**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN MÔN HỌC LẬP TRÌNH THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: Thiết kế và xây dựng ứng dụng quản lý LOA hát Karaoke tại cửa hàng loa Vinh Kiệt**

**Giảng viên hướng dẫn : GIANG HÀO CÔN**

**Sinh viên thực hiện : TRẦN VŨ QUANG TRƯỜNG**

**Mã số sinh viên : 2100010231**

**Lớp : 21DKTPM1C**

**Chuyên ngành : Kỹ thuật phần mềm**

**Khóa : 2021**

*TP. Hồ Chí Minh, tháng 1, năm 2024.*

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN MÔN HỌC LẬP TRÌNH THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: Thiết kế và xây dựng ứng dụng quản lý LOA hát Karaoke tại cửa hàng loa Vinh Kiệt**

**Giảng viên hướng dẫn : GIANG HÀO CÔN**

**Sinh viên thực hiện : TRẦN VŨ QUANG TRƯỜNG**

**Mã số sinh viên : 2100010231**

**Lớp : 21DKTPM1C**

**Chuyên ngành : Kỹ thuật phần mềm**

**Khóa : 2021**

*TP. Hồ Chí Minh, tháng 1, năm 2024.*

LỜI NÓI ĐẦU

Để hoàn thành đồ án môn học này trước tiên chúng em xin được gửi lời cảm ơn của mình đến quý thầy cô giảng viên của trường đại học Nguyễn Tất Thành lời cảm ơn chân thành và sâu sắc nhất.

Đặc biệt, chúng em xin gửi đến thầy **GIANG HÀO CÔN** người đã luôn tận tình hướng dẫn hỗ trợ, góp ý giúp đỡ để chúng em có thể hoàn thành tốt đồ án môn học này lời cảm ơn sâu sắc nhất.

Tuy nhiên trong quá trình nghiên cứu hoàn thành đề tài, do kiến thức chuyên ngành còn hạn chế nên chúng em vẫn còn rất nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đề tài này. Rất mong nhận được sự quan tâm, góp ý nhiệt tình của thầy **GIANG HÀO CÔN** để đề tài của em đầy đủ và hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

Hồ Chí Minh ngày 2 tháng 01 năm 2024

Sinh viên thực hiện

**Trần Vũ Quang Trường**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 7 – NĂM HỌC 2023 - 2024** |

Phiếu chấm tiểu luận / Đồ án

BM-ChT-11

Môn thi: **Lập Trình Thiết Bị Di Động**  Lớp học phần: **21DKTPM1C**

Nhóm sinh viên thực hiện:

1. **Trần Vũ Quang Trường** Mã số sinh viên: **2100010231**

Ngày thi: **02/01/2024** Phòng thi: **L.404A**

Đề tài tiểu luận/báo cáo của sinh viên: **Thiết kế và xây dựng ứng dụng quản lý loa hát karaoke tại cửa hàng loa Vinh Kiệt**

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  | 2 |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  | 2 |  |
| * Lập luận |  | 4 |  |
| * Kết luận |  | 1 |  |
| Trình bày |  | 1 |  |
| **TỔNG ĐIỂM** |  | 10 |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* |

Nhận xét của giảng viên

**Nhận Xét Của Giảng Viên**

Mục lục

[Chương I: Tổng Quan Đề Tài 8](#_Toc154953898)

[1.1. Giới thiệu. 8](#_Toc154953899)

[1.2. Khảo sát hiện trạng. 8](#_Toc154953900)

[1.3. Mục tiêu đề tài. 8](#_Toc154953901)

[1.4. Phạm vi đề tài. 8](#_Toc154953902)

[1.5. Cấu trúc đề tài. 9](#_Toc154953903)

[Chương II: CƠ SỞ LÝ THUYẾT 10](#_Toc154953904)

[2.1. Giới thiệu tổng quan lập trình Android. 10](#_Toc154953905)

[2.2. Các thành phần ứng dụng của Android. 10](#_Toc154953906)

[2.3. Giao diện và xử lý sự kiện trong Android. 12](#_Toc154953907)

[2.4. Tài nguyên Ứng dụng trong Android. 15](#_Toc154953908)

[2.5. Đa ngôn ngữ và xử lý Tập tin. 15](#_Toc154953909)

[2.6. Asset, SharePreferences và lưu trữ dữ liệu. 17](#_Toc154953910)

[2.7. SQLite. 19](#_Toc154953911)

[Chương III: THỰC NGHIỆM VÀ TRIỂN KHAI 22](#_Toc154953912)

[3.1. Thiết kế. 22](#_Toc154953913)

[3.1.1 Mô hình chức năng (BFD). 22](#_Toc154953914)

[3.1.2 Mô hình quan hệ dữ liệu. 23](#_Toc154953915)

[3.2. Cài đặt ứng dụng. 26](#_Toc154953916)

[3.2.1 Màn hình chờ. 26](#_Toc154953917)

[3.2.2 Màn hình chính. 27](#_Toc154953918)

[3.2.3 Màn hình thêm dữ liệu. 30](#_Toc154953919)

[3.2.4 Màn hình xóa dữ liệu. 31](#_Toc154953920)

[3.2.5 Màn hình chỉnh sửa dữ liệu. 32](#_Toc154953921)

[3.2.6 Màn hình chi tiết sản phẩm. 33](#_Toc154953922)

[Chương IV: KẾT LUẬN 34](#_Toc154953923)

[4.1. Kết quả đạt được. 34](#_Toc154953924)

[4.2. Hướng phát triển. 34](#_Toc154953925)

[Tài liệu tham khảo 35](#_Toc154953926)

Danh mục hình ảnh

[Hình 1.1. Sơ đồ tổng quát chức năng BFD. 22](#_Toc154954352)

[Hình 1.2. Bảng mô tả chi tiết chức năng. 23](#_Toc154954353)

[Hình 1.3. Table “LoaTable” trong cơ sở dữ liệu. 24](#_Toc154954354)

[Hình 1.4. Thông tin về các hàng (Row) được thêm vào cơ sở dữ liệu 25](#_Toc154954355)

[Hình 1.5. Giao diện màn hình chờ. 26](#_Toc154954356)

[Hình 1.6. Giao diện của màn hình chính 27](#_Toc154954357)

[Hình 1.7. Menu chuyển trang trong màn hình chính. 28](#_Toc154954358)

[Hình 1.8. Menu sắp xếp sản phẩm trong màn hình chính. 29](#_Toc154954359)

[Hình 1.9. Sắp xếp sản phẩm theo “Sam Sung” 29](#_Toc154954360)

[Hình 1.10. Giao diện trang thêm dữ liệu. 30](#_Toc154954361)

[Hình 1.11. Giao diện xóa sản phẩm. 31](#_Toc154954362)

[Hình 1.12. Sản phẩm được xóa thành công. 32](#_Toc154954363)

[Hình 1.13. Hiển thị trang chỉnh sửa dữ liệu. 32](#_Toc154954364)

[Hình 1.14. Giao diện màn hình chi tiết sản phẩm. 33](#_Toc154954365)

Danh sách viết tắt

|  |  |
| --- | --- |
| **Từ viết tắt** | **Cụm từ chi tiết** |
| UI | User Interface |
| HTTP | Hypertext Transfer Protocol |
| BFD | Business Function Diagram |
| DBMS | Database Management System |

1. Tổng Quan Đề Tài
2. Giới thiệu.

Trong thời đại công nghệ hiện nay, ứng dụng di động đóng vai trò quan trọng trong việc tối ưu hóa quy trình quản lý và sử dụng các thiết bị điện tử. Trong ngữ cảnh này, dự án lập trình ứng dụng Android "Quản lý loa Vinh Kiệt" ra đời nhằm mục đích giải quyết những thách thức trong việc quản lý và tận dụng tối đa tiềm năng của các loa điện tử.

1. Khảo sát hiện trạng.

Trong thời đại số ngày nay, âm thanh đóng vai trò quan trọng trong cuộc sống hàng ngày của chúng ta. Từ việc thưởng thức âm nhạc đến trải nghiệm giải trí tại nhà, loa thông minh đã trở thành một phần quan trọng của không gian sống hiện đại. Để đáp ứng nhu cầu ngày càng tăng về âm thanh chất lượng cao, có rất nhiều cửa hàng bán loa được mở cửa trên khắp cả nước. Với nhu cầu tạo sự thuận lợi cho việc quản lý cũng như phân phối sản phẩm của các cửa hàng. Dự án sẽ tập trung giải quyết những vấn đề này để mang lại trải nghiệm quản lý loa tốt nhất cho người dùng.

1. Mục tiêu đề tài.

Mục tiêu của dự án là xây dựng một ứng dụng di động trên nền tảng Android với khả năng quản lý toàn diện về loa. Ứng dụng sẽ cung cấp thông tin chi tiết về các loại loa, giúp người dùng dễ dàng quản lý sản phẩm của họ với các chức năng như “Thêm”, “Sửa”, “Xóa”. Ngoài ra, ứng dụng sẽ hỗ trợ theo dõi tình trạng bảo dưỡng, cập nhật phần mềm.

1. Phạm vi đề tài.

Phạm vi đề tài bao gồm:

* **Thông tin Loa:** Tổng hợp thông tin chi tiết về các loại loa, bao gồm tên, giá tiền, số lượng có trong kho, nhãn hàng sản xuất, hình ảnh về sản phẩm.
* **Quản lý sản phẩm:** Cho phép người dùng quản lý các sản phẩm có trong kho với các chức năng như: hiển thị giao diện các sản phẩm đang tồn tại trong kho, thêm sản phẩm vào kho, xóa sản phẩm, chỉnh sửa thông tin về sản phẩm khi có những dữ liệu sai về sản phẩm đó.

1. Cấu trúc đề tài.

Cấu trúc đề tài của dự án được trình bày như sau:

* **Tiêu đề.**
* **Giới thiệu.**
* **Tổng quan về đề tài.**
* **Mục tiêu và phạm vi.**
* **Dữ liệu và thu thập dữ liệu**
* **Phân tích dữ liệu**
* **Kết quả dự kiến**
* **Hạn chế và hướng phát triển.**
* **Kết luận.**
* **Tài liệu tham khảo.**

1. CƠ SỞ LÝ THUYẾT
2. Giới thiệu tổng quan lập trình Android.
3. Sơ lược về Android.

Android, hệ điều hành di động phổ biến nhất trên thế giới, đã mở ra một hành tinh số hoá với hàng tỷ thiết bị sử dụng. Được phát triển chính thức bởi Google, là môi trường tích hợp đầy đủ tính năng để phát triển ứng dụng Android. Nó cung cấp trình biên dịch, trình giả lập, gỡ lỗi, và nhiều tiện ích hỗ trợ khác giúp nhà phát triển tối ưu hóa quá trình phát triển. Java và Kotlin là hai ngôn ngữ lập trình chính được sử dụng trong phát triển ứng dụng Android.

1. Cấu trúc dự án Android.

Một dự án Android bao gồm nhiều phần quan trọng như Activity, Fragment, Service, Broadcast Receiver, và Content Provider. Activity là thành phần chính đại diện cho một màn hình giao diện người dùng, trong khi Fragment là một phần nhỏ của Activity, giúp quản lý giao diện hiển thị linh hoạt hơn.

1. Quản lý tài nguyên.

Android có hệ thống quản lý tài nguyên mạnh mẽ, từ hình ảnh đến âm thanh và dữ liệu. Resource Manager giúp tối ưu hóa việc sử dụng bộ nhớ và tài nguyên để đảm bảo ứng dụng chạy mượt mà trên mọi thiết bị.

1. Các thành phần ứng dụng của Android.
2. Activity (Hoạt động).

Activity là một thành phần chính trong Android, đại diện cho một màn hình giao diện người dùng. Mỗi ứng dụng có thể có nhiều Activity, và chúng có thể tương tác với nhau.

1. Service (Dịch vụ).

Service là một thành phần không có giao diện người dùng, chạy ngầm để thực hiện các công việc dài hạn mà không làm gián đoạn trải nghiệm người dùng. Dịch vụ thường được sử dụng để thực hiện các nhiệm vụ như tải dữ liệu từ internet, chơi nhạc ngầm, và các công việc background khác.

1. Broadcast Receiver (Bộ Nhận Tin).

Broadcast Receiver là thành phần nhận và xử lý các sự kiện (broadcasts) từ hệ thống hoặc từ ứng dụng khác. Nó cho phép ứng dụng phản ứng vào các sự kiện như pin yếu, cuộc gọi đến, hoặc tin nhắn.

1. Fragment.

Fragment là một phần của Activity, giúp quản lý giao diện người dùng một cách linh hoạt hơn. Nhiều Fragment có thể tồn tại trong một Activity, và chúng có thể tái sử dụng được.

1. Intent.

Intent là một đối tượng được sử dụng để giao tiếp giữa các thành phần khác nhau của ứng dụng hoặc giữa các ứng dụng khác nhau. Nó được sử dụng để chuyển dữ liệu hoặc kích hoạt một thành phần khác.

1. Layouts và Views.

Layouts định nghĩa cách các thành phần giao diện người dùng được sắp xếp trên màn hình. Views là các thành phần cụ thể như Button, TextView, EditText, và nhiều loại khác, được sử dụng để xây dựng giao diện người dùng.

1. Resource Files.

Resource files bao gồm các tài nguyên như hình ảnh, âm thanh, và chuỗi văn bản được sử dụng trong ứng dụng. Chúng được tổ chức trong các thư mục res/ của dự án.

1. Manifest File.

Manifest là một tệp XML chứa thông tin cơ bản về ứng dụng, bao gồm danh sách các thành phần, quyền truy cập, phiên bản ứng dụng, và các cấu hình khác.

1. Giao diện và xử lý sự kiện trong Android.
2. Giao diện người dùng (UI).

Giao diện người dùng (UI) trong Android được xây dựng bằng cách sử dụng tệp XML để đặc tả cấu trúc của giao diện và Java (hoặc Kotlin) để xử lý sự kiện và tương tác người dùng. Dưới đây là một hướng dẫn cơ bản về cách tạo giao diện và xử lý sự kiện trong Android.

Dưới đây là một ví dụ về tạo giao diện trong XML:

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/buttonClick"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Nhấn vào" />

</RelativeLayout>

1. Xử lý sự kiện.

Các thành phần UI có thể xử lý các sự kiện của người dùng. Sự kiện của người dùng là các hành động mà người dùng thực hiện trên các thành phần UI, chẳng hạn như nhấn vào, kéo hoặc trượt. Để xử lý một sự kiện của người dùng, thành phần UI phải đăng ký một trình xử lý sự kiện. Trình xử lý sự kiện là một phương thức sẽ được gọi khi sự kiện xảy ra.

Ví dụ: để xử lý sự kiện nhấn vào nút, ta có thể đăng ký một trình xử lý sự kiện cho nút như sau:

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Thực hiện hành động khi người dùng nhấn vào nút

}

});

1. Xử lý giao tiếp giữa Components.

Có nhiều cách để xử lý giao tiếp giữa các thành phần (components) khác nhau của ứng dụng. Dưới đây là một số cách phổ biến được sử dụng:

* **Intent:** **Intent** là một cách phổ biến để chuyển dữ liệu giữa các thành phần khác nhau của ứng dụng. có thể sử dụng **Intent** để mở một **Activity** mới, kích hoạt một **Service**, hoặc gửi dữ liệu giữa các thành phần. Dưới đây là một ví dụ:

Gửi dữ liệu từ Activity A đến Activity B:

// Trong Activity A

Intent intent = new Intent(this, ActivityB.class);

intent.putExtra("key", "value");

startActivity(intent);

// Trong Activity B

Intent intent = getIntent();

String data = intent.getStringExtra("key");

* **BroadcastReceiver:** BroadcastReceiver là một cơ chế để gửi và nhận các thông điệp (broadcasts) giữa các thành phần của ứng dụng.

Ví dụ gửi broadcast từ Activity:

Intent intent = new Intent("custom-event");

intent.putExtra("key", "value");

sendBroadcast(intent);

Nhận broadcast trong BroadcastReceiver:

public class MyReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

String data = intent.getStringExtra("key");

// Xử lý dữ liệu nhận được

}

}

1. Xử lý giao tiếp với Server.

Giao tiếp với server là một phần quan trọng trong phát triển ứng dụng di động. Trong Android, có một số cách để thực hiện giao tiếp với server, chẳng hạn như sử dụng HTTP requests, Retrofit, Volley, AsyncTask, và thậm chí là thư viện hỗ trợ như OkHttp.

1. Tài nguyên Ứng dụng trong Android.

Tài nguyên ứng dụng (resources) trong Android bao gồm một loạt các loại tệp và tài nguyên, như hình ảnh, văn bản, biểu tượng, màu sắc, âm thanh, và nhiều loại khác. Các tài nguyên này giúp bạn tách rời mã nguồn ứng dụng từ các giá trị cụ thể và cung cấp một cách dễ dàng để quản lý và thay đổi giao diện người dùng, hình ảnh, và nội dung khác mà không cần phải sửa đổi mã nguồn.

1. **Layouts:** Chứa các tệp XML định nghĩa cấu trúc của các giao diện người dùng (UI).
2. **Menu:** Chứa các tệp XML định nghĩa các menu của ứng dụng.
3. **Drawable:** Chứa hình ảnh, biểu tượng và các tài nguyên về đồ họa khác.
4. **Mipmap:** Chứa biểu tượng ứng dụng cho các kích thước khác nhau.
5. **Strings:** Chứa chuỗi văn bản sử dụng trong ứng dụng.
6. **Colors:** Chứa định nghĩa màu sắc được sử dụng trong ứng dụng.
7. Đa ngôn ngữ và xử lý Tập tin.
8. **Đa ngôn ngữ (Multilingual Support):** Android hỗ trợ đa ngôn ngữ thông qua việc sử dụng tài nguyên chuỗi (strings) dựa trên các tệp resource dành cho mỗi ngôn ngữ cụ thể. Dưới đây là cách hỗ trợ đa ngôn ngữ trong Android:

* **Tạo các tệp Strings cho mỗi ngôn ngữ**.
* **Sử dụng Chuỗi trong Code.**
* **Phân biệt theo ngôn ngữ thiết bị**: Android sẽ tự động chọn tệp strings.xml phù hợp với ngôn ngữ của thiết bị. Khi sử dụng getResources().getString(R.string.hello), Android sẽ tự động chọn giá trị từ tệp strings.xml phù hợp với ngôn ngữ của thiết bị.

1. **Xử Lý Tệp (File Handling):** Android cung cấp nhiều cách để xử lý tệp, chẳng hạn như đọc và ghi dữ liệu vào bộ nhớ trong, đọc tệp từ tài nguyên raw hoặc assets, và thậm chí là thực hiện các thao tác với bộ nhớ SD (nếu được cấp quyền).

* **Đọc tệp từ tài nguyên raw:**

// Đọc tệp từ thư mục res/raw

InputStreaminputStream= getResources().openRawResource(R.raw.my\_file);

inputStream.close();

* **Đọc và ghi tệp trong bộ nhớ ngoại vi (SD Card):**

// Kiểm tra quyền truy cập

if(ContextCompat.checkSelfPermission(this,Manifest.permission.WRITE\_EXTERNAL\_STORAGE)== PackageManager.PERMISSION\_GRANTED) {

// Nếu có quyền, thực hiện các thao tác với SD Card

} else {

ActivityCompat.requestPermissions(this,

new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE},

MY\_PERMISSIONS\_REQUEST\_WRITE\_EXTERNAL\_STORAGE);

}

* **Đọc và ghi tệp trong bộ nhớ trong:**

// Đọc dữ liệu từ tệp trong bộ nhớ trong

FileInputStream fis = openFileInput("my\_file.txt");

// Ghi dữ liệu vào tệp trong bộ nhớ trong

FileOutputStreamfos=openFileOutput("my\_file.txt",Context.MODE\_PRIVATE);

1. Asset, SharePreferences và lưu trữ dữ liệu.
2. **Asset.**

Assets là một thư mục trong dự án Android chứa các tệp tài nguyên không được xử lý bởi Android, chẳng hạn như HTML, CSS, hoặc dữ liệu đa phương tiện. Dữ liệu trong thư mục assets có thể được truy cập thông qua AssetManager.

Đọc dữ liệu từ Asset:

// Mở một InputStream từ tệp trong thư mục assets

AssetManager assetManager = getAssets();

InputStream inputStream = assetManager.open("my\_file.txt");

inputStream.close();

1. **SharedPreferences.**

SharedPreferences là một cách đơn giản để lưu trữ dữ liệu theo cặp key-value. Được sử dụng để lưu trữ các thiết lập, cài đặt, hoặc dữ liệu nhỏ không đòi hỏi cơ sở dữ liệu.

* **Lưu trữ dữ liệu vào SharedPreferences:**

// Lấy ra SharedPreferences

SharedPreferences sharedPreferences =getSharedPreferences("MyPreferences", Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("username", "john\_doe");

editor.putInt("score", 100);

editor.apply();

* **Đọc dữ liệu từ SharedPreferences:**

// Lấy ra SharedPreferences

SharedPreferences sharedPreferences = getSharedPreferences("MyPreferences", Context.MODE\_PRIVATE);

String username = sharedPreferences.getString("username", "");

int score = sharedPreferences.getInt("score", 0);

1. **Lưu Trữ Dữ Liệu.**

* **SQLite Database:** Android có sẵn một cơ sở dữ liệu SQL gọi là SQLite. Bạn có thể sử dụng nó để lưu trữ và truy xuất dữ liệu trong ứng dụng.
* **Room Database:** Room là một thư viện trên cơ sở SQLite giúp đơn giản hóa công việc làm việc với cơ sở dữ liệu.
* **Lưu trữ dưới dạng JSON hoặc XML.**

1. SQLite.

SQLite là một hệ quản trị cơ sở dữ liệu nhỏ, nhẹ, và phổ biến, được tích hợp sẵn trong Android. Nó cung cấp một cơ sở dữ liệu SQL đầy đủ và hoạt động trên nền tảng di động. Dưới đây là cách sử dụng SQLite:

1. **Tạo hoặc Mở Cơ Sở Dữ Liệu:**

public class MyDatabaseHelper extends SQLiteOpenHelper {

private static final String DATABASE\_NAME = "my\_database";

private static final int DATABASE\_VERSION = 1;

public MyDatabaseHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

String createTableQuery = "CREATE TABLE my\_table ("

+ "\_id INTEGER PRIMARY KEY AUTOINCREMENT,"

+ "name TEXT,"

+ "age INTEGER)";

db.execSQL(createTableQuery);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

1. **Thêm dữ liệu:**

// Tạo một đối tượng lớp trợ giúp

MyDatabaseHelper dbHelper = new MyDatabaseHelper(context);

// Đọc hoặc tạo cơ sở dữ liệu

SQLiteDatabase database = dbHelper.getReadableDatabase();

1. **Đọc dữ liệu:**

String[] projection = {"\_id", "name", "age"};

Cursor cursor = database.query(

"my\_table",

projection,

null,

null,

null,

null,

null

);

while (cursor.moveToNext()) {

long itemId = cursor.getLong(cursor.getColumnIndexOrThrow("\_id"));

String name = cursor.getString(cursor.getColumnIndexOrThrow("name"));

int age = cursor.getInt(cursor.getColumnIndexOrThrow("age"));

// Sử dụng dữ liệu

}

cursor.close();

1. **Cập nhật dữ liệu:**

ContentValues updatedValues = new ContentValues();

updatedValues.put("age", 26);

int rowsAffected = database.update(

"my\_table",

updatedValues,

"name=?",

new String[]{"John"}

);

1. **Xóa dữ liệu:**

int rowsDeleted = database.delete(

"my\_table", // Tên bảng

"name=?", // Câu điều kiện WHERE

new String[]{"John"} // Giá trị của câu điều kiện WHERE

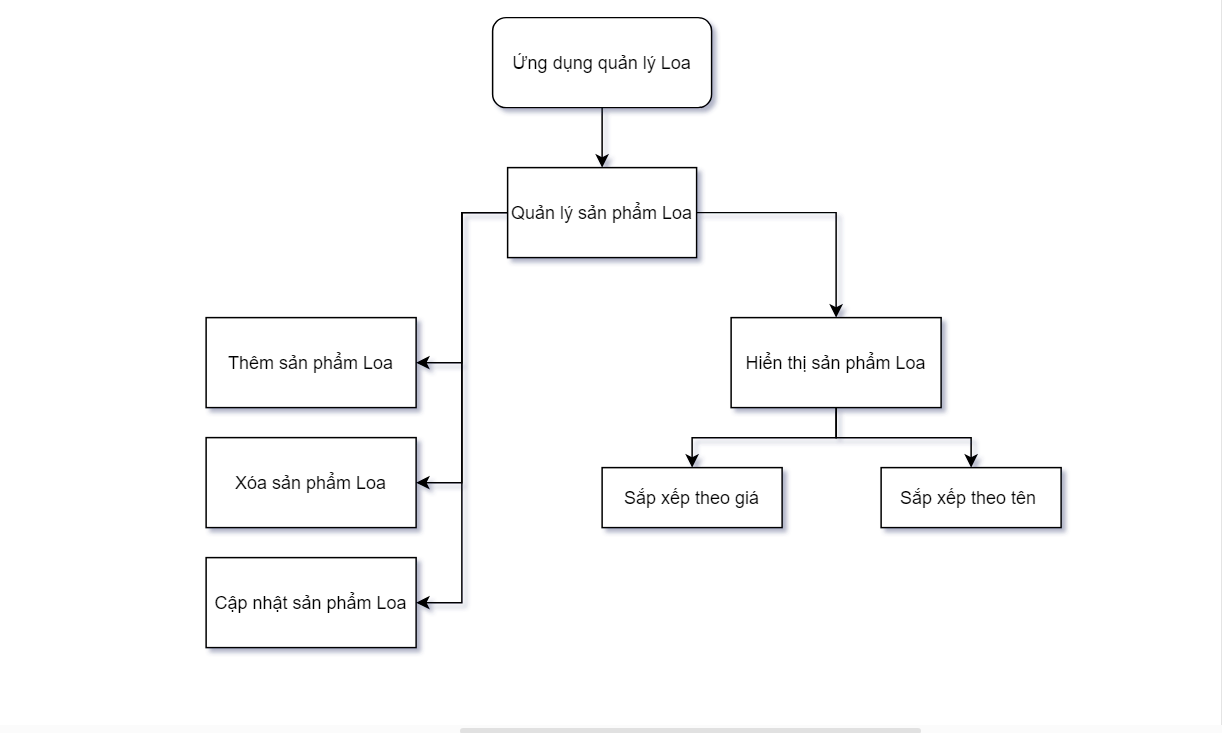
);

1. **Đóng cơ sở dữ liệu:**

dbHelper.close();

1. THỰC NGHIỆM VÀ TRIỂN KHAI
2. Thiết kế.
3. Mô hình chức năng (BFD).

Mô hình chức năng BFD (Business Function Diagram) là một loại sơ đồ mô hình hóa hệ thống thông tin, được sử dụng để mô tả các chức năng của hệ thống theo cách trực quan và dễ hiểu. BFD được sử dụng trong quá trình phân tích chức năng của hệ thống, nhằm xác định các chức năng cần thiết của hệ thống và mối quan hệ giữa các chức năng đó.

Sơ đồ tổng quát mô tả chức năng của ứng dụng quản lý Loa:

1. Sơ đồ tổng quát chức năng BFD.

|  |  |
| --- | --- |
| **Chức năng** | **Chi tiết** |
| Thêm dữ liệu sản phẩm Loa | Nhập tên loa, giá thành, số lượng sản phẩm, hình ảnh sản phẩm, thương hiệu sản xuất. |
| Cập nhật dữ liệu sản phẩm Loa | Chỉnh sửa các dữ liệu: tên loa, giá thành, số lượng sản phẩm, hình ảnh sản phẩm, thương hiệu sản xuất khi cần thiết |
| Xóa dữ liệu sản phẩm Loa | Xóa tất cả dữ liệu liên quan đến sản phẩm Loa được chọn. |
| Hiển thị sản phẩm | Hiển thị đầy đủ thông tin như: tên loa, giá thành, số lượng sản phẩm, hình ảnh sản phẩm, thương hiệu sản xuất lên màn hình. Sắp xếp danh sách sản phẩm hiển thị theo giá thành và nhãn hàng sản xuất. |

1. Bảng mô tả chi tiết chức năng.
2. Mô hình quan hệ dữ liệu.

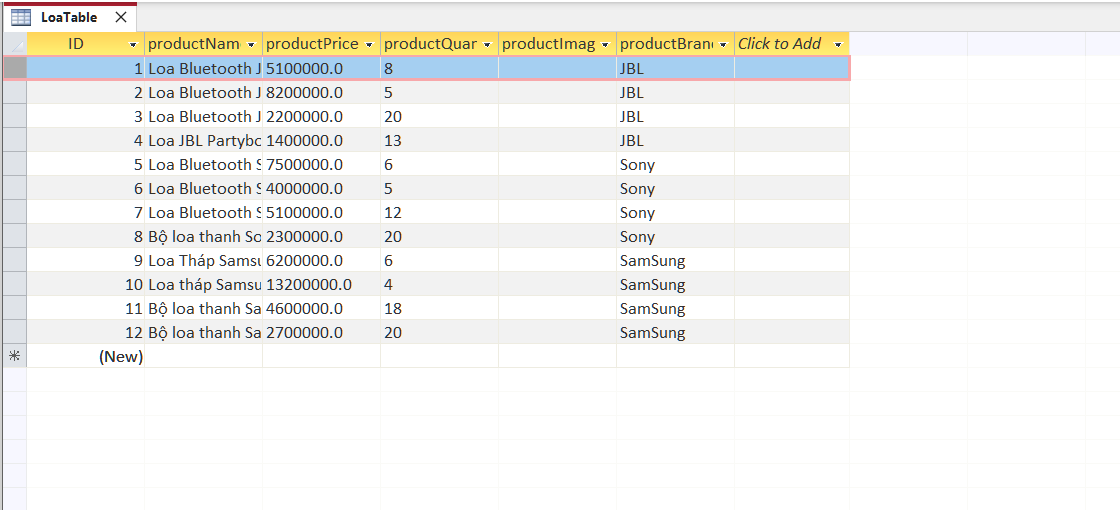
Mô hình dữ liệu quan hệ là một phương pháp tổ chức dữ liệu trong cơ sở dữ liệu quan hệ (Relational Database). Đây là một mô hình dữ liệu phổ biến được sử dụng trong hệ thống quản lý cơ sở dữ liệu (DBMS) để tổ chức và lưu trữ dữ liệu.

Mô hình dữ liệu quan hệ sử dụng các đối tượng như bảng, hàng, cột và khóa để biểu diễn và tổ chức thông tin.

1. Table “LoaTable” trong cơ sở dữ liệu.

Trong dự án các thành phần trên được mô tả tương ứng như sau:

* **Bảng (Table):** “LoaTable” được sử dụng để lưu trữ dữ liệu, đại diện cho sản phẩm Loa của hệ thống.
* **Cột (Column):** Trong “LoaTable” gồm có 6 column:
  + **“ID”:** Đại diện cho thuộc tính **mã số** của sản phẩm.
  + **“productName”:** Đại diện cho thuộc tính **tên** của sản phẩm loa.
  + **“productPrice”:** Đại diện cho thuộc tính **giá tiền** của sản phẩm loa
  + **“productQuantity”:** Đại diện cho thuộc tính **số lượng** của sản phẩm loa
  + **“productImage”:** Đại diện cho thuộc tính **hình ảnh** của sản phẩm loa
  + **“productBrand”:** Đại diện cho thuộc tính **tên thương hiệu** của sản phẩm loa
* **Khóa chính (Primary Key):** Là một hoặc nhiều cột có giá trị duy nhất để định danh mỗi bản ghi trong bảng. Đảm bảo tính duy nhất của mỗi hàng. Khóa chính được sử dụng trong “LoaTable” là column “ID”.
* **Hàng (Row):** Được gọi là bản ghi, mỗi hàng trong bảng chứa thông tin cụ thể về một thực thể. Mỗi hàng thường được định danh bằng một khóa chính.

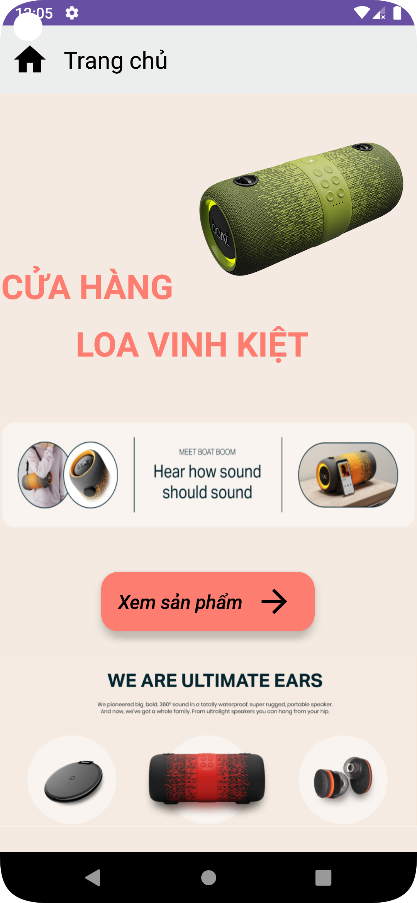


1. Thông tin về các hàng (Row) được thêm vào cơ sở dữ liệu

Mô hình dữ liệu quan hệ giúp tổ chức dữ liệu một cách logic và có thể dễ dàng truy xuất thông tin bằng cách sử dụng ngôn ngữ truy vấn SQL (Structured Query Language). Điều này làm cho việc quản lý và truy xuất dữ liệu trở nên hiệu quả và thuận tiện trong các hệ thống cơ sở dữ liệu lớn.

1. Cài đặt ứng dụng.
2. Màn hình chờ.

Ứng dụng sử dụng và thiết kế một màn hình chờ khi người dùng truy cập vào ứng dụng lần đầu tiên.



1. Giao diện màn hình chờ.

Màn hình chờ được thiết kế trong một file XML, và được thiết lập trong file AndroidManifest được xuất hiện đầu tiên khi ứng dụng khởi động.

Sử dụng các thuộc tính UI của Android như: “TextView” để hiển thị tiêu đề của ứng dụng, “ImageView” để hiển thị các hình ảnh trang trí, “Button” xử lý tác vụ chuyển trang.

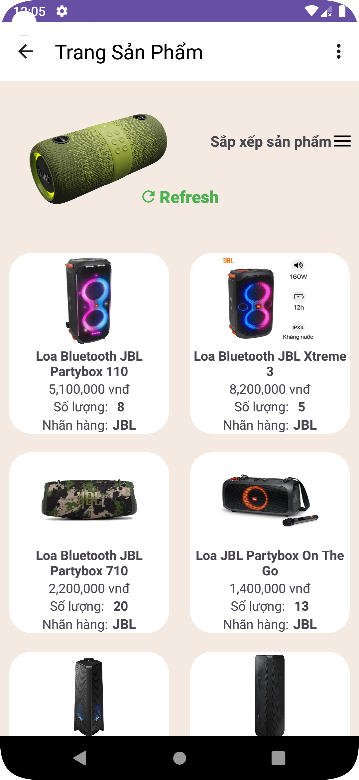
Màn hình chờ có một số tác dụng chính:

* **Thông báo về sự khởi động:** thông báo cho người dùng biết rằng ứng dụng đang được khởi động và chuẩn bị sẵn sàng.
* **Tạo ấn tượng đầu tiên:** màn hình chờ có thể được thiết kế để tạo ra một ấn tượng đầu tiên tích cực và chuyên nghiệp với người dùng.
* **Tăng tính thẩm mỹ và chuyên nghiệp:** một màn hình chờ được thiết kế đẹp mắt và chuyên nghiệp có thể tăng cường sự chuyên nghiệp của ứng dụng và tạo ra ấn tượng tích cực với người dùng.

1. Màn hình chính.

Màn hình chính được hiển thị khi người dùng nhấn vào nút (Button) **“Xem sản phẩm”** trong giao diện màn hình chờ.

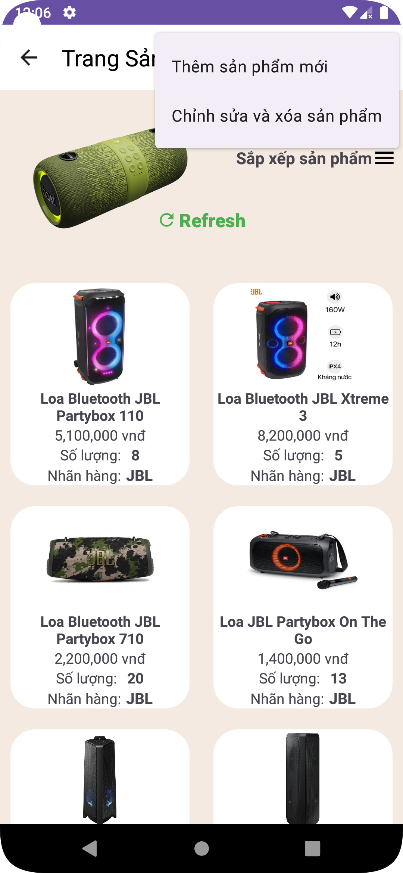
Giao diện màn hình chính được thiết kế bởi các thành phần UI như: “TextView” hiển thị thông tin. “ImageView” hiển thị thông tin hình ảnh, “Button” xử lý các tác vụ được cài đặt, “GridView” hiển thị danh sách sản phẩm, “ActionBar” hiện thị tên trang và quay lại trang trước đó.



1. Giao diện của màn hình chính

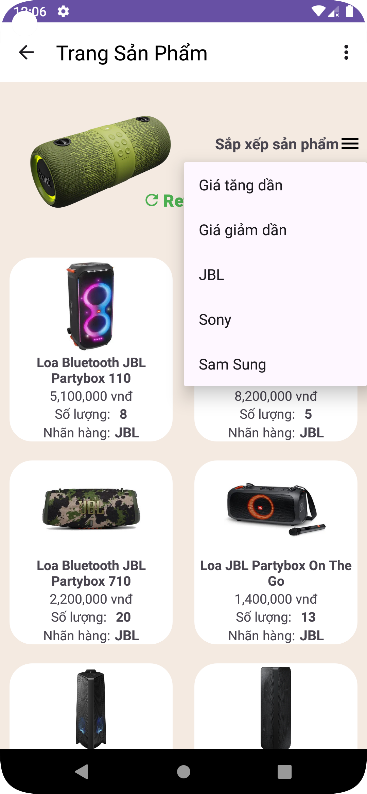
Ngoài ra màn hình chính còn cung cấp các chức năng như:

* **Chuyển trang:** Khi người dùng nhấn vào “Thêm sản phẩm”, “Sửa và xóa sản phẩm” ứng dụng sẽ di chuyển đến các trang tương ứng.



1. Menu chuyển trang trong màn hình chính.

* **Sắp xếp sản phẩm:** Ứng dụng cung cấp sắp xếp sản phẩm theo “Giá tăng dần”, “Giá giảm dần”, “Nhãn hàng được chọn”. Khi người dùng nhấn vào các mục tương ứng, ứng dụng sẽ hiển thị thông tin sản phẩm.



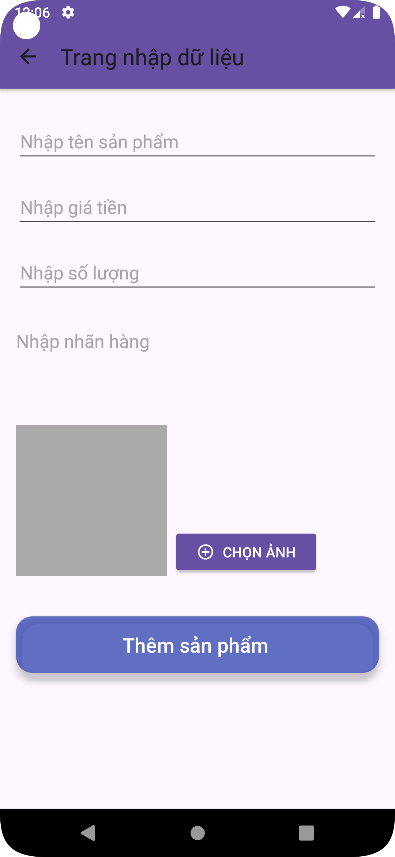
1. Menu sắp xếp sản phẩm trong màn hình chính.

Chọn sắp xếp theo **“Sam Sung”:**

1. Sắp xếp sản phẩm theo “Sam Sung”
2. Màn hình thêm dữ liệu.

Màn hình thêm dữ liệu sản phẩm vào bảng được hiển thị sau khi người dùng nhấn vào mục “Thêm sản phẩm mới” ở mục Menu trong giao diện màn hình chính.

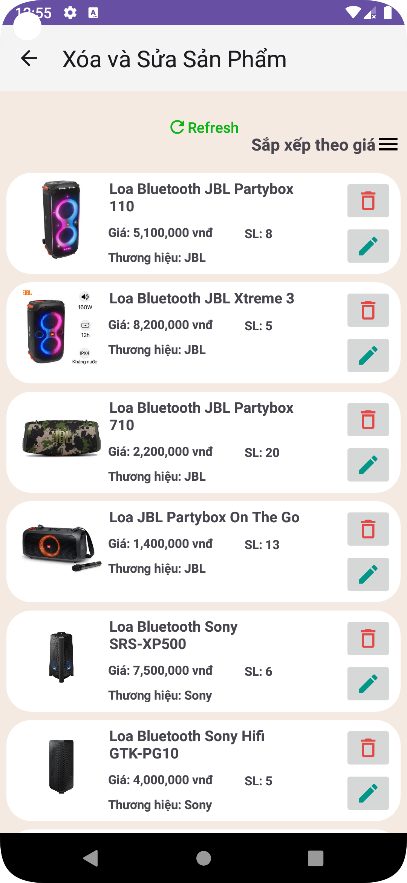
Màn hình thêm dữ liệu được thiết kế bởi các thành phần UI trong Android như: **“EditText”** cung cấp ô nhập dữ liệu dạng văn bản, **“ImageView”** hiển thị hình ảnh được chọn từ thiết bị, **“Button”** chọn hình ảnh từ kho lưu trữ thiết bị và **“Button”** xác nhận thêm dữ liệu vào cơ sở dữ liệu.



1. Giao diện trang thêm dữ liệu.
2. Màn hình xóa dữ liệu.

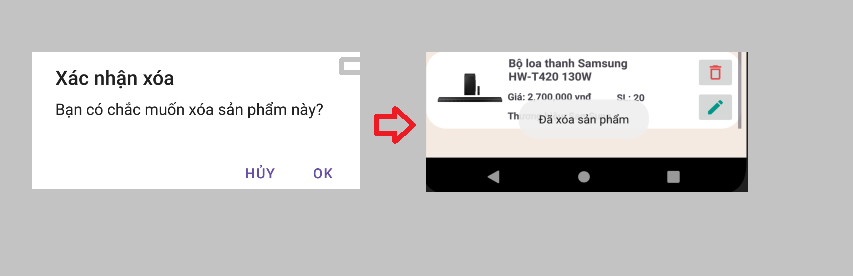
Màn hình xóa dữ liệu được hiển thị sau khi nhấn vào mục “Chỉnh sửa và xóa sản phẩm” trong Menu của màn hình chính.

Màn hình xóa dữ liệu được thiết kế bởi các thành phần giao diện UI như: “GridView” danh sách sản phẩm, “TextView” hiển thỉ thông tin như tên, giá, số lượng, nhãn hàng, “ImageView” hiển thị hình ảnh sản phẩm, “Button Xóa” thiết kế để xóa sản phẩm khi được nhấn vào.



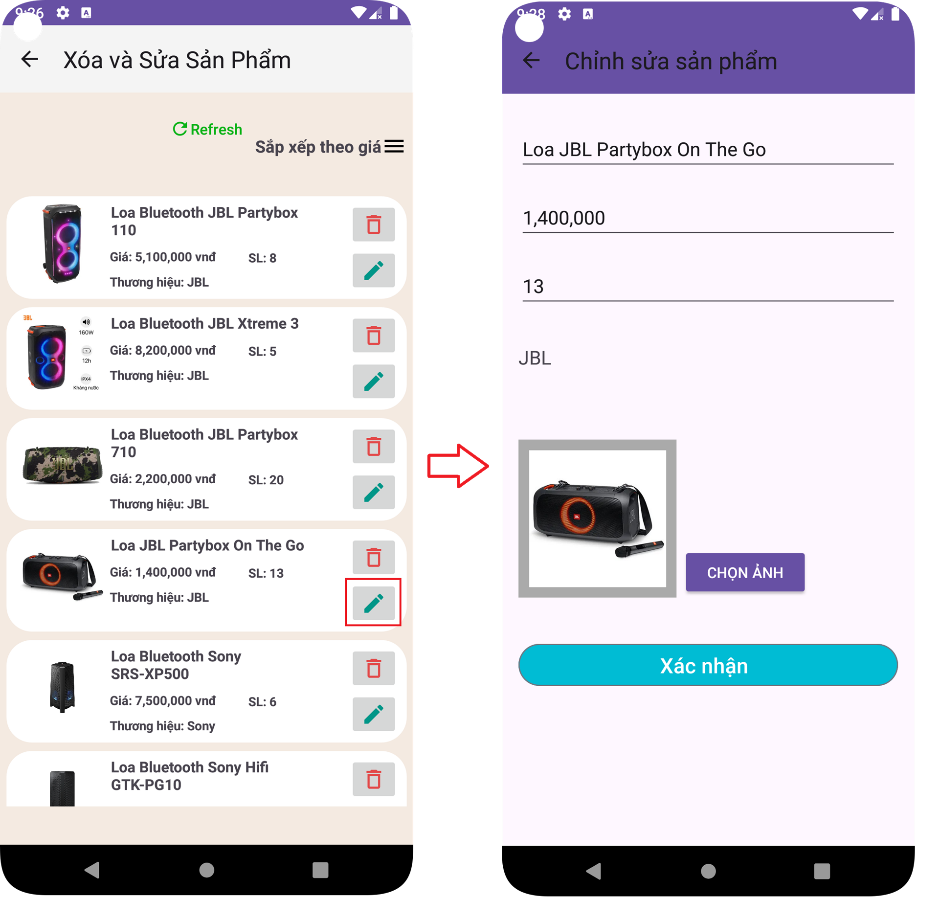
1. Giao diện xóa sản phẩm.

Sau khi nhấn vào “button xóa” ứng dụng sẽ hiện lên một thông báo, nếu chọn đồng ý sản phẩm sẽ được xóa.



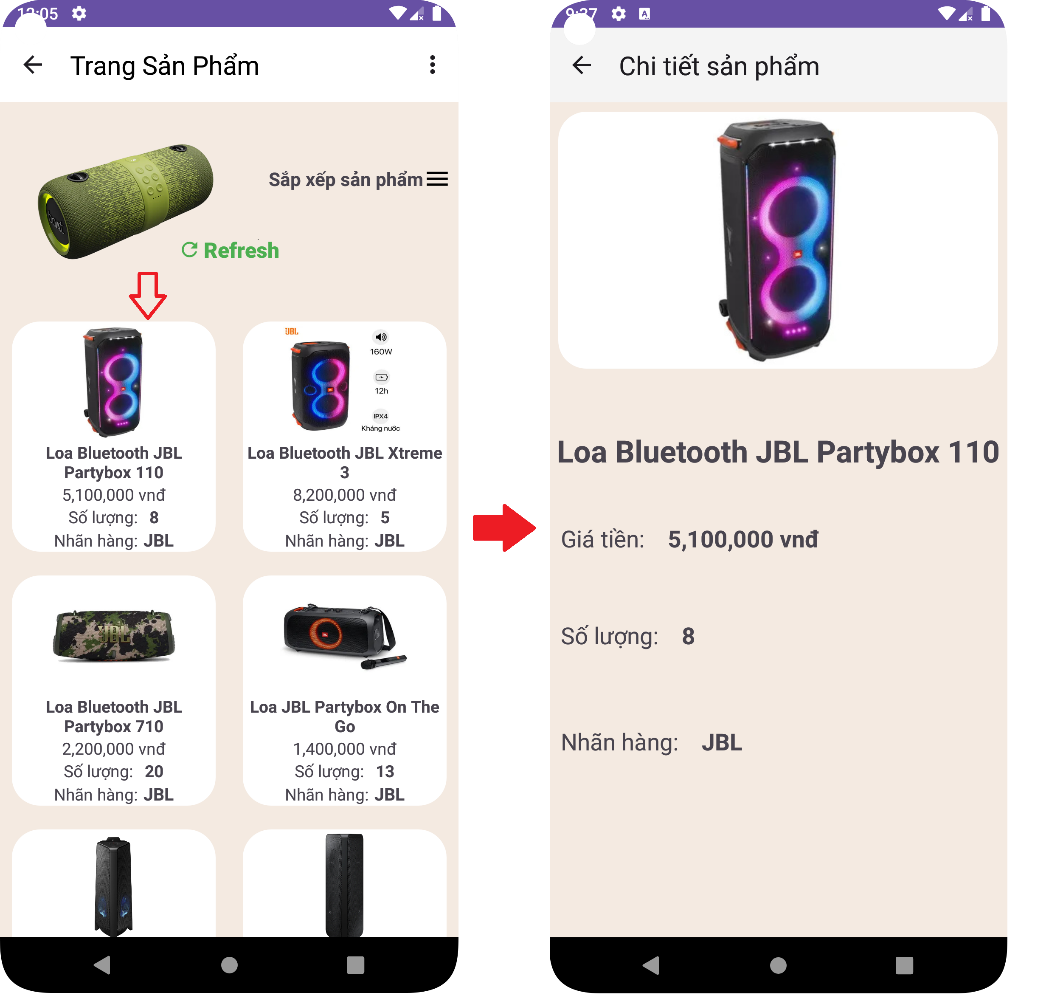
1. Sản phẩm được xóa thành công.
2. Màn hình chỉnh sửa dữ liệu.

Màn hình chỉnh sửa dữ liệu được hiển thị khi nhấn vào “button chỉnh sửa” trong giao diện trang Sửa và Xóa sản phẩm.



1. Hiển thị trang chỉnh sửa dữ liệu.
2. Màn hình chi tiết sản phẩm.

Màn hình hiển thị chi tiết sản phẩm được hiển thị sau khi người dùng nhấn vào sản phẩm trong giao diện màn hình chính.



1. Giao diện màn hình chi tiết sản phẩm.
2. KẾT LUẬN
3. Kết quả đạt được.

Trong quá trình phát triển và phân tích dự án ứng dụng quản lý Loa tôi đã chú trọng vào nhiều khía cạnh quan trọng như giao diện, chức năng, cơ sở dữ liệu. Một số yếu tố tôi đạt được trong quá trình phát triển, cuối cùng kết hợp các yếu tố đó mang lại trải nghiệm sử dụng ứng dụng tốt:

* **Giao diện thân thiện:** Giao diện là một phần không thể thiếu đối với một phần mềm ứng dụng, cho nên tôi đã chú trọng trong việc thiết kế một giao diện người dùng ấn tượng, dễ sử dụng.
* **Cung cấp các chức năng:** Các chức năng được cung cấp trong phần mềm như: Xem sản phẩm, thêm dữ liệu, xóa dữ liệu, cập nhật dữ liệu được thiết kế ổn định khi sử dụng, giảm thiểu tình trạng lỗi khi sử dụng ứng dụng.

1. Hướng phát triển.

Với một ứng dụng được phát triển trong khoảng thời gian ngắn, chúng tôi còn rất nhiều ý tưởng đang được ấp ủ, nhiều tính năng còn đang dang dở. Một số tính năng, ý tưởng đã được lên kế hoạch và cập nhật trong thời gian tới có thể được kể đến:

* + - Thiết kế thêm cơ sở dữ liệu về khách hàng, nhân viên.
    - Cập nhật giao diện người dùng tối ưu hơn.
    - Thêm tính năng quản lý khách hàng và nhân viên.
    - Đưa ứng dụng lên Google Store giúp người dùng dễ dàng tiếp cận.

Tài liệu tham khảo

1. Giao diện hình ảnh: [*liên kết*](https://dribbble.com/shots/17416043-Portable-Speaker-Store-Website)
2. Dữ liệu hình ảnh:[*liên kết*](https://www.dienmayxanh.com/dan-loa-dvd-sony)

Link nộp đề tài

1. Link Google Drive: [*liên kết*](https://drive.google.com/drive/folders/1yKR9IMAXf11NxmumUtDxfUTOhstvOJ5p?usp=drive_link)