BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

NGÔ VĂN HẬU

# PHÁT TRIỂN ỨNG DỤNG MẠNG XÃ HỘI ĐẤU GIÁ TỪ THIỆN DỰA TRÊN REACT NATIVE

KHÓA LUẬN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

TP. HỒ CHÍ MINH, 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

NGÔ VĂN HẬU

# PHÁT TRIỂN ỨNG DỤNG MẠNG XÃ HỘI ĐẤU GIÁ TỪ THIỆN DỰA TRÊN REACT NATIVE

Mã số sinh viên: 1851050043

KHÓA LUẬN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn: Ths. Dương Hữu Thành

TP. HỒ CHÍ MINH, 2021

TRƯỜNG ĐẠI HỌC MỞ **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

THÀNH PHỐ HỒ CHÍ MINH **Độc lập – Tự do – Hạnh phúc**

**KHOA CÔNG NGHỆ THÔNG TIN**

———

# GIẤY XÁC NHẬN

Tôi tên là: Ngô Văn Hậu.....

Ngày sinh: 02-01-2000..... Nơi sinh: Đồng Nai.....

Chuyên ngành: CNTT..... Mã sinh viên: 1851050043.....

Tôi đồng ý cung cấp toàn văn thông tin đồ án/ khóa luận tốt nghiệp hợp lệ về bản quyền cho Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh. Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh sẽ kết nối toàn văn thông tin đồ án/ khóa luận tốt nghiệp vào hệ thống thông tin khoa học của Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh.

Ký tên

*(Ghi rõ họ và tên)*

Ngô Văn Hậu

**Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN/ KHÓA LUẬN TỐT NGHIỆP**

**CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Giảng viên hướng dẫn: Dương Hữu Thành.....**

**Sinh viên thực hiện: Ngô Văn Hậu..... Lớp: DH18IT02.....**

**Ngày sinh: 02-01-2000..... Nơi sinh: Đồng Nai.....**

**Tên đề tài: Xây dựng ứng dụng mạng xã hội đấu giá từ thiện bằng React Native…**

**.....**

**.....**

**.....**

**Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án/ khóa luận trước Hội đồng:**

**.....**

**.....**

**.....**

**.....**

**.....**

**.....**

**.....**

**.....**

**.....**

*Thành phố Hồ Chí Minh, ngày ... tháng ... năm ……*

Người nhận xét

……

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Dương Hữu Thành là giảng viên của trường đại học Mở TP Hồ Chí Minh. Từ lúc em bắt đầu vào ngôi trường cho đến bài đồ án này thì thầy đã rất quan tâm và hỗ trợ giúp em rất nhiệt tình và tâm huyết. Để đạt được ngày hôm nay thì sự hướng dẫn và định hướng của thầy đã giúp em soi sáng và tìm được con đường đúng đắn để đi. Em xin cảm ơn những lời khen quan tâm và sự động viên của thầy trong lúc em làm trong môi trường doanh nghiệp. Và cũng nhờ có thầy mà em đã có được cơ hội đó, một công việc theo ngành đầu đời của em.

Cho đến nay bài đồ án này em xin được phép trú tâm và áp dụng những kiến thức mà được thầy dạy cho đến bây một cách đúng đắn và phù hợp nhất. Dù là bài làm cá nhân nhưng có thêm sự hỗ trợ của thầy sẽ giúp em hoàn thiện nó hơn. Em sẽ cố gắng để đạt được điểm số cao nhất so với những môn khác mà thầy từng dạy. Và một sản phẩm hoàn thiện một bài báo cáo chỉnh chu sẽ chứng minh được điều đó. Sau tất cả em luôn mang trong lòng một sự kính trọng, một lòng biết ơn sâu sắc đến thầy ạ. Em cảm ơn người thầy yêu quý, thầy Thành.

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# TÓM TẮT KHÓA LUẬN

Với mục đích giúp cho xã hội có một môi trường chia sẻ thông tin những hoàn cảnh khó khăn cần sự giúp đỡ và sử dụng với mục đích đấu giá từ thiện, mạng xã hội Kanj Haungo đã được ra đời. Với các tính năng chính là tạo bài viết chia sẻ, đấu giá các sản phẩm với mục đích từ thiện.

Kanj Haungo là một ứng dụng có thể chạy trên nền điện thoại thông minh như Android và IOS. Nhờ ứng dụng được xây dựng bởi React Native nên có thể tạo một lần chạy được trên hai hệ điều hành phổ biến nhất hiện nay này. Về phần Backend là sự kết hợp của framework Django và cơ sở dữ liệu Mysql.

Thêm vào đó hệ thống Kanj Haungo còn có phần giúp cho Admin có thể dễ dàng kiểm soát những bài đăng bị báo cáo vi phạm, hay những bài đăng có lượt tương tác cao từ đó định ra các chính sách và chiến lược phát triển hợp lý.

Các thông tin về hệ thống cũng như là cách thức hoạt động cũng sẽ được trình bày chi tiết tại mục hệ thống mạng xã hội Kanj Haungo. Từ đó giúp cho việc hình dung cũng như thể hiện được ứng dụng một cách rõ ràng.

# MỤC LỤC

# DANH MỤC TỪ VIẾT TẮT

RNB - React Native Bridge

LM - Lifecycle Methods

URL - Uniform Resource Locator

JSON – Javascript Object Notation

UI - User Interface

SDK - Software Development Kit

# DANH MỤC HÌNH VẼ

# DANH MỤC BẢNG

# MỞ ĐẦU

Trong tình hình dịch bệnh khó khăn như ngày nay thì việc được tiếp tế và kêu gọi sự giúp đỡ của những người xung quanh là vô cùng cần thiết, cũng vì điều đó mà hệ thống mạng xã hội Kanj Haungo được xây dựng. Nhằm đến những mục đích từ thiện mỗi khi gặp khó khăn người dùng có thể đăng vấn đề của mình lên để nhờ gọi sự giúp đỡ. Đồng thời người làm từ thiện cũng dễ dàng tiếp đúng lúc đúng đối tượng.

Ngoài ra hệ thống còn cung cấp chức năng đấu giá với mục đích từ thiện. Từ đó giúp người làm từ thiện và mạnh thường quân có thể tương tác giúp đỡ lẫn nhau trong việc làm từ thiện. Những thành quả chương trình của họ cùng nhau tạo ra sẽ thiết thực an tâm hơn.

Về hệ thống, một sản phẩm tốt đồng nghĩa với việc đó là một sản phẩm đẹp về hình thức, phù hợp với người dùng và hơn cả đó là sự an toàn, thân thiện, đúng đắn mới mục đích đã được định ra trước đó. Vì vậy việc kết hợp giữa bộ ba React Native, Django, Mysql là một trong những sự lựa chọn phù hợp và tin cậy.

Về mặt ứng dụng hệ thống Kanj Haungo được lập trình phù hợp để có thể chạy suôn sẻ trên hai hệ điều hành chính hiện nay là Android và IOS. Với các tính năng được thiết kế dễ hiểu dễ sử dụng, cách hoạt động đơn giản nhằm hướng đến hai đối tượng chính là người đi làm từ thiện và các mạnh thường quân.

Hệ thống cũng là một trong những sản phẩm góp phần thúc đẩy sự phát triển của xã hội, tạo ra các giao dịch công bằng và công chính hơn.

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Chương này sẽ sơ lược về một số điểm nhấn của đề tài cũng như là những thông tin cần thiết có liên quan đề tài của dự án.

## 1.1 Giới thiệu đề tài

Trong một năm trở lại đây thì chúng ta luôn phải chống chọi lại với dịch bệnh covid-19 cũng như những hệ lụy của nó một cách mệt mỏi và tốn kém cả người lẫn vật chất. Với những mảnh đời còn khó khăn thì việc đói khát trong những ngày này là không khỏi tránh được, vì vậy rất cần sự giúp đỡ từ những người xung quanh. Từ ý nghĩ đó chúng ta rất cần một sản phẩm có thể chia sẻ và giúp gắn kết mọi người với nhau hơn, cũng như là tuyên truyền những chính sách những cách làm đúng đắn trong mùa dịch bệnh.

Với sự phát triển cao của công nghệ như hiện nay thì việc chia sẻ thông tin đã trở nên dễ dàng hơn hết. Vì vậy cũng cần một sản phẩm tin cậy và an toàn để người dùng có thể an tâm và sử dụng. Và một mạng xã hội từ thiện sẽ giúp cho các bài đăng được đưa lên nhằm chia sẻ trạng với mọi người, một phần khác dùng để tạo các bài đấu giá sản phẩm định hướng đến mục đích từ thiện. Để đạt được điều đó thì việc áp dụng công nghệ mới như React Native để sản phẩm được phát triển nhanh và sử dụng được trên cả Android và các dòng IOS của Apple là một giải pháp phù hợp.

## 1.2 Lý do chọn đề tài

Một trong những lý do quan trọng mà em chọn đề tài này là nó rất phù hợp với tình hình hiện nay của đất nước ta. Chúng ta cần lan tỏa những điều tốt đẹp, chia sẻ khó khăn và khai thác việc đấu giá nhằm mục đích từ thiện. Từ đó cũng giúp cho những mạnh thường quân và những người đi làm từ thiện được kết nối với nhau, cùng làm những hành động tốt đẹp giúp đỡ những mảnh đời khó khăn.

Bên cạnh đó thì việc viết một chương trình mạng xã hội từ thiện cũng là nằm trong khả năng của mình. Đây cũng là một đề tài viết chương trình ứng dụng có nhiều sự tham khảo và tài liệu trên internet.

## 1.3 Mục tiêu nghiên cứu

Một ứng dụng có thể chạy trên cả hai hệ điều hành được nhiều người sử dụng nhất hiện nay là Android và IOS là một trong những thách thức và cũng là lý do React Native được em sử dụng trong ứng dụng Kanj Haungo này. Đây là một framework phù hợp và có tính mới mẻ trong lĩnh vực công nghệ những năm gần đây. Và cũng là một trong những mục tiêu nghiên cứu của em.

Bên cạnh đó việc sử dụng Django để làm server cho app Kanj Haungo cũng có nhiều điểm ưu việt. Django hỗ trợ việc chứng thực người dùng với OAuth2 một giải pháp an toàn và tin cậy được nhiều các công ty lớn sử dụng. Bên cạnh đó còn giúp cho chúng dễ dàng thao tác với dữ liệu hơn nhờ có công nghệ ORM trong Django. Và đây cũng là điểm nhấn mạnh mẽ của framework này.

## 1.4 Phạm vi nghiên cứu, đối tượng sử dụng

Phạm vi nghiên cứu tập trung liên quan đến những kiến thức có trong việc xây dựng ứng dụng React Native phiên bản mới nhất hiện nay. Và nền tảng công nghệ Django. Với mỗi nền tảng thì nghiên cứu về cách thức hoạt động và cách ứng dụng để triển khai chúng sao cho phù hợp nhất.

Hệ thống mạng xã hội từ thiện Kanj Haungo hướng đến hai đối tượng chính là những người đang có hoạt động từ thiện và các mạnh thường quân. Nhằm giúp liên kết, trao đổi trực tiếp với nhau và cùng góp sức vào mục đích chung là từ thiện. Giúp những phần quà được trao đến những hoàn cảnh trực tiếp và ít thủ tục hơn.

## 1.5 Bố cục đề tài

Để cho việc dễ dàng tiếp cận và phân tách các mục rõ ràng thì bố cục của đồ án gồm những chương sau.

Chương một để giới thiệu về đề tài giúp người đọc hứng thú và trình bày những điểm quan trọng và các lý do xoay quanh chúng.

Chương hai trình bày những cơ sở lý thuyết liên quan đến các công nghệ được sử dụng và giải thích chúng dựa trên những kiến thức nền về công nghệ giúp cho người đọc hiểu rõ hơn về các nền tảng ứng dụng.

Chương ba trình bày về hệ thống mạng xã hội từ thiện Kanj Haungo, ở chương này tập chung các kiến thức chuyên ngành liên quan đến mã nguồn và giải thích các chức năng, cách hoạt động của ứng dụng.

Chương bốn là chương viết những kết luận và hướng phát triển của dự án cho sau này. Giúp người đọc hình dung được tiềm năng cũng như là những khuyết điểm hiện tại của ứng dụng.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

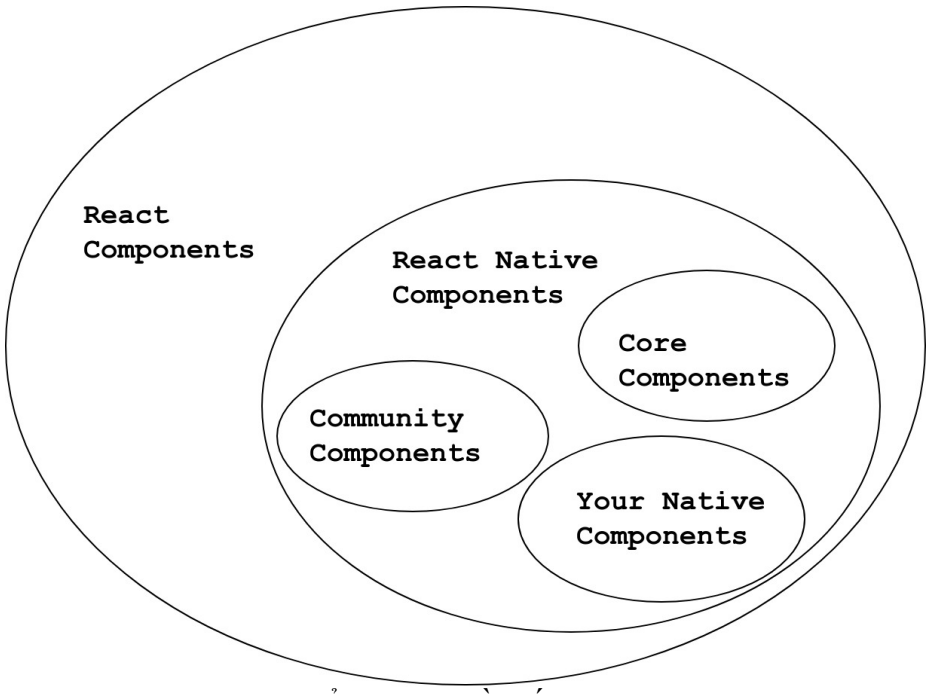
## 2.1 Giới thiệu React Native

React Native là một trong những framework mã nguồn mở nổi tiếng và được sử dụng nhiều gần đây là do sản phẩm sau khi hoàn thiện có thể chạy trên được cả hai hệ điều hành điện thoại lớn là Android và IOS. Ngoài ra các logic xử lý và cú pháp cũng có rất nhiều điểm giống với Reactjs nên việc chuyển đổi từ một web app Reactjs thành một ứng dụng điện thoại cũng dễ dàng. Từ đó giúp giảm rất nhiều về chi phí khi xây dựng một ứng dụng điện thoại.

React Native sử dụng ngôn ngữ Javascript để viết về mặt logic cũng như là về giao diện ứng dụng bằng cách sử dụng các Component. Việc này cũng giúp cho quá trình phát triển có thể tái sử dụng lại những thứ đã viết. Làm cho việt phát triển và bảo trì ứng dụng dễ dàng hơn.

## 2.2 Kiến trúc của React Native

Một trong những thành phần bắt buộc phải tìm hiểu khi sử dụng React Native là kiến trúc của nó. bởi việc xây dựng và chạy được trên cả hai nền tảng thì điều này là vô cùng quan trọng. Trong đó React native cơ bản chia làm hai phần chính quan trọng là Core Components và Native Components.




Hình 2.2.1: Tổng quan về kiến trúc của React Native

### 2.2.1 View

Trong việc phát triển các ứng dụng điện thoại thì View là một trong những thành phần cơ bản của UI. Nó giúp hiển thị các Text, Image và những sự kiện Input của người dùng và thậm chí những thành phần nhỏ hơn như Button hoặc các loại List hiển thị các thành phần. Một trong số chúng có thể kết hợp, bao hàm lẫn nhau giúp cho việc sử dụng được linh hoạt và thiết kế được nhiều mẫu hơn.

### 2.2.2 Native Components

Khi lập trình ứng dụng trên các nền tảng khác nhau, thì chúng ta phải sử dụng các ngôn ngữ tương ứng để viết giao diện như Kotlin, Java trên Android hay Swift, Objective-C trên IOS. Với React Native, Components được viết bằng Javascript sẽ đảm nhiệm cho cả việc hiển thị, kết xuất UI trên cả Android và IOS [1]. Hai View sau khi được tạo ra trên hai nền tảng giống nhau về giao diện, hiệu ứng và cả hiệu suất của của chúng cũng không kém so với những ứng dụng Native khác. Những Components này còn được gọi là Native Components.

Việc kết hợp và sử dụng các Native Components sẽ là bước đầu tiên của ta trong việc xây dựng ứng dụng và chúng được gọi là các Core Component của React Native.

React Native nổi tiếng và được sử dụng rộng rãi nhờ đó bên cạnh các Native Component thì chúng ta còn có thể dụng dụng các Component do cộng đồng phát triển và đóng góp (Community-Contributed-Components).

### 2.2.3 Core Components

Cũng như cách mà chúng ta xây dựng UI trên các nền tảng khác thì việc chia ra các loại thẻ tương ứng với từng chức năng đã là điều vô cùng hiển nhiên và React Native cũng vậy. Có rất nhiều thành phần Core Components với các chức năng khác nhau và cơ bản nhất phải kể đến là 5 thành phần View, Text, Image, ScrollView, TextInput.

Tương ứng có cùng chức năng trên các nền tảng khác ta có bảng so sánh sau đây.

Bảng 2.2.1: So sánh các loại thẻ tương ứng trên các nền tảng khác nhau

| **React Native UI Component** | **Android View** | **iOS View** | **Web Analog** |
|---|---|---|---|
| <View> | <ViewGroup> | <UIView> | A non-scrollling  |
| <Text> | <TextView> | <UITextView> | <p> |
| <Image> | <ImageView> | <UIImageView> | <img> |
| <ScrollView> | <ScrollView> | <UIScrollView> |  |
| <TextInput> | <EditText> | <UITextField> | <input type="text"> |

## 2.3 Cơ chế hoạt động của React Native

Một điểm quan trọng trong cách thức hoạt động của React Native làm nó khác với những framework xây dựng app chạy trên nền webview là React Native có quyền truy cập toàn bộ đến Native APIs và giao diện được xây dựng trên nền của hệ điều hành. Vì vậy chúng ta có thể cảm nhận được hiệu năng như những ứng dụng Native.

Ta cũng chắc chắn rằng React Native được biên dịch từ ngôn ngữ Javascript thành ngôn ngữ máy. Và JavascriptCore (máy ảo Javascript) có sẵn và được cung cấp trên của hai nền tảng Android và IOS sẽ nhận nhiệm vụ biên dịch này. React Native Bridge (RNB) là cầu nối được sử dụng để đảm nhận việc giao tiếp giữa các luồng xử lý Native và Javascript.

Mỗi khi chúng ta chạy app đang phát triển trên nền tảng điện thoại, bộ React Native Cli sẽ đóng gói các Components thành thành một file duy nhất. Và khi app được mở lên, luồng Native (Code gốc trên IOS hoặc Android) sẽ gửi tín hiệu thông qua RNB để khởi động app. JavascriptCore sau khi nhận tín hiệu sẽ chạy các dòng code được đóng gói trước đó. Những dòng code này là logic ban đầu của ứng dụng. Và ngược lại sau khi đọc code JavascriptCore muốn kết xuất giao diện thì sẽ gửi tín hiệu thông qua RNB đến luồng Native để kết xuất chúng.

## 2.4 Các Core Components cơ bản trong React Native

React Native có rất nhiều Core Component để thực hiện các chức năng từ quản lý Form đến thể hiện các chỉ số hoạt động. Trong đó TextInput, ScrollView, ListVIew là ba thành phần cơ bản định nên khung xương và phần nào thể hiện được cách hoạt động của React Native.

### 2.4.1 TextInput

TextInput là một trong những Core Components quan trọng nó cho phép người dùng đưa dữ liệu dạng text vào ô nhập liệu. Theo với đó là các loại sự kiện tương tác kèm theo. Thuộc tính onChangeText nhận vào một hàm và hàm này sẽ được gọi vào mỗi khi dữ liệu trong ô bị thay đổi.

```
<TextInput
        style={{height: 40}}
        placeholder="Enter your name!"
        onChangeText={text => setText(text)}
        defaultValue={text}
/>
```

### 2.4.2 ScrollView

ScrollView là một Components thường để bao hàm nhiều các Component con và các View. Và khi các nội dung bên trong bị tràn ra khỏi màn hình thì ta có thể cuộn theo cả chiều ngang và chiều dọc (mặc định là theo chiều dọc).

### 2.4.3 List View

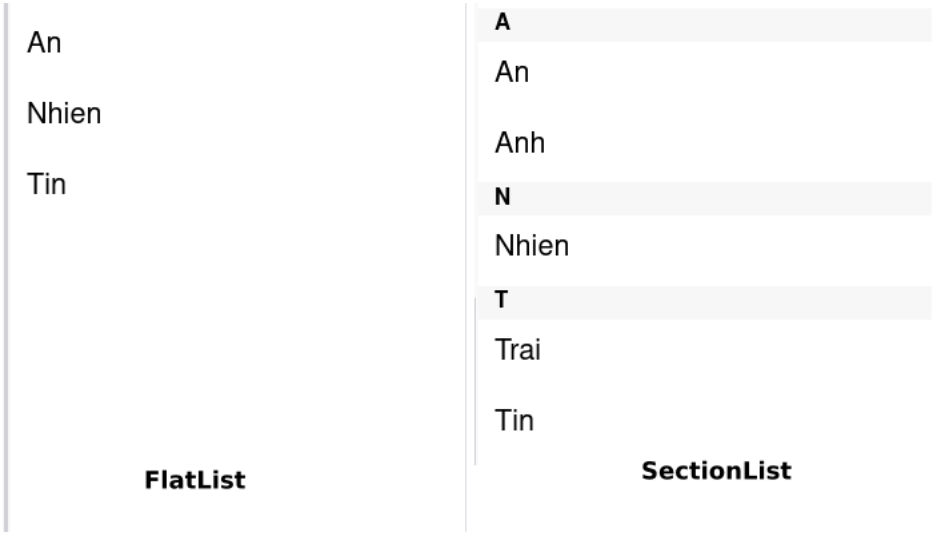
Khi nhu cầu muốn hiển thị dữ liệu dưới dạng danh sách thì chúng ta có thể sử dụng một trong hai Core Component là FlatList hoặc SectionList.

FlatList là Components dùng để hiển thị danh sách với mỗi dòng dữ liệu có cấu trúc đơn giản. Thường hay dùng để hiển thị một danh sách dài, có thể thay đổi theo thời gian và chỉ kết xuất những dòng dữ liệu nào được hiển thị trên màn hình.

FlatList Component bắt buộc hai thuộc tính cần truyền vào là data và renderItem. Trong đó data là thuộc tính nhận dữ liệu danh sách, renderItem nhận một hàm đầu vào là dữ liệu một phần từ trả về một Component để kết xuất.

```
<FlatList
        data={[{key: 'Van A'}, {key: 'Van B'}]}
        renderItem={({item}) => <Text>{item.key}</Text>}
/>
```

Khác với FlatList, SectionList là một Component hiển thị một danh sách cần tựa đề mỗi vùng.




Hình 2.4.1: Sự khác nhau của FlatList và SectionList

## 2.5 Các thành phần quan trọng trong một Component của React Native

Một Component thường được xây dựng trên nhiều Core Component, và mỗi Component thường có những dữ liệu cũng như là logic riêng để đảm nhiệm một mục đích cụ thể. Và Props, State là những thành phần quan trọng để giúp Component thực hiện được mục đích đó.

### 2.5.1 JSX

JSX là một cú pháp được React Native sử dụng để viết lên những phần tử UI đồng thời còn có thể truyền các biến, xử lý điều kiện và truyền các hàm sự kiện của Javascript vào trong giúp cho việc kết xuất được hợp lý tiện lợi hơn.

```
const Student = () => {
  const name = "Hau Ngo";
  return <Text>Hi, I am {name ? name : "a student "}!</Text>;
}
```

### 2.5.2 Props

Props là viết tắt của chữ "properties", đây là biến **chỉ đọc** cho phép truyền các dữ liệu từ Component cha xuống cho các Component con.

Mỗi lần Props thay đổi thì Component con sẽ kết xuất lại một lần, cho nên luôn đảm bảo tính đúng đắn về dữ liệu.

### 2.5.3 State

State là một nơi để lưu trữ dữ liệu cục bộ của một Component, và có thể thay đổi được trong suốt quá trình chạy. Mỗi khi state thay đổi Component lại kết xuất một lần để đảm bảo dữ liệu của biến luôn được cập nhật ở giao diện.

```
const Cat = ({ name }) => {
  const [isClick, setIsClick] = useState(false);
  return <Button onPress={() => {setIsClick(false)}}
        title={isClick ? "Click me" : "Clicked"}/>
}
```

## 2.6 Cách viết một Component

Từ những phiên bản React Native cũ hơn 0.59 chỉ hỗ trợ tạo component bằng Class, nhưng từ các bản React Native sau này thì chúng ta có thể viết Component bằng hai kiểu là Class Component và Function Component.

### 2.6.1 Class Component

Chúng ta phải sử dụng cú pháp Class trong javascript để kế thừa Component của React Native. Và bắt buộc phải override lại phương thức render, phương thức này phải trả về một React Component.

```
import React, { Component } from 'react';
import { Text, View } from 'react-native';
class HelloWorldComponent extends Component {
  render() {
    return (<View><Text>Hello world!</Text></View>);
  }
}
export default HelloWorldComponent;
```

### 2.6.2 Function Component

Function Component là một cách thay thế Class Component để giúp cho việc tạo nhanh hơn, viết code ngắn gọn hơn. Tương tự ta có phần trả về của hàm sẽ tương đương với trả về của phương thức render trong Class Component.

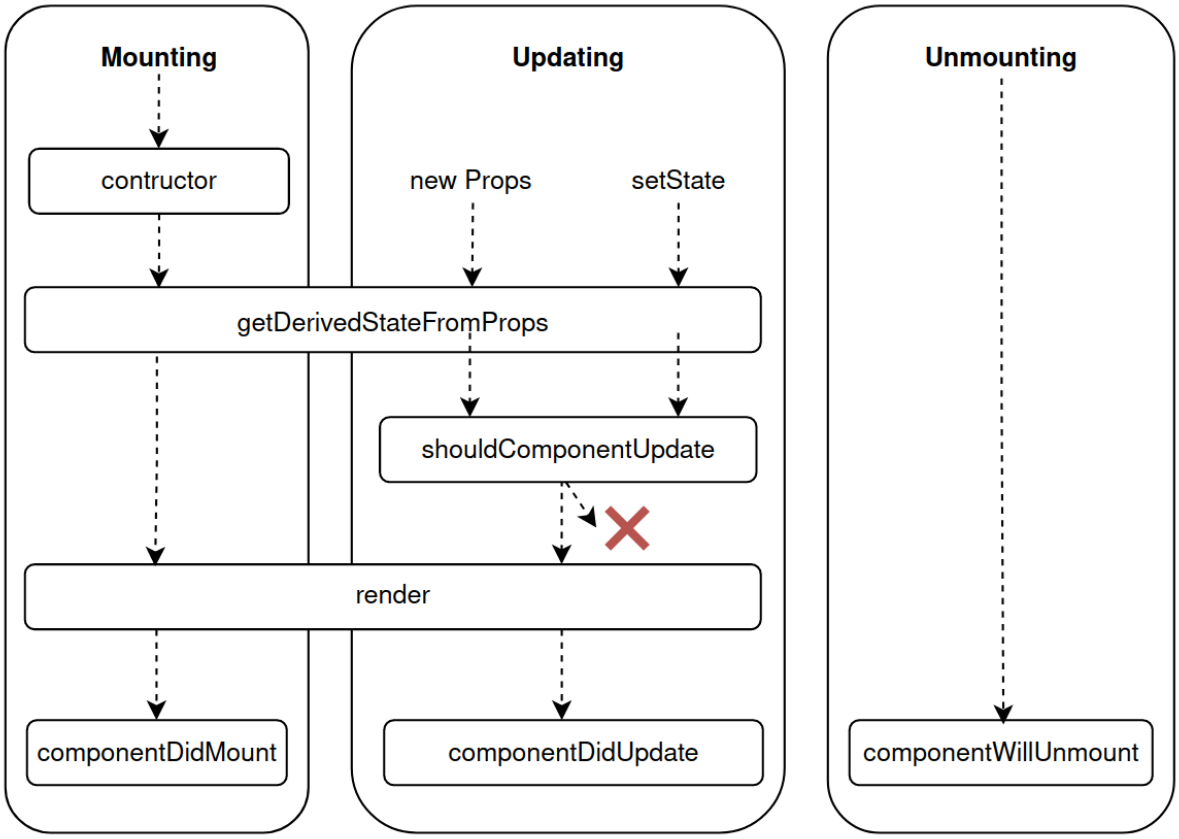
```
import React from 'react';
import { Text, View } from 'react-native';
const HelloWorldComponent = () => {
  return (
    <View><Text>Hello world!</Text></View>
  );
}
export default HelloWorldComponent;
```

## 2.7 Vòng đời của một Component

Từ khi một Component được chạy cho đến khi được kết xuất, được cập nhật lại và cuối cùng là bị gỡ khỏi View, thì chúng qua đi qua hết một vòng đời của của một Component. Gồm có ba giai đoạn chính là Mounting, Updating và Unmounting. Trong đó hai giai đoạn Mounting và Unmounting chỉ chạy duy nhất một lần còn Updating thì có thể được chạy nhiều lần để cập nhật lại giao diện.

Khi một Component được tạo ra thì hàm constructor(props) sẽ được chạy đầu tiên, và có vai trò khởi tạo các state hay gán giá trị của props. Hàm render() là một phương thức bắt buộc phải có của mỗi Class Component vì phần trả về của nó sẽ được kết xuất ra giao diện của Component.

Trong mỗi giai đoạn của một Vòng đời Component sẽ có một số hàm đặc biệt được gọi tự động và chúng ta gọi đó là các Lifecycle Methods (LM).




Hình 2.7.1: Vòng đời của một Component

### 2.7.1 Mounting

Mounting là quá trình khi component được tạo ra vào kết xuất lên View. Lúc này người ta đã có thể nhìn thấy giao diện là một React Component trong phần trả về của hàm render ở Class Component hoặc đối với Function Component là phần trả về của hàm.

Trong quá trình này chúng ta có một LM chính là componentDidMount hàm này sẽ được gọi sau khi giao diện đã được kết xuất lên màn hình. Thường dùng để chạy api lấy dữ liệu về hoặc tạo các hàm lắng nghe sự kiện.

### 2.7.2 Updating

Sau khi một component được render thì những lần sau đó nếu state hay props của Component đó thay đổi thì Component sẽ được re-render lại. Và quá trình này chúng ta gọi là Updating hay nói cách khác là Component bị thay đổi và cập nhật lại giao diện người dùng.

Trong quá trình này chúng ta có 2 LM đi kèm là shouldComponentUpdate và componentDidUpdate, shouldComponentUpdate này cho phép người dùng quyết định

có nên cập nhật lại giao diện hay không và trả về kiểu boolean. Còn componentDidUpdate sẽ chạy ngay sau khi Component đó update xong.

### 2.7.3 Unmounting

Một Component nào đó được gỡ bỏ khỏi giao diện (người dùng không thấy trên màn hình nữa) thì người ta gọi là quá trình Unmounting. Trong quá trình này người ta thường hay xóa các lắng nghe sự kiện để giải phóng bộ nhớ trong componentWillUnmount. Vì componentWillUnmount sẽ chạy trước khi Component đó bị gỡ bỏ khỏi giao diện.

## 2.8 Các hook trong Function Component

Trong Class component ta có những hàm sự kiện được override lại từ Component của React Native như componentDidMount, componentDidUpdate,… Trong Function chúng ta sẽ có những Hook được xem như là những giải pháp thay thế để gọi trong vòng đời.

### 2.8.1 Hook useState

Hook useState là một trong 2 Hook được sử dụng nhiều nhất trong Function Component. Dùng để lưu trữ dữ liệu cục bộ của một Component và không bị thay đổi giá trị mỗi lần Component re-renders, useState nhận vào giá trị khởi tạo ban đầu và trả về một cặp: giá trị hiện tại của state và một hàm cho phép cập nhật lại state đó [2]. Mỗi khi giá trị của state thay đổi thì Component sẽ bị re-renders.

### 2.8.2 Hook useEffect

Hook useEffect là một Hook được sử dụng nhiều nhất trong Function Component giống như useState. Dùng để gọi tải dữ liệu từ server, đăng ký sự kiện hoặc chạy xử lý mỗi lần Component bị re-renders.

Hook useEffect được tạo ra với mục đích thay thế các LM componentDidMount, componentDidUpdate và componentWillUnmount trong Class Component. useEffect nhận vào 2 tham số, đầu tiên là một hàm và cái thứ hai là một mảng, mảng này là danh sách những biến mà khi những biến này thay đổi thì function trong useEffect sẽ được gọi, nếu để trống [] thì hàm trong useEffect chỉ chạy 1 lần duy nhất như

componentDidMount còn nếu biến này không được truyền thì hàm trong useEffect sẽ được chạy mỗi lần Component re-renders. Phần trả về của hàm trong useEffect sẽ có chức năng như của  componentWillUnmount trong Class Component.

## 2.9 Các tạo app đầu tiên với Expo

Expo là một framework và cũng là một nền tảng kết nối tạo ra môi trường cho ứng dụng React Native chạy. Bên cạnh đó Expo còn cung cấp nhiều công cụ và dịch vụ tiện ích trong việc phát triển, xây dựng và triển khai nhanh ứng dụng lên các thiết bị Android, IOS và Web applications [3]. Chúng ta có thể phát triển trên nền Javascript hoặc Typescript thì đều được Expo hỗ trợ.

Để tạo được app thì đầu tiên chúng ta phải cài đặt môi trường Nodejs và trình quản lý gói npm, sau đó thì cài đặt Expo rồi mới cài đặt được React Native app.

- ✔ Bước 1 cài môi trường và trình quản lý gói npm qua trang web chính thức của Nodejs https://nodejs.org/en/
- ✔ Bước 2 cài đặt Expo và tạo app React Native thông qua Cmd trên windows hoặc Terminal trên linux với dòng lệnh sau:

```
npm install --global expo-cli
expo init first-project
```

Vậy là với hai bước trên ta đã tạo được first-project đầu tiên của React Native, việc còn lại là chạy chúng lên bằng cách chúng ta sẽ gõ tiếp các câu lệnh sau:

```
cd first-project
expo start
```

Sau khi chạy câu lệnh trên, trình duyệt mặc định trên máy sẽ được mở với tab mới liên kết với ứng dụng. Và chúng ta chỉ cần tải app Expo trên Appstore với IOS hoặc Ch Play trên Android và quét mã QR là chúng ta sẽ hoàn tất việc cài đặt và tạo app đầu tiên của mình.

## 2.10 Redux

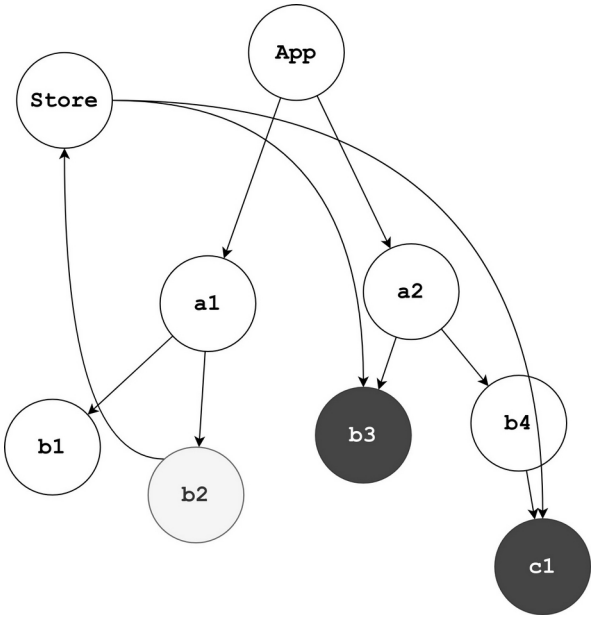
### 2.10.1 Giới thiệu

Redux là một trong những thư viện giúp chúng ta có thể lưu trữ và tương tác với các trạng thái trong app Javascript một cách đơn giản và dễ dàng hơn. Kể cả trên

những môi trường khác nhau (server, client hoặc native) thì ứng dụng vẫn hoạt động được nhất quán [4]. Và trên hết Redux cung cấp cho chúng ta bộ công cụ gỡ lỗi đầy mạnh mẽ có thể quan sát được chúng hoạt động như nào theo thời gian.

Một trong những điểm giúp Redux được sử dụng nhiều trong các app Javascript nhất là chúng ta có thể tương tác với state ở bất cứ Component nào. Từ đó giúp cho việc quản lý nhiều state dễ dàng hơn.

Cách hoạt động của Redux rất đơn giản, chúng ta sẽ lưu trữ tập trung tất cả các state ở duy nhất một store. Và có thể truy cập đến state bất kỳ mà không cần truyền qua các Component.




Hình 2.10.1: Cách hoạt động của Redux

Theo hình 2.4 ta có thể thấy nếu một state trong component b2 thay đổi thì sẽ được cập nhật lên store, sau đó những component b3, c1 có sử dụng những state này thì cũng sẽ được cập nhật theo, mà không cần phải truyền từ component b2, a1 đến App, rồi lại được truyền xuống a2, b3 và a2, b4, c1.

### 2.10.2 Các thành phần chính

Trong Redux chúng ta có 3 thành phần chính là store, action và reducers.

- ✔ Store là nơi lưu trữ tất cả các state của reducers mà đã được cấu hình từ trước.

- ✔ Action là một sự kiện có thể kèm theo dữ liệu, nó giúp kích hoạt hàm reducers thực hiện cập nhật lại state.
- ✔ Reducers là hàm giúp khởi tạo các giá trị của state và dựa vào action để biến đổi state cũ thành state mới.

## 2.11 Giới thiệu Django

Django là một trong những REST framework mạnh mẽ nhất hiện nay, được phát triển trên nền ngôn ngữ Python, kèm với đó là bộ công cụ đầy tính linh hoạt để xây dựng Web APIs. Django có số lượng người dùng đông đảo, những trang web được xây dựng trên nền tảng này thường từ trung bình đến lớn.

Một trong những ưu điểm lớn của Django là giúp sản phẩm được phát triển nhanh chóng, thiết kế có thể tái sử dụng và gọn gàng, linh hoạt. Django có bộ khung xương được phát triển bởi những nhà phát triển nhiều kinh nghiệm, vì vậy chúng ta có thể yên tâm tập trung vào việc viết ứng dụng [5]. Một điểm cộng của Django nữa là một sản phẩm mã nguồn mở nên không cần phải chi trả bất kỳ chi phí nào khi sử dụng.

Django tích hợp nhiều các công nghệ lập trình hiện đại, hỗ trợ lập trình ORM giúp cho việc viết câu truy vấn dễ dàng và không tùy thuộc vào các loại cơ sở dữ liệu. Về bảo mật Django hỗ trợ chứng thực người dùng bằng Oauth2, một trong những công nghệ chứng thực được nhiều ông lớn công nghệ sử dụng [6]. Bên cạnh đó không thể không kể đến Django có nhiều tài liệu rõ ràng và sự hỗ trợ đông đảo của cộng đồng.

### 2.11.1 Model

Model là một cách để định nghĩa một đối tượng bằng các thuộc tính, từ đó giúp chúng ta có thể quy định các hành vi và lưu trữ các thông tin thuộc tính đó vào từng trường trong cơ sở dữ liệu. Việc tạo các model trong Django sẽ tự động ánh xạ xuống cơ sở dữ liệu giúp tự động tạo các bảng với từng cột là tên thuộc tính của model. Trong Django chúng ta có thể tương tác với các đối tượng của model như thêm, sửa, xóa,…sau đó ánh xạ xuống cơ sở dữ liệu bằng các phương thức như save, delete, ….của model.

### 2.11.2 Url

Django có hỗ trợ nhiều loại thiết kế Url và không có giới hạn về một khuôn khổ nào cụ thể, để làm được điều đó thì Django cung cấp cho chúng ta module URLconf để cấu hình Url. Module này có nhiệm vụ ánh xạ các đường dẫn Url đến các hàm của Python (View) để xử lý. Các đường dẫn Url có thể ngắn hoặc dài và cũng thể thể là đường dẫn động.

## 2.12 Giới thiệu về Firebase

Firebase là một trong những dịch vụ hỗ trợ xây dựng và chạy hệ thống Backend, giúp cho việc kết nối và phát triển các sản phẩm web, app,… trở nên dễ dàng hơn.

### 2.12.1 Realtime Database

Realtime Database là một trong những dịch vụ hỗ trợ lưu trữ, đồng bộ với hệ thống NoSql cloud database, các thông tin được đồng bộ theo thời gian thực và có thể lưu trữ để sử dụng khi ngoại tuyến [7]. Các bộ dữ liệu được lưu dưới dạng JSON và đồng bộ ngay khi được kết nối với một hoặc nhiều các thiết bị client.

Bên cạnh đó việc cấu hình các quyền bảo mật cũng được hỗ trợ đầy đủ giúp ứng dụng được an toàn và tin cậy hơn. Bằng cách chỉ cho phép những người có uid trùng với đường dẫn /users/<uid> thì mới được phép ghi dữ liệu thì ta có thể cấu hình như ví dụ sau.

```
{
 "rules": {
  "users": {
   "$uid": {
    ".write": "$uid === auth.uid"
   }
  }
 }
}
```

### 2.12.2 Firebase Authentication

Firebase Authentication là một trong những dịch vụ giúp định danh và xác thực người dùng, thường được sử dụng chung với những dịch vụ khác. Dịch vụ này cung cấp cả SDKs và UI cho các nền tảng khác nhau để dễ dàng xác thực người dùng. Firebase Authentication hỗ trợ nhiều loại xác thực khác nhau như mật khẩu, số điện thoại, và một số dịch vụ liên kết khác như Facebook, Google,… Vì các hàm xác thực được cung cấp trong gói nên chỉ cần gọi tên hàm ra dùng là được (ví dụ dưới).

```
import { getAuth, signInWithEmailAndPassword } from "firebase/auth";

const auth = getAuth();
signInWithEmailAndPassword(auth, email, password)
 .then((userCredential) => {
  // Signed in
  const user = userCredential.user;
  // ...
 })
 .catch((error) => {
  console.log(error.message);
 });
```

# CHƯƠNG 3. HỆ THỐNG MẠNG XÃ HỘI TỪ THIỆN (KANJ HAUNGO)

Giới thiệu về ứng dụng mạng xã hội từ thiện Kanj Haungo về mặt chức năng cũng như là cách hoạt động từ đó giúp chúng ta có cái nhìn tổng quan và hiểu rõ hơn về ứng dụng.

## 3.1 Giới thiệu hệ thống

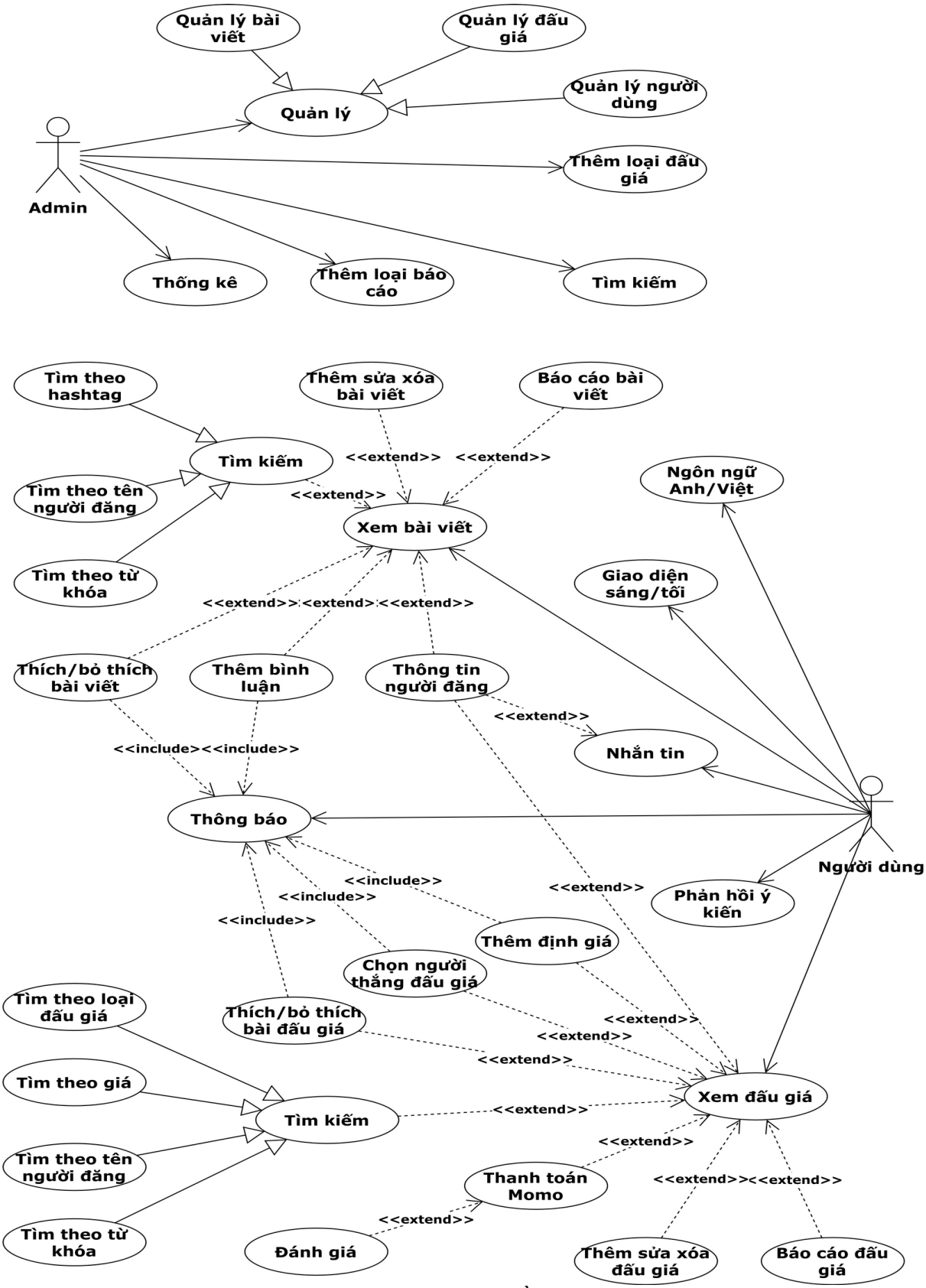
Mạng xã hội từ thiện Kanj Haungo là một ứng dụng mạng xã hội từ thiện chạy trên điện thoại, nhằm mục đích chia sẻ thông tin và tạo các bài đấu giá nhằm mục đích từ thiện. Sau khi đăng ký tài khoản, người dùng có thể tạo các bài viết với ảnh và hashtag kèm theo đồng thời cũng có thể báo cáo các hành vi sai trái.

Một tính năng khác cũng phải kể đến đó là cho phép mở cuộc đấu giá nhằm mục đích từ thiện và có thể thanh toán bằng tiền mặt hoặc qua ví điện tử MoMo. Người đưa sản phẩm đấu giá được phép chọn khách hàng đấu giá theo ý mình. Sau khi giao dịch thành công người mua sẽ được đánh giá cuộc đấu giá đó bao nhiêu sao.

## 3.2 Sơ đồ Use Case

Một trong những chức năng quan trọng của app là đấu giá sản phẩm. Chức năng này cho phép người dùng có thể tạo các bài đấu giá với những thông tin như điều kiện, giá thấp nhất và mục đích từ thiện của việc đấu giá sản phẩm. Sau khi chọn được định giá thì sẽ tiến hành giao dịch trực tiếp hoặc trực tuyến thông qua việc thỏa thuận của cả hai bên mua bán. Trong quá trình đấu giá chỉ có người bán mới được xem hết các định giá của mọi người. Và chỉ có một giao dịch được diễn ra trong suốt quá trình.

Chức năng thống kê dành cho Admin cũng là một chức năng quan trọng, để biết các xu hướng và hiểu rõ hơn hành vi người dùng từ đó đưa ra các chiến lược, chính sách thích hợp.




Hình 3.2.1: Sơ đồ Use Case

## 3.3 Đặc tả chức năng

### 3.3.1 Chức năng đấu giá

Một đấu giá gồm 4 trạng thái: đang đấu giá, đang giao dịch, đấu giá thành công, đấu giá bị huỷ. Một định giá của người mua gồm 4 trạng thái: chưa giao dịch, đang giao dịch, giao dịch bị hủy, giao dịch thành công. Định giá là giá mà người mua đưa ra, đấu giá là bài đăng đấu giá sản phẩm.

Bảng 3.3.1: Use Case tạo đấu giá

| | |
|---|---|
| Mô tả | Use Case này cho phép người bán tạo cuộc đấu giá |
| Actor chính | Người bán |
| Tiền điều kiện | Người bán cần đăng nhập để có thể tạo đấu giá |
| Hậu điều kiện | Cuộc đấu giá được tạo ra ở trạng thái đang đấu giá. |
| Luồng hoạt động | 4. Người bán truy cập vào ứng dụng, bấm tạo đấu giá<br>5. Tiếp tục điền các thông tin cần thiết để tạo đấu giá<br>6. Bấm đăng đấu giá. |
| Ràng buộc | Nếu chọn thanh toán bằng Online thì người mua trả tiền cho App trước, hàng sẽ được vận chuyển sau. Sau khi người mua nhận hàng, tiền sẽ được chuyển qua lại cho người bán. |

Bảng 3.3.2: Use Case để người mua tạo ra định giá

| | |
|---|---|
| Mô tả | Use Case này cho phép người mua đưa ra định giá |
| Actor chính | Người mua |
| Tiền điều kiện | Đấu giá đang ở trạng thái đang giao dịch hoặc đang đấu giá |
| Hậu điều kiện | Người mua tạo được định giá ở trạng thái chưa giao dịch |
| Luồng hoạt động | 1. Bấm vào một bài đấu giá để xem chi tiết<br>2. Nhập nội dung và giá mua ở dưới bài đấu giá<br>3. Bấm gửi bài đấu giá |
| Ràng buộc | Chỉ bán chỉ thấy số lượng định giá và định giá của chính mình. Duy nhất người bán có thể xem nội dung được các định giá. |

Bảng 3.3.3: Use Case bán sản phẩm đấu giá

| | |
|---|---|
| Mô tả | Use Case này cho phép người bán bán sản phẩm thành công |
| Actor chính | Người bán |
| Tiền điều kiện | Đấu giá đang ở trạng thái đang đấu giá và có ít nhất một định giá ở trạng thái chưa giao dịch. |
| Hậu điều kiện | Đấu giá thành công |
| Luồng hoạt động | 1. Vào chi tiết đấu giá rồi chọn một định giá.<br>2. Bấm gửi để bắt đầu một giao dịch (Trạng thái của đấu giá và định giá chuyển về "đang giao dịch")<br>3. Người mua nhận thông báo khi vào chi tiết bài đăng phía dưới cùng sẽ có thông tin liên hệ của người bán<br>Th1: Thanh toán offline<br>4. Hai bên tự giao dịch xong người mua đánh đấu giá thành trạng thái giao dịch thành công<br>Th2: Thanh toán online<br>4. Bấm vào nút thanh toán online trong chi tiết bài đấu giá |

| | |
|---|---|
| | tiếp tục bấm thực hiện giao dịch<br>5. Quá trình sẽ được thực hiện tiếp ở ứng dụng Momo, sau khi người dùng bấm xác nhận thanh toán sẽ quay về app Kanj Haungo để xử lý<br>6. Hiện thông báo thanh toán thành công |
| Luồng ngoại lệ | Giao dịch không thành công thì định giá về trạng thái giao dịch bị hủy và không thể giao dịch lại, đấu sẽ về trạng thái đang giao dịch |

Bảng 3.3.4: Use Case hủy đấu giá

| | |
|---|---|
| Mô tả | Use Case này cho phép người bán hủy đấu giá |
| Actor chính | Người bán |
| Tiền điều kiện | Đấu giá đang không ở trạng thái thành công hay thất bại |
| Hậu điều kiện | Đấu giá thất bại |
| Luồng hoạt động | 1. Bấm vào giao diện chi tiết đấu giá<br>2. Chọn menu góc trên cùng bên phải<br>3. Chọn “Hủy đấu giá”<br>4. Hiện thông báo hủy thành công |
| Ràng buộc | Giao dịch bị hủy không thể tương tác và khôi phục. |

### 3.3.2 Chức năng tương tác với bài đăng

Bảng 3.3.5: Use Case tương tác với bài đăng

| | |
|---|---|
| Mô tả | Use Case này cho phép người dùng tương tác với bài đăng |
| Actor chính | Người dùng ứng dụng |
| Tiền điều kiện | Người dùng phải đăng nhập |
| Hậu điều kiện | Tương tác được với bài đăng |
| Luồng hoạt động | 1. Bấm vào các bài viết trong menu và hiện ra danh sách bài viết. |

| | |
|---|---|
| | 2a. Bấm vào nút like để thích bài viết.<br>2b. Bấm vào nút bình luận để xem chi tiết bài đăng<br>3b. Nhập bình luận và đăng<br>4b. Người đăng nhận thông báo có người bình luận<br>2c. Bấm vào menu bài viết<br>3c. Chọn báo cáo bài viết<br>4c. Nhập nội dung và gửi báo cáo |
| Ràng buộc | Quản trị viên có thể xóa bài viết nếu có trên 20 lượt report. |

### 3.3.3 Chức năng đánh giá người bán

Bảng 3.3.6: Use Case này cho phép đánh giá người bán

| | |
|---|---|
| Mô tả | Use Case này cho phép người mua sau khi mua xong được đánh giá người bán |
| Actor chính | Người mua |
| Tiền điều kiện | Người mua đã giao dịch với người bán thành công |
| Hậu điều kiện | Đánh giá được số sao cho người bán |
| Luồng hoạt động | 4. Chọn đấu giá đã hoàn thành giao dịch online xong<br>5. Lướt xuống dưới cùng của chi tiết bài đấu giá, và đánh giá bằng cách cho số sao từ 1 đến 10. |
| Ràng buộc | Giao dịch phải là thanh toán bằng phương thức online |

### 3.3.4 Chức năng tìm kiếm bài viết

Bảng 3.3.7: Use Case tìm kiếm bài viết theo từ khóa

| | |
|---|---|
| Mô tả | Use Case này cho phép người sử dụng tìm kiếm bài viết |
| Actor chính | Người dùng |
| Tiền điều kiện | Vào phần tìm kiếm bài viết của ứng dụng |

| | |
|---|---|
| | |
| Hậu điều kiện | Hiện kết quả các bài viết đúng thông tin tìm kiếm |
| Luồng hoạt động | 4. Chọn thể loại tìm kiếm (hashtag, tên người đăng hoặc nội dung)<br>5. Nhập từ khóa tìm kiếm<br>6. Bấm nút tìm kiếm<br>7. Hiển thị các bài viết theo kết quả tìm kiếm |
| Ngoại lệ | 4. Không hiển thị bài viết nào vì không đúng theo từ khóa |

### 3.3.5 Chức năng tìm kiếm bài đấu giá

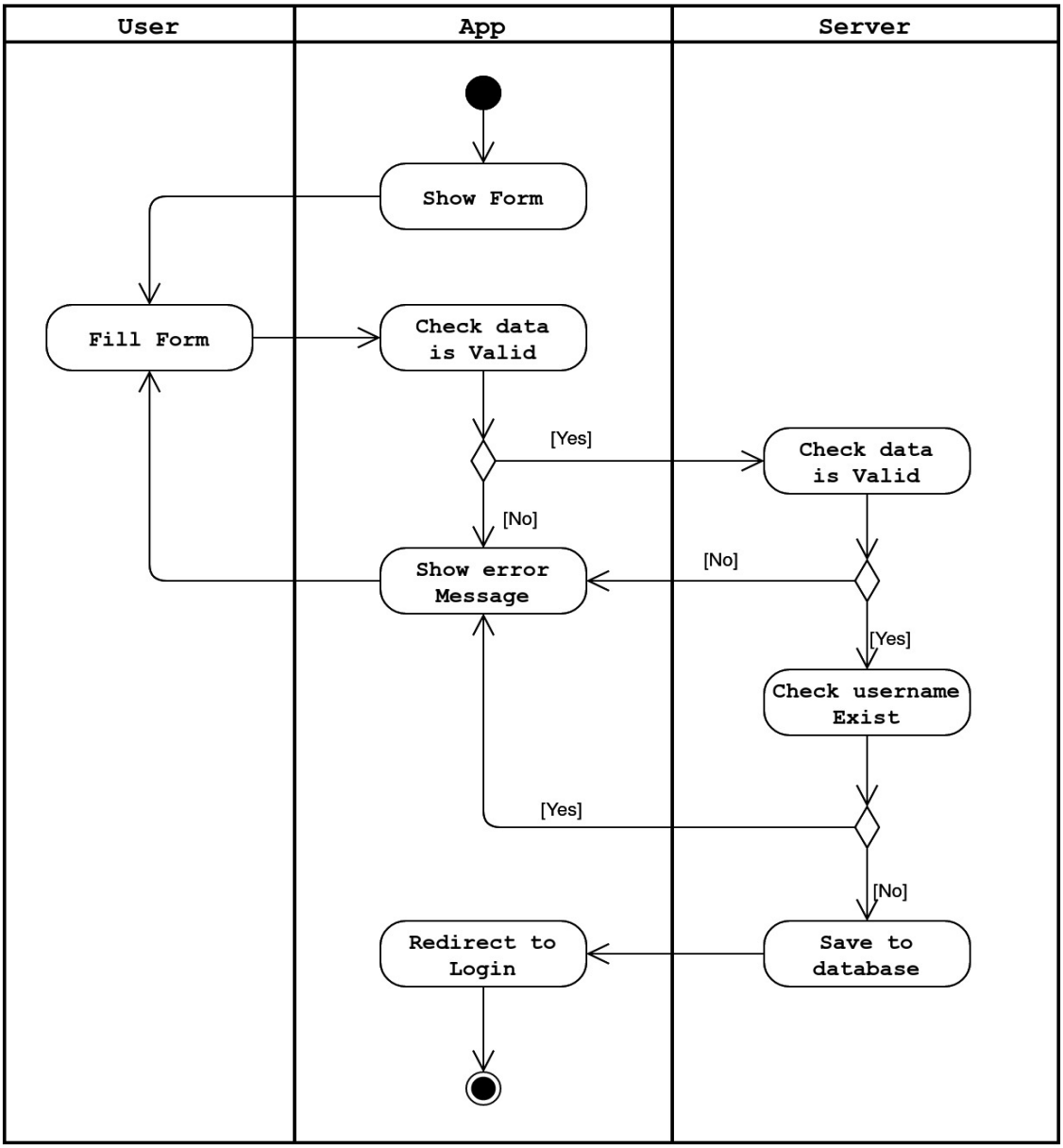
Bảng 3.3.8: Use Case tìm kiếm bài đấu giá

| | |
|---|---|
| Mô tả | Use Case này cho phép người sử dụng tìm kiếm bài đấu giá |
| Actor chính | Người dùng |
| Tiền điều kiện | Vào phần tìm kiếm bài đấu giá của ứng dụng |
| Hậu điều kiện | Hiện kết quả các bài đấu giá đúng thông tin tìm kiếm |
| Luồng hoạt động | 1. Chọn thể loại tìm kiếm (giá thấp nhất, tên người đăng, loại sản phẩm hoặc nội dung)<br>Th1: Thể loại tìm kiếm khác loại sản phẩm<br>2. Nhập từ khóa tìm kiếm<br>Th2: Thể loại tìm kiếm là loại sản phẩm<br>2. Chọn loại sản phẩm theo danh sách đã cho<br>3. Bấm nút tìm kiếm<br>4. Hiển thị các bài đấu giá theo kết quả tìm kiếm |
| Ngoại lệ | 4. Không hiển thị bài đấu giá nào vì không thỏa điều kiện |

## 3.4 Sơ đồ Activity

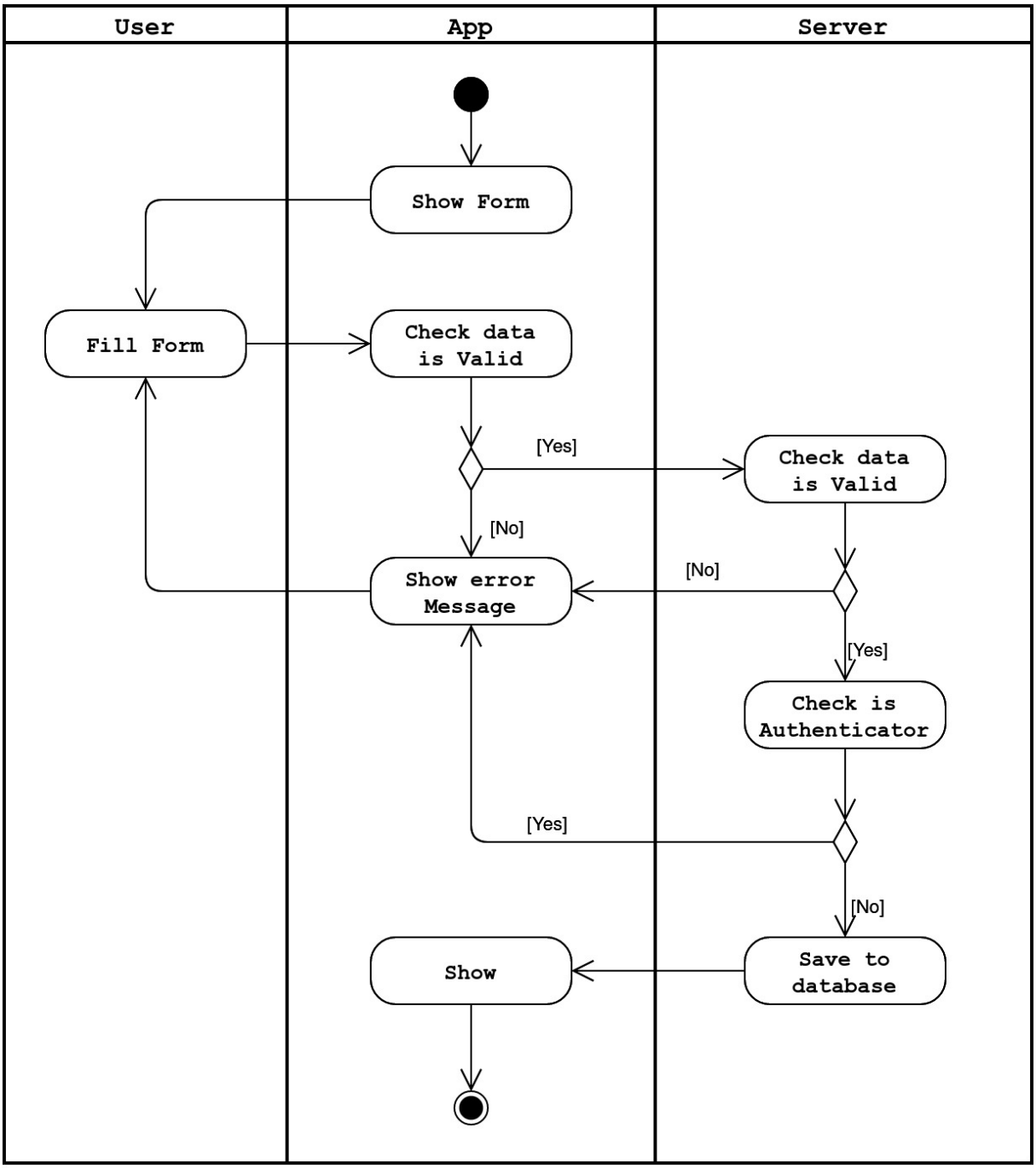
Mô tả cách thức hoạt động, cũng như là cách thức sử dụng của người về một số luồng hoạt động chính của App

Khi người dùng mới sử dụng App thì bắt buộc phải đăng ký tài khoản theo Form cho trước và nếu trùng tên đăng ký thì phải sử dụng tên khác. Sau khi thành công thì sẽ được chuyển qua giao diện đăng nhập với tên tài khoản và mật khẩu đã được tự động điền.



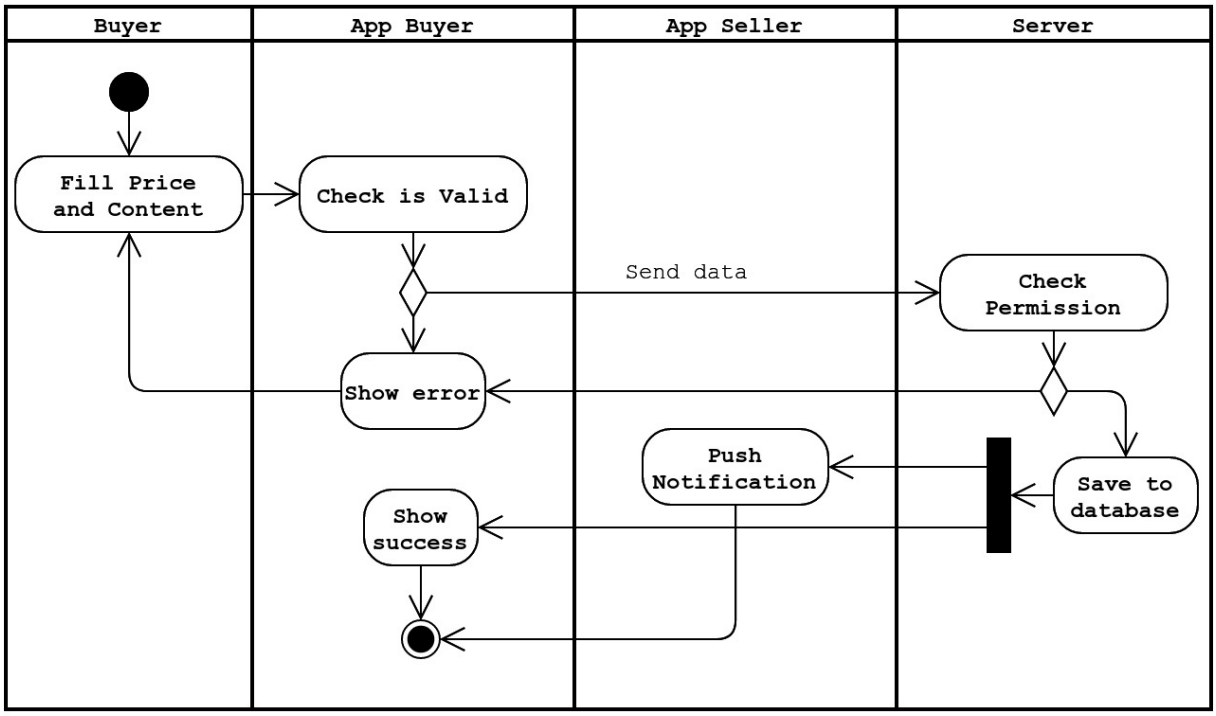

Hình 3.4.1: Sơ đồ đăng ký

Để tạo một đấu giá người dùng phải bấm vào tạo đấu giá xong điền các thông tin cần thiết rồi đăng. Bài đăng sẽ được thêm vào cuối danh sách của tất cả các bài đấu giá.



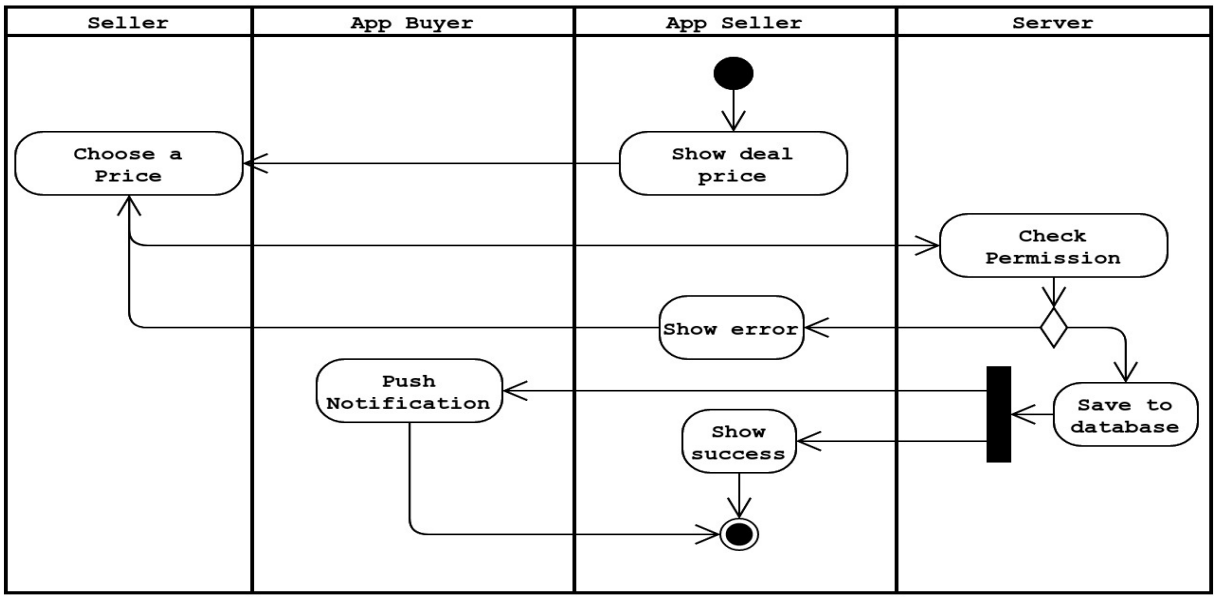

Hình 3.4.2: Sơ đồ tạo đấu giá

Khi muốn đấu giá một sản phẩm thì phải thêm một bình luận vào dưới phần đấu giá với 2 phần là nội dung và giá bạn định cho sản phẩm. Người dùng chỉ được xem có bao nhiêu định giá và định giá của chính mình.



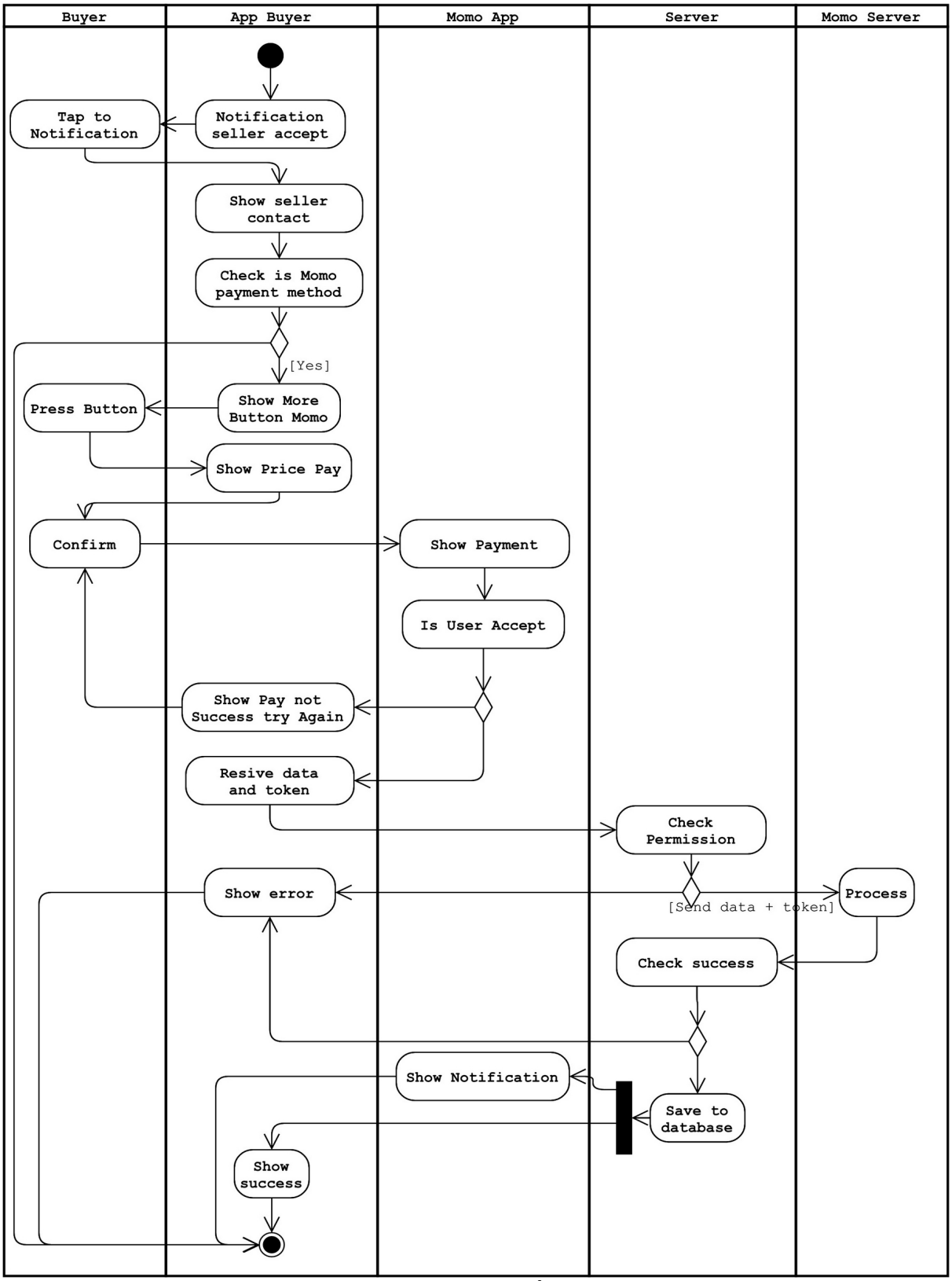

Hình 3.4.3: Sơ đồ quá trình tạo định giá

Sau khi đã có ít nhất một định giá thì người bán có thể chọn định giá nào để bán sản phẩm của mình.




Hình 3.4.4: Sơ đồ quá trình chọn giá bán

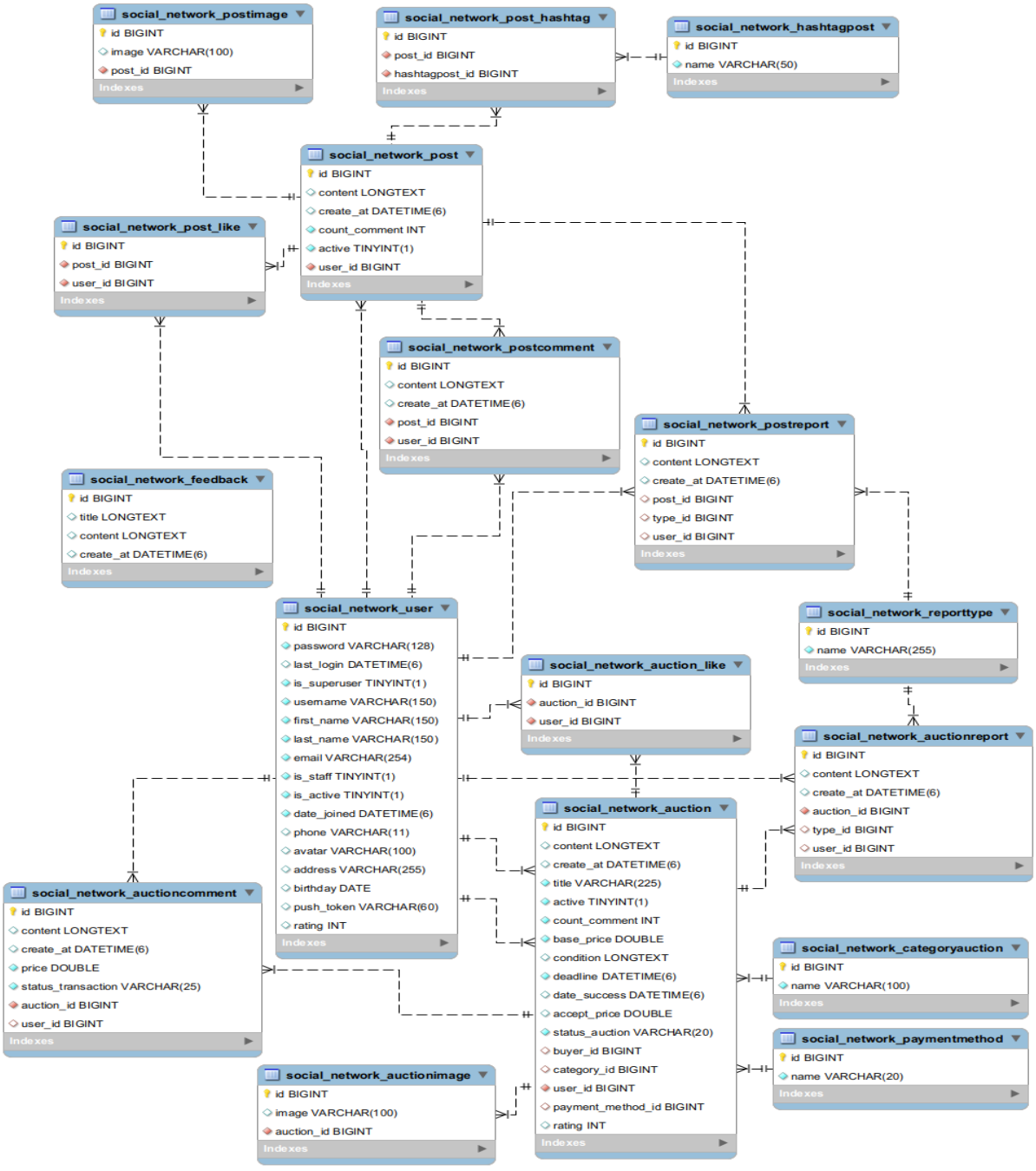
Sau khi người bán đồng ý một định giá thì người đưa ra định giá đó sẽ nhận được thông báo và cả hai thấy thông tin liên lạc với nhau và tiến hành giao dịch.




Hình 3.4.5: Sơ đồ giao dịch

## 3.5 Cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế theo kiểu Sql và được chạy trên nền MySql, một trong những hệ quản trị cơ sở dữ liệu mã nguồn mở lớn nhất thế giới.




Hình 3.5.1: Cơ sở dữ liệu

# CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận tổng quan về đề tài và sản phẩm ứng dụng, những khuyết điểm và thành quả đạt được. Từ tính năng còn thiếu và chưa ưu việt sẽ được đưa ra các hướng phát triển ứng dụng sau này.

## 4.1 Kết luận

App Kanj Haungo đã tích hợp một số tính năng cơ bản để có thể sử dụng một cách bình thường như:

- Quản lý bài đăng (thêm, sửa, xóa), thêm bình luận
- Quản lý bài đấu giá, tạo định giá, chọn định giá, hủy đấu giá
- Thanh toán Momo
- Thích hoặc bỏ thích đối với bài đăng và bài đấu giá
- Thông báo đẩy khi có tương tác với bài đăng hay bài đấu giá
- Chỉnh sửa thông tin cá nhân
- Nhắn tin
- Đánh giá người dùng
- Báo cáo bài viết

Tuy vậy nhưng vẫn chưa chiếm được sự tin tưởng hoàn toàn của khách hàng với sản phẩm. App rất cần thiết một số tính năng để xác thực thông tin và cần có một đội ngũ tham gia kiểm duyệt các sản phẩm được trước khi đưa lên giao dịch.

## 4.2 Hướng phát triển

Tuy app đã có thể sử dụng bình thường nhưng bên cạnh đó vẫn tồn tại một số khuyết điểm cần phải sửa và nâng cấp như:

- App chưa có chức năng đánh giá người bán.
- Thiếu tin cậy trong việc xác nhận thông tin người dùng, hay xác thực các thông tin giao dịch offline.

- App đã test và chạy ổn định trên các dòng điện thoại Android 7 trở lên.
- Gợi ý bài đăng, bài đấu giá phù hợp cho người xem vẫn chưa được thông minh.

Vì vậy việc thêm các tính năng mới và áp dụng các thuật toán chạy AI là điều rất cần thiết để tăng tính chính xác và nâng cao trải nghiệm người dùng.

# TÀI LIỆU THAM KHẢO


[1] "React Native · Learn once, write anywhere". https://reactnative.dev/ (truy cập tháng 10 14, 2021).

[2] "React – A JavaScript library for building user interfaces". https://reactjs.org/ (truy cập tháng 10 14, 2021).

[3] "Introduction to Expo", *Expo Documentation*. https://docs.expo.dev/ (truy cập tháng 10 14, 2021).

[4] "Redux - A predictable state container for JavaScript apps. | Redux". https://redux.js.org/ (truy cập tháng 10 14, 2021).

[5] "The web framework for perfectionists with deadlines | Django". https://www.djangoproject.com/ (truy cập tháng 10 16, 2021).

[6] "Home - Django REST framework". https://www.django-rest-framework.org/ (truy cập tháng 10 16, 2021).

[7] "Firebase Documentation". https://firebase.google.com/docs (truy cập tháng 11 06, 2021).