

# HỆ THỐNG THÔNG BÁO THÔNG TIN

Báo cáo đồ án nhập môn ngành

**Người hướng dẫn**

*Cô Lê Thị Mỹ Hạnh*

**Người thực hiện**

*Trần Hưng Quốc Khải*

*Nguyễn Kim Huy*

*Trần Ngọc Huy*

*Nguyễn Thị Ngọc Huyền*

# BroadcastNow

Trường đại học Bách Khoa Đà Nẵng  
Lớp 18TCLC\_DT3

# MỤC LỤC

MỤC LỤC.....	1
MỞ ĐẦU.....	5
1. Giới thiệu .....	5
2. Vấn đề thực tế .....	6
3. Vấn đề bao quát.....	6
4. Cách giải quyết.....	6
4.1. Mô hình Notitag .....	6
4.2. Ý tưởng.....	6
CHƯƠNG: 1 CƠ SỞ LÝ THUYẾT .....	7
1. Các vấn đề về công nghệ.....	7
2. Các cơ sở công nghệ có thể giải quyết vấn đề .....	7
2.1. Messenger Platform .....	7
2.2. Node.js.....	17
2.3. MongoDB .....	17
CHƯƠNG: 2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....	18
1. Thông báo cho nhóm đối tượng bằng Messenger với mô hình Notitag .....	18
1.1. Ví dụ thực tế: .....	18
1.2. Mô tả .....	18
2. Gửi thông báo từ trang sv.dut.udn.vn về Messenger .....	20
CHƯƠNG: 3 TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ .....	22
1. Các tính năng của hệ thống .....	22
2. Cấu trúc dữ liệu .....	22
3. Mô tả thuật toán theo từng chức năng, tính năng.....	23
3.1. Tính năng của webhook.....	23
3.2. Hàm kiểm tra người dùng có tồn tại hay không.....	24
3.3. Hàm phân tích cú pháp .....	24
3.4. Chức năng create #<tên thẻ> .....	24
3.5. Chức năng follow #<tên thẻ> .....	24

3.6.	Chức năng unfollow #<tên thẻ> .....	24
3.7.	Chức năng send #<tên thẻ> <nội dung> .....	24
3.8.	Chức năng delete #<tên thẻ> .....	24
3.9.	Chức năng Mycreate.....	24
3.10.	Chức năng Myfollow.....	25
3.11.	Chức năng help.....	25
4.	Trình bày mô hình thử nghiệm.....	25
4.1.	Giao diện mở đầu của BroadcastNow.....	25
4.2.	Chức năng khởi tạo thẻ.....	25
4.3.	Chức năng đăng ký nhận tin từ thẻ .....	27
4.4.	Chức năng hủy nhận tin từ thẻ.....	29
4.5.	Chức năng gửi nội dung vào thẻ .....	31
4.6.	Chức năng xóa thẻ.....	33
4.7.	Chức năng xem lại các thẻ đã tạo .....	36
4.8.	Chức năng xem lại các thẻ đã đăng ký nhận tin.....	36
4.9.	Chức năng xem các cú pháp .....	37
4.10.	Ví dụ hoạt động.....	37
4.11.	Gửi thông báo từ trang sv.dut.udn.vn về Messenger .....	38
5.	Đánh giá kết quả.....	40
5.1.	Đối tượng tiếp cận lớn .....	40
5.2.	Tính thuận tiện đến từ sv.dut.udn.vn .....	40
5.3.	Tính cập nhật.....	40
5.4.	Vấn đề về tính ổn định của Messenger: .....	41
5.5.	Việc chuyển đổi từ Email sang Messenger:.....	41
5.6.	Vấn đề về số lượng thẻ đăng kí.....	41
5.7.	Khả năng trở nên phổ biến.....	41
6.	Kết luận và hướng phát triển .....	41
6.1.	Kết luận .....	41
6.2.	Hướng Phát triển.....	42
CHƯƠNG: 4	KẾT LUẬN CHUNG .....	44

TÀI LIỆU THAM KHẢO.....	45
PHỤ LỤC.....	46

### **Danh sách hình vẽ**

Hình 1.....	12
Hình 2.....	13
Hình 3.....	14
Hình 4.....	15
Hình 5.....	16
Hình 6.....	16
Hình 7.....	25
Hình 8.....	26
Hình 9.....	26
Hình 10.....	27
Hình 11.....	27
Hình 12.....	28
Hình 13.....	28
Hình 14.....	29
Hình 15.....	29
Hình 16.....	30
Hình 17.....	30
Hình 18.....	31
Hình 19.....	31
Hình 20.....	32
Hình 21.....	32
Hình 22.....	33
Hình 23.....	33
Hình 24.....	34
Hình 25.....	34
Hình 26.....	35
Hình 27.....	35
Hình 28.....	36
Hình 29.....	36
Hình 30.....	37
Hình 31.....	37
Hình 32.....	38

Hình 33.....	38
Hình 34.....	39
Hình 35.....	39

### **Danh sách bảng biểu**

Bảng 1 các cú pháp nhấn tin.....	20
----------------------------------	----

### **Danh sách từ viết tắt**

## **MỞ ĐẦU**

### **1. Giới thiệu**

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của chính phủ, tổ chức, cũng như các doanh nghiệp, nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.

Cùng với sự phát triển không ngừng về kỹ thuật máy tính và mạng điện tử, ngành công nghệ thông tin liên tục phát triển, đáp ứng nhu cầu đời sống, từng bước đạt đến những đỉnh cao mới. Mạng Internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên không thể thiếu, là nền tảng cho sự truyền tải, trao đổi thông tin trên toàn cầu.

Giờ đây, mọi việc liên quan đến thông tin trở nên dễ dàng với người sử dụng. Bằng internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Một ví dụ điển hình là sự khai sinh và phát triển của hệ thống thông tin trong trường Đại học, Cao đẳng. Trong hoạt động dạy, học và nghiên cứu, hệ thống thông tin khẳng định được vai trò thiết yếu, có sức ảnh hưởng lớn. Trong quá trình sử dụng hệ thống thông tin cho việc học, nhóm chúng em nhận thấy có thể cải tiến một số hoạt động nên đã cùng nhau thực hiện đề án: “Cải tiến hệ thống thông tin trong trường Đại học” với mục tiêu giúp sinh viên và giảng viên dễ dàng hơn trong việc sử dụng hệ thống.

Với sự hướng dẫn tận tình của Cô Lê Thị Mỹ Hạnh, chúng em đã hoàn thành bài báo cáo đề án này. Tuy đã gắng sức phân tích, thiết kế và chỉnh sửa, chắc chắn rằng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và góp ý của quý Thầy cô. Chúng em xin chân thành cảm ơn.

## **2. Vấn đề thực tế**

- Mỗi lần phòng công tác sinh viên có thông báo thì phải đăng tải lên trang web (sv.dut.udn.vn) sau đó sinh viên phải chủ động truy cập để biết thông tin. Nhưng đa số các bạn rất ít truy cập trang web.
- Giảng viên muốn gửi thông báo qua mail cho lớp học phần thì phải thu thập tất cả mail và nhập vào máy tính. Một số smartphone của sinh viên không cài sẵn gmail. Và việc đọc hay kiểm tra hộp thư gmail khá là kèn càng. Một số sinh viên sử dụng gmail không khoa học làm cho gmail nhiều spam và thông tin quan trọng hay bị trôi đi.
- Lớp trưởng muốn thông báo đến lớp thì phải gửi vào “nhóm chat” trong messenger, hoặc “nhóm” trên facebook. Gây ra vấn đề tin nhắn hoặc bài đăng có thể bị trôi, và một số lý do khiến sinh viên bị sót thông tin.

## **3. Vấn đề bao quát**

- Xây dựng một hệ thống hỗ trợ cho việc thông báo đến một nhóm đối tượng trở nên dễ dàng và tối ưu hơn.
- Gửi thông tin mới từ trang sv.dut.udn.vn về nền tảng phổ biến hơn.

## **4. Cách giải quyết**

### **4.1. Mô hình Notitag**

Tương tự như mô hình follow trang cá nhân từ mạng xã hội. Nhưng đối với Notitag, một người có thể tạo nhiều #tag và mọi người muốn nhận thông báo nào thì follow vào #tag đó.

Ví dụ: Giảng viên dạy cho nhiều lớp, mỗi lớp giảng viên sẽ tạo một #tag riêng. Sinh viên chỉ cần đăng ký nhận #tag phù hợp sẽ nhận được thông báo mà không bị làm phiền bởi các thông báo không liên quan.

### **4.2. Ý tưởng**

- Thông báo cho nhóm đối tượng bằng messenger với mô Notitag.
- Khi có thông báo mới từ trang sv.dut.udn.vn, gửi chúng về messenger.

## **CHƯƠNG: 1      CƠ SỞ LÝ THUYẾT**

### **1.    Các vấn đề về công nghệ**

- Một nền tảng cho phép chúng ta có thể giao tiếp để gửi tin nhắn Messenger cho người dùng
- Một hệ thống xử lý các tin nhắn của người dùng và lấy dữ liệu từ trang sv.dut.udn.vn
- Một hệ quản trị cơ sở dữ liệu

### **2.    Các cơ sở công nghệ có thể giải quyết vấn đề**

#### **2.1.    Messenger Platform**

##### **2.1.1.    Các thành phần tích hợp**

Nền tảng Messenger cung cấp tập hợp các API cốt lõi, plugin trên web và một chế độ xem web đầy đủ cung cấp cho bạn những gì bạn cần để tạo trải nghiệm tuyệt vời trong Messenger.

##### **2.1.2.    Thành phần nền tảng có sẵn**

- ID người dùng trong trang (PSID)
- API
- API Gửi
- API Tải lên file đính kèm
- API Khớp ID
- API Giao thức chuyển mạng
- API Mã Messenger
- API Thông tin chi tiết về nhắn tin
- API Khớp khách hàng
- API Trang cá nhân Messenger
- Webhook
- Chế độ xem web của Messenger
- JS SDK tiện ích Messenger
- Tiện ích chat
- Xử lý ngôn ngữ tự nhiên tích hợp
- Plugin

##### **2.1.2.1.    ID người dùng trong trang (PSID)**

Mỗi lần một người bắt đầu cuộc trò chuyện với bot Messenger của bạn, Nền tảng Messenger sẽ gán cho họ một ID người dùng trong trang, thường được gọi là PSID. Đây là ID duy nhất dành cho Trang Facebook của bạn và không thể dùng ID đó với bất kỳ Trang Facebook nào khác. Nói cách khác, một người sẽ có PSID khác nhau đối với mỗi bot Messenger mà họ trò chuyện. Điều này đảm bảo rằng chỉ những bot mà người dùng chọn bắt đầu cuộc trò chuyện mới có thể gửi tin nhắn cho họ.



Bất cứ khi nào gửi tin nhắn cho một người, bạn sẽ thêm PSID của họ vào thuộc tính recipient.id của yêu cầu để xác định ai sẽ nhận tin nhắn. Tương tự, bất cứ khi nào một người thực hiện hành động trong cuộc trò chuyện, chẳng hạn như nhấn vào một nút hoặc gửi một tin nhắn, PSID của họ sẽ được thêm vào thuộc tính sender.id của sự kiện webhook để bot của bạn có thể xác định ai đã thực hiện hành động đó.

#### **2.1.2.2. API**

Nền tảng Messenger cung cấp một loạt REST API nhằm đem đến cho bạn những công cụ bạn cần để tạo ra trải nghiệm Messenger tuyệt vời. Từ gửi tin nhắn đa phương tiện, tìm khách hàng hiện có trên Messenger, đến tùy chỉnh bot của bạn và nhiều hoạt động khác, API của chúng tôi là cách chính để bạn làm việc với Nền tảng Messenger.

#### **2.1.2.3. API Gửi**

API Gửi là điểm tích hợp chính của bạn với Nền tảng Messenger. API này cho phép bạn gửi văn bản đơn giản và mẫu tin nhắn có cấu trúc, cũng như các tài sản như hình ảnh, video, âm thanh và file. API Gửi cũng hỗ trợ các loại tin nhắn và tính năng bổ sung nhằm đem lại cho bạn sự linh hoạt trong việc xây dựng trải nghiệm Messenger, bao gồm cả trả lời nhanh và hành động của người gửi.

#### **2.1.2.4. API Tải lên tệp đính kèm**

API Tải lên tệp đính kèm cho phép bạn tải các nội dung lên Nền tảng Messenger. Bạn có thể tải lên bất kỳ loại nội dung nào từ URL hoặc hệ thống tệp trên máy, bao gồm cả hình ảnh, âm thanh, video và tệp.

Với API này, bạn có thể sử dụng lại các nội dung của mình, từ đó giúp loại bỏ chi phí tải cùng một nội dung lên nhiều lần mỗi khi gửi. Điều này giúp gửi tin nhắn nhanh hơn, cường độ băng thông thấp hơn và hiệu quả hơn. Sau khi tải tài sản lên, API Tải lên file đính kèm sẽ trả về một attachment\_id có thể dùng cho các yêu cầu trong tương lai.

#### **2.1.2.5. API Khớp ID**

Thông thường, một người sẽ tương tác với nhiều Trang và ứng dụng Facebook mà bạn sở hữu. Tuy nhiên, điều này có thể gây ra vấn đề vì họ sẽ có ID khác nhau đối với mỗi Trang và ứng dụng mà họ tương tác. Để giúp bạn có khả năng tạo ra trải nghiệm nhất quán, API Khớp ID cho phép bạn truy xuất ID người dùng trong trang và ID người dùng trong ứng dụng của một người trên tất cả các ứng dụng và Trang của bạn, sử dụng PSID hoặc ASID duy nhất của người đó.

#### **2.1.2.6. API Giao thức chuyển mạng**

Giao thức chuyển mạng cho phép bạn chuyển quyền kiểm soát cuộc trò chuyện giữa nhiều ứng dụng Facebook, cũng như hộp thư trên Trang. Điều này giúp bạn có khả năng thành phần hóa cách thức xây dựng trải nghiệm Messenger, bằng cách cho phép bạn xây dựng nhiều bot chuyên biệt để xử lý các khía cạnh riêng của cuộc trò chuyện, thay vì một ứng dụng khổng lồ (có thể nhanh chóng trở nên phức tạp và khó quản lý). Điều này cũng giúp bạn dễ dàng sử dụng lại các ứng dụng để hỗ trợ cuộc trò chuyện trên nhiều Trang. Ngoài ra, khả năng chuyển quyền kiểm soát sang hộp thư trên Trang sẽ cho phép bạn chuyển cuộc trò chuyện sang trò chuyện trực tiếp bất cứ lúc nào.

#### **2.1.2.7. API Mã Messenger**

Mã Messenger là một cách tuyệt vời để quảng cáo bot Messenger của bạn, đồng thời cũng là cách dễ dàng để mọi người bắt đầu cuộc trò chuyện mới với bạn. Ngoài ra, giao diện của mã này cũng khá đẹp!

Mỗi Trang Facebook đi kèm với một Mã Messenger có thể tạo theo chương trình qua API Mã Messenger. Ngoài ra, API này còn cho phép bạn tạo các mã Messenger thông số có chứa thuộc tính ref tùy chỉnh. Thuộc tính này sẽ được gửi tới webhook của bạn trong sự kiện webhook khi quét Mã. Mã này hữu ích cho nhiều mục đích, chẳng hạn như liên kết cuộc trò chuyện mới với chiến dịch quảng cáo và sự kiện.

#### **2.1.2.8. API Thông tin chi tiết về nhắn tin**

API Thông tin chi tiết nhắn tin cho phép bạn truy xuất theo chương trình các thông tin giống như hiển thị trong tab Thông tin chi tiết Trang của Trang Facebook. Thông tin này bao gồm các số liệu liên quan đến cuộc trò chuyện, chẳng hạn như số lượng cuộc trò chuyện, tình trạng phản hồi, tỷ lệ chặn, v.v. Các dữ liệu và công cụ phân tích khác, chẳng hạn như sự kiện tùy chỉnh cũng có sẵn nhờ có Phân tích trên Facebook.

#### **2.1.2.9. API Khớp khách hàng**

Khả năng là bạn đã có khách hàng, nhưng làm cách nào để tìm thấy họ trên Messenger? Với API Khớp khách hàng, bạn cung cấp một số điện thoại và tên (tùy chọn), Nền tảng Messenger sẽ trả về một người trùng khớp đang sử dụng Messenger. Nếu chúng tôi tìm thấy một người trùng khớp, bạn có thể sử dụng số điện thoại của người đó để bắt đầu cuộc trò chuyện mới với họ. Sau khi họ phản hồi, bạn sẽ nhận được một PSID mà bạn có thể dùng để tiếp tục cuộc trò chuyện.

#### **2.1.2.10. API Trang cá nhân Messenger**

API Trang cá nhân Messenger được dùng để đặt, cập nhật và xóa nhiều cài đặt cho bot Messenger của bạn, bao gồm cả cài đặt cho thanh toán, miền được đưa vào danh sách cho phép, menu ổn định, màn hình chào mừng, v.v.

#### **2.1.3. Gửi tin nhắn hàng loạt**

Theo mặc định, API Tin nhắn hàng loạt sẽ gửi tin nhắn của bạn tới tất cả các cuộc trò chuyện đang mở bằng bot Messenger của bạn. Để cho phép gửi tin nhắn hàng loạt tới một tập con các cuộc trò chuyện, API Tin nhắn hàng loạt hỗ trợ 'custom labels' có thể được liên kết với các PSID cá nhân.

#### **2.1.4. Webhook**

Nền tảng Messenger sẽ gửi sự kiện tới webhook bất cứ khi nào diễn ra một hành động trong cuộc trò chuyện với bot của bạn. Webhook là một điểm cuối HTTPS đơn lẻ (thường là /webhook) mà bạn hiển thị để chấp nhận yêu cầu POST. Đây là nơi bot của bạn xử lý và phản hồi tất cả sự kiện webhook gửi đến.

Nền tảng Messenger hỗ trợ một loạt các sự kiện webhook tiêu chuẩn mà bạn có thể đăng ký webhook trong quy trình thiết lập. Tối thiểu, bạn phải đăng ký sự kiện webhook messages và messaging\_postbacks để có thể triển khai các tính năng Nền tảng cơ bản trong bot của mình.

#### **2.1.5. Chế độ xem web của Messenger**

Chế độ xem web của Messenger là chế độ xem web tiêu chuẩn có thể mở trong ứng dụng Messenger trên các ứng dụng di động và web. Chế độ xem web cho phép bạn cung cấp các trải nghiệm và tính năng yêu cầu giao diện người dùng phức tạp hơn có thể khó tạo được trong cuộc trò chuyện, chẳng hạn như chọn chỗ ngồi để đặt trước hoặc duyệt các ngày có sẵn. Vì đây là chế độ xem web bình thường, tải các trang từ web nên bạn hoàn toàn linh hoạt trong việc tạo ra trải nghiệm phong phú. Trải nghiệm này giúp mở rộng bot Messenger với tất cả các khung, công cụ và thư viện phát triển web mà bạn yêu thích.

#### **2.1.6. SDK JS tiện ích Messenger**

SDK tiện ích Messenger cung cấp quyền truy cập vào các thông tin và tính năng cốt lõi của Messenger trong chế độ xem web của Messenger. Ví dụ: getContext() cung cấp thông tin về cuộc trò chuyện, chẳng hạn như PSID và trang cá nhân của người mở chế độ xem web, beginShareFlow() cho phép bạn chia sẻ nội dung từ chế độ xem web trở lại cuộc trò chuyện và PaymentRequest cho phép bạn chấp nhận khoản thanh toán.

### **2.1.7. Tiện ích trò chuyện**

Tiện ích chat là một tính năng mạnh mẽ của Nền tảng Messenger cho phép bạn tạo ra trải nghiệm Messenger mang tính chia sẻ và cộng tác. Mọi người có thể đưa ngay trải nghiệm này vào cuộc trò chuyện với bạn bè, gia đình và đồng nghiệp. Không giống như hầu hết các tương tác trong Messenger, thường diễn ra trực tiếp giữa người dùng và bot dưới dạng cuộc trò chuyện, Tiện ích trò chuyện cho phép bạn sử dụng chế độ xem web của Messenger để tạo ra trải nghiệm có thể gọi trong bất kỳ cuộc trò chuyện nào. Ví dụ: Tiện ích chat danh sách mua hàng có thể cho phép nhiều người trong cuộc trò chuyện cập nhật một danh sách được chia sẻ.

Bất cứ khi nào một người bắt đầu cuộc trò chuyện với bot Messenger, Tiện ích trò chuyện sẽ tự động hiển thị trong ngăn trình biên soạn của Messenger, sau đó có thể được gọi trong bất kỳ cuộc trò chuyện nào.

### **2.1.8. Xử lý ngôn ngữ tự nhiên tích hợp**

Tính năng xử lý ngôn ngữ tự nhiên tích hợp (NLP) của Nền tảng Messenger giúp đưa nền tảng ngôn ngữ tự nhiên Wit.ai mạnh mẽ đến thẳng bot Messenger, cho phép bạn phát hiện mục đích và ý nghĩa trong các tin nhắn gửi tới bot của mình. Với NLP tích hợp, bạn có thể cho phép nhiều loại thực thể có sẵn của Wit. Các loại thực thể này đã được đào tạo để phân tích cú pháp các thành phần phổ biến trong cuộc trò chuyện, như ngày giờ, truy vấn tìm kiếm, khoảng cách, lời chào, v.v. hoặc bạn có thể tạo các thực thể của riêng mình và đào tạo Wit nhận dạng các cấu trúc ngôn ngữ tự nhiên dành riêng cho bot, chẳng hạn. Khi NLP tích hợp được bật, Nền tảng Messenger sẽ tự động gửi lại phân tích của Wit tới webhook của bạn với mọi tin nhắn.

### **2.1.9. Plugin web**

Nền tảng Messenger hỗ trợ một loạt các plugin giúp người dùng dễ dàng bắt đầu cuộc trò chuyện với bot Messenger của bạn từ web. Plugin web hoạt động bằng cách tích hợp các thành phần chuẩn hóa vào trang web của bạn, chẳng hạn như nút, để tự động mở cuộc trò chuyện với bot trong messenger.com hoặc ứng dụng Messenger. Bot của bạn sẽ nhận được sự kiện webhook bất cứ khi nào bắt đầu cuộc trò chuyện mới từ plugin. Bạn cũng có thể chuyển dữ liệu tùy chỉnh sang plugin, chẳng hạn như thông tin người dùng và nơi tham chiếu.

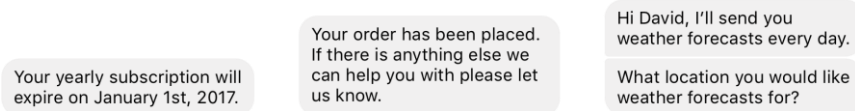
### **2.1.10. Các thành phần trong cuộc trò chuyện**

Cuộc trò chuyện không chỉ gồm các tin nhắn văn bản đơn giản. Mà còn có cả âm thanh, video, hình ảnh, tệp, tin nhắn có cấu trúc (và tin nhắn văn bản!).

#### **2.1.10.1. Các thành phần có trong cuộc trò chuyện**

- Tin nhắn văn bản
- Tập đính kèm
- Mẫu tin nhắn
- Nút
- Trả lời nhanh
- Hành động của người gửi
- Màn hình chào mừng
- Menu ổn định

#### 2.1.10.1.1 Tin nhắn văn bản



*Hình 1*

Nếu mục tiêu của bạn là tạo một trải nghiệm trò chuyện, thì tin nhắn văn bản đơn giản là nền tảng của mọi trải nghiệm trên Messenger và là một trong những công cụ quan trọng nhất mà bạn có thể tùy ý sử dụng. Hãy thử xử lý các tin nhắn văn bản bằng tính năng xử lý ngôn ngữ tự nhiên (NLP) được tích hợp của Nền tảng Messenger để xử lý tất cả các loại tương tác với tin nhắn văn bản đơn giản.


#### 2.1.10.1.2 Tập đính kèm

Ngoài tin nhắn văn bản, Nền tảng Messenger còn cho phép bạn gửi các tài sản đa phương tiện dưới dạng tin nhắn độc lập hoặc được đính kèm vào các mẫu tin nhắn có cấu trúc. Các loại tài sản được hỗ trợ gồm có:

- Âm thanh
- Video
- Hình ảnh
- Tập

Bạn có thể gửi các tài sản từ URL hoặc hệ thống tệp của mình. Đối với các tài sản bạn định gửi nhiều lần, bạn có thể tải chúng lên trước bằng API Tải lên tệp đính kèm hoặc tải chúng lên trong lần đầu tiên gửi chúng bằng API Gửi để loại bỏ chi phí băng thông và thời gian tải lên phát sinh trong mỗi lần gửi. Các tài sản đã lưu được gửi kèm với `attachment_id`. ID này được chỉ định khi tải tài sản lên.

#### 2.1.10.1.3 Mẫu tin nhắn

Please select an option:	
<a href="#">Option 1</a>	<b>Classic White T-Shirt</b> Soft white cotton t-shirt is back in style
<a href="#">Option 2</a>	<a href="#">View Item</a>
<a href="#">Option 3</a>	<a href="#">Bookmark Item</a>

Hình 2

Mẫu tin nhắn là loại tin nhắn có cấu trúc dùng để hỗ trợ các trường hợp sử dụng khác nhau. Các mẫu này giúp hiển thị thông tin trong cuộc trò chuyện khi thông tin này

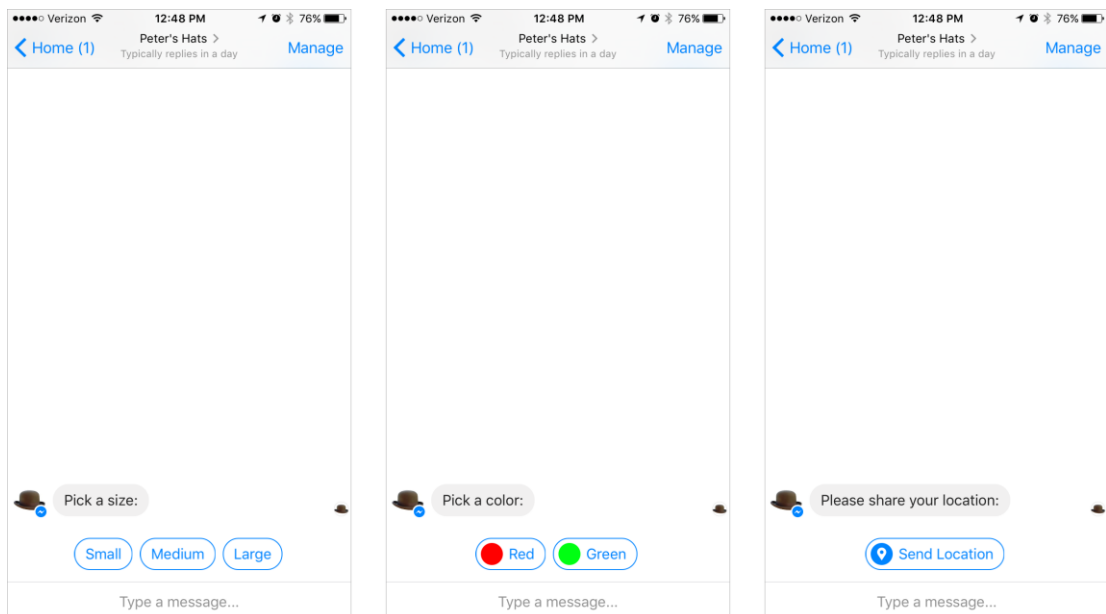
khó hiển thị hoặc trông không ổn với tin nhắn văn bản đơn giản. Các mẫu cũng hỗ trợ những nút giúp mở rộng tính năng của mẫu.

Dưới đây là các mẫu tin nhắn mà bạn có thể sử dụng:

- Mẫu chung
- Mẫu nút
- Mẫu danh sách
- Mẫu Open Graph
- Mẫu biên lai
- Mẫu hàng không
- Mẫu phương tiện

Các mẫu tin nhắn này cũng hỗ trợ một nhóm các nút giúp bổ sung tính năng, chẳng hạn như mở chế độ xem web, gửi yêu cầu đăng lại tới webhook, chia sẻ nội dung v.v.

#### 2.1.10.1.4 Trả lời nhanh



Hình 3

Câu trả lời nhanh cho phép bạn hiển thị rõ ràng ở phía trên trình biên soạn một nhóm các tùy chọn được thiết lập trước cho người nhận tin nhắn. Khi nhấn vào một câu trả lời nhanh, nhóm các tùy chọn này được thay thế bằng một tin nhắn văn bản sẽ được gửi tới webhook của bạn. Bạn cũng có thể thêm hình ảnh vào Trả lời nhanh.

Nền tảng Messenger cũng hỗ trợ tin nhắn trả lời nhanh đặc biệt cho phép người nhận tin nhắn cung cấp cho bot của bạn thông tin vị trí của họ.

#### 2.1.10.1.5 Hành động của người gửi



Hình 4

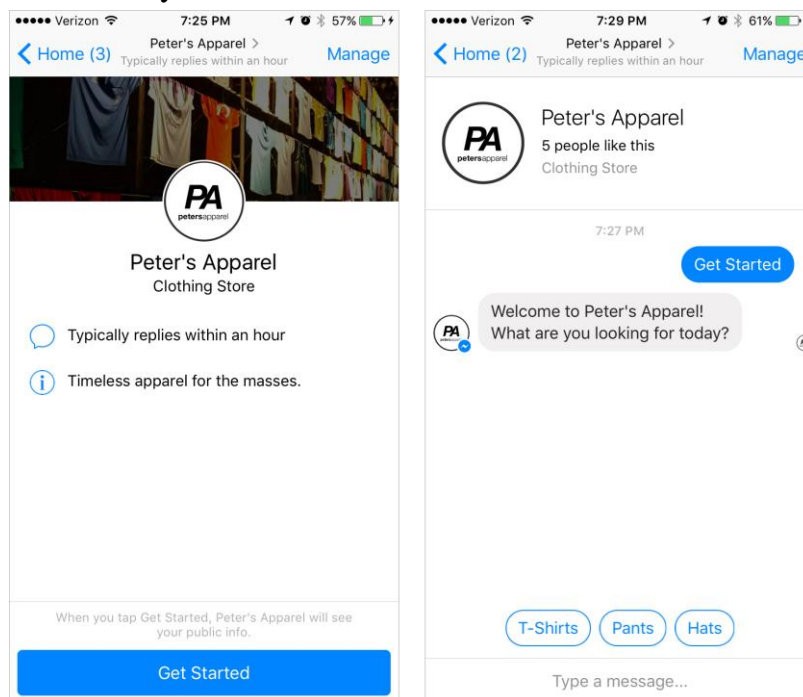
Một khía cạnh quan trọng khi tạo bot Messenger là đặt ra các kỳ vọng. Các hành động của người gửi là công cụ quan trọng để thực hiện việc này nhằm mang đến cho bạn khả năng kiểm soát theo quy trình chỉ báo đang nhập và chỉ báo xác nhận đã đọc trong cuộc trò chuyện trên Messenger tiêu chuẩn. Ví dụ: khi bắt đầu xử lý một tin nhắn, bạn có thể đặt chỉ báo xác nhận đã đọc để người tương tác với bot của bạn biết rằng bạn đã xem tin nhắn của họ, sau đó bạn có thể đặt chỉ báo đang nhập để cho họ biết bạn đang nhập câu trả lời.

#### 2.1.10.1.6 Màn hình chào mừng



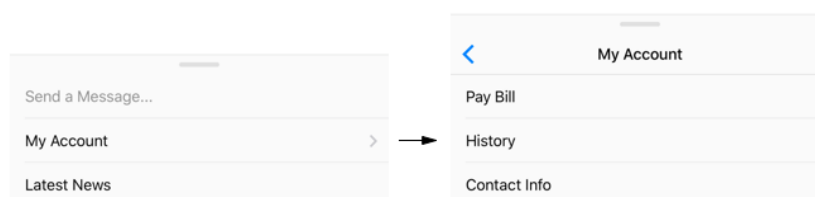
Màn hình chào mừng là điều đầu tiên mọi người nhìn thấy khi bắt đầu cuộc trò chuyện mới với bot Messenger của bạn. Màn hình này hiển thị tên, mô tả, ảnh đại diện và ảnh bìa từ Trang Facebook của bạn. Bạn cũng có thể đặt văn bản chào mừng tùy chọn cho màn hình chào mừng. Phần văn bản này cũng có thể dùng để giới thiệu mục đích của bot.

Một cuộc trò chuyện với bot của bạn bắt đầu khi nhấn vào nút bắt đầu.



Hình 5

#### 2.1.10.1.7 Menu ổn định



Hình 6

Menu ổn định là một thành phần giao diện người dùng luôn bật để giúp mọi người khám phá và dễ dàng truy cập chức năng của bot trong suốt cuộc trò chuyện. Menu này sẽ bao gồm các hành động ở cấp cao nhất mà một người có thể thực hiện ở bất kỳ thời điểm nào. Bạn cũng có thể tùy ý đặt menu ổn định làm cách duy nhất để tương tác với bot bằng cách tắt trình biên soạn.

#### **2.1.10.2. Điểm tham gia trò chuyện**

Cho dù bạn muốn tiếp cận mọi người trên Messenger, Facebook, web hay trong thế giới thực, có rất nhiều cách để bắt đầu cuộc trò chuyện.

#### **2.1.11. Thực tiễn tốt nhất về bot**

Bot Messenger là một cách hiệu quả để kết nối với mọi người. Thực tiễn tốt nhất của chúng tôi sẽ giúp bạn hiểu được những lợi ích và thách thức của việc xây dựng trên Nền tảng.

### **2.2. Node.js**

Node.js là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối thiểu tổng chi phí và tối đại khả năng mở rộng. Node.js bao gồm có V8 JavaScript engine của Google, libUV, và vài thư viện khác.

Node.js được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của Joyent.

#### **2.2.1. Puppeteer**

Puppeteer là một thư viện của Node.js, cung cấp API cấp cao để điều khiển Chromium hoặc Chrome thông qua DevTools Protocol.

### **2.3. MongoDB**

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở thuộc họ NoSQL. Nó được thiết kế theo kiểu hướng đối tượng, các bảng trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ trên bảng không cần tuân theo một cấu trúc nhất định nào cả (điều này rất thích hợp để làm big data).

MongoDB lưu trữ dữ liệu theo hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

## CHƯƠNG: 2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 1. Thông báo cho nhóm đối tượng bằng Messenger với mô hình Notitag

#### 1.1. Ví dụ thực tế:

Cô Hạnh muốn gửi thông báo gặp mặt các sinh viên để làm đồ án Nhập môn ngành.

Đầu tiên, cô tạo thẻ có tên là *#nmn.cohanh* bằng cách nhấn vào trang Facebook tên BroadcastNow, với cú pháp:

*create #nmn.cohanh*

Khi đó, hệ thống sẽ thiết lập thẻ. Các sinh viên muốn nhận thông báo này cần phải đăng ký nhận thông tin từ thẻ *#nmn.cohanh* bằng cách nhấn:

*follow #nmn.cohanh.*

Sau đó, cô tiếp tục nhấn:

*send #nmn.cohanh Ngày mai các em tập trung gặp cô tại phòng H304 lúc 7h30.*

Hệ thống sẽ kiểm tra và gửi thông báo này về Messenger của các bạn sinh viên đã đăng ký nhận thẻ *#nmn.cohanh*.

Sau này, khi có các thông báo khác, cô Hạnh tiếp tục nhấn theo đúng cú pháp thì sinh viên sẽ nhận được các thông báo mới.

Khi đồ án nhập môn ngành kết thúc, sinh viên không muốn nhận thông báo từ thẻ này thì nhấn cú pháp:

*unfollow #nmn.cohanh.*

Cô Hạnh muốn xóa thẻ thì nhấn vào hệ thống BroadcastNow:

*delete #nmn.cohanh*

#### 1.2. Mô tả

##### 1.2.1. Khởi tạo thẻ:

Tất cả mọi người đều có thể khởi tạo thẻ gửi thông tin. Người dùng khởi tạo thẻ bằng cách nhấn cú pháp:

*create #<tên thẻ>*

Sau khi nhấn cú pháp trên vào hệ thống thì hệ thống sẽ kiểm tra *#<tên thẻ>* và thẻ *#<tên thẻ>* sẽ được tạo và được lưu trữ trong hệ thống.

Nếu thẻ có #<tên thẻ> trùng với một thẻ có #<tên thẻ> khác thì hệ thống sẽ thông báo “Thẻ đã tồn tại”, người dùng cần phải chọn #<tên thẻ> khác.

#### **1.2.1.1. Quy tắc đặt tên thẻ:**

- Tên thẻ không dấu
- Tên thẻ không dùng kí tự trắng
- Tên thẻ không chứa các ký tự sau: / \ | : ? \* “ < >
- Tên thẻ không quá 255 kí tự
- Tên thẻ không phân biệt chữ hoa và chữ thường

#### **1.2.2. Đăng ký nhận tin từ thẻ:**

Để nhận thông tin từ một thẻ nào đó, người dùng cần phải đăng ký nhận thông tin từ thẻ đó bằng cú pháp:

follow #<tên thẻ>

Sau khi nhấn cú pháp trên vào hệ thống, khi có thông tin mới từ thẻ #<tên thẻ>, người dùng sẽ nhận được thông tin đó trên Messenger của mình.

Nếu thẻ có #<tên thẻ> chưa được tạo thì hệ thống sẽ thông báo “Thẻ không tồn tại”, người dùng cần kiểm tra lại #<tên thẻ> chính xác.

#### **1.2.3. Hủy nhận tin từ thẻ:**

Khi không muốn nhận thông tin từ một thẻ đã đăng ký, người dùng có thể hủy nhận tin từ thẻ bằng cách nhấn cú pháp:

unfollow #<tên thẻ>

Sau khi hủy nhận thông tin, người dùng vẫn có thể đăng ký nhận thông tin lại từ chính thẻ này.

Nếu thẻ có #<tên thẻ> chưa được tạo thì hệ thống sẽ thông báo “Tên thẻ không tồn tại”, người dùng cần kiểm tra lại #<tên thẻ> chính xác.

Người dùng không thể hủy nhận thông tin từ thẻ nếu chưa đăng ký nhận thông tin từ thẻ này trước đó.

#### **1.2.4. Gửi nội dung vào thẻ:**

Để gửi thông báo vào một thẻ đã được tạo, người dùng cần nhấn cú pháp vào hệ thống:

send #<tên thẻ> <nội dung>

Nếu thẻ có #<tên thẻ> chưa được tạo thì hệ thống sẽ thông báo “Thẻ không tồn tại”, người dùng cần kiểm tra lại <tên thẻ> chính xác.

Hệ thống sẽ kiểm tra xem người có quyền gửi thông báo vào thẻ đó hay không. Chỉ có những người tạo thẻ mới có quyền gửi nội dung cần thông báo vào thẻ đó. Những người khác sẽ không gửi được thông báo vào các thẻ này để tránh tình trạng gửi các thông báo không đúng.

### 1.2.5. Xóa thẻ:

Để xóa thẻ đã được tạo, người dùng cần nhấn cú pháp:

delete #<tên thẻ>

Nếu thẻ có #<tên thẻ> chưa được tạo thì hệ thống sẽ thông báo “Thẻ không tồn tại”, người dùng cần kiểm tra lại <tên thẻ> chính xác.

Hệ thống sẽ kiểm tra người dùng có quyền xóa thẻ hay không.

Chỉ có những người tạo thẻ mới có quyền xóa thẻ đó.

### 1.2.6. Các chức năng khác:

Xem lại các thẻ người dùng đã tạo, nhấn:

mycreate

Xem lại các thẻ người dùng đăng ký nhận thông tin, nhấn:

myfollow

Xem các cú pháp, nhấn:

help

Cú pháp	Ý nghĩa
create #<tên thẻ>	Tạo thẻ <tên thẻ>
follow #<tên thẻ>	Đăng ký nhận tin từ thẻ <tên thẻ>
unfollow #<tên thẻ>	Hủy nhận tin từ thẻ <tên thẻ>
send #<tên thẻ> <nội dung>	Gửi <nội dung> cho tất cả ai đăng ký nhận thẻ <tên thẻ>
delete #<tên thẻ>	Xóa thẻ <tên thẻ> đã tạo
Mycreate	Yêu cầu trang BroadcastNow nhấn lại các danh sách các thẻ của người dùng đã tạo
Myfollow	Yêu cầu trang BroadcastNow nhấn lại các danh sách các thẻ đã được người dùng đăng ký nhận tin
help	Yêu cầu trang BroadcastNow nhấn lại danh sách cú pháp

*Bảng 1 các cú pháp nhắn tin*

## 2. Gửi thông báo từ trang sv.dut.udn.vn về Messenger

Khi người dùng đăng ký nhận thông tin từ trang sv.dut.udn.vn thì hệ thống sẽ kiểm tra và gửi thông báo mới nhất về Messenger cho người dùng đó.

Để đăng ký nhận thông tin, người dùng cần nhấn cú pháp:

follow #public.dut

Để hủy nhận thông tin, người dùng cần nhấn cú pháp:

unfollow #public.dut

## CHƯƠNG: 3 TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

### 1. Các tính năng của hệ thống

- Thiết lập webhook kết nối với Messenger Platform để gửi, nhận tin nhắn. Nhận thông tin người dùng. Gửi tin nhắn hàng loạt.
- Kiểm tra các quyền của người sử dụng khi thực hiện hành động.
- Tương tác với hệ quản trị cơ sở dữ liệu để đọc ghi thông tin.
- Lấy thông báo mới từ trang sv.dut.udn.vn.

### 2. Cấu trúc dữ liệu

Về cơ bản, mỗi người dùng BroadcastNow sẽ được lưu những thông tin bao gồm: tên, ảnh đại diện, giới tính, danh sách thẻ đã tạo, danh sách thẻ đã đăng ký nhận, ID người dùng. Nhưng vì khi truy xuất tất cả các danh sách thẻ người dùng đã tạo hoặc đăng ký thì sẽ bị chậm do chương trình cần tìm đến từng người để thu nhập các thẻ ra so sánh. Vậy nên ta tạo một vùng dữ liệu riêng, là vùng tổng hợp tất cả các thẻ đã tạo. Ở đây ta có 2 kiểu lưu trữ:

Dữ liệu người dùng

Tên  
ảnh đại diện  
giới tính  
ID người dùng  
Danh sách thẻ đã tạo  
    Thẻ thứ 1  
    Thẻ thứ 2  
    Thẻ thứ 3  
    ...  
Danh sách thẻ đã đăng ký  
    Thẻ thứ 1  
    Thẻ thứ 2  
    Thẻ thứ 3  
    ...

Danh sách thẻ mà người dùng đã tạo

Thẻ thứ 1  
Thẻ thứ 2  
Thẻ thứ 3  
...

Để gửi tin nhắn hàng loạt, chúng ta cần liên kết các người dùng nhận chung một thẻ lại với nhau để tiện truy xuất, như vậy ta cần thêm một cấu trúc:

Danh sách thẻ chứa nhóm người dùng

Thẻ thứ 1  
 ID Người dùng 1  
 ID Người dùng 2  
 ID Người dùng 3  
 ...  
 Thẻ thứ 2  
 ...  
 ...

Để lưu thông báo mới từ trang sv.dut.udn.vn, ta có cấu trúc:

Thông báo thứ 1  
 Ngày  
 Tiêu đề  
 Nội dung  
 Danh sách đường dẫn  
 Link 1  
 Tên đường dẫn  
 Đường dẫn  
 Link 2  
 Tên đường dẫn  
 Đường dẫn  
 ...  
 Thông báo thứ 2  
 ...  
 ...

Các cơ sở dữ liệu khác như để lưu các văn bản, hướng dẫn. Ta có cấu trúc:

Danh sách hướng dẫn  
 Hướng dẫn sử dụng

### **3. Mô tả thuật toán theo từng chức năng, tính năng**

#### **3.1. Tính năng của webhook**

Webhook có chức năng nhận các sự kiện từ Messenger Platform, gọi các hàm khác để xử lý và trả lại kết quả cho Messenger Platform.

Đầu tiên, khi nhận được sự kiện một người nào đó nhắn tin đến hệ thống, gọi hàm Kiểm tra người dùng sự tồn tại của người dùng để thêm vào hệ thống nếu là người dùng mới.



Sau đó gọi hàm phân tích cú pháp và trở đến các hàm khác và nhận kết quả.

Khi có kết quả phân tích, ta sẽ thực hiện hành động như Gửi tin nhắn hàng loạt, Gửi tin nhắn cho người dùng nào đó.

### **3.2. Hàm kiểm tra người dùng có tồn tại hay không**

Tìm kiếm ID người dùng trong Dữ liệu người dùng, nếu người dùng không tồn tại thì khởi tạo người dùng đó.

### **3.3. Hàm phân tích cú pháp**

Phân tích cú pháp và trả về lại kết quả là người dùng yêu cầu chức năng gì.

### **3.4. Chức năng create #<tên thẻ>**

Tìm kiếm thẻ trong Danh sách thẻ mà người dùng đã tạo. Nếu thẻ tạo hợp lệ, ghi thêm thẻ đó vào Danh sách thẻ mà người dùng đã tạo và vùng Danh sách thẻ đã tạo trong Dữ liệu người dùng. Ghi thêm thẻ đó vào Danh sách thẻ chứa nhóm người dùng. Sau đó trả về kết quả là thành công hoặc lỗi gặp phải.

### **3.5. Chức năng follow #<tên thẻ>**

Tìm kiếm thẻ trong Danh sách thẻ mà người dùng đã tạo. Nếu thẻ tạo hợp lệ, ghi thêm thẻ đó trong Danh sách thẻ đã đăng ký của Dữ liệu người dùng. Ghi người dùng đó vào tên thẻ trong Danh sách thẻ chứa nhóm người dùng. Sau đó trả về kết quả là thành công hoặc lỗi gặp phải.

### **3.6. Chức năng unfollow #<tên thẻ>**

Tìm kiếm thẻ trong Danh sách thẻ đã đăng ký của Dữ liệu người dùng. Nếu thẻ hợp lệ, gỡ bỏ thẻ đó trong Danh sách thẻ đã đăng ký trong Dữ liệu người dùng. Gỡ người dùng đó ở tên thẻ trong Danh sách thẻ chưa nhóm người dùng. Sau đó trả về kết quả là thành công hoặc lỗi gặp phải.

### **3.7. Chức năng send #<tên thẻ> <nội dung>**

Tìm kiếm thẻ trong Danh sách thẻ mà người dùng đã tạo. Nếu thẻ tạo hợp lệ trả về lệnh Gửi tin nhắn hàng loạt cho webhook bao gồm tên thẻ, nội dung. Nếu có lỗi thì trả về thông tin lỗi.

### **3.8. Chức năng delete #<tên thẻ>**

Tìm kiếm thẻ trong Danh sách thẻ mà người dùng đã tạo. Nếu thẻ tạo hợp lệ gỡ thẻ trong Danh sách thẻ mà người dùng đã tạo, Danh sách thẻ chứa nhóm người dùng, và cũng truy xuất những người dùng có đăng ký thẻ trong Danh sách thẻ chưa nhóm người dùng để gỡ thẻ trong Dữ liệu người dùng. Nếu có lỗi thì trả về thông tin lỗi.

### **3.9. Chức năng Mycreate**

Truy xuất danh sách thẻ đã tạo trong dữ liệu người dùng và trả lại kết quả, hoặc thông tin lỗi nếu có lỗi.

### 3.10. Chức năng Myfollow

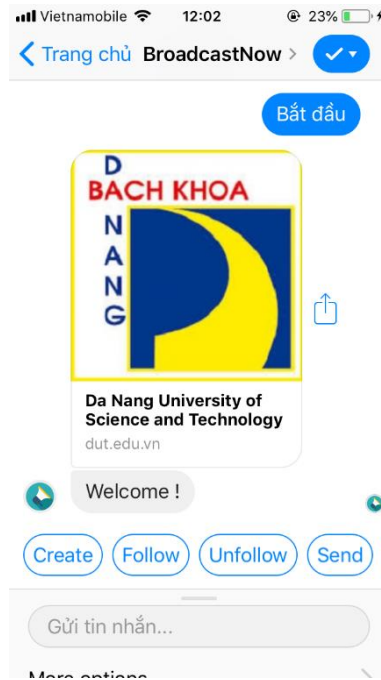
Truy xuất danh sách thẻ đã đăng ký trong dữ liệu người dùng và trả lại kết quả, hoặc thông tin lỗi nếu có lỗi.

### 3.11. Chức năng help

Truy cập hướng dẫn trong Danh sách hướng dẫn và trả kết quả về.

## 4. Trình bày mô hình thử nghiệm

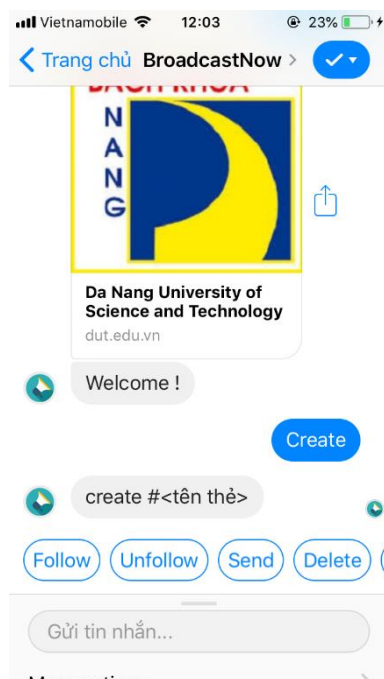
### 4.1. Giao diện mở đầu của BroadcastNow



Hình 7

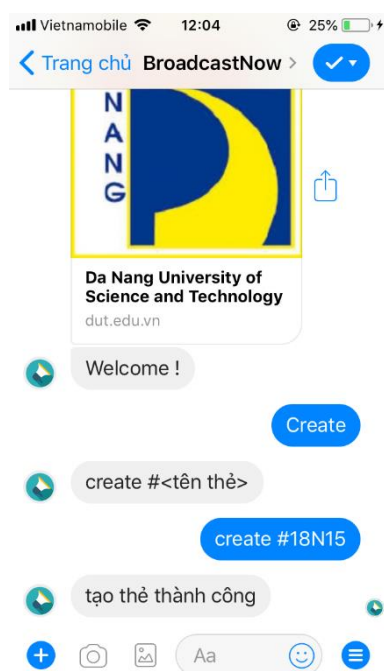
### 4.2. Chức năng khởi tạo thẻ

BroadcastNow khi nhấn cú pháp *create*



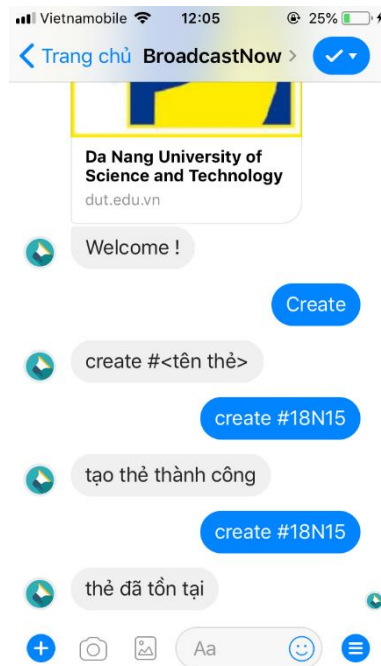
Hình 8

BroadcastNow thông báo khởi tạo thành công thẻ



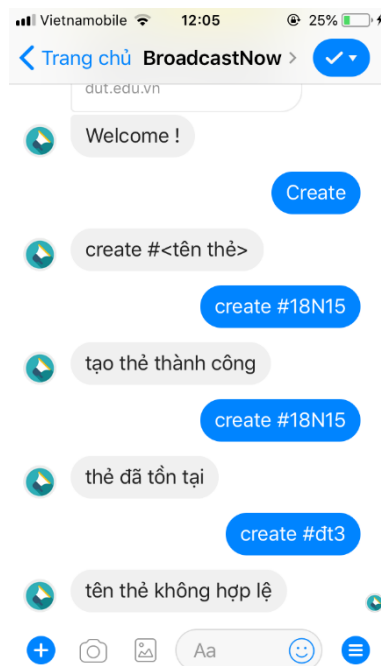
Hình 9

BroadcastNow báo lỗi thẻ đã tồn tại



Hình 10

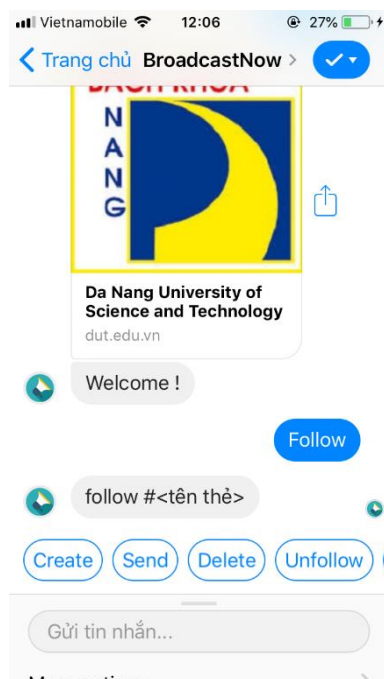
BroadcastNow khi nhập #<tên thẻ> không hợp lệ



Hình 11

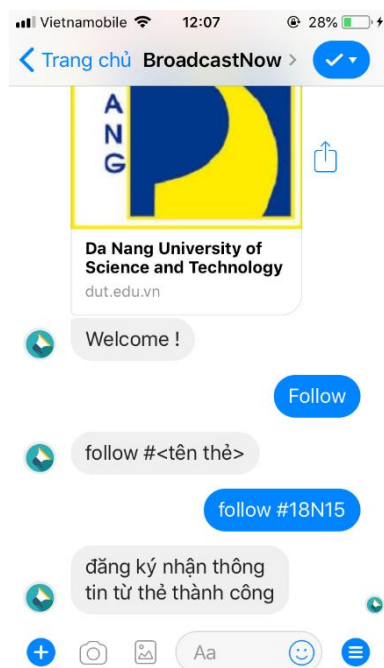
#### 4.3. Chức năng đăng ký nhận tin từ thẻ

BroadcastNow khi nhấn cú pháp *follow*



Hình 12

BroadcastNow thông báo đăng ký nhận thông tin từ thẻ thành công



Hình 13

BroadcastNow báo lỗi thẻ không tồn tại



Hình 14

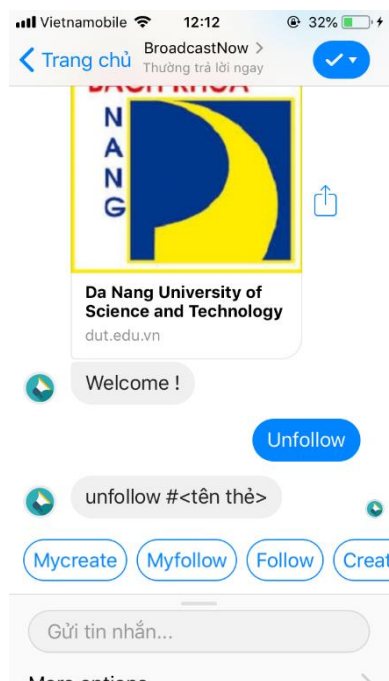
BroadcastNow khi nhập #<tên thẻ> không hợp lệ



Hình 15

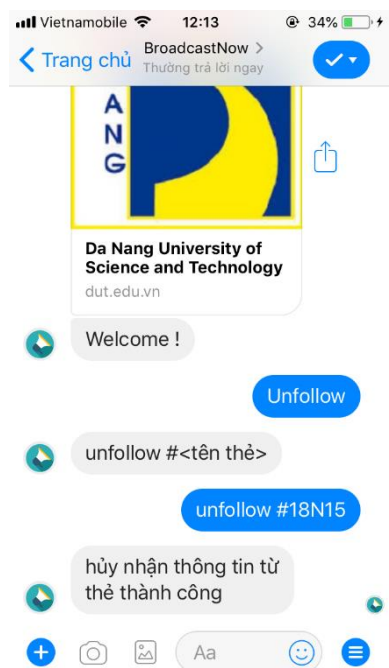
#### 4.4. Chức năng hủy nhận tin từ thẻ

BroadcastNow khi nhấn cú pháp *unfollow*



Hình 16

BroadcastNow thông báo hủy nhận thông tin từ thẻ thành công



Hình 17

BroadcastNow báo lỗi thẻ không tồn tại



Hình 18

BroadcastNow thông báo chưa nhận thông tin từ thẻ

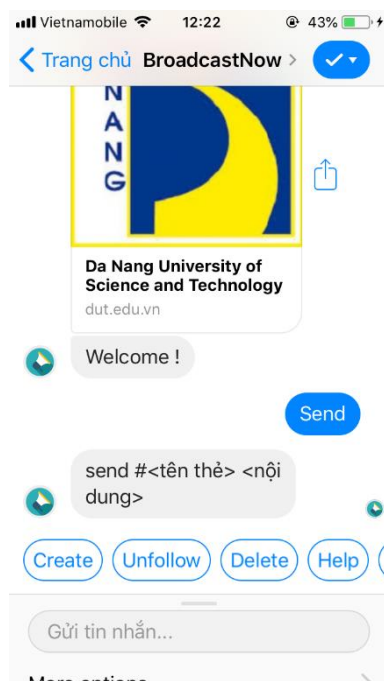


Hình 19

#### 4.5. Chức năng gửi nội dung vào thẻ

BroadcastNow khi nhấn cú pháp *send*





Hình 20

BroadcastNow thông báo gửi nội dung thành công



Hình 21

BroadcastNow báo lỗi thẻ không tồn tại



Hình 22

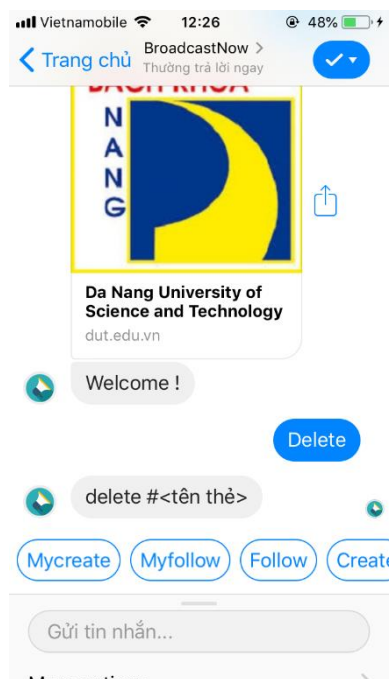
BroadcastNow báo lỗi không có quyền gửi nội dung vào thẻ



Hình 23

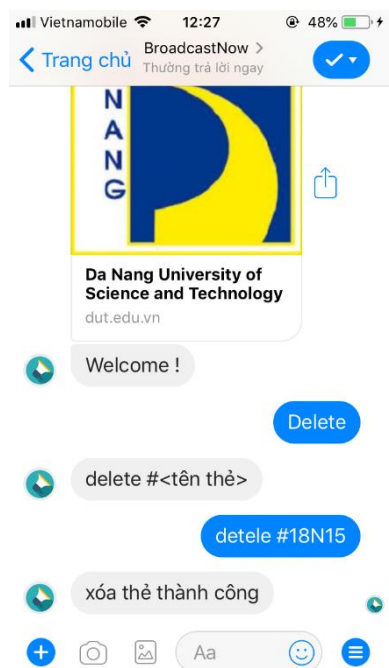
#### 4.6. Chức năng xóa thẻ

BroadcastNow khi nhấn cú pháp *delete*



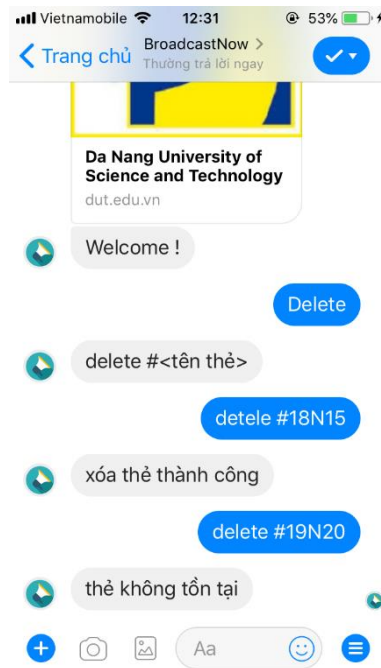
Hình 24

BroadcastNow thông báo xóa thẻ thành công



Hình 25

BroadcastNow báo lỗi thẻ không tồn tại



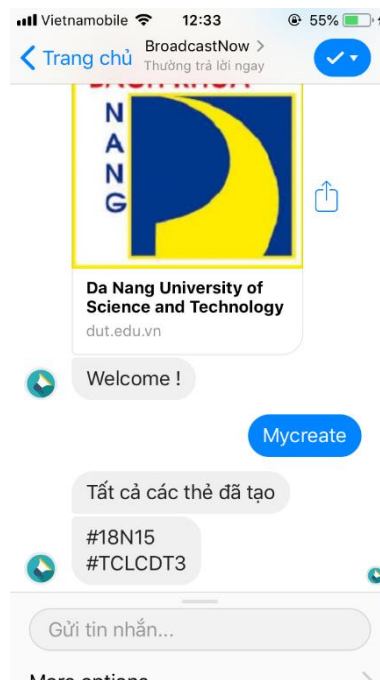
Hình 26

BroadcastNow báo lỗi không có quyền xóa thẻ



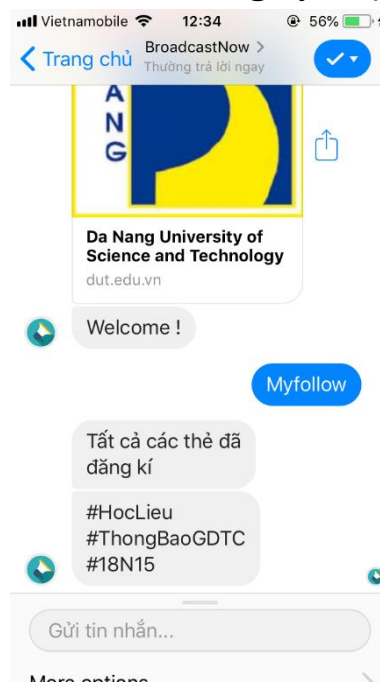
Hình 27

#### 4.7. Chức năng xem lại các thẻ đã tạo



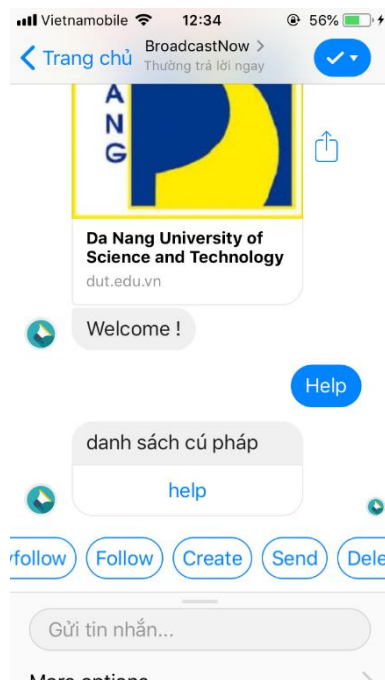
Hình 28

#### 4.8. Chức năng xem lại các thẻ đã đăng ký nhận tin



Hình 29

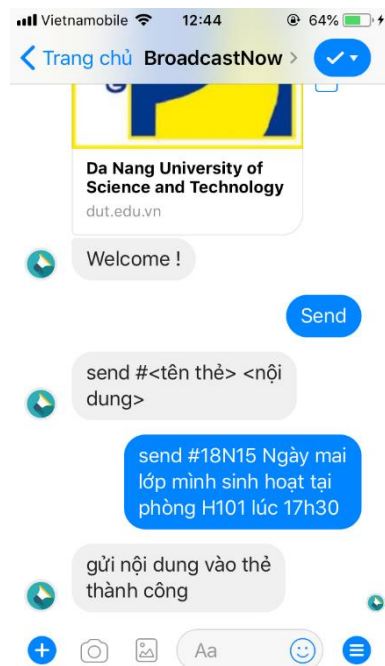
#### 4.9. Chức năng xem các cú pháp



Hình 30

#### 4.10. Ví dụ hoạt động

Giảng viên gửi nhắn vào thẻ #18N15 với nội dung thông báo sinh hoạt lớp



Hình 31

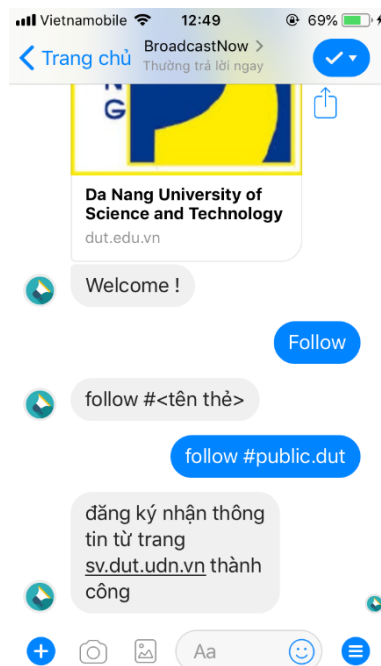
Sinh viên đăng ký nhận tin từ thẻ #18N15 thì sẽ nhận được thông báo



Hình 32

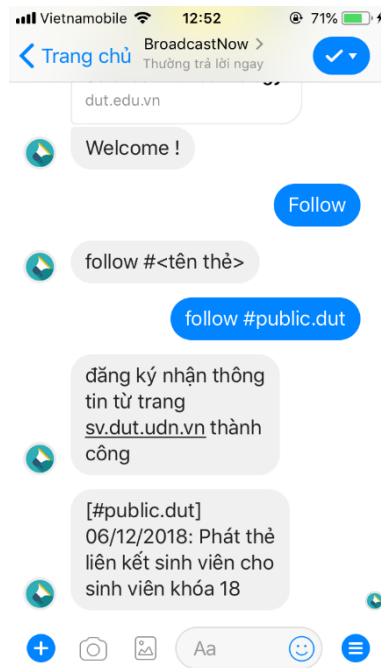
#### 4.11. Gửi thông báo từ trang sv.dut.udn.vn về Messenger

Đăng ký nhận thông tin từ trang sv.dut.udn.vn thành công



Hình 33

Khi có thông báo mới từ trang sv.dut.udn.vn



Hình 34

## Hủy nhận thông tin từ trang sv.dut.udn.vn



Hình 35



## **5. Đánh giá kết quả**

### **5.1. Đối tượng tiếp cận lớn**

Với sự phát triển của mạng xã hội, việc liên kết với nền tảng Messenger với hệ thống riêng ở thời điểm này ngày càng có nhiều người sử dụng.

Độ tuổi sử dụng Messenger tập trung vào khoảng từ 15 đến 35, trong khi đó hệ thống BroadcastNow lại chủ yếu phục vụ cho sinh viên và giảng viên trong quá trình gửi và nhận thông báo nên lượng đối tượng là khá lớn.

### **5.2. Tính thuận tiện đến từ sv.dut.udn.vn**

So với việc truy cập trang web chính “Trang thông tin sinh viên - sv.dut.edu.vn” thì việc đăng nhập vào Messenger thuận tiện và nhanh chóng hơn rất nhiều, chưa kể đến việc đa số sinh viên dùng Messenger để liên lạc, cập nhật tin tức từ bạn bè.

### **5.3. Tính cập nhật**

Tuy hệ thống được coi như một cầu nối trung gian giữa trang thông tin sinh viên với sinh viên nhưng so với trang web chính thì hệ thống BroadcastNow cập nhật thông tin nhanh chóng và chính xác hơn nhiều. Cũng bởi vì là trung gian nên việc xử lý các dữ liệu không quá nặng và phức tạp kèm theo đó sẽ có nhiều chức năng hơn, ví dụ như thu thập thông tin người dùng để giới hạn quyền sử dụng một số cú pháp trong hệ thống.

Việc đưa hệ thống vào thực tế không tốn quá nhiều chi phí cho phần cứng, quan trọng là sự hưởng ứng của giáo viên và sinh viên nên thông qua mạng xã hội là chính sẽ vận động các bạn sử dụng, còn với giảng viên việc sử dụng email đã trở thành truyền thống trong việc trao đổi thông tin thì nhóm sẽ có cố gắng gấp trực tiếp để thuyết phục về vấn đề này.

Việc bảo trì, nâng cấp hệ thống dễ dàng khi không phụ thuộc từ nhiều phía do vậy những thay đổi cần thiết để đáp ứng nhu cầu người dùng sẽ dễ dàng cập nhật và đưa vào sử dụng sớm nhất có thể.

Hệ thống được xây dựng để đáp ứng nhu cầu sinh viên và giảng viên cộng thêm với việc sử dụng trong khuôn khổ trường đại học nên việc triển khai sẽ dễ dàng hơn.

Việc nhà trường thông tin tới sinh viên chủ yếu là online nên sự ra đời của hệ thống là hoàn toàn phù hợp với nhu cầu và hoàn cảnh này, cho thấy khả năng áp dụng vào thực tế là khả thi.

#### **5.4. Vấn đề về tính ổn định của Messenger:**

Hệ thống chỉ hoạt động chính xác khi các bên đều trong trạng thái hoàn hảo nhất, việc liên kết với nền tảng Messenger sẽ gây ra bất tiện vì hệ thống chỉ gửi thông tin qua kênh này nên nếu Messenger đang trong quá trình cập nhật hay bị lỗi thì chắc chắn không tránh khỏi thông tin không được gửi đi kịp thời.

#### **5.5. Việc chuyển đổi từ Email sang Messenger:**

Giảng viên chủ yếu dùng Email để làm việc nên chuyển sang hệ thống mới sẽ khó khăn trong việc tiếp nhận, thêm vào đó là các chức năng của Email cho phép đính kèm các file nặng mà hệ thống chưa phát triển được.

#### **5.6. Vấn đề về số lượng thẻ đăng kí**

Nếu sinh viên đăng kí thẻ một cách thiếu khoa học dẫn đến quên tên thẻ và đăng kí nhiều thẻ khác, việc này làm nặng thêm cho hệ thống.

#### **5.7. Khả năng trở nên phổ biến**

Với kỉ nguyên công nghệ 4.0 thì việc thu thập thông tin càng phải chính xác, nhanh chóng và thuận tiện; ngày càng có nhiều người sử dụng mạng xã hội để cập nhật thông tin, mà một trong số đó là ứng dụng nhắn tin Messenger.

Đáp ứng nhu cầu phần lớn sinh viên bởi đây là Messenger là ứng dụng sinh viên sử dụng hằng ngày thêm vào đó sinh viên cần cập nhật thông tin liên tục từ nhà trường, nên việc vừa có thể nhắn tin trò chuyện với bạn bè vừa nhận thông tin từ nhà trường sẽ tiết kiệm được thời gian hơn.

Đảm bảo mọi thông báo từ nhà trường và giảng viên đều được cập nhật kịp thời đến sinh viên.

Tính hướng đối tượng cao nên thông tin của ai sẽ được gửi đến người đó, điều này là một trong những điều mà sinh viên cần nhất thay vì lên trang thông tin sinh viên để dò tìm những thông báo liên quan đến mình.

### **6. Kết luận và hướng phát triển**

#### **6.1. Kết luận**

Để xây dựng được bài phân tích và thiết kế hệ thống thông tin, việc đầu tiên cần làm trước hết đó là phải xây dựng hệ thống sát với thực tế. Khảo

sát hệ thống là một công việc hết sức quan trọng, giúp thu thập dữ liệu một cách chính xác và chi tiết để tiếp tục xây dựng các bước tiếp theo.

Việc phân tích dữ liệu một cách chính xác sẽ giúp chúng ta thiết lập các chức năng một cách hợp lý nhằm phát huy khả năng sử dụng và điều khiển hệ thống, làm cho hệ thống thân thiện với người sử dụng.

Thiết kế giao diện với màu sắc và bố cục hài hòa, mang xu hướng hiện đại đem lại cho người sử dụng cảm giác thoải mái khi tiếp xúc. Đồng thời, các chức năng đem lại sự hiệu quả hơn trong công việc gửi và nhận thông tin giữa giảng viên – sinh viên. Từ đó, hệ thống góp phần không nhỏ thúc đẩy sự phát triển của quá trình nghiên cứu, dạy và học của sinh viên, giảng viên.

Trên đây là bài báo cáo về đồ án “Cải tiến hệ thống thông tin trong trường Đại học”. Trong quá trình làm đồ án, chúng em đã đạt được một số kết quả sau:

- Hoàn thành phân tích và thiết kế hệ thống.
- Thiết kế giao diện đẹp mắt, thu hút người dùng, không gây nhàm chán trong quá trình sử dụng
- Dựa trên một số nền tảng sẵn có, bước đầu tạo được sản phẩm, đảm bảo một số chức năng đã đề ra. Sản phẩm dễ sử dụng, đáp ứng nhu cầu khách quan của hệ thống.

Hạn chế:

- Tuy đã tối ưu tối đa thuật toán xử lý các chức năng, thời gian chạy của hệ thống vẫn khá lớn nếu có nhiều tags và người dùng. Ngoài ra việc lưu trữ dữ liệu cần cải tiến để đáp ứng được việc gửi nhận thông tin của giảng viên – sinh viên.

## **6.2. Hướng Phát triển**

### **Cải tiến việc đăng kí tín chỉ (đăng kí nhận thông tin từ thẻ)**

Nếu số lượng người dùng truy cập hệ thống lớn sẽ dẫn đến tình trạng nghẽn nút cổ chai. Để giải quyết vấn đề này, nhóm dự định dùng AI để phân bố người dùng đăng kí tín chỉ một cách hợp lý, làm giảm gánh nặng mà hệ thống phải chịu.

### **Nâng cấp hệ thống đầy đủ tính năng:**

Ngoài những tính năng cơ bản đã được phân tích và thiết kế ở trên, trong tương lai, nhóm sẽ tiếp tục nghiên cứu các tính năng khác như hiển thị điểm của sinh viên, đánh giá kết quả học tập, giảng

dạy của sinh viên và giảng viên, thông tin cụ thể về sinh viên và giảng viên, chương trình đào tạo, thông tin cụ thể học phần, lịch sinh hoạt,..

## **CHƯƠNG: 4      KẾT LUẬN CHUNG**

Trong thời kì công nghiệp 4.0 đang diễn ra hiện nay, đổi mới sản phẩm là một yêu cầu bắt buộc đối với mọi dự án để có thể cạnh tranh và không ngừng phát triển. Tất cả dự án không chỉ dừng lại ở ý tưởng tốt mà phải hiện thực hóa dự án đúng với thiết kế và đáp ứng những yêu cầu đề ra.

Với mong muốn giúp sinh viên – giảng viên có môi trường tốt hơn để thực hiện việc gửi nhận thông tin, nhóm đã có ý tưởng “Cải tiến hệ thống thông tin trong trường Đại học”. Qua thời gian phân tích ý tưởng, khảo sát thực tế và đi vào triển khai, nhóm đã thiết kế được cấu trúc và thuật toán của các chức năng cơ bản trong hệ thống. Để thử nghiệm cấu trúc đã thiết kế, nhóm đã cùng nhau làm một sản phẩm đơn giản sử dụng Messenger Platform để đánh giá hệ thống, bước đầu mang lại những kết quả tích cực. Với sản phẩm này, người dùng có thể sử dụng đa số các chức năng mà dự án đã đề ra. Tuy nhiên, dù cố gắng hết sức nhưng hệ thống được thiết kế vẫn không tránh khỏi những hạn chế, bên cạnh đó nhóm cũng nảy ra những ý tưởng mới, cải tiến hệ thống đầy đủ chức năng và thời gian thực thi tác vụ nhanh hơn.

Mặc dù đã bước đầu tạo được sản phẩm nhưng vẫn còn nhiều khuyết điểm, thiếu sót. Trong thời gian tới, nhóm phải không ngừng học tập những kiến thức mới để hoàn thiện sản phẩm như: cơ sở dữ liệu, thuật toán, ngôn ngữ lập trình (Javascript, Nodejs,..). Ngoài ra, từ những nhận xét của các thầy cô giáo trong khoa, chúng em sẽ khắc phục những hạn chế còn tồn tại đồng thời nghiên cứu thêm chức năng, từng bước ứng dụng thực tế, phần nào có thể giúp sinh viên – giảng viên hoàn tất việc dạy và học.

## **TÀI LIỆU THAM KHẢO**

## **PHỤ LỤC**