TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM

KHOA CÔNG NGHỆ THÔNG TIN

-----*-----**

BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN "LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG"

Đề tài:

XÂY DỰNG ỨNG DỤNG DI ĐỘNG ĐA NỀN TẢNG NHẬN DIỆN CHỮ DÙNG FLUTTER VÀ GOOGLE ML KIT

GVHD: ***ThS. Nguyễn Hạnh Phúc***

Sinh viên thực hiện: ***Nguyễn Tuấn Anh – Mã sv: 77141***

Đồng Mai Trinh – Mã sv: 80158

Lê Tú Linh – Mã sv: 78617

Hải Phòng, tháng 10 năm 2019

TRƯỜNG ĐẠI HỌC HÀNG HẢI

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN KHOA HỌC MÁY TÍNH

-----*-----**

BÀI TẬP LỚN

HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

1. Tên đề tài

Xây dựng ứng dụng di động nhận diện chữ dùng flutter và mk kit

2. Mục đích

Xây dựng ứng dụng nhận diện chữ có thể chạy trên nhiền nền tảng (Web, Android, IOS) ứng dụng API của Google Machine learning (ML kit) trên cơ sở là phương pháp lập trinh hướng đối tượng.

3. Công việc cần thực hiện

- Tạo ứng dụng sử dụng flutter, viết bằng ngôn ngữ Dart sử dụng phương pháp lập trinh hướng đối tượng :
  - Kế thừa các lớp có sẵn để tạo giao diện
  - Sử dụng pakage “camera” để truy cập vào camera, chụp ảnh, lưu vào đối tượng File.
- Tích hợp Firebase vào ứng dụng :
  - iOS (https://firebase.google.com/docs/ios/setup)
  - Android (https://firebase.google.com/docs/android/setup)
  - Web (https://firebase.google.com/docs/web/setup)
- Sử dụng ML Kit, dùng hình ảnh chụp được làm parameters khởi tạo đối tượng FirebaseVisionImage, FirebaseVisionImage chứa các thuộc tinh gồm raw image byte của ảnh và thông tin về hướng , kích thướng.

- Khởi tạo đối tượng TextReconizer , khởi tạo đối tượng visionImage bằng phương thức textRecognizer.processImage(visionImage) với visionImage là đối tượng FireBaseVisionImage đã tạo ở đầu. Đối tượng visionImage chứa các thông tin liên quan đến dữ liệu chữ tìm được từ ảnh
- Dùng các thuộc tính, phương thức getter của visionImage để cung cấp dữ liệu cho view model.
- Làm báo cáo bài tập lớn
- Bảo vệ bài tập lớn

4. Yêu cầu

- Kết quả làm bài tập lớn: Báo cáo bài tập lớn
- Báo cáo bài tập lớn phải được trình bày theo mẫu quy định (kèm theo), báo cáo có thể kết xuất thành tệp định dạng PDF và nộp qua email (không bắt buộc phải in ấn)
- Hạn nộp báo cáo bài tập lớn: Tuần 13

5. Tài liệu tham khảo

- Flutter Documentation
- Firebase Documentation
- Dart (DartLang) Introduction: Getting started with Dart/Flutter - Uday Hiwarale
- Firebase ML Kit 101 - Hitanshu Dwahan

Hải Phòng, tháng 10 năm 2019

NGƯỜI HƯỚNG DẪN

MỤC LỤC

MỤC LỤC ..... **i**

DANH MỤC CÁC HÌNH VẼ, BẢNG BIỂU ..... **ii**

DANH MỤC CÁC TỪ VIẾT TẮT ..... **iii**

GIỚI THIỆU **1**

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT **2**

1.1. Giới thiệu ngôn ngữ Dart 2

1.1.1 Dart cơ bản 2

1.2. Giới thiệu Flutter 5

1.2.1. Flutter là gì 5

1.1.2 Điều gì khiến Flutter khác biệt 5

1.3. Giới thiệu Ml Kit 6

CHƯƠNG 2. THIẾT KẾ CHƯƠNG TRÌNH **7**

2.1. Lớp: MyApp – định nghĩa thiết kế của ứng dụng. 7

2.2. Lớp: HomePage – tạo state cho lớp _HomePageState 7

2.3. Lớp: _HomePageState - trang bắt đầu. 8

CHƯƠNG 3. CÀI ĐẶT CÁC LỚP **8**

3.1. MyApp 8

3.2. HomePage 9

3.3. _HomePageState 9

3.4. Kết quả 12

KẾT LUẬN **16**

TÀI LIỆU THAM KHẢO **17**

DANH MỤC CÁC HÌNH VẼ, BẢNG BIỂU

Hình vẽ	**Trang**
Ảnh kết quả 1: Màn hình chinh	13
Ảnh kết quả 2: Nhận diện chữ	14
Ảnh kết quả 3: Dịch	15

DANH MỤC CÁC TỪ VIẾT TẮT

Từ	**Ý nghĩa**
ML	Machine Learning
AI	Artificial Intelligence
OOP	Object-oriented programming
API	Application Programming Interface

GIỚI THIỆU

Bài toán:

Để xây dựng ứng dụng ứng dụng di động nhận diện chữ cần thực hiện hai phần:

- Phần giao diện.
- Phần logic code nhận ảnh, xử lý và cung cấp thông tin phần giao diện.

Yêu cầu:

- Xây dựng phần giao diện gồm các Widget là các đối tượng được kế thừa từ lớp MaterialApp. Lớp này chứa các thành phần giao diện như Button , FloatButton…. có thể được kế thừa và override lại để custom theo ý.
- Phần logic code ứng dụng ML kit, sử dụng các class được cung cấp sắn trong API của ML kit để thu được thông tin về chữ trong ảnh.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Giới thiệu ngôn ngữ Dart

Dart là ngôn ngữ lập trình hướng đối tượng, kiểu tĩnh, có thể được biên dịch thành mã máy, chuyển đổi thành mã JavaScript hoặc chạy như một ngôn ngữ thông dịch (giống Ruby, JavaScript…) thông qua Dart CLI.

Dart là ngôn ngữ lập trình đơn luồng với Event loop giống như JavaScript. Nhưng nó hỗ trợ tạo thêm những luồn khác để chạy các tác vụ nền hay còn được gọi là lập trình bất đồng bộ (asynchronos programming).

Dart là ngôn ngữ thuần hướng đối tượng nên mọi thứ gắn cho biến đều là đối tượng, kể cả số, hàm, null cũng đều là đối tượng. Tất cả đối tượng đều được kế thừa từ lớp Object.

Dart không hỗ trợ public, protected, private giống như c++ hay java. Nếu như tên biến, hàm bắt đầu bằng dấu _ thì nó là private.

Chỉ có 1 hàm main() là nơi chương trình bắt đầu.

Phần mở rông : .dart.

Các vòng lặp hoạt động như trong các ngôn ngữ java,c#...

1.1.1 Dart cơ bản

- **Chương trình Hello World**

```
void main() {
  for (var t = 0; t < 4; t++) {
    print('hello world $t');
  }
}
```

- **Biến và kiểu dữ liệu cơ bản**

String gName = 'Hoho';

Trong đó :

- String là kiểu dữ liệu

- gName là tên biến
- Hoho là giá trị gắn vào biến

```
void main() {
// Khai báo dữ liệu kiểu interger
int a = 6;
//Nếu biến được khai báo mà không có khởi tạo thì sẽ nhận giá trị null
int b; // b = null
//Kiểu dữ liệu có thể được tự động định thông qua quá trình gắn
var c = "hello"; // c là string
//Kiểu của biến không thể bị thay đổi trong runtime
//Chỉ có 2 kiểu dữ liệu cho số là int và Double
}
```

- **Lập trinh hướng đối tượng trong dart**
  - Dart hỗ trợ tất cả chức năng của mô hình lập trinh hướng đối tượng như Lớp, thừa kế, đa hình…
  - Dart dễ học với những người đã có hiểu biết về OOP.
  - Một đối tượng là một thể hiện của một class.
  - Dùng từ khóa class để tạo một class mới trong dart, class có thể có hoặc không có định nghĩa constructor
  - Dart không có accept modifier: dart không có các từ khóa public, protected, private, trong cùng một file dart với nhau thì mọi thứ là public. Nếu tên biến, hàm bắt đầu bằng _ thì là private.
  - Một class có thể có nhiều constructor. Các constructor khác constructor mặc định đều phải có tên theo cú pháp Tên_class.tên_constructor.

```
class Player {
```

```
  Player(String name, int color) {
    this._color = color;
    this._name = name;
  }
  Player.fromPlayer(Player another) {
    this._color = another.getColor();
    this._name = another.getName();
  }
}
new Player.fromPlayer(playerOne);
```

- Trong class có thể dùng các từ khoa getter và setter để tạo các phương thức get / set các thuộc tính của class từ bên ngoai class.

```
class Cat {
  bool _isHungry = true;
  bool get isCuddly => !_isHungry;
  bool get isHungry => _isHungry;
  bool set isHungry(bool hungry) => this._isHungry = hungry;
}
```

- Khi class B là con của class A , class B có thể sử dụng các phương thức của class A. Từ class B có thể dùng từ khoa super để gọi các phương thức của class A.

```
class Television {
  void turnOn() {
    _illuminateDisplay();
    _activateIrSensor();
  }
     // …
}
```

```
class SmartTelevision extends Television {
  void turnOn() {
    super.turnOn();
    _bootNetworkInterface();
    _initializeMemory();
    _upgradeApps();
  }
  // ···
}
```

- Dart hỗ trợ đa kế thừa thông qua từ khóa with.

1.2. Giới thiệu Flutter

1.2.1. Flutter là gì

Flutter là một framework mới, đa nền tảng, hướng đối tượng, mã nguồn mở dùng để phát triển ứng dụng được phát triển bởi Google. Với 1 base code là Dart, ứng dụng dụng viết bằng flutter có thể chạy trên Android, iOS, nền web thông qua dart2js. Điều này giúp nhà phát triển tiết kiệm được thời gian phát triển ứng dụng.

Có nhiều framework với mục đích giống như Flutter như React Native của Facebook, Xamarin của Microsoft,…

1.1.2 Điều gì khiến Flutter khác biệt

- Ngôn ngữ : Flutter sử dụng ngôn ngữ Dart. Cả Dart và Flutter đều cùng được tạo bởi cùng một team creator khiến cho chúng được tối ưu cho nhau về hiệu năng, đặc biệt là với một framework đa nền tảng. Cú pháp của Dart rất giống Java và dựa trên kỹ thuật lập trinh hướng đối tượng nên không mất nhiều thời gian để học nếu đã có kiến thức về OOP từ trước.
- Hiệu năng cao : Flutter biên dịch ra ngôn ngữ native của nền tảng trước thời gian chạy (AOT - pre-compiled ahead of time) nên ứng dụng khi

chạy trên thiết bị của người dùng cuối là ứng dụng native , chạy bằng code kotlin/java Android , swift/c-object iOS, javaScript Web cho hiệu năng cao hơn các framework khác. UI của flutter được vẽ bằng engine Skia dùng trên google chrome viết bằng C++ cho hiệu năng cao lên tới 60fps.

- Flutter tự thu hồi vung nhớ khi đối tượng kết thúc vòng đời của nó,.
- Flutter định nghĩa UI bằng code Dart, giống như với thiết kế web

```
class MyCenteredTextWidget extends StatelessWidget {
@override
Widget build(BuildContext context) {
return new Center(child: new Text("Center of Attention"));
}
}
```

Khi dùng đối tượng MyCenteredTextWidget sẽ cho kể quả là một màn hình với dông Text “Center of Attention” ở giữa.

- Hot reload và Hot restart : Với việc phát triển phần mềm truyền thống, khi có thay đổi nhỏ trong code thì cũng đều cần rebuild lại cả Project gây mất thời gian. Với việc dùng dart vm, flutter có thể cập nhật lại code của app mà không cần rebuild và chạy lại toàn bộ ứng dụng, giúp cho việc phát triển và debu nhanh và dễ dàng hơn nhiều.
- Ví dụ về hot reload .

1.3. Giới thiệu Ml Kit

Ml kit là Project của Goole nhằm cung cấp các API về Machine Learning dành cho những nhà phát triển phần mềm, giúp tích hợp các công cụ về ai và máy học vào ứng dụng mà không cần nhiều nguồn tài nguyên và kiến thức chi tiết về ML.

Các công cụ trong ML kit hiện tại chủ yếu tập trung vào tầm nhìn của máy tính (computer vision) bao gồm:

- Text recognition (nhận dạng văn bản).
- Image labeling (ghi nhãn hình ảnh).
- Barcode scanning (quét mã vạch).
- Face detection (nhận diện khuôn mặt).
- Landmark recognition (nhận diện mốc).
- Smart Reply (trả lời thông minh dựa trên ngữ cảnh).

ML kit có thể được sử dụng và tích hợp trong nhiều nền tảng, ở ví dụ này là Flutter.

CHƯƠNG 2. THIẾT KẾ CHƯƠNG TRÌNH

2.1. Lớp: MyApp – định nghĩa thiết kế của ứng dụng.

- Thừa kế :StatelessWidget (từ thư viện material).
- Phương thức: - build() dựng ui, override từ StatelessWidget.

2.2. Lớp: HomePage – tạo state cho lớp _HomePageState

State là trạng thái của đối tượng, khi có thay đổi về giá trị của đối tượng, trạng thái của đối tượng được làm mới, đối tượng được tạo lại và nạp lại trạng thai đã lưu trước đó, phương thức build() đc gọi lại để cập nhập giao diện.

- Thừa kế: StatefullWidget (từ thư viện material)
- Phương thức: createState() override từ StatefullWidget

2.3. Lớp: _HomePageState - trang bắt đầu.

- Thừa kế:State<HomePage>
- Thuộc tính :
  - File _image : lưu ảnh dưới dạng File.
  - visionImage _visionImage: chứa kết quả nhận diện chữ từ ML kit.
  - String _translated: chứa text sau khi dịch.
- Phương thức:
  - getImage() lấy ảnh từ camera hoặc thư viện ảnh, gán vào _image.
  - getText() xử lý ảnh, trả về visionImage chứa dữ liệu tìm được, gắn vào _visionImage
  - build() dựng ui.
  - translate() dịch sang tiếng việt.

CHƯƠNG 3. CÀI ĐẶT CÁC LỚP

3.1. MyApp

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
```

```
    return new MaterialApp(
      theme: new ThemeData(primaryColor: Colors.blue),
      home: new HomePage(),
    );
  }
}
```

3.2. HomePage

```
class HomePage extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    // TODO: implement createState
    return _HomePageState();
  }
}
```

3.3. _HomePageState

```
class _HomePageState extends State<HomePage> {
  File _image;
  VisionText _visionText;
  String _translated;

  void getImage(String src) async {
    var image = (src == "cam")
        ? await ImagePicker.pickImage(source: ImageSource.camera)
        : await ImagePicker.pickImage(source: ImageSource.gallery);
    if (image != null) {
      setState(() {
        _image = image;
```

```
      });
      getText();
    }
  }

  void getText() async {
    if (this._image == null) return;
    FirebaseVisionImage firebaseVision = FirebaseVisionImage.fromFile(_i
mage);
    TextRecognizer textRecognizer = FirebaseVision.instance.textRecognize
r();
    VisionText visionText = await textRecognizer.processImage(firebaseVis
ion);
    _visionText = visionText;
    setState(() {
      _visionText = visionText;
    });
  }

  void clean() {
    setState(() {
      this._image = null;
      this._translated = null;
    });
  }

  void translate() async {
    final translator = new GoogleTranslator();
    await translator
```

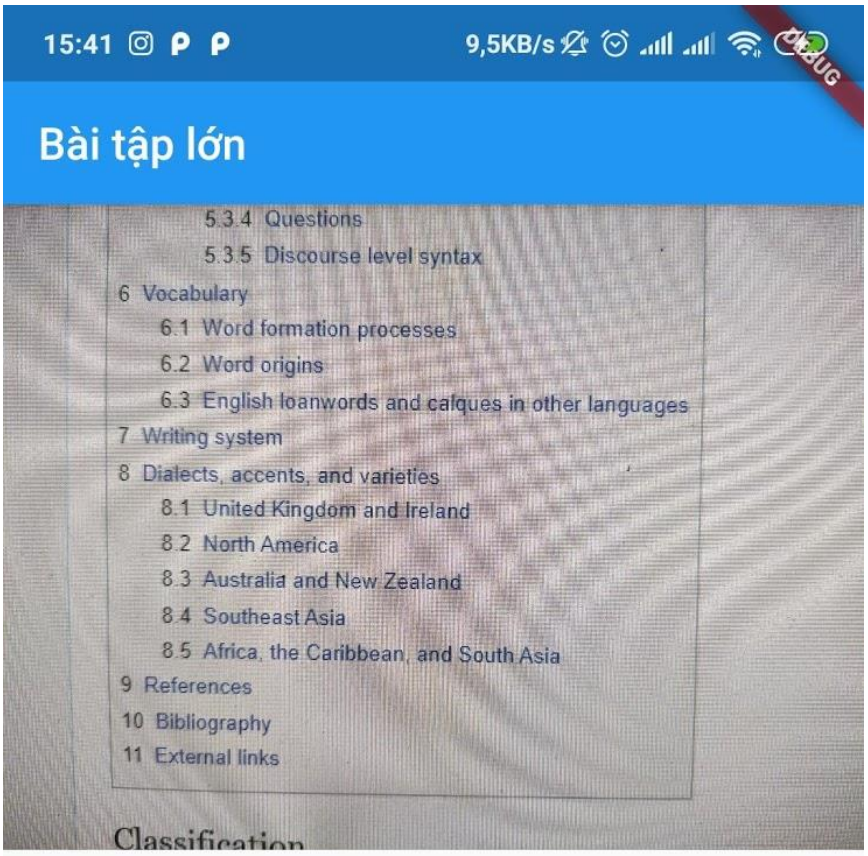
```
      .translate(_visionText.text, to: "vi")
      .then((text) => this._translated = text);
  setState(() {});
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Bài tập lớn"),
    ),
    body: (_image == null)
        ? Center(child: Text("Ấn vào " "+" " để chọn ảnh"))
        : ListView(
            children: <Widget>[
              Image.file(this._image),
              SizedBox(
                height: 20,
              ),
              Text(
                "   Chữ tìm được: ",
                style: TextStyle(fontSize: 15),
              ),
              Text(_visionText.text),
              SizedBox(
                height: 20,
              ),
              (_translated == null) ? SizedBox() : Text(_translated),
              RaisedButton(
```

```
                color: Colors.blue,
                colorBrightness: Brightness.dark,
                child: Text("Dịch"),
                onPressed: () => translate(),
              )
            ],
          ),
        floatingActionButton: SpeedDial(
          child: Icon(Icons.add),
          children: <SpeedDialChild>[
            SpeedDialChild(
              child: Icon(Icons.camera),
              onTap: () => getImage("cam"),
            ),
            SpeedDialChild(
              child: Icon(Icons.photo),
              onTap: () => getImage("photo"),
            ),
            SpeedDialChild(
              child: Icon(Icons.delete),
              onTap: () => clean(),
            ),
          ],
        ),
      );
    }
```

3.4. Kết quả

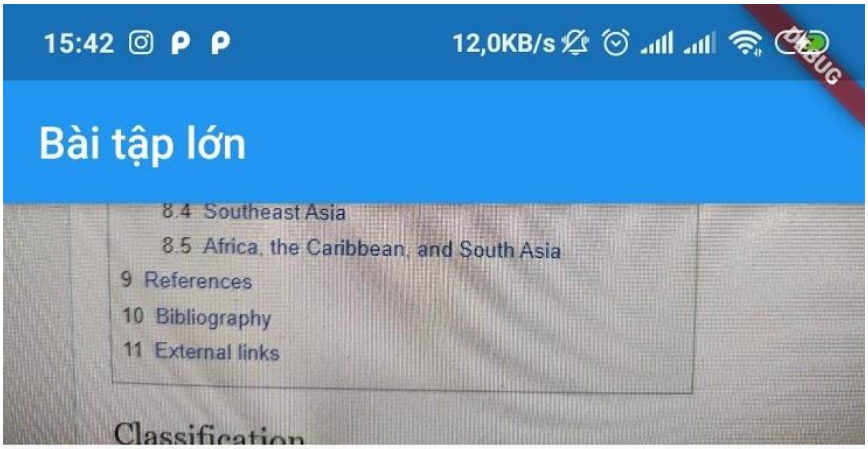
Ảnh kết quả 1: Màn hình chinh



Chữ tìm được:
5.3.4 uestionsS
5.3.5 Discourse level syntax
6 Vocabulary
61Word formation prOcesses
62 Word origins
6.3 English loanwords and calques in other languages
7Writing system
8 Dialects, accents, and varieties
81 United Kingdom and lreland
82 North America
83 Australia and New Zealand
84 Southeast Asia
8.5 Africa, the Caribbean and South Asia
9 References
10 Bibliography
11 External links
Classificatin

Dịch

Ảnh kết quả 2: Nhận diện chữ



Chữ tìm được:
5.3.4 uestionsS
5.3.5 Discourse level syntax
6 Vocabulary
61Word formation prOcesses
62 Word origins
6.3 English loanwords and calques in other languages
7Writing system
8 Dialects, accents, and varieties
81 United Kingdom and Ireland
82 North America
83 Australia and New Zealand
84 Southeast Asia
8.5 Africa, the Caribbean and South Asia
9 References
10 Bibliography
11 External links
Classificatin

5.3.4 uestionsS
5.3.5 cú pháp mức Discourse
6 Từ vựng
quá trình hình thành 61Word
62 nguồn gốc từ
6,3 từ vay mượn tiếng Anh và từ căn ke trong các ngôn ngữ khác
hệ thống 7Writing
8 tiếng địa phương, điểm nhấn, và giống
81 Vương Quốc Anh và Ireland
82 Bắc Mỹ
83 Úc và New Zealand
84 Đông Nam Á
8,5 Phi, Caribê và Nam Á
[illegible]
11 Liên kết ngoại
Classificatin

Ảnh kết quả 3: Dịch

KẾT LUẬN

- Đã giải quyết :
    - Tạo ứng dụng di động bằng flutter
    - Dùng Firebase ML kit vistion để tách chữ từ ảnh và dùng google dịch để dịch sang tiếng việt
- Chưa làm được :
    - Nhận diện chữ trong thời gian thực
    - Không nhận diện được tiếng việt có dấu
- Hướng phát triển :
    - Cải thiện khả năng nhận diện bằng cách dùng ML kit cloud.
    - Nhận diện trong thời gian thực
    - Ứng dụng thêm face dectect,… mở rộng chức năng của app
    - Xây dựng thêm nhiều lớp, nhiêu view mới cho chức năng mới.

TÀI LIỆU THAM KHẢO

- Flutter Documentation
- Firebase Documentation
- Dart (DartLang) Introduction: Getting started with Dart/Flutter – Uday Hiwarale
- Firebase ML Kit 101 - Hitanshu Dwahan