**Phần 1. TỔNG QUAN VỀ ỨNG DỤNG NGHE NHẠC**

**Version 1.0 by Phạm Trung Kiên**

1. **Tên ứng dụng**

* Ứng dụng nghe nhạc Silent Music

1. **Mục đích**

* Giúp người sử dụng nghe nhạc thư giãn online
* Tìm kiếm và nghe nhạc trực tuyến
* Sắp xếp bài hát theo tên, thể loại, ca sĩ,…

1. **Nội dung chính**

* Lấy dữ liệu bài hát online trên mạng internet
* Tìm kiếm nhạc theo tên, ca sĩ, thể loại,..
* Nghe nhạc online
* Sắp xếp nhạc theo tên, thể loại, ca sĩ,…

1. **Số người dự tính dùng ứng dụng**

* Trên 10 người dùng

1. **Tại sao người dùng phải chọn ứng dụng của bạn? Điểm khác biệt của ứng dụng**

* Ứng dụng cho cảm giác trải nghiệm nhanh, các bài hát được lưu trữ hợp lý, sắp xếp gọn gàng
* Giao diện hợp lý, thân thiện, dễ sử dụng
* Là ứng dụng miễn phí hoàn toàn với dung lượng nhỏ

**Phần 2. PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG**

1. **Tác nhân**

* User : những người có nhu cầu nghe nhạc, sắp xếp list nhạc hay tìm kiếm nhạc

1. **Usecase**

* **“Tìm kiếm nhạc”**
* **Tác nhân:** Người dùng
* **Dòng sự kiện:**

1. Mở ứng dụng vào trang chủ
2. Hiển thị chức năng tìm kiếm( nút kính lúp)
3. Nhập thông tin tìm kiếm
4. Hệ thống kiểm tra thông tin nhập vào
5. Hệ thống hiển thị danh sách bài hát được tìm theo dữ liệu đã nhập

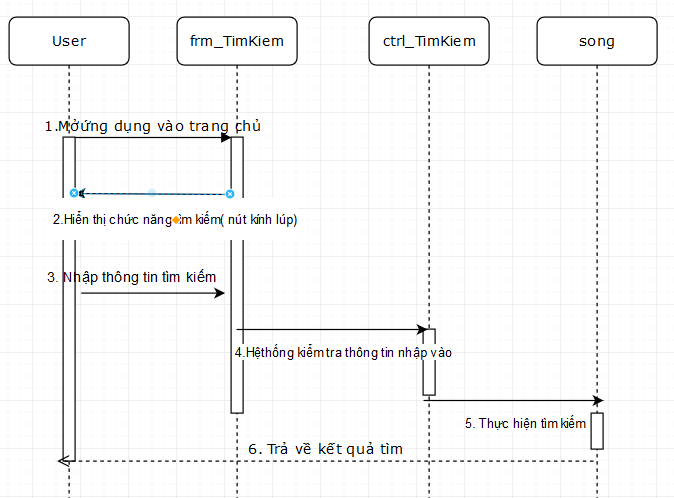
* **Ngoại lệ:**
  1. Hệ thống thông báo “Rất tiếc không có kết quả phù hợp”
* **“Sắp xếp nhạc”**
* **Tác nhân:** Người dùng
* **Dòng sự kiện:**

1. Mở ứng dụng vào trang chủ
2. Vào danh sách bài hát
3. Chọn chức sắp xếp
4. Chọn sắp xếp theo tên, ca sĩ, thời gian thêm vào,…
5. Hệ thống hiển thị danh sách bài hát được sắp xếp

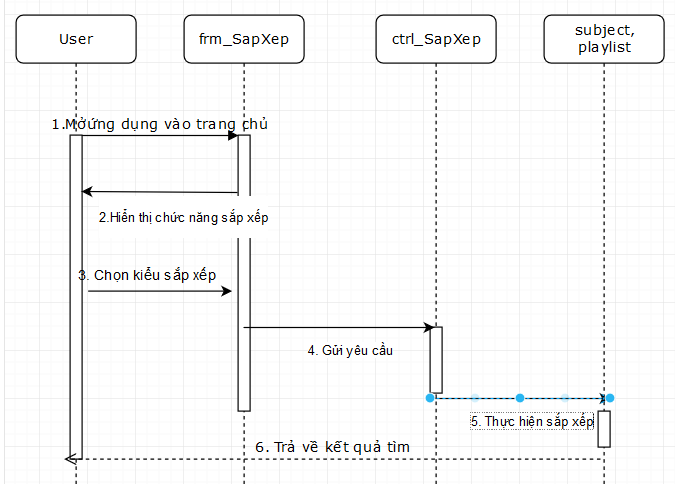
* **“Nghe nhạc”**
* **Tác nhân:** Người dùng
* **Dòng sự kiện:**

1. Mở ứng dụng vào trang chủ
2. Vào danh sách bài hát
3. Chọn bài hát muốn phát
4. Hệ thống hiển thị chi tiết bài hát và các nút chức năng: phát/dừng, bài trước, bài tiếp, …
5. **Biểu đồ trình tự**

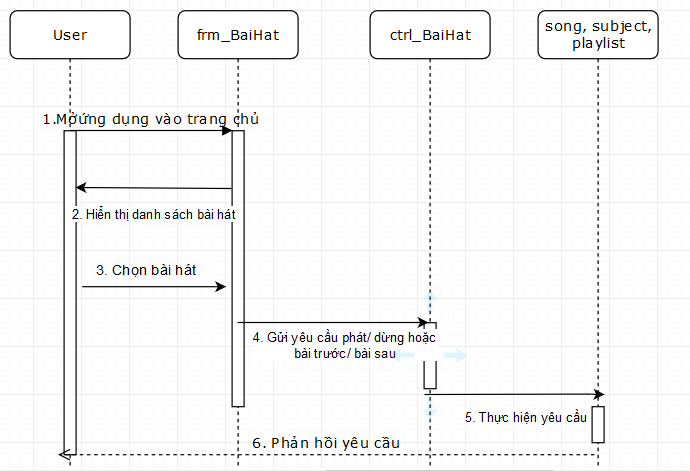
* **Tìm kiếm nhạc**



* **Sắp xếp nhạc**



* **Nghe nhạc**



1. **Cơ sở dữ liệu**

* **Bảng Song**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Định dạng** | **Mô tả** |
| song\_id | int |  |
| name | varchar(255) |  |
| image | text |  |
| artirsts | varchar(255) |  |
| link | varchar(255) |  |
| pid | varchar(255) | Khóa ngoài của Playlist |
| liked | int |  |

* **Bảng Playlist**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Định dạng** | **Mô tả** |
| pid | int |  |
| name | varchar(255) |  |
| bg\_image | text | Ảnh nền cho danh sách nhạc |

* **Bảng Advertisement**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Định dạng** | **Mô tả** |
| id | int |  |
| song\_id | int |  |
| image | text |  |
| content | varchar(255) | Chữ quảng cáo cho bài hát |

1. **Thiết kế giao diện và code**
   1. **Tổng quan**

Ứng dụng sẽ có các màn hình sau:

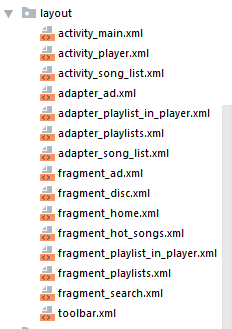
* Màn hình bắt đầu: danh sách các bài hát đang nổi, các bài hát được quảng cáo và danh sách nhạc hot nhất
* Màn hình tìm kiếm
* Màn hình chơi nhạc: trình phát và danh sách các bài được chọn để chơi

Vì đây là ứng dụng nghe nhạc online nên sẽ cần một dịch vụ cung cấp nhạc. Ở đây ta chọn cách tự xây dựng một server nhỏ bằng PHP và MySQL để đỡ mất công cào dữ liệu.

Như vậy phần code sẽ chia ra làm phần chính sau: các Service, các Model, các Activity, mỗi Activity có thể chứa một hoặc nhiểu Fragment hoặc Adapter. Để dễ quản lí code, ta chia các thành phần như vậy vào từng package tương ứng.

* 1. **Quá trình thực hiện**

Trước tiên ta cần xây dựng giao diện để dễ hình dung và thao tác. Từ phân tích trên, ta sẽ có các giao diện sau:



Sơ đồ cây phân cấp giao diện:

* Activity\_main
  + Fragment\_home: màn hình đầu tiên
    - Fragment\_ad: quảng cáo các bài hát
    - Fragment\_playlists: danh sách nhạc được nghe nhiều nhất
    - Fragment\_hot\_songs: danh sách nhạc hot
  + Fragment\_search
* Activity\_song\_list: nhận dữ liệu từ playlists và hiển thị danh sách nhạc
* Activity\_player: màn hình quan trọng nhất của ứng dụng
  + Fragment\_playlist\_in\_player: danh sách các bài hát được chọn
  + Fragment\_disc: hiển thị bài hát đang phát

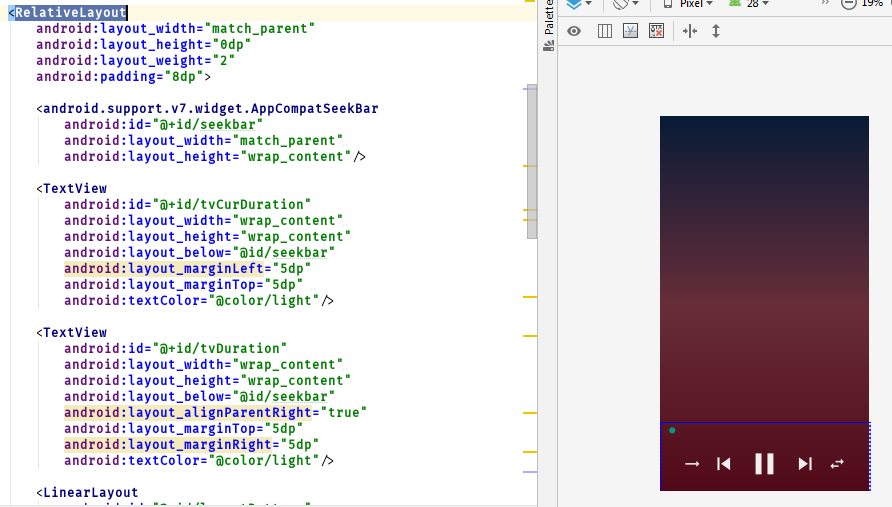
Để thuận tiện cho người dùng không phải chuyển qua lại liên tục giữa các màn hình, ta sử dụng thêm các thành phần mà Android hỗ trợ gồm [ViewPager](https://developer.android.com/reference/android/support/v4/view/ViewPager) và [Fragment](https://developer.android.com/guide/components/fragments). Đồng thời ứng dụng cũng nhanh hơn vì không phải tạo quá nhiểu activity.

Activity\_main sẽ có 1 viewpager chứa 2 fragment là fragment\_home và fragment\_search. Fragment\_home hiển thị các danh sách nhạc (thực ra là 3 fragment chứa nhạc) nên ta sẽ chọn kiểu LinearLayout cho đơn giản. Lưu ý là danh sách có thể dài hơn màn hình nên ta sẽ bọc tất cả trong 1 ScrollView:



Màn hình của activity\_search và activity\_song\_list cũng làm tương tự, nhưng thay vì dùng ScrollView thì dùng RecyclerView. Các danh sách ta đểu chọn ListView hoặc RecyclerView để hiển thị.

Tiếp theo, ta làm activity\_player, lại chia ra 2 phần fragment\_playlist\_in\_player và fragment\_disc bằng ViewPager. Phần playlist thì không có gì mới, phần chính là player sẽ có các nút điều khiển nhạc, thanh tiến trình, tên bài hát, tên nghệ sĩ và một hình đĩa nhạc xoay như máy hát. Các thành phần này không theo thứ tự đơn giản như trên nên ta dùng RelativeLayout để dàn trang. Giao diện sẽ như thế này:



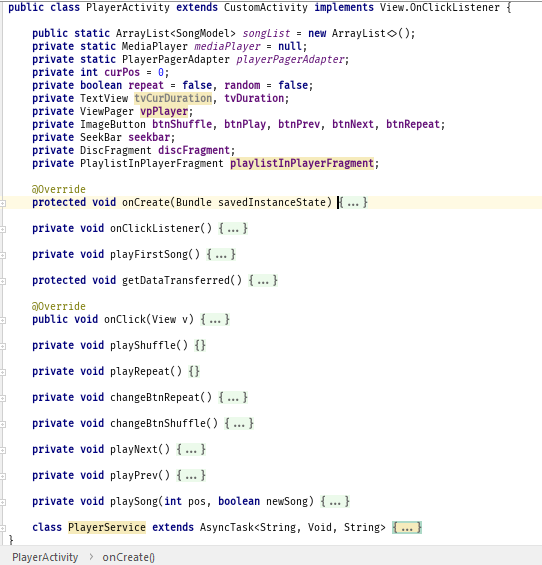
Các nút điểu khiển nằm dưới cùng và xếp hàng ngang, có thể cho vào 1 LinearLayout với orientation=”horizontal”:



Đó là các thành phần cơ bản của giao diện, bây giờ ta sẽ đi vào phân tích mã nguồn.

Phần dịch vụ giao tiếp với server, ta tạo ra các api trả về định dạng JSON rồi dùng Java đọc các chuỗi này và đưa vào các Model tương ứng. Ta xây dựng 1 lớp DataService chịu trách nhiệm lấy dữ liệu, các đối tượng SongModel, AdModel và PlaylistModel hứng dữ liệu trả về của DataService. Các đối tượng này có các thuộc tính giống như các bảng dữ liệu ta phân tích bên trên.

Ta không đi sâu vào phân tích các thành phần phụ mà chỉ phân tích đối tượng quan trọng nhất là trình phát nhạc. Trước hết ta xem tổng quan mã nguồn:

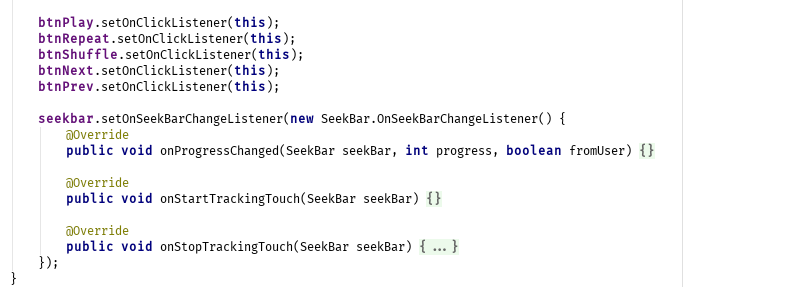


Về giao diện, ta có thể thấy các fragment danh sách nhạc và đĩa nhạc, các nút điều khiển và thanh tiến trình. Các nút này được diều khiển bởi đối tượng MediaPlayer của Android. Ta còn có 1 đối tượng songList chứa các bài hát muốn nghe. Chức năng của trình phát nhạc thường thấy gồm chơi nhạc theo thứ tự, ngẫu nhiên, lặp,… được ánh xạ thành các phương thức playSong(), playShuffle(),…

Để phát nhạc ngầm và không làm ứng dụng bị nghẽn, ta tạo ra 1 lớp bên trong là PlayerService thực hiện việc chơi nhạc bất đồng bộ. Như vậy khi người dùng thao tác sẽ không có cảm giác bị chậm hay giật lag. Các phương thức chơi nhạc đều sử dụng dịch vụ của PlayerService.



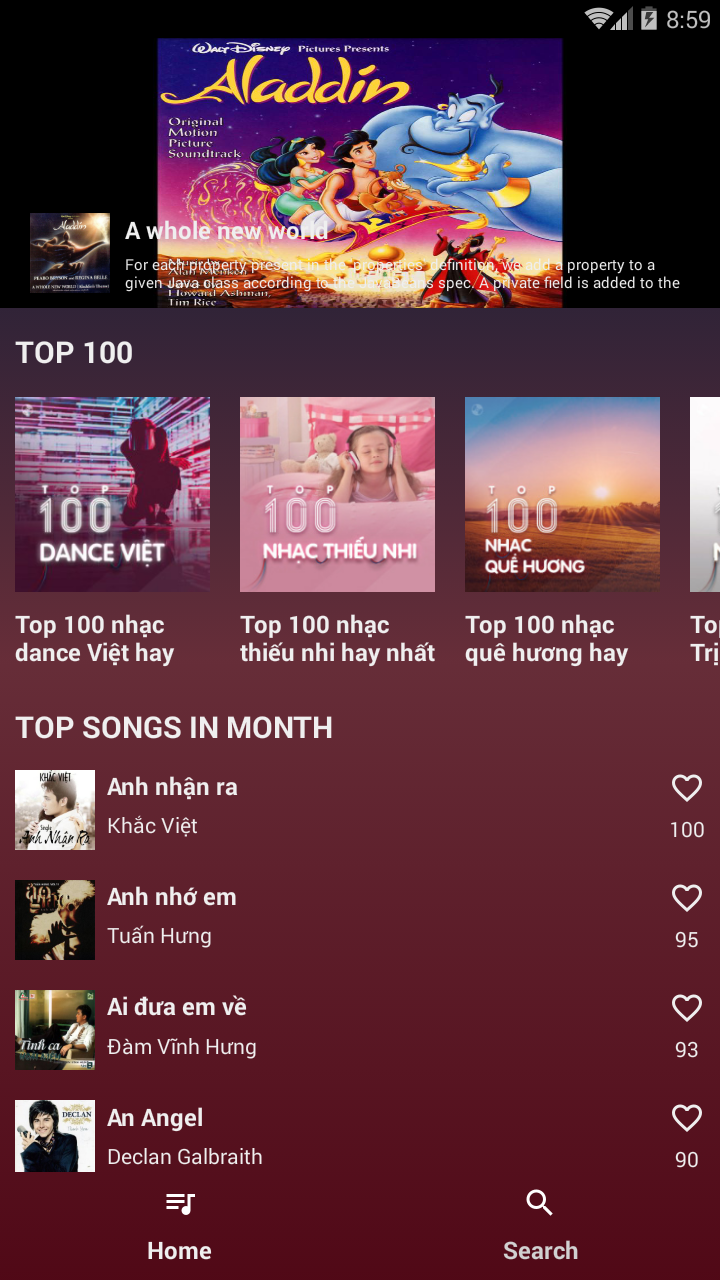
Khi nhận được tín hiệu phát nhạc, lớp này sẽ lấy thông tin bài hát (tên, nghệ sĩ, thời lượng) và thiết lập cho các đối tượng seekBar, textview hiện thông tin.



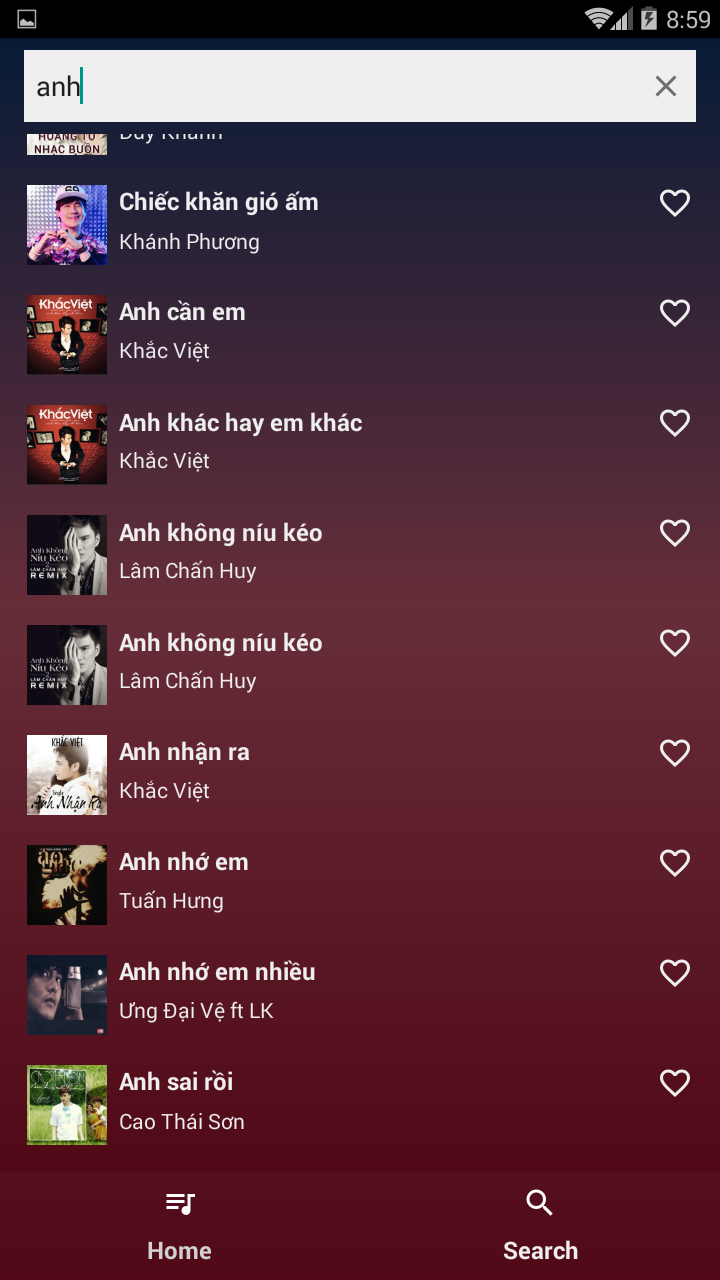
Các nút và seekBar được gán sự kiện xử lí khi chơi nhạc, chính là các phương thức đã nêu. Như vậy ta đã hình dung khá rõ ràng cách thức tạo ra ứng dụng này như thế nào.

Chi tiết dự án có thể xem tại: <https://github.com/shhlkien/silent-music>

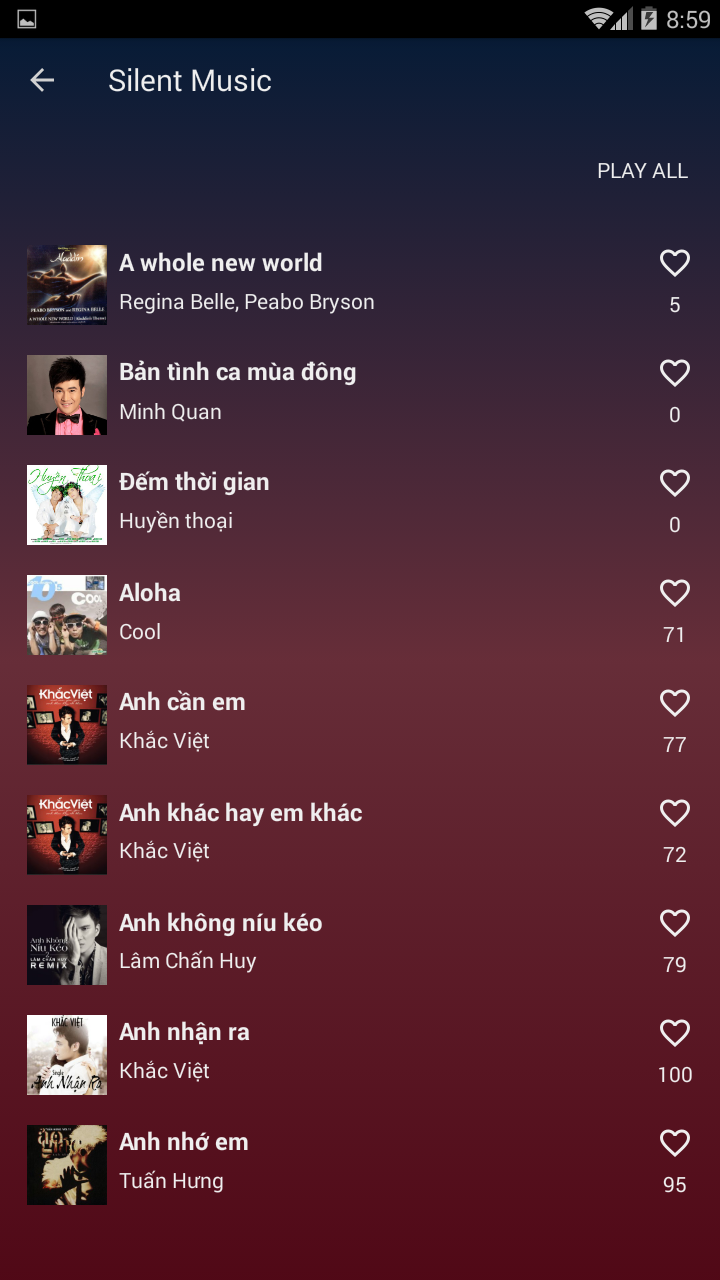
1. **Giao diện**

****

**Màn hình chính**

****

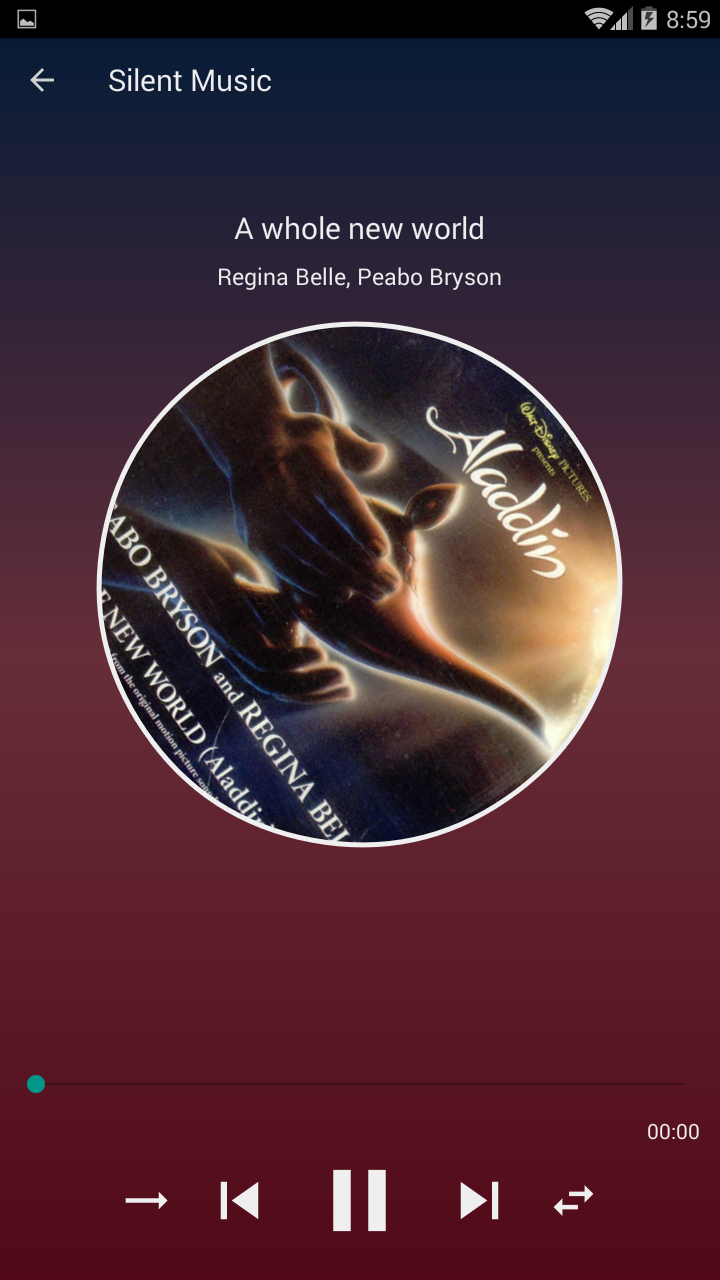
**Màn hình tìm kiếm**

****

**Màn hình danh sách nhạc**

****

**Màn hình danh sách trong trình phát nhạc**

****

**Màn hình phát nhạc**