

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

KHOA: CÔNG NGHỆ THÔNG TIN

**BÁO CÁO KẾT THÚC MÔN HỌC**

**Lập trình di động 2**

**Xây dựng ứng dụng “App Music”**

|  |  |  |  |
| --- | --- | --- | --- |
| Giảng viên hướng dẫn | Trương Bá Thái |  |  |
| Nhóm: 4.2 | Lê Nguyễn Tấn Tài  Trần Thị Lam | Khóa | 2016 |

Thành phố Hồ Chí Minh,22 tháng 4 năm 2019.

**LỜI MỞ ĐẦU**

Trong lời nói đầu tiên của báo cáo Lập trình di động 2 em xin cảm ơn thầy Trương Bá Thái, giảng viên Khoa Công nghệ Thông tin trường Cao đẳng Công nghệ Thủ Đức đã hướng dẫn, nhận xét, hỗ trợ nhóm em trong quá trình thực hiện đề tài.

Nhóm em cũng gửi lời cảm ơn tới các thành viên trong nhóm đã cùng nhau hỗ trợ để hoàn thành môn Lập trình di động 2, giúp các thành viện trong nhóm có kiến thức và tinh thần làm việc nhóm trong quá trình thực hiện đề tài

Em rất mong nhận được ý kiến cũng như nhận xét của thầy để em hoàn thiệt hơn và tiếp tục hoàn thiệt đề tài của nhóm.

Nhóm em xin chân thành cảm ơn.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

1. Về hình thức trình bày báo cáo:

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

2. Nội dung:

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

3. Trình bày bài học của NHÓM:

………………………………………………………………………………….

………………………………………………………………………………….

………………………………………………………………………………….

…………………………………………………………………………………

4. Điểm yếu của NHÓM :……………………………………………

.………………………………………………………………………………….

..…………………………………………………………………………………

5. Điểm mạnh của NHÓM: ………………………………………….

.…………………………………………………………………………………..

..…………………………………………………………………………………

…………………………………………………………………………………...

6. Kết luận:

 Được báo cáo  Không được báo cáo

7. Tổng số điểm của NHÓM đạt được là: . . . /10 điểm

*…………, ngày….tháng….năm……*

**Giảng viên hướng dẫn**

*(ký ghi rõ họ tên)*

Nhật ký hoạt động nhóm 4.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Họ tên | Công việc | Tự đánh giá | Nhóm đánh giá | Giảng viên đánh giá | Ký tên |
| 1 | Lê Nguyễn Tấn Tài | Nghiên cứu đề tài  Viết SDS  Viết SRS  Viết báo cáo  Làm màn hình giao diện nhóm  Làm màn hình nghe nhạc | 50% | 50% |  |  |
| 2 | Trần Thị Lam | Nghiên cứu đề tài  Viết SDS  Viết SRS  Viết báo cáo  Làm màn hình chờ  Làm màn hình trang chủ | 50% | 50% |  |  |

**Mục lục**

Chương 1: MỞ ĐẦU 6

1. 1 Giới thiệu môn học và đề tài thực hiện 6
   * 1. Giới thiệu môn học 6
     2. Giới thiệu đề tài thực hiện 6
     3. Ứng dụng đề tài trong cuộc sống 6

Chương 2: PHÂN TÍCH CẤU TRÚC HỆ THỐNG 6

2.1Phân tích hệ thống 6

* 1. Màn hình giao diện 1: màn hình chờ 7
  2. Màn hình giao diện 2: màn hình trang chủ 8
  3. Màn hình giao diện 3: màn hình giao diện nhóm .10
  4. Màn hình giao diện 4: màn hình phát nhạc 11
  5. Màn hình giao diện 5: màn hình tab 13

Chương 3: CÀI ĐẶT ………………………………………………….13

3.1 Màn hình chờ .. 13

3.2 Màn hình Trang Chủ 14

3.3 Màn hình Play 21

3.4 Màn hình tab 25

Chương 4: KẾT QUẢ ĐẠT ĐƯỢC 25

4.1 Kết quả đạt được 25

4.2 Kết luận 25

**CHƯƠNG 1. MỞ ĐẦU**

* 1. **Giới thiệu môn học và đề tài thực hiện**
  2. Giới thiệu môn học
* Môn lập trình di động 2 là môn nâng cấp của Lập trình di động 1. Môn học này giúp sinh viên phát triển kỹ năng lập trình, tự tìm hiểu, cập nhật kiến thức mới và kiến thức đã học áp dụng cho các dự án thực tế.
* Kiến thức mới: Lấy dữ liệu từ SD card
* Kiến thức đã học: Amination, theme và style, buttom widet, recycler view, soundpool.
* Thành viên thực hiện:
  + Nguyễn Tấn Tài
  + Trần Thị Lam
  1. Giới thiệu đề tài thực hiện
* Tên đề tài: App music
* Mục đích: nghe và phát nhạc.
* Cách thực hiện: Lấy nhạc từ SD card để phát nhạc.
  1. Ứng dụng đề tài trong cuộc sống
* Ngày nay xã hội càng phát triển, nhu cầu giải trí của con người ngày càng cao. Với chiếc điện thoại thông minh không phải lúc nào cũng có internet để có thể nghe nhạc trực tuyến. Vậy nên ứng dụng nghe nhạc offline ra đời để áp ứng nhu cầu của người sử dụng khi muốn nghe nhạc mà không có internet.
* Ứng dụng nghe nhạc offline cho phép người dùng nghe những bản nhạc yêu thích trên SD card đã được tải về trước đó mà không cần kết nối internet.

**CHƯƠNG 2. PHÂN TÍCH CẤU TRÚC HỆ THỐNG**

**2.2. Phân tích hệ thống**

- Tổng quan: Đây là ứng dụng để nghe nhạc trên thiết bị di động mọi lúc mọi nơi khi không có kết nối internet.

- Phạm vi đề tài:

+ Nghe nhạc offline

+ Thiết bị thử nghiệm: Huawei Nova 3i

+ Hệ điều hành: 8.1

+ Độ phân giải: 1080X2340 pixels, 6.3 inch.

1. **Chức năng 1: Màn hình giao diện 1 (Trần Thị Lam)**

* Màn hình chờ



Hình 1. Màn hình chờ

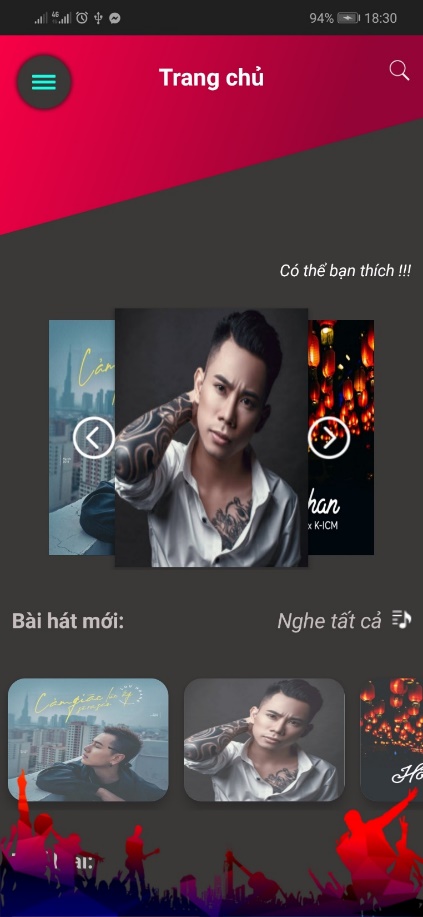
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Quy trình | Kết quả |
| Màn hình chờ | Hiển thị hình chờ | Sau 3s chuyển qua màn hình chính | Hoàn thành |

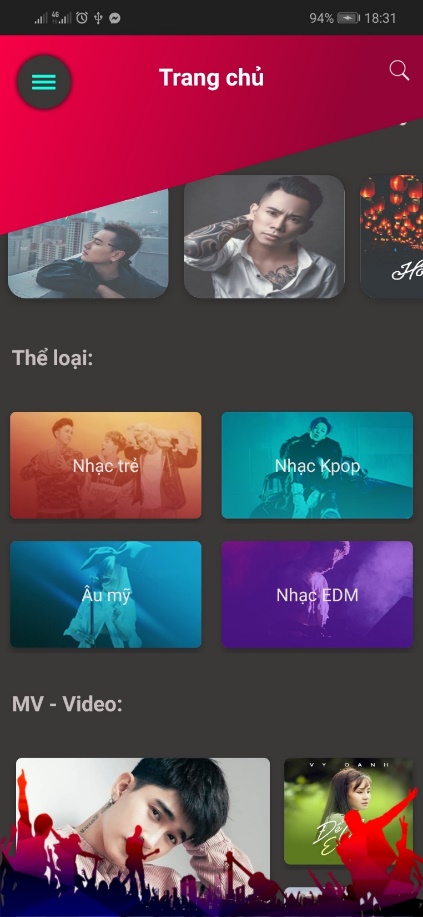
* Làm thế nào: Màn hình chờ
  + Sau khi vào app hiển thị màn hình chờ dùng để load dữ liệu của màn hình tiếp thep trong lúc màn hình chờ xoay. Đợi 3s để chuyển qua màn hình chính.

1. **Chức năng 2: Màn hình giao diện 2 (Trần Thị Lam)**

* Màn hình trang chủ



Hình 2. Màn hình banner



Hình 3. Màn hình thể loại và MV

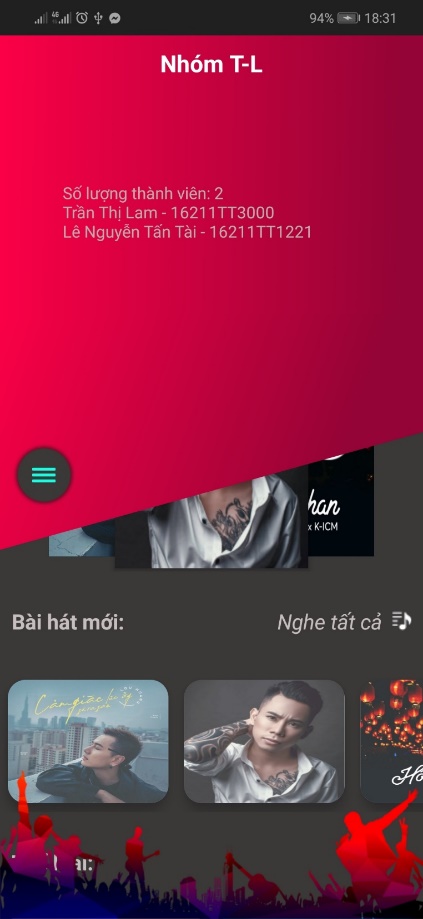
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Quy trình | Kết quả |
| Màn hình slide show | Hiển thị hình ảnh nhạc, click sang có thể chọn hình khác, click vào có thể chuyển sang màn hình phát bài hát | Thiết kế và click | Hoàn thành |
| Recycer view bài hát mới | Hiển thị các bài hát mới và click sẽ chuyển đến màn hình phát nhạc | Click và kéo | Hoàn thành |
| Màn hình thể loại | Hiển thị các thể loại bài hát( Nhạc trẻ, nhạc K-pop, nhac ADM, nhạc Âu-Mĩ), chuyển sang giao diện thể loại | Click | Hoàn thành |

* Làm thế nào: Màn hình giao diện
  + Slide show: Dùng để hiển thị hỉnh ảnh nhạc có thể click hoặc kéo để hiển thị sang màn hình khác.
  + Recycer view bài hát mới: Click
    - Dùng để hiển thị các bài hát mới
  + Image View thể loại: Click
    - Chuyển sang danh mục thể loại bài hát được chọn

1. **Chức năng 3: Màn hình giao diện 3 (Trần Thị Lam)**

* Màn hình giao diện nhóm



Hình 4. Màn hình giao diện nhóm

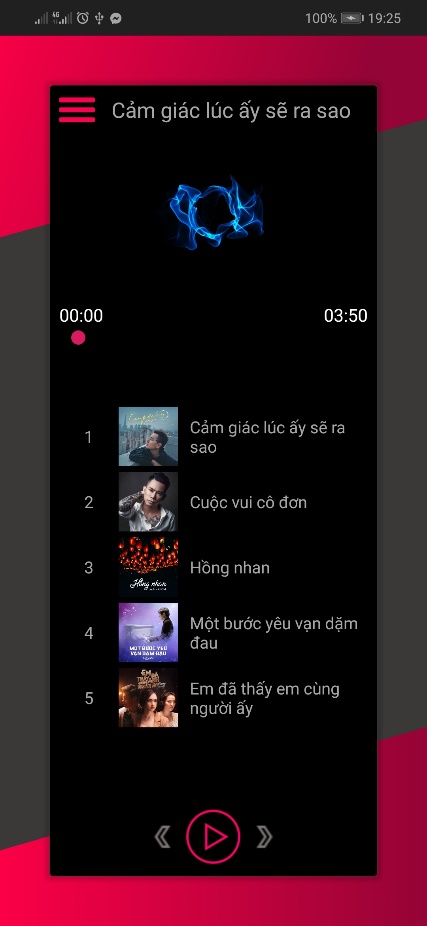
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Quy trình | Kết quả |
| Màn hình giao diện nhóm | Hiển thị thông tin nhóm | Thiết kế | Hoàn thành |

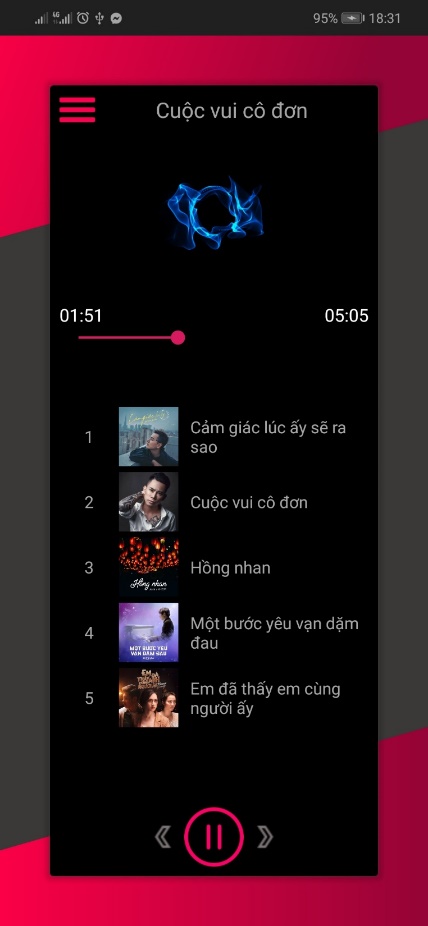
* Làm thế nào: Màn hình giao diện nhóm
  + Linear layout: Thiết kế
    - Dùng để hiển thị thông tin nhóm

1. **Chức năng 4: Màn hình giao diện 4(Lê Nguyễn Tấn Tài)**

* Màn hình phát nhạc



Hình 5. Màn hình phát nhạc (Pause)



Hình 6. Màn hình phát nhạc (Play)

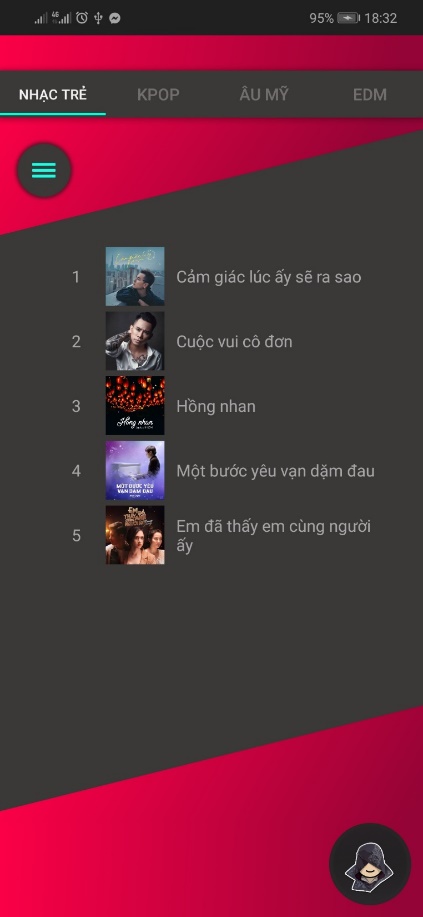
* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Quy trình | Kết quả |
| Màn hình trình phát nhạc (Pause) | Khi ấn vào nút Pause thì bài hát đang phát sẽ dừng lại | Click | Hoàn thành |
| Màn hình trình phát nhạc(Play) | Khi ấn vào nút Play thì nhạc sẽ phát tiếp và thanh thời gian sẽ chạy | Click | Hoàn thành |

* Làm thế nào: Màn hình phát nhạc
  + Image Button(Pauses): Click
    - Dừng phát bài hát đang được phát
  + Image Button (Play): Click
    - Tiếp tục phát bài hát
  + Seekbar: Kéo
    - Dùng để hiển thị thời gian bài hát
    - Kéo dùng để tua đến thời gian mà mình muốn nghe
    - Tự động phát bài mới khi seekbar chạy hết thời gian
  + Image Button (Next, Prev): Click
    - Next: dùng để chuyển sang bài hát tiếp theo.
    - Prev: dùng để quay lại bài hát trước.

1. **Chức năng 4: Màn hình giao diện 4(Lê Nguyễn Tấn Tài)**

* Màn hình Tab



* Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Quy trình | Kết quả |
| Màn hình tab | Khi chọn loại nhạc bên trang hone sẽ chuyển sang màn hình tab hiển thị đúng loại nhạc được chọn. | Click,kéo | Hoàn thành |

* Làm thế nào
  + Tab layout: Kéo
  + Muốn chuyển sang loại nhạc khác cần kéo qua để có thể chuyển thể loại cần chọn.

**CHƯƠNG 3: CÀI ĐẶT**

3.1 Màn hình chờ

a) Code chạy màn hình chờ trong 3 giây

Thread mThread = new Thread(){  
  
 @Override  
 public void run() {  
 try {  
 *sleep*(3000);  
 Intent mySuperIntent = new Intent(SplashScreenActivity.this, MainActivity.class);  
 startActivity(mySuperIntent);  
 finish();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
};  
mThread.start();

b) Code xin quyền truy cập vào thư mục nhạc

// Creating Runtime permission function.  
public void AndroidRuntimePermission(){  
 if(Build.VERSION.*SDK\_INT*>=Build.VERSION\_CODES.*M*){  
  
 if(checkSelfPermission(Manifest.permission.*READ\_EXTERNAL\_STORAGE*)!= PackageManager.*PERMISSION\_GRANTED*){  
  
 if(shouldShowRequestPermissionRationale(Manifest.permission.*READ\_EXTERNAL\_STORAGE*)){  
  
 AlertDialog.Builder alert\_builder = new AlertDialog.Builder(this);  
 alert\_builder.setMessage("External Storage Permission is Required.");  
 alert\_builder.setTitle("Please Grant Permission.");  
 alert\_builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {  
  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
  
 ActivityCompat.*requestPermissions*(  
 SplashScreenActivity.this,  
 new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*},  
 1  
  
 );  
 }  
 });  
  
 alert\_builder.setNeutralButton("Cancel",null);  
  
 AlertDialog dialog = alert\_builder.create();  
  
 dialog.show();  
  
 }  
 else {  
  
 ActivityCompat.*requestPermissions*(  
 SplashScreenActivity.this,  
 new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*},  
 1  
 );  
 }  
 }else {  
  
 }  
 }  
}  
  
@Override  
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults){  
  
 switch(requestCode){  
  
 case 1:{  
  
 if(grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){  
  
 }  
 else {  
  
 }  
 }  
 }  
}

3.2 Màn hình Trang Chủ

a) set control

// setControl  
public void setControl()  
{  
 recyclerView = (RecyclerView)findViewById(R.id.*recycler\_view*);  
 imgAbout = (ImageButton) findViewById(R.id.*imgAbout*);  
 imgNext = (ImageButton) findViewById(R.id.*imgNext*);  
 imgPrev = (ImageButton) findViewById(R.id.*imgPrev*);  
 btnSearch = (ImageButton) findViewById(R.id.*btnSearch*);  
 fmheader = (FrameLayout) findViewById(R.id.*fmheader*);  
 fmAbout = (FrameLayout) findViewById(R.id.*fmAbout*);  
 tvHome = (TextView) findViewById(R.id.*tvHome*);  
 img1 = (ImageView) findViewById(R.id.*img1*);  
 img2 = (ImageView) findViewById(R.id.*img2*);  
 img3 = (ImageView) findViewById(R.id.*img3*);  
 cardNhacTre = (CardView) findViewById(R.id.*cardNhacTre*);  
 cardKpop = (CardView) findViewById(R.id.*cardKpop*);  
 cardAuMy = (CardView) findViewById(R.id.*cardAuMy*);  
 cardEDM = (CardView) findViewById(R.id.*cardEDM*);  
 lnTopThree = (LinearLayout) findViewById(R.id.*lnTopThree*);  
 lnTitleBaiHat = (LinearLayout) findViewById(R.id.*lnTitleBaiHat*);  
 lnDanhSachBaiHat = (LinearLayout) findViewById(R.id.*lnDanhSachBaiHat*);  
}

b) set event

// setEvent  
public void setEvent()  
{  
 Animation slide\_up = AnimationUtils.*loadAnimation*(getApplicationContext(),  
 R.anim.*slide\_up*);  
 lnTopThree.startAnimation(slide\_up);  
 lnTitleBaiHat.startAnimation(slide\_up);  
 lnDanhSachBaiHat.startAnimation(slide\_up);  
  
 imgAbout.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
// Khi click vào nút Bar tên thanh công cụ thì màn hình giao diện nhóm sẽ được đẩy xuống  
 count++;

// click vào nút bar lần 1 thì đẩy giao diện xuống  
 if (count == 1)  
 {  
 fmAbout.setVisibility(View.*VISIBLE*);  
 fmAbout.animate().translationY(0).setDuration(500);  
 fmheader.animate().translationY(800).setDuration(500); // đẩy giao diện xuống 800dp theo chiều Y trong 0.5s  
 imgAbout.animate().translationY(200).setDuration(500);  
 tvHome.animate().alpha(0).setDuration(500);  
 btnSearch.animate().alpha(0).setDuration(500);  
 }

// click vào nút bar lần 2 thì thu lại giao diện  
 if (count == 2)  
 {  
  
 fmAbout.animate().translationY(-800).setDuration(500);  
 fmheader.animate().translationY(0).setDuration(500);  
 imgAbout.animate().translationY(0).setDuration(500);  
 tvHome.animate().alpha(1).setDuration(500); // hiện dần lên trong 0.5s  
 btnSearch.animate().alpha(1).setDuration(500); // tương tự  
 count = 0;  
 }  
 }  
 });  
 imgNext.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 countNext++;  
 if(countNext == 1)  
 {   
 // set màn hình 2  
 img2.animate().translationX(0).setDuration(500);  
 img2.animate().translationZ(10).setDuration(500);  
 img2.animate().scaleX((float)1.1).setDuration(500);  
 img2.animate().scaleY((float)1.1).setDuration(500);  
 img2.setElevation(2);  
  
 // thay doi img1  
 img1.animate().translationX(-190).setDuration(500);  
 img1.animate().translationZ(0).setDuration(500);  
 img1.animate().scaleX((float)1).setDuration(500);  
 img1.animate().scaleY((float)1).setDuration(500);  
 img1.setElevation(1);  
  
 // thay doi img3  
 img3.animate().translationX(190).setDuration(500);  
 img3.animate().translationZ(-5).setDuration(500);  
 img3.animate().scaleX((float)1).setDuration(500);  
 img3.animate().scaleY((float)1).setDuration(500);  
 img3.setElevation(1);  
 countPrev = 2;  
  
 }  
 else if (countNext == 2)  
 {  
 // set màn hình 3  
 img3.animate().translationX(0).setDuration(500);  
 img3.animate().translationZ(10).setDuration(500);  
 img3.animate().scaleX((float)1.1).setDuration(500);  
 img3.animate().scaleY((float)1.1).setDuration(500);  
 img3.setElevation(2);  
  
 // thay doi img2  
 img2.animate().translationX(-190).setDuration(500);  
 img2.animate().translationZ(0).setDuration(500);  
 img2.animate().scaleX((float)1).setDuration(500);  
 img2.animate().scaleY((float)1).setDuration(500);  
 img2.setElevation(1);  
  
 // thay doi img1  
 img1.animate().translationX(190).setDuration(500);  
 img1.animate().translationZ(-5).setDuration(500);  
 img1.animate().scaleX((float)1).setDuration(500);  
 img1.animate().scaleY((float)1).setDuration(500);  
 img1.setElevation(1);  
 countPrev = 1;  
 }  
 else if (countNext == 3)  
 {  
 // set màn hình 1  
 img1.animate().translationX(0).setDuration(500);  
 img1.animate().translationZ(10).setDuration(500);  
 img1.animate().scaleX((float)1.1).setDuration(500);  
 img1.animate().scaleY((float)1.1).setDuration(500);  
 img1.setElevation(2);  
  
 // thay doi img3  
 img3.animate().translationX(-190).setDuration(500);  
 img3.animate().translationZ(0).setDuration(500);  
 img3.animate().scaleX((float)1).setDuration(500);  
 img3.animate().scaleY((float)1).setDuration(500);  
 img3.setElevation(1);  
  
 // thay doi img2  
 img2.animate().translationX(190).setDuration(500);  
 img2.animate().translationZ(-5).setDuration(500);  
 img2.animate().scaleX((float)1).setDuration(500);  
 img2.animate().scaleY((float)1).setDuration(500);  
 img2.setElevation(1);  
 countNext = 0;  
 countPrev = 0;  
 }  
  
 }  
 });  
// lui lại 1 item trong slider top  
 imgPrev.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 countPrev++;

// set lại vị trí cho 3 item trong slide top khi click  
 if (countPrev == 1)  
 {  
 //set man hinh 3  
 img3.animate().translationX(0).setDuration(500);  
 img3.animate().translationZ(10).setDuration(500);  
 img3.animate().scaleX((float)1.1).setDuration(500);  
 img3.animate().scaleY((float)1.1).setDuration(500);  
 img3.setElevation(2);  
  
  
 // thay doi img2  
 img2.animate().translationX(-190).setDuration(500);  
 img2.animate().translationZ(0).setDuration(500);  
 img2.animate().scaleX((float)1).setDuration(500);  
 img2.animate().scaleY((float)1).setDuration(500);  
 img2.setElevation(1);  
  
 // thay doi img1  
 img1.animate().translationX(190).setDuration(500);  
 img1.animate().translationZ(-5).setDuration(500);  
 img1.animate().scaleX((float)1).setDuration(500);  
 img1.animate().scaleY((float)1).setDuration(500);  
 img1.setElevation(1);  
// tăng đếm lên để cập nhật cho lân click vào nút “Next”  
 countNext = 2;  
 }  
 else if (countPrev == 2)  
 {  
  
 // set màn hình 2  
 img2.animate().translationX(0).setDuration(500);  
 img2.animate().translationZ(10).setDuration(500);  
 img2.animate().scaleX((float)1.1).setDuration(500);  
 img2.animate().scaleY((float)1.1).setDuration(500);  
 img2.setElevation(2);  
  
 // thay doi img1  
 img1.animate().translationX(-190).setDuration(500);  
 img1.animate().translationZ(0).setDuration(500);  
 img1.animate().scaleX((float)1).setDuration(500);  
 img1.animate().scaleY((float)1).setDuration(500);  
 img1.setElevation(1);  
  
 // thay doi img3  
 img3.animate().translationX(190).setDuration(500);  
 img3.animate().translationZ(-5).setDuration(500);  
 img3.animate().scaleX((float)1).setDuration(500);  
 img3.animate().scaleY((float)1).setDuration(500);  
 img3.setElevation(1);  
  
  
 countNext = 1;  
 }  
 else if(countPrev == 3)  
 {  
 // set màn hình 1  
 img1.animate().translationX(0).setDuration(500);  
 img1.animate().translationZ(10).setDuration(500);  
 img1.animate().scaleX((float)1.1).setDuration(500);  
 img1.animate().scaleY((float)1.1).setDuration(500);  
 img1.setElevation(2);  
  
 // thay doi img3  
 img3.animate().translationX(-190).setDuration(500);  
 img3.animate().translationZ(0).setDuration(500);  
 img3.animate().scaleX((float)1).setDuration(500);  
 img3.animate().scaleY((float)1).setDuration(500);  
 img3.setElevation(1);  
  
 // thay doi img2  
 img2.animate().translationX(190).setDuration(500);  
 img2.animate().translationZ(-5).setDuration(500);  
 img2.animate().scaleX((float)1).setDuration(500);  
 img2.animate().scaleY((float)1).setDuration(500);  
 img2.setElevation(1);

// set biến đếm của nút “Next” và “Prev” để lặp lại trang thái ban đầu  
 countPrev = 0;  
 countNext = 0;  
 }  
  
 }  
 });  
  
 // Top 3 ramdom click

// click vào vị trí nào trong slider thì chuyển dữ liệu vị trí qua màn hình “Play”   
 img1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intentSong = new Intent(MainActivity.this, PlayActivity.class);  
 intentSong.putExtra("positionBaiHat",0);  
 startActivity(intentSong);  
 }  
 });  
 img2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intentSong = new Intent(MainActivity.this, PlayActivity.class);  
 intentSong.putExtra("positionBaiHat",1);  
 startActivity(intentSong);  
 }  
 });  
 img3.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intentSong = new Intent(MainActivity.this, PlayActivity.class);  
 intentSong.putExtra("positionBaiHat",2);  
 startActivity(intentSong);  
 }  
 });  
  
// Gửi dữ liệu position của item trong recyclerView cho màn hình “Play”  
 recyclerView.addOnItemTouchListener(new RecyclerItemClickListener(this, recyclerView, new RecyclerItemClickListener.OnItemClickListener() {  
 @Override  
 public void onItemClick(View view, int positiona) {  
  
 Intent intentSong = new Intent(MainActivity.this, PlayActivity.class);  
 intentSong.putExtra("positionBaiHat",positiona);  
 startActivity(intentSong);  
 }  
  
 @Override  
 public void onLongItemClick(View view, int position) {  
  
 }  
 }));  
  
  
  
  
  
  
  
  
 // thể loại click  
 cardEDM.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

// khởi tạo intent để chuyển màn hình  
 Intent intentChuDe = new Intent(MainActivity.this, TabsActivity.class);

// Gán dữ liệu cho itent (vị trí của item trong SDCARD)  
 intentChuDe.putExtra("positionChuDe",3);

// Khởi chạy intent qua màn hình thẻ loại( màn hình tab )  
 startActivity(intentChuDe);  
 }  
 });  
 cardAuMy.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

// Tương tự code trên  
 Intent intentChuDe = new Intent(MainActivity.this, TabsActivity.class);  
 intentChuDe.putExtra("positionChuDe",2);  
 startActivity(intentChuDe);  
 }  
 });  
 cardKpop.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

// Tương tự code trên  
 Intent intentChuDe = new Intent(MainActivity.this, TabsActivity.class);  
 intentChuDe.putExtra("positionChuDe",1);  
 startActivity(intentChuDe);  
 }  
 });  
 cardNhacTre.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

// Tương tự code trên  
 Intent intentChuDe = new Intent(MainActivity.this, TabsActivity.class);  
 intentChuDe.putExtra("positionChuDe",0);  
 startActivity(intentChuDe);  
 }  
 });  
}

1. Đọc dữ liệu (gán dữ liệu cho adapter)

public void initView() {  
  
  
  
  
 recyclerView.setHasFixedSize(true);  
 LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this,LinearLayoutManager.*VERTICAL*,false);  
 recyclerView.setLayoutManager(linearLayoutManager);  
 Uri uri = MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
 arrayList = new ArrayList<>();  
 ContentResolver contentResolver = getContentResolver();  
 String selection = MediaStore.Audio.Media.*IS\_MUSIC* + "!=0";  
  
// đọc file từ SDCARD  
  
 try{  
 Cursor cursor ;  
 cursor = this.getApplicationContext().getContentResolver().query(uri, null, selection, null, null);  
  
 if (cursor != null && cursor.getCount() > 0) {  
 if (cursor.moveToFirst()) {  
 do {  
 //ImageView imageView = null;  
 int image = R.drawable.*emda*;  
 String name = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*DISPLAY\_NAME*));  
 //String artist = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.ARTIST));  
 String url = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*DATA*));  
 MediaMetadataRetriever retriever = new MediaMetadataRetriever();  
 retriever.setDataSource(url);  
 byte[] coverBytes = retriever.getEmbeddedPicture();  
 if (coverBytes != null) {  
 Bitmap songCover = BitmapFactory.*decodeByteArray*(coverBytes, 0, coverBytes.length);  
 //Drawable d = new BitmapDrawable(getResources(), songCover);  
 //image = d;  
 }  
 Song song = new Song(name, url, R.drawable.*emda*);  
 arrayList.add(song);  
  
 } while (cursor.moveToNext());  
 }  
  
 cursor.close();  
  
 }  
 }  
 catch (Exception e)  
 {  
 Toast.*makeText*(this,"" + e, Toast.*LENGTH\_LONG*).show();  
 }  
  
 SongVerticalAdapter songVerticalAdapter = new SongVerticalAdapter(arrayList,getApplicationContext());  
 recyclerView.setAdapter(songVerticalAdapter);  
}

3.3 Màn hình Play

a) set control

public void setControl() {  
 recyclerView = (RecyclerView)findViewById(R.id.*recycler\_view*);  
 tvTitleName = (TextView) findViewById(R.id.*tvTitleName*);  
 tvTimeStart = (TextView) findViewById(R.id.*tvTimeStart*);  
 tvTimeEnd = (TextView) findViewById(R.id.*tvTimeEnd*);  
 ibtnNext = (ImageButton) findViewById(R.id.*ibtnNext*);  
 ibtnPlay = (ImageButton) findViewById(R.id.*ibtnPlay*);  
 ibtnPrev = (ImageButton) findViewById(R.id.*ibtnPrev*);  
 sbTimeLine = (SeekBar) findViewById(R.id.*sbTimeLine*);  
  
}

b) setEvent

public void setEvent(){

// Nhấn play  
 ibtnPlay.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

// Load hiệu ứng cho biến animation  
 animation = AnimationUtils.*loadAnimation*(PlayActivity.this,R.anim.*play\_pause\_rotate*);

// Kiểm trang tra xem nhạc có đang chạy không  
 if (mediaPlayer.isPlaying())   
 {

// nếu đang chạy thì dừng nhạc lại  
 mediaPlayer.pause();  
 ibtnPlay.setBackgroundResource(R.drawable.*play*);  
 }  
 else  
 {

// nếu không thì chạy nhạc lên   
 mediaPlayer.start();  
 ibtnPlay.setBackgroundResource(R.drawable.*pause*);  
  
 }

// set animation xoay cho nút play khi click  
 ibtnPlay.startAnimation(animation);

// gọi đến hàm cập nhật thời gian bắt đầu và kết thúc  
 setTimeTotal();

// gọi đến hàm update thời gian  
 updateTimeSong();  
// Toast.makeText(PlayActivity.this, "" + arrayList.get(position).getSzName(), Toast.LENGTH\_SHORT).show();  
 }  
 });

// Set event next  
 ibtnNext.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {

// Tằng position hiện tại lên 1  
 position++;

// kiểm trang nếu position vượt quá size của dữ liệu thì gán lại position 0 để lặp lại trạng thái ban đầu  
 if (position > arrayList.size()-1)  
 {  
 position = 0;  
 }

// kiểm trang nhạc đang phát thì ngưng bài hát lại (reset mediaplayer)  
 if (mediaPlayer.isPlaying())  
 {  
 mediaPlayer.stop();  
 }

// chạy bài hát tiêp theo  
 createMediaPlayer();

// set hình lại thành nút Pause  
 ibtnPlay.setBackgroundResource(R.drawable.*pause*);  
  
 }  
 });

// event nút Prev (tương tự nút Next nhưng chỉ - positon lại)  
  
 ibtnPrev.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 position--;  
 if (position < 0)  
 {  
 position = arrayList.size()-1;  
 }  
 if (mediaPlayer.isPlaying())  
 {  
 mediaPlayer.stop();  
 }  
 createMediaPlayer();  
 ibtnPlay.setBackgroundResource(R.drawable.*pause*);  
 }  
 });

// set sư kiện kéo của seekbar  
 sbTimeLine.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {

// cập nhật line seekbar (đại khái)  
 mediaPlayer.seekTo(sbTimeLine.getProgress());  
 }  
 });

// event recyclerView  
 recyclerView.addOnItemTouchListener(new RecyclerItemClickListener(this, recyclerView, new RecyclerItemClickListener.OnItemClickListener() {  
 @Override  
 public void onItemClick(View view, int positiona) {

// kiểm tra  
 if (mediaPlayer.isPlaying())  
 {  
 mediaPlayer.stop();  
 }

// load animation   
 animation = AnimationUtils.*loadAnimation*(PlayActivity.this,R.anim.*play\_pause\_rotate*);

// cập nhật lại position  
 position = positiona;

// chạy bài hát mới với position mới được cập nhật  
 createMediaPlayer();  
// set hình  
 ibtnPlay.setBackgroundResource(R.drawable.*pause*);

// set animation cho nut play  
 ibtnPlay.startAnimation(animation);  
  
  
 }  
  
 @Override  
 public void onLongItemClick(View view, int position) {  
  
 }  
 }));  
  
 }

1. Load dữ liệu (tương tự trang chủ)

public void initView() {  
  
  
  
  
 recyclerView.setHasFixedSize(true);  
 LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this,LinearLayoutManager.*VERTICAL*,false);  
 recyclerView.setLayoutManager(linearLayoutManager);  
 Uri uri = MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
 arrayList = new ArrayList<>();  
 ContentResolver contentResolver = getContentResolver();  
 String selection = MediaStore.Audio.Media.*IS\_MUSIC* + "!=0";  
  
  
  
 try{  
 Cursor cursor ;  
 cursor = this.getApplicationContext().getContentResolver().query(uri, null, selection, null, null);  
  
 if (cursor != null && cursor.getCount() > 0) {  
 if (cursor.moveToFirst()) {  
 do {  
 //ImageView imageView = null;  
 int image = R.drawable.*emda*;  
 String name = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*DISPLAY\_NAME*));  
 //String artist = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.ARTIST));  
 String url = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*DATA*));  
 MediaMetadataRetriever retriever = new MediaMetadataRetriever();  
 retriever.setDataSource(url);  
 byte[] coverBytes = retriever.getEmbeddedPicture();  
 if (coverBytes != null) {  
 Bitmap songCover = BitmapFactory.*decodeByteArray*(coverBytes, 0, coverBytes.length);  
 //Drawable d = new BitmapDrawable(getResources(), songCover);  
 //image = d;  
 }  
 Song song = new Song(name, url, R.drawable.*emda*);  
 arrayList.add(song);  
  
 } while (cursor.moveToNext());  
 }  
  
 cursor.close();  
  
 }  
 }  
 catch (Exception e)  
 {  
 Toast.*makeText*(this,"" + e, Toast.*LENGTH\_LONG*).show();  
 }  
  
 SongVerticalAdapter songVerticalAdapter = new SongVerticalAdapter(arrayList,getApplicationContext());  
 recyclerView.setAdapter(songVerticalAdapter);  
}

d) Định dạng thời gian

// định dạng lại time  
public void setTimeTotal(){

// set lại định dạng là phút và giây  
 SimpleDateFormat dinhDangGio = new SimpleDateFormat("mm:ss");

// set time time end và thành seekbar  
 tvTimeEnd.setText(dinhDangGio.format(mediaPlayer.getDuration()));  
 sbTimeLine.setMax(mediaPlayer.getDuration());  
}

e) update seekbar theo thời gian của bài hát

// update seekbar theo time của bài hát  
public void updateTimeSong(){  
 final Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 SimpleDateFormat dinhDangGio = new SimpleDateFormat("mm:ss");  
 tvTimeStart.setText(dinhDangGio.format(mediaPlayer.getCurrentPosition())); // vị trí hiện tại của thanh  
  
 // cập nhật thanh seekbar  
 sbTimeLine.setProgress(mediaPlayer.getCurrentPosition());  
  
 // kiểm tra time chạy đến hết bài  
 mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {  
 @Override  
 public void onCompletion(MediaPlayer mp) {  
 position++;  
 if (position > arrayList.size()-1)  
 {  
 position = 0;  
 }  
 if (mediaPlayer.isPlaying())  
 {  
 mediaPlayer.stop();  
 }  
 createMediaPlayer();  
 ibtnPlay.setBackgroundResource(R.drawable.*pause*);  
 }  
 });  
  
 handler.postDelayed(this,500);  
 }  
 },100);  
}

3.4 Màn hình tab

a) set màn hình tab khi được chọn

if ((intent = getIntent()) != null)  
{

// set tab tại vị trí từ intent gửi qua  
 TabLayout.Tab tab = mainTab.getTabAt(intent.getIntExtra("positionChuDe",0));

// hiển thị trang tab được chọn  
 tab.select();  
}

b) tạo dữ liệu cho tab

public void createTabsNav()  
{

// set control

mainTab = (TabLayout) findViewById(R.id.*mainTab*);  
 mainViewPager = (ViewPager) findViewById(R.id.*mainViewPager*);

// khởi tạo 1 tabViewPageAdapter   
 TabViewPagerAdapter tabViewPagerAdapter = new TabViewPagerAdapter(getSupportFragmentManager());

// add fragment cho Adapter  
 tabViewPagerAdapter.addFragment(new FragmentCaSi(), "Nhạc trẻ");  
 tabViewPagerAdapter.addFragment(new FragmentCaSi(), "Kpop");  
 tabViewPagerAdapter.addFragment(new FragmentCaSi(), "Âu mỹ");  
 tabViewPagerAdapter.addFragment(new FragmentCaSi(), "EDM");

// set dữ liệu mới được add vào  
 mainViewPager.setAdapter(tabViewPagerAdapter);  
 mainTab.setupWithViewPager(mainViewPager);  
}

**CHƯƠNG 4: KẾT QUẢ ĐẶT ĐƯỢC**

4.1 Kết quả đạt được

- App chạy và phát được nhạc được lấy từ SD card.

- App có thể chuyển bài tiếp theo và quay lại bài mới.

- App có thể tua thời gian phát nhạc sử dụng seekbar.

- App có thể chọn thể loại bài hát.

4.2. Kết luận

* Những việc đã làm được:
  + Lấy được nhạc từ SD card.
  + Phát được bài hát muốn chọn.
  + Tua được thời gian bài hát.
  + Quay lại bài hát trước
  + Chọn bài hát tiếp theo
* Những việc chưa làm được
  + Chưa làm chức năng reload bài hát.
  + Chưa phát ngẫu nhiên bài hát.