**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

****

**TIỂU LUẬN MÔN HỌC**

**THỰC TẬP DOANH NGHIỆP**

**TÊN TIỂU LUẬN**

**XÂY DỰNG ỨNG DỤNG KẾ TOÁN GIA ĐÌNH SỬ DỤNG CÔNG NGHỆ FLUTTER**

**GVHD: Th.S TRẦN VĂN HỮU**

**SVTH: ĐOÀN CÔNG CHÍNH**

**MSSV: 0933798706**

**LỚP: D17HT01**

**BÌNH DƯƠNG – 12/2020**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

****

**TIỂU LUẬN MÔN HỌC**

**THỰC TẬP DOANH NGHIỆP**

**TÊN TIỂU LUẬN**

**XÂY DỰNG ỨNG DỤNG KẾ TOÁN GIA ĐÌNH SỬ DỤNG CÔNG NGHỆ FLUTTER**

**GVHD: Th.S TRẦN VĂN HỮU**

**SVTH: ĐOÀN CÔNG CHÍNH**

**MSSV: 0933798706**

**LỚP: D17HT01**

**BÌNH DƯƠNG – 12/2020**

MỤC LỤC

[MỤC LỤC ii](#_Toc58977078)

[DANH MỤC HÌNH iv](#_Toc58977079)

[DANH MỤC BẢNG v](#_Toc58977080)

[MỞ ĐẦU 1](#_Toc58977081)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ 2](#_Toc58977082)

[1.1. Android 2](#_Toc58977083)

[1.2. IOS 2](#_Toc58977084)

[1.3. Flutter 3](#_Toc58977085)

[1.4. Firebase 4](#_Toc58977086)

[CHƯƠNG 2. GIỚI THIỆU ỨNG DỤNG 5](#_Toc58977087)

[2.1. Giao diện ứng dụng 5](#_Toc58977088)

[2.1.1. Giao diện đăng nhập hệ thống 5](#_Toc58977089)

[2.1.2. Giao diện trang Home 6](#_Toc58977090)

[2.1.3. Giao diện trang quản lý loại sản phẩm 7](#_Toc58977091)

[2.1.4. Giao diện trang thêm loại sản phẩm 8](#_Toc58977092)

[2.1.5. Giao diện trang quản lý buôn bán 9](#_Toc58977093)

[2.1.6. Giao diện form nhập sản phẩm 10](#_Toc58977094)

[2.1.7. Giao diện trang báo cáo 11](#_Toc58977095)

[2.2. Các chức năng chính của ứng dụng 12](#_Toc58977096)

[2.2.1. Danh sách các Actor 12](#_Toc58977097)

[2.2.2. Biểu đồ hệ thống 12](#_Toc58977098)

[2.2.3. Danh sách các Use case 12](#_Toc58977099)

[2.2.4. Biểu đồ Use case toàn bộ hệ thống 13](#_Toc58977100)

[CHƯƠNG 3. THIẾT KẾ CHƯƠNG TRÌNH 14](#_Toc58977101)

[3.1. Thiết kế CSDL 14](#_Toc58977102)

[3.1.1. Lượt đồ tổng quan 14](#_Toc58977103)

[3.1.2. Bảng thông tin tài khoản (Users) 14](#_Toc58977104)

[3.1.3. Bảng loại sản phẩm(Categories) 14](#_Toc58977105)

[3.1.4. Bảng hoạt động buôn bán (Activities) 15](#_Toc58977106)

[3.1.5. Bảng hình ảnh(Images) 15](#_Toc58977107)

[3.2. Code chương trình 16](#_Toc58977108)

[3.2.1. Code trang chủ 16](#_Toc58977109)

[3.2.2. Code Model sản phẩm 19](#_Toc58977110)

[3.2.3. Code Service Auth 20](#_Toc58977111)

[KẾT LUẬN 22](#_Toc58977112)

[TÀI LIỆU THAM KHẢO 23](#_Toc58977113)

DANH MỤC HÌNH

[**Hình 2.1:** Giao diện trang người dùng 5](#_Toc58977069)

[**Hình 2.2:** Giao diện trang chủ trang quản lý 6](#_Toc58977070)

[**Hình 2.3:** Giao diện trang quản lý loại sản phẩm 7](#_Toc58977071)

[**Hình 2.4:** Giao diện trang thêm loại sản phẩm 8](#_Toc58977072)

[**Hình 2.5:** Giao diện trang quản lý buôn bán 9](#_Toc58977073)

[**Hình 2.6:** Giao diện form nhập sản phẩm 10](#_Toc58977074)

[**Hình 2.7:** Giao diện trang báo cáo 11](#_Toc58977075)

[**Hình 2.8:** Biểu đồ hệ thống 12](#_Toc58977076)

[**Hình 2.9:** Biểu đồ Usecase toàn hệ thống 13](#_Toc58977077)

DANH MỤC BẢNG

[**Bảng 2.1:** Danh sách các Actor 12](#_Toc58977114)

[**Bảng 2.2:** Danh sách các Usecase 13](#_Toc58977115)

[**Bảng 3.1:** Lượt đồ tổng quan 14](#_Toc58977116)

[**Bảng 3.2:** Bảng cở sở dữ liệu tài khoản 14](#_Toc58977117)

[**Bảng 3.3:** Bảng cở sở dữ liệu loại sản phẩm 15](#_Toc58977118)

[**Bảng 3.4:** Bảng cở sở dữ liệu hoạt động 15](#_Toc58977119)

[**Bảng 3.5:** Bảng cở sở dữ liệu hình ảnh 15](#_Toc58977120)

MỞ ĐẦU

Kế toán là một quá trình liên quan đến chi tiết. Nó liên quan đến các quy định, luật lệ, thuế và nhiều tính toán phức tạp khác. Chính vì vầy, sử dụng phần mềm kế toán là một phương pháp tiết kiệm tối đa và hiệu quả nhất quỹ thời gian bận rộn của bạn. Một phần mềm kế toán đã được thiết kế phù hợp với quy định, luật lệ, thuế… hiện hành sẽ giúp doanh nghiệp luôn tuân thủ với luật pháp. Điều doanh nghiệp cần làm là nhập liệu thật chính xác, đầy đủ và in ra báo cáo chỉ với một cú click chuột.

Hầu hết các phần mềm kế toán đều có phần hỗ trợ hướng dẫn cách sử dụng toàn bộ quy trình nhập liệu thông tin khách hàng và nhân viên. Một khi nắm được cách làm, bạn không cần xem lại phần hỗ trợ này nữa và mọi thao tác sẽ nhanh hơn. Hình thức hướng dẫn này góp phần giảm thiểu sai sót và cải thiện chất lượng số sách kế toán của doanh nghiệp.

Độ chính xác là tuyệt đối. Vì dữ liệu được cung cấp bằng phần mềm kế toán mang tính nhất quán cao. Trong khi với công tác kế toán thủ công, thông tin trên một chứng từ co thể do nhiều kế toán viên ghi chép trên nhiều sổ sách theo bản chất nghiệp vụ phụ trách nên dễ dẫn đến trình trạng sai lệch dữ liệu trên các sổ sách khi tổng hợp lại, kéo theo công tác kế toán tổng hợp sai lệch gây thiệt hại nặng cho doanh nghiệp.

Chính vì vậy trong bài báo cáo này tôi chọn đề tài về: “**Xây dựng ứng dụng kế toán gia đình sử dụng công nghệ Flutter**”. Đây là một hệ thống đơn giản nhưng đủ mạnh để cho phép nhanh chóng tính toán các mặt hàng mua bán trong gia đình.

Đề tài tiểu luận được phân các chương như sau:

1. **Giới thiệu tổng quan đề tài**
2. **Cơ sở lý thuyết**
3. **Phân tích thiết kế hệ thống**
4. **Cài đặt ứng dụng “Accouting App”**
5. **Tổng kết**
6. TỔNG QUAN VỀ CÔNG NGHỆ
   1. Android

Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google, sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008.

Chính mã nguồn mở cùng với giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play (cửa hàng ứng dụng chính của Android) ước tính khoảng 25 tỷ lượt. Hiện nay con số này đã giảm xuống do sự ảnh hưởng lớn của iOS từ Apple và một phần nhỏ của Windows Phone, tuy nhiên Android vẫn dẫn đầu thị phần.

* 1. IOS

IOS là hệ điều hành di dộng của Apple được cài sẵn trên các thiết bị iPhone, iPad và iPod touch hay Apple TV của hãng và do chính Apple nghiên cứu và phát triển. Ban đầu iOS được gọi là iPhoneOS và đến tháng 6 năm 2010 thì chính thức được đổi gọi tên là “iOS” như hiện nay.

Hệ điều hành iOS được công bố cùng chiếc iPhone 2G tại Macworld Conference & Expo vào ngày 9/1/2007 và chính thức phát hành trong tháng 6 sau đó. Khi ra mắt, hệ điều hành này không không hỗ trợ các ứng dụng từ các nhà phát triển thứ 3 (có thể hiểu là các lập trình viên ứng dụng không thuộc Apple). Nguyên nhân của việc từ chối các ứng dụng từ bên thứ 3 được Steve Jobs đưa ra là các nhà phát triển có thể xây dựng ứng dụng trùng chức năng với các ứng dụng gốc trên iPhone. iOS ra đời đã đánh dấu và mở ra một kỉ nguyên mới cho sự phát triển của điện thoại màn hình cảm ứng.

* 1. Flutter

Flutter được phát triển nhằm giải quyết bài toán thường gặp trong mobile là Fast Development và Native Performance. Nếu như React Native chỉ đảm bảo Fast Development và code native thuần chỉ đảm bảo Native Performance thì Flutter làm được cả 2 điều trên.

Flutter phát triển ứng dụng nhanh chóng: Tính năng hot reload của Flutter giúp bạn nhanh chóng và dễ dàng thử nghiệm, xây dựng giao diện người dùng, thêm tính năng và sửa lỗi nhanh hơn. Trải nghiệm tải lại lần thứ hai, mà không làm mất trạng thái, trên emulator, simulator và device cho iOS và Android.

UI đẹp và biểu cảm: Thỏa mãn người dùng của bạn với các widget built-in đẹp mắt của Flutter theo Material Design và Cupertino (iOS-flavor), các API chuyển động phong phú, scroll tự nhiên mượt mà và tự nhận thức được nền tảng.

Framework hiện đại và reactive: Dễ dàng tạo giao diện người dùng của bạn với framework hiện đại, reactive của Flutter và tập hợp các platform, layout và widget phong phú. Giải quyết các thách thức giao diện người dùng khó khăn của bạn với các API mạnh mẽ và linh hoạt cho 2D, animation, gesture, hiệu ứng và hơn thế nữa.

Truy cập các tính năng và SDK native: Làm cho ứng dụng của bạn trở nên sống động với API của platform, SDK của bên thứ ba và native code. Flutter cho phép bạn sử dụng lại mã Java, Swift và ObjC hiện tại của mình và truy cập các tính năng và SDK native trên iOS và Android.

Phát triển ứng dụng thống nhất: Flutter có các công cụ và thư viện để giúp bạn dễ dàng đưa ý tưởng của mình vào cuộc sống trên iOS và Android. Nếu bạn chưa có kinh nghiệm phát triển trên thiết bị di động, thì Flutter là một cách dễ dàng và nhanh chóng để xây dựng các ứng dụng di động tuyệt đẹp. Nếu bạn là một nhà phát triển iOS hoặc Android có kinh nghiệm, bạn có thể sử dụng Flutter cho các View của bạn và tận dụng nhiều code Java / Kotlin / ObjC / Swift hiện có của bạn.

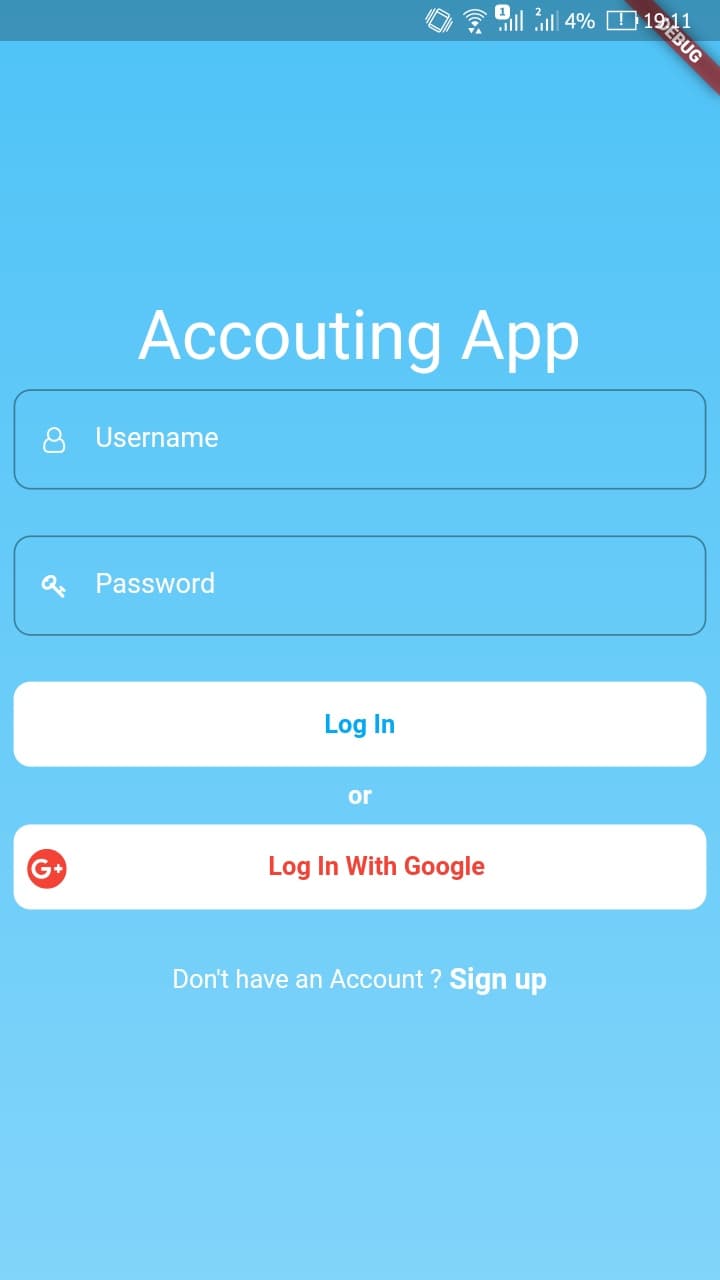
* 1. Firebase

Firebase là một nền tảng để phát triển ứng dụng di động và trang web, bao gồm các API đơn giản và mạnh mẽ mà không cần backend hay server. Lợi ích của Firebase là gì? Firebase còn giúp các lập trình viên rút ngắn thời gian triển khai và mở rộng quy mô của ứng dụng mà họ đang phát triển.

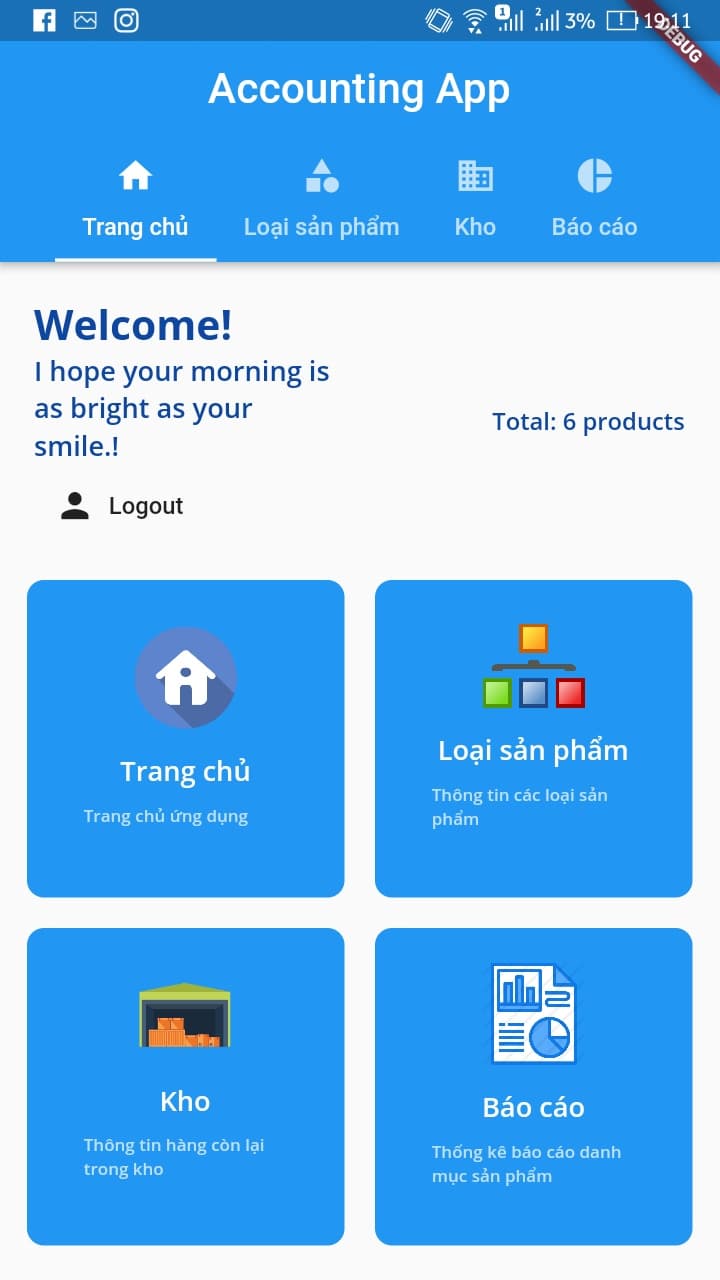
Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu. Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

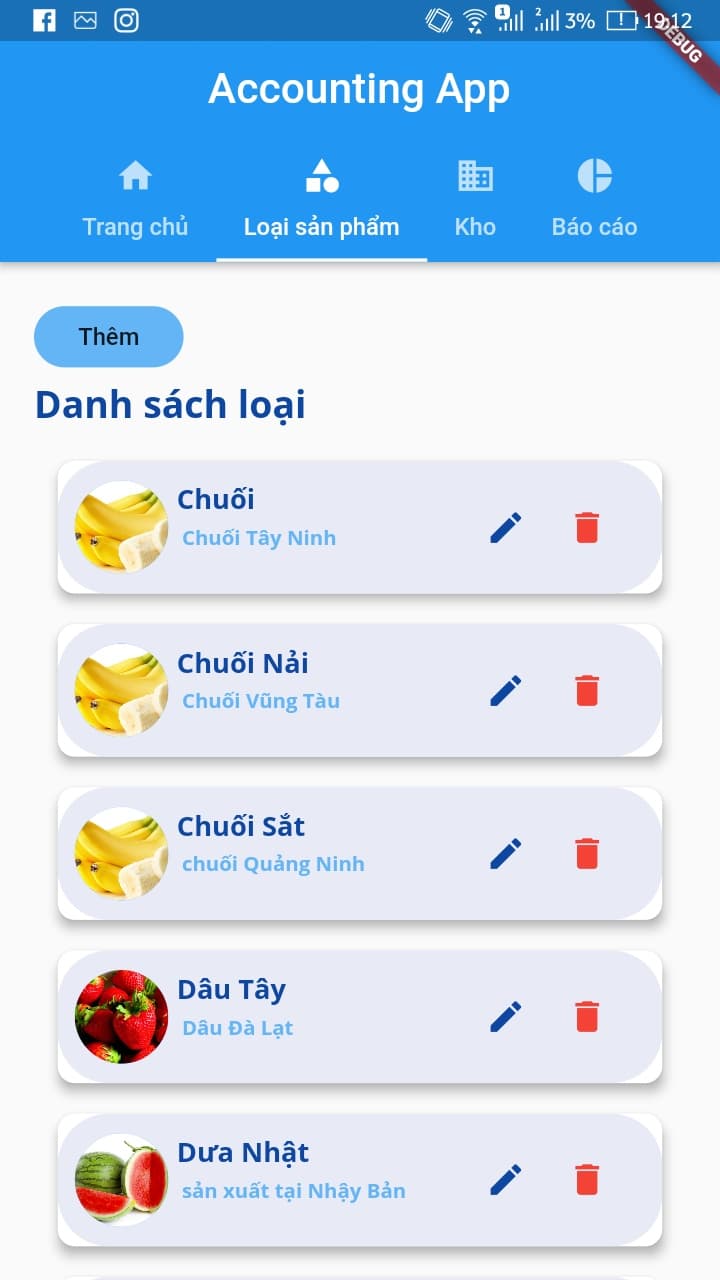
2. GIỚI THIỆU ỨNG DỤNG
   1. Giao diện ứng dụng
      1. Giao diện đăng nhập hệ thống



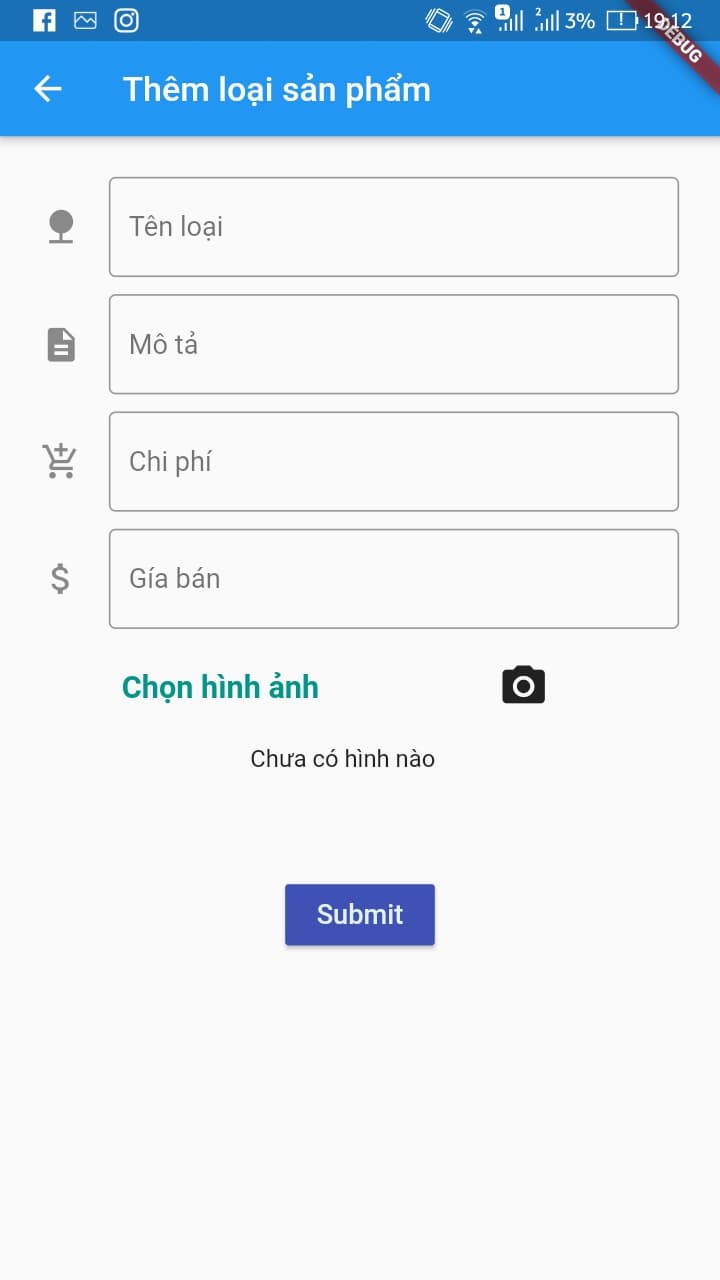
* + - 1. Giao diện trang người dùng
* **Chức năng chính:** Đăng nhập, đăng xuất, xem bảng xếp hạng, xem các bài tập, nộp bài tập, vào xem đội
  + 1. Giao diện trang Home



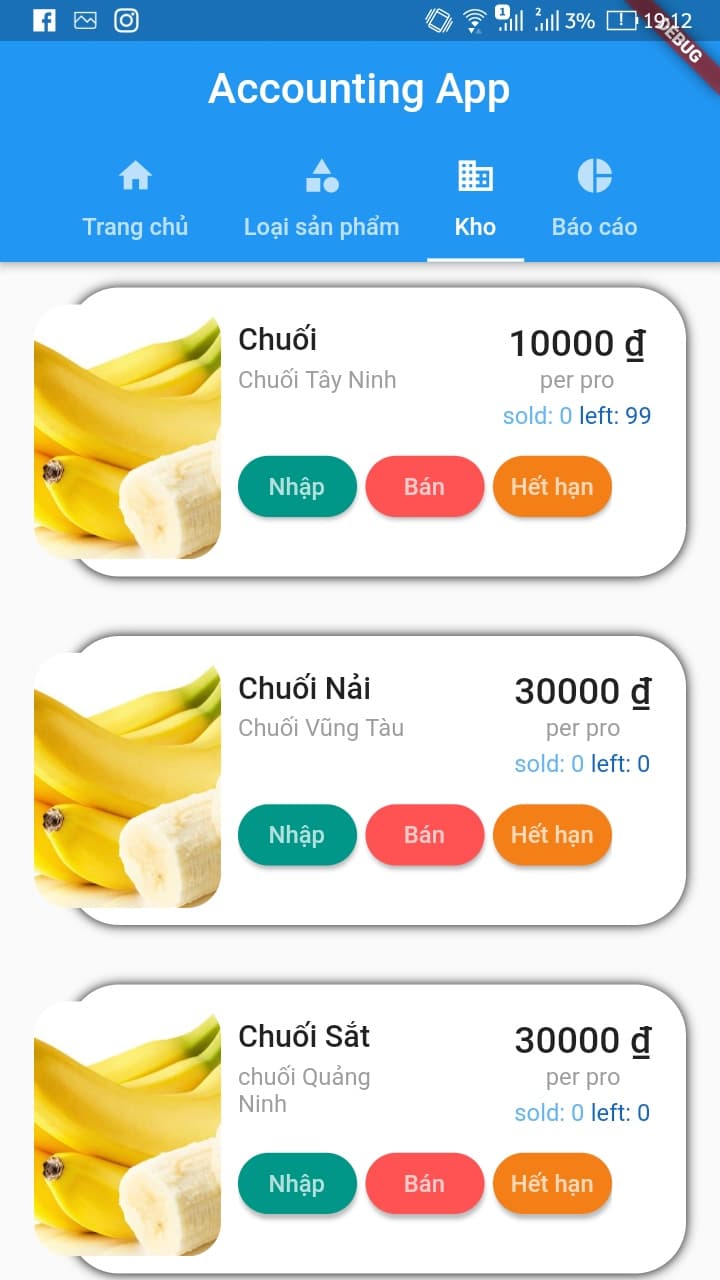
* + - 1. Giao diện trang chủ trang quản lý
* **Chức năng chính:** Vào trang quản lý loại sản phẩm, vào trang quản lý buôn bán, báo cáo, đăng xuất.
  + 1. Giao diện trang quản lý loại sản phẩm



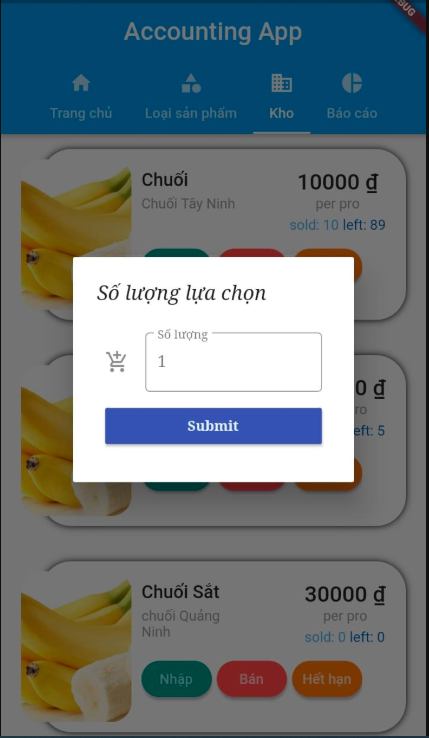
* + - 1. Giao diện trang quản lý loại sản phẩm
* **Chức năng chính:** Thêm sản phẩm, xem danh sách loại sản phẩm, sửa sản phẩm, xóa sản phẩm.
  + 1. Giao diện trang thêm loại sản phẩm



* + - 1. Giao diện trang thêm loại sản phẩm
* **Chức năng chính:** Nhập tên loại, nhập mô tả, nhập chi phí sản xuất, nhập giá bán, chọn hình ảnh từ thư viện hoặc chụp hình, gửi yêu cầu.
  + 1. Giao diện trang quản lý buôn bán



* + - 1. Giao diện trang quản lý buôn bán
* **Chức năng chính:** Xem danh sách sản phẩm, nhập sản phẩm, bán sản phẩm, sản phẩm hết hạn
  + 1. Giao diện form nhập sản phẩm



* + - 1. Giao diện form nhập sản phẩm
* **Chức năng chính:** nhập số lương, gửi yêu cầu.
  + 1. Giao diện trang báo cáo

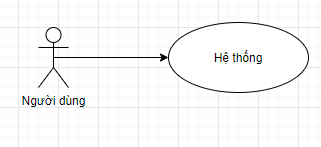


* + - 1. Giao diện trang báo cáo
* **Chức năng chính:** Xem báo cáo
  1. Các chức năng chính của ứng dụng
     1. Danh sách các Actor

Một actor hay tác nhân ngoài là một vai trò của một hay nhiều người hay vật thể trong sự tương tác với hệ thống.

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Mô tả** |
| 1 | Người dùng | Là người sử dụng hệ thống, có quyền truy cập vào hệ thống |

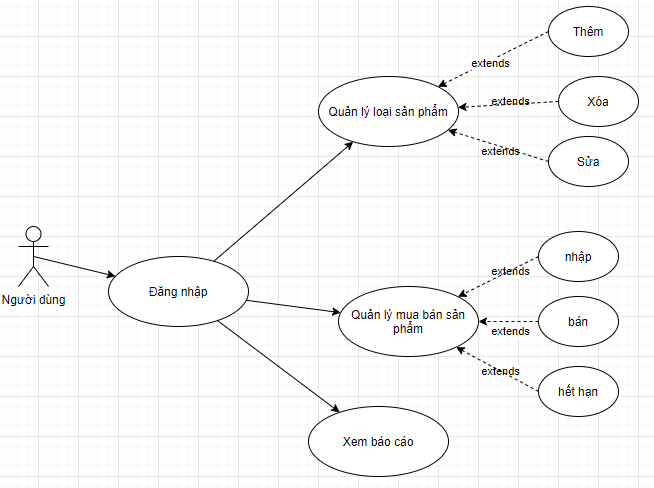
* + - * 1. Danh sách các Actor
    1. Biểu đồ hệ thống



* + - 1. Biểu đồ hệ thống
    1. Danh sách các Use case

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Use case | Ý nghĩa | Actor |
| 1 | Đăng nhập | Cho phép người sử dụng truy cập vào hệ thống | Người dùng |
| 2 | Đăng xuất | Cho phép người sử dụng đăng xuất khỏi hệ thống | Người dùng |
| 3 | Xem danh sách loại sản phẩm | Cho phép người sử dụng xem danh sách loại sản phẩm đã thêm | Người dùng |
| 4 | Thêm loại sản phẩm | Cho phép người sử thêm một sản phẩm | Người dùng |
| 5 | Chỉnh sửa loại sản phẩm | Cho phép người sử dụng có thể thay đổi loại sản phẩm đã thêm | Người dùng |
| 6 | Xóa loại sản phẩm | Cho phép người sử dụng có thể xóa một loại sản phẩm | Người dùng |
| 7 | Xem danh sách các sản phẩm bán | Cho phép người sử dụng có thể xem danh sách các sản phẩm đang hoạt động | Người dụng |
| 8 | Nhập hàng | Cho phép người sử dụng thêm số lượng hàng hóa | Người dùng |
| 9 | Bán hàng | Cho phép người sử dụng bán hàng hóa | Người dùng |
| 10 | Hết hạn | Cho phép người sử dụng xóa các hàng hóa bị hết hạn | Người dùng |
| 11 | Xem báo cáo | Cho phép người sử dụng xem báo cáo các mặt hàng đã thêm hoặc bán trong thời gian nhất định | Người dùng |

* + - * 1. Danh sách các Usecase
    1. Biểu đồ Use case toàn bộ hệ thống



* + - 1. Biểu đồ Usecase toàn hệ thống

1. THIẾT KẾ CHƯƠNG TRÌNH
   1. Thiết kế CSDL
      1. Lượt đồ tổng quan

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Mô tả** |
| 1 | tbl\_users | Danh sách thông tin thông báo |
| 2 | tbl\_activities | Danh dánh mối quan hệ thông báo và quyền |
| 3 | tbl\_categories | Danh sách thông tin quyền |
| 4 | tbl\_images | Danh sách thông tin biểu tượng |

* + - * 1. Lượt đồ tổng quan
    1. Bảng thông tin tài khoản (Users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tbl\_users | | | | |
|  | **Tên trường** | **Loại dữ liệu** | **Giới hạn chiều dài** | **Mô tả** |
| 1 | ID | String | 255 | Lưu mã tài khoản( Khóa chính) |
| 2 | Username | String | 255 | Lưu tên đăng nhập |
| 3 | Password | String | 255 | Lưu mật khẩu |

* + - * 1. Bảng cở sở dữ liệu tài khoản
    1. Bảng loại sản phẩm(Categories)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tbl\_categories | | | | |
|  | **Tên trường** | **Loại dữ liệu** | **Giới hạn chiều dài** | **Mô tả** |
| 1 | ID | String | 255 | Lưu mã loại sản phẩm( Khóa chính) |
| 2 | Cost | Int |  | Lưu giá sản phẩm |
| 3 | Description | String | 255 | Lưu mô tả sản phẩm |
| 4 | imageUrl | String | 255 | Lưu đường dẫn hình ảnh |
| 5 | Name | String | 255 | Lưu tên sản phẩm |
| 6 | Price | Int |  | Lưu giá sản phẩm |
| 7 | Quantify | Int |  | Lưu số lượng sản phẩm |
| 8 | Hiden | Bool |  | Lưu trạng thái sản phẩm |

* + - * 1. Bảng cở sở dữ liệu loại sản phẩm
    1. Bảng hoạt động buôn bán (Activities)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tbl\_activities | | | | |
|  | **Tên trường** | **Loại dữ liệu** | **Giới hạn chiều dài** | **Mô tả** |
| 1 | ID | String | 255 | Lưu mã hoạt động( Khóa chính) |
| 2 | cateID | String | 255 | Lưu mã loại( Khóa ngoại) |
| 3 | createAt | Datetime |  | Lưu ngày hoạt động |
| 4 | Quantify | Int |  | Lưu số lượng |
| 5 | Type | Int |  | Lưu loại hoạt động |

* + - * 1. Bảng cở sở dữ liệu hoạt động
    1. Bảng hình ảnh(Images)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tbl\_images | | | | |
|  | **Tên trường** | **Loại dữ liệu** | **Giới hạn chiều dài** | **Mô tả** |
| 1 | ID | String | 255 | Lưu mã hình ảnh( Khóa chính) |
| 2 | Date | String | 1000000 | Lưu dữ liệu hình ảnh |

* + - * 1. Bảng cở sở dữ liệu hình ảnh
  1. Code chương trình
     1. Code trang chủ

**return** Scaffold(  
 appBar: AppBar(  
 centerTitle: **true**,  
 title: **const** Text(  
 **'Accounting App'**,  
 style: TextStyle(  
 fontSize: 25.0,  
 fontWeight: FontWeight.*w600*,  
 color: Colors.*white*,  
 ),  
 ),  
 bottom: TabBar(  
 controller: **\_controller**,  
 isScrollable: **true**,  
 labelColor: Colors.*white*,  
 unselectedLabelColor: Colors.*white70*,  
 indicatorColor: Colors.*white*,  
 tabs: choices.map<Widget>(  
 (Choice choice) {  
 **return** Tab(  
 text: choice.**title**,  
 icon: Icon(  
 choice.**icon**,  
 ),  
 );  
 },  
 ).toList(),  
 ),  
 ),  
 body: TabBarView(  
 controller: **\_controller**,  
 children: <Widget>[  
 Column(  
 children: <Widget>[  
 SingleChildScrollView(  
 scrollDirection: Axis.**vertical**,  
 child: Padding(  
 padding: **const** EdgeInsets.all(20.0),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.**spaceBetween**,  
 children: <Widget>[  
 Column(  
 crossAxisAlignment: CrossAxisAlignment.**start**,  
 children: <Widget>[  
 Text(  
 **'Welcome!'**,  
 style: GoogleFonts.*openSans*(  
 textStyle: TextStyle(  
 color: Colors.*blue*[900],  
 fontSize: 24,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 ),  
 Container(  
 width: 180,  
 child: Text(  
 **'I hope your morning is as bright as your smile.!'**,  
 style: GoogleFonts.*openSans*(  
 textStyle: TextStyle(  
 color: Colors.*blue*[900],  
 fontSize: 16,  
 fontWeight: FontWeight.*w600*,  
 ),  
 ),  
 ),  
 ),  
 FlatButton.icon(  
 onPressed: () **async** {  
 **await** \_auth.signOut();  
 },  
 icon: Icon(Icons.*person*),  
 label: Text(**"Logout"**),  
 )  
  
 ],  
 ),  
 Text(  
 **'Total:** ${**total**} **products'**,  
 overflow: TextOverflow.**ellipsis**,  
 style: GoogleFonts.*openSans*(  
 textStyle: TextStyle(  
 color: Colors.*blue*[900],  
 fontSize: 14,  
 fontWeight: FontWeight.*w600*,  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 Flexible(  
 child: GridView.count(  
 childAspectRatio: 1.0,  
 padding: EdgeInsets.only(left: 16.0, right: 16.0),  
 crossAxisCount: 2,  
 crossAxisSpacing: 18,  
 mainAxisSpacing: 18,  
 children: choices.map(  
 (data) {  
 **return** GestureDetector(  
 onTap: () {  
 **\_controller**.**index** = choices.indexOf(data);  
 },  
 child: Container(  
 decoration: BoxDecoration(  
 color: Theme.*of*(context).**primaryColor**,  
 borderRadius: BorderRadius.circular(10.0),  
 ),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.**center**,  
 children: <Widget>[  
 Image.asset(  
 data.**imageUrl**,  
 width: 60,  
 fit: BoxFit.**cover**,  
 ),  
 SizedBox(  
 height: 14,  
 ),  
 Text(  
 data.**title**,  
 style: GoogleFonts.*openSans*(  
 textStyle: TextStyle(  
 color: Colors.*white*,  
 fontSize: 16.0,  
 fontWeight: FontWeight.*w600*,  
 ),  
 ),  
 ),  
 SizedBox(  
 height: 8.0,  
 ),  
 Container(  
 width: 120,  
 child: Text(  
 data.**subtitle**,  
 style: GoogleFonts.*openSans*(  
 textStyle: TextStyle(  
 color: Colors.*white70*,  
 fontSize: 10.0,  
 fontWeight: FontWeight.*w600*,  
 ),  
 ),  
 maxLines: 2,  
 ),  
 ),  
 SizedBox(  
 height: 14.0,  
 ),  
 ],  
 ),  
 ),  
 );  
 },  
 ).toList(),  
 ),  
 )  
 ],  
 ),  
 **new** category\_tab.CategoryTab(),  
 **new** store\_tab.StoreTab(),  
 **new** report\_tab.ReportTab(),  
 ],  
 ),  
);

* + 1. Code Model sản phẩm

import 'package:firebase\_database/firebase\_database.dart';  
  
class Category {  
 String id;  
 String name;  
 String description;  
 String imageUrl;  
 int cost;  
 int price;  
 int quantify;  
 bool hiden;  
  
 Category(this.name, this.description, this.imageUrl, this.cost, this.price,  
 this.quantify)  
 : this.hiden = false;  
  
 Category.empty()  
 : id = '0',  
 name = '',  
 description = '',  
 quantify = 0,  
 imageUrl = '',  
 hiden = false;  
  
 Category.fromMap(Map<String, dynamic> map)  
 : name = map['name'],  
 hiden = map['hiden'],  
 description = map['description'],  
 imageUrl = map['imageUrl'],  
 cost = map['cost'],  
 price = map['price'],  
 quantify = map['quantify'];  
  
 Category.fromSnapshot(DataSnapshot snapshot)  
 : id = snapshot.key,  
 name = snapshot.value['name'],  
 hiden = snapshot.value['hiden'],  
 description = snapshot.value['description'],  
 imageUrl = snapshot.value['imageUrl'],  
 cost = snapshot.value['cost'],  
 price = snapshot.value['price'],  
 quantify = snapshot.value['quantify'];  
  
 toJson() {  
 return {  
 'name': name,  
 'description': description,  
 'imageUrl': imageUrl,  
 'cost': cost,  
 'hiden': hiden,  
 'price': price,  
 'quantify': quantify  
 };  
 }  
}

* + 1. Code Service Auth

import 'package:accountingapp/models/user\_model.dart';  
import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:google\_sign\_in/google\_sign\_in.dart';  
  
class AuthService {  
 final FirebaseAuth \_auth = FirebaseAuth.*instance*;  
 GoogleSignIn \_googleSignIn = GoogleSignIn(scopes: ['email']);  
 *// create user obj based on Firebase* User \_userFromFirebaseUser(FirebaseUser user) {  
 return user != null ? User(uid: user.uid) : null;  
 }  
  
 *// auth change user stream* Stream<User> get user {  
 return \_auth.onAuthStateChanged  
 .map(\_userFromFirebaseUser);  
 }  
  
 *// sign in anon* Future signInAnon() async {  
 try {  
 AuthResult result = await \_auth.signInAnonymously();  
 FirebaseUser user = result.user;  
 return \_userFromFirebaseUser(user);  
 } catch (e) {  
 print("failed: " + e.toString());  
 return null;  
 }  
 }  
  
 *// sign in with email & password* Future signInUsernamePassword(String username, String password) async {  
 username += "@abc.com";  
 try {  
 AuthResult result = await \_auth.signInWithEmailAndPassword(  
 email: username, password: password);  
 FirebaseUser user = result.user;  
 return \_userFromFirebaseUser(user);  
 } catch (e) {  
 print("failed: " + e.toString());  
 return null;  
 }  
 }  
  
 *// sign in with google* Future googleSignIn() async {  
 try {  
 GoogleSignInAccount googleUser = await \_googleSignIn.signIn();  
 GoogleSignInAuthentication googleAuth = await googleUser.authentication;  
 final AuthCredential credential = GoogleAuthProvider.*getCredential*(  
 accessToken: googleAuth.accessToken,  
 idToken: googleAuth.idToken,  
 );  
 final FirebaseUser user = (await \_auth.signInWithCredential(credential)).user;  
 return \_userFromFirebaseUser(user);  
 } catch (e) {  
 print("failed: " + e.toString());  
 return null;  
 }  
 }  
  
 *// register with email & password* Future registerWithEmailAndPassword(String username, String password,  
 String fullname) async {  
 username += "@abc.com";  
 try {  
 AuthResult result = await \_auth.createUserWithEmailAndPassword(  
 email: username, password: password);  
 FirebaseUser user = result.user;  
 UserUpdateInfo userUpdateInfo = new UserUpdateInfo();  
 userUpdateInfo.displayName = fullname;  
 await user.updateProfile(userUpdateInfo);  
 await user.reload();  
 return \_userFromFirebaseUser(user);  
 } catch (e) {  
 print(e.toString());  
 return null;  
 }  
 }  
  
 *// sign out* Future signOut() async {  
 try {  
 return await \_auth.signOut();  
 } catch (e) {  
 print(e.toString());  
 return null;  
 }  
 }  
}

KẾT LUẬN

Sau một khoảng thời gian bắt đầu xây dựng với sự hướng dẫn của thầy Trần Văn Hữu, tôi đã hoàn thành đề tài “**Xây dựng ứng dụng kế toán gia đình sử dụng công nghệ Flutter**”. Tuy nhiên, trong quá trình phân tích và thiết kế do thời gian có hạn và khả năng và kinh nghiệm của bản thân còn hạn chế nên báo cáo này vẫn chưa thể tối ưu hết mực. Em mọng nhận được những lời nhận xét, đánh giá từ phía thầy, cô để em có thể khác phục và hoàn thiện hơn nữa về hệ thống ứng dụng này.

**1. Kết quả đạt được**

* + Giao diện có tính tương tác cao
  + Có đầy đủ chức năng thông kê vào báo cáo
  + Dễ thao tác và sử dụng

**2. Hướng phát triển của đề tài**

Vì kiến thức còn hạn hẹp nên ứng dụng còn thô sơ, trong thời gian sắp tới tôi sẽ cố gắng mở rộng ứng dụng để có thể áp dụng cho những doanh nghiệp lớn thay vì chỉ áp dụng cho mô hình gia đình như hiện tại.

TÀI LIỆU THAM KHẢO

1. Sách: Android Programming for Beginners
2. Sách: Head First Android Development
3. Sách: Beginning Android 4 Application Development
4. Sách: Học lập trình Google Flutter qua 65 ví dụ
5. Website: <https://hiepsiit.com/khoa-hoc/android/gioithieu/53/0>
6. Firebase: <https://firebase.google.com/docs>