

# Báo cáo nhóm FiveStar

## 1 GIỚI THIỆU TỔNG QUAN

### 1.1 Phân tích chợ tốt

- Cách tiếp cận:
  - ✓ Nhóm xem CHỢ TỐT giống như chợ truyền thống thì sẽ có các gian hàng cố định (cửa hàng), các gánh hàng rong (những cái nhân bán) và người mua.
  - ✓ CHỢ TỐT hiện thực hóa chợ truyền thống thành online.
  - ✓ CHỢ TỐT tạo ra một nền tảng cho phép kết nối trực tiếp giữa người bán và người mua. Quá đó giúp cho việc mua bán trao đổi trở nên dễ và với phạm vi rộng hơn chợ truyền thống.
- Chiến lược ban đầu của CHỢ TỐT là đánh trên lĩnh vực mua bán đồ củ. Sau đó phát triển dần lên các lĩnh vực khác.
- CHỢ TỐT hiện có hơn 10 triệu users, vài triệu món đồ.
- Tập khách hàng mục tiêu với lĩnh vực đồ điện:
  - ✓ Sinh viên
  - ✓ Những người có mức thu nhập trung bình trở lại.

#### 1.1.1 Định hướng phát triển của chợ tốt:

- **Cái nhân hóa:** hướng tới cái nhân từng user.
  - ✓ **Đề xuất sản phẩm:** có thể cho phép user chọn danh mục họ quan tâm cũng như phân tích dữ liệu user để từ đó đề xuất các sản phẩm phù hợp cho user.
  - ✓ **Giao diện:** cho phép tùy chỉnh cách hiển thị và màu sắc của ứng dụng theo user.
- **Hiệu quả:** tăng số lượng giao dịch, giúp mua bán diễn ra dễ dàng, nhanh chóng hơn.
  - ✓ Đề xuất thêm các sản phẩm mua kèm cùng với sản phẩm user đang chọn.
  - ✓
- **Chất lượng:** gia tăng chất lượng các mặt hàng được đăng, cũng như chất lượng của nền tảng.
  - ✓ Tốc độ dịch vụ nhanh.
  - ✓ UI/UX tốt.

### 1.1.2 Đối thủ của Chợ tốt có gì ?

- Các cửa hàng bán đồ củ uy tín (như cửa hàng bán Iphone củ Hoàng Kiên (hoangkien.com), cửa hàng bán lap top củ thế giới số TLD (24hlaptop.vn))

- ✓ Điểm mạnh:

- Sản phẩm củ được test kỹ càng trước khi đến tay người tiêu dùng.
- Có bảo hành.
- Có một nơi để người mua đến khi có vấn đề với sản phẩm.

→ đảm bảo sự an tâm hơn về chất lượng.

- ✓ Điểm yếu:

- Không đa dạng về mặt hàng cũng như lĩnh vực mua bán. Mà chỉ chuyên theo một lĩnh vực nào đó.

- ✓ Tính năng học được:

- **Dùng thử sản phẩm:** cho phép người mua dùng thử sản phẩm trong một thời gian (3 ngày, 1 tuần, ...), nếu hài lòng họ sẽ mua sản phẩm đó nếu không thì họ phải trả một khoản phí như là phí thuê sản phẩm sử dụng theo thời gian. Điều kiện là chợ tốt phải là bên trung gian để nắm giữ tiền cọc (bằng giá trị sản phẩm) và thu tiền thuê sản phẩm cho người bán. Xu thế cho thuê các sản phẩm hiện cũng đang là trào lưu mới ở Mỹ và các nước Châu Âu thời gian gần đây.

- Các group và trang bán đồ củ trên facebook

- ✓ Điểm mạnh:

- Họ thường là những trang thông tin theo dòng sản phẩm của từng hãng.
- Bản chất là một cộng đồng hỗ trợ nhau về các lỗi trong quá trình sử dụng sản phẩm. Những người mua hoặc bán ở trên này thường là fan của các hàng đó.
- Cho phép người dùng bình luận trên các bài đăng, cũng như nhận được các thông báo về các bài đăng bán sản phẩm trong các group mình theo dõi.
- Facebook mới có tính năng mới là marketplace cho phép đăng bán và xem các mặt hàng bán trên facebook.
- Lượng người dùng facebook lớn

- ✓ Điểm yếu:

- ✓ Tính năng học được:

- **Phát triển kênh thông tin:** Tạo ra các kênh thông tin về sản phẩm, quy trình test thử sản phẩm củ, các lỗi phát sinh khi sử dụng và góc câu chuyện cảnh giác (nơi chia sẻ các thủ đoạn lừa đảo trên chợ tốt đến cộng đồng). Đồng thời chợ tốt có thể bán quảng cáo trên các kênh

này, hay quảng cáo các sản phẩm trên chợ tốt ở trang này như cách mà thế giới di động cũng như các mạng xã hội đang làm.

- Ứng dụng GET IT
  - ✓ Điểm mạnh:
    - Cho phép tìm các sản phẩm gần mình.
    - Cho phép bình luận trong các mặt hàng được đăng.
  - ✓ Điểm yếu:
    - Mới ra, chưa được nhiều người biết đến.
  - ✓ Tính năng học được:
    - Tìm sản phẩm gần mình
    - **Bình luận trong các mặt hàng đăng:** cho phép nhưng người đã đi xem sản phẩm review lại về sản phẩm cho những người mua sau.

## 1.2 Vấn đề của chợ tốt

Từ những phân tích trên nhóm xin đưa ra các vấn đề sau:

- Liệt kê theo các bước thực hiện quá trình mua hàng và nêu ra các vấn đề:
  - ✓ Bước 1: Vào trang home
    - **Hiện thị sản phẩm mẫu theo danh mục:** Khi một người vào chợ tốt họ không biết đang muốn tìm kiếm gì ? lướt trang home chợ tốt chưa show ra được các sản phẩm mẫu theo các danh mục.
    - **Đề xuất sản phẩm:** Chưa có đề xuất các sản phẩm theo sở thích và hành vi của user.
  - ✓ Bước 2: Chọn các điều kiện lọc và tìm kiếm
    - Phần Lọc thiếu:
      - **Lọc các sản phẩm gần mình gần nhất:** Có những món hàng nào đang được bán gần mình (bao nhiêu km quanh khu vực user) ? hoặc là user có xe đạp nên chỉ đi xem được các sản phẩm trong vòng 2km thôi, xa hơn không được thì làm sao ?
  - ✓ Bước 3: chọn sản phẩm để xem chi tiết
    - Sản phẩm bán rồi hay không không biết ?
    - **Đề xuất thêm các sản phẩm mua kèm:** Các sản phẩm mua kèm này nên là sản phẩm mới, được bán bởi các cửa hàng uy tín, được giao tận nhà và thanh toán online. Ví dụ mua điện thoại củ thì có thể bán kèm kính cường lực, ốp lưng, gậy tự xoay; mua laptop thì có thể bán kèm túi chống sốc, đế tản nhiệt,...
    - Chưa hiển thị các **sản phẩm khác của cùng người bán** để user có thể đến xem cùng lúc nhiều sản phẩm và mua chúng.
    - **Không có các sản phẩm tương tự:** đề xuất cho user các sản phẩm tương tự để lựa chọn.

- Chưa cho phép bình luận trong các mặt hàng đăng.
- **Việc đánh dấu sản phẩm:** Có nhiều sản phẩm được chọn (đánh dấu) thì chưa đưa ra các tiêu chí giúp người mua dễ đưa ra quyết định. (Như *mức giá đã rẻ bao nhiêu % so với sản phẩm mới ? chất lượng còn bao nhiêu % so với sản phẩm mới ? Khoảng cách như thế nào so với người mua ?*)
- **Trường hợp không tìm thấy sản phẩm phù hợp:** chưa cho phép đặt hàng (tạo thông báo khi có sp phù hợp) bắt người mua phải mỗi ngày vào ứng dụng để xem những sản phẩm mới nào được đăng.
- ✓ Bước 4: Liên hệ người bán
  - Hiện phần này khá khá ổn. có thể nhắn tin trực tiếp hoặc gọi điện.
- ✓ Bước 5: Hẹn gặp người bán xem và quyết định mua
  - **Chưa có địa điểm bán:** chưa cho phép người bán chọn các vị trí họ muốn bán sản phẩm. Ví dụ: các ngày trong tuần đi làm nên muốn bán ở công ty, cuối tuần thì ở quán cafe gần nhà.
  - **Chưa cho dùng thử sản phẩm:** user đặt cọc tiền tương ứng với giá sản phẩm trên chợ tốt, có thể chợ tốt kết nối với bên giao hàng để giao sản phẩm từ người bán đến người mua để họ trả nghiệm.
  - **Chưa có trang thông tin:** cách test sản phẩm cũ, các tình huống lừa gạt, giúp nâng cao trình độ của user giúp họ tự tin hơn khi sử dụng chợ tốt.
- Hiện tượng cò môi: Người mua muốn mua máy từ chính người bán máy thật không có, làm sao để hạn chế ?

### 1.3 Mục tiêu

Phát triển một ứng dụng để giải quyết vấn đề của Chợ Tốt:

- Xây dựng tính năng recommendation cho người mua.
- Dẫn người mua tới cái người ta muốn nhanh nhất có thể.

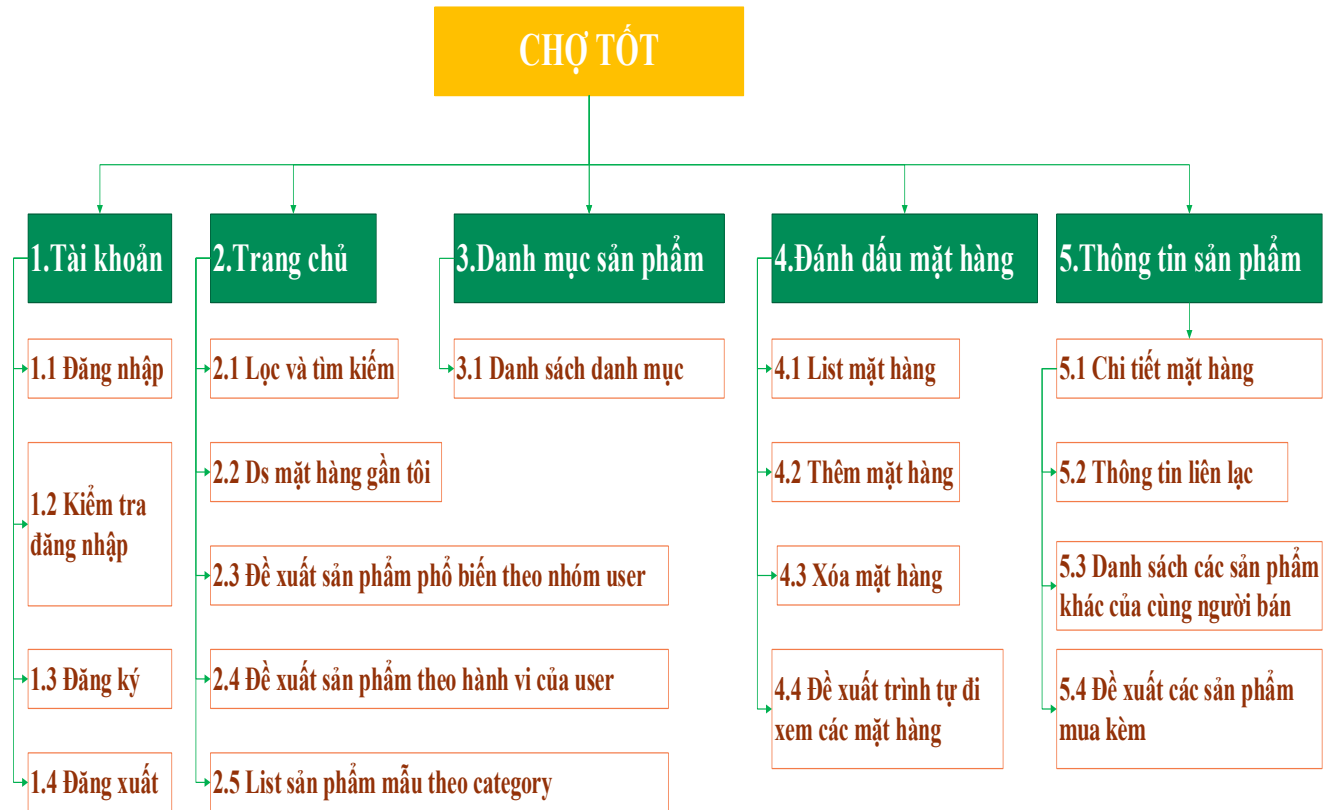
### 1.4 Tiêu chí thành công

- Tất cả các thành viên không ai bỏ cuộc
- Hoàn thành được ít nhất 80% các tính năng đề ra và bàn giao sản phẩm đúng hạn.

### 1.5 Phạm vi dự án

Nhóm thực hiện dự án này với mong muốn tạo một công cụ đề xuất tự động các sản phẩm đến với người tiêu dùng. Hệ thống giúp người tiêu dùng tiếp cận được các sản phẩm mà họ muốn và người bán sẽ bán được nhanh sản phẩm của họ.

Xét theo thời gian cũng như năng lực của team, mọi người trong team quyết định lựa chọn thực hiện các tính năng sau:



## 1.6 Tổ chức dự án

### 1.6.1 Giới thiệu team

- Các thành viên tham gia

STT	Tên thành viên	Lớp	Liên hệ
1	Nguyễn Bá Khánh	Product management	0366 126 070, nguyenbakhanhcntt96@gmail.com
2	Trương Thi Hồng Thắm	Mobile Development	0378144450, 14520824@gm.uit.edu.vn
3	Trần Hoài Hưng	Mobile Development	0967 884 005, tranhoaihung05@gmail.com
4	Lê Trần Anh Đăng	Data Science	0349 646 816, anh dang140296@gmail.com
5	Triệu Đức Huy	Data Science	0388 168 095, huytrieu2712@gmail.com

- Các công nghệ và trách nhiệm

<i>STT</i>	<b>Các phần</b>	<b>Công nghệ</b>	<b>Trách nhiệm</b>
1	Quản lý task	Scrum	Khánh
2	Lên ý tưởng phân tích thiết kế	Figma tạo prototype	Cả team
3	Xây dựng Front End	React Native	Thắm, Hưng
4	Xây dựng Back End	Nodejs	Khánh, Thắm
5	Xây dựng Recommendation system	python	Huy, Đăng

Nhóm phân chia mỗi phần hai người để có thể trao đổi và hỗ trợ nhau trong quá trình làm.

- Các quy tắc trong Team:
  - ✓ Hoàn thành task đúng hạn, nếu trễ bị phạt 5k/ lần.
  - ✓ Cam kết mỗi thành viên bỏ ra 20h làm hoặc hành thành task được giao.
  - ✓ Vắng mặt trong các buổi họp team phạt 10k/lần.

### 1.6.2 Quản lý Srum:

- Sprint:
  - ✓ Thời gian: 1 tuần. Bắt đầu vào thứ 2 kết thúc vào chủ nhật hằng tuần.
  - ✓ Chiều chủ nhật hằng tuần sẽ họp kết thúc sprint và bắt đầu sprint mới: Báo cáo kết quả làm được trong tuần, khó khăn gặp phải, bàn hướng giải quyết và nêu ra nhiệm vụ phải làm ở tuần mới.
- Meeting:
  - ✓ Mỗi tuần họp nhóm vào tối thứ 4 sau khi học xong.

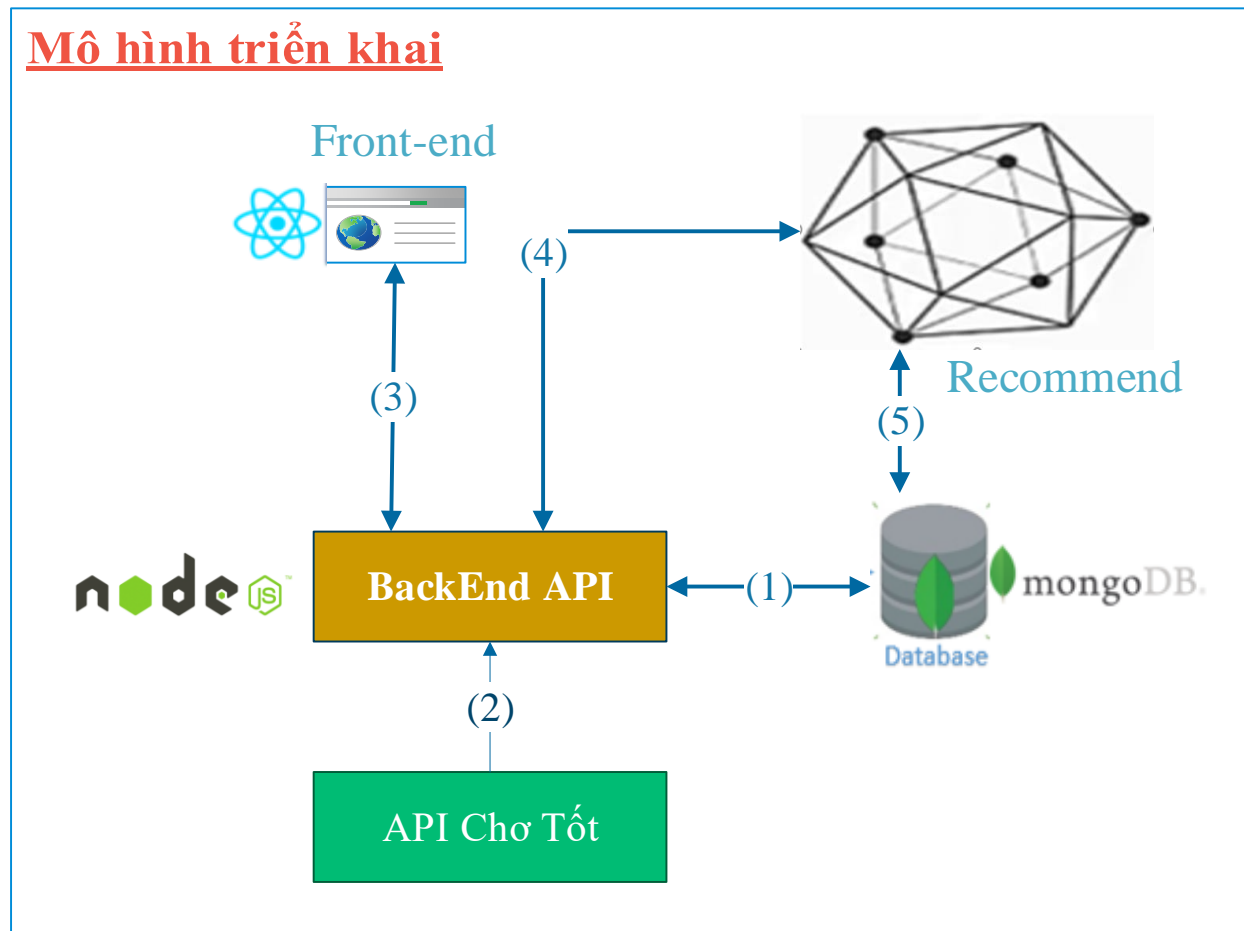
### 1.6.3 Phần mềm và công cụ sử dụng:

- Design: Figma
- Data science: Pycharm.
- Backend: VSCode, Nodejs, npm
- FrontEnd: VSCode, React Native, Expo

## 2 QUÁ TRÌNH THỰC HIỆN

### 2.1 Mô hình thực hiện

#### Mô hình triển khai



Dự án bao gồm 3 phần:

- Backend
- FrontEnd
- Recommendation system

### 2.2 Back End

Phần này sẽ giới thiệu về các công nghệ được sử dụng để thực hiện tầng Backend và liệt kê các API

#### 2.2.1 Chuẩn bị và môi trường

- Hệ điều hành: Ubuntu 16.04
- Nodejs version 10.17.0
- Npm version 6.11.3

## 2.2.2 Xây dựng phần BackEnd

### a) Nodejs

NodeJS là một môi trường runtime open-source, đa nền tảng cho phép các lập trình viên tạo ra các ứng dụng và công cụ phía server trên ngôn ngữ Javascript. Môi trường runtime này có thể được sử dụng bên ngoài ngữ cảnh của browser (chạy trực tiếp trên máy tính hoặc hệ điều hành của server).

NodeJS có các lợi ích sau:

- Hiệu suất tuyệt vời do Node được thiết kế để tối ưu thông lượng (throughput) và khả năng mở rộng (scale).
- Trình quản lý npm với hàng trăm package hỗ trợ.
- NodeJS đặc biệt thích hợp với các ứng dụng thời gian thực do là một ngôn ngữ hướng sự kiện và khả năng xử lý bất đồng bộ.

Express là một web application framework dành cho NodeJS. Nó cung cấp cho các lập trình viên các chức năng mạnh mẽ để viết server-side logic trên ngôn ngữ Javascript. Các tính năng quan trọng:

- Thiết lập các lớp trung gian để trả về các HTTP request.
- Định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

### b) Json Web Token (Jwt)

JWT (Json web token) token, là một chuẩn mở (RFC 7519) định nghĩa cách thức truyền tin an toàn giữa các bên bằng một đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng HMAC hoặc RSA.

Những đặc điểm nổi bật của JWT:

- Kích thước nhỏ: JWT có thể được truyền thông qua URL, hoặc qua giao thức POST, hay nhét vào bên trong phần HTTP Header. Kích thước nhỏ hơn ứng với công việc truyền tải sẽ nhanh hơn. Dưới đây là cách thức truyền token vào trong HTTP Header sử dụng Bearer Schema: Authorization: Bearer <token>
- Khép kín: Phần Payload (hiểu nôm na là khối hàng) chứa toàn bộ những thông tin mà chúng ta cần tới, ví dụ như thông tin của người dùng (thay vì phải truy vấn cơ sở dữ liệu nhiều lần).



### c) Danh sách các API

STT	Path API	Chức năng
1	/api/v1/register	Đăng ký tài khoản gồm phone, password và role (buyer, seller)
2	/api/v1/authentication/login	Đăng nhập với phone và password
3	/api/v1/authentication/logout	Đăng xuất
4	/api/v1/authentication/check-if-logged-in	Kiểm tra thiết bị hiện tại đã đăng nhập hay chưa
5	/api/v1/events/create	Ghi nhận lại sự kiện click của user
6	/api/v1/events/list_all	Trả về tất cả các event data lưu chuyển cho model học
7	/api/v1/events/delete	Xóa các event
8	/api/v1/item	Trả về danh sách các item theo các tham số truyền vào
9	/api/v1/item/detail	Trả chi tiết của item theo item_id
10	/api/v1/recommend	Lấy danh sách item được đề xuất theo userid

## 2.3 Front End

### 2.3.1 Chuẩn bị và môi trường

Nhóm chọn framework React Native làm framework chính để hiển thị hình ảnh để tương tác với client thông qua App.

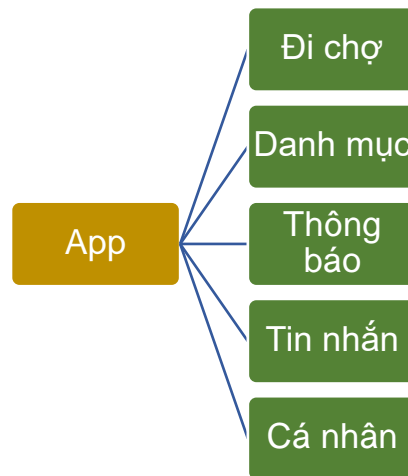
Phần mềm design: Figma,

Phần mềm code: Visual studio code,

### 2.3.2 Phân tích thiết kế

Trong phần phân tích và thiết nhóm chú trọng *xây dựng các tính năng chính tập trung vào người dùng* (ở đây là người muốn mua hàng, chính vì vậy nhóm không phát triển tính năng đối với người bán.).

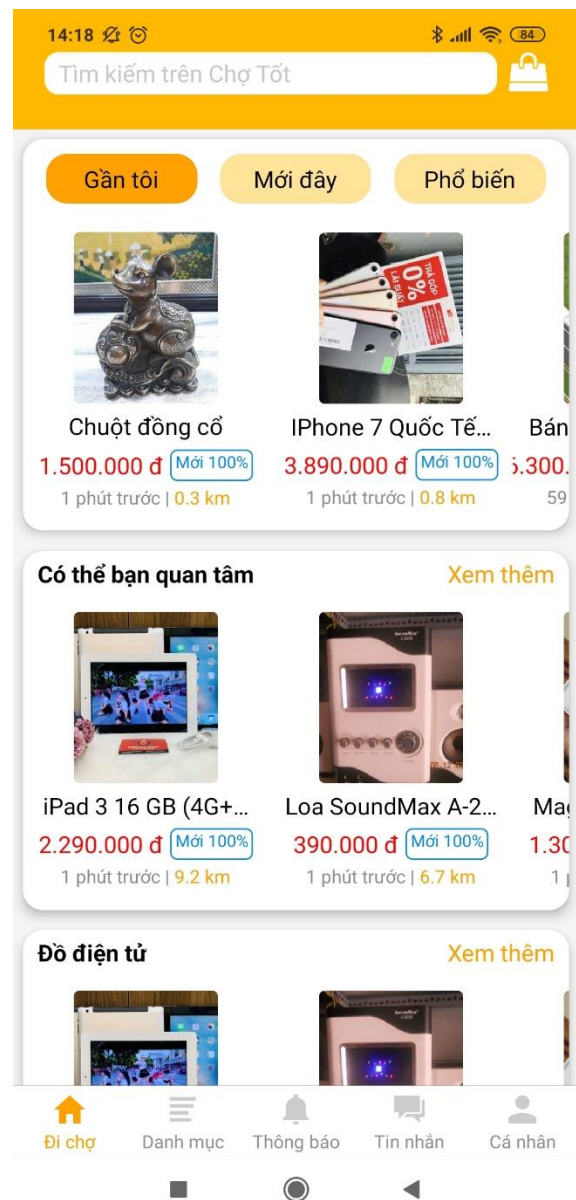
Cấu trúc các màn hình của app:



Ở trang Home: trang chính hiện tại của CHỢ TỐT là các danh mục để người dùng lựa chọn và từ đó đi vào các sản phẩm theo danh mục đã chọn. Ví dụ chọn vào Đồ điện tử sẽ đi vào trang chứa list các đồ điện tử. Việc thiết kế như này thật sự thích hợp đối với những người đã biết chính sản phẩm họ muốn mua rồi. Vậy đối với những người chưa biết họ sẽ cần mua sản phẩm gì thì sao? Họ có thể chọn từng danh mục để vào xem sản phẩm rồi đi ra lại trang home, rồi cứ thế lần lượt cho các sản phẩm. Việc thao tác như vậy sẽ tốn nhiều thời gian và làm cho người muốn mua đồ cảm thấy bị mất nhiều thời gian. Chính vì vấn đề trên, nhóm thay mục home hiện tại của app CHỢ TỐT bằng danh sách các sản phẩm theo các danh mục. Từ đó người mua sẽ đi dạo qua từng danh mục và xem sản phẩm của danh mục đó tại trang home. (Như *hình 2.3.1a* và *2.3.1b*)



Hình 2.3.1a: Mục trang Home hiện tại



Hình 2.3.1b: Mục trang Home của nhóm

Trong danh mục home của nhóm thiết kế, ở phần hiển thị đầu tiên là các danh sách các phẩm gợi ý cho người mua. Đối với các lựa chọn:

- **Gần tôi:** sẽ hiển thị những sản phẩm đang được bán gần với người mua.
- **Mới đây:** sẽ hiển thị những sản phẩm vừa mới được đăng bán.
- **Phổ biến:** sẽ hiển thị những sản phẩm được tìm kiếm nhiều nhất.

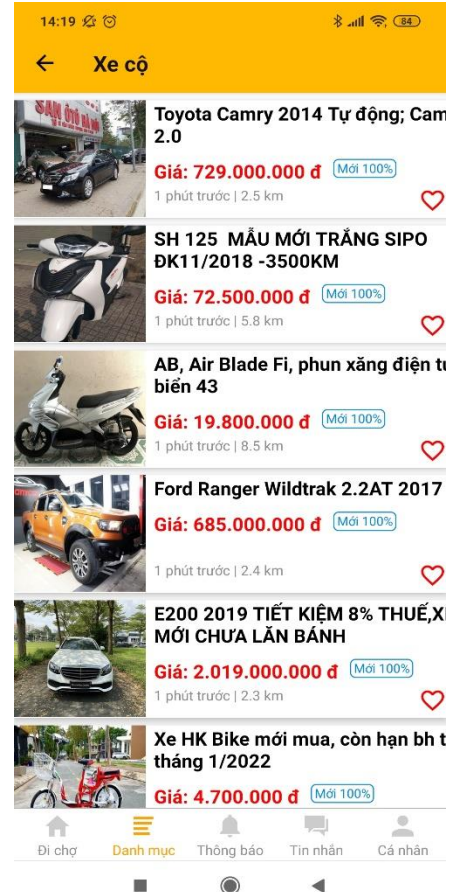
Đối với mục **có thể bạn quan tâm:** sẽ hiển thị kết quả từ quá trình recommend, những sản phẩm đề xuất cho người mua.

Các danh mục còn lại sẽ hiển các sản phẩm của danh mục đó.

Trong trang danh mục, hiển thị các danh mục sản phẩm. Nhận thấy ở danh mục các sản phẩm hiện tại của CHỢ TỐT phần chữ hiển thị tên các danh mục nổi lên trên hình ảnh. Chính vì điều này tạo ra 1 hạn chế với người dùng khi nhìn vào. Chính vì vậy nhóm đã thiết kế lại như hình 2.3.2



Hình 2.3.2: trang lựa chọn các danh mục

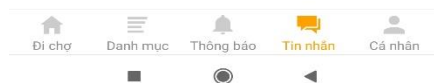


Hình 2.3.2: Danh sách các sản phẩm theo danh mục

Vì thời gian nghiên cứu, và thực hiện có giới hạn nên các tính năng ở màn hình “**Thông báo**”, “**Tin nhắn**”, “**Cá nhân**” nhóm không phát triển. Sau đây là hình ảnh các màn hình:



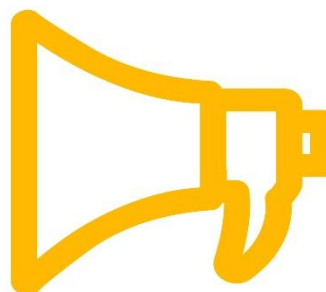
Bạn chưa có tin nhắn!



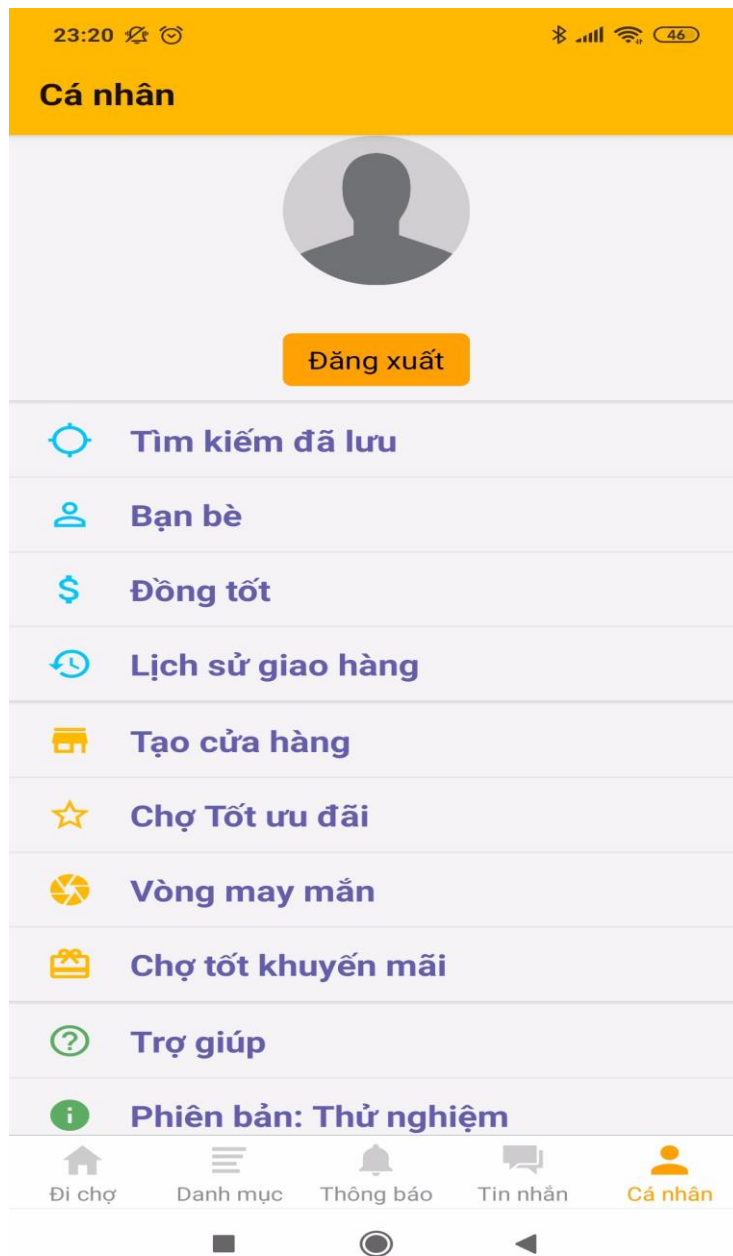
*Hình 2.3.3: Màn hình tin nhắn*



Bạn chưa có thông báo!



*Hình 2.3.4: Màn hình thông báo*



Hình 2.3.5: Màn hình trang cá nhân

←

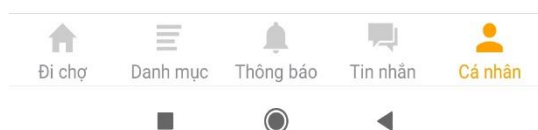
ĐĂNG KÝ

Phone

Mật khẩu

Nhập lại mật khẩu

Đăng ký



Hình 2.3.6: Màn hình đăng ký

18:48

←

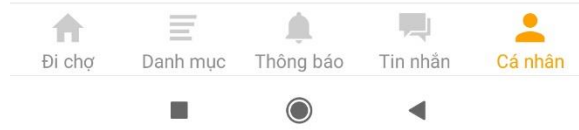
Đăng nhập

Tên đăng nhập

Mật khẩu

Đăng nhập

Tạo tài khoản Quên mật khẩu?



Hình 2.3.7: Màn hình đăng nhập

## 2.4 Recommendation System

### 2.4.1 Thông tin khai thác từ tập dữ liệu

#### Thông tin cơ bản về tập dữ liệu:

- Tổng số lượng người dùng (**user**): **1000 user**.
- Tổng số lượng sự kiện (**event**): 11.043.41
- Thời gian: 15-09-2019 đến 23-09-2019
- Mỗi sự kiện được lưu lại sẽ gồm 19 giá trị đặc trưng. Trong đó,



- Có 2 loại event chính là CLICK và LOAD. Trong đó, CLICK event là 291.864.
  - ✓ Bao gồm 2 loại sự kiện chính là CLICK và LOAD. Trong đó, số lượng của CLICK là 291.864.

Do các event chỉ được tổng kết trong khoảng thời gian ngắn, có thể dẫn đến một số nhận định chủ quan về xu hướng quan tâm của người dùng.

Số lượng CLICK giữa các người dùng khá chênh lệch nhau. Và cũng với mỗi người dùng, số lượng CLICK ở những thời điểm khác nhau là không đồng đều, có những thời điểm chênh lệch rất lớn.

fc2f39cf-2adc-434f-9a39-8e5b356e8a42	2019-09-14	4
	2019-09-15	221
	2019-09-16	92

fd0a647d-f703-4586-a60b-2a908342cf2f	2019-09-15	239
	2019-09-16	79
	2019-09-17	148
	2019-09-18	211
	2019-09-19	84
	2019-09-20	86
	2019-09-21	99
	2019-09-22	60
	2019-09-23	41

Ngoài ra, ở một thời điểm nhất định thì xu hướng quan tâm của người dùng cũng khác, có thể là dựa trên nhu cầu hiện tại của người dùng. Thông thường, người dùng sẽ tập trung vào một hoặc một số mặt hàng xác định. Vì vậy ta sẽ xác định việc gợi ý cho người dùng những mặt hàng liên quan đến nhu cầu hiện tại thay vì dựa trên toàn bộ lịch sử thao tác của người dùng.

Dựa vào những phản hồi của người dùng, ta chia làm hai loại là explicit feedback và implicit feedback. Explicit feedback bao gồm các hành động mua hàng, bình luận, đưa vào danh sách yêu thích. Implicit feedback bao gồm các hành động nhấn vào sản phẩm, xem thông tin sản phẩm, nhấn gửi tin nhắn, nhấn gọi người dùng. Đa phần các loại sự kiện trong dữ liệu đều là implicit feedback. Tuy là implicit feedback, nhưng mỗi sự kiện đều thể hiện một mức độ quan tâm khác nhau của người dùng với từng sản phẩm. Với dữ liệu hiện tại, ta chỉ thực hiện việc đánh giá dựa trên sự kiện CLICK của người dùng. Ta đánh trọng số cho mỗi sự kiện như sau:



```

EVENT_WEIGHTS = {
    'ADLISTING/CLASSIFYAD_CLICK': 1.5,
    'ADLISTING/ADVIEW_CLICK': 1.5,
    'ADVIEW/SHOW_PHONE_CLICK': 2.5,
    'ADVIEW/SHOW_PHONE_CLICK_BODY': 2.5,
    'ADVIEW/CHAT_CLICK': 3,
    'ADVIEW/SMS_CLICK': 4,
    'ADVIEW/CALL_CLICK_BODY': 5,
    'ADVIEW/CALL_CLICK': 5
}

```

Thông tin về các đặc tính của các item có nhiều sự khác biệt lớn. Không những giữa các mặt hàng khác loại với nhau (điện thoại – loa) mà giữa các mặt hàng cùng loại với nhau cũng có sự khác biệt. Do đó, khá là khó khăn khi định nghĩa các đặc tính cụ thể chung để phân biệt các mặt hàng với nhau. Điều này ảnh hưởng tới khá nhiều về việc lựa chọn thuật toán phục vụ cho việc gợi ý.

### 2.4.2 Lựa chọn thuật toán

Dựa vào những nhận định trên, ta thấy việc xây dựng thuật toán dựa trên content-based khá là khó khăn. Dựa vào user thì số lượng user khá ít, có thể không mang đến sự đa dạng về dữ liệu. Về việc dựa vào thông tin của các item thì cũng đã phân tích như trên. Ta khó có thể tìm và định nghĩa được các đặc tính chung giữa các item với nhau. Do đó, việc thực hiện gợi ý dựa trên content-based sẽ khó mà thực hiện được.

Collaborative Filtering là lựa chọn phù hợp hơn cả và cụ thể là Item-based Collaborative Filtering. Và thuật toán được tham khảo từ bài báo: “TencentRec: Real-time Stream Recommendation in Practice”.

### 2.4.3 Item-based Collaborative Filtering. (ICF)

ICF bao gồm hai thành phần: item similarity computation và prediction computation.

Item similarity computation:

$$sim(i_p, i_q) = \frac{\vec{i}_p \cdot \vec{i}_q}{\|\vec{i}_p\| \|\vec{i}_q\|} = \frac{\sum_{u \in U} r_{u,p} r_{u,q}}{\sqrt{\sum r_{u,p}^2} \sqrt{\sum r_{u,q}^2}}$$

Giả sử trong hệ thống ta có  $n$  users và  $m$  items. Ta có một ma trận  $R$  có kích thước  $n * m$  là ma trận thể hiện mức độ quan tâm, đánh giá của người dùng. Trong đó,  $r_{p,q}$  thể hiện mức độ quan tâm của user  $u_p$  trên item  $i_q$ . Mỗi item có thể được thể hiện bằng một

vector đại diện cho sự quan tâm của  $n$  users,  $i_q = \langle r_{1,q}, r_{2,q}, \dots, r_{n,q} \rangle$ . Với tập  $U$  là tập hợp các users. Nếu user  $u_p$  chưa thực hiện đánh giá lên item  $i_q$  thì  $r_{p,q}$  được cho là bằng 0.

Và ở đây ta dùng cosine similarity để thực hiện tính toán.

Prediction computation:

$$\hat{r}_{u,p} = \frac{\sum_{i_q \in N^k(i_p)} sim(i_p, i_q) r_{u,i_q}}{\sum_{i_q \in N^k(i_p)} sim(i_p, i_q)}$$

Để thực hiện tính toán mức độ quan tâm mà ta có thể dự đoán được cho một user cụ thể, ta thực hiện tính toán theo công thức như trên. Trong đó, các items trong tập  $N^k(i_p)$  là các item có độ tương đồng cao nhất với item  $i_p$ .

Tuy nhiên ta có một số điều chỉnh về việc tính toán similarity giữa các item với nhau.

$$sim(i_p, i_q) = \frac{\sum_{u \in U} \min(r_{u,p}, r_{u,q})}{\sqrt{\sum r_{u,p}} \sqrt{\sum r_{u,q}}}$$

Vì ta thay đổi phần tử của công thức, do đó dẫn đến việc ta định nghĩa lại thành phần mẫu. Ở đây, ta không còn dùng chuẩn L2-norm nữa.

Từ đây, ta tách các thành phần ra như sau:

$$itemCount(i_p) = \sum r_{u,p}$$

$$pairCount(i_p, i_q) = \sum_{u \in U} co-rating(i_p, i_q)$$

$$co-rating(i_p, i_q) = \min(r_{u,p}, r_{u,q})$$

Và ta được công thức như sau:

$$sim(i_p, i_q) = \frac{pairCount(i_p, i_q)}{\sqrt{itemCount(i_p)} \sqrt{itemCount(i_q)}}$$

Cách tính similarity này dựa vào ba thành phần là  $pairCount(i_p, i_q)$ ,  $itemCount(i_p)$  và  $itemCount(i_q)$ . Ba thành phần nói trên có thể được tính toán độc lập với nhau, và giá trị similarity sẽ là sự kết hợp giữa ba giá trị này với nhau. Và theo một

cách trực quan thì các thành phần này sẽ theo xu hướng tăng dần. Do đó quá trình cập nhật giá trị similarity mới sẽ được tiến hành như sau:

$$\begin{aligned} sim(i_p, i_q)' &= \frac{pairCount(i_p, i_q)'}{\sqrt{itemCount(i_p)'} \sqrt{itemCount(i_q)'}} \\ &= \frac{pairCount(i_p, i_q) + \Delta co-rating(i_p, i_q)}{\sqrt{itemCount(i_p) + \Delta r_{u_p}} \sqrt{itemCount(i_q) + \Delta r_{u_q}}} \end{aligned}$$

Khi cập nhật giá trị mới, ta chỉ xem xét đến  $\Delta co-rating(i_p, i_q)$ ,  $\Delta r_{u_p}$  và  $\Delta r_{u_q}$  dựa trên việc cập nhật các sự kiện.

Real-time Pruning:

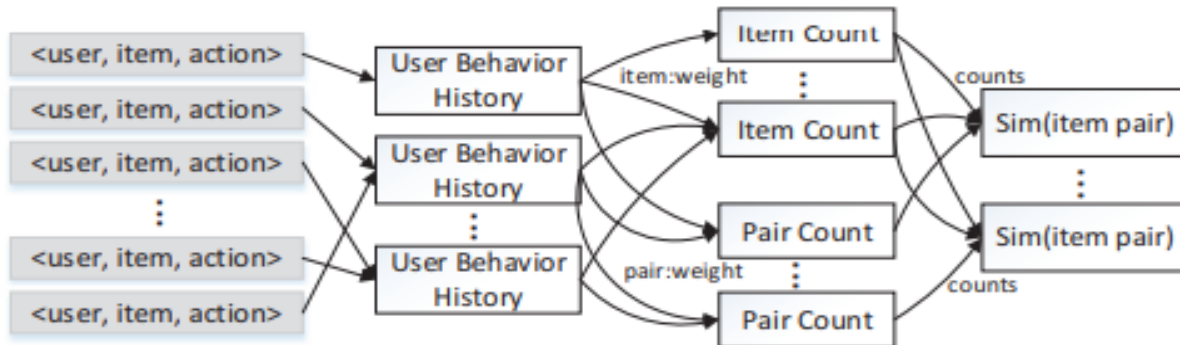
Một vấn đề khác trong hệ thống recommendation system là số lượng các cặp item-item theo thời gian sẽ tăng lên thành một khối lượng khổng lồ, dẫn đến sự tăng đột biến về khối lượng tính toán. Vì vậy, việc cắt bỏ bớt là một điều rất cần thiết. Để giải quyết vấn đề, ta dùng Hoeffding bound theory.

Với  $x$  là một giá trị ngẫu nhiên,  $0 \leq x \leq R$ . Giả sử ta có  $n$  quan sát trên biến này, và tính được giá trị trung bình là  $\hat{x}$ . Với xác suất  $1 - \delta$ , thì giá trị trung bình thực sự của biến sẽ có giá trị lớn nhất là  $\hat{x} + \epsilon$ , với

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

Dựa vào đó, với các cặp item-item có một số lượng cập nhật giá trị nhất định ta sẽ có được một giá trị similarity ước tính mà các cặp item-item tối thiểu phải đạt được. Nếu không vượt quá giá trị đó thì cặp item-item sẽ bị loại bỏ. Do đó, nếu số lượng user tương tác càng cao sẽ giúp cho ta lọc được các cặp item-item nào có sự liên quan với nhau nhất.

#### 2.4.4 Tiến hành



Về phần tính toán similarity computation, gồm User Behavior History, Item Count, Pair Count và Sim(item pair). User Behavior History sẽ ghi nhận lại các event (user, item, action), dựa vào các giá trị này ta sẽ tính toán rating và corating mới. Bên cạnh đó, ta cũng sẽ lưu các giá trị cũ trước đó và sẽ thực hiện so sánh để thực hiện cập nhật cho Item Count và Pair Count các giá trị mới. Sau đó sẽ tính toán Sim(item pair).

---

**Input:** user rating action recording user  $u$  and item  $i$

```

1 Get  $L_i$ 
2 for each item  $j$  rated by user  $u$  do
3   if  $j$  in  $L_i$  then
4     | Continue
5   end
6   Update pairCount( $i, j$ )
7   Get itemCount( $i$ ) and itemCount( $j$ )
8   Compute  $sim(i, j)$  using Equation 5
9   Increment  $n_{ij}$ 
10  Get threshold  $t_1$  of  $i$ 's similar-items list
11  Get threshold  $t_2$  of  $j$ 's similar-items list
12   $t = \min(t_1, t_2)$ 
13  Compute  $\epsilon$  using Equation 9
14  if  $\epsilon < t - sim(i, j)$  then
15    | Add  $j$  to  $L_i$ 
16    | Add  $i$  to  $L_j$ 
17  end
18 end

```

---

Quá trình thực hiện real-time pruning được tích hợp trong quá trình tính toán trên, theo các bước minh họa trên. Trong đó,  $L_i$  là tập gồm các item đã được loại bỏ và được cho rằng là không có liên qua tới item  $i$ .

### 3 THỰC NGHIỆM DEMO

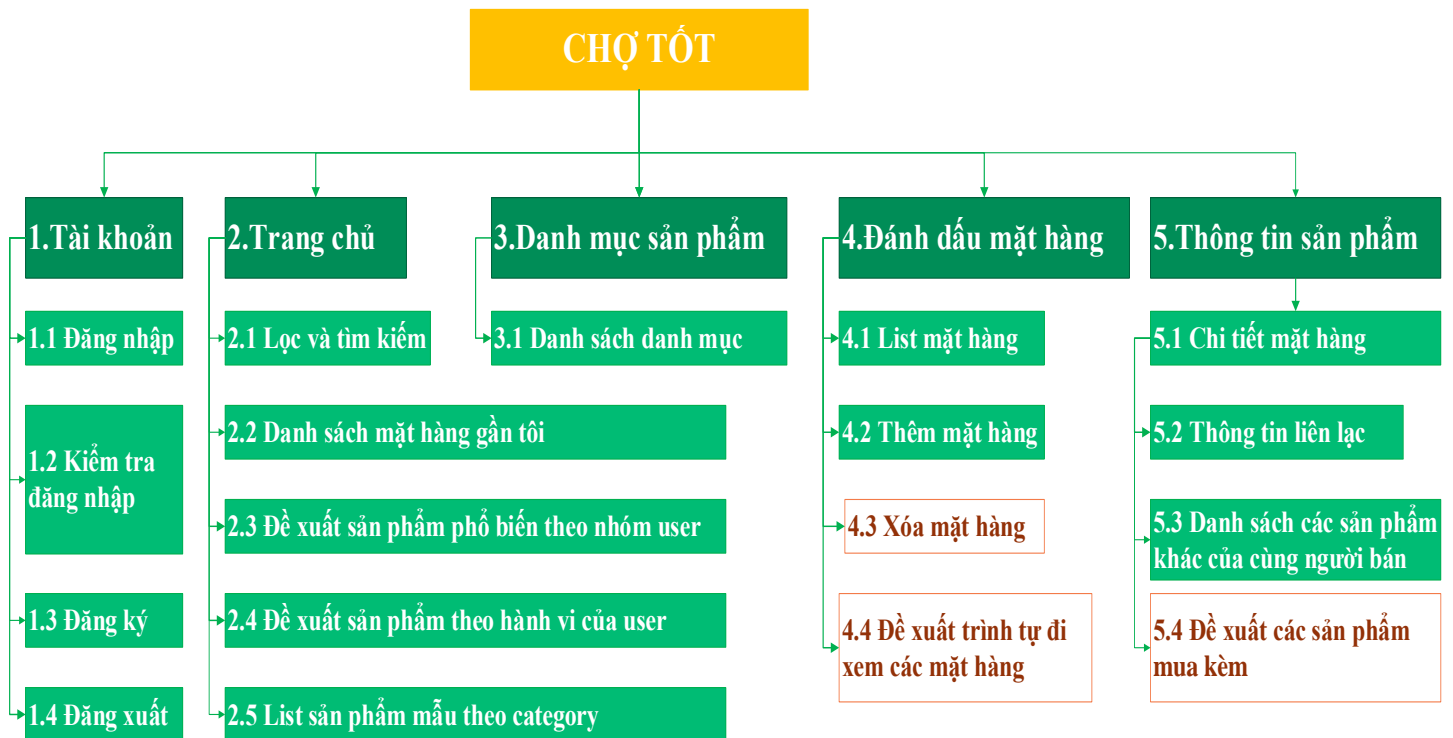
Link video demo:

[https://drive.google.com/drive/folders/1Q0\\_s\\_gG20P0lpzQQbVTgJ9nkB2PPWuZZ?usp=sharing](https://drive.google.com/drive/folders/1Q0_s_gG20P0lpzQQbVTgJ9nkB2PPWuZZ?usp=sharing)

### 4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

#### 4.1 Kết luận: những gì đã làm được

- Những gì đạt được:
  - ✓ Hoàn thành được 15/18 tính năng (những tính năng được tô màu) đề ra.
  - ✓ Không có thành viên nào bỏ cuộc giữa chừng



- **Khó khăn:**
  - ✓ Nhiều kiến thức mới, team còn non trẻ
  - ✓ Thời gian ngắn
  - ✓ Tài nguyên phân cứng
  - ✓ Nguồn tiếp cận đến dữ liệu còn hạn chế

## 4.2 Hướng phát triển

- Tiếp tục hoàn thiện hơn tính năng Recommendation System
- Bổ xung các tính năng còn thiếu như: cho phép đặt đặt hàng, cho phép dùng thử sản phẩm, thanh toán online và giao hàng tận nhà.