TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

KHOA CÔNG NGHỆ THÔNG TIN

--------------------------------

BÁO CÁO KẾT THÚC MÔN HỌC

Lập trình di động 3

**TÊN ĐỀ TÀI**

Giảng viên hướng dẫn: **Trương Bá Thái**

Sinh viên thực hiện:

1. NGUYỄN ANH TOÀN
2. LÊ MINH ĐẠT
3. NGUYỄN PHƯƠNG LINH
4. NGUYỄN ĐÔNG DUY

Ngành: Công nghệ thông tin Lớp: CD16TT1 Khoá: 12

*Tp. Hồ Chí Minh*, ngày tháng năm

**NHẬT KÝ HOẠT ĐỘNG NHÓM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Họ và tên** | **Công việc đã thực hiện** | **Tự đánh giá** | **Nhóm đánh giá** | **Chữ ký** |
| 1 | Nguyễn Văn A | 1. Công việc 1  2. Công việc 2… | 9 | 10 |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |

MỤC LỤC

DANH MỤC BẢNG BIỂU, HÌNH VẼ, SƠ ĐỒ 4

CHƯƠNG 1. TỔNG QUAN VỀ REACT NATIVE 5

1. Giới thiệu tổng quan về React native 5

2. Kiến thức cơ bản về ES6 trong React native 5

3. Component trong react native 5

4. API Trong React Native 5

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 6

2.1 Phân tích hệ thống 6

2.1.1. Feature/Component #1: MyMobile Registration screen 6

2.1.1.1 User Interfaces 6

2.1.1.2. Functional Requirements 6

2.2 Thiết kế hệ thống 8

2.2.1. Sudoku Main Screen 8

2.2.1.1. Screen Shot for Sudoku Main Screen 8

2.2.1.2. Objects and actions for Sudoku Main Screen 9

CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ 10

3.1 Cài đặt 10

3.2 Kiểm thử 10

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC 11

4.1 Kết quả đạt được 11

4.2 Các kết luận và kiến nghị 11

PHỤ LỤC 12

TÀI LIỆU THAM KHẢO 13

# DANH MỤC BẢNG BIỂU, HÌNH VẼ, SƠ ĐỒ

# CHƯƠNG 1. TỔNG QUAN VỀ REACT NATIVE

## 1. Giới thiệu tổng quan về React native

- Sự phát triển về CNTT ngày càng khủng khiếp. Những công nghệ mới đã và đang làm thay đổi thế giới một cách nhanh chóng. Từ lúc mới ra đời cho đến nay smartphone đã có những bước tiến mạnh mẽ vì vậy mà những công nghệ kèm theo cũng đòi hỏi những nhà phát triển phần mềm viết ra nhiều phần mềm hơn để có thể sử dụng được trên smartphone chứ không còn đơn thuần là những ứng dụng nhắn tin gọi điện thông thường. Kèm theo đó là số lượng người dùng di động (smartphone) tăng lên chóng mặt.

- Nhiều công ty sản xuất phần mềm trên di động hoặc những công ty lớn nhìn thấy đây là mảnh đất màu mỡ và tiềm năng phát triển của nó rất lớn. Đồng thời dựa theo số liệu thống kê như trên thì bắt đầu người dùng đã chịu chi tiền cho smartphone nhiều hơn, số lượng người dùng chịu bỏ tiền ra mua ứng dụng để sử dụng trên smartphone cũng tăng lên theo từng năm. Do đó kéo theo nhu cầu về công việc liên quan đến mảng smartphone nói chung và lập trình mobile nói riêng tăng lên chóng mặt. Đưa ra những tiềm năng và thử thách dành cho những ai muốn phát triển theo hướng lập trình ứng dụng di động. Nhu cầu dạy và học lập trình ứng dụng di động đang dần trở thành xu hướng của xã hội hiện nay.

Ngày nay chúng ta có hai nền tảng cạnh tranh chính là iOS và Android, mỗi kẻ một thế lực làm khuynh đảo giới lập trình. Từ đó sinh ra các công cụ hỗ trợ nhà phát triển ứng dụng có thể phát triển mỗi ứng dụng một cách dễ dàng chẳng hạn như Apple thì có XCode, Android thì có Android Studio.

- Nhưng cũng chính vì lẽ mỗi kẻ một thế lực nên không ai chơi với ai, mỗi bên xưng đế một “phương” cuối cùng con dân lập trình viên là những người phải chịu trận. Và cũng từ lẽ đó mà giang hồ sinh ra hai môn vang danh thiên hạ là Native và Hybrid? Trước khi đi sâu vào React Native thì chúng ta nên biết Native và Hybrid là cái chi chi và những lợi và hại giữa hai môn phái này. Thông qua đó chúng ta sẽ biết được tại sao mà React/React Native lại được khai sinh ra đời để giải quyết vấn đề nan giải của con dân lập trình viên di động.

- React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động. Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái luôn, sợ chưa!!! Một điểm hay ho nữa mà mình có đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là bạn không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa.

## 2. Kiến thức cơ bản về ES6 trong React native

- ES6 là chữ viết tắt của ECMAScript 6, đây được coi là một tập hợp các kỹ thuật nâng cao của Javascript và là phiên bản mới nhất của chuẩn ECMAScript

Đây là danh sách tính năng cơ bản của ES6

* Arrow function
* Block Scoped:
* Destructuring Assignments
* Default Parameters
* Template String
* Weak, Set, Map

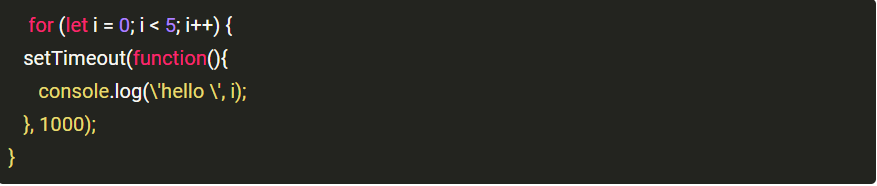
1. Arrow function

Có thể tạo hàm bằng cách sử dụng dấu mũi tên =>.



1. Block scoped

Định nghĩa biến với từ khóa let, cách định nghĩa này thì biến chỉ tồn tại trong phạm vi khối của nó (Block Scope)



Kết quả trả về sẽ là:

hello 1

hello 2

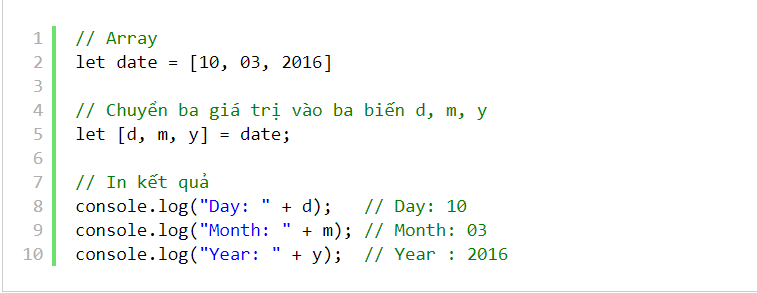
hello 3

hello 4

hello 5

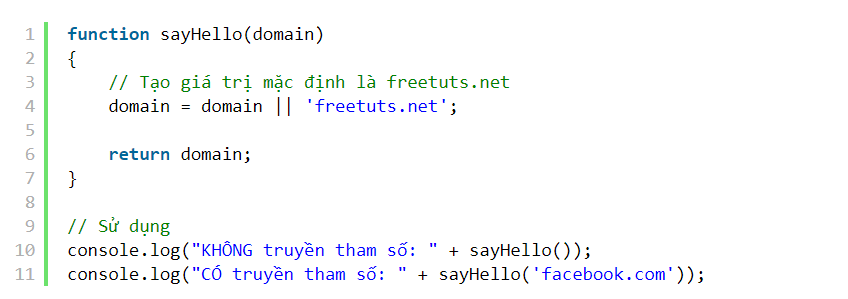
1. Destructuring Assignments

Có thể khởi tạo các biến từ một mảng bằng một dòng code đơn giản



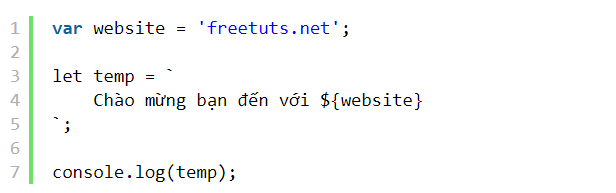
1. Default Parameters

Là giá trị mặc định của tham số khi truyền vào các function.

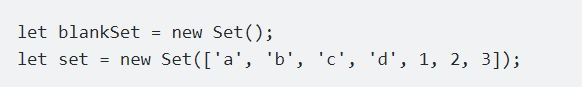


1. Template String

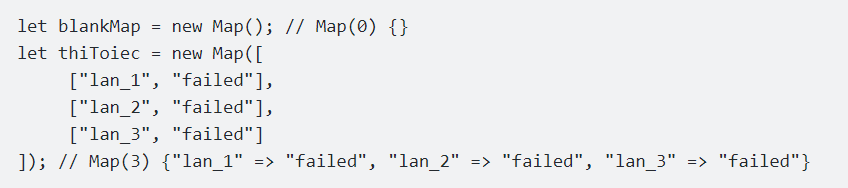
Thay thế cách nối chuỗi thông thường



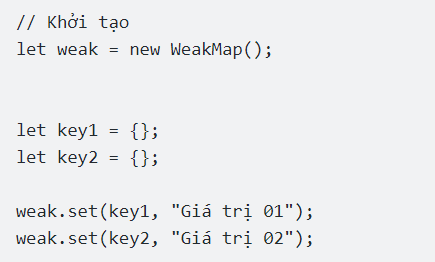
1. Weak, Set, Map
   * Set: tập hợp các phần tử khác nhau và không có key



* Map: một kiểu dữ liệu tương tự như Set, tuy nhiên với Map thì có cấu trúc dạng key => value còn với Set thì chỉ có value



* WeakMap: một loại kiểu dữ liệu giống như Map . Chỉ khác là với WeakMap thì key truyền vào phải là một biến và biến này phải là một Object



* WeakSet: thì lại giống như Set, tuy nhiên với WeakSet thì dữ liệu truyền vào luôn phải là một đối tượng



## 3. Component trong react native

* **Component** chính là 1 trong những lí do khiến ứng dụng viết bằng React/React Native có khả năng tái sử dụng và tăng khả năng mở rộng
* Không chỉ có vậy, điểm mạnh thứ 2 liên quan đến Component chính là việc React Native cung cấp cho chúng ta rất nhiều các component mặc định, ví dụ như Text. Khi chúng ta build app trên iOS, React Native sẽ đảm nhận việc chuyển component này thành UIView chứa đoạn text, còn khi build trên Android thì sẽ thành TextView native.

Props là viết tắt của Properties. Một điều mà bạn cần phải nhớ khi sử dụng props đó là không bao giờ nên thay đổi giá trị của nó, hay nói cách khác, đây là một dữ liệu immutable.

- Các component nhận props từ component cha. Bạn không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra thôi. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha => các component con.



- **State** thì hoạt động khác với props. state là dữ liệu nội bộ của một Component, trong khi props là dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu mutable.



## 4. API Trong React Native

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành. api Mỗi bộ API dành cho các hệ điều hành khác nhau là hoàn toàn khác nhau và không có sự tương thích với nhau. API dành cho các hệ điều hành Windows và Linux là hoàn toàn khác nhau. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Nói nôm na API là một loại công cụ để tạo ra phần mềm, cũng giống như các viên gạch xây nên một toà nhà. Chúng ta chia việc xây nhà ra các phần khác nhau, API cũng như viên gạch là một bộ phận của ngôi nhà, nhưng việc tạo ra nó là một công đoạn riêng hoàn toàn, chúng ra có thể tự làm hoặc đơn giản là đi mua của các nhà cung cấp. Web API là một trong những công nghệ mới của Microsoft dùng để xây dựng dịch vụ thành phần phân tán. Web API là mô hình dùng để hỗ trợ MVC bao gồm: routing, controller, action result, filter, filter, loc container, model binder, unit test, injection. Bên cạnh đó nó còn hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu.

React Native cung cấp module Fetch API để sử dụng cho việc kết nối network. Fetch sẽ rất thân thuộc nếu như bạn đã đã từng sử dụng XMLHttpRequest hoặc các networking APIs trước đây

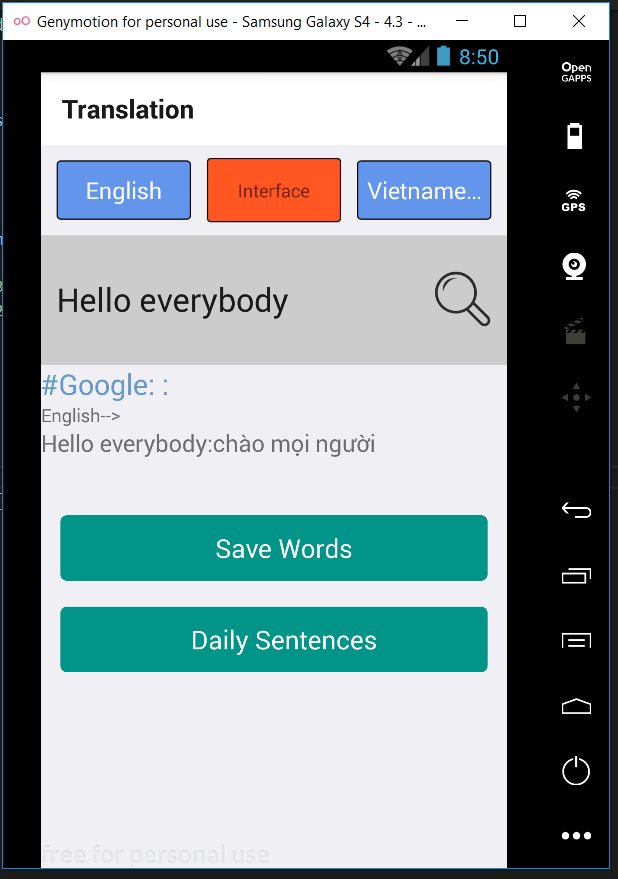
Bạn không bị giới hạn đối với các thành phần và API đi kèm với React Native. React Native là một cộng đồng gồm hàng ngàn nhà phát triển. Nếu bạn đang tìm kiếm một thư viện làm một cái gì đó cụ thể, hãy tìm kiếm sổ đăng ký npm để biết các gói đề cập đến Reac -igen hoặc xem Awesome React Native để biết danh sách

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Phân tích hệ thống

### 2.1.1. Feature/Component #1: Dictionary screen

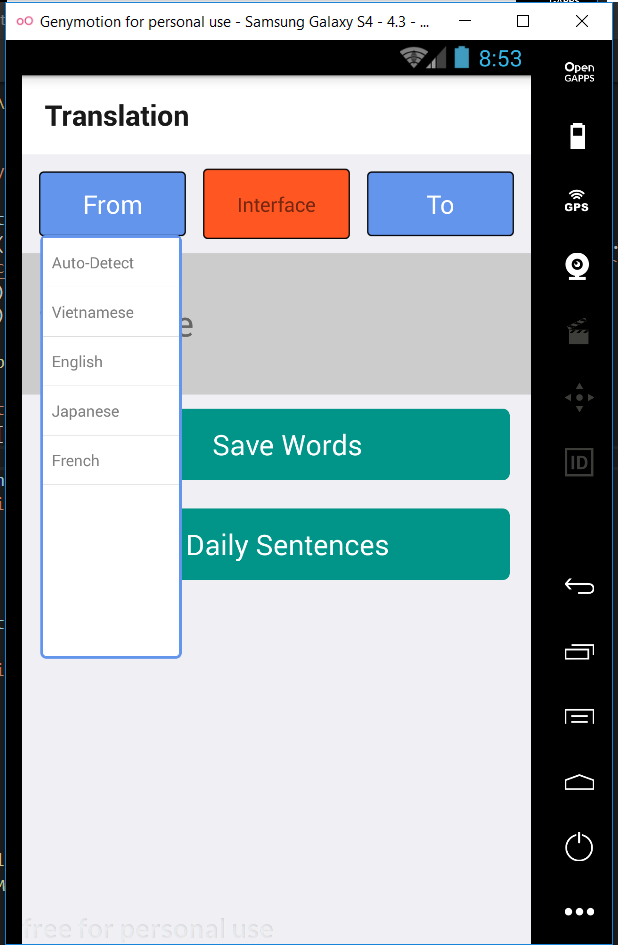
1. **Chức năng dịch từ**



**Mô tả:**

- Chức năng dịch cho phép người dùng nhập vào 1 từ hay 1 đoạn văn. Ứng dụng sẽ dịch sang nghĩa mà người dùng chọn ở chức năng chọn ngôn ngữ

1. **Chức năng chọn ngôn ngữ**

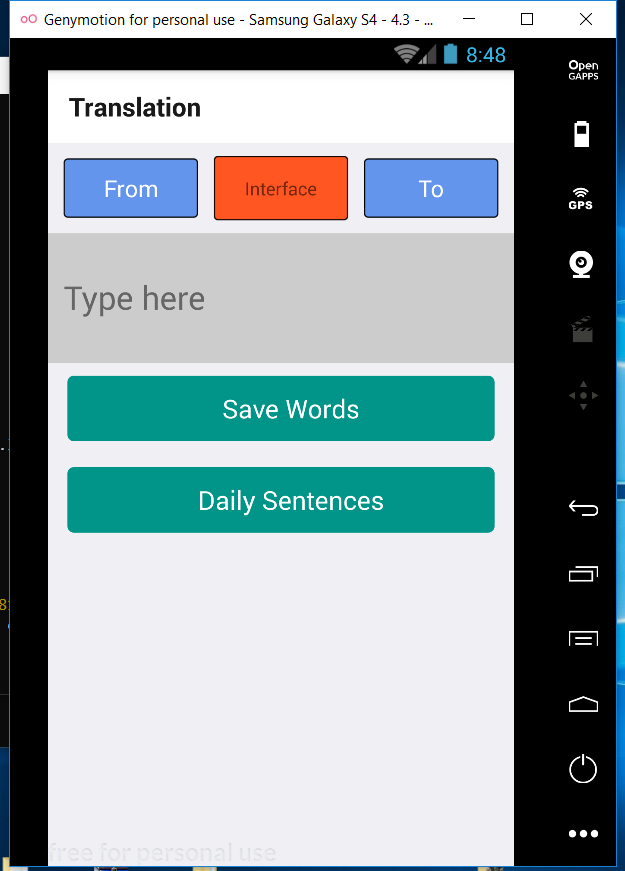
****

**Mô tả:**

Người dùng có thể chọn ngôn ngữ mình nhập vào giúp cho ứng dụng trở nên linh hoạt hơn có thể dịch sang nhiều thứ tiếng khác nhau

1. **Chức năng lưu từ**

#### 2.1.1.1 User Interfaces



*Image1: Registration*

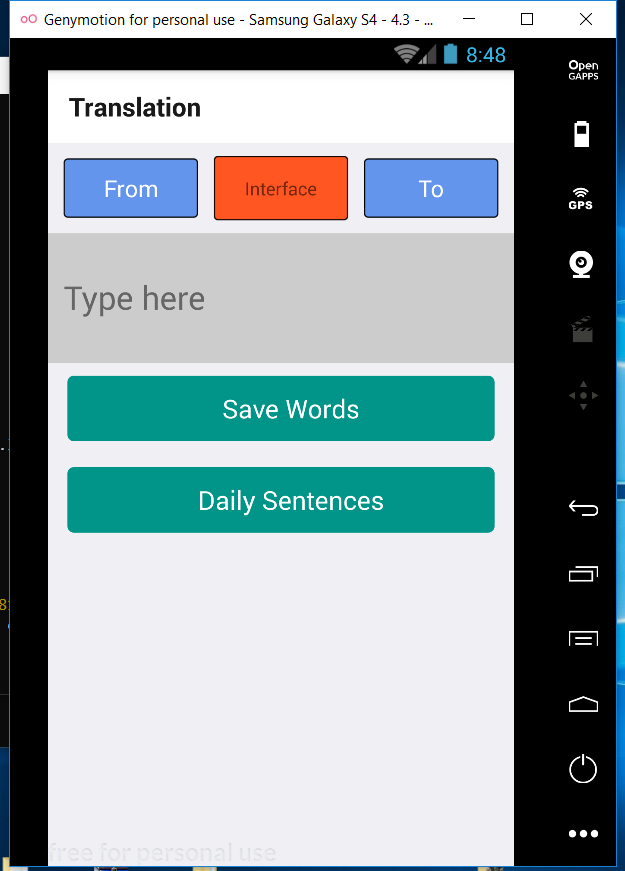
#### 2.1.1.2. Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **Dictionary screen** | Dictionary Screen has 5 buttons:   * English * Interface * VietNamese * Save words * Daily sentences * Text input |  |  |
| **Text Input** | Use to input word or document want to translate | Input with keyboard | The text input will appearance word or document user input |
|  |  |  |
| **Button From** | Select input language. The language user choose will appearance in button | Tap on from button | A list language for user choice |
| **Button To** | Select output language. The language user choose will appearance in button | Tap on the button | A list language for user choice |
| **Button Interface** | Go back to main screen | Tap on the button | Go to interface user |
| **Save words** |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 2.2 Thiết kế hệ thống

### 2.2.1. Sudoku Main Screen

#### 2.2.1.1. Screen Shot for Dictionary Main Screen

****

#### 2.2.1.2. Objects and actions for Dictionany Main Screen

Objects:

* Interface
* Save words
* From
* To

Actions:

* Main screen
* Save new words
* Tranfer language input
* Tranfer language output

# CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ

## 3.1 Cài đặt

Mô tả cách thức hiện thực hoá cho từng Activity, Service… của hệ thống

## 3.2 Kiểm thử

Các Test case của hệ thống là gì?

Kết quả kiểm thử ra sao?

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

## 4.1 Kết quả đạt được

Mô tả các kết quả khi chạy ứng dụng (chụp các màn hình khi chương trình chạy, mô tả cách chạy, kết quả ra sao…)

## 4.2 Các kết luận và kiến nghị

- Những điểm đã làm được

- Những điểm chưa làm được

- Các chức năng bổ sung nếu có thêm thời gian…

# PHỤ LỤC

# TÀI LIỆU THAM KHẢO