TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN – ĐIỆN TỬ

Logo

Description automatically generated

**TIỂU LUẬN CUỐI KỲ**

**“MESBOX”** **QUẢN LÝ DỮ LIỆU TÀI KHOẢN VÀ TIN NHẮN**

**Môn học: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

Giảng viên: PGS.TS. Hoàng Văn Dũng

**Danh sách sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| **Mã số SV** | **Họ và tên** | **Mức độ**  **đóng góp (%)** |
| 20139003 | Tô Gia Huy | 100% |
| 20139084 | Lê Đức Phong | 100% |
| 20139072 | Nguyễn Lương Phú Gia | 100% |
| 20139040 | Trần Nguyễn Quang Lâm | 100% |

*TP. Hồ Chí Minh, tháng 05 năm 2022*

**NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

TP. Hồ Chí Minh, ngày 29 tháng 05 năm 2022

Giảng viên chấm điểm

**LỜI CẢM ƠN**

Để hoàn thành tốt đề tài và bài báo cáo này,chúng em xin gửi lời cảm ơn chân thành

đến giảng viên, Hoàng Văn Dũng,người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm ơn thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn đã đề ra.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với

những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được nhữngýkiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau.Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc thầy luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

Tp.Hồ Chí Minh, ngày 30 tháng 5 năm 2022

Nhóm sinh viên thực hiện

# 

MỤC LỤC

[**PHẦN I : MỞ ĐẦU 1**](#_Toc104847641)

[**1.LÝ DO CHỌN ĐỀ TÀI 1**](#_Toc104847642)

[**Tên đề tài : MesBox : Quản lý cơ sở dữ liệu và tin nhắn 1**](#_Toc104847643)

[**2.TỔNG QUAN ĐỀ TÀI 1**](#_Toc104847644)

[**2.1 Chức năng : 1**](#_Toc104847645)

[**2.2 Mục tiêu đề tài : 1**](#_Toc104847646)

[**PHẦN II: CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP SỬ DỤNG 2**](#_Toc104847647)

[**1. CƠ SỞ LÝ THUYẾT 2**](#_Toc104847648)

[**2.PHƯƠNG PHÁP SỬ DỤNG 2**](#_Toc104847649)

[**PHẦN III : THIẾT KẾ CHƯƠNG TRÌNH 5**](#_Toc104847650)

[**1. CHỨC NĂNG VÀ THUẬT TOÁN 5**](#_Toc104847651)

[**1.1 Chức năng của chương trình 5**](#_Toc104847652)

[**1.2 Thuật toán và lưu đồ thuật toán 6**](#_Toc104847653)

[**2.CƠ SỞ DỮ LIỆU 13**](#_Toc104847654)

[**2.1. Cài đặt các phương pháp và thuật toán trong chương trình 13**](#_Toc104847655)

[**1.2. Thiết kế cơ sở dữ liệu: 20**](#_Toc104847656)

[**3.KẾT QUẢ 27**](#_Toc104847657)

[**3.1 Hướng dẫn sử dụng 27**](#_Toc104847658)

[**3.2 Kết quả đề tài 28**](#_Toc104847659)

[**PHẦN 4: KẾT LUẬN 32**](#_Toc104847660)

[**1.KẾT LUẬN 32**](#_Toc104847661)

[**2.HƯỚNG PHÁT TRIỂN VÀ MỞ RỘNG 32**](#_Toc104847662)

# PHẦN I : MỞ ĐẦU

## 1.LÝ DO CHỌN ĐỀ TÀI

Tên đề tài : MesBox : Quản lý cơ sở dữ liệu và tin nhắn

Hiện nay nhu cầu nhắn tin, trao đổi thông tin là cần thiết và không thể thiếu không chỉ những người trẻ mà còn với tất cả mọi người. Trao đổi rất quan trọng, thông tin còn cần được lưu trữ và dễ dàng tìm lại bất cứ khi nào và cần có sự bảo mật. Kết hợp với những yếu tố trên nhóm chúng em thực hiện đề tài này để nhằm tạo ra một ứng dụng quản lý tin nhắn cá nhân một cách hiệu quả, tiện lợi.

Để kết thúc môn học Giải Thuật và Cấu Trúc Dữ Liệu của thầy Hoàng Văn Dũng , nhóm em đã vận dụng các kiến thức đã được học ở lớp trong thời gian qua để làm một dự án mà tụi em đặt tên nó là Mesbox. Đây là một chương trình có tính thực tế vận dụng kiến thức về thuật toán mà chúng em đã học ở lớp. Phần mềm có tính bảo mật, quản lí tin nhắn được gửi, nhận… Giúp mọi người quản lí thông tin tin nhắn của mình một cách rõ ràng, tiện lợi.

## 2.TỔNG QUAN ĐỀ TÀI

### 2.1 Chức năng :

MesBox được thiết kế để tạo ra một ứng dụng cho người dùng có thể khởi tạo tài khoản hoàn toàn miễn phí và sử dụng như một công cụ vận chuyển / nhận thư giữa cá nhân với cá nhân, danh sách tin nhắn đã nhận, gửi, đã xóa, gắn sao, tìm kiếm các thư được gửi/ nhận từ một user. Giúp quản lí dữ liệu dễ dàng, minh bạch.

### 2.2 Mục tiêu đề tài :

Đề tài mà chúng em trình bày hướng đến việc quản lý tin nhắn cá nhân, trao đổi thông tin giữa các người dùng vì thế mục tiêu của đề tài là đáp ứng được khả năng nhận / gửi thao tác với tin nhắn , thể hiện được đầy đủ chính xác nội dung, thời gian và thông tin người nhận gửi , dễ dàng thao tác với chương trình.

# PHẦN II: CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP SỬ DỤNG

## 1. CƠ SỞ LÝ THUYẾT

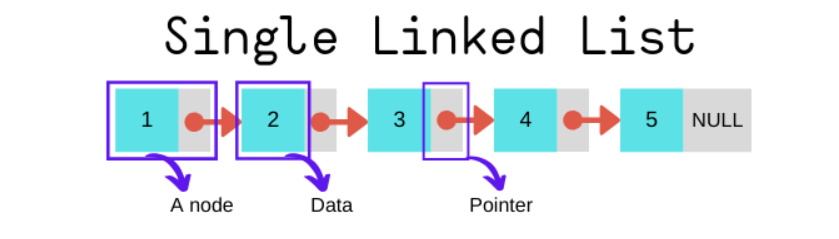
* Ngôn ngữ và công cụ sử dụng: với đề tài này nhóm em lập trình trên ngôn ngữ C++ và công cụ lập trình DEVC++, VSCode.
* Tài liệu về cấu trúc dữ liệu và giải thuật của thầy Hoàng Văn Dũng giảng dạy.

## 2.PHƯƠNG PHÁP SỬ DỤNG

**Lý thuyết về danh sách liên kết đơn:**

Danh sách liên kết đơn (Single Linked List) là một cấu trúc dữ liệu động, nó là một danh sách mà mỗi phần tử đều liên kết với phần tử đúng sau nó trong danh sách. Mỗi phần tử (được gọi là một node hay nút) trong danh sách liên kết đơn là một cấu trúc có hai thành phần:

* Thành phần dữ liệu: lưu thông tin về bản thân phần tử đó.
* Thành phần liên kết: lưu địa chỉ phần tử đứng sau trong danh sách, nếu phần tử đó là phần tử cuối cùng thì thành phần này bằng NULL.



*Minh họa về danh sách liên kết đơn*

Đặc điểm của danh sách liên kết đơn

Do danh sách liên kết đơn là một cấu trúc dữ liệu động, được tạo nên nhờ việc cấp phát động nên nó có một số đặc điểm sau đây:

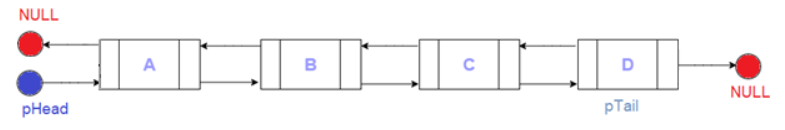
* Được cấp phát bộ nhớ khi chạy chương trình
* Có thể thay đổi kích thước qua việc thêm, xóa phần tử
* Kích thước tối đa phụ thuộc vào bộ nhớ khả dụng của RAM
* Các phần tử được lưu trữ ngẫu nhiên (không liên tiếp) trong RAM

Và do tính liên kết của phần tử đầu và phần tử đứng sau nó trong danh sách liên kết đơn, nó có các đặc điểm sau:

* Chỉ cần nắm được phần tử đầu và cuối là có thể quản lý được danh sách
* Truy cập tới phần tử ngẫu nhiên phải duyệt từ đầu đến vị trí đó
* Chỉ có thể tìm kiếm tuyến tính một phần tử

**Lý thuyết về danh sách liên kết đôi:**

* Danh sách liên kết đôi (Doubly Linked List) là một dạng mở rộng của **danh sách liên kết đơn**. Trong danh sách liên kết đôi cũng bao gồm nhiều các Node (nút) và điều đặc biệt ở danh sách liên kết đôi là các Node liên kết với nhau với 2 con trỏ và theo hai chiều trước – sau. Nghĩa là một Node có thể trỏ đến địa chỉ của Node kế tiếp và cũng có thể trỏ đến địa chỉ của Node phía trước nó.



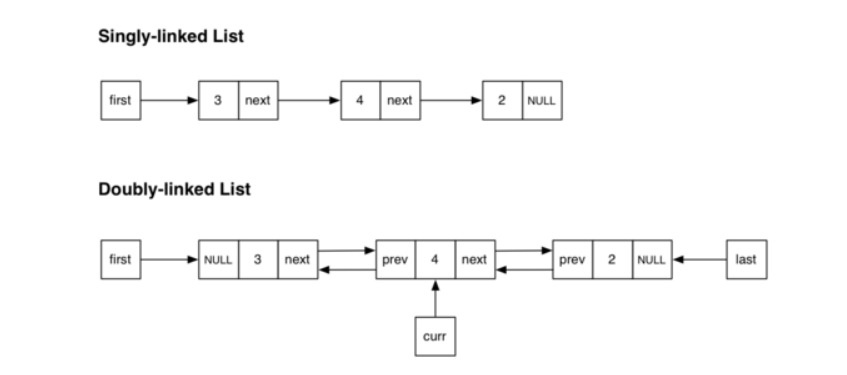
Qua hình ta có thể thấy:

Các node có 2 mối liên kết là trước và sau cũng chính là 2 con trỏ, previous và next.

* Node đầu tiên của danh sách có previous = NULL (mũ tên trỏ về phần hình tròn màu đỏ là địa chỉ NULL).



* Một Node (hay phần tử) trong danh sách liên kết đôi bao gồm 3 thành phần chính đó là:
* Phần data: lưu trữ dữ liệu của node
* Previous: Lưu trữ địa chỉ của node (hay phần tử) đứng trước. Trong trường hợp node này là node đầu tiên của danh sách thì sẽ có previous =NULL
* Next: Lưu trữ địa chỉ của node (hay phần tử) đừng sau. Trong trường hợp node này là node cuối cùng của danh sách thì sẽ có next = NULL



*Khác biệt giữa danh sách liên kết đơn với danh sách liên kết đôi*

**Thuật toán tìm kiếm tuần tự**

Tìm kiếm tuần tự (Sequential Search hay Linear Search) là một giải thuật đơn giản, rất dễ cài đặt. Bắt đầu từ đối tượng a\_1,a 1 ​ , duyệt qua tất cả các đối tượng, cho tới khi tìm thấy đối tượng có khóa mong muốn, hoặc duyệt hết toàn bộ dãy mà không tìm thấy khóa đó.

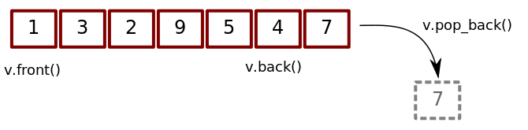


**Mô phỏng giải thuật**

**Lý thuyết về Vector**

Vector C++ là một lớp, một cấu trúc dữ liệu có trong thư viện của C++ Standard Template Library dùng để lưu các đối tượng khác giống với mảng (Array). C++ rất mạnh về tính linh hoạt và cấu trúc chính vì thế vector được sinh ra nhằm khắc phục các mặt hạn chế của mảng. Nếu như mảng chỉ có một kích thước cố định khi khai báo thì vector hoàn toàn có thể thay đổi trong khi chương trình đang hoạt động. Tức là một đối tượng vector có thể linh động về kích thước tương tự như con trỏ.

Kích thước không cố định chính là điểm nổi bật nhất của vector. Tuy nhiên không dừng lại ở đó, đối tượng này còn được trang bị thêm khá nhiều phương thức xử lý thông dụng như sort, insert, delete, leght . . . Chính nhờ tính linh hoạt của nó, đôi khi sẽ hữu ích hơn trong việc giải quyết các bài toán lập trình.



*Nguyên tắc hoạt động của vector*

# PHẦN III : THIẾT KẾ CHƯƠNG TRÌNH

## 1. CHỨC NĂNG VÀ THUẬT TOÁN

### 1.1 Chức năng của chương trình

Chương trình gồm 3 menu :

* Menu chính : đây là nơi thực hiện thao tác đầu tiên với chương trình
* Thoát khỏi chương trình .
* Tạo tài khoản mới.
* Đăng nhập vào một tài khoản của bạn .
* Xóa một tài khoản đang tồn tại.
* Thay đổi mật khẩu của tài khoản .
* Menu Thao tác với tài khoản người dùng:
* Đăng xuất khỏi tài khoản
* Gửi một tin nhắn
* Kiểm tra tin nhắn đến
* Xem tin nhắn đã gửi
* Tìm kiếm tin nhắn đã gửi
* Tìm kiếm tin nhắn tin nhắn được nhận
* Xem tin nhắn đã ghim trong InBox
* Xem tin nhắn đã ghim trong SentBox
* Menu lựa chọn thao tác với tin nhắn
* Thoát
* Xoa tin nhắn
* Ghim/ Bỏ ghim tin nhắn

### 1.2 Thuật toán và lưu đồ thuật toán

#### 1.2.1 Ý tưởng thuật toán :

- Dùng double linked list để quản lý lưu trữ tài khoản người dùng và hai singly linked list cho mỗi người dùng để lưu trữ các tin nhắn đã gửi và đã nhận.

- Người dùng có thể tạo tài khoản, đăng nhập và gửi tin nhắn cho người dùng hiện có khác.

- Người dùng đã đăng nhập có thể đăng xuất và tạo tài khoản khác, đăng nhập vào tài khoản hiện có khác, thay đổi mật khẩu của tài khoản hiện có hoặc xóa tài khoản hiện có.

- Người dùng khi đăng nhập có thể xem danh sách các tin nhắn đã nhận, đã gửi, đã xóa, có ghim và cũng có thể tìm kiếm các tin nhắn được gửi đến / nhận từ người dùng.

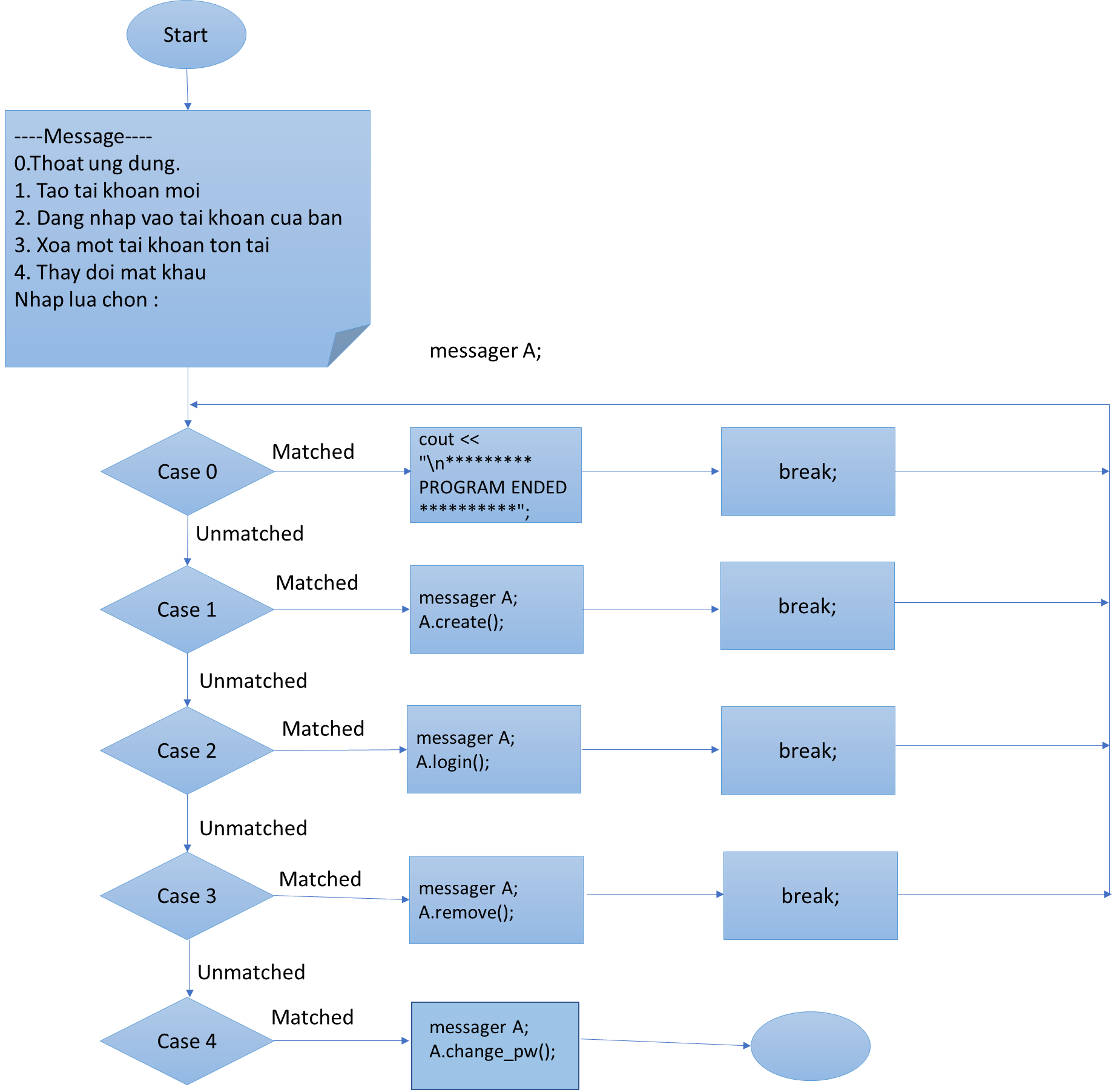
- Đối với mỗi tin nhắn đã gửi / nhận, người dùng có thể đọc nó, xóa nó, đánh dấu nó là quan trọng (dấu sao) / bỏ dấu sao.

- Để hiển thị danh sách các tin nhắn được gửi đến / nhận từ người dùng hoặc danh sách các tin nhắn được ghim sao trong các singly linked list đã gửi / nhận, chúng em đã sử dụng *thuật toán tìm kiếm tuần tự*

- Mỗi khi thấy thông báo thỏa mãn điều kiện bắt buộc, thì tham chiếu đó sẽ được lưu trữ trong một vectơ. Vectơ này giúp đọc, xóa hoặc ghim sao / bỏ ghim một thư trong danh sách nhận được do tìm kiếm tuần tự. Các tham chiếu của các tin nhắn đã xóa đã được lưu trữ trong một vectơ (thùng rác).

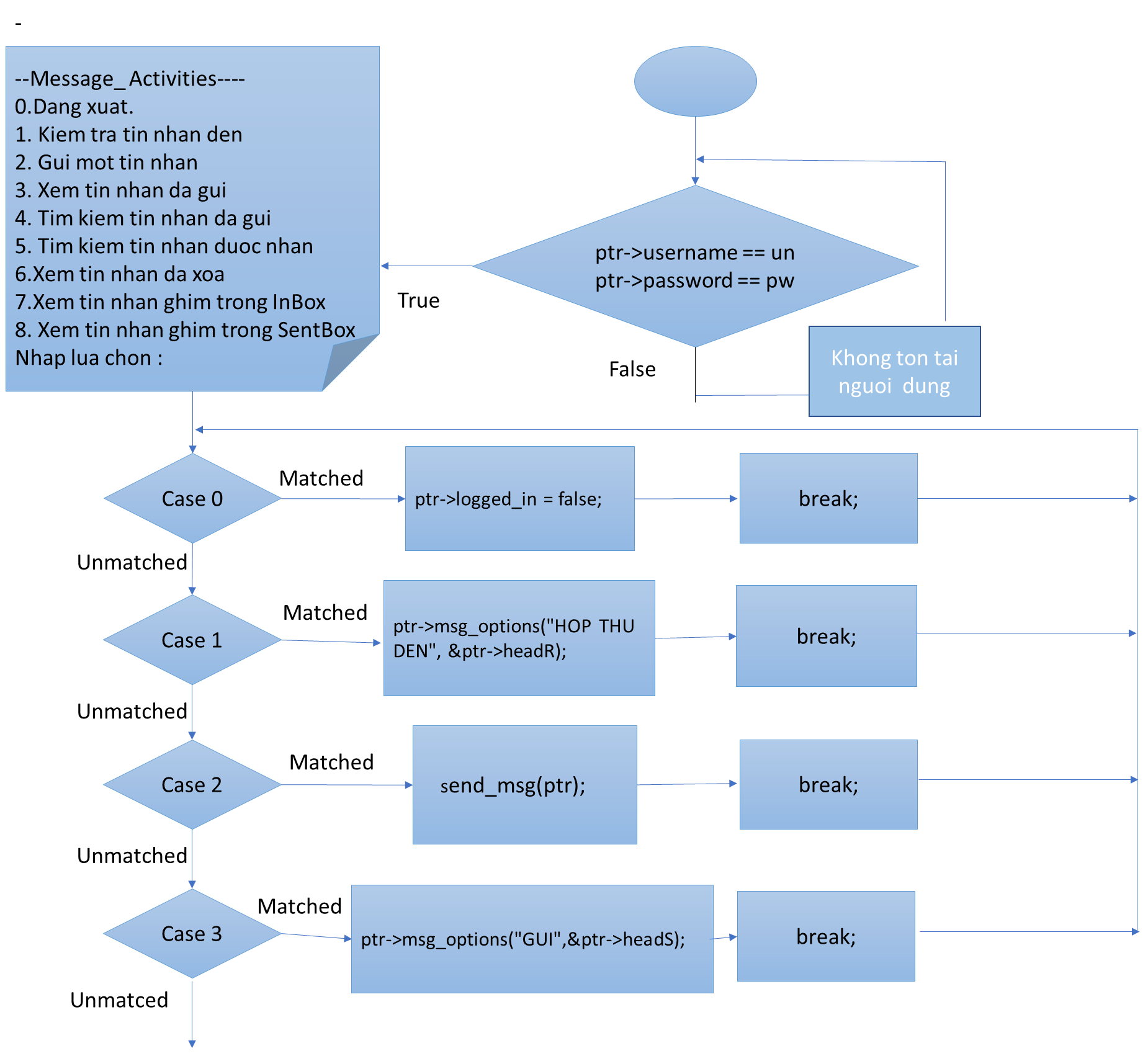
- Người dùng có thể xóa vĩnh viễn tin nhắn khỏi thùng rác hoặc chỉ cần xem nó.

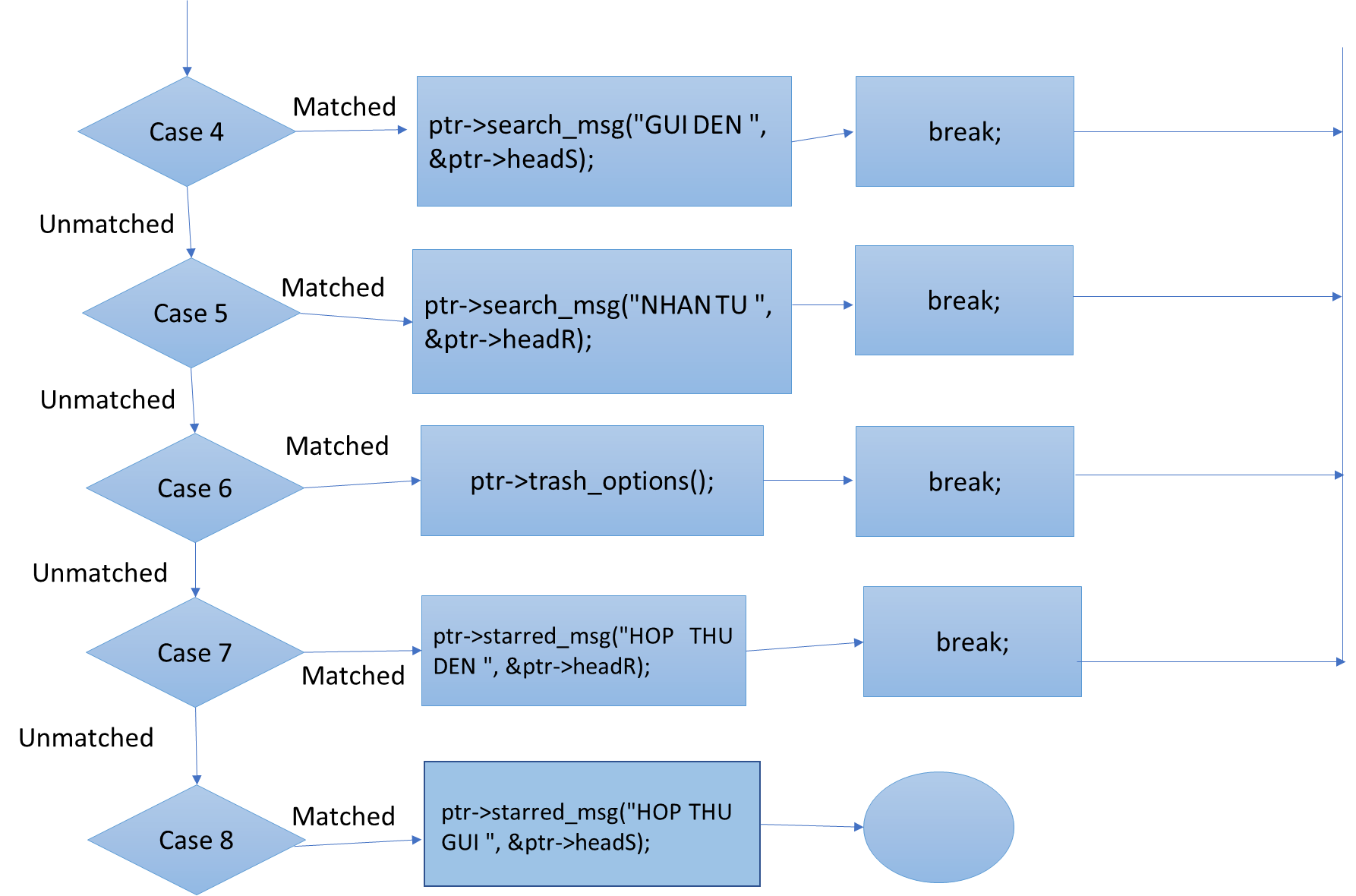
#### 1.2.2. Lưu đồ thuật toán



***Lưu đồ lựa chọn Menu chính***

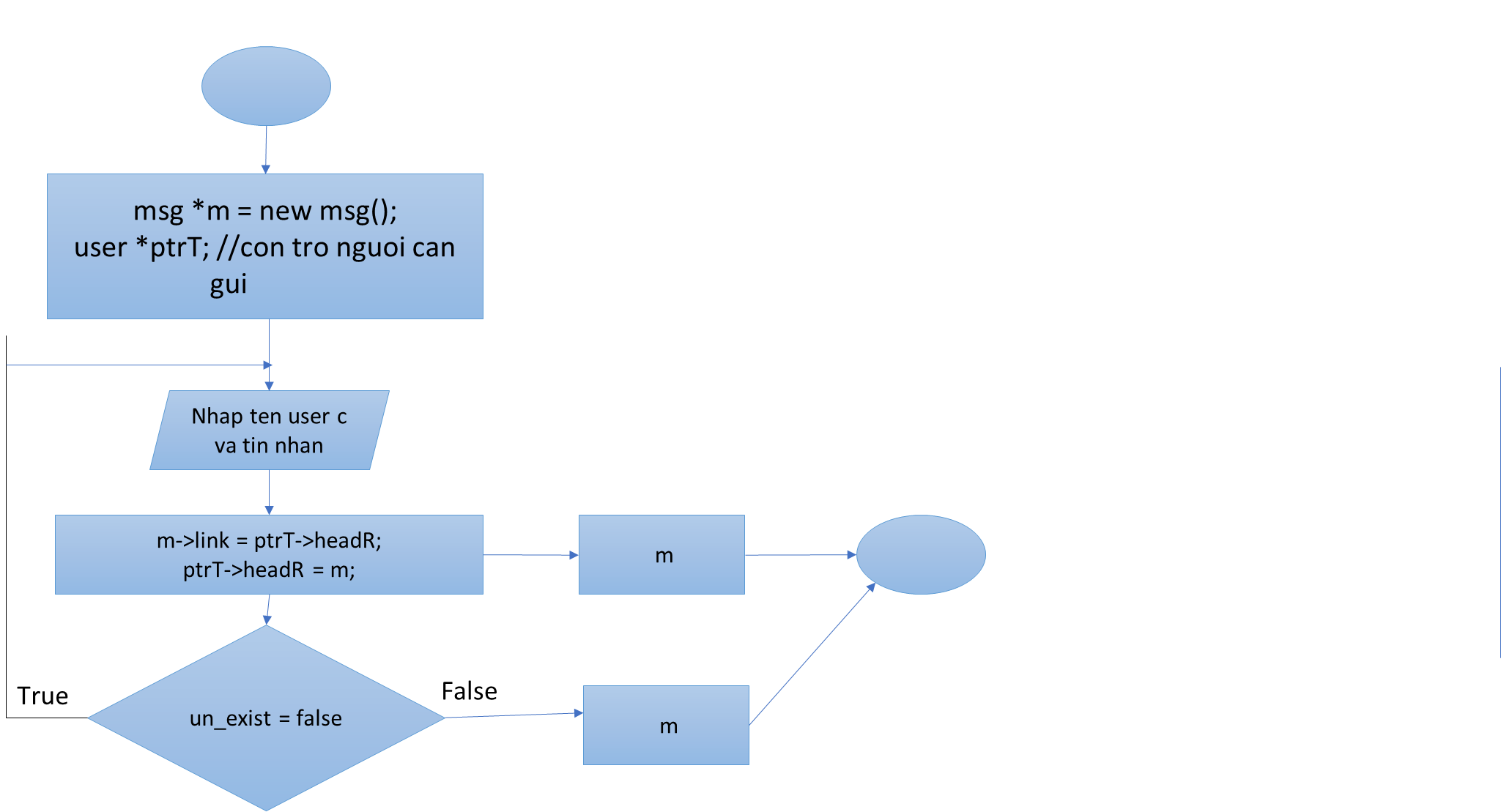
**Message Activities** : các thao thác với tin nhắn khi login vào chương trình thảo mãn username và passworrd đã tồn tại .



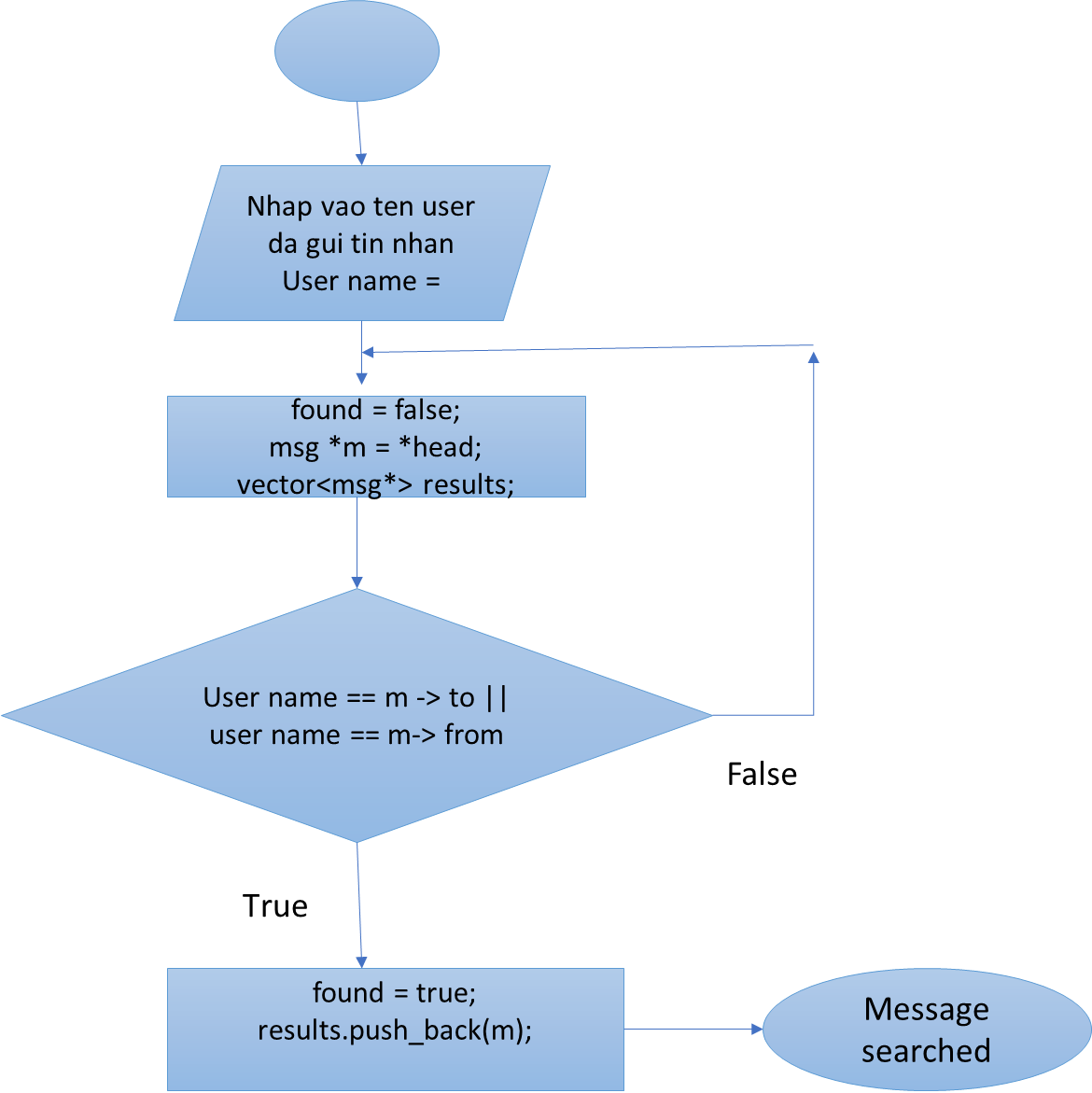


***Lưu đồ thao tác tin***

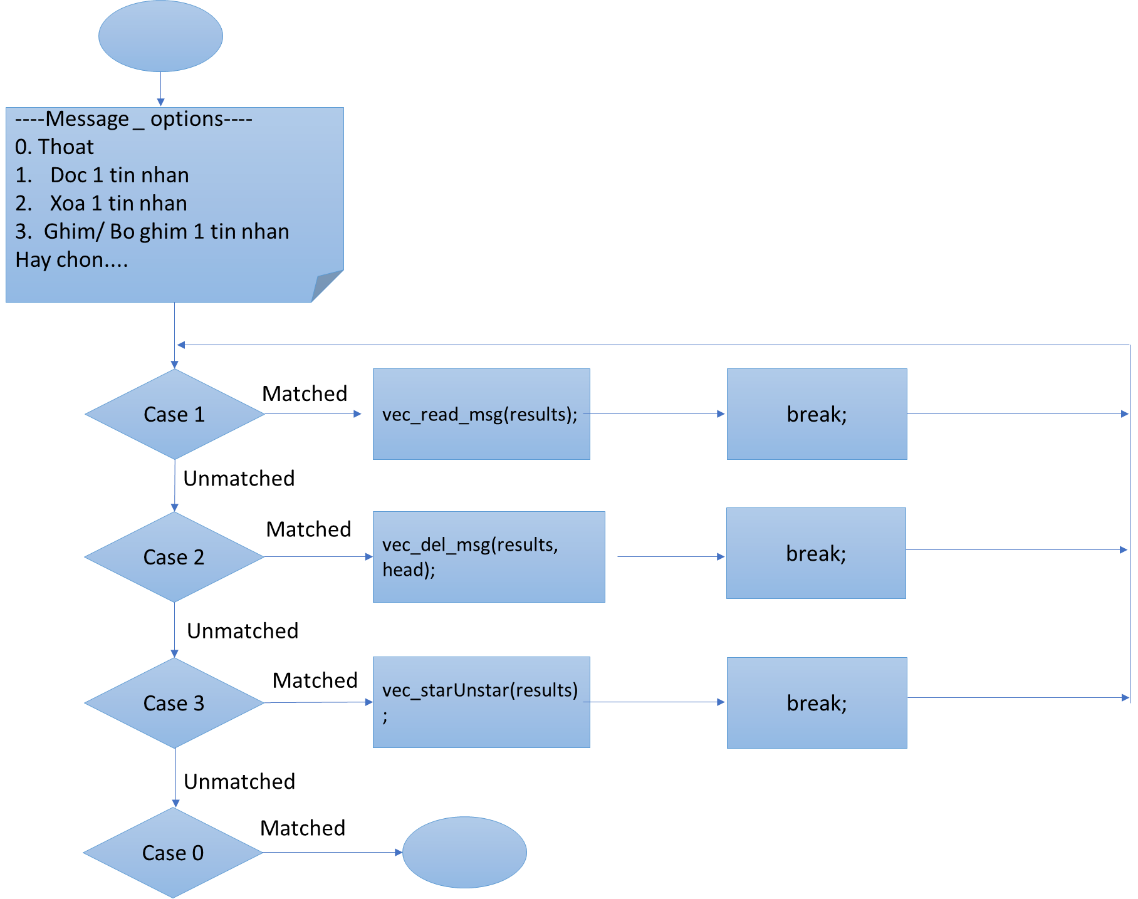
Nhập tên người dùng cần gửi , nếu tên người dùng tồn tại tiếp tục nhập nội dung tin nhắn , lưu nội dung vào m và cập nhật lại m , nếu không tồn tại tin tài khoản thi trả về con trỏ m ban đầu.



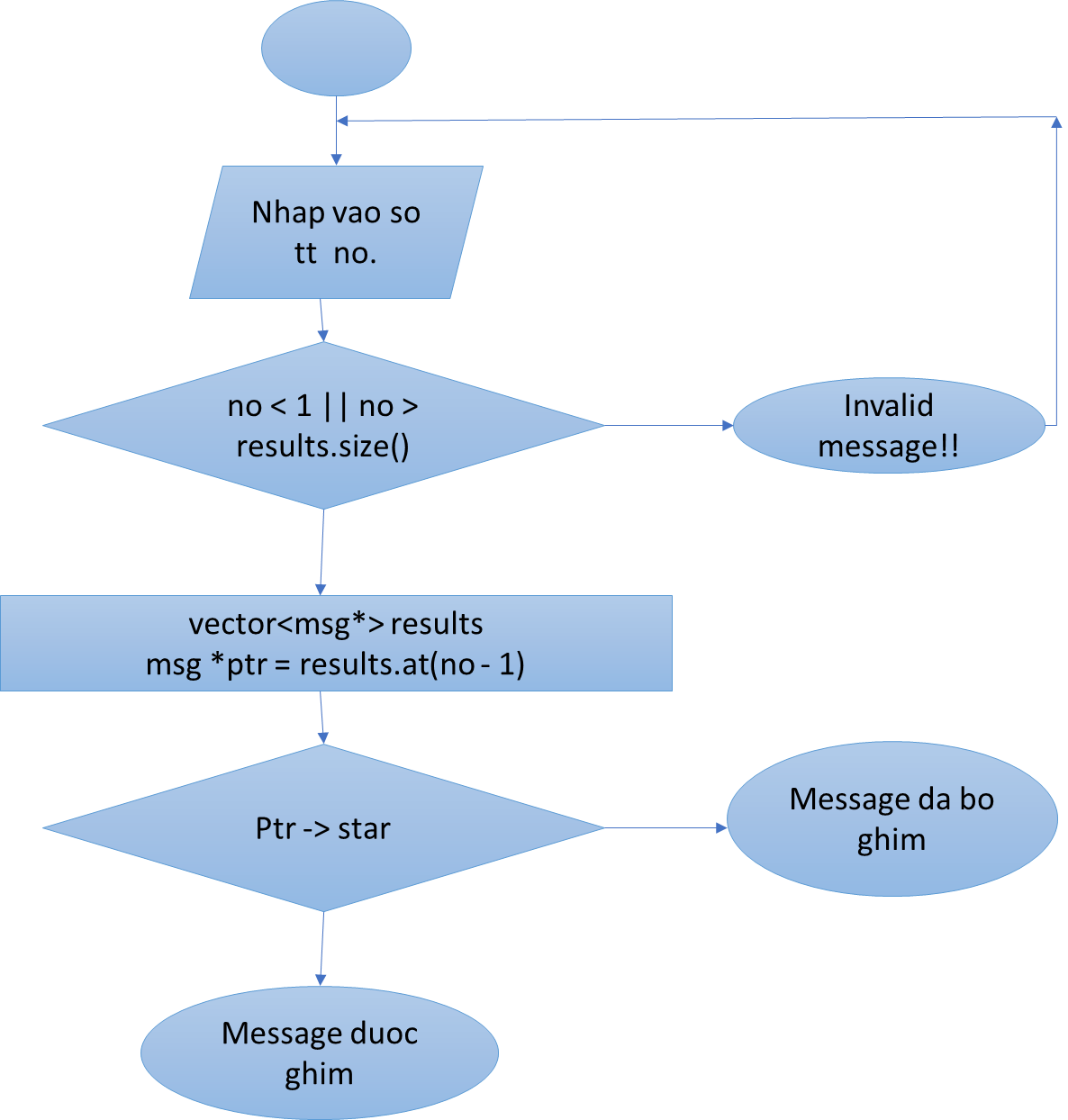
***Lưu đồ thể hiện thao tác gửi tin nhắn***



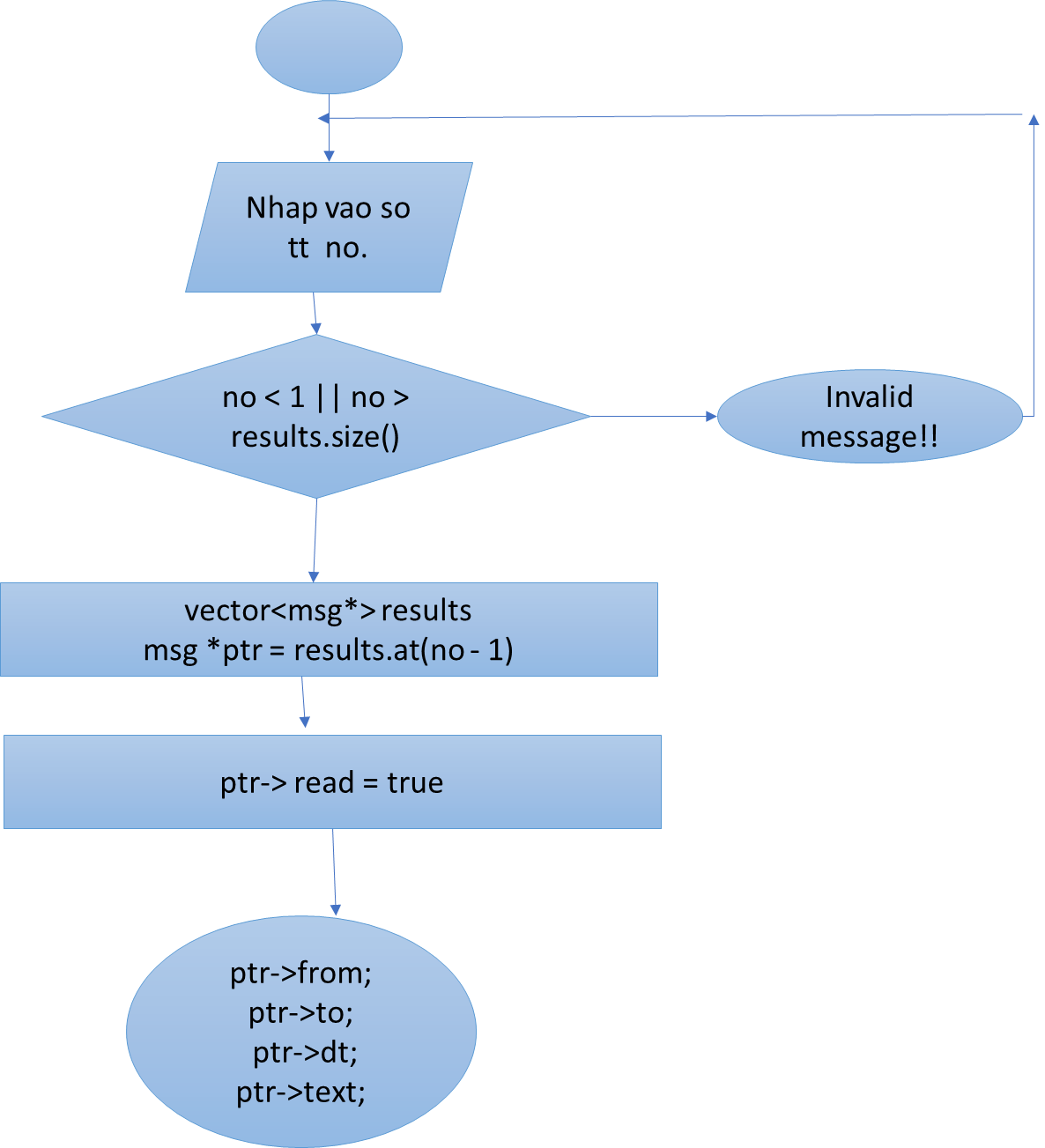
***Lưu đồ tìm kiếm tin nhắn theo tên người nhận***



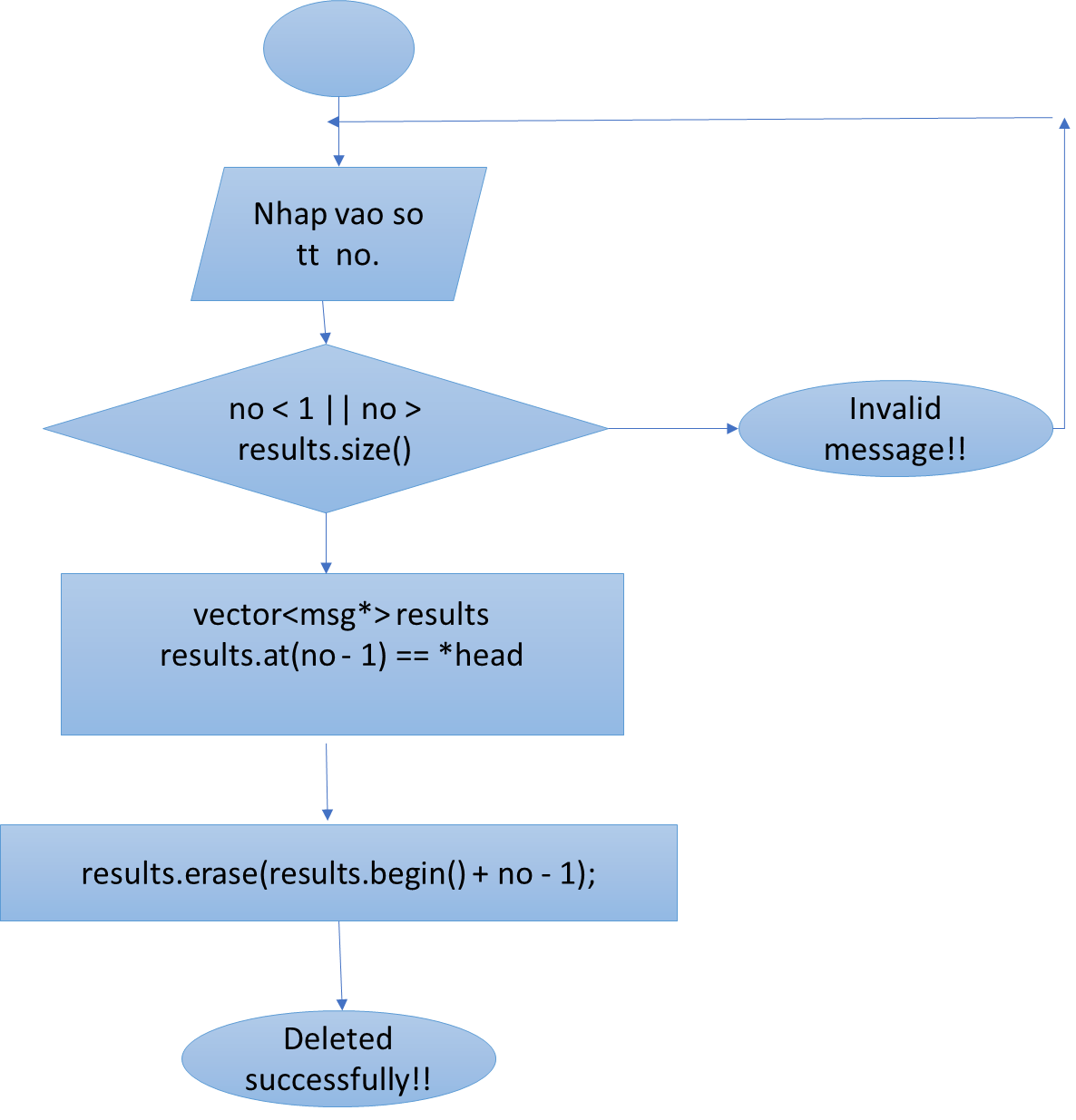
***Lưu đồ lựa chọn tin nhắn xem , xóa , ghim, bỏ ghim tin nhắn lưu trữ bằng ve***



***Lưu đồ ghim /bỏ ghim tin nhắn bằng số thứ tự “No. “ lưu trũ từ vecto tin nhắn***



***Lưu đồ đọc tin nhắn bằng nhập số thứ tự “no..***



***Lưu đồ xóa tin nhắn bằng nhập số thứ tự “no. “***

## 2.CƠ SỞ DỮ LIỆU

### 2.1. Cài đặt các phương pháp và thuật toán trong chương trình

#### 2.1.1.Cài đặt sử dụng danh sách liên kết đơn và danh sách liên kết đôi trong chương trình

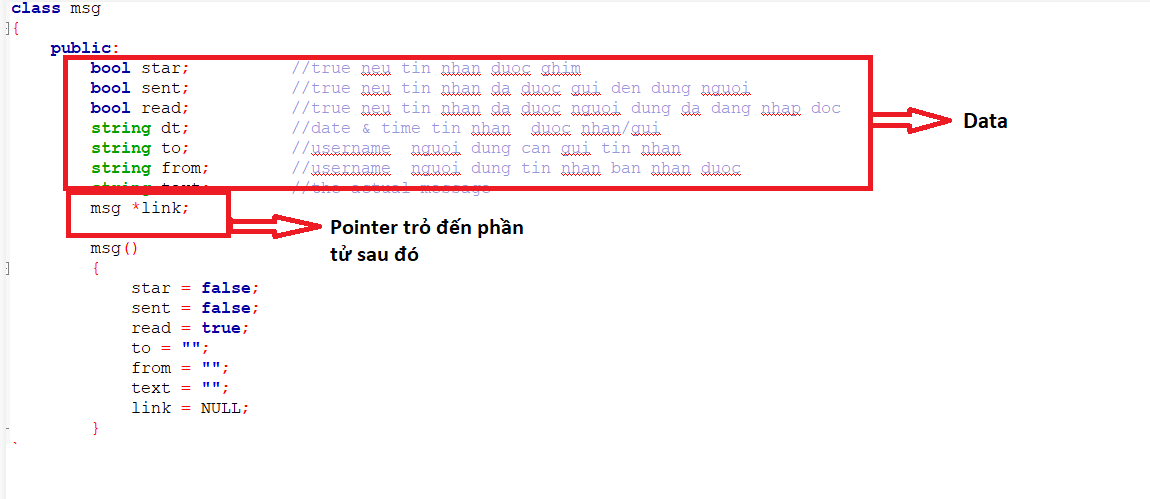
**Cài đặt danh sách liên kết đơn áp dụng trong dự án**

Danh sách liên kết đơn được ứng dụng vào trong chương trình của chúng em nhằm áp dụng để gửi và nhận tin nhắn cho mỗi tài khoản.

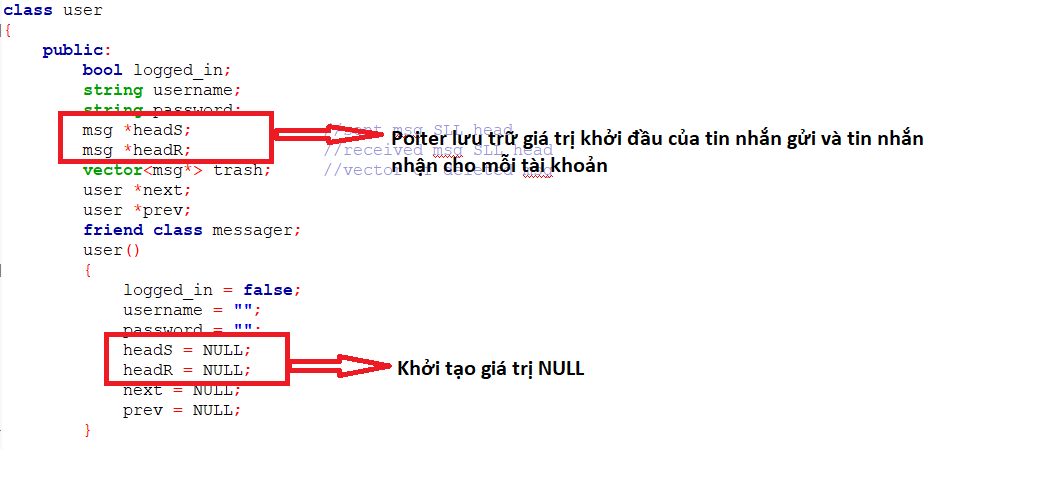
Chúng em sử dụng 2 danh sách liên kết đơn (Two Singly Linked List) cho mỗi tài, một dung để gửi và một dung để nhận các tin nhắn.

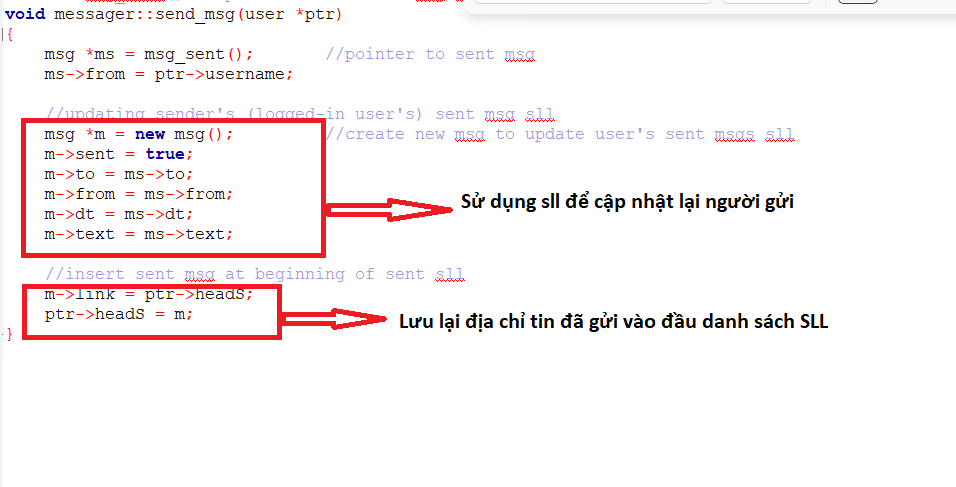
Đầu tiên, chúng ta đi vào phần khởi tạo các SLLs:

**Tạo Node**

****

*Node singly linked list dùng để lưu địa chỉ gửi, nhận tin nhắn*

****

******

*Hàm áp dụng SLL*

**Cài đặt danh sách liên kết đôi áp dụng trong dự án**

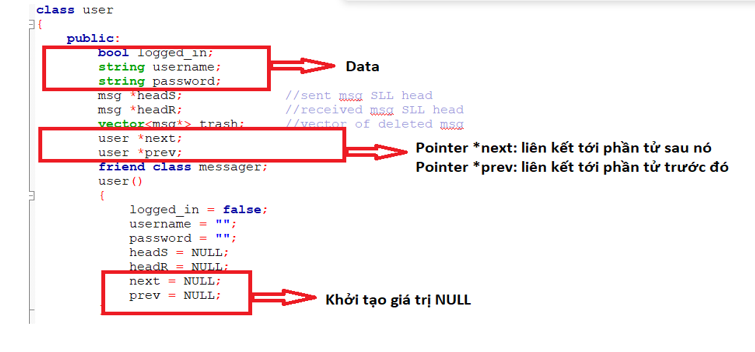
* Tương tự như danh sách liên kết đơn, các thư viện được dùng để cài đặt danh sách liên kết đôi vẫn là 2 thư viện:



Ở chương trình của chúng em, chúng em khai báo các thư viện iostream và iomanip để khai báo và sử dụng được double linked list.

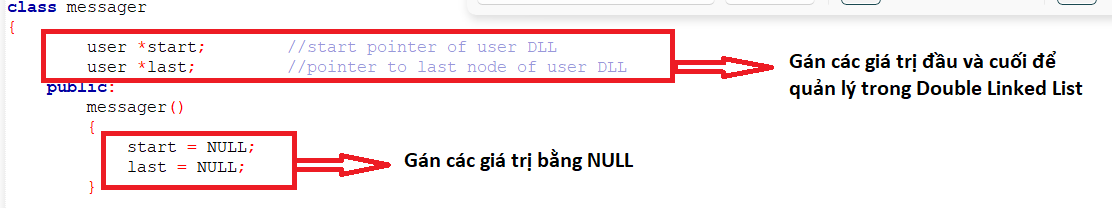
**Khai báo & khởi tạo**

- Khác một chút với dslk đơn, ngoài con trỏ next, chúng ta sẽ có thêm con trỏ prev để liên kết với Node trước nó. Chúng ta vẫn sẽ để Data như dưới hình dùng để lưu trữ thông tin của tài khoản. Ngay sau khai báo, chúng ta sẽ gán giá trị khởi tạo cho các pointer là NULL bởi vì ban đầu nó chưa trỏ đến giá trị nào của danh sách liên kết đôi.

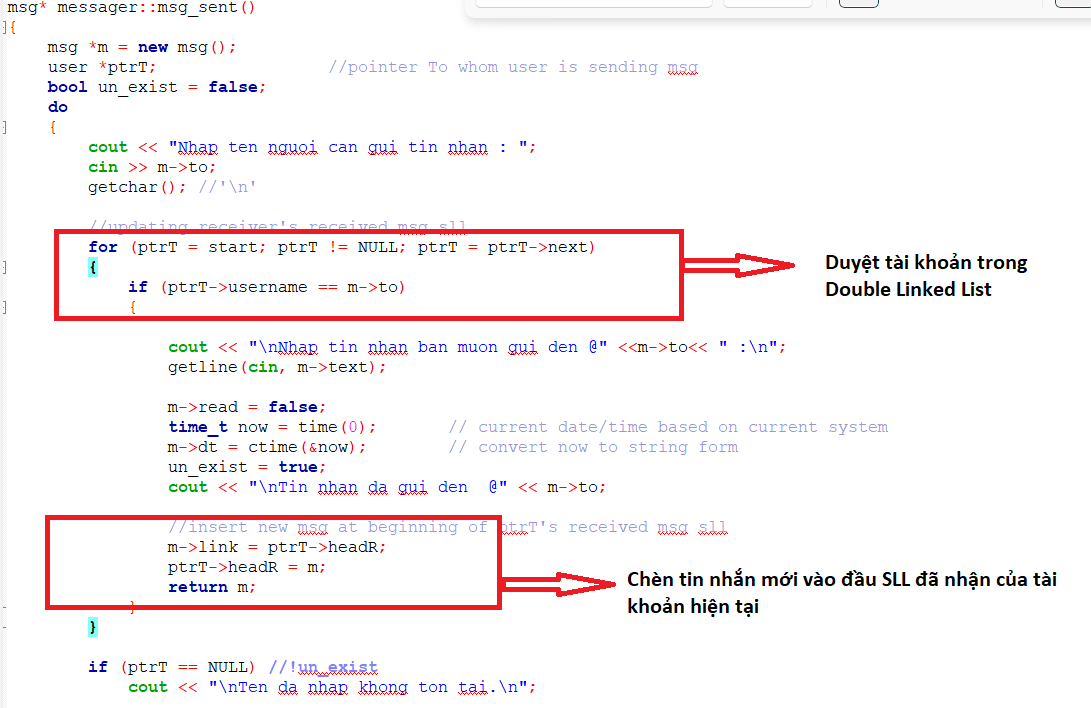


*Khởi tạo node cho double link list để sử dụng lưu trữ tài khoản*

Khởi tạo danh sách liên kết đôi (doule linked list) quản lý bằng giá trị khởi tạo đầu và cuối:

****

Danh sách liên kết đôi ban đầu khi được khai báo để lưu trữ tài khoản sẽ hoàn toàn không có phần tử nào. Vì thế các thành phần **user \*start** và **user \*last** của danh sách được coi là rỗng (NULL). Vì vậy, hàm khởi tạo là thao tác gán giá trị con trỏ quản lý địa chỉ đầu (**user \*start**) và cuối (**user \*last**) của danh sách về con trỏ NULL.

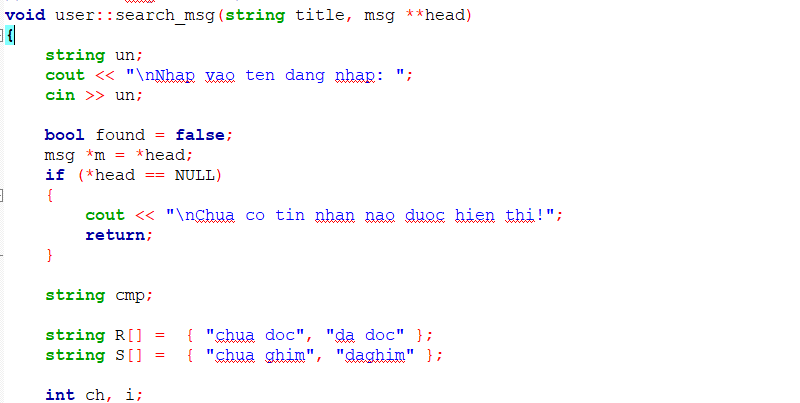


*Một số hàm sử dụng Double Linked List*

#### 2.1.2 Ý tưởng giải thuật tìm kiếm tuần tự được áp dụng trong chương trình:

Ý tưởng thuật toán tìm kiếm tuần tự được đưa ra nhằm mục đích tạo ra một hàm để tìm kiếm và kiệt kê ra các tin nhắn đã gửi hay đã nhận của mỗi tài khoản, nó còn giúp chúng ta chỉ ra các tin nhắn được đánh dấu (star) là quan trọng.

Đầu tiên, ta định nghĩa một hàm search\_msg thuộc class user, hàm này sử dụng thuật toán tìm kiếm tuần tự , khi được gọi ra nó cho phép chúng ta tìm kiếm tin nhắn đã gửi, đã nhận hay đã ghim từ user.



***Khai báo thuộc tính cần thiết trước khi bắt đầu giải thuật tìm* kiếm**



*Thuật toán tìm kiếm tuần tự*

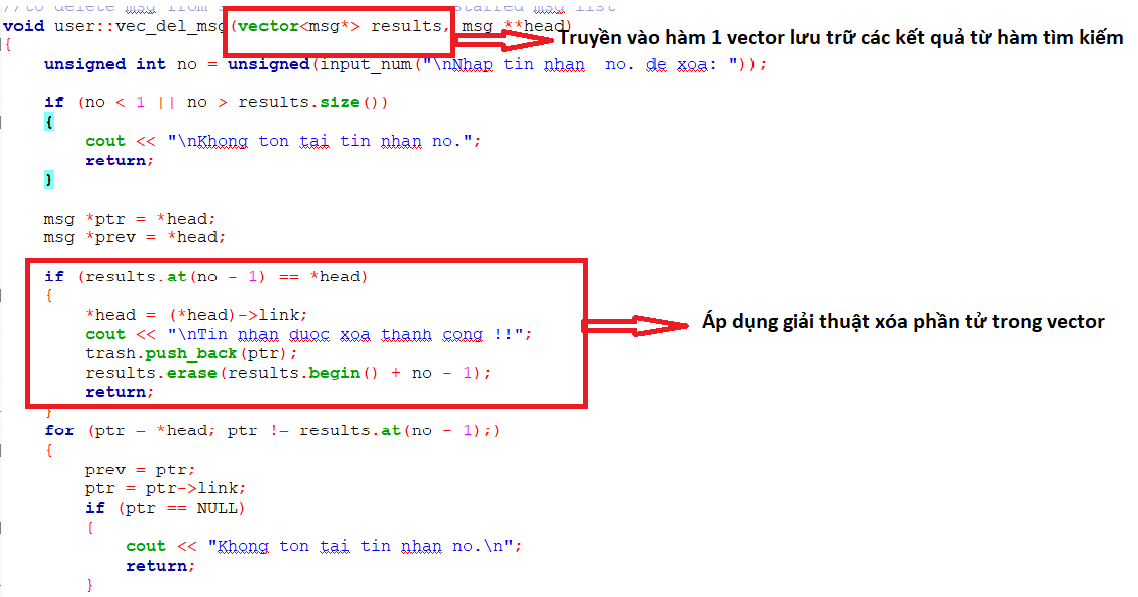
#### 2.1.3 Áp dụng cấu trúc vector cho việc xây dựng đồ án

Ở chương trình của chúng em, chúng em áp dụng Vector (Cấu trúc Vector) để gắn hay bỏ dấu sao ( tin nhắn đã ghim) của mỗi tin nhắn. Ngoài ra vector còn dung để lưu trữ các tin nhắn bị xóa, để có thể xóa vĩnh viễn

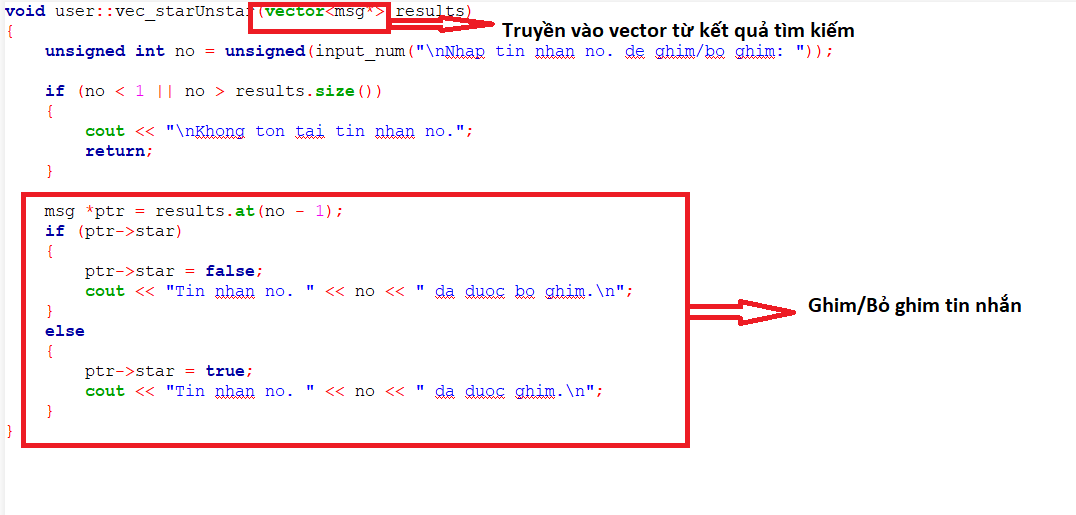
Đầu tiên, để khai báo và sử dụng vector ta khởi tạo thư viện vector trong ngôn ngữ C++:



Tiếp theo, ta khai báo và sử dụng vector cho nhưng hàm cần dung đến nó:

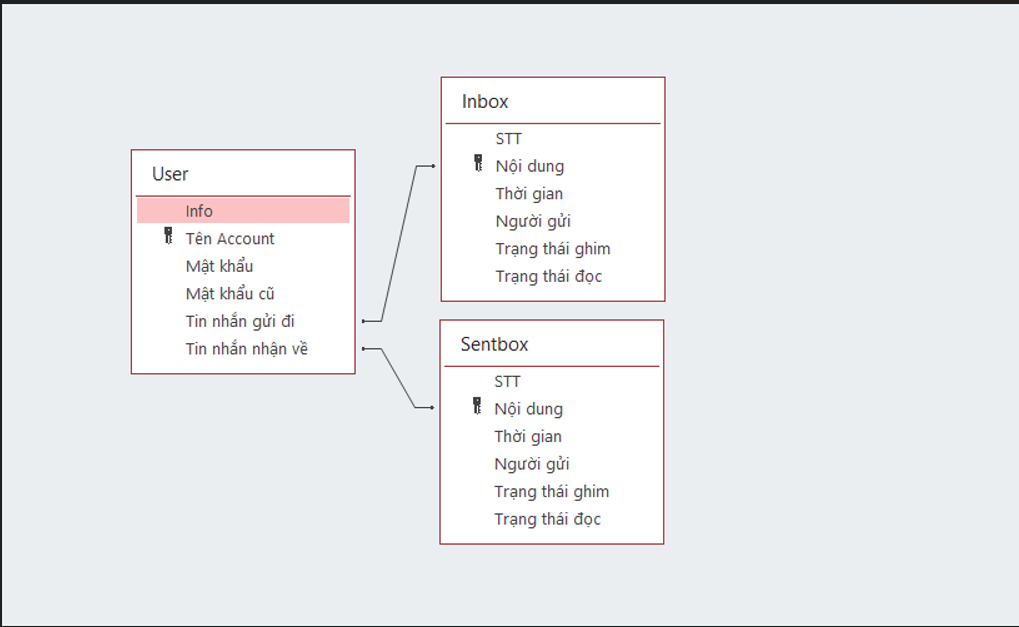


*Hàm sử dụng dữ liệu vector truyền vào để xóa tin nhắn khỏi kết quả tìm kiếm hoặc khỏi danh sách tin nhắn có gắn dấu sao*

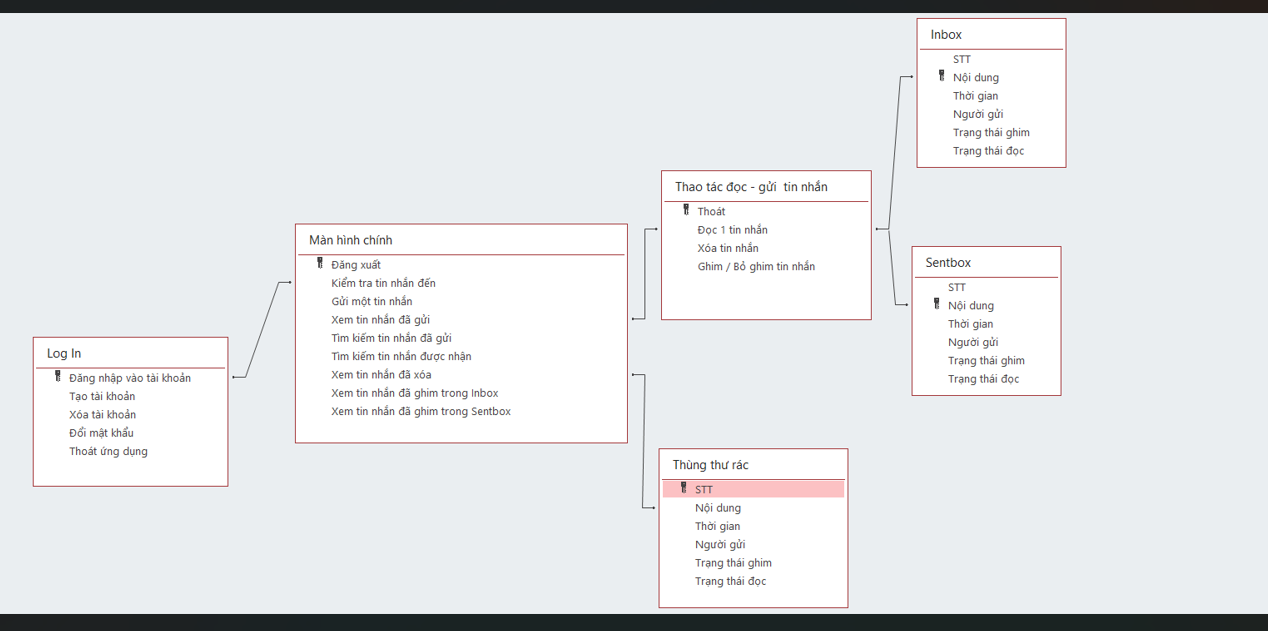


*Một ứng của khác của vector vào hàm ghim hoặc bỏ ghim một tin nhắn*

### 1.2. Thiết kế cơ sở dữ liệu:



Mô hình quan hệ giữa các menu của chương trình :

*Mô hình quan hệ*

**Chương trình gồm có 3 class chính:**

* **Class msg:** nhiệm vụ của class msg là đánh dấu các trạng thái của tin nhắn( sent, read, star) lưu nội dung và tất cả các thông tin mà tin nhắn đó có ( người gửi, người nhận, giờ gửi…)

Text

Description automatically generated

|  |  |
| --- | --- |
| msg \*link sử dụng msg \*link để khai báo singly linked list, trỏ đến tin nhắn tiếp theo |  |
|  |  |
| **bool star;** (True nếu có gắn dấu\*) |  |
| **bool sent;** (True nếu đã gửi ) |  |
| **bool read;** True nếu người nhận đã đọc) |  |
| **string text;** (để lưu nội dung tin nhắn) |  |
| **string to;** (lưu người nhận là ai) |  |
| **string from;** ( lưu người gửi là ai) |  |
| **string dt;** (lưu thời gian gửi) |  |
| Tất cả dữ liệu sẽ được khởi tạo ở **msg()** để khi vừa gọi class, sẽ gọi **msg()** 1 cách tự động |  |

* **Class User**: nhiệm vụ của class User là lưu thông tin **( Password, Username**) của User, Trạng thái logged\_in, lưu các User trước và sau (doubly linked list) .Và là hàm bạn( friend class) của class messager tiếp theo. Ngoài ra, hàm class user sẽ có các hàm con để làm các tiện ích của chương trình.

Text

Description automatically generated

Text

Description automatically generated

|  |  |
| --- | --- |
| logged\_in = True nếu người dùng đã đăng nhập vào tại khoản |  |
| Lưu username của người dùng |  |
| Lưu password của người dùng |  |
| Con trỏ headS với kiểu dữ liệu msg để quản lý danh sách các tin nhắn được gửi đi (danh sách theo kiểu singly linked list) |  |
| Con trỏ headR với kiểu dữ liệu msg để quản lý danh sách các tin nhắn được nhận (danh sách theo kiểu singly linked list) |  |
| Vector với kiểu dữ liệu msg\* để thêm các tin nhắn (kiểu msg) gồm tất cả các dữ liệu đi kèm với 1 tin nhắn. Thêm vào vector để thành 1 thùng rác, cần thì xem lại, cần thì xóa. |  |
| User \*next và \*prev để trỏ tới các phần tử thuộc kiểu class user trước và sau (doubly linked list) |  |
| Gán hàm bạn user và messager để 2 class có thể truy cập các phần tử của nhau mà không bị lỗi |  |
| Hàm user sẽ được gọi tự động mỗi khi có 1 phần tử user được khai báo. Hàm này nhằm khởi tạo các thuộc tính của user |  |

Ngoài ra, ở class user còn có các chương trình con để thực hiện các tiện ích của chương trình.

Ở phần này là về nhiệm vụ của từng hàm con. Việc đưa các dữ liệu từ vào từ hàm con, cách xử lý của hàm con sẽ nói chi tiết ở phần sau.

|  |  |
| --- | --- |
|  | Chương trình trình bày các tin nhắn. Dữ liệu vào sẽ là  Title: nhiệm vụ khi display  \*head: để trỏ đến phần tử đầu của linked list. |
|  | ở đây con trỏ \*\*head mục đích là để trỏ vào con trỏ head. Nhằm sửa nội dung của các phần tử tin nhắn. Và dữ liệu sửa sẽ được lưu tại ( Biến toàn cục)  nếu chỉ input là \*head thì dữ liệu sẽ được lưu biến cục bộ. |
|  | Đọc tin nhắn,ở chương trình này sử dụng tìm kiến tuần tự(sequence search). Input vào con trỏ đầu danh sách liên kết. |
|  | Như đã giải thích ở trên về con trỏ \*\*head.  Việc cần làm ở hàm này là xóa cả phần tử nên cần input 1 biến toàn cục |
|  | Hàm đánh dấu sao tin nhắn |
|  | Sử dụng dữ liệu vector để truy cập đến tin nhắn nhanh hơn. |
|  |  |
|  |  |

* **Class messager**

Text

Description automatically generated

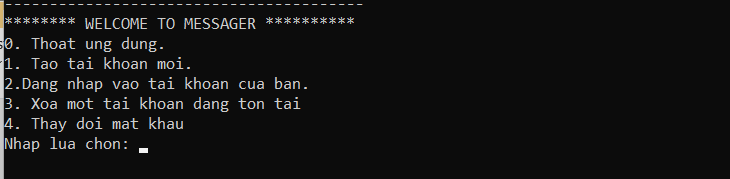
|  |  |
| --- | --- |
|  | 2 biến này để quản lí start và last của doubly linked list user, nếu ko có user thì last==start== NULL |
|  | Return true khi không có user else return false |
|  | Hàm để tạo tài khoản.  CÓ LƯU ĐỒ |
|  | Kiểm tra xem các user xem có hay không để gán start và last |
|  | Hàm để đăng nhập |
|  | Xóa tài khoản, sử dụng cách xóa của douly linked list.  CÓ LƯU ĐỒ |
|  | Đổi mật khẩu |
|  | Hiển thị các hoạt động của chương trình cho người dùng chọn. Sau đó sẽ gọi các chương trình con theo yêu cầu của người dùng. |
|  | Được gọi sau  void send\_msg(user \*ptr);  được gọi. để chương trình minh bạch hơn |
|  | Cấp phát bộ nhớ động để tạo tin nhắn. Và gọi chương trình msg\*\_sent() Để tạo tin nhắn gửi đến |

## 3.KẾT QUẢ

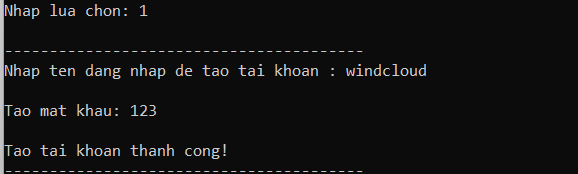
### 3.1 Hướng dẫn sử dụng

**+ Tạo tài khoản để bắt đầu sử dụng ứng dụng**

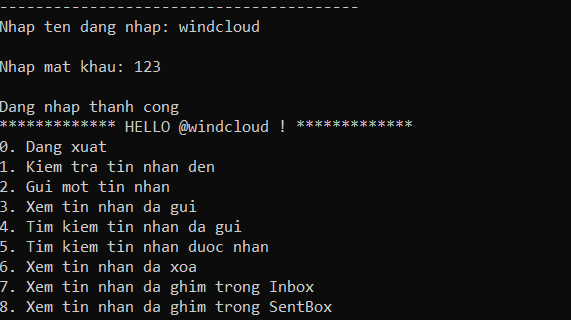
**Bước 1: Nhập vào sự lựa chọn của bạn:**

****

**Bước 2: Chọn 1 – Tạo 1 tài khoản mới: Nhập tên đăng nhập và mật khẩu để login 1 tài khoản mới**

****

**Bước 3: Chọn 2 để đăng nhập vào tài khoản của bạn**

****

Sau khi đăng nhập tạo tài khoản người dùng có thể thực hiện các thao tác với các chức năng của chương trình

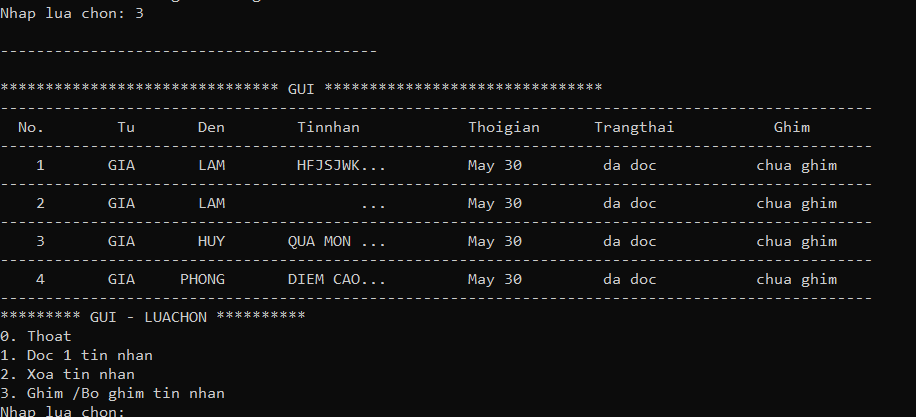
### 3.2 Kết quả đề tài

**-Thao tác gửi tin nhắn**

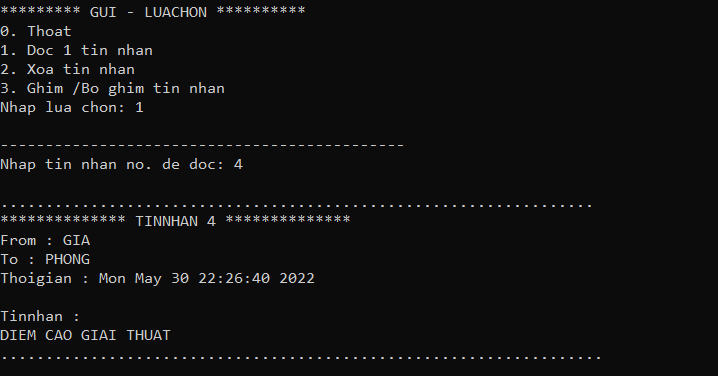
**Kết quả gửi tin nhắn sẽ bao gồm các thông tin:**



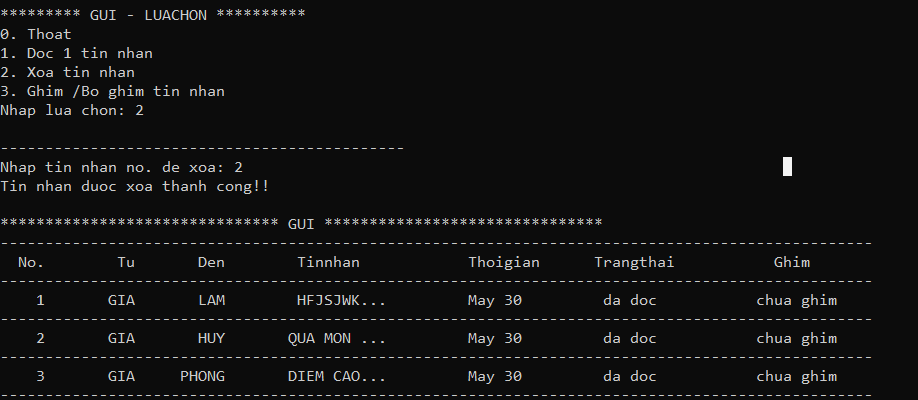
**-Thao tác xem tin nhắn : nhập lựa chọn 3**



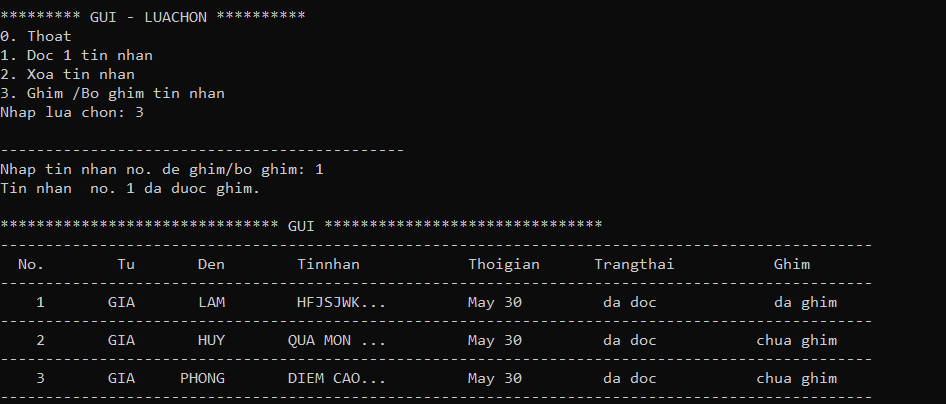
**+ Để xem chi tiết tin nhắn thì người dùng cần nhập no. của tin nhắn**



**+ Xóa tin nhắn nhất định thì nhập no. của tin nhắn**

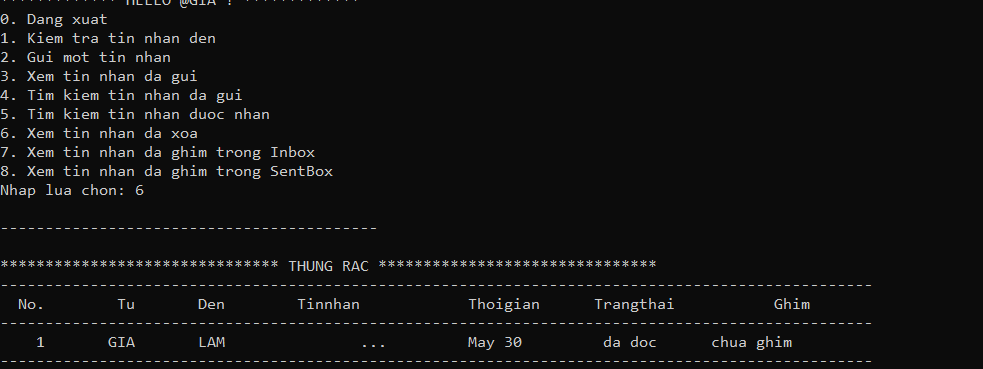


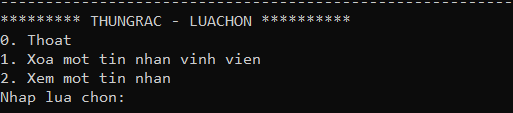
**+ Ghim / bỏ ghim tin nhắn :**



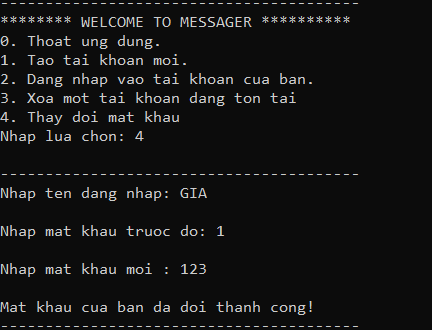
**-Thao tác xem tin nhắn đã xóa:**

**+ Người dùng có thể xem tin nhắn và có thể xóa hoàn toàn tin nhắn**

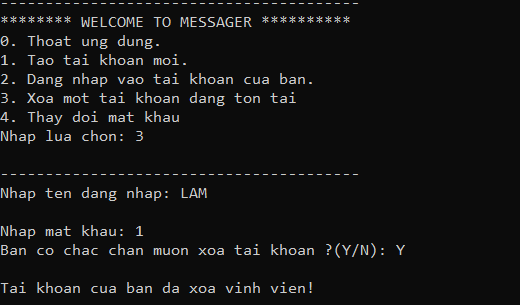
****



**-Thao tác đổi mật khẩu tài khoản**



**-Thao tác xóa tài khoản tồn tại**



# PHẦN 4: KẾT LUẬN

## 1.KẾT LUẬN

* **Về cơ bản chương trình của nhóm chúng em đã hoàn thành được các mục tiêu đã đề ra trước đó và có thêm nhiều tính năng như đổi mật khẩu, xóa tìa khoản ….. chúng em đã sử dụng và áp dụng các phương pháp đã học vào trong chương trình .**
* **Về ưu điểm đề tài:**
* **Đề tài có thao tác dễ dàng , trực quan , dễ tiếp cận cho người mới sử dụng**
* **Chương trình chạy ổn định cho ra kết quả chính xác về nội dung , thời gian, thông tin người gửi / nhận tin nhắn .**
* **Dễ chỉnh sửa bởi các chương trình, hàm con được sắp xếp theo logic, thứ tự , dễ dàng debug và tìm kiếm sửa lỗi.**
* **Về nhược điểm:**
* **Giao diện cần được tối ưu hóa và thiết kế thẩm mỹ hơn**
* **Cần cả thiện một số thuật toán để chương trình được tối ưu, giúp cho chương trình xử lý nhanh hơn.**

## 2.HƯỚNG PHÁT TRIỂN VÀ MỞ RỘNG

* **Chương trình có thể được thêm nhiều chức năng mới như : gửi tệp tin , thư mục**
* **Thêm tính năng in danh sách thông tin (profile) người nhận gửi thư**
* **Nâng cấp thêm giao diện để dễ nhìn hơn**
* **Thêm chế độ người quản trị quản lý các người dùng hệ thống**
* **Thêm chức năng tìm kiếm người dùng đã đăng nhập**

**TÀI LIỆU THAM KHẢO**

Giáo trình Cấu trúc dữ liệu và giải thuật/ Lê Văn Vinh. -- Tp. Hồ Chí Minh: Đại học Quốc gia Tp. Hồ Chí Minh, 2013 113tr.

<https://www.geeksforgeeks.org/linked-list-set-2-inserting-a-node/>

<https://www.geeksforgeeks.org/linked-list-set-1-introduction/>

<https://www.geeksforgeeks.org/doubly-linked-list/>

<https://www.youtube.com/watch?v=mDt53JLj8sM>

<https://viblo.asia/p/c-mfc-dynamic-link-libraries-dll-924lJmX8ZPM>

<https://tuhocict.com/cac-thuat-toan-tim-kiem-co-ban-tim-kiem-tuan-tu-tim-kiem-nhi-phan/>

<https://viblo.asia/p/bai-toan-tim-kiem-va-cac-phuong-phap-giai-thong-dung-djeZ1dEGKWz>

<https://nguyenvanhieu.vn/danh-sach-lien-ket-doi/#21-khai-bao-khoi-tao>

<https://viblo.asia/p/arraylist-vs-linkedlist-vs-vector-3Q75wkb35Wb>

[https://duongdinh24.com/vector-trong-c/#:~:](https://duongdinh24.com/vector-trong-c/#:~:text=Vector%20C%2B%2B%20l%C3%A0%20g%C3%AC%3F,m%E1%BA%B7t%20h%E1%BA%A1n%20ch%E1%BA%BF%20c%E1%BB%A7a%20m%E1%BA%A3ng)