

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC



BÁO CÁO KẾT THÚC MÔN HỌC

Lập Trình Di Động 2

Nhóm sinh viên nhóm 1:

Nguyễn Xuân Nguyên Hạnh

Nguyễn Văn Phúc

Dương Quốc Vĩnh

Nguyễn Văn Thái

Phạm Lê Minh Phú

Giảng viên hướng dẫn:

Trương Bá Thái

Ngày 20/04/2019

NHẬT KÝ HOẠT ĐỘNG NHÓM 1

STT	HỌ TÊN	CÔNG VIỆC	TƯ ĐÁNH GIÁ	NHÓM ĐÁNH GIÁ	KÝ TÊN
1	Nguyễn Xuân Nguên Hạnh	Nghiêm cứu đề tài Viết SRS Viết SDS Viết báo cáo Báo cáo cuối kỳ	20%	20%	
2	Dương Quốc Vĩnh	Nghiêm cứu đề tài Viết SRS Viết SDS Báo cáo cuối kỳ	20%	20%	
3	Phạm Lê Minh Phú	Nghiêm cứu đề tài Viết SRS Viết SDS Báo cáo cuối kỳ	20%	20%	
4	Nguyễn Văn Thái	Nghiêm cứu đề tài Viết SRS Viết SDS Báo cáo cuối kỳ	20%	20%	
5	Nguyễn Văn Phúc	Nghiêm cứu đề tài Viết SRS Viết SDS Làm màn hình 3 Báo cáo cuối kỳ	20%	20%	

Danh mục

NHẬT KÝ HOẠT ĐỘNG NHÓM 1	2
CHƯƠNG I. MỞ ĐẦU	4
1. Giới thiệu môn học và nhóm thực hiện.....	4
2. Mô tả ứng dụng	4
3. Ứng dụng trong cuộc sống.....	4
CHƯƠNG II. PHÂN TÍCH CẤU TRÚC HỆ THỐNG	5
1. Phân tích thiết hệ thống.....	5
2.Thiết kế hệ thống.....	26
2.1 Màn hình Navigation Drawer (Nguyễn Văn Phúc).....	27
2.2 Màn hình Playlists (Nguyễn Văn Phúc)	31
2.3 Màn hình chi tiết Playlists (Nguyễn Văn Phúc).....	34
2.4 Giao diện danh mục Ablums (Nguyễn Xuân Nguyên Hạnh)	42
2.5 Màn hình bài hát trong Albums (Nguyễn Xuân Nguyên Hạnh).....	44
2.9 Màn hình phát nhạc (Dương Quốc Vĩnh).....	45
2.10 Giao diện màn hình danh sách nghệ sĩ – Artist (Phạm Lê Minh Phú).....	47
2.11 Chức năng Shuffle all trên thanh menu (Phạm Lê Minh Phú).....	49
CHƯƠNG III. CÀI ĐẶT	52
3.1 Màn hình Navigation Drawer (Nguyễn Văn Phúc).....	52
3.2 Màn hình Playlists (Nguyễn Văn Phúc)	53
3.3 Màn hình chi tiết Playlists (Nguyễn Văn Phúc).....	55
3.4 Giao diện danh mục Ablums (Nguyễn Xuân Nguyên Hạnh)	58
3.5 Màn hình bài hát trong Albums (Nguyễn Xuân Nguyên Hạnh).....	60
3.6 Màn hình Playing Queue (Nguyễn Văn Thái)	64
3.7 Màn hình phát nhạc – Dương Quốc Vĩnh.....	66
3.8 Màn hình Artists (Phạm Lê Minh Phú)	68
3.9 Màn hình View as (Phạm Lê Minh Phú)	69
3.10 Chức năng Shuffle all (Phạm Lê Minh Phú)	70
CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC	70
4.1 Kết quả đạt được.....	71
4.2 Kết luận.....	71
4.3 Tài liệu kham khảo	71

CHƯƠNG I. MỞ ĐẦU

1. Giới thiệu môn học và nhóm thực hiện

- Môn di động 2 là môn nâng cao của di động 1. Môn học này giúp sinh viên phát triển kỹ năng lập trình, tự tiêm kiém, cập nhật kiến thức mới giúp áp dụng cho các dự án thực tế.
- Nâng cao kỹ năng lập trình Android.
- Rèn luyện kỹ năng hoàn thành 1 dự án theo yêu cầu.
- Rèn luyện kỹ năng Unit Test
- Rèn luyện kỹ năng tự đọc tài liệu, tìm kiếm tài liệu.
- Nâng cao kỹ năng làm việc nhóm.

2. Mô tả ứng dụng

- Tên ứng dụng: App Music
- Đọc nhạc từ thẻ SD Card
- Android 4.3+
- Phát nhạc trên thẻ nhớ
- Danh sách bài hát
- Bài hát yêu thích
- Quản lý nhạc thông minh theo nhạc sĩ, ca sĩ, yêu thích.
- Tìm kiếm bài hát theo ca sĩ, tên bài hát.
- Playlist thích

3. Ứng dụng trong cuộc sống

- App Music đây là ứng dụng miễn phí trên di động và máy tính bảng. Nghe nhạc mọi nơi mà không cần internet. Giao diện đơn giản dễ dàng sử dụng, thân thiện người dùng.

CHƯƠNG II. PHÂN TÍCH CẤU TRÚC HỆ THỐNG

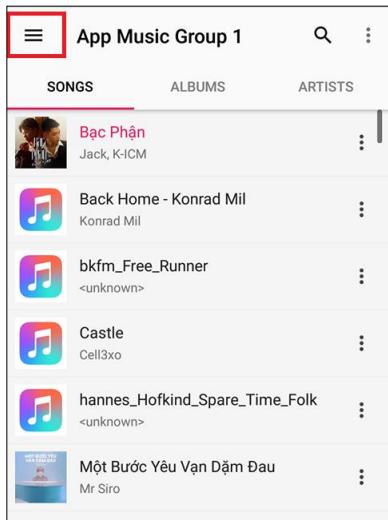
1. Phân tích thiết hệ thống

1.1 Màn hình Navigation Drawer (Nguyễn Văn Phúc)

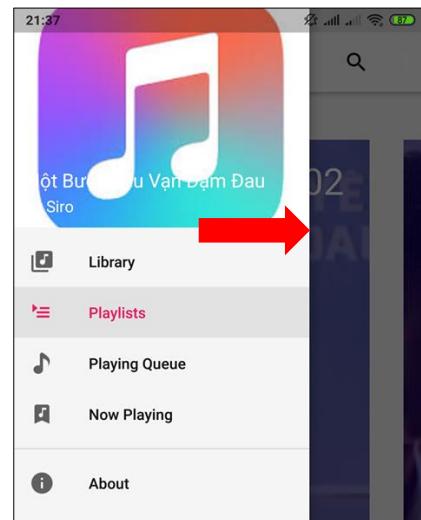
1.1.1 Tổng quan

Màn hình có nhiệm vụ chủ yếu chuyển các màn hình cần thiết để thuận tiện nhất cho người sử dụng.

Chọn mục tô đỏ để vào Navigation Drawer



Hình 1 Giao diện danh mục



Hình 2 Giao diện menu

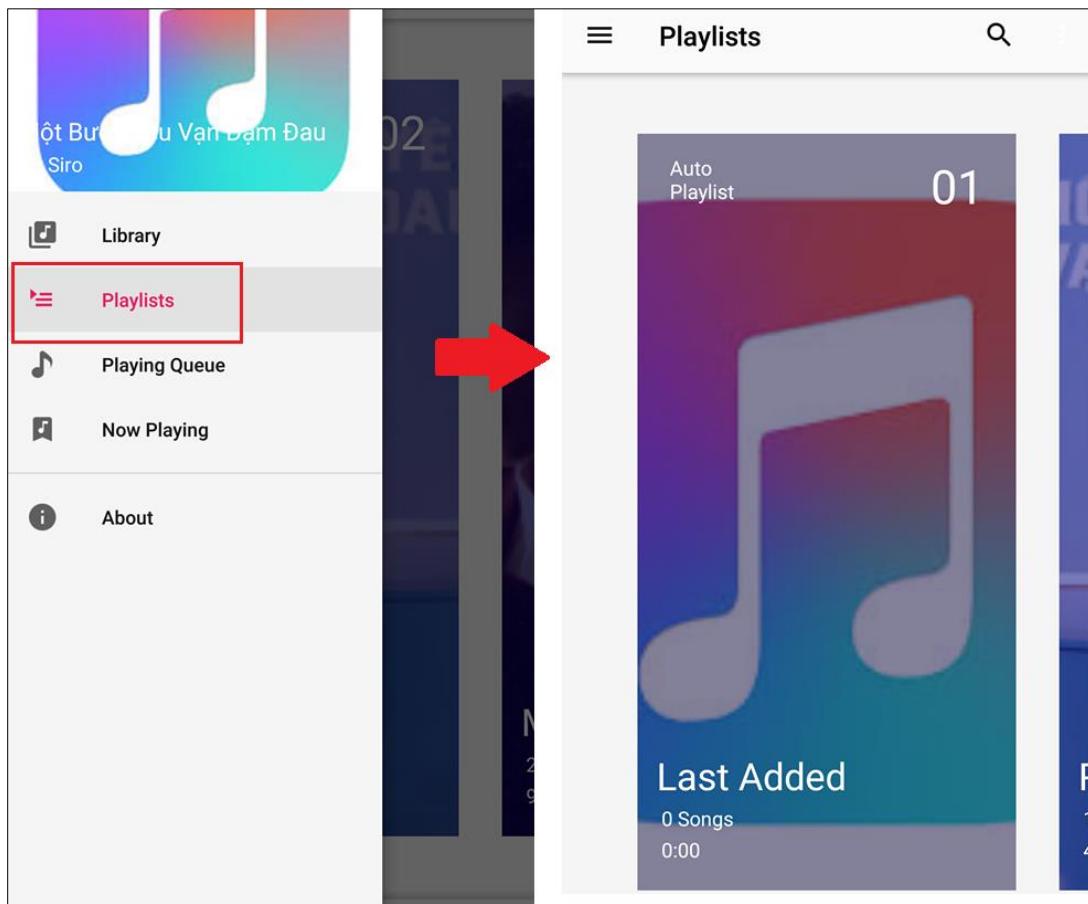
1.1.2 Yêu cầu hệ thống

Tiêu đề	Mô tả	Qui trình	Kết quả
Button navigation	- Chuyển sang giao diện Navigation Drawer	Click	Chuyển sang giao diện mục Navigation Drawer
Group Items Library	Chuyển sang giao diện thư viện nhạc	Click	Chuyển sang giao diện thư viện nhạc
Group Items Playlists	Chuyển sang giao diện Playlist	Click	Chuyển sang giao diện Playlist
Group Items Playing Queue	Chuyển sang giao diện danh sách nhạc chờ phát	Click	Chuyển sang giao diện danh sách nhạc chờ phát
Group Items Now Playing	Chuyển sang giao diện phát nhạc	Click	Chuyển sang giao diện phát nhạc
Group Items About	Chuyển sang giao diện giới thiệu	Click	Chuyển sang giao diện giới thiệu

1.2 Màn hình Playlists (Nguyễn Văn Phúc)

1.2.1 Tổng quan

Màn hình có nhiệm vụ chủ yếu để người dùng biết được số lượng bài hát them vào, bài hát vừa nghe gần đây, các bài hát được nghe nhiều nhất và có thêm tạo 1 playlist nhạc mà mình yêu thích.



Hình 3 Màn hình playlists

1.2.1 Yêu cầu hệ thống

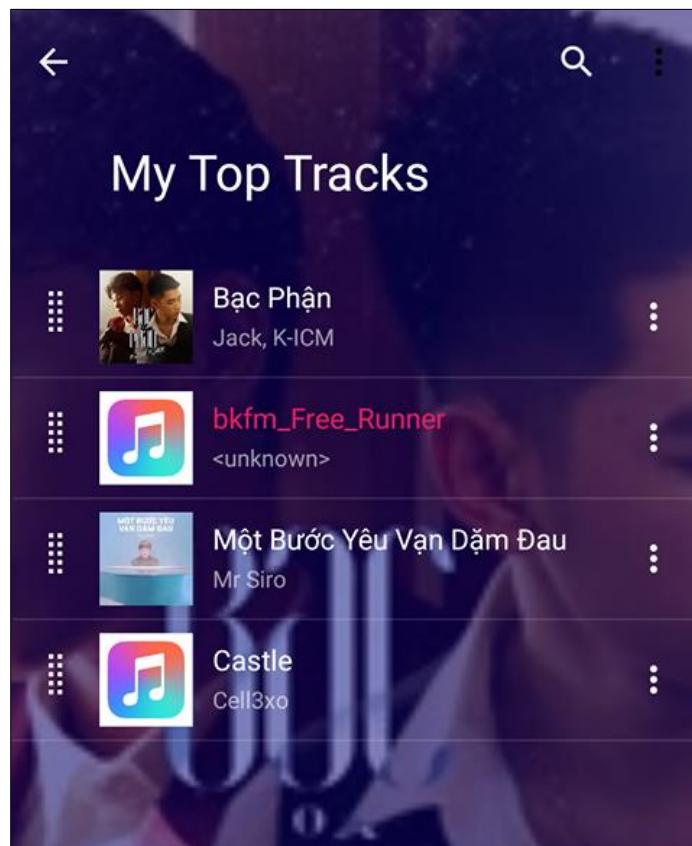
Tiêu đề	Mô tả	Qui trình	Kết quả
Item RecyclerView Last Added	Chuyển sang giao diện Playlist Last Added	Click	Chuyển sang giao diện Playlist Last Added
Item RecyclerView Recently Played	Chuyển sang giao diện danh sách phát gần đây	Click	Chuyển sang giao diện danh sách phát gần đây
Item RecyclerView My Top Tracks	Chuyển sang giao diện bài hát được nghe nhiều nhất	Click	Chuyển sang giao diện bài hát được nghe nhiều nhất

Item	RecyclerView	Chuyển sang giao diện playlist được tạo mới	Click	Chuyển sang giao diện playlist được tạo mới
Playlist được tạo mới				

1.3 Màn hình chi tiết Playlists (Nguyễn Văn Phúc)

1.3.1 Tổng quan

Màn hình có nhiệm vụ để người xem chi tiết playlist của mình và có thể thực hiện các chức năng (xóa, xem ca sĩ, albums, xóa bài hát,...)



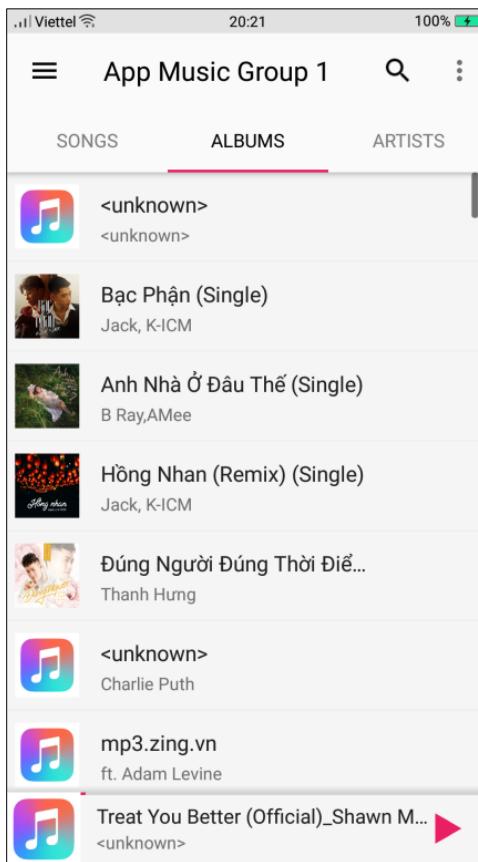
Hình 4 Chi tiết Playlist

1.3.2 Yêu cầu hệ thống

Tiêu đề	Mô tả	Qui trình	Kết quả

Danh sách bài hát	Play nhạc	Click	Phát bài hát
Button menu	Hiển thị các tùy chọn	Click	Hiển thị chức năng để người dùng lựa chọn
Item Menu Remove from playlist	Xóa bài hát ra khỏi Playlist	Click	Xóa bài hát ra khỏi Playlist
Item menu Play	Phát bài hát	Click	Phát bài hát
Item menu play next	Bài hát sẽ được phát sau bài hát đang phát	Click	Chuyển bài hát vào danh sách chờ
Item menu queue	Chuyển bài hát vào danh sách chờ	Click	Chuyển bài hát vào danh sách chờ
Item menu Add to playlist	Bài hát sẽ được thêm vào playlist được chọn	Click	Bài hát sẽ được thêm vào playlist được chọn
Item menu Go to album	Chuyển đến album của bài hát	Click	Chuyển đến album của bài hát
Item menu Go to artist	Chuyển đến ca sĩ của bài hát	Click	Chuyển đến ca sĩ của bài hát
Menu item Delete form device	Xóa bài hát được chọn	Click	Xóa bài hát được chọn

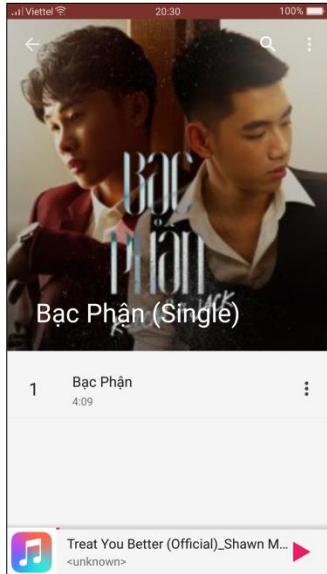
1.4 Giao diện danh mục Ablums (Nguyễn Xuân Nguyên Hạnh)



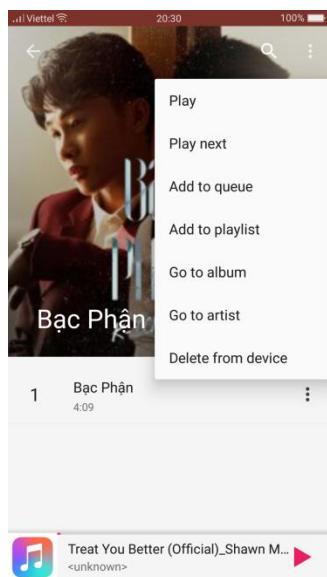
Hình 5 Màn hình Albums

Tiêu đề	Mô tả	Qui trình	Kết quả
TextView ‘Bạc Phận(Single)’	- TextView hiển thị ‘tên Album’	Click	Chuyển màn hình list bài hát có trong Albums đó
TextView ‘Jack, K-ICM’	- TextView hiển thị ‘tên Ca sĩ/ Nhạc sĩ’ của ‘Album’	Click	Chuyển màn hình list bài hát có trong Albums đó
ImageView của Alums	Hiển thị hình ảnh của Album đó	Click	Chuyển màn hình list bài hát có trong Albums đó

1.5 Màn hình bài hát trong Albums (Nguyễn Xuân Nguyên Hạnh)



Hình 6 Màn hình chi tiết bài hát trong Album

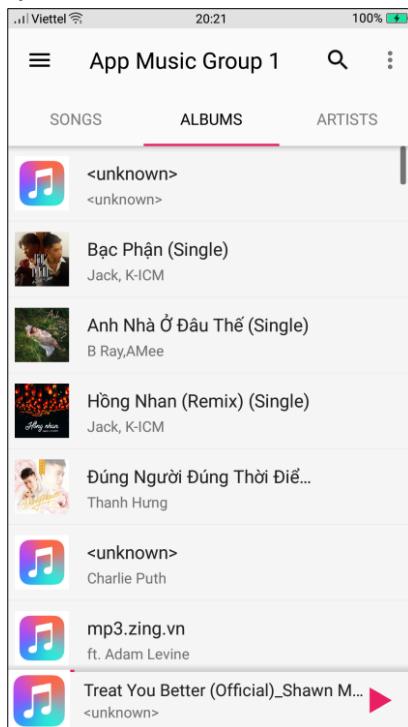


Hình 7 Màn hình chi tiết bài hát trong Album

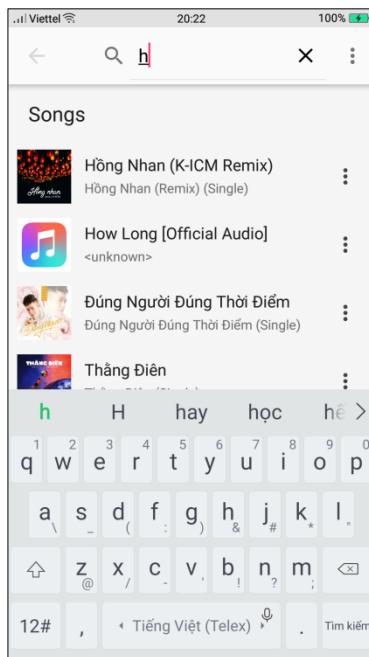
Tiêu đề	Mô tả	Qui trình	Kết quả
TextView ‘Bạc Phận (Single)’	Textview hiển thị ‘tên Album’	Click	Chuyển màn hình phát nhạc bài hát đó
TextView ‘Bạc Phận’	Textview hiển thị ‘tên bài hát’ của Albums đó	Click	Chuyển màn hình phát nhạc bài hát đó
TextView ‘4:09’	Hiển thị thời gian của bài hát đó trong Albums	Click	Chuyển màn hình phát nhạc bài hát đó
ImageView của Albums	Hiển thị hình ảnh của Albums đó	Click	Không có gì
TextView ‘1’	Hiển thị số thứ tự bài hát đó trong Albums	Click	Chuyển màn hình phát nhạc bài hát đó
Menu popupsong	Hiển thị menu bài hát	Click	Hiển thị danh mục menu cho bài hát.
ItemMenu ‘Play’	Item Play bài hát đó trong Albums	Click	Chuyển sang màn hình phát nhạc bài hát đó
ItemMenu ‘Next Play’	Item Next Play bài hát tiếp theo trong Albums	Click	Chuyển sang màn hình phát nhạc bài hát tiếp theo
ItemMenu ‘Add to queue’	Item Add to queue thêm bài hát vào mục ‘Playing queue’	Click	Thêm bài hát vào mục Playing queue
ItemMenu ‘Add to playlist’	Item Add to playlist thêm bài hát vào mục ‘Playlist’	Click	Thêm bài hát vào mục Playlist
ItemMenu ‘Go to Ablum’	Item Go to Album chuyển bài hát vào mục Album	Click	Chuyển bài hát vào mục Album

ItemMenu ‘Go to Artist’	Item Go to Artist chuyển bài hát vào mục Artist	Click	Chuyển bài hát vào mục Artist
ItemMenu ‘Delete from device’	Item Delete from device xóa bài hát khỏi thiết bị	Click	Xóa bài hát khỏi thiết bị

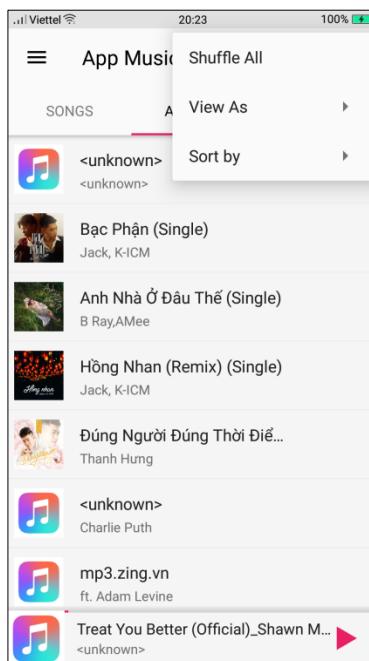
1.6 Màn hình Menu chức năng sắp xếp và chức năng tìm kiếm bài hát. (Nguyễn Xuân Nguyên Hạnh)



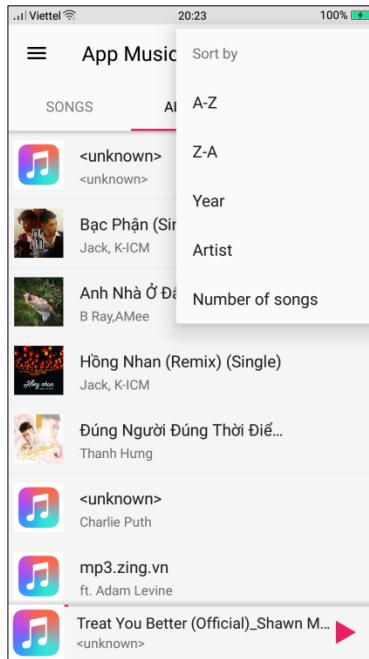
Hình 8 Màn hình Album chức năng tìm kiếm, sắp xếp bài hát, ca sĩ



Hình 9 Màn hình Album chức năng tìm kiếm, sắp xếp bài hát, ca sỹ



Hình 10 Màn hình Album chức năng tìm kiếm, sắp xếp bài hát, ca sỹ



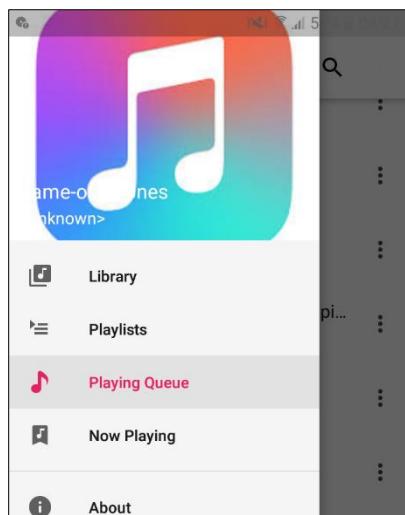
Hình 11 Màn hình Album chức năng tìm kiếm, sắp xếp bài hát, ca sỹ

Tiêu đề	Mô tả	Qui trình	Kết quả
MenuItem Search	Item Search tìm kiếm bài hát	- Click - Nhập keyword tìm kiếm	Hiện thông tin bài hát tìm kiếm và hiện MenuItem ‘X’ để xóa keyword vừa search
MenuItem ‘Sort by’	Item Sort by trỏ chuột vào sẽ hiện danh mục của MenuItem Sort by	- Mouse Hover	Hiện danh mục của MenuItem Sortby
MenuItem ‘A-Z’	Item A-Z sắp xếp bài hát từ theo thứ tự từ A-Z	Click	Danh sách bài hát sẽ được sắp xếp theo thứ tự từ kí tự A – kí tự Z
MenuItem ‘Z-A’	Item Z-A sắp xếp bài hát từ theo thứ tự từ Z-A	Click	Danh sách bài hát sẽ được sắp xếp theo thứ tự từ kí tự Z – kí tự A
MenuItem ‘Year’	Item Year sắp xếp theo năm bài hát	Click	Danh sách bài hát sẽ sắp xếp theo năm bài hát

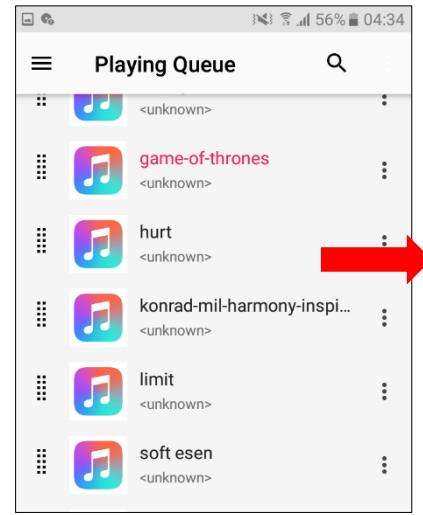
MenuItem ‘Artist’	Item Artist sắp xếp theo ca sỹ	Click	Danh sách bài hát sẽ sắp xếp theo ca sỹ
MenuItem ‘Number of songs’	Item Artist sắp xếp theo số bài hát	Click	Danh sách bài hát sẽ sắp xếp theo số bài hát

1.7 Màn hình Playing Queue (Nguyễn Văn Thái)

Màn hình có nhiệm vụ chuyển người dùng đến danh sách bài hát mình chọn để phát (có thể hiểu là bài hát mình thích) sau khi người dùng chọn chức năng Add to queue ở list nhạc trước.



Hình 12 Chọn danh mục

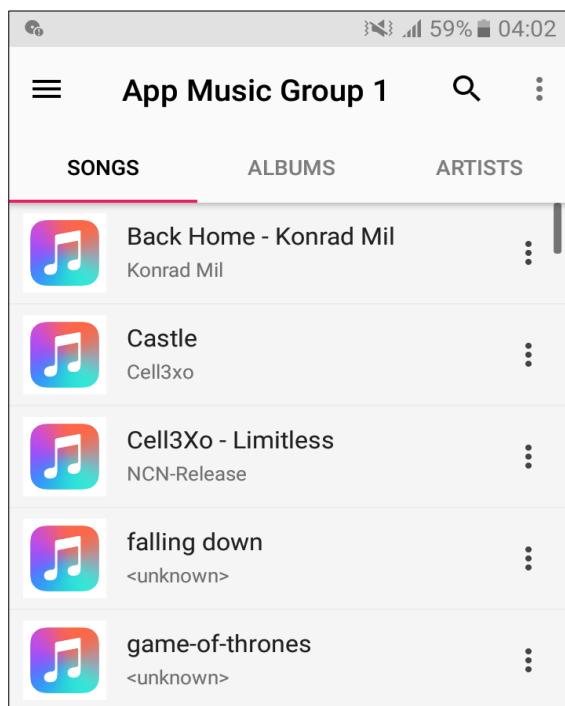


Hình 13 Chọn danh mục

Tiêu đề	Mô tả	Qui trình	Kết quả
Danh sách bài hát (Playing Queue)	Play nhạc	Click	Phát bài hát

Button menu	Hiển thị các tùy chọn	Click	Hiển thị chức năng để người dùng lựa chọn
Item Menu Remove from Queue	Xóa bài hát ra khỏi Playing Queue	Click	Xóa bài hát ra khỏi Playing Queue
Item menu Play	Phát bài hát	Click	Phát bài hát
Item menu Add to playlist	Bài hát sẽ được thêm vào playlist được chọn	Click	Bài hát sẽ được thêm vào playlist được chọn
Item menu Go to album	Chuyển đến album của bài hát	Click	Chuyển đến album của bài hát
Item menu Go to artist	Chuyển đến ca sĩ của bài hát	Click	Chuyển đến ca sĩ của bài hát

1.8 Màn hình chi tiết Songs (Nguyễn Văn Thái)



Hình 14 Chi tiết Songs

Tiêu đề	Mô tả	Qui trình	Kết quả
Danh sách bài hát	Play nhạc	Click	Phát bài hát
Button menu	Hiển thị các tùy chọn	Click	Hiển thị chức năng để người dùng lựa chọn
Item menu Play	Phát bài hát	Click	Phát bài hát
Item menu play next	Bài hát sẽ được phát sau bài hát đang phát	Click	Chuyển bài hát vào danh sách chờ
Item menu queue	Chuyển bài hát vào danh sách chờ	Click	Chuyển bài hát vào danh sách chờ
Item menu Add to playlist	Bài hát sẽ được thêm vào playlist được chọn	Click	Bài hát sẽ được thêm vào playlist được chọn
Item menu Go to album	Chuyển đến album của bài hát	Click	Chuyển đến album của bài hát
Item menu Go to artist	Chuyển đến ca sĩ của bài hát	Click	Chuyển đến ca sĩ của bài hát
Menu item Delete from device	Xóa bài hát được chọn	Click	Xóa bài hát được chọn

1.9 Chức Năng Màn hình phát nhạc (Đương Quốc Vĩnh)

Màn hình phát nhạc

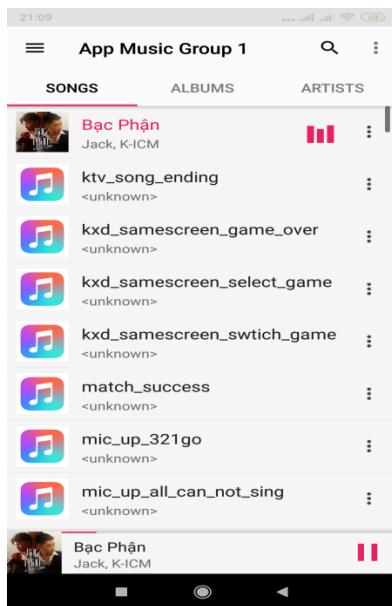


Hình 15 Màn hình phát nhạc

o Yêu cầu chức năng

Tiêu đề	Mô tả	Qui trình	Kết quả
Image View	Hiển thị hình của bài hát	View	Hiển thị hình bài hát
TextView ‘Bạc Phận, Jack, K-ICM’	- TextView hiển thị ‘Tên bài hát, tên ca sỹ’	View	Hiển Thị View tên bài hát, tên ca sỹ
Button Pause	- Bấm vào sẽ dừng bài hát đang phát, hiện Button Resume.	Click	Bấm vào sẽ dừng bài hát đang phát, hiện Button Resume.
Button Next	- Bấm vào chuyển bài hát tiếp theo.	Click	Bấm vào chuyển bài hát tiếp theo.

Button Previous	-Bấm vào chuyển bài hát trước đó.	Click	Bấm vào chuyển bài hát tiếp theo.
TextView TimeStart	-Hiển thị thời gian bài hát bắt đầu chạy tới hết thời gian bài hát	View	Thời gian bài hát bắt đầu chạy tới hết thời gian bài hát
TextView TimeResult	-Hiển thị thời tổng thời gian bài hát đếm lùi tới hết bài hát là thời gian 00:00	View	Tổng thời gian bài hát đếm lùi tới hết bài hát là thời gian 00:00
Seebar TimelineSong	-Hiển thị vạch thời gian bài hát, có thể di chuyển vạch bài hát	Touch	Vạch thời gian bài hát di chuyển, kéo thời gian bài hát tới đâu thì bài phát sẽ phát tới khúc đấy
Button icon “Lặp lại bài hát”	-Click button 1 lần thì lặp lại liên tục bài hát đang phát, click 2 lần lặp lại 1 lần.	Click	Lặp lại bài hát theo tùy chọn



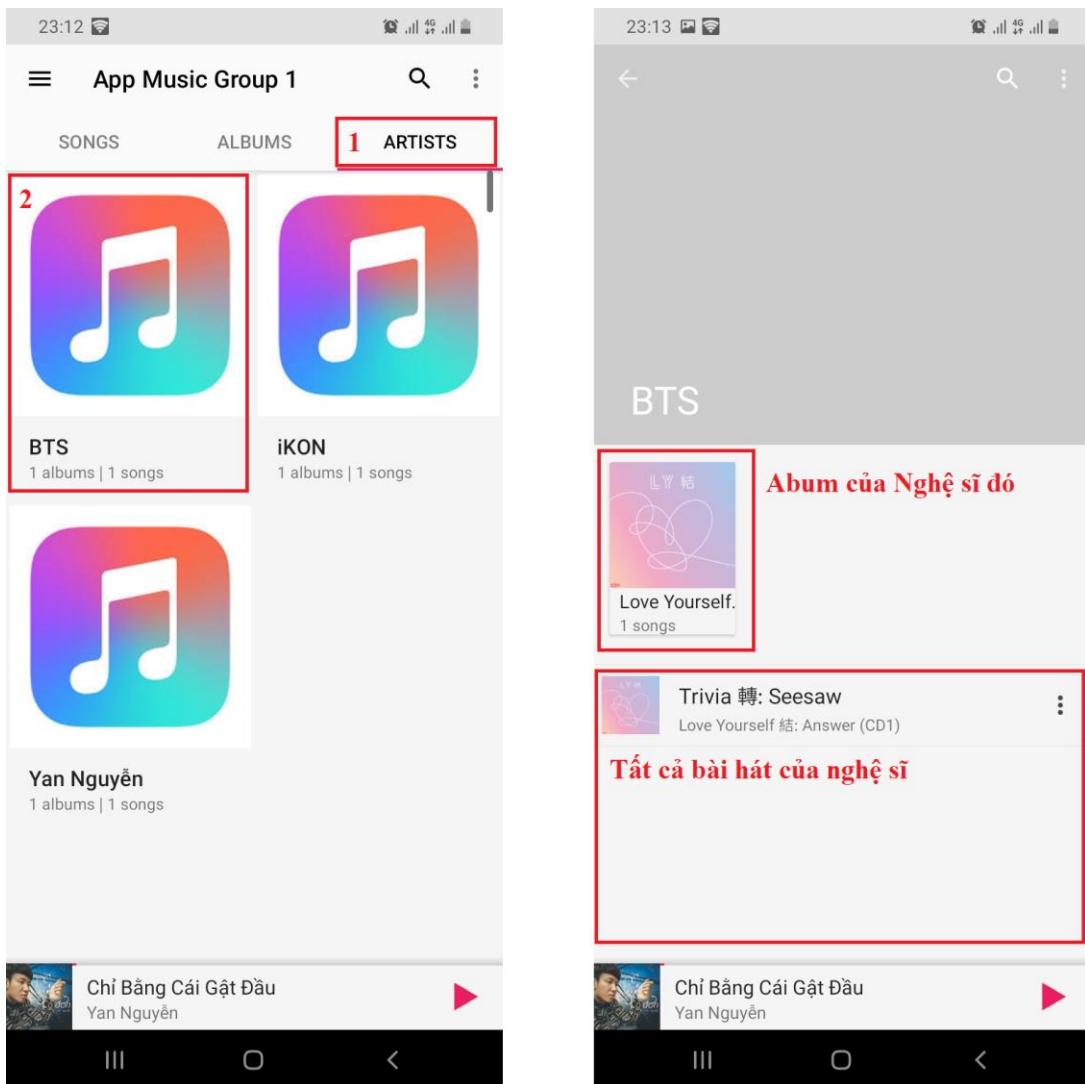
Hình 16 Màn hình đang phát nhạc khi chuyển màn hình khác

Tiêu đề	Mô tả	Qui trình	Kết quả
Bottom View	Hiển thị bài hát đang phát hiện tại.	Phát nhạc, sau đó chuyển sang màn hình khác	Hoàn thành
Image view	Hiển thị hình của bài hát	Hiển thị	Hoàn thành
Text view “Bạc phận, jack, K-ICM”	Hiển thị tên bài hát và ca sỹ	Hiển thị	Hoàn thành
Button pause, play	Chuyển thành button pause nếu đang phát và ngược lại	Click	Phát nhạc nếu đang dừng, hoặc dừng nếu đang phát

1.10 Giao diện màn hình danh sách nghệ sĩ – Artist (Phạm Lê Minh Phú)

1.10.1 Tổng quan

Màn hình Artist có chức năng hiển thị danh sách các nghệ sĩ. Khi click vào mỗi nghệ sĩ sẽ chuyển sang thông kê ra các Album của họ và tất cả các bài hát của họ.



Hình 17 Chức năng Shuffle all trên thanh menu

1.10.2 Yêu cầu hệ thống

Màn hình Artists

Tiêu đề	Mô tả	Qui trình	Kết quả
Hình ảnh Adapter Item	Hiển thị hình ảnh của Nghệ sĩ đó	- Click	Chuyển sang màn hình chi tiết Nghệ sĩ Artist Detail
Textview Adapter Item Tên Nghệ Sĩ	Hiển thị tên của nghệ sĩ đó	- Click	Chuyển sang màn hình chi tiết Nghệ sĩ Artist Detail
Textview Adapter Item Albums	Thông kê số lượng Album	- Click	Chuyển sang màn hình chi tiết Nghệ sĩ Artist Detail
Textview Adapter Item Songs	Thông kê số lượng bài hát	- Click	Chuyển sang màn hình chi tiết Nghệ sĩ Artist Detail

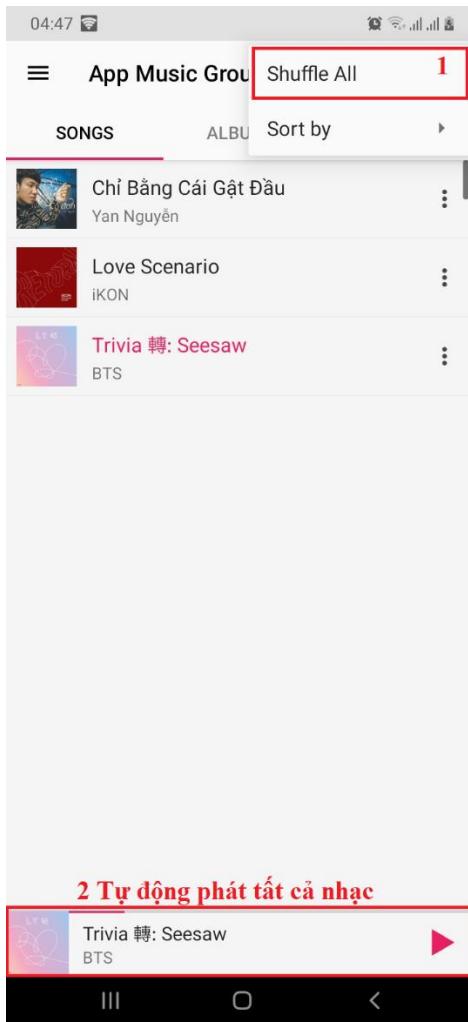
Màn hình Artist Detail

Tiêu đề	Mô tả	Qui trình	Kết quả
Hình ảnh Adapter Item	Hiển thị hình ảnh của Nghệ sĩ đó	- Show	Hiển thị hình ảnh của Nghệ sĩ đó
Textview Adapter Item Tên Nghệ Sĩ	Hiển thị tên của nghệ sĩ đó	- Show	Hiển thị tên của nghệ sĩ đó
GridView Album	Hiện thị danh sách Album của nghệ sĩ	- Click	Chuyển sang màn hình Album
Listview Songs	Hiện thị tất cả bài hát của nghệ sĩ	- Click	Chuyển sang màn hình PlayingNow để phát bài nhạc đó

1.11 Chức năng Shuffle all trên thanh menu (Phạm Lê Minh Phú)

1.11.1 Tổng quan

Cần có một phím tắt để phát nhanh tất cả các bài nhạc. Đây cũng là thói quen của người dùng.



Hình 18 Chức năng Shuffle all trên thanh menu

1.11.2 Yêu cầu hệ thống

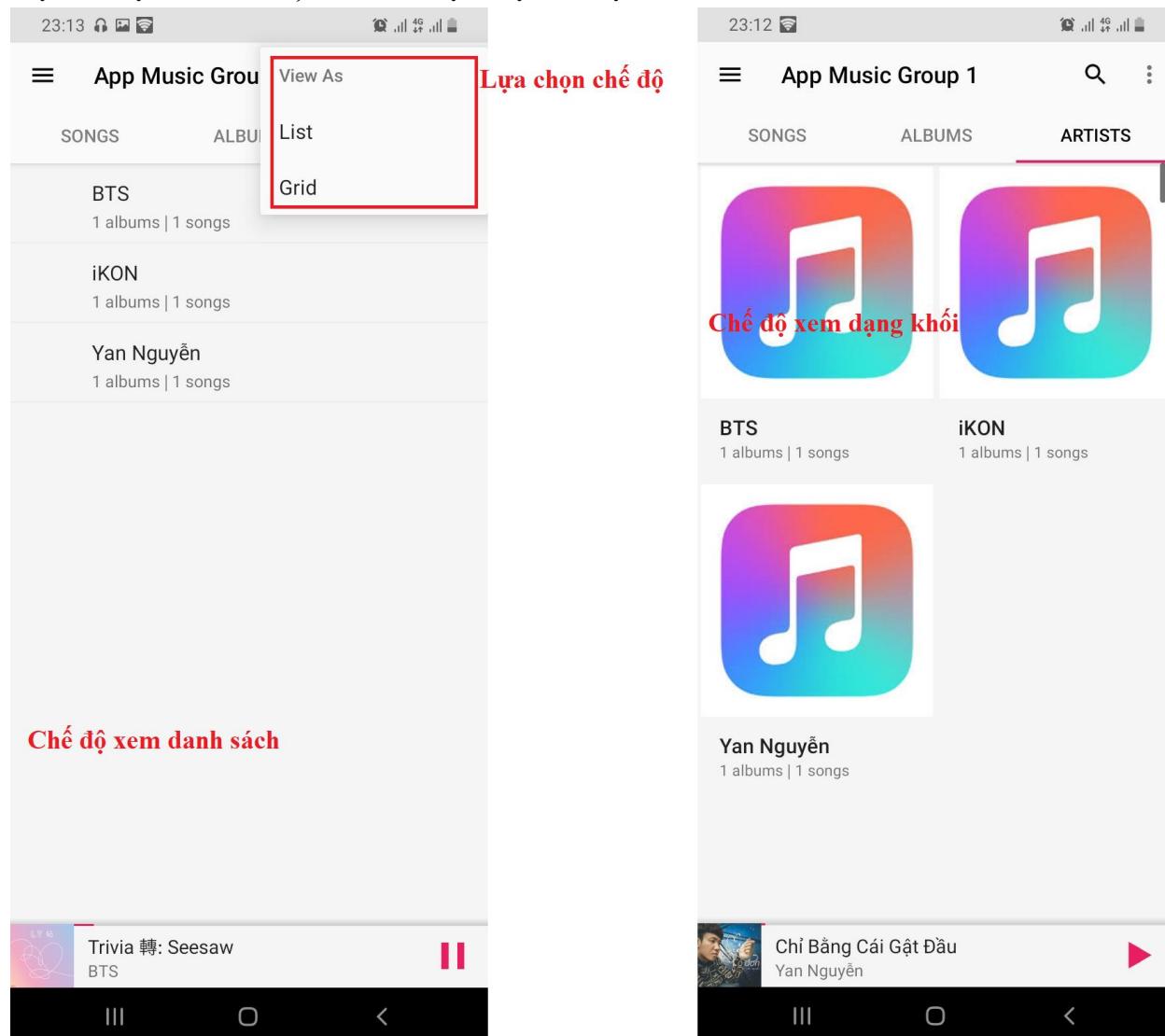
Tiêu đề	Mô tả	Qui trình	Kết quả
Menu Item	Phát tất cả các bài hát	- Click	Tự động phát tất cả các bài hát có trong máy

Playing Now	Nhận tất cả bài hát vào danh sách phát	- Playing	Phát tất cả các bài nhạc
-------------	--	-----------	--------------------------

1.12 Chức năng chế độ xem (View as) trên thanh menu (Phạm Lê Minh Phú)

1.12.1 Tổng quan

Có thể thay đổi chế độ hiện thị ở Abum và Artist là List và Grid. Với List là hiện thị chế độ danh sách, Grid là hiện thị chế độ khối.



Hình 19 Chức năng chế độ xem (View as)

1.12.2 Yêu cầu hệ thống

Tiêu đề	Mô tả	Qui trình	Kết quả
---------	-------	-----------	---------

Menu Item – View as	Chế độ xem	- Click	Chuyển sang menu cấp 2: List và Grid
Menu Item – List	Hiện thị theo dạng danh sách	- Click	Chuyển danh sách hiện tại sang dạng List
Menu Item – Grid	Hiện thị theo dạng khối	- Click	Chuyển danh sách hiện tại sang dạng Grid

2.Thiết kế hệ thống

Tổng quan: Tài liệu dùng để cho nhóm trưởng phân công công việc cho các thành viên giúp phân công công việc 1 cách dễ dàng, bố trí việc làm và phân bổ công việc hợp lý. Mô tả chi tiết công việc cho từng màn hình.

Yêu cầu chức năng:

- a. **Màn hình splash o Giao diện người dùng**



Hình 20 Splash screen

o Yêu cầu chức năng

Tiêu đề	Mô tả	Qui trình	Kết quả
Splash screen	Màn hình Splasd	Sau 3s chuyển màn hình	Chuyển qua màn hình chính

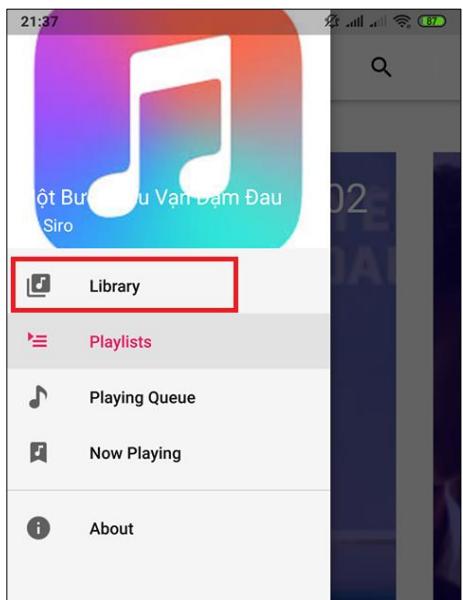
Làm thế nào: Splash screen

- Chạy ứng dụng
 - Chương trình sẽ chạy màn hình Splash
- Sau 3s sẽ tự động chuyển qua màn hình chính

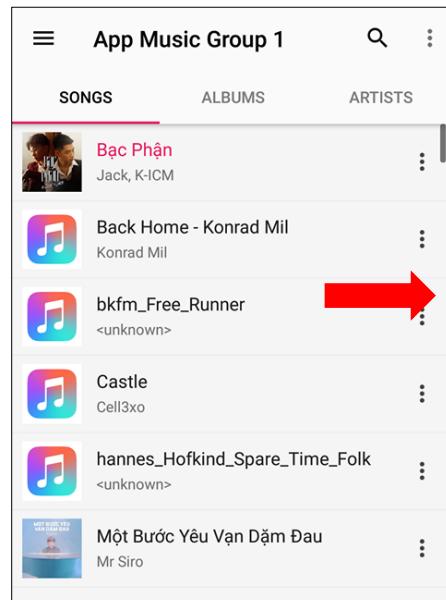
2.1 Màn hình Navigation Drawer (Nguyễn Văn Phúc)

2.1.1 Library

Chọn Library sẽ được chuyển đến danh mục chính gồm (Songs, Albums, Artists)



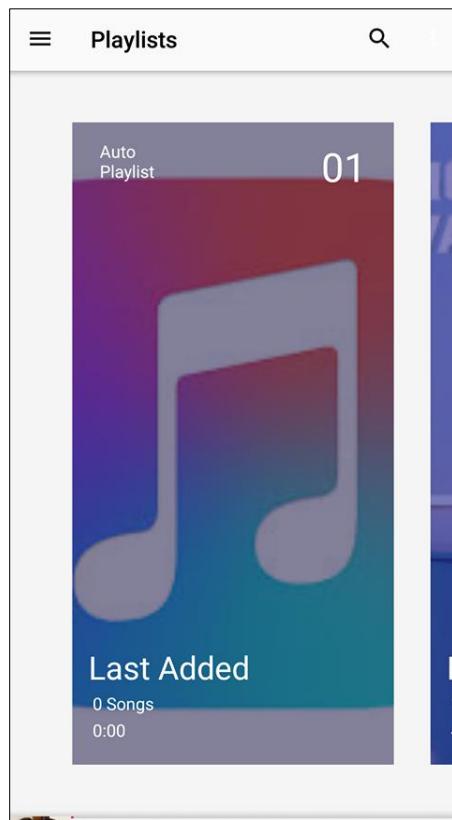
Hình 21 Chọn Library



Hình 22 Giao diện Library

2.1.2 Playlists

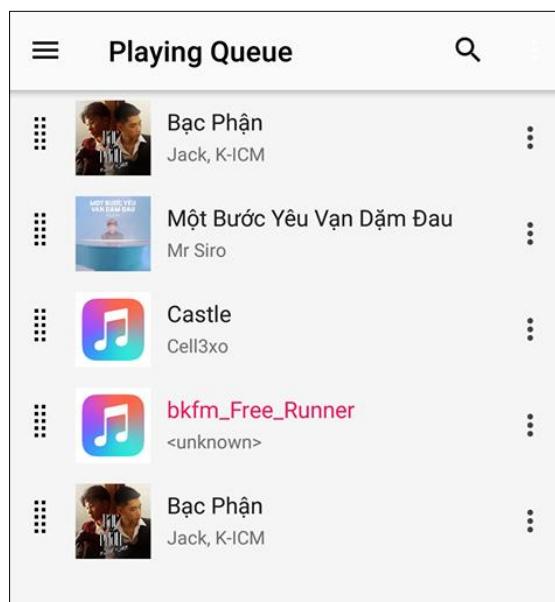
Chọn Playlist sẽ được chuyển đến giao diện Playlist



Hình 23 Giao diện Playlist

2.1.3 Playing Queue

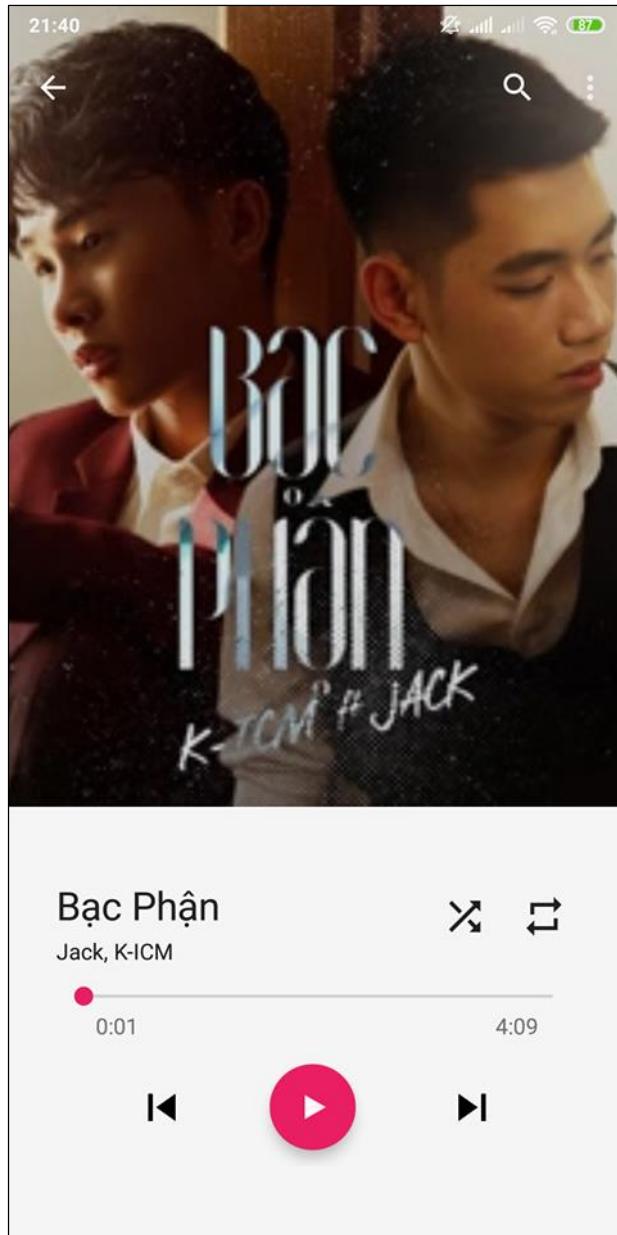
Chọn Playing Queue sẽ được chuyển sang giao diện



Hình 24 Giao diện Playing Queue

2.1.4 Now Playing

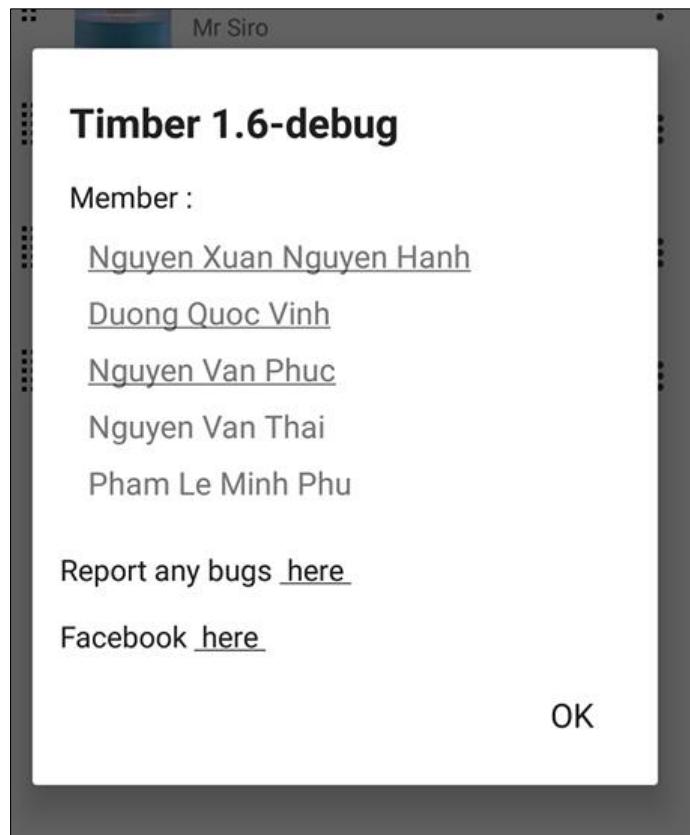
Chọn Now Playing để chuyển sang giao diện Now playing (Phát nhạc)



Hình 25 Giao diện phát nhạc

2.1.5 About

Chọn About để chuyển đến màn hình giới thiệu

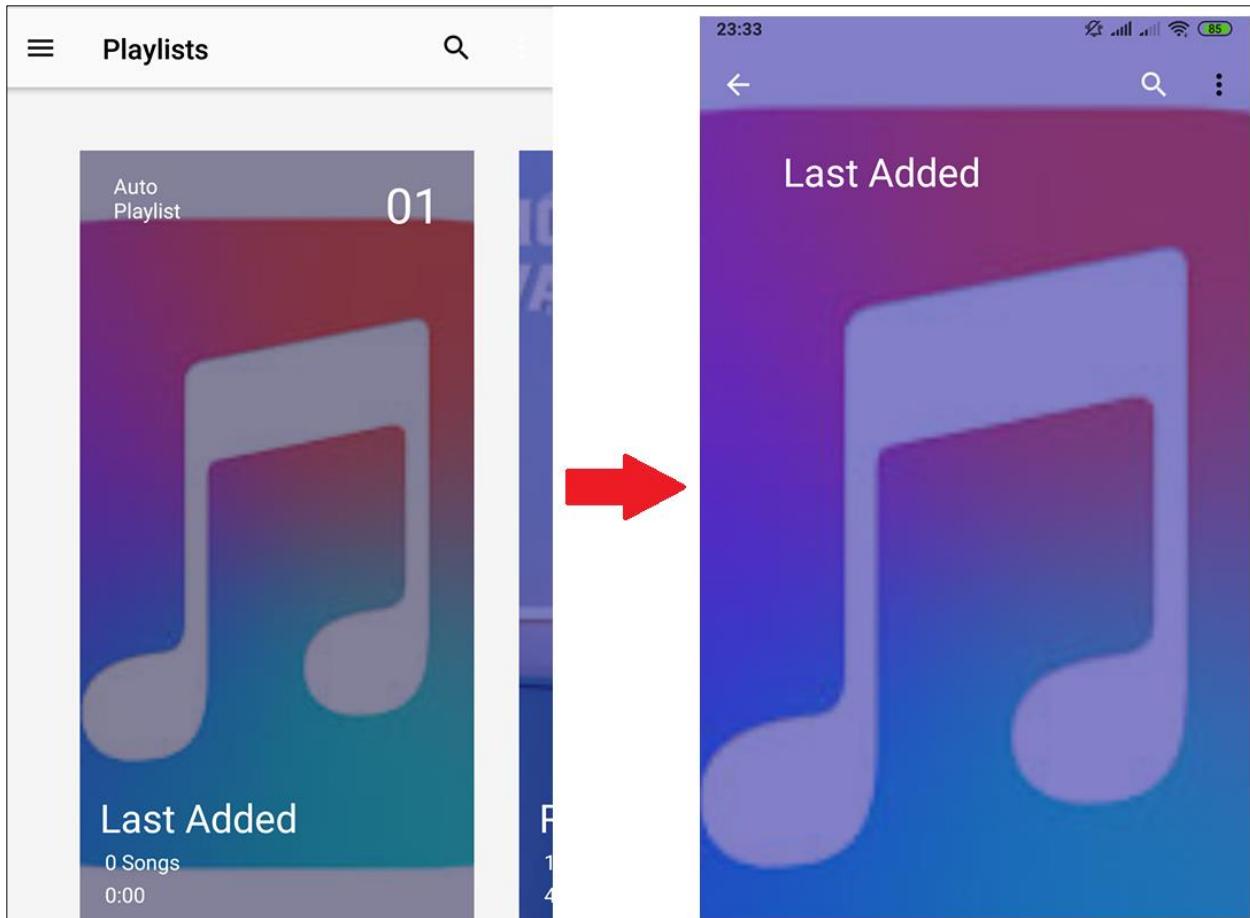


Hình 26 Màn hình giới thiệu

2.2 Màn hình Playlists (Nguyễn Văn Phúc)

2.2.1 Last Added

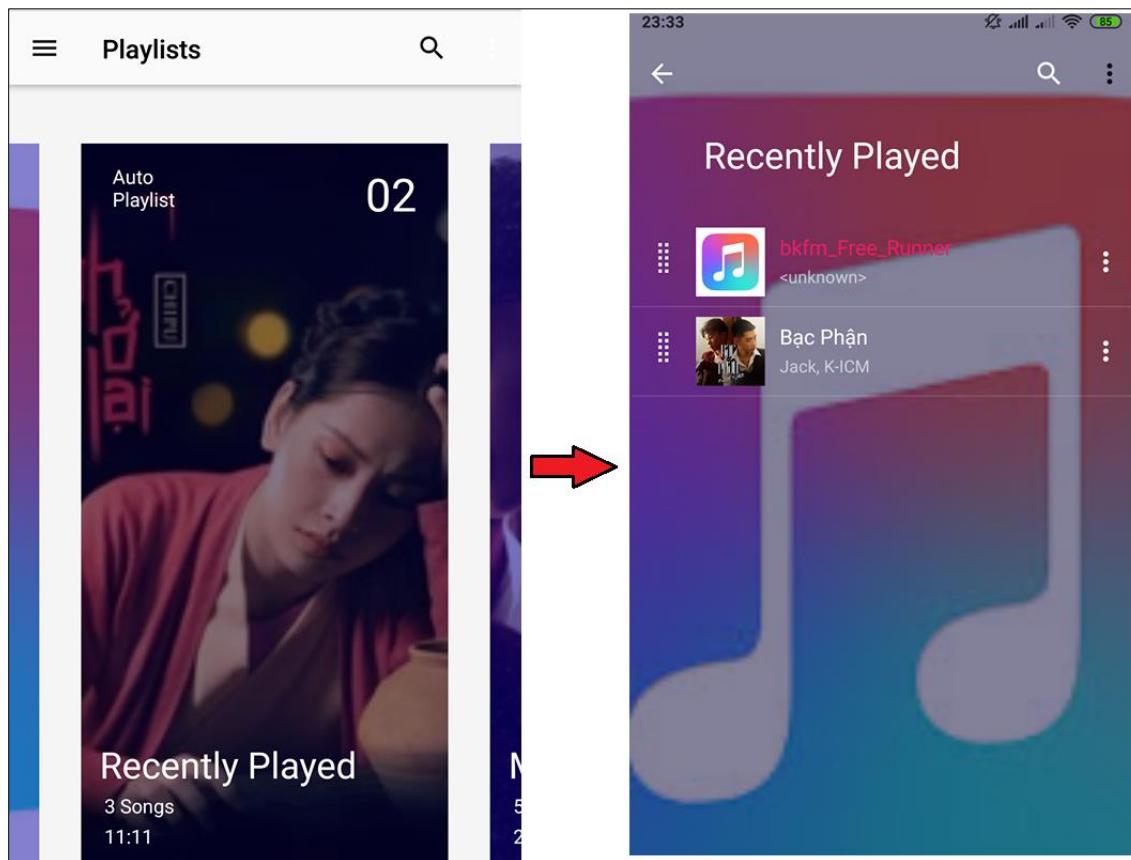
Chọn vào Item RecyclerView Last Added sẽ được chuyển vào Giao diện Last Added



Hình 27 Giao diện Last Added

2.2.2 Recently Played

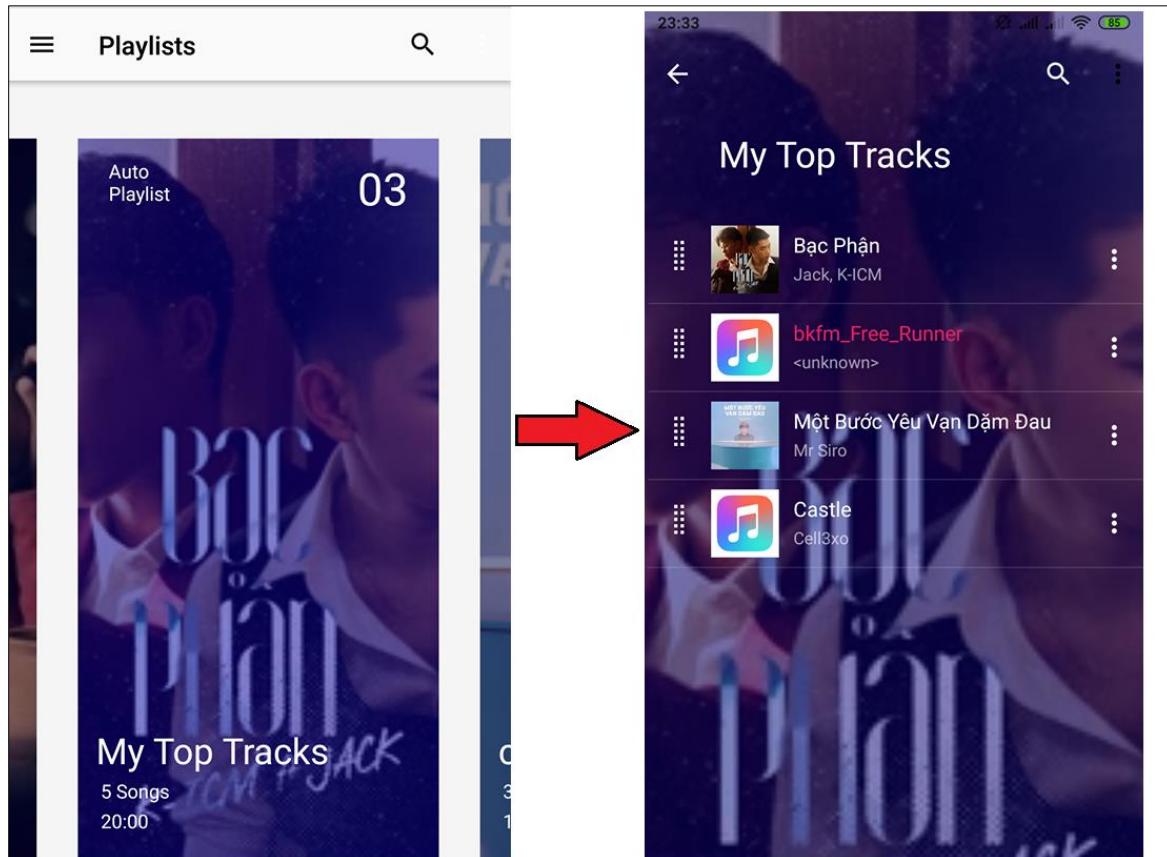
Chọn vào Item Recycler View Recentlt Played sẽ được chuyển vào Giao diện Recentlt Played



Hình 28 Giao diện Recently Played

2.2.3 My top tracks

Chọn Item RecyclerView My Top Tracks sẽ chuyển tới giao diện My Top Tracks

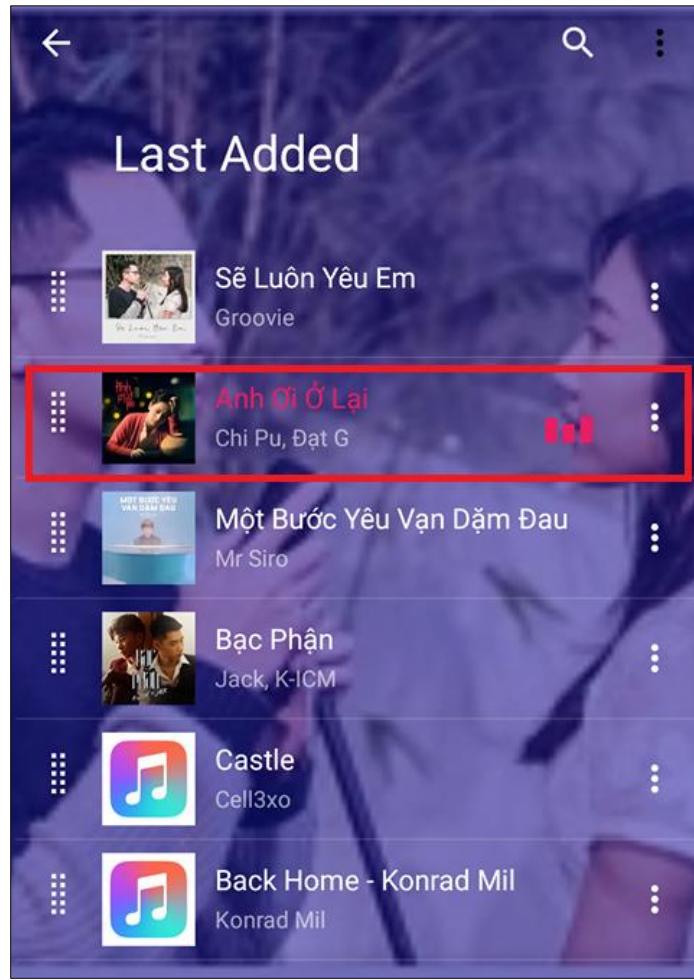


Hình 29 Giao diện My Top Tracks

2.3 Màn hình chi tiết Playlists (Nguyễn Văn Phúc)

2.3.1 Chọn phát nhạc

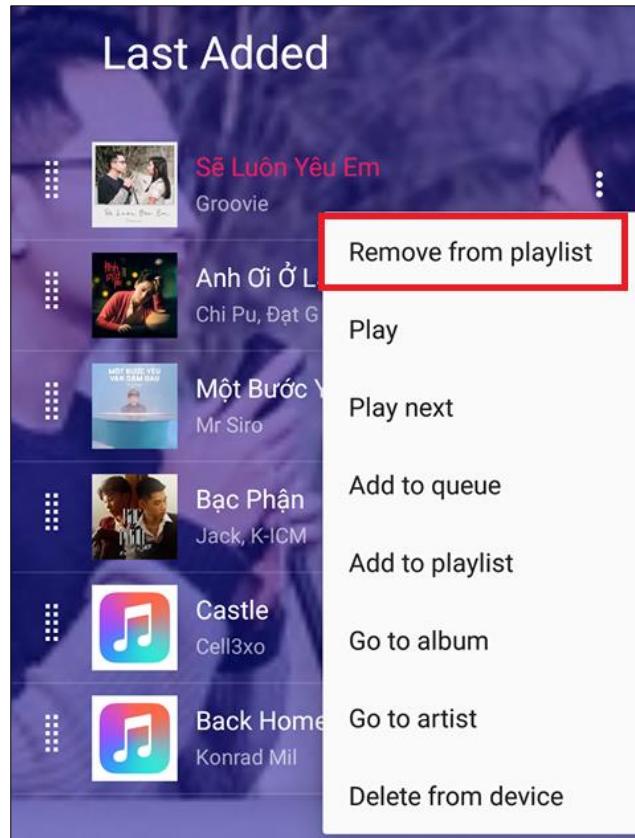
Chọn bài hát cần phát



Hình 30 Phát bài hát

2.3.2 Item Menu Remove from playlist

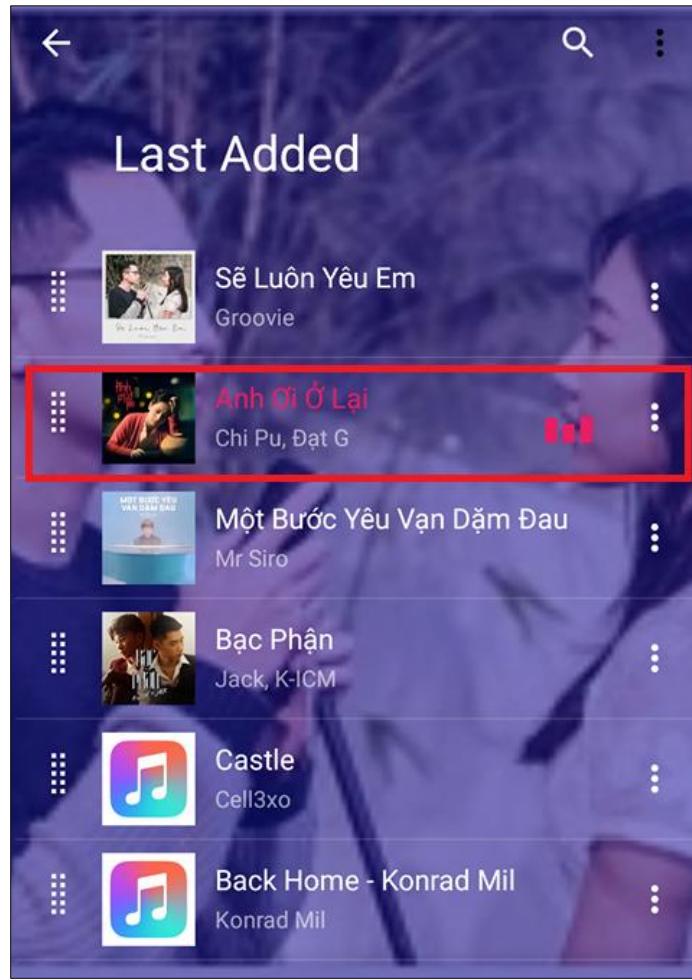
Trên menu chọn Remove from playlist bài hát sẽ xóa khỏi playlist



Hình 31 Xóa bài hát khỏi playlist

2.3.3 Item Menu play

Chọn Play trên menu bài hát sẽ được phát



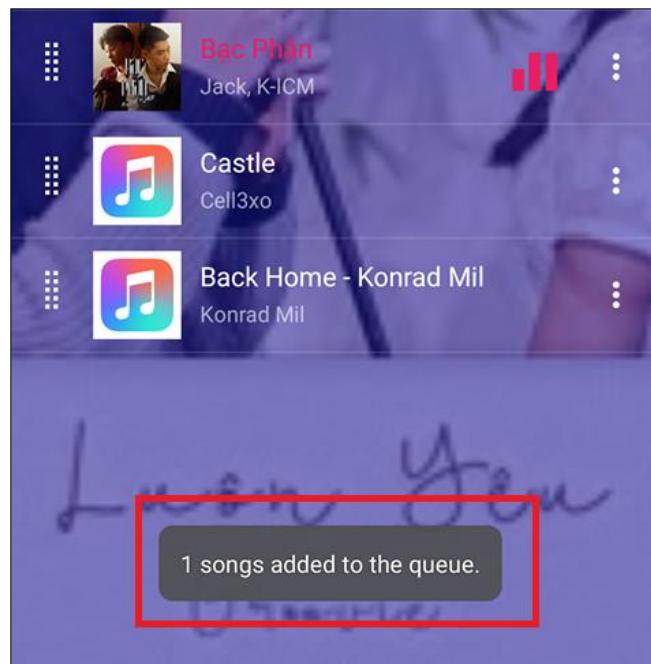
Hình 32 Phát bài hát

2.3.4 Item Menu play next

Chọn Play next trên menu bài hát được chọn sẽ phát sau khi bài hát hiện tại kết thúc

2.3.5 Item Menu Addto queue

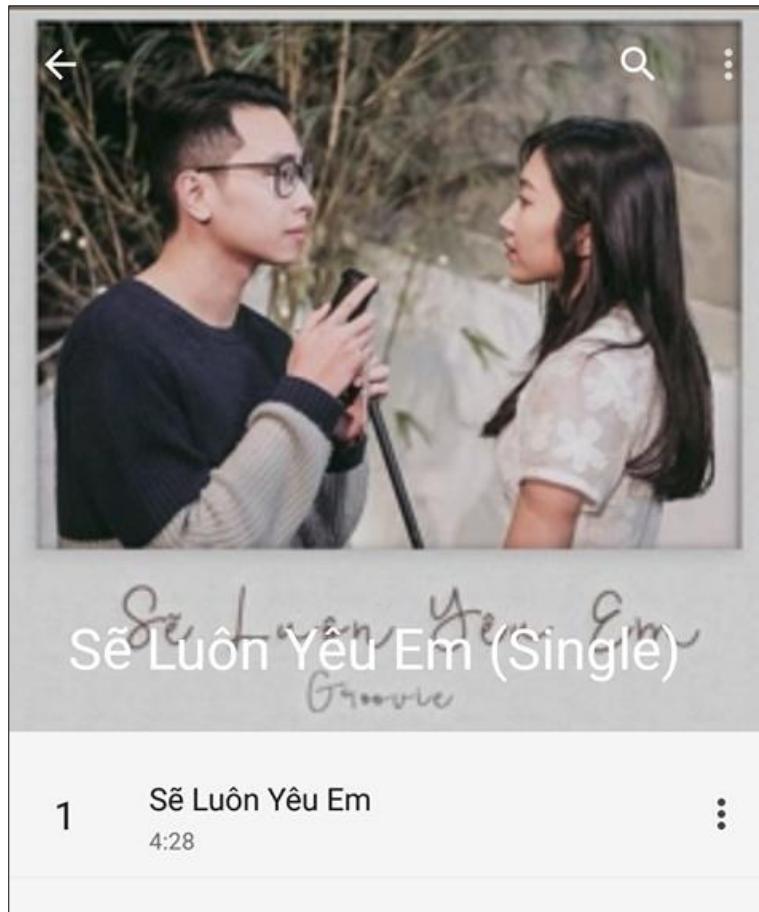
Chọn Add to queue trên menu bài hát được chọn sẽ thêm vào danh sách đợi phát



Hình 33 Bài hát được thêm vào danh sách đợi

2.3.6 Item Menu Go to Album

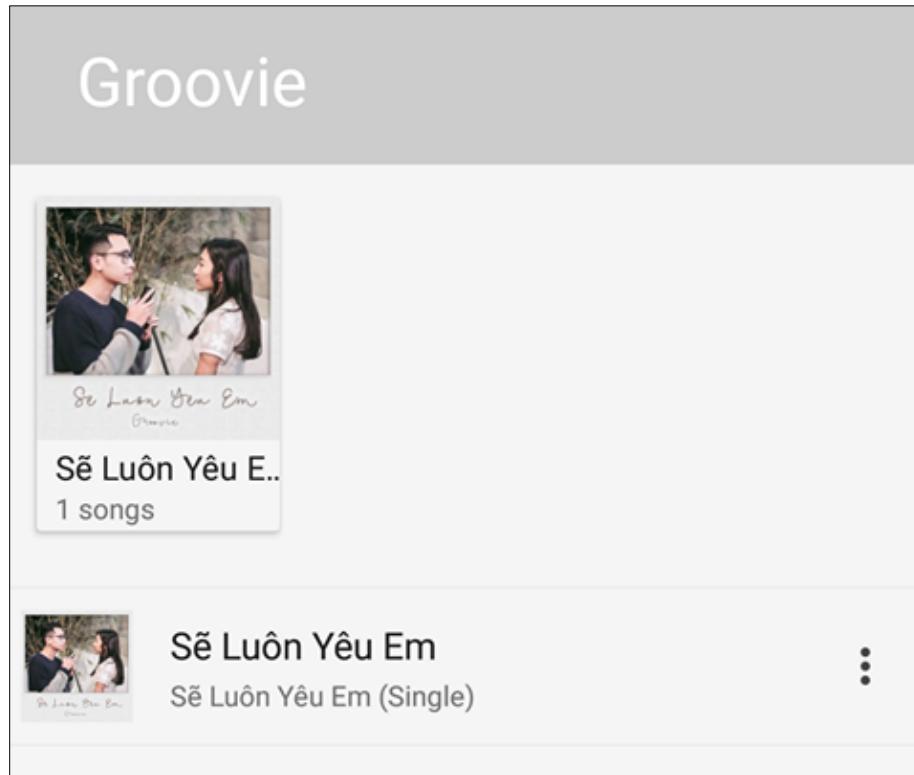
Bài hát được chọn sẽ chuyển đến màn hình album có bài hát được chọn



Hình 3410 Album bài hát

2.3.7 Item Menu Go to Artists

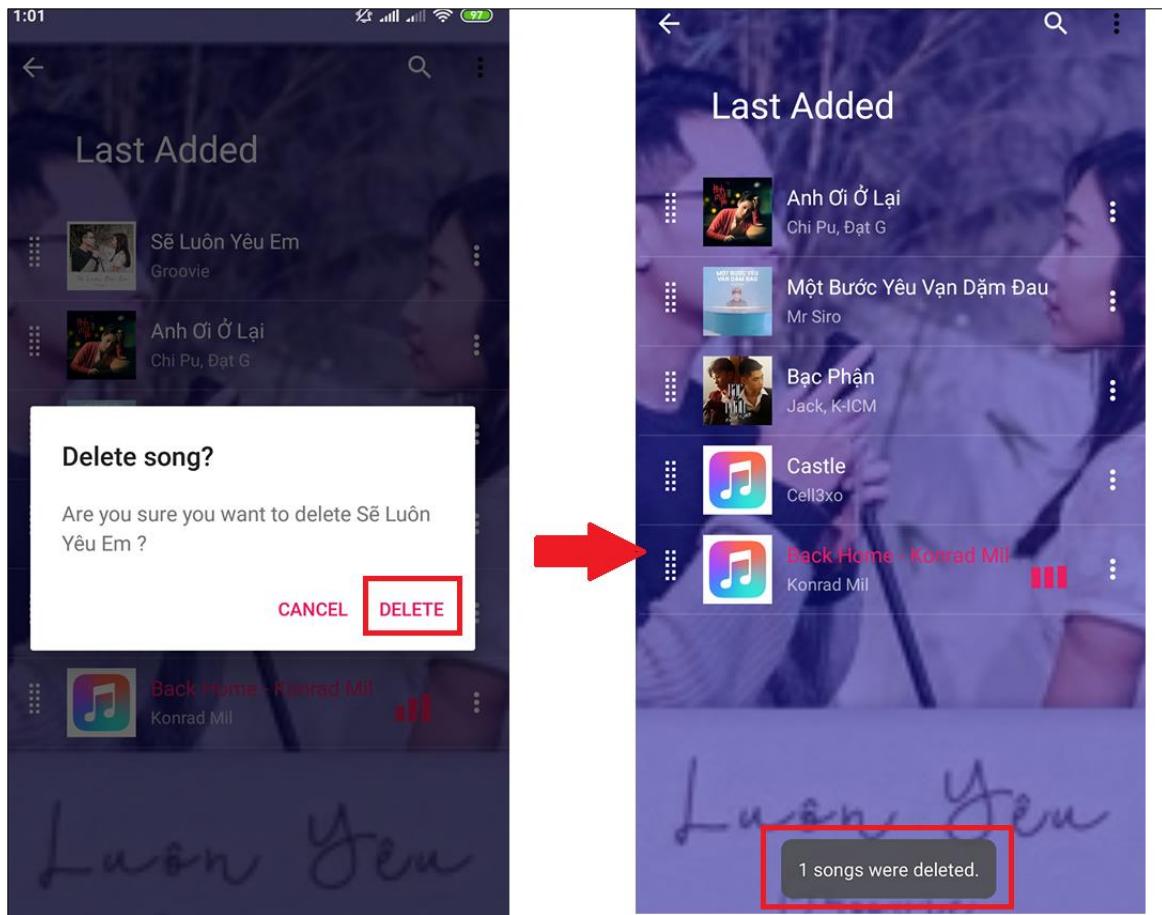
Bài hát được chọn sẽ chuyển đến màn hình ca sĩ thể hiện bài hát được chọn



Hình 35 Ca sĩ bài hát được chọn

2.3.8 Item Menu Delete form device

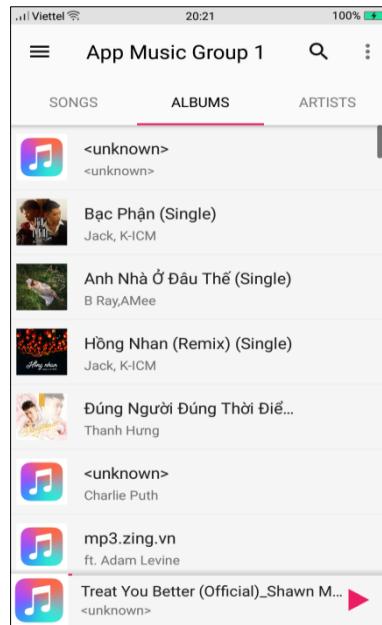
Bài hát được chọn sẽ xóa khỏi thiết bị người dùng khi người dùng chọn và xác nhận



Hình 36 Chọn xóa và xác nhận

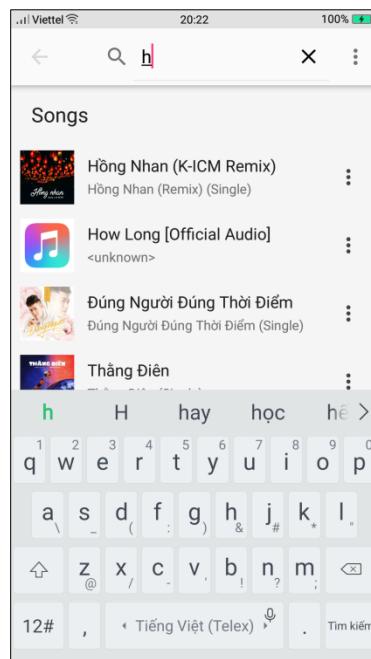
2.4 Giao diện danh mục Ablums (Nguyễn Xuân Nguyên Hạnh)

2.4.1 Giao diện chính màn hình Albums



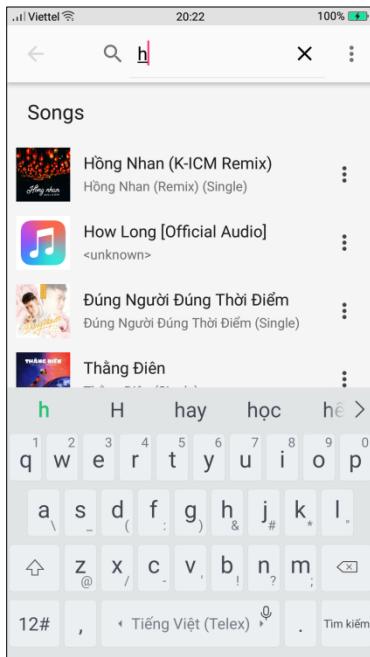
Hình 37 Danh sách bài hát trong mục Albums

- Hiện danh sách bài hát trong từng mục Albums, ngoài ra còn có MenuItem Search và danh sách MenuItem trong MenuItem Sort by.

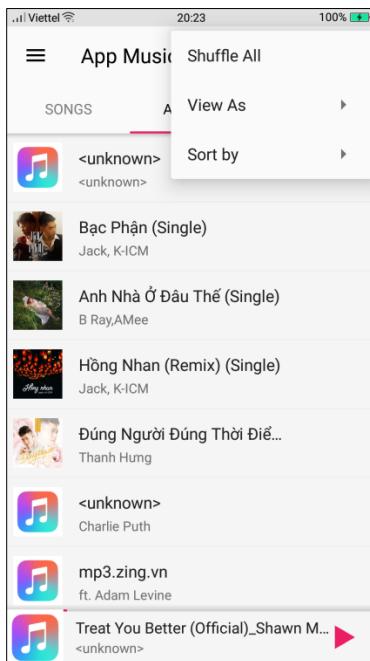


Hình 38 Danh sách bài hát trong mục Albums

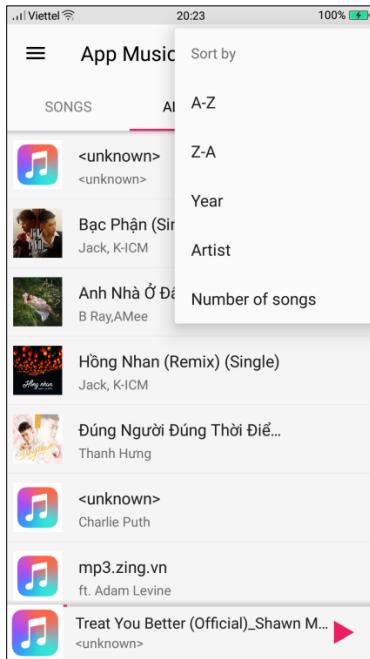
- *Chức năng tìm kiếm:* Nhập keyword để search, sau khi nhập keyword tìm kiếm bài hát, ca sỹ, sau khi search với từ khóa search được thông tin bài hát ca và màn hình sẽ tự chuyển động sang màn hình Sóng thì trên thanh công cụ tìm kiếm menu tự nhiên xuất hiện Item 'X' hủy.



Hình 39 Màn hình chính Albums với các chức năng tìm kiếm



Hình 40 Màn hình chính Albums với các chức năng sắp xếp



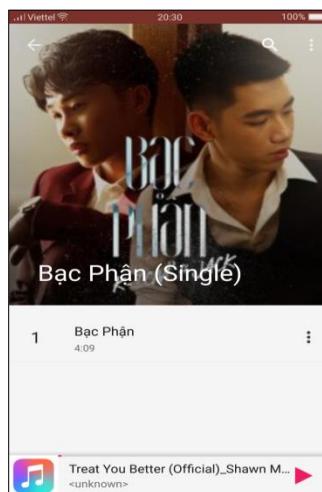
Hình 11 Màn hình chính Albums với các chức năng sắp xếp

- *Chức năng sắp xếp trong mục Albums* danh mục Item sắp xếp: *Sắp xếp thông tin, bài hát ở đâu theo tên bài hát, tên tác giả* theo kí tự từ A-Z và Z-A, ngoài ra còn những itemMenu khác như *sắp xếp theo Năm bài hát, số lượng bài hát và sắp xếp theo ca sĩ*

2.5 Màn hình bài hát trong Albums (Nguyễn Xuân Nguyên Hạnh)

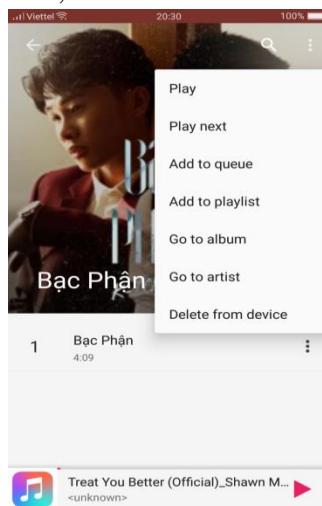
2.5.1 Giao diện chính màn hình chi tiết trong Albums

- Màn hình chi tiết của mục Album (danh sách bài hát trong mục Albums), ở màn hình này mọi thông tin về tất cả bài hát trong 1 mục Album nào đó sẽ hiện lên hết: số lượng bài hát trong 1 Albums, tên bài hát, thời gian của bài hát đó, tên Album, hình ảnh của Album,...



Hình 12 Màn hình chi tiết của Albums

- Ngoài ra ở màn hình này còn 1 số chức năng như: Di chuyển màn hình đến Albums có chứa bài hát này, Di chuyển màn hình đến màn hình của Ca sĩ chưa bài hát đó, thêm bài hát vào Playlist, thêm bài hát vào mục Playing Queue, xóa bài hát khỏi thiết bị, phát nhạc, phát bài kế đó,...



Hình 13 Màn hình chi tiết Album với nhiều chức năng khác.

- Play: Chuyển tới màn hình phát nhạc(Playing now) phát bài hát đó.
- PlayNext: Chuyển tới màn hình phát nhạc nhưng mà phát bài hát kế tiếp đó trong Album.
- Add to queue: Thêm bài hát vào mục Playing Queue.
- Add to Playlist: Thêm bài hát vào mục Playlist.
- Go to Album: Di chuyển màn hình tới mục Album có chứa bài hát đó.
- Go to Artist: Di chuyển màn hình tới mục Artist có chứa bài hát đó.
- Delete from devices: Xóa bài hát đó ra khỏi an toàn

2.9 Màn hình phát nhạc (Đương Quốc Vĩnh)

Yêu cầu chức năng:

- a. **Màn hình phát nhạc**
 - o **Giao diện người dùng**



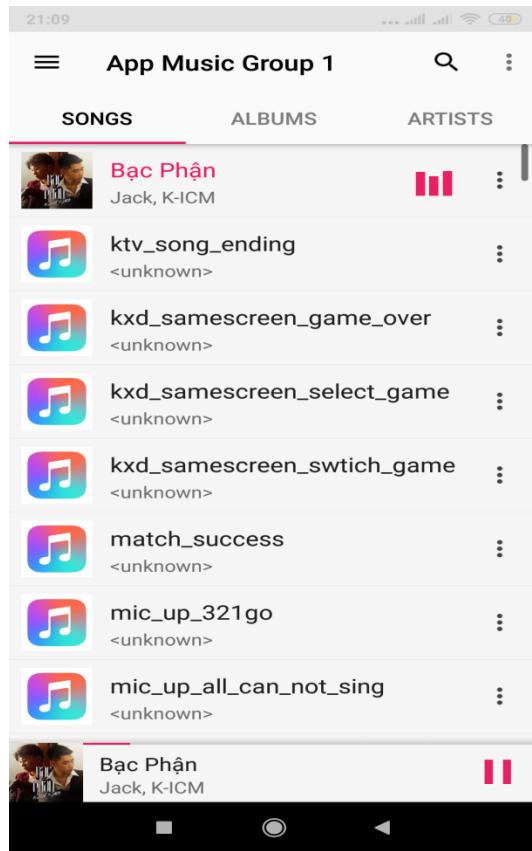
Hình 14 màn hình phát nhạc

Làm thế nào: Màn hình phát nhạc

- Chạy ứng dụng, nó sẽ load các bài hát từ bộ nhớ trong của thiết bị
- Khi click vào 1 bài hát bất kì sẽ tự động chuyển sang màn hình phát nhạc
- Xử lý các button khi click:
 - o Button pause: dừng phát nhạc
 - o Button Play: phát nhạc
 - o Button Next: Chuyển sang bài hát tiếp theo trong danh sách phát
 - o Button Pre: Lùi lại bài hát trước
 - o Button icon: Lặp lại bài hát một lần, hoặc lặp lại vô hạn.

b. Màn hình phát nhạc khi chuyển màn hình

o **Giao diện người dùng**



Hình 15 màn hình phát nhạc

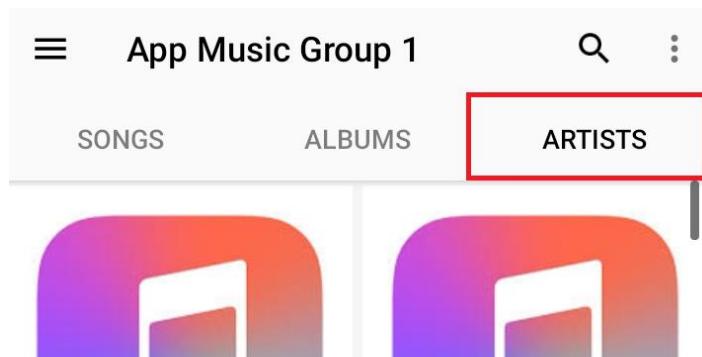
Làm thế nào: Màn hình phát nhạc

- Khi click vào 1 bài hát bất kì sẽ tự động chuyển sang màn hình phát nhạc
- Sau đó bấm quay lại để chuyển sang màn hình khác bất kì.
- Sự kiện các button khi click:
 - o Button pause: dừng phát nhạc
 - o Button Play: phát nhạc

2.10 Giao diện màn hình danh sách nghệ sĩ – Artist (Phạm Lê Minh Phú)

2.10.1 Chọn chế độ Artists

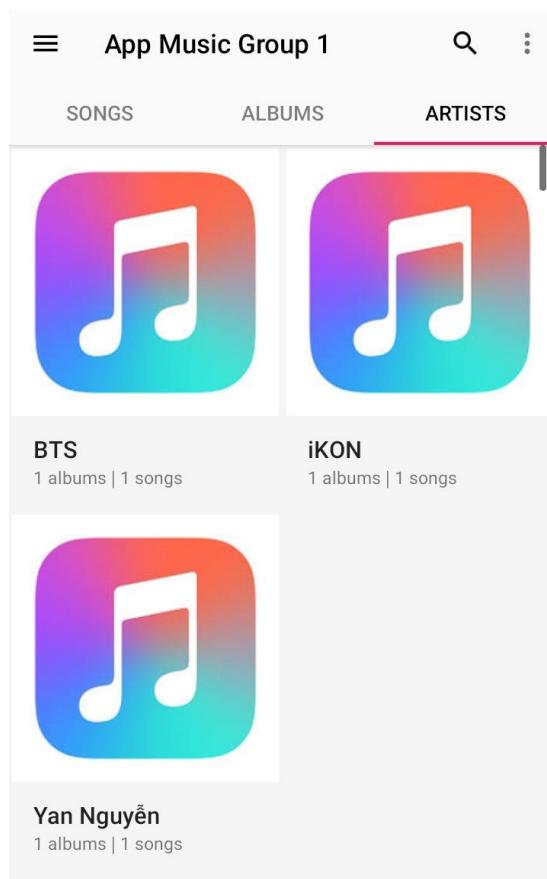
Từ màn hình chính ban đầu, Chọn ARTISTS trên thanh menu



Hình 16 Chọn chế độ Artists

2.10.2 Xem chi tiết Artist

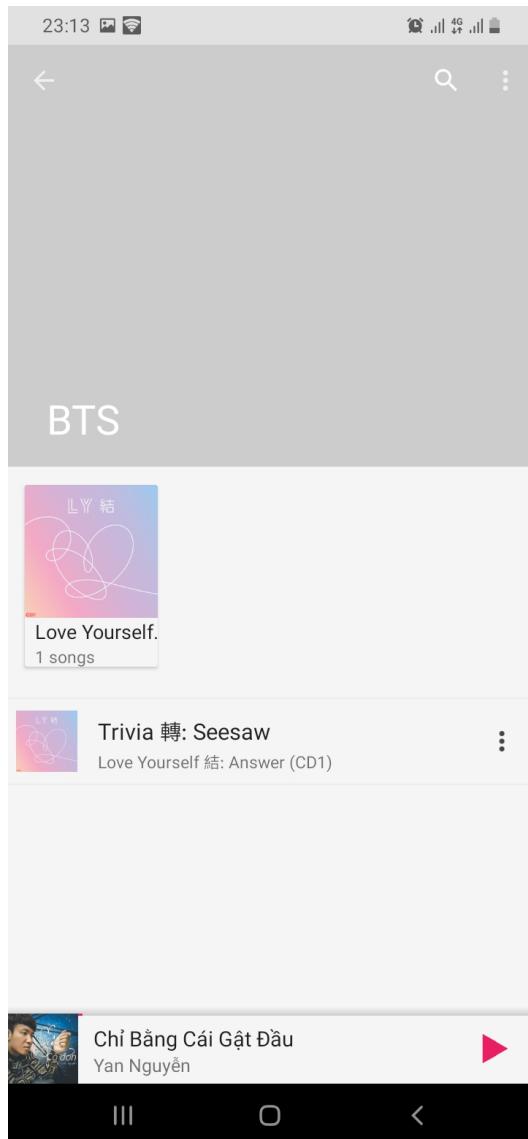
Chọn vào một nghệ sĩ muốn xem chi tiết.



Hình 17 Xem chi tiết Artist

2.10.3 Màn hình chi tiết Artist

Chọn Album để xem Album của nghệ sĩ đó, chọn vào bài hát để phát bài hát của nghệ sĩ đó

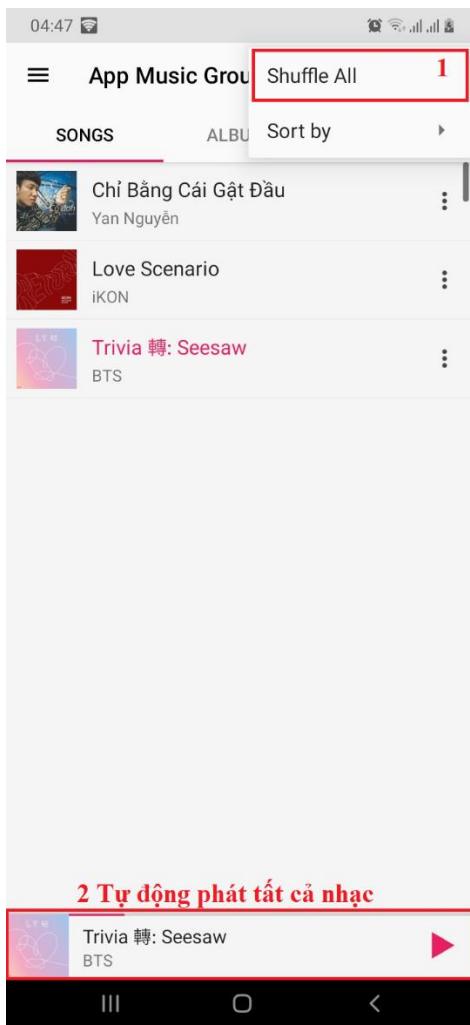


Hình 48 Màn hình chi tiết Artist

2.11 Chức năng Shuffle all trên thanh menu (Phạm Lê Minh Phú)

2.11.1 Chọn Shuffle all – Phát tất cả nhạc

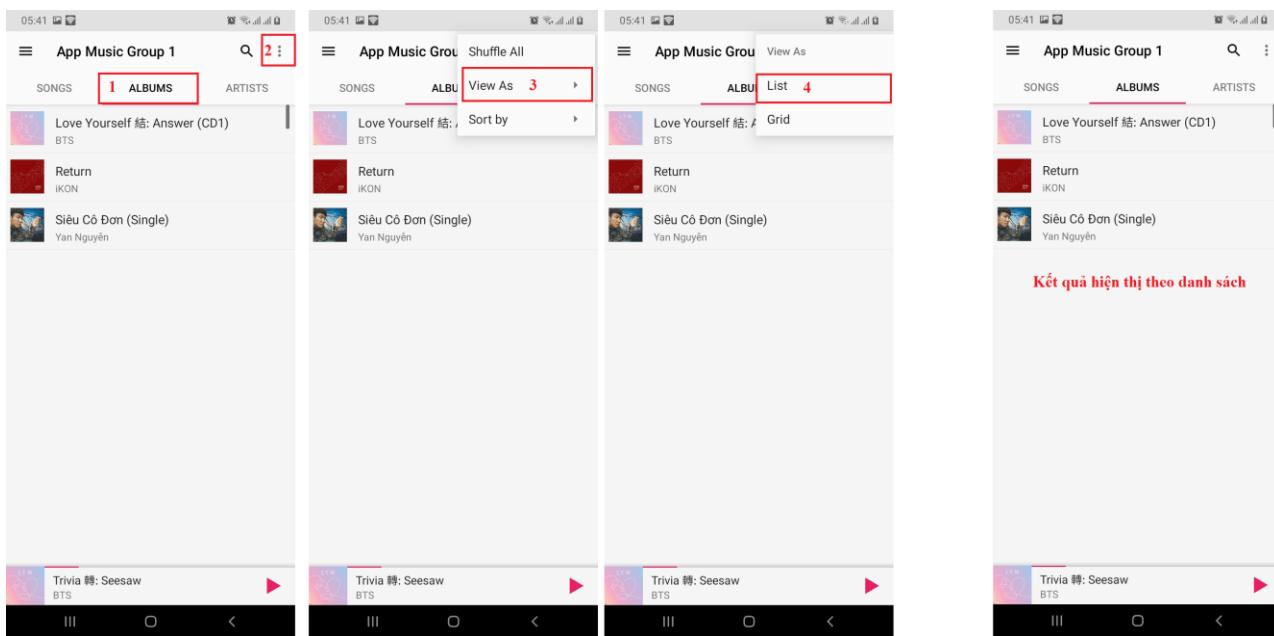
Trên thanh menu chọn Shuffle all để phát tất cả nhạc.

*Hình 18 Chọn Shuffle all*

2.12 Chức năng chế độ xem (View as) trên thanh menu (Phạm Lê Minh Phú)

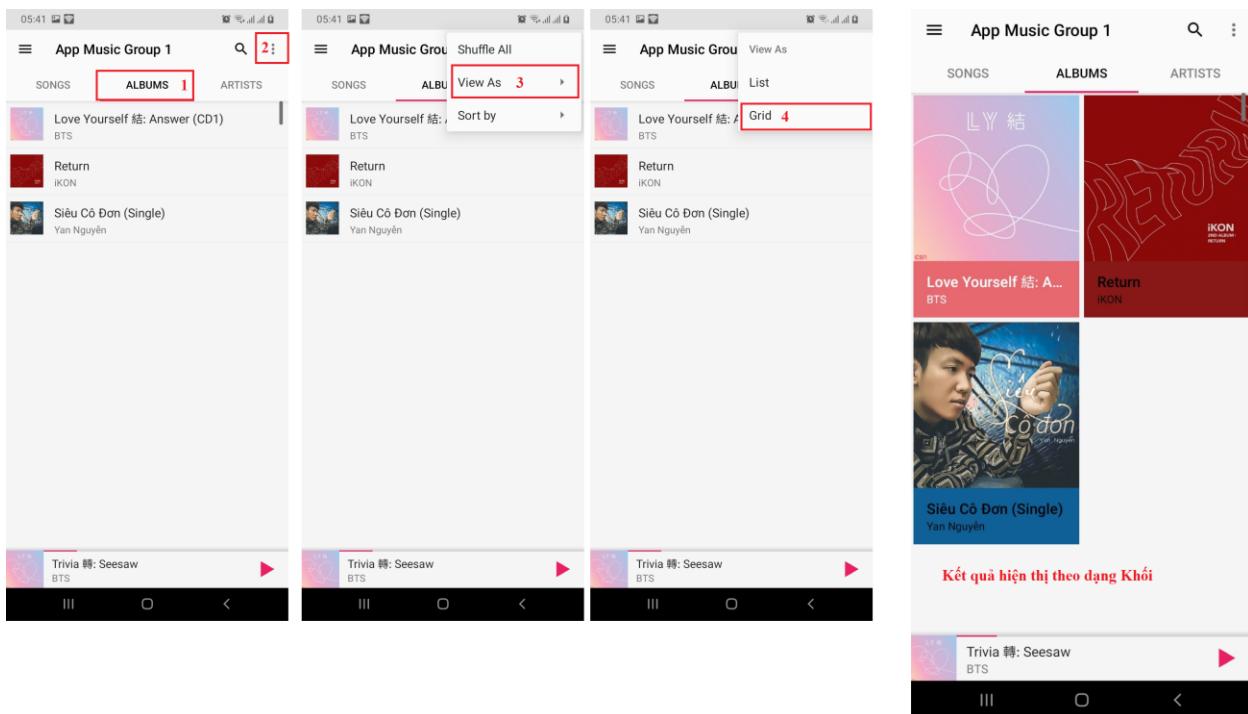
2.12.1 Chọn chế độ xem cho Album

Cách xem dạng list



Hình 19 Chọn chế độ xem cho Abum - Cách xem dạng List

Cách xem dạng Khối Grid



Hình 20 Chọn chế độ xem cho Abum - Cách xem dạng Khối Grid

CHƯƠNG III. CÀI ĐẶT

3.1 Màn hình Navigation Drawer (Nguyễn Văn Phúc)

3.1.1 Navigation Drawer



```
<?xml version="1.0" encoding="utf-8" ?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <group
        android:id="@+id/group1"
        android:checkableBehavior="single">
        <item
            android:id="@+id/nav_library"
            android:title="Library" />
        <item
            android:id="@+id/nav_playlists"
            android:title="Playlists" />
        <item
            android:id="@+id/nav_queue"
            android:title="Playing Queue" />
        <item
            android:id="@+id/nav_nowplaying"
            android:checkable="false"
            android:title="Now Playing" />
    </group>

    <group
        android:id="@+id/group2"
        android:checkableBehavior="none">
        <item
            android:id="@+id/nav_about"
            android:title="@string/about" />
    </group>
</menu>
```

Hình 21 Code Navigation Drawer

3.1.2 Xử lý chuyển sang màn hình (Library, Playlist, Queue, Playing, About)

```

private void updatePosition(final MenuItem menuItem) {
    runnable = null;

    switch (menuItem.getItemId()) {
        case R.id.nav_library:
            runnable = navigateLibrary;
            break;

        case R.id.nav_playlists:
            runnable = navigatePlaylist;
            break;

        case R.id.nav_nowplaying:
            if (getCastSession() != null) {
                startActivity(new Intent(packageContext: MainActivity.this, ExpandedControlsActivity.class));
            } else {
                NavigationUtils.navigateNowplaying(context: MainActivity.this, withAnimations: false);
            }
            break;

        case R.id.nav_queue:
            runnable = navigateQueue;
            break;

        case R.id.nav_about:
            mDrawerLayout.closeDrawers();
            Handler handler = new Handler();
            handler.postDelayed(() -> {
                Helpers.showAbout(activity: MainActivity.this);
            }, delayMillis: 350);

            break;
    }

    if (runnable != null) {
        menuItem.setChecked(true);
        mDrawerLayout.closeDrawers();
        Handler handler = new Handler();
        handler.postDelayed(() -> { runnable.run(); }, delayMillis: 350);
    }
}

```

Hình 22 Code xử lý chuyển màn hình

3.2 Màn hình Playlists (Nguyễn Văn Phúc)

3.2.1 Code xử lý Item RecyclerView

Code xử lý chọn vào Item RecyclerView để chuyển vào màn hình tương ứng

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_playlist_detail);

    action = getIntent().getAction();

    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setTitle("");

    playlistsMap.put(Constants.NAVIGATE_PLAYLIST_LASTADDED, playlistLastAdded);
    playlistsMap.put(Constants.NAVIGATE_PLAYLIST_RECENT, playlistRecents);
    playlistsMap.put(Constants.NAVIGATE_PLAYLIST_TOPTRACKS, playlistToptracks);
    playlistsMap.put(Constants.NAVIGATE_PLAYLIST_USERCREATED, playlistUsercreated);

    recyclerView = (RecyclerView) findViewById(R.id.recyclerview);
    blurFrame = (ImageView) findViewById(R.id.blurFrame);
    playlistname = (TextView) findViewById(R.id.name);
    foreground = findViewById(R.id.foreground);

    recyclerView.setLayoutManager(new LinearLayoutManager(context: this));

    setAlbumart();

    animate = getIntent().getBooleanExtra(Constants.ACTIVITY_TRANSITION, defaultValue: false);
    if (animate && TimberUtils.isLollipop()) {
        getWindow().getEnterTransition().addListener(new EnterTransitionListener());
    } else {
        setUpSongs();
    }
}

```

Hình 23 Code xử lý RecyclerView

3.2.2 Load bài hát từ bộ nhớ

```

private void loadEverything() {
    Runnable navigation = navigationMap.get(action);
    if (navigation != null) {
        navigation.run();
    } else {
        navigateLibrary.run();
    }
    new initQuickControls().execute("");
}

private void checkPermissionAndThenLoad() {
    //check for permission
    if (Nammu.checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) && Nammu.checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
        loadEverything();
    } else {
        if (Nammu.shouldShowRequestPermissionRationale(activity: this, Manifest.permission.READ_EXTERNAL_STORAGE)) {
            Snackbar.make(panelLayout, text: "Timber will need to read external storage to display songs on your device.",
                Snackbar.LENGTH_INDEFINITE)
                .setAction(text: "OK", (OnClickListener) (view) -> {
                    Nammu.askForPermission(activity: MainActivity.this, new String[]{Manifest.permission.READ_EXTERNAL_STORAGE, Manifest.permission.WRITE_EXTERNAL_STORAGE}, permissionReadstorageCallback)
                }).show();
        } else {
            Nammu.askForPermission(activity: this, new String[]{Manifest.permission.READ_EXTERNAL_STORAGE, Manifest.permission.WRITE_EXTERNAL_STORAGE}, permissionReadstorageCallback);
        }
    }
}

```

Hình 24 Code Load bài hát từ bộ nhớ

3.3 Màn hình chi tiết Playlists (Nguyễn Văn Phúc)

3.3.1 Code xử lý phát nhạc

Click vào bài hát

```

public static final boolean isPlaying() {
    if (mService != null) {
        try {
            return mService.isPlaying();
        } catch (final RemoteException ignored) {
        }
    }
    return false;
}

```

Hình 25 Code play nhạc

3.3.2 Code xử lý xóa bài hát khỏi play list

Click vào bài hát cần xóa khỏi playlist

```

menu.setOnMenuItemClickListener((item) -> {
    switch (item.getItemId()) {
        case R.id.popup_song_remove_playlist:
            TimberUtils.removeFromPlaylist(mContext, arraylist.get(position).id, playlistId);
            removeSongAt(position);
            notifyItemRemoved(position);
            break;

        @Override
        public void removeSongAt(int i) {
            arraylist.remove(i);
            updateDataSet(arraylist);
        }
    }
}

```

Hình 26 Code xử lý xóa bài hát khỏi playlist

3.3.3 Code xử lý chuyển đến album bài hát

Click vào bài hát cần xem album

```

private void setOnPopupMenuListener(ItemHolder itemHolder, final int position) {
    itemHolder.menu.setOnClickListener((v) -> {
        final PopupMenu menu = new PopupMenu(mContext, v);
        menu.setOnMenuItemClickListener((item) -> {
            switch (item.getItemId()) {
                case R.id.popup_song_play:
                    MusicPlayer.playAll(mContext, songIDs, position, sourceld: -1, TimberUtils.IdType.NA, forceShuffle: false);
                    break;
                case R.id.popup_song_play_next:
                    long[] ids = new long[1];
                    ids[0] = arraylist.get(position).id;
                    MusicPlayer.playNext(mContext, ids, sourceld: -1, TimberUtils.IdType.NA);
                    break;
                case R.id.popup_song_goto_album:
                    NavigationUtils.goToAlbum(mContext, arraylist.get(position).albumId);
                    break;
                case R.id.popup_song_goto_artist:
                    NavigationUtils.goToArtist(mContext, arraylist.get(position).artistId);
                    break;
                case R.id.popup_song_addto_queue:
                    long[] id = new long[1];
                    id[0] = arraylist.get(position).id;
                    MusicPlayer.addToQueue(mContext, id, sourceld: -1, TimberUtils.IdType.NA);
                    break;
                case R.id.popup_song_addto_playlist:
                    AddPlaylistDialog.newInstance(arraylist.get(position)).show(((AppCompatActivity) mContext).getSupportFragmentManager(), tag: "ADD_PLAYLIST");
                    break;
                case R.id.popup_song_delete:
                    long[] deleteIds = {arraylist.get(position).id};
                    TimberUtils.showDeleteDialog(mContext, arraylist.get(position).title, deleteIds, adapter: AlbumSongsAdapter.this, position);
                    break;
            }
            return false;
        });
        menu.inflate(R.menu.popup_song);
        menu.show();
    });
}

```

Hình 57 Code xử lý chuyển đến album

3.3.4 Code xử lý chuyển đến ca sĩ thẻ hiện bài hát

Click vào bài hát cần xem ca sĩ

```

private void setOnPopupMenuListener(ItemHolder itemHolder, final int position) {
    itemHolder.menu.setOnClickListener((v) -> {
        final PopupMenu menu = new PopupMenu(mContext, v);
        menu.setOnMenuItemClickListener((item) -> {
            switch (item.getItemId()) {
                case R.id.popup_song_play:
                    MusicPlayer.playAll(mContext, songIDs, position, sourceld: -1, TimberUtils.IdType.NA, forceShuffle: false);
                    break;
                case R.id.popup_song_play_next:
                    long[] ids = new long[1];
                    ids[0] = arraylist.get(position).id;
                    MusicPlayer.playNext(mContext, ids, sourceld: -1, TimberUtils.IdType.NA);
                    break;
                case R.id.popup_song_goto_album:
                    NavigationUtils.goToAlbum(mContext, arraylist.get(position).albumId);
                    break;
                case R.id.popup_song_goto_artist:
                    NavigationUtils.goToArtist(mContext, arraylist.get(position).artistId);
                    break;
                case R.id.popup_song_addto_queue:
                    long[] id = new long[1];
                    id[0] = arraylist.get(position).id;
                    MusicPlayer.addToQueue(mContext, id, sourceld: -1, TimberUtils.IdType.NA);
                    break;
                case R.id.popup_song_addto_playlist:
                    AddPlaylistDialog.newInstance(arraylist.get(position)).show((AppCompatActivity) mContext).getSupportFragmentManager(), tag: "ADD_PLAYLIST");
                    break;
                case R.id.popup_song_delete:
                    long[] deleteIds = {arraylist.get(position).id};
                    TimberUtils.showDeleteDialog(mContext, arraylist.get(position).title, deleteIds, adapter: AlbumSongsAdapter.this, position);
                    break;
            }
            return false;
        });
        menu.inflate(R.menu.popup_song);
        menu.show();
    });
}
}

```

Hình 28. Code xử lý chuyển đến ca sĩ

3.3.5 Code xử lý xóa bài hát khỏi thiết bị

Click vào bài hát cần xóa

```

private void setOnPopupMenuListener(ItemHolder itemHolder, final int position) {
    itemHolder.menu.setOnClickListener((v) -> {
        final PopupMenu menu = new PopupMenu(mContext, v);
        menu.setOnMenuItemClickListener((item) -> {
            switch (item.getItemId()) {
                case R.id.popup_song_play:
                    MusicPlayer.playAll(mContext, songIDs, position, sourceId: -1, TimberUtils.IdType.NA, forceShuffle: false);
                    break;
                case R.id.popup_song_play_next:
                    long[] ids = new long[1];
                    ids[0] = arraylist.get(position).id;
                    MusicPlayer.playNext(mContext, ids, sourceId: -1, TimberUtils.IdType.NA);
                    break;
                case R.id.popup_song_goto_album:
                    NavigationUtils.goToAlbum(mContext, arraylist.get(position).albumId);
                    break;
                case R.id.popup_song_goto_artist:
                    NavigationUtils.goToArtist(mContext, arraylist.get(position).artistId);
                    break;
                case R.id.popup_song_addto_queue:
                    long[] id = new long[1];
                    id[0] = arraylist.get(position).id;
                    MusicPlayer.addToQueue(mContext, id, sourceId: -1, TimberUtils.IdType.NA);
                    break;
                case R.id.popup_song_addto_playlist:
                    AddPlaylistDialog.newInstance(arraylist.get(position)).show(((AppCompatActivity) mContext).getSupportFragmentManager(), tag: "ADD_PLAYLIST");
                    break;
                case R.id.popup_song_delete:
                    long[] deleteIds = {arraylist.get(position).id};
                    TimberUtils.showDeleteDialog(mContext, arraylist.get(position).title, deleteIds, adapter: AlbumSongsAdapter.this, position);
                    break;
            }
        });
        menu.inflate(R.menu.popup_song);
        menu.show();
    });
}
}

```

Hình 28. Code xử lý chuyển đến ca sĩ

3.4 Giao diện danh mục Ablums (Nguyễn Xuân Nguyên Hạnh)

Hàm Load danh sách bài hát:

```

public static ArrayList<Song> getSongsForAlbum(Context context, long albumID)
{
    Cursor cursor = makeAlbumSongCursor(context, albumID);
    ArrayList arrayList = new ArrayList();
    if ((cursor != null) && (cursor.moveToFirst()))
        do {
            long id = cursor.getLong(0);
            String title = cursor.getString(1);
            String artist = cursor.getString(2);
            String album = cursor.getString(3);
            int duration = cursor.getInt(4);
            int trackNumber = cursor.getInt(5);
            /*This fixes bug where some track numbers displayed as 100 or
            200*/
            while (trackNumber >= 1000) {
                trackNumber -= 1000; //When error occurs the track numbers
                have an extra 1000 or 2000 added, so decrease till normal.
            }
            long artistId = cursor.getInt(6);
            long albumId = albumID;
        }
        while (cursor.moveToNext());
    cursor.close();
    return arrayList;
}

```

```

        arrayList.add(new Song(id, albumId, artistId, title, artist,
album, duration, trackNumber));
    }
    while (cursor.moveToNext());
    if (cursor != null)
        cursor.close();
    return arrayList;
}
}

```

Hàm Search bài hát, ca sỹ, albums :

```

@Override
public void onBindViewHolder(final ItemHolder itemHolder, int i) {
    switch (getItemViewType(i)) {
        case 0:
            Song song = (Song) searchResults.get(i);
            itemHolder.title.setText(song.title);
            itemHolder.songArtist.setText(song.albumName);
            ImageLoader.getInstance().displayImage(TimberUtils.getAlbumArtUri(song.albumId).toString(), itemHolder.albumArt,
                new DisplayImageOptions.Builder().cacheInMemory(true)
                    .cacheOnDisk(true)
                    .showImageOnFail(R.drawable.ic_empty_music2)
                    .resetViewBeforeLoading(true)
                    .displayLayer(new FadeInBitmapDisplayer(400))
                    .build());
            setOnPopupMenuListener(itemHolder, i);
            break;
        case 1:
            Album album = (Album) searchResults.get(i);
            itemHolder.albumTitle.setText(album.title);
            itemHolder.albumArtist.setText(album.artistName);
            ImageLoader.getInstance().displayImage(TimberUtils.getAlbumArtUri(album.id).toString(), itemHolder.albumArt,
                new DisplayImageOptions.Builder().cacheInMemory(true)
                    .cacheOnDisk(true)
                    .showImageOnFail(R.drawable.ic_empty_music2)
                    .resetViewBeforeLoading(true)
                    .displayLayer(new FadeInBitmapDisplayer(400))
                    .build());
            break;
        case 2:
            Artist artist = (Artist) searchResults.get(i);
            itemHolder.artistTitle.setText(artist.name);
            String albumNumber = TimberUtils.getLabel(mContext, R.plurals.Nalbums, artist.albumCount);
            String songCount = TimberUtils.getLabel(mContext, R.plurals.Nsongs, artist.songCount);
            itemHolder.albumSongCount.setText(TimberUtils.makeCombinedString(mContext, albumNumber, songCount));
            LastFmClient.getInstance(mContext).getArtistInfo(new ArtistQuery(artist.name), new ArtistInfoListener() {

                @Override
                public void artistInfoSuccess(LastfmArtist artist) {
                    if (artist != null && itemHolder.artistImage != null) {
                        ImageLoader.getInstance().displayImage(artist.mArtwork.get(1).mUrl, itemHolder.artistImage,
                            new DisplayImageOptions.Builder().cacheInMemory(true)
                                .cacheOnDisk(true)
                                .showImageOnFail(R.drawable.ic_empty_music2)
                                .resetViewBeforeLoading(true)
                                .displayLayer(new FadeInBitmapDisplayer(400))
                                .build());
                    }
                }

                @Override
                public void artistInfoFailed() {
                }
            });
            break;
        case 10:
            itemHolder.sectionHeader.setText((String) searchResults.get(i));
        case 3:
            break;
    }
}
}

```

Hàm Option menu màn hình Ablums :

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menu_sort_by_az:
            mPreferences.setAlbumSortOrder(SortOrder.AlbumSortOrder.ALBUM_A_Z);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_za:
            mPreferences.setAlbumSortOrder(SortOrder.AlbumSortOrder.ALBUM_Z_A);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_year:
            mPreferences.setAlbumSortOrder(SortOrder.AlbumSortOrder.ALBUM_YEAR);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_artist:
            mPreferences.setAlbumSortOrder(SortOrder.AlbumSortOrder.ALBUM_ARTIST);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_number_of_songs:
            mPreferences.setAlbumSortOrder(SortOrder.AlbumSortOrder.ALBUM_NUMBER_OF_SONGS);
            reloadAdapter();
            return true;
        case R.id.menu_show_as_list:
            mPreferences.setAlbumsInGrid(false);
            isGrid = false;
            updateLayoutManager(1);
            return true;
        case R.id.menu_show_as_grid:
            mPreferences.setAlbumsInGrid(true);
            isGrid = true;
            updateLayoutManager(2);
            return true;
    }
    return super.onOptionsItemSelected(item);
}

```

Hàm sắp xếp album theo kí tự từ A-Z, Z-A, Artist, Year, Number of Song.

```

/**
 * Album sort order entries.
 */
public interface AlbumSortOrder {
    /* Album sort order A-Z */
    String ALBUM_A_Z = MediaStore.Audio.Albums.DEFAULT_SORT_ORDER;

    /* Album sort order Z-A */
    String ALBUM_Z_A = ALBUM_A_Z + " DESC";

    /* Album sort order songs */
    String ALBUM_NUMBER_OF_SONGS = MediaStore.Audio.Albums.NUMBER_OF_SONGS
        + " DESC";

    /* Album sort order artist */
    String ALBUM_ARTIST = MediaStore.Audio.Albums.ARTIST;

    /* Album sort order year */
    String ALBUM_YEAR = MediaStore.Audio.Albums.FIRST_YEAR + " DESC";
}

```

3.5 Màn hình bài hát trong Albums (Nguyễn Xuân Nguyên Hạnh)

Hàm hiển thị thông tin chi tiết bài hát trong Albums

```
private void setAlbumDetails() {

    String songCount = TimberUtils.makeLabel(getActivity(), R.plurals.Nsongs, album.songCount);

    String year = (album.year != 0) ? (" - " + String.valueOf(album.year)) : "";
    albumTitle.setText(album.title);
    albumDetails.setText(album.artistName + " - " + songCount + year);

}
```

Hàm hiển thị danh sách bài hát trong Albums

```
private void setUpAlbumSongs() {

    List<Song> songList = AlbumSongLoader.getSongsForAlbum(getActivity(), albumID);
    mAdapter = new AlbumSongsAdapter(getActivity(), songList, albumID);
    recyclerView.addItemDecoration(new DividerItemDecoration(getActivity(), DividerItemDecoration.VERTICAL_LIST));
    recyclerView.setAdapter(mAdapter);

}
```

Hàm hiển thị danh sách item menu trong Album

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menu_go_to_artist:
            NavigationUtils.goToArtist(getContext(), album.artistId);
            break;
        case R.id.popup_song_addto_queue:
            MusicPlayer.addToQueue(context, mAdapter.getSongIds(), -1, TimberUtils.IdType.NA);
            break;
        case R.id.popup_song_addto_playlist:
            AddPlaylistDialog.newInstance(mAdapter.getSongIds()).show(mContext.getSupportFragmentManager(), "ADD_PLAYLIST");
            break;
        case R.id.menu_sort_by_az:
            mPreferences.setAlbumSongSortOrder(SortOrder.AlbumSongSortOrder.SONG_A_Z);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_za:
            mPreferences.setAlbumSongSortOrder(SortOrder.AlbumSongSortOrder.SONG_Z_A);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_year:
            mPreferences.setAlbumSongSortOrder(SortOrder.AlbumSongSortOrder.SONG_YEAR);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_duration:
            mPreferences.setAlbumSongSortOrder(SortOrder.AlbumSongSortOrder.SONG_DURATION);
            reloadAdapter();
            return true;
        case R.id.menu_sort_by_track_number:
            mPreferences.setAlbumSongSortOrder(SortOrder.AlbumSongSortOrder.SONG_TRACK_LIST);
            reloadAdapter();
            return true;
    }
    return super.onOptionsItemSelected(item);
}
```

Hàm thêm đi tới Artist có bài hát đó

```
public static void goToArtist(Context context, long artistId) {
    Intent intent = new Intent(context, MainActivity.class);
    intent.setAction(Constants.NAVIGATE_ARTIST);
    intent.putExtra(Constants.ARTIST_ID, artistId);
    context.startActivity(intent);
}
```

Hàm thêm đi tới Album có bài hát đó

```
public static void goToAlbum(Context context, long albumId) {
    Intent intent = new Intent(context, MainActivity.class);
    intent.setAction(Constants.NAVIGATE_ALBUM);
    intent.putExtra(Constants.ALBUM_ID, albumId);
    context.startActivity(intent);
}
```

Hàm xóa bài hát ra khỏi thiết bị.

```

public static void deleteTracks(final Context context, final long[] list) {
    final String[] projection = new String[]{
        BaseColumns._ID, MediaStore.MediaColumns.DATA, MediaStore.Audio.AudioColumns.ALBUM_ID
    };
    final StringBuilder selection = new StringBuilder();
    selection.append(BaseColumns._ID + " IN (");
    for (int i = 0; i < list.length; i++) {
        selection.append(list[i]);
        if (i < list.length - 1) {
            selection.append(",");
        }
    }
    selection.append(")");
    final Cursor c = context.getContentResolver().query(
        MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, projection, selection.toString(),
        null, null);
    if (c != null) {
        // Step 1: Remove selected tracks from the current playlist, as well
        // as from the album art cache
        c.moveToFirst();
        while (!c.isAfterLast()) {
            // Remove from current playlist
            final long id = c.getLong(0);
            MusicPlayer.removeTrack(id);
            // Remove the track from the play count
            SongPlayCount.getInstance(context).removeItem(id);
            // Remove any items in the recents database
            RecentStore.getInstance(context).removeItem(id);
            c.moveToNext();
        }
        // Step 2: Remove selected tracks from the database
        context.getContentResolver().delete(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
            selection.toString(), null);
    }

    // Step 3: Remove files from card
    c.moveToFirst();
    while (!c.isAfterLast()) {
        final String name = c.getString(1);
        final File f = new File(name);
        try { // File.delete can throw a security exception
            if (!f.delete()) {
                // I'm not sure if we'd ever get here (deletion would
                // have to fail, but no exception thrown)
                Log.e("MusicUtils", "Failed to delete file " + name);
            }
            c.moveToNext();
        } catch (final SecurityException ex) {
            c.moveToNext();
        }
    }
    c.close();
}

final String message = makeLabel(context, R.plurals.NNNtracksdeleted, list.length);

Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
context.getContentResolver().notifyChange(Uri.parse("content://media"), null);
MusicPlayer.refresh();
}

```

Hàm thêm bài hát vào mục PlayingQueue

```

public static void addToQueue(final Context context, final long[] list, long sourceId,
    IdType sourceType) {
    if (mService == null) {
        return;
    }
    try {
        mService.enqueue(list, MusicService.LAST, sourceId, sourceType.mId);
        final String message = makeLabel(context, R.plurals.NNNtrackstoqueue, list.length);
        Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
    } catch (final RemoteException ignored) {
    }
}

```

Hàm thêm bài hát vào mục Playlist: có 2 loại Playlist 1 là Playlist có sẵn, 2 là Playlist chưa tạo, nếu thêm vào Playlist chưa tạo App sẽ hiển thị mục tạo Playlist cho bạn và bài hát sẽ được thêm vào đó.

```

public static AddPlaylistDialog newInstance(Song song) {
    long[] songs = new long[1];
    songs[0] = song.id;
    return newInstance(songs);
}

public static AddPlaylistDialog newInstance(long[] songList) {
    AddPlaylistDialog dialog = new AddPlaylistDialog();
    Bundle bundle = new Bundle();
    bundle.putLongArray("songs", songList);
    dialog.setArguments(bundle);
    return dialog;
}

```

Hàm thêm đánh số thứ tự bài hát trong mục Album Detail

```

public static final String makeLabel(final Context context, final int pluralInt,
                                    final int number) {
    return context.getResources().getQuantityString(pluralInt, number, number);
}

```

Hàm thêm đánh tính thời gian bài hát.

```

public static final String makeShortTimeString(final Context context, long secs) {
    long hours, mins;

    hours = secs / 3600;
    secs %= 3600;
    mins = secs / 60;
    secs %= 60;

    final String durationFormat = context.getResources().getString(
        hours == 0 ? R.string.durationformatshort : R.string.durationformatlong);
    return String.format(durationFormat, hours, mins, secs);
}

```

```

        JSONArray jsonArrayWeather = jsonObjectList.getJSONArray("weather");
        JSONObject jsonObjectWeather = jsonArrayWeather.getJSONObject(0);
        String status = jsonObjectWeather.getString("description");
        String icon = jsonObjectWeather.getString("icon");

        mangthoitiet.add(new ThoiTiet(Day, status, icon, NhietdoMax, NhietdoMin));
    }
    customAdapter_thoitiet.notifyDataSetChanged();

} catch (JSONException e) {
    e.printStackTrace();
}

},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }
});
requestQueue.add(stringRequest);
}

```

3.6 Màn hình Playing Queue (Nguyễn Văn Thái)

3.6.1. Navigation Drawer



```
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <group
        android:id="@+id/group1"
        android:checkableBehavior="single">
        <item
            android:id="@+id/nav_library"
            android:title="Library" />
        <item
            android:id="@+id/nav_playlists"
            android:title="Playlists" />
        <item
            android:id="@+id/nav_queue"
            android:title="Playing Queue" />
        <item
            android:id="@+id/nav_nowplaying"
            android:checkable="false"
            android:title="Now Playing" />
    </group>

    <group
        android:id="@+id/group2"
        android:checkableBehavior="none">
        <item
            android:id="@+id/nav_about"
            android:title="@string/about" />
    </group>
</menu>
```

Hình 19. Code Navigation Drawer

3.6.2 Xử lý chuyển sang màn hình Playing Queue.

```

private void updatePosition(final MenuItem menuItem) {
    runnable = null;

    switch (menuItem.getItemId()) {
        case R.id.nav_library:
            runnable = navigateLibrary;
            break;

        case R.id.nav_playlists:
            runnable = navigatePlaylist;
            break;

        case R.id.nav_nowplaying:
            if (getCastSession() != null) {
                startActivity(new Intent(packageContext: MainActivity.this, ExpandedControlsActivity.class));
            } else {
                NavigationUtils.navigateToNowplaying(context: MainActivity.this, withAnimations: false);
            }
            break;

        case R.id.nav_queue:
            runnable = navigateQueue;
            break;

        case R.id.nav_about:
            mDrawerLayout.closeDrawers();
            Handler handler = new Handler();
            handler.postDelayed(() -> {
                Helpers.showAbout(activity: MainActivity.this);
            }, delayMillis: 350);
            break;
    }

    if (runnable != null) {
        menuItem.setChecked(true);
        mDrawerLayout.closeDrawers();
        Handler handler = new Handler();
        handler.postDelayed(() -> { runnable.run(); }, delayMillis: 350);
    }
}

```

Hình 10. Code xử lý chuyển màn hình

```

menu.setOnMenuItemClickListener((item) -> {
    switch (item.getItemId()) {
        case R.id.popup_song_play:
            MusicPlayer.playAll(mContext, getSongIds(), position, sourceld: -1, TimberUtils.IdType.NA, forceShuffle: false);
            break;
        case R.id.popup_song_play_next:
            long[] ids = new long[1];
            ids[0] = arraylist.get(position).id;
            MusicPlayer.playNext(mContext, ids, sourceld: -1, TimberUtils.IdType.NA);
            break;
        case R.id.popup_song_goto_album:
            NavigationUtils.goToAlbum(mContext, arraylist.get(position).albumId);
            break;
        case R.id.popup_song_goto_artist:
            NavigationUtils.goToArtist(mContext, arraylist.get(position).artistId);
            break;
        case R.id.popup_song_addto_queue:
            long[] id = new long[1];
            id[0] = arraylist.get(position).id;
            MusicPlayer.addToQueue(mContext, id, sourceld: -1, TimberUtils.IdType.NA);
            break;
        case R.id.popup_song_addto_playlist:
            AddPlaylistDialog.newInstance(arraylist.get(position)).show(mContext.getSupportFragmentManager(), tag: "ADD_PLAYLIST");
            break;
        case R.id.popup_song_delete:
            long[] deleteIds = {arraylist.get(position).id};
            TimberUtils.showDeleteDialog(mContext, arraylist.get(position).title, deleteIds, qAdapter: BaseQueueAdapter.this, position);
            break;
    }
    return false;
});

```

Xử lý các Item menu

Hình 10. Code xử lý khi lựa chọn các Item menu

3.7 Màn hình phát nhạc – Dương Quốc Vĩnh

Các Bước Thực Hiện

1. Load danh sách các bài hát từ thẻ SSD

```

void display() {
    final ArrayList<File> mySongs = findSong(Environment.getExternalStorageDirectory());
    items = new String[mySongs.size()];
    for (int i = 0; i < mySongs.size(); i++) {
        items[i] = mySongs.get(i).getName().toString().replace( target: ".mp3", replacement: "").replace( target: ".wav", replacement: "");
    }
    ArrayAdapter<String> myAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, items);
    myListviewForSongs.setAdapter(myAdapter);
    myListviewForSongs.setOnItemClickListener((parent, view, i, l) -> {
        String songName = myListviewForSongs.getItemAtPosition(i).toString();
        startActivity(new Intent(getApplicationContext(), NowPlayingActivity.class)
            .putExtra( name: "songs", mySongs).putExtra( name: "songnam", songName)
            .putExtra( name: "pos", i));
    });
}

```

- Load thông tin từ bài hát lên view (tên nghệ sĩ, hình ảnh)

```

public boolean loadArtistAndAlbumImages() {
    if (!mPreferences.getBoolean(ARTIST_ALBUM_IMAGE, true)) {
        if (!mPreferences.getBoolean(ARTIST_ALBUM_IMAGE_MOBILE, true)) {
            if (connManager == null) connManager = (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY_SERVICE);
            NetworkInfo ni = connManager.getActiveNetworkInfo();
            return ni != null && ni.getType() == ConnectivityManager.TYPE_WIFI;
        }
        return true;
    }
    return false;
}

```

2. Xử lý các button ()

- Hàm dừng bài hát

```

public void pause() {
    if (D) Log.d(TAG, "Pausing playback");
    synchronized (this) {
        mPlayerHandler.removeMessages(FADEUP);
        if (mIsSupposedToBePlaying) {
            final Intent intent =
                new Intent(AudioEffect.ACTION_CLOSE_AUDIO_EFFECT_CONTROL_SESSION);
            intent.putExtra(AudioEffect.EXTRA_AUDIO_SESSION, getAudioSessionId());
            intent.putExtra(AudioEffect.EXTRA_PACKAGE_NAME, getPackageName());
            sendBroadcast(intent);

            mPlayer.pause();
            notifyChange(META_CHANGED);
            setIsSupposedToBePlaying(false, true);
        }
    }
}

```

- Hàm phát nhạc

```

mPlayer.start();
mPlayerHandler.removeMessages(FADEDOWN);
mPlayerHandler.sendMessage(FADEUP);

setIsSupposedToBePlaying(true, true);

cancelShutdown();
updateNotification();
notifyChange(META_CHANGED);
} else if (mPlaylist.size() <= 0) {
    setShuffleMode(SHUFFLE_AUTO);
}
}

```

- Hàm lặp lại bài hát

```

public int getRepeatMode() {
    return mRepeatMode;
}

public void setRepeatMode(final int repeatmode) {
    synchronized (this) {
        mRepeatMode = repeatmode;
        setNextTrack();
        saveQueue(false);
        notifyChange(REPEATMODE_CHANGED);
    }
}

```

- Hàm random khi next nhạc

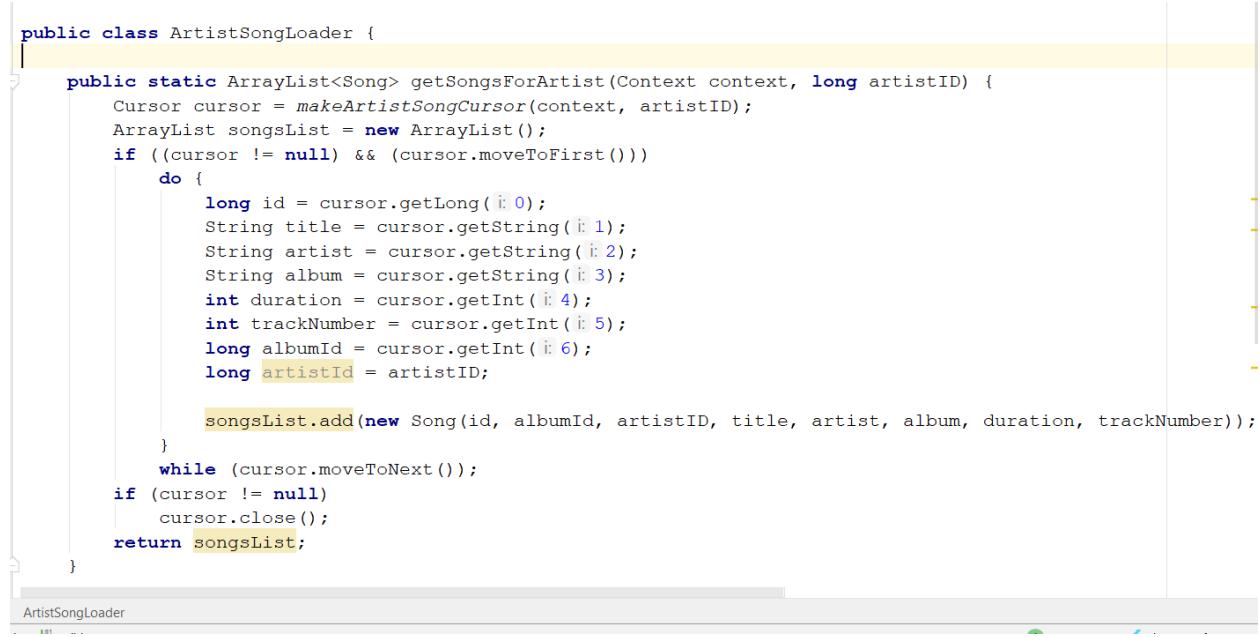
```

private void doAutoShuffleUpdate() {
    boolean notify = false;
    if (mPlayPos > 10) {
        removeTracks(0, mPlayPos - 9);
        notify = true;
    }
    final int toAdd = 7 - (mPlaylist.size() - (mPlayPos < 0 ? -1 : mPlayPos));
    for (int i = 0; i < toAdd; i++) {
        int lookback = mHistory.size();
        int idx = -1;
        while (true) {
            idx = mShuffler.nextInt(mAutoShuffleList.length);
            if (!wasRecentlyUsed(idx, lookback)) {
                break;
            }
            lookback /= 2;
        }
        mHistory.add(idx);
        if (mHistory.size() > MAX_HISTORY_SIZE) {
            mHistory.remove(0);
        }
        mPlaylist.add(new MusicPlaybackTrack(mAutoShuffleList[idx], -1, TimberUtils.IdType.NA, -1));
        notify = true;
    }
    if (notify) {
        notifyChange(QUEUE_CHANGED);
    }
}

```

3.8 Màn hình Artists (Phạm Lê Minh Phú)

- Hàm này load danh sách bài hát vào mục artist



```

public class ArtistSongLoader {
    public static ArrayList<Song> getSongListForArtist(Context context, long artistID) {
        Cursor cursor = makeArtistSongCursor(context, artistID);
        ArrayList<Song> songsList = new ArrayList();
        if ((cursor != null) && (cursor.moveToFirst()))
            do {
                long id = cursor.getLong(0);
                String title = cursor.getString(1);
                String artist = cursor.getString(2);
                String album = cursor.getString(3);
                int duration = cursor.getInt(4);
                int trackNumber = cursor.getInt(5);
                long albumId = cursor.getLong(6);
                long artistId = artistID;

                songsList.add(new Song(id, albumId, artistID, title, artist, album, duration, trackNumber));
            } while (cursor.moveToNext());
        if (cursor != null)
            cursor.close();
        return songsList;
    }
}

```

Hình 27 Hàm này load danh sách bài hát vào mục artist

- Chi tiết Artist:

```

private long artistID = -1;
private ImageView artistArt;
private Toolbar toolbar;
private CollapsingToolbarLayout collapsingToolbarLayout;
private AppBarLayout appBarLayout;
private boolean largeImageLoaded = false;
private int primaryColor = -1;
private ArtistSongAdapter mAdapter;

public static ArtistDetailFragment newInstance(long id, boolean useTransition, String transitionName) {
    ArtistDetailFragment fragment = new ArtistDetailFragment();
    Bundle args = new Bundle();
    args.putLong(Constants.ARTIST_ID, id);
    args.putBoolean("transition", useTransition);
    if (useTransition)
        args.putString("transition_name", transitionName);
    fragment.setArguments(args);
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        artistID = getArguments().getLong(Constants.ARTIST_ID);
    }
}

```

Hình 28 Chi tiết Artist

3.9 Màn hình View as (Phạm Lê Minh Phú)

- Xem theo dạng khôi:

```
//hàm xử lí xem danh sách artist theo grid
public void setArtistsInGrid(final boolean b) {
    final SharedPreferences.Editor editor = mPreferences.edit();
    editor.putBoolean(TOGGLE_ARTIST_GRID, b);
    editor.apply();
}
```

Hình 29 Xem danh sách artist theo grid

```
public void setAlbumsInGrid(final boolean b) {
    final SharedPreferences.Editor editor = mPreferences.edit();
    editor.putBoolean(TOGGLE_ALBUM_GRID, b);
    editor.apply();
}
```

Hình 30 Xem danh sách album theo grid

3.10 Chức năng Shuffle all (Phạm Lê Minh Phú)

```
private static final class Shuffler {
    private final LinkedList<Integer> mHistoryOfNumbers = new LinkedList<~>();
    private final TreeSet<Integer> mPreviousNumbers = new TreeSet<~>();
    private final Random mRandom = new Random();
    private int mPrevious;
    public Shuffler() { super(); }
    public int nextInt(final int interval) {
        int next;
        do {
            next = mRandom.nextInt(interval);
        } while (next == mPrevious && interval > 1
                 && !mPreviousNumbers.contains(Integer.valueOf(next)));
        mPrevious = next;
        mHistoryOfNumbers.add(mPrevious);
        mPreviousNumbers.add(mPrevious);
        cleanUpHistory();
        return next;
    }
    private void cleanUpHistory() {
        if (!mHistoryOfNumbers.isEmpty() && mHistoryOfNumbers.size() >= MAX_HISTORY_SIZE) {
            for (int i = 0; i < Math.max(1, MAX_HISTORY_SIZE / 2); i++) {
                mPreviousNumbers.remove(mHistoryOfNumbers.removeFirst());
            }
        }
    }
}
```

Hình 31 Class Chức năng Shuffle all

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

4.1 Kết quả đạt được

Về mặt kỹ năng:

- Nâng cao kỹ năng lập trình Android.
- Rèn luyện kỹ năng hoàn thành 1 dự án theo yêu cầu.
- Rèn luyện kỹ năng Unit Test
- Rèn luyện kỹ năng tự đọc tài liệu, tìm kiếm tài liệu.
- Nâng cao kỹ năng làm việc nhóm.

Về mặt kiến thức:

- Thiết kế giao diện nâng cao trong android.
- Điều khiển màn hình tương tác và cảm biến.
- Đồ họa và hoạt cảnh
- Đa phương tiện và truyền thông thoại.
- Bản đồ trực tuyến map và location
-

4.2 Kết luận

Những việc đã làm đợt:

- Lấy thông tin bài hát
- Đọc dữ liệu từ bộ nhớ trong, thẻ Sdcard
- Tìm kiếm bài hát theo tên bài hát, ca sĩ

Những việc chưa làm đợt:

- Giao diện chưa tối ưu

4.3 Tài liệu kham khảo

- <https://stackoverflow.com/questions/6674249/how-to-save-file-to-sd-card>
- <https://pixabay.com/images/search/music/>

- <https://www.android-examples.com/android-multiple-auto-image-slider-example-tutorial-using-online-url/>
- <https://www.youtube.com/watch?v=f2MOYNTe2oU&t=299s>
- <https://www.youtube.com/watch?v=8Lq3HyBCuAA>
- <https://www.youtube.com/watch?v=o-1dMLJwOe0&t=756s>
- <https://www.youtube.com/watch?v=OknMZUnTyds&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt>
- <https://www.youtube.com/watch?v=urLA8z6-l3k&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt&index=2>
- <https://www.youtube.com/watch?v=fPFr0So1LmI&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt&index=5>
- <https://www.youtube.com/watch?v=6Trdd9EnmqY&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt&index=8>
- <https://www.youtube.com/watch?v=DhYofrJPzII&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt&index=11>
- https://www.youtube.com/watch?v=mN6kM_1M0cY&t=8s