|  |  |
| --- | --- |
|  |  |
| HÀ QUANG HƯNG | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| **BÁO CÁO ĐỒ ÁN TỐT NGHIỆP** |
| **NGÀNH CÔNG NGHỆ THÔNG TIN** |
| **TÊN ĐỀ TÀI**  **Xây dựng ứng dụng và phát triển ứng dụng nghe nhạc trên nền tảng Android** |
|  |
|  |
| **CBHD: *TS. Đặng Trọng Hợp*** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên: *Hà Quang Hưng*** |
| **Mã số sinh viên: *2018603268*** |
|  |
|  |
|  |
| Hà Nội – Năm 2022 |

MỤC LỤC

[Bảng danh mục hình ảnh 4](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813662)

[CHƯƠNG I: GIỚI THIỆU TỔNG QUAN 7](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813663)

[**1.1 Lý do chọn đề tài 7**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813664)

[**1.2 Mục tiêu của đề tài 7**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813665)

[**1.3 Giới thiệu công nghệ sử dụng 7**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813666)

[1.3.1 Mô hình phát triển MVVM 8](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813667)

[1.3.2 Ngôn ngữ Kotlin 9](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813668)

[1.3.3 Hệ điều hành Android 10](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813669)

[**1.4 Những mặt còn hạn chế: 18**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813670)

[1.4.1 Ngôn ngữ kotlin 18](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813671)

[1.4.2 Nền tảng android 18](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813672)

[CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 19](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813673)

[**2.1 Khảo sát đề tài 19**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813674)

[2.1.1 Phương pháp: Phỏng vấn 19](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813675)

[2.1.2 Mẫu phiếu phỏng vấn: 21](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813676)

[**2.2 Yêu cầu hệ thông 22**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813677)

[2.2.1 Yêu cầu người dùng 22](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813678)

[2.2.2 Yêu cầu hệ thống 23](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813679)

[**2.3 Mô tả chi tiết Use Case 23**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813680)

[2.3.1 Xác định chức năng 23](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813681)

[2.3.2 Sơ đồ UseCase 24](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813682)

[2.2.3 Đặc tả UseCase 26](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813683)

[2.3.4 Các biểu đồ trình tự 34](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813684)

[**2.5 Thiết kế biểu đồ lớp 46**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813685)

[**2.6 Cơ sở dữ liệu 47**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813686)

[**2.7 Thiết kế giao diện 48**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813687)

[CHƯƠNG III: CÀI ĐẶT THỬ NGHIỆM 52](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813688)

[**3.1 Giới thiệu môi trường phát triển ứng dụng tích hợp Android Studio 52**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813689)

[**3.2 Các thành phần chính của Android studio 53**](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813690)

[3.2.1 Quản lý file và tệp 53](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813691)

[3.2.2 Hệ thống android build 54](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813692)

[3.2.3 Bộ nhớ và màn hình GPU 55](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813693)

[3.2.4 Truy cập file dữ liệu: 56](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813694)

[KẾT LUẬN 58](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813695)

[Tài liệu tham khảo 59](file:///C:\Users\HurioFly\Downloads\BaoCaoDoAnTotNghiep_TranHaiSon_k12.docx#_Toc71813696)

# Bảng danh mục hình ảnh

|  |  |
| --- | --- |
| **Số thứ tự** | **Tên hình ảnh** |
| Hình 1.1 | Mô tả hoạt động mô hình MVVM |
| Hình 1.2 | Thanh thông báo được trượt xuống từ cạnh trên màn hình. |
| Hình 1.3 | Hình ảnh logo của Google Play |
| Hình 1.4 | Sơ đồ kiến trúc hệ thống android |
| Hình 1.5 | Cấp phát quyền cho ứng dụng hoạt động |
| Hình 2.1 | Biểu đồ UseCase chính |
| Hình 2.2 | Biểu đồ UseCase xem danh sách bài hát |
| Hình 2.3 | Biểu đồ UseCase hiển thị chi thiết bài hát |
| Hình 2.4 | Biểu đồ trình tự UseCase Xem danh sách bài hát |
| Hình 2.5 | Biểu đồ trình tự UseCase Hiển thị chi tiết bài hát |
| Hình 2.6 | Biểu đồ trình tự UseCase Tìm kiếm |
| Hình 2.7 | Biểu đồ trình tự UseCase Play |
| Hình 2.8 | Biểu đồ trình tự UseCase Pause |
| Hình 2.9 | Biểu đồ trình tự UseCase Next |
| Hình 2.10 | Biểu đồ trình tự UseCase Previous |
| Hình 2.11 | Biểu đồ trình tự UseCase Random |
| Hình 2.12 | Biểu đồ trình tự UseCase Repeat |
| Hình 2.13 | Biểu đồ trình tự UseCase SeekBar |
| Hình 2.14 | Biểu đồ trình tự UseCase Chọn bài hát yêu thích |
| Hình 2.15 | Biểu đồ trình tự UseCase Hẹn giờ |
| Hình 2.16 | Biểu đồ các lớp và mối quan hệ |
| Hình 2.17 | Giao diện chính |
| Hình 2.18 | Giao diện khi chọn bài hát |
| Hình 2.19 | Giao diện khi xem chi tiết bài hát |
| Hình 2.20 | Giao diện khi nhấn drawer để chọn màn |
| Hình 3.1 | Quản lý file và tệp trên android studio |
| Hình 3.2 | Màn hình hiển thị thông số GPU của thiết bị |
| Hình 3.3 | Kiểm tra cấu hình cài đặt Android studio |

**LỜI MỞ ĐẦU**

Trong những năm gần đây, nền tảng công nghệ thông tin phát triển mạnh mẽ, cùng với đó là sự ra đời của smartphone, máy tính…các thiết bị này không chỉ ngày càng phổ biến mà còn mang đến cho con người hàng loạt khả năng mới trên mọi lĩnh vực như: trao đổi thông tin, làm việc di động, giải trí mọi lúc mọi nơi... Trong đó điện thoại thực sự đã giúp thay đổi toàn diện cuộc sống theo hướng tích cực hơn.

Điện thoại di động đã là một bước tiến trong việc liên lạc, tuy nhiên với những chiếc điện thoại di động thời xưa, con người chỉ có thể truyền và nhận những thông điệp đơn giản với âm thanh và tin nhắn ký tự. Ngày nay với smartphone, dù đang ở bất cứ lúc nào hay ở bất cứ nơi đâu, chỉ cần một vài thao tác là bạn đã có vô số lựa chọn để kết nối với người thân, từ gọi video, gửi tin nhắn có hình, chat tức thời hay đơn giản là cập nhật trạng thái/hình ảnh trên Facebook, Instagram, Twitter,... Thay vì nhắn tin và gọi điện thu phí như trước đây, người dùng hiện nay có thể thỏa mãn với các ứng dụng liên lạc miễn phí trên di động bởi sự phát triển nhanh chóng của 4G và wifi.

Nhận thức được tầm quan trọng của nhu cầu giải trí trên điện thoại di động. Em đã chọn đề tài “Xây dựng ứng dụng nghe nhạc” để nghiên cứu và viết báo cáo, qua đó với sự giúp đỡ của thầy Nguyễn Bá Nghiễn và các anh chị cùng Công ty, em hy vọng sẽ hiểu thêm quá trình thiết kế, xây dựng, bảo trì, và hoạt động của ứng dụng từ những bước cơ bản nhất.

Qua bài báo cáo này em xin được gửi lời cảm ơn chân thành của thầy Nguyễn Bá Nghiễn và các anh chị cùng công ty đã tận tình hướng dẫn và giúp đỡ để em có thể hoàn thành tốt bài báo cáo trong kì thực tập này.

**Em xin chân thành cảm ơn!**

# CHƯƠNG I: GIỚI THIỆU TỔNG QUAN

**1.1 Lý do chọn đề tài**

Xã hội hiện đại đưa cuộc sống của chúng ta lên một mức sống mới, văn minh, nhộn nhịp, nhưng cũng kéo theo nhiều điều tiêu cực, nổi bật trong đó là stress công việc, mệt mỏi với cuộc sống hàng ngày.

Trong khi đó, âm nhạc, được ví như ngôn ngữ của vũ trụ, Âm nhạc là loại thần dược của tâm hồn và sức khỏe của con người, giúp cho chúng ta tỉnh táo và năng động khi được nghe những bản nhạc có tiết tấu nhanh. Ngày nay âm nhạc trong cuộc sống giúp cho người bệnh lấy lại trạng thái tinh thần sau những tổn thương và những cú sốc về tình cảm.

Và cũng vì bản thân rất thích nghe nhạc mỗi khi stress công việc, học tập, cũng như rảnh rỗi. Vì vậy em đã quyết định chọn đề tài “ Xây dựng ứng dụng nghe nhạc” trên nền tảng Android để phát triển, sản phẩm này sẽ mang đến nguồn cảm giác và học tập vô hạn cho chúng ta, giúp chúng ta lên dây cót tinh thần khi cuộc sống mệt mỏi, sau một ngày dài lao động.

**1.2 Mục tiêu của đề tài**

Xây dựng được ứng dụng nghe nhạc với giao diện đẹp mắt dễ dàng sử dụng và đầy đủ chức năng cho người dùng

**1.3 Giới thiệu công nghệ sử dụng**

Đối với việc phát triển ứng dụng, nền tảng android là một nền tảng tốt với rất nhiều ưu điểm, kết hợp với mô hình phát triển ứng dụng nổi tiếng MVVM.

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

**1.3.1 Mô hình phát triển MVVM**

MVVM là viết tắt của Model - View - ViewModel, ở đó view (tức giao diện người dùng) sẽ được cập nhật bởi ViewModel và việc xử lý Logic hoặc trình bày dữ liệu sẽ do Model đảm nhận. Mô hình này khá giống với MVC (Model - View - Controller), sự khác biệt duy nhất là nẳm ở cách xây dựng nên C (Controller) của MVC và VM (ViewModel) của MVVM.



*Hình 1.1:**Mô tả hoạt động mô hình MVVM*

**Model**: Trong MVVM thì model sẽ thể hiện cho dữ liệu + trạng thái + các logic của đối tượng. Nó không có ràng buộc với **View** hoặc **Controller** vì vậy có thể được xử dụng lại dễ dàng

**View**: Liên kết các biến quan sát và hành động bởi **View Model**. Quan hệ giữa **View Model** và **View** là 1-n, nghĩa là nhiều **View** có thể liên kết với 1 **ViewModel**

**ViewModel**: Chứa các model và chuẩn bị các dữ liệu quan sát cho View. Nó cung cấp các móc để truyền dữ liệu từ View sang Model. Một điều cần phải ghi nhớ là ***ViewModel sẽ không ràng buộc vào View***.

MVVM có thể nói là mô hình kiến trúc được rất nhiều các cư dân trong cộng đồng ưa chuộng. Điểm tinh hoa của kiến trúc này là ở ViewModel, mặc dù rất giống với Presenter trong MVP tuy nhiên có 2 điều làm nên tên tuổi của kiến trúc này đó là:

ViewModel không hề biết gì về View, một ViewModel có thể được sử dụng cho nhiều View (one-to-many). ViewModel sử dụng Observer design pattern để liên lạc với View (thường được gọi là binding data, có thể là 1 chiều hoặc 2 chiều tùy nhu cầu ứng dụng). Chính đặc điểm này MVVM thường được phối hợp với các thư viện hỗ trợ Reactive Programming hay Event/Data Stream, đây là triết lý lập trình hiện đại và hiệu quả phát triển rất mạnh trong những năm gần đây.

**1.3.2 Ngôn ngữ Kotlin**

Kotlin là ngôn ngữ lập trình được giới thiệu bởi JetBrains, là một công ty phát triển phần mềm, họ tạo ra một IDE nổi tiếng để lập trình Java có tên là Intellij IDEA. Kotlin là một ngôn ngữ mạnh mẽ chạy trên JVM. Trong năm 2017, Google đã công bố Kotlin là ngôn ngữ chính thức để phát triển Android. Nó là một ngôn ngữ lập trình nguồn mở hỗ trợ cả hai phướng pháp lập trình hướng đối tượng (object-oriented programming) cũng như lập trình hướng thủ tục (functional programming). Nội dung của tutorial này sẽ được chia thành các bài khác nhau với các chủ đề liên quan và kèm theo ví dụ đơn giản và hữu ích.

Kotlin là một ngôn ngữ lập trình mã nguồn mở mới giống như Java, JavaScript, PHP,… Đây là ngôn ngữ bậc cao hỗ trợ cả phương pháp lập trình hướng đối tượng và hướng thủ tục. Hiện tại, Kotlin đang target đến Java và JavaScript. Nó chạy trên JVM (Java Virtual Machine).

Kotlin bị ảnh hưởng bởi các ngôn ngữ lập trình khác như Java, Scala, Groovy, Gosu,… Cú pháp (syntax) của Kotlin có thể không hoàn toàn giống với JAVA, tuy nhiên, bên trong Kotlin phụ thuộc vào các thư viện Java Class để tạo ra kết quả tuyệt vời cho các lập trình viên.

**Một vài ưu điểm của ngôn ngữ kotlin:**

* Ngôn ngữ dễ học.
* Code ngắn gọn.
* Thời gian chạy nhỏ và hiệu suất tốt hơn.
* Khả năng tương tác tốt.
* Android Studio có hỗ trợ
* 100% khả năng tương tác với java.

**1.3.3 Hệ điều hành Android**

**1.3.3.1 Tổng quan hệ thống Android**

Android là một hệ điều hành mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt sau đó 2 năm (2007) cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào năm 2008.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

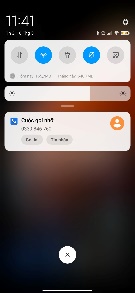
Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian OS vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo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

**1.3.3.2 Giao diện của Android**

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

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

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



*Hình 1.2* *Thanh thông báo được trượt xuống từ cạnh trên màn hình.*

**1.3.3.3 Kho ứng dụng Android**

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



*Hình 1.3* *Hình ảnh logo của Google Play*

**1.3.3.4 Cộng đồng mã nguồn mở**

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

**1.3.3.5 Tính bảo mật và riêng tư**

Các ứng dụng Android chạy trong một "hộp cát", là một khu vực riêng rẽ với hệ thống và không được tiếp cận đến phần còn lại của tài nguyên hệ thống, trừ khi nó được người dùng trao quyền truy cập một cách công khai khi cài đặt. Trước khi cài đặt ứng dụng, Cửa hàng Play sẽ hiển thị tất cả các quyền mà ứng dụng đòi hỏi: ví dụ như một trò chơi cần phải kích hoạt bộ rung hoặc lưu dữ liệu vào thẻ nhớ SD, nhưng nó không nên cần quyền đọc tin nhắn SMS hoặc tiếp cận danh bạ điện thoại. Sau khi xem xét các quyền này, người dùng có thể chọn đồng ý hoặc từ chối chúng, ứng dụng chỉ được cài đặt khi người dùng đồng ý.

Hệ thống hộp cát và hỏi quyền làm giảm bớt ảnh hưởng của lỗi bảo mật hoặc lỗi chương trình có trong ứng dụng, nhưng sự bối rối của lập trình viên và tài liệu hướng dẫn còn hạn chế đã dẫn tới những ứng dụng hay đòi hỏi những quyền không cần thiết, do đó làm giảm đi hiệu quả của hệ thống này. Một số công ty bảo mật, như Lookout Mobile Security, AVG Technologies, và McAfee, đã phát hành những phần mềm diệt virus cho các thiết bị Android. Phần mềm này không có hiệu quả vì cơ chế hộp cát vẫn áp dụng vào các ứng dụng này, do vậy làm hạn chế khả năng quét sâu vào hệ thống để tìm nguy cơ.

Điện thoại thông minh Android có khả năng báo cáo vị trí của điểm truy cập Wi-Fi, phát hiện ra việc di chuyển của người dùng điện thoại, để xây dựng những cơ sở dữ liệu có chứa vị trí của hàng trăm triệu điểm truy cập. Những cơ sở dữ liệu này tạo nên một bản đồ điện tử để tìm vị trí điện thoại thông minh, cho phép chúng chạy các ứng dụng như Foursquare, Google Latitude, Facebook Places, và gửi những đoạn quảng cáo dựa trên vị trí. Phần mềm theo dõi của bên thứ ba như TaintDroid, một dự án nghiên cứu trong trường đại học, đôi khi có thể biết được khi nào thông tin cá nhân bị gửi đi từ ứng dụng đến các máy chủ đặt ở xa.



*Hình 1.4* *Cấp phát quyền cho ứng dụng hoạt động*

**1.4 Những mặt còn hạn chế:**

**1.4.1 Ngôn ngữ kotlin**

* **Khai báo namespace:** Kotlin cho phép các nhà phát triển khai báo các chức năng ở cấp cao nhất. Tuy nhiên, khi khai báo cùng một function ở nhiều nơi trong ứng dụng thì chúng ta sẽ khó để biết được function nào đang được gọi.
* **Không có khai báo tĩnh “static”**: Kotlin không có “static” modifier như Java, điều này có thể gây ra một số vấn đề cho nhà phát triển Java.

**1.4.2 Nền tảng android**

* Tuy nên tảng android đang chiếm thị phần lớn nhất nhưng IOS cũng đang là một đối thủ cạnh tranh khốc liệt.
* Người dùng android đa phần là người dùng phổ thông, ít chi tiêu cho ứng dụng của họ hơn là IOS

**CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1 Khảo sát đề tài**

Mục tiêu:

* Tìm hiểu nhu cầu giải trí bằng âm nhạc của các bạn trẻ
* Đối tượng khảo sát: Các đối tượng khảo sát bao gồm: Các bạn học sinh, sinh viên.

**2.1.1 Phương pháp: Phỏng vấn**

|  |  |
| --- | --- |
| Kế hoạch phỏng vấn | |
| Người được hỏi: Nguyễn Trung Minh Hiếu | Người phỏng vấn: Trần Hải Sơn |
| Địa chỉ: Trường đại học công nghiệp Hà nội | Thời gian hẹn: Chủ nhật (10/01/2021)  Thời điểm bắt đầu: 8h  Thời điểm kết thúc: 11h |
| Đối tượng:  Đối tượng được hỏi là: Sinh viên lớp CNTT1-K12  Cần thu thập dữ liệu: Quản lý điểm cần những gì? Mong muốn hệ thông ứng dụng chơi nhạc như thế nào? | Các yêu cầu đòi hỏi:  Thường xuyên sử dụng điện thoại để giải trí |
| Chương trình:   * Giới thiệu * Tổng quan về dự án * Tổng quan về phỏng vấn * Chủ đề sẽ đề cập. Xin phép được ghi âm * Chủ đề 1: Bạn đã dùng điện thoại được lâu chưa?   ->Câu trả lời   * Chủ đề 2: Bạn thường sử dụng hãng điện thoại gì?   ->Câu trả lời   * Chủ đề 3: Bạn thường sử dụng hệ điều hành nào?   ->Câu trả lời   * Chủ đề 4: Bạn có hay nghe nhạc để giải trí trong cuộc sống không?   ->Câu trả lời   * Chủ đề 5: Bạn thường nghe nhạc qua thiết bị gì?   ->Câu trả lời   * Chủ đề 6: Bạn thường chú ý đến những thể thoại nhạc gì?   ->Câu trả lời   * Chủ đề 7: Nếu có một ứng dụng nghe nhạc, bạn mong muốn ứng dụng mang đến những tiêu chí gì?   ->Câu trả lời   * Chủ đề 8: Theo bạn ứng dụng nên giúp bạn giải quyết những vấn đề gì?   ->Câu trả lời   * Chủ đề 9: Bạn có thể nêu một số gợi ý về giao diện ứng dụng bạn mong muốn?   ->Câu trả lời   * Chủ đề 10: Bạn có thể nếu một số gợi ý về chức năng ứng dụng?   ->Câu trả lời   * Tổng hợp các nội dung chính và ý kiến của người được hỏi   Kết thúc (thỏa thuận) | Ước lượng thời gian  1 phút  2 phút  1 phút  5 phút  1 phút  1 phút  1 phút  1 phút  3 phút  3 phút  3 phút  3 phút  2 phút  5 phút  5 phút |

**2.1.2 Mẫu phiếu phỏng vấn:**

* Sinh viên đại học

|  |  |
| --- | --- |
| **PHIẾU PHỎNG VẤN** | |
| Dự án: Xây dựng ứng dụng chơi nhạc | |
| Người được hỏi: Sinh viên Nguyễn Trung Minh Hiếu, năm cuối Đại học công nghiệp Hà nội | Người hỏi: Trần Hải Sơn  Ngày: Chủ nhật( ngày 10/01/2021) |
| **Câu hỏi:** | **Ghi chú:** |
| Câu 1: Bạn đã dùng điện thoại được lâu chưa? | Trả lời: Mình đã sử dụng điện thoại từ năm 2014 |
| Câu 2: Bạn thường sử dụng hãng điện thoại gì? | Trả lời: Mình thường sử dụng điện thoại Samsung, LG, và sony |
| Câu 3: Bạn thường sử dụng hệ điều hành nào? | Trả lời: Mình sử dụng chủ yếu là android do nó đơn giản, thân thiện |
| Câu 4: Bạn có hay nghe nhạc để giải trí trong cuộc sống không? | Trả lời: Mình rất thích nghe nhạc để giải trí. |
| Câu 5: Bạn thường nghe nhạc qua thiết bị gì? | Trả lời: Mình thường nghe nhạc qua các app nghe nhạc trên điện thoại như zing mp3 |
| Câu 6: Bạn thường chú ý đến những thể thoại nhạc gì? | Trả lời: Mình thường nghe nhạc trẻ Việt Nam, Nhạc rap và thỉnh thoảng là nhạc buồn. |
| Câu 7: Nếu có một ứng dụng nghe nhạc, bạn mong muốn ứng dụng mang đến những tiêu chí gì? | Trả lời: Nếu có một ứng dụng như vậy thì thật tuyệt vời, mình muốn nó có thể dễ dàng sử dụng, tiện lợi nghe nhạc bất cứ đâu. |
| Câu 8: Theo bạn ứng dụng nên giúp bạn giải quyết những vấn đề gì? | Trả lời: Muốn muốn ứng dụng giúp mình tìm nhạc trong điện thoại, bật nhạc ngay cả khi màn hình tắt, dùng được cả tai nghe để nghe nhạc. |
| Câu 9: Bạn có thể nêu một số gợi ý về giao diện ứng dụng bạn mong muốn? | Trả lời: Mình muốn giao diện ứng dụng có màu sắc hài hòa, và dễ sử dụng cho mọi người. |
| Câu 10: Bạn có thể nếu một số gợi ý về chức năng ứng dụng? | Trả lời: Một số chức năng mình nghĩ là hữu ích: Tìm bài hát, lọc ra theo danh sách, next, previous, pause, play, ramdom,…. |
| Đánh giá chung: Thông tin quản lý cung cấp khá đầy đủ. Cuộc trò chuyện diễn ra vui vẻ, thời gian hợp lý. | |

**2.2 Yêu cầu hệ thông**

Từ khảo sát người dùng, là những người sẽ trực tiếp sử dụng hệ thông, Nhằm hướng đến một sản phẩm có chất lượng, yêu cầu ứng dụng cần đạt được nhưng tiêu chí tiêu chuẩn của một ứng dụng android.

* + 1. **Yêu cầu người dùng**
* Giao diện thân thiệt, bắt mắt.
* Dễ dàng sử dụng, thao tác.
* Chức năng hoạt động ổn định, đúng như mong muốn.
  + 1. **Yêu cầu hệ thống**
* Nhẹ, mượt mà, không giật.
* Tiết kiệm tài nguyên hệ thống.
* Ước tính cấu hình tối thiểu:
* Màn hình: 4.7 inch trở lên.
* Ram: 1GB trở lên.
* Chip: 1.5 Ghz trở lên
* Phiên bản android tối thiểu.
* Andoird API Level (SDK) 21 trở lên.
* Phiên bản Android 4.1 trở lên.

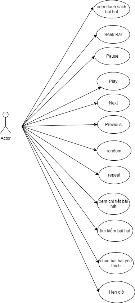
**2.3 Mô tả chi tiết Use Case**

**2.3.1 Xác định chức năng**

1. Đăng ký, đăng nhập
2. Xem, sửa thông tin tài khoản
3. Xem danh sách món
4. Giỏ hàng
5. Thêm món vào giỏ hàng
6. Đặt hàng
7. Thêm món vào danh sách yêu thích
8. Xem lịch sử đơn hàng

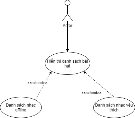
### 2.3.2 Sơ đồ UseCase

**2.3.2.1 Biểu đồ UseCase tổng quát**

****

*Hinh 2.1 Biểu đồ UseCase chính*

**2.3.2.2 Biểu đồ UseCase chức năng “Xem danh sách bài hát”**

****

*Hinh 2.2 Biểu đồ UseCase “Xem danh sách bài hát”*

**2.3.2.3 Biểu đồ UseCase chức năng “Hiển thị chi tiết bài hát”**

*Hinh 2.3 Biểu đồ UseCase “Hiển thị chi tiết bài hát*

### 2.2.3 Đặc tả UseCase

**2.2.3.1 Mô tả use case “Xem danh sách bài hát”**

Mô tả tóm tắt:

* Tên usecase : Xem danh sách bài hát
* Mục đích: Use case này cho phép người dùng xem danh sách bài hát trong máy điện thoại, và những bài hát yêu thích do mình lựa chọn, hoặc danh sách bài hát online
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
      * 1. Use case này bắt đầu khi Khách hàng mở ứng dụng lên
        2. Ứng dụng sẽ lấy toàn bộ nhạc từ SongManager và tiến hành hiển thị danh sách lên activity dưới dạng fragment
        3. Hiển thị bài hát dưới tên bài hát và ca sĩ
    - Luồng rẽ nhánh:

1. Danh sách rỗng: Nếu danh sách chưa có bài nhạc nào thì sẽ hiện thị thông báo danh sách rỗng.

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Đóng activity thì sẽ quay trở về màn xem danh sách bài hát.

**2.2.3.2 Mô tả use case “Hiển thị chi tiết bài hát”**

Mô tả tóm tắt:

* Tên usecase : Hiển thị chi tiết bài hát
* Mục đích: Use case này cho phép người dùng xem chi tiết bài hát
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
      * 1. Use case này bắt đầu khi Khách hàng chọn thanh chơi nhạc. Hệ thống sẽ dựa vào ID của bài hát đó lấy thông tin chi tiết về bài hát có ID đấy.
        2. Lấy được thông tin sẽ hiển thị lên màn hình.
        3. Nhạc sẽ đồng thời phát khi xem
    - Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Khi phát nhạc trong service, update lại thông tin trên Notification.

Tiền điều kiện: Khách hàng đã cho phép ứng dụng truy cập thẻ nhớ để đọc file nhạc.

Hậu điều kiện: Danh sách phát tạm thời sẽ được tạo ra, những thao tác từ giờ sẽ thao tác trong danh sách phát tạm thời.

Điểm mở rộng: Nhạc sẽ được phát cả khi người dùng tắt màn hình, tắt ứng dụng

**2.2.3.3 Mô tả UseCase “Tìm kiếm”**

Mô tả tóm tắt:

* Tên usecase : Tìm kiếm
* Mục đích: Use case này cho phép người dùng tìm kiếm bài hát cần tìm
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
  1. Use case này bắt đầu khi Khách hàng chọn vào thanh tìm kiếm
  2. Hệ thống sẽ tìm kiếm theo tên ca sĩ hoặc tên bài hát sau đó hiển thị lên màn hình
     + Luồng rẽ nhánh:

1. Danh sách rỗng: Nếu không có bài hát được tìm kiếm hệ thống sẽ thông báo không tìm thấy bài hát hoặc ca sĩ cần tìm.

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.4 Mô tả UseCase “Chọn bài hát yêu thích”**

Mô tả tóm tắt:

* Tên usecase : Chọn bài hát yêu thích
* Mục đích: Use case này cho phép người dùng chọn bài hát yêu thích, hoặc bỏ bài hát yêu thích
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng chọn vào menu bên cạnh bài hát và nhấn yêu thích
* Hệ thống sẽ thêm bài hát đó vài danh sách bài hát yêu thích
  + - Luồng rẽ nhánh:
* Nếu ấn lần thứ hai thì bài hát đó sẽ bị xóa khỏi danh sách yêu thích

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.5 Mô tả UseCase “Play”**

Mô tả tóm tắt:

* Tên usecase : Play
* Mục đích: Use case này cho phép người dùng phát tiếp bài hát khi bị dừng
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn vào nút Play. Hệ thống sẽ cho phép bài hát được phát tiếp tục
  + - Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.6 Mô tả UseCase “Pause”**

Mô tả tóm tắt:

* Tên usecase : Pause
* Mục đích: Use case này cho phép người dùng dừng bài bài hát khi bài hát đang được phát
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn vào nút Pause. Hệ thống sẽ cho phép bài hát dừng khi bài hát đang phát
  + - Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.7 Mô tả UseCase “Next”**

Mô tả tóm tắt:

* Tên usecase : Next
* Mục đích: Use case này cho phép người dùng chuyển bài hát tiếp theo
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn vào nút Next. Hệ thống sẽ chuyển bài hát sang bài tiếp theo
  + - Luồng rẽ nhánh:
* Use case Play hoặc Pause sẽ kết thúc

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.8 Mô tả UseCase “Previous”**

Mô tả tóm tắt:

* Tên usecase : Previous
* Mục đích: Use case này cho phép người dùng chuyển bài hát trước đó
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn vào nút Pause. Hệ thống sẽ chuyển bài hát sang bài trước đó
  + - Luồng rẽ nhánh:
* Use case Play hoặc Pause sẽ kết thúc

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.9 Mô tả UseCase “Random”**

Mô tả tóm tắt:

* Tên usecase : Random
* Mục đích: Use case này cho phép người dùng chuyển bài hát một cách ngẫu nhiên
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn vào nút Random. Hệ thống sẽ chuyển bài hát một cách ngẫu nhiên
  + - Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

**2.2.3.10 Mô tả UseCase “Repeat”**

Mô tả tóm tắt:

* Tên usecase : Repeat
* Mục đích: Use case này cho phép người dùng nghe một bài liên tục
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn vào nút Repeat. Khi kết thúc bài hát hệ thống sẽ phát lại bài hát vừa phát
  + - Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có

**2.2.3.11 Mô tả UseCase “Seekbar”**

Mô tả tóm tắt:

* Tên usecase : Seekbar
* Mục đích: Use case này cho phép người dùng xem thời gian của bài hát và có thể tua hoặc nghe tại một vị trí tùy thích
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn chọn vào Seekbar. Hệ thống sẽ phát nhạc tại vị trí khách hàng chọn trên seekbar
  + - Luồng rẽ nhánh: Không có

Các yêu cầu đặc biệt: Không có

Tiền điều kiện: Không có.

Hậu điều kiện: Không có

**2.2.3.12 Mô tả UseCase “Hẹn Giờ”**

Mô tả tóm tắt:

* Tên usecase : Hẹn Giờ
* Mục đích: Use case này cho phép người dùng hẹn giờ để dừng bài hát
* Tác nhân : Người dùng

Luồng sự kiện:

* + - Luồng cơ bản:
* Use case này bắt đầu khi Khách hàng nhấn chọn vào icon đồng hồ. Hệ thống sẽ sang màn khác để cho người dùng chọn thời gian muốn dừng nhạc
* Khi nhấn vào nút hẹn giờ thì hệ thống sẽ chuyển về màn trước đó và tính toán đến thời gian để dừng nhạc
  + - Luồng rẽ nhánh:
* Khi quay lại màn nếu nhấn vào icon đồng hồ lần nữa UseCase “Hẹn giờ” kết thúc

Các yêu cầu đặc biệt: Không có

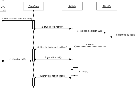
Tiền điều kiện: Không có.

Hậu điều kiện: Không có

Điểm mở rộng: Khi tắt màn hình, tắt ứng dụng nhạc vẫn phát thì UseCase “Hẹn giờ” vẫn hoạt động

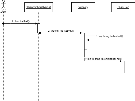
**2.3.4 Các biểu đồ trình tự**

**2.3.4.1 Biểu đồ trình tự cho UseCase “Xem danh sách bài hát”**



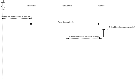
*Hình 2.4 Biểu đồ trình tự UseCase Xem danh sách bài hát*

**2.3.4.2 Biểu đồ trình tự cho UseCase “Hiển thị chi tiết bài hát”**



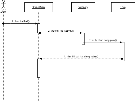
*Hình 2.5 Biểu đồ trình tự UseCase Hiển thị chi tiết bài hát*

**2.3.4.3 Biểu đồ trình tự cho UseCase “Tìm kiếm”**



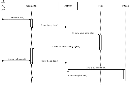
*Hình 2.6 Biểu đồ trình tự UseCase Tìm kiếm*

**2.3.4.4 Biểu đồ trình tự cho UseCase “Play”**



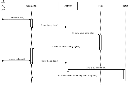
*Hình 2.7 Biểu đồ trình tự UseCase Play*

**2.3.4.5 Biểu đồ trình tự cho UseCase “Pause”**



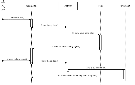
*Hình 2.8 Biểu đồ trình tự UseCase Pause*

**2.3.4.6 Biểu đồ trình tự cho UseCase “Next”**



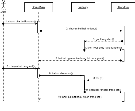
*Hình 2.9 Biểu đồ trình tự UseCase Next*

**2.3.4.7 Biểu đồ trình tự cho UseCase “Previous”**



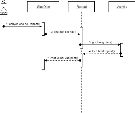
*Hình 2.10 Biểu đồ trình tự UseCase Previous*

**2.3.4.8 Biểu đồ trình tự cho UseCase “Random”**



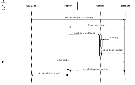
*Hình 2.11 Biểu đồ trình tự UseCase Random*

**2.3.4.9 Biểu đồ trình tự cho UseCase “Repeat”**



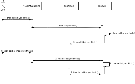
*Hình 2.12 Biểu đồ trình tự UseCase Repeat*

**2.3.4.10 Biểu đồ trình tự cho UseCase “SeekBar”**



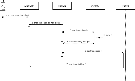
*Hình 2.13 Biểu đồ trình tự UseCase SeekBar*

**2.3.4.11 Biểu đồ trình tự cho UseCase “Chọn bài hát yêu thích”**

****

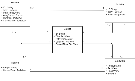
*Hình 2.14 Biểu đồ trình tự UseCase Chọn bài hát yêu thích*

**2.3.4.12 Biểu đồ trình tự cho UseCase “ Hẹn giờ ”**



*Hình 2.15 Biểu đồ trình tự UseCase Hẹn giờ*

## 2.5 Thiết kế biểu đồ lớp



*Hình 2.16* *Biểu đồ các lớp và mối quan hệ*

## 2.6 Cơ sở dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu** | **Chú thích** | **Mô tả** |
| id | String | Primary Key | ID của bài hát |
| tiltle | Varchar(50) | Not null | Tên bài hát |
| subTitle | Varchar(50) | Not null | Tên ca sĩ |
| path | Varchar(50) | Not Null | Đường dẫn bài hát |

*Chi tiết các trường của bảng Song*

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu** | **Chú thích** | **Mô tả** |
| id | String | Primary Key | ID của bài hát |
| tiltle | Varchar(50) | Not null | Tên bài hát |
| subTitle | Varchar(50) | Not null | Tên ca sĩ |
| path | Varchar(50) | Not Null | Đường dẫn bài hát |
| duration | Int | Not Null | Thời lượng bài hát |

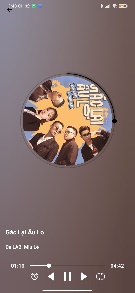
*Chi tiết các trường của bảng SongDetail*

## 2.7 Thiết kế giao diện

*Hình 2.17 Giao diện chính*

**

*Hình 2.18 Giao diện khi chọn bài hát*

**

*Hình 2.19 Giao diện khi xem chi tiết bài hát*

**

*Hình 2.20 Giao diện khi nhấn drawer để chọn màn*

**CHƯƠNG III: CÀI ĐẶT THỬ NGHIỆM**

**3.1 Giới thiệu môi trường phát triển ứng dụng tích hợp Android Studio**

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà bạn có thể sử dụng hàng ngày.

Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác.

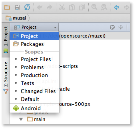
Android Studio khiến việc tạo ứng dụng trở nên dễ dàng hơn đáng kể so với phần mềm không chuyên dụng. Đối với người mới bắt đầu, có rất nhiều thứ để học và nhiều thông tin có sẵn, thậm chí thông qua các kênh chính thức nhưng chúng có thể đã lỗi thời hoặc quá nhiều thông tin khiến họ cảm thấy choáng ngợp.

Android Studio cung cấp:

* Hệ thống Gradle-based linh hoạt
* Xây dựng các biến thể và tạo nhiều tệp APK
* Code các mẫu template để hỗ trợ các tính năng app thông thường
* Chỉnh sửa bố cục đa dạng với khả năng kéo và thả theme
* Công cụ lint giúp nắm bắt hiệu suất, khả năng sử dụng, phiên bản tương thích và các vấn đề khác.
* ProGuard và ứng dụng ký app-signing
* Hỗ trợ tích hợp trên Google Cloud Platform, cho phép dễ dàng tích hợp Google Cloud Messaging và App Engine.
* Và nhiều tính năng khác nữa.

**3.2 Các thành phần chính của Android studio**

**3.2.1 Quản lý file và tệp**



*Hình 3.1* *Quản lý file và tệp trên android studio*

Theo mặc định, Android Studio hiển thị các tệp project trong chế độ xem project trên Android. Chế độ xem này cho phép xem cấu trúc project theo lát cắt, cung cấp truy cập nhanh vào các tệp source chính của các project trên Android và giúp bạn làm việc với hệ thống Gradle-based. Chế độ xem dự án Android:

* + Hiển thị các thư mục gốc quan trọng nhất ở cấp cao nhất của phân cấp module.
  + Nhóm các build file cho tất cả các module vào một thư mục chung.
  + Nhóm tất cả các file kê khai cho từng module vào một thư mục chung.
  + Hiển thị các tệp tài nguyên từ tất cả các tập nguồn nguồn Gradle.
  + Nhóm các tệp tài nguyên cho các khu vực và kiểu màn hình khác nhau trong một nhóm duy nhất cho mỗi loại tài nguyên.

**3.2.2 Hệ thống android build**

Hệ thống Android Build là bộ công cụ bạn sử dụng để xây dựng, thử nghiệm, chạy và đóng gói ứng dụng của mình. Hệ thống xây dựng này thay thế hệ thống Ant được sử dụng với Eclipse ADT. Nó có thể chạy như một công cụ tích hợp từ menu Android Studio và độc lập với dòng lệnh. Bạn có thể sử dụng các tính năng của build system để:

* + Tùy chỉnh, cấu hình và mở rộng quá trình thiết kế, kiến trúc.
  + Sử dụng lại mã và tài nguyên trên các tập hợp nguồn
* Gỡ lỗi và hiệu suất:

Android Studio cung cấp một số cải tiến hỗ trợ việc gỡ lỗi và cải thiện hiệu suất code, bao gồm các công cụ được cải tiến như công cụ quản lý thiết bị ảo, inline debug và phân tích hiệu suất.

* Trình quản lý thiết bị ảo của Android (AVD)
  + AVD Manager đã cập nhật các link lên màn hình để giúp bạn chọn các cấu hình thiết bị phổ biến nhất, lựa chọn kích thước màn hình và độ phân giải trong chế độ xem trước.
  + Trình quản lý AVD có liên kết với bộ mô phỏng cho các thiết bị Nexus 6 và Nexus 9. AVD cũng hỗ trợ tạo giao diện thiết bị Android tùy chỉnh dựa trên các đặc tính mô phỏng cụ thể và gán các giao diện đó cho cấu hình phần cứng.
  + Trình quản lý AVD có liên kết với bộ mô phỏng cho các thiết bị Nexus 6 và Nexus 9. AVD cũng hỗ trợ tạo giao diện thiết bị Android tùy chỉnh dựa trên các đặc tính mô phỏng cụ thể và gán các giao diện đó cho cấu hình phần cứng.
* Inline debugging

Sử dụng inline debugging để tăng cường code walk-throughs??? trong chế độ xem debugging với xác thực các tham chiếu, biểu thức và các giá trị biến. Thông tin Inline debugging bao gồm:

* + Giá trị biến inline
  + Tham chiếu các đối tượng (các đối tượng này là tham chiếu của một đối tượng đã chọn)
  + Phương thức trả về giá trị
  + Lambda và biểu thức toán tử
  + Các giá trị tooltip

**3.2.3 Bộ nhớ và màn hình GPU**

Android Studio hỗ trợ một chế độ xem bộ nhớ và màn hình CPU để bạn có thể dễ dàng theo dõi hiệu suất và mức sử dụng bộ nhớ của ứng dụng. Từ đó, có thể theo dõi mức sử dụng CPU, tìm đối tượng deallocated, xác định vị trí lỗi bộ nhớ, theo dõi dung lượng bộ nhớ đang được thiết bị kết nối sử dụng. Với ứng dụng chạy trên thiết bị hoặc trình mô phỏng, bạn nhấp vào tab Android ở góc dưới bên trái của cửa sổ runtime để khởi động cửa sổ Android runtime. Click tab Memory hoặc CPU.



*Hình 3.2* *Màn hình hiển thị thông số GPU của thiết bị*

**3.2.4 Truy cập file dữ liệu:**

Các công cụ SDK của Android, như Systrace, logcat và Traceview, tăng hiệu suất và gỡ lỗi dữ liệu giúp phân tích ứng dụng chi tiết.

Để xem các file dữ liệu khả dụng, click Capture ở góc bên trái trên cửa sổ runtime. Trong danh sách các tệp được tạo, click đúp vào tệp muốn xem dữ liệu. Chuột phải vào bất kỳ tệp .hprof nào để chuyển đổi chúng sang định dạng tệp .hprof tiêu chuẩn.

Trong Android Studio, lint đã được cấu hình và các kiểm tra IDE khác sẽ tự động chạy bất cứ khi nào bạn thực hiện biên dịch chương trình. Ngoài các kiểm tra lint đã cấu hình, kiểm tra mã IntelliJ bổ sung và chạy xác thực chú thích để review dòng code.

Với Android Studio, bạn cũng có thể chạy kiểm tra lint cho một biến build xác định hoặc cho tất cả các biến build. Bạn có thể cấu hình kiểm tra lint bằng cách thêm thuộc tính lintOptions vào cài đặt Android trong tệp build.gradle.



*Hình 3.3* *Kiểm tra cấu hình cài đặt Android studio*

# KẾT LUẬN

Qua thời gian tìm hiểu và nghiên cứu em đã học hỏi được một số công nghệ sau :

* Ngôn ngữ lập trình Kotlin
* Room database
* Xây dựng giao diện xml
* Lập trình ứng dụng android
* Mô hình MVVM

Trong đó đặc biệt em đã vận dụng chính ngôn ngữ Kotlin, Room và mô hình MVVM để hoàn thành sản phẩm của mình

Sau khi nghiên cứu và thực hiện đề tài em đã hoàn thành được sản phẩm ứng dụng ‘Music’ gồm các chức năng sau :

1. Chơi được nhiều định dang file audio
2. Có khả năng tự tìm và nhận dạng các file audio trong thiết bị.
3. Hiện thị danh sách nhạc online, offline.
4. Hiển thị thông tin chi tiết bài nhạc (Tên, album, thời lượng, ....)
5. Trình chơi nhạc đầy đủ chức năng (Play, Pause, Next, Previous, Tua nhạc)
6. Thay đổi thứ tự phát nhạc (Ngẫu nhiên, xáo trộn, lặp lại)
7. Phát nhạc khi đã ẩn ứng dụng, khi tắt màn hình.
8. Hẹn giờ phát nhạc.
9. Tìm kiếm bài hát.
10. Chọn bai hát yêu thích

Hướng phát triển đề tài của em là em sẽ nâng cấp thêm các chức năng như lời bài hát, hát karaoke, chat với bạn bè, share nhạc. Đồng thời chỉnh sửa giao diện sao cho đẹp hơn và phù hợp với xu hướng xây dựng ứng dụng.

**Tài liệu tham khảo**

**[1]. Giáo trình phân tích thiết kế hệ thống Trường Đại học Công Nghiệp Hà Nội.**

**[2].** [**https://developer.android.com**](https://developer.android.com)

**[3].** [**http://stackoverflow.com**](http://stackoverflow.com)

**[5].** [**https://vietjack.com/android/**](https://vietjack.com/android/)

**[6].** [**http://o7planning.org**](http://o7planning.org)

**[7]. https://viblo.asia/**