



5

Lab

Gửi và Nhận Mail trong C#

Sending & Receiving Email in C#

Thực hành Lập trình mạng căn bản

GVHD: Phan Trung Phát

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Nắm được cách giao tiếp với Email Server, gửi và nhận email.

2. Môi trường

- IDE Microsoft Visual Studio 2010 trở lên.

3. Liên quan

- Các kiến thức nền tảng về lập trình, ngôn ngữ lập trình C#, Windows Forms Application.
- Các kiến thức nền tảng về cơ chế hoạt động của thư điện tử, các giao thức phục vụ cho thư điện tử: SMTP, POP3, IMAP.
- Tham khảo tài liệu (Mục E) để có kiến thức cơ bản về C#, Winforms.

B. KIẾN THỨC VỀ EMAIL VÀ NGUYÊN TẮC HOẠT ĐỘNG

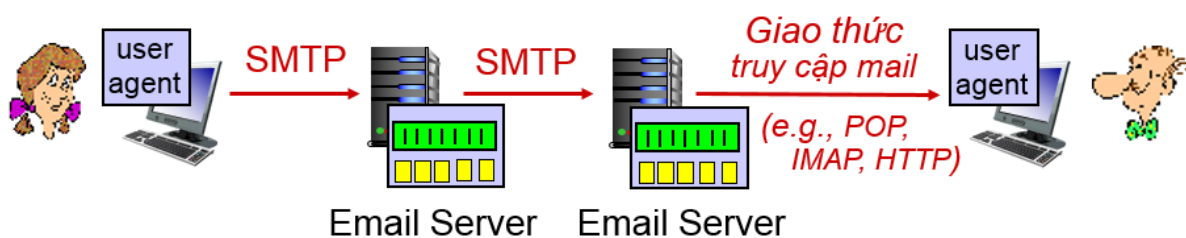
Email (electronic mail) là thư điện tử được sử dụng trong môi trường Internet để người dùng có thể trao đổi thông điệp với nhau.

Mỗi email có địa chỉ duy nhất, dạng: <username>@<domain name> trong đó:

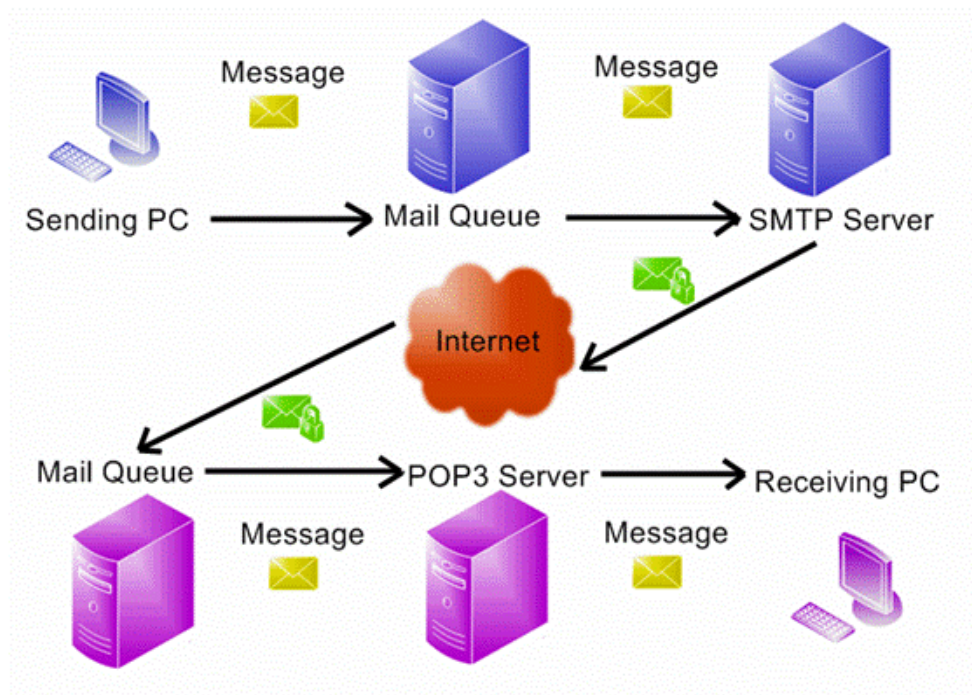
- <domain name> duy nhất trên hệ thống DNS toàn cầu.
- <username> chỉ cần duy nhất trong mail server của người nhận.

Ví dụ: 21520010@gm.uit.edu.vn, tranvana@gmail.com

Hình dưới đây mô tả nguyên tắc hoạt động, các giao thức sử dụng để gửi, nhận email.



Cụ thể, email không được gửi trực tiếp từ người gửi đến người nhận mà thông qua các Mail Server. Khi người gửi soạn và gửi đi một email, thư này sẽ được xếp tại hàng chờ (Mail Queue) của Mail Server (người gửi). Mail Server của người gửi sẽ chuyển email này đến Mail Server của người nhận. Lúc này người nhận sử dụng các giao thức truy xuất Email để lấy thư về.



➤ **Gửi email:**

Giao thức SMTP là giao thức dùng để gửi mail. SMTP hoạt động dựa trên giao thức TCP, sử dụng port 25 để truyền thông dữ liệu.

➤ **Nhận email:**

Có 2 giao thức dùng để truy xuất mail chính là POP (port 110) và IMAP (port 143). Sự khác nhau cơ bản giữa 2 giao thức này là nguyên lý lưu trữ email. POP là phương thức được đề xuất khi người dùng muốn truy cập email chỉ từ một thiết bị. Ngược lại, IMAP là phương thức được đề xuất khi người dùng cần kiểm tra email của họ từ một số thiết bị khác nhau, chẳng hạn như điện thoại, máy tính xách tay và máy tính bảng.

➤ **Webmail:**

Ngoài việc sử dụng các giao thức trên để gửi và nhận mail, có thể sử dụng giao thức HTTP (dựa trên Webmail) để gửi và nhận email.

➤ **Namespace System.Net.Mail và các lớp dẫn xuất của nó**

System.Net.Mail chứa các lớp được sử dụng để gửi email đến Email Server, và chờ Email Server gửi đi.

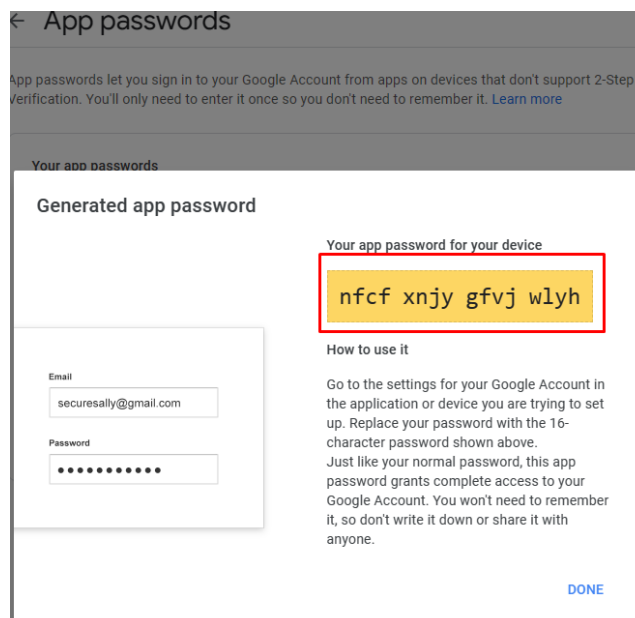
Sinh viên có thể tìm hiểu và tham khảo thêm chi tiết tại

<https://docs.microsoft.com/en-us/dotnet/api/system.net.mail?view=netframework-4.8>, cũng như tham khảo lại nội dung lý thuyết đã học.

C. THỰC HÀNH

Hướng dẫn sử dụng Gmail Email Server: để sử dụng được tài khoản Gmail chúng ta không thể sử dụng trực tiếp mật khẩu của tài khoản Google mà phải tạo App password để đăng nhập.

- B1: Truy cập: <https://myaccount.google.com/apppasswords>
- B2: Tạo một app password và lưu lại để sử dụng cho việc xác thực với gmail server.



	Host	Port	SSL
IMAP	imap.gmail.com	993	Yes
SMTP	smtp.gmail.com	465	Yes
POP	pop.gmail.com	995	Yes

Bài 1: Viết ứng dụng cho phép gửi mail.

Giao diện như hình minh họa:

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a "SEND" button on the left. To its right, there are two text boxes: "From:" containing "phatpt@uit.edu.vn" and "To:" containing "baonv@uit.edu.vn". Below these is a "Subject:" text box containing "Test email for Lab05". At the bottom, there is a "Body:" text area containing the text "Hello, Tese email. Best regard!".

Gợi ý:

- Sử dụng **SmtpClient** trong System.Net.Mail để gửi email.
- Sử dụng **NetworkCredential** để xác thực với Mail Server.

```
var client = new SmtpClient();
client.Connect(SMTPHost, SMTPPort, true); // smtp host, port, use ssl.
client.Authenticate(username, password); // gmail account, app password

var message = new MimeMessage();
message.From.Add(new MailboxAddress(name, from));
message.To.Add(new MailboxAddress("", to));
message.Subject = subject;

message.Body = new TextPart("plain") // gửi ở dạng plain text, hoặc có thể thay
bằng "html" nếu muốn gửi email dạng html
```

```
{
    Text = body
};

client.Send(message);
```

Bài 2: Viết ứng dụng cho phép đọc mail (IMAP).

Giao diện như hình minh họa:

Email	From	Thời gian
Welcome to the MDaemon email system for domain tea...	"MDaemon at mail.teacher.nt106" <...>	16/05/2019 12:00:0...
Microsoft Outlook Test Message	"Microsoft Outlook" <hiendo@teache...	01/01/0001 12:00:0...
Test mail telnet	lando@teacher.nt106	01/01/0001 12:00:0...
Test email from App	lando@teacher.nt106	16/05/2019 12:00:0...
System.Windows.Forms.TextBox, Text: Test mail from a...	lando@teacher.nt106	16/05/2019 12:00:0...
Test mail 3	"Windy" <lando@teacher.nt106>	16/05/2019 12:00:0...

Gợi ý:

```
var client = new ImapClient();
client.Connect(server, port, true); // imap host, port, use ssl.
client.Authenticate(username, password); // gmail account, app password.
var inbox = client.Inbox;
inbox.Open(FolderAccess.ReadOnly);
for (int i = 0; i < limit; i++)
{
    var message = inbox.GetMessage(i);
    // xử lý để hiển thị email lên listview: message.Subject; message.From;
    message.Date
}
```

Bài 3: Viết ứng dụng cho phép đọc mail (POP).

Giao diện tham khảo tương tự ở trên, sinh viên có thể linh động ghép Bài 2 và Bài 3 thành 1 và có button chọn sử dụng phương thức nào để đọc mail.

Bài 4: Viết ứng dụng Email Client.

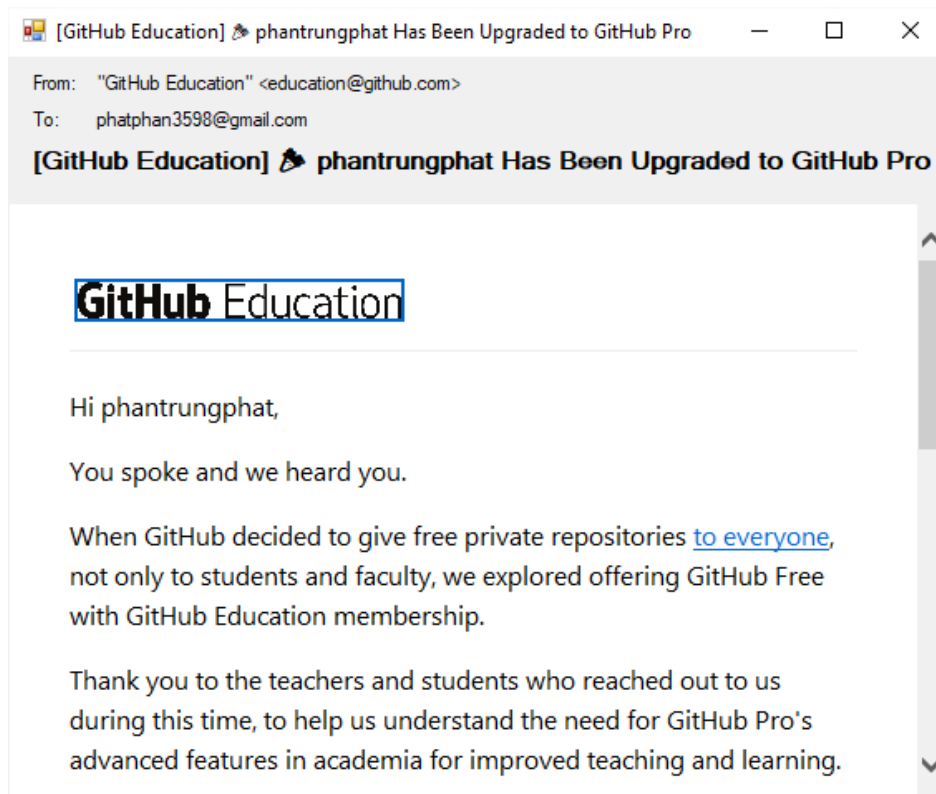
Từ các ví dụ trên, viết ứng dụng có giao diện cho phép đăng nhập, đăng xuất, duyệt mail, đọc mail cụ thể, reply, gửi mail, đính kèm tệp tin khi gửi, với giao diện tùy biến (Có thể tham khảo Outlook, Gmail).

Giao diện như hình minh họa:

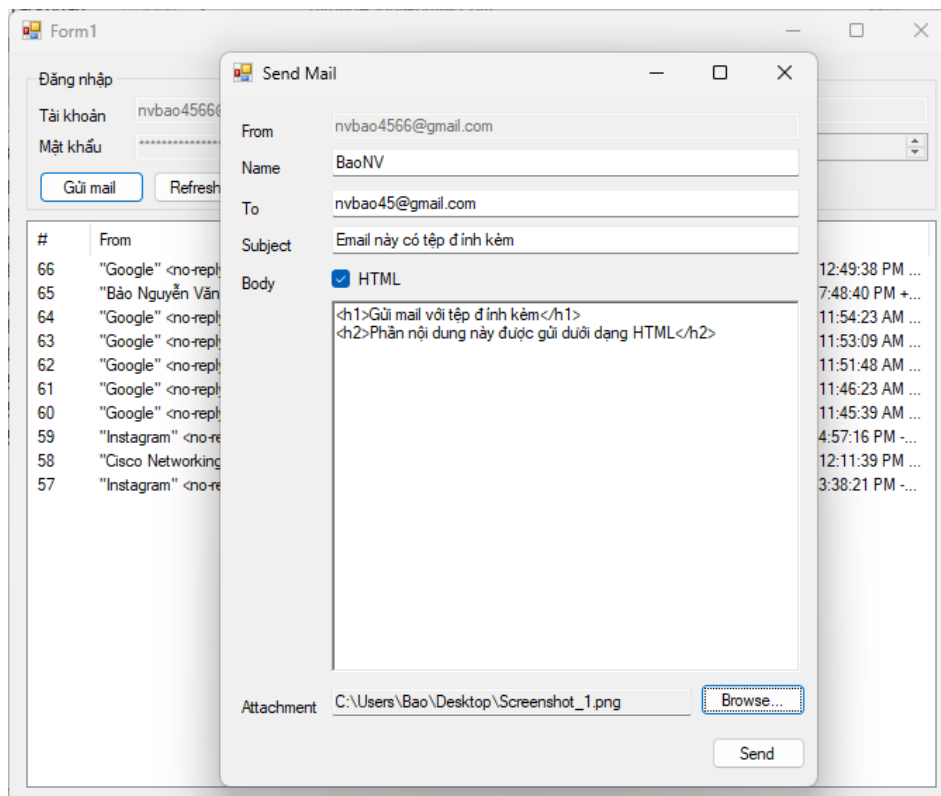
Hình 1. Đăng nhập vào gmail

#	From	Subject	Datetime
0	"Nhóm phụ trách cộng đồng Google" <...>	Phat, chào mừng bạn đến với Tài khoản Googl...	1/16/2019 4:55:34 AM -...
1	"GitHub" <noreply@github.com>	[GitHub] Please verify your email address.	1/16/2019 4:56:38 AM -...
2	"GitHub" <support@github.com>	[GitHub] Thanks for verifying your email address.	1/16/2019 12:56:46 PM ...
3	"GitHub" <noreply@github.com>	[GitHub] An email address was added to your a...	1/16/2019 1:13:01 PM +...
4	"GitHub" <noreply@github.com>	[GitHub] A third-party OAuth application has be...	1/16/2019 1:26:14 PM +...
5	"Trello" <do-not-reply@trello.com>	Xác thực tài khoản Trello	1/16/2019 2:52:11 PM +...
6	"Taco từ Trello" <taco@trello.com>	Chào mừng đến với Trello!	1/16/2019 2:54:05 PM +...
7	"Taco From Trello" <taco@trello.com>	Are you a work from home hermit?	1/23/2019 4:20:36 PM +...
8	"GitHub Education" <education@github...>	[GitHub Education] 🐛 phantrungphat Has Bee...	2/9/2019 12:45:24 AM +...
9	"Taco from Trello" <taco@trello.com>	Travel with Trello	2/12/2019 4:17:56 PM +...

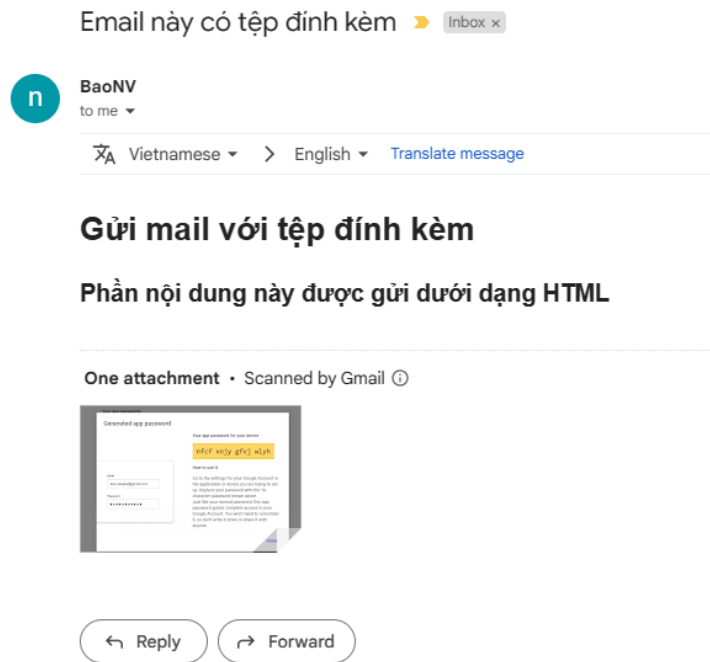
Hình 2. Hiển thị list email



Hình 3. Đọc email cụ thể



Hình 4. Gửi kèm email với tệp đính kèm



Hình 5. Kiểm tra lại kết quả

D. YÊU CẦU & NỘI BÀI

1. Yêu cầu

- Code “sạch” [2], đặt tên biến rõ ràng.
- Nội bài không đầy đủ; lỗi, không chạy được; nộp trễ; không giải thích báo cáo; sao chép code bạn khác, nguồn có sẵn: *xử lý tùy theo mức độ*.

2. Nội bài

- Sinh viên thực hành và nộp bài theo **Nhóm (Nhóm trưởng nộp)** tại website môn học theo thời gian quy định.
- Bài nộp bao gồm toàn bộ **Source-code** của các bài tập liên quan tại github và **trình bày báo cáo gồm Ảnh chụp màn hình kèm mô tả, giải thích** các bước hoạt động, quay Video thực hiện từng kết quả cụ thể:

Toàn bộ project đặt vào 1 file nén (.zip) với tên theo quy tắc sau:

Mã lớp-LabX-MSSV1-MSSV2

Ví dụ: *NT106.M21.MMCL.1-Lab05-20520001-20520002*

E. THAM KHẢO

- [1] Microsoft (2018). C# Guide. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/>
- [2] Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.
- [3] Network Programming in the .NET Framework: <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/>

HẾT