**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 6](#_Toc27075759)

[1.1. Phát biểu bài toán 6](#_Toc27075760)

[1.1.1. Đặt vấn đề 6](#_Toc27075761)

[1.1.2. Mục tiêu 6](#_Toc27075762)

[1.1.3. Công cụ sử dụng 6](#_Toc27075763)

[1.1.4. Nền tảng phát triển 7](#_Toc27075764)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 8](#_Toc27075765)

[2.1. Giới thiệu hệ điều hành Android 8](#_Toc27075766)

[2.2. Giao diện Android 8](#_Toc27075767)

[2.3. Ứng dụng android 9](#_Toc27075768)

[2.4. Quản lý bộ nhớ Android 9](#_Toc27075769)

[2.5. Nhân Linux 10](#_Toc27075770)

[2.6. Giới thiệu về React Native 11](#_Toc27075771)

[2.7. Cách hoạt động của React Native 11](#_Toc27075772)

[2.8. Cài đặt 12](#_Toc27075773)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 13](#_Toc27075774)

[3.1. Kiến trúc hệ thống 13](#_Toc27075775)

[3.3. Thiết kế hệ thống 14](#_Toc27075776)

[3.3.1. Cơ sở dữ liệu 14](#_Toc27075777)

[3.3.2. Mô hình phân rã chức năng 15](#_Toc27075778)

[3.3.3. Sơ đồ Use-case 16](#_Toc27075779)

[3.3.4. Sơ đồ Sequence diagram 17](#_Toc27075780)

[Sequence diagram Login 17](#_Toc27075781)

[Sequence diagram Logout 18](#_Toc27075782)

[Sequence diagram Movie API 18](#_Toc27075783)

[Sequence diagram User Info 19](#_Toc27075784)

[Sequence diagram Recommend 19](#_Toc27075785)

[CHƯƠNG 4: HIỆN THỰC ỨNG DỤNG 20](#_Toc27075786)

[4.1. Danh sách các màn hình chính 20](#_Toc27075787)

[4.2. Giao diện các màn hình 20](#_Toc27075788)

[4.2.1. Màn hình khởi động 20](#_Toc27075789)

[4.2.2. Màn hình Home 23](#_Toc27075790)

[4.2.3. Màn hình Tính năng hiện thị thông tin phim 24](#_Toc27075791)

[4.2.4. Màn hình xem phim 25](#_Toc27075792)

[4.2.5. Màn hình Đăng nhập 26](#_Toc27075793)

[4.2.6. Màn hình Đăng nhập 27](#_Toc27075794)

[4.2.7. Màn hình Đề xuất Phim 28](#_Toc27075795)

[4.2.7. Màn hình thể loại phim 29](#_Toc27075796)

[4.2.8. Màn hình thể loại phim cùng thể loại 30](#_Toc27075797)

[4.2.9. Màn hình tìm kiếm phim 31](#_Toc27075798)

[4.3. Data Mining Trong chức năng đề xuất cho người dung 31](#_Toc27075799)

[4.3.1. Neighborhood-Based Collaborative Filtering 31](#_Toc27075800)

[4.3.2. Neighborhood-Based Collaborative Filtering 32](#_Toc27075801)

[CHƯƠNG 5: KẾT LUẬN 35](#_Toc27075802)

[5.1. Môi trường phát triển và môi trường triển khai 35](#_Toc27075803)

[5.1.1. Môi trường phát triển 35](#_Toc27075804)

[5.1.2. Môi trường triển khai 35](#_Toc27075805)

[5.2. Kết quả đạt được 35](#_Toc27075806)

[5.2.1. Kết quả 35](#_Toc27075807)

[5.2.2. Hạn chế 35](#_Toc27075808)

[5.2. Hướng phát triển 35](#_Toc27075809)

# CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## 1.1. Phát biểu bài toán

### 1.1.1. Đặt vấn đề

Với sự phát triển của các ứng dụng trong môi trường công nghệ thông tin hiện nay nói chung, đặc biệt trong môi trường di động nói riêng, ứng dụng di động đã ngày càng trở nên phổ biến và phục vụ cho con người trong mọi mặt từ giải trí, học tập, cung cấp thông tin đến kinh doanh, và là công cụ marketing hiệu quả giữa người sử dụng, người tiêu dùng và nhà sản xuất. Vì vậy việc ngày nay càng có nhiều ứng dụng được phát triển liên tục để phục vụ nhu cầu đó, làm sao để các ứng dụng ngày càng trở nên thân thiện và tiện dụng đối với người dùng là một thách thức lớn đối với nhà phát triển. Có nhiều loại ứng dụng khác nhau đa dạng về thể loại, trong đó có ứng dụng thể loại cung cấp thông tin cho người dung ngày càng được phổ biến, điển hình là ứng dụng Dự báo thời tiết.

Ta có thể để ý rằng, khi cầm một chiếc điện thoại, tablet,…với nhu cầu giải trí ngày càng nhiều và đa số các máy đều cài youtube, netfix… .Nhận thấy đây tuy là đề tài không mới nhưng tính ứng dụng của nó còn rất nhiều, và vẫn còn được mọi người sử dụng rộng rãi nên nhóm chúng em quyết định lựa chọn đề tài “Xây dựng app Movie theater” viết trên nền tảng di động sử dụng React Native.

Thông qua ứng dụng, nhóm chúng em sẽ cung cấp được các thông tin về phim dựa trên trải nghiệm và đánh giá của người dùng về những thứ họ đã xem để có thể đoán được những bộ phim mà người dùng sẽ thích. Từ đó mang đến cho người dung nhiều trải nghiệm mới thông qua việc sử dụng Data mining trong phân tích và đánh giá dữ liệu.

### 1.1.2. Mục tiêu

Xây dựng một app hoàn chỉnh để người dung có thể sử dụng và trải nghiệm các tính năng của ứng dụng.

Ứng dụng sẽ bao gồm các tính năng sau:

* Cho phép người dùng đăng nhập, đăng xuất, đăng ký
* Stop, Play, watch full screen, tua nhanh 10s, tua về 10s
* Hiển thị thông tin người dùng dã đăng ký
* Hiển thị thông tin phim, thông tin nhà phát hành
* Và cuối cùng là chức năng dựa vào thông tin đã được người dùng đánh giá để có thể giới thiệu những bộ phim mới cho người dùng

### 1.1.3. Công cụ sử dụng

Trong quá trình thực hiện, nhóm đã sử dụng một số phần mềm phục vụ cho việc tìm và xây dựng đề tài:

* Android Studio.
* Subline Text 3/ Visual Studio Code/ Atom
* Microsoft Visio

Tất cả các phần mềm trên được nhóm cài đặt và sử dụng trên hệ điều hành Microsoft Windows 10. Việc tương thích các phần mềm trên với điều kiện các hệ điều hành khác không nằm trong phạm vi nghiên cứu của đề tài này.

### 1.1.4. Nền tảng phát triển

Vì là môn phát tiển ứng dụng trên thiết bị di động nên ứng dụng được viết trên nền tảng Android và iOS sử dụng framework React Native được phát triển bới đội ngũ Facebook.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Giới thiệu hệ điều hành Android

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 cầm 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.

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 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. 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.

## 2.2. Giao diện Android

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng. Những thiết bị phần cứng bên trong như gia tốc kế, con quay hồi chuyển và cảm biến khoảng cách được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị, giống như đang điều khiển vô lăng.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hính chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẩu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung theoe tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ mà không cần phải mở ứng dụng gọi điện ra. Thông báo sẽ luôn nằm đó cho đến khi người dùng đã đọc hoặc xóa nó đi.

## 2.3. Ứng dụng 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. 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.

## 2.4. Quản lý bộ nhớ Android

Vì các thiết bị Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa mức tiêu thụ điện năng, trái với hệ điều hành máy tính để bàn luôn cho rằng máy tính sẽ có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động ngưng nó trong bộ nhớ - trong khi ứng dụng về mặt kỹ thuật vẫn "mở", những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào (như năng lượng pin hay năng lượng xử lý) và nằm đó cho đến khi nó được cần đến. Cách làm như vậy có lợi kép là vừa làm tăng khả năng phản hồi nói chung của thiết bị Android, vì ứng dụng không nhất phải đóng rồi mở lại từ đầu, vừa đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

Android quản lý các ứng dụng trong bộ nhớ một cách tự động: khi bộ nhớ thấp, hệ thống sẽ bắt đầu diệt ứng dụng và tiến trình không hoạt động được một thời gian, sắp theo thời điểm cuối mà chúng được sử dụng (tức là cũ nhất sẽ bị tắt trước). Tiến trình này được thiết kế ẩn đi với người dùng, để người dùng không cần phải quản lý bộ nhớ hoặc tự tay tắt các ứng dụng. Tuy nhiên, sự che giấu này của hệ thống quản lý bộ nhớ Android đã dẫn đến sự thịnh hành của các ứng dụng tắt chương trình của bên thứ ba trên cửa hàng Google Play; những ứng dụng kiểu như vậy được cho là có hại nhiều hơn có lợi.

## 2.5. Nhân Linux

Android có một hạt nhân dựa trên nhân Linux phiên bản 2.6, kể từ Android 4.0 Ice Cream Sandwich (bánh ngọt kẹp kem) trở về sau, là phiên bản 3.x, với middleware, thư viện và API viết bằng C, còn phần mềm ứng dụng chạy trên một nền tảng ứng dụng gồm các thư viện tương thích với Java dựa trên Apache Harmony. Android sử dụng máy ảo Dalvik với một trình biên dịch động để chạy 'mã dex' (Dalvik Executable) của Dalvik, thường được biên dịch sang Java bytecode. Nền tảng phần cứng chính của Android là kiến trúc ARM. Người ta cũng hỗ trợ x86 thông qua dự án Android x86, và Google TV cũng sử dụng một phiên bản x86 đặc biệt của Android.

Một số tính năng cũng được Google đóng góp ngược vào nhân Linux, đáng chú ý là tính năng quản lý nguồn điện có tên wakelock, nhưng bị những người lập trình chính cho nhân từ chối vì họ cảm thấy Google không có định sẽ tiếp tục bảo trì đoạn mã do họ viết. Google thông báo vào tháng 4 năm 2010 rằng họ sẽ thuê hai nhận viên để làm việc với cộng đồng nhân Linux, nhưng Greg KroahHartman, người bảo trì nhân Linux hiện tại của nhánh ổn định, đã nói vào tháng 12 năm 2010 rằng ông ta lo ngại rằng Google không còn muốn đưa những thay đổi của mình vào Linux dòng chính nữa. Một số lập trình viên Android của Google tỏ ý rằng "nhóm Android thấy chán với quy trình đó," vì nhóm họ không có nhiều người và có nhiều việc khẩn cấp cần làm với Android hơn.

Vào tháng 8 năm 2011, Linus Torvalds rằng "rốt cuộc thì Android và Linux cũng sẽ trở lại với một bộ nhân chung, nhưng điều đó có thể sẽ không xảy ra trong 4 hoặc 5 năm nữa". Vào tháng 12 năm 2011, Greg Kroah-Hartman thông báo kích hoạt Dự án Dòng chính Android, nhắm tới việc đưa một số driver, bản vá và tính năng của Android ngược vào nhân Linux, bắt đầu từ Linux 3.3. Linux cũng đưa tính năng autosleep (tự nghỉ hoạt động) và wakelocks vào nhân 3.5, sau nhiều nỗ lực phối trộn trước đó. Tương tác thì vẫn vậy nhưng bản hiện thực trên Linux dòng chính cho phép hai chế độ nghỉ: bộ nhớ (dạng nghỉ truyền thống mà Android sử dụng), và đĩa (là ngủ đông trên máy tính để bàn). Việc trộn sẽ hoàn tất kể từ nhân 3.8, Google đã công khai kho mã nguồn trong đó có những đoạn thử nghiệm đưa Android về lại nhân 3.8.

Bộ lưu trữ flash trên các thiết bị Android được chia thành nhiều phân vùng, như "system" dành cho hệ điều hành và "/data" dành cho dữ liệu người dùng và cài đặt ứng dụng. Khác với các bản phân phối Linux cho máy tính để bàn, người sở hữu thiết bị Android không được trao quyền truy cập root vào hệ điều hành và các phân vùng nhạy cảm như /system được thiết lập chỉ đọc. Tuy nhiên, quyền truy cập root có thể chiếm được bằng cách tận dụng những lỗ hổng bảo mật trong Android, điều mà cộng đồng mã nguồn mở thường xuyên sử dụng để nâng cao tính năng thiết bị của họ, kể cả bị những người ác ý sử dụng để cài virus và phần mềm ác ý. Việc Android có được xem là một bản phân phối Linux hay không vẫn còn là vấn đề gây tranh cãi, tuy được Linux Foundation và Chris DiBona, trưởng nhóm mã nguồn mở Google, ủng hộ. Một số khác, như linux-magazine.com thì không đồng ý, do Android không không hỗ trợ nhiều công cụ GNU, trong đó có glibc.

## 2.6. Giới thiệu về React Native

React Native được phát triển bởi Facebook với mục đính 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 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.

React Native giúp cho lập trình viên web có thể viết ứng dụng native để khắc phục các điểm yếu của ứng dụng web và hybrid. Chỉ với một lập trình viên thành thạo javascript, bạn có thể chiến đấu trên mọi mặt trận web, desktop, server và bây giờ là mobile. Điều này không những có lợi cho lập trình viên web mà nó giúp cho các doanh nghiệp phát triển sản phẩm đầu cuối với ít nhân lực hơn.

Khi xây dựng React Native, điều tuyệt vời là đã được tích hợp tính năng Live Reload – tương tự tính năng Hot Replacement Module trong Webpack. Tính năng Live Reload khác tính năng Reload, trong khi Live Reload chỉ tải lại chức năng/tập tin nào thay đổi, thì Reload sẽ tải lại toàn bộ mã nguồn. Ngoài ra, bạn cũng dễ dàng debug javascript trong Chrome và Safari. Đối với những lỗi thuộc Native thì phải cần đến XCode cho iOS hoặc Android Studio cho Android.

## 2.7. Cách hoạt động của React Native

Ứng dụng viết bằng React Native được chia làm 2 phần: phần view(hiển thị) và phần xử lý.

Phần hiển thị được biên dịch từ javascript sẽ map với những component của hệ thống ví dụ: điều hướng, tab, touch…Phần view này được lấy cảm hứng từ Virtual DOM của React JS, mọi xử lý view được thực hiện trên một cây DOM ảo, sau đó được React Native render lại bằng native view.

Phần xử lý vẫn được thực hiện trực tiếp bằng ngôn ngữ javascript: ví dụ “1+1=2” :)), biểu thức này được xử lý dưới bộ core thực thi Javascript, không phải thông dịch qua Java hay Swift/Objective-C rồi mới làm phép tính đâu nhé.

**KHUYẾT ĐIỂM CỦA REACT NATIVE**

React Native là một giải pháp tuyệt vời cho phát triển ứng dụng trên điện thoại di động, tuy nhiên đến thời điểm hiện tại, vẫn còn tồn tại một số khuyết điểm:

* Vẫn còn thiếu các component view cho Android: Map, Modal, Slider, Spinner hoặc các module như Camera Roll, Media, PushNotificationIOS.
* Không hỗ trợ Window Phone: đây sẽ là điểm trừ lớn nếu so với ứng dụng hybrid. Tuy nhiên, nếu bạn chỉ tập trung cho iOS và Android thì tất nhiên đây không phải là vấn đề lớn.
* Không build được ứng dụng iOS trên Window và Linux: tất nhiên đây không phải là điểm yếu kém của bản thân React Native, mà đó là do yêu cầu từ Apple.
* React Native không thể build được ứng dụng “quá phức tạp” nếu bạn không biết Swift/Objecive-C, Java – tính phức tạp ở đây là ứng dụng của bạn cần phải chỉnh sửa các component. Bạn nên nhớ là để viết được 1 ứng dụng native bằng javascript “luôn luôn” có sẵn các component đã được viết từ Swift/Objective-C (iOS) và Java (Android) với tính chất 1-1. Trường hợp bạn muốn chỉnh sửa 1 component nào đó: thay đổi thành phần hoặc thêm API thì bạn phải “tự viết” bằng chính ngôn ngữ tương ứng của iOS hoặc Android. Tin vui cho bạn là cũng nhiều lập trình viên khác đã viết nhiều component cần thiết cho hầu hết ứng dụng (đây cũng là lý do vì sao Facebook biến React Native thành mã nguồn mở).
* Không nên dùng để viết game có tính đồ họa và cách chơi phức tạp.
* Dùng ES2015/ES6: đây là cấu trúc mới cho Javascript từ 2015, vì khá là mới nên những cấu trúc của nó có thể bạn chưa quen, dẫn tới việc khó khăn trong việc tiếp cận.
* Bạn có thể phải mất nhiều thời gian để theo hết hệ sinh thái của React, tuy nhiên tất cả là để làm cho đời lập trình viên fullstack đẹp đẽ và những sản phẩm chất lượng hơn, bao gồm: React Js, React Native, Flux, Relay, GraphQL. Bạn không cần phải quan tâm những thứ này chỉ để viết ứng dụng với React native.

## 2.8. Cài đặt

Yêu cầu:

* Mac OS X (để biên dịch iOS), nếu chỉ biên dịch Android thì có thể sử dụng Window hoặc Linux
* XCode 7.0+
* Android Studio/Vistual Studio Code/ Atome,…
* JDK 1.8
* Node.js 4.0+
* Khác:
* Watchman (để view error)
* Flow

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. Kiến trúc hệ thống

A close up of text on a white background

Description automatically generated

## 3.3. Thiết kế hệ thống

### 3.3.1. Cơ sở dữ liệu

A sign on the side of a building

Description automatically generated

### 3.3.2. Mô hình phân rã chức năng

A close up of text on a white background

Description automatically generated

### 3.3.3. Sơ đồ Use-case

A close up of a map

Description automatically generated

### 3.3.4. Sơ đồ Sequence diagram

### Sequence diagram Login

A close up of a map

Description automatically generated

Các thành phần trong hệ thống trong chức năng user: Giao diện Login, Rest API Service.

Khái quát các bước trong chức năng đăng nhập dựa vào sơ đồ:

- Người dùng: Người dùng chọn chức năng đăng nhập.

- Ứng dụng cụ thể là giao diện đăng nhập: các field Email, password và nút login user tiến hành nhập email và password sau đó nhấn nút login. Nếu thành công chuyển về giao diện Home screen. Ngược lại hiện popup thông báo “Bạn đã sai email hoặc mật khẩu”.

- Rest API service : Từ client gửi lên thông tin username và password lên server, server kiểm tra thông tin user thông qua hàm validation nếu tồn tại user thì tiến hành tạo token và trả thông tin token cho client và client lưu lại thông tin token vào local storge để dùng sau này ngược lại nếu user không tồn tại thì tiền hành trả về 404 not found.

### Sequence diagram Logout

A screenshot of a cell phone

Description automatically generated

Các thành phần trong hệ thống trong chức năng user: Giao diện Login, Rest API Service.

Khái quát các bước trong chức đăng xuất :

- Người dùng: Người dùng mở navigation chọn vào nút logout

- Client: gửi thông tin token cho server nếu server trả về thông tin huỷ thành công thì xoá token trong local storge.

- Rest API Service: nhận thông tin token được gửi từ client sau đó huỷ token thông qua hàm remove token.

### Sequence diagram Movie API

A screenshot of a cell phone

Description automatically generated

Các thành phần trong hệ thống trong chức năng Movie API: Tất cả các giao diện có liên quan đến phim.

Khái quát các bước trong chức năng dựa vào sơ đồ:

- Người dùng: người dùng mở các screen có chưa movie.

- Ứng dụng cụ thể là giao diện chính: load dữ liệu lên màn hình

- Rest API: trả về thông tin mà client yêu cầu với phương thức GET

### Sequence diagram User Info

A close up of a map

Description automatically generated

Khái quát các bước trong dung chức năng thông tin người dùng dựa vào sơ đồ:

- Người dùng: người dùng bấm vào button unifor ở navigation.

- Ứng dụng cụ thể là giao diện chính: load thông tin dữ liệu người dùng lên màn hình

- Rest API: tìm user và trả về user cho client

### Sequence diagram Recommend

A close up of a map

Description automatically generated

Khái quát các bước trong dung chức năng recommend dựa vào sơ đồ:

- Người dùng:sau khi người dùng đăng nhập thành công và chuyển về trang home screen.

- Ứng dụng cụ thể là giao diện chính: gửi thông tin user lên server và nhận lại data là một danh sách list phim và thêm flatlist này vào màn hình home screen

- Rest API: server tìm user trong model lấy ra id của user sau đó tra với danh sách đề cử phim cho user đó ở file excel đã được training trước đó. Và gửi trả lại danh sách phim cho client.

# CHƯƠNG 4: HIỆN THỰC ỨNG DỤNG

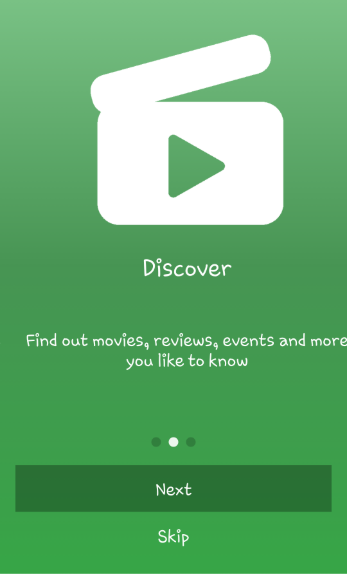
## 4.1. Danh sách các màn hình chính

## 4.2. Giao diện các màn hình

### 4.2.1. Màn hình khởi động

A close up of a sign

Description automatically generated



A close up of a sign

Description automatically generated

Mục đích: Hiển thị lên màn hình khởi động để chuẩn bị chuyển sang màn hình Home các chức năng chính.

Điều hướng và tương tác người dung:

Sau ba slide người dùng sẽ kết thúc bằng cach nhấn vào button done và chuyển về giao diện home sceen.

### 4.2.2. Màn hình Home

A screenshot of a cell phone

Description automatically generated

Mục đích: Hiển thị lên danh các phim

Điều hướng và tương tác người dùng: Khi bấm vào button

(1) Sẽ reload lại phim.

(2) sẽ load thêm phim.

(3) danh sách button trong navigation

Tùy theo người dung ấn chọn bào button chức năng nào sẽ chuyển đến giao diện hoặc thực thi chức năng

### 4.2.3. Màn hình Tính năng hiện thị thông tin phim

A screen shot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Mục đích: Hiển thị lên danh các phim

Điều hướng và tương tác người dùng: Khi bấm vào button

(1) trở về trang trước đó.

(2) đi đến trang phim có cùng thể loại.

(3) show more thông tin phim

(4) đi đến trang phim

Tùy theo người dung ấn chọn bào button chức năng nào sẽ chuyển đến giao diện hoặc thực thi chức năng

### 4.2.4. Màn hình xem phim



### 4.2.5. Màn hình Đăng nhập

A screenshot of a cell phone

Description automatically generated

Mục đích: đăng nhập

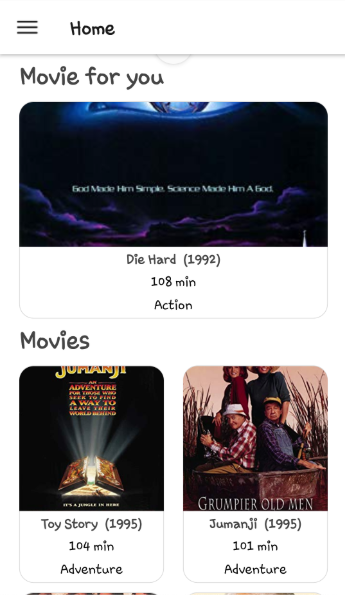
### 4.2.6. Màn hình Đăng nhập

A screenshot of a cell phone

Description automatically generated

Mục đích: đăng nhập

### 4.2.7. Màn hình Đề xuất Phim



### 4.2.7. Màn hình thể loại phim

A picture containing screenshot

Description automatically generated

### 4.2.8. Màn hình thể loại phim cùng thể loại



### 4.2.9. Màn hình tìm kiếm phim

A screenshot of a cell phone

Description automatically generated

## 4.3. Data Mining Trong chức năng đề xuất cho người dung

### 4.3.1. Neighborhood-Based Collaborative Filtering

Ý tưởng cơ bản của NBCF là xác định mức độ quan tâm của một user tới một item dựa trên các users khác gần giống với user này. Việc gần giống nhau giữa các users có thể được xác định thông qua mức độ quan tâm của các users này tới các items khác mà hệ thống đã biết. Ví dụ, A, B đều thích phim Cảnh sát hình sự, tức đều rate bộ phim này 5 sao. Ta đã biết A cũng thích Người phán xử, vậy nhiều khả năng B cũng thích bộ phim này.

Các bạn có thể đã hình dung ra, hai câu hỏi quan trọng nhất trong một hệ thống Neighborhood-based Collaborative Filtering là:

* Làm thế nào xác định được sự giống nhau giữa hai users?
* Khi đã xác định được các users gần giống nhau (similar users) rồi, làm thế nào dự đoán được mức độ quan tâm của một user lên một item?

Việc xác định mức độ quan tâm của mỗi user tới một item dựa trên mức độ quan tâm của similar users tới item đó còn được gọi là User-user collaborative filtering. Có một hướng tiếp cận khác được cho là làm việc hiệu quả hơn là Item-item collaborative filtering. Trong hướng tiếp cận này, thay vì xác định user similarities, hệ thống sẽ xác định item similarities. Từ đó, hệ thống gợi ý những items gần giống với những items mà user có mức độ quan tâm cao.

### 4.3.2. Chuẩn hoá dữ liệu

A screenshot of a cell phone

Description automatically generated

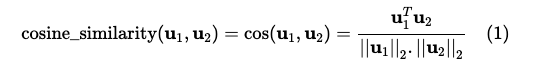
**Hình** **2a)** là giá trị trung bình của ratings cho mỗi user. Giá trị cao tương ứng với các user dễ tính và ngược lại. Khi đó, nếu tiếp tục trừ từ mỗi rating đi giá trị này và thay các giá trị chưa biết bằng 0, ta sẽ được normalized utility matrix như trong **Hình 2b).** Bạn có thể thắc mắc tại sao bước chuẩn hoá này lại quan trọng, câu trả lời ở ngay đây:

* Việc trừ đi trung bình cộng của mỗi cột khiến trong trong mỗi cột có những giá trị dương và âm. Những giá trị dương tương ứng với việc user thích item, những giá trị âm tương ứng với việc user không thích item. Những giá trị bằng 0 tương ứng với việc chưa xác định được liệu user có thích item hay không.
* Về mặt kỹ thuật, số chiều của utility matrix là rất lớn với hàng triệu users và items, nếu lưu toàn bộ các giá trị này trong một ma trận thì khả năng cao là sẽ không đủ bộ nhớ. Quan sát thấy rằng vì số lượng ratings biết trước thường là một số rất nhỏ so với kích thước của utility matrix, sẽ tốt hơn nếu chúng ta lưu ma trận này dưới dạng sparse matrix, tức chỉ lưu các giá trị khác không và vị trí của chúng. Vì vậy, tốt hơn hết, các dấu ‘?’ nên được thay bằng giá trị ‘0’, tức chưa xác định liệu user có thích item hay không. Việc này không những tối ưu bộ nhớ mà việc tính toán similarity matrix sau này cũng hiệu quả hơn.

Sau khi đã chuẩn hoá dữ liệu như trên, một vài similiraty function thường được sử dụng là:

**Cosine Similarity:**

Đây là hàm được sử dụng nhiều nhất, và cũng quen thuộc với các bạn nhất. Nếu các bạn không nhớ công thức tính coscos của góc giữa hai vector u1,u2u1,u2 trong chương trình phổ thông, thì dưới đây là công thức:



Trong đó u1,2u1,2 là vectors tương ứng với users 1, 2 đã được chuẩn hoá như ở trên.

Độ similarity của hai vector là 1 số trong đoạn [-1, 1]. Giá trị bằng 1 thể hiện hai vector hoàn toàn similar nhau. Hàm số coscos của một góc bằng 1 nghĩa là góc giữa hai vector bằng 0, tức một vector bằng tích của một số dương với vector còn lại. Giá trị coscos bằng -1 thể hiện hai vector này hoàn toàn trái ngược nhau. Điều này cũng hợp lý , tức khi hành vi của hai users là hoàn toàn ngược nhau thi similarity giữa hai vector đó là thấp nhất.

Ví dụ về cosine\_similarity của các users trong **Hình 2b)** được cho trong **Hình 2c).** Similarity matrix SS là một ma trận đối xứng vì coscos là một hàm chẵn, và nếu user A giống user B thì điều ngược lại cũng đúng. Các ô màu xanh trên đường chéo đều bằng 1 vì đó là coscos của góc giữa 1 vector và chính nó, tức cos(0)=1cos(0)=1. Khi tính toán ở các bước sau, chúng ta không cần quan tâm tới các giá trị 1 này. Tiếp tục quan sát các vector hàng tương ứng với u0,u1,u2u0,u1,u2, chúng ta sẽ thấy một vài điều thú vị:

* u0 gần với u1 và u5 (độ giống nhau là dương) hơn các users còn lại. Việc similarity cao giữa u0 và u1 là dễ hiểu vì cả hai đều có xu hướng quan tâm tới i0,i1,i2i0,i1,i2 hơn các items còn lại. Việc u0 gần với u5 thoạt đầu có vẻ vô lý vì u5 đánh giá thấp các items mà u0 đánh giá cao (Hình 2a)); tuy nhiên khi nhìn vào ma trận utility đã chuẩn hoá ở Hình 2b), ta thấy rằng điều này là hợp lý. Vì item duy nhất mà cả hai users này đã cung cấp thông tin là i1i1 với các giá trị tương ứng đều là tích cực.
* u1 gần với u0 và xa các users còn lại.
* u2 gần với u3,u4,u5,u6u3,u4,u5,u6 và xa các users còn lại.

Từ similarity matrix này, chúng ta có thể phân nhóm các users ra làm hai nhóm (u0,u1)(u0,u1) và (u2,u3,u4,u5,u6)(u2,u3,u4,u5,u6). Vì ma trận SS này nhỏ nên chúng ta có thể dễ dàng quan sát thấy điều này; khi số users lớn hơn, việc xác định bằng mắt thường là không khả thi. Việc xây dựng thuật toán phân nhóm các users (users clustering) rất có thể sẽ được trình bày ở một trong các bài viết tiếp theo.

Có một chú ý quan trọng ở đây là khi số lượng users lớn, ma trận SS cũng rất lớn và nhiều khả năng là không có đủ bộ nhớ để lưu trữ, ngay cả khi chỉ lưu hơn một nửa số các phần tử của ma trận đối xứng này. Với các trường hợp đó, mới mỗi user, chúng ta chỉ cần tính và lưu kết quả của một hàng của similarity matrix, tương ứng với việc độ giống nhau giữa user đó và các users còn lại.

Trong bài viết này, tôi sẽ sử dụng similarity function này.

**Rating prediction:**

Công thức phổ biến được sử dụng để dự đoán rating của u cho i là:

A screenshot of a cell phone

Description automatically generated

Một ví dụ về việc tính normalized rating của u1 cho i1 được cho trong Hình 2e) với số nearest neighbors là k=2. Các bước thực hiện là:

1. Xác định các users đã rated i1, đó là u0,u3,u5u0,u3,u5.
2. Xác định similarities của u1 với các users này ta nhận được 0.83,−0.40,−0.230.83,−0.40,−0.23. Hai (k=2) giá trị lớn nhất là 0.830.83 và −0.23−0.23 tương ứng với u0 và u5.
3. Xác định các normalized ratings của u0,u5u0,u5 cho i1, ta thu được hai giá trị lần lượt là 0.750.75 và 0.50.5.
4. Dự đoán kết quả:

A picture containing object

Description automatically generated

Việc quy đổi các giá trị ratings đã chuẩn hoá về thang 5 có thể được thực hiện bằng cách cộng các cột của ma trận Y^ với giá trị rating trung bình của mỗi user như đã tính trong Hình 2a).

Việc hệ thống quyết định recommend items nào cho mỗi user có thể được xác định bằng nhiều cách khác nhau. Có thể sắp xếp unrated items theo thứ tự tự lớn đến bé của các predicted ratings, hoặc chỉ chọn các items có normalized predicted ratings dương - tương ứng với việc user này có nhiều khả năng thích hơn.

# CHƯƠNG 5: KẾT LUẬN

## 5.1. Môi trường phát triển và môi trường triển khai

### 5.1.1. Môi trường phát triển

Hệ điều hành: Microsoft Windows 10.

Môi trường:

* Android 8
* NodeJS: v8.12.0
* React: 16.3.1
* React-native: 0.55.4
* Dữ liệu thời tiết: Sử dụng API của trang OpenWeather.com

Công cụ phát triển: Android Studio/ Visual Studio Code/ Syblime Text 3.

Nền tảng: App React native chạy trên cả Android và iOS

### 5.1.2. Môi trường triển khai

Android 5.0 trở lên.

## 5.2. Kết quả đạt được

### 5.2.1. Kết quả

App Movie Theater không là hướng đi mới, tuy nhiên nó vẫn còn được nhiều người sử dụng, chứng tỏ mức quan trọng của app trên thiết bị di động.

### 5.2.2. Hạn chế

Vì thời gian thực hiện và nghiên cứu còn giới hạn, và do yếu tố chủ quan từ nhóm nên App hiện tại vẫn chưa đẹp về hình ảnh, một số chức năng còn chưa thực hiện xong, cũng như chưa cải tiến được cho phù hợp với thị hiếu người dung.

## 5.2. Hướng phát triển

Như đã trình bày, từ những thiếu sót chưa thực hiện được, nhóm chúng em sẽ cải tiến và hoàn thành đầy đủ các chức năng của app. Cải tiến giao diện cho đẹp mắt và thu hút. Thêm một số tính năng khác như: Tìm chính xác hơn vị trí để cung cấp thời tiết hiện tại, giải quyết các vấn đề đa tiến trình async task cho hiệu quả, cải thiện hiệu năng chạy.