

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU THÀNH PHỐ HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO TIỂU LUẬN**

**MÔN: LẬP TRÌNH MẠNG**

**ĐỀ TÀI:**

**XÂY DỰNG CHƯƠNG TRÌNH MAIL CLIENT**

Sinh viên thực hiện : NGÔ VĂN CHIẾN - 5951071004

: NGUYỄN THANH QUI - 5951071086

: TRẦN QUANG HUY - 5951071037

Lớp : CQ.59.CNTT

*TP Hồ Chí Minh, năm 2021*

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU THÀNH PHỐ HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO TIỂU LUẬN**

**MÔN: LẬP TRÌNH MẠNG**

**ĐỀ TÀI:**

**XÂY DỰNG CHƯƠNG TRÌNH MAIL CLIENT**

Sinh viên thực hiện : NGÔ VĂN CHIẾN - 5951071004

: NGUYỄN THANH QUI - 5951071086

: TRẦN QUANG HUY - 5951071037

Lớp : CQ.59.CNTT

*TP Hồ Chí Minh, năm 2021*

## NHIỆM VỤ BÁO CÁO TIỂU LUẬN

-----\*\*\*-----

**Họ tên SV:** Ngô Văn Chiến – Nguyễn Thanh Qui – Trần Quang Huy

**Khóa:** 59

**Lớp:** CQ.59. CNTT

### 1. Tên đề tài:

XÂY DỰNG CHƯƠNG TRÌNH MAIL CLIENT

### 2. Mục đích, yêu cầu:

#### a. Mục đích:

**Tìm hiểu các giao thức SMTP, POP3, IMAP. Mã hóa Base64, ...  
Quá trình làm việc của MailClient.**

#### b. Yêu cầu:

##### - Yêu cầu công nghệ:

- Sử dụng ngôn ngữ lập trình Java.
- Sử dụng công cụ Apache NetBeans.

##### - Yêu cầu chức năng:

- Cho phép đăng nhập vào Gmail.
- Cho phép tự động download các file đính kèm
- Cho phép xóa tất cả mail.

### **3. Nội dung và phạm vi đề tài**

#### **a. Nội dung:**

- Tổng quan bài toán.
- Tổng quan về các công nghệ đang sử dụng.
- Khảo sát bài toán.
- Lập trình xây dựng.
- Kiểm thử và chạy thực nghiệm.
- Kết quả thu được

#### **b. Phạm vi:**

- Nghiên cứu công cụ Apache NetBeans và ngôn ngữ Java.

### **4. Công nghệ, công cụ và ngôn ngữ lập trình**

- Công cụ: Apache NetBeans.
- Ngôn ngữ lập trình: Java.

### **5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

- Xây dựng ứng chương trình Mail Client.

Ngày ..... tháng ..... năm 2021

**Trưởng BM Công nghệ Thông tin**

## LỜI CẢM ƠN

Lời đầu tiên em xin gửi lời cảm ơn chân thành đến quý thầy, cô giáo trong **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông vận tải**.

Những người đã truyền dạy, đã trang bị cho em kho tàng kiến thức về bầu trời công nghệ thông tin rộng lớn.

Ở đây, em không chỉ học được kiến thức về sách vở mà em còn học được các bài học, kỹ năng sống trước khi tạm biệt mái trường đại học thân yêu này và tiến ra biển đời mênh mông rộng lớn. Đặc biệt, em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy **PHAN THANH HY**, người đã dạy em môn **LẬP TRÌNH MẠNG**

Trong quá trình học tập và tìm hiểu em đã nỗ lực rất nhiều với mong muốn hoàn thành dự án một cách tốt nhất, nhưng đời người sẽ có những thiếu sót không thể tránh khỏi, và với những người chưa chững chạc và trưởng thành như em thì sai lầm là không thể không mắc phải. Em mong thầy, cô bộ môn có thể thông cảm và cho em những ý kiến, đóng góp để em có thể hoàn thành đồ án của mình một cách trọn vẹn nhất trước khi rời xa ngôi trường thân yêu này.

Sau cùng, em xin kính chúc Quý Thầy Cô trong **Bộ môn Công nghệ thông tin** lời chúc sức khỏe, luôn hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Tp. Hồ Chí Minh, ngày ..... tháng ..... năm ...*

**Giảng viên hướng dẫn**

# MỤC LỤC

NHIỆM VỤ BÁO CÁO TIỂU LUẬN .....	3
LỜI CẢM ƠN.....	5
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN .....	6
DANH MỤC HÌNH ẢNH, BẢNG BIỂU .....	8
MỞ ĐẦU .....	9
1. Lý do chọn đề tài. ....	9
2. Mục tiêu nghiên cứu:.....	9
3. Phương pháp nghiên cứu.....	9
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	10
1.1. Lý thuyết cơ sở.....	10
1.2. Nội dung chính và các giao thức mail client.....	10
a.quá trình làm việc của mailclient: .....	10
b.giao thức tcp/ip: .....	11
c.giao thức pop3:.....	14
d.giao thức imap: .....	15
e.giao thức smtp: .....	16
f.giao thức và số cổng tương ứng: .....	17
g.Mã hóa Base64: .....	17
CHƯƠNG 2: THỬ NGHIỆM ỨNG DỤNG .....	19
2.1. Môi trường cài đặt.....	19
2.2. Triển khai ứng dụng .....	19
2.3. DEMO .....	20
2.3.1 Ứng dụng download attachment bằng pop3 .....	20
2.3.2 Ứng dụng xóa mail sử dụng Imap .....	24
CHƯƠNG 3: KẾT LUẬN.....	27
3.1 Kết quả đạt được .....	27
3.2 Hạn chế .....	27
TÀI LIỆU THAM KHẢO .....	29

## **DANH MỤC HÌNH ẢNH, BẢNG BIỂU**

*Hình 1.1. Các tầng trong giao thức tcp/ip*

*Bảng 1.2. Bảng giao thức mạng và số cổng tương ứng*

*Bảng 1.3. Bảng mã chuyển đổi file*

*Bảng 1.4. Bảng tra mã*

*Hình 2.1. Code download file đính kèm*

*Hình 2.2. Code download file đính kèm*

*Hình 2.3. Code download file đính kèm*

*Hình 2.4. Code download file đính kèm*

*Hình 2.5. Kết quả download file đính kèm*

*Hình 2.6. Code xóa email*

*Hình 2.7. Code xóa email*

*Hình 2.8. Code xóa email*

*Hình 2.9. Kết quả xóa email*



# MỞ ĐẦU

## 1. Lý do chọn đề tài.

- Email hay Gmail là một bước tiến lớn trong quá trình phát triển của nhân loại khi mà giờ đây thư, tin nhắn hay những cuộc trò chuyện không còn dừng trên giấy hay phải gặp nhau mới truyền đạt được mà nó sẽ được chuyển dưới dạng số một cách tức thì
- Mail cũng là một công cụ mà ai cũng cần, là nền tảng đầu tiên để phát triển facebook, Instagram, ...
- Để hiểu hơn về cách thức hoạt động cũng như bước tiến quan trọng này của nhân loại thì đề tài đăng nhập mail download các attachment là đề tài nhóm chúng em chọn để hiểu hơn và ứng dụng một cách trực quan và thực tiễn những lý thuyết khô khan ở giảng đường đại học

### Hướng tiếp cận của đề tài:

- Tìm hiểu giao thức mạng TCP/IP, POP3, IMAP, SMTP.
- Các thư viện hỗ trợ lập trình mạng: javax.mail
- Xây dựng chương trình minh họa:
  - 2 chương trình demo :
    - download tất cả các attachment.
    - Xoá tất cả các thư trong phần inbox.

## 2. Mục tiêu nghiên cứu:

- Tìm hiểu các giao thức SMTP, POP3, IMAP, Mã hóa Base 64, ...
- Quá trình làm việc của MailClient

## 3. Phương pháp nghiên cứu

- Tài liệu: Tìm hiểu và khai thác các tài liệu liên quan đến các giao thức mạng TCP/IP, POP3, IMAP, SMTP, Mã hóa Base 64.
- Thực hành: Tiến hành xây dựng demo cho ứng dụng Đăng nhập vào Gmail và thực hiện các chức năng sử dụng các giao thức trên.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1. Lý thuyết cơ sở.

SocKet là một giao diện lập trình (API – Application Program Interface) ứng dụng mạng thông qua giao diện này có thể lập trình điều khiển việc truyền thông giữa 2 máy sử dụng các giao thức mức thấp như TCP,UDP... , Có thể tưởng tượng nó như một thiết bị truyền thông 2 chiều tương tự như tệp tin, chúng ta gửi/nhận dữ liệu giữa 2 máy, tương tự như việc đọc ghi trên tệp tin.

Trong không gian mạng mail là một bước tiến quan trọng của nhân loại cũng như đổi mới phương thức giao tiếp giữa người và người

## 1.2. Nội dung chính và các giao thức mail client.

Trong phần này nhóm chúng em xin phép đi trực tiếp vào nội dung chính của đề tài nghiên cứu báo cáo của tiểu luận các nội dung chính:

### **a.quá trình làm việc của mailclient:**

Email client (từ kỹ thuật là Mail User Agent (MUA), trong tiếng Việt là trình duyệt mail) là một phần mềm máy tính được dùng để truy cập và quản lý email của người dùng.

Các giao thức phổ biến để lấy email bao gồm POP3 và IMAP4, gửi mail thường được thực hiện bằng cách sử dụng giao thức SMTP.

Một tiêu chuẩn quan trọng được hỗ trợ bởi hầu hết các email clients là MIME (Multipurpose Internet Mail Extensions), được sử dụng để gửi email đính kèm tệp tin nhị phân. File đính kèm là các tệp tin không phải là một phần của email, nhưng được gửi đi với email.

Các mail client phổ biến bao gồm Microsoft Outlook, IBM Lotus Notes, Mozilla Thunderbird,, Evolution và Apple Inc.'s Mail.

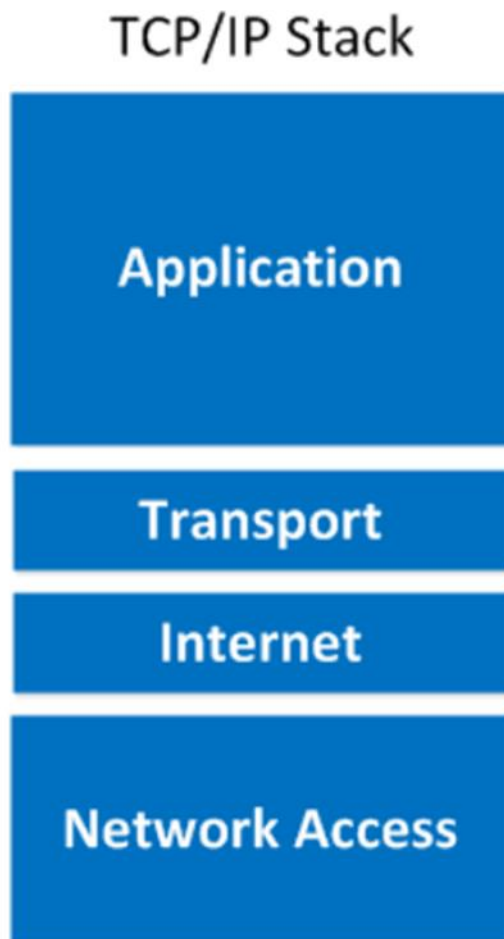
**b.giao thức tcp/ip:**

- Bộ giao thức TCP/IP, ngắn gọn là TCP/IP (tiếng Anh: Internet protocol suite hoặc IP suite hoặc TCP/IP protocol suite - bộ giao thức liên mạng), là một bộ các giao thức truyền thông cài đặt chồng giao thức mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (Giao thức Điều khiển Giao vận) và IP (Giao thức Liên mạng). Chúng cũng là hai giao thức đầu tiên được định nghĩa.

- Như nhiều bộ giao thức khác, bộ giao thức TCP/IP có thể được coi là một tập hợp các tầng, mỗi tầng giải quyết một tập các vấn đề có liên quan đến việc truyền dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn. Về mặt logic, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các giao thức tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể được truyền đi một cách vật lý.

- Mô hình OSI miêu tả một tập cố định gồm 7 tầng mà một số nhà sản xuất lựa chọn và nó có thể được so sánh tương đối với bộ giao thức TCP/IP. Sự so sánh này có thể gây

nhằm lần hoặc mang lại sự hiểu biết sâu hơn về bộ giao thức TCP/IP.



*Hình 1.1. Các tầng trong giao thức tcp/ip*

- **Tầng ứng dụng:**

Các nhà thiết kế TCP/IP cảm thấy rằng các giao thức mức cao nên bao gồm các tầng trình bày và tầng phiên. Để đơn giản, họ tạo ra một tầng ứng dụng kiểm soát các giao thức mức cao, các vấn đề của tầng Trình diễn, mã hóa và điều khiển hội thoại. TCP/IP tập hợp tất cả các vấn đề liên quan đến ứng dụng vào trong một tầng, và đảm bảo dữ liệu được đóng gói một cách thích hợp cho tầng kế tiếp.

- **Tầng giao vận:**

Tầng vận chuyển đề cập đến các vấn đề chất lượng dịch vụ như độ tin cậy, điều khiển luồng và sửa lỗi. Một trong các giao thức của nó là TCP, TCP cung cấp các phương thức linh hoạt và hiệu quả để thực hiện các hoạt động truyền dữ liệu tin cậy, hiệu suất cao và ít lỗi. TCP là giao thức có tạo cầu nối (connection – oriented). Nó tiến hành hội thoại giữa nguồn và đích trong khi bọc thông tin tầng ứng dụng thành các đơn vị gọi là segment. Tạo cầu nối không có nghĩa là tồn tại một mạch thực giữa hai máy tính, thay vì vậy nó có nghĩa là các segment của tầng 4 di chuyển tới và lui giữa hai host để công nhận kết nối tồn tại một cách luận lý trong một khoảng thời gian nào đó. Điều này coi như chuyển mạch gói (packet switching).

- **Tầng Internet:**

Mục tiêu của tầng Internet là truyền các gói tin bắt nguồn từ bất kỳ mạng nào trên liên mạng và đến được đích trong điều kiện độc lập với đường dẫn và các mạng mà chúng đã trải qua. Giao thức đặc trưng không chế tầng này được gọi là IP. Công việc xác định đường dẫn tốt nhất và hoạt động chuyển mạch gói diễn ra tại tầng này.

- **Tầng truy xuất mạng**

Tên của tầng này có nghĩa khá rộng. Nó cũng được gọi là tầng host – to – network. Nó là tầng liên quan đến tất cả các vấn đề mà một gói tin IP yêu cầu để tạo một liên kết vật lý thực sự, và sau đó tạo một liên kết vật lý khác. Nó bao gồm các chi tiết kỹ thuật LAN và WAN, và tất cả các chi tiết trong tầng liên kết dữ liệu cũng như tầng vật lý của mô hình OSI.

Mô hình OSI hướng đến tối đa độ linh hoạt tại tầng ứng dụng cho người phát triển phần mềm. Tầng vận chuyển liên quan đến hai giao thức TCP và UDP (User Datagram Protocol). Tầng cuối cùng, tầng truy xuất mạng liên kết các kỹ thuật LAN hay WAN đang được sử dụng.

Trong mô hình TCP/IP không cần quan tâm đến ứng dụng nào yêu cầu các dịch vụ mạng, và không cần quan tâm đến giao thức vận chuyển nào đang được dùng,

chỉ có một giao thức mạng IP. Đây là một quyết định thiết kế được cân nhắc rất kỹ. IP phục vụ như một giao thức đa năng cho phép bất kỳ máy tính nào, ở bất cứ đâu truyền dữ liệu vào bất cứ thời điểm nào.

### **c.giao thức pop3:**

Giao thức POP3 là viết tắt của Post Office Protocol version 3, là một giao thức tầng ứng dụng, dùng để lấy thư điện tử từ server mail, thông qua kết nối TCP/IP. POP3 được sử dụng để kết nối tới server email và tải email xuống máy tính cá nhân thông qua ứng dụng email client như Outlook, Thunderbird, Windows Mail, Mac Mail...

POP là một giao thức nhận mail có lịch sử lâu đời. Nó ra đời từ máy tính còn bị giới hạn bởi tốc độ, băng thông, vậy nên các kỹ sư đã tạo ra POP, một nỗ lực để làm đơn giản nhất có thể để tải các bản copy của email để đọc khi offline, sau đó xóa những email này từ remote server.

#### **-Ưu điểm của Giao Thức POP3**

Mail được lưu cục bộ, tức luôn có thể truy cập ngay cả khi không có kết nối Internet.

Kết nối Internet chỉ dùng để gửi và nhận mail.

Tiết kiệm không gian lưu trữ trên server.

Được lựa chọn để lại bản sao mail trên server.

Hộp nhất nhiều tài khoản email và nhiều server vào một hộp thư đến.

#### **- Nhược điểm của Giao Thức POP3**

Mỗi lần nhận mail, POP sẽ download email đó về máy local (và mặc định xóa mail trên server đi) nên bạn sẽ không thể sử dụng nhiều thiết bị để quản lý cùng một tài khoản email qua giao thức POP. Tuy nhiên, bạn có thể cấu hình email client để POP3 không xóa email trên server mà chỉ “mark as read” – đánh dấu đã đọc với những email đó.

**d.giao thức imap:**

IMAP là chữ viết tắt của Internet Message Access Protocol là một giao thức được ra đời vào năm 1986 cho phép bạn truy cập email của mình mọi lúc mọi nơi, từ bất kỳ thiết bị.

Khi bạn đọc email bằng cách dùng IMAP, bạn thực sự không phải tải xuống hoặc lưu trữ nó trên máy tính của mình. Thay vào đó, bạn đang đọc nó từ dịch vụ email. Nhờ đó, bạn có thể kiểm tra email bạn từ thiết bị khác nhau như điện thoại, máy tính,... ở bất kỳ đâu trên thế giới.

IMAP chỉ tải xuống thư khi bạn bấm vào nó và phân đính kèm không tự động được tải xuống. Cách này giúp bạn có thể kiểm tra thư của bạn nhanh chóng nhiều hơn POP.

**-Cách thức hoạt động:**

Kết nối đến server

Lấy nội dung được người dùng yêu cầu và lưu vào bộ nhớ đệm trên thiết bị cục bộ.

VD: danh sách các email mới, danh sách email theo một truy vấn tìm kiếm...

Hiển thị các nội dung đã lấy trên phần mềm của người dùng, xử lý các biên tập thêm/xóa/sửa của người dùng lên các email.

Ngắt kết nối.

Các bước làm việc của IMAP phức tạp hơn so với POP. Về cơ bản, email vẫn được lưu trên server, người dùng chỉ copy một bản lưu tạm trên thiết bị cá nhân (Một số phần mềm cho phép tải một hoặc nhiều email được chỉ định hoàn toàn về thiết bị).

**-Ưu điểm của IMAP:**

Mail được lưu trữ trên server, có thể truy cập từ nhiều thiết bị và địa điểm khác nhau (điện thoại, máy tính...). Xem nhanh hơn do chỉ cần tải trước danh sách email với

các tiêu đề với các yêu cầu rõ ràng thay vì phải đồng bộ toàn bộ như POP. Tiết kiệm không gian lưu trữ trên thiết bị cục bộ đặc biệt là điện thoại vẫn cho phép lưu trữ email cục bộ (cài đặt trên phần mềm). Việc sử dụng IMAP trên điện thoại sẽ làm cho việc đồng bộ dữ liệu dễ dàng giữa điện thoại máy tính và server.

**-Nhược điểm của IMAP:**

Không như POP3, IMAP yêu cầu truy cập liên tục trong suốt quá trình làm việc với các e-mail. Vì IMAP lưu các email trên một remote mail server, nên dung lượng hộp thư của bạn sẽ bị giới hạn bởi các nhà cung cấp dịch vụ mail. Nếu bạn có một lượng lớn email cần lưu trữ, bạn sẽ gặp nhiều vấn đề khi gửi nhận mail khi hộp thư bị đầy. Để giải quyết vấn đề này bằng cách tạo một bản sao copy của các email đó thông qua mail client, sau đó xóa bỏ email gốc trên server.

**e.giao thức smtp:**

SMTP là viết tắt của từ Simple Mail Transfer Protocol dịch ra có nghĩa là giao thức truyền tải thư tín đơn giản hóa, giao thức này thực hiện nhiệm vụ chính là gửi mail còn việc nhận mail hay truy xuất dữ liệu server mail sẽ có các phương thức khác để nhận như IMAP hay POP3.

Những máy chính chủ giúp bạn thực hiện gửi thư người ta thường gọi là SMTP server chúng thực hiện gửi thư qua giao thức TCP hoặc IP.

**-Ưu điểm của SMTP:**

Được phát triển từ rất sớm, và là chuẩn sử dụng của hầu hết server mail trên thế giới

**-Nhược điểm:**

Chỉ sử dụng định dạng NVT 7 bit ASCII không có chức năng nhận thực bản tin gửi đi không được mã hóa dễ bị spam, giả danh địa chỉ người gửi



#### **f.giao thức và số cổng tương ứng:**

giao thức	Cách sử dụng	Số cổng không mã hoá	Số cổng mã hoá
POP3	Thư đến	110	995
IMAP	Thư đến	143	993
SMTP	Thư đi	25	465/587

*Bảng 1.2. Bảng giao thức mạng và số cổng tương ứng*

#### **g.Mã hóa Base64:**

Base64 là một chương trình mã hóa chuỗi ký tự bằng cách dùng thay thế các ký tự trong bảng mã ASCII 8 bit thông dụng thành bảng mã 6 bit. Nó thường được sử dụng để mã hóa các tập tin đa phương tiện (hình ảnh, âm thanh, video,...). Ký tự 64 trong Base64 là đại diện cho 64 ký tự trong bảng mã ASCII

Base64 thường được sử dụng trong việc truyền tải email. Tuy nhiên, ngày nay người ta đã sử dụng nó vào việc truyền tải hình ảnh trên website.

Chuẩn Base64 là một tập hợp gồm các ký tự (theo đúng thứ tự) : từ A đến Z, từ a đến z, từ 0 đến 9, dấu +, dấu /

Tổng cộng là 64 ký tự biểu diễn 64 giá trị từ 0 đến 63. Như vậy, ký tự từ A đến Z biểu diễn cho các giá trị từ 0 đến 25, từ a đến z biểu diễn cho giá trị từ 26 đến 51, từ 0 đến 9 biểu diễn cho giá trị từ 52 đến 61, dấu + biểu diễn cho giá trị 62, dấu / biểu diễn cho giá trị 63.

Một ký tự biểu diễn theo mã ASCII sẽ dùng 8 bits. Một ký tự theo Base64 sẽ dùng 6 bits. Như vậy, một file ở dạng Base64 sẽ có kích thước lớn hơn khi ở dạng ASCII. Cụ thể, sẽ lớn gấp 4/3 lần (8 bits/6 bits).

**Để chuyển đổi file sang dạng Base64, ta thực hiện theo các bước như sau :**

1. Đọc nội dung file dưới dạng bit.
2. Cứ 6 bits ta tách thành một nhóm để xử lý.

3. Tra bảng mã Base64, mỗi nhóm 6 bits sẽ có giá trị tương ứng với một ký tự.

4. Ghi ra file các ký tự đó. Vậy là xong.

Ta sẽ thực hiện các thao tác trên qua một ví dụ cụ thể.

Chuyển từ Man sang mã base64 sẽ là TWFu

Trong ví dụ trên, Mã ASCII, các ký tự M, a, n sẽ là 77, 97, và 110 với 8-bit nhị phân là 01001101, 01100001, và 01101110. Có 3 giá trị được nối lại với nhau thành một chuỗi 24bit cụ thể là 010011010110000101101110. Một nhóm 6bit có tối đa  $2^6 = 64$  các giá trị nhị phân khác nhau sẽ được tính từ trái sang phải. (trong trường hợp này 24bit ta sẽ chia thành 4 nhóm từ trái sang phải

010011 010110 000101 101110 sau đó tra bảng mã base64 sẽ được giá trị base64 tương ứng).

Text content	M	a	n
ASCII	77 (0x4d)	97 (0x61)	110 (0x6e)
Bit pattern	01001101	01100001	01101110
Index	19	22	5
Base64-encoded	T	W	Fu

Bảng 1.3. Bảng mã chuyển đổi file

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Base 64 table codes

*Bảng 1.4. Bảng tra mã*

Khi số bytes không chia hết cho 3 điều đó có nghĩa là thiếu 1 hay 2 bytes thì ta thêm phần bytes phụ với giá trị bằng 0 và thực hiện chuyển đổi.

## CHƯƠNG 2: THỬ NGHIỆM ỨNG DỤNG

### 2.1. Môi trường cài đặt

Từ những cơ sở lý thuyết, phương pháp phân tích được đề cập ở chương 1. Chúng em đã xây dựng và phát triển thành công ứng dụng Gmail trên môi trường Apache NetBeans, viết bằng ngôn ngữ lập trình Java.

### 2.2. Triển khai ứng dụng

- Để triển khai sử dụng được mail client do bạn tự code ra thì trước tiên ta phải bật cho phép ứng dụng ít an toàn truy xuất mail

1. Open your Google Admin console (admin.google.com). Click Security > Basic settings.

2. Under Less secure apps, select Go to settings for less secure apps.
3. In the subwindow, select the Enforce access to less secure apps for all users radio button.
4. (You can also use the Allow users to manage their access to less secure apps, but don't forget to turn on the less secure apps option in users settings then!)
5. Click the Save button.

## 2.3. DEMO

### 2.3.1 Ứng dụng download attachment bằng pop3

```
* @author admin
*/
public class EmailAttachmentReceiver {
    private String saveDirectory;
    private boolean expunge;

    /**
     * Sets the directory where attached files will be stored.
     * @param dir absolute path of the directory
     */
    public void setSaveDirectory(String dir) {
        this.saveDirectory = dir;
    }

    /**
     * Downloads new messages and saves attachments to disk if any.
     * @param host
     * @param port
     * @param userName
     * @param password
     */
    public void downloadEmailAttachments(String host, String port,
        String userName, String password) {
        Properties properties = new Properties();
        // server setting
        properties.put("mail.pop3.host", host);
        properties.put("mail.pop3.port", port);
        // SSL setting
        properties.setProperty("mail.pop3.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        properties.setProperty("mail.pop3.socketFactory.fallback", "false");
        properties.setProperty("mail.pop3.socketFactory.port",
            String.valueOf(port));
        Session session = Session.getDefaultInstance(properties);
```

*Hình 2.1. Code download file đính kèm*

```

try {
    // connects to the message store
    Store store = session.getStore("pop3");
    store.connect(userName, password);
    // opens the inbox folder
    Folder folderInbox = store.getFolder("INBOX");
    folderInbox.open(Folder.READ_ONLY);
    // fetches new messages from server
    Message[] arrayMessages = folderInbox.getMessages();
    for (int i = 0; i < arrayMessages.length; i++) {
        Message message = arrayMessages[i];
        Address[] fromAddress = message.getFrom();
        String from = fromAddress[0].toString();
        String subject = message.getSubject();
        String sentDate = message.getSentDate().toString();
        String contentType = message.getContentType();
        String messageContent = "";
        // store attachment file name, separated by comma
        String attachFiles = "";
        if (contentType.contains("multipart")) {
            // content may contain attachments
            Multipart multiPart = (Multipart) message.getContent();
            int numberOfParts = multiPart.getCount();
            for (int partCount = 0; partCount < numberOfParts; partCount++) {
                MimeBodyPart part = (MimeBodyPart) multiPart.getBodyPart(partCount);
                if (Part.ATTACHMENT.equalsIgnoreCase(part.getDisposition())) {
                    // this part is attachment
                    String fileName = part.getFileName();
                    attachFiles += fileName + ", ";
                    part.saveFile(saveDirectory + File.separator + fileName);
                } else {
                    // this part may be the message content
                    messageContent = part.getContent().toString();
                }
            }
        }
    }
}

```

*Hình 2.2. Code download file đính kèm*

```

    }
    if (attachFiles.length() > 1) {
        attachFiles = attachFiles.substring(0, attachFiles.length() - 2);
    }
} else if (contentType.contains("text/plain")
    || contentType.contains("text/html")) {
    Object content = message.getContent();
    if (content != null) {
        messageContent = content.toString();
    }
}

// print out details of each message
System.out.println("Message #" + (i + 1) + ":");
System.out.println("\t From: " + from);
System.out.println("\t Subject: " + subject);
System.out.println("\t Sent Date: " + sentDate);
System.out.println("\t Message: " + messageContent);
System.out.println("\t Attachments: " + attachFiles);
System.out.println("message"+i+1+"attachment downloaded");
message.setFlag(Flags.Flag.DELETED, true);

}

// boolean expunge = true;
// folderInbox.close(expunge);
// disconnect
folderInbox.close(false);
store.close();
store.close();
} catch (NoSuchProviderException ex) {
    System.out.println("No provider for pop3.");
    ex.printStackTrace();
} catch (MessagingException ex) {
    System.out.println("Could not connect to the message store");
    ex.printStackTrace();
}

```

*Hình 2.3. Code download file đính kèm*

```

public static void main(String[] args) {
    String host = "pop.gmail.com";
    String port = "995";

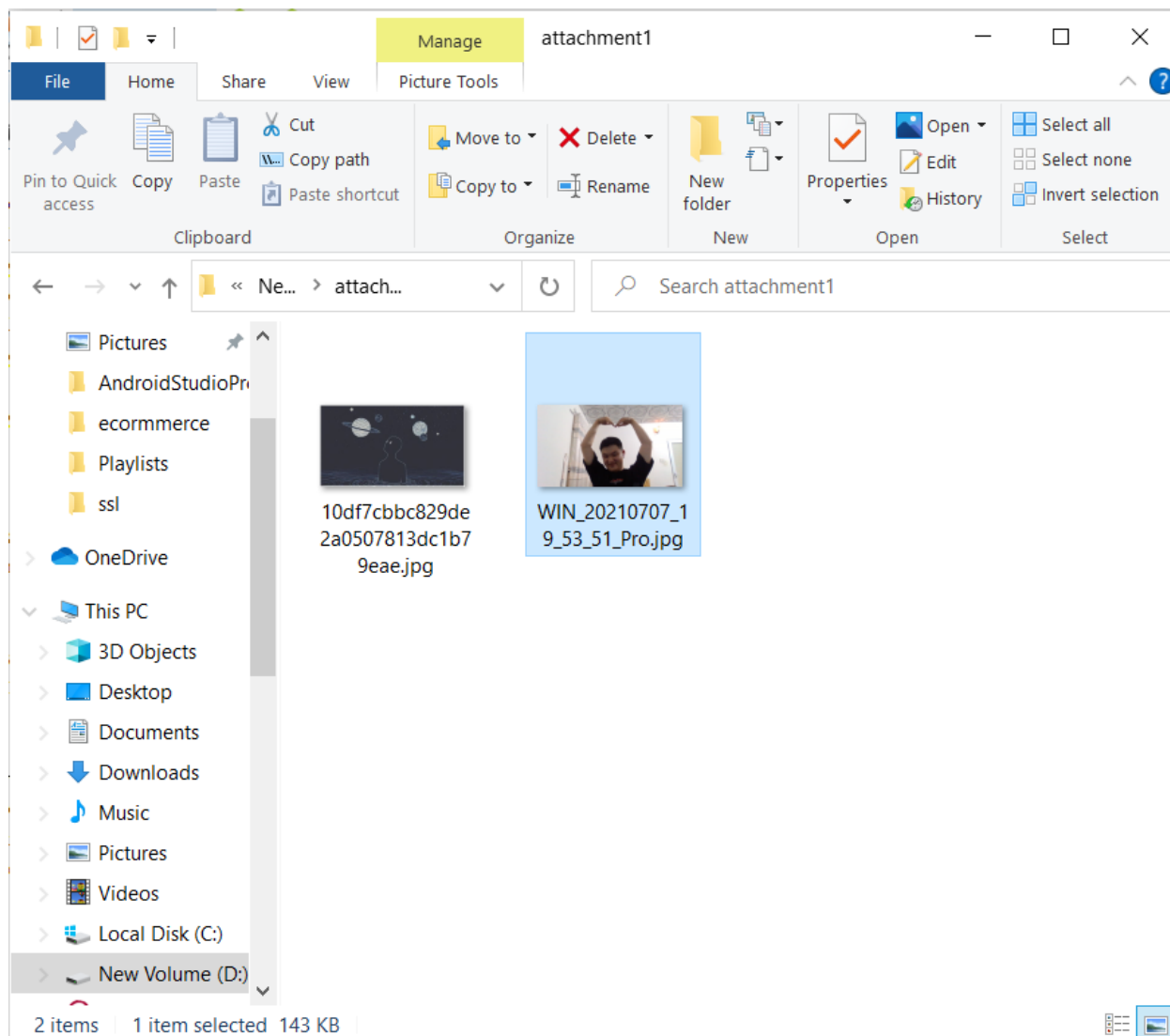
    String userName = "laptrinhmangbc@gmail.com";
    String password = "A1122334455";

    String saveDirectory = "D:/attachment1";
    EmailAttachmentReceiver receiver = new EmailAttachmentReceiver();
    receiver.setSaveDirectory(saveDirectory);
    receiver.downloadEmailAttachments(host, port, userName, password);
    System.out.println("download ended");
}

```

*Hình 2.4. Code download file đính kèm*

## Kết quả download file đính kèm:



Hình 2.5. Kết quả download file đính kèm

### 2.3.2. Ứng dụng xóa mail sử dụng Imap

```
public void deleteMessages(String Host, String Port,
    String username, String password) {
    Properties prpt = new Properties();

    // server setting
    prpt.put("mail.imap.host", Host);
    prpt.put("mail.imap.port", Port);

    // SSL setting
    prpt.setProperty("mail.imap.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    prpt.setProperty("mail.imap.socketFactory.fallback", "false");
    prpt.setProperty("mail.imap.socketFactory.port", String.valueOf(Port));
    Session ss = Session.getDefaultInstance(prpt);
    System.out.println(username);
    System.out.println(password);
    try {
        // connects to the message store
        Store str = ss.getStore("imap");
        System.out.println(username);
        System.out.println(password);
        str.connect(username, password);

        // opens the inbox folder
        Folder folderInbox = str.getFolder("INBOX");
        folderInbox.open(Folder.READ_WRITE);

        // fetches new messages from server
        Message[] arrayMessages = folderInbox.getMessages();

        for (int i = 0; i < arrayMessages.length; i++) {
            Message message = arrayMessages[i];
            String subject = message.getSubject();

            message.setFlag(Flags.Flag.DELETED, true);
        }
    }
}
```

Hình 2.6. Code xóa email



```

        message.setFlag(Flags.Flag.DELETED, true);
        System.out.println("Marked DELETE for message: " + (i+1));
    }

    // expunges the folder to remove messages which are marked deleted
    boolean expunge = true;
    folderInbox.close(expunge);

    // another way:
    //folderInbox.expunge();
    //folderInbox.close(false);

    // disconnect
    str.close();
} catch (NoSuchProviderException ex) {
    System.out.println("No provider.");
    ex.printStackTrace();
} catch (MessagingException ex) {
    System.out.println("Could not connect to the message store.");
    ex.printStackTrace();
}
}

```

*Hình 2.7. Code xóa email*

```

/**
 * Runs this program to delete e-mail messages on a Gmail account
 * via IMAP protocol.
 */
public static void main(String[] args) {
    String host = "imap.gmail.com";
    String port = "993";
    String userName = "laptrinhmangbc@gmail.com";
    String password = "A1122334455";
    EmailMessageRemover remover = new EmailMessageRemover();

    // try to delete all messages contain this string its Subject field
    String subjectToDelete = "Newsletter";
    remover.deleteMessages(host, port, userName, password);
}
}

```

Hình 2.8. Code xóa email

### Kết quả xóa email:

```

laptrinhmangbc@gmail.com
A1122334455
laptrinhmangbc@gmail.com
A1122334455
Marked DELETE for message: 1
Marked DELETE for message: 2
Marked DELETE for message: 3
Marked DELETE for message: 4
Marked DELETE for message: 5
BUILD SUCCESSFUL (total time: 9 seconds)

```

Hình 2.9. Kết quả xóa email

## **CHƯƠNG 3: KẾT LUẬN**

### **3.1 Kết quả đạt được**

- Về tri thức:

Trong suốt khoảng thời gian 2 tuần, chúng em đã học được rất nhiều về kiến thức nền tảng, các phương pháp tìm và đọc tài liệu.

- Về ứng dụng:

Đã đạt được các yêu cầu đề về xây dựng chương trình Gmail như:

- Cho phép đăng nhập vào Gmail.
- Cho phép tự động download các file đính kèm và xóa mail.

- Về con người: khoảng thời gian 2 tuần là khoảng thời gian đủ để chúng em hoàn thành tiểu luận, mặc dù cũng gặp đôi chút khó khăn. Tuy nhiên chúng em đã hoàn thành xong bài tiểu luận được giao.

### **3.2 Hạn chế**

Vì thời gian thực hiện đề tài cũng như kiến thức của chúng em là có giới hạn nên vẫn còn vài thiếu sót như: chưa có giao diện.

## BẢNG KHỐI LƯỢNG CÔNG VIỆC

1. Họp team về đề tài 27/6
2. Họp team về đề tài 30/6 chia công việc
3. Họp team về đề tài 3/7
4. Tìm kiếm tài liệu
5. Xây dựng demo
6. Làm báo cáo word

Tham gia 100%	Tham gia 60%	Tham gia 30%
---------------	--------------	--------------

	1	2	3	4	5	6
HUY						
CHIẾN						
QUI						

## TÀI LIỆU THAM KHẢO

1. <https://gpcoder.com/3753-huong-dan-su-dung-thu-vien-java-mail/>
2. <https://www.codejava.net/java-ee/javamail/get-attachments-in-e-mail-messages-using-javamail>
3. <https://www.hostinger.vn/huong-dan/giai-thich-giao-thuc-pop3-smtp-imap-la-gi-va-port-cua-chung>
4. <https://braniumacademy.net/c%C3%A1c%20Kh%C3%B3a%20h%E1%BB%8Dc/khoa-hoc-lap-trinh-java-voi-netbeans/>