**CÔNG TY CÔNG NGHỆ & KỸ THUẬT STECH**

**VIỆN NGHIÊN CỨU & SÁNG TẠO KHOA HỌC MÁY TÍNH – IRICS**

**A logo with orange letters

Description automatically generated**

**BÁO CÁO NGHIÊN CỨU FLUTTER**

**TASK 4 – NGHIÊN CỨU FLUTTER**

Người hướng dẫn**: MR. NGUYỄN TẤN HÙNG ANH**

**MR. VÕ THÀNH LUÂN**

Người thực hiện**: ĐINH NGỌC AN THƠ**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**TÓM TẮT**

Ứng dụng Flutter đã xây dựng là một ví dụ tiêu biểu cho sức mạnh và linh hoạt của nền tảng phát triển ứng dụng di động Flutter. Mục tiêu của ứng dụng là tạo ra một giao diện người dùng đơn giản nhưng hữu ích, cho phép người dùng chọn và xem hình ảnh của các loài động vật.

Quy trình phát triển ứng dụng bao gồm các bước cơ bản từ việc tạo dự án mới đến thiết kế giao diện người dùng và triển khai logic ứng dụng. Bằng cách sử dụng các công cụ và tính năng của Flutter như widget, state management và event handling, chúng tôi đã xây dựng một ứng dụng hoàn chỉnh và chức năng.

Ứng dụng không chỉ cung cấp một trải nghiệm người dùng tốt mà còn mang lại lợi ích thực tế bằng cách giúp người dùng dễ dàng tương tác và tìm hiểu về thế giới tự nhiên xung quanh chúng. Đồng thời, việc xây dựng ứng dụng này đã mở ra một cánh cửa mới cho việc thử nghiệm và phát triển ứng dụng Flutter trong tương lai.

**MỤC LỤC**

[CHƯƠNG I : CODE FILE XYXOPHONE ( MẪU) 5](#_Toc167982223)

[I. Package & Main 5](#_Toc167982224)

[II. Class 5](#_Toc167982225)

[III. Interface 6](#_Toc167982226)

[CHƯƠNG 2: CODE FILE PROJECT TASK 4 8](#_Toc167982227)

[I. Package & Main 8](#_Toc167982228)

[II. Class 8](#_Toc167982229)

[III. Interface 11](#_Toc167982230)

[CHƯƠNG 3 : TỔNG HỢP 12](#_Toc167982231)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

[Hình 1: Code Package và Main của app 6](#_Toc167982370)

[Hình 2: Code Class và Expanded của app 7](#_Toc167982371)

[Hình 3: Code Interface của app 8](#_Toc167982372)

[Hình 4 : package và Main của project task 4 9](#_Toc167982373)

[Hình 5 : Các Class của project task 4 10](#_Toc167982374)

[Hình 6 : Hàm starReplayTimer 11](#_Toc167982375)

[Hình 7: Expanded buildKey 11](#_Toc167982376)

[Hình 8: Giao diện người dùng 12](#_Toc167982377)

1. CHƯƠNG I : CODE FILE XYXOPHONE ( MẪU)
   1. Package & Main

**A screen shot of a computer

Description automatically generated**

Hình 1: Code Package và Main của app

* ‘import ‘package :

/audioplayers.dart : Nhập thư viện ‘audioplayers’, cho phép ứng dụng có thể phát âm thanh

/material.dart : Nhập thư viện ‘material’, chứa các widget cơ bản của Flutter để xây dựng giao diện người dùng theo phong cách Materia; Design

* ‘void main() => runApp(XylophoneApp());’ : Hàm main() là điểm bắt đầu của ứng dụng. Và ‘runApp(XylophoneApp());’ để khởi chạy ứng dụng và truyền vào widget gốc là ‘XylophoneApp’
  1. Class

**A screen shot of a computer program

Description automatically generated**

Hình 2: Code Class và Expanded của app

* ‘class XylophoneApp extends StateWidget’ : Định nghĩa class ‘XylophoneApp’ được kế thừa từ ‘StatelessWidget’ là một lớp chính của ứng dụng
* ‘void playSound(int soundNumber)’ : định nghĩa phương thức ‘playSound’ nhận số nguyên ‘soundNumber’
* ‘final player = AudioPlayer();’ : Tạo đối tượng tên ‘AudioPlayer’
* ‘player.play(AssetSource(‘note$soundNumber.wav’))’: Sử dụng player để phát âm thanh từ tài nguyên note$soundNumber.wav , trong đó $soundnumber được thay thế bởi soundNumber
* ‘Expanded buildKey ({required Color color, required int soundNumber})’ : Định nghĩa phương thức là buildKey để nhận vào hai tham số là color và soundNumber để trả về widget Expanded
* ‘return Expanded()’ : Trả về widget Expanded, mở rộng kích thước chiều dọc hoặc ngang
* ‘child: TextButton(

Child : SizeBox.shrink(),

Sytle : ButtonStyle(

backgroundColor: MaterialStateProperty.all<color>(color),’

không hiển thị nội dung TextButton vì ‘SizedBox.shrink()’ tạo ra một widget có kích thước tối thiểu.

Và style : ButtonStyle dùng để thiết lập kiểu cho TextButton với màu nền là màu được đặt bởi ‘color’

* ‘onPressed(){

playSound(soundNumber);}’

Khi nút được nhấn, gọi phương thức ‘playSound(soundNumber)’

* 1. Interface

**A screen shot of a computer program

Description automatically generated**

Hình 3: Code Interface của app

-‘@override’ : Ghi đè phương thức ‘build’ của class ‘StatelessWidget’

- ‘Widget build (BuildContext context)’ : Phương thức build để xây dựng giao diện người dùng

- ‘return MaterialApp()’ : Trả về MaterialApp là gốc của ứng dụng

- ‘home : Scaffold()’ : Scaffold cung cấp cấu trúc cơ bản của giao diện, gồm AppBar, Body, BottomNavigationBar,…

-‘backgroundColor : Colors.black’: Đặt màu nền của Scaffold là màu đen

-‘body : SafeArea()’ : SafeArea để đảm bảo nội dung không bị che bởi các phần của thiết bị như notch, status bar,…

-‘child: Column()’ : Column sắp xếp các con theo chiều dọc

-‘crossAxisAlignment :CrossAxisAlignment.stretch;’ : căn các con của Column trên dài theo chiều ngang

-‘children : <Widget> [] ‘: danh sách các con của Column

Gọi ‘buildKey’ là phương thức để tạo các phim xylophone với màu sắc và âm thanh qua ‘buildKey(color: Colors.red soundNumber:1)’ , tương tự với các phím khác. Mỗi ‘buildKey’ tạo ra một nút có màu và âm thanh tương ứng, khi được nhấn sẽ phát ra nốt nhạc tương ứng với assets đã khai báo

1. CHƯƠNG 2: CODE FILE PROJECT TASK 4
   1. Package & Main

**A screen shot of a computer

Description automatically generated**

Hình 4 : package và Main của project task 4

* import ‘package :

/audioplayers.dart : Nhập thư viện ‘audioplayers’, cho phép ứng dụng có thể phát âm thanh

/material.dart : Nhập thư viện ‘material’, chứa các widget cơ bản của Flutter để xây dựng giao diện người dùng theo phong cách Materia; Design

/dart:async : cung cấp các lớp và phương thức liên quan đến lập lịch và xử lý bất đồng bộ trong Dart gồm Future, Stream và Timer

* ‘void main() => runApp(MyApp());’ : Hàm main() là điểm bắt đầu của ứng dụng. Và ‘runApp(MyAppeApp());’ để khởi chạy ứng dụng và truyền vào widget gốc là ‘MyApp’
  1. Class

**A screen shot of a computer program

Description automatically generated**

Hình 5 : Các Class của project task 4

1. **Lớp MyApp và StatefulWidget**

* **‘**MyApp’ :widget đại diện cho cả ứng dụng
* Class MyApp kế thừa từ StatefulWidget vì sẽ có trạng thái có thể thay đổi trong quá trình thực thi

1. **Lớp ‘\_MyAppState’ và ‘State’**

* **‘**\_MyAppState’ là lớp con của ‘MyApp’ đại diện cho trạng thái của app
* Class ‘\_MyAppState’ mở rộng từ ‘State<MyApp>’ thay đổi khi gọi hàm ‘setState’ để cập nhật giao diện người dùng

1. **Biến và Đối Tượng**

List<int> playedSounds = [];

Timer? replayTimer;

final player = AudioPlayer();

* ‘playedSounds’ : Danh sách mà các số nguyên đã đặt đại diện cho âm thanh được phát
* ‘replayTimer’ : Đối tượng Timer để quản lý việc phát lại âm thanh
* ‘player’ : một tính năng của class ‘AudioPlayer’ từ ‘audioplayers’ được sử dụng để phát các tệp âm thanh

1. **‘playSound’**

* Hàm ‘playSound’ được sử dụng để phát một âm thanh khi người dùng nhấn vào một nút âm thanh.
* Trong hàm này:

Sử dụng ‘player.play’ để phát tệp âm thanh tương ứng với ‘soundNumber’.

* Sau khi âm thanh được phát thành công, ‘soundNumber ‘được thêm vào danh sách ‘playedSounds’ và giao diện người dùng được cập nhật bằng cách gọi ‘setState’.
* ‘startReplayTimer’ được gọi để thiết lập lại hẹn giờ, để chuẩn bị cho việc phát lại các âm thanh sau khi đã phát âm thanh mới.

1. **Hàm ‘startReplayTimer’**

**A screen shot of a computer code

Description automatically generated**

Hình 6 : Hàm starReplayTimer

* ‘startReplayTimer’ : được sử dụng để khởi động hoặc thiết lập lại hẹn giờ để phát lại các âm thanh đã được phát trước đó.
* Nếu có hẹn giờ đang chạy (replayTimer không null), nó sẽ bị hủy bỏ.
* Sau đó, một hẹn giờ mới được thiết lập với thời gian là 3 giây và hàm ‘replaySounds’ được đặt làm hàm ‘callback’.

1. **‘buildKey’**

**A screen shot of a computer program

Description automatically generated**

Hình 7: Expanded buildKey

* ‘buildKey’ được sử dụng để tạo ra các nút âm thanh hiển thị trên giao diện người dùng, nhận vào một màu và một số âm thanh, và trả về widget ‘Expanded’ chứa một ‘TextButton’.
* Mỗi nút âm thanh sẽ hiển thị số âm thanh và một biểu tượng dấu check nếu âm thanh đã được phát**.**
  1. Interface

**A screen shot of a computer program

Description automatically generated**

Hình 8: Giao diện người dùng

* Trong ‘build’, trả về widget ‘MaterialApp’ là đại diện cho ứng dụng Flutter.
* MaterialApp chứa một Scaffold, định nghĩa cấu trúc cơ bản của giao diện ứng dụng.
* ‘debugShowCheckedModeBanner: false’ : không hiện dòng debug khi chạy app
* Scaffold có màu nền đen (backgroundColor: Colors.black) và chứa một phần thân (body) là một hàng ngang (Row) được đặt trong SafeArea.Trong child: Row , mỗi nút âm thanh được tạo bằng cách gọi hàm buildKey với màu và số âm thanh tương ứng.

1. CHƯƠNG 3 : TỔNG HỢP

* Project mini trên giúp hiểu và sử dụng được package đúng cách, cách viết mã nguồn ứng dụng, cách sử dụng audio trong lập trình.
* Quản lý trạng thái và Logic phát âm thanh : các code xử lý trạng thái của ứng dụng : logic phát âm thanh, bộ đếm thời gian ghi nhớ và phát lại
* Về phần xây dựng Interface : biết cách tạo nút cho mỗi âm thanh và thiết lập bố cục rõ ràng
* Các câu lệnh tạo ra một ứng dụng Flutter đơn giản cho phép phát ra âm thanh khi bấm các nút và các màu khác nhau. Và sau mỗi lần phát âm thanh, sẽ có một Icon kiểm tra hiện lên cho thấy rằng âm thanh đã được phát và sau một khoảng thời gian đã thiết lập các âm thanh đó sẽ tự động phát lại