

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG
CHO THIẾT BỊ DI ĐỘNG

CHỦ ĐỀ 5: ỨNG DỤNG THƯƠNG MẠI ĐIỆN TỬ

ĐỀ TÀI 103 : XÂY DỰNG ỨNG DỤNG ĐẶT VÉ XEM PHIM
TRÊN NỀN TẢNG ANDROID SỬ DỤNG ANDROID STUDIO

Sinh viên thực hiện	Lớp	Khóa
Bùi Minh Tùng	DCCNTT.12.10.9	K12
Đặng Văn Tùng	DCCNTT.12.10.9	K12
Nguyễn Duy Vũ	DCCNTT.12.10.9	K12
Vũ Duy Tùng	DCCNTT.12.10.9	K12
Lê Quang Khánh Nguyên	DCCNTT.12.10.9	K12

Bắc Ninh, năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

**HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG
CHO THIẾT BỊ DI ĐỘNG**

Nhóm 4

CHỦ ĐỀ 5: ỨNG DỤNG THƯƠNG MẠI ĐIỆN TỬ

Đề tài 103: Xây dựng ứng dụng đặt vé xem phim
trên nền tảng android sử dụng android studio

STT	Sinh viên thực hiện	Mã sinh viên	Điểm bằng số	Điểm bằng chữ
1	Bùi Minh Tùng	20212543		
2	Đặng Văn Tùng	20212588		
3	Nguyễn Duy Vũ	20212607		
4	Vũ Duy Tùng	20212450		
5	Lê Quang Khánh Nguyên	20212683		

CÁN BỘ CHẤM 1

(Ký và ghi rõ họ tên)

CÁN BỘ CHẤM 2

(Ký và ghi rõ họ tên)

Bắc Ninh, năm 2024

MỤC LỤC

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC CÁC BẢNG BIỂU

LỜI NÓI ĐẦU	1
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	2
1.1. Tổng quan về ứng dụng cho thiết bị di động	2
1.1.1. Xu hướng phát triển ứng dụng cho thiết bị di động.....	2
1.1.2. Tổng quan về các hệ điều hành thiết bị di động, các công cụ lập trình.	3
1.1.3. Hệ điều hành Android	4
1.1.4. Tổng quan về Android Studio	5
1.2. Phát triển ứng dụng trên nền tảng Android sử dụng Java	5
1.2.1. Cấu trúc ứng dụng Android	6
1.2.2. Các thành phần phát triển ứng dụng Android	6
1.3. Giới thiệu về Firebase	7
1.3.1. Firebase là gì?	7
1.3.2. Lịch sử hình thành và phát triển nền tảng Firebase	7
1.3.3. Firebase thuộc sở hữu của ai?	8
1.3.4. Các tính năng chính của Firebase	8
1.3.5. Chức năng của Firebase	11
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	15
2.1. Xác định yêu cầu bài toán.....	15
2.2. Phân tích hệ thống.....	17
2.2.2. Danh sách các tác nhân	18
2.3. Thiết kế chương trình của hệ thống	20
2.3.1. Đặc tả use case tổng quát của hệ thống	20
2.3.2. Đặc tả use case chức năng “Quản lý tài khoản”	21
2.3.2.1. Biểu đồ use case chức năng “Quản lý tài khoản.....	21
2.3.2.2. Biểu đồ hoạt động chức năng “Quản lý tài khoản”	22
2.3.2.3. Đặc tả use case chức năng “Quản lý tài khoản”	23

2.3.3. Đặc tả use case chức năng “Đặt vé”	24
2.3.2.1. Sơ đồ use case chức năng “Đặt vé”	24
2.3.2.2. Sơ đồ hoạt động chức năng “Đặt vé”	25
2.3.3.3. Đặc tả use case chức năng “Đặt vé”	26
2.3.4. Đặc tả use case chức năng “Quản lý đặt vé”	27
2.3.4.1. Sơ đồ Use Case chức năng “Quản lý đặt vé”	27
2.3.4.2. Sơ đồ hoạt động chức năng “Quản lý đặt vé”	28
2.3.4.3. Đặc tả use case chức năng “Quản lý đặt vé”	29
2.3.5. Đặc tả use case chức năng “Thanh toán”	30
2.3.5.1. Sơ đồ Use Case chức năng “Thanh toán”	30
2.3.5.2. Sơ đồ hoạt động chức năng “Thanh toán”	31
2.3.5.3. Đặc tả use case chức năng “Thanh toán”	32
2.4. Biểu đồ lớp hệ thống	33
2.4.1. Biểu đồ lớp tài khoản	33
2.4.2. Biểu đồ lớp phòng	33
2.4.3. Biểu đồ lớp combo	34
2.5. Thiết kế cơ sở dữ liệu	35
2.5.1. Biểu đồ cơ sở dữ liệu hệ thống	35
CHƯƠNG 3. CÀI ĐẶT CHƯƠNG TRÌNH	38
3.1. Tổ chức chương trình	38
3.2. Giao diện chính của chương trình	40
3.2.1. Giao diện người dùng	40
3.2.2. Giao diện quản lý hệ thống	51
CHƯƠNG 4. KẾT LUẬN	63
TÀI LIỆU THAM KHẢO	65
PHỤ LỤC	66

DANH MỤC CÁC HÌNH ẢNH

Hình 1.3. 1.1. Khái niệm Firebase là gì?	7
Hình 1.3.3. 1. Firebase thuộc quyền sở hữu của google	8
Hình 1.3.4. 1. RL là một cơ sở dữ liệu theo thời gian thực Authentication	9
Hình 1.3.4. 2. Cloud Storage cho phép lưu trữ và quản lý nội dung đã tạo ra	9
Hình 1.3.4. 3. Hosting giúp người dùng sử dụng dịch vụ một cách dễ dàng hơn ..	10
Hình 1.3.4. 4. Analytics giúp phân tích hành vi của người sử dụng.....	11
Hình 1.3.5. 1. Firebase có chức năng phát triển ứng dụng di động.....	11
Hình 1.3.5. 2. Firebase có chức năng lưu trữ và đồng bộ dữ liệu.....	12
Hình 1.3.5. 3. Firebase phân tích dữ liệu để giúp lập trình hiệu hơn	13
Hình 2.2. 1. Chức năng hệ thống quản lý	17
Hình 2.2. 2. Chức năng hệ thống người dùng	17
Hình 2.3.1. 1. Use case tổng quát hệ thống	20
Hình 2.3.1. 4. Sơ đồ Use Case chức năng “Quản lý đặt vé”	27
Hình 2.3.1. 5. Sơ đồ Use Case chức năng “Thanh toán”	30
Hình 2.3.2. 1. Sơ đồ Use Case chức năng “Quản lý tài khoản”	21
Hình 2.3.2.2. 1. Sơ đồ hoạt động chức năng “Đặt vé”	25
Hình 2.3.4.2. 1. Sơ đồ hoạt động chức năng “Quản lý đặt vé”	28
Hình 2.3.5.2. 1. Sơ đồ hoạt động chức năng “Thanh toán”	31
Hình 2.4.1. 1. Biểu đồ lớp TaiKhoan	33
Hình 2.4.2. 1. Biểu đồ lớp Phong	33
Hình 2.4.3. 1. Biểu đồ lớp Combo	34
Hình 2.5.1. 1. Biểu đồ cơ sở dữ liệu tổng quát	35
Hình 3.2.1. 1. Form “Đăng nhập”	40
Hình 3.2.1. 2. Form đăng ký	41
Hình 3.2.1. 3. Form “Thông tin tài khoản”	43
Hình 3.2.1. 4. Form “Chọn phim”	44
Hình 3.2.1. 5. Form “Chọn rạp”	45
Hình 3.2.1. 6. Form “Chọn chỗ ngồi”	46
Hình 3.2.1. 7. Form “Chọn combo”	47
Hình 3.2.1. 8. Form “Thanh toán hóa đơn”	48
Hình 3.2.1. 9. Form “Xem vé đã đặt vé”	49
Hình 3.2.1. 10. Form “Lịch chiếu theo rạp”	50
Hình 3.2.2. 1. Form “Quản lý HotCinema”	51
Hình 3.2.2. 2. Form “Danh sách tài khoản”	52

<i>Hình 3.2.2. 3. Form “Thêm tài khoản”</i>	53
<i>Hình 3.2.2. 4. Form “Sửa, xóa tài khoản”</i>	54
<i>Hình 3.2.2. 5. Form “Danh sách Combo HotCinema”</i>	55
<i>Hình 3.2.2. 6. Form “Thêm Combo”</i>	56
<i>Hình 3.2.2. 7. Form “Sửa, xóa Combo”</i>	57
<i>Hình 3.2.2. 8. Form “Quản lý phòng”</i>	58
<i>Hình 3.2.2. 9. Form “Thêm phòng”</i>	59
<i>Hình 3.2.2. 10. Form “Sửa, xóa phòng”</i>	60
<i>Hình 3.2.2. 11. Form “Ghế ngồi”</i>	61
<i>Hình 3.2.2. 12. Form “Phòng vé”</i>	62

DANH MỤC CÁC BẢNG BIỂU

<i>Bảng 2.2.1. 1. Tác nhân người dùng.....</i>	<i>18</i>
<i>Bảng 2.2.1. 2. Các ca sử dụng</i>	<i>19</i>
<i>Bảng 2.3.2.3. 1. Đặc tả use case chức năng “Quản lý tài khoản”</i>	<i>23</i>
<i>Bảng 2.3.3.3. 1. Đặc tả use case chức năng “Đặt vé”</i>	<i>26</i>
<i>Bảng 2.3.4.3. 1. Đặc tả use case chức năng “Quản lý đặt vé”</i>	<i>29</i>
<i>Bảng 2.3.5.3. 1. Đặc tả use case chức năng “Thanh toán”</i>	<i>32</i>
<i>Bảng 2.5.2. 1. Phim.....</i>	<i>35</i>
<i>Bảng 2.5.2. 2. Phòng.....</i>	<i>36</i>
<i>Bảng 2.5.2. 3. Loại ghế</i>	<i>36</i>
<i>Bảng 2.5.2. 4. Ghế</i>	<i>36</i>
<i>Bảng 2.5.2. 5. Combo.....</i>	<i>36</i>
<i>Bảng 2.5.2. 6. Vé.....</i>	<i>37</i>
<i>Bảng 2.5.2. 7. Tài khoản</i>	<i>37</i>

LỜI NÓI ĐẦU

Ngày nay, sự phát triển vượt bậc của khoa học kỹ thuật, đặc biệt là công nghệ thông tin, đã tạo ra những thay đổi sâu sắc trong nhiều lĩnh vực của đời sống. Các ứng dụng di động đang trở thành công cụ quan trọng, đáp ứng nhu cầu đa dạng của con người, từ quản lý công việc, giải trí, cho đến mua sắm và dịch vụ. Trước đây, nhiều công việc như mua vé xem phim thường phải thực hiện trực tiếp, đòi hỏi thời gian và công sức. Tuy nhiên, với sự ra đời của các ứng dụng đặt vé trực tuyến, quy trình này đã trở nên thuận tiện và nhanh chóng hơn bao giờ hết.

Trong bối cảnh đó, việc phát triển một ứng dụng di động để đặt vé xem phim là cần thiết và có ý nghĩa thực tiễn cao. Ứng dụng này không chỉ giúp người dùng dễ dàng tra cứu thông tin về các bộ phim đang chiếu, chọn lựa suất chiếu phù hợp mà còn đặt vé một cách nhanh chóng và tiện lợi ngay trên điện thoại di động. Điều này không chỉ giúp tiết kiệm thời gian mà còn tạo ra một trải nghiệm mới mẻ và hiện đại cho người dùng.

Chương trình đặt vé xem phim có vai trò quan trọng trong việc cải thiện trải nghiệm của khách hàng khi sử dụng dịch vụ giải trí. Bằng việc xây dựng một ứng dụng mobile với giao diện thân thiện, dễ sử dụng, và tích hợp nhiều tính năng tiện ích như thanh toán trực tuyến, chọn chỗ ngồi, và nhận thông báo về các chương trình khuyến mãi, ứng dụng sẽ góp phần nâng cao chất lượng dịch vụ, đồng thời tạo ra sự khác biệt trong cạnh tranh.

Ứng dụng này không chỉ mang lại lợi ích cho người dùng mà còn hỗ trợ các rạp chiếu phim trong việc quản lý lượng khách hàng, tối ưu hóa hoạt động kinh doanh và tiếp cận với khách hàng tiềm năng. Ngoài ra, nó cũng giúp giảm thiểu các sai sót trong quá trình đặt vé và thanh toán, đồng thời tăng cường tính minh bạch và chính xác.

Với những lợi ích thiết thực mà ứng dụng đặt vé xem phim mang lại, việc nghiên cứu và phát triển một ứng dụng di động bằng ngôn ngữ lập trình hiện đại như Java là một hướng đi đúng đắn và cần thiết. Java là một ngôn ngữ lập trình mạnh mẽ, dễ học, và có khả năng tích hợp với nhiều nền tảng di động khác nhau, phù hợp cho việc xây dựng các ứng dụng có tính ứng dụng cao trong thực tế. Ứng dụng này không chỉ giải quyết được các vấn đề hiện tại mà còn có khả năng mở rộng và phát triển trong tương lai, đáp ứng nhu cầu ngày càng cao của người dùng trong việc trải nghiệm dịch vụ giải trí hiện đại.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về ứng dụng cho thiết bị di động

Ứng dụng cho thiết bị di động là các phần mềm được thiết kế và phát triển để chạy trên các thiết bị di động như điện thoại thông minh, máy tính bảng. Đây là một phần không thể thiếu trong cuộc sống hàng ngày của con người hiện đại, cung cấp nhiều tính năng và tiện ích từ giải trí, giáo dục, làm việc, đến mua sắm và chăm sóc sức khỏe.

Phân loại ứng dụng di động:

Ứng dụng gốc (Native apps): Được phát triển riêng cho một hệ điều hành cụ thể (Android, iOS) bằng ngôn ngữ lập trình tương ứng (Java/Kotlin cho Android, Swift/Objective-C cho iOS). Ưu điểm của loại này là tối ưu hóa về hiệu năng và giao diện người dùng, nhưng nhược điểm là cần phát triển riêng lẻ cho từng hệ điều hành.

Ứng dụng web di động (Mobile Web Apps): Là các trang web được thiết kế để hiển thị tốt trên thiết bị di động. Chúng chạy trên trình duyệt và không cần cài đặt qua cửa hàng ứng dụng. Ưu điểm là không phụ thuộc vào hệ điều hành, nhưng thường kém mượt mà và không thể truy cập nhiều tính năng của thiết bị như GPS, camera.

Ứng dụng lai (Hybrid apps): Kết hợp giữa ứng dụng gốc và ứng dụng web. Chúng được phát triển bằng các công nghệ web (HTML, CSS, JavaScript) và sau đó được đóng gói thành ứng dụng gốc thông qua các nền tảng như Cordova, React Native. Ưu điểm là phát triển nhanh và tiết kiệm chi phí, nhưng hiệu năng không cao bằng ứng dụng gốc.

1.1.1. Xu hướng phát triển ứng dụng cho thiết bị di động

Xu hướng phát triển ứng dụng cho thiết bị di động đang tập trung vào việc tối ưu hóa trải nghiệm người dùng và khai thác các công nghệ tiên tiến để tạo ra các ứng dụng thông minh, hiệu quả và an toàn hơn. Sự gia tăng về nhu cầu cá nhân hóa, bảo mật dữ liệu và hiệu suất ứng dụng đã thúc đẩy sự phát triển của các công nghệ mới như trí tuệ nhân tạo, máy học, và kết nối 5G.

Các ứng dụng hiện đại không chỉ cần đáp ứng các yêu cầu cơ bản về chức năng mà còn phải tích hợp nhiều dịch vụ trong một, đảm bảo tính linh hoạt và tiện lợi cho người dùng. Đồng thời, với sự phát triển của công nghệ, ứng dụng di động cũng đang trở nên ngày càng phức tạp hơn, đòi hỏi sự đổi mới liên tục từ các nhà phát triển để duy trì sức cạnh tranh trên thị trường.

Sự kết hợp giữa tốc độ kết nối nhanh hơn, trải nghiệm người dùng mượt mà hơn, và yêu cầu bảo mật cao đã và đang tạo ra những thay đổi đáng kể trong cách phát triển và triển khai các ứng dụng di động. Tương lai của các ứng dụng này sẽ tiếp tục được định hình bởi những tiến bộ công nghệ và những thay đổi trong hành vi sử dụng của người dùng.

Một số xu hướng quan trọng đang định hình tương lai của các ứng dụng di động:

Trí tuệ nhân tạo (AI) và Machine Learning (ML).

Ứng dụng đa nền tảng.

Công nghệ 5G.

Ứng dụng không cần cài đặt (Progressive Web Apps - PWA).

Tăng cường thực tế (AR) và Thực tế ảo (VR).

Bảo mật và quyền riêng tư.

Ứng dụng di động dành cho doanh nghiệp (Enterprise Mobile Apps).

1.1.2. Tổng quan về các hệ điều hành thiết bị di động, các công cụ lập trình

Hệ điều hành (HĐH) cho thiết bị di động là phần mềm hệ thống giúp quản lý tài nguyên phần cứng và phần mềm, cung cấp nền tảng cho các ứng dụng di động hoạt động.

Dưới đây là các hệ điều hành di động phổ biến nhất:

Android: Được phát triển bởi google là hệ điều hành di động phổ biến nhất thế giới, chiếm hơn 70% thị phần toàn cầu.

Đặc điểm: Android là hệ điều hành mã nguồn mở, dựa trên nhân Linux. Nó hỗ trợ nhiều thiết bị từ các nhà sản xuất khác nhau. Android có một cộng đồng phát triển ứng dụng rộng lớn và đa dạng.

iOS: Được phát triển bởi Apple chiếm khoảng 25-30% thị phần toàn cầu, tập trung chủ yếu ở các quốc gia phát triển.

Đặc điểm: iOS là hệ điều hành độc quyền của Apple, chạy trên các thiết bị như iPhone, iPad. iOS được biết đến với tính bảo mật cao, giao diện người dùng mượt mà và tích hợp sâu với hệ sinh thái của Apple.

Các Công Cụ Lập Trình Phát Triển Ứng Dụng Di Động:

Android Studio: Dành cho phát triển ứng dụng Android.

Xcode: Dành cho phát triển ứng dụng iOS.

React Native: Dành cho phát triển ứng dụng đa nền tảng (Android và iOS).

Flutter: Dành cho phát triển ứng dụng đa nền tảng (Android và iOS).

Xamarin: Dành ch phát triển ứng dụng đa nền tảng (Android và iOS).

Unity: Dành cho phát triển trò chơi di động và các ứng dụng 3D.

1.1.3. Hệ điều hành Android

Android là một hệ điều hành mã nguồn mở được phát triển bởi Google, dựa trên nhân Linux. Ban đầu, hệ điều hành này được thiết kế chủ yếu cho các thiết bị di động như điện thoại thông minh và máy tính bảng, nhưng sau đó đã mở rộng sang nhiều loại thiết bị khác như đồng hồ thông minh (Wear OS), TV thông minh (Android TV), và thậm chí cả xe hơi (Android Auto). Android hiện là hệ điều hành di động phổ biến nhất thế giới, với hơn 70% thị phần toàn cầu.

Lịch sử phát triển:

Khởi nguồn: Android được phát triển bởi Android Inc., một công ty công nghệ nhỏ được thành lập vào năm 2003 bởi Andy Rubin, Rich Miner, Nick Sears, và Chris White. Google đã mua lại Android Inc. vào năm 2005.

Ra mắt chính thức: Phiên bản đầu tiên của Android (Android 1.0) ra mắt vào tháng 9 năm 2008, cùng với chiếc điện thoại T-Mobile G1 (còn gọi là HTC Dream), đánh dấu sự xuất hiện của hệ điều hành này trên thị trường.

Phát triển qua các phiên bản: Android đã trải qua nhiều phiên bản nâng cấp, mỗi phiên bản thường được đặt tên theo các món tráng miệng theo thứ tự bảng chữ cái (ví dụ: Cupcake, Donut, Eclair). Những phiên bản gần đây đã chuyển sang cách đánh số phiên bản đơn giản hơn (Android 10, 11, 12...).

Đặc điểm nổi bật:

Mã nguồn mở: Android là hệ điều hành mã nguồn mở, có nghĩa là mã nguồn của nó được phát hành công khai. Điều này cho phép các nhà sản xuất thiết bị và nhà phát triển phần mềm có thể tùy chỉnh và điều chỉnh Android cho các nhu cầu cụ thể của họ.

Đa dạng thiết bị: Android không chỉ giới hạn ở điện thoại thông minh mà còn được sử dụng rộng rãi trên máy tính bảng, TV thông minh, đồng hồ thông minh, ô tô, và nhiều thiết bị IoT khác. Điều này tạo ra một hệ sinh thái phong phú và đa dạng, phục vụ cho nhiều nhu cầu sử dụng khác nhau.

Google Play Store: Một trong những thành phần cốt lõi của Android là Google Play Store, nơi người dùng có thể tải xuống hàng triệu ứng dụng, trò chơi, phim, sách và nhiều nội dung khác. Google Play Store đóng vai trò quan trọng trong việc mở rộng hệ sinh thái Android, cung cấp nền tảng cho các nhà phát triển tiếp cận người dùng.

Tùy biến cao: Android cho phép người dùng tùy chỉnh giao diện và chức năng của thiết bị theo nhiều cách khác nhau, từ việc thay đổi hình nền, chủ đề, đến cài đặt các launcher khác nhau để thay đổi cách sử dụng.

1.1.4. Tổng quan về Android Studio

Android Studio là IDE cụm từ viết tắt của Integrated Development Environment, môi trường để phát triển tích hợp các ứng dụng Android. Phần mềm này đã dựa trên INTELLIJ IDEA, một môi trường phát triển của phần mềm Java.

Để giúp phát triển được ứng dụng bên trong hệ điều hành Android thì cần phải có phần mềm hệ thống công nghệ được xây dựng dựa trên Gradle, trình giả lập, tích hợp Github, mẫu mã. Mỗi một dự án ở trong Android Studio có 1 hoặc nhiều các phương thức, mã nguồn và các tệp tài nguyên. Những phương thức này sẽ bao gồm mô-đun cho ứng dụng, thư viện và Google App Engine.

Một số tính năng nổi bật:

Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).

Chức năng dò và sửa lỗi nhanh, hướng Android.

Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.

Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.

Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.

1.2. Phát triển ứng dụng trên nền tảng Android sử dụng Java

Trong phần này, sẽ thảo luận về việc phát triển ứng dụng trên nền tảng Android bằng cách sử dụng ngôn ngữ lập trình Java. Đây là ngôn ngữ chính được sử dụng để xây dựng các ứng dụng Android, cho phép các nhà phát triển tạo ra các ứng dụng mạnh mẽ và linh hoạt. Bài viết sẽ bao gồm từ cơ bản đến nâng cao về các khái niệm, công cụ và phương pháp cần thiết để phát triển ứng dụng Android.

Lý do chọn Java: Java là một ngôn ngữ lập trình hướng đối tượng, dễ học và mạnh mẽ. Java có cộng đồng phát triển lớn và được hỗ trợ rộng rãi, khiến nó trở thành lựa chọn lý tưởng cho việc phát triển ứng dụng trên Android.

Các công cụ phát triển: Android Studio là công cụ chính thức của Google để phát triển ứng dụng Android. Ngoài ra, còn có các công cụ khác như Gradle để quản lý build, ADB để kiểm tra và gỡ lỗi, và các thư viện hỗ trợ khác.

1.2.1. Cấu trúc ứng dụng Android

Cấu trúc của một ứng dụng Android bao gồm các thành phần chính như Activity, Fragment, Service, Content Provider, và Broadcast Receiver. Mỗi thành phần này đóng một vai trò quan trọng trong việc xây dựng và quản lý ứng dụng Android.

Activity: Đây là thành phần cơ bản nhất của một ứng dụng Android, đại diện cho một màn hình giao diện người dùng. Mỗi Activity có một vòng đời riêng biệt và có thể tương tác với các Activity khác.

Fragment: Fragment là một phần của giao diện người dùng hoặc hành vi trong một Activity. Nó cho phép bạn tạo ra các giao diện người dùng linh hoạt và tái sử dụng được giữa các Activity khác nhau.

Service: Service là thành phần chạy ngầm trong hệ thống, không có giao diện người dùng. Service có thể được sử dụng để thực hiện các tác vụ dài hạn, chẳng hạn như phát nhạc trong nền hoặc tải dữ liệu từ mạng.

1.2.2. Các thành phần phát triển ứng dụng Android

Để phát triển một ứng dụng Android hoàn chỉnh, cần hiểu rõ về các thành phần cơ bản và các công cụ hỗ trợ:

Manifest File: Đây là tập tin cấu hình chính của ứng dụng, chứa các thông tin như tên ứng dụng, biểu tượng, và các quyền mà ứng dụng yêu cầu.

Resources: Tài nguyên của ứng dụng bao gồm các tập tin hình ảnh, chuỗi văn bản, bố cục giao diện người dùng, và các tập tin XML khác.

Gradle: Gradle là công cụ quản lý xây dựng mạnh mẽ, giúp tự động hóa các tác vụ như biên dịch mã nguồn, kiểm tra và triển khai ứng dụng.

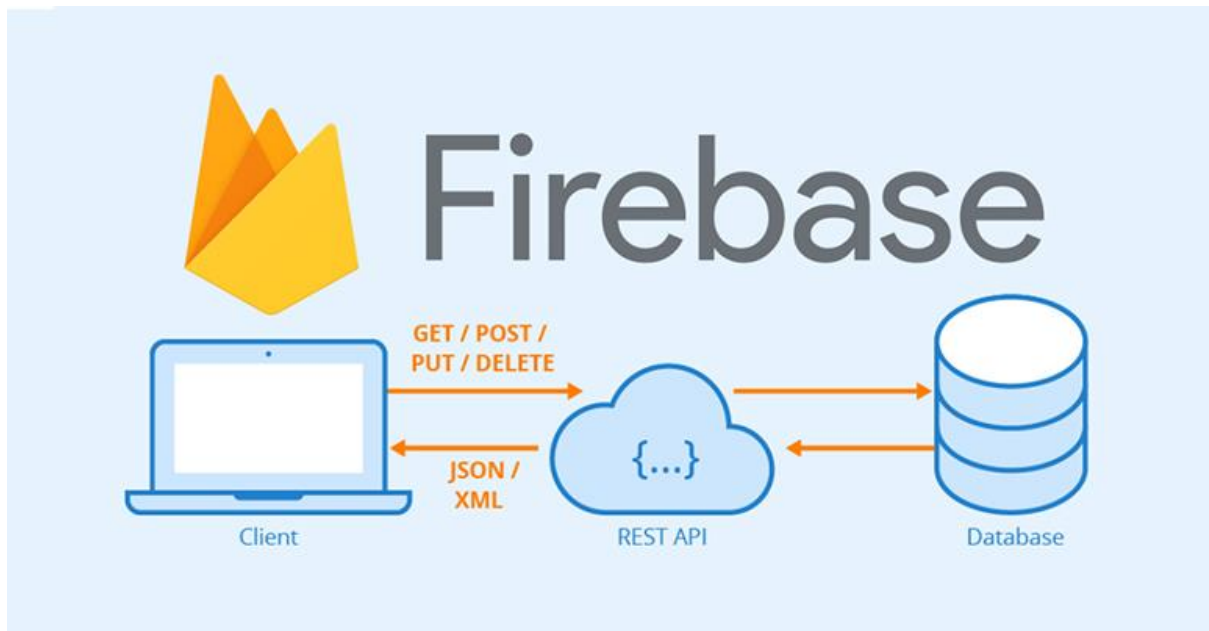
Android Jetpack: Android Jetpack là một bộ thư viện được phát triển bởi Google để giúp đơn giản hóa quá trình phát triển ứng dụng. Nó bao gồm các thư viện để quản lý dữ liệu, giao diện người dùng, và tương tác với hệ điều hành.

1.3. Giới thiệu về Firebase

1.3.1. Firebase là gì?

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.

Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.



Hình 1.3. 1.1. Khái niệm Firebase là gì?

1.3.2. Lịch sử hình thành và phát triển nền tảng Firebase

Vào năm 2011, James Tamplin và Andrew Lee đã cho ra mắt Evolve. Đây là một nền tảng có cấu trúc khá đơn giản chuyên cung cấp các API cần thiết để tích hợp tính năng trò chuyện vào các trang web. Tuy nhiên, họ nhận ra rằng nền tảng này đang được sử dụng để truyền dữ liệu ứng dụng chứ không đơn giản là chat. Sau đó, họ đã phát triển Envole và tạo nên Firebase.

Đến tháng 4 năm 2012, Firebase đã được công bố trên toàn cầu dưới dạng một công ty riêng biệt. Những năm sau đó, Firebase đã thực hiện nhiều cuộc huy động vốn và ra mắt các sản phẩm mới.

Đến tháng 10 năm 2014, Firebase đã chính thức được Google mua lại và trở thành một ứng dụng đa năng trên nền tảng di động và website.

1.3.3. Firebase thuộc sở hữu của ai?

Ban đầu, Firebase là một sáng chế của James Tamplin và Andrew Lee. Tuy nhiên trải qua nhiều cuộc huy động vốn, Firebase đã chính thức được Google mua lại vào năm 2014. Vì vậy, tính đến thời điểm hiện tại, nền tảng này là một phần và thuộc quyền sở hữu của Google.



Hình 1.3.3. 1. Firebase thuộc quyền sở hữu của google

1.3.4. Các tính năng chính của Firebase

Firebase là một nền tảng đa năng giúp người dùng dễ dàng lập trình ứng dụng trên nền tảng di động và website. Vậy các tính năng chính của Firebase là gì?

Realtime Database:

Realtime Database là một cơ sở dữ liệu thời gian thực. Ngay sau khi bạn đăng ký tài khoản trên Firebase, bạn sẽ nhận được Realtime Database được lưu trữ dưới dạng JSON và được đồng bộ hóa theo thời gian thực đối với mọi kết nối.

Đối với các ứng dụng được xây dựng trên đa nền tảng như Android, IOS và WebApp, tất cả client sẽ cùng sử dụng một cơ sở dữ liệu. Bên cạnh đó, hệ thống dữ liệu này sẽ tự động cập nhật khi lập trình viên phát triển ứng dụng. Sau đó, tất cả dữ liệu này sẽ được truyền tải thông qua các kết nối SSL có 2048 bit.



Hình 1.3.4. 1. RL là một cơ sở dữ liệu theo thời gian thực Authentication

Authentication là tính năng giúp xác thực danh tính của người dùng ứng dụng. Firebase cung cấp các bước xác thực thông qua Email, Facebook, Twitter, GitHub hay Google. Điều này giúp cho các thông tin cá nhân của khách hàng được bảo vệ một cách tốt nhất, hạn chế được tình trạng bị hacker đánh cắp. Đồng thời việc xác thực danh tính qua Firebase sẽ giúp người dùng tiếp cận sản phẩm nhanh chóng và an toàn hơn.

Cloud Storage:

Cloud Storage là tính năng cho phép lưu trữ và quản lý nội dung đã tạo ra như ảnh, video, nội dung, văn bản,... Firebase Storage cung cấp các API hỗ trợ bạn upload và download các file từ ứng dụng một cách trơn tru mà không cần quan tâm đến chất lượng đường truyền mạng với độ bảo mật cao.



Hình 1.3.4. 2. Cloud Storage cho phép lưu trữ và quản lý nội dung đã tạo ra

Cloud Firestore:

Cloud Firestore được phát triển từ tính năng Realtime Database. Trải qua nhiều lần nâng cấp, Cloud Firestore có giao diện trực quan và khả năng mở rộng ưu việt hơn so với Realtime Database. Tính năng này của Firebase giúp đồng bộ mọi dữ liệu trên các ứng dụng thông qua việc đăng ký thời gian thực và cung cấp hỗ trợ ngoại tuyến cho thiết bị di động cũng như website.

Hosting:

Hosting được phân phối thông qua tiêu chuẩn công nghệ bảo mật SSL từ hệ thống mạng CDN. CDN là một mạng lưới máy chủ giúp lưu trữ các bản sao của các nội dung tĩnh trên website. Thông qua CDN, người dùng có thể truy cập và sử dụng các dịch vụ trên web khi cài Firebase Hosting một cách nhanh chóng và ổn định hơn.



Hình 1.3.4. 3. Hosting giúp người dùng sử dụng dịch vụ một cách dễ dàng hơn

Cloud Functions:

Cloud Functions giúp chạy code backend tự động nhằm phản hồi các sự kiện được kích hoạt bởi tính năng của Firebase và HTTPS request. Cloud Functions có tính bảo mật cao và luôn thực hiện chính xác những gì mà bạn muốn.

Analytics:

Analytics giúp bạn có thể phân tích hành vi của người sử dụng ứng dụng của bạn. Qua đó, bạn sẽ biết được khách hàng thường xuyên truy cập tính năng nào và các thông tin về hiệu quả quảng cáo, tình trạng trả phí,... để có thể đưa ra được chiến

lược phát triển phù hợp. Để thực hiện tính năng Analytics của Firebase, bạn cần cài đặt Software Development Kit (SDK).



Hình 1.3.4. 4. Analytics giúp phân tích hành vi của người sử dụng

1.3.5. Chức năng của Firebase

Firebase hiện đang là một trong những nền tảng được nhiều nhà lập trình sử dụng nhất. Nguyên nhân là bởi Firebase mang đến nhiều chức năng ưu việt giúp tối ưu hóa thời gian và công sức xây dựng ứng dụng. Vậy chức năng của Firebase là gì?

Phát triển ứng dụng di động:

Với cơ sở dữ liệu không lồ, Firebase giúp lập trình viên phát triển ứng dụng di động một cách nhanh chóng và giúp giảm thiểu chi phí. Song song đó, Firebase cho phép phát triển ứng dụng trên cả nền tảng iOS và Android nên có thể đáp ứng mọi yêu cầu về hệ điều hành



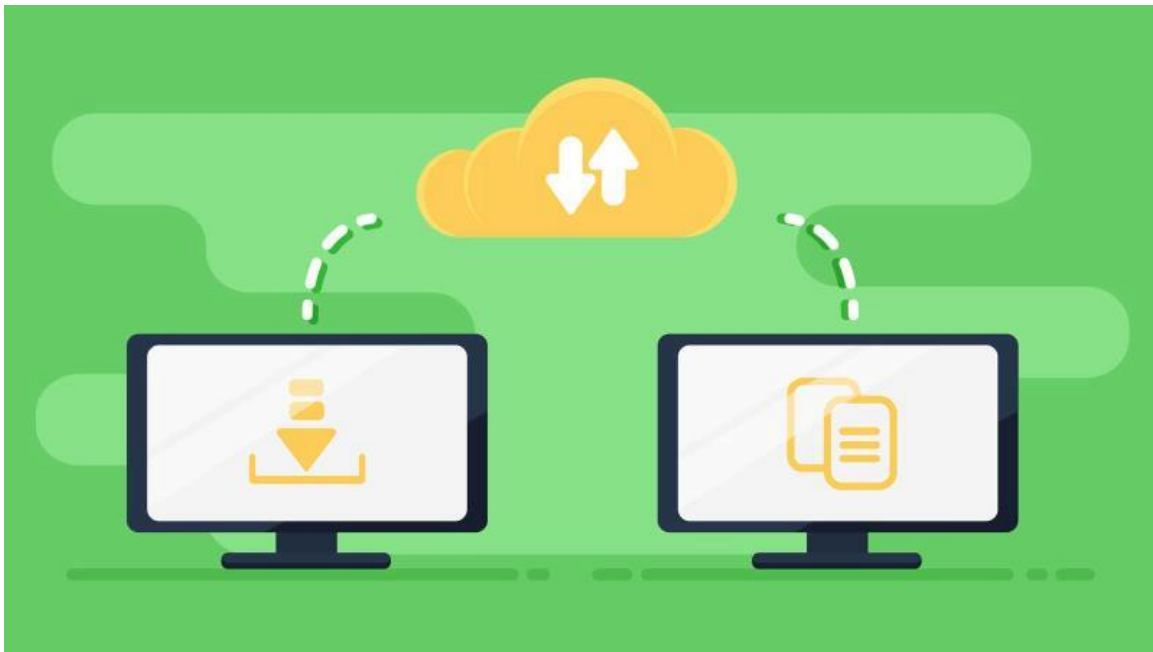
Hình 1.3.5. 1. Firebase có chức năng phát triển ứng dụng di động

Phát triển ứng dụng web:

Ngoài việc hỗ trợ phát triển ứng dụng di động, Firebase còn giúp xây dựng ứng dụng website. Các API đơn giản và hữu dụng giúp bạn có thể xây dựng ứng dụng mà không cần backend và server.

Lưu trữ và đồng bộ dữ liệu:

Firebase thường xuyên sao lưu dữ liệu nhằm đảm bảo tính sẵn có của dữ liệu. Bên cạnh đó, chức năng này của Firebase còn giúp cho mọi thông tin và dữ liệu của người dùng sẽ được bảo mật một cách tốt nhất nếu có trường hợp xấu xảy ra. Đặc biệt, khi sử dụng gói dịch vụ mất phí, bạn có thể dễ dàng sử dụng cấu hình cơ sở dữ liệu thời gian thực của Firebase để thực hiện các bản sao lưu tự động.



Hình 1.3.5. 2. Firebase có chức năng lưu trữ và đồng bộ dữ liệu

Quản lý đăng nhập:

Firebase Authentication giúp nhà phát triển quản lý người dùng ứng dụng một cách đơn giản và an toàn. Cụ thể, Firebase cung cấp nhiều phương pháp quản lý đăng nhập thông qua email với mật khẩu Google và Facebook. Việc đăng nhập này vừa đảm bảo xác thực được người dùng, vừa không để lộ thông tin cá nhân.

Phân tích dữ liệu:

Chức năng phân tích dữ liệu của Firebase giúp lập trình viên có một cái nhìn tổng quan về ứng dụng thông qua các số liệu thống kê, biểu đồ, bảng biểu,... Cụ thể, bạn sẽ biết được số liệu về số khách hàng đã truy cập vào app và web, khách hàng

đang sử dụng trong 30 phút vừa qua,.... Qua đó, bạn có thể xây dựng được những chiến lược phát triển ứng dụng sao cho phù hợp với nhu cầu của thị trường.



Hình 1.3.5. 3. Firebase phân tích dữ liệu để giúp lập trình hiệu hơn

Hosting:

Chức năng Hosting giúp lập trình viên đơn giản hóa các thao tác với cơ sở dữ liệu khi thiết lập cấu trúc website. Qua đó, bạn sẽ tiết kiệm được thời gian và công sức khi xây dựng backend cho trang web của mình.

Ưu điểm, nhược điểm của Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web được phát triển bởi Google. Dưới đây là một số ưu và nhược điểm của Firebase:

Ưu điểm:

Tích hợp dịch vụ toàn diện: Firebase cung cấp nhiều dịch vụ như cơ sở dữ liệu, xác thực, lưu trữ, chức năng đám mây, phân tích và thông báo đẩy, giúp các nhà phát triển không cần phải tích hợp nhiều công cụ từ bên ngoài.

Realtime Database: Cơ sở dữ liệu thời gian thực của Firebase cho phép đồng bộ dữ liệu ngay lập tức trên tất cả các thiết bị, rất phù hợp cho các ứng dụng cần cập nhật liên tục như chat hoặc ứng dụng đa người dùng.

Cloud Firestore: Là cơ sở dữ liệu linh hoạt và dễ mở rộng, hỗ trợ các truy vấn phức tạp hơn so với Realtime Database, và có khả năng tự động sao lưu dữ liệu.

Đễ dàng tích hợp và sử dụng: Firebase có tài liệu phong phú và dễ hiểu, với SDKs được hỗ trợ trên nhiều nền tảng (iOS, Android, Web). Điều này giúp cho việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng.

Khả năng mở rộng: Firebase được xây dựng trên hạ tầng của Google Cloud, giúp các ứng dụng dễ dàng mở rộng khi lượng người dùng tăng lên.

Xác thực người dùng: Firebase Authentication hỗ trợ nhiều phương thức xác thực (email/password, OAuth, số điện thoại), giúp bảo mật và quản lý người dùng dễ dàng.

Không cần quản lý backend: Firebase cung cấp các dịch vụ backend mạnh mẽ mà không cần quản lý máy chủ, tiết kiệm thời gian và công sức cho nhà phát triển.

Nhược điểm:

Giới hạn query trong Realtime Database: Realtime Database có giới hạn về query và có thể trở nên phức tạp khi làm việc với các tập dữ liệu lớn.

Chi phí có thể tăng cao: Mặc dù Firebase cung cấp gói miễn phí, nhưng khi ứng dụng phát triển và yêu cầu tài nguyên nhiều hơn, chi phí có thể tăng nhanh chóng.

Phụ thuộc vào hệ sinh thái của Google: Firebase là một sản phẩm của Google, nên nếu có thay đổi về chính sách hoặc ngừng dịch vụ, bạn có thể gặp rủi ro.

Khả năng tùy biến backend hạn chế: Do Firebase tập trung vào các dịch vụ không máy chủ (serverless), điều này có thể hạn chế khả năng tùy biến backend cho những ứng dụng có yêu cầu đặc thù.

Không hỗ trợ SQL: Firebase sử dụng NoSQL, nên nếu ứng dụng của bạn cần các tính năng của SQL như join bảng hoặc các thao tác phức tạp trên dữ liệu quan hệ, bạn sẽ gặp hạn chế.

Khó khăn trong việc di chuyển dữ liệu: Khi muốn chuyển đổi sang nền tảng khác, việc xuất và nhập dữ liệu từ Firebase có thể không dễ dàng và yêu cầu nhiều công sức.

Firestore là một giải pháp mạnh mẽ và tiện dụng cho nhiều loại ứng dụng, nhưng cũng có những hạn chế cần được xem xét khi lựa chọn.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Xác định yêu cầu bài toán

Ứng dụng đặt vé xem phim cần cung cấp một nền tảng tiện lợi cho người dùng để tìm kiếm, chọn lựa và mua vé cho các suất chiếu phim tại các rạp phim.

Đối tượng người dùng:

Khách hàng người muốn đặt vé xem phim.

Quản trị viên: Quản lý toàn bộ hệ thống, bao gồm người dùng, phim, suất chiếu rạp chiếu, chỗ ngồi.

Yêu cầu chức năng.

Quản lý tài khoản người dùng:

Đăng ký tài khoản: Người dùng có thể đăng ký tài khoản mới bằng email hoặc số điện thoại.

Đăng nhập/Đăng xuất: Người dùng có thể đăng nhập vào tài khoản của mình và đăng xuất khi cần thiết.

Quản lý thông tin cá nhân: Người dùng có thể cập nhật thông tin cá nhân như tên, email, số điện thoại, mật khẩu.

Tìm kiếm và xem thông tin phim:

Tìm kiếm phim: Người dùng có thể tìm kiếm phim theo tên, thể loại, ngày chiếu, và rạp phim.

Xem chi tiết phim: Hiển thị thông tin chi tiết về phim, bao gồm tóm tắt nội dung, thể loại, diễn viên, đạo diễn, thời lượng, và trailer.

Xem lịch chiếu: Hiển thị lịch chiếu cho từng phim tại các rạp khác nhau.

Đặt vé xem phim:

Chọn suất chiếu và chỗ ngồi: Người dùng có thể chọn suất chiếu và chỗ ngồi mong muốn từ sơ đồ chỗ ngồi.

Xác nhận đặt vé: Sau khi chọn chỗ, người dùng xác nhận thông tin đặt vé.

Thanh toán: Hỗ trợ nhiều phương thức thanh toán như thẻ tín dụng, ví điện tử, và chuyển khoản ngân hàng.

Quản lý lịch sử đặt vé:

Xem lịch sử đặt vé: Người dùng có thể xem lại các vé đã đặt trước đó, bao gồm thông tin chi tiết về phim, suất chiếu, và chỗ ngồi.

Yêu cầu về giao diện người dùng.

Thân thiện và trực quan: Giao diện người dùng phải dễ hiểu, dễ sử dụng với các bước đặt vé rõ ràng.

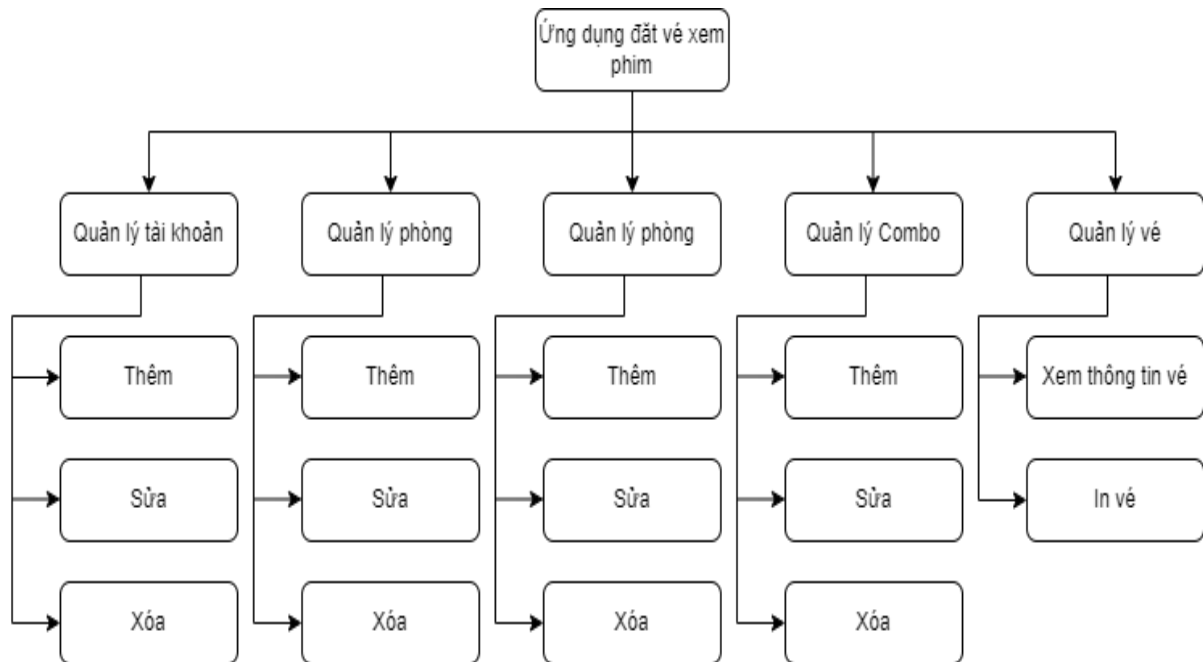
Đa ngôn ngữ: Ứng dụng cần hỗ trợ nhiều ngôn ngữ để phục vụ người dùng ở nhiều quốc gia khác nhau.

Thiết kế đáp ứng: Giao diện phải tương thích với nhiều loại thiết bị và kích thước màn hình khác nhau.

2.2. Phân tích hệ thống

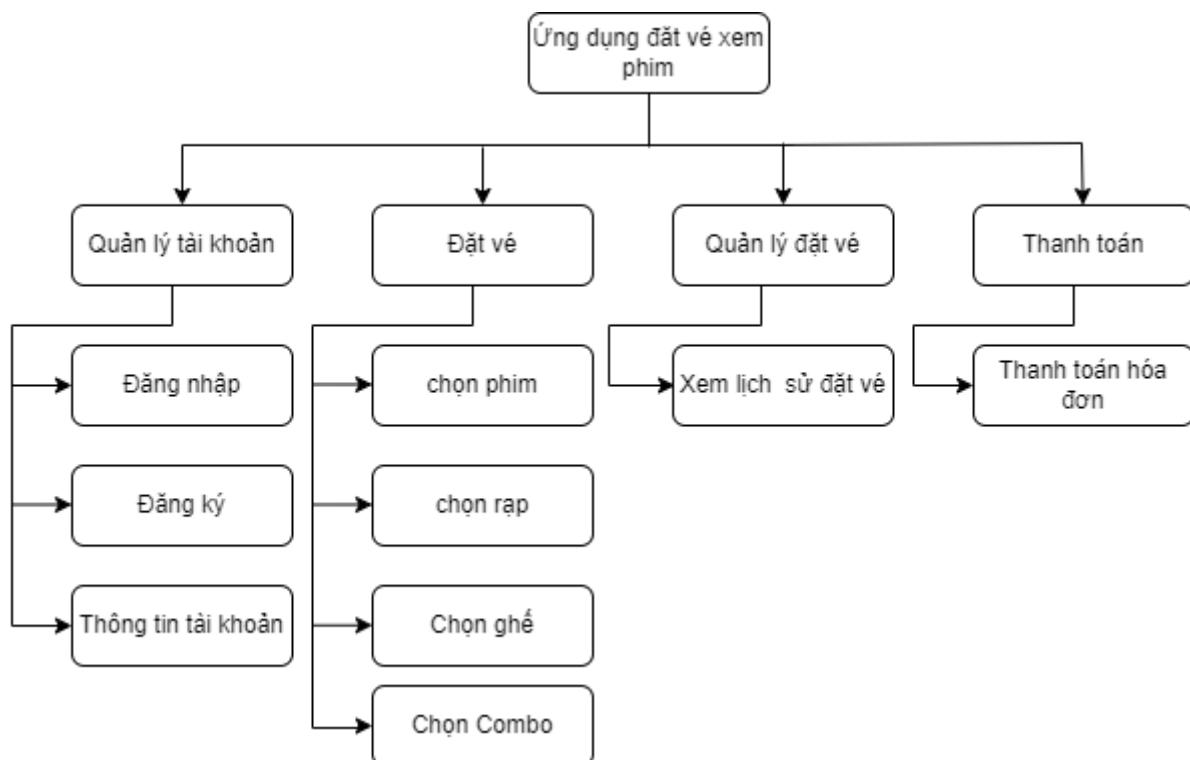
2.2.1. Sơ đồ chức năng của ứng dụng

Phần quản lý:



Hình 2.2. 1. Chức năng hệ thống quản lý

Phần người dùng:



Hình 2.2. 2. Chức năng hệ thống người dùng

2.2.2. Danh sách các tác nhân

Tác nhân: người dùng, quản lý.

Bảng 2.2.2. Tác nhân người dùng

STT	Tên tác nhân	Mô tả	Phân loại
1	Người quản lý	Quản lý toàn bộ hệ thống bao gồm: Quản lý tài khoản, quản lý phòng, quản lý ghế ngồi, quản lý vé, quản lý combo. Giải quyết các vấn đề.	Phức tạp
2	Người dùng	Người dùng đăng ký tài khoản và sử dụng dịch vụ của ứng dụng. Người dùng có thể theo dõi các bộ phim và đặt vé xem phim, thanh toán, theo dõi tình trạng vé đã đặt và thông tin cá nhân của mình	Phức tạp

2.2.3. Xác định các ca sử dụng use case

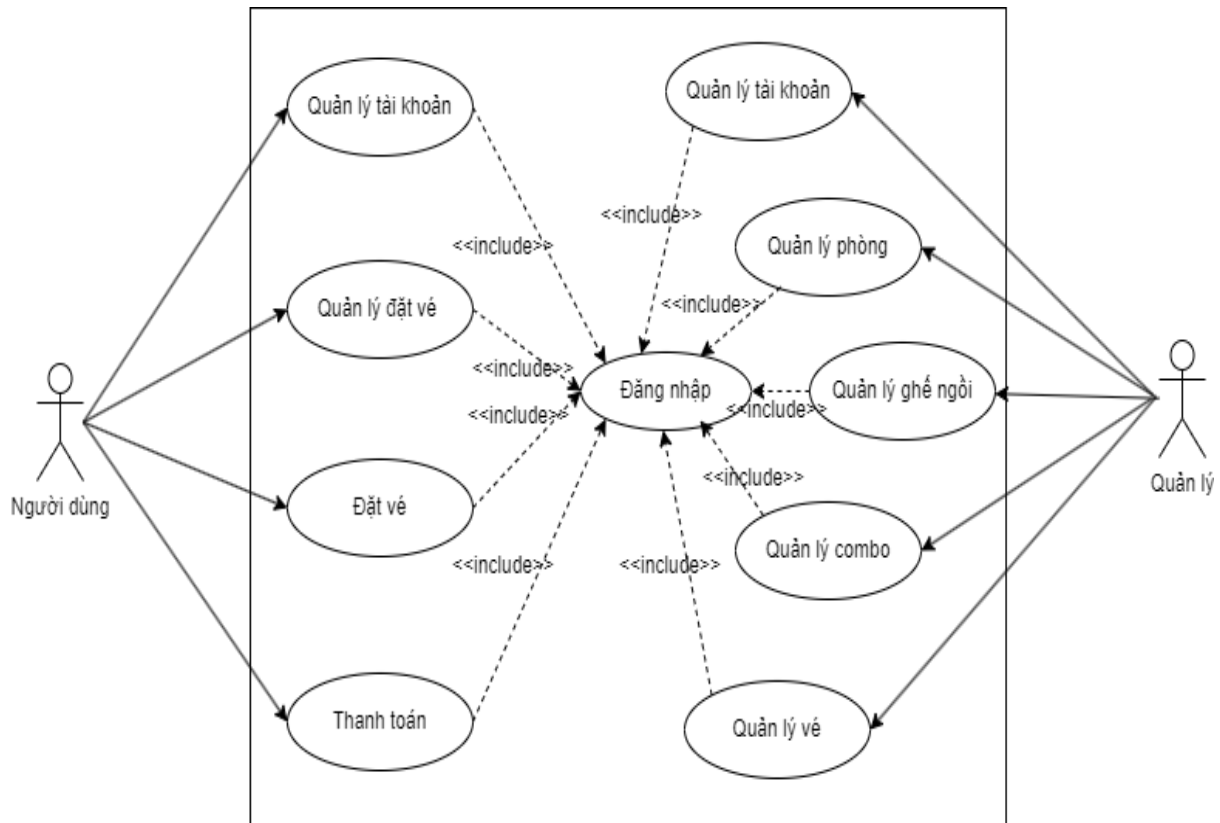
Dựa trên việc mô tả bài toán và việc phân tích để tìm ra các tác nhân, ta xác định được các ca sử dụng như sau:

Bảng 2.2.3. Các ca sử dụng

Tác nhân	Ca sử dụng
Người quản lý	Đăng nhập. Quản lý tài khoản. Quản lý phòng. Quản lý vé. Quản lý ghế ngồi. Quản lý combo.
Người dùng	Đăng ký tài khoản Đăng nhập Xem chi tiết phim Đặt vé Thanh toán Xem lịch sử đặt vé Thông tin tài khoản

2.3. Thiết kế chương trình của hệ thống

2.3.1. Đặc tả use case tổng quát của hệ thống



Hình 2.3.1. 1. Use case tổng quát hệ thống

Các hành động chính của người dùng và quản lý bao gồm: Quản lý tài khoản, Quản lý đặt vé, đặt vé, thanh toán, quản lý phòng, quản lý ghế ngồi, quản lý combo, quản lý vé.

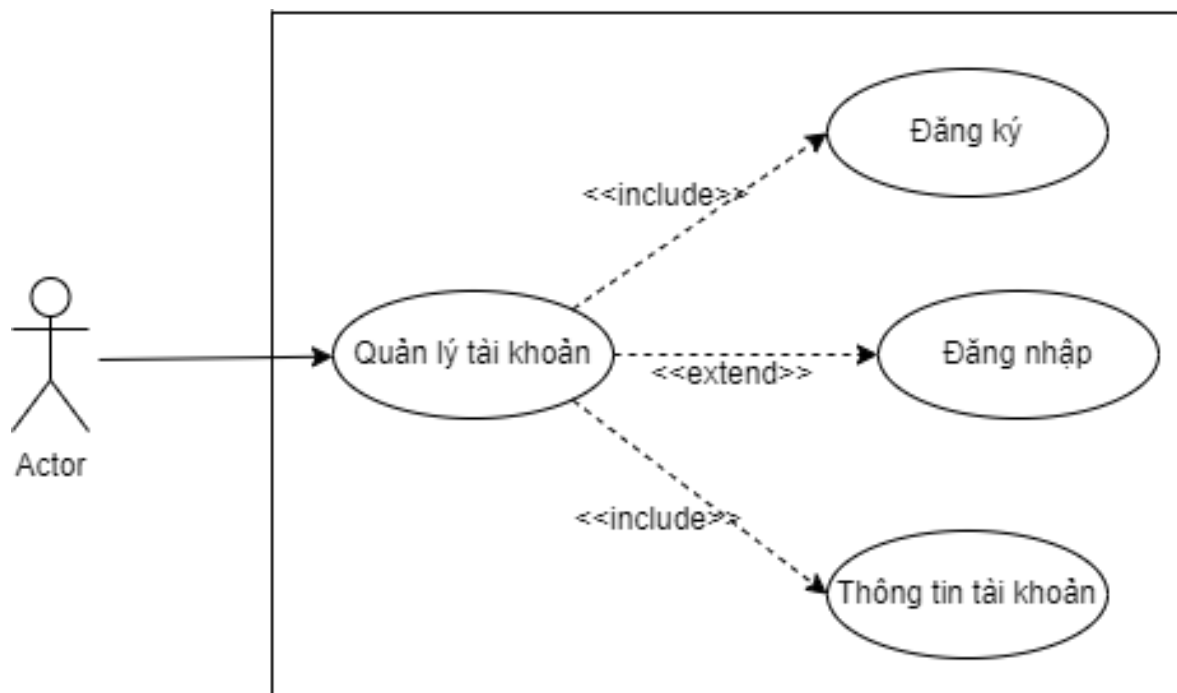
Quan hệ giữa các trường hợp sử dụng (Relationships):

Ca sử dụng "Đăng nhập" là một phần bắt buộc cho hầu hết các hành động:

Mọi hành động như quản lý tài khoản, quản lý đặt vé, đặt vé và thanh toán, quản lý phòng, quản lý ghế ngồi, quản lý combo, quản lý vé. đều có mối liên hệ với chức năng đăng nhập. Điều này được thể hiện thông qua mối quan hệ «include», có nghĩa là để thực hiện các chức năng này, người dùng cần phải đăng nhập vào hệ thống trước.

2.3.2. Đặc tả use case chức năng “Quản lý tài khoản”

2.3.2.1. Biểu đồ use case chức năng “Quản lý tài khoản”



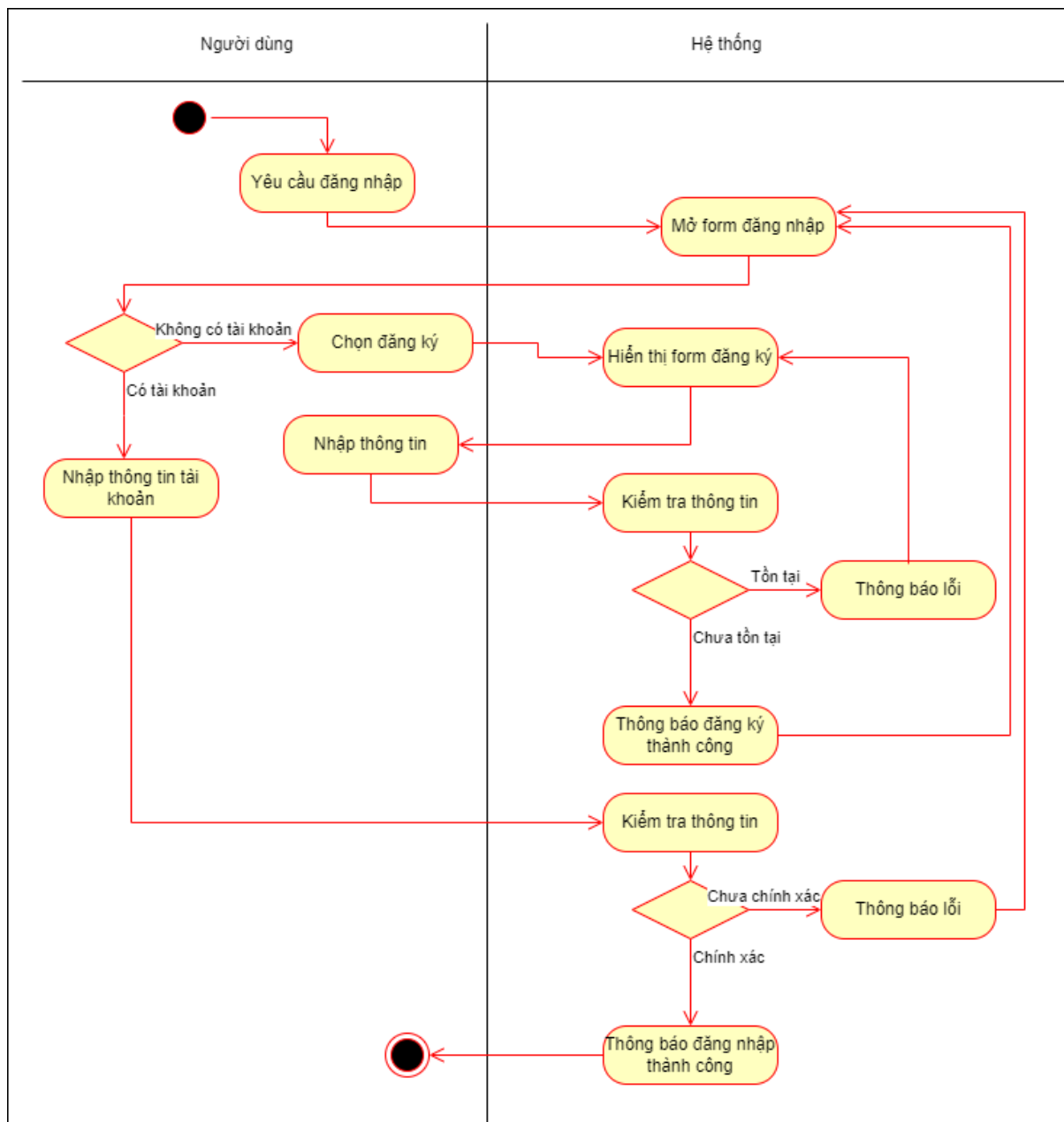
Hình 2.3.2. 1. Sơ đồ Use Case chức năng “Quản lý tài khoản”

Sơ đồ này mô tả các trường hợp sử dụng liên quan đến quản lý tài khoản, bao gồm các hành động như đăng ký, đăng nhập và xem, chỉnh sửa thông tin tài khoản.

Mối quan hệ <<include>> cho thấy rằng chức năng "Đăng ký" và "Thông tin tài khoản" là một phần của chức năng "Quản lý tài khoản". Điều này có nghĩa là trong quá trình quản lý tài khoản, việc đăng ký và xem thông tin tài khoản là bắt buộc và không thể tách rời.

Mối quan hệ <<extend>> giữa "Quản lý tài khoản" và "Đăng nhập" chỉ ra rằng chức năng đăng nhập là một phần mở rộng, được thực hiện khi có tài khoản đã tồn tại. Điều này có nghĩa là đăng nhập chỉ xảy ra khi người dùng đã có tài khoản, và nó không phải là một phần bắt buộc mỗi khi quản lý tài khoản.

2.3.2.2. Biểu đồ hoạt động chức năng “Quản lý tài khoản”



Hình 2.3.2.2. Biểu đồ hoạt động chức năng “Quản lý tài khoản”

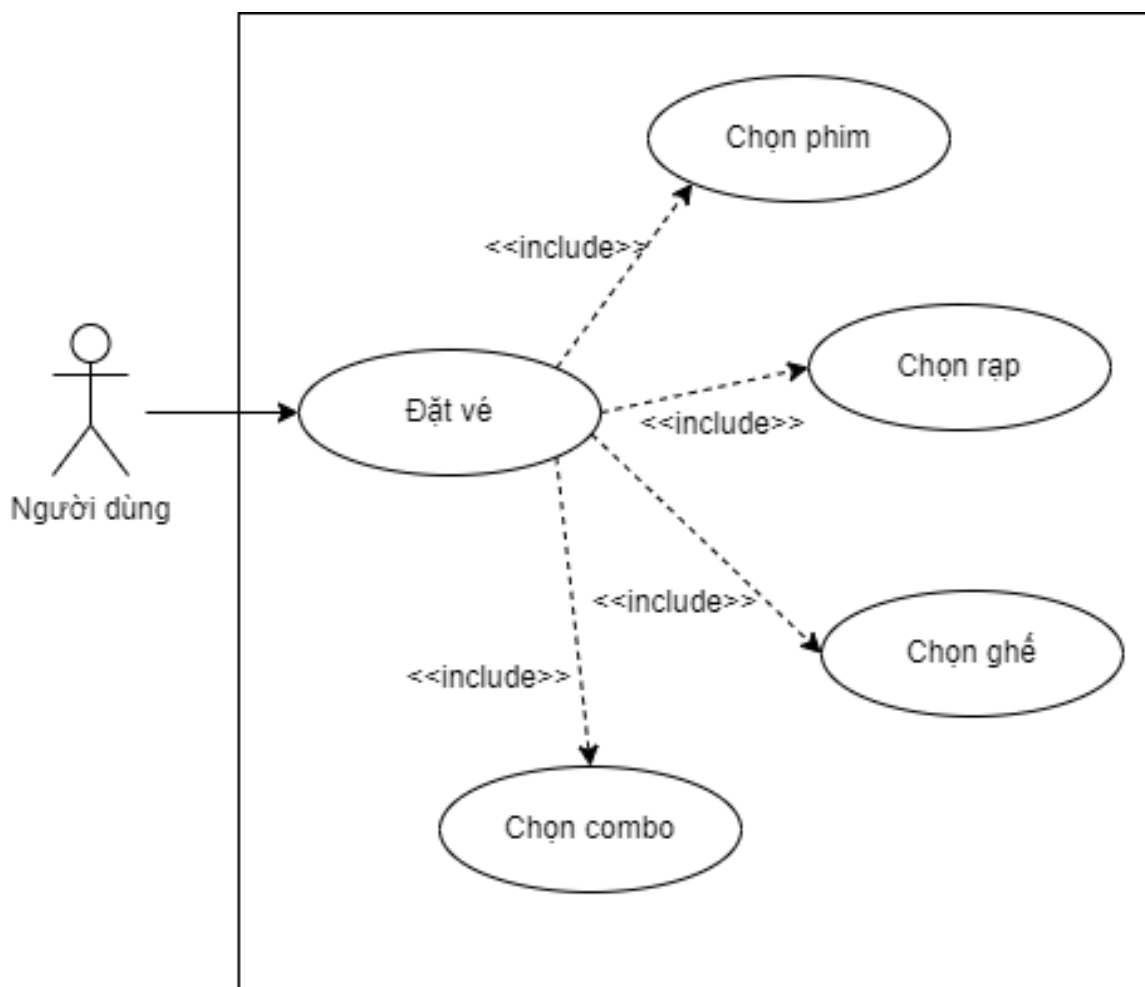
2.3.2.3. Đặc tả use case chức năng “Quản lý tài khoản”

Bảng 2.3.2.3. 1. Đặc tả use case chức năng “Quản lý tài khoản”

Tên Use case		Quản lý tài khoản
Tác nhân thực hiện		Khách hàng
Mục đích		Quản lý tài khoản (bao gồm đăng ký, đăng nhập và cập nhật thông tin)
Tiền điều kiện thực hiện		Người dùng chưa đăng nhập hoặc đã có tài khoản
Hậu điều kiện	Thành công	Đăng nhập hoặc đăng ký thành công, thông tin tài khoản được cập nhật vào hệ thống
	Lỗi	Đưa ra thông báo lỗi nếu xuất hiện lỗi không thực hiện được
Luồng nghiệp vụ chính/kịch bản chính		<p>1. Mở giao diện "Quản lý tài khoản" cho phép thực hiện các chức năng.</p> <p>1.1. Đăng ký tài khoản mới: - Nhập thông tin: Email, Mật khẩu, Họ tên</p> <p>1.2. Đăng nhập tài khoản: - Nhập email và mật khẩu.</p> <p>1.3. Cập nhật thông tin tài khoản: - Cập nhật email, mật khẩu, hoặc thông tin cá nhân (nếu cần).</p> <p>2. Hệ thống hiển thị thông báo thành công.</p> <p>3. Thoát.</p>
Luồng nghiệp vụ chính/kịch bản phụ		<p>1. Nếu thông tin đăng ký không hợp lệ (email đã tồn tại, mật khẩu yếu,...), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu sửa lại thông tin.</p> <p>2. Nếu đăng nhập không thành công (mật khẩu sai, tài khoản không tồn tại,...), hệ thống sẽ thông báo lỗi.</p>

2.3.3. Đặc tả use case chức năng “Đặt vé”

2.3.2.1. Sơ đồ use case chức năng “Đặt vé”

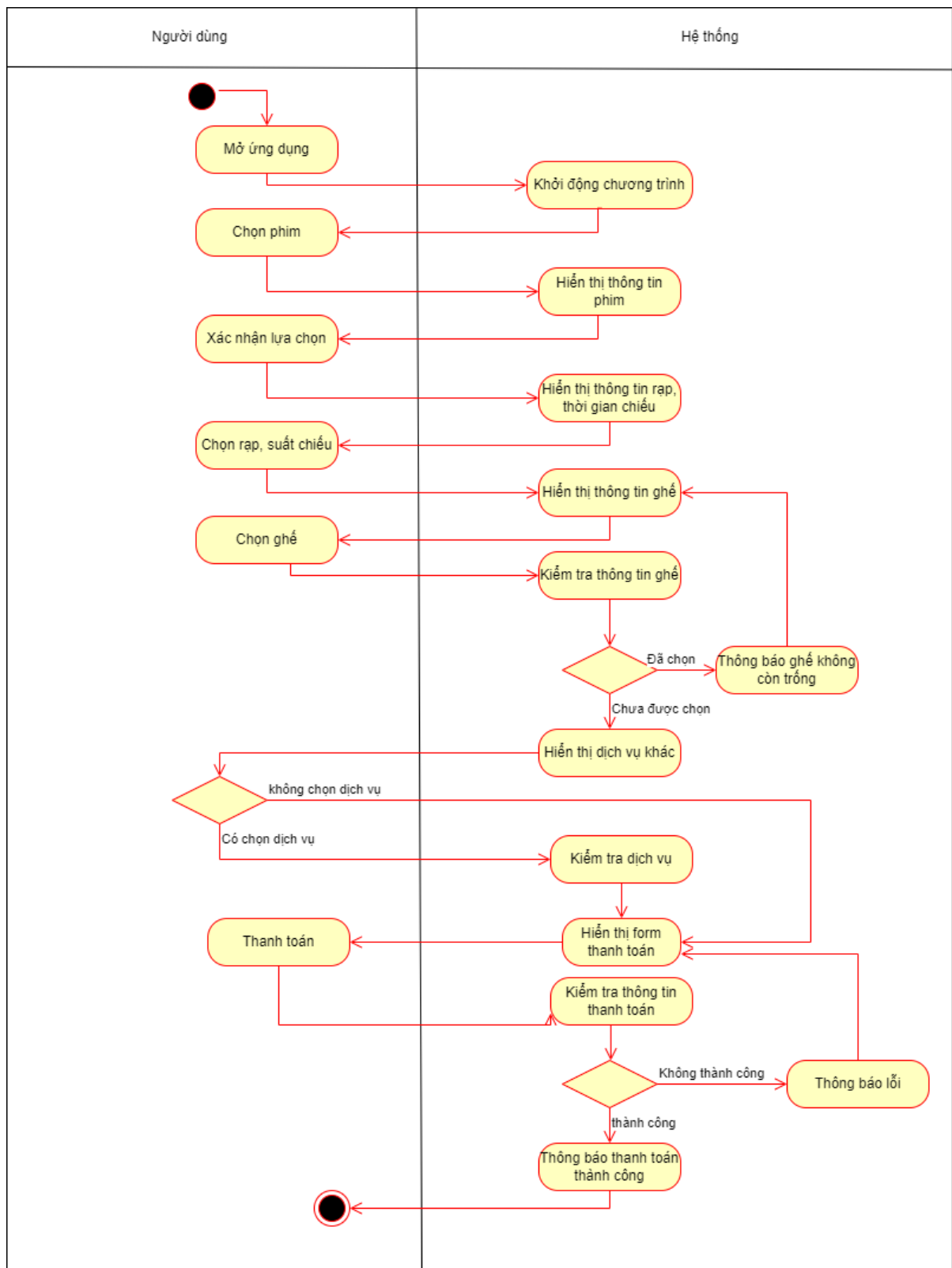


Hình 2.3.3. 1. Sơ đồ Use Case chức năng “Đặt vé”

Sơ đồ này mô tả quy trình “Đặt vé” chi tiết với các bước cần thực hiện, bao gồm Chọn phim, Chọn rạp, Chọn ghế, và Chọn combo. Tất cả các bước này được liên kết với hành động chính Đặt vé thông qua quan hệ <<include>>, cho thấy các bước này là cần thiết hoặc có thể xảy ra trong quy trình đặt vé.

Mỗi quan hệ <<include>> chỉ ra rằng tất cả các hành động Chọn phim, Chọn rạp, Chọn ghế, và Chọn combo đều là một phần của quy trình Đặt vé. Điều này có nghĩa là người dùng phải thực hiện tất cả các bước trên để hoàn tất quy trình đặt vé.

2.3.2.2. Sơ đồ hoạt động chức năng “Đặt vé”



Hình 2.3.2.2. 1. Sơ đồ hoạt động chức năng “Đặt vé”

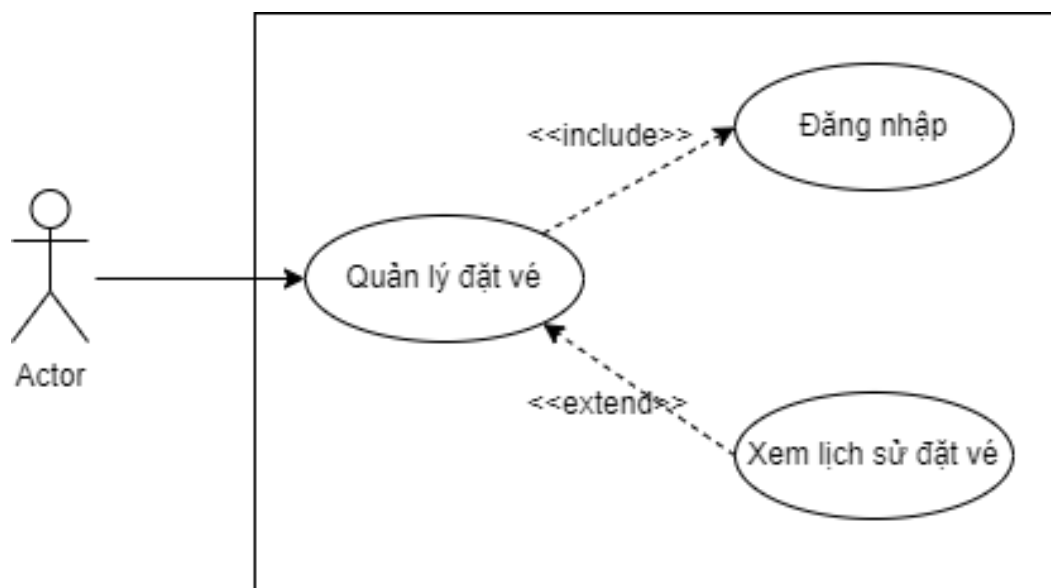
2.3.3.3. Đặc tả use case chức năng “Đặt vé”

Bảng 2.3.3.3. 1. Đặc tả use case chức năng “Đặt vé”

Tên Use case		Đặt vé
Tác nhân thực hiện		Khách hàng
Mục đích		Đặt vé xem phim
Tiền điều kiện thực hiện		Ca sử dụng bắt đầu khi người dùng đăng nhập thành công vào hệ thống
Hậu điều kiện	Thành công	Thực hiện chọn phim, rạp, ghế, combo thành công
	Lỗi	Đưa ra thông báo lỗi nếu xuất hiện lỗi không thực hiện được
Luồng nghiệp vụ chính/kịch bản chính		<ol style="list-style-type: none"> 1. Mở ứng dụng <ol style="list-style-type: none"> 1.1. Chọn phim 1.2. Chọn rạp 1.3. Chọn ghế 1.4. Chọn combo 2. Xác nhận các thông tin và chọn đặt vé 3. Hiện thị thông báo kết quả đặt vé thành công 4. Thoát
Luồng nghiệp vụ chính/kịch bản phụ		Hệ thống hiện thị thông báo lỗi khi thông tin vé không hợp lệ hoặc có vấn đề trong quá trình đặt vé (ví dụ: ghế đã được đặt trước, thông tin phim không có sẵn, ...).

2.3.4. Đặc tả use case chức năng “Quản lý đặt vé”

2.3.4.1. Sơ đồ Use Case chức năng “Quản lý đặt vé”



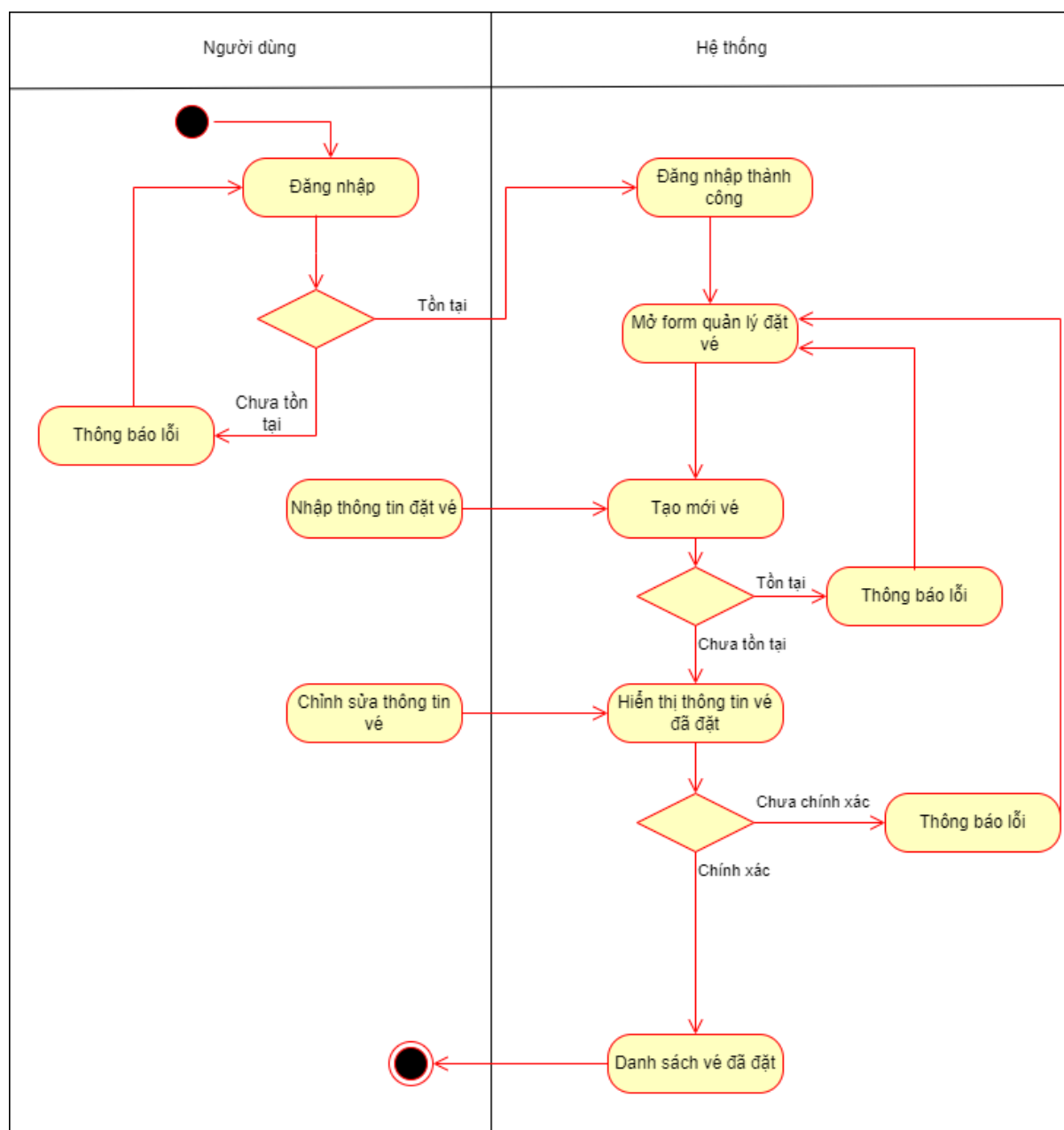
Hình 2.3.1. 2. Sơ đồ Use Case chức năng “Quản lý đặt vé”

Quản lý đặt vé là chức năng chính mà Actor thực hiện. Trường hợp sử dụng "Quản lý đặt vé" là hành động chính mà Actor tham gia.

Đăng nhập: là một trường hợp sử dụng phụ mà quản lý đặt vé yêu cầu. Sơ đồ này dùng ký hiệu <<include>> để chỉ rằng việc đăng nhập là bắt buộc khi thực hiện quản lý đặt vé. Điều này có nghĩa là chức năng quản lý đặt vé luôn cần phải bao gồm bước đăng nhập.

Xem lịch sử đặt vé: là một chức năng tùy chọn mở rộng, không bắt buộc. Ký hiệu <<extend>> cho thấy rằng Actor có thể mở rộng hành vi của quản lý đặt vé bằng cách thực hiện thêm chức năng xem lịch sử đặt vé, nhưng điều này không phải lúc nào cũng xảy ra.

2.3.4.2. Sơ đồ hoạt động chức năng “Quản lý đặt vé”



Hình 2.3.4.2. Sơ đồ hoạt động chức năng “Quản lý đặt vé”

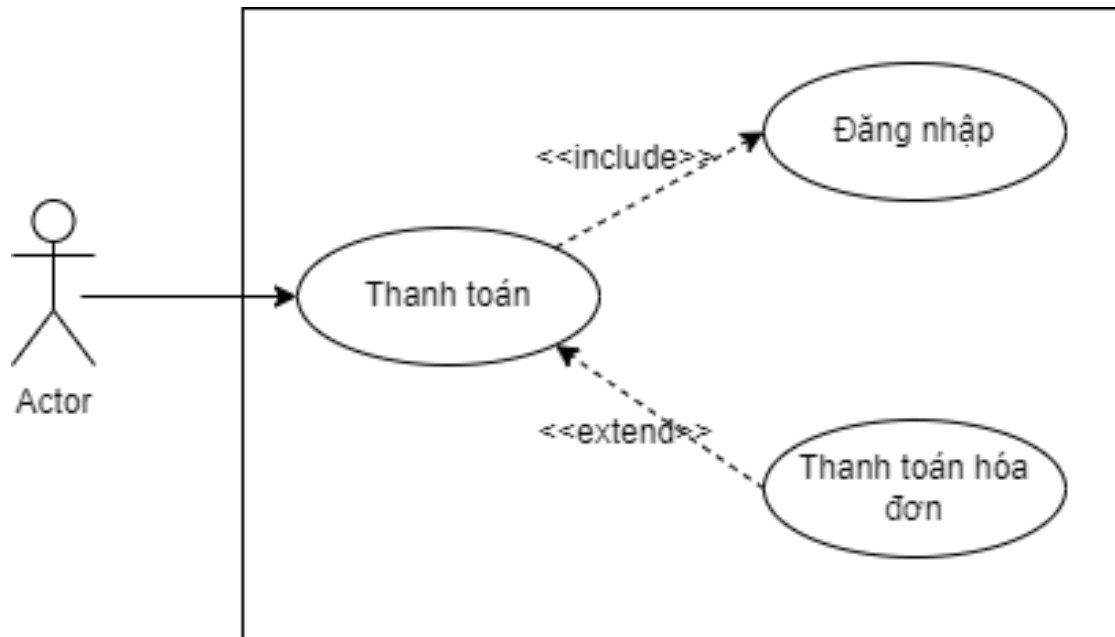
2.3.4.3. Đặc tả use case chức năng “Quản lý đặt vé”

Bảng 2.3.4.3. Đặc tả use case chức năng “Quản lý đặt vé”

Tên Use case		Quản lý đặt vé
Tác nhân thực hiện		Người quản lý
Mục đích		Quản lý đặt vé(bao gồm thông tin vé: tên phim, thời gian, địa điểm, chỗ ngồi, giá vé)
Tiền điều kiện thực hiện		Người dùng đã đăng nhập
Hậu điều kiện	Thành công	Quản lý đặt vé thành công, thông tin vé được cập nhật vào hệ thống
	Lỗi	Hệ thống đưa ra thông báo lỗi nếu có sự cố khi thực hiện thao tác
Luồng nghiệp vụ chính/kịch bản chính		<ol style="list-style-type: none"> Mở giao diện "Quản lý đặt vé" cho phép thực hiện các chức năng quản lý vé. <ol style="list-style-type: none"> Tạo mới vé: Nhập thông tin vé: tên phim, thời gian chiếu, địa điểm chiếu, chỗ ngồi, giá vé. Chỉnh sửa vé đã đặt: Cập nhật thông tin vé như thời gian, chỗ ngồi, hoặc hủy vé. Xem danh sách vé đã đặt: Hiển thị danh sách các vé đã đặt theo thông tin người dùng, phim, thời gian, hoặc trạng thái vé. Hệ thống hiển thị thông báo thành công khi thao tác hoàn tất. Thoát khỏi giao diện quản lý đặt vé.
Luồng nghiệp vụ chính/kịch bản phụ		<ol style="list-style-type: none"> Nếu thông tin vé không hợp lệ (ví dụ: thời gian chiếu đã qua, chỗ ngồi đã được đặt), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng sửa lại thông tin. Nếu việc tạo, chỉnh sửa, hoặc hủy vé không thành công (do lỗi hệ thống hoặc thông tin không hợp lệ), hệ thống sẽ thông báo lỗi.

2.3.5. Đặc tả use case chức năng “Thanh toán”

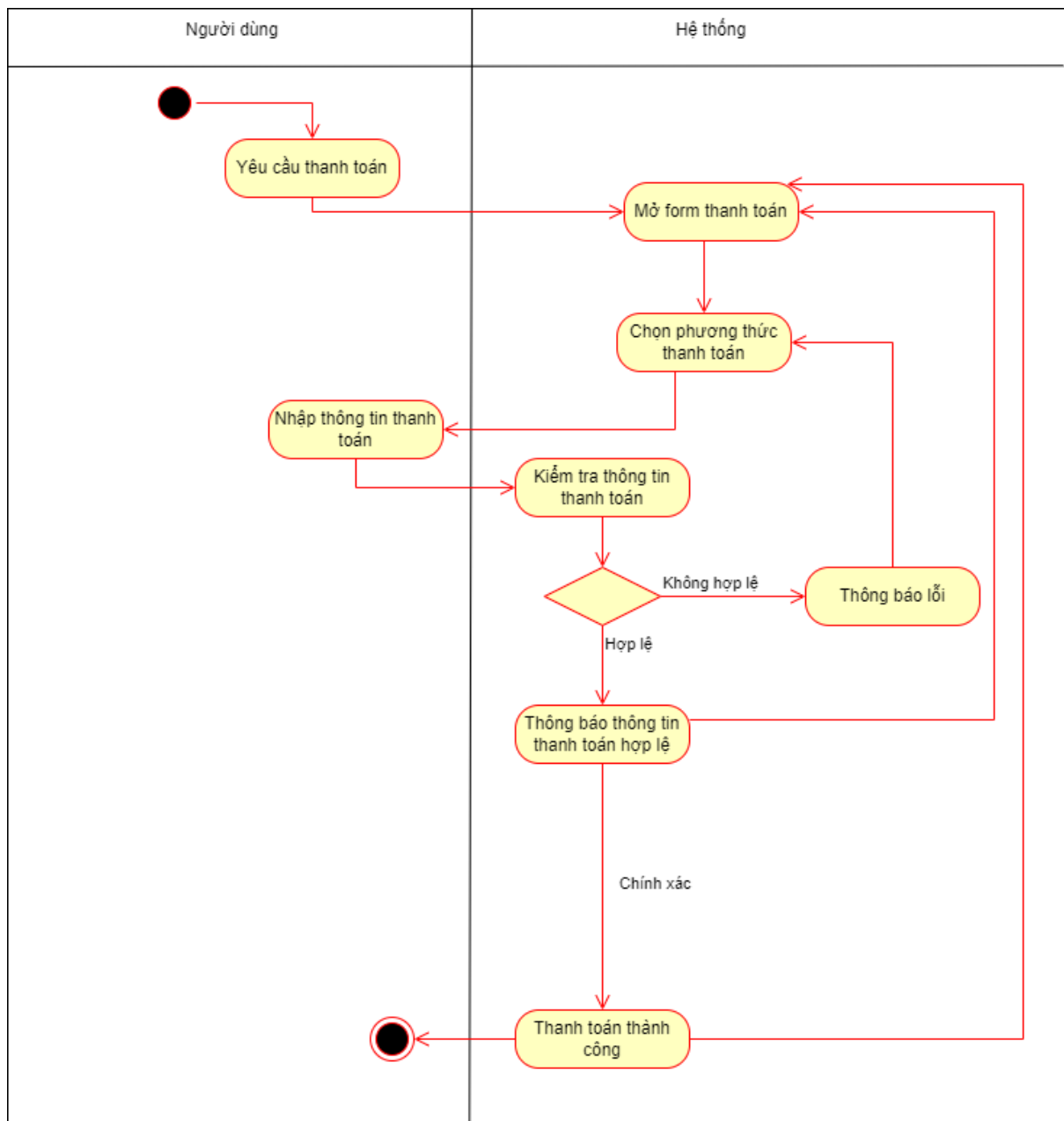
2.3.5.1. Sơ đồ Use Case chức năng “Thanh toán”



Hình 2.3.1. 3. Sơ đồ Use Case chức năng “Thanh toán”

Đăng nhập: là một trường hợp sử dụng phụ mà thanh toán yêu cầu. Sơ đồ này dùng ký hiệu <<include>> để chỉ rằng việc đăng nhập là bắt buộc khi thực hiện quản lý đặt vé. Điều này có nghĩa là chức năng quản lý đặt vé luôn cần phải bao gồm bước đăng nhập.

2.3.5.1. Sơ đồ hoạt động chức năng “Thanh toán”



Hình 2.3.5.2. Sơ đồ hoạt động chức năng “Thanh toán”

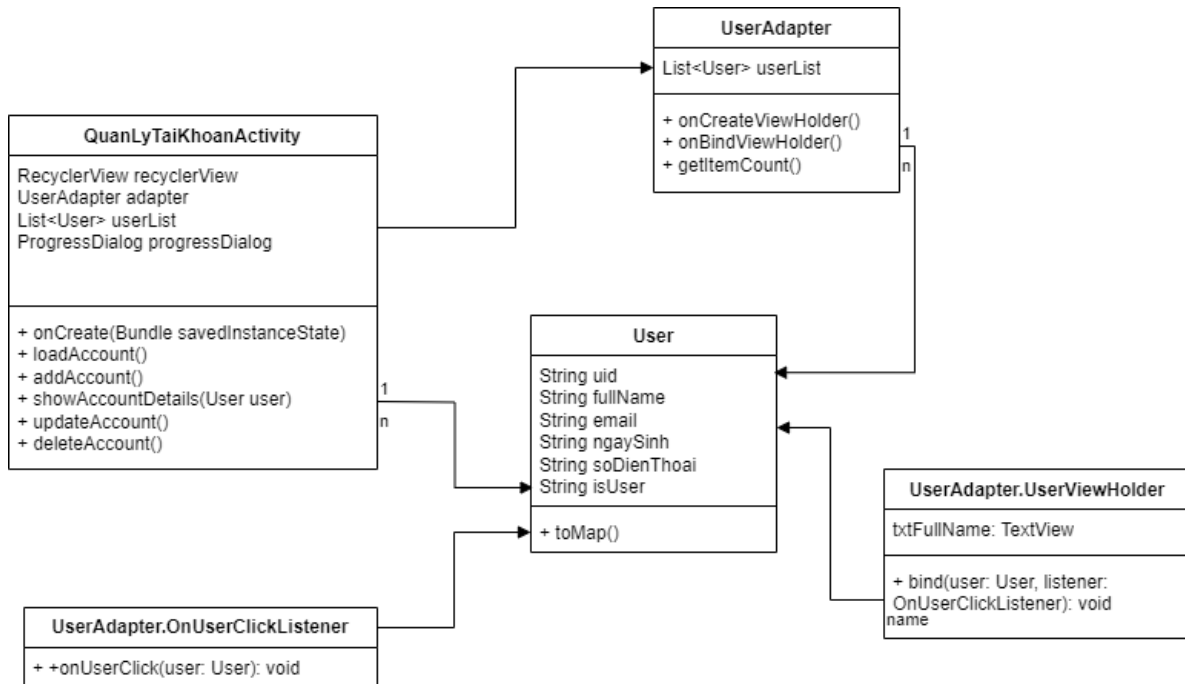
2.3.5.3. Đặc tả use case chức năng “Thanh toán”

Bảng 2.3.5.3. Đặc tả use case chức năng “Thanh toán”

Tên Use case		Quản lý thanh toán
Tác nhân thực hiện		Người quản lý
Mục đích		Thực hiện thanh toán cho vé (bao gồm thông tin: tổng giá trị vé, phương thức thanh toán, trạng thái thanh toán)
Tiền điều kiện thực hiện		Người dùng đã đăng nhập và đã chọn vé cần thanh toán
Hậu điều kiện	Thành công	Thanh toán thành công, thông tin thanh toán được cập nhật vào hệ thống
	Lỗi	Hệ thống đưa ra thông báo lỗi nếu có sự cố khi thực hiện thanh toán
Luồng nghiệp vụ chính/kịch bản chính		<ol style="list-style-type: none"> Người dùng mở giao diện "Thanh toán" trên hệ thống sau khi chọn vé. <ol style="list-style-type: none"> Chọn phương thức thanh toán: Người dùng có thể chọn một trong các phương thức thanh toán (thẻ tín dụng, ví điện tử, chuyển khoản ngân hàng, v.v.). Nhập thông tin thanh toán: Người dùng nhập thông tin cần thiết như số thẻ, ngày hết hạn, mã xác thực, hoặc các thông tin liên quan đến phương thức thanh toán. Xác nhận thanh toán: Người dùng kiểm tra lại thông tin và nhấn nút "Xác nhận thanh toán". Hệ thống xử lý giao dịch và hiển thị thông báo thành công nếu thanh toán hoàn tất. Người dùng thoát khỏi giao diện thanh toán.
Luồng nghiệp vụ chính/kịch bản phụ		<ol style="list-style-type: none"> Nếu thông tin thanh toán không hợp lệ (ví dụ: số thẻ không đúng, tài khoản không đủ tiền, hoặc thông tin không đầy đủ), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin. Nếu quá trình thanh toán gặp lỗi (ví dụ: kết nối không ổn định, hệ thống thanh toán gặp sự cố), hệ thống sẽ thông báo lỗi và yêu cầu người dùng thử lại sau.

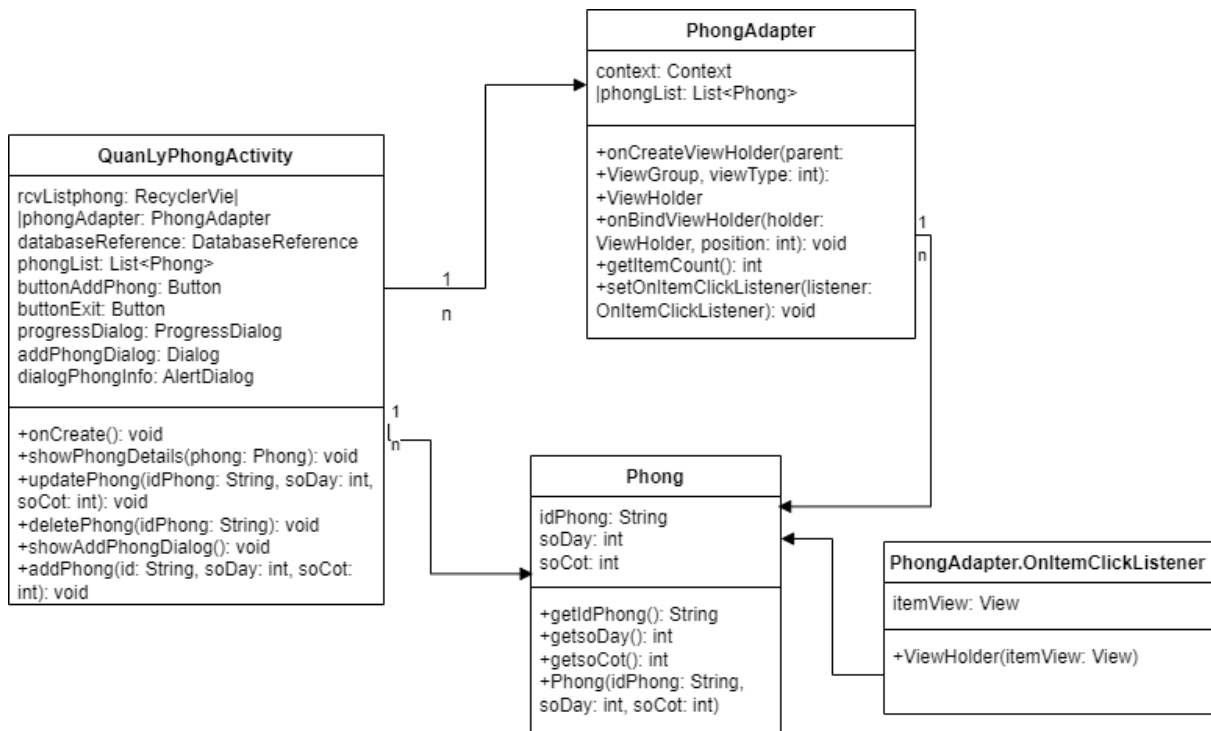
2.4. Biểu đồ lớp hệ thống

2.4.1. Biểu đồ lớp tài khoản



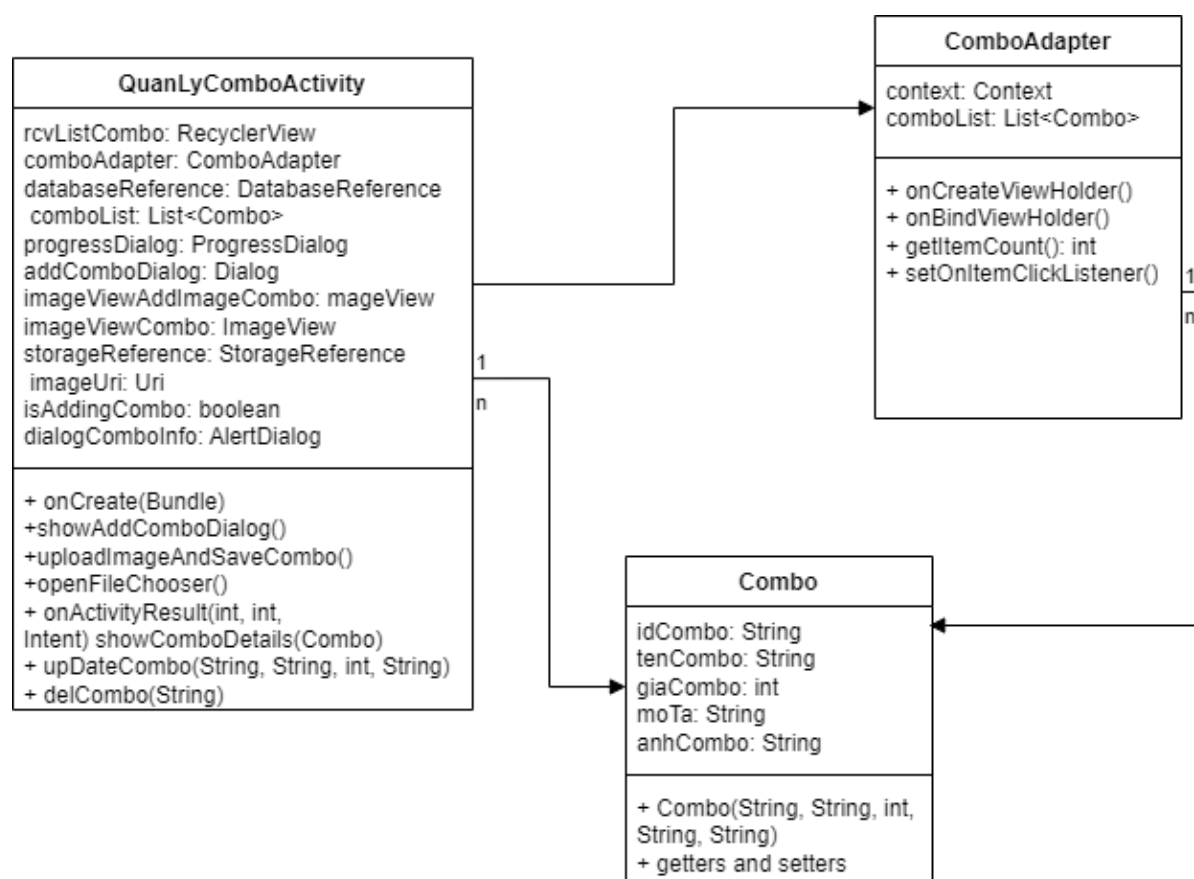
Hình 2.4.1. 1. Biểu đồ lớp TaiKhoan

2.4.2. Biểu đồ lớp phòng



Hình 2.4.2. 1. Biểu đồ lớp Phong

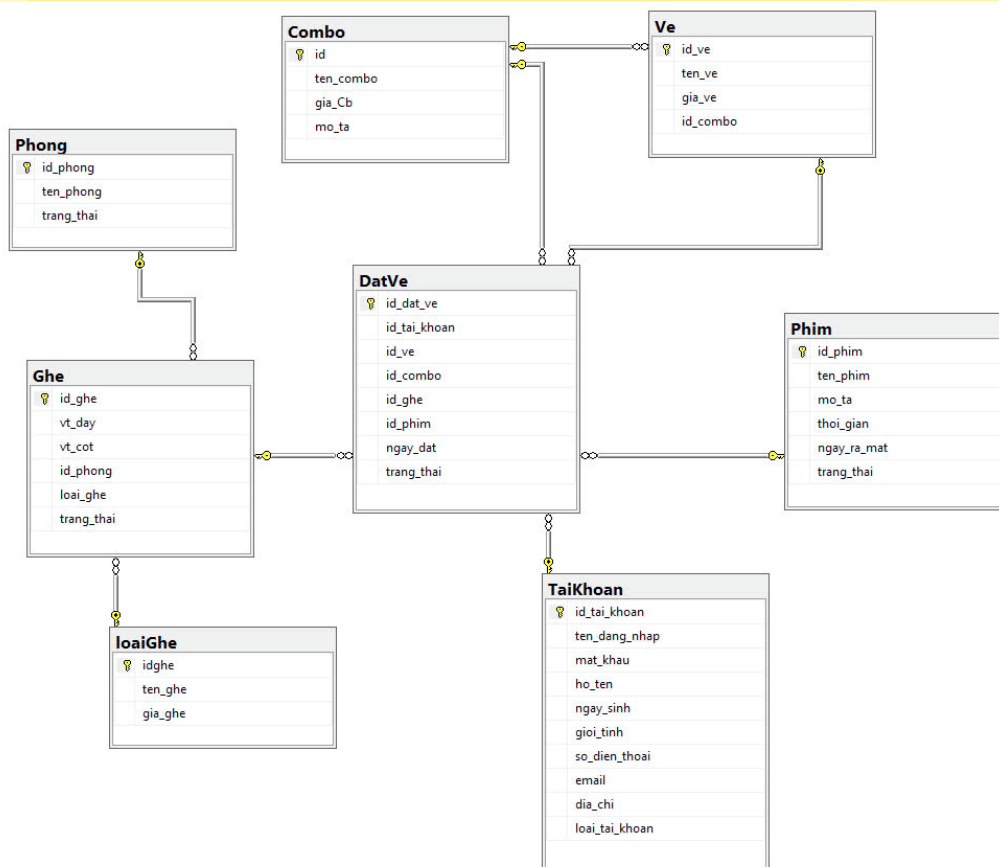
2.4.3. Biểu đồ lớp combo



Hình 2.4.3. 1. Biểu đồ lớp Combo

2.5. Thiết kế cơ sở dữ liệu

2.5.1. Biểu đồ cơ sở dữ liệu hệ thống



Hình 2.5.1. 1. Biểu đồ cơ sở dữ liệu tổng quát

2.5.2. Thiết kế database

Bảng 2.5.2. 1. Phim

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id_phim	Int	không	Mã phim
2	Ten_phim	Nvarchar	100	Tên phim
3	Mo_ta	Nvarchar	255	Mô tả phim
4	Ngay_ra_mat	Date	Không	Ngày ra mắt phim
5	Trang_thai	Nvarchar	50	Trạng thái phim

Bảng 2.5.2. 2. Phòng

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id_phong	Int	không	Mã phòng
2	ten_phong	Nvarchar	50	Tên phòng
3	trang_thai	Nvarchar	50	Trạng thái phòng

Bảng 2.5.2. 3. Loại ghế

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	idghe	Int	không	Mã loại ghế
2	ten_ghe	Nvarchar	50	Tên ghế
3	gia_ghe	int	Không	Giá ghế

Bảng 2.5.2. 4. Ghế

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id_ghe	Int	không	Mã ghế
2	vt_day	Nvarchar	50	Vị trí dãy
3	vt_cot	Int	Không	Vị trí cột
5	trang_thai	Nvarchar	50	Trạng thái ghế

Bảng 2.5.2. 5. Combo

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id	Int	không	Mã Combo
2	ten_combo	Nvarchar	100	Tên Combo
3	mo_ta	Nvarchar	255	Mô tả Combo
4	gia_cd	Int	Không	Giá Combo

Bảng 2.5.2. 6. Vé

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id_ve	Int	không	Mã vé
2	ten_ve	Nvarchar	100	Tên vé
3	gia_ve	Int	Không	Giá vé

Bảng 2.5.2. 7. Tài khoản

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id_tai_khoan	Int	không	Mã tài khoản
2	ten_dang_nhap	Nvarchar	100	Tên đăng nhập
3	mat_khau	Nvarchar	50	Mật khẩu
4	ho_ten	Nvarchar	50	Họ và tên
5	trang_thai	Nvarchar	50	Trạng thái phim
6	gioi_tinh	Nvarchar	50	Giới tính
7	ngay_sinh	Date	Không	Ngày sinh
8	so_dien_thoai	Int	Không	Số điện thoại
9	loai_tai_khoan	Nvarchar	50	Loại tài khoản

Bảng 2.5.2.8. Đặt vé

STT	Tên thuộc tính	Kiểu dữ liệu	Kích thước	Ý nghĩa
1	id_dat_ve	Int	không	Mã đặt vé
2	ngay_dat	Nvarchar	100	Ngày đặt
5	trang_thai	Nvarchar	50	Trạng thái vé

CHƯƠNG 3. CÀI ĐẶT CHƯƠNG TRÌNH

3.1. Tổ chức chương trình

Thiết kế giao diện (UI):

Tạo các layout cần thiết: Sử dụng XML trong Android Studio để thiết kế giao diện, bao gồm màn hình đăng nhập, danh sách phim, chi tiết phim, chọn ghế ngồi, xác nhận đặt vé, v.v.

Sử dụng RecyclerView: Để hiển thị danh sách phim hoặc lịch chiếu.

Navigation Component: Để quản lý điều hướng giữa các màn hình

Tích hợp API và dịch vụ:

Firebase: Dùng cho xác thực người dùng, lưu trữ hình ảnh hoặc thông báo đẩy.

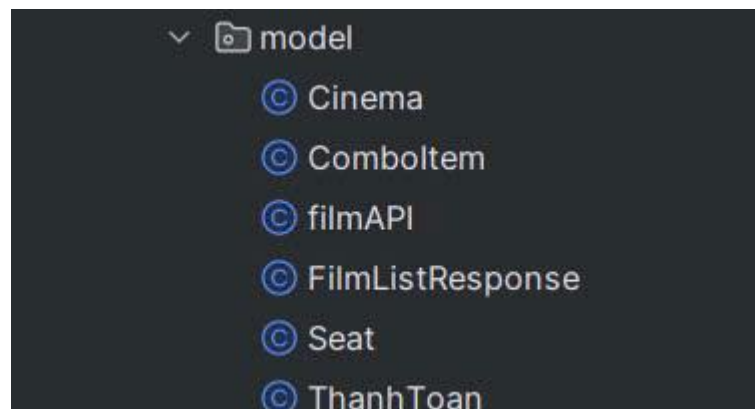
Retrofit: Giao tiếp với API để lấy dữ liệu phim, lịch chiếu.

3.1.1. Gói model

Chức năng: Đảm nhận việc xử lý dữ liệu và các logic nghiệp vụ của hệ thống.

Vai trò: Model chịu trách nhiệm truy xuất, lưu trữ, và xử lý dữ liệu. Nó đại diện cho các đối tượng dữ liệu trong ứng dụng và thường bao gồm các quy tắc nghiệp vụ (business logic).

Thành phần chính: Các lớp mô hình dữ liệu, các phương thức truy vấn, xử lý dữ liệu, kết nối cơ sở dữ liệu (Database), hoặc API backend.



Hình 3.1.1. Gói model phân người dùng

3.1.2. Gói View

Chức năng: Hiển thị thông tin và tương tác với người dùng.

Vai trò: View lấy dữ liệu từ Model và hiển thị chúng cho người dùng dưới dạng giao diện đồ họa (UI). Nó cũng nhận các thao tác từ người dùng và chuyển đến Controller để xử lý.

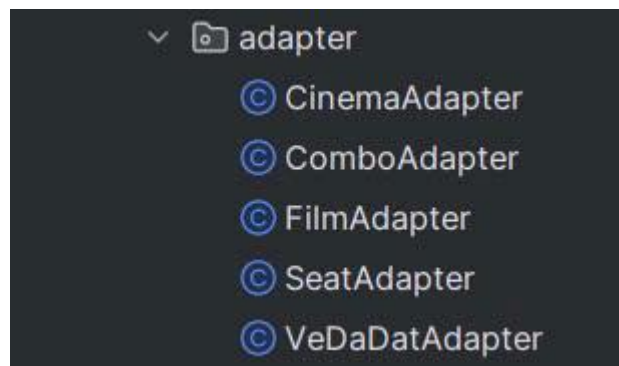
Thành phần chính: Các thành phần UI như XML layout trong Android, v.v.

3.1.3. Gói adapter

Chức năng: Điều phối luồng xử lý giữa Model và View.

Vai trò: Controller nhận các yêu cầu từ View, xử lý các yêu cầu đó bằng cách gọi các phương thức từ Model, và sau đó cập nhật lại View với dữ liệu từ Model.

Thành phần chính: Các lớp xử lý yêu cầu, điều hướng (navigation), kiểm soát sự kiện người dùng (user interaction), v.v.



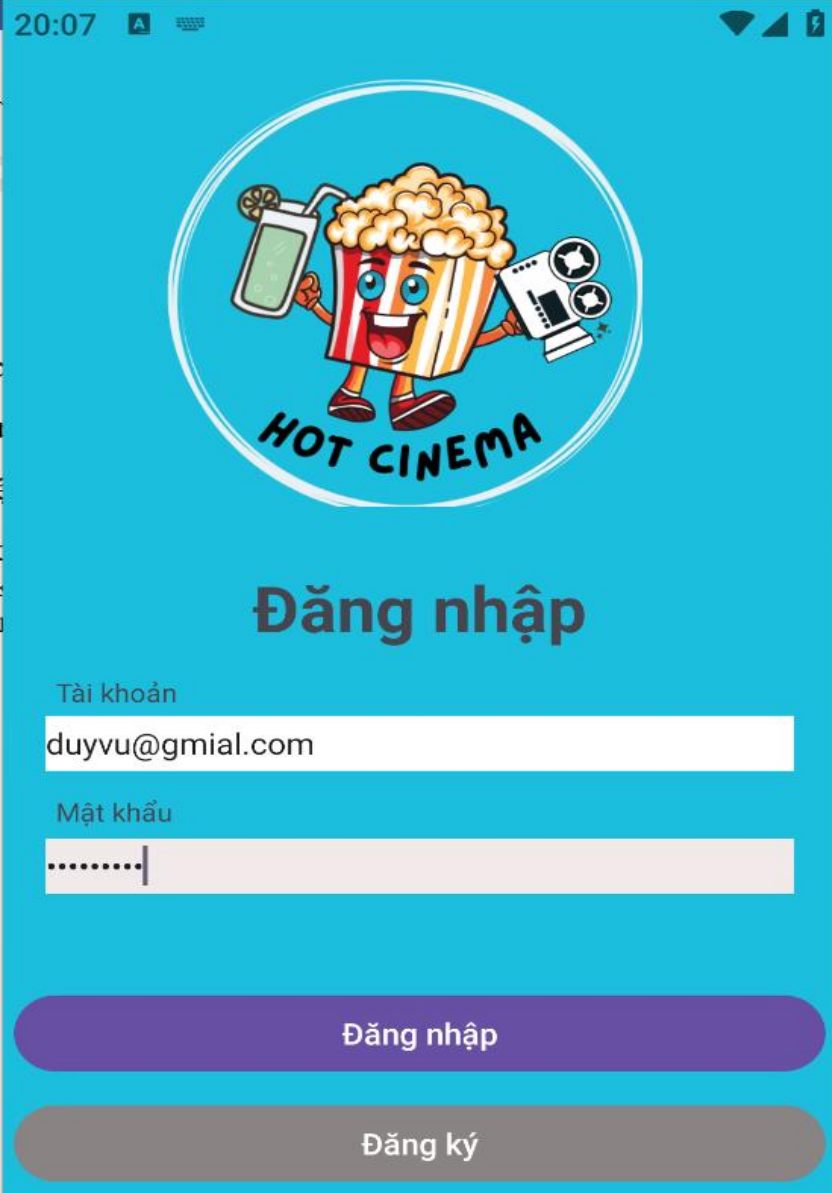
Hình 3.1.3. Gói adapter phần người dùng

3.2. Giao diện chính của chương trình

3.2.1. Giao diện người dùng

Quản lý tài khoản:

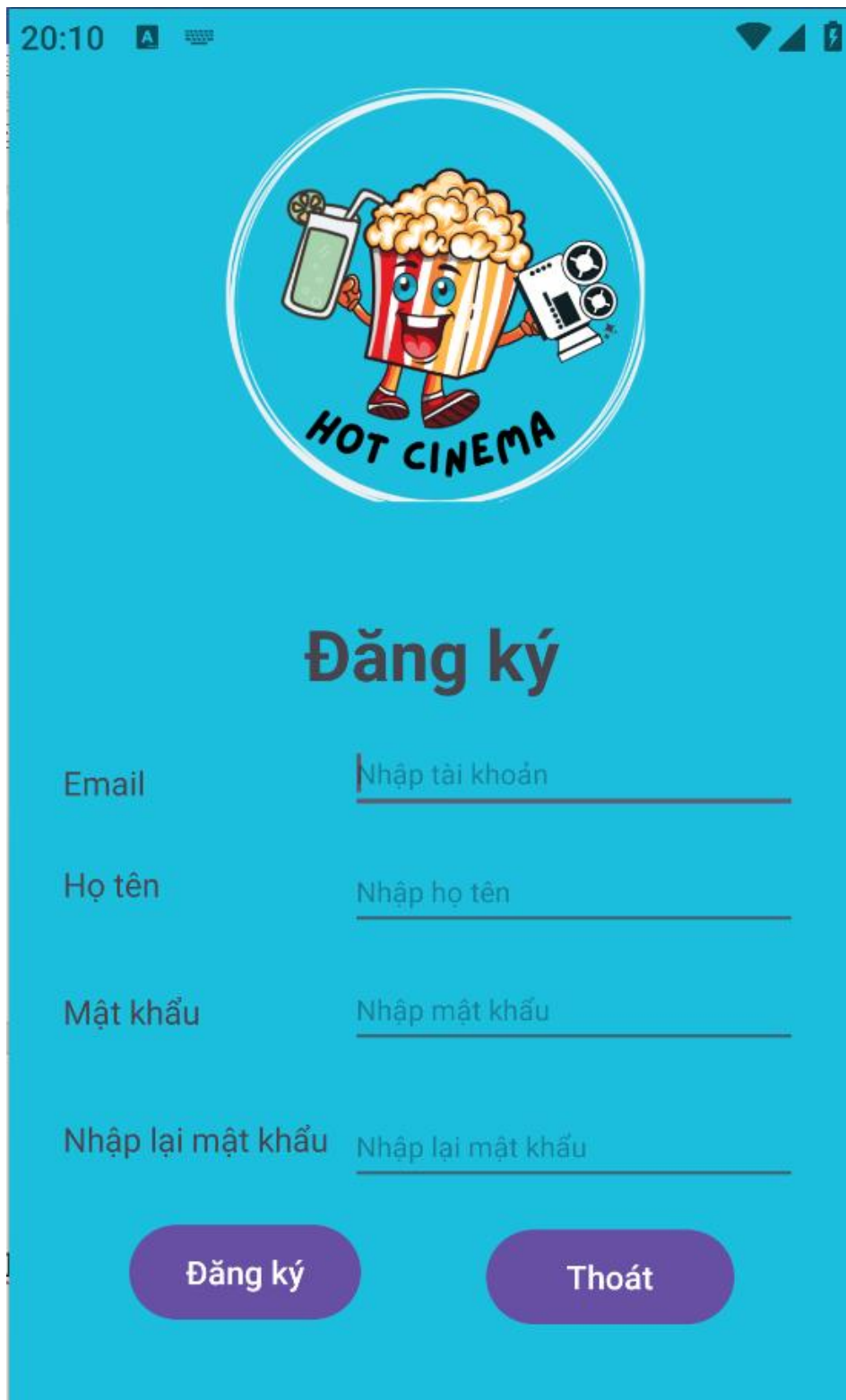
Form đăng nhập:



The image shows a mobile application interface for 'HOT CINEMA'. At the top, there is a status bar with the time 20:07 and various icons. Below the status bar is a circular logo featuring a cartoon popcorn character holding a smartphone and a clapperboard, with the text 'HOT CINEMA' underneath. The main heading is 'Đăng nhập' (Login). Below this, there are two input fields: 'Tài khoản' (Username) with the value 'duyvu@gmial.com' and 'Mật khẩu' (Password) with a masked input. At the bottom, there are two buttons: a purple 'Đăng nhập' button and a grey 'Đăng ký' (Register) button.

Hình 3.2.1. 1. Form “Đăng nhập”

Form đăng ký:



The image shows a mobile application registration form on a blue background. At the top, there is a status bar with the time 20:10 and icons for signal, Wi-Fi, and battery. Below the status bar is a circular logo featuring a smiling popcorn character holding a smartphone and a movie camera, with the text "HOT CINEMA" underneath. The main title "Đăng ký" (Registration) is centered in a large, bold font. Below the title are four input fields, each with a label on the left and a placeholder text on the right: "Email" with "Nhập tài khoản", "Họ tên" with "Nhập họ tên", "Mật khẩu" with "Nhập mật khẩu", and "Nhập lại mật khẩu" with "Nhập lại mật khẩu". At the bottom, there are two purple buttons: "Đăng ký" (Register) and "Thoát" (Exit).

20:10

Đăng ký

Email

Họ tên

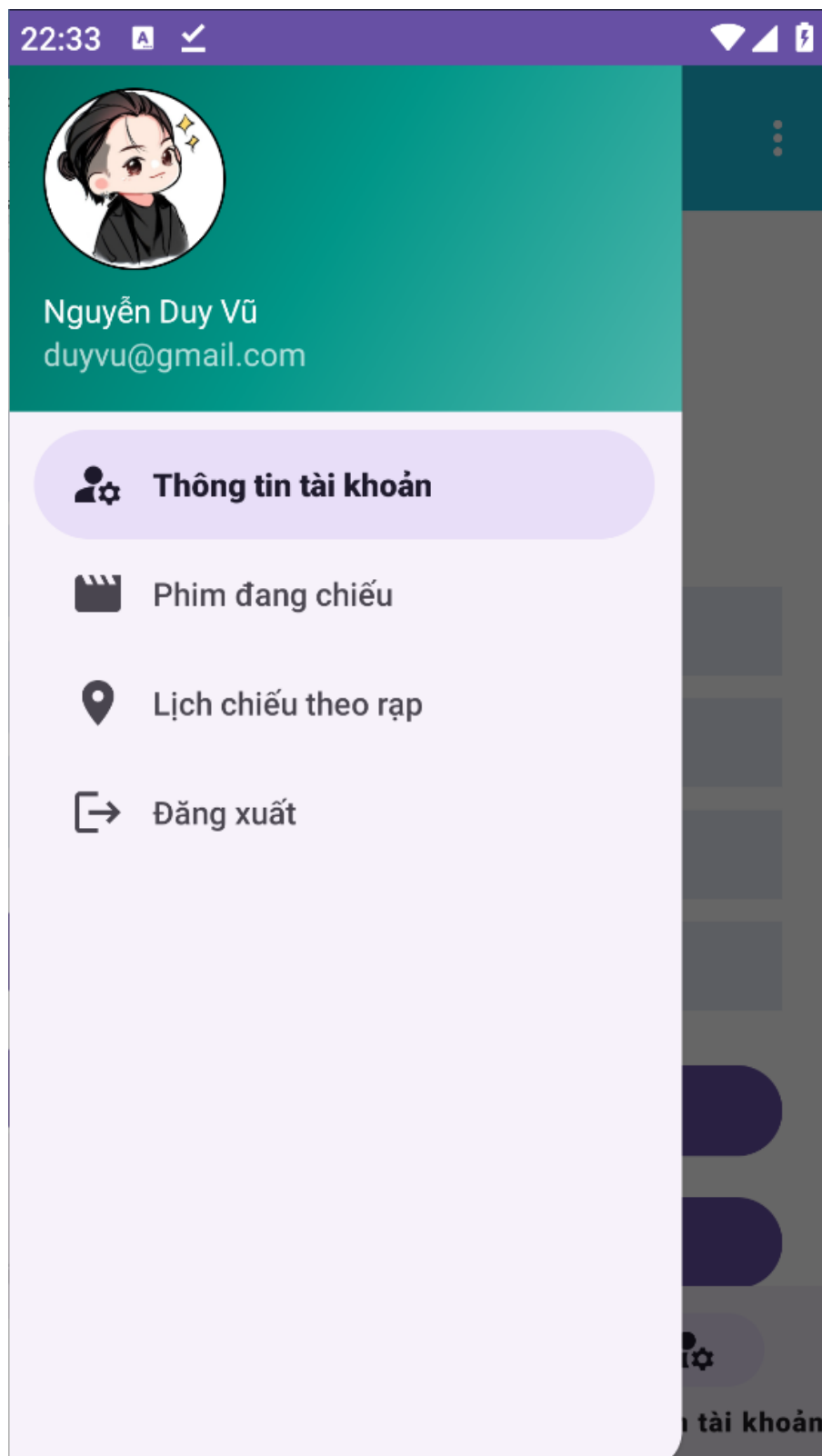
Mật khẩu

Nhập lại mật khẩu

Đăng ký **Thoát**

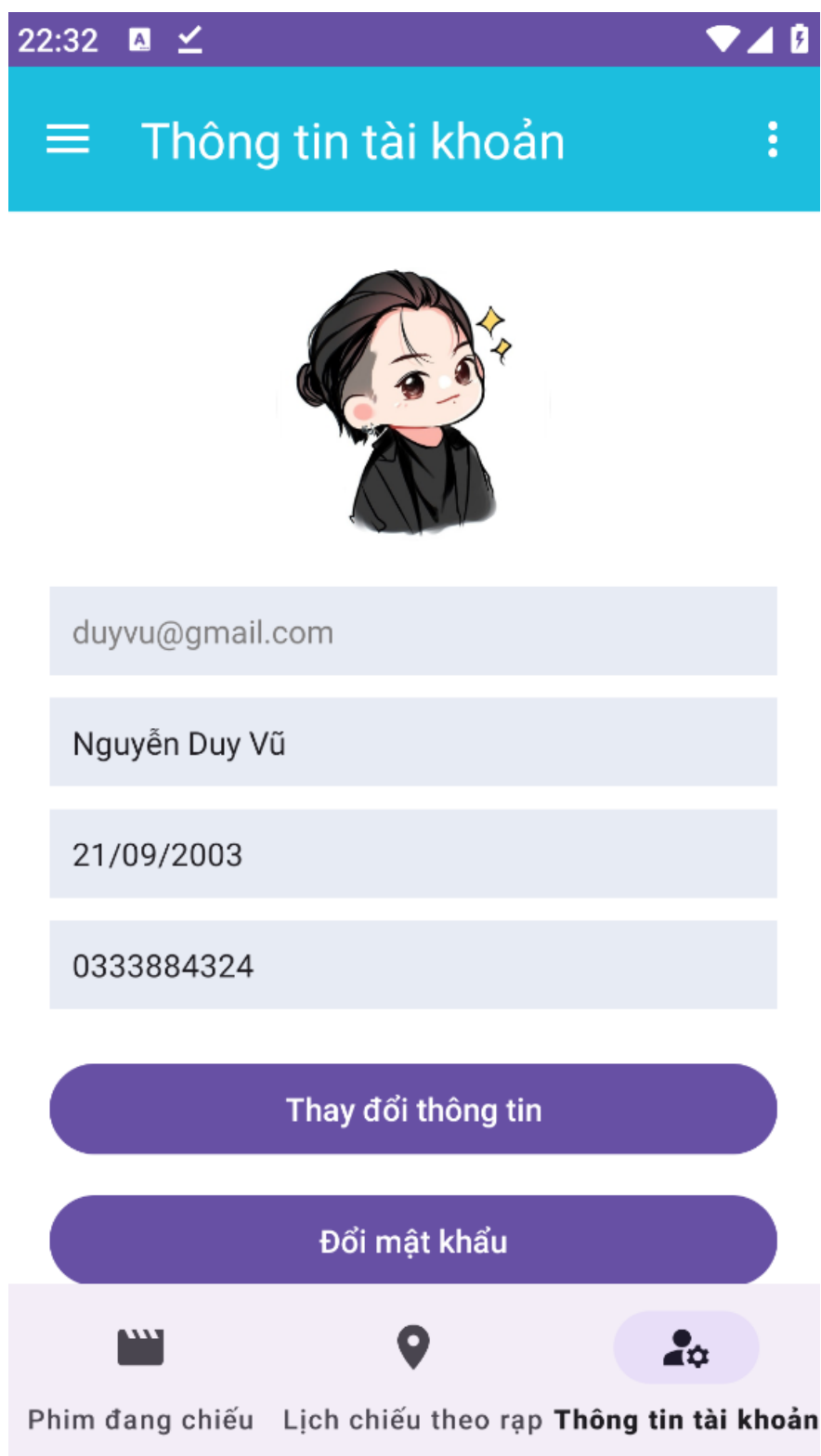
Hình 3.2.1. 2. Form đăng ký

Form menu HotCinema:




Hình 3.2.1. 3. Form “Menu HotCinema”

Form thông tin tài khoản:



22:32

≡ Thông tin tài khoản



duyvu@gmail.com

Nguyễn Duy Vũ

21/09/2003

0333884324

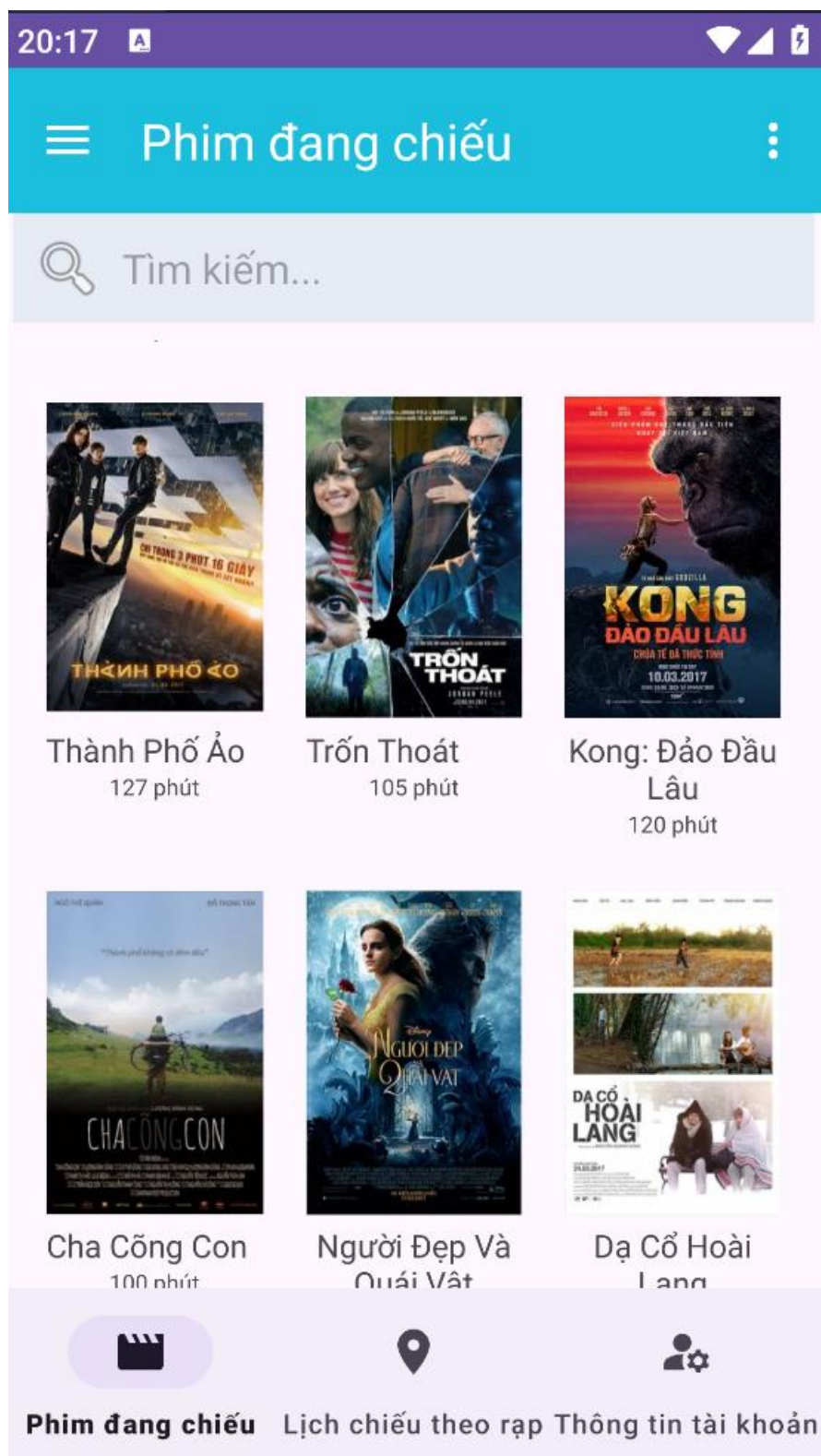
Thay đổi thông tin

Đổi mật khẩu

Phim đang chiếu Lịch chiếu theo rạp **Thông tin tài khoản**

Hình 3.2.1. 4. Form “Thông tin tài khoản”

Form chọn phim:



Hình 3.2.1. 5. Form “Chọn phim”

Form chọn rạp:

20:14

Đặt vé

Quá Nhanh Quá Nguy Hiểm 8

160 phút

Chi tiết phim

26/09
Thứ năm

27/09
Thứ sáu

28/09
Thứ bảy

29/09
Chủ nhật

Chọn khu vực

Hà Nội

CGV Aeon Long Biên

Tầng 4 - TTTM AEON Long Biên, Số 27 Cổ Linh,
Quận Long Biên, Hà Nội

CGV Hồ Gươm Plaza

Tầng 3, TTTM Hồ Gươm Plaza, 110 Trần Phú,
Phường Mỗ Lao, Quận Hà Đông, Hà Nội

CGV IPH Hà Nội

CGV MIPEC Tower

Hình 3.2.1. 6. Form “Chọn rạp”

Form chọn chỗ ngồi:

20:21

Chọn chỗ ngồi

Trốn Thoát

22:00 | null, null
CGV Aeon Long Biên

Ghế trống Ghế đang được giữ Ghế đã bán

Ghế thường 45.000đ Ghế VIP 50.000đ Ghế đôi 120.000đ

Màn hình chiếu

A01 A02 A03 A04 A05 A06 A07 A08 A09 A10
B01 B02 B03 B04 B05 B06 B07 B08 B09 B10
C01 C02 C03 C04 C05 C06 C07 C08 C09 C10
D01 D02 D03 D04 D05 D06 D07 D08 D09 D10
E01 E02 E03 E04 E05

Ghế đã chọn: B03

Tổng tiền: 45.000 đ

Tiếp tục

Hình 3.2.1. 7. Form “Chọn chỗ ngồi”

Form chọn combo:

20:26

Thanh toán

Trốn Thoát

105 phút

Rạp chiếu:

CGV Hồ Gươm Plaza

Ngày chiếu:

Thứ sáu, 27/09


Giờ chiếu:

22:30

Ghế:

Ghế đã chọn:
C08

COMBO ƯU ĐÃI HOT




Combo See -tinh

TIẾT KIỆM 56K!!! càng nhanh càng rẻ

59.000 đ

- 0 +




Combo FA

TIẾT KIỆM 10K!!! sở hữu ngay: 1 nước + 1 bắp

50.000 đ

- 0 +



Combo Family

TIẾT KIỆM 30K!!! sở hữu ngay: 2 nước + 2 bắp + combo cơm

Tổng tiền:

50.000 đ

Thanh Toán

Hình 3.2.1. 8. Form “Chọn combo”

Form thanh toán hóa đơn:

20:28

Thanh toán

Trốn Thoát

105 phút

Thông tin vé đã đặt

Email: duyvu@gmail.com
Tên phim: Trốn Thoát
Rạp: CGV Hồ Gươm Plaza
Địa chỉ rạp: Tầng 3, TTTM Hồ Gươm Plaza, 110 Trần Phú, Phường Mỗ Lao, Quận Hà Đông, Hà Nội
Ngày giờ chiếu: Thứ sáu, 27/09 22:30
Ghế đã chọn: C08,
Danh sách combo đã chọn:
Tổng số tiền: 50.000 đ



OK

 TIẾT KIỂM 30K!!! sở hữu ngay: 2 nước + 2 bắp + combo cơm

Tổng tiền: 50.000 đ

Thanh Toán

Hình 3.2.1. 9. Form “Thanh toán hóa đơn”

Form xem vé đã đặt vé:

20:31 A

Vé đã đặt

Thông tin Vé đã đặt

ID: BXtc5IEV6xMklZrYHHWKbZye29e2

Email: duyvu@gmail.com

Tên phim: Trốn Thoát

Tên rạp: CGV Hồ Gươm Plaza

Thời gian: 22:30 | Thứ sáu, 27/09

Vị trí ghế: C08,

Combo:

Thành tiền: 50.000 VND

Thoát

Hình 3.2.1. 10. Form “Xem vé đã đặt vé”

Form lịch chiếu theo rạp:

20:36

≡ Lịch chiếu theo rạp ⋮

CGV Aeon Bình Tân
Tầng 3, Trung tâm thương mại Aeon Mall Bình Tân,
Số 1 đường số 17A, khu phố 11, phường Bình Trị
Đông B, quận Bình Tân, TPHCM

CGV Celadon Tân Phú

CGV Crescent Mall
Lầu 5, Crescent Mall Đại lộ Nguyễn Văn Linh, Phú Mỹ
Hưng Quận 7 TP. Hồ Chí Minh

CGV CT Plaza

CGV Gò Vấp

CGV Golden Plaza

CGV Hoàng Văn Thụ
Tầng 1 và 2, Gala Center, số 415, Hoàng Văn Thụ,
Phường 2, Quận Tân Bình, TPHCM

Phim đang chiếu **Lịch chiếu theo rạp** Thông tin tài khoản

Hình 3.2.1. 11. Form “Lịch chiếu theo rạp”

3.2.2. Giao diện quản lý hệ thống

Form quản lý HotCinema:



Hình 3.2.2. 1. Form “Quản lý HotCinema”

Form danh sách tài khoản:



20:52

Danh Sách tài khoản

ID: 74YssMzhRSgqb3cWg0Nule7fRG23 Email: user@gmail.com	Quyền: Người dùng
ID: BXtc5IEV6xMklZrYHHWKbZye29e2 Email: duyvu@gmail.com	Quyền: Người dùng
ID: KO3HGYJR7ZXEzdx0LtWpuk1YnhH3 Email: ca7nh9@gmail.com	Quyền: Admin
ID: LsrGFMzJkBeRe7tMCSqV7QPTSwG2 Email: 123@gmail.com	Quyền: Người dùng
ID: QwT1HgmL86fgktnUeLOzZ8yNrzb3 Email: 1234@gmail.com	Quyền: Người dùng
ID: g5FkUswFZzRuqNy99ihOK2odR5M2 Email: minh tungne@gmail.com	Quyền: Người dùng
ID: j7tUQh05MBYAy9rVQnSSBPP65Dv2 Email: minhtung76@gmail.com	Quyền: Người dùng
ID: sQBrpPHZzYYKfoSn2yeH7NnDExy2 Email: mtung12@gmail.com	Quyền: Người dùng
ID: vq4Kzzaq5uZwZOjoAS0tUppPIPk2 Email: tungdang123@gmail.com	Quyền: Người dùng

Thêm Tài Khoản

Thoát

Hình 3.2.2. 2. Form “Danh sách tài khoản”

Form thêm tài khoản:



The image shows a mobile application interface. At the top, a dark blue header bar contains the title "Danh Sách tài khoản" (Account List) in white text. Below this, a white modal form titled "Thêm Tài Khoản" (Add Account) is displayed. The form contains several input fields with labels: "Email", "PassWord", "Họ tên" (Full Name), "Nhập Ngày sinh (DD/MM/YYYY)" (Enter Date of Birth), "Số điện thoại" (Phone Number), and "Quyền (Admin/ User)". At the bottom of the form are two blue buttons labeled "Thêm" (Add) and "Thoát" (Exit). The background of the application shows a list of accounts, with visible text like "ID:", "E", and "Thoát" partially obscured by the modal.

Hình 3.2.2. 3. Form “Thêm tài khoản”

Form sửa, xóa tài khoản:

20:56

Danh Sách tài khoản

Thông tin Tài khoản

ID: BXtc5IEV6xMklZrYHHWKbZye29e2

ID: duyvu@gmail.com

ID: Nguyễn Duy Vũ

ID: 21/09/2003




ID: 0333884324

ID: Người dùng






Xóa Sửa Thoát

Hình 3.2.2. 4. Form “Sửa, xóa tài khoản”

Form danh sách Combo HotCinema:

20:57   

Danh Sách Combo HotCinema

	Combo See -tinh 59.000 đ
	Combo FA 50.000 đ
	Combo Family 120.000 đ
	Combo See-Mê 99.000 đ
	Combo Tình Yêu 69.000 đ

Thêm Combo

Thoát

Hình 3.2.2. 5. Form “Danh sách Combo HotCinema”

Form thêm Combo:

20:59 A

Danh Sách Combo HotCinema

Thêm Combo mới

ID Combo

Tên Combo

Giá Combo

Mô tả

Chọn ảnh

Thêm Thoát

Thoát

Hình 3.2.2. 6. Form “Thêm Combo”

Form sửa, xóa Combo:

21:00

Danh Sách Combo HotCinema

Thông tin Combo

3

Combo Family

120000

TIẾT KIỂM 30K!!! sở hữu ngay: 2 nước + 2 bắp + combo cơm

Chọn ảnh

BIG COMBO

Xóa Sửa Thoát

Thoát

Hình 3.2.2. 7. Form “Sửa, xóa Combo”

Form quản lý phòng:

22:51

Danh Sách Phòng Chiếu HotCinema

Phòng: 101

Số dãy: 10

Số cột: 5

Phòng: 102

Số dãy: 10

Số cột: 5

Thêm phòng

Thoát

Hình 3.2.2. 8. Form “Quản lý phòng”

Form thêm phòng:

22:51

Danh Sách Phòng Chiếu HotCinema

Phòng: 101
Số dãy: 10
Số cột: 5

Phòng:
Số dãy:
Số cột:

Thêm Phòng

ID Phòng

Số Dãy

Số Cột

Thêm Thoát

Thêm phòng

Thoát

Hình 3.2.2. 9. Form “Thêm phòng”

Form sửa, xóa phòng:

22:52

Danh Sách Phòng Chiếu HotCinema

Phòng: 101

Số dãy: 10

Thông tin Phòng

102

10

5

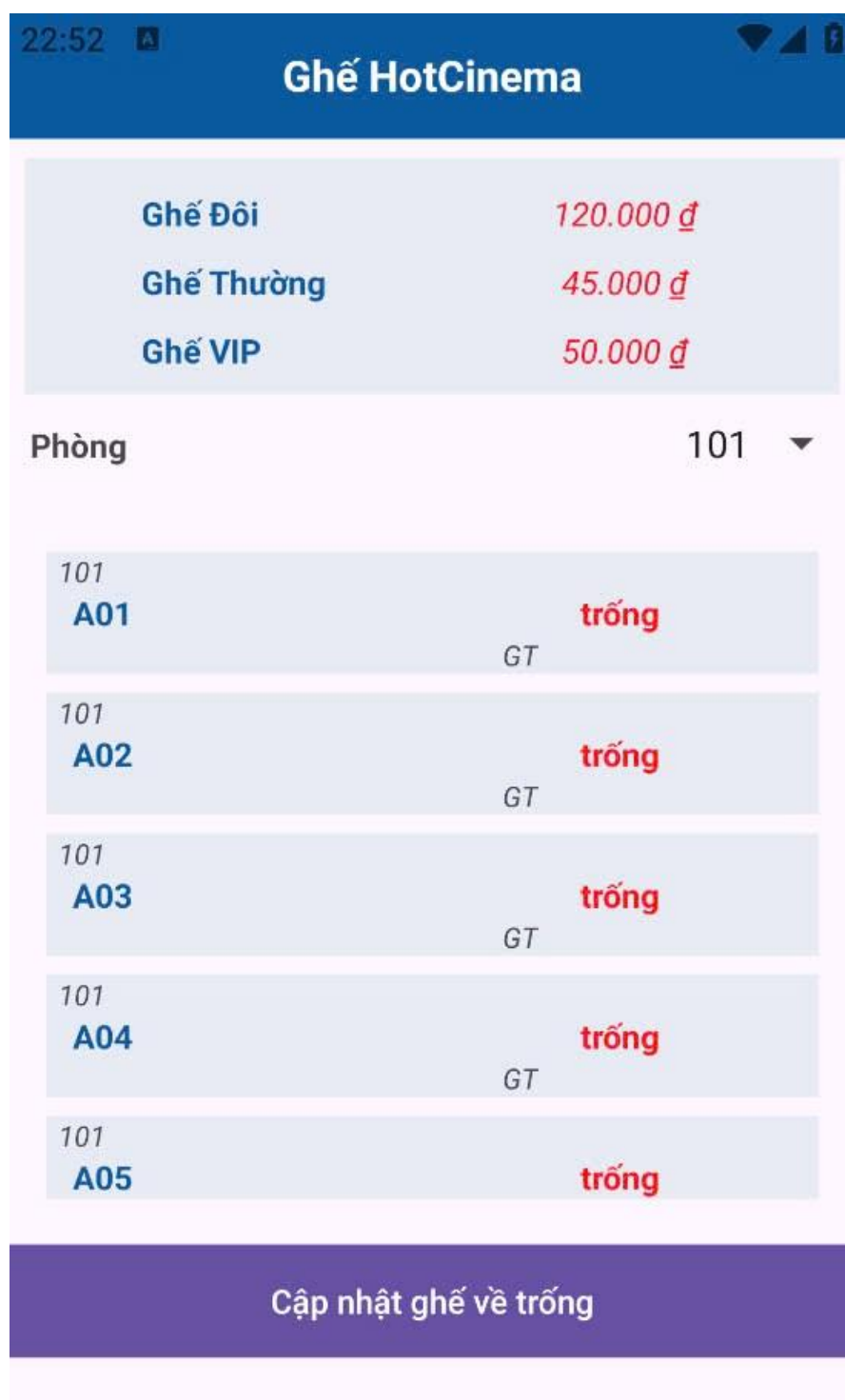
Xóa Sửa Thoát

Thêm phòng

Thoát

Hình 3.2.2. 10. Form “Sửa, xóa phòng”

Form ghế ngồi:



Ghế HotCinema

Ghế Đôi	120.000 đ
Ghế Thường	45.000 đ
Ghế VIP	50.000 đ

Phòng 101 ▼

101 A01	GT	trống
101 A02	GT	trống
101 A03	GT	trống
101 A04	GT	trống
101 A05		trống

Cập nhật ghế về trống

Hình 3.2.2. 11. Form “Ghế ngồi”

Form phòng vé:

The screenshot displays the 'Phòng vé HotCinema' app interface. At the top, a dark blue header bar contains the time '22:53', a small 'A' icon, and the app title 'Phòng vé HotCinema' in white. Below the header, the background is a light pink. Two movie listings are shown in light blue rectangular boxes. The first listing is for 'Cô Gái Và Gã Khổng Lồ' at CGV Hồ Gươm Plaza, scheduled for 21:30 on Friday, 26/09, with seats C03 and C04. The second listing is for 'Trốn Thoát' at the same location, scheduled for 22:30 on Saturday, 27/09, with seat C08. The text 'Vị trí:' is in red for both listings.

22:53 A Phòng vé HotCinema

CGV Hồ Gươm Plaza
Cô Gái Và Gã Khổng Lồ Vị trí: C03, C04,
21:30 | Thứ năm, 26/09

CGV Hồ Gươm Plaza
Trốn Thoát Vị trí: C08,
22:30 | Thứ sáu, 27/09

Hình 3.2.2. 12. Form “Phòng vé”

CHƯƠNG 4. KẾT LUẬN

Kết quả đạt được:

Sau khi nghiên cứu và thực hiện bài tập lớn với đề tài “Xây dựng ứng dụng đặt vé xem phim trên nền tảng android sử dụng android studio” sử dụng android studio, nhóm chúng em đã đạt được các kết quả sau:

Vẽ được sơ đồ use case nghiệp vụ:

Mô tả chi tiết từng use case.

Vẽ các biểu đồ hoạt động, tuần tự và cộng tác của từng use case.

Xây dựng được cơ sở dữ liệu:

Thiết kế và triển khai cơ sở dữ liệu phù hợp với yêu cầu của hệ thống.

Thiết kế được giao diện của hệ thống:

Tạo giao diện người dùng thân thiện và dễ sử dụng.

Phát triển chương trình bằng Androi studio:

Viết mã nguồn hoàn chỉnh cho hệ thống quản lý và ứng dụng đặt vé xem phim.

Xây dựng thành công ứng dụng “Đặt vé xem phim”:

Hệ thống đã có thể thực hiện các chức năng quản lý, và người dùng cơ bản.

Hạn chế:

Thiết kế giao diện còn sơ sài: Giao diện người dùng cần cải thiện để thân thiện và trực quan hơn.

Chưa tối ưu được một số chức năng trong hệ thống: Một số tính năng chưa hoạt động mượt mà hoặc chưa được tối ưu.

Chức năng thanh toán còn hạn chế: Chưa thể cập nhập khi thanh toán xong.

Hướng phát triển:

Chúng em sẽ tiếp tục phát triển và hoàn thiện hệ thống với các mục tiêu sau:

Cải thiện giao diện người dùng: Thiết kế lại giao diện để dễ sử dụng và đầy đủ chức năng hơn.

Thêm chức năng báo cáo thống kê: Phát triển chức năng cho phép quản lý doanh thu

Tối ưu hóa các chức năng hiện có: Nâng cao hiệu suất và tính ổn định của hệ thống.

Phát triển chức năng thống kê và báo cáo: Thống kê danh sách doanh thu và số lượng các combo, vé.

Bảo trì và hỗ trợ khách hàng: Đảm bảo hệ thống hoạt động ổn định và đáp ứng được nhu cầu của người dùng sau khi triển khai.

TÀI LIỆU THAM KHẢO

- [1]. Firebase là gì? Truy cập ngày 22/09/2024 từ <https://fptcloud.com/firebase-la-gi/>.
- [2]. Hệ điều hành Android là gì? Truy cập ngày 20/09/2024 từ <https://bkhost.vn/blog/he-dieu-hanh-android/>
- [3]. Tổng quan về Android studio Truy cập ngày 21/09/2024 từ <https://itnavi.com.vn/blog/android-studio-la-gi>
- [4]. Danh sách các api trong rạp chiếu phim. Truy cập ngày 20/09/2024 từ <https://rapchieuphim.com/api>

PHỤ LỤC

Bảng phân công công việc.

STT	Người thực hiện	Vai trò	Nhiệm vụ	Đánh giá
1	Bùi Minh Tùng	Nhóm trưởng	Phân công công việc Phân tích chức năng Thực hiện code	25%
2	Đặng Văn Tùng	Thành viên	Thực hiện code Thiết kế giao diện	25%
3	Nguyễn Duy Vũ	Thành viên	Tổng hợp báo cáo Hỗ trợ code	20%
4	Vũ Duy Tùng	Thành viên	Hỗ trợ tổng hợp báo cáo Hỗ trợ test	15%
5	Lê Quang Khánh Nguyên	Thành viên	Hỗ trợ tổng hợp báo cáo Hỗ trợ test	15%