

**Trường Đại Học Kỹ Thuật – Công Nghệ Cần Thơ**

**Khoa Công Nghệ Thông Tin**

****

**Đồ Án 2**

**Tạo một ứng dụng có sử dụng Java hoặc C#**

**Design Pattern Strategy**

**SVTH: Võ Triệu Long**

**MSSV: 1500205**

**GVHD: Th.s Hà Xuân Sơn**

**Cần Thơ, ngày 1 tháng 6 năm 2018**

Contents

MỤC LỤC

[**LỜI NÓI ĐẦU** 4](#_Toc515780906)

[**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN** 6](#_Toc515780907)

[**CHƯƠNG 1: TỔNG QUAN** 7](#_Toc515780908)

[**1/ Đặt vấn đề:** 7](#_Toc515780909)

[**2/ Mục đích đề tài:** 7](#_Toc515780910)

[**3/ Phạm vi đề tài:** 7](#_Toc515780911)

[**4/ Phạm vi thực hiện:** 7](#_Toc515780912)

[**5/ Những chức năng chính của ứng dụng là là:** 7](#_Toc515780913)

[**CHƯƠNG 2: KHẢO SÁT VÀ ĐẶT BÀI TOÁN** 8](#_Toc515780914)

[**I/ Khảo sát thực tế** 8](#_Toc515780915)

[**II/ Đặc tả yêu cầu** 8](#_Toc515780916)

[**1. Tác nhân lên ứng dụng** 8](#_Toc515780917)

[**2. Yêu cầu đạt được của ứng dụng** 8](#_Toc515780918)

[**CHƯƠNG 3** 9](#_Toc515780919)

[**Tìm Hiểu Design Pattern Và Strategy Pattern** 9](#_Toc515780920)

[**I/ Tìm Hiểu Design Pattern** 9](#_Toc515780921)

[1. Giới thiệu và phân loại thiết kế mẫu ( Design Pattern) 9](#_Toc515780922)

[**2. PATTERN là gì?** 9](#_Toc515780923)

[**3. Đặc điểm chung** 9](#_Toc515780924)

[**3.1. Pattern được hiểu theo nghĩa tái sử dụng ý tưởng hơn là mã lệnh.** 9](#_Toc515780925)

[**3.2. Pattern hỗ trợ tái sử dụng kiến trúc và mô hình thiết kế phần mềm theo quy mô lớn.** 9](#_Toc515780926)

[**3.3. Pattern đa tương thích.** 10](#_Toc515780927)

[**4. Phân loại pattern** 10](#_Toc515780928)

[**5. Tác dụng của các Design Pattern** 10](#_Toc515780929)

[**6*.* Ưu điểm của design pattern** 10](#_Toc515780930)

[**II/Strategy Pattern** 11](#_Toc515780931)

[**1.Strategy Pattern là gì?** 11](#_Toc515780932)

[**2.Tại sao chúng ta cần sử dụng Strategy Pattern?** 11](#_Toc515780933)

[**3.Thuật toán Strategy trong object** 11](#_Toc515780934)

[**4.Lược đồ Strategy Pattern** 11](#_Toc515780935)

[**5. Lược đồ hoạt động Strategy Pattern** 12](#_Toc515780936)

[**CHƯƠNG 4. GIỚI THIỆU CÔNG NGHỆ** 13](#_Toc515780937)

[**I. Ngôn ngữ lập trình Java** 13](#_Toc515780938)

[**1. Khái niệm** 13](#_Toc515780939)

[**2.Một số đặc điểm nổi bật của ngôn ngữ Java** 13](#_Toc515780940)

[**II.Giới thiệu về Android Studio** 14](#_Toc515780941)

[**1.Android Studio là gì**? 14](#_Toc515780942)

[**2. Giao diện người dùng** 14](#_Toc515780943)

[**III/ Chuẩn bị môi trường làm việc với android studio** 14](#_Toc515780944)

[**1. Tải và cài đặt Java JDK** 14](#_Toc515780945)

[**2. Tải và cài đặt android studio** 14](#_Toc515780946)

[**3. Tải và cài đặt Genymotion** 19](#_Toc515780947)

[**CHƯƠNG 5: PHÂN TÍCH HỆ THỐNG** 23](#_Toc515780948)

[**1. Sơ đồ use case** 23](#_Toc515780949)

[**2. Các giải thuật** 23](#_Toc515780950)

[**CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 24](#_Toc515780951)

[**1.Kết quả đạt được** 24](#_Toc515780952)

[**2.Việc chưa làm được** 26](#_Toc515780953)

[**3. Hướng phát triển** 26](#_Toc515780954)

[**Tài Liệu Tham Khảo** 26](#_Toc515780955)

# **LỜI NÓI ĐẦU**

Trong thời đại công nghệ thông tin hiện nay, các ngôn ngữ lập trình càng ngày đa dạng hơn để nhớ hết tất cả những chi tiết là một công việc vô cùng khó khăn. Với việc không nắm bắt được tổng quát công việc đem lại nhiều phiền toái trong công việc. Khi đó, mã nguồn đang viết có thể vẫn còn làm việc tốt đẹp, nhưng trừ khi lập trình viên bao quát được bức tranh lớn hơn, lúc đó mã nguồn mới thực sự hoàn hảo.

Những vấn đề nghiêm trong thực sự thhường xuyên xuất hiện khi ta chỉnh sửa chúng trong quá trình lâp trình. Những nhà phát triển thường tự mình khắc phục bằng cách tìm ra lỗi và chỉnh sữa chúng cho hết lỗi. Tuy nhiên trong môi trường làm việc, các lập trình viên thường phải bỏ ra nhiều thời gian để bảo trì và chỉnh sửa những lỗi xải ra của công việc cũ và ít có thời gian để tập trung vào một công việc mới.

Và thật khó khăn khi cứ phải làm những công việc cũ như chỉnh sửa, tìm lỗi,... Giải pháp tốt nhất chính là đưa vào một quy chuẩn tổng hợp trong thiết kế và bảo trì, giúp cho người lập trình viên tránh đi những phiền toái và các rắc rối phát sinh trong ứng dụng, tạo điều kiện cho người lập trình viên tiết kiệm được thời gian và dễ dàng cho việc bảo trì ứng dụng.

Bằng những khinh nghiệm của các lập trình viên đi trước, các mẫu Design Pattern được ra đời nhằm tiết kiệm thời gian và công sức của người lập trình đới với những vấn đề thường gặp trong quá trình viết ứng dụng và bảo trì. Đươc coi là một dạng kỹ thuật lập trình chung cho tất cả lập trình viên, nên Design Pattern còn làm cho những dòng code khi lập trình viên thiết kế cũng dễ hiểu và dễ phân tích hơn, giúp giai đoạn sữa chữa, bảo trì cũng như nâng cấp phần mềm được dễ dàng hơn.

Ở đồ án 2 lần này, với yêu cầu viết ra một ứng dụng bằng ngôn ngữ lập trình Java hoặc C# có sử dụng Design Pattern.Với việc nghiên cứu về ngôn ngữ lập trình Java và làm việc trên môi trường Android Studio, em đã viết ra một ứng dụng nghe nhạc trên thiết bị thông minh chạy trên nền tảng hệ điều Android và có sử dụng Design Pattern trên khuôn mẫu Strategy Pattern. Em xin chân thành cám ơn thầy Hà Xuân Sơn đã hướng dẫn tận tình, giúp em hoàn thành đồ án này.

Sinh viên thực hiện:

Võ Triệu Long

# **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

# **CHƯƠNG 1: TỔNG QUAN**

Ứng dụng App nghe nhạc là một ứng dụng của của sự phát triển công nghệ hiện nay với những ưu điểm:

* Giúp cho người sử dụng thiết bị công nghệ như điện thoại có thể nghe nhạc mọi lúc mà không cần tới internet.
* App nghe nhạc có giao diện đơn giản giúp cho người sử dụng đễ theo tác trên App trong quá điều chỉnh âm thanh và chọn bài hát mình yêu thích.
* Ứng dụng App nghe nhạc trên điện thoại di động trên nền tảng Android với các phiên bảng khác nhau trên điện thoại di động như:

## **1/ Đặt vấn đề:**

* Khi thiết kế code, ở trường hợp cụ thể rằng một lớp chỉ có một thể hiện (instance) duy nhất và trong đó cung cấp một cổng giao tiếp chung nhất để truy cập vào lớp đó. Chúng ta cần sử dụng Strategy Pattern để giải quyết bài toán trên để tái sử dụng 1 lớp mà không cần phải khai báo lại nhiều lần, giảm bớt thời gian code và những lỗi vặt.

## **2/ Mục đích đề tài:**

Đề tài thực hiện nhằm xây dựng một ứng dụng phát nhạc có sử dụng Stategy Pattern

## **3/ Phạm vi đề tài:**

Ứng dụng phát nhạc có sử dụng Strategy Pattern

## **4/ Phạm vi thực hiện:**

Ứng dụng phát nhạc lấy file nhạc từ thư mục của thiết bị người dùng

## **5/ Những chức năng chính của ứng dụng là là:**

* Lấy file nhạc trực tiếp trong thư mục thiết bị của người dùng
* Chức năng nhấn chọn một bài hát
* Chức năng phát nhạc (play), tạm dừng (pause)
* Chức năng phát bài tiếp theo (next), phát lại bài trước (previous)

**CHƯƠNG 2: KHẢO SÁT VÀ ĐẶT BÀI TOÁN**

**I/ Khảo sát thực tế**

Trong thời hiên đại bây giời thì nhu cầu về giải trí của con người ngày càng tăng lên họ muống được giải trí mọi lúc mọi nơi như nghe những ca khúc họ yêu thích. Ngày xưa thì họ nghe nhạc thì chỉ qua băng đĩa hoặc các kênh truyền hình hoặc họ nghe nhạc trực tuyến qua internet. Dựa trên nhu cầu đó, đã tạo ra một ứng dụng nghe nhạc trên điện thoại của họ bằng cách họ sẽ tải những bài hát họ yêu thích về và nghe trên ứng dụng đó. Không cần internet thì họ vẫn có thể giải trí và nghe được những ca khúc mà họ yêu thích. Về giao diện của ứng dụng thì dễ nhìn và dễ thảo tác trên đó như chon bài hát, nhấn nút phát nhạc để bài nhạc bắt đầu chạy, nhấn nút chuyển bài để chuyển qua bài tiếp theo, nhấn nút trở về để trở về bài hát trước đó...

**II/ Đặc tả yêu cầu**

**1. Tác nhân lên ứng dụng**

Người quản trị:

- Là tác nhân chính lên ứng dụng

- Người quản trị có thể bảo trì, cập nhật ứng dụng, phát triển ứng dụng.

### **2. Yêu cầu đạt được của ứng dụng**

Giúp cho người quản trị có thể dễ dàng bảo trì, cập nhật và phát triển hệ thống

# **CHƯƠNG 3**

## **Tìm Hiểu Design Pattern Và Strategy Pattern**

## **I/ Tìm Hiểu Design Pattern**

### 1. Giới thiệu và phân loại thiết kế mẫu ( Design Pattern)

Trong phát triển phần mềm hiện đại, kiến trúc tổng thể của dự án đóng một vai trò quan trọng, đặc biệt với bộ khung (framework) và mẫu thiết kế (design pattern). Bài viết này sẽ giúp các bạn hiểu được một cách tổng quan về pattern cũng như cách thức thiết kế một số pattern tiêu biểu.

### **2. PATTERN là gì?**

Pattern mô tả một giải pháp chung đối với một vấn đề nào đó trong thiết kế thường được “lặp lại” trong nhiều dự án. Nói một cách khác, một pattern có thể được xem như một “khuôn mẫu” có sẵn áp dụng được cho nhiều tình huống khác nhau để giải quyết một vấn đề cụ thể. Trong bất kỳ hệ thống phần mềm hướng đối tượng nào chúng ta cũng có thể bắt gặp các vấn đề lặp lại.

### **3. Đặc điểm chung**

### **3.1. Pattern được hiểu theo nghĩa tái sử dụng ý tưởng hơn là mã lệnh.**

Pattern cho phép các nhà thiết kế có thể cùng ngồi lại với nhau và cùng giải quyết một vấn đề nào đó mà không phải mất nhiều thời gian tranh cãi. Trong rất nhiều trường hợp, dự án phần mềm thất bại là do các nhà phát triển không có được sự hiểu biết chung trong các vấn đề về kiến trúc phần mềm. Ngoài ra, pattern cũng cung cấp những thuật ngữ và khái niệm chung trong thiết kế. Nói một cách đơn giản, khi đề cập đến một pattern nào đấy, bất kỳ ai biết pattern đó đều có thể nhanh chóng hình dung ra “bức tranh” của giải pháp. Và cuối cùng, nếu áp dụng pattern hiệu quả thì việc bảo trì phần mềm cũng được tiến hành thuận lợi hơn, nắm bắt kiến trúc hệ thống nhanh hơn.

### **3.2. Pattern hỗ trợ tái sử dụng kiến trúc và mô hình thiết kế phần mềm theo quy mô lớn.**

Cần phân biệt design pattern với framework. Framework hỗ trợ tái sử dụng mô hình thiết kế và mã nguồn ở mức chi tiết hơn. Trong khi đó, design pattern được vận dụng ở mức tổng quát hơn, giúp các nhà phát triển hình dung và ghi nhận các cấu trúc tĩnh và động cũng như quan hệ tương tác giữa các giải pháp trong quá trình thiết kế ứng dụng đối với một chuyên khu riêng biệt.

### **3.3. Pattern đa tương thích.**

Pattern không phụ thuộc vào ngôn ngữ lập trình, công nghệ hoặc các nền tảng lớn như J2EE của Sun hay Microsoft .NET Framework. Tim năng ứng dụng của pattern là rất lớn. Các thiết kế dựa trên pattern được sử dụng khá nhiều ở các phần mềm mã nguồn mở, trong nền tảng J2EE hoặc .NET... Trong các dạng ứng dụng này, có thể dễ dàng nhận ra một s+ố tên lớp chứa các tiền tố hoặc hậu tố như Factory, Proxy, Adapter...

### **4. Phân loại pattern**

Pattern được phân loại ra làm 3 nhóm chính sau đây:

• Nhóm cấu thành (Creational Pattern): Gồm Factory, Abstract Factory, Singleton, Prototype, Builder... Liên quan đến quá trình khởi tạo đối tượng cụ thể từ một định nghĩa trừu tượng (abstract class, interface).

• Nhóm cấu trúc tĩnh (Structural Pattern): Gồm Proxy, Adapter, Wrapper, Bridge, Facade, Flyweight, Visitor... Liên quan đến vấn đề làm thế nào để các lớp và đối tượng kết hợp với nhau tạo thành các cấu trúc lớn hơn.

• Nhóm tương tác động (Behavioral Pattern): Gồm Observer, Strategy, Command, Iterator... Mô tả cách thức để các lớp hoặc đối tượng có thể giao tiếp với nhau.

### **5. Tác dụng của các Design Pattern**

- Sự thể hiện chung – Điều này cho phép các đối tượng được tạo ra trong hệ thống không cần phải định nghĩa một đặc tả kiểu lớp trong mã nguồn

- Đơn giản – Một vài mẫu làm cho việc khởi tạo đối tượng trở nên dễ dàng, vì vậy lớp “gọi” khởi tạo đối tượng không phải viết mã nhiều cũng như phức tạp.

### **6. Ưu điểm của design pattern**

- Design pattern giúp cho dự án của chúng ta dễ bảo trì, nâng cấp và mở rộng

- Hơn nữa do design pattern đã được các nhà nghiên cứu đúc kết ra nên khi sử dụng design pattern thì chúng ta sẽ hạn chế được các lỗi tiềmm ẩn.

- Giảm bớt thời gian và công sức suy nghĩ của lập trình viên khi gặp những trường hợp cụ thể trong giai đoạn thiết kế code

- Và cuối cùng là khi sử dụng design pattern thì sẽ giúp code của chúng ta sẽ dễ đọc hơn. Điều này rất có lợi khi làm việc nhóm.

## **II/Strategy Pattern**

### **1.Strategy Pattern là gì?**

Strategy pattern (mẫu chiến lược): hiểu một cách đơn giản thì đây là mẫu thiết kế giúp bạn trừu tượng hóa những hành vi (behavior, method, function) của một đối tượng bằng cách đưa ra những cài đặt vào những lớp khác nhau.

### **2.Tại sao chúng ta cần sử dụng Strategy Pattern?**

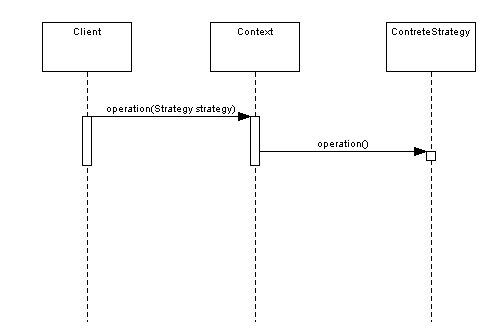
Trong phát triển phần mềm, đôi lúc ta có gặp một số tình huống mà ở đó một số class nó chỉ khác nhau ở thuật toán để thực hiện công việc. Thay vì tạo nhiều class gần nhứ giống nhau thì ta chỉ tách những phần thuật toán khác nhau ra thành những class khác. Sau đó việc lựa chọ sử dụng thuật toán nào đó sẽ quyết định.

### **3.Thuật toán Strategy trong object**

### 

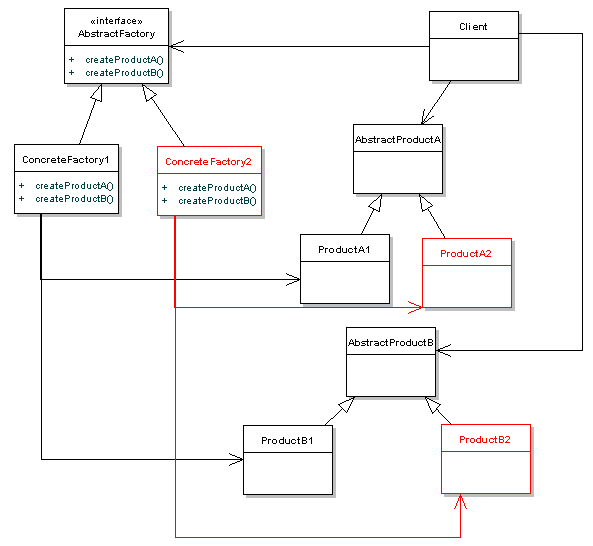
### **4.Lượ đồ Strategy Pattern**

Theo lược đồ trên thì Context được tạo bởi Stategy. Phần Context có thể có bất cứ thứ gì khi nó yêu cầu thay đổi hành vi. Strategy đơn giản là interface, vì thế chúng ta có thể hoán đổi các thuật toán ConreteStrategy mà không ảnh hưởng tới Context.



### **5. Lược đồ hoạt động Strategy Pattern**

Dùng Context từ client có nhiều hướng dùng, client có thể gọi Context, sau đó Context tự quyết định việc xử lý tiếp theo thay cho client. Tốt hơn là để nguyên quyết định này của Context, vì nó loại bỏ các câu lệnh chuyển đổi kiểu mà chúng ta đã thấy trong Factory pattern như hình dưới.



# **CHƯƠNG 4. GIỚI THIỆU CÔNG NGHỆ**

## **I. Ngôn ngữ lập trình Java**

### **1. Khái niệm**

Java là một ngôn ngữ lập lập trình, được phát triển bởi Sun Microsystem vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

### **2.****Một số đặc điểm nổi bật của ngôn ngữ Java**

**Thông dịch:**

Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi \*.java đầu tiên được biên dịch thành tập tin có đuôi \*.class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.

**Độc lập nền:**

Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, …) với điều kiện ở đó có cài đặt máy ảo java (Java Virtual Machine).

**Hướng đối tượng:**

Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.

**Đa nhiệm - đa luồng (MultiTasking - Multithreading):**

Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm và tương tác với nhau.

**Khả chuyển (portable):**

Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java. “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).

**Hỗ trợ mạnh cho việc phát triển ứng dụng:**

Công nghệ Java phát triển mạnh mẽ nhờ vào “đại gia Sun Microsystem” cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như:

J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server.

J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các ứng dụng thương mại.

J2ME (Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, …

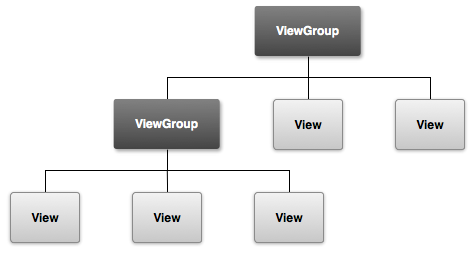
## **II.Giới thiệu về Android Studio**

### **1.Android Studio là gì**?

**Android Studio** là một phầm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

**2. Giao diện người dùng**

Giao diện người dùng của từng thành phần trong ứng dụng của bạn được định nghĩa bằng cách sử dụng một phân cấp của các đối tượng [View](https://developer.android.com/reference/android/view/View.html?hl=vi) và [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html?hl=vi) như minh họa trong hình 1. Mỗi nhóm dạng xem là một bộ chứa vô hình có chức năng tổ chức các dạng xem con, trong khi dạng xem con có thể là điều khiển nhập liệu hoặc các widget khác để vẽ một phần nào đó của UI. Cây phân cấp này có thể đơn giản hoặc phức tạp như nhu cầu của bạn (nhưng đơn giản sẽ tốt cho hiệu năng).



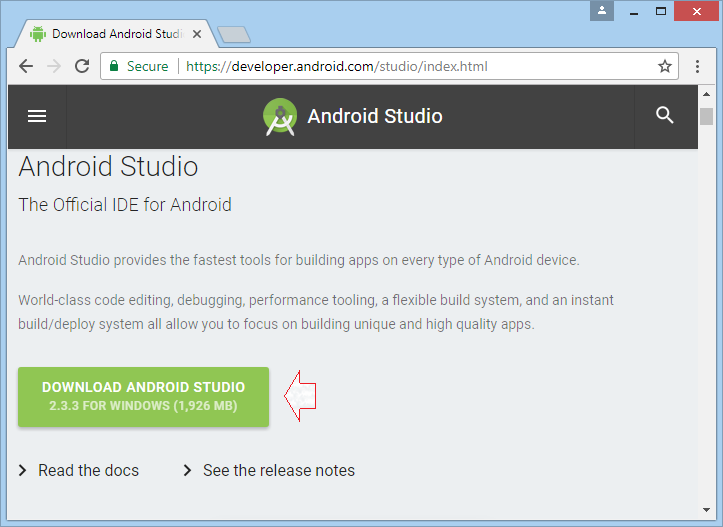
*Hình1: Minh họa một phân cấp dạng xem có chức năng định nghĩa một bố trí UI.*

**III/ Chuẩn bị môi trường làm việc với android studio**

**1. Tải và cài đặt Java JDK**

**2. Tải và cài đặt android studio**

\* bươc 1: tải android studio



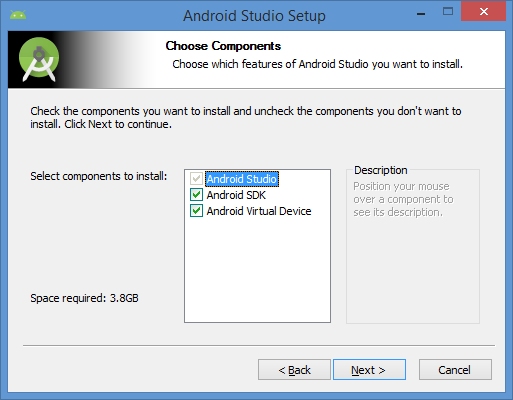
Hình 2: tải android studio về máy

\* bước 2: khơi động tệp android studio vừa tải



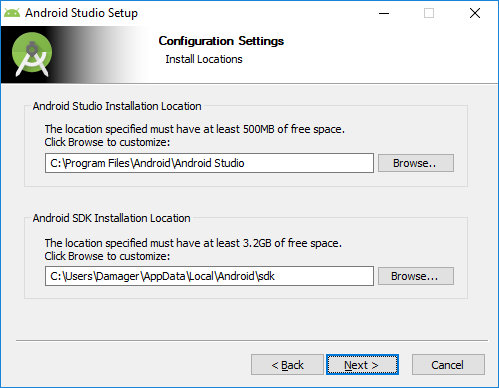
*Hình 3: Khi khởi động tệp android studio*

\* bước 3: chọn theo mặt định của android studio ( như hình 4)



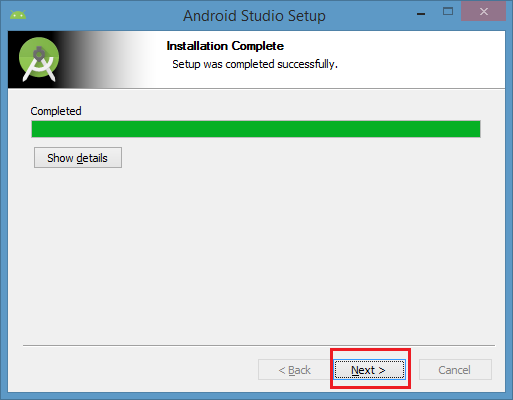
*Hình 4:chọn môi trường và máy ảo của android studio*

\* bước 4: chọn nơi lưu trương trình android studio



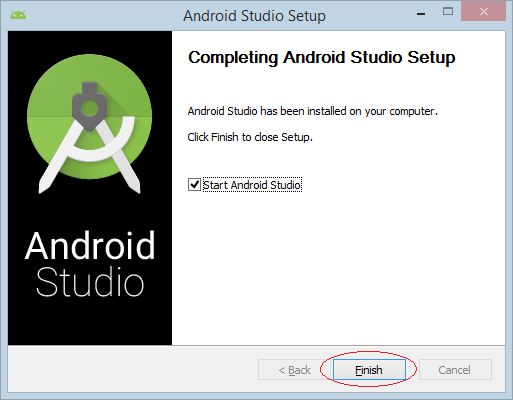
*Hình 5:chon thư mục cần lưu trương trình*

\* bước 5: đợi android studio cài đặt

**

*Hình 6:đợi android studio tự động cài đặt*

\* bước 6: nhấn finish để



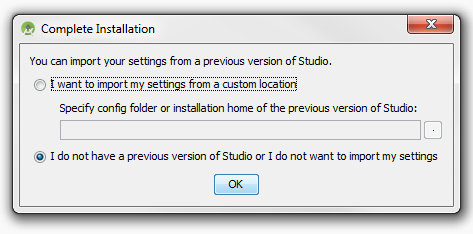
*Hình 7: Nhấn Finish để bắt đầu android studio*

\* bước 7: khi lên giao điện như hình 8 ta có hai lựa chọn

- Lựa chon 1: là import tất cả phần cài đặt trước hay không

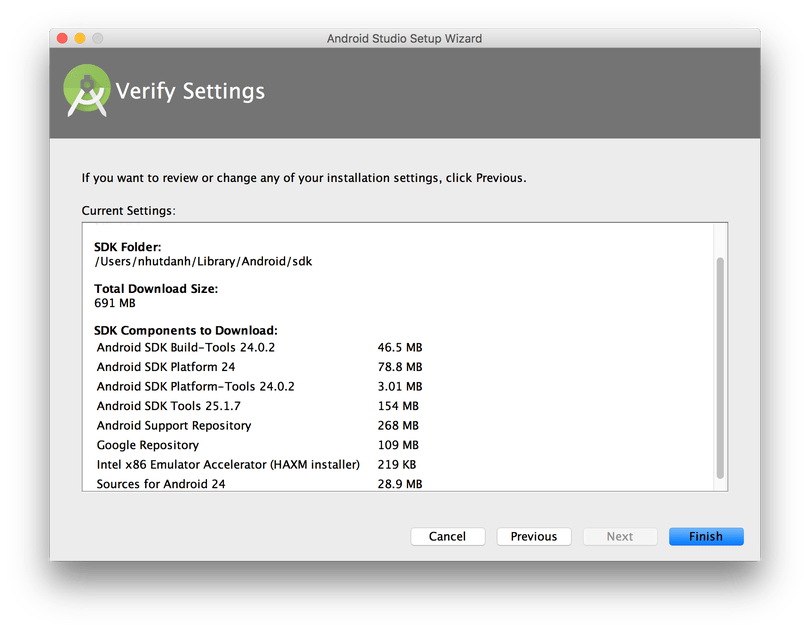
- Lựa chon 2: là không có phần cài đặt trước đó

-sau khi lựa chọn xong nhân ok và nhấn next tới hình của bước 8



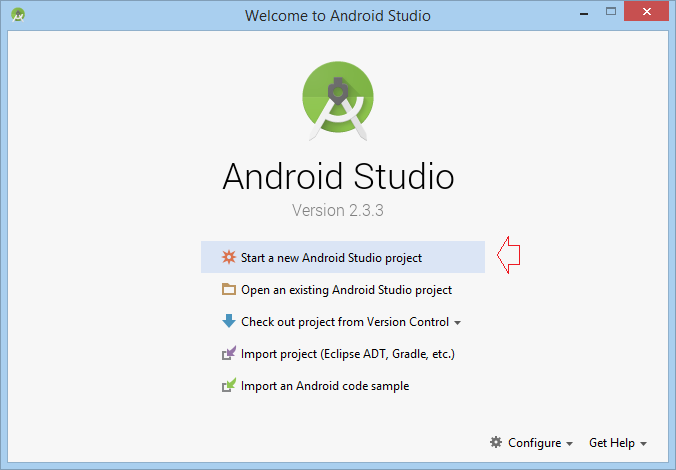
*Hình 8: sau khi nhấn finish*

\* bước 8: lần đàu tiên chạy sẽ yêu cầu cài thêm một số thứ như hình



*Hình 9: yêu cầu cài thêm một số trong android studio*

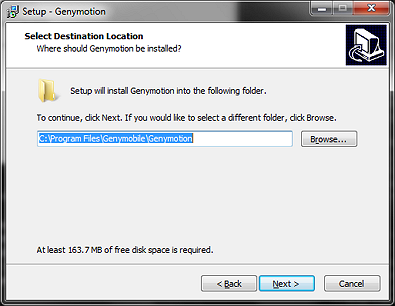
\* bước 9: Giao diện đầu tiên của android studio



*Hình 10: Giao diện đầu tiên của android studio*

**3. Tải và cài đặt Genymotion**

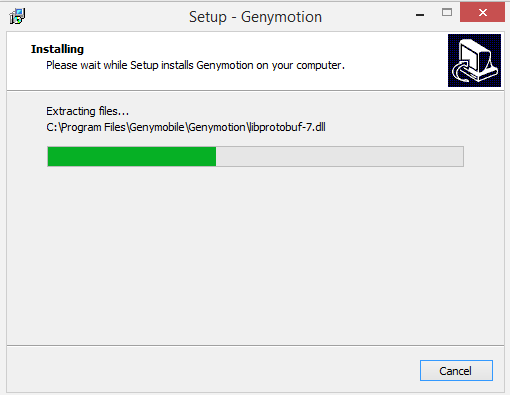
\* bước 1: chọn thư mục chứa Genymotion khi cài đặt



*Hình 11: chon thư mục để cài Genymotion*

\* lick next cho tới bước 2

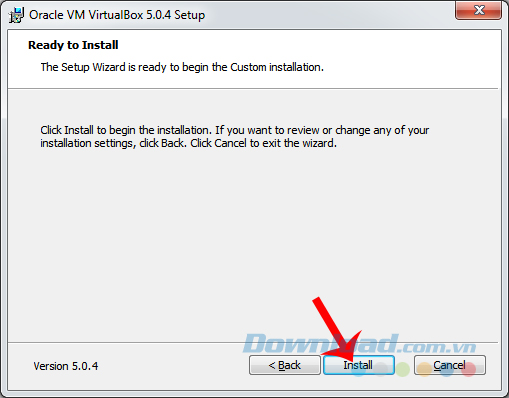
\* bước 2: đợi Genymotion cài đặt



*Hình 12: Genymotion đang cài đặt*

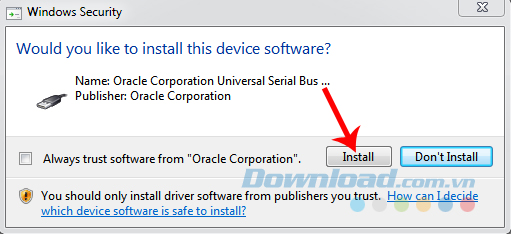
\* lick next tới bước 3

\* bước 3: Install



*Hình 13: Install*

Sau khi nhấn Install nếu được hỏi như dưới đây. Với trường hợp cá nhân người viết, phải tick chọn mục Alwáy trust software... sau đó chọn lại Install mới có thể cài đặt



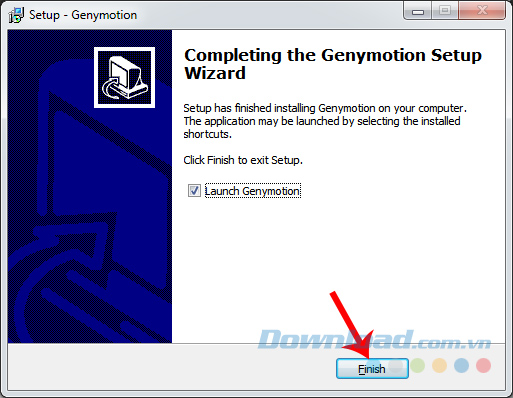
*Hình 14*

\* Bước 4: Bỏ dấu tick ở Start Oracle WM VirtualBox và chọn vào Finish



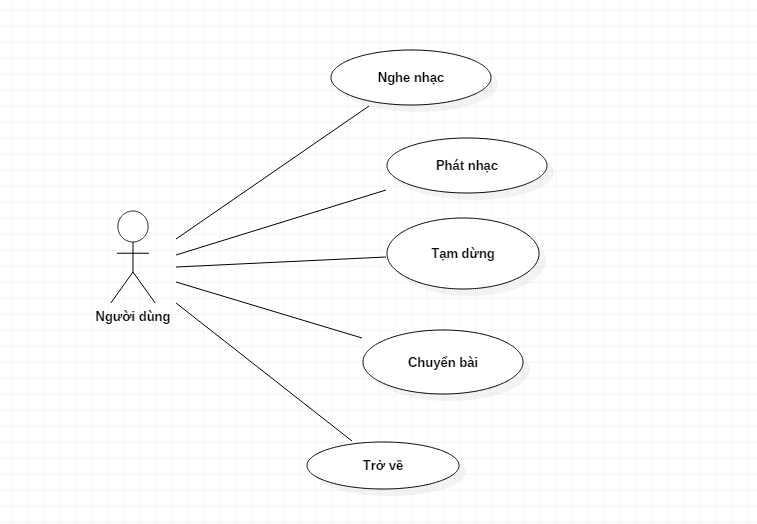
*Hình 15:*

\* Bước 5 Lúc này cửa sổ cài đặt cuối cùng của Genymotion xuất hiện, Finish để kết thúc quá trình.



# **CHƯƠNG 5: PHÂN TÍCH HỆ THỐNG**

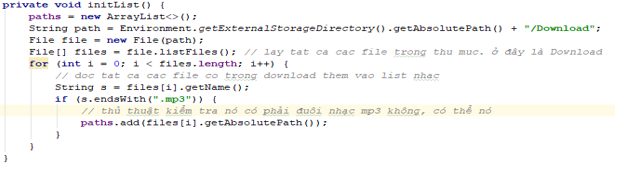
## **1. Sơ đồ use case**



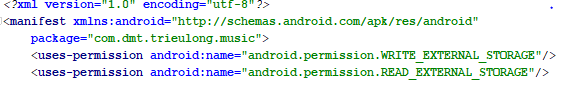
- Người dùng thực hiện các chức năng trên app nghe nhạc như: Nghe nhạc, Phát nhạc, Tạm dùng, Chuyển bài, Trở về

## **2. Các giải thuật**

- Truy cập vào file download của thiết bị như điện: thoai thoại...



- Khai báo cho ứng dụng quyền truy cập vào file của máy



**CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Xây dựng được một ứng dụng nghe nhạc cơ bản cho người dùng android một cách thuận tiện hơn

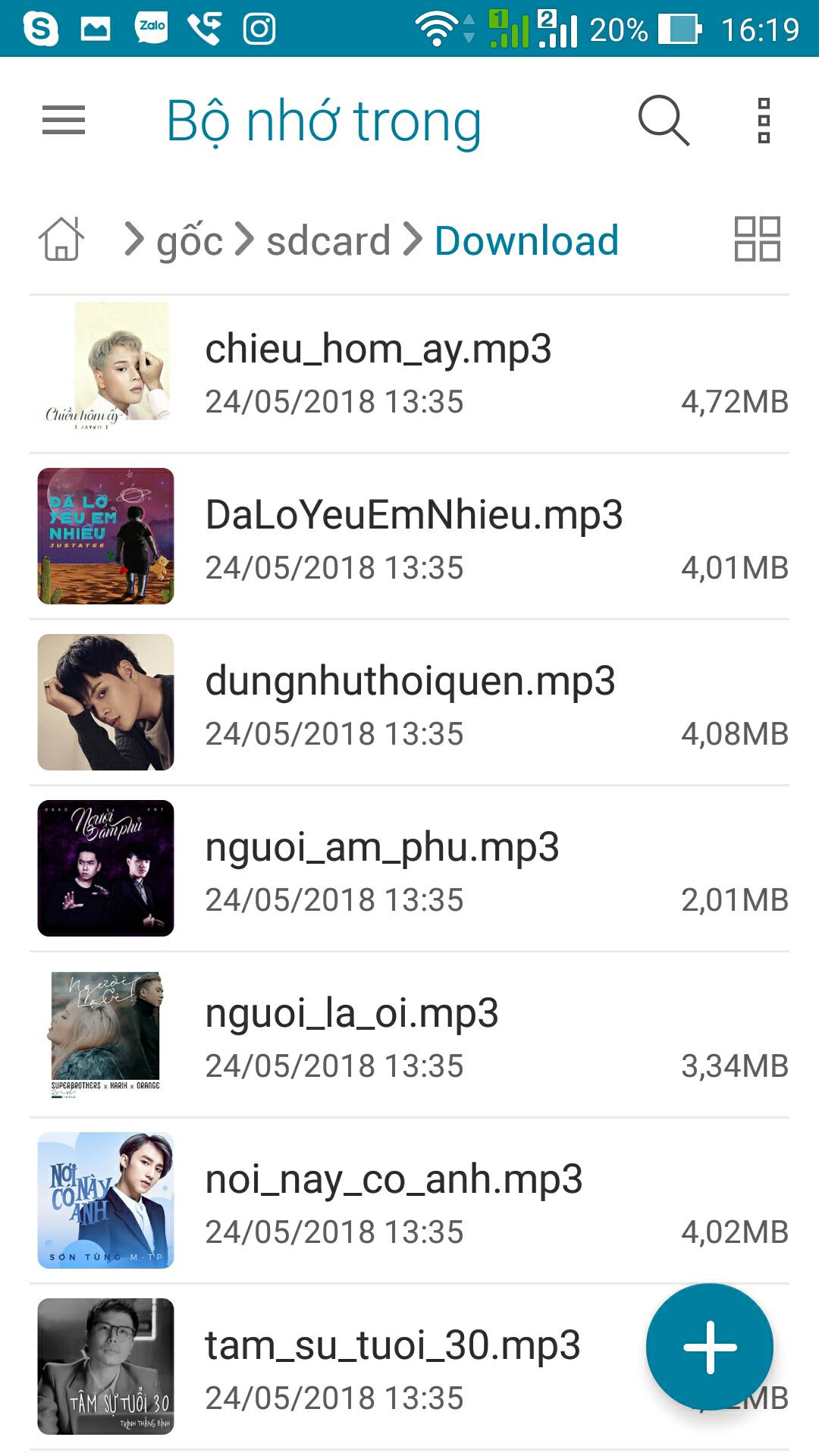
**1.Kết quả đạt được**

\* Về Ứng dụng

- Giao diện ứng dụng



- Ứng dụng sẽ truy cập vào file download của thiết bị



\* Về lý thuyết:

- Tim hiểu được về nguôn ngữ lập trình Java, nắm được cầu trúc cơ bản của Java

- Tim hiểu được về Design Pattern có những loại nào và cách áp dụng chúng để thu gon và dễ dàng trong lập trình.

- Tim hiều được về Strategy Pattern là mẫu chiến lước và cách áp dụng Strategy Pattern

- Biết cách lập trình trên nền tảng Android Studio và cách hoạt động của Android Studio

- Sử dụng được máy ảo Genymotion để phục vụ cho run ứng dụng.

Sau khi làm xong đồ án này, vận dụng được các kiến thức căn bản về Java và biết thêm một nền tảng mới phục vụ cho việt ứng dụng.

## **2.Việc chưa làm được**

Mặt dù đã cố gắng tham khảo nhiều nguồn tại liệu khác nhau nhưng em chưa đưa được Strategy vào bài cụ thể.

## **3. Hướng phát triển**

Phát triển ứng dụng có thể tải trực tiếp từ nguồn internet về ứng dụng.

# **Tài Liệu Tham Khảo**

<https://viblo.asia/p/tim-hieu-strategy-pattern-znmMdy7YGr69>

<https://allaravel.com/tutorials/strategy-pattern-xay-dung-chien-luoc-trong-luc-chay/>

Thiết-kế-mẫu-Design-Pattern%20(1).pdf