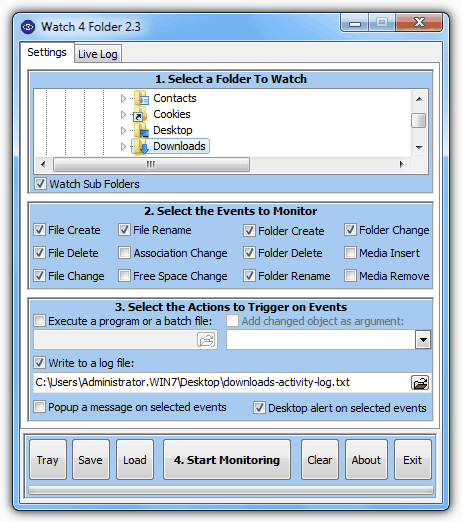
* Foder changes view



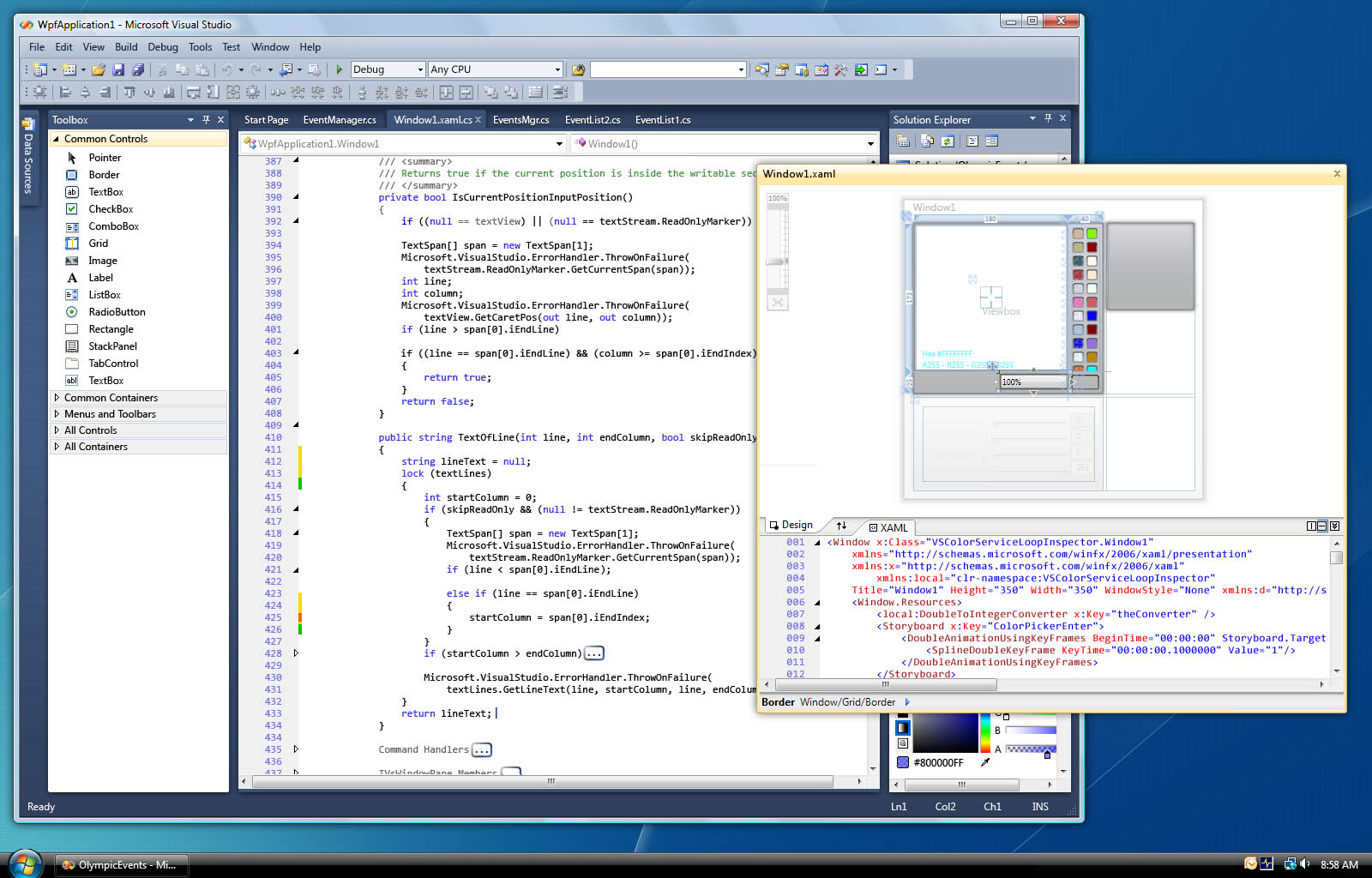
* The folder spy

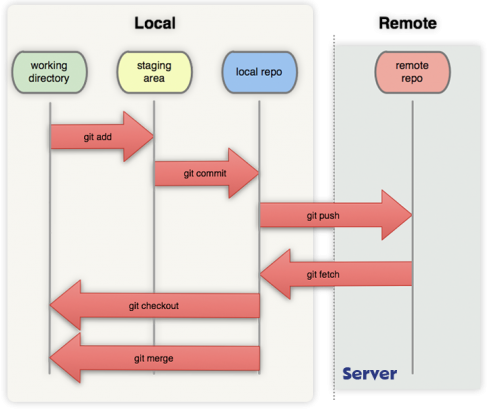


* Watch 4 folder



* 1. **Các công nghệ kỹ thuật liên quan**
     1. **Microsoft visual studio**
* Visual studio là một môi trường phát triển, quản lý mã nguồn từ Microsoft. Dùng để phát triển ứng dụng cho chương trình máy tính Microsoft Windows, cũng như các trang Web và các ứng dụng Web, dịch vụ Web. Visual studio sử dụng các nền tảng như Windows API, Windows Form, Windows PresentationFoundation, Windows store và Microsoft Silverlight



* + 1. **GitHub**
* ****GitHub là một dịch vụ lưu trữ trên web cho các dự án phát triển phần mềm trong đó có sử dụng các hệ thống kiểm soát bởi Git. GitHub cung cấp cả có phí hoặc miễn phí tùy vào nhu cầu sử dụng.
* Hiện nay GitHub đang có hơn 3,4 triệu lượt người dùng và trở thành máy chủ lớn nhất trên thế giới.
* GitHub sẽ được dùng để lưu trữ mã nguồn của đề tài này.

**PHẦN 2: PHÂN TÍCH THIẾT KẾ**

**2.1 Mô hình ứng dụng**

Lưu thông tin

Giám giát các ổ đĩa

Duyệt các ổ đĩa

Đọc file Config.xml

Đăng ký event log

**2.2 Cơ sở dữ liệu**

**InformationFile**

date

changer

path

name

2.3 Các hàm chức năng

* Class service1
* Class được kế thừa từ class serviceBase dùng để chạy ứng dụng ở chế độ service
* Phương thức OnStart() sẽ chỉ ra cần phải làm những gì khi bắt đầu ứng dụng. ThreadPool.QueueUserWorkItem() để đợi phương thức thi hành.
* Phương thức Onstop() chỉ ra khi ứng dụng dừng cần làm gì.
* Class InputControl(String)
  + Dùng để bắt đầu bắt đầu.
  + Tham số đầu vào: Đường dẫn file cấu hình.
  + Tham số đầu ra: Không có.
  + Gọi khi muốn bắt đầu chạy chương trình.
* Phương thức InitReadFileConfig(String Config\_file):
  + Dùng để kiểm tra và đọc file cấu hình cũng như là phương thức chính để đăng ký event, duyệt và theo dõi các ổ đĩa.
  + Tham số đầu vào: Đường dẫn chứa file cấu hình
  + Tham số đầu ra: Cho biết ứng dụng có bị lỗi không.
  + Phương thức gọi khi bắt ứng dụng thi hành.
* OutController.WriteEventLog(String):
  + Dùng để đăng ký ứng dụng vào Event viewer.
  + Tham số đầu vào: Tên ứng dụng muốn ghi vào event log.
  + Tham số đàu ra: Cho biêt thành công hay thất bại khi đăng ký.
  + Phương thức gọi khi muốn đăng ký event Log
* Remove.RemoveFolder():
  + Dùng để loại các thư mục không phải theo dõi.
  + Tham số đầu vào : Không.
  + Tham số đàu ra: Không.
  + Đươc gọi Khi muốn loại bỏ thư mục không muốn theo dõi.
* Class ScanDisk
  + Dùng để tìm và theo dõi các ổ đĩa trên máy tính.
  + Tham số: Không có.
* ScanDisk.OnDiskHander(object , DiskEventArgs)
  + Dùng để sử lý khi ổ đĩa có sự thay đổi.
  + Tham số đầu vào: Cho biết đối tương làm thay đổi và cho biết sự thay đổi là gì.
  + Tham số đầu ra: Không có.
  + Gọi khi có sự thay đổi ở ổ đĩa.
* Class WatchFile(String):
  + Dùng để đăng ký sư kiện theo dõi.
  + Tham số đầu vào: Cho biết thư mục hay file nào cần theo dõi.
  + Tham số đầu ra: Không có
  + Gọi khi muốn theo dõi vơi một đường dẫn biết trước.
* WatcherFile.WatchCreate(object , DiskEventArgs)
  + Dùng để khi sử lý sự kiện thư mục đăng ký theo dõi có sự kiện khởi tạo.
  + Tham số đầu vào: Cho biết đối tương làm thay đổi và cho biết sự thay đổi là gì.
  + Tham số đầu ra : Không có.
  + Gọi khi có sự kiện tạo ra trong thư mục đăng ký.
* WatcherFile.WatchDelete(object , DiskEventArgs)
  + Dùng để khi sử lý sự kiện thư mục(file) đăng ký theo dõi có sự kiện xóa.
  + Tham số đầu vào: Cho biết đối tương làm thay đổi và cho biết sự thay đổi là gì.
  + Tham số đầu ra : Không có.
  + Gọi khi có sự kiện đăng ký trong thư mục (file) đăng ký.
* WatcherFile.WatchChanger(object , DiskEventArgs)
  + Dùng để khi sử lý sự kiện thư mục(file) đăng ký theo dõi có sự kiện thay đổi.
  + Tham số đầu vào: Cho biết đối tương làm thay đổi và cho biết sự thay đổi là gì.
  + Tham số đầu ra : Không có.
  + Gọi khi có sự kiện thay đổi trong thư mục (file) đăng ký.
* WatcherFile.WatchRename(object , DiskEventArgs)
  + Dùng để khi sử lý sự kiện thư mục(file) đăng ký theo dõi có sự kiện đổi tên.
  + Tham số đầu vào: Cho biết đối tương làm thay đổi và cho biết sự thay đổi là gì.
  + Tham số đầu ra : Không có.
  + Gọi khi có sự kiện đổi tên trong thư mục (file) đăng ký.
* GetValue.\_GetWatcherNumber.Add(new WatchFile(String))
  + Dùng để thêm vào danh sách cần theo dõi.
  + Tham số đầu vào: Đối tượng của class WatchFile với đường dẫn muốn theo giám sát.
  + Tham số đầu ra: Không có.
  + Gọi khi muốn thêm vào danh sách theo dõi.
* ScanDisk.Run(bool):
  + Bắt đầu giám sát các ổ đĩa.
  + Tham số đầu vào: Thông báo bắt đầu.
  + Tham số đầu ra: Không có.
  + Gọi khi muốn bắt đầu theo dõi ổi đĩa.
* SaveOnDatabase.InitDatabase(String)
  + Bắt đầu lưu vào database.
  + Tham số đầu vào: Tên của database.
  + Tham số đàu ra: Cho biết có kết nối được database không.
  + Gọi khi muốn lưu vào database.
* OutController.WriteLog(String)
  + Bắt đâu khi muốn lưu vào file log.txt.
  + Tham số đầu vào: Ghi thông tin vào file log.txt.
  + Tham số đầu ra: Không có.
  + Gọi khi muốn ghi thông tin vào file log.txt.
* GetUserName.Name()
  + Cho biết name gây ra sự kiện(Đọc sự kiện đăng nhâp trong Event viewer).
  + Tham số đầu vào: Không có.
  + Tham số đầu ra: Người dùng đang thao tác trên máy.
  + Gọi khi muốn biết người nào đang thao tác trên máy tính thời điểm hiện tại.

**PHẦN 3: SẢN PHẨM VÀ KẾT QUẢ**

* Ứng dụng sẽ chạy luôn khi cài đặt kết thúc.
* Không phải chạy ứng dụng mỗi lần đăng nhập vào máy.

