

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

-----

**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH HỆ THỐNG THÔNG TIN**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG QUẢN LÝ VÀ NGHE NHẠC  
TRÊN NỀN TẢNG ANDROID SỬ DỤNG KOTLIN**

**CBHD : TS. Lê Thị Anh**  
**Sinh viên : Hà Thị Mai Linh**  
**Mã sinh viên : 2021600279**

**Hà Nội – 2024**

HÀ THỊ MAI LINH

HỆ THỐNG THÔNG TIN

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  
**NGÀNH HỆ THỐNG THÔNG TIN**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG QUẢN LÝ VÀ NGHE NHẠC**  
**TRÊN NỀN TẢNG ANDROID SỬ DỤNG KOTLIN**

**CBHD : TS. Lê Thị Anh**  
**Sinh viên : Hà Thị Mai Linh**  
**Mã sinh viên : 2021600279**

**Hà Nội – Năm 2024**

## LỜI CẢM ƠN

Trước tiên, em xin được bày tỏ lòng biết ơn chân thành và sâu sắc nhất đến tất cả các cá nhân và tổ chức đã hỗ trợ, đồng hành cùng em trong suốt quá trình nghiên cứu và thực hiện đồ án tốt nghiệp.

Trước hết, em xin gửi lời cảm ơn trân trọng đến tập thể quý thầy cô Khoa Công nghệ Thông tin, Trường Đại học Công nghiệp Hà Nội, những người đã tận tình giảng dạy và truyền đạt kiến thức quý báu trong suốt những năm tháng học tập tại trường. Chính sự dìu dắt, định hướng và động viên từ quý thầy cô đã tạo nền tảng vững chắc giúp em có thể hoàn thành đồ án tốt nghiệp này với đề tài: “Phát triển ứng dụng di động quản lý và nghe nhạc trên nền tảng Android sử dụng Kotlin.”.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến cô TS. Lê Thị Anh, người trực tiếp hướng dẫn, luôn tận tình hỗ trợ, góp ý và truyền cảm hứng cho em trong suốt quá trình nghiên cứu và hoàn thiện đồ án. Những ý kiến đóng góp quý giá của cô không chỉ giúp em hoàn thiện nội dung đồ án mà còn mang lại nhiều bài học kinh nghiệm thiết thực, quý báu cho quá trình phát triển bản thân và nghề nghiệp của em sau này.

Em nhận thức rõ rằng, mặc dù đã dành nhiều tâm huyết và nỗ lực, đồ án vẫn không thể tránh khỏi những thiếu sót do hạn chế về kinh nghiệm và thời gian thực hiện. Em kính mong nhận được sự chỉ dẫn và đóng góp ý kiến từ quý thầy cô để có thể hoàn thiện hơn trong tương lai.

Em xin chân thành cảm ơn!

## MỤC LỤC

LỜI CẢM ƠN .....	i
DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT ..	vi
DANH MỤC HÌNH VẼ .....	vii
DANH MỤC BẢNG BIỂU .....	x
MỞ ĐẦU .....	xi
CHƯƠNG 1: GIỚI THIỆU VỀ DỰ ÁN PHẦN MỀM .....	1
1.1. Giới thiệu chung .....	1
1.2. Công cụ xây dựng ứng dụng .....	2
1.2.1. Android Studio .....	2
1.2.2. Android Virtual Device .....	4
1.2.3. Kotlin .....	5
1.2.4. Media Player.....	6
1.2.5. Mô hình MVC .....	9
1.2.6. Google Firebase.....	12
1.3. Bản quyền và quyền tác giả .....	16
CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....	18
2.1. Khảo sát sơ bộ .....	18
2.1.1. Mục tiêu .....	18
2.1.2. Phương pháp.....	18
2.1.3. Kết quả sơ bộ.....	22
2.2. Khảo sát chi tiết .....	22
2.2.1. Các yêu cầu chức năng .....	22

2.2.2. Các yêu cầu phi chức năng .....	23
2.2.3. Giao diện.....	23
2.3. Xác định tác nhân .....	24
2.4. Biểu đồ use case.....	25
2.5. Mô tả chi tiết các Use case .....	27
2.5.1. Mô tả use case Đăng ký .....	27
2.5.2. Mô tả use case Đăng nhập .....	28
2.5.3. Mô tả use case Đăng xuất .....	29
2.5.4. Mô tả use case Quên mật khẩu .....	29
2.5.5. Mô tả use case Thay đổi mật khẩu .....	30
2.5.6. Mô tả use case Phát nhạc .....	31
2.5.7. Mô tả use case Phát ngẫu nhiên và lặp lại bài .....	32
2.5.8. Mô tả use case Xem và phát thể loại nhạc .....	33
2.5.9. Mô tả use case Tìm kiếm bài hát .....	33
2.5.10. Mô tả use case Quản lý danh sách yêu thích .....	34
2.5.11. Mô tả use case Viết Phản hồi.....	35
2.5.12. Mô tả use case Xem Contact.....	36
2.5.13. Mô tả use case Thêm bài hát.....	37
2.5.14. Mô tả use case Quản lý bài hát .....	38
2.5.15. Mô tả use case Xem Phản hồi.....	40
2.6. Sơ đồ hoạt động .....	41
2.6.1. Chức năng đăng ký tài khoản.....	41
2.6.2. Chức năng đăng nhập .....	41
2.6.3. Chức năng đăng xuất tài khoản.....	42

2.6.4. Chức năng quên mật khẩu .....	42
2.6.5. Chức năng thay đổi mật khẩu .....	43
2.6.6. Chức năng phát nhạc .....	43
2.6.7. Chức năng phát ngẫu nhiên và lặp lại bài .....	44
2.6.8. Chức năng xem và phát thể loại nhạc .....	44
2.6.9. Chức năng tìm kiếm bài hát.....	45
2.6.10. Chức năng viết phản hồi.....	45
2.6.11. Chức năng xem Contact.....	46
2.6.12. Chức năng thêm bài hát.....	46
2.6.13. Chức năng sửa bài hát.....	47
2.6.14. Chức năng xoá bài hát .....	47
2.7. Thiết kế cơ sở dữ liệu Realtime database .....	48
2.7.1. Cấu trúc cơ sở dữ liệu.....	48
2.7.2. Kiểu dữ liệu.....	48
2.8. Thiết kế giao diện .....	50
<b>CHƯƠNG 3: KẾT QUẢ THỰC HIỆN VÀ KIỂM THỬ HỆ THỐNG.....</b>	<b>51</b>
3.1. Cài đặt môi trường.....	51
3.1.1. Cài đặt môi trường Java .....	51
3.1.2. Cài đặt Android Studio .....	51
3.2. Cài đặt chương trình .....	52
3.3. Kết quả thu được.....	54
3.3.1. Chức năng đăng nhập .....	56
3.3.2. Chức năng đăng xuất .....	56
3.3.3. Chức năng đăng ký.....	57

3.3.4. Chức năng thay đổi mật khẩu .....	57
3.3.5. Chức năng quên mật khẩu .....	58
3.3.6. Chức năng phát nhạc .....	58
3.3.7. Chức năng phát nhạc theo thể loại .....	59
3.3.8. Chức năng quản lý danh sách yêu thích .....	59
3.3.9. Chức năng tìm kiếm bài hát .....	60
3.3.10. Chức năng viết Phản hồi .....	60
3.3.11. Chức năng xem contact .....	61
3.3.12. Chức năng thêm bài hát .....	61
3.3.13. Chức năng quản lý bài hát .....	62
3.3.14. Chức năng xem phản hồi .....	62
3.4. Kiểm thử hệ thống .....	63
3.5. Kịch bản kiểm thử .....	64
3.5.1. Kiểm thử chức năng đăng nhập .....	64
3.5.2. Kiểm thử chức năng đăng ký .....	64
3.5.3. Kiểm thử chức năng phát nhạc theo thể loại .....	64
3.5.4. Kiểm thử chức năng tìm kiếm bài hát .....	65
3.5.5. Kiểm thử chức năng quản lý danh sách yêu thích .....	66
3.5.6. Kiểm thử chức năng thêm bài hát .....	66
3.5.7. Kiểm thử chức năng sửa bài hát .....	67
3.5.8. Kiểm thử chức năng xóa bài hát .....	67
3.6. Kết luận kiểm thử .....	68
KẾT LUẬN VÀ KIẾN NGHỊ .....	69
TÀI LIỆU THAM KHẢO .....	71

**DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU  
VÀ CÁC CHỮ VIẾT TẮT**

<b>STT</b>	<b>Viết tắt</b>	<b>Dịch nghĩa</b>
1	MVC	Model – View – Controller
2	AVD	Android Virtual Devices
3	CSDL	Cơ sở dữ liệu
4	JDK	Java Development Kit
5	SDK	Software Development Kit
6	XML	Extensible Markup Language
7	IDE	Integrated Development Environment



## DANH MỤC HÌNH VẼ

Hình 1.1: Android Studio .....	2
Hình 1.2: Máy ảo AVD .....	4
Hình 1.3: Sơ đồ hoạt động của MVC.....	10
Hình 1.4: Luồng xử lý trong MVC .....	11
Hình 1.5: Các chức năng nổi bật của Firebase .....	13
Hình 2.1: Biểu đồ use case tổng quát.....	25
Hình 2.2: Sơ đồ hoạt động chức năng đăng ký tài khoản .....	41
Hình 2.3: Sơ đồ hoạt động chức năng đăng nhập.....	41
Hình 2.4: Sơ đồ hoạt động chức năng đăng xuất.....	42
Hình 2.5: Sơ đồ hoạt động chức năng quên mật khẩu .....	42
Hình 2.6: Sơ đồ hoạt động chức năng thay đổi mật khẩu .....	43
Hình 2.7: Sơ đồ hoạt động chức năng phát nhạc .....	43
Hình 2.8: Sơ đồ hoạt động chức năng phát ngẫu nhiên và lặp lại bài .....	44
Hình 2.9: Sơ đồ hoạt động chức năng xem và phát thể loại nhạc .....	44
Hình 2.10: Sơ đồ hoạt động chức năng tìm kiếm bài hát .....	45
Hình 2.11: Sơ đồ hoạt động chức năng viết phản hồi.....	45
Hình 2.12: Sơ đồ hoạt động chức năng xem Contact .....	46
Hình 2.13: Sơ đồ hoạt động chức năng thêm bài hát.....	46
Hình 2.14: Sơ đồ hoạt động chức năng sửa bài hát .....	47
Hình 2.15: Sơ đồ hoạt động chức năng xóa bài hát.....	47
Hình 2.16: Cấu trúc dạng Json.....	48
Hình 2.17: Cấu trúc dạng Tree .....	48
Hình 2.18: Giao diện một số màn hình được thiết kế bằng Figma .....	50

Hình 3.1. Một số phiên bản JDK trên trang Oracle .....	51
Hình 3.2. Giao diện của Android Studio sau khi cài đặt thành công .....	52
Hình 3.3. Cài đặt chương trình.....	52
Hình 3.4. Giao diện Android Studio sau khi đã nhập mã nguồn .....	53
Hình 3.5. Giao diện ban đầu của chương trình sau khi build .....	54
Hình 3.6. Giao diện chức năng đăng nhập.....	56
Hình 3.7. Giao diện màn hình chức năng đăng xuất .....	56
Hình 3.8. Giao diện màn hình chức năng đăng ký.....	57
Hình 3.9. Giao diện màn hình chức năng thay đổi mật khẩu .....	57
Hình 3.10. Giao diện màn hình chức năng quên mật khẩu .....	58
Hình 3.11. Giao diện màn hình chức năng phát nhạc .....	58
Hình 3.12. Giao diện màn hình chức năng phát nhạc theo thể loại.....	59
Hình 3.13. Giao diện màn hình chức năng quản lý danh sách yêu thích.....	59
Hình 3.14. Giao diện màn hình chức năng tìm kiếm bài hát.....	60
Hình 3.15. Giao diện màn hình chức năng viết phản hồi.....	60
Hình 3.16. Giao diện màn hình chức năng xem contact .....	61
Hình 3.17. Giao diện màn hình chức năng thêm bài hát.....	61
Hình 3.18. Giao diện màn hình chức năng quản lý bài hát .....	62
Hình 3.19. Giao diện màn hình chức năng xem phản hồi.....	62
Hình 3.20: Kiểm thử chức năng đăng nhập.....	64
Hình 3.21: Kiểm thử chức năng đăng ký .....	64
Hình 3.22: Kiểm thử chức năng phát nhạc theo thể loại .....	65
Hình 3.23: Kiểm thử chức năng tìm kiếm bài hát .....	65
Hình 3.24: Kiểm thử chức năng quản lý danh sách yêu thích .....	66

Hình 3.25: Kiểm thử chức năng thêm bài hát .....	66
Hình 3.26: Kiểm thử chức năng sửa bài hát.....	67
Hình 3.27: Kiểm thử chức năng xoá bài hát.....	67
Hình 3.28: Kết quả kiểm thử .....	68

**DANH MỤC BẢNG BIỂU**

Bảng 2.1: Kế hoạch phỏng vấn.....	18
Bảng 2.2: Các tác nhân.....	24
Bảng 2.3: Kiểu dữ liệu của Phản hồi .....	49
Bảng 2.4: Kiểu dữ liệu của Bài hát .....	49

xi  
**MỞ ĐẦU**

Trong thời đại công nghệ số, sự phát triển mạnh mẽ của công nghệ thông tin đã thay đổi đáng kể cách con người sống và làm việc. Các ứng dụng di động đã trở thành công cụ hỗ trợ không thể thiếu, mang lại giải pháp tiện ích cho nhiều lĩnh vực như công việc, học tập, kết nối xã hội và đặc biệt là giải trí. Trong số đó, các ứng dụng nghe nhạc trực tuyến đã và đang chiếm giữ một vị trí quan trọng trong thói quen hằng ngày của nhiều người. Những ứng dụng này không chỉ cung cấp trải nghiệm âm nhạc phong phú, đa dạng mà còn giúp người dùng dễ dàng tiếp cận hàng triệu bài hát từ khắp nơi trên thế giới chỉ với vài thao tác đơn giản.

Sự phát triển của ngôn ngữ lập trình Kotlin cùng nền tảng Firebase đã mở ra nhiều cơ hội mới trong việc xây dựng các ứng dụng di động hiện đại. Kotlin nổi bật với khả năng linh hoạt, hiệu suất cao, đồng thời giảm bớt những phức tạp trong việc phát triển ứng dụng. Trong khi đó, Firebase cung cấp các giải pháp toàn diện từ cơ sở dữ liệu, phân tích, quản lý người dùng, cho đến lưu trữ tệp và xử lý dữ liệu thời gian thực. Các công nghệ này cho phép phát triển những ứng dụng nghe nhạc thông minh, đáp ứng tốt nhu cầu ngày càng cao của người dùng.

## 1. Lý do chọn đề tài

Công nghệ thông tin ngày nay không chỉ đóng vai trò quan trọng trong việc phát triển kinh tế - xã hội mà còn góp phần cải thiện chất lượng cuộc sống, nâng cao hiệu suất lao động và thay đổi tư duy làm việc. Trong xu thế phát triển mạnh mẽ của cuộc cách mạng công nghiệp 4.0, công nghệ thông tin đã trở thành một trong những trụ cột kinh tế mũi nhọn của đất nước, góp phần thúc đẩy sự phát triển toàn diện trên nhiều lĩnh vực.

Xuất phát từ niềm đam mê với cả công nghệ và âm nhạc, em quyết định chọn đề tài: “Phát triển ứng dụng di động quản lý và nghe nhạc trên nền tảng Android sử dụng Kotlin”. Việc thực hiện đề tài này không chỉ là cơ hội để em phát triển kỹ năng chuyên môn về lập trình mà còn là dịp để khám phá và áp dụng các công nghệ hiện đại vào thực tiễn. Đặc biệt, sử dụng Kotlin - ngôn ngữ lập trình được đánh giá cao bởi tính hiện đại và thân thiện, cùng với nền tảng Firebase - một công cụ mạnh mẽ hỗ trợ xây dựng và quản lý ứng dụng, em mong muốn tạo ra một sản phẩm vừa có tính ứng dụng cao, vừa mang lại giá trị thiết thực và trải nghiệm thú vị cho người dùng.

Đề tài này cũng thể hiện khát vọng của em trong việc không ngừng học hỏi, sáng tạo và ứng dụng công nghệ để mang đến những sản phẩm hữu ích, đồng thời đáp ứng xu hướng phát triển không ngừng của ngành công nghệ thông tin và nhu cầu giải trí ngày càng đa dạng của người dùng.

## 2. Mục tiêu đề tài

Mục tiêu đề ra của đề tài:

- Nghiên cứu về: Kotlin, Firebase, MVC,...;
- Phân tích thiết kế và viết tài liệu đặc tả yêu cầu phần mềm trên việc khảo sát một số ứng dụng: Spotify, ZingMP3,...
- Thiết kế và cài đặt phần mềm dựa trên tài liệu đặc tả
- Xây dựng được ứng dụng di động có các chức năng: đăng ký, đăng nhập, nghe nhạc, quản lý bài hát, ...
- Xây dựng được ứng dụng đạt yêu cầu về nghiệp vụ.

## 3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là các ứng dụng di động nghe nhạc trên nền tảng Android, đặc biệt là những ứng dụng sử dụng ngôn ngữ lập trình Kotlin và nền tảng Firebase. Trong đó, các tính năng cơ bản như đăng ký, đăng nhập, nghe nhạc, quản lý bài hát, và các chức năng liên quan đến việc xử lý dữ liệu người dùng và tương tác với hệ thống cơ sở dữ liệu sẽ được tập trung nghiên cứu và phát triển.

Phạm vi nghiên cứu của đề tài bao gồm các bước từ khảo sát, phân tích, thiết kế đến triển khai ứng dụng di động. Đề tài chủ yếu nghiên cứu các công nghệ, công cụ và phương pháp phát triển ứng dụng di động như Kotlin, Firebase, và mô hình phát triển phần mềm MVC. Cụ thể, phạm vi nghiên cứu của đề tài bao gồm:

- Nghiên cứu và áp dụng các công nghệ phát triển ứng dụng di động
- Khảo sát các ứng dụng nghe nhạc phổ biến như Spotify, ZingMP3, để phân tích và học hỏi từ các tính năng và thiết kế của các ứng dụng hiện tại.
- Xây dựng và triển khai một ứng dụng di động hoàn chỉnh với các tính năng cơ bản, đảm bảo hoạt động hiệu quả trên nền tảng Android.

#### 4. Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học: Đề tài góp phần vào nghiên cứu ứng dụng các công nghệ mới như Kotlin và Firebase trong phát triển phần mềm di động, giúp tối ưu hiệu suất và bảo mật. Mô hình MVC giúp tăng khả năng mở rộng và bảo trì ứng dụng trong tương lai.

Ý nghĩa thực tiễn: Đề tài mang lại ứng dụng nghe nhạc di động chất lượng cao, đáp ứng nhu cầu giải trí trực tuyến. Việc sử dụng Kotlin và Firebase không chỉ cải thiện hiệu suất mà còn tạo ra một sản phẩm dễ sử dụng, phù hợp với xu hướng công nghệ hiện đại. Đề tài giúp nâng cao kỹ năng lập trình và khả năng phát triển phần mềm trong ngành công nghệ thông tin.

#### 5. Cấu trúc báo cáo

Cấu trúc chính của báo cáo bao gồm 4 chương:

- Chương 1: Giới thiệu về dự án phần mềm
  - Giới thiệu chung
  - Mô tả bài toán
  - Giới thiệu về công nghệ áp dụng trong phát triển ứng dụng nghe nhạc
- Chương 2: Khảo sát và phân tích thiết kế hệ thống
  - Khảo sát sơ bộ
  - Khảo sát chi tiết
  - Xác định tác nhân
  - Biểu đồ use case
  - Mô tả chi tiết các use case
  - Sơ đồ hoạt động
  - Thiết kế cơ sở dữ liệu Realtime database



- Thiết kế màn hình
- Chương 3: Kết quả thực hiện và kiểm thử hệ thống
- Cài đặt môi trường
  - Kết quả thu được
  - Kiểm thử hệ thống
  - Kịch bản kiểm thử

## CHƯƠNG 1: GIỚI THIỆU VỀ DỰ ÁN PHẦN MỀM

### 1.1. Giới thiệu chung

Trong thời đại hiện nay, việc thưởng thức âm nhạc trở thành một phần không thể thiếu trong cuộc sống hàng ngày của mọi người. Sự phát triển của công nghệ đã tạo ra nhiều cơ hội mới cho việc trải nghiệm âm nhạc, đặc biệt là qua các ứng dụng nghe nhạc trực tuyến. Điều này đã thúc đẩy sự gia tăng về sự phát triển của các ứng dụng nghe nhạc, và việc sử dụng ngôn ngữ lập trình Kotlin cùng nền tảng Firebase của Google là một bước tiến quan trọng trong quá trình phát triển ứng dụng này.

Ứng dụng nghe nhạc không chỉ là một công cụ giải trí, mà còn là một cách để kết nối con người với âm nhạc một cách thuận tiện và hiệu quả. Bằng cách sử dụng Kotlin, một ngôn ngữ lập trình hiện đại và linh hoạt, kết hợp với sức mạnh của Firebase - một nền tảng phát triển ứng dụng đám mây mạnh mẽ của Google, ứng dụng này hứa hẹn mang đến trải nghiệm nghe nhạc tuyệt vời cho người dùng.

Với đề tài này, em sẽ đi sâu vào việc phát triển một ứng dụng nghe nhạc đa năng, cung cấp cho người dùng trải nghiệm âm nhạc tốt nhất cùng với các tính năng như tìm kiếm, tạo danh sách yêu thích, ... Sử dụng Kotlin và Firebase sẽ giúp chúng ta xây dựng ứng dụng một cách linh hoạt, đồng thời đảm bảo tính năng và hiệu suất của ứng dụng.

Với sự kết hợp giữa công nghệ và đam mê âm nhạc, em hy vọng rằng đề tài này sẽ mang lại những giá trị và trải nghiệm đáng nhớ cho cả người phát triển và người dùng.

## 1.2. Công cụ xây dựng ứng dụng

### 1.2.1. Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android, dựa trên IntelliJ IDEA.



*Hình 1.1: Android Studio*

Ngoài các công cụ dành cho nhà phát triển và biên tập mã mạnh mẽ của IntelliJ, Android Studio cung cấp nhiều tính năng hơn nữa để nâng cao năng suất của bạn khi xây dựng các ứng dụng Android, chẳng hạn như:

- ✓ Hệ thống xây dựng dựa trên Gradle linh hoạt.
- ✓ Trình giả lập nhanh và nhiều tính năng.
- ✓ Một môi trường hợp nhất, nơi bạn có thể phát triển cho tất cả các thiết bị Android.
- ✓ Áp dụng các thay đổi để đẩy mã và thay đổi tài nguyên cho ứng dụng đang chạy của bạn mà không cần khởi động lại ứng dụng của bạn.
- ✓ Các mode được tích hợp trên GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến và nhập mã mẫu.
- ✓ Các công cụ và khuôn khổ thử nghiệm mở rộng.
- ✓ Các công cụ của Lint để nắm bắt hiệu suất, khả năng sử dụng, khả năng tương thích của phiên bản và các vấn đề khác.
- ✓ Hỗ trợ C++ và NDK.

- ✓ Hỗ trợ tích hợp cho Google Cloud Platform, giúp dễ dàng tích hợp Google Cloud Messaging và App Engine.

### 1.2.2. Android Virtual Device

Thiết bị Android ảo (Android Virtual Device – AVD) là trình giả lập Android bao gồm một bộ cảm biến và tính năng hoàn chỉnh để tương tác với môi trường Android ảo. AVD thường được tích hợp khi bạn cài đặt bộ môi trường phát triển tích hợp Android Studio. Với AVD, bạn có thể kiểm tra các ứng dụng Android của mình trên nhiều loại thiết bị ảo cho mục đích phát triển, thử nghiệm và trình diễn. AVD nhanh, cài đặt đơn giản và mạnh mẽ nhờ các tiện ích cảm biến và tính năng tương tác thân thiện với người dùng. Nó có sẵn cho các hệ điều hành Windows, macOS và Linux.



*Hình 2.2: Máy ảo AVD*

### 1.2.3. Kotlin

Kotlin là một ngôn ngữ lập trình kiểu tĩnh chạy trên máy ảo Java (JVM) và có thể được biên dịch sang mã nguồn Java hay sử dụng cơ sở hạ tầng trình biên dịch LLVM. Nó được tài trợ và phát triển bởi JetBrains. Mặc dù cú pháp không tương thích với Java, những bản thực hiện JVM của thư viện chuẩn Kotlin được thiết kế để tương tác với mã Java và dựa vào mã Java từ Java Class Library có sẵn, ví dụ như collections framework. Kotlin sử dụng suy luận kiểu một cách tích cực để xác định kiểu của giá trị và biểu thức vốn không được nêu rõ. Điều này giúp giảm tính dài dòng của ngôn ngữ so với Java, vốn thường đòi hỏi toàn bộ đặc kiểu một cách dư thừa mãi đến phiên bản 10. Mã Kotlin có thể chạy trên JVM đến phiên bản Java 11 mới nhất.

Kể từ Android Studio 3.0 (phát hành vào tháng 10 năm 2017), Kotlin được Google hỗ trợ đầy đủ để sử dụng cho việc lập trình ứng dụng cho hệ điều hành Android của họ và được nhúng trực tiếp vào trong gói cài đặt của IDE đó để thay thế cho trình biên dịch Java tiêu chuẩn. Trình biên dịch Android Kotlin cho phép người dùng chọn lựa giữa hướng đến mã bytecode tương thích với Java 6, hay Java 8.

Vào tháng 7 năm 2011, JetBrains đã giới thiệu Project Kotlin là một ngôn ngữ mới cho JVM đã được phát triển trong một năm. Vào tháng 2 năm 2012, JetBrains mở nguồn dự án theo giấy phép Apache2. Kotlin v1.0 được phát hành vào ngày 15 tháng 2 năm 2016. Đây được coi là phiên bản chính thức ổn định đầu tiên và JetBrains đã cam kết tương thích ngược dài hạn kể từ phiên bản này.

Tại hội nghị Google I/O 2017, Google công bố hỗ trợ hạng nhất (first class support) cho Kotlin trên Android. Kotlin v1.2 được phát hành vào ngày 28 tháng 11 năm 2017. Tính năng chia sẻ mã nguồn giữa nền tảng JVM và Javascript mới được thêm vào bản phát hành này. Kotlin v1.3 được phát hành vào ngày 29 tháng 10 năm 2018, với các coroutine API cho lập trình

bất đồng bộ.

#### **Ưu điểm:**

- ✓ **Có thể thay thế cho Java:** Kotlin có thể được sử dụng thay thế cho Java, đặc biệt trong phát triển ứng dụng Android, nhờ khả năng tương thích hoàn toàn với mã Java.
- ✓ **Dễ học:** Cú pháp của Kotlin gọn gàng và dễ hiểu, làm cho ngôn ngữ này dễ học, đặc biệt đối với những người đã có kinh nghiệm với Java.
- ✓ **Kết hợp những gì tốt nhất của lập trình hàm và thủ tục:** Kotlin hỗ trợ cả lập trình hướng đối tượng và lập trình hàm, mang lại sự linh hoạt cho lập trình viên.
- ✓ **Code ngắn gọn hơn:** Kotlin cho phép viết code ngắn gọn và dễ đọc hơn so với Java, giúp tăng năng suất và giảm thiểu lỗi.

#### **Nhược điểm:**

- ✓ **Hiệu suất runtime:** Kotlin có thể yêu cầu thời gian chạy (runtime) lâu hơn so với Java, do một số tính năng nâng cao của ngôn ngữ.
- ✓ **Không dễ tiếp cận với người mới bắt đầu:** Mặc dù dễ học với người đã biết Java, Kotlin vẫn có thể khó khăn cho người mới bắt đầu hoàn toàn do cú pháp và khái niệm mới.
- ✓ **Thiếu tài liệu và hỗ trợ chính thức:** So với Java, Kotlin có ít tài liệu và nguồn hỗ trợ hơn, dù điều này đang cải thiện theo thời gian.
- ✓ **Công cụ hỗ trợ và tích hợp:** Mặc dù hầu hết các IDE lớn như IntelliJ IDEA và Android Studio hỗ trợ Kotlin tốt, nhưng có thể gặp khó khăn với một số công cụ và plugin khác chưa hỗ trợ đầy đủ.

#### **1.2.4. Media Player**

Trong phát triển ứng dụng nghe nhạc trực tuyến, việc phát nhạc là chức năng cốt lõi và quan trọng nhất. Android cung cấp một API mạnh mẽ, đó là MediaPlayer, giúp dễ dàng phát các tệp âm thanh và video từ nhiều nguồn khác nhau. MediaPlayer hỗ trợ nhiều định dạng tệp và có thể được sử dụng để phát nhạc trực tuyến.

- Các chức năng của MediaPlayer: MediaPlayer là một lớp trong Android API, cung cấp các chức năng cần thiết để điều khiển quá trình phát lại âm thanh và video.
- Một số chức năng chính của MediaPlayer bao gồm:
  - Phát nhạc từ nhiều nguồn khác nhau
  - Nguồn trực tuyến: MediaPlayer có thể phát nhạc từ các URL trực tuyến, giúp dễ dàng stream nhạc từ internet.
  - Nguồn cục bộ: MediaPlayer có thể phát nhạc từ các tệp lưu trữ trên thiết bị hoặc thẻ nhớ.
- Điều khiển quá trình phát lại
  - Phát nhạc: Bắt đầu phát nhạc từ vị trí hiện tại hoặc từ đầu.
  - Tạm dừng nhạc: Tạm dừng phát nhạc và có thể tiếp tục từ vị trí đã dừng.
  - Dừng nhạc: Dừng phát nhạc và chuẩn bị lại MediaPlayer để phát nhạc khác hoặc phát lại từ đầu.
  - Tua nhạc: Chuyển đến một vị trí cụ thể trong bài hát.
- Quản lý các trạng thái phát nhạc
  - Đang phát: Trạng thái khi MediaPlayer đang phát nhạc.
  - Đã dừng: Trạng thái khi MediaPlayer đã dừng phát nhạc.
  - Hoàn tất: Trạng thái khi MediaPlayer đã phát xong bài hát.
- Hỗ trợ Streaming từ các nguồn trực tuyến: MediaPlayer hỗ trợ phát nhạc trực tuyến (Streaming) từ các URL, giúp ứng dụng có thể phát nhạc mà không cần tải về toàn bộ tệp nhạc.

**Ưu điểm:**



- ✓ **Dễ sử dụng:** MediaPlayer cung cấp một API đơn giản và dễ hiểu, giúp nhà phát triển nhanh chóng tích hợp chức năng phát nhạc vào ứng dụng.
- ✓ **Hỗ trợ nhiều định dạng:** MediaPlayer hỗ trợ nhiều định dạng âm thanh và video phổ biến, giúp ứng dụng có thể phát nhiều loại tệp khác nhau.
- ✓ **Hỗ trợ streaming:** MediaPlayer hỗ trợ phát nhạc trực tuyến, giúp người dùng có thể nghe nhạc từ internet mà không cần tải về.
- ✓ **Quản lý tài nguyên tự động:** MediaPlayer tự động quản lý tài nguyên hệ thống, giúp tiết kiệm bộ nhớ và tối ưu hóa hiệu suất.

#### **Nhược điểm:**

- ✓ **Giới hạn trong việc xử lý âm thanh phức tạp:** MediaPlayer không hỗ trợ xử lý âm thanh phức tạp, như hiệu ứng âm thanh hoặc xử lý đa kênh.
- ✓ **Phụ thuộc vào trạng thái mạng:** Khi phát nhạc trực tuyến, MediaPlayer phụ thuộc vào chất lượng kết nối mạng, có thể gây ra gián đoạn nếu mạng không ổn định.
- ✓ **Thời gian chuẩn bị:** MediaPlayer cần thời gian để chuẩn bị trước khi phát nhạc, đặc biệt khi phát nhạc từ nguồn trực tuyến.

**Kết luận:** MediaPlayer của Android là một công cụ mạnh mẽ và linh hoạt để phát nhạc trong ứng dụng di động. Với khả năng phát nhạc từ nhiều nguồn khác nhau, điều khiển quá trình phát lại, và hỗ trợ streaming, MediaPlayer là lựa chọn lý tưởng cho các ứng dụng nghe nhạc trực tuyến. Tuy nhiên, MediaPlayer cũng có một số hạn chế nhất định, như khả năng xử lý âm thanh phức tạp và phụ thuộc vào chất lượng kết nối mạng.

Bằng cách kết hợp MediaPlayer với các dịch vụ của Firebase, chúng ta có thể xây dựng một ứng dụng nghe nhạc trực tuyến hoàn chỉnh, cung cấp trải nghiệm mượt mà cho người dùng. Việc hiểu và sử dụng hiệu quả

MediaPlayer sẽ giúp chúng ta phát triển các ứng dụng âm nhạc với các tính năng phong phú và hiệu suất cao.

### 1.2.5. Mô hình MVC

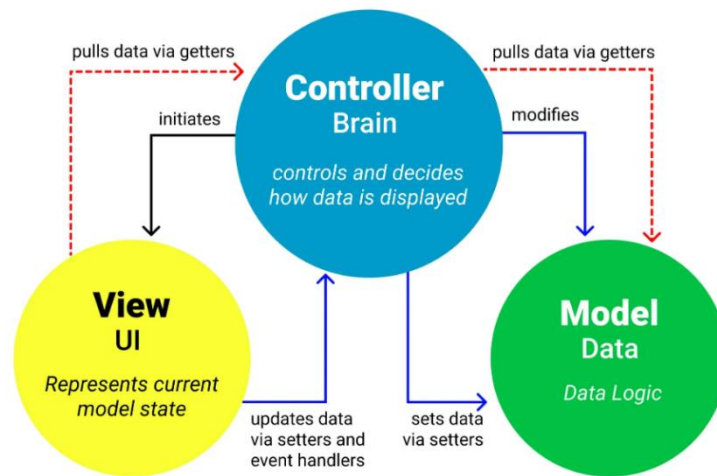
**MVC** là viết tắt của cụm từ “**Model – View – Controller**”. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng.

Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác:

- **Model:** Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản.
- **View:** Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
- **Controller:** Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.

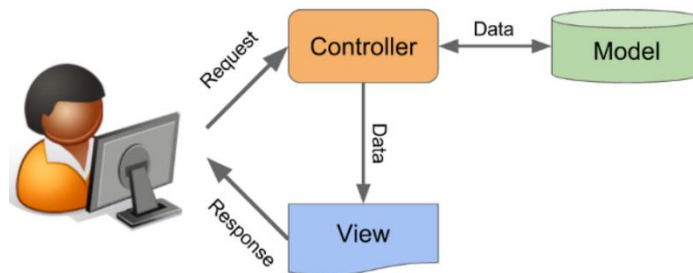
## MVC Architecture Pattern



*Hình 1.3: Sơ đồ hoạt động của MVC*

Luồng xử lý trong MVC:

- Khi một yêu cầu được gửi từ máy khách (Client) đến Server, Controller trong MVC sẽ tiếp nhận yêu cầu này để xác định xem đó là URL request hay sự kiện từ người dùng.
- Controller xử lý và kiểm soát dữ liệu đầu vào từ người dùng, sau đó gửi yêu cầu tương ứng đến Model.
- Model sẽ thực hiện các hoạt động xử lý dữ liệu, bao gồm truy vấn cơ sở dữ liệu, tính toán hoặc tương tác với các nguồn dữ liệu khác, và chuẩn bị dữ liệu để trả về cho Controller.
- Controller nhận kết quả từ Model và quyết định cách hiển thị dữ liệu đó bằng cách chọn View phù hợp.
- Controller gửi dữ liệu được chuẩn bị từ Model cho View để hiển thị lên giao diện người dùng, thông qua các phương thức và tương tác được định nghĩa sẵn.



*Hình 1.4: Luồng xử lý trong MVC*

View không giao tiếp trực tiếp với Model trong luồng xử lý này. Mọi tương tác giữa View và Model đều thông qua Controller, giữ cho mô hình luôn duy trì tính tách biệt và linh hoạt.

#### **Ưu điểm:**

- ✓ **Kiểm tra đơn giản và dễ dàng:** MVC trong Android cho phép kiểm tra phần mềm một cách dễ dàng và hiệu quả trước khi bàn giao cho người dùng. Việc tách biệt các thành phần Model, Controller và View giúp dễ dàng xác định và sửa lỗi trong mã nguồn.
- ✓ **Tách biệt các thành phần:** Một trong những lợi ích chính của MVC trong Android là khả năng tách biệt các thành phần Model, Controller và View. Điều này giúp cho việc phát triển ứng dụng trở nên linh hoạt và dễ dàng bảo trì.
- ✓ **Controller tối ưu:** Mô hình MVC trong Android tạo điều kiện cho Controller hoạt động một cách hiệu quả và tối ưu trên nhiều nền tảng ngôn ngữ khác nhau, giúp tạo ra các ứng dụng chất lượng cao.
- ✓ **Dễ dàng duy trì ứng dụng:** MVC giúp cho việc duy trì ứng dụng trở nên dễ dàng hơn bởi các thành phần được tách biệt với nhau. Điều này giúp cho việc cập nhật và mở rộng ứng dụng trở nên thuận tiện và hiệu quả.
- ✓ **Tính linh hoạt trong phát triển:** MVC trong Android cho phép chia nhỏ công việc và phân chia cho nhiều nhà phát triển làm việc đồng thời mà không ảnh hưởng đến nhau, giúp tăng hiệu suất và tiết kiệm thời gian.
- ✓ **Hỗ trợ Test-Driven Development (TTD):** MVC hỗ trợ việc thực

hiện Test-Driven Development (TTD) thông qua việc tạo và thực thi các unit test và test case một cách dễ dàng và linh hoạt.

- ✓ **Responsive và linh hoạt:** Phiên bản mới nhất của MVC hỗ trợ thiết kế responsive cho website và cung cấp các mẫu thiết kế cho thiết bị di động. Điều này giúp phát triển ứng dụng trên nhiều nền tảng một cách linh hoạt và tiện lợi.

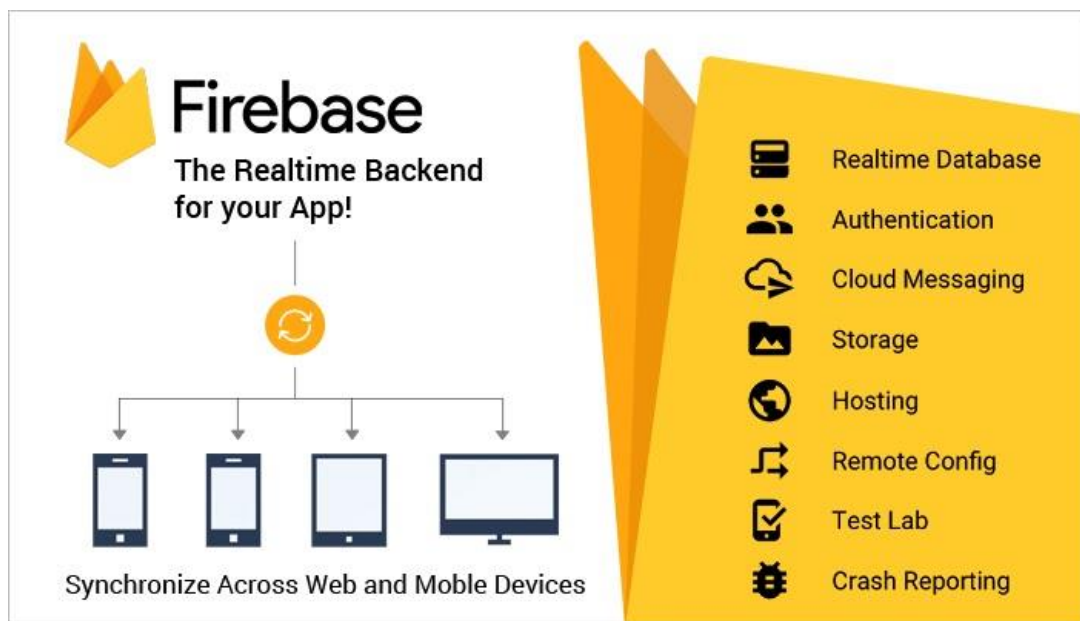
#### **Nhược điểm:**

- ✓ **Phức tạp khi ứng dụng lớn:** Khi ứng dụng trở nên lớn và phức tạp, việc quản lý các thành phần trong MVC có thể trở nên khó khăn. Điều này có thể dẫn đến việc tăng độ phức tạp của mã nguồn và làm giảm khả năng tái sử dụng mã.
- ✓ **Sự phụ thuộc cao giữa các thành phần:** Trong mô hình MVC, Controller thường phải tương tác với cả Model và View. Sự phụ thuộc cao giữa các thành phần này có thể làm cho mã nguồn trở nên khó hiểu và khó bảo trì.
- ✓ **Khó thử nghiệm:** Việc thử nghiệm (testing) trong mô hình MVC có thể trở nên khó khăn do sự phụ thuộc chặt chẽ giữa các thành phần. Điều này làm cho việc viết các bài kiểm tra (unit tests) hoặc kiểm tra tự động (automated tests) trở nên phức tạp.
- ✓ **View Logic phức tạp:** Trong MVC, thường có sự hiện diện của logic trong lớp View, đặc biệt là khi View trở nên phức tạp. Điều này có thể dẫn đến việc trộn lẫn logic hiển thị với logic xử lý dữ liệu, làm cho mã nguồn trở nên khó bảo trì và tái sử dụng.
- ✓ **Khó tái sử dụng giao diện người dùng:** Một trong những mục tiêu của mô hình MVC là tách biệt logic xử lý dữ liệu và logic hiển thị. Việc tái sử dụng giao diện người dùng (UI) có thể gặp khó khăn khi có sự ràng buộc mạnh mẽ giữa Model và View.

#### **1.2.6. Google Firebase**

**Firebase** là một nền tảng dịch vụ đám mây của Google cung cấp cho

các nhà phát triển các công cụ để xây dựng và triển khai ứng dụng di động và web. Nền tảng này cung cấp nhiều tính năng như cơ sở dữ liệu thời gian thực, phân tích, lưu trữ, xác thực người dùng, push notification và nhiều hơn nữa. Với Firebase, các nhà phát triển không cần phải tự xây dựng các phần mềm phức tạp để quản lý cơ sở dữ liệu và hạ tầng, mà có thể tập trung vào việc phát triển ứng dụng của mình. Firebase được sử dụng rộng rãi trong cộng đồng phát triển ứng dụng và web hiện nay.



*Hình 1.5: Các chức năng nổi bật của Firebase*

Chức năng hoạt động của **Firebase** là gì? Kể từ sau khi Google chính thức mua lại và tiếp tục phát triển, cho đến nay Firebase gồm có các hoạt động điển hình như sau:

- **Firebase Authentication:**

Hoạt động nổi trội nhất của Firebase chính là xây dựng những bước xác định người dùng thông qua Email, Facebook, Twitter, GitHub hay Google. Ngoài ra, hoạt động Firebase Authentication cũng hỗ trợ xác thực nặc danh cho những ứng dụng. Hoạt động xác thực của Firebase có thể giúp cho thông tin cá nhân của những người sử dụng được an toàn hơn. Điều này cũng đảm bảo tài khoản và các thông tin cá nhân của người dùng không bị đánh cắp.

- **Firebase Cloud Storage:**

Lưu trữ đám mây của Firebase cung cấp bộ nhớ đám mây an toàn, có thể mở rộng, dùng để lưu trữ và phân phát nội dung do người dùng tạo, chẳng hạn như hình ảnh, video và các tệp khác. Nó cung cấp các API đơn giản để tải lên, tải xuống và quản lý tệp, cùng với quyền kiểm soát truy cập cụ thể.

- **Firestore Database:**

Firestore Database có dạng một JSON đã được đồng bộ thời gian đến với tất cả các kết nối client. Để có được hoạt động này thì các lập trình viên cần phải đăng ký tài khoản ở trên Firebase. Dữ liệu ở trong database sẽ tự động cập nhật một cách liên tục khi phát triển ứng dụng. Sau khi đã được cập nhật thì những dữ liệu này sẽ được truyền tải thông qua các kết nối SSL có 2048 bit.

**Ưu điểm:**

- ✓ **Sử dụng dễ dàng:** Những người dùng có thể đăng ký một tài khoản Firebase thông qua tài khoản Google. Đồng thời, người dùng cũng có thể sử dụng nền tảng này trong quá trình phát triển ứng dụng một cách đơn giản nhất.
- ✓ **Tốc độ phát triển nhanh:** Ưu điểm tiếp theo chính là Firebase hỗ trợ cho việc phát triển ứng dụng rất nhanh chóng. Điều này sẽ giúp lập trình viên giảm bớt được thời gian để phát triển cũng như tiếp thị ứng dụng.
- ✓ **Cung cấp nhiều dịch vụ:** Firebase còn cung cấp đa dịch vụ cho mục đích phát triển trang web. Người dùng có thể lựa chọn database Firestore hoặc Realtime theo đúng ý muốn của mình.
- ✓ **Nền tảng cho Google phát triển:** google firebase là gì? Firebase được Google mua lại và trở thành một phần của Google. Ứng dụng này sẽ khai thác được triệt để sức mạnh cũng như các dịch vụ hiện đang sẵn có của Google.
- ✓ **Giao diện người dùng được chú trọng:** Firebase sẽ cho phép các lập trình viên tập trung hơn vào việc phát triển giao diện của người dùng

thông qua kho Backend mẫu vô cùng đa dạng.

- ✓ **Firestore app không có máy chủ:** Chính điều này sẽ giúp cho Firestore có được khả năng tối ưu hóa nhất về hiệu suất làm việc nhờ vào việc mở rộng cụm database.
- ✓ **Học máy:** Ứng dụng Firestore sẽ cung cấp học máy cho các lập trình viên để hỗ trợ tốt nhất cho việc phát triển ứng dụng.
- ✓ **Tạo lưu lượng truy cập:** Firestore App sẽ hỗ trợ việc tạo lập các chỉ mục. Đồng thời, Firestore cũng sẽ giúp nâng cao thứ hạng của ứng dụng ở trên bảng xếp hạng của Google. Nhờ vậy mà lượt traffic sẽ tăng lên.
- ✓ **Theo dõi lỗi:** Đây là một công cụ để phát triển cũng như khắc phục lỗi vô cùng tuyệt vời. Nhờ vậy khi sử dụng, bạn không cần lo lắng mình sẽ để sót lỗi.
- ✓ **Chức năng sao lưu:** Cách sử dụng firestore sao lưu một cách thường xuyên và đảm bảo tính sẵn có. Đồng thời, chức năng này cũng giúp cho thông tin và dữ liệu được bảo mật một cách an toàn nhất.

#### **Nhược điểm:**

- ✓ **Firestore không là mã nguồn mở:** Điều này sẽ giúp cho ứng dụng trở thành một lựa chọn không quá tối ưu đối với nhiều nhà phát triển. Người dùng không thể sửa đổi được mã nguồn Firestore.
- ✓ **Người dùng không truy cập được mã nguồn:** Đối với những ứng dụng lớn thì việc chuyển đổi sang các nhà cung cấp khác thực sự không dễ dàng. Để làm được điều này thì toàn bộ Backend cần phải được xây dựng lại từ đầu.
- ✓ **Nền tảng không hoạt động nhiều quốc gia:** Firestore chính là một Subdomain của Google. Trang web chính thức của Firestore hiện tại đang bị chặn ở nhiều quốc gia trên thế giới, trong đó có cả Trung Quốc.
- ✓ **Firestore chỉ hoạt động với CSDL NoSQL:** Người dùng sẽ không thể xử lý được dữ liệu một cách nhanh chóng. Firestore chỉ sử dụng JSON và hầu như không có SQL. Chính vì vậy, để di chuyển từ cơ sở dữ liệu



sẽ không hề dễ dàng.

- ✓ **Firestore chỉ chạy trên Google Cloud:** Firestore trở thành một phần của Google và tất cả cơ sở hạ tầng của ứng dụng đều hoạt động trên Google Cloud. Người dùng không thể chạy ứng dụng trên những đơn vị cung cấp đám mây khác.
- ✓ **Truy vấn khám chậm:** Không phải các cơ sở dữ liệu đều phù hợp cho mọi trường hợp và tất nhiên Cloud Firestore cũng không là ngoại lệ. Điều này khiến bạn rất mất thời gian khi sử dụng ứng dụng.
- ✓ **Các dịch vụ cung cấp không phải đều miễn phí:** Không phải tất cả các dịch vụ được cung cấp đều miễn phí. Các chức năng trên đám mây sẽ chỉ có thể khả dụng trên gói Blaze và dĩ nhiên người dùng cũng không thể sử dụng được các dịch vụ ở trong gói Spark.
- ✓ **Thiếu hợp đồng doanh nghiệp:** Firestore không có những tùy chọn Dedicated Servers hoặc các hợp đồng doanh nghiệp. Để có thể sử dụng được Firestore chính là sử dụng cấu trúc Serverless ít linh hoạt hơn.
- ✓ **Không cung cấp API GraphQL:** Ứng dụng không cung cấp API GraphQL tương tự như một phần của quá trình thiết lập tiêu chuẩn. Mặc dù còn có những giải pháp thay thế khác thế nhưng REST vẫn là một tùy chọn mặc định của nền tảng này.

### 1.3. Bản quyền và quyền tác giả

Trong quá trình phát triển ứng dụng nghe nhạc, em đã gặp phải thách thức trong việc tìm kiếm nguồn nhạc phù hợp cho dự án mà không vi phạm quyền tác giả. Do hạn chế về tài nguyên và khả năng tiếp cận các nguồn nhạc miễn phí, em đã sử dụng một số nội dung có bản quyền trong phạm vi dự án.

Tuy nhiên, em xin nhấn mạnh rằng ứng dụng này được thực hiện hoàn toàn với mục đích học tập và nghiên cứu, không phục vụ bất kỳ mục đích thương mại nào. Em không có kế hoạch phát hành ứng dụng hoặc thu lợi nhuận từ việc sử dụng các nội dung này. Việc sử dụng các bản nhạc có bản quyền chỉ nhằm mục tiêu hoàn thiện đồ án tốt nghiệp và nâng cao kỹ năng

lập trình của bản thân.

Em cam kết tuân thủ nghiêm túc các quy định pháp luật liên quan đến quyền sở hữu trí tuệ và luôn tôn trọng quyền tác giả trong mọi hoạt động phát triển ứng dụng của mình. Em rất mong nhận được sự thông cảm và ủng hộ từ phía nhà trường cũng như cộng đồng đối với các hạn chế mang tính khách quan trong quá trình thực hiện dự án.

CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Khảo sát sơ bộ

2.1.1. Mục tiêu

Xây dựng được một ứng dụng nghe nhạc có những chức năng chính như sau:

- Phát nhạc, trình phát nhạc mượt mà, dễ sử dụng
- Cung cấp chức năng tìm kiếm, yêu thích bài hát
- Cung cấp chức năng thêm, sửa, xoá bài hát dễ sử dụng
- Cung cấp thông tin liên hệ, đưa ra ý kiến để cải thiện ứng dụng

2.1.2. Phương pháp

Phỏng vấn:

Bảng 2.1: Kế hoạch phỏng vấn

Kế hoạch phỏng vấn	
Người được hỏi: Lê Lê Na	Người phỏng vấn: Hà Thị Mai Linh
Địa chỉ : Nhà hàng ngay cổng chính trường Đại học Công Nghiệp Hà Nội	Thời gian phỏng vấn : 20/10/2024 Thời gian hẹn : 9h Thời gian bắt đầu : 9h30 Thời gian kết thúc : 10h30
Đối tượng : Người đang muốn xây dựng một ứng dụng nghe nhạc	Các yêu cầu đòi hỏi
Chương trình	Ước lượng thời gian
Giới thiệu	2 phút
Tổng quan về dự án	7 phút
Tổng quan về phỏng vấn	3 phút
Chủ đề sẽ đề cập. Xin phép được ghi âm	8 phút

Tổng hợp các nội dung chính, ý kiến của người được hỏi	10 phút
Kết thúc	
	Dự kiến tổng thời lượng : 30 phút

Phiếu phỏng vấn	
Dự án : Xây dựng ứng dụng nghe nhạc trực tuyến	
Người được hỏi: Lê Lê Na	Người hỏi: Hà Thị Mai Linh Ngày: 20/10/2024
Câu hỏi:	Trả lời:
Câu 1. Bạn muốn xây dựng ứng dụng nghe nhạc trực tuyến cho mục đích gì?	
Câu 2. Bạn đã nghiên cứu những ứng dụng tương tự chưa?	
Câu 3. Bạn đã xác định các tính năng và hiệu suất cần thiết cho ứng dụng của mình chưa?	
Câu 4. Bạn đã nghĩ về thiết kế và trải nghiệm người dùng của ứng dụng chưa?	
Câu 5. Những yêu cầu đặc biệt nào mà bạn muốn có?	

Câu 6: Bạn có mong muốn gì về hệ thống quản lý bài hát trong ứng dụng của mình?	
Đánh giá chung	

**Phiếu điều tra :**

<p style="text-align: center;">Phiếu khảo sát hoạt động của ứng dụng nghe nhạc trực tuyến (Dành cho người dùng)</p> <p>1. Vui lòng cho biết tên của bạn? .....</p> <p>2. Bạn đang ở độ tuổi nào?</p> <p>a. Dưới 18 tuổi</p> <p>b. Từ 18-25 tuổi</p> <p>c. Từ 26-35 tuổi</p> <p>d. Từ 36-45 tuổi</p> <p>e. Từ 46-55 tuổi</p> <p>f. Trên 55 tuổi</p> <p>3. Bạn là nam hay nữ?</p> <p>a. Nam</p> <p>b. Nữ</p> <p>c. Khác</p> <p>4. Bạn có thường xuyên nghe nhạc không?</p> <p>a. Có</p> <p>b. Không</p> <p>c. Thỉnh thoảng</p> <p>5. Bạn thường nghe nhạc vào thời gian nào trong ngày?</p> <p>a. Sáng</p> <p>b. Trưa</p>
---

- c. Chiều
  - d. Tối
  - e. Bất kỳ lúc nào
6. Bạn nghe nhạc bao lâu mỗi ngày?
- a. Dưới 1 giờ
  - b. 1-2 giờ
  - c. 2-3 giờ
  - d. Trên 3 giờ
7. Bạn chủ yếu nghe nhạc trên thiết bị nào?
- a. Điện thoại di động
  - b. Máy tính bảng
  - c. Máy tính để bàn, laptop
  - d. Thiết bị khác
8. Bạn đánh giá chất lượng âm thanh của ứng dụng như thế nào?
- a. Tốt
  - b. Bình thường
  - c. Trung bình
  - d. Kém
9. Bạn đánh giá giao diện người dùng của ứng dụng như thế nào?
- a. Dễ sử dụng
  - b. Bình thường
  - c. Khó sử dụng
10. Bạn sẵn sàng giới thiệu ứng dụng của chúng tôi cho người khác không?
- a. Rất sẵn sàng
  - b. Sẵn sàng
  - c. Không chắc chắn
  - d. Không sẵn sàng
11. Bạn có bất kỳ đề xuất nào để cải thiện ứng dụng không? Nếu có, vui lòng ghi rõ.

.....

12. Tính năng nào bạn muốn thấy trong các phiên bản tương lai của ứng dụng?

.....

Cảm ơn bạn đã dành thời gian để hoàn thành bảng khảo sát của chúng tôi. Ý kiến của bạn rất quan trọng đối với chúng tôi để cải thiện trải nghiệm của người dùng ứng dụng của chúng tôi.

### 2.1.3. Kết quả sơ bộ

Làm rõ được nhu cầu của người dùng, nắm được các nghiệp vụ của hệ thống và cách thức quản lý của người quản trị hệ thống.

## 2.2. Khảo sát chi tiết

### 2.2.1. Các yêu cầu chức năng

#### - Người dùng:

- **Đăng ký tài khoản:** Chức năng này giúp cho người dùng có thể đăng ký tài khoản để lưu các thông tin và thực hiện các chức năng khác.
- **Đăng nhập:** Chức năng này giúp người dùng đăng nhập và hệ thống để thực hiện các chức năng khác.
- **Phát bài hát, sử dụng trình phát nhạc:** Chức năng này giúp người dùng phát nhạc, điều hướng phát nhạc như: chuyển bài tiếp theo, chuyển bài trước, tạm dừng.
- **Phát theo thể loại nhạc, sử dụng trình phát nhạc:** Chức năng này giúp người dùng phát nhạc theo thể loại, điều hướng phát nhạc như: chuyển bài tiếp theo, chuyển bài trước, tạm dừng, phát ngẫu nhiên, phát lại bài hát.
- **Tìm kiếm bài hát:** Chức năng này cho phép người dùng tìm kiếm bài hát theo tên bài hát.
- **Thêm bài hát vào danh sách yêu thích:** Chức năng này cho phép người dùng yêu thích bài hát, bài hát sẽ được thêm vào danh sách yêu thích.

- **Cài đặt:** Ý kiến phản hồi, liên hệ nhà phát triển, thay đổi mật khẩu.
- **Người quản trị:**
  - **Đăng ký tài khoản:** Chức năng này giúp cho người quản trị có thể đăng ký tài khoản để lưu các thông tin và thực hiện các chức năng khác.
  - **Đăng nhập:** Chức năng này giúp người quản trị đăng nhập và hệ thống để thực hiện các chức năng khác.
  - **Phát bài hát, sử dụng trình phát nhạc:** Chức năng này giúp người quản trị phát nhạc, điều hướng phát nhạc như: chuyển bài tiếp theo, chuyển bài trước, tạm dừng.
  - **Tìm kiếm bài hát:** Chức năng này cho phép người quản trị tìm kiếm bài hát theo tên bài hát.
  - **Đọc ý kiến phản hồi:** Chức năng này cho phép người quản trị đọc các ý kiến phản hồi của người dùng.
  - **Quản lí bài hát:** Người quản trị có thể thêm, cập nhật, xoá thông tin của bài hát.

### 2.2.2. Các yêu cầu phi chức năng

- Dữ liệu hiển thị: Các thông tin của bài hát phải đầy đủ và rõ ràng.
- Chức năng của hệ thống: Các chức năng của ứng dụng phải dễ dàng sử dụng, tập trung vào những nhu cầu cần thiết và tránh các thủ tục rườm rà.

### 2.2.3. Giao diện

- Giao diện đơn giản, hài hòa, thân thiện để người dùng dễ dàng sử dụng.
- Các tính năng chính hiển thị lên màn hình có điểm nhấn, dễ dàng nhận biết.
- Thiết kế ứng dụng phải tương thích với nhiều thiết bị khác nhau.



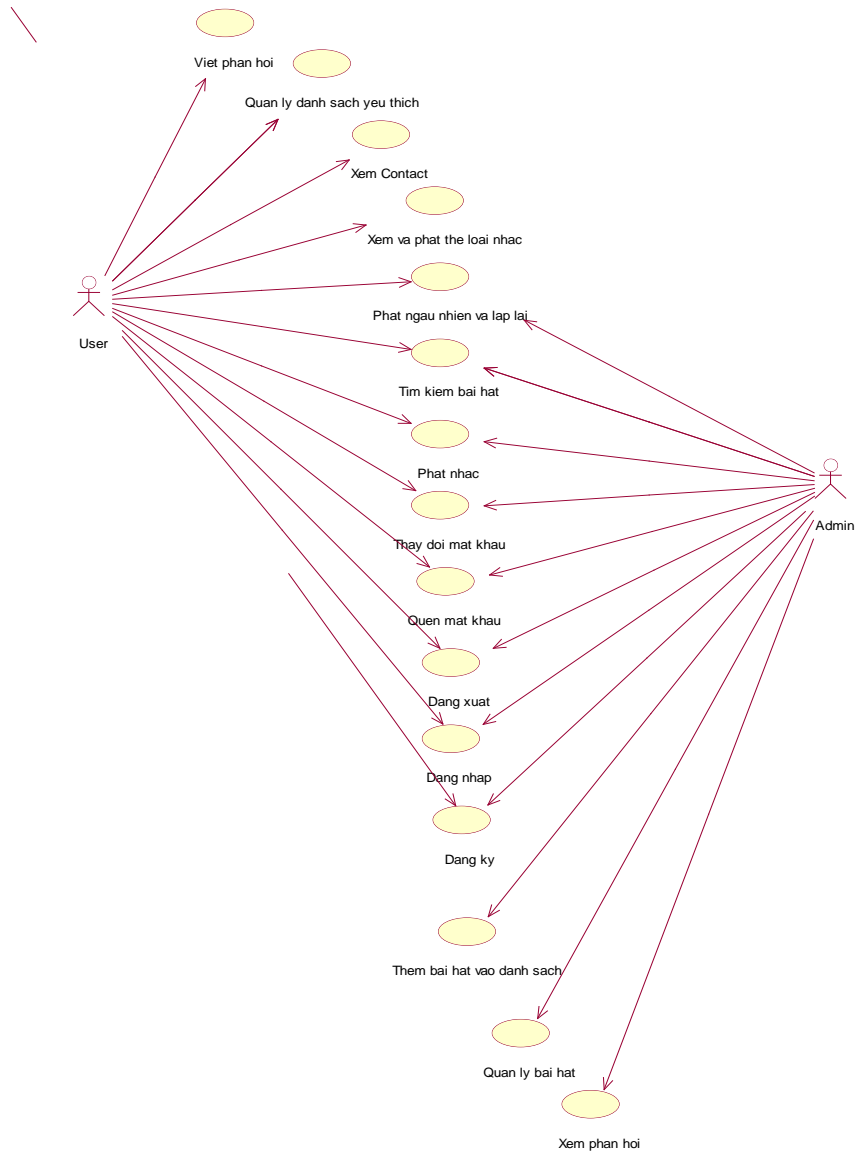
### 2.3. Xác định tác nhân

*Bảng 2.2: Các tác nhân*

STT	Vai trò	Ca sử dụng
1	Người dùng	Đăng nhập Đăng ký Phát nhạc Phát nhạc theo thể loại Điều khiển trình phát nhạc Tìm kiếm Thêm bài hát vào mục yêu thích Ý kiến phản hồi Liên hệ nhà phát triển Thay đổi mật khẩu
2	Người quản trị	Đăng nhập Đăng ký Phát nhạc Điều khiển trình phát nhạc Tìm kiếm Đọc kiến phản hồi Thay đổi mật khẩu Quản lí bài hát

## 2.4. Biểu đồ use case

### Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

1. **Đăng ký:** cho phép admin/user đăng ký tài khoản để thực hiện chức năng đăng nhập vào hệ thống.
2. **Đăng nhập:** cho phép admin/user đăng nhập vào hệ thống để thực hiện các chức năng của mình.
3. **Đăng xuất:** cho phép người dùng đăng xuất tài khoản ra khỏi hệ thống.
4. **Quên mật khẩu:** cho phép người dùng lấy lại mật khẩu.
5. **Thay đổi mật khẩu:** cho phép người dùng thay đổi mật khẩu.
6. **Phát nhạc:** cho phép người dùng nghe nhạc.
7. **Phát ngẫu nhiên và lặp lại bài:** cho phép người dùng có thể phát ngẫu nhiên bài hát hoặc lặp lại bài hát đang được phát.
8. **Xem và phát thể loại nhạc:** cho phép người dùng có thể xem và phát danh sách bài hát theo thể loại.
9. **Tìm kiếm bài hát:** cho phép người dùng tìm kiếm bài hát
10. **Quản lý danh sách yêu thích:** cho phép người dùng xem và phát danh sách yêu thích, thêm, xóa bài hát trong danh sách yêu thích.
11. **Viết Feedback (Ý kiến phản hồi):** cho phép người dùng viết ý kiến phản hồi về ứng dụng.
12. **Xem Contact:** cho phép người dùng xem các nền tảng mạng xã hội của nhà phát triển.
13. **Thêm bài hát:** cho phép người quản trị thêm bài hát vào hệ thống
14. **Quản lý bài hát:** cho phép người quản trị xem, sửa, xóa thông tin bài hát
15. **Xem Feedback (Xem ý kiến phản hồi):** cho phép người quản trị xem danh sách các ý kiến phản hồi của người dùng.

## 2.5. Mô tả chi tiết các Use case

### 2.5.1. Mô tả use case Đăng ký

Use case cho phép admin/user đăng ký tài khoản để thực hiện chức năng đăng nhập vào hệ thống.

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng kích vào nút “Đăng ký” , hệ thống hiển thị màn hình đăng ký.

- 2) Người dùng nhập đầy đủ thông tin email, mật khẩu, lựa chọn vai trò người dùng (admin hoặc user) và ấn nút đăng ký, hệ thống kiểm tra thông tin hợp lệ chuyển sang màn hình chính.

Use case kết thúc.

- Luồng rẽ nhánh:

- 1) Tại bước 2 nếu người dùng đăng ký thông tin không hợp lệ, hệ thống hiển thị thông báo đăng ký không thành công.

- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Không có

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.2. Mô tả use case Đăng nhập

Use case cho phép admin/user đăng nhập vào hệ thống để thực hiện các chức năng của mình.

- **Luồng sự kiện:**

- Luồng cơ bản:

- i. Use case bắt đầu khi admin/user lần đầu sử dụng ứng dụng hoặc kích vào nút “Đăng nhập”, hệ thống hiển thị màn hình đăng nhập.

- ii. Người dùng nhập đầy đủ thông tin email, mật khẩu và ấn nút Sign in, hệ thống kiểm tra tài khoản hợp lệ chuyển sang màn hình chính.

Use case kết thúc.

- Luồng rẽ nhánh:

- 1) Tại bước 2 nếu người dùng đăng nhập thông tin không hợp lệ, hệ thống hiển thị thông báo đăng nhập không thành công.

- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Không có

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.3. Mô tả use case Đăng xuất

Use case cho phép người dùng đăng xuất tài khoản ra khỏi hệ thống.

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng kích vào mục cài đặt, hệ thống hiển thị tên tài khoản và danh sách cài đặt.
- 2) Người dùng ấn vào chữ “Đăng xuất”, hệ thống chuyển về màn hình đăng nhập.  
Use case kết thúc.

- Luồng rẽ nhánh:

- 1) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.4. Mô tả use case Quên mật khẩu

Use case cho phép người dùng lấy lại mật khẩu

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng kích vào mục “Quên mật khẩu” ở màn hình đăng nhập, hệ thống hiển thị yêu cầu nhập email.
- 2) Người dùng điền email và ấn vào chữ “Thiết lập lại mật khẩu”, hệ thống hiển thị thông báo “Mật khẩu đã cập nhật thành công. Vui lòng kiểm tra email của bạn”.

Use case kết thúc.

○ Luồng rẽ nhánh:

- 1) Tại bước 2, khi người dùng điền email không hợp lệ, hệ thống sẽ hiển thị thông báo “Email của bạn không hợp lệ”
- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

• **Các yêu cầu đặc biệt:**

Không có

• **Tiền điều kiện:**

Không có

• **Hậu điều kiện:**

Không có

• **Điểm mở rộng:**

Không có

### 2.5.5. Mô tả use case Thay đổi mật khẩu

Use case cho phép người dùng thay đổi mật khẩu

• **Luồng sự kiện:**

○ Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng kích vào mục cài đặt, hệ thống hiển thị tên tài khoản và danh sách cài đặt.
- 2) Người dùng ấn vào chữ “Thay đổi mật khẩu”, hệ thống hiển thị màn hình thay đổi mật khẩu.
- 3) Người dùng điền thông tin yêu cầu (mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới) rồi nhấn vào “Thay đổi mật khẩu”. Hệ thống hiển thị thông báo thay đổi mật khẩu thành công.

Use case kết thúc.

○ Luồng rẽ nhánh:

- 1) Tại bước 3, khi người dùng điền mật khẩu cũ không hợp lệ, hệ thống sẽ hiển thị thông báo “Mật khẩu cũ của bạn không đúng”.

- 2) Tại bước 3, khi người dùng điền mật khẩu mới và xác nhận mật khẩu mới không hợp lệ, hệ thống sẽ hiển thị thông báo “Xác nhận mật khẩu mới không trùng khớp”.
- 3) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

#### 2.5.6. Mô tả use case Phát nhạc

Use case cho phép người dùng nghe nhạc

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng kích vào bài hát ở bất cứ trang giao diện nào (Trang chủ, slide, mục tìm kiếm, danh sách yêu thích)
- 2) Hệ thống phát nhạc cho người dùng, người dùng có thể thực hiện các thao tác: Tạm dừng bài hát, tiếp tục phát sau khi tạm dừng, chuyển bài hát tiếp theo, chuyển bài hát trước đó, phát ngẫu nhiên các bài hát trong danh sách, lặp lại bài hát hiện tại.
- 3) Người dùng ấn tạm dừng hoặc biểu tượng “x” ở thanh trình phát nhạc.  
Use case kết thúc.

- Luồng rẽ nhánh:

- 1) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông



báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.7. Mô tả use case Phát ngẫu nhiên và lặp lại bài

Use case cho phép người dùng có thể phát ngẫu nhiên bài hát hoặc lặp lại bài hát đang được phát

- **Luồng sự kiện:**

- Luồng cơ bản:

1) Use case bắt đầu khi người dùng chọn vào biểu tượng phát ngẫu nhiên hoặc lặp lại bài, hệ thống thay đổi chế độ phát theo yêu cầu của người dùng.

2) Người dùng bỏ chọn cả 2 biểu tượng.

Use case kết thúc

- Luồng rẽ nhánh:

1) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống và truy cập vào giao diện phát nhạc của ứng dụng

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.8. Mô tả use case Xem và phát thể loại nhạc

Use case cho phép người dùng có thể xem và phát danh sách bài hát theo thể loại

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng chọn vào View all và Play all theo từng thể loại ở màn hình chính
- 2) Người dùng thoát khỏi danh sách bài hát theo thể loại và dừng phát nhạc.

Use case kết thúc

- Luồng rẽ nhánh:

- 1) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.9. Mô tả use case Tìm kiếm bài hát

Use case cho phép người dùng tìm kiếm bài hát

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng điền từ khoá vào mục tìm kiếm trên ứng dụng và ấn vào biểu tượng tìm kiếm.
- 2) Hệ thống tìm kiếm và hiển thị các kết quả phù hợp với từ khoá tìm kiếm
- 3) Người dùng chọn một trong các kết quả.

Use case kết thúc

○ Luồng rẽ nhánh:

- 1) Tại bước 2, nếu tìm kiếm không có kết quả, hệ thống hiển thị thông báo không có kết quả phù hợp.
- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

• **Các yêu cầu đặc biệt:**

Không có

• **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

• **Hậu điều kiện:**

Không có

• **Điểm mở rộng:**

Không có

#### 2.5.10. Mô tả use case Quản lý danh sách yêu thích

Use case cho phép người dùng xem và phát danh sách yêu thích, thêm, xoá bài hát trong danh sách yêu thích.

• **Luồng sự kiện:**

○ Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng ấn vào mục “Yêu thích”
- 2) Hệ thống hiển thị danh sách các bài hát đã được người dùng thêm vào
- 3) Người dùng có thể thực hiện các hoạt động như thêm, xoá, phát bài hát từ danh sách hoặc phát danh sách.
- 4) Người dùng thoát khỏi mục “Yêu thích”.

Use case kết thúc.

○ Luồng rẽ nhánh:

- 1) Tại bước 2, nếu danh sách yêu thích trống, sẽ hiển thị thông báo danh sách đang trống
- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

• **Các yêu cầu đặc biệt:**

Không có

• **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

• **Hậu điều kiện:**

Không có

• **Điểm mở rộng:**

Không có

### 2.5.11. Mô tả use case Viết Phản hồi

Use case cho phép người dùng viết ý kiến phản hồi về ứng dụng

• **Luồng sự kiện:**

○ Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng ấn vào mục Cài đặt và ấn vào mục Phản hồi
  - 2) Hệ thống hiển thị màn hình Phản hồi bao gồm: Tên, số điện thoại, email và ý kiến của người dùng.
  - 3) Người dùng ấn vào nút “Gửi phản hồi”, hệ thống hiển thị gửi ý kiến thành công.
- Use case kết thúc.

○ Luồng rẽ nhánh:

- 1) Tại bước 2, nếu người dùng để trống mục tên hoặc ý kiến, hệ thống sẽ thông báo không được để trống
- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông

báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống với vai trò User (Người dùng)

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

#### 2.5.12. Mô tả use case Xem Contact

Use case cho phép người dùng xem các nền tảng mạng xã hội của nhà phát triển

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng ấn vào mục Cài đặt và ấn vào mục Liên hệ
- 2) Hệ thống hiển thị màn hình Liên hệ bao gồm: Logo, các tài khoản xã hội của nhà phát triển.
- 3) Người dùng ấn vào biểu tượng quay lại.
- 4) Use case kết thúc.

- Luồng rẽ nhánh:

- 1) Tại bước 2, nếu người dùng click vào bất kỳ biểu tượng mạng xã hội nào, hệ thống sẽ chuyển đến liên kết đó.
- 2) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối”.  
Use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống với vai trò User (Người dùng)

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Không có

### 2.5.13. Mô tả use case Thêm bài hát

Use case cho phép người quản trị thêm bài hát vào hệ thống

- **Luồng sự kiện**

- **Luồng cơ bản**

1. Use case này bắt đầu khi người quản trị kích vào biểu tượng dấu cộng ở góc màn hình trang chủ.

2. Thêm bài hát

a. Hệ thống hiển thị biểu mẫu nhập thông tin gồm: tên bài hát, tên tác giả, ảnh minh họa, link âm thanh, , thể loại của bài hát.

b. Người quản trị nhập thông tin tên bài hát, tên tác giả, ảnh minh họa, link âm thanh, lựa chọn thể loại của bài hát và kích vào nút “Add”. Hệ thống sẽ có một bài hát mới và hiển thị danh sách các bài hát đã được cập nhật.

Use case kết thúc.

- **Luồng rẽ nhánh**

1. Tại bước 2b trong luồng cơ bản nếu người quản trị nhập thông tin bài hát không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào biểu tượng quay lại để kết thúc.

2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt**

Use case này chỉ cho phép một số vai trò như người quản trị thực hiện

- **Tiền điều kiện**

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

- **Hậu điều kiện:**

Nếu use case kết thúc thành công thì thông tin về bài hát sẽ được cập nhật trong cơ sở dữ liệu Firebase.

- **Điểm mở rộng:**

Không có.

#### 2.5.14. Mô tả use case Quản lý bài hát

Use case cho phép người quản trị xem, sửa, xóa thông tin bài hát

- **Luồng sự kiện**

- **Luồng cơ bản**

1. Use case này bắt đầu khi người quản trị kích vào trang chủ. Hệ thống lấy thông tin chi tiết của sản phẩm gồm: Tên bài hát, tên tác giả, ảnh minh họa, thể loại và hiển thị danh sách các bài hát lên màn hình trang chủ.

2. Sửa bài hát

a. Người quản trị kích vào icon “Sửa” trên một bài hát. Hệ thống sẽ lấy thông tin cũ của bài hát được chọn gồm: tên bài hát, tên tác giả, ảnh minh họa, link âm thanh, , thể loại của bài hát và hiển thị lên màn hình.

b. Người quản trị nhập thông tin mới cho bài hát và kích vào nút “Sửa”. Hệ thống sẽ sửa thông tin của sản phẩm được chọn trong và hiển thị danh sách bài hát đã cập nhật.

3. Xóa bài hát

a. Người quản trị kích vào icon “Xóa” trên một bài hát. Hệ

thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.

- b. Người quản trị kích vào nút “OK”. Hệ thống sẽ xóa bài hát khỏi hệ thống và hiển thị danh sách các bài hát đã cập nhật.

Use case kết thúc.

#### ○ **Luồng rẽ nhánh**

1. Tại bước 2b trong luồng cơ bản nếu người quản trị nhập thông tin bài hát không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào biểu tượng quay lại để kết thúc.
2. Tại bước 2b trong luồng cơ bản nếu người quản trị kích vào biểu tượng quay lại, hệ thống sẽ bỏ qua thao tác cập nhật tương ứng và hiển thị danh sách các bài hát.
3. Tại bước 3b trong luồng cơ bản nếu người quản trị kích vào nút “Cancel” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các bài hát.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

#### ● **Các yêu cầu đặc biệt**

Use case này chỉ cho phép một số vai trò như người quản trị thực hiện

#### ● **Tiền điều kiện**

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

#### ● **Hậu điều kiện**

Nếu use case kết thúc thành công thì thông tin về bài hát sẽ được cập nhật trong cơ sở dữ liệu.

#### ● **Điểm mở rộng**

Không có.



### 2.5.15. Mô tả use case Xem Phản hồi

Use case cho phép người quản trị có thể xem danh sách các phản hồi của người dùng

- **Luồng sự kiện:**

- Luồng cơ bản:

- 1) Use case bắt đầu khi người quản trị chọn vào mục Phản hồi ở giao diện admin

- 2) Hệ thống hiển thị danh sách các phản hồi của người dùng bao gồm: tên người dùng, ý kiến đóng góp

- 3) Người quản trị chuyển sang mục khác.

- 4) Use case kết thúc.

- Luồng rẽ nhánh:

- 1) Nếu không kết nối được cơ sở dữ liệu tại bất kỳ thời điểm nào trong quá trình thực hiện use case thì hệ thống sẽ hiển thị thông báo “lỗi kết nối” và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Use case này chỉ cho phép một số vai trò như người quản trị thực hiện

- **Tiền điều kiện:**

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

- **Hậu điều kiện:**

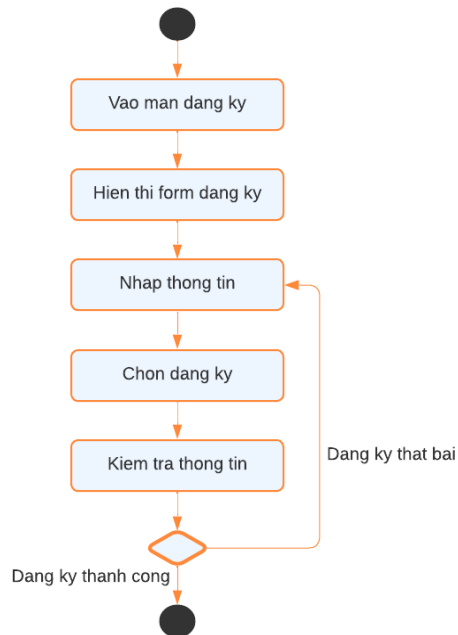
Không có

- **Điểm mở rộng:**

Không có

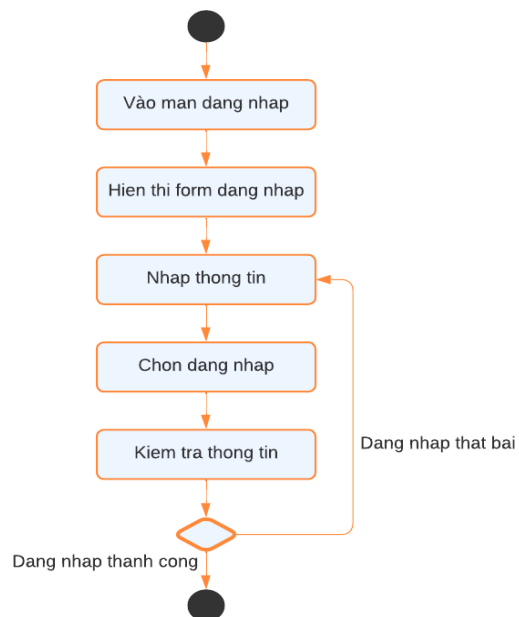
## 2.6. Sơ đồ hoạt động

### 2.6.1. Chức năng đăng ký tài khoản



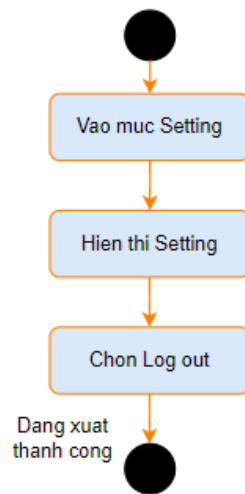
Hình 2.2: Sơ đồ hoạt động chức năng đăng ký tài khoản

### 2.6.2. Chức năng đăng nhập



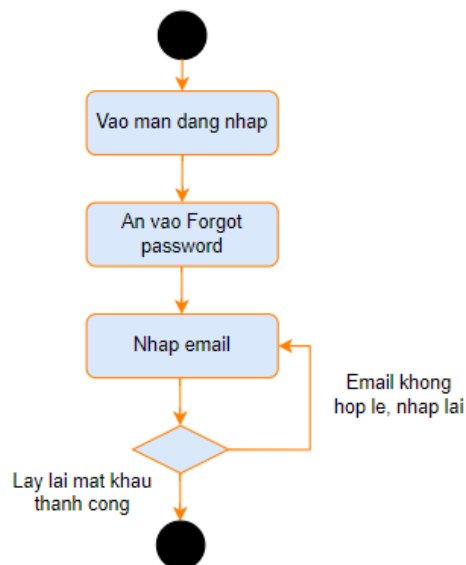
Hình 2.3: Sơ đồ hoạt động chức năng đăng nhập

### 2.6.3. Chức năng đăng xuất tài khoản



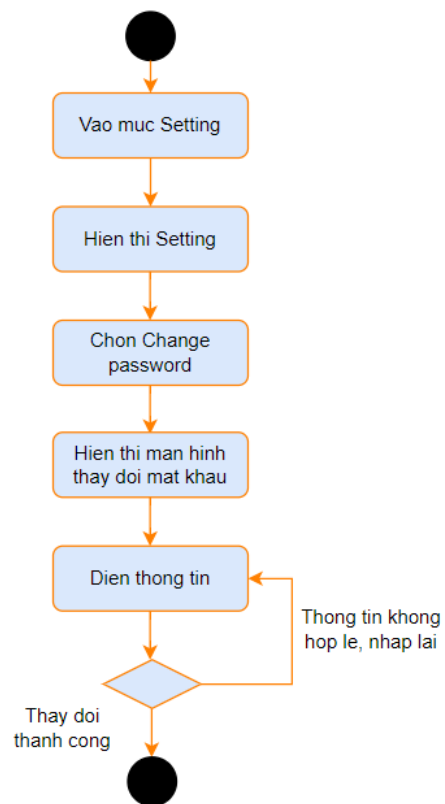
Hình 2.4: Sơ đồ hoạt động chức năng đăng xuất

### 2.6.4. Chức năng quên mật khẩu



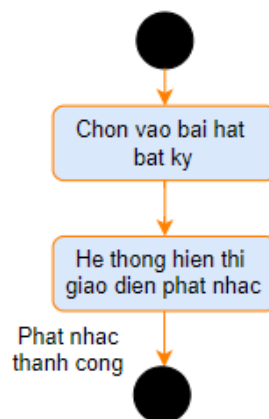
Hình 2.5: Sơ đồ hoạt động chức năng quên mật khẩu

### 2.6.5. Chức năng thay đổi mật khẩu



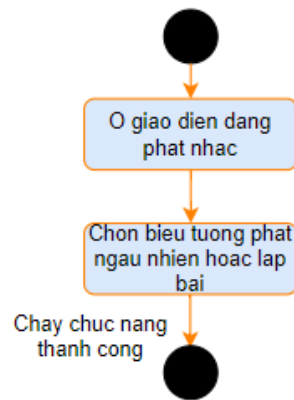
Hình 2.6: Sơ đồ hoạt động chức năng thay đổi mật khẩu

### 2.6.6. Chức năng phát nhạc



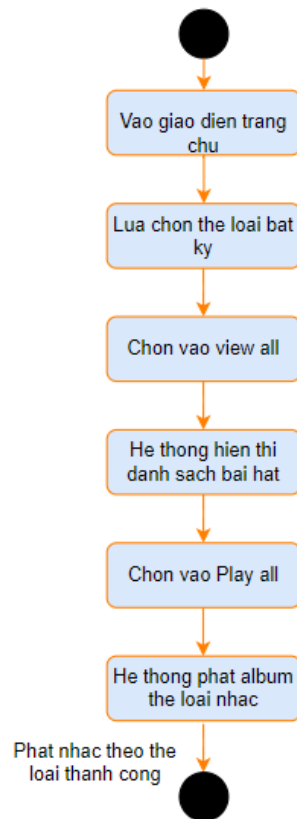
Hình 2.7: Sơ đồ hoạt động chức năng phát nhạc

### 2.6.7. Chức năng phát ngẫu nhiên và lặp lại bài



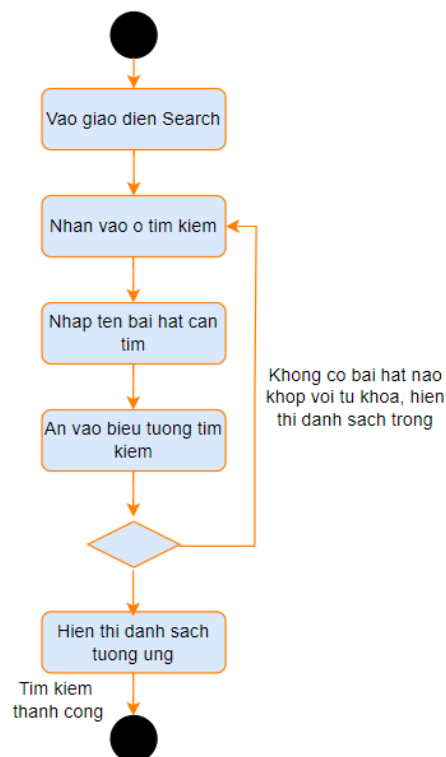
Hình 2.8: Sơ đồ hoạt động chức năng phát ngẫu nhiên và lặp lại bài

### 2.6.8. Chức năng xem và phát thể loại nhạc



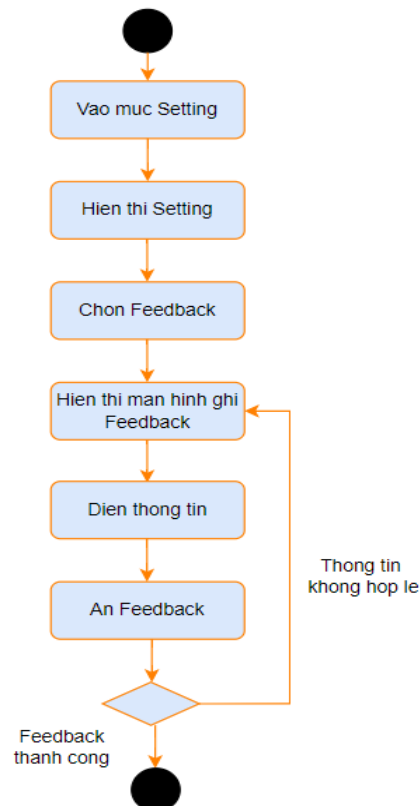
Hình 2.9: Sơ đồ hoạt động chức năng xem và phát thể loại nhạc

### 2.6.9. Chức năng tìm kiếm bài hát



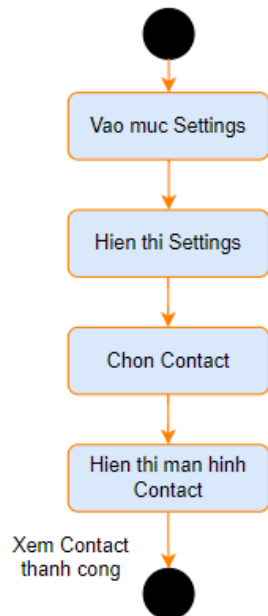
Hình 2.10: Sơ đồ hoạt động chức năng tìm kiếm bài hát

### 2.6.10. Chức năng viết phản hồi



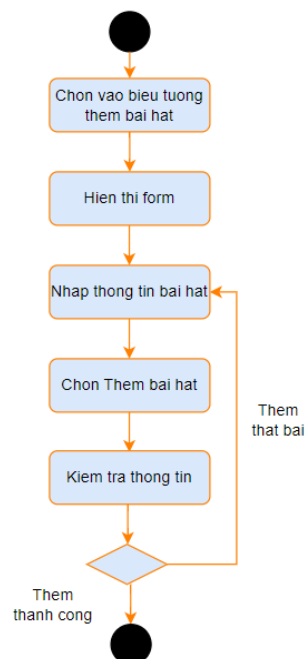
Hình 2.11: Sơ đồ hoạt động chức năng viết phản hồi

### 2.6.11. Chức năng xem Contact



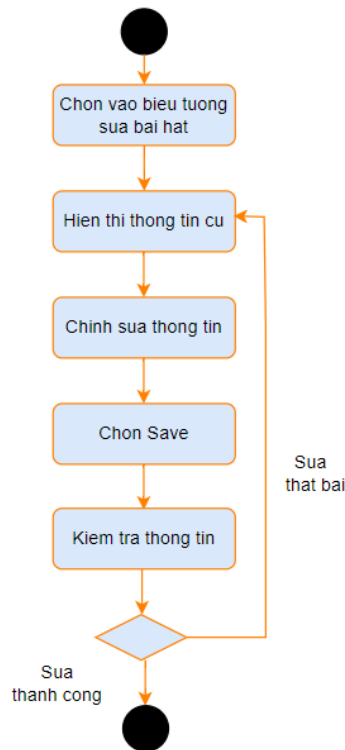
Hình 2.12: Sơ đồ hoạt động chức năng xem Contact

### 2.6.12. Chức năng thêm bài hát



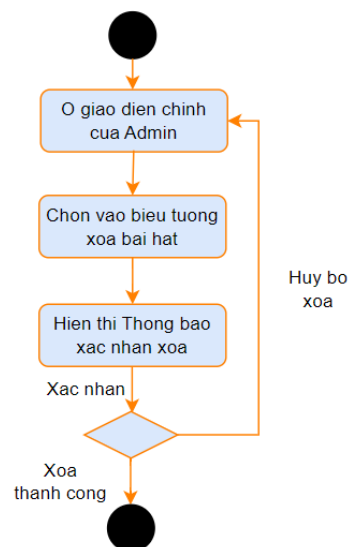
Hình 2.13: Sơ đồ hoạt động chức năng thêm bài hát

### 2.6.13. Chức năng sửa bài hát



Hình 2.14: Sơ đồ hoạt động chức năng sửa bài hát

### 2.6.14. Chức năng xoá bài hát



Hình 2.15: Sơ đồ hoạt động chức năng xoá bài hát



## 2.7. Thiết kế cơ sở dữ liệu Realtime database

### 2.7.1. Cấu trúc cơ sở dữ liệu

```
"feedback": {
  "1671591727009": {
    "comment": "App very good",
    "email": "mailinh@gmail.com",
    "name": "Mai Linh",
    "phone": "0986868687"
  }
},
"songs": {
  "artist": "Mohamed Aliba",
  "artistId": 2,
  "category": "Jazz",
  "categoryId": 1,
  "count": 82,
  "id": 2,
  "image": "https://images.wallpaperscraft",
  "title": "Headphones Song Music",
  "url": "https://www.soundhelix.com/exam"
```

Hình 2.16: Cấu trúc dạng Json

```
{
  feedback : {
    1715709100630 : { 4 props }
  }
  songs : {
    1 : {
      artist : Lukas Graham ft. Vu
      count : 1222
      featured : ☒ true
      id : 1
      image : https://firebasestorage.google[
        ae293.appspot.com/o/images%2f
        alt=media&token=8a78c0fc-abb:
      latest : ☐ false
      title : Happy For You
      url : https://firebasestorage.google[
        ae293.appspot.com/o/mp3%2F9%20t
        0Video.mp3?alt=media&token=c31:
    }
  }
}
```

Hình 2.17: Cấu trúc dạng Tree

### 2.7.2. Kiểu dữ liệu

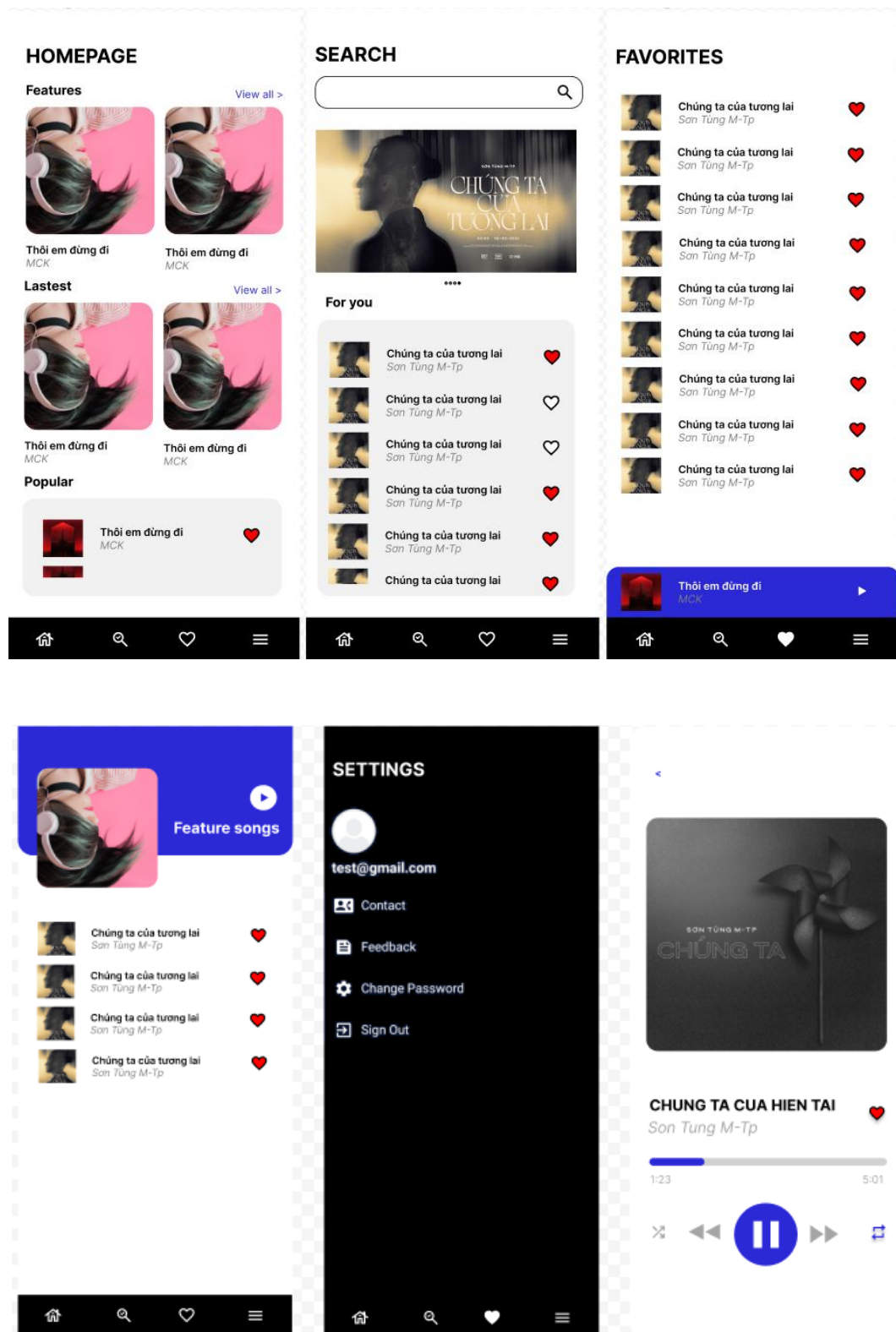
Bảng 1.3: Kiểu dữ liệu của Phản hồi

Phản hồi			
STT	THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
1	comment	String	Góp ý của người dùng
2	email	String	Tài khoản email
3	name	String	Tên người dùng
4	phone	String	Số điện thoại

Bảng 2.4: Kiểu dữ liệu của Bài hát

Bài hát			
STT	THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
1	Id	Int	Mã bài hát
2	count	Int	Số lượt nghe
3	artist	String	Tên ca sĩ
4	image	String	Đường dẫn ảnh
5	latest	Boolean	Tên thể loại
6	featured	Boolean	Tên thể loại
7	title	String	Tên bài hát
8	url	String	Đường dẫn ảnh

## 2.8. Thiết kế giao diện



Hình 2.18: Giao diện một số màn hình được thiết kế bằng Figma

## CHƯƠNG 3: KẾT QUẢ THỰC HIỆN VÀ KIỂM THỬ HỆ THỐNG

### 3.1. Cài đặt môi trường

#### 3.1.1. Cài đặt môi trường Java

##### - Cài đặt JDK – Java Development Kit

- Bước 1: Truy cập đường link sau: [Java Downloads | Oracle](#)
- Bước 2: Chọn JDK phù hợp với máy tính để tải xuống, sau đó tiến hành cài đặt.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux	macOS	Windows
Product/file description	File size	Download
Arm 64 Compressed Archive	179.90 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_linux-aarch64_bin.tar.gz">https://download.oracle.com/java/19/latest/jdk-19_linux-aarch64_bin.tar.gz</a> ( sha256)
Arm 64 RPM Package	159.99 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_linux-aarch64_bin.rpm">https://download.oracle.com/java/19/latest/jdk-19_linux-aarch64_bin.rpm</a> ( sha256)
x64 Compressed Archive	181.11 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_linux-x64_bin.tar.gz">https://download.oracle.com/java/19/latest/jdk-19_linux-x64_bin.tar.gz</a> ( sha256)
x64 Debian Package	154.77 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_linux-x64_bin.deb">https://download.oracle.com/java/19/latest/jdk-19_linux-x64_bin.deb</a> ( sha256)
x64 RPM Package	161.63 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_linux-x64_bin.rpm">https://download.oracle.com/java/19/latest/jdk-19_linux-x64_bin.rpm</a> ( sha256)

JDK Script-friendly URLs

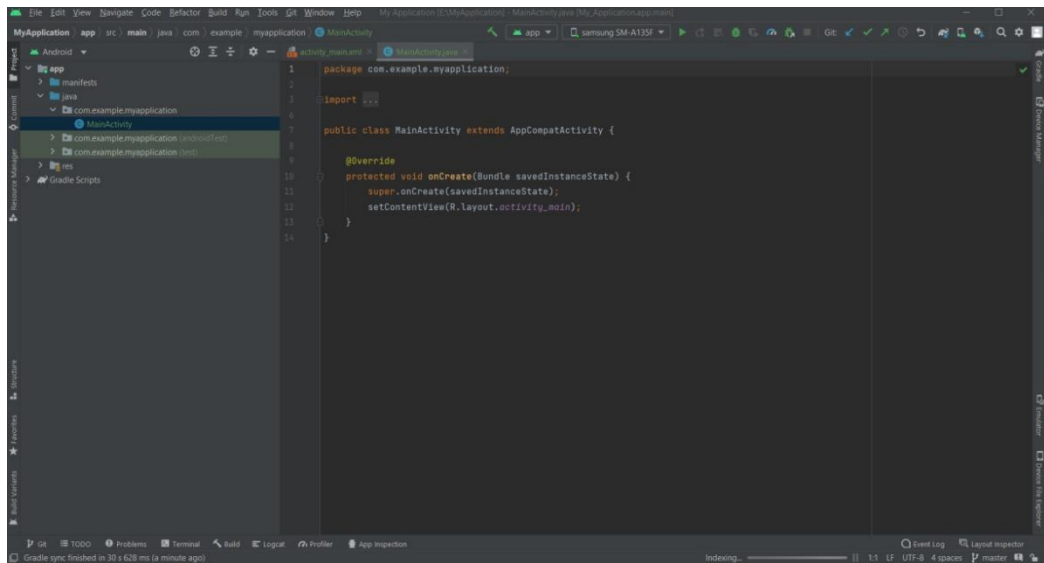
The URLs listed above will remain the same for JDK update releases to allow their use in scripts.

[Learn more about automating the downloads of JDK](#)

Hình 3.1. Một số phiên bản JDK trên trang Oracle

#### 3.1.2. Cài đặt Android Studio

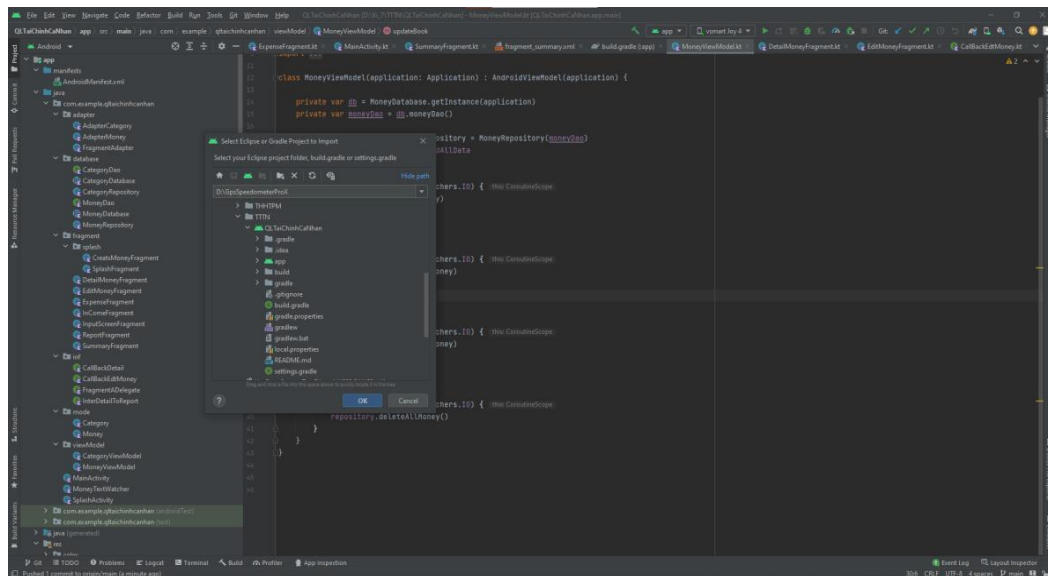
- Bước 1: Truy cập vào trang android theo đường link sau: [Download Android Studio & App Tools - Android Developers](#)
- Bước 2: Tải xuống Android Studio, sau đó cài đặt và lựa chọn thông số máy ảo sao cho phù hợp với cấu hình của máy tính:



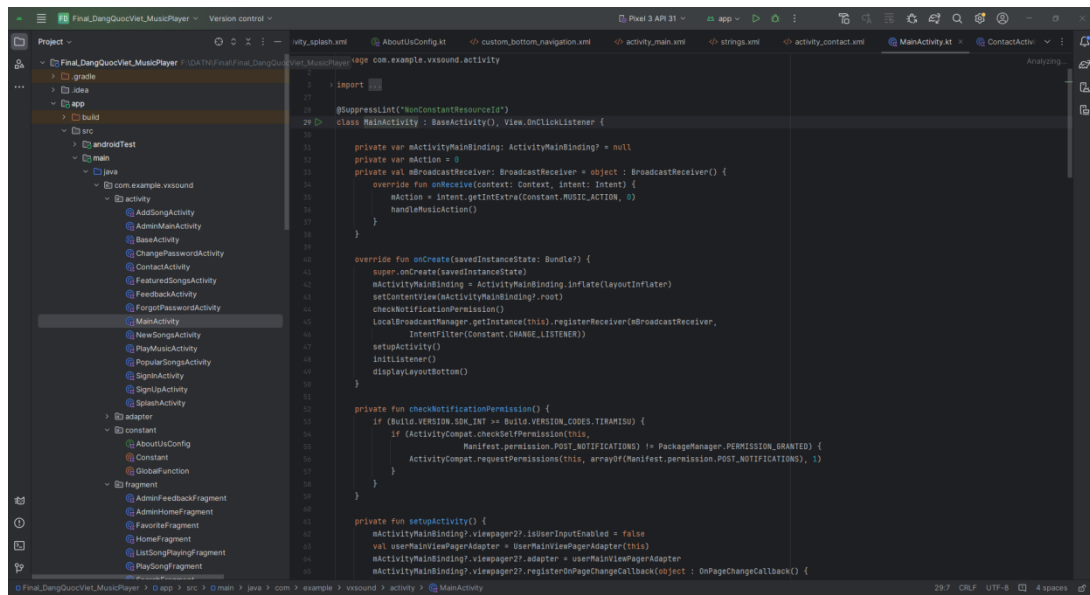
Hình 3.2. Giao diện của Android Studio sau khi cài đặt thành công

### 3.2. Cài đặt chương trình

Để mở file chương trình, chọn File, sau đó chọn Open, rồi chọn thư mục có chứa chương trình:



Hình 3.3. Cài đặt chương trình



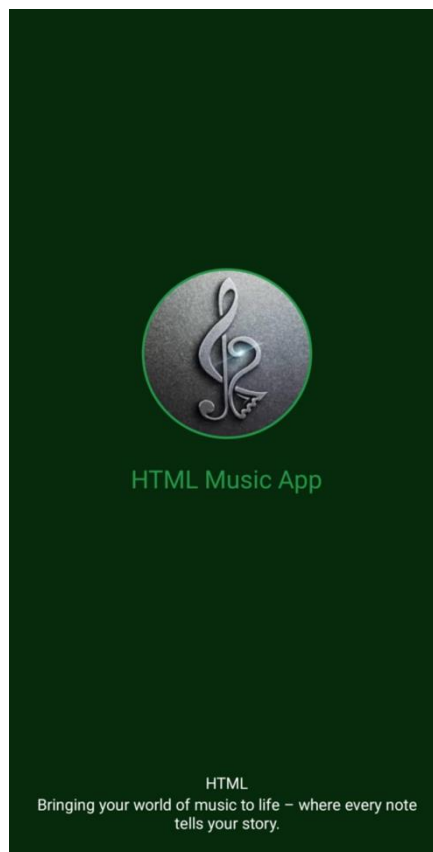
Hình 3.4. Giao diện Android Studio sau khi đã nhập mã nguồn

Giao diện chính gồm các thành phần sau:

- ❖ Ở bên trái là cấu trúc của project:
  - Thư mục **manifests** chứa tập tin AndroidManifests.xml mô tả các thành phần của ứng dụng, quyền của ứng dụng và thứ tự thực thi.
  - Thư mục **java** chứa các package hoặc các class khi xây dựng chương trình.
  - Thư mục **res** chứa tất cả các resources của chương trình như hình ảnh bitmap, layout, mipmap, values, các tập tin XML chứa các đoạn strings hiển thị:
  - Thư mục **drawable** chứa các ảnh khác nhau của chương trình.
  - Thư mục **layout** sẽ chứa tất cả các file XML Layout đã sử dụng để xác định giao diện người dùng của ứng dụng.
  - Thư mục **mipmap** sẽ chứa các biểu tượng ứng dụng (launcher icon) được sử dụng để hiển thị trên màn hình chính. Các loại biểu tượng sẽ có tỷ trọng khác nhau như hdpi, mdpi, xhdpi, xxhdpi, xxxhdpi, để sử dụng dựa trên kích thước của thiết bị.
  - Thư mục **values** sẽ chứa một số file XML khác nhau, chẳng hạn như strings quy định mã cho các chuỗi hiển thị (2 file strings cho 2 ngôn

ngữ khác nhau), colors quy định màu hiển thị (2 bảng màu cho 2 chế độ màn hình khác nhau), styles quy định kiểu cách hiển thị.

- Thư mục **Gradle Scripts**: Trong Android, Gradle là công cụ xây dựng hệ thống, Gradle được tích hợp sẵn vào Android Studio và được điều khiển một cách tự động thông qua Android Studio.
- ❖ Ở bên phải là nội dung chi tiết của tập tin mà bạn đang chỉnh sửa.
  - Tại đây chúng ta có thể xây dựng giao diện trên tập tin XML.
  - Xây dựng chương trình bằng code Kotlin trên tập tin JAVA.



*Hình 3.5. Giao diện ban đầu của chương trình sau khi build*

### 3.3. Kết quả thu được

Hệ thống xây dựng và hoàn thiện các chức năng:

- Quản lý tài khoản: Đăng ký, đăng nhập, thay đổi mật khẩu, quên mật khẩu.
- Phát nhạc, phát nhạc theo thể loại, quản lý trình phát nhạc (chuyển bài, mở ngẫu nhiên hoặc lặp lại bài)

- Quản lý danh sách yêu thích
- Tìm kiếm bài hát
- Gửi phản hồi
- Quản lý các bài hát.



### 3.3.1. Chức năng đăng nhập

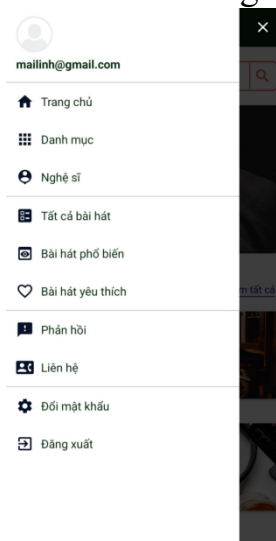
Chức năng đăng nhập cho phép người dùng nhập thông tin tài khoản để truy cập vào ứng dụng và sử dụng các tính năng của ứng dụng.

Bạn chưa có tài khoản? [Đăng ký](#)

Hình 3.6. Giao diện chức năng đăng nhập

### 3.3.1. Chức năng đăng xuất

Chức năng đăng xuất giúp người dùng thoát khỏi tài khoản, đảm bảo bảo mật và bảo vệ thông tin cá nhân khi không sử dụng ứng dụng.



Hình 3.7. Giao diện màn hình chức năng đăng xuất

### 3.3.3. Chức năng đăng ký

Chức năng đăng ký cho phép người dùng tạo tài khoản mới để bắt đầu sử dụng ứng dụng.

←

**Đăng ký**

Email...

Mật khẩu...

☐ Admin ☒ User

**ĐĂNG KÝ**

Bạn đã có tài khoản? [Đăng nhập](#)

Hình 3.8. Giao diện màn hình chức năng đăng ký

### 3.3.4. Chức năng thay đổi mật khẩu

Chức năng thay đổi mật khẩu cho phép người dùng cập nhật mật khẩu của tài khoản nếu cần thiết, giúp đảm bảo tính bảo mật.

☰ **Đổi mật khẩu**

Mật khẩu cũ...

Mật khẩu mới...

Xác nhận mật khẩu mới...

**ĐỔI MẬT KHẨU**

Hình 3.9. Giao diện màn hình chức năng thay đổi mật khẩu

### 3.3.5. Chức năng quên mật khẩu

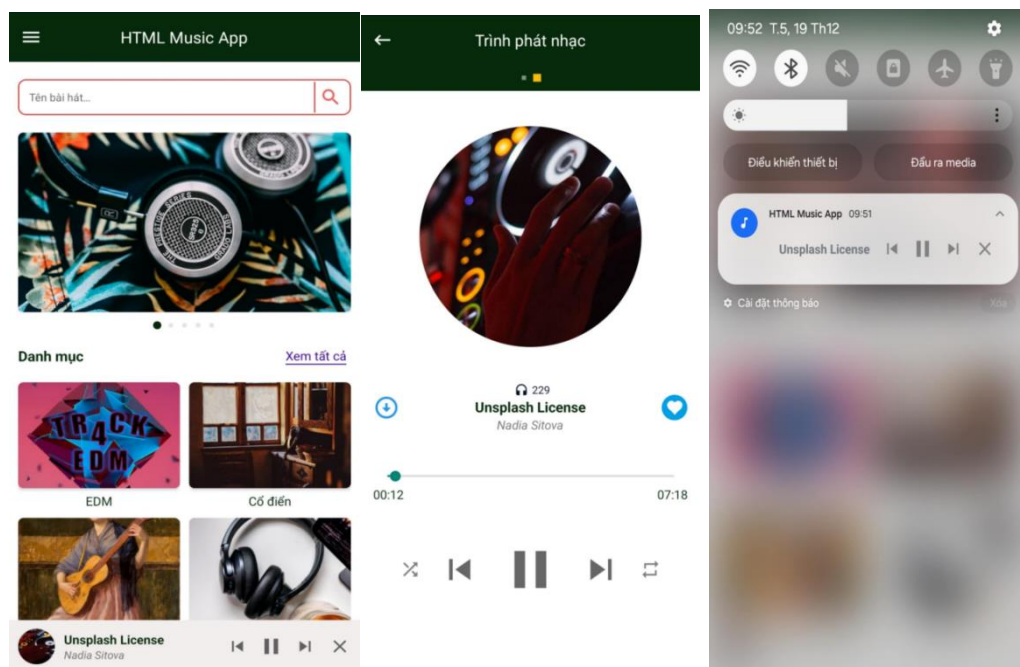
Chức năng quên mật khẩu hỗ trợ người dùng khôi phục mật khẩu khi không nhớ.



Hình 3.10. Giao diện màn hình chức năng quên mật khẩu

### 3.3.6. Chức năng phát nhạc

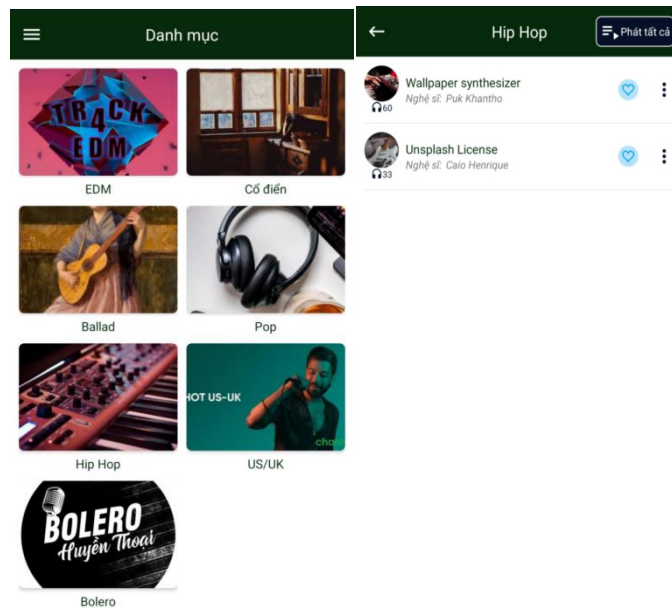
Chức năng phát nhạc cho phép người dùng nghe bài hát ngay trên giao diện ứng dụng, với các tùy chọn điều khiển cơ bản như phát, tạm dừng,...



Hình 3.11. Giao diện màn hình chức năng phát nhạc

### 3.3.7. Chức năng phát nhạc theo thể loại

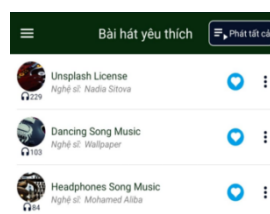
Chức năng này giúp người dùng khám phá và nghe nhạc theo thể loại yêu thích, tạo ra một trải nghiệm nghe nhạc đa dạng và thú vị.



Hình 3.12. Giao diện màn hình chức năng phát nhạc theo thể loại

### 3.3.8. Chức năng quản lý danh sách yêu thích

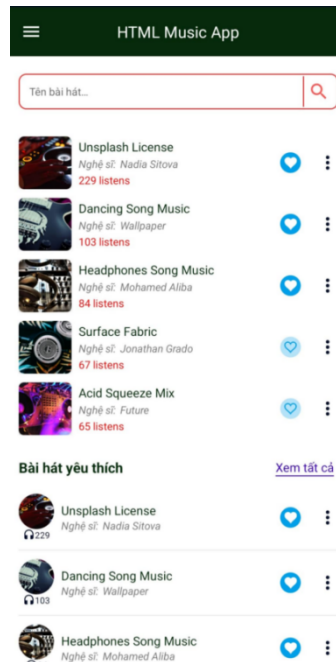
Chức năng quản lý danh sách yêu thích cho phép người dùng lưu lại các bài hát yêu thích để dễ dàng truy cập và nghe lại bất cứ lúc nào.



Hình 3.13. Giao diện màn hình chức năng quản lý danh sách yêu thích

### 3.3.9. Chức năng tìm kiếm bài hát

Chức năng tìm kiếm bài hát giúp người dùng nhanh chóng tìm ra các bài hát theo tên thông qua công cụ tìm kiếm tích hợp trong ứng dụng.



Hình 3.14. Giao diện màn hình chức năng tìm kiếm bài hát

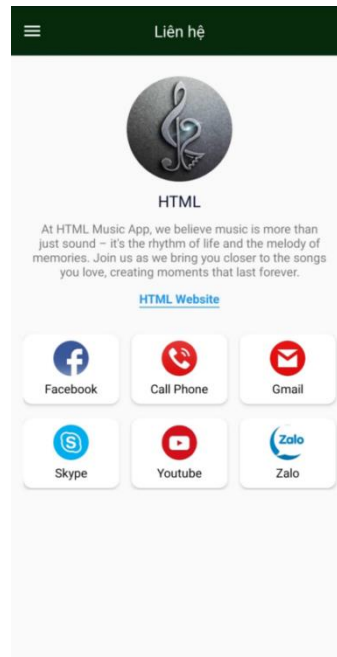
### 3.3.10. Chức năng viết Phản hồi

Chức năng viết phản hồi cho phép người dùng gửi ý kiến hoặc phản hồi về ứng dụng, giúp cải thiện chất lượng và trải nghiệm người dùng.

Hình 3.15. Giao diện màn hình chức năng viết phản hồi

### 3.3.11. Chức năng xem contact

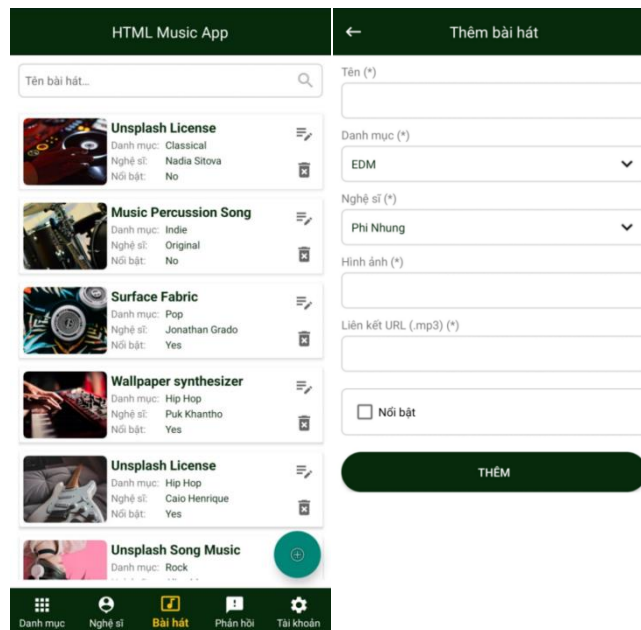
Chức năng xem contact giúp người dùng xem thông tin liên lạc của nhà phát triển.



Hình 3.16. Giao diện màn hình chức năng xem contact

### 3.3.12. Chức năng thêm bài hát

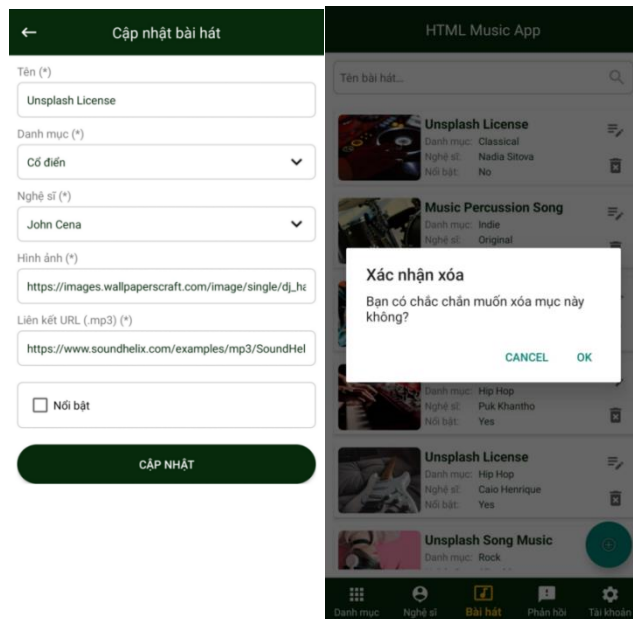
Chức năng thêm bài hát cho phép người dùng thêm các bài hát mới vào ứng dụng, nâng cao khả năng tùy biến và cá nhân hóa trải nghiệm âm nhạc.



Hình 3.17. Giao diện màn hình chức năng thêm bài hát

### 3.3.13. Chức năng quản lý bài hát

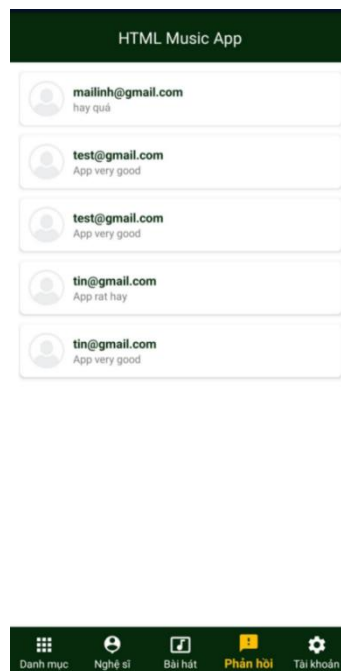
Chức năng quản lý bài hát giúp người dùng chỉnh sửa, xóa các bài hát trong thư viện ứng dụng.



Hình 3.18. Giao diện màn hình chức năng quản lý bài hát

### 3.3.14. Chức năng xem phản hồi

Chức năng xem phản hồi cho phép người quản trị đọc các phản hồi đã gửi từ người dùng về ứng dụng.



Hình 3.19. Giao diện màn hình chức năng xem phản hồi

### 3.4. Kiểm thử hệ thống

- Mục tiêu: Xác minh tính hoạt động chính xác và đầy đủ của một số chức năng trong ứng dụng nghe nhạc trực tuyến.

- Phạm vi:

Kiểm thử các chức năng sau:

- Đăng nhập
  - Đăng ký
  - Phát nhạc theo thể loại, sử dụng các tính năng điều hướng
  - Tìm kiếm bài hát
  - Quản lý bài hát yêu thích
  - Thêm, sửa, xoá bài hát.
- Tiêu chuẩn và tiêu chí kiểm thử
    - Độ chính xác: Xác định xem các chức năng có hoạt động chính xác không.
    - Độ ổn định: Xác định xem các chức năng có ổn định không, không gây ra lỗi hệ thống.
    - Tương tác người dùng: Đảm bảo các chức năng dễ sử dụng và giao diện người dùng thân thiện.
    - Hiệu suất: Kiểm tra hiệu suất của các chức năng và đảm bảo thời gian phản hồi hợp lý.



### 3.5. Kịch bản kiểm thử

#### 3.5.1. Kiểm thử chức năng đăng nhập

Function name: Đăng nhập					
Test requirement: Đăng nhập					
Tester: Hà Thị Mai Linh					
Pass	Fail	N/A	Untested	Number of Test cases	
2	0	0	0	2	
ID	Test Case Title	Preconditions	Test Case Description	Expected Output	Result
[Đăng nhập-1]	Kiểm tra đăng nhập thành công	Người dùng đang ở giao diện đăng nhập	1. Người dùng nhập vào email, mật khẩu Email: mailinh28@gmail.com Mật khẩu: 28282828 Tích role: User 2. Kiểm tra	Hệ thống truy cập vào màn hình chính của ứng dụng	Passed
[Đăng nhập-2]	Kiểm tra đăng nhập không thành công	Người dùng đang ở giao diện đăng nhập	1. Người dùng nhập vào email, mật khẩu không đúng định dạng hoặc chưa được đăng ký 2. Kiểm tra	Hệ thống hiển thị thông báo đăng nhập không thành công, hiển thị lỗi	Passed

Hình 3.20: Kiểm thử chức năng đăng nhập

#### 3.5.2. Kiểm thử chức năng đăng ký

Function name: Đăng ký					
Test requirement: Đăng ký					
Tester: Hà Thị Mai Linh					
Pass	Fail	N/A	Untested	Number of Test cases	
3	0	0	0	3	
ID	Test Case Title	Preconditions	Test Case Description	Expected Output	Result
[Đăng ký-1]	Kiểm tra giao diện đăng ký	Người dùng đang ở màn hình đăng ký	1. Click vào "Đăng ký" ở trang đăng nhập.	Hệ thống hiển thị màn hình đăng ký	Passed
			2. Kiểm tra		
[Đăng ký-2]	Kiểm tra đăng ký không thành công	Người dùng đang ở màn hình đăng ký	1. Nhập thông tin email, password sai định dạng hoặc email đã được đăng ký với hệ thống trước đó (mailinh28@gmail.com).	Hệ thống hiển thị thông báo đăng ký không thành công	Passed
			2. Kiểm tra		
[Đăng ký-3]	Kiểm tra đăng ký thành công	Người dùng đang ở màn hình đăng ký	1. Nhập thông tin email, password đúng định dạng, chọn ký email chưa được sử dụng với hệ thống.	Hệ thống hiển thị thông báo đăng ký thành công, truy cập vào trang chủ ứng dụng	Passed
			2. "Đăng ký".		
			3. Kiểm tra		

Hình 3.21: Kiểm thử chức năng đăng ký

#### 3.5.3. Kiểm thử chức năng phát nhạc theo thể loại

<b>Function name</b>	Phát nhạc theo thể loại				
<b>Test requirement</b>	Phát nhạc theo thể loại				
<b>Tester</b>	Hà Thị Mai Linh				
<b>Pass</b>	<b>Fail</b>	<b>N/A</b>	<b>Untested</b>	<b>Number of Test cases</b>	
2	0	0	0	2	
<b>ID</b>	<b>Test Case Title</b>	<b>Preconditions</b>	<b>Test Case Description</b>	<b>Expected Output</b>	<b>Result</b>
[Phát nhạc theo thể loại-1]	Kiểm tra phát nhạc theo thể loại thành công	Người dùng đã đăng nhập vào hệ thống và đang ở màn hình chính của ứng dụng	1. Người dùng lựa chọn thể loại bất kỳ 2. Người dùng chọn vào "Play all" 3. Kiểm tra	Phát nhạc thành công, danh sách phát bao gồm các bài hát cùng thể loại	Passed
[Phát nhạc theo thể loại-2]	Kiểm tra các tính năng điều hướng thành công	Thực hiện từ [Phát nhạc theo thể loại-1]	1. Người dùng thực hiện các chức năng: chuyển bài trước, chuyển bài tiếp theo, tạm dừng, phát tiếp, mở bài ngẫu nhiên, lặp lại bài hiện tại 2. Kiểm tra	Các chức năng hoạt động đúng theo yêu cầu	Passed

Hình 3.22: Kiểm thử chức năng phát nhạc theo thể loại

### 3.5.4. Kiểm thử chức năng tìm kiếm bài hát

<b>Function name</b>	Tìm kiếm bài hát				
<b>Test requirement</b>	Tìm kiếm bài hát				
<b>Tester</b>	Hà Thị Mai Linh				
<b>Pass</b>	<b>Fail</b>	<b>N/A</b>	<b>Untested</b>	<b>Number of Test cases</b>	
3	0	0	0	3	
<b>ID</b>	<b>Test Case Title</b>	<b>Preconditions</b>	<b>Test Case Description</b>	<b>Expected Output</b>	<b>Result</b>
[Tìm kiếm bài hát-1]	Kiểm tra giao diện tìm kiếm	Người dùng đã đăng nhập vào hệ thống	1. Người dùng chọn biểu tượng Tìm kiếm trên thanh công cụ 2. Kiểm tra	Hệ thống chuyển sang màn hình có mục Tìm kiếm	Passed
[Tìm kiếm bài hát-2]	Kiểm tra danh sách tìm kiếm nhập hợp lệ	Thực hiện từ [Tìm kiếm bài hát-1]	1. Người dùng nhập vào một từ hoặc đầy đủ tên bài hát hợp lệ, có trong hệ thống ("Cha", "Cha và con gái", ...) 2. Kiểm tra	Hệ thống hiển thị danh sách các bài hát có tên chứa các từ ngữ được tìm kiếm	Passed
[Tìm kiếm bài hát-3]	Kiểm tra danh sách tìm kiếm nhập không hợp lệ	Thực hiện từ [Tìm kiếm bài hát-1]	1. Người dùng nhập vào từ khóa không hợp lệ hoặc bài hát không có trong hệ thống 2. Kiểm tra	Hệ thống hiển thị danh sách trống và thông báo không có bài hát nào trong hệ thống	Passed

Hình 3.23: Kiểm thử chức năng tìm kiếm bài hát

### 3.5.5. Kiểm thử chức năng quản lý danh sách yêu thích

<b>Function name</b>	Quản lý danh sách yêu thích				
<b>Test requirement</b>	Quản lý danh sách yêu thích				
<b>Tester</b>	Hà Thị Mai Linh				
<b>Pass</b>	<b>Fail</b>	<b>N/A</b>	<b>Untested</b>	<b>Number of Test cases</b>	
3	0	0	0	3	
<b>ID</b>	<b>Test Case Title</b>	<b>Preconditions</b>	<b>Test Case Description</b>	<b>Expected Output</b>	<b>Result</b>
[Quản lý danh sách yêu thích-1]	Kiểm tra giao diện danh sách yêu thích	Người dùng đã đăng nhập vào hệ thống	1. Người dùng chọn biểu tượng Yêu thích trên thanh công cụ 2. Kiểm tra	Hệ thống chuyển sang màn hình danh sách các bài hát yêu thích	Passed
[Quản lý danh sách yêu thích-2]	Thêm bài hát vào danh sách yêu thích thành công	Người dùng đã đăng nhập vào hệ thống	1. Người dùng chọn vào biểu tượng yêu thích trên bài hát bất kỳ 2. Kiểm tra	Chuyển sang màn hình danh sách yêu thích, bài hát vừa được chọn đã được thêm vào danh sách	Passed
[Quản lý danh sách yêu thích-3]	Xoá bài hát khỏi danh sách yêu thích thành công	Thực hiện từ [Quản lý danh sách yêu thích-1]	1. Người dùng chọn lại vào biểu tượng yêu thích của bài hát bất kỳ trong danh sách yêu thích 2. Kiểm tra	Bài hát được chọn bị xoá khỏi danh sách yêu thích	Passed

Hình 3.24: Kiểm thử chức năng quản lý danh sách yêu thích

### 3.5.6. Kiểm thử chức năng thêm bài hát

<b>Function name</b>	Thêm bài hát				
<b>Test requirement</b>	Thêm bài hát				
<b>Tester</b>	Hà Thị Mai Linh				
<b>Pass</b>	<b>Fail</b>	<b>N/A</b>	<b>Untested</b>	<b>Number of Test cases</b>	
3	0	0	0	3	
<b>ID</b>	<b>Test Case Title</b>	<b>Preconditions</b>	<b>Test Case Description</b>	<b>Expected Output</b>	<b>Result</b>
[Thêm bài hát-1]	Kiểm tra giao diện màn hình thêm bài hát	Người dùng đã đăng nhập vào hệ thống với vai trò admin	1. Người quản trị chọn biểu tượng "+" ở góc màn hình chính 2. Kiểm tra	Hệ thống hiển thị màn hình thêm bài hát	Passed
[Thêm bài hát-2]	Thêm bài hát mới vào hệ thống thành công	Thực hiện từ [Thêm bài hát-1]	1. Người quản trị nhập vào thông tin bài hát: Tên bài hát, tên tác giả, link url của bài hát, link hình ảnh, tích chọn thể loại cho bài hát đầy đủ và hợp lệ 2. Click "Thêm" 3. Kiểm tra	Bài hát được thêm vào hệ thống thành công, màn hình chính của admin hiển thị danh sách có bài hát mới	Passed
[Thêm bài hát-3]	Thêm bài hát mới vào hệ thống không thành công	Thực hiện từ [Thêm bài hát-1]	1. Người quản trị nhập vào thông tin bài hát: Tên bài hát, tên tác giả, link url của bài hát, link hình ảnh, tích chọn thể loại cho bài hát không đầy đủ hoặc không hợp lệ 2. Click "Thêm" 3. Kiểm tra	Hệ thống hiển thị thông báo không được để trống hoặc thông tin không hợp lệ	Passed

Hình 3.25: Kiểm thử chức năng thêm bài hát

### 3.5.7. Kiểm thử chức năng sửa bài hát

Function name	Sửa bài hát				
Test requirement	Sửa bài hát				
Tester	Hà Thị Mai Linh				
Pass	Fail	N/A	Untested	Number of Test cases	
3	0	0	0	3	
ID	Test Case Title	Preconditions	Test Case Description	Expected Output	Result
[Sửa bài hát-1]	Kiểm tra giao diện màn hình sửa bài hát	Người dùng đã đăng nhập vào hệ thống với vai trò admin	1. Người quản trị chọn biểu tượng Sửa trên bài hát cần sửa 2. Kiểm tra	Hệ thống hiển thị màn hình sửa bài hát	Passed
[Sửa bài hát-2]	Sửa thông tin bài hát thành công	Thực hiện từ [Sửa bài hát-1]	1. Người dùng sửa thông tin bất kỳ của bài hát: Tên bài hát, tên tác giả, link url của bài hát, link hình ảnh, tích chọn thể loại cho bài hát hợp lệ 2. Click "Cập nhật" 3. Kiểm tra	Bài hát hiển thị thông tin mới sau khi được sửa	Passed
[Sửa bài hát-3]	Sửa thông tin bài hát không thành công	Thực hiện từ [Sửa bài hát-1]	1. Người dùng sửa thông tin bất kỳ của bài hát: Tên bài hát, tên tác giả, link url của bài hát, link hình ảnh, tích chọn thể loại cho bài hát không đầy đủ hoặc không hợp lệ 2. Click "Cập nhật" 3. Kiểm tra	Hệ thống hiển thị thông báo không được để trống hoặc thông tin không hợp lệ	Passed

Hình 3.26: Kiểm thử chức năng sửa bài hát

### 3.5.8. Kiểm thử chức năng xóa bài hát

Function name	Xóa bài hát				
Test requirement	Xóa bài hát				
Tester	Hà Thị Mai Linh				
Pass	Fail	N/A	Untested	Number of Test cases	
3	0	0	0	3	
ID	Test Case Title	Preconditions	Test Case Description	Expected Output	Result
[Xóa bài hát-1]	Kiểm tra chức năng xóa bài hát	Người dùng đã đăng nhập vào hệ thống với vai trò admin	1. Người quản trị chọn biểu tượng Xóa trên bài hát cần sửa 2. Kiểm tra	Hệ thống hiển thị pop-up bao gồm: - Thông báo "Bạn chắc chắn muốn xóa bài hát?" - Nút "OK" và nút "Hủy"	Passed
[Xóa bài hát-2]	Kiểm tra nút "OK"	Thực hiện từ [Xóa bài hát-1]	1. Click vào nút "OK" trên pop-up 2. Kiểm tra	Bài hát được chọn bị xóa khỏi hệ thống và hiển thị danh sách sau khi đã xóa bài hát	Passed
[Xóa bài hát-3]	Kiểm tra nút "Hủy"	Thực hiện từ [Xóa bài hát-1]	1. Click vào nút "Hủy" trên pop-up 2. Kiểm tra	Hệ thống hủy bỏ lệnh xóa bài hát được chọn và hiển thị màn hình danh sách	Passed

Hình 3.27: Kiểm thử chức năng xóa bài hát

### 3.6. Kết luận kiểm thử

- Quá trình kiểm thử đã được tiến hành trên các chức năng chính của hệ thống, bao gồm: Đăng ký, Đăng nhập, Phát nhạc theo thể loại, Tìm kiếm bài hát, Quản lý danh sách yêu thích, Thêm bài hát, Sửa bài hát và Xoá bài hát.
- Kết quả kiểm thử chi tiết:
  - Tổng cộng có 22 trường hợp kiểm thử (test cases) được thực hiện trên các chức năng khác nhau.
  - Kết quả cho thấy tất cả các trường hợp đều đạt (Pass), không có lỗi (Fail).
- Nhận xét và kết luận
  - Tất cả các chức năng đã được thực hiện đúng theo yêu cầu đề ra.
  - Các test cases đạt tiêu chuẩn, đảm bảo hệ thống hoạt động ổn định.
  - Đặc biệt, các tính năng liên quan đến việc quản lý bài hát như Thêm, Sửa và Xoá bài hát được xác nhận không phát sinh lỗi trong các trường hợp kiểm thử.

STT	Function name	Number of Test cases	Pass	Fail
1	Đăng ký	3	3	0
2	Đăng nhập	2	2	0
3	Phát nhạc theo thể loại	2	2	0
4	Tìm kiếm bài hát	3	3	0
5	Quản lý danh sách yêu thích	3	3	0
6	Thêm bài hát	3	3	0
7	Sửa bài hát	3	3	0
8	Xoá bài hát	3	3	0

Hình 3.28: Kết quả kiểm thử

## KẾT LUẬN VÀ KIẾN NGHỊ

### ❖ Kết quả đạt được

Sau quá trình nghiên cứu và thực hiện đồ án, em đã đạt được các kết quả sau:

- Nắm vững hơn về quy trình phát triển ứng dụng di động và các giai đoạn triển khai một dự án công nghệ.
- Hiểu và sử dụng thành thạo công cụ Android Studio để thiết kế và phát triển ứng dụng.
- Tiến hành khảo sát, phân tích yêu cầu của hệ thống, từ đó hiện thực hóa các chức năng theo đúng đặc tả kỹ thuật.
- Vận dụng hiệu quả các kiến thức đã học để phân tích và đặc tả hệ thống.
- Thiết kế cơ sở dữ liệu sử dụng Firebase Realtime Database, đảm bảo khả năng quản lý và lưu trữ dữ liệu hiệu quả.
- Xây dựng các chức năng quản lý tài khoản và phân quyền người dùng thông qua Firebase Authentication.
- Sử dụng ngôn ngữ lập trình Kotlin, mô hình MVC, Media Player, và các thư viện liên quan để hoàn thiện ứng dụng.

### ❖ Hướng phát triển trong tương lai

Trong tương lai, em dự kiến tiếp tục phát triển và nâng cấp ứng dụng với các hướng sau:

- Tích hợp chức năng cho phép người dùng tạo danh sách phát cá nhân.
- Nâng cao trải nghiệm người dùng bằng cách cải thiện hiệu năng và tối ưu hóa các tính năng của ứng dụng.
- Thiết kế lại giao diện người dùng để thân thiện, hiện đại và đáp ứng tốt hơn nhu cầu của người dùng.

Qua quá trình thực hiện đồ án, em đã học hỏi được nhiều kiến thức và kỹ năng quý báu. Tuy nhiên, em nhận thấy sản phẩm vẫn còn nhiều hạn chế cần khắc phục và hoàn thiện. Em rất mong nhận được sự đóng góp ý kiến và nhận xét từ quý thầy cô và nhà trường để có cơ hội phát triển, nâng cao chất lượng sản phẩm cũng như năng lực chuyên môn của bản thân trong thời gian tới.



## TÀI LIỆU THAM KHẢO

- [1] Tổ HTTT Đại học Công Nghiệp Hà Nội (2011) – *Giáo trình phân tích thiết kế hệ thống*, nhà xuất bản Giáo dục Việt Nam.
- [2] Nguyễn Thành Công (2018) – *Lập trình ứng dụng di động với Kotlin*, Nhà xuất bản Khoa học và Kỹ thuật.
- [3] Hồng Anh, Lê Minh Tân (2020) – *Công nghệ phát triển ứng dụng di động trên nền tảng Android*, Nhà xuất bản Đại học Sư phạm Kỹ thuật TP.HCM.
- [4] Nguyễn Bá Nghiễn (2022) - *Phát triển ứng dụng cho thiết bị di động*, nhà xuất bản Thống kê.
- [5] <https://developer.android.com/training/basics/firstapp> [Online].
- [6] <https://firebase.google.com/docs> [Online].
- [7] <https://stackoverflow.com> [Online].
- [8] <https://kotlinlang.org/docs/home> [Online]. Tài liệu chính thức về Kotlin từ website chính thức của Kotlin.