

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

□ □ □ □



BÁO CÁO ĐỒ ÁN CUỐI KỲ
MÔN LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

Đề tài:

ỨNG DỤNG QUẢN LÝ, MUA GAME
GAMEHUB

Giảng viên hướng dẫn : Thầy Huỳnh Tuấn Anh

Lớp : SE346.J11

Nhóm: HelloWorld

Sinh viên thực hiện:

Nguyễn Ngọc Duy 15520165

Vũ Khắc Hợi 15520269

TP. Hồ Chí Minh, ngày 19 tháng 12 năm 2018

LỜI CẢM ƠN

Đầu tiên, nhóm tác giả xin gửi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô Khoa Công nghệ phần mềm đã giúp cho nhóm tác giả có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, nhóm tác giả xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Thầy Huỳnh Tuấn Anh. Thầy đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báu giúp nhóm tác giả hoàn thành tốt đồ án môn học của mình.

Trong thời gian một học kỳ thực hiện đề tài, nhóm tác giả đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm tác giả vận dụng tối đa những gì đã thu thập được để hoàn thành đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm tác giả không tránh khỏi những thiếu sót. Chính vì vậy, nhóm tác giả rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà nhóm tác giả đã học tập và là hành trang để nhóm tác giả thực hiện tiếp các đề tài khác trong tương lai.

Xin chân thành cảm ơn quý Thầy Cô !

Nhóm sinh viên thực hiện

NHẬN XÉT CỦA GIẢNG VIÊN

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

MỤC LỤC

LỜI CẢM ƠN.....	1
MỤC LỤC.....	3
GIỚI THIỆU	5
CHƯƠNG 1. GIỚI THIỆU CÔNG NGHỆ THIẾT BỊ DI ĐỘNG VÀ HỆ ĐIỀU HÀNH ANDROID	6
1.1 GIỚI THIỆU VỀ ĐIỆN THOẠI DI ĐỘNG:	6
1.1.1 Khái niệm	6
1.1.2 Lịch sử phát triển:	6
1.1.3 Điện thoại thông minh	6
1.1.4 Ứng dụng trên thiết bị di động.....	7
1.2 HỆ ĐIỀU HÀNH ANDROID	8
1.2.1 Giới thiệu.....	8
1.2.2 Vị thế của Android	9
1.2.4 Ứng dụng trong Android	9
1.2.5 Thị phần và tỉ lệ sử dụng của Android.....	10
1.2.6 Cộng đồng mã nguồn mở	11
1.3 GIỚI THIỆU VỀ REACT NATIVE.....	12
1.3.1 React Native	12
1.3.2 Ưu điểm	12
1. React Native sử dụng Javascript.....	12
2. Hot reloading! Không cần tốn thời gian để recompiling	13
3. Thực sự Native	13
4. Code một lần, chạy nhiều chỗ.....	13
5. Một cộng đồng hỗ trợ rất lớn	13
1.3.3 Nhược điểm	14
CHƯƠNG 2. WEB API – ASP.net Core.....	15
2.1 WEB API.....	15
2.1.1 Giới thiệu ASP.net Core.....	15
2.1.2 Tại sao dùng ASP.net Core	15
3.1 ĐẶC TẢ VÀ YÊU CẦU ỨNG DỤNG	16

3.2 CÁCH THỨC HOẠT ĐỘNG CỦA ỨNG DỤNG	16
CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG	17
4.1 XÂY DỰNG ỨNG DỤNG	17
4.1.1 Giới thiệu tổng quan kiến trúc chương trình	17
4.1.2 Chi tiết các thành phần	17
4.2 CÁC THƯ VIỆN SỬ DỤNG TRONG ỨNG DỤNG	18
4.2.1 Axios	18
- Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests).....	18
4.2.1 ImageSlider	18
4.2.2 React Native Base	18
4.3 GIAO DIỆN CHƯƠNG TRÌNH	19
4.3.1 Giao diện Login.....	19
4.3.2 Giao diện đăng kí	20
4.3.3 Giao diện Home	21
4.3.4 Giao diện Game Detail	22
4.3.5 Giao Game List	23
23	
4.3.6 Giao diện Game Library	24
26	
4.4 CÀI ĐẶT VÀ KIỂM THỬ	27
4.3.1 Môi trường	27
4.3.2 Thử nghiệm và đánh giá kết quả	27
4.3.3 Thử nghiệm và đánh giá kết quả	27
KẾT LUẬN VÀ HƯỚNG MỞ RỘNG.....	28
MÃ NGUỒN MỞ VÀ TÀI LIỆU THAM KHẢO.....	29

GIỚI THIỆU

Hiện nay, tìm kiếm thông tin là một nhu cầu hàng đầu và thiết yếu trong xã hội, tìm kiếm đã trở thành một điều cần thiết cho hầu hết tất cả mọi người dù cho người đó là ai, ngành nghề hoặc nhu cầu cá nhân nào. Từ việc tìm kiếm văn bản, tìm kiếm hình ảnh cho đến tìm kiếm bản đồ, trang web, tất tần tật mọi thứ sẽ có nếu bạn muốn tìm kiếm nó trên Internet. Có nhiều công cụ tìm kiếm, nhưng nổi bật và nắm giữ nhiều thị phần tìm kiếm nhất là Google Search của Google.

Ở đây chúng tôi xin đề cập đến dịch vụ tìm kiếm bản đồ Google Maps. Google map được xem là dịch vụ bản đồ số được Google phát triển với mục đích thay thế cho các loại bản đồ giấy thông thường. Giờ đây chỉ bằng chiếc smartphone nhỏ gọn bạn có thể tự do lựa chọn những địa điểm mà mình muốn đến và nhanh chóng tiếp cận được những dịch vụ xung quanh các địa điểm đó. Để sử dụng Google map một cách chính xác nhất bạn cần GPS- hệ thống định vị toàn cầu giúp bạn có thể biết rõ vị trí hiện tại của bản thân và thông qua GPS con người có thể dễ dàng xác định được phương hướng và đường đi một cách nhanh chóng nhất có thể

Chính vì vậy, chúng tôi đã ứng dụng dịch vụ tìm kiếm ưu việt này vào ứng dụng của mình, thông qua việc sử dụng API do Google Maps cung cấp. Thông qua API chúng tôi có thể lấy được bản đồ từ Google và thỏa sức tùy chỉnh cho nó.

Trước đây, việc tìm kiếm một địa chỉ ăn uống phần lớn thông qua các công cụ tìm kiếm, chẳng hạn như Google. Hiện tại có thể thói quen đó vẫn đang còn, tuy nhiên đứng trước cả đại dương thông tin như vậy thì việc sàng lọc cũng như xem xét các đánh giá về địa điểm ăn uống trở nên phức tạp và tốn nhiều thời gian tìm kiếm.

Ứng dụng Find Best Restaurant của chúng tôi, cho phép chỉ vài giây với vài thao tác đơn giản người dùng đã tìm kiếm và lựa chọn cho mình địa điểm ăn uống tốt nhất, thích hợp nhất và trong khoảng cách mong muốn. Đồng thời sẽ chỉ dẫn cụ thể các bước di chuyển theo bản đồ để người dùng có thể đến được địa điểm đấy. Ngoài ra người dùng có thể lưu lại địa điểm ăn uống yêu thích vào thiết bị của mình chỉ bằng một thao tác đơn giản.

Như vậy, việc của bạn chỉ là mở ứng dụng lên và chọn khoảng cách tìm kiếm mong muốn, việc còn lại hãy để chúng tôi.

CHƯƠNG 1. GIỚI THIỆU CÔNG NGHỆ THIẾT BỊ DI ĐỘNG VÀ HỆ ĐIỀU HÀNH ANDROID

1.1 GIỚI THIỆU VỀ ĐIỆN THOẠI DI ĐỘNG:

1.1.1 Khái niệm

Điện thoại di động (ĐTDD), còn gọi là điện thoại cầm tay, là loại điện thoại kết nối dựa trên sóng điện từ vào mạng viễn thông. Nhờ có kết nối sóng, còn gọi là "kết nối không dây", mà ĐTDD thực hiện trao đổi thông tin khi đang di chuyển.

Phạm vi di chuyển mà vẫn giữ trao đổi thông suốt phụ thuộc vào vùng phủ sóng của nhà cung cấp dịch vụ, gọi gọn là *nhà mạng*. Nhà cung cấp dịch vụ lắp đặt mạng lưới các trạm thu phát, mỗi trạm đảm bảo kết nối trong phạm vi 8 đến 13 km. Chất lượng sóng phụ thuộc vào thiết bị mạng và phần nào địa hình nơi sử dụng máy chứ ít khi bị giới hạn về không gian.

Vào thời kỳ phát triển đầu tiên, hồi thế kỷ 20, ĐTDD được chế tạo chủ yếu phục vụ liên lạc lời nói. Ngày nay ĐTDD vẫn được chế tạo với kích cỡ bỏ túi, có thêm phần nghe nhạc, chụp ảnh,...

1.1.2 Lịch sử phát triển:

Ngày 10 tháng 3 năm 1876 được coi là mốc son đánh dấu sự ra đời của điện thoại. Cha đẻ của chiếc điện thoại đầu tiên là Alexander Graham Bell. Chiếc máy thô sơ có thể truyền được giọng nói này đã mở ra một kỷ nguyên phát triển mới trong lịch sử thông tin liên lạc, thay thế cho điện tín.

Năm 1967, chiếc điện thoại được coi là "di động" đầu tiên trình làng với tên gọi Carry Phone, rất cồng kềnh cho việc di chuyển vì nó nặng đến 4,5 kg.

Điện thoại di động chính thức ra đời vào ngày 3 tháng 4 năm 1973, mang tên Motorola Dyna Tac, phát minh bởi nhà sáng chế John F. Mitchell và Martin Cooper^[1]. Motorola Dyna Tac mang hình dáng gần giống điện thoại di động ngày nay mặc dù vẫn còn khá cồng kềnh (nặng khoảng 1 kg) và không phổ biến.

Từ đó đến nay, chiếc điện thoại di động phát triển không ngừng phát triển theo hướng nhỏ gọn hơn rất nhiều tổ tiên của nó và ngày càng được tích hợp nhiều chức năng hơn chứ không còn đơn thuần là nghe và gọi.

1.1.3 Điện thoại thông minh

Điện thoại thông minh hay smartphone là khái niệm để chỉ chiếc điện thoại tích hợp một nền tảng hệ điều hành di động với nhiều tính năng hỗ trợ

tiên tiến về điện toán và kết nối dựa trên nền tảng cơ bản của điện thoại di động thông thường.

Ban đầu điện thoại thông minh bao gồm các tính năng của điện thoại di động thông thường kết hợp với các thiết bị phổ biến khác như PDA, thiết bị điện tử cầm tay, máy ảnh kỹ thuật số, hệ thống định vị toàn cầu GPS. Điện thoại thông minh ngày nay bao gồm tất cả chức năng của laptop như duyệt web, Wi-Fi, các ứng dụng của bên thứ 3 trên di động và các phụ kiện đi kèm cho máy.

Các ngành công nghiệp tin rằng điện thoại thông minh đa số có một màn hình độ phân giải cao hơn so với điện thoại truyền thống và điện thoại thông minh như một máy tính di động, vì nó có một hệ điều hành riêng biệt vì có thể hiển thị phù hợp các trang website bình thường và người dùng có thể thay đổi một giao diện và sở hữu khả năng mở ứng dụng tiện hơn và dễ dàng cài đặt lẫn gỡ bỏ ứng dụng; điện thoại thông minh có màn hình cảm ứng độ phân giải cao, và sẵn sàng để gọi bàn phím ảo và viết chữ tay.

Có thể tiến hành đa tác vụ thao tác, và có một đa phương tiện mạnh mẽ, Email, Internet và hoàn toàn có thể thay đổi các thiết bị truyền thống như MP3 và PDA điện thoại thông minh có thể thay thế xử lý các vấn đề máy tính văn phòng và các vấn đề khác, nó có thể giao tiếp với mạng duy trì một thời gian kết nối liền mạch với thời gian, đồng thời có thể vô hiệu hóa mạng bất cứ lúc nào, và đồng bộ hóa dữ liệu với máy tính, máy tính xách tay và các thiết bị khác.

Những điện thoại thông minh phổ biến nhất hiện nay dựa trên nền tảng của hệ điều hành Android của Google và iOS của Apple.

1.1.4 Ứng dụng trên thiết bị di động

Một phần mềm ứng dụng trên thiết bị di động, còn được gọi tắt là ứng dụng di động, hoặc chỉ ứng dụng, (tiếng Anh: Mobile app hoặc app) là phần mềm ứng dụng được thiết kế để chạy trên điện thoại thông minh, máy tính bảng và các thiết bị di động khác.

Các ứng dụng thường có sẵn thông qua các nền tảng phân phối ứng dụng, bắt đầu xuất hiện vào năm 2008 và thường được điều hành bởi các chủ sở hữu của hệ điều hành di động, như Apple App Store, Google Play, Windows Phone Store, và BlackBerry App World. Một số ứng dụng miễn phí, trong khi một số ứng dụng phải được mua.

Thuật ngữ "ứng dụng" là một rút ngắn của thuật ngữ "phần mềm ứng dụng". Trong tiếng Anh, thường được viết là app và đã trở thành rất phổ biến và trong năm 2010 đã được liệt kê như là " từ ngữ của năm" do Hiệp hội American Dialect Society chọn lọc.

Ứng dụng di động ban đầu được cung cấp với mục đích thông tin tổng quát và các dịch vụ thông dụng trên mạng toàn cầu, bao gồm email, lịch, danh bạ, và thị trường chứng khoán và thông tin thời tiết. Tuy nhiên, nhu cầu chung của những người sử dụng thiết bị di động và khả năng phát triển của các nhà lập trình đã mở rộng thành các loại khác, chẳng hạn như trò chơi di động, tự động hóa nhà máy, GPS và các dịch vụ dựa trên địa điểm, định vị và ngân hàng, để theo dõi, mua vé và các ứng dụng y tế di động gần đây. Sự bùng nổ về số lượng và sự đa dạng của các ứng dụng đã tạo ra 1 tiềm năng và thị trường lớn.

Sự phổ biến của các ứng dụng di động đã tiếp tục tăng.^[2] Theo công ty nghiên cứu thị trường Gartner, 102 tỷ ứng dụng sẽ được tải về trong năm 2013 (91% trong số đó là miễn phí) nhưng chúng vẫn sẽ tạo ra 26 tỷ USD, tăng 44,4% so với 18 tỷ USD vào năm 2012.^[3] Báo cáo phân tích ước tính rằng nền kinh doanh ứng dụng tạo ra doanh thu hơn 10 tỷ €cho mỗi năm trong Liên minh châu Âu, trong khi hơn 529.000 công ăn việc làm đã được tạo ra trong 28 quốc gia EU do sự tăng trưởng của thị trường ứng dụng.

1.2 HỆ ĐIỀU HÀNH ANDROID

1.2.1 Giới thiệu

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. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cảm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

1.2.2 Vị thế của Android

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết 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, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, 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.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý. Những dự án này bổ sung các tính năng cao cấp cho những người dùng thích tìm tòi hoặc đưa Android vào các thiết bị ban đầu chạy hệ điều hành khác.

Android chiếm 87,7% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 2 năm 2017, với tổng cộng 2 tỷ thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là "cuộc chiến điện thoại thông minh" giữa các công ty công nghệ.

1.2.4 Ứng dụng trong Android

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 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.^[44] 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 tỷ.

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK). 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, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) 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.

Để vượt qua những hạn chế khi tiếp cận các dịch vụ của Google do sự Kiểm duyệt Internet tại Cộng hòa Nhân dân Trung Hoa, các thiết bị Android bán tại Trung Quốc lục địa thường được điều chỉnh chỉ được sử dụng dịch vụ đã được duyệt.

1.2.5 Thị phần và tỉ lệ sử dụng của Android

Công ty nghiên cứu thị trường Canalys đã ước tính trong quý 2 năm 2009 rằng Android có 2,8% thị phần điện thoại thông minh được bán ra toàn cầu. Đến quý 4 năm 2010 con số này tăng lên 33% thị phần, trở thành nền tảng điện thoại thông minh bán chạy hàng đầu. Đến quý 3 năm 2011 Gartner ước tính rằng hơn một nửa (52,5%) thị trường điện thoại thông minh thuộc về Android. Đến quý 3 năm 2012 Android đã có 75% thị phần điện thoại thông minh toàn cầu theo nghiên cứu của hãng IDC.

Vào tháng 7 năm 2011, Google nói rằng có 550.000 thiết bị Android mới được kích hoạt mỗi ngày, đỉnh điểm là 400.000 máy một ngày vào tháng 5, và có hơn 100 triệu thiết bị đã được kích hoạt với mức tăng 4,4% mỗi tuần. Vào tháng 9 năm 2012, 500 triệu thiết bị đã được kích hoạt với 1,3 triệu lượt kích hoạt mỗi ngày.

Thị phần của Android có khác nhau theo khu vực. Vào tháng 7 năm 2012, thị phần Android tại Mỹ là 52%, nhưng lên tới 90% tại Trung Quốc.

Tỷ lệ sử dụng các phiên bản khác nhau tính đến tháng 4 năm 2014. Phần lớn các thiết bị Android cho tới nay vẫn chạy hệ điều hành phiên bản 4.1.x Jelly Bean được phát hành ngày 9 tháng 7 năm 2012 nhờ tính ổn định và hỗ trợ tốt các máy có cấu hình thấp.

1.2.6 Cộng đồng mã nguồn mở

Android có một cộng đồng các lập trình viên và những người đam mê rất năng động. Họ sử dụng mã nguồn Android để phát triển và phân phối những phiên bản chỉnh sửa của hệ điều hành. Các bản Android do cộng đồng phát triển thường đem những tính năng và cập nhật mới vào nhanh hơn các kênh chính thức của nhà sản xuất/nhà mạng, tuy không được kiểm thử kỹ lưỡng cũng như không có đảm bảo chất lượng; cung cấp sự hỗ trợ liên tục cho các thiết bị cũ không còn nhận được bản cập nhật chính thức; hoặc mang Android vào những thiết bị ban đầu chạy một hệ điều hành khác, như HP Touchpad. Các bản Android của cộng đồng thường được root sẵn và có những điều chỉnh không phù hợp với những người dùng không rành rẽ, như khả năng ép xung hoặc tăng/giảm áp bộ xử lý của thiết bị. CyanogenMod là firmware của cộng đồng được sử dụng phổ biến nhất, và hoạt động như một tổ chức của số đông khác.

Trước đây, nhà sản xuất thiết bị và nhà mạng tỏ ra thiếu thiện chí với việc phát triển firmware của bên thứ ba. Những nhà sản xuất còn thể hiện lo ngại rằng các thiết bị chạy phần mềm không chính thức sẽ hoạt động không tốt và dẫn đến tổn tiền hỗ trợ. Hơn nữa, các firmware đã thay đổi như CyanogenMod đôi khi còn cung cấp những tính năng, như truyền tải mạng (tethering), mà người dùng bình thường phải trả tiền nhà mạng mới được sử dụng. Kết quả là nhiều thiết bị bắt đầu đặt ra hàng rào kỹ thuật như khóa bootloader hay hạn chế quyền truy cập root. Tuy nhiên, khi phần mềm do cộng đồng phát triển ngày càng trở nên phổ biến, và sau một thông cáo của Thư viện Quốc hội Hoa Kỳ cho phép "jailbreak" (vượt ngục) thiết bị di động, các nhà sản xuất và nhà mạng đã tỏ ra mềm mỏng hơn với các nhà phát triển thứ ba, thậm chí một số hãng như HTC, Motorola, Samsung và Sony, còn hỗ trợ và khuyến khích phát triển. Kết quả của việc này là dần dần nhu cầu tìm ra các hạn chế phần cứng để cài đặt được firmware không chính thức đã bớt đi do ngày càng nhiều thiết bị được phát hành với bootloader đã mở khóa sẵn hoặc có thể mở khóa, tương tự như điện thoại dòng Nexus, tuy rằng thông thường họ sẽ yêu cầu người dùng từ bỏ chế độ bảo hành nếu họ làm như vậy. Tuy nhiên, tuy được sự chấp thuận của nhà sản xuất, một số nhà mạng tại Mỹ vẫn bắt buộc điện thoại phải bị khóa.

Việc mở khóa và "hack" điện thoại thông minh và máy tính bảng vẫn còn là tác nhân gây căng thẳng giữa cộng đồng và công nghiệp. Cộng đồng luôn biện hộ rằng sự hỗ trợ không chính thức ngày càng trở nên quan trọng trước việc nền công nghiệp không cung cấp các bản cập nhật thường xuyên và/hoặc ngưng hỗ trợ cho chính các thiết bị của họ.

1.3 GIỚI THIỆU VỀ REACT NATIVE

1.3.1 *React Native*

React Native được phát triển bởi Facebook với mục đích ban đầu là áp dụng vào mạng xã hội lớn nhất hành tinh: Facebook. Do đặc tính công nghệ của mạng xã hội, Facebook cần phải tạo ra nền tảng phát triển ứng dụng di động đa nền tảng có hiệu năng không thua kém so với ứng dụng được phát triển độc lập cho từng nền tảng. React Native hiện tại chỉ hỗ trợ phát triển ứng dụng di động hệ điều hành Android và iOS, ít hơn so với Ionic (Android, iOS, Windows Phone).

React Native chính thức trở thành mã nguồn mở vào tháng 3 năm 2015. Cho đến nay, React Native được áp dụng trong nhiều ứng dụng của ta và cả nước ngoài.

React Native tạo ra ứng dụng native chứ không phải hybrid app như ionic,... Với 100% Native UI, React Native tạo nên sự “mượt mà” trong việc render UI và xóa nhòa khoảng cách về hiệu năng giữa ứng dụng đa nền tảng so với ứng dụng độc lập nền tảng.

1.3.2 *Ưu điểm*

1. React Native sử dụng Javascript.

- Javascript là một ngôn ngữ tương đối dễ dàng Ngôn ngữ

Javascript là tương đối dễ dàng để tìm hiểu và bao gồm các cú pháp đó là gần tiếng Anh. Nó sử dụng mô hình DOM cung cấp nhiều tính năng được viết sẵn cho các đối tượng khác nhau trên các trang giúp bạn dễ dàng phát triển một kịch bản để giải quyết một mục đích tùy chỉnh.

- Javascript tương đối nhanh đối với người dùng cuối

Khi mã được thực hiện trên máy tính của người dùng, kết quả và quá trình xử lý được hoàn thành hầu như ngay lập tức tùy thuộc vào nhiệm vụ (nhiệm vụ trong javascript trên các trang web thường rất đơn giản để tránh bị hog bộ nhớ)

Không cần phải được xử lý trong máy chủ web của trang web và gửi lại cho người dùng tiêu tốn băng thông của địa phương cũng như máy chủ.

2. Hot reloading! Không cần tốn thời gian để recompiling

- Với Hot Reloading, trạng thái của ứng dụng được duy trì, và nó sẽ reload lại chính màn hình đang sửa code, tất cả ngay trước mắt bạn trong chỉ ít hơn vài giây.

3. Thực sự Native

- Với React Native, những component cơ bản(View, Image, Button, Checkbox) đều là những component native. Do đó ứng dụng sẽ mang một trải nghiệm liền mạch với từng hệ điều hành. Điều đó thực sự làm lên một khác biệt lớn so với các framework lập trình cross-platform khác.

- Nó thực sự ấn tượng hơn nữa khi bạn lại code bằng ngôn ngữ javascript và render ra các component native từ OS.

4. Code một lần, chạy nhiều chỗ

- Bạn không cần biết Objective-C, Swift hay Java. Với Javascript và JSX, bạn có thể xây dựng một ứng dụng hoạt động đa hệ điều hành rất tốt.

- Từ kinh nghiệm, mình thấy rằng gần 95% code được chia sẻ giữa iOS và Android, và chỉ cần tinh chỉnh nhỏ(tweak) để hoàn thiện sản phẩm cuối cùng cho từng hệ điều hành. Đó có phải là điều bạn hướng tới không? Bạn không cần phải có nhiều team và codebase để hỗ trợ các hệ điều hành khác nhau

5. Một cộng đồng hỗ trợ rất lớn

- Nhiều năm trở lại đây, React Native đang trở lên rất phổ biến, nhiều nhà lập trình đang đóng góp để làm React Native tốt hơn mỗi ngày. Đặc biệt là nó được chống lưng bởi tập đoàn Facebook.

- React Native Github repro là một nguồn mở và có hàng nghìn cộng tác viên hoạt động rất năng nổ.

- Có một diễn đàn thảo luận mới về React Native và bạn cũng có thể là một phần trong đó nếu muốn.

- Stack Overflow là một địa chỉ nữa nơi mà có nhiều tài liệu và câu hỏi đã được trả lời về React Native.

Cộng đồng rất lớn và đang dần phát triển hơn nữa. Nhiều vấn đề đã và đang được giải quyết và bạn sẽ không cần phải tốn thời gian để nghiên cứu lại trong suốt quá trình phát triển của bản thân.

1.3.3 Nhược điểm

- Vẫn đòi hỏi native code.

- Hiệu năng sẽ thấp hơn với app thuần native code.
- Bảo mật không cao do dựa trên JS.
- Quản lý bộ nhớ.
- Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

CHƯƠNG 2. WEB API – ASP.net Core

2.1 WEB API

2.1.1 Giới thiệu ASP.net Core

ASP.NET Core là một framework mã nguồn mở, hiệu suất cao và đa nền tảng dùng để xây dựng các ứng dụng hiện đại có kết nối với Internet và dựa trên mô hình đám mây. ASP.NET Core được phát triển bởi tập đoàn Microsoft và cộng đồng lập trình viên. ASP.NET Core cũng là một framework kiểu module có khả năng thực thi trên framework .NET, Windows và .NET Core đa nền tảng

2.1.2 Tại sao dùng ASP.net Core .

ASP.NET Core MVC cung cấp các tính năng để xây dựng API Web và ứng dụng web như sau:

- Mô hình MVC (Model-View-Controller) giúp tạo ra các API web và ứng dụng web có khả năng thực nghiệm.
- Tích hợp hỗ trợ cho các định dạng đa dữ liệu và việc dàn xếp nội dung cho phép các API web tiếp cận nhiều khách hàng hơn, bao gồm trên các trình duyệt và thiết bị mobile khác nhau
- Ràng buộc mô hình tự động ánh xạ dữ liệu từ các yêu cầu HTTP đến các tham số phương thức hành động.
- Xác thực mô hình tự động thực thi xác thực dữ liệu phía client và server.

2.1.3 Rest full API

RESTful Web Service là các Web Service được viết dựa trên kiến trúc REST. REST đã được sử dụng rộng rãi thay thế cho các Web Service dựa trên SOAP và WSDL. RESTful Web Service nhẹ (lightweigh), dễ dàng mở rộng và bảo trì. Những khái niệm đầu tiên về REST (REpresentational State Transfer) được đưa ra vào năm 2000 trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Trong luận văn ông giới thiệu khá chi tiết về các ràng buộc, quy ước cũng như cách thức thực hiện với hệ thống để có được một hệ thống REST. REST định nghĩa các quy tắc kiến trúc để bạn thiết kế Web services, chú trọng vào tài nguyên hệ thống, bao gồm các trạng thái tài nguyên được định dạng như thế nào và được truyền tải qua HTTP, và được viết bởi nhiều ngôn ngữ khác nhau. Nếu tính theo số dịch vụ

mạng sử dụng, REST đã nổi lên trong vài năm qua như là một mô hình thiết kế dịch vụ chiếm ưu thế. Trong thực tế, REST đã có những ảnh hưởng lớn và gần như thay thế SOAP và WSDL vì nó đơn giản và dễ sử dụng hơn rất nhiều.

REST là một bộ quy tắc để tạo ra một ứng dụng Web Service, mà nó tuân thủ 4 nguyên tắc thiết kế cơ bản sau:

- 1. Sử dụng các phương thức HTTP một cách rõ ràng*
- 2. Phi trạng thái*
- 3. Hiển thị cấu trúc thư mục như các Urls*
- 4. Truyền tải JavaScript Object Notation (JSON), XML hoặc cả hai.*

CHƯƠNG 3. ĐẶC TẢ YÊU CẦU

3.1 ĐẶC TẢ VÀ YÊU CẦU ỨNG DỤNG

Những yêu cầu đặt ra là:

- Cho phép người dùng đăng kí tài khoản, đăng nhập vào cửa hàng để mua game
- Có giao diện hiển thị danh sách các game theo mục (đang Sale, Hot, v.v...) để người dùng lựa chọn, mua
- Người dùng có thể xem được những game mình đã mua, đang sở hữu
- Giao diện hiển thị chi tiết của Game để người dùng có thể xem thông tin của Game rõ hơn.
- Tìm kiếm, sắp xếp theo nội dung, giá tiền.

3.2 CÁCH THỨC HOẠT ĐỘNG CỦA ỨNG DỤNG

- Người dùng đăng kí tài khoản
- Đưa ra danh sách game hiện tại của shop.
- Người dùng chọn game, mua

- Game được đưa vào thư viện của người dùng.

CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG

4.1 XÂY DỰNG ỨNG DỤNG

4.1.1 Giới thiệu tổng quan kiến trúc chương trình

Chương trình xây dựng dựa trên 6 Activity chính để thực hiện các chức năng chính:

- Login
- Register
- Home
- Game List
- Game Detail
- Game Library
- Payment

4.1.2 Chi tiết các thành phần

Login: Cung cấp cho người dùng các trường để nhập tài khoản, mật khẩu để đăng nhập vào phần mềm. Khi đăng nhập, phần mềm gửi thông tin đăng nhập tới backend API để kiểm tra. Nếu thông tin đăng nhập đúng, API sẽ trả về kết quả true và 1 user-id của người dùng.

Register: Cung cấp cho người dùng các trường để nhập thông tin cho tài khoản cần đăng ký vào phần mềm. Khi ấn đăng ký, phần mềm gửi thông tin đăng ký tới backend API để kiểm tra. Nếu thông tin đăng ký không bị trùng, không bị lỗi định dạng, API sẽ trả về kết quả thành công. (Ngược lại sẽ gửi thông tin lỗi).

Home: Cung cấp các danh mục quảng cáo cho game như đang sale, đang hot, nhiều người chơi gần đây nhất. Các thông tin được lấy từ API. (Backend có 1 bảng chứa thông tin quảng cáo: GameID nào đang sale, sale bao nhiêu %).

Game list: Chứa một danh sách tất cả các game của cửa hàng cùng với một số thông tin như tên game, nhà sản xuất, giá. Có các chức năng cơ bản như tìm kiếm theo tên, sắp xếp theo giá, lọc game free và có phí.

Game Detail: Hiển thị thông tin chi tiết của game như hình ảnh, content, thể loại, độ tuổi người chơi.

Game Library: Hiển thị những game đã được người dùng mua.

Payment: Khi người dùng chọn một game và tiến hành mua, giao diện thanh toán sẽ được hiển thị chứa thông thanh toán của game, số tiền dư còn lại của tài khoản.

4.2 CÁC THƯ VIỆN SỬ DỤNG TRONG ỨNG DỤNG

4.2.1 *Axios*

- Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests).

4.2.1 *ImageSlider*

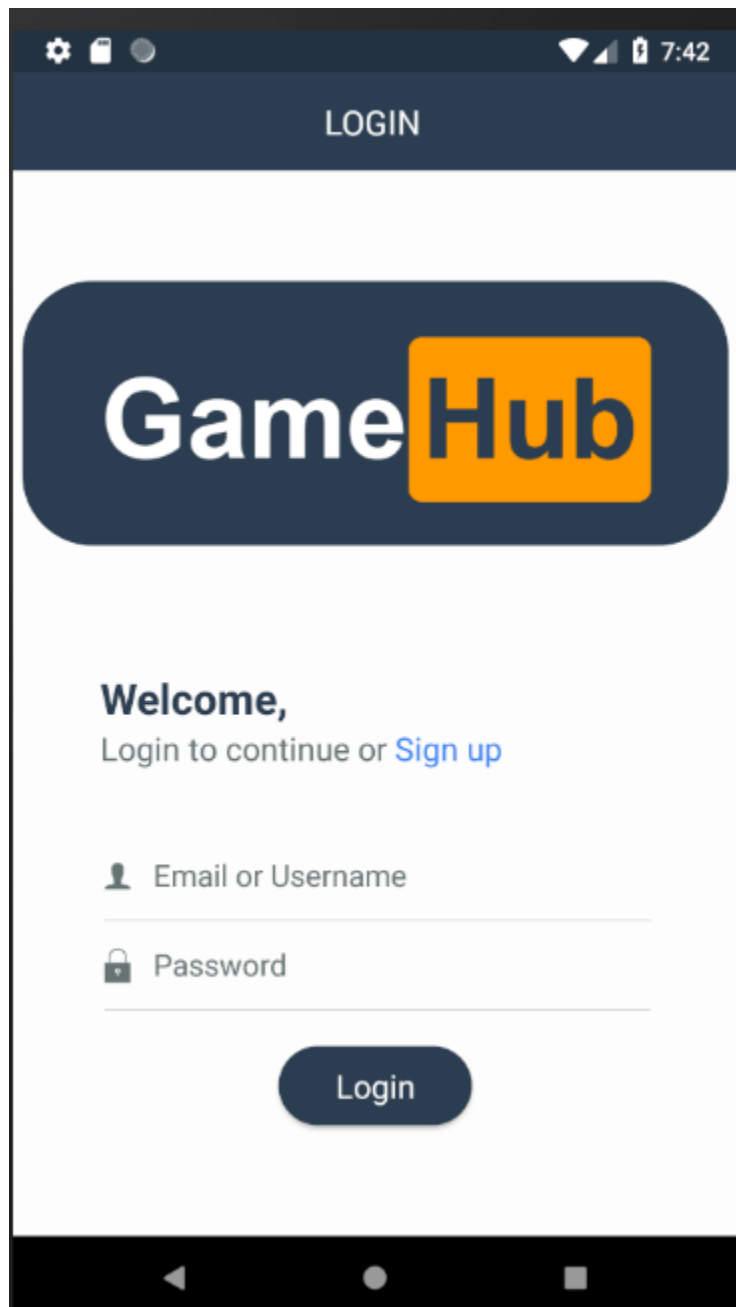
- Một thư viện cung cấp một view hỗ trợ xem ảnh dưới dạng lướt, có hỗ trợ tự động chuyển ảnh với animation.

4.2.2 *React Native Base*

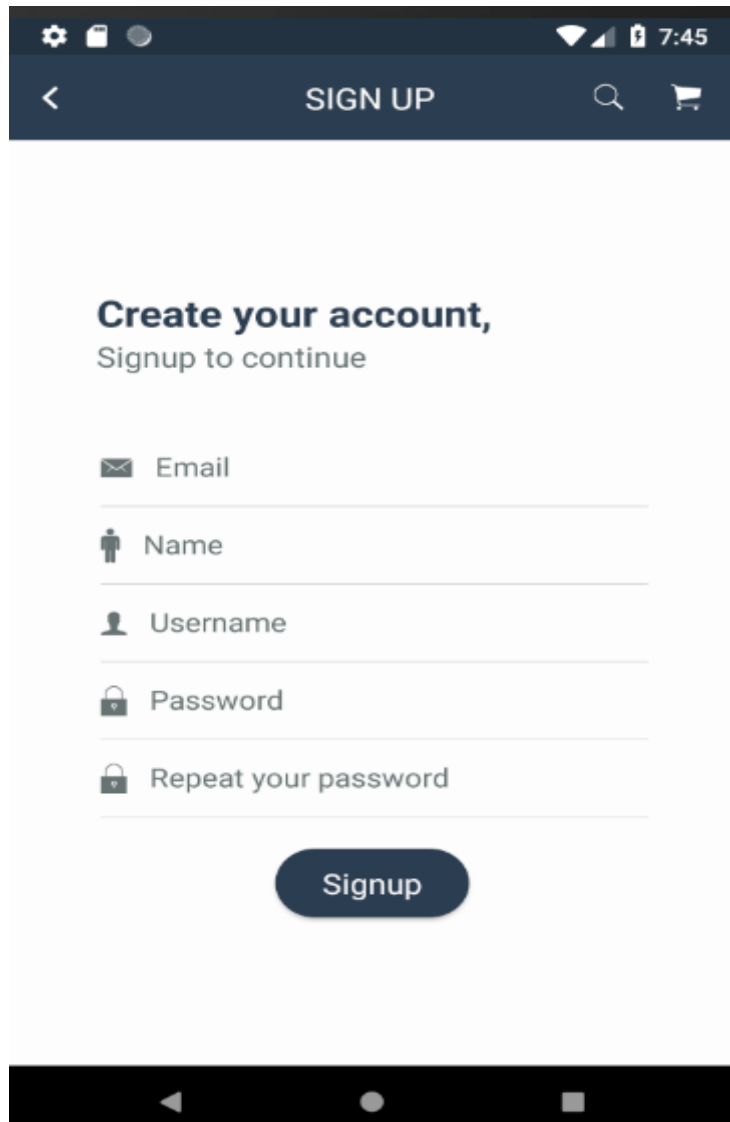
- Một thư viện hỗ trợ sẵn những component rất tiện lợi dựa trên thư viện React Native

4.3 GIAO DIỆN CHƯƠNG TRÌNH

4.3.1 Giao diện Login



4.3.2 Giao diện đăng kí



SIGN UP

Create your account,
Signup to continue

Email

Name

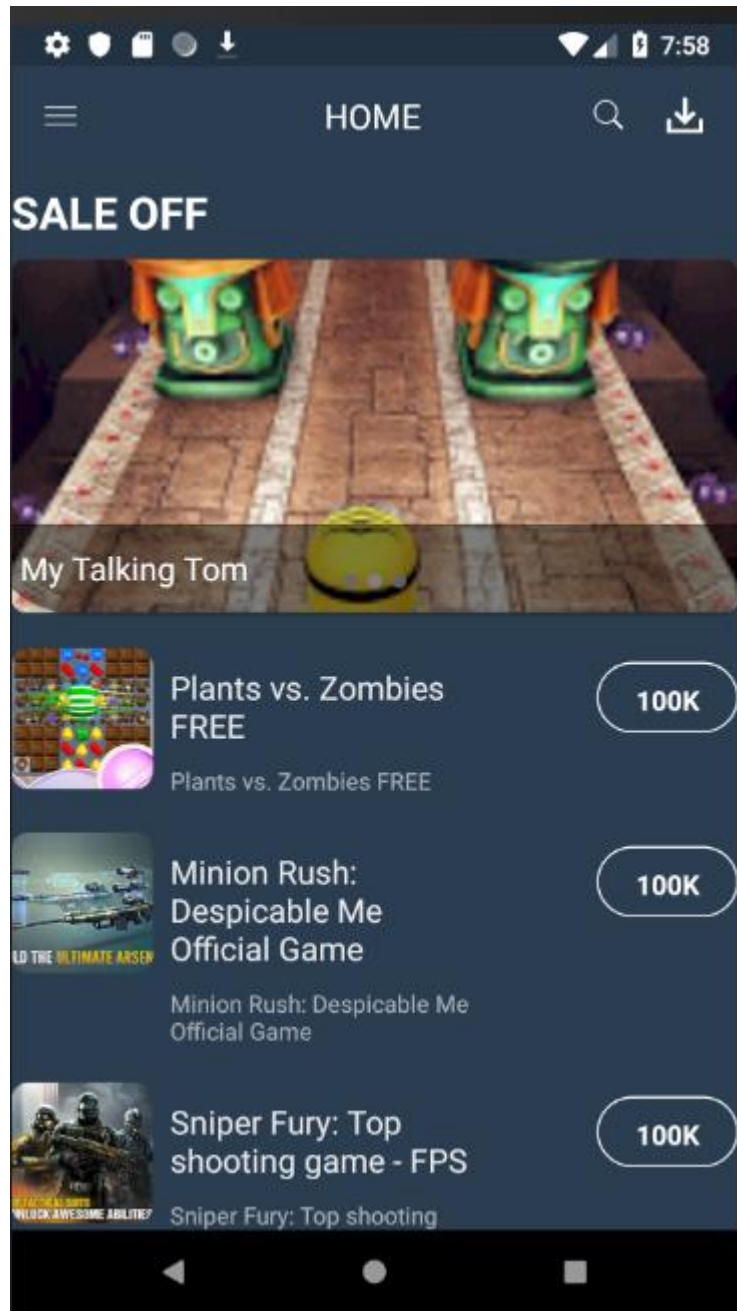
Username

Password

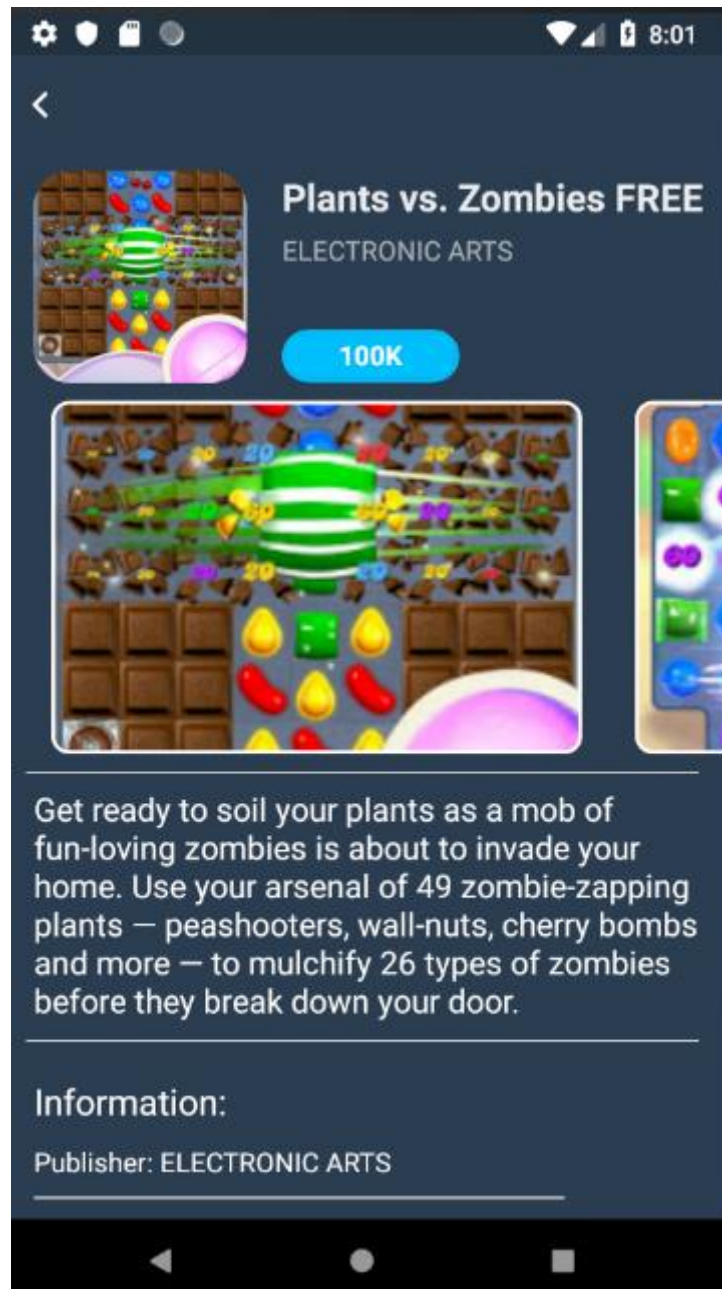
Repeat your password

Signup

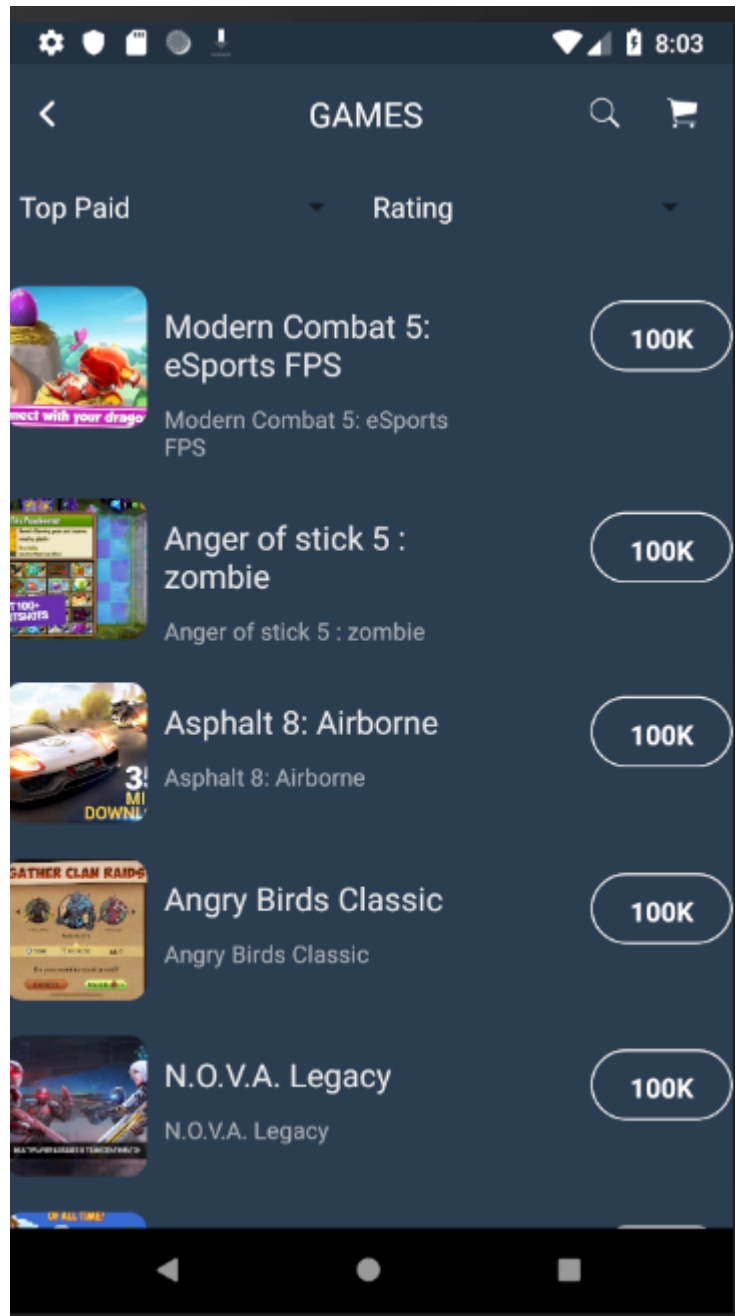
4.3.3 Giao diện Home



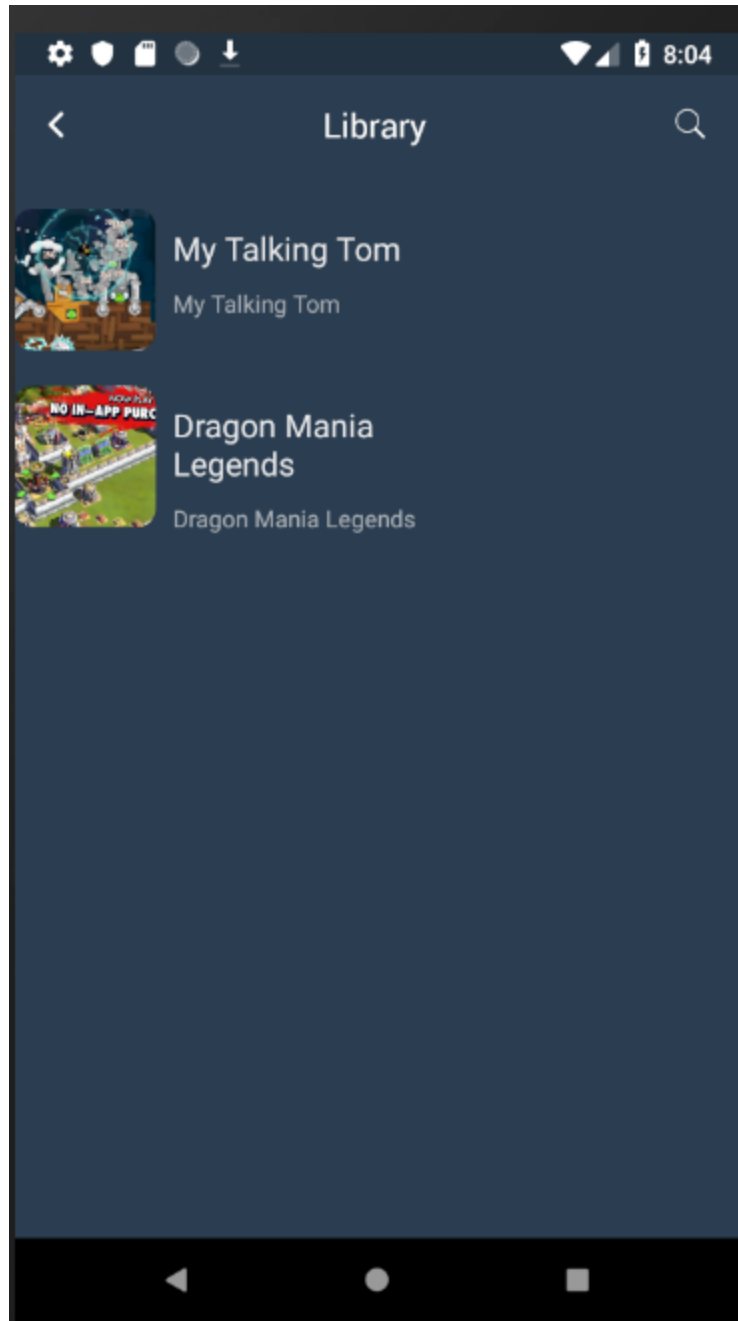
4.3.4 Giao diện Game Detail



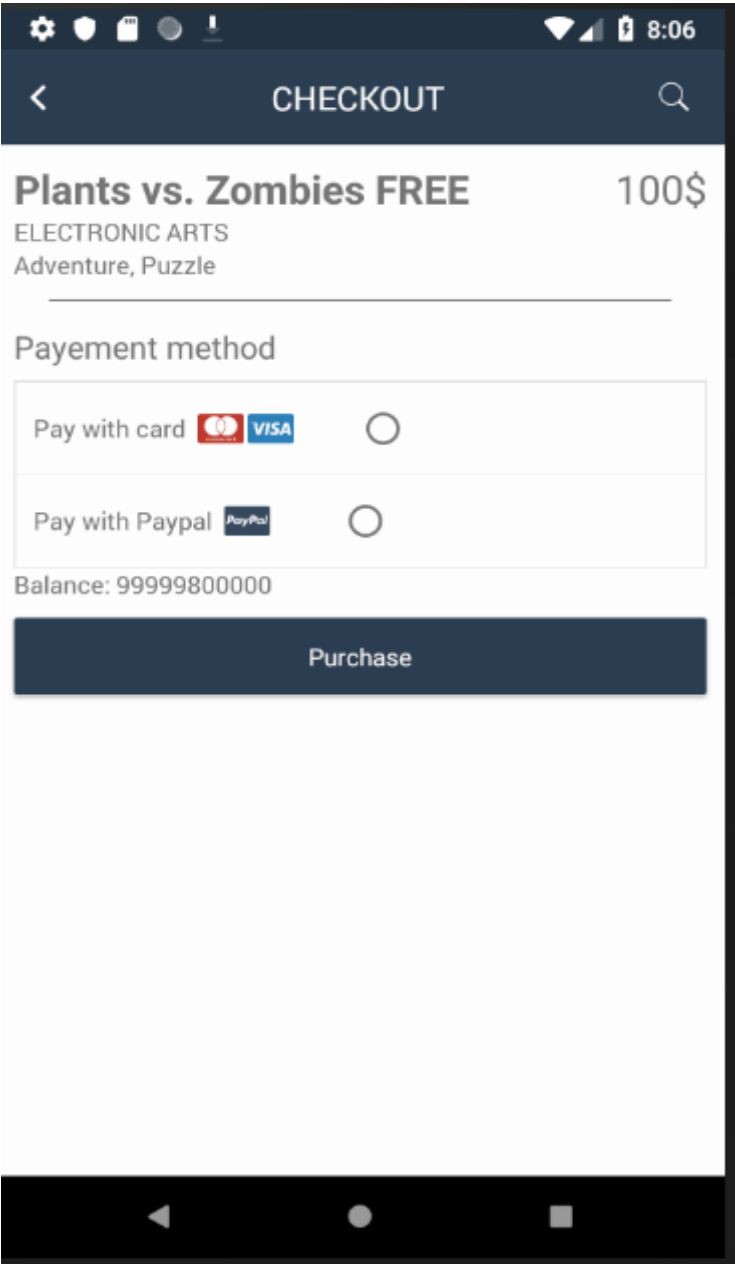
4.3.5 Giao Game List



4.3.6 Giao diện Game Library



4.3.7 Giao diện Payment



4.4 CÀI ĐẶT VÀ KIỂM THỬ

4.3.1 Môi trường

- +Phát triển dựa trên môi trường Android
- +Phiên bản android yêu cầu là Android 6.0 hay API 24 trở lên.
- +Thiết bị có cài đặt Google Service 11.2 trở lên
- +ASP.net core
- + SQL Server

4.3.2 Thử nghiệm và đánh giá kết quả

- Cài đặt Server
 - + Run Project Server với android studio 2017
 - +Gõ lệnh update-database trong console nuget để cập nhật database

4.3.3 Thử nghiệm và đánh giá kết quả

Thử nghiệm:

- Quá trình thử nghiệm diễn ra có và trục trặc nhưng đã được khắc phục.
- Các chức năng cuối cùng đã được hoàn thành và đúng như mong muốn.
- Cơ sở dữ liệu và file nén Project được sao lưu.

Đánh giá kết quả:

- Về cơ bản đã hoàn thành xong các chức năng mà yêu cầu đặt ra.
- Giao diện bắt mắt, thân thiện, dễ nhìn.
- Nhìn chung đã hoàn thành tốt ứng dụng một cách cố gắng nhất.

KẾT LUẬN VÀ HƯỚNG MỞ RỘNG

Sau thời gian dài tìm hiểu và hoàn thành đồ án, chúng em cảm thấy rất vui vì những kiến thức mới đã đạt được. Và cảm thấy vui vì những chỉ dẫn tận tình của Thầy Cô.

Đồ án nhóm chúng em nhìn chung đã hoàn thành tất cả các chức năng yêu cầu đặt tả đặt ra, nhưng đâu đó chắc chắn vẫn còn những thiếu sót, mà trong tầm nhìn hạn hẹp chúng em vẫn chưa thấy được. Nên nhóm chúng em quyết định sẽ dành thời gian tìm hiểu nâng cấp thêm nữa các chức năng bổ sung và giao diện bắt mắt hơn nữa cho ứng dụng trong thời gian sắp tới.

Hướng mở rộng sẽ là thêm tài khoản đăng nhập và lưu trữ các thông tin cá nhân của người dùng, dựa vào đó nhà phát triển có thể thông tin đến người dùng một cách nhanh chóng và đầy đủ nhất. Ngoài ra còn liên kết với các trang Web của từng nhà hàng, sẽ có chi tiết hơn nữa cho mỗi nhà hàng được chọn.

Cuối cùng nhóm em xin cảm ơn Thầy Cô đã tận tình chỉ dạy và xin hứa sẽ nghiên cứu và học tập thật tốt trong thời gian sắp tới.

MÃ NGUỒN MỞ VÀ TÀI LIỆU THAM KHẢO

[1] [*https://facebook.github.io/react-native/*](https://facebook.github.io/react-native/)

[2] [*https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2*](https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2)