TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**Project 3**

**TÌM HIỂU VỀ WEBCRAWLER VÀ**

**ỨNG DỤNG TRÊN TRANG WEB SIS HUST**

Sinh viên thực hiện: **Nguyễn Đình Minh**

Lớp: CN – CNTT2

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

HÀ NỘI 11 - 2014

# PHIẾU GIAO NHIỆM VỤ MÔN HỌC

**1. Thông tin về sinh viên**

Họ và tên sinh viên: Nguyễn Đình Minh

Điện thoại liên lạc: 0963040827 Email: ndminh9x@gmail.com

Lớp: CN - CNTT 2 Hệ đào tạo: Đại học chính quy

**2. Sản phẩm mục tiêu của môn học**

- Tìm hiểu về Web Crawler và ứng dụng trên trang web SIS HUST

**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/MinhDNguyen/CrawlerWeb

**Các tính năng tùy chọn:**

* Lấy thông tin trên trang SIS về học phần có mã nhập từ bàn phím:

<http://sis.hust.edu.vn/ModuleProgram/CourseLists.aspx>

* Thực hiện gửi gói tin POST/GET để lấy đăng nhập được vào trang SIS.
* Lấy thông tin trên trang SIS về sinh viên có mã số nhập từ bàn phím.

**MỤC LỤC**

[PHIẾU GIAO NHIỆM VỤ MÔN HỌC 2](#_Toc404874055)

[MỤC LỤC 3](#_Toc404874056)

[DANH MỤC HÌNH VẼ 4](#_Toc404874057)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 4](#_Toc404874058)

[PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG 5](#_Toc404874059)

[1.1. Các sản phẩm tương tự 5](#_Toc404874060)

[1.2. Các công nghệ kỹ thuật liên quan 5](#_Toc404874061)

[1.2.1. HttpFox (Add-in FireFox) 5](#_Toc404874062)

[1.2.2. Post Man (Add-in Chrome). 6](#_Toc404874063)

[1.2.3. XPath 6](#_Toc404874064)

[1.2.4. HTML Agility Pack 7](#_Toc404874065)

[1.2.5. GitHub 7](#_Toc404874066)

[PHẦN 2: PHÂN TÍCH THIẾT KẾ 8](#_Toc404874067)

[2.1. Mô hình mạng 8](#_Toc404874068)

[1. Mô hình Web Crawler 8](#_Toc404874069)

[2. Mô hình tương tác với Web Server 9](#_Toc404874070)

[3. Mô hình HTML. 10](#_Toc404874071)

[2.2. Mô hình giao diện 10](#_Toc404874072)

[2.3. Các hàm chức năng 11](#_Toc404874073)

[PHẦN 3: SẢN PHẨM KẾT QUẢ 13](#_Toc404874074)

[1. Giao diện cơ bản. 13](#_Toc404874075)

[2. Tra thông tin về môn học. 14](#_Toc404874076)

[3. Tra thông tin về cá nhân sinh viên. 15](#_Toc404874077)

[LỜI CÁM ƠN. 16](#_Toc404874078)

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

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Từ viết tắt** | **Ý nghĩa** |
| 1 | DOM | Document Object Model |
| 2 | HAP | HTML Agility Pack |
| 3 | HTTP | Hyper Text Transfer Protocol |

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

## Các sản phẩm tương tự

Một số nguồn tham khảo:

* Tạo và gửi các gói tin POST/GET sử dụng Web Request:

<http://msdn.microsoft.com/en-us/library/debx8sh9%28VS.80%29.aspx>

* Hướng dẫn sử dụng HTMLAgilityPack:

<http://www.codefall.io/15-jun-2014/htmlagilitypack-getting-started.html>

* Lấy toàn bộ Cookie từ Response:

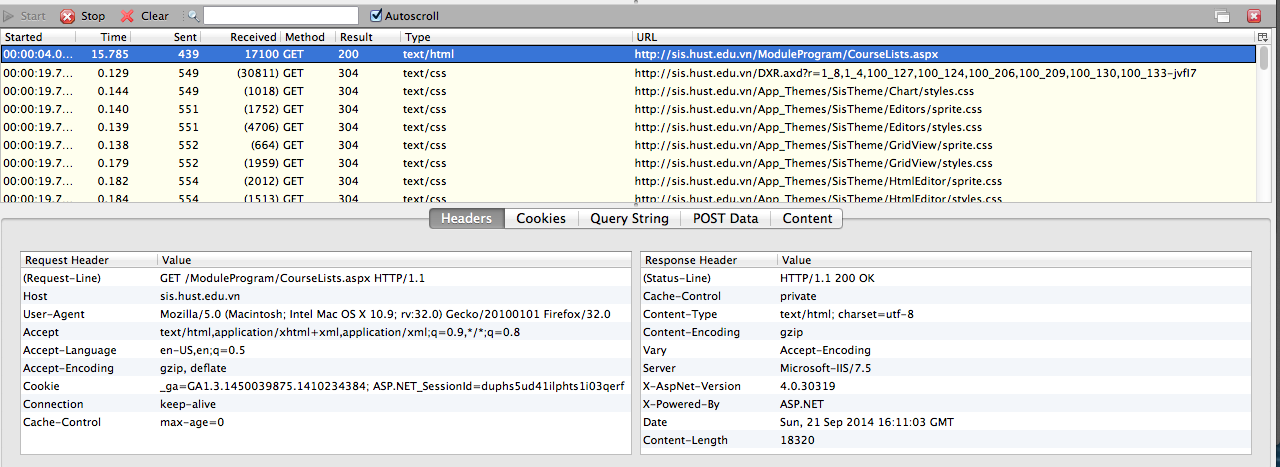
<http://stackoverflow.com/questions/10964177/httpwebrequest-and-cookiecontainer>

* Tạo HTTP Request để Login, giữ lại Cookie:

<http://stackoverflow.com/questions/15206644/how-to-pass-cookies-to-htmlagilitypack-or-webclient>

## Các công nghệ kỹ thuật liên quan

### HttpFox (Add-in FireFox)

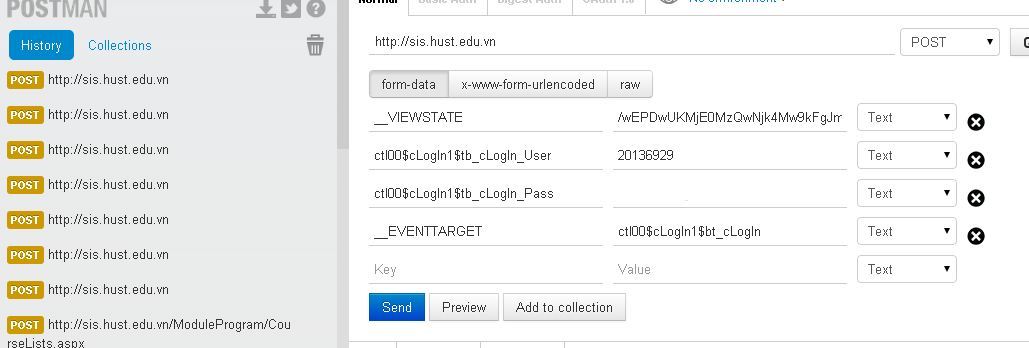


*Hình 1: Giao diện add-in HttpFox trên FireFox*

HttpFox là Add-in trên trình duyệt FireFox. Nó giúp xem các thông tin gửi đi và nhận về khi gõ địa chỉ trang Web lên trình duyệt.

Khi gõ vào địa chỉ trang web lên trình duyệt thì Client sẽ gửi rất nhiều các gói tin khác nhau với các *Request Header* khác nhau. Mỗi gói tin lại có nhiệm vụ riêng, như ảnh trên có thể thấy đầu tiên client sẽ gửi truy vấn GET *ModuleProgram/CourseLists.aspx* lên trang SIS*,* với loại thông tin là *text/html* sau đó Web Server sẽ gửi trả về *HttpResponse* gồm các *header* như trên. Ngoài ra client còn gửi rất nhiều các *Http Request* tới Web Server, không chỉ gửi các gói tin kiểu dạng *text/html* mà còn các dạng như các file *css*, hay là các hình ảnh, banner, biểu tượng trong trang web.

### Post Man (Add-in Chrome).

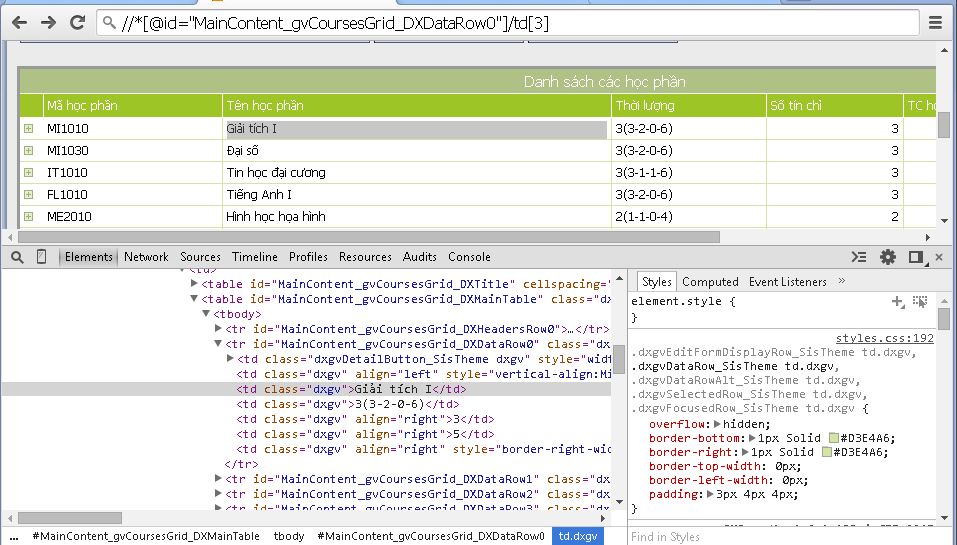


*Hình 2: Giao diện add-in PostMan trên Chrome*

Post man là một công cụ trên trình duyệt Chrome. Là một công cụ rất trực quan cho phép gửi và nhận các Request một cách thủ công lên Server và xem được nội dung trả về sẽ cho kết quả ra sao.

Cách thức thực hiện đầu tiên nhập URL địa chỉ mà muốn gửi một request lên. Tiếp đó chọn cách thức để gửi truy vấn, đó có thể là GET, POST, PUT, hay HEAD … Rồi ấn Submit để gửi gói tin, và chờ response của webserver. Ngoài ra khi thức hiện gửi bằng phương thức POST, có thể chọn các tham số cần thiết để gửi đi để được kết quả như mong muốn.

### XPath



*Hình 3: Truy cập tới các phần từ cần tìm qua XPath*

XPath, một thành phần hỗ trợ giúp truy xuất thông tin trong tập tin XML làm tiền đề cho việc áp dụng stylesheet kết hợp XML để tạo ra kết xuất tùy theo yêu cầu. Bên cạnh đó XPath cũng làm nền tảng cho việc hỗ trợ truy vấn parsing dữ liệu của tài liệu XML cực kỳ nhanh chóng hiệu quả.

|  |  |
| --- | --- |
| **Ký hiệu** | **Mô tả** |
| *nodename* | Chọn toàn bộ những node với tên “*nodename*” |
| / | Chọn từ node gốc |
| // | Chọn node trong văn bản bất kỳ với tên phù hợp không quan trọng chúng ở đâu |
| . | Chọn node hiện tại đang truy cập |
| .. | Chọn node cha của node hiện tại |
| @ | Lựa chọn thuộc tính |

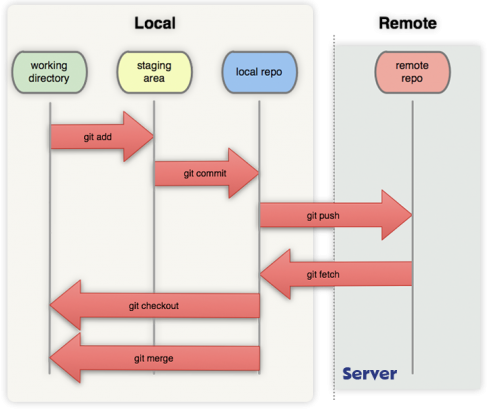
Tương tự như vậy, để tìm ra vị trí thông tin cần quan tâm trên cây phân cấp xml, sử dụng Chrome, với công cụ Copy Xpath để tìm đường dẫn DOM tới các thực thể cần lấy giá trị.

### HTML Agility Pack

HAP là một bộ thư viện mã nguồn mở miễn phí được thiết kế nhằm mục đích đơn giản hóa việc đọc và ghi tài liệu HTML, nó phục vụ đắc lực cho việc rút trích dữ liệu trên bất kỳ một tài liệu HTML nào. Có thể dùng nó để rút trích, sửa đổi dữ liệu trên một trang web mong muốn. Toàn bộ quá trình sẽ được tự động hóa hoàn toàn, chỉ việc chạy đọan mã đã chuẩn bị sẵn sẽ có kết quả mong muốn.

Thư viện này cho phép duyệt qua nội dung HTML như là một tài liệu XML bằng cú pháp của XPath. Từ đó thu thập trên cây DOM của tài liệu HTML, di chuyển từ node này sang node khác, hoặc chuyển đến node con một cách dễ dàng.

### 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

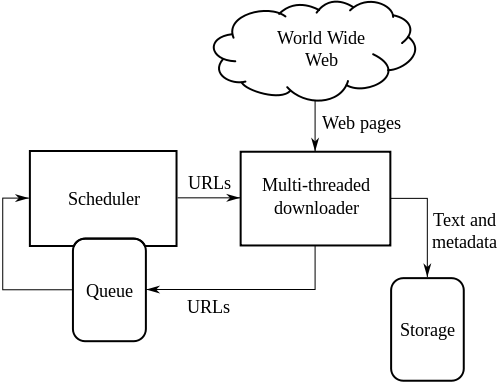
*Hình 4: Sơ đồ hoạt động GitHub*

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

# PHẦN 2: PHÂN TÍCH THIẾT KẾ

## Mô hình mạng

### Mô hình Web Crawler



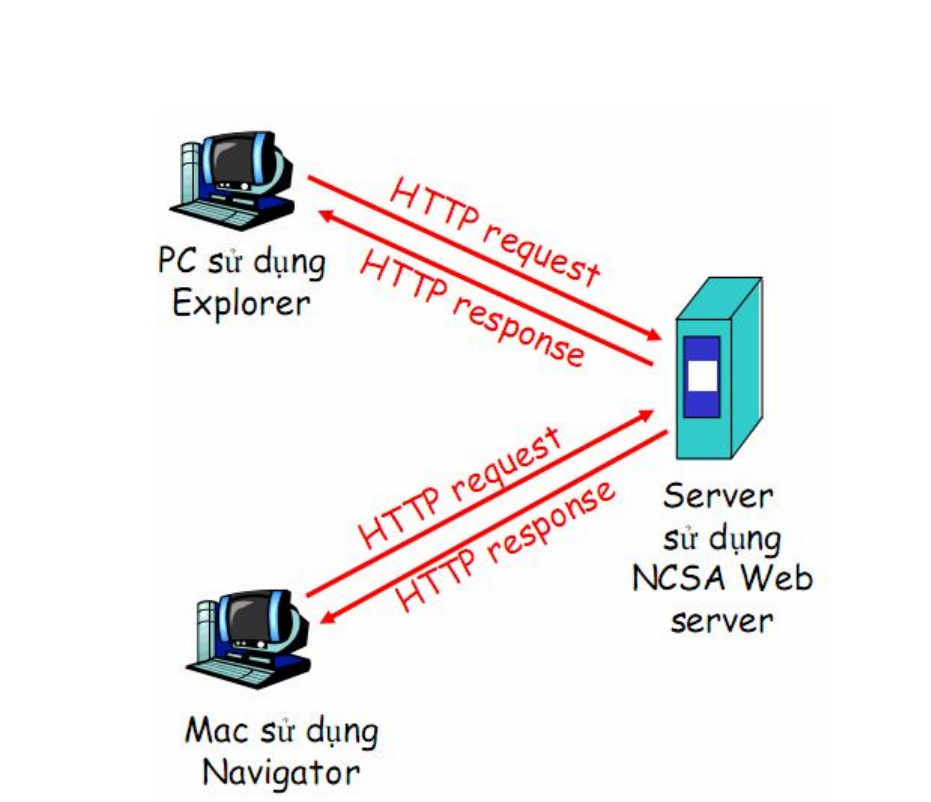
*Hình 5: Mô hình kiến trúc của một Web Crawler chuẩn*

Trình thu thập web (Web crawler) là một chương trình khai thác cấu trúc đồ thị của web di chuyển từ trang này qua trang khác. Các trình thu thập thường bắt đầu bằng cách chọn một số các đường dẫn (URL) ứng với các trang web sẽ ghé thăm đầu tiên, các trang này được gọi là các trang hạt giống. Khi ghé thăm một trang hạt giống, trình thu thập sẽ đọc nội dung trang web, lọc ra tất cả các siêu liên kết (hyperlink) có trong trang web đó và đưa các URL tương ứng với chúng vào một danh sách. Dựa vào danh sách này, trình thu thập tiếp tục quá trình duyệt đệ quy để ghé thăm tất cả các URL chưa được duyệt. Quá trình này được thực hiện đa luồng và chỉ dừng lại khi trình thu thập đã thu thập đủ số trang yêu cầu hoặc danh sách rỗng, tức là không còn URL để duyệt nữa.

Sau khi đã có được một danh sách các URL dùng cho việc thu thập, ta sẽ thực hiện quá trình lấy trang. Tất cả các trang được lấy một lần và được lưu vào một kho lưu trữ giống như cơ sở dữ liệu của công cụ tìm kiếm, đến đây không cần thu thập thêm.

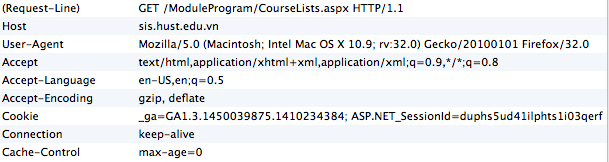
Các trang web chủ yếu được viết bằng các ngôn ngữ đánh dấu như HTML, XHTML chứa đựng nhiều thông tin có ích mà con người có thể muốn thu thập và lưu trữ lại, chính vì thế mà cần phải có những kỹ thuật bóc tách và trích xuất thông tin theo một cơ chế tự động. Trong đồ án này sử dụng bóc tách dữ liệu (parsing) qua bộ thư viện mã nguồn mở HAP.

### Mô hình tương tác với Web Server



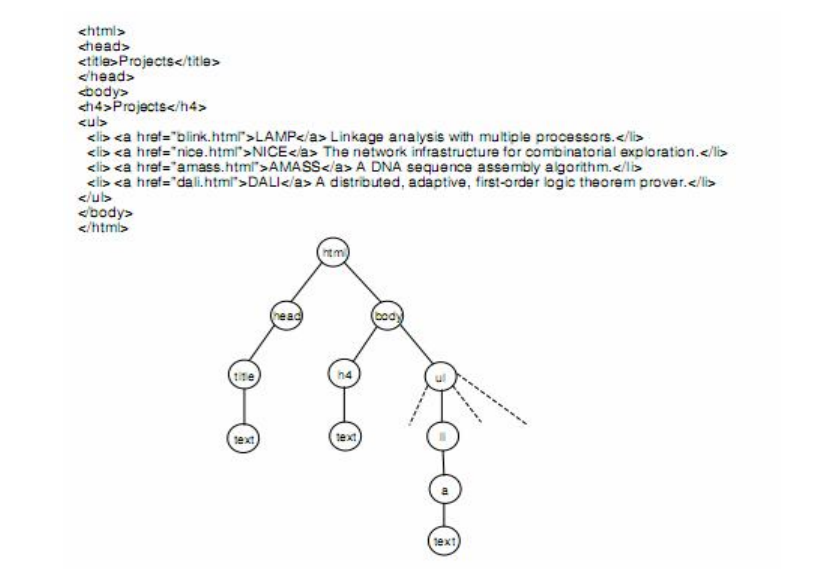
*Hình 6: Mô hình giao thức HTTP*

Giao thức truyền tải siêu văn bản viết tắt là HTTP, là giao thức quan trọng và cơ bản nhất của web sử dụng. HTTP xác định cách các thông điệp được định dạng và truyền tải ra sao, và những hành động nào mà webserver và các trình duyệt Web (browser) phải làm để đáp lại. Có hai loại thông điệp HTTP là : Thông điệp yêu cầu (HTTP Request) và thông điệp trả lời (HTTP Response). Cụ thể trong đồ án sử dụng hai phương thức truy vấn của HTTP là POST/GET qua các Header.



*Hình 7: Ví dụ về HTTP Request qua các Header*

### Mô hình HTML.



*Hình 8: Mô hình kiến trúc của một Web Crawler chuẩn*

Các trình thu thập có thể lấy ra giá trị của các URL hoặc một nội dung bất kỳ trong một trang web bằng cách kiểm tra phạm vi thẻ tag HTML chứa chúng. Để làm được điều này, trình thu thập có thể sử dụng mô hình thẻ HTML dạng cây và phân tích cấu trúc DOM của mô hình này. Phân tích cấu trúc DOM giúp trình thu thập có thể duyệt các node trên cây này và chỉ lấy ra phần nội dung mà nó cần. Trong đồ án này sẽ sử dụng XPath như đã nói ở trên để tìm đường dẫn DOM, qua đó sẽ sử dụng HAP để bóc tách dữ liệu.

## Mô hình giao diện

Chương trình gồm hai chức năng: một là lấy thông tin về môn học trên trang: <http://sis.hust.edu.vn/ModuleProgram/CourseLists.aspx> bằng cách người dùng nhập mã số môn học cần tra. Hai là lấy thông tin cá nhân người dùng bằng cách nhập thông tin tài khoản người dùng vào. Phần mềm sẽ tự động Crawl dữ liệu hiển thị kết quả mong muốn.

## Các hàm chức năng

private static String GenerateLoginPostData(String username, String password)

{

…

}

* Mục đích: Người dùng truyền vào thông tin tài khoản đăng nhập và sinh ra dữ liệu POST.
* Tham số:
  + Username: nội dung của tên đăng nhập
  + Password: nội dung mật khẩu của username
* Hàm trả về nội dung gói POST.

private static String GenerateSubjectsPostData(String subjectsName)

{

…

}

* Mục đích: Người dùng truyền vào mã số môn học và sinh ra dữ liệu POST
* Tham số:
  + subjectName: mã số môn học cần tra
* Tạo ra dữ liệu cho nội dung gói POST, phục vụ cho việc kiểm tra môn học.

private static Boolean VerifyWebContent(String page, String expected\_string)

{

…

}

* Mục đích: Kiểm tra có đúng nội dung cần tìm không.
* Tham số:
  + Page: nội dung trang web
  + Expected\_string: chuỗi muốn kiểm tra có trong trang web không
* Nếu khớp thì trả về true, sai thì trả về false.

private static String Send\_POST(String URL, String referer, String content)

{

…

}

* Mục đích: Gửi gói tin HTTP POST cho WebServer.
* Tham số:
  + URL: có dạng “/ModuleProgram/CourseLists.aspx”
  + Referer: có dạng “/ModuleProgram/CourseLists.aspx”
  + Content: nội dung gói POST đã được thực hiện ở trên, tùy nội dung mong muốn.
* Nội dung tin gói tin Response trả về dưới dạng String.

private static String Send\_GET(String URL, String referer)

{

…

}

* Mục đích: Gửi gói tin HTTP GET cho WebServer.
* Tham số:
  + URL: có dạng “/ModuleProgram/CourseLists.aspx”
  + Referer: có dạng “/ModuleProgram/CourseLists.aspx”
* Nội dung tin gói tin Response trả về dưới dạng String.

private static void CrawlerSubjects(String content)

{

…

}

private static void CrawlerInfo(String content)

{

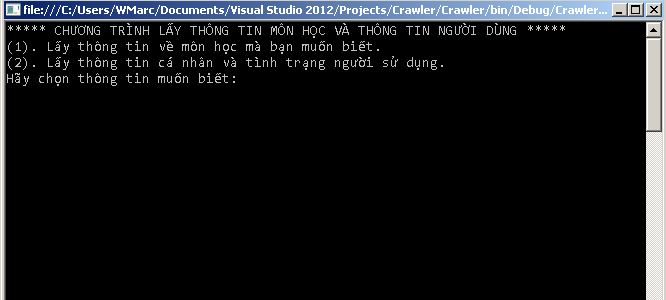
…

}

* Mục đích: Crawl dữ liệu từ trang SIS.
* Tham số:
  + Content: nội dung HTML để Crawl
* Thông tin cần thiết sau khi đã Crawl xong.

# PHẦN 3: SẢN PHẨM KẾT QUẢ

## Giao diện cơ bản.

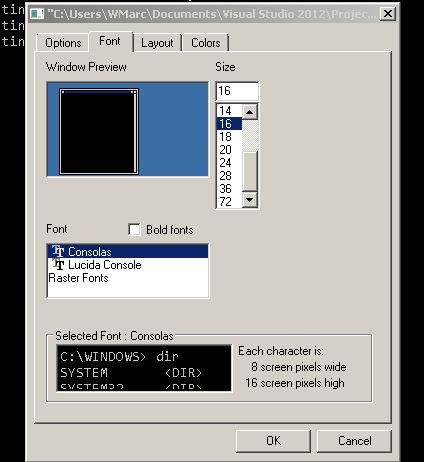


*Hình 9*

* **Để có thể hiển thị được tiếng Việt, sử dụng dòng lệnh sau:**

Console.OutputEncoding = Encoding.UTF8;

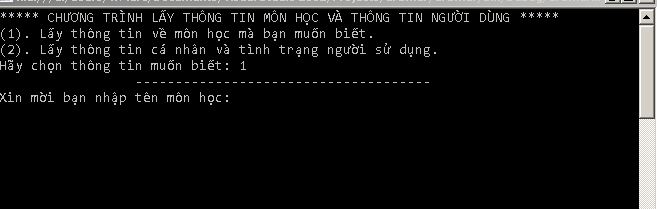
* **Sau đó chọn chuột phải vào màn hình Console và thiết lập như hình vẽ:**

****

*Hình 10*

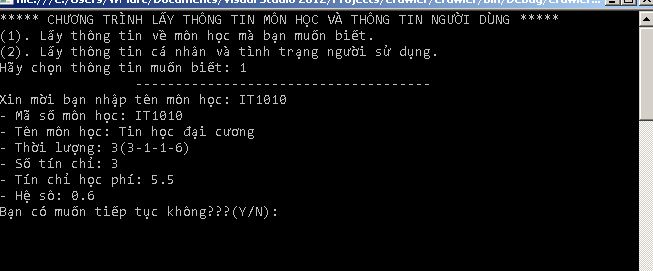
## Tra thông tin về môn học.

* Nhập từ bàn phím số 1 để kiểm tra thông tin về môn học.

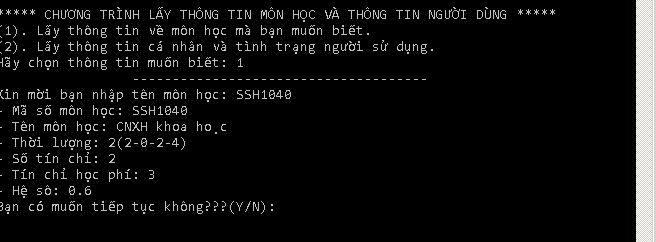


*Hình 11*

* Nhập thông tin về môn học cần quan tâm. Kết quả sẽ tự động hiện ra như bên dưới.

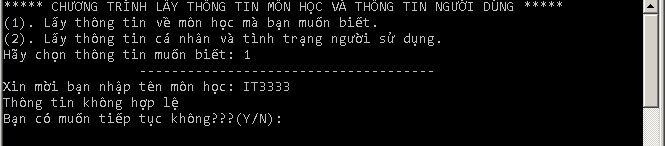


*Hình 12*



*Hình 13*

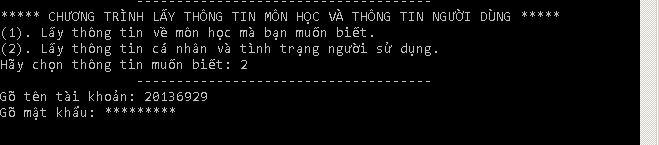
* Khi nhập sai tên môn học, chương trình sẽ thông báo không tìm thấy.



*Hình 14*

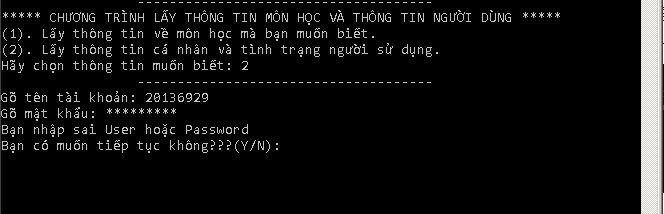
## Tra thông tin về cá nhân sinh viên.

* Nhập từ bàn phím số 2 để kiểm tra thông tin về sinh viên.
* Vì phải đăng nhập mới có thông tin về sinh viên nên hãy gõ tên tài khoản và password người dùng. Chương trình sẽ tự động kiểm tra thông tin.



*Hình 15*

* Thông báo khi người dùng nhập sai thông tin tài khoản.



*Hình 16*

# LỜI CÁM ƠN.

*Em xin gửi lời cảm ơn chân thành đến thầy giáo* ***Th.s******Nguyễn Đức Tiến****, giảng viên Bộ môn Kỹ Thuật Máy Tính - trường DH Bách Khoa Hà Nội người đã tận tình hướng dẫn, đóng góp ý kiến, chỉ bảo em trong suốt quá trình thực hiện đồ án.*

*Tuy vậy, do kỹ năng còn hạn chế nên trong bài báo cáo đồ án này sẽ không tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được đóng góp ý kiến của thầy để em có điều kiện bổ sung, nâng cao kiến thức cũng như khả năng của mình.*

*Lần nữa em xin chân thành cám ơn thầy!*