**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT - CÔNG NGHỆ**

****

**TIỂU LUẬN CUỐI KỲ**

**MÔN HỌC:**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG ĐA NỀN TẢNG**

**ĐỀ TÀI: XÂY DỰNG VÀ PHÁT TRIỂN ỨNG DỤNG BÁN CÂY XANH**

**GVHD : TRẦN VĂN HỮU**

**NHÓM THỰC HIỆN**

|  |  |
| --- | --- |
| ***Cao Văn Tâm*** | ***1824801030183*** |
| ***Phan Thế Nhựt*** | ***1824801030209*** |
| ***Lưu Bá Đức*** | ***1824801030145*** |

*Bình Dương, tháng 4 năm 2021*

**LỜI CẢM ƠN**

Trong thời gian nghiên cứu và học tập môn Phát triển ứng dụng di động đa nền tảng,cùng với việc tìm hiểu và đọc các tài liệu trên thư viện Trường Đại học Thủ Dầu Một cũng như tìm hiểu thêm tài liệu học tập trên internet, nhóm em đã gặp không ít khó khăn về việc xây dựng ứng dụng bán hàng sử dụng React Native và Node js. Với sự giúp đỡ của các bạn trong lớp, nhóm em đã hoàn thành Ứng dụng bán hang cụ thể là một App bán cây xanh

Đồng thời chúng em xin gửi lời cảm ơn đặc biệt về sự giảng dạy nhiệt tình của thầy **Trần Văn Hữu** đã giúp chúng em cũng cố kiến thức cho môn học, cùng với sự góp ý, hỗ trợ nhiệt tình của các bạn trong lớp**.**

Tuy nhiên, kinh nghiệm còn thiếu nên trong sản phẩm cũng như bài báo cáo này chắc chắn sẽ không tránh khỏi những thiếu sót, hạn chế nhất định. Những ý kiến nhận xét và góp ý quý báu của thầy và các bạn là cơ sở để nhóm em học hỏi thêm và hoàn thiện thêm kiến thức và củng cố thêm kinh nghiệm của bản thân mình. Mặc dù đã rất nỗ lực và cố gắng nhưng chắc chắn rằng đề tài vẫn còn nhiều thiếu sót, chúng em mong sẽ nhận đuợc sự góp ý, phê bình của thầy và các bạn để đề tài hoàn thiện hơn.

**Em xin trân thành cảm ơn!**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

*Thủ Dầu Một, Ngày tháng năm 2019*

## Giáo viên hướng dẫn

**LỜI NÓI ĐẦU**

Ngày nay công nghệ thông tin trở thành một phần không thể thiếu trong cuộc sống, công nghệ thông tin đã và đang đóng vai trò quan trọng trong các ngành khoa học kĩ thuật, kinh tế xã hội của mỗi nước trên thế giới. Nó không những giải quyết công việc một cách nhanh chóng mà còn đem lại được hiệu quả kinh tế cao.

Sau quá trình học tập và nghiên cứu về lập trình viết ứng dụng và được sự giảng dạy nhiệt tình của thầy, nhóm chúng em cũng đã hoàn thành xong báo cáo đồ án môn học Lập Trình ứng dụng đa nền tảng “ Xây Dựng và phát triển ứng dụng bán cây xanh “.

Quá trình thực hiện báo cáo gặp nhiều khó khăn vì ngôn ngữ lập trình mới, kinh nghiệm thực tế còn hạn chế, chúng em mong sẽ nhận được sự góp ý, phê bình của thầy để đề tài này hoàn thiện hơn.

Chúng em xin chân thành cảm ơn sự giúp đỡ của thầy!

**MỤC LỤC**

**CHƯƠNG I:KHẢO SÁT VÀ THU THẬP YÊU CẦU**

**1.Tên đề tài**

**-**Xậy dựng ứng dụng Bán cây xanh thân thiện với môi trường

**1.1 Thông tin cá nhân**

**-**Họ tên sinh viên: Cao Văn Tâm

-Lớp:D18PM03

-MSSV: 1824801030183

**-**Họ tên sinh viên: Phan Thế Nhựt

-Lớp:D18PM03

-MSSV: 1824801030209

**-**Họ tên sinh viên: Lưu Bá Đức

-Lớp:D18PM03

-MSSV:

**1.2 Các công nghệ trong đề tài**

* **React Native là gì?**

React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái luôn, sợ chưa!!!

Một điểm hay ho nữa mà mình có đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là bạn không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa

* **Dịch vụ cơ sở dữ liệu Firebase**

**Platform do Google cung cấp, nhằm hỗ trợ việc tạo ra các web application, mobile application với chất lượng cao.**

Với việc sử dụng Firebase, developer có thể tập trung vào việc phát triển application mà không cần lo về việc sản phẩm của mình sẽ hoạt động và được quản lý thể nào ở phía Backend.

Khái niệm Backend nhắc tới trong câu trên để chỉ việc xử lý và lưu trữ dữ liệu trong service, ở vùng "phía sau" mà người dùng không nhìn thấy.

Firebase là một trong những BaaS (Backend as a service).

Serverless architecture được chú ý như một giải pháp đáp ứng yêu cầu release một dịch vụ trong thời gian ngắn, trong bối cảnh đó, BaaS ra mắt người dùng.

BaaS hướng đến thiết bị chủ yếu là mobile nên còn được gọi với một tên gọi khác là MBaaS.

* **Các chức năng chính của Firebase**

**+** Firebase Analytics:

Bằng Firebase Analytics, ta có thể phân tích sự tương tác của người dùng với ứng dụng cũng như tình trạng sử dụng ứng dụng đó.

Firebase có sẵn chức năng tạo report. Nhìn report ấy, chúng ta có thể nắm được một cách rõ ràng hoạt động của người dùng.

Bằng việc cài đặt SDK (Software Development Kit, cụ thể hơn với trường hợp này là FirebaseAnalytics.unitypackage), chức năng phân tích các access trở nên khả dụng. Khi đó, ta không chỉ xem được hoạt động của người dùng mà còn có thể biết được thông tin về thuộc tính hoạt động hay hiệu quả quảng cáo, tình trạng trả phí, v.v.

+ Firebase Hosting:

Firebase Hosting là dịch vụ deploy trang web và web app chỉ bằng những thao tác đơn giản.

Khi tạo một app, nhiều trường hợp nhà phát triển sẽ muốn xử lý một số chức năng trên trang web.

Ví dụ tiêu biểu nhất là trường hợp của Điều khoản dịch vụ (Terms of service) hoặc Chính sách bảo mật (Privacy policy), ta thường sẽ sử dụng trang web có sẵn để hiển thị chúng thay vì tạo riêng một trang trong application.

Khi sử dụng Firebase Hosting, ta có thể sử dụng web page cho ứng dụng iOS và Android của mình.

Với các ưu điểm thao tác nhanh, đơn giản, tính an toàn cao, Firebase hosting giúp các nhà phát triển tạo lập ứng dụng một cách đơn giản hơn so với việc tự tạo hay thuê server thực.

+ Firebase Cloud Messaging:

Với Firebase Cloud Messaging, ta có thể gửi nhận tin nhắn miễn phí. Khái niệm "Gửi message" ở đây còn dùng để chỉ việc push thông báo.

Khi tên nhắn mới được gửi tới, người dùng có thể nhận được thông báo. Ví dụ, với trường hợp người dùng sử dụng iPhone, khi có mail hay thông báo từ application, ở góc trên bên phải icon của app đó sẽ xuất hiện dấu tròn nhỏ màu đỏ, hay tiện lợi hơn, trong dấu tròn đỏ ấy sẽ hiển thị số lượng thông báo, tin nhắn mới.

Thêm vào đó, ta cũng có thể thiết lập để tin nhắn chỉ gửi tới những đối tượng mà ta mong muốn, ví dụ như việc gửi thông báo yêu cầu viết feedback sản phẩm cho những khách hàng thường xuyên sử dụng sản phẩm.

+Firebase Authentication:

Firebase Authentication là chức năng dùng để xác thực người dùng bằng Password, số điện thoại hoặc tài khoản Google, Facebook hay Twitter, v.v.

Việc xác thực người dùng là một chức năng quan trọng trong phát triển ứng dụng. Tuy nhiên, việc đối ứng với nhiều phương pháp xác thực khác nhau sẽ tốn nhiều thời gian và công sức.

Firebase Authentication giúp thực hiện việc chia sẻ ID giữa các ứng dụng, giúp người dùng dễ dàng tiếp cận sản phẩm hơn. Vì thế, nó là một chức năng rất quý.

+Firebase Cloud Storage:

Firebase Cloud Storage hỗ trợ việc quản lý, chia sẻ các content người dùng upload lên như ảnh, video; cũng như sử dụng những tài nguyên ấy cho ứng dụng của bạn.

Data được lưu trữ trong Google Cloud Storage buckets, và có thể được access từ server.

Đây cũng là một chức năng quan trọng và tiện lợi cho việc phát triển dịch vụ.

+Firebase Remote Config:

Firebase Remote Config giúp thực hiện việc thay đổi UI cũng như hoạt động của ứng dụng mà không cần phải update app đó.

Việc này thực hiện bằng cách set giá trị ban đầu (giá trị default) cho giao diện và hoạt động của app, sau đó sử dụng Remote Config REST API hoặc Firebase Console để ghi đè giá trị mới lên setting default.

+Firebase Test Lab:

Firebase Test Lab được sử dụng để test ứng dụng trên nền tẳng cloud.

Bằng cách sử dụng Test Lab, ta có thể test hoạt động của ứng dụng trên nhiều thiết bị đa dạng, nhờ đó có thể xác minh được app sẽ chạy thế nào trên thiết bị thực của người dùng.

+Firebase Crashlytics:

Firebase Crashlytics là công cụ báo cáo các hoạt động bất thường (crash) phát sinh trên ứng dụng theo thời gian thực (real time).

Tool giúp ích cho việc truy vết các vấn đề bất thường hạ thấp chất lượng sản phẩm, nhờ đó ta có thể đặt thứ tự ưu tiên để có phương hướng xử lý thích hợp.

Công cụ này còn hỗ trợ việc chia nhóm các bất thường phát sinh, truy xuất tình trạng có liên quan đến crash ấy, giúp tiết kiệm thời gian xử lý sự cố.

+Firebase App Indexing:

Firebase App Indexing là chức năng hiển thị content trong ứng dụng trong kết quả tìm kiếm Google.

Với user đã cài đặt sẵn ứng dụng trên thiết bị của mình, khi user chọn kết quả tìm kiếm có chứa content trong ứng dụng thì application sẽ được khởi động.

+Firebase Dynamic Links:

Firebase Dynamic Links là chức năng phân chia trang đích một cách thích hợp tuỳ vào tình trạng người dùng.

+Firebase Console:

Để sử dụng những chức năng được giới thiệu ở trên, ta cần đến Firebase console.

Để login vào console, ta cần có tài khoản Google, vậy bước trước tiên cần được thực hiện là đăng ký một tài khoản Google nếu bạn chưa sở hữu, và đăng nhập với account đó.

Thêm vào đó, bằng việc sử dụng Notifications Composer của Firebase Console, ta có thể gửi tin nhắn thông báo tới thiết bị iOS hay Android.

**1.3 Khảo sát hiện trạng**

* **Đặt vấn đề**

Thương mại điện tử là hình thức kinh doanh **trực tuyến** sử dụng nền tảng công nghệ thông tin với sự hỗ trợ của Internet để thực hiện các giao dịch mua bán, trao đổi, thanh toán trực tuyến. Đồng thời, [thương mại](https://thuongtruong.com.vn/thuong-truong/thuong-mai.html?utm_source=danhmuc&utm_medium=danhmucnoidung&utm_campaign=th%C6%B0%C6%A1ng%20m%E1%BA%A1i) điện tử cũng là xu hướng của thời đại toàn cầu hóa, đây là lĩnh vực tiềm năng để các doanh nghiệp vừa và nhỏ sinh lợi và phát triển, cơ hội cho những ai muốn khởi nghiệp kinh doanh theo mô hình mới. Mô hình kinh doanh [thương mại](https://thuongtruong.com.vn/thuong-truong/thuong-mai.html?utm_source=danhmuc&utm_medium=danhmucnoidung&utm_campaign=th%C6%B0%C6%A1ng%20m%E1%BA%A1i) điện tử được xem như một trong những giải pháp thúc đẩy sự phát triển của nền [kinh tế](https://thuongtruong.com.vn/kinh-te.html?utm_source=danhmuc&utm_medium=danhmucnoidung&utm_campaign=kinh%20t%E1%BA%BF) quốc gia.

Do đó, thương mại điện tử hay bán lẻ đều là một hình thức kinh doanh rất phổ biến và ngày càng phát triển khi nhu cầu mua sắm của người tiêu dùng thay đổi. Ngày nay, các cơ sở bán hàng truyền thống, các trung tâm siêu thị không còn là giải pháp duy nhất và có sức hút đối với người tiêu dùng, mà họ có thêm sự lựa chọn rất tiện tích đó là mua sắm trực tuyến thông qua nền tảng thương mại điện tử.

* **Xu thế bán hàng trực tuyến hiện nay**

Theo Báo cáo của Google-Temasek năm 2018, thị trường thương mại điện tử Việt Nam sẽ bùng nổ vào năm 2025, với doanh thu ở mức 7,5 tỷ USD. Đây là con số rất hấp dẫn để các doanh nghiệp bán lẻ thúc đẩy kênh bán hàng online, đón đầu sự phát triển của thị trường.

Còn theo số liệu của Bộ Công Thương, doanh thu thương mại điện tử tiếp tục tăng trưởng ổn định 20%/năm; dự kiến đến năm 2020, doanh số bán hàng thương mại điện tử đạt 10 tỷ USD, chiếm 5% tổng mức bán lẻ hàng hóa và doanh thu dịch vụ tiêu dùng của cả nước; và khoảng 30% dân số sẽ mua sắm online và đạt mốc 350 USD/người/năm.

Với đa phần số dân trong độ tuổi trẻ, Việt Nam nhiều năm liền nằm trong nhóm thị trường bán lẻ hấp dẫn nhất thế giới. Ngành Bán lẻ vì vậy cũng trải qua quá trình biến chuyển và tiến hóa cùng “cơn lốc” mua sắm trực tuyến để đáp ứng nhu cầu tiêu dùng đang thay đổi từng ngày của khách hàng.

Vì vậy đối với xu thế của thế giới hiện nay việc tạo ra một App bán hang hay một kênh bán hang online là một điều tất yếu cần có cho sự phát triển và xu thế của thế giới trong tương lai sau này cũng như hiện tại.

* **Những tồn tại hạn chế**

Tuy đạt được những kết quả nhất định trong phát triển kinh doanh bán lẻ trên nền thương mại điện tử (TMĐT), song theo nhận định của các chuyên gia trong ngành, một số hạn chế vẫn còn tồn tại về lĩnh vực này.

Cụ thể, việc ứng dụng TMĐT phục vụ hoạt động sản xuất - kinh doanh của doanh nghiệp đã trở nên phổ biến, nhưng phần lớn doanh nghiệp bán lẻ chưa có chiến lược, cũng như đầu tư thích đáng để khai thác các ứng dụng TMĐT theo chiều sâu, chưa chuẩn bị sẵn sàng để bắt kịp những tiến bộ công nghệ mới về TMĐT của cuộc cách mạng công nghiệp 4.0. Hệ sinh thái cho TMĐT bao gồm dịch vụ thanh toán, logistics và các dịch vụ phụ trợ còn chưa hoàn thiện. Tỷ lệ thanh toán bằng tiền mặt trong giao dịch TMĐT vẫn còn cao. Tuy nhiên, hạ tầng, chất lượng vận chuyển, thời gian vận chuyển, thủ tục thông quan, chi phí, các dịch vụ theo dõi và truy xuất đơn hàng vẫn chưa phát triển đồng bộ và còn bộc lộ nhiều hạn chế.

* **Cách khắc phục**

Nghiên cứu, đề xuất, sửa đổi, bổ sung các quy định về xử lý vi phạm hành chính trong lĩnh vực TMĐT, đặc biệt với đối tượng cung cấp thông tin hàng giả, hàng không rõ nguồn gốc xuất xứ, hàng cấm trên môi trường điện tử. Đồng thời, triển khai các giải pháp toàn diện nhằm xây dựng hệ sinh thái cho TMĐT và kinh tế số, hỗ trợ doanh nghiệp ứng dụng các công nghệ tiên tiến của cách mạng công nghiệp 4.0.

**1.4 Mô tả bài toán**

**Đăng kí:** Người dùng sẽ đăng kí để có thế truy cập vào ứng dụng nếu người dùng chưa có tài khoản đăng nhập, để đăng kí thành công vào ứng dụng khác hang sẽ đăng kí theo Gmail và password sau đó xác nhận lại một lần nữa và tiến hành đăng kí tài khoản.

**Đăng nhập:** Người dùng đăng nhập vào hệ thống bằng tài khoản vốn có của họ sau đó màn hình sẽ chuyển sang giao diện chính ứng dụng bán hàng

**Tìm kiếm:** Người dùng khi muốn tìm kiếm một sản phẩm nào đó có thể nhấp vào biểu tượng tìm kiếm sau đó nhập từ khoá cần tìm sau đó sẽ hiện ra kết quả mà mình tìm kiếm

**Chi tiết sản phẩm:** Người dùng sau khi chọn sản phẩm muốn xem sẽ chuyển sang màn hình chi tiết sản phẩm bao gồm các thông tin về sản phẩm như giá,số lượng,mô tả sản phẩm.

**Giỏ hàng:** Người dùng muốn mua hàng có thể nhấp vào nút them giỏ hàng cùng với số lượng mà khác hàng muốn mua sau đó chuyển tới màn hình của trang giỏ hàng, tại đây khách hàng có thể xem được sản phẩm mình đã them vào giỏ hàng sau đó có thể tiến hành thanh toán hoặc là xoá giỏ hàng và giảm số lượng giỏ hàng nếu khách hàng đổi ý.

**1.5 Mô tả yêu cầu**

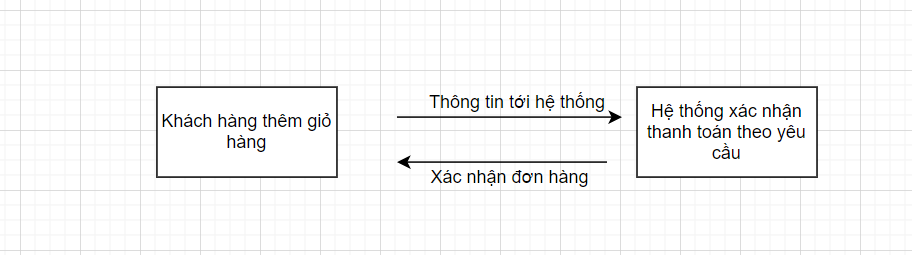
* **Yêu cầu chung về ứng dụng bán hàng**

-Hệ thống bao gồm các chức năng như bán hàng, đặt hàng thông tin trong tin trong cơ sở dữ liệu

* **Thanh toán hoá đơn**

-Chọn các sản phẩm mình mua trong danh mục sản phẩm mà khách hàng yêu cầu(số lượng,tên sản phẩm,đơn giá)

-Thanh toán và xác nhận đơn hàng

****

**Hình 1.5. Yêu cầu của bài toán**

**1.6 Chức năng của bài toán**

* **Chức năng đăng ký**

-Khách hàng chọn nút đăng kí

-Nhập Email và passworld muốn đăng kí

-Xác nhận passworld

-Nhấn nút đăng kí để xác nhận thông tin và tạo tài khoản

* **Chức năng đăng nhập**

**-**Khách hàng nhập Email và passworld với tài khoản đã đăng kí

-Xác nhận đăng nhập bằng nút đăng nhập

-Chuyển tới giao diện chính của ứng dụng

* **Chức năng tìm kiếm**

-Khách hàng chọn biểu tượng tìm kiếm

-Nhập từ khoá muốn tìm kiếm

-Hiện thị kết quả theo yều cầu cảu khách hàng

* **Chức năng xem chi tiết sản phẩm**

-Chọn sản phẩm muốn xem

-Ứng dụng chuyển sang màn hình trang chi tiết sản phẩm

* **Chức năng giỏ hàng**

-Thêm sản phẩm muốn mua tại trang chi tiết sản phẩm

-Chỉnh số lượng sản phẩm muốn mua

-Mở giỏ hàng lên kiểm tra và chỉnh sửa giỏ hàng thêm,xoá các sản phẩm

-Xác nhận thanh toán đơn hàng

**CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. **Chức năng phía ứng dụng**

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Mô tả |
| 1 | Xem hàng | Người xem có thể xem thông tin về các loại cây bao gồm:Tên cây,loại cây,mô tả,giá |
| 2 | Đặt hàng | Người dùng có thể chọn các sản phẩm mình muốn mua và đặt hàng |
| 3 | Tìm kiếm sản phẩm | Người dùng có thể tìm kiểm sản phẩm theo các từ khoá |
| 4 | Đăng nhập | Người dùng đăng nhập vào hệ thống với các thông tin đăng kí |

1. **Xác định các use case,xây dựng các biểu đồ use case chi tiết**

Tác nhân khách hàng có các use case như sau:

-use case toàn hệ thống

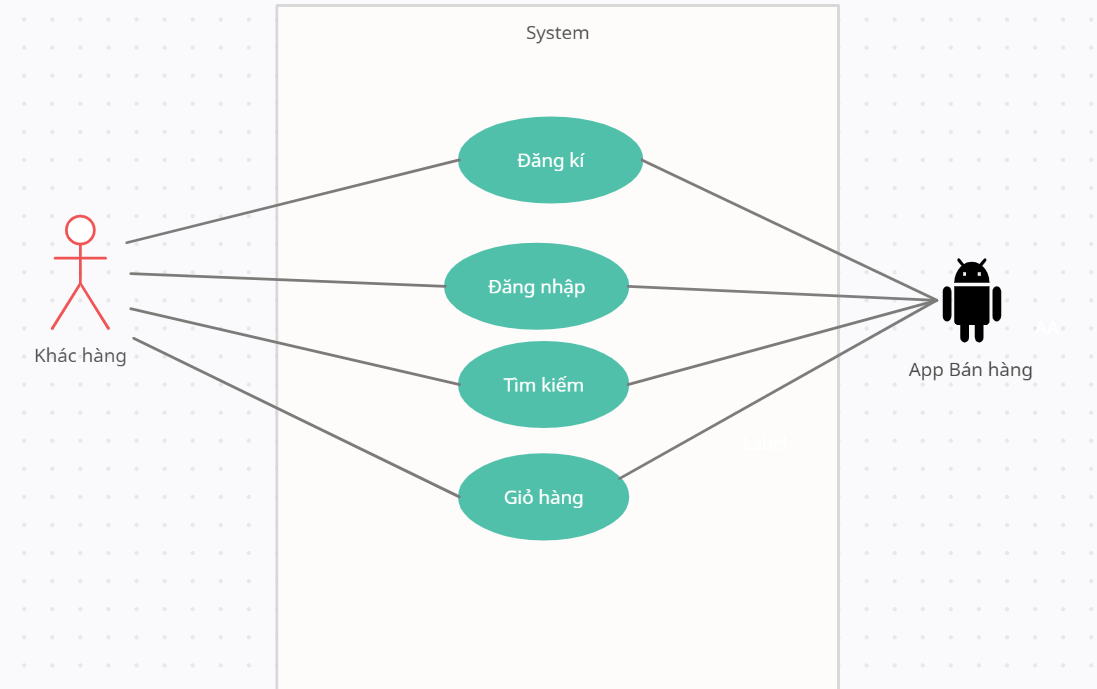
-use case đăng nhập, đăng kí

-use case tìm kiếm

-use case giỏ hàng (thêm,xoá,sủa)

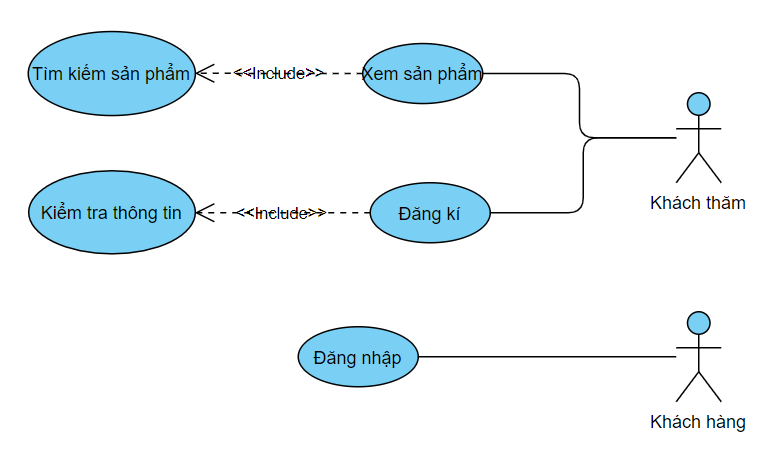
-use case xem sản phẩm

* **UC toàn hệ thống**

****

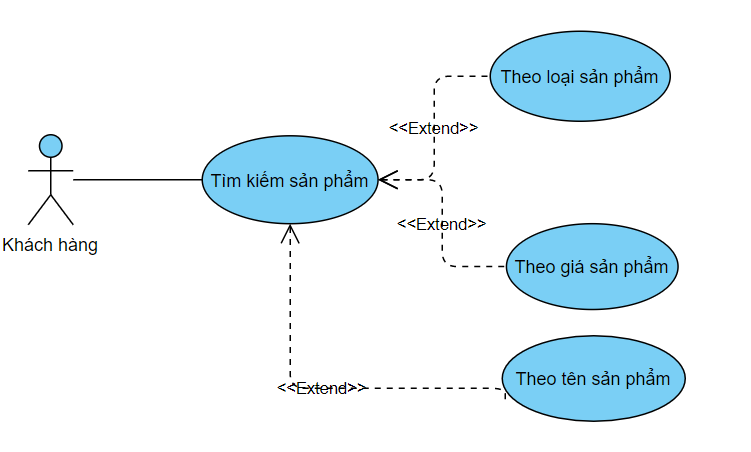
**Hình 2.1. Use case toàn hệ thống**

* **UC đăng nhập,đăng kí**



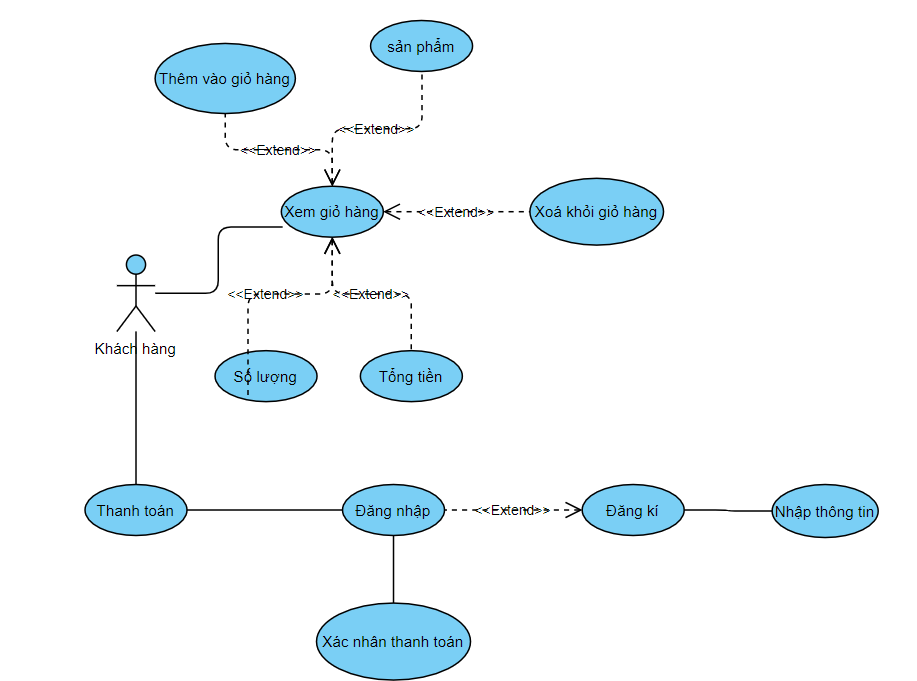
**Hình 2.3. Use case đăng nhập, đăng kí**

* **UC tìm kiếm**



**Hình 2.4 Use case tìm kiếm sản phẩm**

* **UC giỏ hàng**

****

**Hình 2.5 Use case giỏ hàng**

1. **Đặc tả use case**

**3.1 Đăng kí tài khoản**

a.Tác nhân :khách hàng

b.Mô tả: cho phép khách hàng đăng kí tài khoản cho hệ thống

c.Tiền điều kiện: khách hàng sử dụng ứng dụng

d.Luồng sự kiện chính:

-Khách hàng chọn chức năng đăng kí tài khoản

-Màn hình đăng kí hiển thị

-Khách hàng điền thông tin đăng nhập

-Nhấn nút đăng kí

e.Luồng rẽ nhánh

f.Hậu điều kiện:khách hàng đăng kí thành công tài khoản

**3.2 .Đăng nhập tài khoản**

a.Tác nhân:khách hàng

b.Mô tả: UC cho phép khách hàng đăng nhập vào hệ thống

c.Tiền điều kiện: Khách hàng chưa đăng nhập vào hệ thống

d.Luồng sự kiện chính:

-Khách hàng chọn chức năng đăng nhập tài khoản

-Màn hình đăng nhập hiển thị

-Khách hàng điền Email và passworld

-Kiểm tra tài khoản khách hàng nếu đúng sẽ chuyển sang màn hình chính nếu sai nhập lại mật khẩu

-UC kết thúc

e.Luồng rẻ nhánh:

f.Hậu điều kiện:đăng nhập thành công,khách hàng có thể sử dụng các chức năng của hệ thống

**3.3 Tìm kiếm**

a.Tác nhân:khách hàng

b.Mô tả: UC cho phép khách hàng tìm kiếm sản phẩm

c.Tiền điều kiện: Khách hàng đã đăng nhập vào hệ thống

d.Luồng sự kiện chính:

-Khách hàng chọn biểu tượng tìm kiếm trên màn hình chính

-Màn hình hiện thị giao diện tìm kiếm

-Khách hàng nhập từ khoá cần tìm

-Nhấn nút tìm kiếm

-Hiện thị kết quả tìm kiếm

e.Luồng rẻ nhánh:

f.Hậu điều kiện:

**3.4 Xem thông tin chi tiết sản phẩm**

a.Tác nhân:khách hàng

b.Mô tả: UC cho phép khách hàng xem thông tin sản phẩm

c.Tiền điều kiện: Khách hàng đã đăng nhập vào hệ thống

d.Luồng sự kiện chính:

-Khách hàng chọn sản phẩm muốn xem

-Màn hình hiện thị trang chi tiết sản phẩm

e.Luồng rẻ nhánh:

f.Hậu điều kiện:Hiện thi chi tiết các sản phẩm

**3.5 Thêm vào giỏ hàng**

a.Tác nhân:khách hàng

b.Mô tả: UC cho phép khách hàng đưa các sản phẩm đã chọn vào giỏ hàng

c.Tiền điều kiện: sản phẩm đã được chọn

d.Luồng sự kiện chính:

-Khách hàng chọn chức năng thêm sản phẩm vào giỏ hàng và lưu trữ,nếu chưa chọn sản phẩm mà chọn nút giỏ hàng thì thực hiện luồng rẽ nhánh A1

-Màn hình hiện thị thông tin sản phẩm đưa vào giỏ hàng

-Thanh toán giỏ hàng

-UC kết thúc

e.Luồng rẻ nhánh A1: chưa chọn sản phẩm cần mua

-Hệ thống báo sản phẩm chưa được chọn

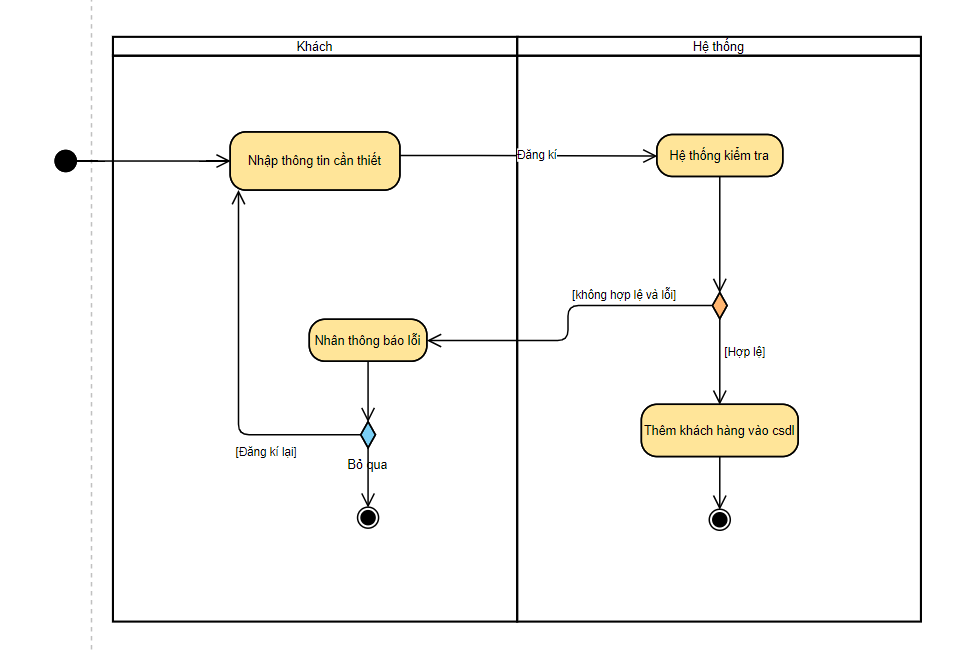
-Hiển thị sản phẩm để khách hàng chọn

-Quay lại bước 1 của luồng sự kiện chính

f.Hậu điều kiện:Thanh toán thành công giỏ hàng

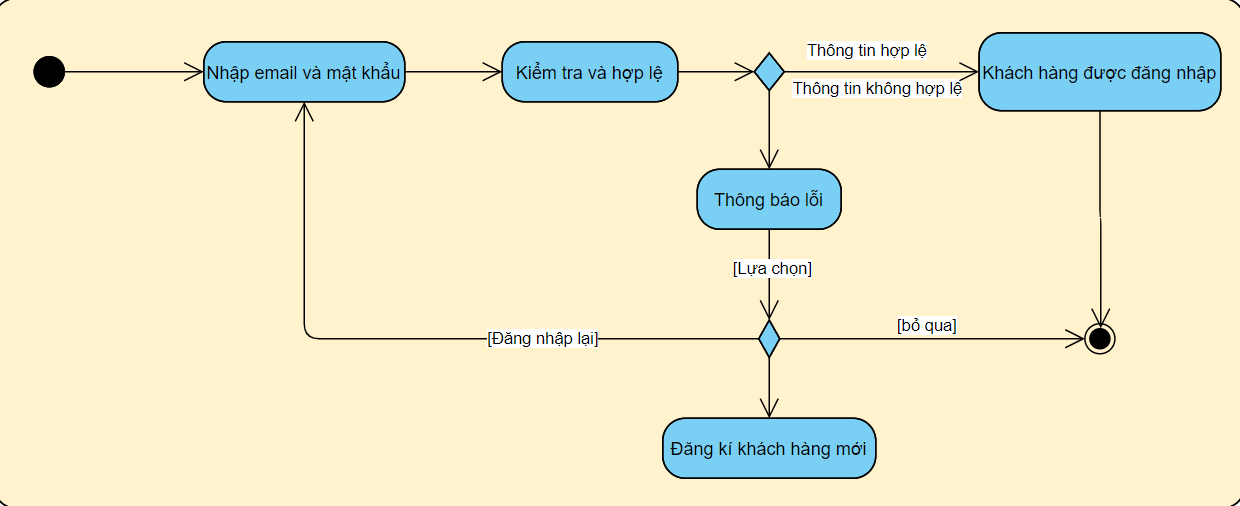
**4. Biểu đồ hoạt động**

**4.1 Đăng kí tài khoản**

****

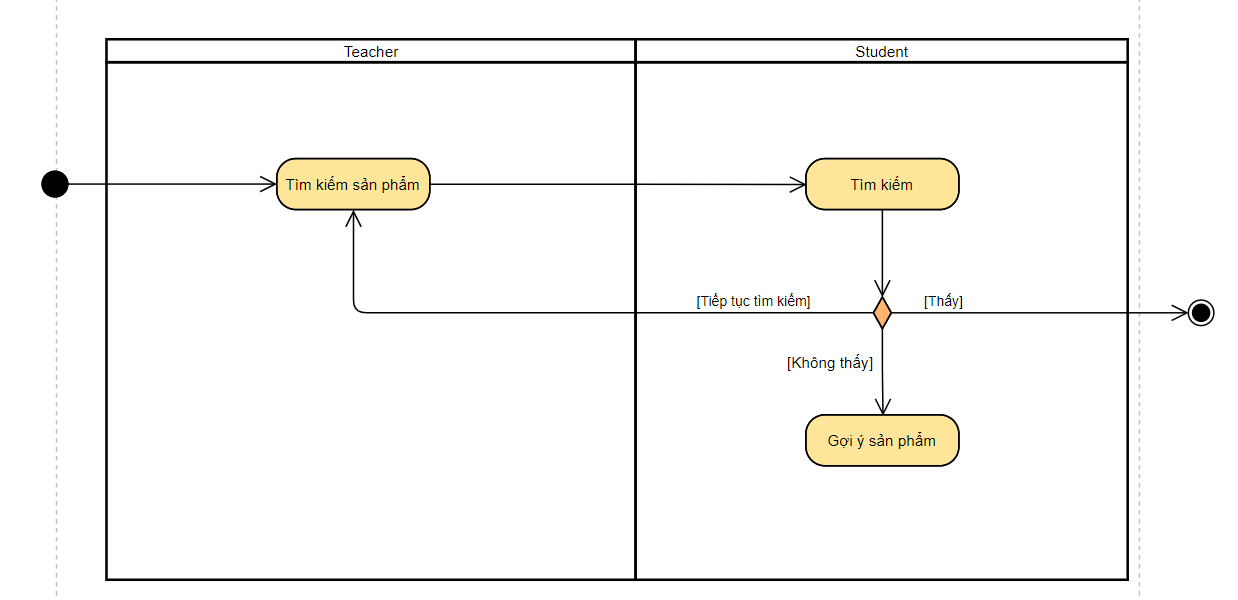
**Hình 4.1 Biểu đồ hoạt động đăng kí tài khoản**

**4.2 Đăng nhập vào hệ thống**

****

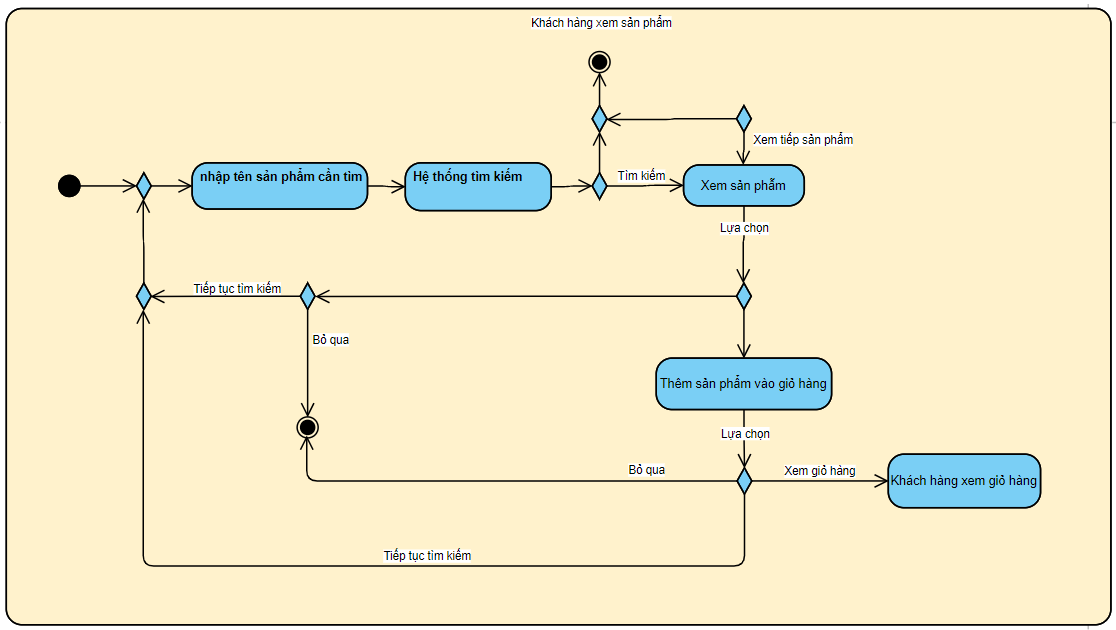
**Hình 4.2 biểu đồ hoạt động đăng nhập**

**4.3 Tìm kiếm sản phẩm**

****

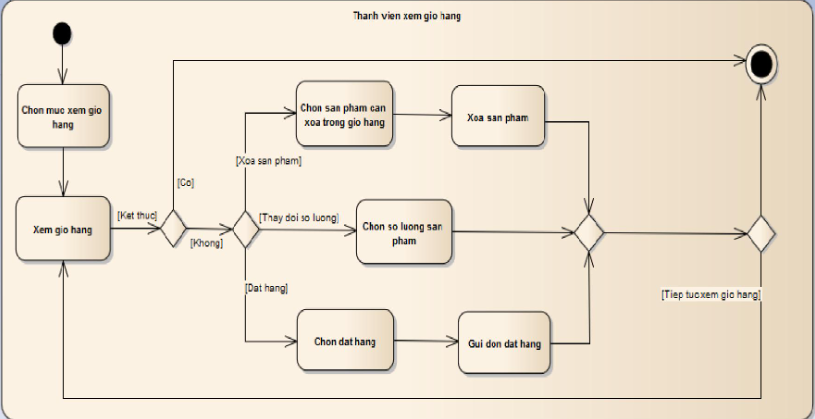
**Hình 4.3 biểu đồ hoạt động tìm kiếm**

**4.4 Xem sản phẩm**



**Hình 4.4 biểu đồ hoạt động xem sản phẩm**

**4.5 Xem giỏ hàng**

****

**Hình 4.5 biểu đồ hoạt động xem giỏ hàng**

**CHƯƠNG III:THIẾT KẾ CƠ SỞ DỮ LIỆU**

**1.Xác định các thực thể**

-Users(Email,Passworld,UsersName)

-Products(IdProducts,NameProducts,Image,Price, description)

**2. Mô tả chi tiết các thực thể**

**2.1 Users: Tài khoản người dùng**

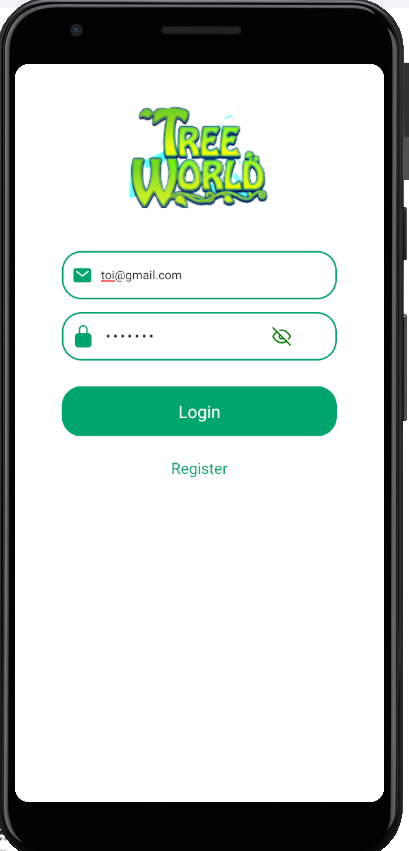
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu | Null | Ràng buộc | Mô tả |
| Email | String | X |  | Email khách hàng |
| Password | String |  |  | Mật khẩu khách hàng |
| UsersName | String |  |  | Tên khách hàng |

**2.2 Products: Thông tin sản phẩm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu | Null | Ràng buộc | Mô tả |
| idProducts | String |  |  | ID sản phẩm |
| NameProducts | String |  |  | Tên sản phẩm |
| Image | String |  |  | Hình ảnh sản phẩm |
| Price | Double |  |  | Giá sản phẩm |
| description | String |  |  | Mô tả sản phẩm |

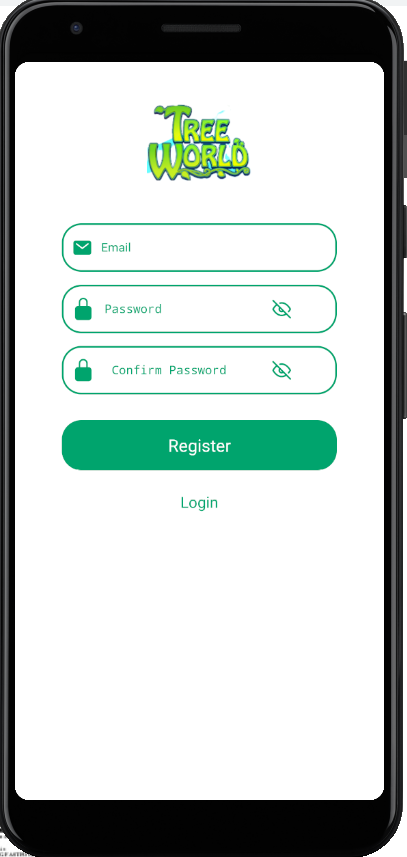
**CHƯƠNG IV: CÀI ĐẶT CHƯƠNG TRÌNH**

1. **Giao diện chức năng đăng nhập**

****

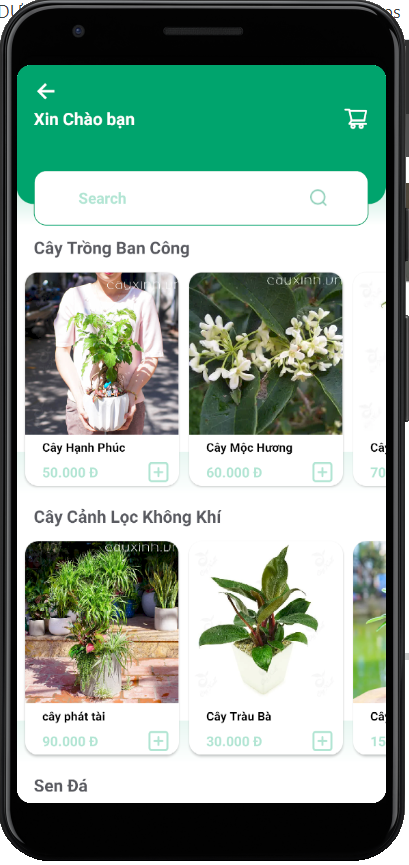
**Hình 1.1 Giao diện đăng nhập**

1. **Giao diện chức năng đăng kí**

****

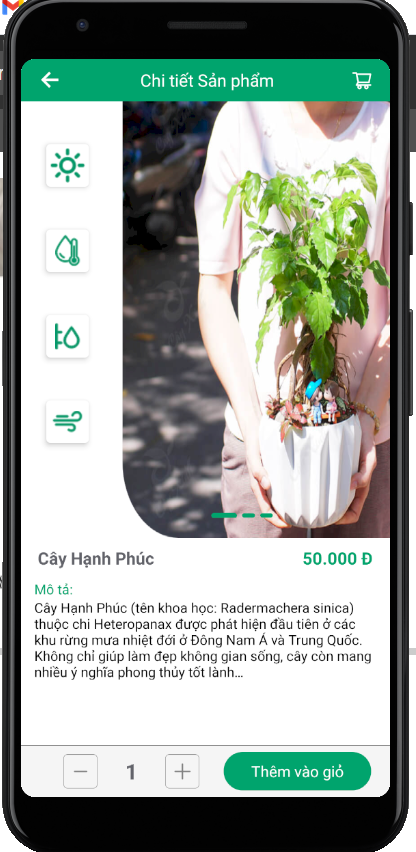
**Hình 1.2 Giao diện đăng kí**

1. **Giao diện trang chủ**

****

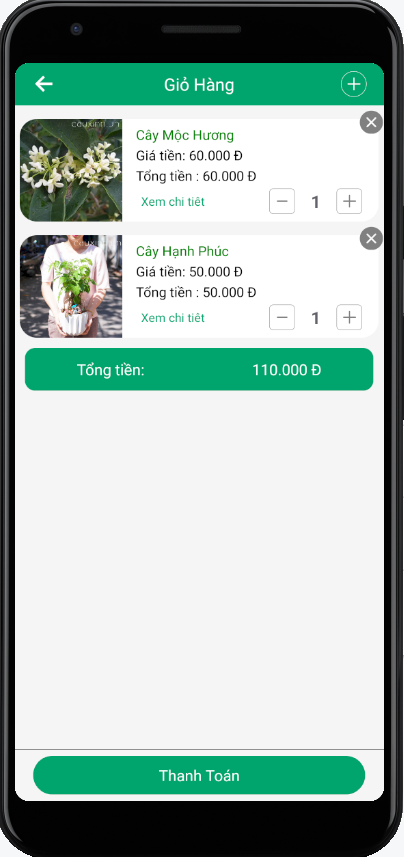
**Hình 1.3 Giao diện trang chủ**

1. **Giao diện chức năng xem chi tiết sản phẩm**

****

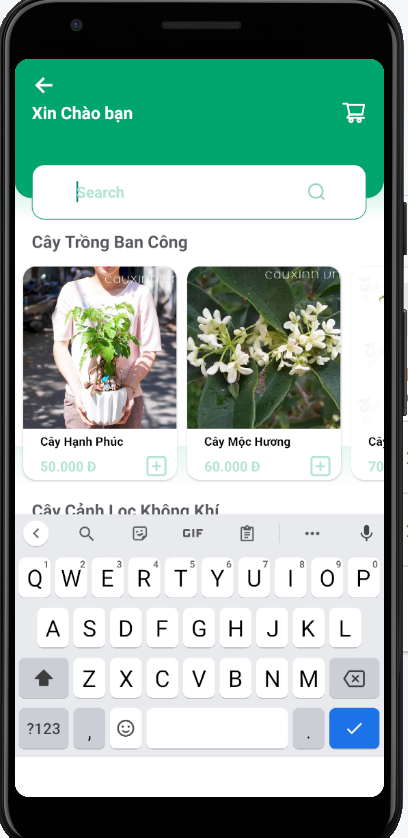
**Hình 1.4 Giao diện chi tiết sản phẩm**

1. **Giao diện chức năng giỏ hàng**

****

**Hình 1.5 Giao diện giỏ hàng**

1. **Giao diện chức năng tìm kiếm**

****

**Hình 1.6 Giao diện đăng kí**

**2. Quá trình thực hiện các chức năng và giao diện**

**2.1 DataProducts**

import image from './Image';

export default restaurantData = [

  {

    id: 1,

    name: 'Cây Trồng Ban Công ',

    menu: [

      {

        menuId: 11,

        nameProducts: 'Cây Hạnh Phúc',

        photo: [

          {

            imageId: 12,

            path: image.cayhanhphuc1,

          },

          {

            imageId: 22,

            path: image.cayhanhphuc2,

          },

          {

            imageId: 32,

            path: image.cayhanhphuc3,

          },

        ],

        avatarImage: image.cayhanhphuc1,

        description:

          'Cây Hạnh Phúc (tên khoa học: Radermachera sinica) thuộc chi Heteropanax được phát hiện đầu tiên ở các khu rừng mưa nhiệt đới ở Đông Nam Á và Trung Quốc. Không chỉ giúp làm đẹp không gian sống, cây còn mang nhiều ý nghĩa phong thủy tốt lành…',

        price: 50000,

      },

      {

        menuId: 21,

        nameProducts: 'Cây Mộc Hương',

        photo: [

          {

            imageId: 22,

            path: image.caymochuong,

          },

        ],

        avatarImage: image.caymochuong,

        description:

          'Cây Mộc Hương sở hữu những bông hoa với mùi hương ngọt ngào, là một trong những loài cây được đặc biệt nhiều người Việt Nam yêu thích. Đặc biệt, đây chính là sự lựa chọn không thể thiếu trong sân nhà.',

        price: 60000,

      },

      {

        menuId: 31,

        nameProducts: 'Cây Thiết Mộc Lan',

        photo: [

          {

            imageId: 31,

            path: image.thietmoclan,

          },

          {

            imageId: 32,

            path: image.thietmoclan1,

          },

        ],

        avatarImage: image.thietmoclan,

        description:

          'Thiết Mộc Lan (tên khoa học: Mass Cane  hay Dracaena Fragrans) là loại cây trồng trong nhà dễ chăm sóc, có ý nghĩa mang lại tài lộc, vận may đến với gia chủ…',

        price: 70000,

      },

      {

        menuId: 4,

        nameProducts: 'Hoa Dừa Cạn',

        photo: [

          {

            imageId: 41,

            path: image.hoaduacan,

          },

        ],

        avatarImage: image.hoaduacan,

        description:

          'Hoa Dừa Cạn là một trong số những sự lựa chọn cây trồng ban công vô cùng phổ biến tại Việt Nam. Một chậu hoa Dừa Cạn sẽ mang đến cho không gian sống thêm màu sắc và giàu sức sống…',

        price: 80000,

      },

      {

        menuId: 51,

        nameProducts: 'Cây Hồng Leo',

        photo: [

          {

            imageId: 51,

            path: image.hoahongleo,

          },

        ],

        avatarImage: image.hoahongleo,

        description:

          'Hoa hồng leo là một trong những loại cây cảnh trồng ban công đang rất được ưa chuộng. Với vẻ ngoài bắt mắt, kiêu sa mang đến một không gian xanh mát, tự nhiên…',

        price: 45000,

      },

      {

        menuId: 61,

        nameProducts: 'Cây Hoa giấy',

        photo: [

          {

            imageId: 61,

            path: image.hoagiay,

          },

          {

            imageId: 62,

            path: image.hoagiay1,

          },

          {

            imageId: 63,

            path: image.hoagiay2,

          },

        ],

        avatarImage: image.hoagiay,

        description:

          'Hoa Giấy (tên khoa học: Bougainvillea spectabilis) được biết đến là một trong những loại cây trồng ban công, cổng nhà vô cùng quen thuộc. Cây có vẻ ngoài nhẹ nhàng, mộc mạc. Không chỉ có tác dụng trang trí, cây Hoa Giấy còn mang đến cho gia đình những ý nghĩa phong thủy tốt đẹp.',

        price: 50000,

      },

    ],

  },

  {

    id: 2,

    name: 'Cây Cảnh Lọc Không Khí',

    menu: [

      {

        menuId: 111,

        nameProducts: 'cây phát tài',

        photo: [

          {

            imageId: 111,

            path: image.phatai,

          },

          {

            imageId: 112,

            path: image.phatai1,

          },

          {

            imageId: 113,

            path: image.phatai2,

          },

        ],

        avatarImage: image.phatai,

        description:

          'Cây Phát Tài được tìm thấy ở Zambia, Tanzania và Tây Phi. Loại cây này có lá màu xanh sẫm, tán lá xòe rộng và có đường gân vàng nổi bật, thường được trồng trong nhà giúp thanh lọc không khí hiệu quả…',

        price: 90000,

      },

      {

        menuId: 112,

        nameProducts: 'Cây Tràu Bà',

        photo: [

          {

            imageId: 221,

            path: image.trauba,

          },

          {

            imageId: 222,

            path: image.trauba1,

          },

        ],

        avatarImage: image.trauba,

        description:

          'Cây Trầu Bà (tên khoa học: Philodendron Imperial) là một loài thực vật có hoa họ Ráy (Araceae), tượng trưng cho sự may mắn, thành đạt và bình an cho gia chủ…',

        price: 30000,

      },

      {

        menuId: 113,

        nameProducts: 'Cây Trường Sinh',

        photo: [

          {

            imageId: 331,

            path: image.truongsinh,

          },

          {

            imageId: 332,

            path: image.truongsinh1,

          },

          {

            imageId: 333,

            path: image.truongsinh,

          },

        ],

        avatarImage: image.truongsinh,

        description:

          'Cây Trường Sinh (tên khoa học: Peperomia Obtusifolia) là loại thực vật xanh tốt quanh năm, có khả năng sinh tồn mạnh mẽ, sức chống chịu cao. Trong phong thủy cây tượng trưng cho sức khỏe, sự trường tồn, ý chí mạnh mẽ và quyết liệt,.…',

        price: 150000,

      },

    ],

  },

  {

    id: 3,

    name: 'Sen Đá',

    menu: [

      {

        menuId: 1111,

        nameProducts: 'Sen Đá Da Rắn',

        photo: [

          {

            imageId: 4441,

            path: image.sendadaran,

          },

          {

            imageId: 4442,

            path: image.sendadaran1,

          },

          {

            imageId: 4443,

            path: image.sendadaran2,

          },

        ],

        avatarImage: image.sendadaran,

        description:

          'Sen Đá Da Rắn (tên khoa học: Senecio Scaposus Silver Coral) có những đặc điểm hình dáng rất đặc biệt, độc đáo. Cây cũng có tác dụng thanh lọc không khí, bảo vệ sức khỏe con người hiệu quả…',

        price: 100000,

      },

      {

        menuId: 1112,

        nameProducts: 'Sen Đá kim Cương',

        photo: [

          {

            imageId: 5551,

            path: image.sendakimcuong,

          },

          {

            imageId: 5552,

            path: image.sendakimcuong1,

          },

          {

            imageId: 5553,

            path: image.sendakimcuong2,

          },

        ],

        avatarImage: image.sendakimcuong,

        description:

          'Sen Đá Kim Cương (tên khoa học: Pachyphytum compactum) nổi bật bởi có màu xanh nhẹ nhàng và hình dáng nhỏ nhắn. Cây mang ý nghĩa của tình yêu vĩnh cửu, vẹn nguyên và dài lâu…',

        price: 75000,

      },

      {

        menuId: 1113,

        nameProducts: 'Sen Đá Pha Lê',

        photo: [

          {

            imageId: 6661,

            path: image.sendaphale,

          },

          {

            imageId: 6662,

            path: image.sendaphale1,

          },

          {

            imageId: 6663,

            path: image.sendaphale2,

          },

        ],

        avatarImage: image.sendaphale,

        description:

          'Sen Đá Pha Lê (tên khoa học: Graptosedum Francesco Baldi) có nguồn gốc từ Mexico và Arizona. Cây mang vẻ đẹp nhẹ nhàng tinh tế và rất bắt mắt được trồng nhiều tại ban công, cửa sổ…',

        price: 55000,

      },

      {

        menuId: 1114,

        nameProducts: 'Sen Đá Giva',

        photo: [

          {

            imageId: 7771,

            path: image.sendagiva,

          },

          {

            imageId: 7772,

            path: image.sendagiva,

          },

          {

            imageId: 7773,

            path: image.sendagiva2,

          },

        ],

        avatarImage: image.sendagiva,

        description:

          'Sen Đá Giva (tên khoa học: Echeveria Agavoides) có màu xanh tươi đặc trưng, đầu lá như những chiếc gai nhọn mọc xen kẽ nhau một cách cân đối. Lá cây mọc đều xung quanh gốc, xếp xòe ra rất đẹp…',

        price: 35000,

      },

    ],

  },

];

**2.2 index Home**

import React, {useState, useRef, useEffect} from 'react';

import {

  View,

  Text,

  Image,

  ImageBackground,

  SafeAreaView,

  FlatList,

  Keyboard,

  TouchableOpacity,

  ScrollView,

  TextInput,

  Dimensions,

  ActivityIndicator,

} from 'react-native';

// import {TextInput, ScrollView} from 'react-native-gesture-handler';

import LinearGradient from 'react-native-linear-gradient';

import Icon from 'react-native-vector-icons/Feather';

import Icon1 from 'react-native-vector-icons/AntDesign';

import Icon2 from 'react-native-vector-icons/Ionicons';

import DataProduct from '../../Data/DataProduct';

import auth from '@react-native-firebase/auth';

import styles from './styles';

import {convertToNumberCommas} from '../../utilities/index';

const {height, width} = Dimensions.get('window');

const DataProducts = DataProduct;

const Home = ({navigation}) => {

  const [dataSearch, setDataSearch] = useState([]);

  const [term, setTerm] = useState('');

  let timeout = useRef(null);

  const [textinput, setTextinput] = React.useState('');

  const [itemAll, setItemAll] = useState([]);

  const [isLoading, setIsLoading] = useState(true);

  function onSearch(text) {

    setIsLoading(true);

    let term = text;

    let AllProducts = itemAll;

    if (term && term.length >= 1 && AllProducts.length != 0) {

      if (timeout) clearTimeout(timeout);

      timeout = setTimeout(() => {

        var menu = AllProducts.filter(menuname =>

          menuname.nameProducts.toLowerCase().includes(term.toLowerCase()),

        );

        setDataSearch(menu);

        setIsLoading(false);

      }, 3000);

    }

  }

  useEffect(() => {

    let AllProducts = [];

    const Data = DataProducts;

    Data.map((item, index) => {

      item.menu.map((itemProduct, index) => {

        AllProducts.push(itemProduct);

      });

    });

    // Data.push({id: 1234456, name: 'Tất Cả', menu: AllProducts});

    // console.log(Data);

    setItemAll(AllProducts);

  }, []);

  const renderItemProduct = ({item, index}) => {

    return (

      <View>

        <View style={styles.ViewTitileProducts}>

          <View>

            <Text style={styles.TxtTitleProducts}>{item.name}</Text>

          </View>

          {/\* <View style={{flex: 1, alignItems: 'flex-end'}}>

    <View style={styles.ViewTitileMore}>

      <Text style={styles.TxtTitileMore}>More</Text>

    </View>

  </View> \*/}

        </View>

        {/\* view list of products \*/}

        <ScrollView

          horizontal

          showsHorizontalScrollIndicator={false}

          style={{height: 270}}>

          <LinearGradient

            colors={['rgba(0,164,109,0.09)', 'transparent', '#FFF']}

            style={styles.ViewLinearGradient}

          />

          {item.menu.map((itemProduct, index) => {

            return (

              <>

                <TouchableOpacity

                  activeOpacity={1}

                  key={index}

                  onPress={() =>

                    navigation.navigate('Detail', {dataItem: itemProduct})

                  }

                  style={styles.ViewProducts}>

                  <Image

                    source={itemProduct.avatarImage}

                    style={styles.ViewImageProducts}

                  />

                  <View style={styles.ViewTitilePro}>

                    <Text

                      style={{

                        fontWeight: 'bold',

                        paddingHorizontal: 10,

                      }}>

                      {itemProduct.nameProducts}

                    </Text>

                  </View>

                  {/\* <Text

      style={{

        paddingHorizontal: 10,

        fontWeight: 'bold',

        color: '#b1e5d3',

        paddingTop: 3,

      }}>

      RUSSIA

    </Text> \*/}

                  <View

                    style={[

                      styles.ViewTitilePro,

                      {justifyContent: 'space-between', alignItems: 'center'},

                    ]}>

                    <Text

                      style={{

                        paddingHorizontal: 10,

                        fontWeight: 'bold',

                        color: '#b1e5d3',

                        fontSize: 16,

                        // paddingBottom: 3,

                      }}>

                      {convertToNumberCommas(itemProduct.price)} Đ

                    </Text>

                    <Icon

                      name="plus-square"

                      size={28}

                      color="#b1e5d3"

                      // style={{paddingBottom: 3}}

                    />

                  </View>

                </TouchableOpacity>

              </>

            );

          })}

        </ScrollView>

      </View>

    );

  };

  const renderItemDataSearch = ({item, index}) => {

    return (

      <TouchableOpacity

        activeOpacity={1}

        key={index}

        onPress={() => navigation.navigate('Detail', {dataItem: item})}

        style={styles.ViewProducts}>

        <Image source={item.avatarImage} style={styles.ViewImageProducts} />

        <View style={styles.ViewTitilePro}>

          <Text

            style={{

              fontWeight: 'bold',

              paddingHorizontal: 10,

            }}>

            {item.nameProducts}

          </Text>

        </View>

        <View

          style={[

            styles.ViewTitilePro,

            {justifyContent: 'space-between', alignItems: 'center'},

          ]}>

          <Text

            style={{

              paddingHorizontal: 10,

              fontWeight: 'bold',

              color: '#b1e5d3',

              fontSize: 16,

              // paddingBottom: 3,

            }}>

            {item.price}

          </Text>

          <Icon

            name="plus-square"

            size={28}

            color="#b1e5d3"

            // style={{paddingBottom: 3}}

          />

        </View>

      </TouchableOpacity>

    );

  };

  return (

    <SafeAreaView style={styles.container}>

      <View>

        {/\* ViewHeader \*/}

        <View style={styles.ViewHeader}>

          <TouchableOpacity

            style={{marginTop: 15}}

            onPress={() => navigation.goBack()}>

            {/\* <Image

            source={require('../../images/17.png')}

            style={{marginVertical: 40, width: 25}}

          /> \*/}

            <Icon2 name="arrow-back" size={30} color="white" />

          </TouchableOpacity>

          <View style={styles.ViewTitile}>

            <View style={{width: '50%'}}>

              <Text style={styles.TxtTitle}>Xin Chào bạn</Text>

            </View>

            <View style={{width: '50%', alignItems: 'flex-end'}}>

              <TouchableOpacity

                onPress={() => navigation.navigate('ShoppingCart')}>

                <Icon1 name="shoppingcart" size={30} color="white" />

              </TouchableOpacity>

              {/\* <Image

                source={require('../../images/g.png')}

                style={{height: 60, width: 60}}

              /> \*/}

            </View>

          </View>

        </View>

        {/\* View Tìm kiếm  \*/}

        <LinearGradient

          colors={['rgba(0,164,109,0.4)', 'transparent']}

          style={styles.ViewSearchContainer}>

          <View style={styles.ViewSearch}>

            <TextInput

              placeholder="Search"

              placeholderTextColor="#b1e5d3"

              style={styles.TxtTextInput}

              onChangeText={value => setTextinput(value)}

              onEndEditing={() => {

                Keyboard.dismiss();

                onSearch(textinput);

              }}

            />

            <TouchableOpacity

              hitSlop={{top: 10, left: 10, right: 10, bottom: 10}}

              onPress={() => {

                Keyboard.dismiss();

                onSearch(textinput);

              }}>

              <Image

                source={require('../../images/3.png')}

                style={{height: 20, width: 20}}

              />

            </TouchableOpacity>

          </View>

        </LinearGradient>

        {/\* view the title of the products all \*/}

        {textinput.length <= 1 ? (

          <View style={{marginBottom: 360}}>

            <FlatList

              data={DataProducts}

              renderItem={renderItemProduct}

              keyExtractor={(item, index) => index.toString()}

            />

          </View>

        ) : (

          <>

            <Text style={{marginHorizontal: 16, marginVertical: 6}}>

              Kết quả tìm kiếm{' '}

            </Text>

            {isLoading ? (

              <ActivityIndicator size="large" color="#00a46c" />

            ) : (

              <View>

                {dataSearch.length != 0 ? (

                  <FlatList

                    horizontal

                    data={dataSearch}

                    renderItem={renderItemDataSearch}

                    keyExtractor={(item, index) => index.toString()}

                  />

                ) : (

                  <Text>Chưa có kết quả nào</Text>

                )}

              </View>

            )}

          </>

        )}

      </View>

    </SafeAreaView>

  );

};

export default Home;

**2.3 index chi tiết sản phẩm**

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

import {

  View,

  Text,

  Image,

  SafeAreaView,

  ScrollView,

  Dimensions,

} from 'react-native';

import SwiperComponent from '../../components/SwiperComponent';

import {TouchableOpacity} from 'react-native-gesture-handler';

import styles from './styles';

import Icon from 'react-native-vector-icons/AntDesign';

import Icon1 from 'react-native-vector-icons/Ionicons';

import auth from '@react-native-firebase/auth';

import firestore from '@react-native-firebase/firestore';

import {convertToNumberCommas} from '../../utilities/index';

const {width, height} = Dimensions.get('window');

const Detail = ({navigation, route}) => {

  const [numberProduct, setNumberProduct] = useState(

    route?.params?.count ? route?.params?.count : 1,

  );

  const [dataDetail, setDataDetail] = useState(route?.params?.dataItem);

  useEffect(() => {

    console.log(dataDetail);

  }, []);

  const onhandleShoppingCart = async () => {

    const user = auth().currentUser;

    console.log(user);

    let data = dataDetail;

    let body = {

      amount: numberProduct,

      idProduct: data.menuId,

      nameProduct: data.nameProducts,

      priceProduct: data.price,

      email: user?.email,

    };

    await firestore().collection('ShoppingCart').add(body);

    navigation.navigate('ShoppingCart');

  };

  return (

    <SafeAreaView style={styles.container}>

      {/\* Header \*/}

      <View

        style={{

          height: 50,

          // width: width,

          flexDirection: 'row',

          justifyContent: 'space-between',

          paddingHorizontal: 20,

          backgroundColor: '#00a46c',

          alignItems: 'center',

        }}>

        <TouchableOpacity onPress={() => navigation.goBack()}>

          {/\* <Image

            source={require('../../images/17.png')}

            style={{marginVertical: 40, width: 25}}

          /> \*/}

          <Icon1 name="arrow-back" size={30} color="white" />

        </TouchableOpacity>

        <Text style={{fontSize: 20, color: 'white'}}>Chi tiết Sản phẩm</Text>

        <TouchableOpacity onPress={() => navigation.navigate('ShoppingCart')}>

          <Icon name="shoppingcart" size={25} color="white" />

        </TouchableOpacity>

      </View>

      <ScrollView

        showsHorizontalScrollIndicator={false}

        showsVerticalScrollIndicator={false}>

        <View

          style={{

            flexDirection: 'row',

            width: '100%',

            height: height \* 0.7 - 50,

          }}>

          <View style={{width: width \* 0.2, paddingLeft: 30}}>

            <View style={styles.ViewCharacteristics}>

              <Image source={require('../../images/11.png')} />

            </View>

            <View style={styles.ViewCharacteristics}>

              <Image source={require('../../images/12.png')} />

            </View>

            <View style={styles.ViewCharacteristics}>

              <Image source={require('../../images/13.png')} />

            </View>

            <View style={styles.ViewCharacteristics}>

              <Image source={require('../../images/14.png')} />

            </View>

          </View>

          <View style={{width: width \* 0.8}}>

            <SwiperComponent photo={dataDetail.photo} />

          </View>

        </View>

        <View style={styles.ViewNameProduct}>

          <Text numberOfLines={1} style={styles.TxtName}>

            {dataDetail.nameProducts}

          </Text>

          <Text style={styles.TxtPrice}>

            {convertToNumberCommas(dataDetail.price)} Đ

          </Text>

        </View>

        <View style={{marginVertical: 12}}>

          <Text style={styles.TxtIntroduce}>Mô tả:</Text>

          <Text style={[styles.TxtIntroduce, {color: 'black'}]}>

            {dataDetail.description}

          </Text>

        </View>

      </ScrollView>

      <View style={styles.ViewBottomContainer}>

        <View style={styles.ViewNumProduct}>

          <TouchableOpacity

            onPress={() => {

              if (numberProduct > 0) {

                setNumberProduct(numberProduct - 1);

              }

            }}

            style={styles.ButtomPlus}>

            <Icon name="minus" size={25} color="gray" />

          </TouchableOpacity>

          <Text

            style={{

              color: '#62636a',

              fontWeight: 'bold',

              fontSize: 25,

            }}>

            {numberProduct}

          </Text>

          <TouchableOpacity

            onPress={() => {

              setNumberProduct(numberProduct + 1);

            }}

            style={styles.ButtomPlus}>

            <Icon name="plus" size={25} color="gray" />

          </TouchableOpacity>

        </View>

        <TouchableOpacity

          onPress={() => onhandleShoppingCart()}

          style={styles.ViewButtom}>

          <Text

            style={{

              color: '#FFF',

              fontSize: 18,

            }}>

            Thêm vào giỏ

          </Text>

        </TouchableOpacity>

      </View>

    </SafeAreaView>

  );

};

export default Detail;

**2.4 Index Login**

import React, {useState} from 'react';

import {

  Text,

  View,

  Image,

  TextInput,

  TouchableOpacity,

  SafeAreaView,

  Alert,

  Dimensions,

  // AsyncStorage,

} from 'react-native';

import Icon from 'react-native-vector-icons/Ionicons';

import Feather from 'react-native-vector-icons/Feather';

import styles from './styles';

import \* as Animatable from 'react-native-animatable';

// import firestore from '@react-native-firebase/firestore';

import auth from '@react-native-firebase/auth';

// import {AuthContext} from './context';

import AsyncStorage from '@react-native-async-storage/async-storage';

const {height, width} = Dimensions.get('window');

const Login = ({navigation}) => {

  const [data, setData] = React.useState({

    username: 'toi@gmail.com',

    password: '1234567',

    check\_textInputChange: false,

    secureTextEntry: true,

    isValidUser: true,

    isValidPassword: true,

  });

  const isEmail = value => {

    // eslint-disable-next-line no-useless-escape

    const regex = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)\*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;

    return !!regex.test(String(value).toLowerCase());

  };

  const textInputChange = val => {

    if (val.trim().length >= 4) {

      setData({

        ...data,

        username: val,

        check\_textInputChange: isEmail(val),

        isValidUser: isEmail(val),

      });

    } else {

      setData({

        ...data,

        username: val,

        check\_textInputChange: isEmail(val),

        isValidUser: isEmail(val),

      });

    }

  };

  const handlePasswordChange = val => {

    if (val.trim().length >= 6) {

      setData({

        ...data,

        password: val,

        isValidPassword: true,

      });

    } else {

      setData({

        ...data,

        password: val,

        isValidPassword: false,

      });

    }

  };

  const updateSecureTextEntry = () => {

    setData({

      ...data,

      secureTextEntry: !data.secureTextEntry,

    });

  };

  const handleValidUser = val => {

    if (val.trim().length >= 4) {

      setData({

        ...data,

        isValidUser: isEmail(val),

      });

    } else {

      setData({

        ...data,

        isValidUser: false,

      });

    }

  };

  const loginHandle = async () => {

    // navigation.navigate('SignUpScreen')

    const {username, password} = data;

    if (data.username.length === 0 || data.password.length === 0) {

      Alert.alert(

        'Wrong Input!',

        'Username or password field cannot be empty.',

        [{text: 'Okay'}],

      );

      return;

    } else {

      await auth()

        .signInWithEmailAndPassword(username, password)

        .then(async function (result) {

          // Alert.alert('Thông báo!', 'Đăng nhập thành công', [{text: 'Okay'}]);

          let body = {

            username: username,

            password: password,

          };

          await AsyncStorage.setItem('@Account', JSON.stringify(body));

          navigation.navigate('Home');

          // result.user.tenantId should be ‘TENANT\_PROJECT\_ID’.

        })

        .catch(function (error) {

          if (error.code === 'auth/invalid-email') {

            Alert.alert('Thất bại!', 'Email bạn nhập không hợp lệ!', [

              {text: 'Okay'},

            ]);

          } else {

            if (error.code === 'auth/wrong-password') {

              Alert.alert(

                'Thông báo!',

                'Thông tin email hoặc password không chính xác!',

                [{text: 'Okay'}],

              );

            } else {

              Alert.alert(

                'Thông báo!',

                'Đăng nhập thất bại! Vui lòng thử lại!',

                [{text: 'Okay'}],

              );

            }

          }

          console.log(error);

        });

    }

  };

  return (

    <SafeAreaView style={{backgroundColor: '#FFF', height: '100%', flex: 1}}>

      <Image

        source={require('../../images/nameLogin.png')}

        style={{

          width: 163,

          height: 119,

          alignSelf: 'center',

          marginTop: 50,

        }}

      />

      <View

        style={{

          flexDirection: 'row',

          alignItems: 'center',

          marginHorizontal: 55,

          // marginRight: 20,

          borderWidth: 2,

          marginTop: 50,

          paddingHorizontal: 10,

          borderColor: '#00a46c',

          borderRadius: 23,

          paddingVertical: 2,

        }}>

        <Icon name="mail" color="#00a46c" size={24} />

        <TextInput

          placeholder="Email"

          placeholderTextColor="#00a46c"

          style={{paddingHorizontal: 10, marginRight: 10, width: width / 2.2}}

          autoCapitalize="none"

          onChangeText={val => textInputChange(val)}

          onEndEditing={e => handleValidUser(e.nativeEvent.text)}

          defaultValue={data.username}

        />

        {data.check\_textInputChange ? (

          <Animatable.View animation="bounceIn">

            <Feather name="check-circle" color="green" size={20} />

          </Animatable.View>

        ) : null}

      </View>

      {data.isValidUser ? null : (

        <Animatable.View animation="fadeInLeft" duration={500}>

          <Text style={styles.errorMsg}>Vui lòng nhập email hợp lệ</Text>

        </Animatable.View>

      )}

      <View

        style={{

          flexDirection: 'row',

          alignItems: 'center',

          marginHorizontal: 55,

          borderWidth: 2,

          marginTop: 15,

          paddingHorizontal: 10,

          borderColor: '#00a46c',

          borderRadius: 23,

          paddingVertical: 2,

        }}>

        <Icon name="md-lock-closed" color="#00a46c" size={28} />

        <TextInput

          placeholder="Password"

          placeholderTextColor="#00a46c"

          style={{paddingHorizontal: 10, marginRight: 10, width: width / 2.2}}

          secureTextEntry={data.secureTextEntry ? true : false}

          autoCapitalize="none"

          onChangeText={val => handlePasswordChange(val)}

          defaultValue={data.password}

        />

        <TouchableOpacity

          onPress={updateSecureTextEntry}

          style={{justifyContent: 'flex-end'}}>

          {data.secureTextEntry ? (

            <Feather name="eye-off" color="green" size={22} />

          ) : (

            <Feather name="eye" color="green" size={22} />

          )}

        </TouchableOpacity>

      </View>

      {data.isValidPassword ? null : (

        <Animatable.View animation="fadeInLeft" duration={500}>

          <Text style={styles.errorMsg}>Mật khẩu phải có ít nhất 6 ký tự.</Text>

        </Animatable.View>

      )}

      <TouchableOpacity

        onPress={() => {

          loginHandle(data.username, data.password);

        }}

        style={{

          marginHorizontal: 55,

          alignItems: 'center',

          justifyContent: 'center',

          marginTop: 30,

          backgroundColor: '#00a46c',

          paddingVertical: 16,

          borderRadius: 23,

        }}>

        <Text

          style={{

            color: 'white',

            fontFamily: 'SemiBold',

            fontSize: 20,

          }}>

          Login

        </Text>

      </TouchableOpacity>

      <TouchableOpacity onPress={() => navigation.navigate('Register')}>

        <Text

          style={{

            alignSelf: 'center',

            color: '#00a46c',

            fontFamily: 'SemiBold',

            paddingVertical: 25,

            fontSize: 18,

          }}>

          Register

        </Text>

      </TouchableOpacity>

    </SafeAreaView>

  );

};

export default Login;

**2.5 Index Register**

import React, {useState} from 'react';

import {

  Text,

  View,

  Image,

  TextInput,

  TouchableOpacity,

  SafeAreaView,

  Alert,

  Dimensions,

} from 'react-native';

import Icon from 'react-native-vector-icons/Ionicons';

import \* as Animatable from 'react-native-animatable';

import FontAwesome from 'react-native-vector-icons/FontAwesome';

import Feather from 'react-native-vector-icons/Feather';

import Fontisto from 'react-native-vector-icons/Fontisto';

import auth from '@react-native-firebase/auth';

import firestore from '@react-native-firebase/firestore';

const {height, width} = Dimensions.get('window');

const Register = ({navigation}) => {

  const [data, setData] = React.useState({

    username: '',

    password: '',

    confirm\_password: '',

    check\_textInputChange: false,

    secureTextEntry: true,

    confirm\_secureTextEntry: true,

  });

  const textInputChange = val => {

    if (val.length !== 0) {

      setData({

        ...data,

        username: val,

        check\_textInputChange: isEmail(val),

      });

    } else {

      setData({

        ...data,

        username: val,

        check\_textInputChange: false,

      });

    }

  };

  const handlePasswordChange = val => {

    setData({

      ...data,

      password: val,

    });

  };

  const handleConfirmPasswordChange = val => {

    setData({

      ...data,

      confirm\_password: val,

    });

  };

  const updateSecureTextEntry = () => {

    setData({

      ...data,

      secureTextEntry: !data.secureTextEntry,

    });

  };

  const updateConfirmSecureTextEntry = () => {

    setData({

      ...data,

      confirm\_secureTextEntry: !data.confirm\_secureTextEntry,

    });

  };

  const isEmail = value => {

    // eslint-disable-next-line no-useless-escape

    const regex = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)\*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;

    return !!regex.test(String(value).toLowerCase());

  };

  const InserUserToTable = async () => {

    let body = {

      email: data.username,

      password: data.password,

      favorite: [],

      id: new Date().toISOString(),

    };

    await firestore().collection('users').add(body);

    console.log('add');

  };

  const onRegister = () => {

    if (data.username.length == 0 || data.password.length == 0) {

      Alert.alert('Thông báo', 'Vui lòng nhập đầy đủ thông tin', [

        {text: 'Okay'},

      ]);

    } else if (data.password !== data.confirm\_password) {

      Alert.alert('Thông báo', 'Mật khẩu không trùng khớp!', [{text: 'Okay'}]);

    } else if (!isEmail(data.username)) {

      Alert.alert('Thông báo', 'Vui lòng nhập đúng định dạng email', [

        {text: 'Okay'},

      ]);

    } else if (data.password.length < 6) {

      Alert.alert('Thông báo', 'Mật khẩu phải nhiều hơn 6 ký tự', [

        {text: 'Okay'},

      ]);

    } else {

      try {

        auth()

          .createUserWithEmailAndPassword(data.username, data.password)

          .then(() => {

            InserUserToTable();

            navigation.goBack();

          })

          .catch(error => {

            if (error.code === 'auth/email-already-in-use') {

              Alert.alert('Thất bại!', 'Email đã được sử dụng!', [

                {text: 'Okay'},

              ]);

            }

            if (error.code === 'auth/invalid-email') {

              Alert.alert('Thất bại!', 'Email bạn nhập không hợp lệ!', [

                {text: 'Okay'},

              ]);

            }

            console.error(error);

          });

      } catch (error) {

        Alert.alert('Register fail!', error, [{text: 'Okay'}]);

      }

    }

  };

  return (

    <SafeAreaView style={{backgroundColor: '#FFF', height: '100%', flex: 1}}>

      <Image

        source={require('../../images/nameLogin.png')}

        style={{

          width: 163 \* 0.75,

          height: 119 \* 0.75,

          alignSelf: 'center',

          marginTop: 50,

        }}

      />

      <View

        style={{

          flexDirection: 'row',

          alignItems: 'center',

          marginHorizontal: 55,

          borderWidth: 2,

          marginTop: 50,

          paddingHorizontal: 10,

          borderColor: '#00a46c',

          borderRadius: 23,

          paddingVertical: 2,

        }}>

        <Icon name="mail" color="#00a46c" size={24} />

        <TextInput

          placeholder="Email"

          placeholderTextColor="#00a46c"

          style={{paddingHorizontal: 10, marginRight: 10, width: width / 2.2}}

          autoCapitalize="none"

          onChangeText={val => textInputChange(val)}

        />

        {data.check\_textInputChange ? (

          <Animatable.View animation="bounceIn">

            <Feather name="check-circle" color="green" size={20} />

          </Animatable.View>

        ) : null}

      </View>

      <View

        style={{

          flexDirection: 'row',

          alignItems: 'center',

          marginHorizontal: 55,

          borderWidth: 2,

          marginTop: 15,

          paddingHorizontal: 10,

          borderColor: '#00a46c',

          borderRadius: 23,

          paddingVertical: 2,

        }}>

        <Icon name="md-lock-closed" color="#00a46c" size={28} />

        <TextInput

          placeholder="Password"

          placeholderTextColor="#00a46c"

          style={{paddingHorizontal: 10, marginRight: 10, width: width / 2.2}}

          secureTextEntry={data.secureTextEntry ? true : false}

          autoCapitalize="none"

          onChangeText={val => handlePasswordChange(val)}

        />

        <TouchableOpacity onPress={updateSecureTextEntry}>

          {data.secureTextEntry ? (

            <Feather name="eye-off" color="#00a46c" size={22} />

          ) : (

            <Feather name="eye" color="#00a46c" size={22} />

          )}

        </TouchableOpacity>

      </View>

      <View

        style={{

          flexDirection: 'row',

          alignItems: 'center',

          marginHorizontal: 55,

          borderWidth: 2,

          marginTop: 15,

          paddingHorizontal: 10,

          borderColor: '#00a46c',

          borderRadius: 23,

          paddingVertical: 2,

        }}>

        <Icon name="md-lock-closed" color="#00a46c" size={28} />

        <TextInput

          placeholder=" Confirm Password"

          placeholderTextColor="#00a46c"

          style={{paddingHorizontal: 10, marginRight: 10, width: width / 2.2}}

          secureTextEntry={data.confirm\_secureTextEntry ? true : false}

          autoCapitalize="none"

          onChangeText={val => handleConfirmPasswordChange(val)}

        />

        <TouchableOpacity onPress={updateConfirmSecureTextEntry}>

          {data.confirm\_secureTextEntry ? (

            <Feather name="eye-off" color="#00a46c" size={22} />

          ) : (

            <Feather name="eye" color="#00a46c" size={22} />

          )}

        </TouchableOpacity>

      </View>

      <TouchableOpacity

        onPress={() => {

          onRegister();

        }}

        style={{

          marginHorizontal: 55,

          alignItems: 'center',

          justifyContent: 'center',

          marginTop: 30,

          backgroundColor: '#00a46c',

          paddingVertical: 16,

          borderRadius: 23,

        }}>

        <Text

          style={{

            color: 'white',

            fontFamily: 'SemiBold',

            fontSize: 20,

          }}>

          Register

        </Text>

      </TouchableOpacity>

      <TouchableOpacity onPress={() => navigation.navigate('Login')}>

        <Text

          style={{

            alignSelf: 'center',

            color: '#00a46c',

            fontFamily: 'SemiBold',

            paddingVertical: 25,

            fontSize: 18,

          }}>

          Login

        </Text>

      </TouchableOpacity>

    </SafeAreaView>

  );

};

export default Register;

**2.6 Index giỏ hàng**

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

import {

  View,

  Text,

  Image,

  SafeAreaView,

  ScrollView,

  Dimensions,

  TouchableHighlight,

  Alert,

  //   TouchableOpacity,

} from 'react-native';

import SwiperComponent from '../../components/SwiperComponent';

import {TouchableOpacity} from 'react-native-gesture-handler';

import styles from './styles';

import Icon from 'react-native-vector-icons/AntDesign';

import Icon1 from 'react-native-vector-icons/Ionicons';

import auth from '@react-native-firebase/auth';

import DataProduct from '../../Data/DataProduct';

import firestore from '@react-native-firebase/firestore';

import {convertToNumberCommas} from '../../utilities';

const {width, height} = Dimensions.get('window');

const DataProducts = DataProduct;

const ShoppingCart = ({navigation}) => {

  const [numberProduct, setNumberProduct] = useState(1);

  const [dataShopping, setDataShopping] = useState([]);

  const [dataItemAll, setDataItemAll] = useState([]);

  const [count, setCount] = useState(1);

  const [totalPrice, setTotalPrice] = useState(0);

  useEffect(() => {

    const users = auth().currentUser;

    const getShoppingByUser = async () => {

      let AllProducts = [];

      const Data = DataProducts;

      Data.map((item, index) => {

        item.menu.map((itemProduct, index) => {

          AllProducts.push(itemProduct);

        });

      });

      setDataItemAll(AllProducts);

      try {

        const shoppingCart = await firestore().collection('ShoppingCart').get();

        let box = shoppingCart.docs.map(doc => {

          return {

            id: doc.id,

            ...doc.data(),

          };

        });

        let shoppingbyUser = box.filter(element => {

          return element.email === users?.email;

        });

        for (let i = 0; i < shoppingbyUser.length; i++) {

          for (let y = 0; y < AllProducts.length; y++) {

            if (shoppingbyUser[i].idProduct == AllProducts[y].menuId) {

              // let photo = AllProducts[i].photo;

              let avatarImage = AllProducts[y].avatarImage;

              shoppingbyUser[i] = Object.assign(shoppingbyUser[i], {

                avatarImage,

              });

            }

          }

        }

        setDataShopping(shoppingbyUser);

        converTotalPrice(shoppingbyUser);

      } catch (error) {

        console.log('error\_error', error);

      }

    };

    getShoppingByUser();

  }, []);

  const handledel = async (item, index) => {

    let DataShopping = [];

    for (let i = 0; i < dataShopping.length; i++) {

      if (item.idProduct != dataShopping[i].menuId && i != index) {

        DataShopping.push(dataShopping[i]);

      }

    }

    // console.log(DataShopping);

    setDataShopping(DataShopping);

    converTotalPrice(DataShopping);

    await firestore().collection('ShoppingCart').doc(item.id).delete();

  };

  const handleDetail = (id, count) => {

    // console.log(dataItemAll);

    if (dataItemAll.length != 0) {

      for (let i = 0; i < dataItemAll.length; i++) {

        if (dataItemAll[i].menuId == id) {

          navigation.navigate('Detail', {

            count: count,

            dataItem: dataItemAll[i],

          });

        }

      }

    }

  };

  const onHandleShopping = async (value, item, index) => {

    console.log(item.amount, index);

    if (item.amount > 0 || value == 1) {

      let data = dataShopping;

      let Count = 0;

      for (let i = 0; i < dataShopping.length; i++) {

        let count = data[i].amount;

        if (dataShopping[i].menuId == item.menuId && i == index) {

          data[i].amount = count + value;

          Count = data[i].amount;

        }

      }

      await setDataShopping(data);

      await firestore()

        .collection('ShoppingCart')

        .doc(item.id)

        .update({amount: Count});

      setCount(count + value);

      converTotalPrice(data);

    }

  };

  const converTotalPrice = totalArr => {

    let CountTotal = 0;

    for (let i = 0; i < totalArr.length; i++) {

      CountTotal = CountTotal + totalArr[i].amount \* totalArr[i].priceProduct;

    }

    setTotalPrice(CountTotal);

  };

  const handlePay = async () => {

    for (let i = 0; i < dataShopping.length; i++) {

      await firestore()

        .collection('ShoppingCart')

        .doc(dataShopping[i].id)

        .delete();

    }

    Alert.alert('Thông báo', 'Thanh toán thành công', [

      {text: 'OK', onPress: () => console.log('OK Pressed')},

    ]);

    setDataShopping([]);

    setTotalPrice(0);

  };

  return (

    <SafeAreaView style={styles.container}>

      {/\* Header \*/}

      <View

        style={{

          height: 50,

          // width: width,

          flexDirection: 'row',

          justifyContent: 'space-between',

          paddingHorizontal: 20,

          backgroundColor: '#00a46c',

          alignItems: 'center',

        }}>

        <TouchableOpacity onPress={() => navigation.goBack()}>

          <Icon1 name="arrow-back" size={30} color="white" />

        </TouchableOpacity>

        <Text style={{fontSize: 20, color: 'white'}}>Giỏ Hàng</Text>

        <TouchableOpacity

          onPress={() => navigation.navigate('Home')}

          style={styles.TopPlus}>

          <Icon name="plus" size={20} color="white" />

        </TouchableOpacity>

      </View>

      <ScrollView

        showsHorizontalScrollIndicator={false}

        showsVerticalScrollIndicator={false}>

        {dataShopping.map((item, index) => {

          return (

            <View

              key={index}

              style={{

                marginTop: 16,

                height: 120,

                marginHorizontal: 6,

                backgroundColor: 'white',

                borderRadius: 20,

                flexDirection: 'row',

              }}>

              <View

                style={{

                  position: 'absolute',

                  right: -5,

                  top: -10,

                }}>

                <TouchableOpacity

                  hitSlop={{top: 10, left: 10, right: 10, bottom: 10}}

                  onPress={() => {

                    handledel(item, index);

                  }}

                  style={{

                    height: 27,

                    width: 27,

                    justifyContent: 'center',

                    alignItems: 'center',

                  }}>

                  <Icon name="closecircle" size={27} color="gray" />

                </TouchableOpacity>

              </View>

              <Image

                source={item.avatarImage}

                style={{

                  height: 120,

                  width: 120,

                  borderBottomLeftRadius: 20,

                  borderTopLeftRadius: 20,

                }}

              />

              <View

                style={{

                  marginHorizontal: 12,

                  width: width - 100 - 32,

                  paddingVertical: 6,

                }}>

                <Text

                  numberOfLines={1}

                  style={{

                    fontSize: 16,

                    color: 'green',

                    fontWeight: '500',

                    lineHeight: 24,

                  }}>

                  {' '}

                  {item.nameProduct}

                </Text>

                <Text

                  numberOfLines={1}

                  style={{

                    fontSize: 16,

                    color: 'black',

                    fontWeight: '500',

                    lineHeight: 24,

                  }}>

                  {' '}

                  Giá tiền: {convertToNumberCommas(item.priceProduct)} Đ

                </Text>

                <Text

                  numberOfLines={1}

                  style={{

                    fontSize: 16,

                    color: 'black',

                    fontWeight: '500',

                    lineHeight: 24,

                  }}>

                  {' '}

                  Tổng tiền :{' '}

                  {convertToNumberCommas(

                    Number(item.amount) \* Number(item.priceProduct),

                  )}{' '}

                  Đ

                </Text>

                <View

                  style={{

                    height: 36,

                    flexDirection: 'row',

                    justifyContent: 'space-between',

                    marginHorizontal: 10,

                    alignItems: 'center',

                  }}>

                  <TouchableOpacity

                    onPress={() => {

                      handleDetail(item.idProduct, item.amount);

                    }}>

                    <Text

                      style={{

                        color: '#00a46c',

                        // fontWeight: 'bold',

                        fontSize: 14,

                        // textDecorationLine: 0.2,

                      }}>

                      Xem chi tiêt

                    </Text>

                  </TouchableOpacity>

                  <View style={styles.ViewNumProduct}>

                    <TouchableOpacity

                      onPress={() => onHandleShopping(-1, item, index)}

                      style={styles.ButtomPlus}>

                      <Icon name="minus" size={20} color="gray" />

                    </TouchableOpacity>

                    <Text

                      style={{

                        color: '#62636a',

                        fontWeight: 'bold',

                        fontSize: 20,

                      }}>

                      {item.amount}

                    </Text>

                    <TouchableOpacity

                      onPress={() => onHandleShopping(1, item, index)}

                      style={styles.ButtomPlus}>

                      <Icon name="plus" size={20} color="gray" />

                    </TouchableOpacity>

                  </View>

                </View>

              </View>

            </View>

          );

        })}

        {totalPrice ? (

          <View

            style={{

              height: 50,

              // width: width,

              flexDirection: 'row',

              justifyContent: 'space-around',

              alignItems: 'center',

              borderRadius: 12,

              backgroundColor: '#00a46c',

              margin: 12,

            }}>

            <Text style={{fontSize: 18, color: 'white'}}>Tổng tiền: </Text>

            <Text style={{fontSize: 18, color: 'white'}}>

              {convertToNumberCommas(totalPrice)} Đ

            </Text>

          </View>

        ) : null}

      </ScrollView>

      <View style={styles.ViewBottomContainer}>

        <TouchableOpacity

          onPress={() => {

            handlePay();

          }}

          style={styles.ViewButtom}>

          <Text

            style={{

              color: '#FFF',

              fontSize: 18,

            }}>

            Thanh Toán

          </Text>

        </TouchableOpacity>

      </View>

    </SafeAreaView>

  );

};

export default ShoppingCart;

**KẾT LUẬN**

* **Kết quản đạt được**

Trong thời gian nghiên cứu và vận dụng những kiến thức đã được trang bị ở trường và các lĩnh vực khác, em đã viết được Ứng dụng bán hàng nhằm mục đích bán hàng. Thông qua dự án này em cũng hiểu thêm nhiều hơn về kiến thức lập trình hướng đối tượng , Tuy nhiên vẫn còn hạn chế về việc chưa phát triển sâu hơn về ứng dụng và mới chỉ đạt được những yêu cầu cơ bản của ứng dụng. Nhóm em rất mong nhận được sự đóng góp của mọi người để có thể hoàn thiện hơn. Em hy vọng những kiến thức em có được sau khoá học này sẽ là tiền đề giúp em phát triển mạnh hơn về những chương trình khác, những ứng dụng khác chất lượng hơn để có thể đóng góp một phần trong làng công nghệ thông tin.

* **Hướng phát triển**

Bên cạnh việc phát triển ứng dụng mang tính khéo léo đòi hỏi sự kiên nhẫn thì những ứng dụng mang tính chất phổ biến cũng trở thành những ứng dụng hot trong những năm trở lại đây. Chính vì vậy mô hình ứng dụng bán hàng sẽ là đối tượng em tiếp tục nghiên cứu trong thời gian tới, đồng thời em sẽ cải thiện hơn về mặt giao diện và chức năng