**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM**

TRẦN HOÀNG MINH MSSV: N14DCCN116 Tên Đề tài: Lớp: D14CQCP01-N 2014-2019

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*: *“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”***

**Người hướng dẫn : TS.GVC VÕ XUÂN THỂ**

**Sinh viên thực hiện : TRẦN HOÀNG MINH**

**Mã số sinh viên : N14DCCN116**

**Lớp : D14CQCP01-N**

**Khoá** **: 2014**

**Hệ** **: Đại học chính quy**

TP.

HCM

2018

**TP.HCM, tháng 08/2018**

**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*: *“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”***

**Người hướng dẫn : TS.GVC VÕ XUÂN THỂ**

**Sinh viên thực hiện : TRẦN HOÀNG MINH**

**Mã số sinh viên : N14DCCN116**

**Lớp : D14CQCP01-N**

**Khoá** **: 2014**

**Hệ** **: Đại học chính quy**

**TP.HCM, tháng 08/2018**

# MỞ ĐẦU

Ngày nay, cùng với sự phát triển mạnh mẽ của các ứng dụng Android, nhưng vẫn chưa có nhiều dịch vụ cung cấp các sản phẩm đa phương tiện như trò chơi, hình, video, nhạc một cách đa dạng nào mà chỉ hỗ trợ một hoặc hai loại sản phẩm đa phương tiện. Ngoài ra, việc thanh toán, giao dịch vẫn chưa thật sự thuận tiện, dễ dàng đối với thị trường Việt Nam. Tại sao chúng ta không tạo ra một ứng dụng chuyên cung cấp các sản phẩm đa phương tiện mà các phương thức thanh toán, giao dịch một cách dễ dàng và nhanh chóng? Nơi chúng ta có thể dễ dàng tìm kiếm những ứng dụng trò chơi đa dạng, các loại hình ảnh phong phú đẹp mắt, những video hay kèm những bài hát tuyệt hảo. Chỉ với một thao tác thanh toán, chúng ta đã sỡ hữu ngay cho mình một sản phẩm đa phương tiện thích thú. Với một chiếc thẻ cào tiện lợi, chúng ta đã có thể nạp tiền một cách nhanh chóng và thật dễ dàng. Một kho các sản phẩm đa phương tiện phong phú và đa dạng đang chờ được khám phá.

Chính vì vậy, đề tài ***“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”*** là một ứng dụng được thiết kế và thực hiện nhằm cung cấp các sản phẩm đa phương tiện đến tận tay người dùng chỉ với một chiếc thẻ cào.

# LỜI CẢM ƠN

Để hoàn thành đồ án thực tập tốt nghiệp “***Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào***” này trước hết em xin gửi lời cảm ơn đến tập thể Ban Giám Hiệu, Khoa Công Nghệ Thông Tin, giảng viên, cán bộ các phòng ban chức năng Học Viện Công Nghệ Bưu Chính Viễn Thông tại TP.HCM. Em xin bày tỏ lòng cảm ơn sâu sắc chân thành về sự giúp đỡ đó.

Đặc biệt, em xin gởi đến thầy TS.GVC Võ Xuân Thể, người đã tận tình hướng dẫn, giúp đỡ em hoàn thành đồ án thực tập tốt nghiệp này lời cảm ơn sâu sắc nhất.

Sau cùng, em xin kính chúc quý Thầy Cô trong Khoa Công Nghệ Thông Tin và thầy TS.GVC Võ Xuân Thể thật dồi dào sức khỏe, niềm tin để tiếp tục sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Em xin trân trọng cảm ơn.

TP. Hồ Chí Minh, ngày 10 tháng 8 năm 2017

**Sinh viên thực hiện**

Trần Hoàng Minh

# MỤC LỤC

[MỞ ĐẦU 3](#_Toc520061689)

[LỜI CẢM ƠN 4](#_Toc520061690)

[MỤC LỤC 2](#_Toc520061691)

[DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH 4](#_Toc520061692)

[THUẬT NGỮ, CHỮ VIẾT TẮT 4](#_Toc520061693)

[ DVM: Dalvik Virtual Machine. 4](#_Toc520061694)

[ ART: Android RunTime. 4](#_Toc520061695)

[ JSON: JavaScript Object Notation. 4](#_Toc520061696)

[ SSD: Solid-State Drive - ổ cứng thể rắn. 4](#_Toc520061697)

[ CDN: Mạng lưới gồm nhiều máy chủ lưu trữ đặt tại nhiều vị trí địa lý khác nhau. 4](#_Toc520061698)

[ DDOS: Distributed Denial Of Service - tấn công từ chối dịch vụ phân tán. 4](#_Toc520061699)

[ MVC: Model View Controller. 4](#_Toc520061700)

[ MVP: Model View Presenter. 4](#_Toc520061701)

[ Mod: Moderator. 4](#_Toc520061702)

[ Admin: Administrator. 4](#_Toc520061703)

[DANH SÁCH HÌNH ẢNH 5](#_Toc520061704)

[CHƯƠNG I: TỔNG QUAN 6](#_Toc520061705)

[1.1 Bối cảnh chọn đề tài 6](#_Toc520061706)

[1.2 MỤC ĐÍCH VÀ Ý NGHĨA CỦA ĐỀ TÀI 6](#_Toc520061707)

[1.2.1 Mục đích của đề tài 6](#_Toc520061708)

[1.2.2 Ý nghĩa của đề tài: 7](#_Toc520061709)

[1.3 Mục tiêu và những yêu cầu khi xây dựng hệ thống 7](#_Toc520061710)

[1.3.1 Mục tiêu xây dựng hệ thống 7](#_Toc520061711)

[1.3.2 Những yêu cầu đặt ra cho hệ thống 7](#_Toc520061712)

[1.3.3 Hoạt động của hệ thống 8](#_Toc520061713)

[1.3.4 Hoạt động của người dùng 8](#_Toc520061714)

[1.3.5 Hoạt động của người cộng tác 8](#_Toc520061715)

[1.3.6 Hoạt động của người quản trị 8](#_Toc520061716)

[1.4 Phương pháp giải quyết các vấn đề 9](#_Toc520061717)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 10](#_Toc520061718)

[2.1 Về lý thuyết 10](#_Toc520061719)

[2.2 Về chương trình 10](#_Toc520061720)

[2.3 Phương pháp triển khai 11](#_Toc520061721)

[2.4 Cơ sở lý thuyết liên quan đến đề tài 11](#_Toc520061722)

[2.4.1 Lý thuyết về Android 11](#_Toc520061723)

[2.4.2 Lí thuyết về Firebase 18](#_Toc520061724)

[2.5 Các bước phát triển một ứng dụng Android 21](#_Toc520061725)

[2.5.1 Phát triển ứng dụng Android gốc 21](#_Toc520061726)

[2.5.2 Thách thức phải đối mặt đối với các nhà phát triển ứng dụng Android 21](#_Toc520061727)

[2.5.3 Xu hướng gần đây 22](#_Toc520061728)

[2.5.4 Xuất bản ứng dụng Android 23](#_Toc520061729)

[2.5.5 Kiếm tiền từ ứng dụng Android 23](#_Toc520061730)

[2.6 MVP Pattern cho Android 24](#_Toc520061731)

[2.6.1 Mô hình MVP là gì? 24](#_Toc520061732)

[2.6.2 Tại sao sử dụng MVP? 25](#_Toc520061733)

[2.6.3 Triển khai MVP trên Android 25](#_Toc520061734)

[2.7 Multi-Thread và Service trong android 26](#_Toc520061735)

[2.7.1 Multi-Thread 26](#_Toc520061736)

[2.7.2 Service 28](#_Toc520061737)

[CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 30](#_Toc520061738)

[3.1 Tổng quan ứng dụng 30](#_Toc520061740)

[3.1.1 Đối tượng sử dụng ứng dụng 30](#_Toc520061741)

[3.1.2 Các chức năng chính của hệ thống 30](#_Toc520061742)

[3.2 Thiết kế cơ sở dữ liệu 31](#_Toc520061743)

[3.1.1 Tổng quan cơ sở dữ liệu 31](#_Toc520061744)

[3.1.2 Chi tiết cơ sở dữ liệu 32](#_Toc520061745)

[CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG TRÊN ANDROID DỊCH VỤ CUNG CẤP CÁC SẢN PHẨM MULTIMEDIA QUA THẺ CÀO 37](#_Toc520061746)

[4.1 Giới thiệu các công nghệ sử dụng 37](#_Toc520061748)

[4.1.1 Các công nghệ sử dụng trong việc xây dựng ứng dụng 37](#_Toc520061749)

[4.1.2 Các bước thực hiện xây dựng ứng dụng 37](#_Toc520061750)

[4.2 Kết quả 37](#_Toc520061751)

[4.3 Chức năng chính trong ứng dụng 37](#_Toc520061752)

[4.3.1 Đăng kí (Chức năng #1): 37](#_Toc520061753)

[4.3.2 Đăng nhập (Chức năng #2): 39](#_Toc520061754)

[4.3.3 Hiển thị sản phẩm (Chức năng #3): 40](#_Toc520061755)

[4.3.4 Hiển thị chi tiết sản phẩm (Chức năng #4): 43](#_Toc520061756)

[4.3.5 Hiển thị hồ sơ người dùng (Chức năng #5): 45](#_Toc520061757)

[4.3.6 Hiển thị danh sách ưu thích (Chức năng #6): 46](#_Toc520061758)

[4.3.7 Upload, cập nhật sản phẩm (Chức năng #7): 47](#_Toc520061759)

[4.3.8 Nạp tiền (Chức năng #8): 47](#_Toc520061760)

[4.3.9 Thanh toán (Chức năng #9): 47](#_Toc520061761)

[4.3.10 Tìm kiếm (Chức năng #10): 47](#_Toc520061762)

[4.3.11 Hệ thống đánh giá, bình chọn (Chức năng #11): 47](#_Toc520061763)

[4.3.12 Các chức năng của người quản trị (Chức năng #12): 47](#_Toc520061764)

[CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 48](#_Toc520061765)

[5.1 Kết quả đạt được 48](#_Toc520061767)

[5.2 Hạn chế 48](#_Toc520061768)

[5.3 Hướng phát triển 48](#_Toc520061769)

[TÀI LIỆU THAM KHẢO 49](#_Toc520061770)

# DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

**KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT**

DVM: Dalvik Virtual Machine.

ART: Android RunTime.

JSON: JavaScript Object Notation.

SSD: Solid-State Drive - ổ cứng thể rắn.

CDN: Mạng lưới gồm nhiều máy chủ lưu trữ đặt tại nhiều vị trí địa lý khác nhau.

DDOS: Distributed Denial Of Service - tấn công từ chối dịch vụ phân tán.

MVC: Model View Controller.

MVP: Model View Presenter.

Mod: Moderator.

Admin: Administrator.

# DANH SÁCH HÌNH ẢNH

[Hình 1: Mô hình Client/Server 6](file:///C:\Users\huyco\AppData\Roaming\Skype\My%20Skype%20Received%20Files\BAOCAO_TRANLEANHTHE.docx#_Toc490522077)

[Hình 2: NodeJS 13](#_Toc490522078)

[Hình 3: Mô hình MVC 18](#_Toc490522079)

[Hình 4: Single Page App 19](#_Toc490522080)

[Hình 5: MongoDB 36](#_Toc490522081)

[Hình 6: Minh họa Bộ sưu tập 40](#_Toc490522082)

[Hình 7:Database 47](#_Toc490522083)

[Hình 8:Code mã hóa Password. 51](#_Toc490522084)

[Hình 9:Giao diện trang đăng ký. 52](#_Toc490522085)

[Hình 10:Kết quả trả về sau khi đăng nhập thành công. 52](#_Toc490522086)

[Hình 11:Giao diện trang đăng nhập. 53](#_Toc490522087)

[Hình 12:Giao diện trang chủ. 53](#_Toc490522088)

[Hình 13:Giao diện trang đội bóng. 54](#_Toc490522089)

[Hình 14:Giao diện trang tạo thách đấu 55](#_Toc490522090)

[Hình 15:Giao diện trang Team Profile(của Team khác) 56](#_Toc490522091)

[Hình 16: Giao diện trang Message 56](#_Toc490522092)

[Hình 17:Giao diện trang My Matches 57](#_Toc490522093)

[Hình 18:Giao diện trang Tạo Team 58](#_Toc490522094)

[Hình 19: Cài đặt NodeJS 58](#_Toc490522095)

[Hình 20: Chạy npm-install 59](#_Toc490522096)

[Hình 21: Chạy bower installl 59](#_Toc490522097)

[Hình 22: Chạy Gulp 60](#_Toc490522098)

[Hình 23: Chạy Gulp serve-dev 60](#_Toc490522099)

# CHƯƠNG I: TỔNG QUAN

## Bối cảnh chọn đề tài

Xã hội ngày càng phát triển, đời sống của con người trở nên thoải mái hơn, nhu cầu vui chơi, giải trí cũng ngày càng đa dạng và phong phú hơn. Nói đến giải trí, điện thoại và các thiết bị cầm tay được xem là phổ biến nhất, đặc biệt là thiết bị di động nền tảng Android, những trò chơi chiến lược, phối hợp hay những bản nhạc, video ưu thích hoặc đơn giản là những tấm hình đẹp sẽ giúp chúng ta giải stress. Tuy nhiên để có được những sản phẩm đa phương tiện bản quyền như vậy, thì việc thanh toán, nạp tiền không phải là điều dễ dàng cho tất cả mọi người, đặt biệt là những người không thật sự biết công nghệ. Tất cả những khó khăn trên đã dẫn đến việc hình thành một ứng dụng cung cấp các sản phẩm đa phương tiện bằng nạp thẻ cào nhằm giúp người dùng tiếp cận dễ dàng với các sản phẩm đa phương tiện bản quyền khi sử dụng những chiếc thẻ cào thanh toán đang phổ biến hiện nay.

Ứng dụng là hoàn toàn miễn phí, nhưng để chơi sở hữu được các sản phẩm đa phương tiện trong hệ thống thì chúng ta phải mua bản quyền của sản phẩm đó. Nếu chúng ta muốn sử dụng các sản phẩm đa phương tiện đã mua trên một thiết bị khác thì phải làm sao? Với ứng dụng này chuyện đó hết sức đơn giản bởi vì chúng ta chỉ việc đăng nhập tài khoản của mình vào máy đó là sẽ có thể tải các ứng dụng đã sỡ hữu.

Thông qua việc sử dụng ứng dụng này, chúng ta có thể dễ dàng lựa chọn những sản phẩm đa phương tiện bản quyền, nạp tiền bằng thẻ cào điện thoại một cách gần gũi, linh hoạt, theo dõi các sản phẩm ưu thích hoặc đóng góp đánh giá sản phẩm một cách tin dùng. Những ý tưởng này đã thúc đẩy việc hình thành nên ứng dụng cung cấp các sản phẩm đa phương tiện qua thẻ cào.

## MỤC ĐÍCH VÀ Ý NGHĨA CỦA ĐỀ TÀI

### Mục đích của đề tài

Mục đích của đề tài ***“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”*** được thực hiện dựa trên hai mục đích chính sau:

* Thứ nhất là nghiên cứu các công nghệ lập trình như: lập trình ứng dụng với Android.
* Thứ hai là phát triển ứng dụng có tính thực tiễn cao, có khả năng triển khai ứng dụng vào thực tế, giúp những người có nhu cầu giải trí đa phương tiện có thể thao tác dễ dàng và tiện lợi.

Em hi vọng dựa trên nền tảng lý thuyết đã được thầy cô truyền thụ lại, kết hợp với sự tìm hiểu công nghệ của cá nhân, em tin đồ án thực tập tốt nghiệp của em sẽ đạt được mục đích như mong đợi.

### Ý nghĩa của đề tài:

Trước hết, đề tài *“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”* là một ứng dụng được xây dựng như một đề tài thể hiện việc áp dụng những kiến thức quý báu đã được các thầy cô đã nhiệt tình truyền thụ lại cho chúng em. Và đặc biệt là sự theo dõi và quan tâm giúp đỡ của thầy TS.GVC Võ Xuân Thể trong suốt thời gian em thực hiện đề tài này.

Thứ hai là những kinh nghiệm quý báu mà em có được trong quá trình thực hiện đề tài sẽ là hành trang tuyệt vời giúp ích rất nhiều cho công việc sau này của mình.

Thứ ba, trong thời đại công nghệ thông tin phát triển như vũ bão ngày nay, công nghệ không ngừng được cải tiến, những công nghệ mới ra đời đòi hỏi người lập trình viên phải cố gắng nghiên cứu và tìm hiểu để áp dụng vào thực tiễn. Điều này cũng được thể hiện qua sự cố gắng của em khi quyết định thực hiện đề tài này với các công nghệ được coi là mới nhất, hay nhất và được dự báo sẽ phát triển mạnh mẽ trong tương lai. Do vậy, em rất hi vọng sẽ tạo nên một sản phẩm hoàn chỉnh nhất có thể để đáp lại sự nhiệt tình giúp đỡ từ thầy cô và cả tâm huyết của em khi quyết định thực hiện đề tài này.

Cuối cùng, em tin với ứng dụng này có thể đáp ứng được nhu cầu của người sử dụng cho việc giải trí đa phương tiện cũng như làm thỏa mãn thị trường ứng dụng di động đang phát triển mạnh mẽ hiện nay.

## Mục tiêu và những yêu cầu khi xây dựng hệ thống

### Mục tiêu xây dựng hệ thống

* Đáp ứng được nhu cầu cung cấp các sản phẩm đa phương tiện.
* Nạp tiền và thanh toán nhanh chóng, tiện lợi bằng thẻ cào.
* Quản lý người dùng và các công cụ quản trị.

### Những yêu cầu đặt ra cho hệ thống

* Người dùng (user):
  + Dễ dàng đăng kí, đăng nhập.
  + Phân loại sản phẩm đa phương tiện hợp lí.
  + Thanh toán nhanh chóng, nạp tiền tiện lợi bằng thẻ cào.
  + Quản lý tài khoản, quản lý sản phẩm đã mua.
  + Đánh giá, xếp hạng, lưu lại các sản phẩm quan tâm.
* Người cộng tác (mod):
* Thêm, sửa, ngưng hoạt động sản phẩm đa phương tiện do mình tạo.
* Người quản trị (admin):
  + Thêm, sửa, ngưng hoạt động tất cả sản phẩm đa phương tiện.
  + Thay đổi quyền, kích hoạt, ngừng kích hoạt tài khoản.
  + Thay đổi thông tin người dùng.

### Hoạt động của hệ thống

Trong hoạt động của hệ thống có 3 đối tượng và có các hoạt động chính:

* Hoạt động của người dùng.
* Hoạt động của người cộng tác.
* Hoạt động của người quản trị.

### Hoạt động của người dùng

Người dùng là đối tượng sử dụng hệ thống nhằm mục đích xem được những sản phẩm đa phương tiện, nạp tiền và thanh toán qua thẻ cào, đánh giá, xếp loại sản phẩm, quản lý tài khoản cá nhân.

### Hoạt động của người cộng tác

Người cộng tác là người được ủy quyền, cho phép đăng các sản phẩm đa phương tiện mới, chỉnh sửa và ngưng hoạt động các sản phẩm của mình. Đồng thời nhận được khoản tiền từ người dùng mua sản phẩm do chính người cộng tác tạo.

### Hoạt động của người quản trị

Ứng dụng có các chức năng của người quản trị:

*Thêm, sửa, ngưng hoạt động sản phẩm*: Người quản trị có thể tạo các sản phẩm, sửa và ngưng hoạt động của mọi sản phẩm.

*Thêm, sửa, ngưng hoạt động, thay đổi quyền người dùng*: Người quản trị cũng có thể tạo tài khoản, sửa, quyền người dùng và ngưng hoạt động tài khoản người dùng.

*Thêm, sửa, ngưng hoạt động thẻ cào*: Người quản trị còn có thể quản lý thẻ cào.

## Phương pháp giải quyết các vấn đề

Từ các yêu cầu, phân tích hoạt động đặt ra của đề tài, em tiến hàng phân tích và đưa ra phương hướng giải quyết theo các bước như sau:

* Tìm hiểu các yêu cầu cần thiết của một ứng dụng cung cấp sản phẩm đa phương tiện, từ đó phân tích các chức năng sẽ đưa vào trong ứng dụng.
* Tìm hiểu về nghiệp vụ thẻ cào bao gồm nạp tiền, thanh toán và quản lý thẻ cào.
* Tìm hiểu về các ngôn ngữ lập trình, môi trường, các dạng cơ sở dữ liệu từ đó đưa ra giải pháp phù hợp để tiến hành xây dựng.
* Xây dựng giao diện, cơ sở dữ liệu, xử lý để làm rõ thiết kế của ứng dụng.
* Tiến hành xây dựng ứng dụng theo thiết kế.
* Tiến hành kiểm tra và chạy thử ứng dụng.
* Thay đổi, bổ sung, khắc phục các sự cố, lỗi bảo mật, từ đó đưa ứng dụng vào sử dụng trong thực tiễn.

# CHƯƠNG II. CƠ SỞ LÝ THUYẾT

## Về lý thuyết

Đề tài ***“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”*** là một ứng dụng chạy trên điện thoại sử dụng hệ điều hành Android. Ứng dụng được xây dựng và chạy trên điện thoại được phát triển trên nền Android SDK (Software development kit). Dưới đây là những chức năng chính mà em dự định phát triển ở phiên bản đầu tiên của ứng dụng này:

* Đề tài ***“Phát triển ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào”*** trên nền tảng Android 4.1 Jelly Bean.
* Người dùng tải ứng dụng hoàn toàn miễn phí về điện thoại có sử dụng hệ điều hành Android từ 4.1 trở lên.
* Người dùng chạy ứng dụng có thể đăng nhập hoặc đăng kí để tạo tài khoản bằng Gmail hoặc tài khoản ứng dụng. Khi đã đăng nhập, người dùng có thể xem danh sách các sản phẩm đa phương tiện mà người quản lý đã cập nhật trong cơ sở dữ liệu. Người dùng cũng có thể mua các sản phẩm bằng cách nạp tiền vào ví tiền bằng thẻ cào để mua các sản phẩm. Sau khi người dùng đã mua, số tiền người dùng trả sẽ được cộng vào túi tiền của người quản lý. Người dùng sẽ có quyền tải các sản phẩm do mình mua. Thông tin thanh toán sẽ được chuyển về và lưu vào cơ sở dữ liệu thông qua cơ sở dữ liệu của Firebase. Ngoài ra, ứng dụng còn tích hợp một công cụ quản lý dành cho người quản trị để xử lý nghiệp vụ về tài khoản bao gồm người quản lý và người dùng, thẻ cào và các thông tin mà khách hàng gởi về từ ứng dụng Android, xử lý các yêu cầu của người dùng.
* Mục tiêu của đề tài này là:
  + Tìm hiểu nghiệp vụ nạp tiền, thanh toán và quản lý thẻ cào.
  + Tìm hiểu cách thức xây dựng một ứng dụng trên Android.
  + Xây dựng thành công ứng dụng trên Android dịch vụ cung cấp các sản phẩm Multimedia qua thẻ cào.
  + Tìm hiểu bộ công cụ Android Studio.
  + Tìm hiểu thao tác kết nối với Firebase và cách hoạt động của cơ sở dữ liệu thời gian thực.

## Về chương trình

Chương trình thực hiện được các yêu cầu đề ra của đề tài tốt nghiệp, ứng dụng đã phần nào xây dựng và đáp ứng được một số chức năng chính:

* Client:
  + Đăng ký tài khoản.
  + Nạp thẻ cào.
  + Quản lý tài khoản.
  + Tải các sản phẩm: nhạc, phim, game, video.
* Server:
  + Các chức năng bảo mật các tài khoản: quản lý và khách hàng.
  + Quản lý kho các sản phẩm Multimedia.
  + Quản lý các tài khoản thẻ cào.

## Phương pháp triển khai

Xây dựng hệ thống bao gồm ứng dụng chạy trên điện thoại và cơ sở dữ liệu để quản lý các thông tin cần rất nhiều thời gian, tâm huyết và cả công sức. Vì vậy trong phạm vi thực tập tốt nghiệp lần này, em quyết định thực hiện các phần cơ bản đến các phần nâng cao theo các bước sau:

Khảo sát lấy yêu cầu người sử dụng điện thoại về các ứng dụng cung cấp sản phẩm đa phương tiện trên mobile để xây dựng ý tưởng.

Tiếp đến, em sẽ nghiên cứu công nghệ phát triển ứng dụng chạy trên hệ điều hành Android. Xây dựng các module có thể cài đặt trên điện thoại chạy Android đáp ứng các yêu cầu cơ bản của hệ thống.

Sau khi hoàn thành phần ứng dụng chạy trên điện thoại, em sẽ nghiên cứu đến việc lấy dữ liệu từ server về sau đó đưa lên ứng dụng. Ở đây, em sẽ sử dụng công nghệ hiện đại do Google tạo ra đó là Firebase một cơ sở dữ liệu cho phép quản lý dữ liệu, trả về theo thời gian thực của ứng dụng.

Lựa chọn một mô hình thiết kế phù hợp cho Android đó là MVP (Model-View-Presenter) vì ứng dụng chạy theo đa luồng, nhận dữ liệu bất đồng bộ do đó mô hình MVP là hoàn toàn phù hợp với các sự kiện trả về.

Sau đây là danh sách các công nghệ sử dụng để xây dựng ứng dụng:

* Android SDK
* Firebase
* MVP structure pattern

## Cơ sở lý thuyết liên quan đến đề tài

### Lý thuyết về Android

#### Android là gì?

Android là một hệ điều hành 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 và sau này được chính Google mua lại vào năm 2005.

Chính mã nguồn mở của Android cùng với tính không ràng buộc nhiều đã cho phép các nhà phát triển thiết bị di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị.

Cho đến hiện tại, Google đã bán ra 200.000 thiết bị Android mỗi ngày và Play Store của họ vượt mức 90.000 ứng dụng. Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

#### Kỉ nguyên của Android

**Ứng dụng**: Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên, Android 1.0, được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake (bánh cupcake), Donut (bánh vòng), Eclair (bánh Eclair), Froyo (ya-ua lạnh), Gingerbread (bánh gừng), Honeycomb (mật ong), Ice Cream Sandwich (bánh kẹp kem), Jelly Bean (kẹo dẻo), Kitkat (bánh Kitkat) và phiên bản hiện tại Lollipop (kẹo mút).

Kỉ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán ra tại Mĩ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng như giao diện đặc sản của hệ điều hành này đã khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này.

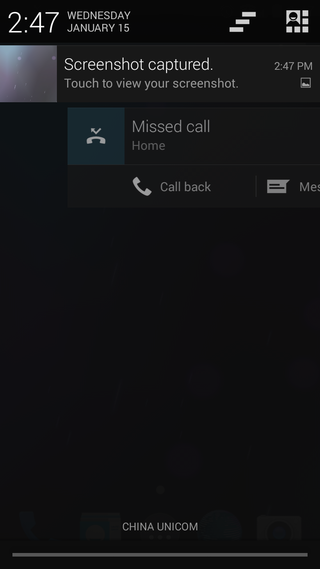
Ngay từ những ngày đầu tiên của Android, thanh thông báo đã đánh dấu một bước quan trọng mà trước đây chưa hề có hệ điều hành nào làm được - đưa tất cả thông tin tin nhắn, tin thoại hoặc các cuộc gọi nhỡ chỉ với thao tác vuốt xuống. Apple cũng đã học tập tính năng này và đưa vào hệ điều hành iOS của mình sau 3 năm kể từ khi phiên bản iOS đầu tiên ra mắt. Thanh notification này vẫn được Android sử dụng đến nay và đã cải tiến rất nhiều so với phiên bản đầu tiên.

Màn hình chính (Home Screen) và các widget: Một điểm khác biệt giữa Android so với các hệ điều hành khác là phần màn hình chính của mình. Bên cạnh việc thay đổi được hình nền, Android còn cho phép người dùng tùy biến màn hình chính của mình với nhiều widgets kèm theo, chẳng hạn như đồng hồ, lịch, trình nghe nhạc, đưa các icon ứng dụng ra ngoài hoặc thậm chí có thể can thiệp sâu hơn để thay đổi toàn bộ giao diện màn hình Home Screen này.

Đồng bộ và tích hợp chặt chẽ với Gmail: Vào thời điểm điện thoại G1 được bán ra, Gmail đã hỗ trợ giao thức POP và IMAP để tích hợp với các trình email trên di động. Tuy nhiên, lúc bấy giờ không có bất kì sản phẩm nào có thể hỗ trợ được hoàn toàn những tính năng ưu việt này của Gmail. Mãi cho đến khi Android 1.0 xuất hiện, vấn đề này đã được khắc phục và G1 trở thành chiếc điện thoại mang lại trải nghiệm Gmail tốt nhất trên thị trường lúc bấy giờ.

Kho ứng dụng Android Market: Thật khó có thể tưởng tượng một chiếc smartphone mà không hề có kho ứng dụng, nhưng vào thời điểm Android mới ra mắt, gần như không có bất kì điện thoại nào có kho ứng dụng nào được tích hợp và chính Android đã mở đầu cuộc cách mạng ứng dụng di động này. Android Market trên G1 thời bấy giờ có rất ít ứng dụng và giao diện cực kỳ đơn giản, hơn nữa tính năng mua ứng dụng trên phiên bản này vẫn chưa được xuất hiện mãi cho đến năm sau - những vấn đề này dễ hiểu vì thời điểm này Android chỉ mới được khai sinh nên mọi thứ còn khá thô sơ.

**Giao diện**: Google đã phát triển giao diện Android phiên bản 1.0 với sự hỗ trợ từ TAT, viết tắt từ The Astonishing Tribe, một công ty thiết kế tương tác của Thụy Điển. Dấu ấn rõ ràng nhất mà TAT để lại trên phiên bản Android từ phiên bản 1.0 cho đến 2.2 chính là widget đồng hồ kim nằm ngoài Home Screen tuy đơn giản nhưng rất đẹp mắt. Công ty này sau đó ngừng hợp tác với Google và bị RIM mua lại để tập trung phát triển sản phẩm Blackberry cũng như nền tảng BBX sau này.



Hình 1: Giao diện Android tiêu chuẩn

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK (Android application package) từ trang web khác. Các ứng dụng trên Cửa hàng Play cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Cửa hàng Play được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về, và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dùng. Đến tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Cửa hàng Play ước tính đạt 25 tỉ



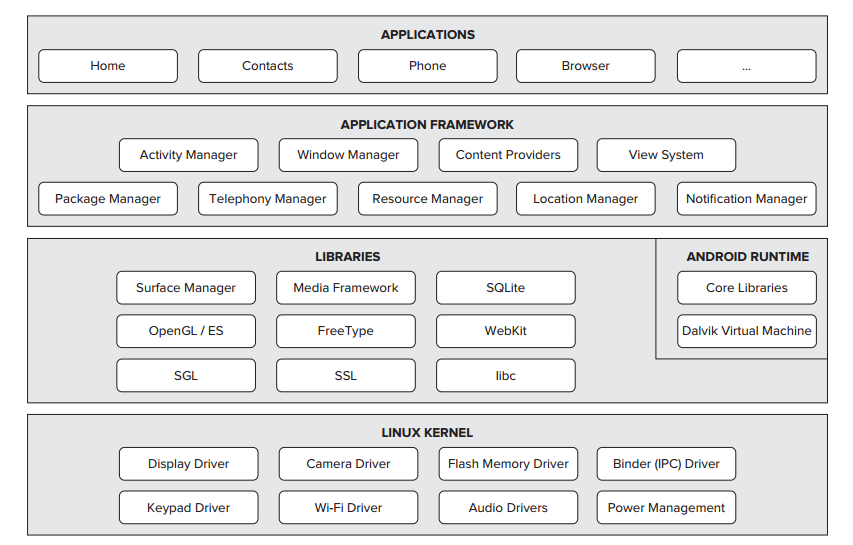
Hình 2: Biểu tượng Google play

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK - Software Development Kit). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên QEMU (generic and open source machine emulator and virtualizer), tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE - Integrated Development Environment) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốc dành cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.

#### Kiến trúc cơ bản của hệ điều hành Android

Android gồm 5 phần chính như sau và được chứa trong 4 lớp:

* Nhân Linux: Đây là nhân nền tảng mà hệ điều hành Android dựa vào nó để phát triển. Đây là lớp chứa tất cả các thiết bị giao tiếp ở mức thấp dùng để điều khiển các phần cứng khác trên thiết bị Android.
* Thư viện: Chứa tất cả các mã nguồn mở mà cung cấp những tính năng chính của hệ điều hành Android, đối với đề tài này thì Firebase là thư viện cung cấp việc hộ trợ làm việc với database dùng để chứa dữ liệu trên kho lưu trữ Google.
* Android runtime: Là tầng cùng với lớp thư viện Android runtime cung cấp một tập các thư viện cốt lõi để cho phép các lập trình viên phát triển viết ứng dụng bằng việc sử dụng ngôn ngữ lập trình Java. Android Runtime bao gốm máy ảo DVM (Dalvik Virtual Machine) (ở các version < 4.4, hiện tại là phiên bản máy ảo ART (Android RunTime) được cho là mạnh mẽ hơn trong việc xử lý biên dịch). Là cái để điều khiển mọi hoạt động của ứng dụng Android chạy trên nó (máy ảo Dalvik sẽ biên dịch ứng dụng để nó có thể chạy (thực thi) được, tương tự như các ứng dụng được biên dịch trên máy ảo Java vậy). Ngoài ra máy ảo còn giúp tối ưu năng lượng pin cũng như CPU của thiết bị Android.
* Android framework: Là phần thể hiện các khả năng khác nhau của Android (kết nối, thông báo, truy xuất dữ liệu) cho nhà phát triển ứng dụng, chúng có thể được tạo ra để sử dụng trong các ứng dụng của họ.
* Application: Tầng ứng dụng là tầng bạn có thể tìm thấy giao diện của các thiết bị Android như Contact, trình duyệt…Và mọi ứng dụng bạn viết đều nằm trên tầng này.



Hình 3: Kiến trúc hệ điều hành Android

#### Ưu điểm

*Mã nguồn mở:* Trong khi các hệ điều hành khác sử dụng mã nguồn đóng thì Android lại đi ngược lại bằng mã nguồn mở nên khả năng tùy biến cao, cung cấp cho người dùng tùy ý chỉnh sửa. Do đó, trước khi các nhà sản xuất cài đặt hệ điều hành Android trên thiết bị của mình thì họ đã bổ sung rất nhiều thứ từ giao diện, đến vô số những tính năng thông minh, tiện lợi khác

*Giá cả hợp lý từ bình dân đến cao cấp:* Đa dạng sản phẩm, rất nhiều hãng điện thoại lớn đã tung ra hàng loạt các sản phẩm đủ mọi kiểu dáng, kích cỡ với giá thành từ 1 đến 20 triệu đồng bạn đã có muôn vàn những lựa chọn từ bình dân vừa túi tiền đến những chiếc điện thoại cao cấp sở hữu nhiều tính năng hữu ích.

*Kho ứng dụng Google Play Store đồ sộ*: Hệ điều hành Android mang lại một kho ứng dụng Google Play đồ sộ với hơn 650.000 ứng dụng, trong đó có nhiều ứng dụng miễn phí mà vô cùng tiện dụng, chỉ sợ bạn không đủ bộ nhớ để cài mà thôi. Mặt khác, với tín đồ công nghệ, việc tự tạo ứng dụng trên Android chưa bao giờ dễ dàng đến thế dù việc này có thể làm cho Android tràn ngập ứng dụng “rác”.

*Dễ sử dụng:* Với 2 môi trường làm việc là màn hình chính (Home Screen) và menu chính. Home Screen giống như ngôi nhà gồm các mặt bức tường nơi bạn có thể thỏa sức thể hiện óc sáng tạo như trang trí đồng hồ, lịch, máy nghe nhạc, báo thức,... Còn Menu chính thì giống như tủ đồ cá nhân, nơi lưu trữ mục tin nhắn, điện thoại, nghe nhạc, máy ảnh,.... Tuy nhiên, đánh giá này cũng tùy thuộc quan điểm từng cá nhân, vì sẽ có nhiều người thấy rằng việc chỉ có 2 môi trường là hơi nhàm chán và quen thuộc.

#### Nhược điểm

*Khả năng dọn rác kém:* Sau một thời gian sử dụng thiết bị Android, các dữ liệu của ứng dụng đã được lưu trong bộ nhớ đệm, mà máy không có quá trình tự động để dọn dẹp chúng đi. Những dữ liệu này nếu không được dọn dẹp sẽ vẫn nằm ở đó và chiếm bộ nhớ của máy, đặc biệt là bộ nhớ RAM, vì vậy nếu sử dụng các thiết bị có bộ nhớ RAM thấp chúng ta sẽ thường xuyên gặp phải tình trạng giật, lag khi đang sử dụng.

*Quá nhiều phiên bản dẫn đến việc phân mảnh*: “Đem con bỏ chợ” là những gì mà đa phần người dùng than phiền Google khi sử dụng hệ điều hành Android phiên bản cũ và không được hỗ trợ nâng cấp lên phiên bản mới. Do tốc độ phát triển nhanh, kèm theo việc nâng cấp phiên bản thường xuyên dẫn đến việc có rất nhiều phiên bản Android trên thị trường. Thêm vào đó là rất nhiều hãng sử dụng hệ điều hành Android mà không thể nào hỗ trợ nâng cấp hết cho tất cả các thiết bị. Một phần cũng dễ hiểu vì họ luôn chạy theo và tập trung phát triển cho cái mới nên việc bỏ quên các thiết bị cũ là không hề tránh khỏi.

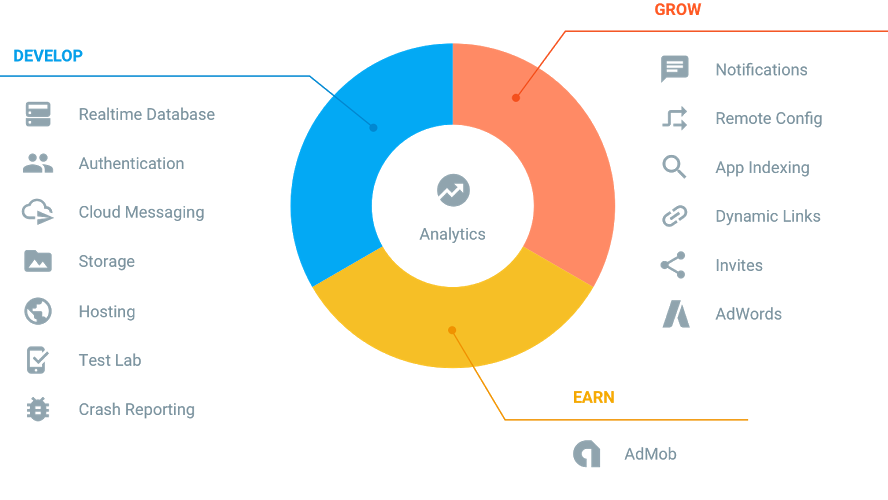
*Bảo mật kém*: Do đặc điểm là mã nguồn mở, nên đây là điều kiện để những người có ý đồ xấu tìm hiểu và khai thác lỗ hổng bảo mật trên thiết bị Android. Ví dụ: trước đây có một ứng dụng tên là iCalendar là một ứng dụng lịch báo thông thường với những icon và hiệu ứng lạ mắt, nhưng ẩn bên dưới là những đoạn mã độc dùng để “móc túi” người dùng, những đoạn mã độc này sẽ tự động gửi tin nhắn về tổng đài đã được định sẵn khi chạm 5 lần trên màn hình mà người dùng không hề hay biết. Sau khi vụ việc được phát hiện thì ứng dụng đó đã bị gỡ bỏ khỏi Google Play. Dần dần qua các phiên bản, Google cũng đã cải thiện khả năng bảo mật, nhưng “liệu có còn tồn tại những lỗ hổng nữa không?” thì vẫn chưa thể trả lời chính xác được.

*Kho ứng dụng quá nhiều dẫn đến khó khăn trong việc lựa chọn*:  Thật không dễ dàng để tìm kiểm một ứng dụng phù hợp trong kho lưu trữ khổng lồ Google Play Store. Khi chúng ta gõ một từ khóa tìm kiếm trên Play Store tiếp đó sẽ xuất hiện một danh sách thật dài các ứng dụng có cùng chức năng. Vì vậy việc lựa chọn không hề dễ dàng.

### Lí thuyết về Firebase

#### Firebase là gì

Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.



Hình 4: Các dịch vụ của Firebase

#### Các chức năng chính của Google Firebase

Với Google Firebase, bạn có thể tạo ra các ứng dụng chat như Yahoo Message của ngày xưa hoặc như Facebook Messenger của ngày nay trong thời gian cực ngắn như khoảng một ngày thậm chí là vài giờ bởi đơn giản là bạn chỉ cần lo phần client còn phần server và database đã có firebase lo. Firebase là sự kết hợp giữa nền tảng cloud với hệ thống máy chủ cực kì mạnh mẽ tới từ Google, để cung cấp cho chúng ta những API (Application Programming Interface) đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database. Cụ thể hơn Google Firebase cung cấp tới chúng ta những chức năng chính sau:

* *Real time Database* – Cơ sở dữ liệu thời gian thực: Firebase lưu trữ dữ liệu database dưới dạng JSON (JavaScript Object Notation) và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là bạn có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi. Ngoài ra Firebase còn cho phép bạn phân quyền một các đơn giản bằng cú pháp tương tự như JavaScript.
* *Firebase Authentication* – Hệ thống xác thực của Firebase: Với Firebase bạn có thể dễ dàng tích hợp các công nghệ xác thực của Google, Facebook, Twitter,… hoặc một hệ thống xác thực mà bạn tự mình tạo ra vào trong ứng dụng của bạn ở bất kì nền tảng nào như Android, iOS hoặc Web.
* *Firebase storage*: Với firebase storage các lập trình viên có thể lưu trữ dữ liệu trực tiếp trên server của firebase như hình ảnh, video, tập tin một cách dễ dàng. Firebase còn bổ sung Google security để tải lên và tải về các ứng dụng firebase của bạn. Bạn có thể sử dụng nó để lưu trữ hình ảnh, âm thanh, video hoặc nội dung do người dùng tạo ra. Quả thật đây là một dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và hiệu quả chi phí.
* *Firebase Hosting:* Firebase hosting cung cấp nhanh chóng và an toàn cho việc lưu trữ ứng dụng web của bạn. Với Hosting, bạn có thể nhanh chóng và dễ dàng triển khai các ứng dụng web và các nội dung tĩnh cho một mạng nội dung phân phối toàn cầu (CDN - mạng lưới gồm nhiều máy chủ lưu trữ đặt tại nhiều vị trí địa lý khác nhau) với một lệnh duy nhất.

#### Điểm nổi bật:

*Server với kết nối an toàn*: Cấu hình SSL được thêm vào Firebase Hosting nên nội dung luôn được phân phối một cách an toàn.

*Tốc độ cao*: Mỗi tập tin bạn tải lên sẽ được lưu trữ trên ổ SSD (Solid-State Drive - ổ cứng thể rắn) tại CDN. Chính vì vậy mà nội dung được cung cấp nhanh chóng.

*Triển khai nhanh chóng*: Sử dụng Firebase CLI (command line), bạn có thể có được ứng dụng và chạy trong vài giây. Chỉ với một vài lệnh command line để thêm vào ứng dụng của bạn.

*One-click rollbacks*: Triển khai nhanh chóng là rất tốt, nhưng có thể lùi lại sai lầm thậm chí còn tốt hơn. Firebase Hosting cung cấp versioning đầy đủ cho phép rollbacks một cú nhấp chuột.

#### Những lợi ích từ việc sử dụng Google Firebase

Ở phía trên là các chức năng của google firebase, vậy các chức năng đó sẽ đem lại cho bạn những lợi ích như sau:

*Giảm thời gian lập trình*: Với Firebase bạn có thể giảm bớt rất nhiều thời gian cho việc viết các dòng code để quản lý và đồng bộ cơ sở dữ liệu, mọi việc sẽ diễn ra hoàn toàn tự động với các API của Firebase. Không chỉ có vậy Firebase còn hỗ trợ đã nền tảng nên bạn sẽ càng đỡ mất thời gian rất nhiều khi ứng dụng bạn muốn xây dựng là ứng dụng đa nền tảng.

*Nhanh và đơn giản* Không chỉ nhanh chóng trong việc xây dựng database, Google Firebase còn giúp ta đơn giản hóa quá trình đăng kí và đăng nhập vào ứng dụng bằng các sử dụng hệ thống xác thực do chính Firebase cung cấp.

*Bảo mật*: Firebase hoạt động dựa trên nền tảng cloud và thực hiện kết nối thông qua giao thức bảo mật SSL, chính vì vậy bạn sẽ bớt lo lắng rất nhiều về việc bảo mật của dữ liệu cũng như đường truyền giữa client và server. Không chỉ có vậy, việc cho phép phân quyền người dùng database bằng cú pháp JavaScript cũng nâng cao hơn nhiều độ bảo mật cho ứng dụng của bạn, bởi chỉ những user mà bạn cho phép mới có thể có quyền chỉnh sửa cơ sở dữ liệu.

*Tính linh hoạt và khả năng mở rộng:* Sử dụng Firebase sẽ giúp bạn dễ dàng hơn rất nhiều mỗi khi cần nâng cấp hay mở rộng dịch vụ. Ngoài ra firebase còn cho phép bạn tự xây dựng server của riêng mình để bạn có thể thuận tiện hơn trong quá trình quản lý.

*Không ràng buộc:* Việc Firebase sử dụng NoSQL, giúp cho database của bạn sẽ không bị bó buộc trong các bảng và các trường mà bạn có thể tùy ý xây dựng database theo cấu trúc của riêng bạn.

*Sự ổn định*: Firebase hoạt động dựa trên nền tảng cloud đến từ Google vì vậy hầu như bạn không bao giờ phải lo lắng về việc sập server, tấn công mạng như DDOS (Distributed Denial Of Service - tấn công từ chối dịch vụ phân tán), tốc độ kết nối lúc nhanh lúc chậm, … nữa, bởi đơn giản là Firebase hoạt động trên hệ thống server của Google. Hơn nữa nhờ hoạt động trên nền tảng Cloud nên việc nâng cấp, bảo trì server cũng diễn ra rất đơn giản mà không cần phải dừng server để nâng cấp như truyền thống.

*Giá thành rẻ:* Google Firebase có rất nhiều gói dịch vụ với các mức dung lượng lưu trữ cũng như băng thông khác nhau với mức giá dao động từ Free đến $1500 đủ để đáp ứng được nhu cầu của tất cả các đối tượng. Chính vì vậy bạn có thể lựa chọn gói dịch vụ phù hợp nhất với nhu cầu của mình. Điều này giúp bạn tới ưu hóa được vốn đầu tư và vận hành của mình tùy theo số lượng người sử dụng. Ngoài ra bạn còn không mất chi phí để bảo trì, nâng cấp, khắc phục các sự cố bởi vì những điều này đã có Firebase lo.

## Các bước phát triển một ứng dụng Android

### Phát triển ứng dụng Android gốc

Có rất nhiều cách tiếp cận khác nhau để phát triển ứng dụng Android. Tuy nhiên, chúng ta sẽ chỉ tập trung vào phương pháp phát triển ứng dụng Android gốc bởi vì nó cung cấp hiệu suất vượt trội, chạy với yêu cầu tối thiểu, và sử dụng những khả năng tốt nhất của hệ điều hành Android.

Các ứng dụng Android gốc được phát triển bằng cách sử dụng ngôn ngữ lập trình Java, và trên các thiết bị chạy Android 5.0 hoặc cao hơn, tận dụng runtime được gọi là Android Runtime (ART) mặc định. Để tạo ra các ứng dụng như vậy, hầu hết các nhà phát triển chọn sử dụng Android Studio, Android IDE (Integrated Development Environment) chính thức được phát triển bởi Google. Android Studio miễn phí và chạy trên Windows, OS X, và hầu hết các bản phân phối Linux.

Android Studio 1.5.1, phiên bản ổn định hiện nay (tại thời điểm 2018), có một lượng lớn các tính năng để đơn giản hóa tất cả các khía cạnh của sự phát triển ứng dụng Android, chẳng hạn như một trình soạn thảo code mạnh mẽ, giao diện trực quan và trình biên tập theme, vector và tài nguyên hình ảnh, và nhiều hơn nữa. Android Studio 2.0, hiện vẫn là một phiên bản preview, có nhiều tính năng nâng cao như chạy tức thời và GPU profile để cải thiện hơn nữa trải nghiệm phát triển ứng dụng.

Android Studio cũng đi kèm với một emulator giống thiết bị thật mà bạn có thể sử dụng nếu bạn không sở hữu một thiết bị Android. Bởi vì nhân của nó phù hợp với tất cả phiên bản Android có sẵn nên bạn có thể tải về và chạy bất kỳ phiên bản nào của Android trên nó.

### Thách thức phải đối mặt đối với các nhà phát triển ứng dụng Android

Phát triển một ứng dụng Android chạy tốt trên một điện thoại hoặc máy tính bảng Android cụ thể là điều đơn giản. Tuy nhiên, việc phát triển một ứng dụng chạy tốt trên tất cả các thiết bị Android, có thể vô cùng khó khăn bởi những khác biệt lớn ở các phiên bản hệ điều hành, kích thước màn hình và khả năng của thiết bị.

*Hỗ trợ các phiên bản Android cũ hơn:* Không giống như các thiết bị iOS của Apple, hầu hết các thiết bị Android hiếm khi nhận được các bản cập nhật phần mềm, nếu không muốn nói là tất cả. Marshmallow, phiên bản mới nhất của Android, chạy trên không đến 1,5% tất cả các thiết bị Android ngày nay, ngay cả khi nó được phát hành vài tháng trước đây.

Điều này có nghĩa là, nếu bạn quyết định phát triển một ứng dụng, bạn phải hỗ trợ các phiên bản cũ hơn của Android, như KitKat và Lollipop. Hơn nữa, nếu người dùng chính của bạn là từ các nước đang phát triển, việc hỗ trợ các phiên bản cũ hơn như Android Gingerbread, là điều nên làm. Tuy nhiên, làm như vậy, thường là dễ dàng, nhờ không chỉ Android Support Library, mà còn có một số thư viện mã nguồn mở của bên thứ ba.

*Xử lý mật độ điểm ảnh và kích thước màn hình khác nhau*: Trong những ngày đầu của Android, chỉ có một số ít các thiết bị có thể chạy Android và hầu như tất cả chúng là điện thoại. Tuy nhiên, trong những năm qua, với ngày càng nhiều hơn các nhà sản xuất thiết bị áp dụng nền tảng, thì các kích cỡ màn hình và mật độ điểm ảnh bắt đầu trở nên khác nhau.

Với việc phát hành Android Honeycomb, Android cũng bắt đầu hỗ trợ các thiết bị màn hình lớn, chẳng hạn như máy tính bảng và smart TV. Do đó, các nhà phát triển ngày nay dự kiến sẽ sử dụng nhiều kích thước hình ảnh và bố cục để hỗ trợ tất cả các màn hình khác nhau.

Trong khi việc tạo ra các layout vẫn còn là một quá trình khá tốn thời gian, thì việc quản lý các kích thước hình ảnh khác nhau đã trở nên dễ dàng hơn nhờ vào Vector Asset Studio.

*Xử lý thiết bị cảm biến có sẵn*: Chỉ có một vài thiết bị cao cấp có tất cả các thiết bị cảm biến mà Android SDK hỗ trợ. Vì vậy, các nhà phát triển cần viết ứng dụng mà có thể xác định khả năng phần cứng của thiết bị lúc chạy và giảm bớt một cách nhẹ nhàng nếu có thể.

Android SDK (Software Development Kit) có một Sensor Framework để đơn giản hóa công việc này. Ngoài ra, các nhà phát triển có thể ngăn chặn người dùng cài đặt ứng dụng trên thiết bị thiếu phần cứng cần thiết bằng cách sử dụng bộ lọc của Google Play.

### Xu hướng gần đây

Nền tảng Android không ngừng phát triển và các nhà phát triển ứng dụng phải theo kịp với các xu hướng hiện nay để có thể xây dựng các ứng dụng đáp ứng sự mong đợi của người dùng.

*Material Design:* Material Design là một tập hợp các hướng dẫn thiết kế đã được hoạch định một cách tỉ mỉ mà bạn có thể làm theo để xây dựng các ứng dụng có giao diện phù hợp với giao diện gốc của Android Lollipop và Marshmallow. Do sự đơn giản và vẻ hiện đại của nó, Material Design được các nhà phát triển và các nhà thiết kế rất thích.

*Android Wear*: Những chiếc đồng hồ thông minh chạy Android Wear đang trở nên ngày càng phổ biến bởi vì các mẫu mới nhất trông tốt hơn và có tính năng mới và được cải tiến chẳng hạn như cử chỉ và mặt đồng hồ làm cho việc sử dụng chúng trở nên thú vị hơn. Do đó, ngày càng nhiều các nhà phát triển đang sử dụng Wear API trong ứng dụng của họ để đảm bảo rằng các ứng dụng có thể tương tác với thiết bị đeo này.

### Xuất bản ứng dụng Android

Một khi bạn đã phát triển xong một ứng dụng, bạn có thể xuất bản nó trực tuyến để cho mọi người khắp nơi trên thế giới có thể tải về và sử dụng nó. Có rất nhiều chợ ứng dụng Android mà bạn có thể xuất bản các ứng dụng của bạn ở trên đó, nhưng một trong những nơi phổ biến nhất hiện nay là Google Play bởi vì nó được cài đặt sẵn trên hầu hết thiết bị Android.

Để có thể xuất bản ứng dụng của bạn trên Google Play, bạn phải trả một khoản phí đăng ký 25$ và tạo một tài khoản nhà phát triển trên đó. Một khi bạn có một tài khoản, việc xuất bản một ứng dụng chủ yếu liên quan đến việc tải lên tập tin APK của nó- một gói tập tin nén mà bạn tạo ra bằng cách sử dụng Android Studio - cùng với một vài ảnh chụp màn hình và cung cấp các thông tin chi tiết, chẳng hạn như tên ứng dụng và mô tả.

Sau khi gửi tất cả các chi tiết, bạn sẽ phải chờ một vài giờ các ứng dụng được xuất bản. Vào đầu năm 2015, Google Play giới thiệu một quy trình phê duyệt ứng dụng để đảm bảo rằng các ứng dụng được xuất bản tuân thủ các chính sách nội dung của Google.

### Kiếm tiền từ ứng dụng Android

Việc phát triển ứng dụng Android là thú vị và không có gì hơn là thấy ứng dụng của bạn có mặt trên Google Play. Tuy nhiên, nếu bạn đã đặt rất nhiều nỗ lực để xây dựng nên ứng dụng, tại sao không kiếm tiền từ nó? Nhiều người ngày nay sống bằng cách phát triển ứng dụng Android, và nếu ứng dụng của bạn là đủ tốt, thì bạn cũng có thể.

Có rất nhiều cách bạn có thể kiếm tiền từ ứng dụng. Dưới đây là ba cách phổ biến, tất cả đều sử dụng dịch vụ do Google cung cấp:

* *Bán trên Google Play*: Ở hầu hết các nước, các nhà phát triển được cho phép xuất bản các ứng dụng premium trên Google Play. Không giống như các ứng dụng miễn phí, một ứng dụng premium chỉ có thể được tải về sau khi người dùng trả tiền cho nó. Đây là cách dễ nhất để kiếm tiền từ ứng dụng vì bạn không phải viết bất kỳ code nào cho nó.
* *Hiển thị quảng cáo*: Bằng cách sử dụng Mobile Ads SDK (Mobile Advertisement Software Development Kit) của Google, bạn có thể hiển thị biểu ngữ hoặc quảng cáo trong ứng dụng của bạn. Với các thiết lập này, bạn thường được trả mỗi khi người dùng nhìn thấy một quảng cáo hoặc nhấp vào.
* *Bán hàng bên trong ứng dụng*: Bằng cách sử dụng In-app Billing API, bạn có thể bán các mặt hàng kỹ thuật số đến người dùng đã sử dụng ứng dụng của bạn. Ví dụ, bạn có thể làm cho ứng dụng của bạn miễn phí và sau đó có thể tính một khoản phí nhỏ để mở khóa một tính năng đặc biệt trong nó. Loại ứng dụng này thường được gọi là các ứng dụng freemium.

## MVP Pattern cho Android

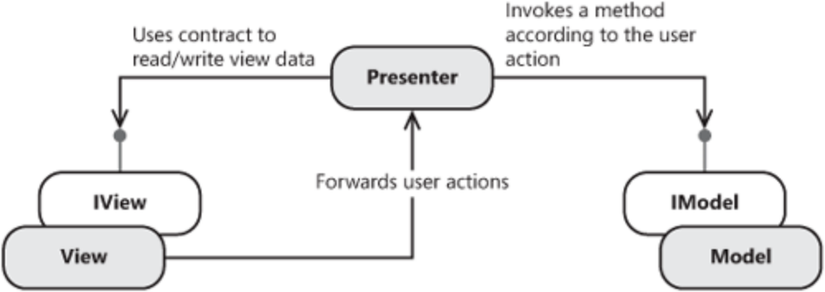
Mô hình MVP (Model View Presenter) là một dẫn xuất từ mô hình MVC (Model View Controller), hiện tại đang được áp dụng trong nhiều ứng dụng Android.

### Mô hình MVP là gì?

MVP là một mô hình kiến trúc hướng giao diện người dùng, được thiết kế để tạo thuận lợi cho việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC.

Mô hình MVP cho phép tách tầng trình diễn (Presenter) ra khỏi tầng dữ liệu (Model), vì vậy tương tác với giao diện được tách biệt với cách chúng ta biểu diễn nó trên màn hình (View), hay nói cách khác, tất cả logic khi người dùng tương tác được tách ra và đưa vào tầng trình diễn. Thiết kế lý tưởng nhất là với cùng một logic được áp dụng cho nhiều View khác nhau và hoán đổi được cho nhau. Trong đó:

* Model là một interface xác định cách mà dữ liệu được hiển thị trong giao diện người dùng.
* View là một giao diện người dùng thụ động hiển thị dữ liệu (Model) và tiếp nhận tương tác người dùng và truyền tới để Presenter xử lý tương tác.
* Presenter được ví như middle-man. Khi người dùng tương tác với View, Presenter tiếp nhận tương tác người dùng và cập nhật Model. Khi Model được cập nhật hay có thay đổi, Presenter lấy dữ liệu từ Model, định dạng và đưa tới View để hiển thị.



Hình 5: Mô hình Model - View - Presenter

**Điểm khác biệt dễ thấy nhất ở đây khi so sánh mô hình MVP với mô hình MVC chính là vị trí cũng như chức năng của tầng Presenter và View so với tầng Controller:**

**Trong mô hình MVP, tầng View là tầng duy nhất tiếp nhận tương tác người dùng thay vì cả 2 tầng View và tầng Controller trong mô hình MVC. Logic xử lý tương tác cũng như logic xử lý dữ liệu hiển thị được tách ra trong tầng Presenter thay vì được gộp chung với tiếp nhận tương tác trong Controller.**

### ****Tại sao sử dụng MVP?****

**Trong Android, có một vấn đề phát sinh từ thực tế là các cơ chế xử lý tương tác trong Android được kết hợp chặt chẽ giữa giao diện người dùng và xử lý, truy cập dữ liệu.**

**Một ví dụ điển hình là Cursor Adapter, đây là sự kết hợp của việc định dạng dữ liệu với xử lý tương tác giao diện. Đôi khi trong đó còn bao gồm cả xử lý dữ liệu ở mức sâu hơn (như tương tác với cơ sở dữ liệu) thông qua Cursor.**

**Việc kết hợp này có thể giúp giảm thiểu lượng code trong ứng dụng cũng như gộp các xử lý, logic trong ứng dụng vào cùng một nơi. Nhưng đối với một ứng dụng liên tục phát triển, hay một ứng dụng lớn, việc này khiến cho lượng code trên mỗi logic trở nên rất lớn, các logic xen lần, chồng chéo lên nhau, rất khó cho việc đọc hiểu, bảo trì cũng như mở rộng.**

**Phân tầng, phân lớp ứng dụng có thể giảm đi hiệu năng do ứng dụng cần thêm nhiều tài nguyên, nhưng tính khả chuyển của ứng dụng cũng tăng gấp nhiều lần., không những thế còn mở rộng khả năng cho phép kiểm thử từng phần của ứng dụng trở nên dễ dàng hơn. Chính vì thế cân nhắc giữa hiệu năng và tính dễ dàng mở rộng, bảo trì là hết sức quan trọng.**

**Trong Android, kiểm thử là một vấn đề khó khăn vì mối liên kết chặt chẽ giữa giao diện, logic và dữ liệu. MVP tách biệt giao diện khỏi dữ liệu, chia ứng dụng ra thành ít nhất ba lớp khác nhau, từ đó có thể kiểm thử một các độc lập. Với MVP ta có thể kiểm thử đo đạc một cách tối đa các logic của ứng dụng.**

### ****Triển khai MVP trên Android****

Có rất nhiều biến thể cũng như phương pháp triển khai MVP, tất cả mọi người có thể điều chỉnh mô hình này tùy theo nhu cầu và cách họ cảm thấy thoải mái hơn. Các mô hình này, về cơ bản khác nhau ở số lượng chức năng mà tầng Presenter đảm nhận.

Một view nhận tương tách từ người dùng disable hoặc enable progress bar, liệu có nên giao nhiệm vụ này cho một presenter? Một Activity sẽ nhận sự kiện click vào nút settings trên Action Bar hay sự kiện này sẽ được một đối tượng presenter trong Activity đảm nhận?

Những câu hỏi như vậy tạo ra nhiều cách để chúng ta triển khai mô hình MVP, và trên thực tế chưa có một tiêu chuẩn chính xác nào được đưa ra.

#### Tầng trình diễn - Presenter

Tầng trình diễn có trách nhiệm như một middle-man giữa View và Model. Nó lấy dữ liệu từ Model, định dạng và trả về cho View. Nhưng không giống như MVC, nó cũng quyết định những gì sẽ xảy ra khi người dùng tương tác với View, hay nói cách khác nó hàm chứa logic ứng dụng.

#### Tầng logic dữ liệu - Model

Trong một ứng dụng với thiết kế kiến trúc tốt, mô hình này sẽ chỉ là một gateway giữa tầng domain và tầng business logic. Trong mô hình Clean Architecture của Uncle Bob, Model sẽ là một interactor thực thi một use case. Để đơn giản, ở đây Model đơn thuần được nhìn nhận như một data source - cung cấp dữ liệu chúng ta muốn hiển thị trong giao diện ứng dụng.

#### Tầng giao diện - View

View, thường được implement bởi một Activity (hoặc có thể là một Fragment, một View ... tùy thuộc vào cấu trúc ứng dụng), Activity này sẽ chứa một thuộc tính là một lớp Presenter. Lý tưởng nhất Presenter nên được cung cấp bởi một Dependency Injection framework như Dagger, nhưng trong trường hợp ứng dụng không sử dụng một thư viện hay framework như vậy ta hoàn toàn có thể tạo ra các đối tượng Presenter này.

## ****Multi-Thread và Service trong android****

### ****Multi-Thread****

#### ****Thread là gì?****

**Thread là một luồng của một chương trình đang chạy. Một chương trình đang chạy thường sẽ có 1 luồng chính chủ đạo và có thể tạo ra được nhiều các luồng khác chạy xung quanh để phục vụ cho luồng chính đó. Các luồng này có thể chạy song song với nhau luồng đều có call stack riêng với các phương thức được gọi, tham số và biến địa phương của chúng.**

#### ****Cách tạo 1 thread?****

Chúng ta có 2 cách chính để tạo 1 thread: Tạo một lớp thread mới kế thừa từ lớp Thread. Ví dụ:

public class NewThread extends Thread{

public void run() {

print(“Hello”)

}

}

Tạo một biến mới của lớp Thread với đối tượng Runnable

Thread background = new Thread(new Runnable() {

public void run() {

print(“Hello”)

}

}

Và để chạy bất kỳ một thread nào ta đều phải gọi phương thức start();

background.start()

#### Tác dụng của thread trong 1 chương trình

* Các luồng có thể dùng chung tài nguyên của tiến trình để thực hiện các thao tác cần thiết.
* Có thể tách các nhiệm vụ của ứng dụng ra thành các công việc nhỏ cho từng luồng xử lý.
* Sử dụng luồng ta có thể thực thi song song các công việc 1 lúc một cách thực sự.
* Sử dụng thread sẽ làm cho chương trình chạy một cách trơn chu hơn, nhanh hơn chứ không phải đợi chờ lâu.
* Bên cạnh đó ta cũng gặp một số vấn đề về việc xảy ra deadlock khi có nhiều luồng cùng sử dụng 1 tài nguyên cố định. Hoặc thay đổi tài nguyên đó một cách không có kiểm soát sẽ dẫn đến sai lệch kết quả.

#### Liên kết giữa thread chính với thread phụ

Trong android, các thread tương tác với nhau thông qua các đối tượng được gọi là Handler. Khi một tiến trình được tạo cho một ứng dụng, luồng chính của nó được dành riêng để chạy một message queue, queue này quản lý các đối tượng bậc cao của ứng dụng (activity, intent receiver,…) và các cửa sổ mà chúng tạo ra.

Khi ta tạo một Handler mới, nó được gắn với message queue của thread tạo ra nó – từ đó trở đi, nó sẽ gửi các message và các runnable tới message queue đó và thực thi chúng khi chúng ra khỏi message queue.

Nhiệm vụ chính của Handler là: xếp lịch cho các message và runnable cần được thực thi vào thời điểm nào đó trong tương tai, và xếp hàng một action cần thực hiện tại một thread khác Mỗi luồng phụ cần liên lạc với luồng chính thì phải yêu cầu một message token bằng cách dùng phương thức obtainMessage().

Khi đã lấy được token, thread phụ đó dùng phương thức sendMessage() để ghi dữ liệu vào token và gắn nó vào message queue của Handler.

Ví dụ:

Handler handler = new Handler();

String data = “Messages 1”;

Message msg = handler.obtainMessage(1, (String)data);

handler.sendMessage(msg);

Handler liên tục xử lý các message mới được gửi tới thread chính bằng phương thức handleMessage(). Mỗi message được lấy ra từ queue của thread có thể trả về dữ liệu cho thread chính hoặc yêu cầu chạy các đối tượng runnable qua phương thức post().

### ****Service****

#### ****Service là gì?****

Service là một thành phần ứng dụng chạy ở chế độ nền, không tương tác với người sử dụng, trong một thời gian không xác định thời gian.

Điều này có nghĩa rằng, service của bạn sẽ làm bất cứ CPU chuyên sâu (như chơi nhạc MP3) hoặc ngăn chặn (như mạng) hoạt động, nó sẽ sinh ra tiến trình của mình, trong đó để làm công việc đó.

Mỗi Service phải có một khai báo tương ứng trong AndroidManifest.xml của gói của nó.

Dịch vụ có thể được bắt đầu với Context.startService () và Context.bindService ().

Service có thể được khởi tạo nhiều lần Context.startService () nhưng sẽ được ngừng lại chỉ với một lần gọi Context.stopService () hoặc stopSelf ().

Một Service có thể được bắt đầu và cho phép chạy cho đến khi một người nào đó hoặc chính nó tự dừng lại.

Chỉ có một stopService () gọi là cần thiết để dừng dịch vụ, không có vấn đề bao nhiêu lần startService () được gọi.

#### Các tạo và khởi chạy service

Giả sử hoạt động chính MyService muốn tương tác với một dịch vụ gọi là MyService1. Các hoạt động chính là chịu trách nhiệm về các công việc sau:

* Khởi tạo và start MyService1:

Intent intentMyService = new Intent(this, MyService1.class);

Service myService = startService(intentMyService);

* Định nghĩa IntentFilter và đăng ký BroadcastReceiver:

IntentFilter mainFilter = new IntentFilter("matos.action.GOSERVICE3");

BroadcastReceiver receiver = new MyMainLocalReceiver();

registerReceiver(receiver, mainFilter);

* Implement và override phương thức onReceive

public void onReceive(Context localContext, Intent callerIntent)

* Dừng hoạt động của Service:

stopService(new Intent(intentMyService) );

* Sử dụng phương pháp onDestroy của dịch vụ để đảm bảo rằng tất cả các hoạt động của nó chạy được chấm dứt và hủy đăng ký dịch vụ

myService.onDestroy()

unregisterReceiver(receiver);

# CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



## Tổng quan ứng dụng

### Đối tượng sử dụng ứng dụng

#### Người dùng (User)

Người dùng là đối tượng sử dụng ứng dụng để thực hiện các nhu cầu xem các sản phẩm đa phương tiện, nạp tiền và thanh toán. Ngoài ra người dùng có thể đóng góp thông qua chức năng đánh giá và xếp hạng.

#### Người cộng tác (Mod)

Đối với người cộng tác sẽ có chức năng đăng các sản phẩm đa phương tiện của họ và nhận tiền thông qua việc người dùng mua các ứng dụng đó.

#### Người quản lí (Admin)

Người quản lí là người chịu trách nhiệm với toàn bộ hệ thống. Toàn quyền với người dùng và người cộng tác.

### Các chức năng chính của hệ thống

#### Đối với người dùng

Ứng dụng cần có các chức năng để hỗ trợ và giải quyết các nhu cầu của người dùng như sau:

* Hiển thị các sản phẩm đa phương tiện.
* Đăng ký, đăng nhập qua Gmail hoặc tài khoản thường.
* Nạp tiền qua các loại card và nhiều mệnh giá.
* Mua hàng bằng chức năng một chạm.
* Đánh giá, xếp hạng với các sản phẩm.
* Lưu lại các sản phẩm hứng thú.
* Xem, sửa hồ sơ tài khoản
* Thống kê các sản phẩm đã mua

#### Đối với người cộng tác

Ứng dụng cần có các chức năng để hỗ trợ việc quản lý sản phẩm của người cộng tác như sau:

* Thêm, sửa, ngưng hoạt động đối với sản phẩm do mình tạo.
* Các chức năng khác giống với người dùng.

#### Đối với người quản trị

Ứng dụng sẽ có các chức năng để hỗ trợ quản lí toàn quyền của người quản trị như sau:

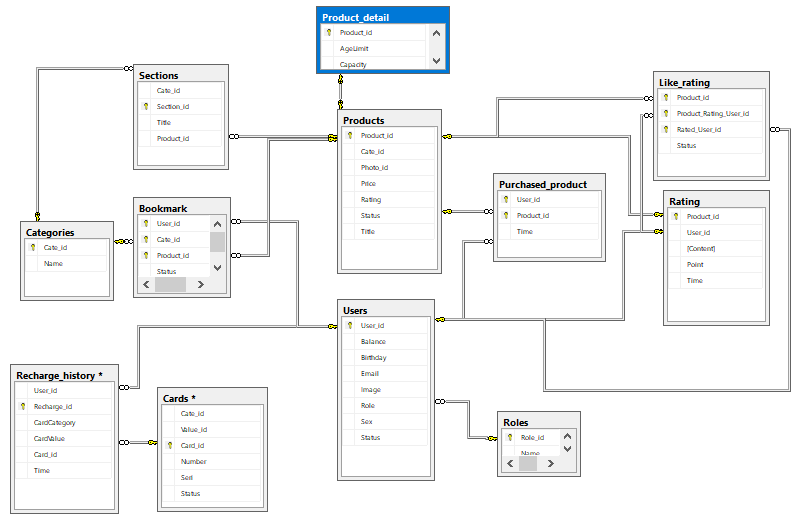
* Toàn quyền quản lí (thêm, sửa, ngưng hoạt động) với tất cả sản phẩm đa phương tiện.
* Toàn quyền quản lí (thêm, sửa, ngưng hoạt động) với tất cả tài khoản người dùng.
* Toàn quyền quản lí (thêm, sửa, ngưng hoạt động) với tất cả thẻ cào.

## Thiết kế cơ sở dữ liệu

### Tổng quan cơ sở dữ liệu

Từ những phân tích tổng quán của ứng dụng, đưa đến việc phát triển cơ sở dữ liệu nhằm lưu trữ và truy xuất dữ liệu.

Hệ quản trị cơ sở dữ liệu sử dụng: Firebase Database Real time



Hình 6: Diagram database của đồ án

### Chi tiết cơ sở dữ liệu

Danh sách các thực thể có trong cơ sở dữ liệu

* *Bookmark*: Lưu thông tin ứng dụng được đánh dấu bởi người dùng.
* *Cards*: Thông tin thẻ cào.
* *Categories*: Thông tin loại sản phẩm bao gồm:
  + Trò chơi
  + Hình ảnh
  + Video
  + Âm thanh
* *Like\_rating*: Lưu thông tin người dùng thích các đánh giá.
* *Products:* Lưu thông tin sản phẩm.
* *Product\_detail:* Lưu chi tiết sản phẩm.
* *Purchased\_product:* Lưu các sản phẩm đã được mua bởi người dùng.
* *Rating:* Lưu các bài đánh giá.
* *Recharge\_history:* Lưu lịch sử nạp tiền của người dùng
* *Roles:* Nơi các quyền của ứng dụng được lưu bao gồm:
  + Admin
  + Mod
  + User
* *Sections:* Lưu các thể loại của sản phẩm.
* *Users:* Lưu thông tin người dùng

#### Thực thể Bookmark lưu trữ thông tin:

Thực thể bookmark chứa thông tin các sản phẩm được lưu bởi người dùng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | User\_id | Mã người dùng |
| 2 | Cate\_id | Mã loại sản phẩm |
| 3 | Product\_id | Mã sản phẩm |
| 4 | Status | Trạng thái lưu (0: đã xóa, 1: đã lưu) |

*Bảng 4.1. Bảng mô tả thuộc tính của thực thể Bookmark*

#### Thực thể Cards lưu trữ thông tin:

Thực thể Cards chứa các thông tin của thẻ cào trong ứng dụng, nhằm giả lập việc nạp tiền và thanh toán đối với các loại thẻ.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Cate\_id | Mã loại thẻ cào (Viettel, Vina, Mobi, Vietnamobile, Garena) |
| 2 | Value\_id | Mã mệnh giá thẻ (10, 20, 50, 100, 200, 500) |
| 3 | Card\_id | Mã thẻ |
| 4 | Number | Mã nạp tiền |
| 5 | Seri | Mã seri |
| 6 | Status | Trạng thái thẻ (0: chưa nạp, 1: đã nạp) |

*Bảng 4.2. Bảng mô tả thuộc tính của thực thể Cards*

#### Categories

Thực thể categories chứa thông tin của loại sản phẩm

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Cate\_id | Mã loại sản phẩm |
| 2 | Name | Tên loại sản phẩm (Game, Image, Video, Music) |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Categories*

#### Like\_rating

Thực thể Like\_rating chứa các thông tin của người dùng đã thích các bài đánh giá.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Product\_id | Mã sản phẩm |
| 2 | Product\_Rating\_User\_id | Mã người dùng bài đánh giá |
| 3 | Rated\_User\_id | Mã người dùng thích bài |
| 4 | Status | Trạng thái |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Like\_rating*

#### Products

Thực thể products chứa các thông tin của sản phẩm đa phương tiện.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Product\_id | Mã sản phẩm |
| 2 | Cate\_id | Mã loại sản phẩm |
| 3 | Photo\_id | Tên ảnh đại diện |
| 4 | Price | Giá sản phẩm |
| 5 | Rating | Điểm đánh giá (0 đến 5) |
| 6 | Status | Trạng thái sản phẩm |
| 7 | Title | Tiêu đề sản phẩm |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Products*

#### Product\_detail

Thực thể product\_detail chứa các thông tin chi tiết của sản phẩm đa phương tiện.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Product\_id | Mã sản phẩm |
| 2 | AgeLimit | Tuổi giới hạn |
| 3 | Capacity | Kích thước (theo KB) |
| 4 | Description | Mô tả |
| 5 | Downloaded | Số lượt tải |
| 6 | ImageList | Danh sách hình ảnh phụ |
| 7 | Intro | Giới thiệu |
| 8 | Owner\_id | Mã người sở hữu |
| 9 | Video | Link youtube |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Product\_Detail*

#### Purchased\_product

Thực thể purchased\_product chứa các thông tin của sản phẩm đã được mua bởi người dùng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | User\_id | Mã người dùng |
| 2 | Product\_id | Mã sản phẩm |
| 3 | Time | Thời gian |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Purchased\_product*

#### Rating

Thực thể Rating chứa các thông tin của những bài đánh giá bởi người dùng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Product\_id | Mã sản phẩm |
| 2 | User\_id | Mã người dùng |
| 3 | Content | Nội dung |
| 4 | Point | Điểm đánh giá (0 đến 5) |
| 5 | Time | Thời gian |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Rating*

#### Recharge\_history

Thực thể Recharge\_history chứa các thông tin của những lịch sử nạp tiền của người dùng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | User\_id | Mã người dùng |
| 2 | Recharge\_id | Mã giao dịch |
| 3 | CardCategory | Mã loại thẻ cào |
| 4 | CardValue | Mã mệnh giá thẻ |
| 5 | Card\_id | Mã thẻ |
| 6 | Time | Thời gian |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Recharge\_history*

#### Roles

Thực thể Roles chứa các thông tin của quyền trong ứng dụng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Role\_id | Mã quyền |
| 2 | Name | Tên quyền (Admin, Mod, User) |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Roles*

#### Sections

Thực thể Section chứa các thông tin phân loại của sản phẩm.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Cate\_id | Mã loại sản phẩm |
| 2 | Section\_id | Mã nhóm sản phẩm |
| 3 | Title | Tiêu đề |
| 4 | Product\_id | Danh sách mã sản phẩm |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Sections*

#### Users

Thực thể Users chứa các thông tin chi tiết của người dùng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | User\_id | Mã người dùng |
| 2 | Balance | Số tiền trong tài khoản |
| 3 | Birthday | Ngày sinh |
| 4 | Email | Địa chỉ email |
| 5 | Image | Tên hình đại diện |
| 6 | Name | Tên người dùng |
| 7 | Role | Mã quyền |
| 8 | Sex | Mã giới tính (0: Nam, 1: Nữ, 2: Không xác đinh) |
| 9 | Status | Trạng thái (0: Ngưng kích hoạt, 1: Đang kích hoạt) |

*Bảng 4.4. Bảng mô tả thuộc tính của thực thể Users*

# CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG TRÊN ANDROID DỊCH VỤ CUNG CẤP CÁC SẢN PHẨM MULTIMEDIA QUA THẺ CÀO



## Giới thiệu các công nghệ sử dụng

### Các công nghệ sử dụng trong việc xây dựng ứng dụng

* Môi trường: Android studio.
* Cơ sở dữ liệu, xác thực người dùng: Firebase
* Cấu trúc: MVP structure pattern

### Các bước thực hiện xây dựng ứng dụng

*Bước 1*: Phân tích, thiết kế, mô tả các chức năng cần thiết của ứng dụng.

*Bước 2*: Phân tích, thiết kế cơ sở dữ liệu để phù hợp với các chức năng và thực hiện hiệu quả việc truy vấn dữ liệu.

*Bước 3*: Phân tích, thiết kế, lên ý tưởng giao diện ứng dụng.

*Bước 4*: Phân tích, lựa chọn loại design pattern là MVP

*Bước 5*: Xây dựng quá trình thao tác dữ liệu giữa ứng dụng và cơ sở dữ liệu đồng thời hiển thị cho người dùng

*Bước 6*: Xây dựng, thiết kế bảo mật.

*Bước 7*: Chạy thử ứng dụng, sửa lỗi, thêm hiệu ứng, làm đẹp ứng dụng.

## Kết quả

Sau khi thực hiện tuần tự các bước như đã đặt ra, ứng dụng đã hoạt động và cơ bản đảm bảo các chức năng yêu cầu đặt ra ban đầu, bên cạnh đó cũng có một số chức năng cải tiến mới và đảm bảo thuận lợi nhất cho người sử dụng khi sử dụng ứng dụng.

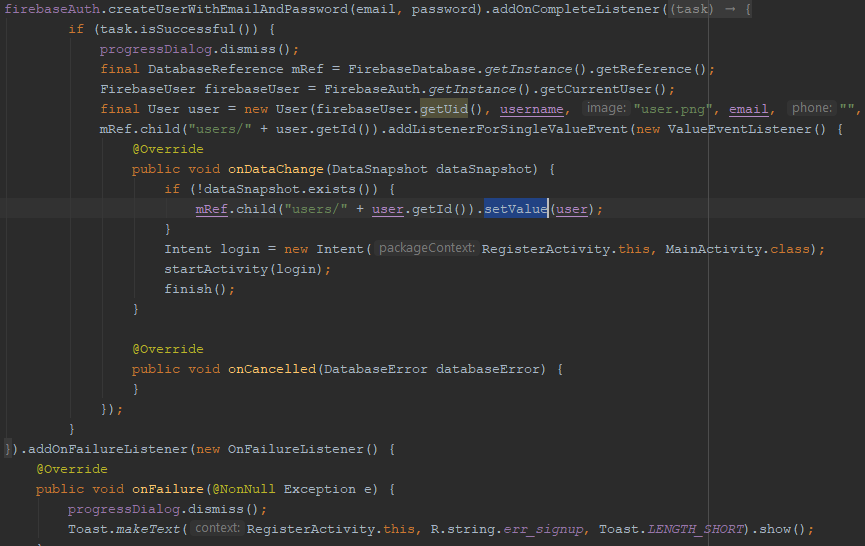
## Chức năng chính trong ứng dụng

### Đăng kí (Chức năng #1):

* Giao diện đăng kí:
* Yêu cầu chức năng:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Đăng kí | Đăng kí tài khoản Local của ứng dụng | Thiết kế  Xử lý | Hoàn thành |

* Hướng giải quyết: Sử dụng Authentication của Firebase Authentication để quản lí thông tin đăng kí người dùng, chúng ta có hai cách đăng kí:
  + Tạo tài khoản bằng sử dụng tài khoản Google.
  + Tạo tài khoản bằng username và password tự nhập.
* Cách thức làm:
  + Màn hình hiển thị các Edit Text cho biết những thông tin mà người dùng cần cung cấp.
  + Nếu đăng kí thành công thì sẽ chuyển sang MainActivity.
* Phần cài đặt:



Hình 7: Hình mã nguồn đăng kí

### Đăng nhập (Chức năng #2):

* Giao diện đăng nhập
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Đăng nhập thông thường | Sử dụng tài khoản  Local của ứng dụng | Xử lý | Hoàn thành |
| Đăng nhập Google | Sử dụng bằng tài khoản google để đăng nhập | Xử lý | Hoàn thành |

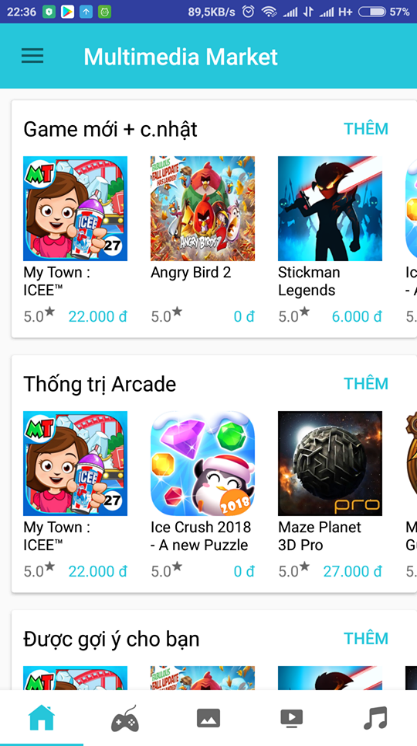
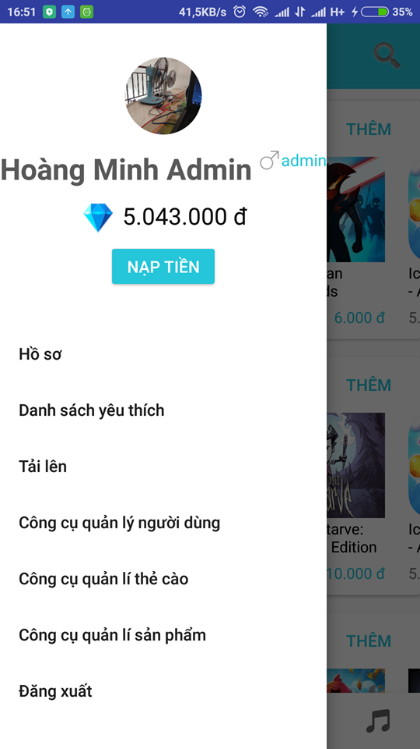
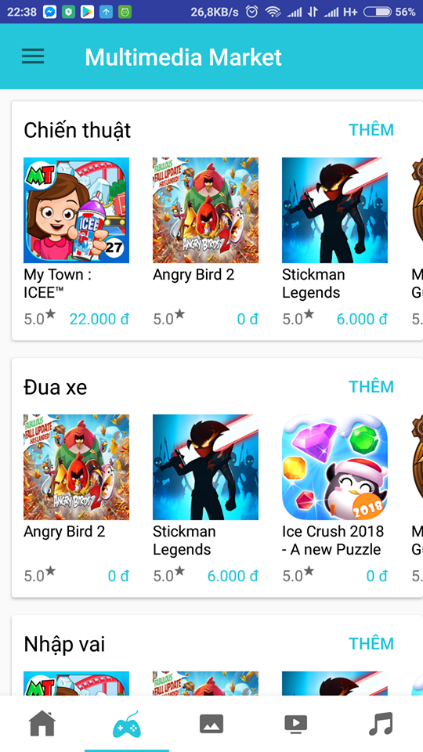
* Hướng giải quyết:
  + Chúng ta cũng sẽ sử dụng Firebase Authentication để quản lí đăng nhập.
  + Có hai cách đăng nhập:
    - Sử dụng tài khoản Google để đăng nhập.
    - Sử dụng tài khoản local của ứng dụng để đăng nhập.
  + Nếu là lần đầu tiên đăng nhập thì thông tin cá nhân sẽ được tạo trong bảng User. Đồng thời, tài khoản đăng nhập cũng sẽ được kiểm tra nếu trạng thái cho phép hoạt động thì mới được tiếp tục.
* Cách thức làm:
  + Màn hình đăng nhập dùng để hiển thị khi người dùng cần đăng nhập.
  + Button:
    - Chuyển sang MainActivity khi đăng nhập thành công.
    - Chuyển sang RegisterActivity khi bấm vào Button “đăng kí”.
* Phần Cài đặt:



Hình 8: Hình mã nguồn đăng nhập

### Hiển thị sản phẩm (Chức năng #3):

* Giao diện sản phẩm

*Hình 9: Giao diện màn hình chính*

* Yêu cầu chức năng

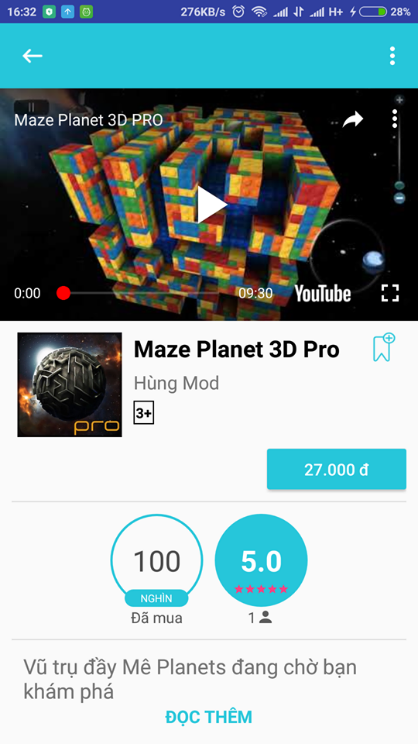
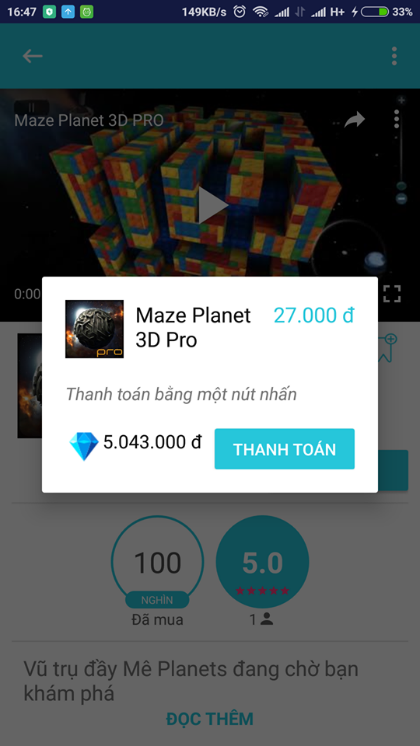
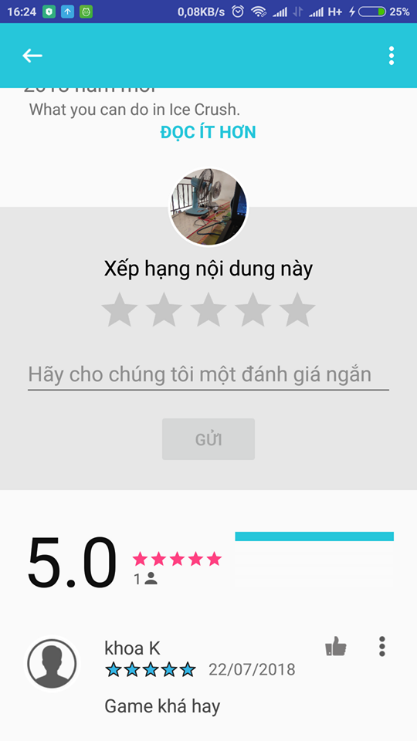
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Hiển thị thông tin sản phẩm | Sử dụng card view và recycle view để hiển thị.  Sử dụng view pager để vuốt | Thiết kế  Xử lý | Hoàn thành |
| Hiển thị Header navigation | Bao gồm: tên người dùng, giới tính, quyền, ví tiền. | Thiết kế  Xử lý | Hoàn thành |
| Hiển thị menu navigation | Bao gồm: Hồ sơ, danh sách ưu thích, tải lên, các công cụ quản trị và đăng xuất | Thiết kế  Xử lý | Hoàn thành |

* Hướng giải quyết:
  + Khi người dùng đăng nhập thành công, trang chủ sẽ hiển thị các tab quản lý sản phẩm đa phương tiện bao gồm:
    - Tab home.
    - Tab game.
    - Tab image.
    - Tab video.
    - Tab music.
  + Trong mỗi Tab, các sản phẩm đa phương tiện sẽ được phân loại rõ ràng. Dựa theo thông tin trong bảng Sections, chúng sẽ được sắp xếp hợp lí, bao gồm:
    - Home:
      * "Game mới + c.nhật"
      * "Thống trị Arcade"
      * "Được gợi ý cho bạn"
      * "Lập chiến lược và chinh phục"
      * "Đề xuất cho bạn"
    - Game:
      * "Chiến thuật"
      * "Đua xe"
      * "Nhập vai"
      * "Phiêu lưu"
      * "Thể thao"
      * "Câu đố"
      * "Giáo dục"
      * "Mô phỏng"
    - Image:
      * "Animal"
      * "Nature"
      * "Autumn"
      * "Flower"
      * "Love"
      * "Texure"
      * "Wallpaper"
      * "Business"
* Với danh sách các Sections, sẽ không thể nào hiển thị toàn bộ vì kích thước quá nặng, request quá nhiều. Do đó, em xử dụng kĩ thuật phân trang, với một tab em chỉ cho phép tải lên 5 sections trên một tab. Khi đó sẽ giảm thiểu được lượng dữ liệu mà người dùng chưa nhìn thấy. Chỉ khi kéo xuống, ứng dụng sẽ tải tiếp 5 sections nữa.
* Cũng giống với kĩ thuật phân trang, được áp dụng cho mỗi section, chúng ta sẽ chỉ cho tải lên 5 sản phẩm đa phương tiện trong mỗi section.
* Các trang trong màn hình chính
  + Trang màn hình chính:
    - Khi trang MainActivity được bắt đầu, sẽ tạo 5 trang SectionFagment có các chức năng hiển thị các sản phẩm khác nhau.
    - Mỗi trang sẽ lấy dữ liệu dựa theo từ khóa được MainActivity truyền vào thông qua intent.
  + Trang navigation menu:
    - Khi bắt đầu, Trang MainActivity sẽ gọi presenter để lấy dữ liệu người dùng hiển thị lên trang navigation.
* Cách thức làm:
  + Màn hình chính dùng để hiển thị danh sách sản phẩm đa phương tiện:
    - Chuyển sang ProductActivity khi bấm vào nút “More”.
    - Chuyển sang ProductDetailActivity khi bấm vào sản phẩm.
    - Mở thêm trang NavigationView khi bấm vào nút “Home”
* Phần Cài đặt:

Hình 9: Hình mã nguồn

### Hiển thị chi tiết sản phẩm (Chức năng #4):

* Giao diện sản phẩm

*Hình 11: Giao diện màn hình chi tiết sản phẩm*

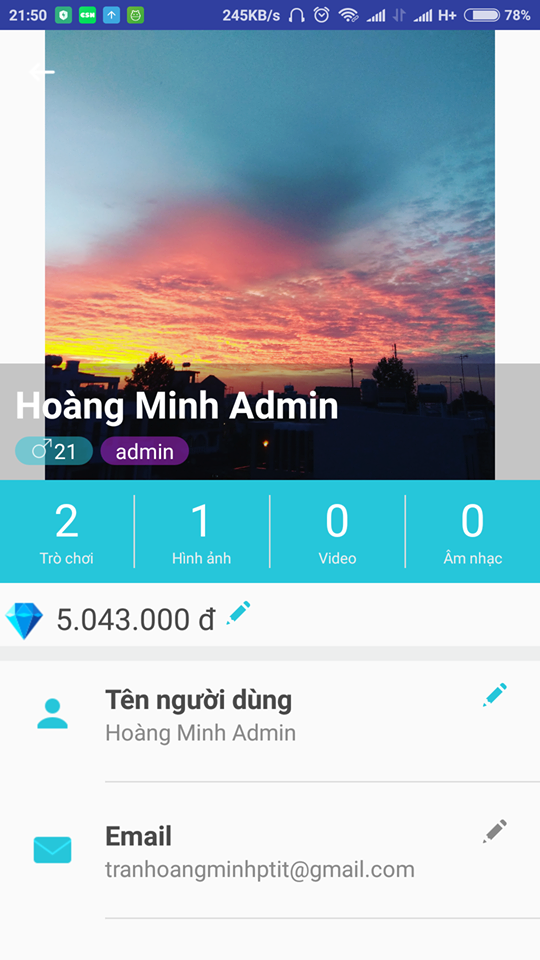
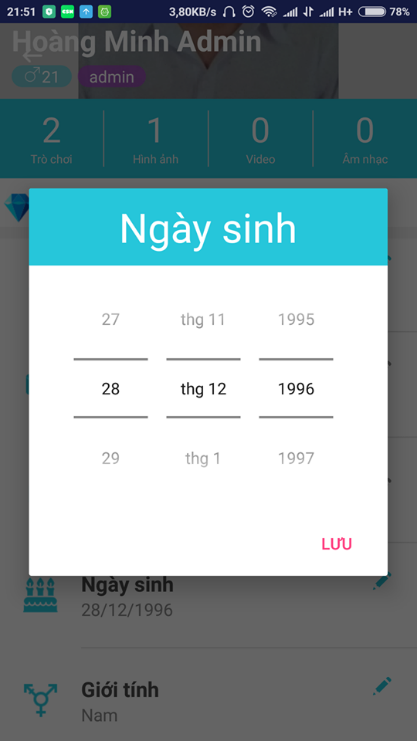
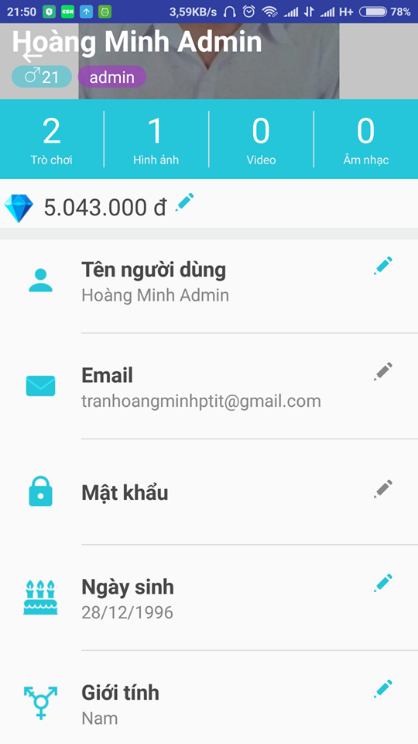
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Hiển thị thông tin chi tiết sản phẩm | Bao gồm: tên, chủ sở hữu, tuổi giới hạn, giá, giới thiệu, mô tả | Thiết kế  Xử lý | Hoàn thành |
| Hiển thị video | Sử dụng frame layout để hiển thị YouTube Player | Thiết kế  Xử lý | Hoàn thành |
| Lưu sản phẩm ưa thích | Đánh dấu bookmark lưu sản phẩm ưu thích | Xử lý | Hoàn thành |
| Hiển thị hệ thống đánh giá, xếp hạng | Thống kê điểm, luợt mua, số luợt bình chọn | Thiết kế  Xử lý | Hoàn thành |
| Cho phép người dùng đánh giá, xếp hạng | Sử dụng rating layout để đánh giá | Thiết kế  Xử lý | Hoàn thành |
| Thích bình luận | Người dùng nhấn like để thích phẩn bình luận | Xử lý | Hoàn thành |

* Hướng giải quyết:
  + Khi người dùng nhấp vào sản phẩm tại trang chủ, màn hình sẽ chuyển sang trang chi tiết sản phẩm. Người dùng có thể:
    - Xem video youtube giới thiệu sản phẩm.
    - Xem lượt mua và hệ thống đánh giá (điểm, lượt đánh giá).
    - Thanh toán sản phẩm bằng một nút nhấn.
    - Tự đánh giá bằng hệ thống đánh giá từ 0 đến 5 và gồm cả bình luận.
    - Xem danh sách đánh giá của người dùng với sản phẩm này, người dùng có thể nhấn thích bình luận
* Các trang trong chi tiết sản phẩm
  + Trang màn hình chi tiết sản phẩm:
    - Khi trang ProductDetailActivity bắt đầu sẽ lấy dữ liệu từ Firebase database và hiển thị lên người dùng
    - Mỗi trang sẽ lấy dữ liệu dựa theo id của sản phẩm truyền vào thông qua intent.
  + Menu tùy chọn:
    - Admin: Hiển thị toàn quyền với sản phẩm: sửa, ngừng kích hoạt hoặc kích hoạt
    - Mode: Hiển thị toàn quyền nếu là chủ sở hữu sản phẩm.
    - User: Ẩn menu.
  + Hộp thoại thanh toán:
    - Khi nhấn vào button thanh toán, hộp thoại thanh toán sẽ hiển thị, nếu người dùng đủ tiền sẽ cho thanh toán nếu không sẽ hiển thị nút nạp tiền. Khi thanh toán thành công, người dùng sẽ bị trừ tiền và khoản tiền sẽ được chuyển cho chủ sở hữu sản phẩm
* Phần Cài đặt:

### Hiển thị hồ sơ người dùng (Chức năng #5):

* Giao diện sản phẩm

*Hình 11: Giao diện màn hình hồ sơ người dùng*

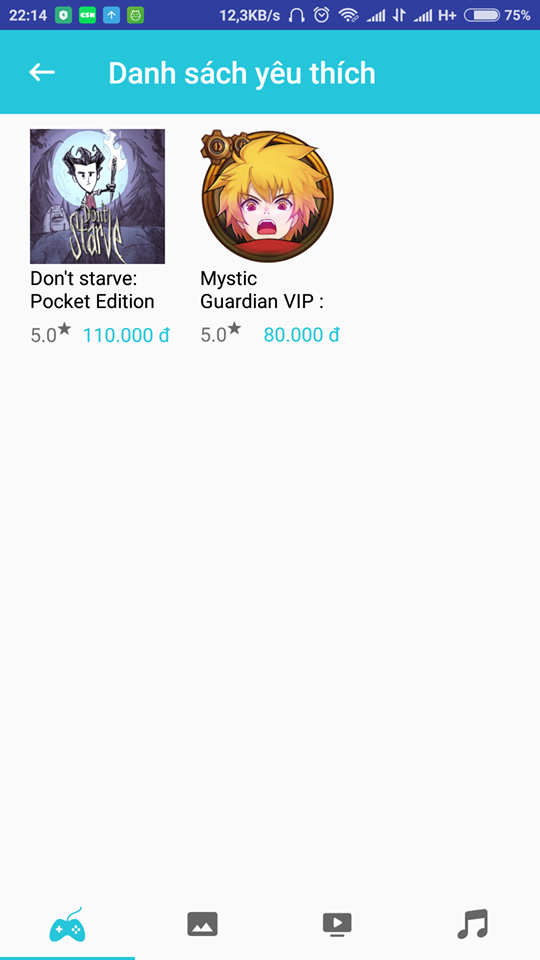
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Hiển thị thông tin người dùng | Bao gồm: họ tên, giới tính, tuổi, quyền, thống kê sản phẩm đã mua | Thiết kế  Xử lý | Hoàn thành |
| Cho phép chỉnh sửa thông tin | Hiển thị các Alert Dialog để nhập liệu | Thiết kế  Xử lý | Hoàn thành |
| Đổi tài khoản hoặc mật khẩu | Chỉ cho phép với tài khoản Local | Xử lý | Hoàn thành |
| Xem sản phẩm đã mua | Xem được các sản phẩm đã từng mua | Xử lý | Hoàn thành |

* Hướng giải quyết:
  + Khi người dùng nhấp vào hồ sơ tại menu navigation của trang chủ, màn hình sẽ chuyển sang trang hồ sơ người dùng. Người dùng có thể:
    - Xem thông tin cơ bản của mình.
    - Chỉnh sửa thông tin cơ bản.
    - Đổi tài khoản hoặc mật khẩu nếu là tài khoản local.
    - Xem lại các sản phẩm đã mua.
* Các trang trong chi tiết sản phẩm
  + Trang màn hình hồ sơ ngươi dùng:
    - Khi trang ProfileActivity bắt đầu sẽ lấy dữ liệu từ Firebase database từ mã người dùng và hiển thị lên.
    - Khi nhấp vào các sản phẩm đã mua, trang sẽ gửi mã loại và mã người dùng cho ProductActivity để hiển thị sản phẩm.
  + Alert Dialog:
    - Hiển thị khi người dùng bấm vào C:\Users\tranh\Desktop\MM\ic_pencil.png sẽ mở ra hộp thoại cho phép người dùng nhập vào thông tin cần chỉnh sửa.
* Phần Cài đặt:

### Hiển thị danh sách ưu thích (Chức năng #6):

* Giao diện sản phẩm



*Hình 11: Giao diện màn hình danh sách ưu thích*

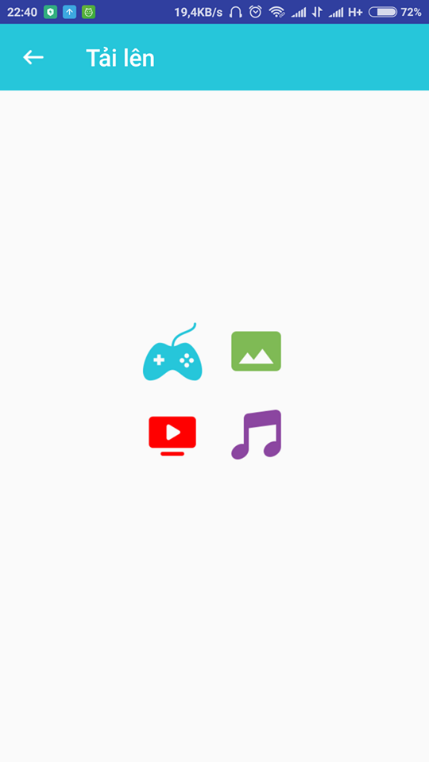
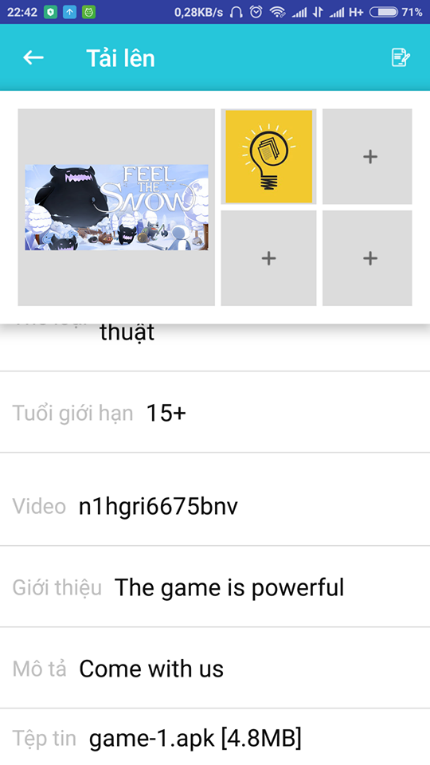
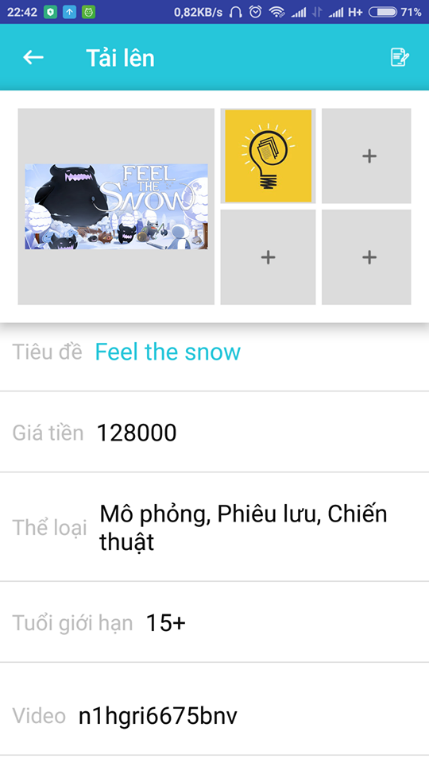
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Hiển thị sản phẩm đã đánh dấu ưu thích | Bao gốm: trò chơi, hình ảnh, video, nhạc | Thiết kế  Xử lý | Hoàn thành |

* Hướng giải quyết:
  + Khi người dùng nhấp vào danh sách yêu thích trong menu navigation của trang chủ, màn hình sẽ chuyển sang trang danh sách các sản phẩm đã đánh dấu bookmark. người dùng có thể:
    - Xem lại các sản phẩm mình đã lưu.
* Các trang trong sản phẩm đã lưu:
  + Trang màn hình thông tin sản phẩm:
    - Khi trang BookmarkActivity bắt đầu sẽ lấy dữ liệu từ Firebase database từ mã người dùng và loại sản phẩm để hiển thị.
    - Khi nhấp vào các sản phẩm đã lưu, trang sẽ gửi mã sản phẩm cho ProductDetailActivity để hiển thị chi tiết sản phẩm.
* Phần Cài đặt:

### Upload, cập nhật sản phẩm (Chức năng #7):

* Giao diện sản phẩm

*Hình 11: Giao diện màn hình upload / chỉnh sửa sản phẩm*

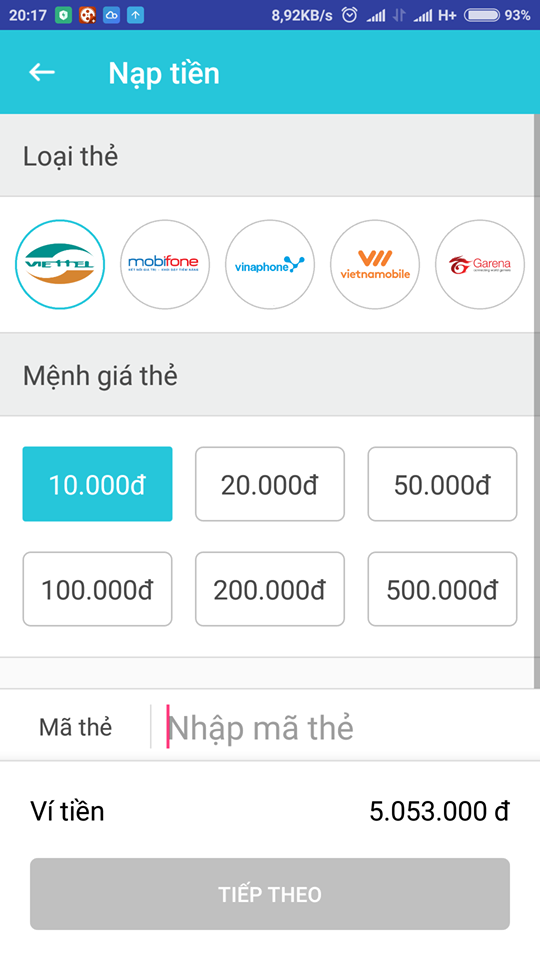
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Chọn loại sản phẩm | Bao gốm: trò chơi, hình ảnh, video, nhạc | Thiết kế  Xử lý | Hoàn thành |
| Chọn hình đại diện, hình mô tả thêm | Sử dụng camera hoặc thư viện | Thiết kế  Xử lý | Hoàn thành |
| Cho phép nhập dữ liệu | Bao gồm: tiêu đề, giá tiền, thể loại, video, giới thiệu, mô tả | Thiết kế  Xử lý | Hoàn thành |
| Chọn tuổi giới hạn và tệp tin | Hiển thị dialog và thư viện bộ nhớ trong | Thiết kế  Xử lý | Hoàn thành |

* Hướng giải quyết:
  + Khi người cộng tác nhấp vào nút upload trong menu navigation của trang chủ, màn hình sẽ chuyển sang trang upload chọn loại sản phẩm. Sau khi chọn loại sản phẩm. Người cộng tác có thể nhập thông tin sản phẩm và upload hoặc chỉnh sửa sản phẩm, người cộng tác có thể:
    - Tạo mới và upload sản phẩm.
    - Chỉnh sửa sản phẩm
* Các trang trong upload/chỉnh sửa sản phẩm:
  + Fragment chọn loại sản phẩm:
    - Khi trang OptionModifyProductFragment bắt đầu sẽ hiển thị loại sản phẩm.
    - Khi nhấp vào loại sản phẩm, trang sẽ gửi mã sản phẩm cho ModifyProductFragment để hiển thị ô nhập liệu phù hợp.
  + Fragment nhập thông tin sản phẩm:
    - Khi nhấn lưu, trang sẽ lưu thông tin sản phẩm mới lên Firebase database hoặc chỉnh sửa sản phẩm.
* Phần Cài đặt:

### Nạp tiền (Chức năng #8):

* Giao diện sản phẩm



*Hình 11: Giao diện màn hình nạp tiền của người dùng*

* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Chọn loại sản phẩm | Bao gốm: trò chơi, hình ảnh, video, nhạc | Thiết kế  Xử lý | Hoàn thành |
| Chọn hình đại diện, hình mô tả thêm | Sử dụng camera hoặc thư viện | Thiết kế  Xử lý | Hoàn thành |
| Cho phép nhập dữ liệu | Bao gồm: tiêu đề, giá tiền, thể loại, video, giới thiệu, mô tả | Thiết kế  Xử lý | Hoàn thành |
| Chọn tuổi giới hạn và tệp tin | Hiển thị dialog và thư viện bộ nhớ trong | Thiết kế  Xử lý | Hoàn thành |

* Hướng giải quyết:
  + Khi người cộng tác nhấp vào nút upload trong menu navigation của trang chủ, màn hình sẽ chuyển sang trang upload chọn loại sản phẩm. Sau khi chọn loại sản phẩm. Người cộng tác có thể nhập thông tin sản phẩm và upload hoặc chỉnh sửa sản phẩm, người cộng tác có thể:
    - Tạo mới và upload sản phẩm.
    - Chỉnh sửa sản phẩm
* Các trang trong upload/chỉnh sửa sản phẩm:
  + Fragment chọn loại sản phẩm:
    - Khi trang OptionModifyProductFragment bắt đầu sẽ hiển thị loại sản phẩm.
    - Khi nhấp vào loại sản phẩm, trang sẽ gửi mã sản phẩm cho ModifyProductFragment để hiển thị ô nhập liệu phù hợp.
  + Fragment nhập thông tin sản phẩm:
    - Khi nhấn lưu, trang sẽ lưu thông tin sản phẩm mới lên Firebase database hoặc chỉnh sửa sản phẩm.
* Phần Cài đặt:

### Thanh toán (Chức năng #9):

### Tìm kiếm (Chức năng #10):

### Hệ thống đánh giá, bình chọn (Chức năng #11):

### Các chức năng của người quản trị (Chức năng #12):

#### Công cụ quản lí người dùng:

#### Công cụ quản lí thẻ cào:

#### Công cụ quản lí sản phẩm:

# CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN



## Kết quả đạt được

Website Phủi (mạng xã hội thu nhỏ dành cho những tín đồ đam mê đá bóng) về cơ bản đã giải quyết một phần nào khó khăn hiện nay của các đội bóng bao gồm các chức năng cho phép người dùng dễ dàng trong việc tìm kiếm đội bóng cũng như chiêu mộ thành viên, tạo trận đấu với những đội bóng mình mong muốn.

Về các yêu cầu phi chức năng, website có tốc độ hiển thị khá tốt và giao diện bắt mắt, chuyên nghiệp.

Các chức năng yêu cầu của website như thiết kế ban đầu đã được hoàn thành, đi từ việc hiển thị những Team đang muốn tìm đội để thi đấu, chấp nhận thách đấu, tạo thách đấu, đề nghị tham gia Team, chấp nhận gia nhập Team.

Trong quá trình làm đồ án, bản thân em cũng đã học hỏi và tìm hiểu được rất nhiều các công nghệ mới cũng như áp dụng vào thực tế đề tài nhiều công nghệ như đã trình bày ở trong báo cáo.

## Hạn chế

Tuy nhiên, do thời gian ngắn và số lượng công việc cũng như tìm hiểu khá nhiều công nghệ cho nên việc xảy ra lỗi xảy ra nhiều làm cho quá trình hoàn thành bị chậm. Bên cạnh đó một số các chức năng dự kiến thêm vào vẫn chưa được triển khai như tạo sân bóng, tạo bảng giá sân bóng ở các khung giờ, chủ sân cập nhật tỉ số các trận đấu, tính điểm sau mỗi trận đấu,thông báo đến từng thành viên trong đội khi có trận đấu.

## Hướng phát triển

* Về mặt đối tượng sử dụng, có thể mở rộng quy mô của hệ thống khi thêm các đối tượng sử dụng khác vào hệ thống như chủ sân để quản lý, đăng bảng giá sân, khuyến mãi sân, cũng như cập nhật tỉ số các trận đấu…v..v.
* Về mặt hiển thị thông báo, có thể gửi thông báo đến từng thành viên trước 2h khi có trận đấu sắp diễn ra.
* Về mặt trò chuyện, có thể nhắn tin giữa các người dùng để bàn bạc thêm.
* Có thể thêm chức năng giải đấu.

**DANH MỤC TÀI LIỆU THAM KHẢO**

**Danh mục các Website tham khảo:**

1. http://androidseriestutorial.blogspot.com/2014/08/gioi-thieu-so-luoc-ve-he-ieu-hanh.html
2. https://vi.wikipedia.org/wiki/Android\_(h%E1%BB%87\_%C4%91i%E1%BB%81u\_h%C3%A0nh)
3. https://www.dienmayxanh.com/kinh-nghiem-hay/nhung-uu-nhuoc-diem-khi-chon-mua-dien-thoai-androi-542188
4. https://androidcoban.com/tong-quan-ve-firebase-google-phien-ban-moi.html
5. https://code.tutsplus.com/vi/articles/android-from-scratch-an-overview-of-android-application-development--cms-25972
6. https://viblo.asia/p/mvp-pattern-for-android-1qm6RWzOveJE

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập- Tự do- Hạnh phúc**

TP. Hồ Chí Minh, ngày tháng năm 2018

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

1. **Tên đề tài:**
2. **Sinh viên: Lớp:**
3. **Giáo viên hướng dẫn:**
4. **Nơi công tác:**

**NỘI DUNG NHẬN XÉT**

1. Đánh giá chung:

………………………………………………………………………………………………………………………………………………………………………………

1. Đánh giá chi tiết:

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

1. Nhận xét về tinh thần, thái độ làm việc:

………………………………………………………………………………………………………………………………………………………………………………

1. Kết luận:

………………………………………………………………………………………………………………………………………………………………………………

1. Điểm hướng dẫn ():

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký, ghi rõ họ tên)