

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



TRÌNH NGỌC TÌNH

PHÁT TRIỂN ỨNG DỤNG MOBILE: COFFEE SHOP

ĐỒ ÁN NGÀNH
NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2021

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



TRÌNH NGỌC TÌNH

PHÁT TRIỂN ỨNG DỤNG MOBILE: COFFEE SHOP

Mã số sinh viên: 1751012071

ĐỒ ÁN NGÀNH
NGÀNH KHOA HỌC MÁY TÍNH

Giảng viên hướng dẫn:
TS. TRƯƠNG HOÀNG VINH

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: TRẦN NGỌC TÌNH

Ngày sinh: 15/10/1999 Nơi sinh: Quảng Nam.....

Chuyên ngành: Cơ sở dữ liệu Mã sinh viên: 1751012071

Tôi đồng ý cung cấp toàn văn thông tin đồ án 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 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)

Trần Ngọc Tình

**Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN
CỦA GIẢNG VIÊN HƯỚNG DẪN**

Giảng viên hướng dẫn: TS. TRƯƠNG HOÀNG VINH

Sinh viên thực hiện: Trình Ngọc Tình **Lớp:** DH17–TH05

Ngày sinh: 15/10/1999 **Nơi sinh:** Quảng Nam

Tên đề tài: Phát triển ứng dụng Mobile: Coffee Shop

.....

.....

.....

.....

**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

TS. Trương Hoàng Vinh

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án, do tình hình dịch bệnh nên em đã gặp nhiều khó khăn nhưng nhờ sự giúp đỡ, đóng góp ý kiến và sự hướng dẫn nhiệt tình của quý thầy cô, bạn bè và đặc biệt là giảng viên hướng dẫn nên e đã hoàn thành được đồ án.

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến TS. Trương Hoàng Vinh, giảng viên Khoa Công nghệ Thông tin – Trường Đại học Mở Thành phố Hồ Chí Minh đã dành thời gian hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện đồ án.

Tiếp theo, em tỏ lòng biết ơn đến Ban giám hiệu nhà trường cùng quý Thầy Cô giảng viên trong Khoa đã tạo môi trường dạy và học thật tốt, cho em học hỏi và tiếp thu kiến thức về các môn đại cương và chuyên ngành, giúp em có thêm những cơ sở lý thuyết vững vàng làm nền tảng trong việc thực hiện đồ án.

Cuối cùng, em xin cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm và động viên em về mặt tinh thần trong suốt quá trình học tập và hoàn thành đồ án.

Vì còn là sinh viên nên kinh nghiệm của chúng em còn hạn chế, bài báo cáo đồ án còn gặp nhiều sai sót. Em rất mong nhận được những ý kiến đóng góp của quý thầy cô để em bổ sung, cải thiện và nâng cao kiến thức của mình!

Em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, ngày ... tháng ... năm 2021

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN.....	1
1.1. Giới thiệu.....	1
1.2. Mục tiêu.....	1
1.3. Giới hạn đề tài	2
1.4. Phương pháp nghiên cứu.....	2
1.5. Bố cục bài báo cáo	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1. Giới thiệu.....	3
2.2. Cơ sở lý thuyết.....	3
2.2.1. React Native	3
2.2.1.1. Tổng quan	3
2.2.1.2. Native App và Hybrid App.....	5
2.2.1.3. Nguyên lý hoạt động	5
2.2.1.4. Native Components và Core Components	6
2.2.1.5. Core Concepts.....	6
2.2.1.6. Cài đặt ReactNative	11
2.2.1.7. Tạo ứng dụng React Native đầu tiên	12
2.2.1.8. Một số ứng dụng sử dụng ReactNative	13
2.2.2. FireBase.....	13
2.2.2.1. Tổng quan	13
2.2.2.2. Cách thức hoạt động	14
2.2.2.3. Cài đặt Firebase	16
2.2.2.4. Ưu nhược điểm của Firebase	16
CHƯƠNG 3. THỰC NGHIỆM ỨNG DỤNG	18
3.1. Mục đích.....	18
3.2. Chức năng của hệ thống	18
3.2.1. Thực nghiệm giao diện và chức năng quản trị người dùng	18

3.2.2. Thực nghiệm giao diện và chức năng quản lý sản phẩm của admin	23
3.2.3. Thực nghiệm giao diện và chức năng quản lý hóa đơn của admin...	26
3.2.4. Thực nghiệm giao diện với vai trò người dùng	28
3.2.4.1. Đăng nhập và đăng ký	28
3.2.4.2. Trang chủ ứng dụng.....	32
3.2.4.3. Thông tin tài khoản người dùng	36
CHƯƠNG 4. TỔNG KẾT	37
4.1. Kết quả đạt được	37
4.2. Hạn chế.....	37
4.3. Hướng phát triển đề tài	37

DANH MỤC TỪ VIẾT TẮT

DANH MỤC HÌNH VẼ

Hình 2.1.	Giới thiệu lịch sử React Native.....	3
Hình 2.2.	Nguyên lý hoạt động của React.....	5
Hình 2.3.	Mô tả code JSX.....	8
Hình 2.4.	Mô tả biểu thức JavaScript của JSX.....	8
Hình 2.5.	Ứng dụng React Native đầu tiên.....	13
Hình 2.6.	Firebase Realtime Database.....	14
Hình 2.7.	Firebase Authentication.....	15
Hình 2.8.	Firebase Hosting.....	15
Hình 3.1.	Thực nghiệm giao diện quản lý của Admin.....	18
Hình 3.2.	Mô tả các chức năng tìm kiếm, thêm, sửa, xóa và vô hiệu hóa tài khoản.....	19
Hình 3.3.	Nhập thông tin tìm kiếm.....	19
Hình 3.4.	Kết quả tìm kiếm user.....	19
Hình 3.5.	Nhập thông tin User.....	20
Hình 3.6.	Thêm thành công User mới.....	20
Hình 3.7.	Hướng dẫn Sửa mật khẩu của user.....	20
Hình 3.8.	Email thông báo chứa link thay đổi mật khẩu.....	21
Hình 3.9.	Xóa tài khoản người dùng.....	21
Hình 3.10.	Thông báo tài khoản đã bị xóa.....	22
Hình 3.11.	Vô hiệu hóa tài khoản người dùng.....	22
Hình 3.12.	Thông báo tài khoản đã bị vô hiệu hóa.....	23
Hình 3.13.	Danh sách sản phẩm của hệ thống.....	23
Hình 3.14.	Thông tin chi tiết sản phẩm cần xem.....	24
Hình 3.15.	Nhập thông tin sản phẩm cần thêm.....	24
Hình 3.16.	Hình sản phẩm đã được thêm vào CSDL.....	25
Hình 3.17.	Xóa sản phẩm Trà Đào.....	25
Hình 3.18.	Sửa giá của sản phẩm Bạc xỉu.....	26
Hình 3.19.	Tổng quan về quản lý hóa đơn trên web admin.....	26
Hình 3.20.	Tạo hạng mục phân loại đơn đặt hàng.....	27
Hình 3.21.	Nhập thông tin đơn hàng.....	27
Hình 3.22.	Thông tin đơn hàng có ID: 1.....	28

Hình 3.23. Sửa và xóa thông tin đơn đặt hàng	28
Hình 3.24. Giao diện đăng nhập App	29
Hình 3.25. Giao diện đăng ký của App	30
Hình 3.26. Thông báo đăng ký tài khoản thành công.....	30
Hình 3.27. Thông báo email đã được sử dụng không thể đăng ký.....	31
Hình 3.28. Thông báo địa chỉ email không hợp lệ để đăng ký.....	31
Hình 3.29. Trang chủ ứng dụng Coffee Shop.....	32
Hình 3.30. Chính sách ưu đãi của ứng dụng.....	33
Hình 3.31. Danh sách các sản phẩm.....	34
Hình 3.32. Thêm các sản phẩm vào giỏ hàng.....	35
Hình 3.33. Thanh toán giỏ hàng	36
Hình 3.34. Giao diện thông tin account.....	36

DANH MỤC BẢNG

MỞ ĐẦU

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu

Ngày nay, cách mạng công nghiệp 4.0 đã quá đỗi quen thuộc với nhiều người. Trên đà phát triển của xã hội, đồ điện tử xuất hiện ngày càng nhiều trong cuộc sống của chúng ta. Chúng mang đến nhiều lợi ích cho con người, tiết kiệm về thời gian, tiền bạc... Bên cạnh đó không thể kể đến Smartphone.

Smartphone ngày càng phổ biến rộng rãi, dần dần thay thế Feature Phone và trở thành vật bất ly thân với mọi người. Nhiều người lạm dụng việc sử dụng Smartphone quá mức sẽ ảnh hưởng xấu đến bản thân, nhưng không thể phủ nhận lợi ích Smartphone mang lại đến cho thế giới con người là rất nhiều. Nhất là trong lĩnh vực kinh doanh.

Hiện nay, hầu hết mọi người đều dần dần tìm hiểu và sử dụng các ứng dụng trên Smartphone, từ học sinh, sinh viên cho đến các doanh nhân, social networker... Smartphone có sức hút mạnh mẽ nên việc các công ty, cá nhân sử dụng để quảng bá thương hiệu, sản phẩm của họ ngày càng nhiều. Như giải trí (game: PUBG mobile, Candy Crush...; đọc sách: Wattpad, Ibooks, BlueBook...), học tập (Tiếng Anh: Duolingo, Hello English...; giải toán: PhotoMath, Quanda...), chăm sóc sức khỏe (tập thể dục: Belife, Yoga...; tư vấn dinh dưỡng: Fooducate, Calorie Counter Pro...) và đặc biệt là về ăn uống (GrabFood, The Coffee House...).

Với việc xã hội ngày càng phát triển, con người ngày càng bận rộn với công việc và ảnh hưởng của đại dịch covid-19, nhu cầu tiêu dùng online trở nên mạnh mẽ. Con người ngày càng ưa chuộng với việc sử dụng app mobile để đặt thức uống mà không mất quá nhiều thời gian.

Với những lý do trên, em quyết định chọn đề tài phát triển ứng dụng mobile: Coffee Shop cho đồ án của mình.

1.2. Mục tiêu

- Xây dựng và phát triển một ứng dụng mobile về buôn bán coffee, có thể đưa vào thực tế cho người dùng sử dụng
- Có thể sử dụng khi đi phỏng vấn tìm việc làm

- Cố gắng sử dụng các công nghệ vào project: thanh toán bằng thẻ tín dụng, kết nối với mạng xã hội khác (facebook, google...)
- Tích lũy kiến thức, kinh nghiệm và rèn dũa bản thân trong quá trình thực hiện đồ án.

1.3. Giới hạn đề tài

- Các dữ liệu về thông tin sản phẩm: tên sản phẩm, giá cả, hình ảnh, kích thước...
- Thuộc lĩnh vực đồ uống: coffee, trà sữa, nước ngọt...

1.4. Phương pháp nghiên cứu

Thông qua quá trình thu thập, nghiên cứu và tổng hợp kiến thức:

- Chọn lọc dữ liệu, tạo dữ liệu online
- Thu thập ý kiến từ thầy cô, bạn bè và người thân để lên nội dung cho app
- Lựa chọn các công nghệ mới, dễ tìm hiểu và sử dụng để áp dụng vào nghiên cứu đồ án
- Tham khảo thêm một số các app mobile khác: The Coffee House, Trung Nguyên Legend...

1.5. Bố cục bài báo cáo

Bài báo cáo được chia thành 4 chương:

- Chương 1: Giới thiệu tổng quan về đề tài, đối tượng, mục tiêu nghiên cứu và các phương pháp nghiên cứu đề tài
- Chương 2: Cơ sở lý thuyết về React Native và Firebase
- Chương 3: Thực nghiệm của ứng dụng về mục đích, chức năng và phương thức hoạt động của ứng dụng
- Chương 4: Tổng kết, đánh giá tổng quan đề tài, nêu lên kết quả đạt được và những hạn chế khi thực hiện đề tài, hướng phát triển đề tài.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu

Các công nghệ được áp dụng vào trong đề tài phát triển ứng dụng mobile: Coffee Shop bao gồm:

- React Native
- Firebase

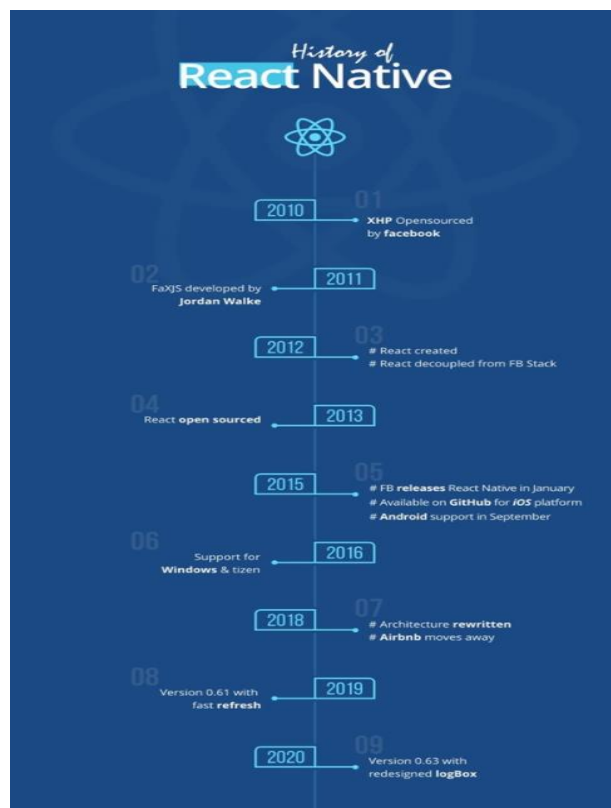
2.2. Cơ sở lý thuyết

2.2.1. React Native

2.2.1.1. Tổng quan

React Native là một trong những framework phát triển ứng dụng di động được thiết kế và phát triển bởi Facebook.

Năm 2015, Facebook đã công khai mã nguồn của React Native và có trên GitHub. Vào tháng 9, React Native cho Android cũng được ra mắt. Vào những năm 2016, các lập trình viên Facebook cũng công bố React Native có thể hỗ trợ cả trên nền tảng Windows và Tizen.



Hình 2.1. Giới thiệu lịch sử React Native

React Native là một JavaScript Framework cho phép lập trình viên xây dựng các ứng dụng di động trên cả Android và iOS.

Đây là một framework cho phép sử dụng trên nhiều nền tảng khác nhau cùng một cơ sở mã. Được phát triển tối ưu hóa hiệu năng Hybrid và tối giản ngôn ngữ Native di động.

Ưu và nhược điểm của React Native

Ưu điểm

- Có thể tái sử dụng code: developer có thể tái sử dụng hầu như 80-90% đoạn code thay vì viết và tạo các ứng dụng riêng biệt cho các nền tảng khác.
- Tiết kiệm tiền và thời gian.
- Team phát triển nhỏ.
- Cộng đồng người dùng lớn: Theo khảo sát của Stack Overflow vào năm 2019, React Native được đánh giá là một trong những Framework được yêu thích nhất.
- Tính ổn định và khả năng tối ưu cao: Hiệu năng ổn định khá cao, các mã React Native đơn giản hóa quá trình xử lý dữ liệu.
- Mang đến người dùng các trải nghiệm thú vị.

Nhược điểm

- Vẫn còn thiếu nhiều component quan trọng.
- Hiệu năng kém hơn so với native app.
- Quản lý bộ nhớ.
- Bảo mật chưa tốt do dùng JavaScript.
- Ảnh hưởng bởi một số đặc điểm của JavaScript: dễ làm dễ sai, khó duy trì về sau.
- Không thích hợp khi ứng dụng vào các app cần khả năng tính toán cao.

2.2.1.2. Native App và Hybrid App

Native App

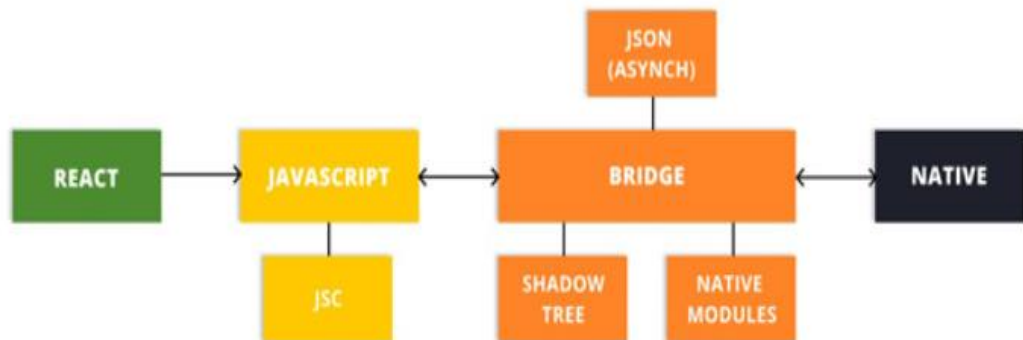
- Native App là những ứng dụng được xây dựng và phát triển trên những công cụ do chính nhà phát triển cung cấp
- Nó được tạo ra bằng ngôn ngữ của hệ điều hành và các tính năng có sẵn.
- Các ứng dụng vận hành với tốc độ cao mà không thông qua bất kì ứng dụng hay engine nào
- Ngôn ngữ: Android (Java, Kotlin), iOS (Swift)

Hybrid App

- “đứa con” giữa ứng dụng web và mobile
- Có khả năng cài đặt trên điện thoại vừa có thể sử dụng trên nền tảng web
- Ngôn ngữ: HTML, CSS và JS

2.2.1.3. Nguyên lý hoạt động

FrameWork này hoạt động dựa trên tích hợp 2 luồng với nhau. Đó là Main Thread và JS Thread với vai trò riêng biệt:



Hình 2.2. Nguyên lý hoạt động của React

- Main Thread: vai trò cập nhật giao diện người dùng (UI) và xử lý tương tác người dùng.
- JS Thread: đảm bảo hệ thống hoạt động hiệu quả thông qua việc thực thi và xử lý code JavaScript.
- Hai luồng kết nối với nhau thông qua một Bridge, duy trì sự giao tiếp nhưng không phụ thuộc nhau

2.2.1.4. Native Components và Core Components

- Native Components: Các thành phần React Native được hỗ trợ bởi các chế độ xem giống như Android và iOS nên các ứng dụng React Native có giao diện, hoạt động giống như các ứng dụng khác. Các thành phần được nền tảng hỗ trợ này là Native Components
- Core Components: React Native có nhiều Core Components cho mọi thứ từ điều khiển biểu mẫu đến chỉ báo hoạt động. Chúng được ghi lại trong phần API, các Core Components như:

REACT NATIVE UI COMPONENT	ANDROID VIEW	IOS VIEW	WEB ANALOG
<code><View></code>	<code><ViewGroup></code>	<code><UIView></code>	A non-scrollling <code><div></code>
<code><Text></code>	<code><TextView></code>	<code><UITextView></code>	<code><p></code>
<code><Image></code>	<code><ImageView></code>	<code><UIImageView></code>	<code></code>
<code><ScrollView></code>	<code><ScrollView></code>	<code><UIScrollView></code>	<code><div></code>
<code><TextInput></code>	<code><EditText></code>	<code><UITextField></code>	<code><input type="text"></code>

hình

2.2.1.5. Core Concepts

Component

Xác định Cat Component, trước tiên hãy sử dụng JavaScript import để nhập Text Core Component

```
import React from 'react';
import { Text } from 'react-native';
```

Component bắt đầu dưới dạng một function

```
const Cat = () => {};
```

Bất cứ gì mà một thành phần hàm trả về đều được hiển thị dưới dạng một phần tử React cho phép bạn mô tả những gì bạn muốn xem trên màn hình. Cat Component sẽ hiển thị một `<Text>` phần tử

```
const Cat = () => {  
  return <Text>Hello, I am your cat!</Text>;  
};
```

Xuất function component bằng JavaScript “export default” để sử dụng trong toàn bộ ứng dụng:

```
const Cat = () => {  
  return <Text>Hello, I am your cat!</Text>;  
};  
  
export default Cat;
```

Đây chỉ là một trong nhiều cách để xuất component. Phụ thuộc và cấu trúc tệp của ứng dụng mà bạn có thể sử dụng các quy ước khác.

JSX

- React và React Native sử dụng JSX, đây là một cú pháp cho phép bạn viết các yếu tố bên trong JavaScript như .
- Vì JSX là JavaScript nên bạn có thể sử dụng các biến bên trong nó. Ví dụ khai báo tên cho con mèo và nhúng nó với dấu ngoặc nhọn bên trong `<Text>`, kết quả: “Hello, I am Maru!”

```
import React from 'react';
import { Text } from 'react-native';

const Cat = () => {
  const name = "Maru";
  return (
    <Text>Hello, I am {name}!</Text>
  );
}

export default Cat;
```

Hình 2.3. Mô tả code JSX

- Mọi biểu thức JavaScript sẽ hoạt động giữa các dấu ngoặc nhọn, bao gồm các lệnh gọi hàm như: {getFullName("Trinh", "Ngoc", "Tinh")}

```
import React from 'react';
import { Text } from 'react-native';

const getFullName = (firstName, secondName, thirdName) => {
  return firstName + " " + secondName + " " + thirdName;
}
```

Hình 2.4. Mô tả biểu thức JavaScript của JSX

- JSX được bao gồm trong thư viện React, nó sẽ không hoạt động nếu bạn không có import React from 'react' ở đầu tệp của ứng dụng

Props

- Props là tên viết tắt của properties (thuộc tính). Props cho phép bạn tùy chỉnh các thành phần React
- Hầu hết các Core Component của ReactNative có thể tùy chỉnh bằng Props. Ví dụ: khi sử dụng Image, chuyển nó đến một prop có tên source để xác định hình ảnh mà nó hiển thị:

```
<Image
  source={{uri:
    "https://reactnative.dev/docs/assets/p_cat1.png"}}
  style={{width: 200, height: 200}}
/>
```

Hình

- Chú ý dấu ngoặc nhọn kép `{ { }` xung quanh style width và height. Nó có tác dụng chuyển một đối tượng JS trong JSX.

State

- Ngoài ra bạn có thể xây dựng nhiều thứ với props và Core Component (Text, Image, View). Nhưng để xây dựng bạn sẽ cần tới State
- State giống như một bộ lưu trữ dữ liệu cá nhân của component. State hữu ích để xử lý dữ liệu thay đổi theo thời gian hoặc từ sự tương tác của người dùng, cung cấp cho bộ nhớ các component.
- Ví dụ có hai con mèo đói đang chờ chủ cho ăn. Con đói của mèo được lưu trữ dưới dạng trạng thái. Để cho mèo ăn, hãy nhấn các nút của chúng, nút này sẽ cập nhật trạng thái của chúng. Đầu tiên hãy nhập useState từ React:

```
import React, { useState } from 'react';
```

- Tiếp tục khai báo State của component bằng các gọi useState bên trong hàm của nó. Tạo một isHungry biến trạng thái:

```
const Cat = (props) => {  
  const [isHungry, setIsHungry] = useState(true);  
  // ...  
};
```

- Tiếp theo bạn thêm Button Core Component và cung cấp cho nó một onPress props:

```
<Button  
  onPress={() => {  
    setIsHungry(false);  
  }}  
  //..  
</>
```

- Nếu kích hoạt onPress sẽ gọi setIsHungry(false). Khi này đặt biến trạng thái isHungry thành false. Khi đó, giá trị Button được đặt thành true và nó cũng thay đổi:

```
<Button
  //..
  disabled={!isHungry}
  title={isHungry ? 'Pour me some milk, please!' : 'Thank you!'}
/>
```

- Một số lưu ý: Bạn có thể sử dụng useState để theo dõi bất kỳ loại dữ liệu nào (Strings, numbers, Booleans, arrays, objects)

Handling Text Input

- Handling Text Input TextInput là Core Component cho phép người dùng nhập văn bản.
- TextInput có một onChangeText hỗ trợ nhận một chức năng khi được gọi mỗi khi văn bản thay đổi và một onSubmitEditing hỗ trợ nhận một chức năng được gọi khi văn bản được gửi

ScrollView

- Các ScrollView là một container di chuyển chung có thể chứa nhiều components và views
- Các scrollable items có thể không đồng nhất, bạn có thể cuộn theo cả chiều dọc và chiều ngang bằng cách sử dụng thuộc tính “horizontal”
- Trên iOS, ScrollView được sử dụng để người dùng thu phóng nội dung bằng cách thiết lập maximumZoomScale và minimumZoomScale để thực hiện các cử chỉ chụm và mở rộng để phóng to thu nhỏ
- ScrollView hoạt động tốt nhất để trình bày một số lượng nhỏ những thứ có kích thước hạn chế

List Views

- Dùng để trình bày danh sách dữ liệu: FlatList và SectionList

- FlatList hiển thị danh sách các dữ liệu thay đổi nhưng có cấu trúc tương tự. Hoạt động tốt cho danh sách dữ liệu dài, hiển thị các phần tử hiện đang hiển thị trên màn hình chứ không phải tất cả các phần tử. Bao gồm data (nguồn thông tin cho danh sách) và renderItem (lấy từ nguồn và trả về một thành phần được định dạng để hiển thị).
- SectionList hiển thị một tập hợp dữ liệu được chia thành các phần logic. Nó tương tự như UITableView (iOS).

2.2.1.6. Cài đặt ReactNative

Cài đặt NodeJS

- Chúng ta nên cài đặt Node để có trình quản lý package npm, sau này phát triển ứng dụng React Native ta sử dụng đến console nhiều.
- Vào trang web nodejs.org để download Node JS. Có nhiều phiên bản để chọn, gợi ý:



Cài đặt React Native

- Đầu tiên cài đặt CLI của React Native (Command Line Interface). Đây là trình hỗ trợ dòng lệnh console của React Native. Mở console và cài đặt CLI bằng câu lệnh:

```
npm install -g react-native-cli
```

- Tiếp theo cài đặt React Native bằng câu lệnh:

```
npm install -g create-react-native-app
```

Cài đặt môi trường phát triển Android: Để phát triển ứng dụng, chúng ta cần phải cài đặt môi trường phát triển:

- Cài đặt JDK
- Cài đặt Android Studio
- Thêm biến môi trường Java và Android_Home

- Thêm biến môi trường adb (file adb.exe trong Android SDK)

Cài đặt môi trường phát triển iOS: Tương tự như Android, để React Native build ra một ứng dụng iOS, chúng ta cần:

- Cài đặt MacOS
- Cài đặt XCode để lập trình

Lựa chọn IDE phù hợp để làm việc: Một số IDE hữu ích:

- Nuclide
- Atom
- Sublime Text
- Visual Studio Code

2.2.1.7. Tạo ứng dụng React Native đầu tiên

- Tạo project React Native mới có tên “reactTutorialApp” bằng dòng lệnh:

```
react-native init reactTutorialApp
```

- Chuyển đến thư mục project đã tạo bằng lệnh:

```
cd reactTutorialApp
```

- Chạy sẵn một máy ảo (máy ảo Android mặc định cung cấp bởi AVD).
Sau đó build project bằng lệnh:

```
react-native run-android
```

- Kết quả:



Hình 2.5. Ứng dụng React Native đầu tiên

2.2.1.8. Một số ứng dụng sử dụng ReactNative

- Facebook
- Instagram
- Uber Eats
- Tesla

2.2.2. Firebase

2.2.2.1. Tổng quan

Firebase là một nền tảng để phát triển ứng dụng di động và trang web, nó gồm các API đơn giản nhưng mạnh mẽ mà không cần tới backend hay server

Tiền thân của Firebase là Envolv. Đây là nền tảng cung cấp cho chúng ta những API cần thiết để tích hợp tính năng chat vào trang web. Bên cạnh đó nó còn được sử dụng để truyền và đồng bộ hóa dữ liệu cho nhiều ứng dụng khác. Năm 2012, Firebase ra đời với sản phẩm cung cấp là “dịch vụ Backend-as-a-Service” đến năm 2014 thì được Google mua lại và được phát triển thành dịch vụ đa chức năng.

Vì Firebase là một nền tảng đơn giản nên nó giúp tiết kiệm thời gian triển khai, xây dựng và mở rộng của ứng dụng của các lập trình viên.

2.2.2.2. Cách thức hoạt động

Hiện nay Firebase bao gồm các phương thức hoạt động sau:

Firestore Realtime Database

- Hoạt động nổi bật nhất của Firestore là Firestore Realtime Database (dữ liệu thời gian thực)



Hình 2.6. Firestore Realtime Database

- Khi ta đăng ký tài khoản trên Firestore để sử dụng, chúng ta có một cơ sở dữ liệu thời gian thực dưới dạng JSON. Dữ liệu này luôn được đồng bộ thời gian thực đến mọi client. Khi mất mạng thì dữ liệu vẫn được lưu lại ở local vì thế nó sẽ luôn tự động cập nhật lên Server của Firestore.
- Các dữ liệu ở local cũ hơn với Server thì cũng tự động cập nhật.
- Lấy phiên bản cơ sở dữ liệu cho ứng dụng:

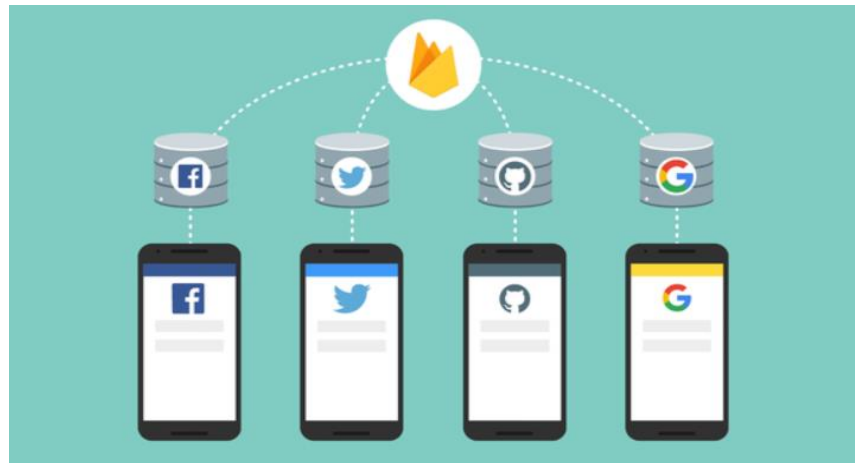
```
const databaseForDefaultApp = firebase.database();
```

- Một số Methods:
 - *getServerTime*: trả về thời gian máy chủ CSDL Firestore dưới dạng đối tượng ngày JS: `getServerTime(): Date`;
 - *goOffline*: ngắt kết nối khỏi máy chủ: `goOffline(): Promise <void>`;
 - *goOnline*: kết nối lại với máy chủ và đồng bộ hóa CSDL với trạng thái máy chủ: `goOnline(): Promise <void>`;
 - *ref*: Trả về reference đại diện cho vị trí trong CSDL tương ứng ref (path?: undefined | string): Reference;

- *useEmulator*: sửa đổi phiên bản CSDL để giao tiếp với CSDL
 Firebase: *useEmulator* (host: string, port: number): void;
- *ServerValue*: trả về các giá trị cụ thể của máy chủ:
 Database.ServerValue: ServerValue;

Firestore Authentication

- Firestore xây dựng các hành động tự động đăng nhập cho các ứng dụng bằng cách xác thực danh tính



Hình 2.7. Firestore Authentication

- Xác thực bằng Facebook, Twitter, Github, Google. Bên cạnh đó còn xác thực ẩn danh nhằm bảo vệ thông tin cá nhân người dùng và không bị đánh cắp

Firestore Hosting

- Đây là cách thức cung cấp các hosting. Hosting được phân phối qua tiêu chuẩn công nghệ bảo mật SSL từ mạng CDN.



Hình 2.8. Firestore Hosting

- SSL là một tiêu chuẩn của công nghệ bảo mật truyền thông mã hóa giữa máy chủ Web server và trình duyệt. Nó đảm bảo tất cả ác dữ liệu được truyền mang tính riêng tư và tách rời.
- CDN là một mạng lưới máy chủ lưu giữ bản sao của các nội dung bên trong website. Nó sẽ được trả về cho người dùng khi họ truy cập website.

2.2.2.3. Cài đặt Firebase

Cài đặt mô-đun ứng dụng ReactNative Firebase vào thư mục gốc của đồ án bằng lệnh npm:

```
npm install --save @react-native-firebase/app
```

2.2.2.4. Ưu nhược điểm của Firebase

Ưu điểm

- Tạo tài khoản bằng cách đăng nhập Google và sử dụng dễ dàng
- Tốc độ phát triển nhanh do không cần truy cập vào Server và Host để tạo hay bảo trì cơ sở dữ liệu
- Firebase cung cấp đầy đủ các dịch vụ hỗ trợ cho Developer trong quá trình phát triển
- Tập trung phát triển giao diện người dùng bằng việc tạo mã Frontend cho các ứng dụng di động, nó trở nên thuận tiện và giảm chi phí đáng kể
- Kiến trúc không có máy chủ (Serverless Architecture)
- Machine Learning (học máy) với các APO có nhiều tính năng khác nhau giúp nhận diện văn bản, khuôn mặt, quét mã vạch...
- Tạo lưu lượng truy cập
- Theo dõi lỗi bằng tính năng Crashlytics, nó giúp tìm và khắc phục sự cố nhanh chóng
- Sao lưu

Nhược điểm

- Không phải mã nguồn mở: không thể sửa đổi và truy cập mã nguồn nên đây là một lựa chọn không tối ưu
- Trang web Firebase: <https://firebase.google.com> còn bị chặn ở nhiều quốc gia (Trung Quốc..)
- Chỉ hoạt động với Cơ sở dữ liệu NoSQL, việc sử dụng JSON và hầu như không có tính năng SQL nên việc di chuyển dữ liệu không dễ dàng.
- Truy vấn chậm

Một số ứng dụng sử dụng Firebase

- Alibaba
- Thời báo NewYork
- Todoist
- eBay Motors

CHƯƠNG 3. THỰC NGHIỆM ỨNG DỤNG

3.1. Mục đích

Báo cáo được thiết kế đầy đủ theo giao diện của ứng dụng nhằm tạo cái nhìn khái quát nhất của hệ thống. Được minh họa bằng hình ảnh để người xem có thể hiểu rõ và có cái nhìn trực quan, chi tiết hơn về ứng dụng mobile: Coffee Shop.

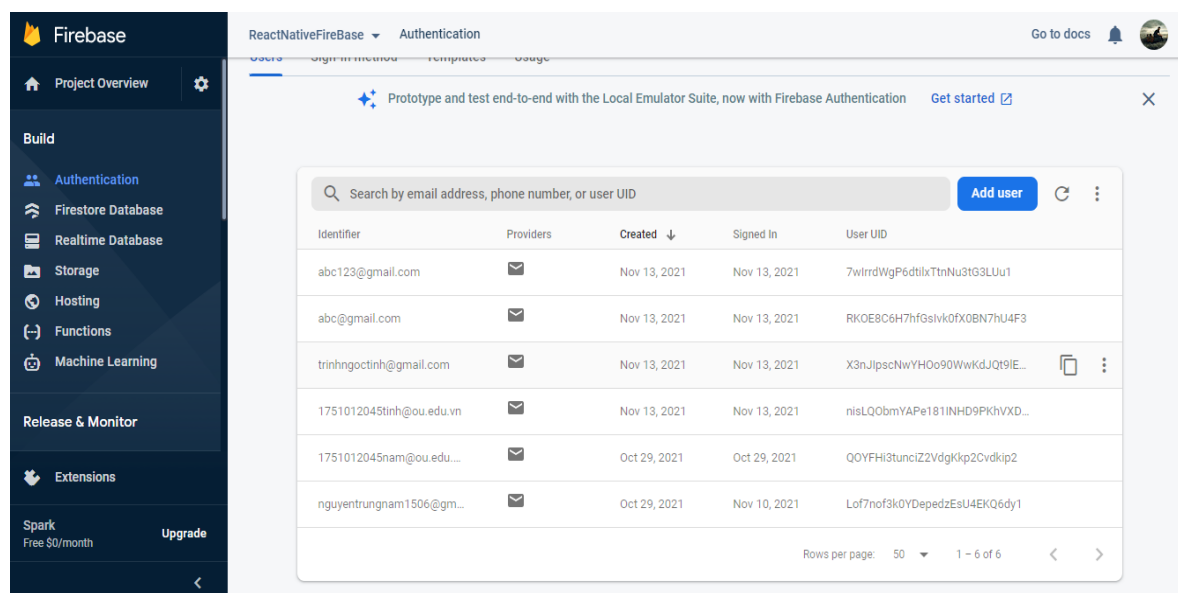
3.2. Chức năng của hệ thống

Giao diện gồm 2 phần chính

- ADMIN: là người quản trị hệ thống, có thể thực hiện đầy đủ các tính năng
- USER: là người dùng, thực hiện các tác vụ của một khách hàng

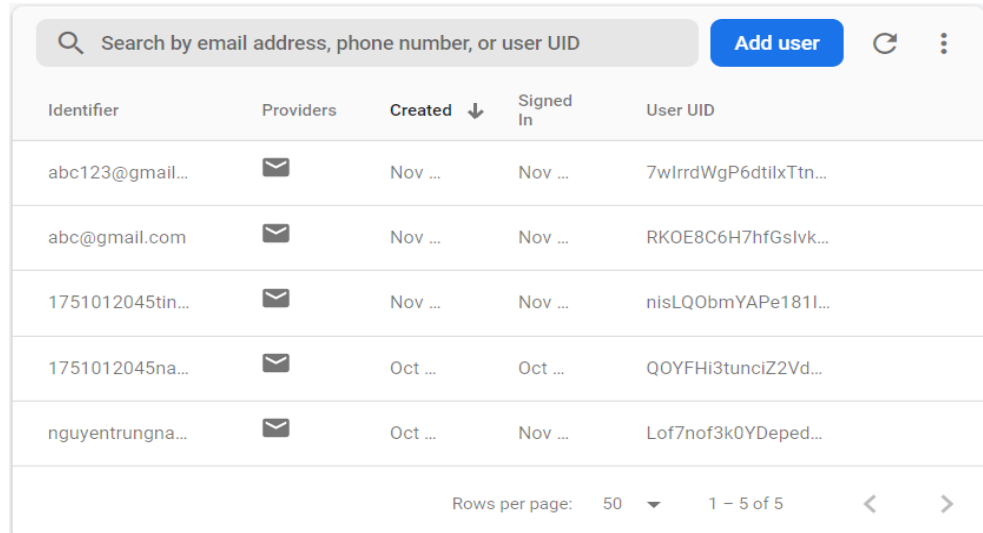
3.2.1. Thực nghiệm giao diện và chức năng quản trị người dùng

Khi quản trị viên (admin) đăng nhập vào hệ thống bằng tài khoản người quản trị trên web firebase: <https://console.firebase.google.com/>. Admin có thể thực hiện mọi chức năng trong hệ thống như xem thông tin, thêm, xóa, sửa và quản lý người dùng ngay trên web. Các dữ liệu sẽ tự động cập nhật trên web và app mobile.



Hình 3.1. Thực nghiệm giao diện quản lý của Admin

Tìm kiếm, thêm, sửa, xóa và vô hiệu hóa tài khoản



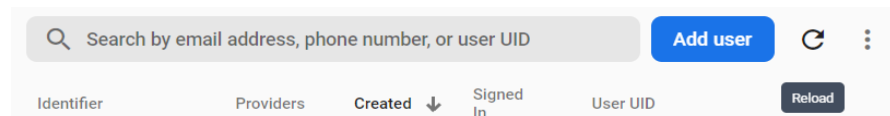
The screenshot shows a user management interface. At the top, there is a search bar with the placeholder text "Search by email address, phone number, or user UID" and a blue "Add user" button. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed In, and User UID. The table contains five rows of user data. At the bottom right, there are pagination controls showing "Rows per page: 50" and "1 - 5 of 5".

Identifier	Providers	Created	Signed In	User UID
abc123@gmail...	✉	Nov ...	Nov ...	7wlrrdWgP6dtlxTtn...
abc@gmail.com	✉	Nov ...	Nov ...	RKOE8C6H7hfGslvk...
1751012045tin...	✉	Nov ...	Nov ...	nisLQObmYAPe181l...
1751012045na...	✉	Oct ...	Oct ...	QOYFHI3tunciZ2Vd...
nguyentrungna...	✉	Oct ...	Nov ...	Lof7nof3k0YDeped...

Hình 3.2. Mô tả các chức năng tìm kiếm, thêm, sửa, xóa và vô hiệu hóa tài khoản

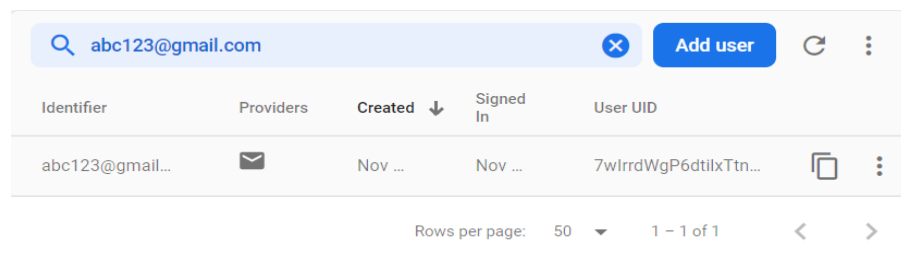
Tìm kiếm người dùng

- Trên giao diện quản lý, admin truy cập vào ô Search để nhập thông tin tìm kiếm của người dùng và ấn vào nút Reload để cho ra kết quả:



Hình 3.3. Nhập thông tin tìm kiếm

- Kết quả:



Hình 3.4. Kết quả tìm kiếm user

Thêm người dùng

Trên giao diện quản lý, admin nhấn vào ô Add user để thêm một người dùng, sau đó nhập thông tin email và password để thêm một user vào hệ thống

Add an Email/Password user

Email

Password

Cancel

Hình 3.5. Nhập thông tin User

User đã được thêm vào hệ thống:

Search by email address, phone number, or user UID <input type="button" value="Add user"/>				
Identifier	Providers	Created ↓	Signed In	User UID
trinhngoctinh@gmail.com		Nov 13, 2021		gelPDYNWoMdDdTT14Ja01vy5yz...

Hình 3.6. Thêm thành công User mới

Sửa mật khẩu

Admin muốn thay đổi mật khẩu của một người dùng bất kỳ, chọn vào view more options sao đó chọn reset password:

Reset password

Disable account

Delete account

Reset password

Send a password reset email. [Review email template](#)

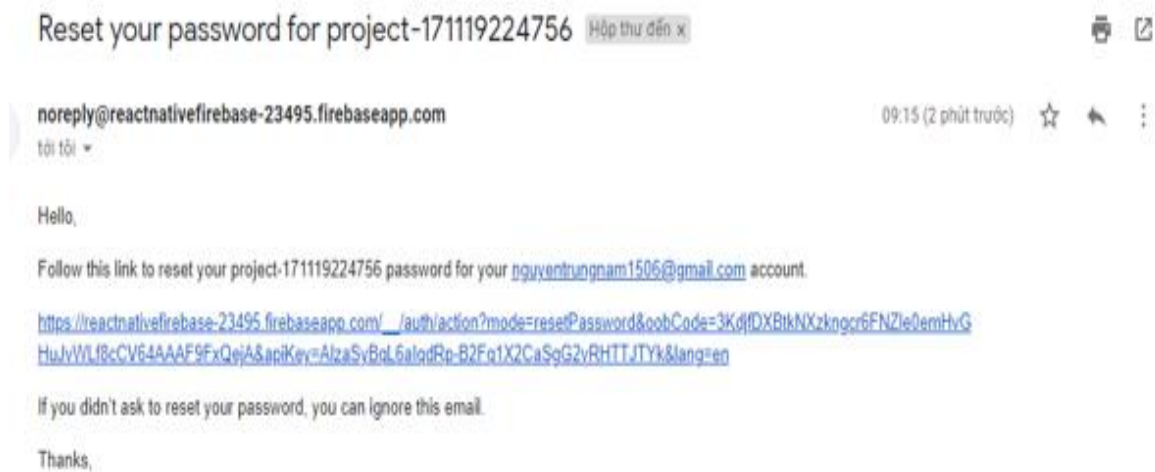
User account

abc123@gmail.com

Cancel

Hình 3.7. Hướng dẫn Sửa mật khẩu của user

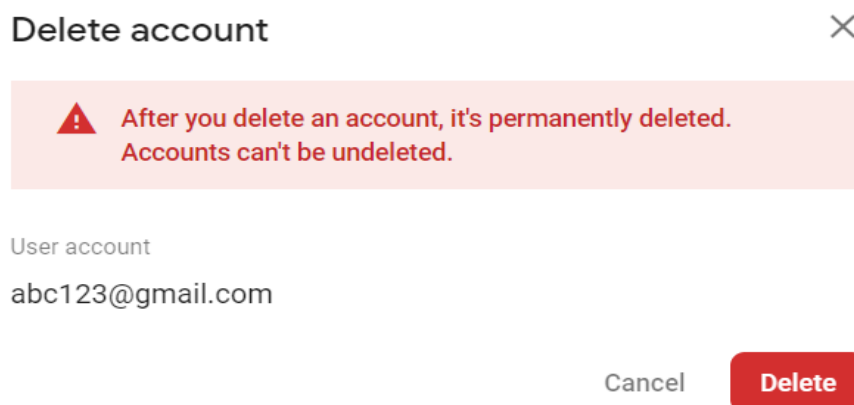
Người dùng sẽ nhận được một email thông báo có chứa đường link để thay đổi mật khẩu account của mình. Đường link này dùng để thay đổi mật khẩu cho account của người dùng.



Hình 3.8. Email thông báo chứa link thay đổi mật khẩu

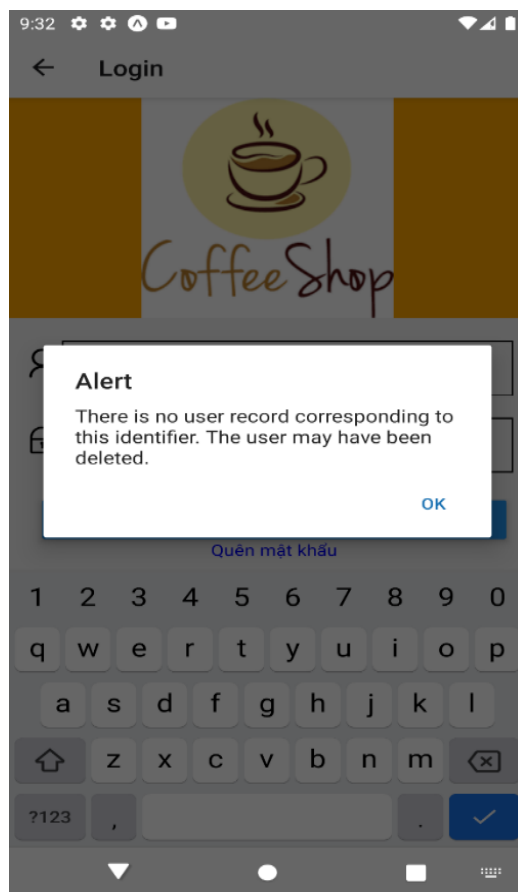
Xóa tài khoản

Tương tự như thay đổi mật khẩu, admin muốn xóa tài khoản chọn vào view more options và chọn vào Delete account:



Hình 3.9. Xóa tài khoản người dùng

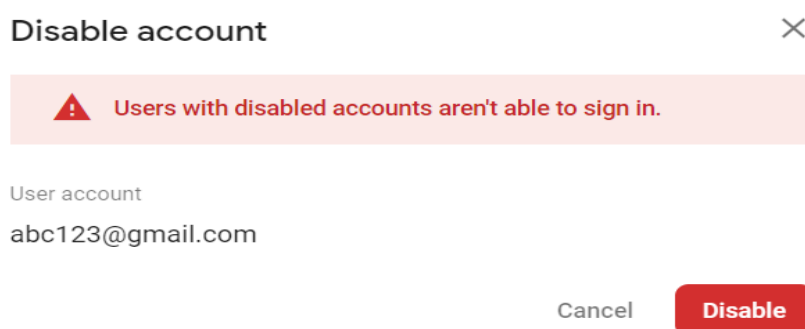
Khi bị xóa tài khoản thì người dùng sẽ nhận được thông báo khi đăng nhập



Hình 3.10. Thông báo tài khoản đã bị xóa

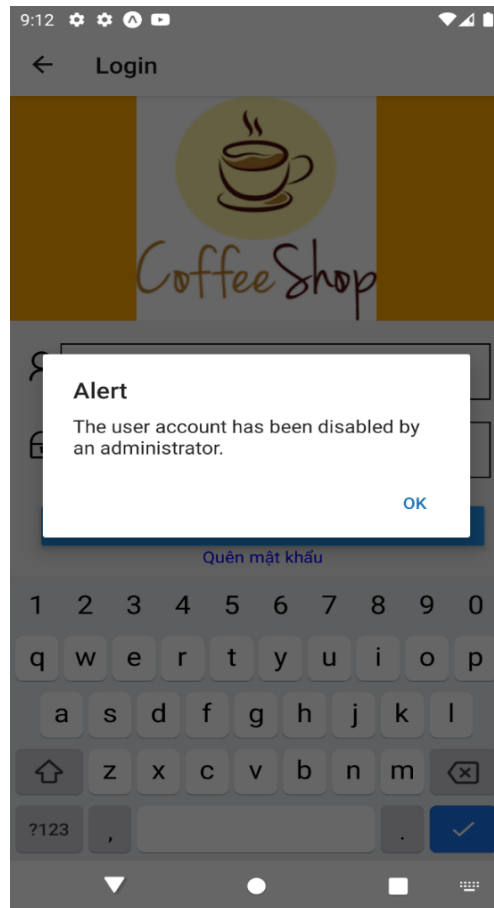
Vô hiệu hóa tài khoản

Tương tự như xóa tài khoản, admin muốn vô hiệu hóa tài khoản chọn vào view more options và chọn vào Disable account. Lúc này hệ thống web hiển thị lên mỗi thông báo rằng bạn User sau khi bị vô hiệu hóa thì không thể đăng nhập vào app mobile nữa:



Hình 3.11. Vô hiệu hóa tài khoản người dùng

Sau khi admin chấp nhận vô hiệu hóa, User khi vào app mobile bằng tài khoản đã bị vô hiệu hóa sẽ nhận được thông báo:



Hình 3.12. Thông báo tài khoản đã bị vô hiệu hóa

3.2.2. Thực nghiệm giao diện và chức năng quản lý sản phẩm của admin

Quản trị viên truy cập vào Realtime Database trên web hệ thống. Admin có thể thực hiện mọi chức năng trong hệ thống như xem thông tin, thêm, xóa, sửa và quản lý sản phẩm ngay trên web. Các dữ liệu sẽ tự động cập nhật trên web và app mobile.



Hình 3.13. Danh sách sản phẩm của hệ thống

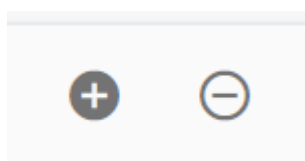
Xem thông tin sản phẩm

Quản trị viên chỉ việc click vào “+” trước mỗi tên sản phẩm để xem thông tin của sản phẩm đó:



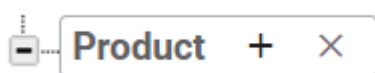
Hình 3.14. Thông tin chi tiết sản phẩm cần xem

Ngoài ra, có thể xem thông tin tất cả các sản phẩm hoặc thu gọn thông tin bằng cách sử dụng các nút sau:



Thêm sản phẩm

Ta chọn dấu + bên trong Product của hệ thống



Nhập thông tin của sản phẩm muốn thêm, có thể ấn dấu + để thêm các thuộc tính mà quản trị muốn. Sau đó ấn Add:

```
Product X
├─ Name Trà Đào + X
│   ├── Name info: Value Trà Đào Thơm X
│   └── Name price: Value $10.0 X
└─ Cancel Add
```

Hình 3.15. Nhập thông tin sản phẩm cần thêm

Sản phẩm Trà Đào đã được thêm vào cơ sở dữ liệu



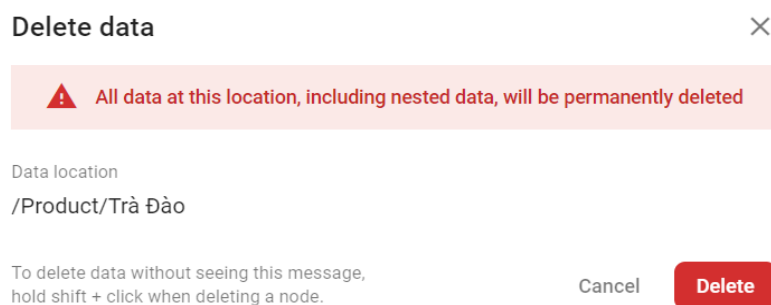
Hình 3.16. Hình sản phẩm đã được thêm vào CSDL

Xóa sản phẩm

Muốn xóa một sản phẩm quản trị viên kéo chuột tới sản phẩm đó rồi chọn dấu X sau tên sản phẩm:



Xuất hiện thông báo xóa sản phẩm:

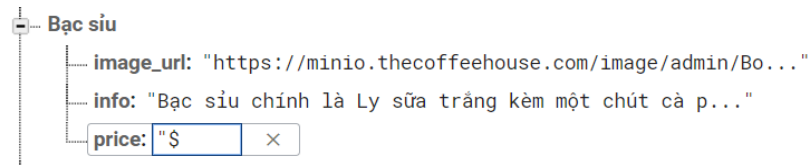


Hình 3.17. Xóa sản phẩm Trà Đào

Sau khi xóa thì tất cả các dữ liệu của sản phẩm biến mất, bao gồm cả dữ liệu lồng nhau cũng bị xóa vĩnh viễn.

Sửa sản phẩm

Muốn sửa sản phẩm quản trị viên cần show thông tin sản phẩm ra sau đó có thể chỉnh sửa trực tiếp thông tin sản phẩm:

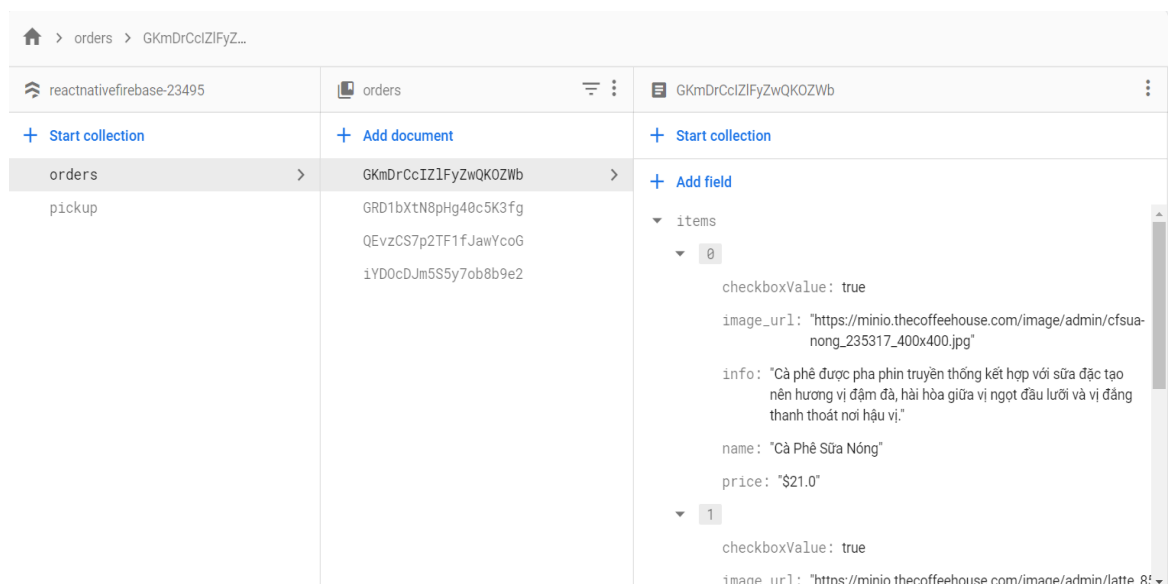


Hình 3.18. Sửa giá của sản phẩm Bạc xỉu

Sau khi sửa thì thông tin của sản phẩm được cập nhật

3.2.3. Thực nghiệm giao diện và chức năng quản lý hóa đơn của admin

Quản trị viên truy cập vào Firestore Database trên web hệ thống để quản lý hóa đơn của ứng dụng



Hình 3.19. Tổng quan về quản lý hóa đơn trên web admin

Quản trị viên có thể tạo các hạng mục để phân loại với từng đơn đặt hàng tại Start collection. Ví dụ tạo hạng mục người dùng đặt đến lấy tại cửa hàng (Pickup):

Start a collection

✓ Give the collection an ID

2 Add its first document

Document parent path ?

/Pickup

Document ID ?

Auto-ID

Field	Type	Value
	= string	

Cancel
Save

Hình 3.20. Tạo hạng mục phân loại đơn đặt hàng

Tạo đơn hàng bằng cách ấn vào Add document. Sau đó nhập vào thông tin của đơn hàng:

Add a document

Parent path /orders

Document ID ?

1

Field	Type	Value
Bạc xỉu	= string	Bạc xỉu 50% đá
Matcha	= string	atcha 50% đường

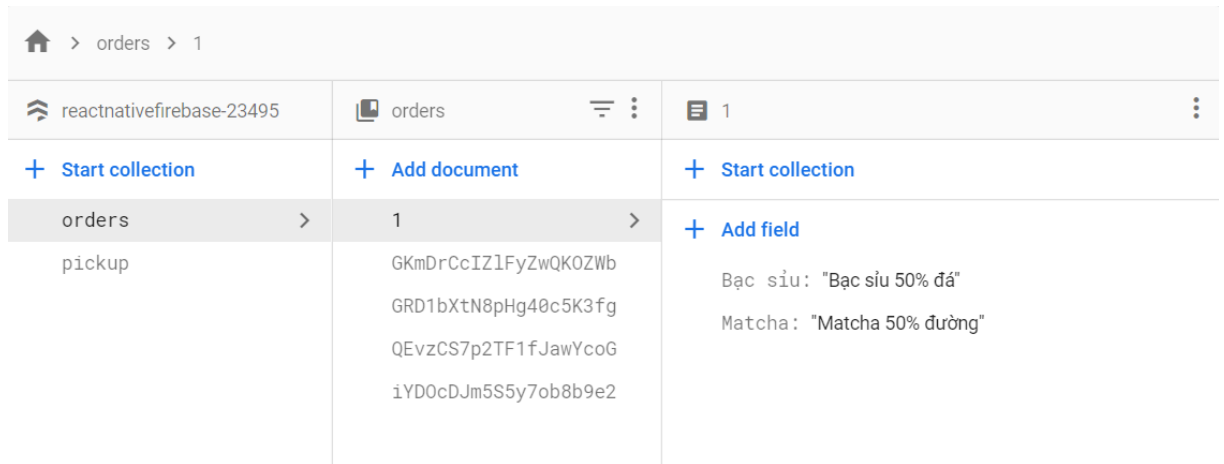
+

 Add field

Cancel
Save

Hình 3.21. Nhập thông tin đơn hàng

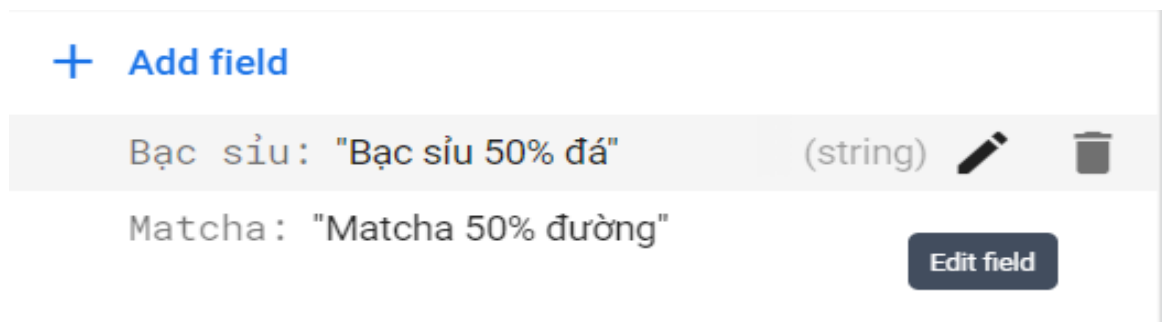
Đơn hàng sau khi thêm được hiển thị trên hệ thống:



reactnativefirebase-23495	orders	1
+ Start collection	+ Add document	+ Start collection
orders >	1 >	+ Add field
pickup	GKmDrCcIZlFyZwQK0ZWb GRD1bXtN8pHg40c5K3fg QEvzCS7p2TF1fJawYcoG iYD0cDJm5S5y7ob8b9e2	Bạc xỉu: "Bạc xỉu 50% đá" Matcha: "Matcha 50% đường"

Hình 3.22. Thông tin đơn hàng có ID: 1

Đơn hàng sau khi thêm có thể sửa và xóa trên thông tin đơn đặt hàng:



+ Add field

Bạc xỉu: "Bạc xỉu 50% đá" (string) [edit icon] [delete icon]

Matcha: "Matcha 50% đường" [Edit field button]

Hình 3.23. Sửa và xóa thông tin đơn đặt hàng

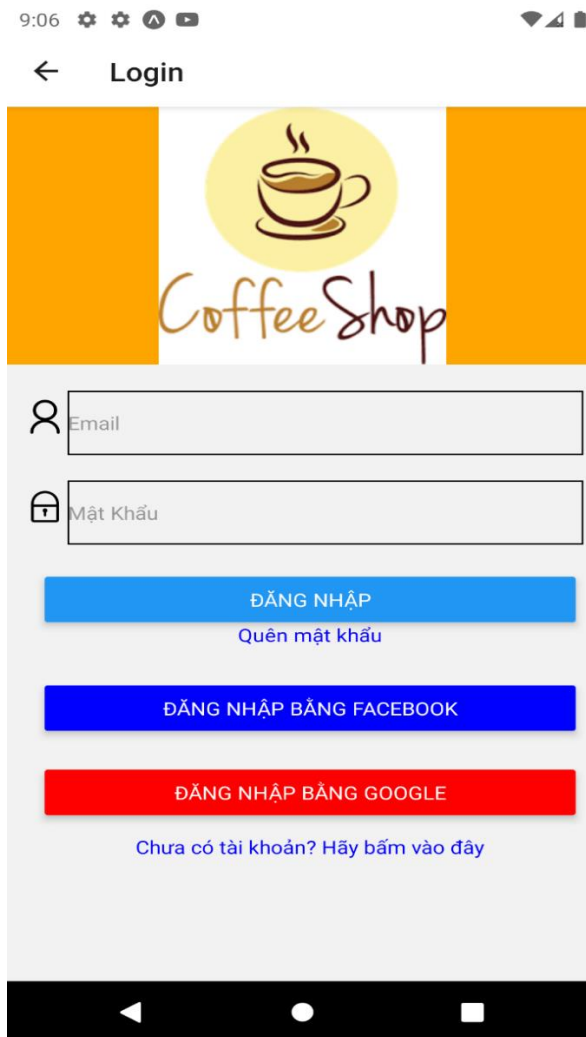
3.2.4. Thực nghiệm giao diện với vai trò người dùng

3.2.4.1. Đăng nhập và đăng ký

Đăng nhập

Mô tả: nhập thông tin tài khoản gồm email và mật khẩu vào form đăng nhập của trang web rồi ấn vào Đăng nhập. Nếu tài khoản đúng thì sẽ chuyển đến trang chủ của ứng dụng, nếu sai thì hệ thống sẽ báo sai thông tin tài khoản.

Đăng nhập giúp người dùng có thể sử dụng ứng dụng ở vai trò User

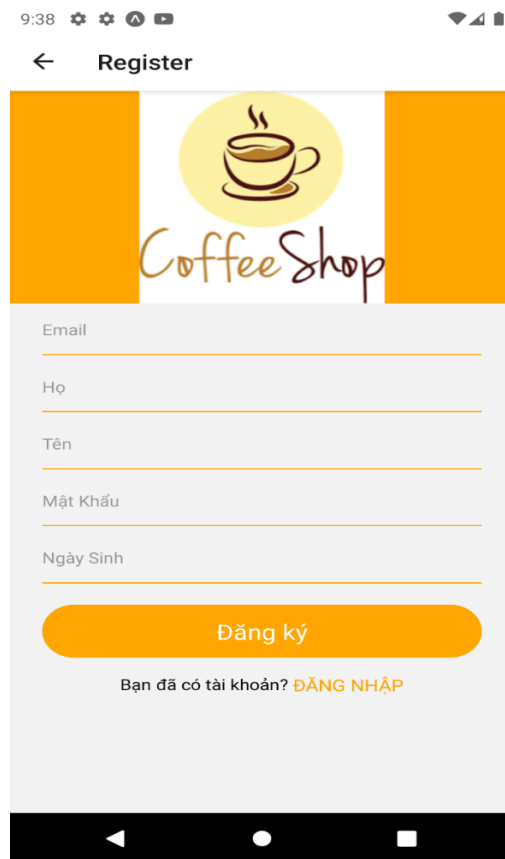


Hình 3.24. Giao diện đăng nhập App

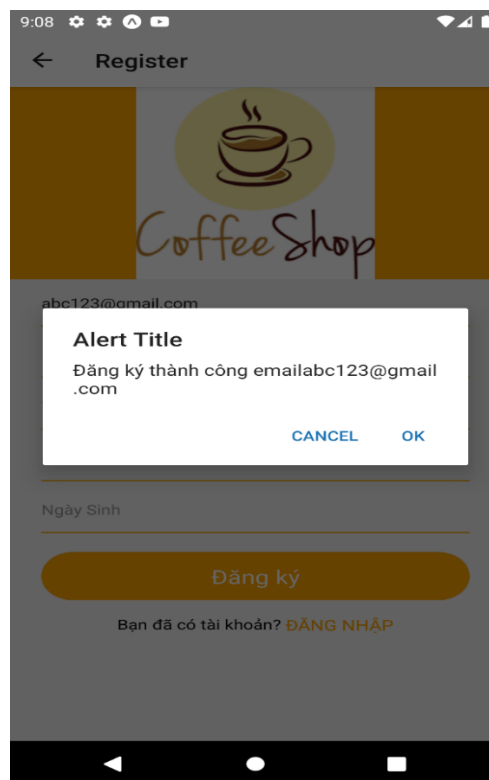
Đăng ký

Mô tả: Người dùng nhập thông tin tài khoản bao gồm: Email, Họ, Tên, Mật khẩu và Ngày sinh để tạo tài khoản sử dụng ứng dụng. Nếu đăng ký thành công, hệ thống sẽ chuyển về giao diện Đăng nhập.

Tạo cho người dùng một tài khoản để sử dụng ứng dụng

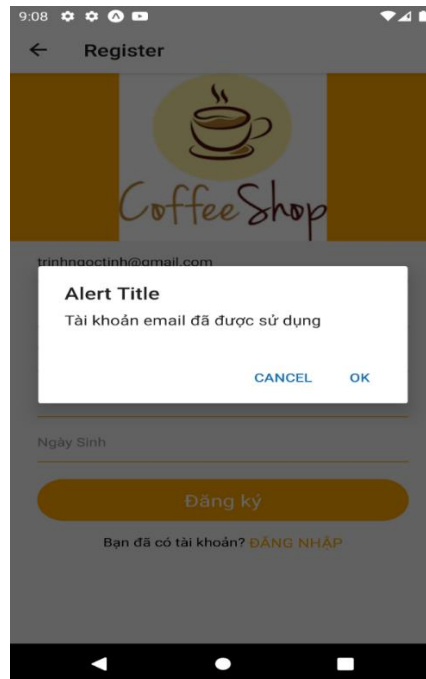


Hình 3.25. Giao diện đăng ký của App
Đăng ký thành công hệ thống xuất thông báo



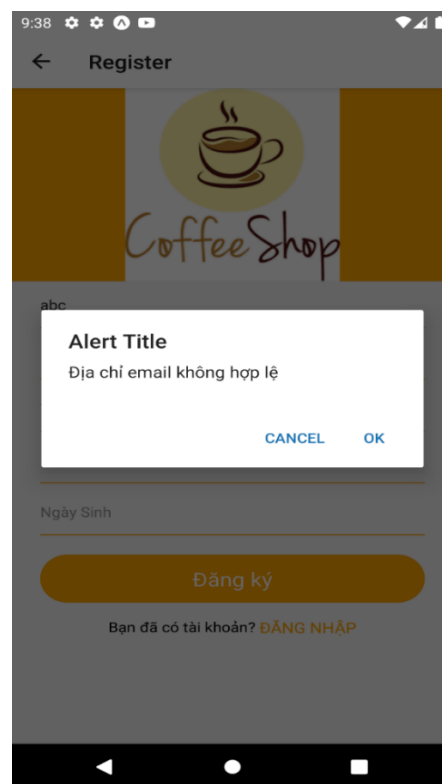
Hình 3.26. Thông báo đăng ký tài khoản thành công

Nếu email đã được đăng ký, hệ thống sẽ xuất thông báo Tài khoản email đã được sử dụng:



Hình 3.27. Thông báo email đã được sử dụng không thể đăng ký

Nếu người dùng nhập email không hợp lệ, hệ thống sẽ xuất thông báo Địa chỉ email không hợp lệ:



Hình 3.28. Thông báo địa chỉ email không hợp lệ để đăng ký

3.2.4.2. Trang chủ ứng dụng

Trang chủ:

Trang chủ của ứng dụng chứa các hình ảnh về sản phẩm, có các mục để khách hàng lựa chọn như Cửa hàng, Đặt hàng, Thông báo, Khác và nhiều lựa chọn khác. Giúp khách hàng có thể lựa chọn các chức năng mà cần sử dụng.



Hình 3.29. Trang chủ ứng dụng Coffee Shop

Xem các ưu đãi của chương trình bằng các thông tin sản phẩm, thời gian ưu đãi và chính sách ưu đãi để khách hàng có thể dễ dàng chọn lựa những sản phẩm vừa ý

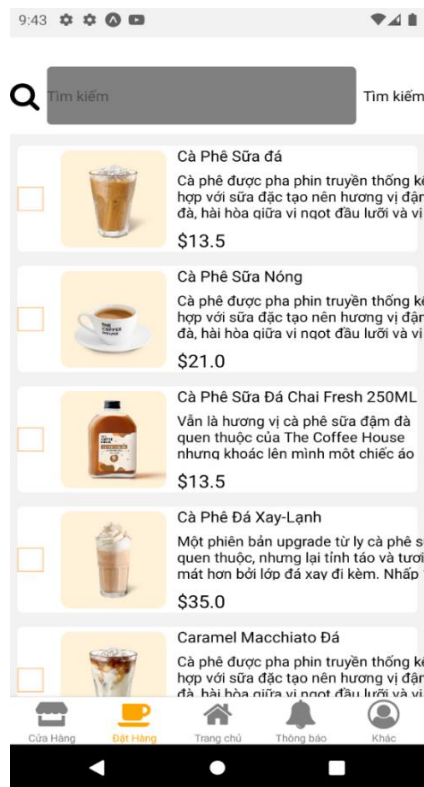


Hình 3.30. Chính sách ưu đãi của ứng dụng

Xem danh sách và tìm kiếm sản phẩm

Mô tả: Người dùng muốn xem và tìm kiếm sản phẩm thì qua chức năng đặt hàng để sử dụng. Trên ô tìm kiếm người dùng nhập keyword để tìm kiếm loại sản phẩm người dùng cần, người dùng có thể xem thông tin sản phẩm tìm kiếm được ngay trên giao diện màn hình ứng dụng về tên, thông tin chi tiết và giá tiền của sản phẩm

Chức năng này giúp người dùng nhanh chóng tìm kiếm sản phẩm theo ý muốn

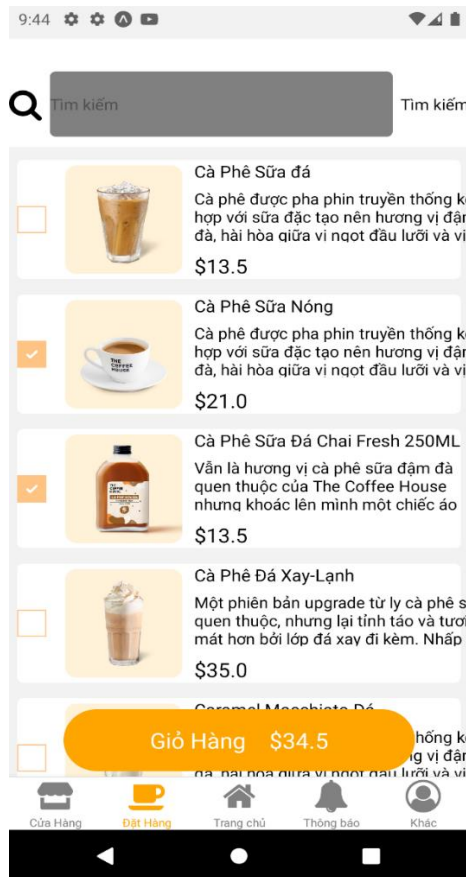


Hình 3.31. Danh sách các sản phẩm

Thêm sản phẩm vào giỏ hàng

Mô tả: Khách hàng chọn sản phẩm cần thêm vào giỏ hàng bằng cách tick vào ô trước hình ảnh sản phẩm cần mua. Có thể chọn một hoặc nhiều sản phẩm và hệ thống sẽ hiển thị ô thông báo nhỏ phía dưới về số tiền trong giỏ hàng mà quý khách đã thêm vào giỏ.

Chức năng này giúp cho người dùng có thể chọn được nhiều sản phẩm khi mua và dễ dàng quản lý các sản phẩm cũng như giá tiền trong giỏ hàng

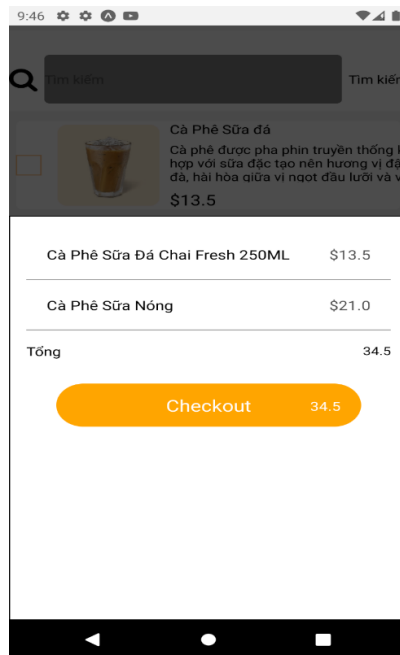


Hình 3.32. Thêm các sản phẩm vào giỏ hàng

Xem giỏ hàng và thanh toán

Mô tả: Các sản phẩm được bỏ vào giỏ hàng sẽ được người dùng xem qua trước khi người dùng muốn ấn đặt hàng. Người dùng có thể bỏ chọn những sản phẩm không muốn mua nữa và sau đó xác nhận thanh toán bằng nút Checkout.

Giúp cho người dùng xem xét kỹ và tiến hành thanh toán một cách nhanh chóng



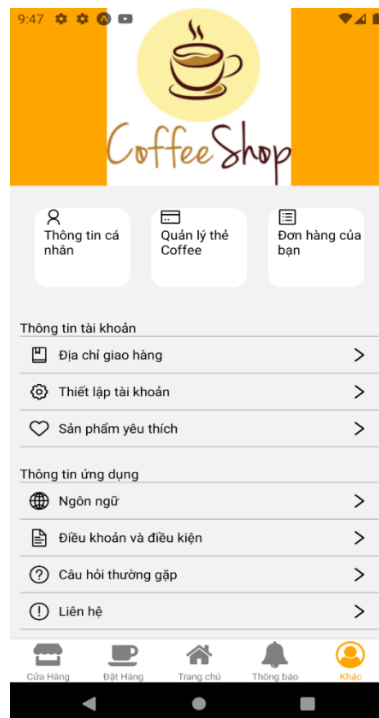
Hình 3.33. Thanh toán giỏ hàng

3.2.4.3. Thông tin tài khoản người dùng

Khách hàng xem thông tin của mình trong mục Khác. Chứa các thông tin cá nhân, Địa chỉ giao hàng...

Ngoài ra khách hàng còn có các chức năng khác chưa hoàn thiện

Tại đây khách hàng có thể đăng xuất tài khoản của mình



Hình 3.34. Giao diện thông tin account

CHƯƠNG 4. TỔNG KẾT

4.1. Kết quả đạt được

- Đề tài đã phát triển tương đối các tính năng của một ứng dụng mobile về bán hàng (coffee, thức uống)
- Giao diện dễ nhìn và dễ sử dụng
- Tìm hiểu và học hỏi được nhiều kiến thức mới
- Áp dụng được các kiến thức về ReactNative và quản lý dữ liệu trên Firebase
- Được thực tế với các ứng dụng, tiếp cận hơn với nhu cầu tiêu dùng bên ngoài
- Trao đổi và có thêm kinh nghiệm làm việc với nhiều công nghệ, kiến thức mới thuận tiện cho công việc sau này.

4.2. Hạn chế

- Đề tài này khá phổ biến, không mới mẻ và chưa có nhiều sáng tạo
- Chưa tạo được thống kê, biểu đồ về doanh số
- Một số chức năng sơ sài, chưa hoàn thiện
- Dữ liệu nhập tay nên còn hạn chế

4.3. Hướng phát triển đề tài

- Nâng cao khả năng truy vấn API để giúp tốc độ xử lý nhanh hơn
- Tạo thêm phần tương tác với người dùng
- Hy vọng xây dựng lên một trang web tích hợp kết nối với app mobile
- Đưa dự án vào thực tế, có một số ít người dùng thử và lấy ý kiến để phát triển thêm

TÀI LIỆU THAM KHẢO

1. Giáo trình Quản trị hệ cơ sở dữ liệu. Giảng viên: ThS. Hồ Quang Khải – Đại học mở Tp HCM.
2. Tài liệu tham khảo Internet:
 - <http://viblo.asia>
 - <http://stackoverflow.com>
 - <http://reactnative.dev>
 - <http://w3schools.com>
 - <http://rnfirebase.io>
 - <http://firebase.google.com>

PHỤ LỤC

Để truy cập vào Database của đồ án trong FireBase ta thực hiện các bước:

- Truy cập vào website: <https://firebase.google.com/>
- Chọn Đăng nhập với tài khoản và mật khẩu:
 - Email: nguyentrungnam1506@gmail.com
 - Mật khẩu: nguyentrungnam
- Chọn Get Started
- Chọn Mục ReactNativeFireBase để vào Giao diện quản lý của Admin

Sau khi truy cập vào giao diện có thể nắm quyền Admin của hệ thống.