**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 3 – 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 1: GIẢI THÍCH CODE TRONG MAIN 6](#_Toc167808713)

[I. FILE “XUCXAC” 6](#_Toc167808714)

[1. Main() : 6](#_Toc167808715)

[2. Phương thức : 7](#_Toc167808716)

[II FILE “ LIST\_SAMPLE” 11](#_Toc167808717)

[1. Main 11](#_Toc167808718)

[1. Class 12](#_Toc167808719)

[2. Giao diện 13](#_Toc167808720)

[CHƯƠNG 2: TỔNG HỢP MINI-PROJECT 15](#_Toc167808721)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

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

[Hình 1: Main của file XucXac 6](#_Toc167808970)

[Hình 2: Phương thức thay đổi của Xúc xắc 7](#_Toc167808971)

[Hình 4: Main file List\_Sample 10](#_Toc167808972)

[Hình 5: Class file List\_Sample 11](#_Toc167808973)

[Hình 6 : Interface file List\_Sample 12](#_Toc167808974)

CHƯƠNG 1: GIẢI THÍCH CODE TRONG MAIN

* 1. FILE “XUCXAC”

1. Main() :

**A screen shot of a computer program

Description automatically generated**

Hình 1: Main của file XucXac

a. Cấu trúc MaterialApp

**-**Đoạn mã sử dụng ngôn ngữ Dart

**-** ‘import ‘dart math’ : Thư viện cung cấp hàm toán học, các hàm sinh số ngẫu nhiên, hàm tính toán lượng giác

**-** ‘import ‘ package : flutter/material.dart’;’ : Thư viện cung cấp widget, công cụ để thiết kế giao diện trong flutter

**-**‘void main()’ : Hàm được gọi khi ứng dụng khởi chạy

**-** ‘runApp()’ : hàm chạy ứng dụng truyền widget gốc

**-**‘MaterialApp’ : widget gốc cung cấp nhiều tính năng của ứng dụng như Theme, Navigation,…

- ‘debugShowCheckedModeBanner: false,’ : dùng để ẩn biểu ngữ “ DEBUG” góc bên phải màn hình khi ứng dụng chạy ở chế độ debug

-‘ home’ : Định nghĩa widget hiển thị ứng dụng là ‘Scaffold’ - khung để thêm các yếu tố thiết kế widget phổ biến như AppBars, Drawers,…

b. Cấu trúc Scaffold

- ‘backgroungColor : Colors.blueAccent’ : Thiết lập màu nền

- **‘**appBar’ : Định nghĩa thanh ứng dụng ( nằm ở đầu) của Scaffold :

-‘title:Text(‘Xúc Xắc’) : Tên tiêu đề của AppBar là “Xúc Xắc”

- ‘backgroundColor: Colors.blueAccent’ : Đặt màu nền của AppBar là màu xanh

-‘body: DicePage(),’ : định nghĩa phần nội dung của Scafflold sẽ ở phần sau của ứng dụng. ‘DicePage’ là một widget tùy chỉnh chịu trách nhiệm hiển thị và xử lý logic cho trò chơi xúc xắc

1. **Phương thức** :

A computer screen shot of a code

Description automatically generated

Hình 2: Phương thức thay đổi của Xúc xắc

1. Định nghĩa widget ‘DicePage’ :

* ‘class DicePage extends StatefulWidget’ : tạo widget có thể thay đổi trạng thái là ‘DisPage’
* ‘@override’ : thể hiện phương thức ‘createState’ đang ghi đề một phương thức của lớp cha
* ‘\_DicePageStatte createState() =>\_DicePageState ();’ : Phương thức tạo ra instance chứa trạng thái của ‘DicePage’ cần dùng

1. Định nghĩa state ‘DicePage’ :

class \_DicePageState extends State<DicePage> {

int leftDiceNumber = 1;

int rightDiceNumber = 1;

* ‘class\_DicePageState extends State<DicePage>’ : Định nghĩa class ‘\_DícePageState’ quản lý ‘DicePage’

‘int leftDiceNumber = 1’ và ‘int rightDiceNumber = 1’ : khai báo biến trạng thái ‘left’ và ‘right’ với giá trị là 1. Đây là hai biến sẽ lưu trữ số mặt của hai xúc xắc

1. Phương thức đổi mặt xúc xắc :

void changeDiceFace() {

setState(() {

leftDiceNumber = Random().nextInt(6) + 1;

rightDiceNumber = Random().nextInt(6) + 1;

});

}

* ‘void changeDiceFace() ‘ : phương thức thay đổi giá trị của biến
* ‘setState()’: hàm thông báo cho biết có sự thay đổi trạng thái và cần cập nhật giao diện của người dùng. Bên trong ‘setState’, sẽ cần cập nhật biến
* ‘leftDiceNumber = Random().nextInt(6) + 1’ : Sinh số ngẫu nhiên từ 1->6 cho xúc xắc bên trái
* ‘rightDiceNumber = Random().nextInt(6) + 1’ : : Sinh số ngẫu nhiên từ 1->6 cho xúc xắc bên phải

1. **Giao diện**

A screen shot of a computer program

Description automatically generated

Hình 3; Giao diện file XucXac

* ‘override’ : ghi đè ‘build’ để thiết kế giao diện của widget ‘DicePage’
* ‘center’ : căn giữa Widget con trong màn hình
* ‘row’ : sắp xếp các widget con theo hàng ngang

Expanded(

child: TextButton(

child: Image.asset(

'images/dice$leftDiceNumber.png',

),

onPressed: () {

changeDiceFace();

},

),

),

-‘Expanded’ : Widget mở rộng lấp để làm đầy phần không gian còn lại trong ‘Row’

-‘TextButton’ : Widget tạo nút bấm không có đường viền

- ‘child: Image.asset('images/dice$leftDiceNumber.png')’ : Hiển thị hình ảnh của xúc xắc được lấy từ thư mục ‘images’ đã được khai báo assets trong pubspec.yaml

- ‘onPressed: ()’ : khi nút bấm được nhấn thì ‘changeDiceFace’

Đây là đoạn mã dùng để thay đổi mặt ngẫu nhiên khi người dùng nhân vào xúc xắc bất kỳ, phương thức ‘changeDiceFace’ sẽ được gọi để cập nhật cả hai xúc xắc với các giá trị ngẫu nhiên khác nhau

1. FILE “ LIST\_SAMPLE”
2. Main

**A screen shot of a computer program

Description automatically generated**

Hình 4: Main file List\_Sample

1. Các câu lệnh import :

* ‘import 'package:flutter/material.dart’ : lệnh cung cấp nhiều widget và chức năng để thiết kế giao diện
* ‘import 'dart:ui’’ : cung cấp quyền truy cập vào render để sử dụng đồ họa nâng cao hoặc vẽ tùy chỉnh

1. Main :

* ‘Main()’: điểm bắt đầu của app. Khi runApp nhận widget và làm cho widget trở thành gốc của cây widget
* ‘runApp(MyApp())’ : yêu cầu Flutter khỏi động app và hiển thị widget được định nghĩa trong class ‘MyApp’

1. Lớp ‘MyApp’:

* ‘class MyApp extends StatelessWidget’ : Định nghĩa lớp new widget là’MyApp’ mở rộng từ ‘StatelessWidget’-loại widget không yêu cầu trạng thái có thể thay đổi ( ko thay đổi theo thời gian)
* ‘@override’ : ghi đè lên phương thức của lớp cha tuy ko bắt buộc nhưng giúp code dễ hiểu hơn
* ‘Widget build(BuildContext context)’ : mô tả cách hiển thị widget, được gọi khi widget được xây dụng
* ‘return MaterialApp’ : Hàm trả về widget là ‘MaterialApp’-một widget gốc của app, nó quản lý cấu hình tổng thể của ứng dụng ( tiêu đề, chủ đề)

1. Class

**A screen shot of a computer

Description automatically generated**

Hình 5: Class file List\_Sample

* ‘class AnimalSelectorScreen extends StatefulWidget’ : Class main của Screen chọn động vật, mở rộng từ ‘StatefulWidget’ giúp thay đổi trạng thái
* ‘createState() => \_AnimalSelectorScreenState()’ : tạo và trả về class được gọi “\_AnimalSelectorScreenState” là AnimalSelectorScreen
* ‘String? selectedAnimal’ : Lưu trữ tên động vật được chọn
* ‘final Map<String, String> animalImages’ : liên kết tên động vật với hình ảnh tương ứng

1. Giao diện

**A screen shot of a computer program

Description automatically generated**

Hình 6 : Interface file List\_Sample

* ‘Widget build(BuildContext context)’ : phương thức mô tả giao diện người dùng của widget. ‘return Scaffold’ và trả về widget ‘Scaffold’ cung cấp cấu trúc cơ bản của màn hình với thanh công cụ là AppBar và phần nội dung chính
* ’appBar: AppBar(title: Text('Animal Selector'))’ : Thiết lập thanh công cụ ở đầu màn hình với nội dung “Animal Selector”
* Về phần Body :
* ‘center’ : giúp căn giữa phần thân của Scaffold giữa các widget con
* ‘child : Column()’ : sắp xếp widget con theo chiều dọc
* ‘mainAxisAlignment: MainAxisAlignment.center’ : Căn giữa các widget con theo trục chính của cột ( theo chiều dọc )
* ‘DropdownButton<String>’ : Tạo nút thả xuống để người dùng chọn giá trị từ danh sách
* ‘hint :Text (‘Chọn một loài động vật’) : Hiển thị loại gợi ý khi chưa chọn giá trị
* ‘value: selectedAnimal’ : Giá trị hiện tại của nút thả xuống
* ‘items: animalImages.keys.map((String animal) {...}).toList()’ : Tạo danh sách các mục thả xuống của ‘animalImages’. Mỗi key là tên một loài động vật
* ‘onChanged: (String? newValue) {...}’ : Xử lý sự kiện khi người dùng chọn mục khác trong nút thả xuống
* ‘setState’ : để cập nhật ‘selectedAnimal’ với giá trị mới (‘newValue’)

Đoạn mã xây dựng trên màn hình một nút thả xuống để người dùng chọn một loại động vật. Khi được chọn hình ảnh của nó sẽ được hiển thị dưới nút thả xuống

CHƯƠNG 2: TỔNG HỢP MINI-PROJECT

Qua mini-project có thể xây dựng được một ứng dụng flutter đơn giản với chức năng chọn Cho phép người dùng chọn động vật từ một danh sách và hiển thị hình ảnh của động vật đã chọn

và hiển thị hình ảnh động vật. Và qua đó tổng hợp được những nội dung đã học được :

1. Hiểu rõ về flutter : Phát triển giao diện người dùng và logic ứng dụng
2. Hiểu rõ về Dart : ngôn ngữ lập trình để phát triển Flutter
3. Thiết kế được giao diện người dùng : bằng cách sử dụng widget của Flutter để tạo một giao diện người dùng gồm một nút thả xuống và một vùng hiển thị hình ảnh
4. Xử lí sự kiện : Xây dựng logic để xử lú khi người dùng chọn một mục từ danh sách
5. Sử dụng flutter để hiển thị hình ảnh