TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

KHOA CÔNG NGHỆ THÔNG TIN

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

BÁO CÁO KẾT THÚC MÔN HỌC

Lập trình di động 3

**DICTIONARY MOBILE**

**APPLICATION**

Giảng viên hướng dẫn: **TRƯƠNG BÁ THÁI**

Sinh viên thực hiện:

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

Ngành: Công nghệ thông tin Lớp: Lập trình di động 3 - Khoá: 16

*Tp. Hồ Chí Minh*, ngày 20 tháng 12 năm 2018

**NHẬT KÝ HOẠT ĐỘNG NHÓM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Họ và tên** | **Công việc thực hiện** | **Tự đánh giá** | **Nhóm đánh giá** | **Chữ ký** |
| 1 | Nguyễn Anh Toàn | 1. Thiết kế giao diện  2. Cài đặt, sử dụng react-native  3. Cài đặt, thiết lập, sử dụng google-translate-api | 8 | 9 |  |
| 2 | Nguyễn Đông Duy | 1. Thiết kế giao diện: Daily Senetences  2. Sử dụng API để lấy tin tức theo ngày | 8 | 9 |  |
| 3 | Nguyễn Phương Linh | 1. Đọc và lưu từ vựng lại sau khi search  2. Thiết kế json để khi lưu từ vựng đúng định dạng | 8 | 9 |  |
| 4 | Lê Minh Đạt | 1. Chỉnh sửa, thiết kế các components  2. Trang trí giao diện ( các file style, css..)  3. Tìm hiểu các components | 8 | 9 |  |

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. Cụ thể theo những báo cáo mới đây mà các nhà khoa học đã thống kê thì thời gian dành cho smartphone trung bình qua khảo sát hàng tỉ người thì rơi vào khoảng 3 tiếng một ngày.

Đồng thời giá smartphone đang ngày càng rẻ qua từng năm, vì các dòng smartphone giá rẻ đến từ các quốc gia phát triển và đông dân như Ấn Độ, Trung Quốc. Cùng với đó công nghệ sản xuất càng dần hoàn thiện hơn kéo giá smartphone xuống tới các tầng lớp phổ thông của xã hội. Từ đó smartphone trở thành một thiết bị có thể được sở hữu dễ dàng.

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.

Giới thiệu với các bạn, gương mặt đứa con của chúng ta hôm nay đây rồi React Native. 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.

Và điểm lợi hại kế tiếp của React Native đó chính là chúng ta chỉ cần sử dụng JS để phát triển được một ứng dụng di động hoàn chỉnh, đồng thời giải quyết được các vấn đề mà Native App gặp phải mà mình đã nêu ở trên. Và rồi còn cả kết hợp với code native như Swift, Java, v.v…

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

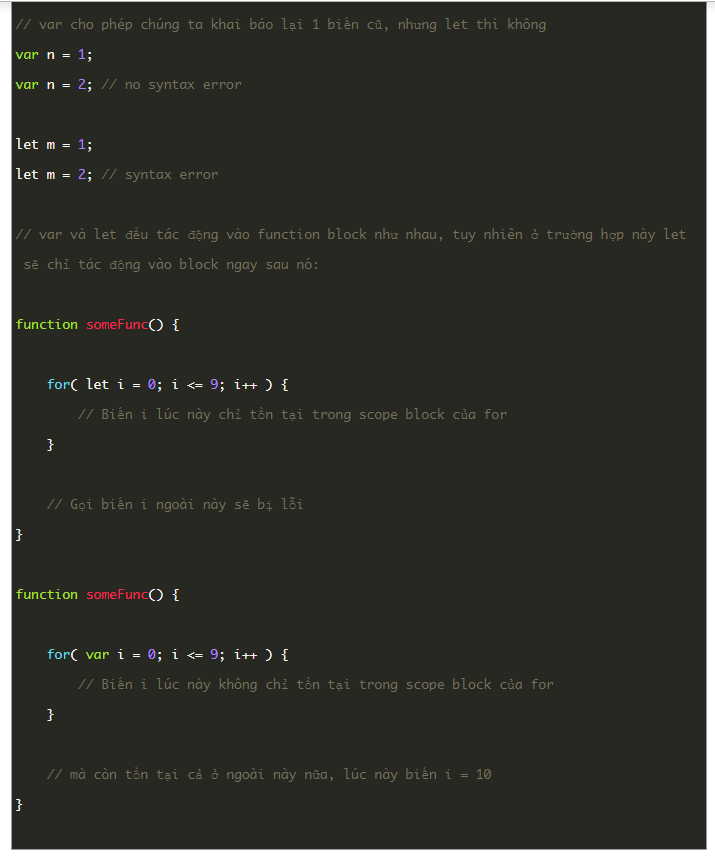
Dưới đây là các plugin ES6/7 mà React Native đang sử dụng và bài viết này mình tập trung vào những syntax mà mình thấy quan trọng nhất trong React Native.



### **Khai báo biến với**var**,**let**, và**const

Khi khai báo biến với Const, biến đó sẽ là immutable variable, nghĩa là sẽ không thay đổi được giá trị của biến.

Với var và let, chúng ta đều có thể khai báo được 1 biến bất kỳ, biến này có thể thay đổi được giá trị. Điểm khác biệt giữ var và let đó là:



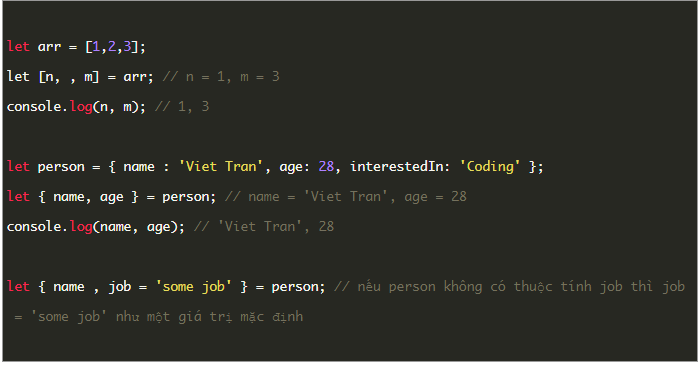
Arrow function cũng như function bình thường, chỉ khác về syntax và binding context:

### **Module import / export**

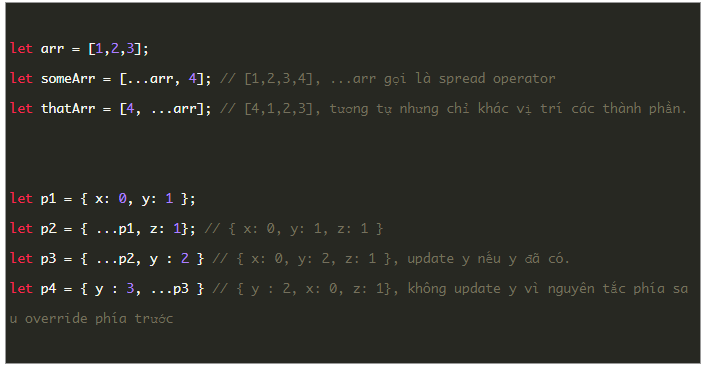
Ứng dụng RN thường được phát triển trên nhiều file JS mà ta thường gọi là các module / component. Tất cả các biến và function trong 1 file JS sẽ chỉ được truy xuất trong file (hay còn gọi là file private). Để cho phép các thành phần có thể sử dụng từ các file khác, chúng ta cần dùng tới từ khoá export và import.

### **Function Parameter: default và rest**

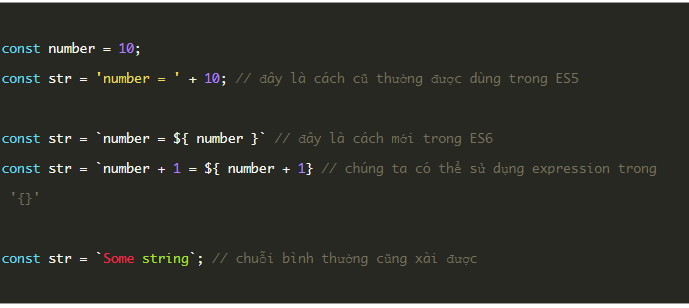
### **Object/Array Matching, Short Hand Notation**



### **Spread Operator**

Đây là một trong những operator quan trọng chúng ta rất hay dùng trong RN. Trong clip hướng dẫn RN cơ bản mình cũng có demo quản lý style component con với Spread Operator.

### **String interpolation**





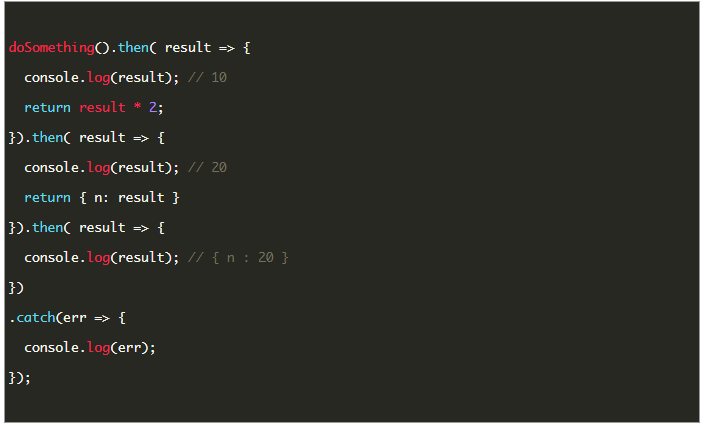
### **Classes**

### **Promise và parallel promise**

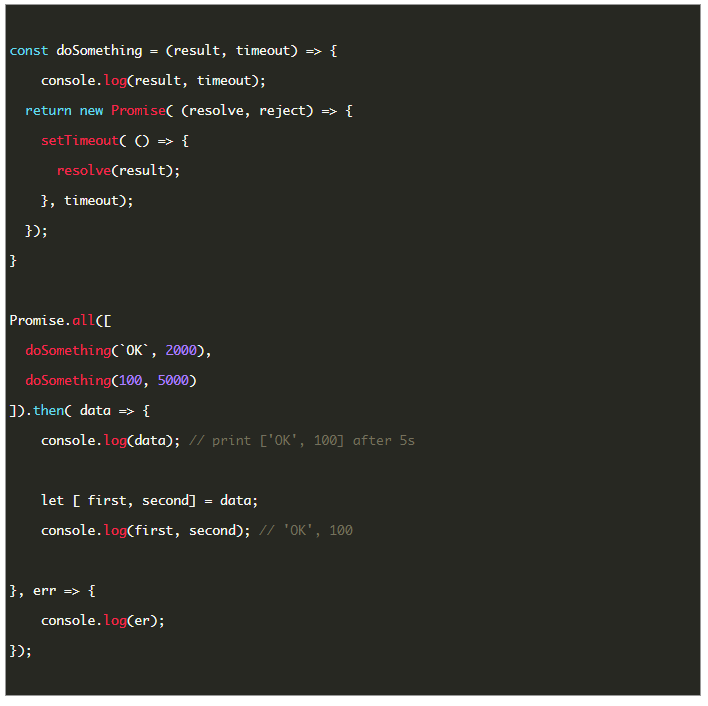
Việc sử dụng Promise là một giải pháp hiệu qủa khi làm việc với các hàm callback, sourcecode chúng ta sẽ dễ đọc hơn.



Chúng ta có thể sử dụng hàm then như một middleware, ở mỗi bước then ta có thể return để làm tham số cho hàm then tiếp theo

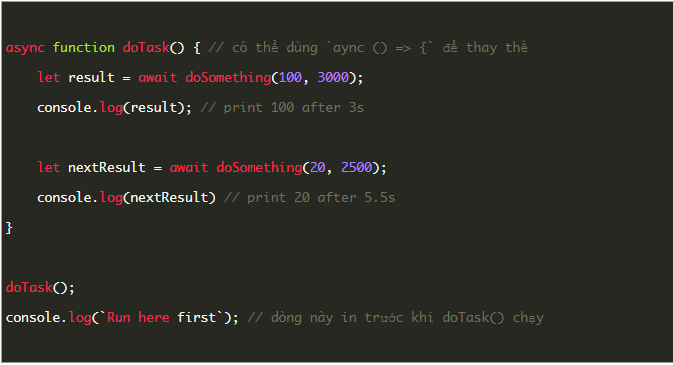


Chạy cùng lúc nhiều Promise với Promise.all. Việc này rất hiệu quả khi ta cần load 1 lúc nhìu APIs hoặc nhiều async tasks song song.



### **Async và Await**

Thế giới của JS đầy rẫy những callback function và promise, thế nhưng lắm lúc ta lại cần chúng chạy synchonize bình thường, hay nói đúng hơn ta sẵn sàng đợi cho chúng chạy xong. Source code sẽ chạy lần lượt từ trên xuống dưới.

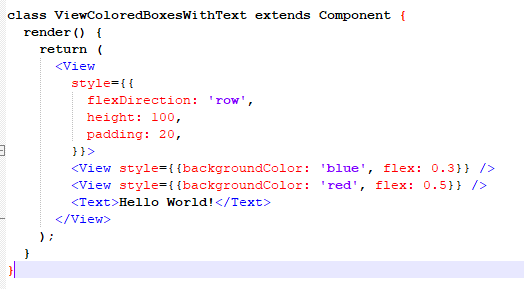


## lưu ý: Từ khoá wait chỉ chạy trong function được khai báo với từ khoá async. Vì function này async nên sẽ chạy bất đồng bộ (chạy ở 1 thread khác) nên ở trên ta thấy Run here first sẽ được in ra đầu tiên. Trong function doTask, từ khoá await sẽ khiến doSomething chạy như synchronize (block thread hiện tại để đợi kết quả).

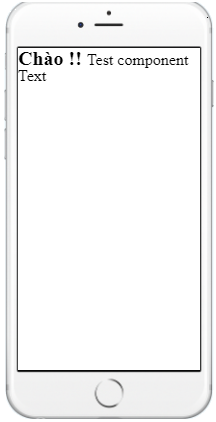
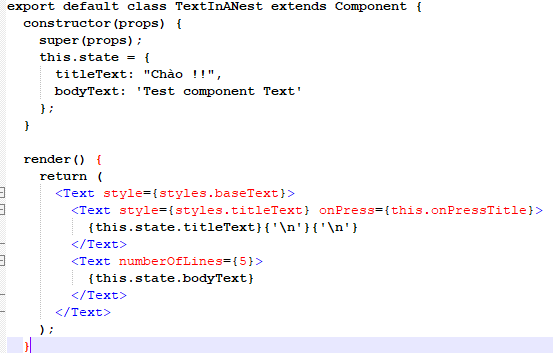
## 3. Component trong react native

1. **The basic component**

* View



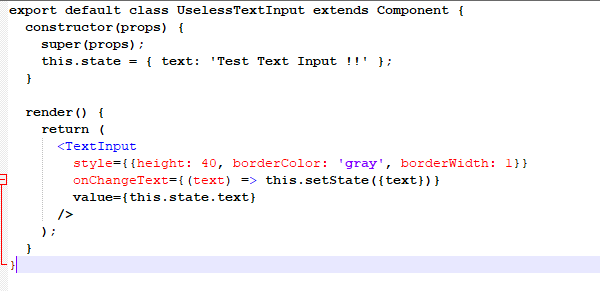
* Text : Giúp hiện thị văn bản lên màn hình

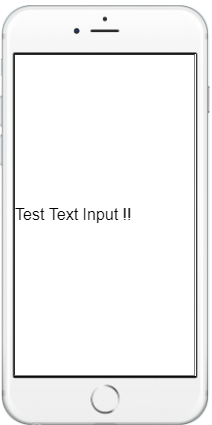


* Image: Đưa hình ảnh vào màn hình

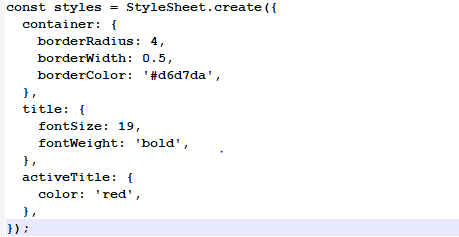


* Text Input : Nhập văn bản thông qua bàn phím

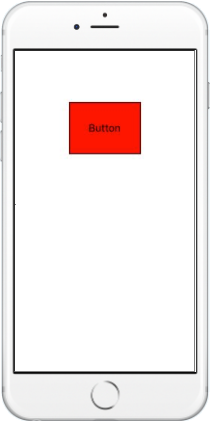


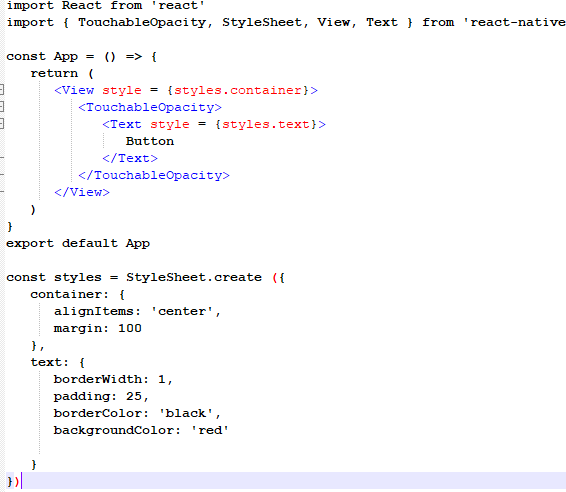


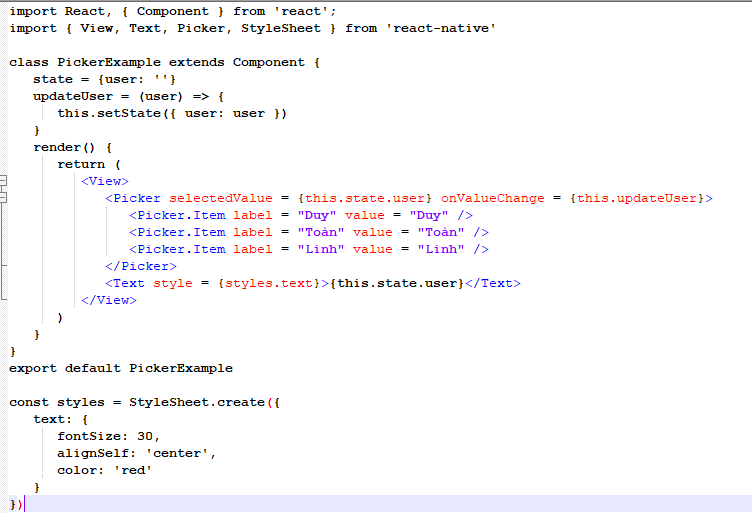
* StyleSheet : Thành phần dùng để viết CSS

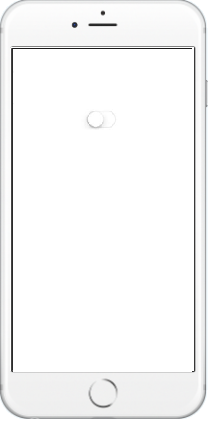
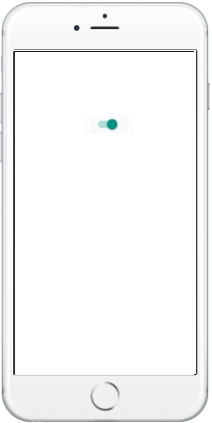


1. **User interface**

* Button



* Picker
* Switch



## 

## 4. API Trong React Native

React Native cung cấp module [Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/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 có thể sẽ cần phải tham khảo hướng dẫn sử dụng Fetch [tại đây](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch) để có được thêm các thông tin

#### Tạo một HTTP Request

Khi bạn muốn lấy nội dung bằng cách gọi đơn giản nhất từ một URL, rất đơn giản bạn chỉ cẩn đặt URL đó trong fetch:



Fetch sẽ có một số tùy chọn tham số để bạn có thể tùy chỉnh HTTP request. Ví như bạn muốn thêm cụ thể một header nào đó và muốn gọi với phương thức POST. Ví dụ dưới đây sẽ cho các bạn thấy một cách đơn giản các tùy chọn:



#### Xử lý response

Networking bản chất là một hình thức bất đồng bộ (Lan man một chút, vì sự bất đồng bộ này mà ở Android từ API 11 trở lên hệ điều hành đã ngăn cản việc chạy Network trên main thread để ngăn cản độ trễ của chương trình trong thời gian chờ dữ liệu mạng được trả về dưới client). Phương thức Fetch sẽ trả về một [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise), điều này sẽ dễ dàng để các bạn có thể viết các đoạn code xử lý cho các thao tác bất đồng bộ:



Bạn cũng có thể sử dụng mẫu cấu trúc ES2017 về async/await trong ứng dụng React Native:



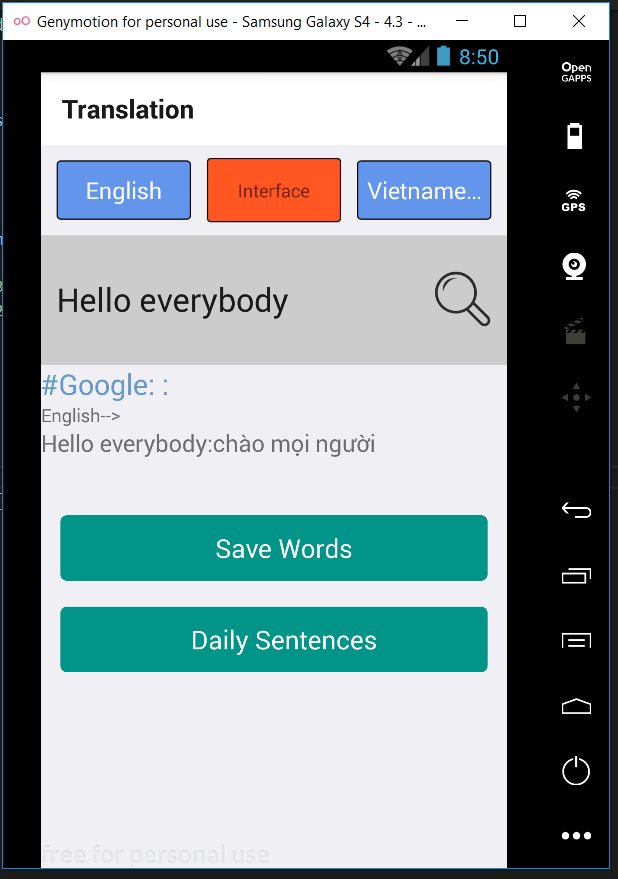
Đừng quên câu lệnh catch để bắt bất kỳ một lỗi nào xảy ra khi mà bạn thực hiện fetch. Bên cạnh đó chúng ta cũng không nên âm thầm bỏ qua các lỗi.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Phân tích hệ thống

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

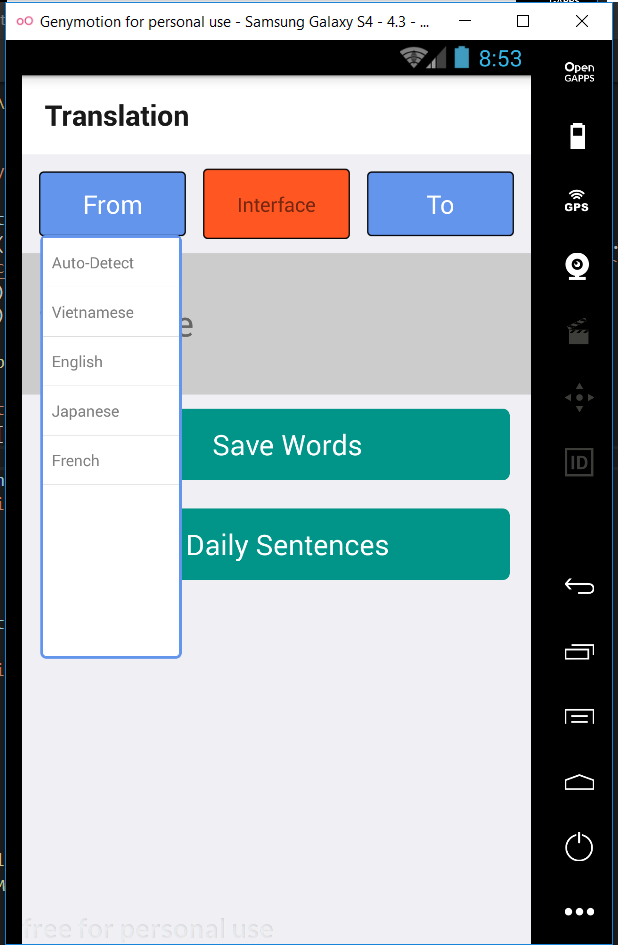
1. **Chức năng dịch từ**



**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

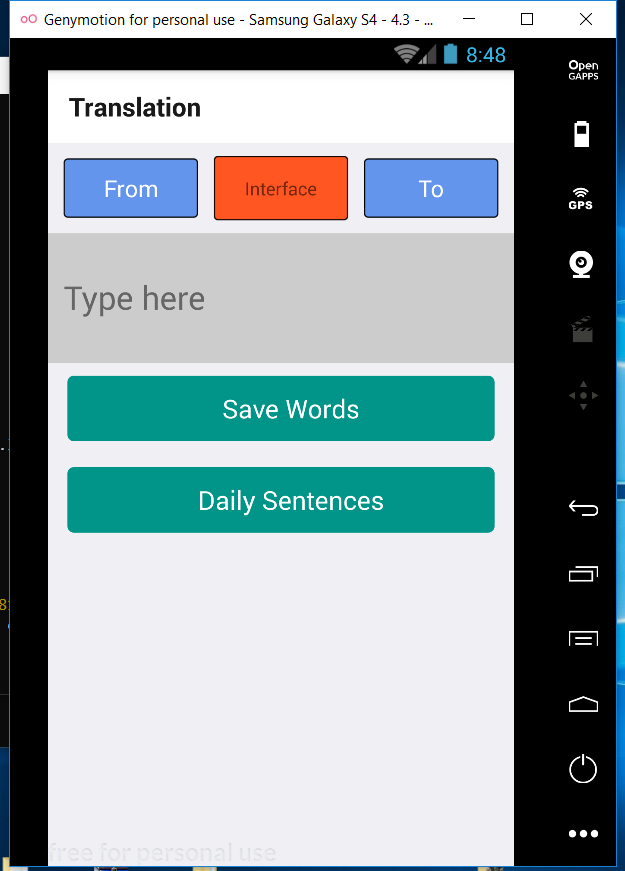
1. **Chức năng chọn ngôn ngữ**

****

**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

#### 2.1.1.1 User Interfaces



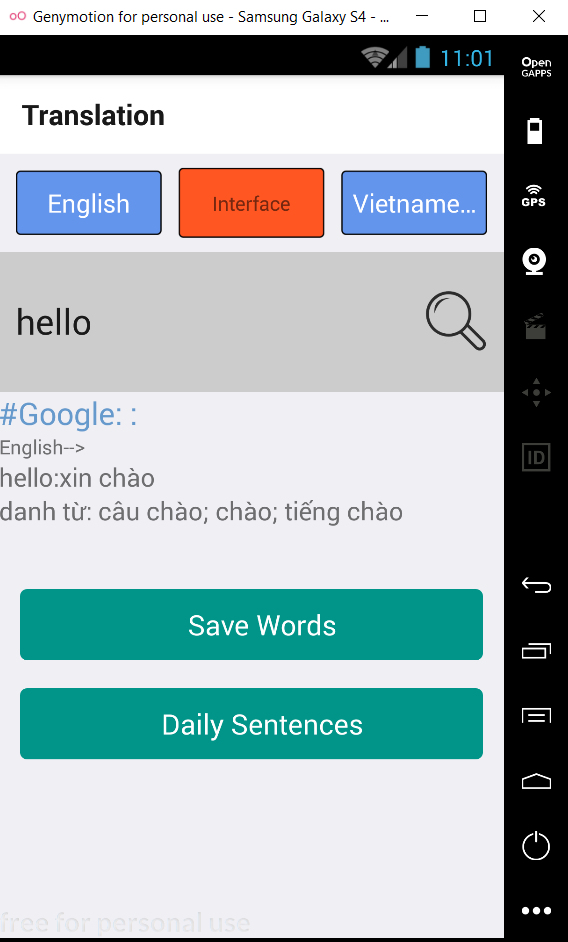
# 2.1.1.2. Functional Requirements

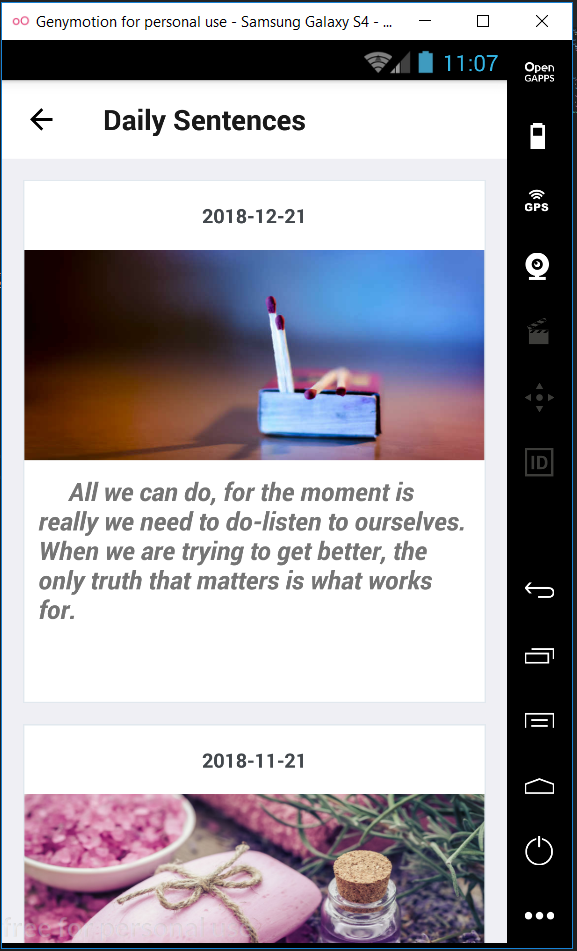
|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **Chọn ngôn ngữ gốc** | Bấm vào button chọn ngôn ngữ gốc, để muốn dịch sang ngôn ngữ đích.  Có 4 loại ngôn ngữ:   * Tiếng Nhật * Tiếng Anh * Tiếng Việt * Tiếng Pháp | Bấm vào Button | Hiển thị ngôn ngữ được chọn |
| **Chọn Interface (dùng nhiều loại API khác nhau: google, bing…)** | Dùng nhiều loại API để dịch từ vựng khác nhau.  Có thể sử dụng API của Google, Bing, Youdao, Baidu… | Bấm vào Button | Hiển thị các loại API muốn sử dụng |
| Có thể chọn dịch một lúc nhiều API | Bấm chọn checkbox | Checkbox ticked |
| **Chọn ngôn ngữ cần dịch sang** | Bấm vào button chọn ngôn ngữ muốn dịch sang  Có 4 loại ngôn ngữ:   * Tiếng Nhật * Tiếng Anh * Tiếng Việt * Tiếng Pháp | Bấm vào Button | Hiển thị ngôn ngữ được chọn |
| **Điền từ cần dịch vào inputText** | Điền từ cần dịch vào ô inputText | Bấm chọn vào inputText, điền word cần dịch vào | Hiển thị từ được nhập vào lên inputText cho người dùng nhìn thấy |
| **Button tìm kiếm** | Gọi hàm tìm kiếm và lấy từ cần dịch ở inputText | Bấm vào button kính lúp | Dịch từ vựng, và hiển tự kết quả vào listview |
| **Button Save Words** | Lưu các kết quả từ vựng đã tìm kiếm vào file json | Bấm vào button | Hiển thị popup đã lưu từ vựng |
| **Exit** | Closed application | Tap on Exit button | Application is closed |
| **Button Daily Sentences** | Hiển thị các nội dung, câu châm ngôn, hình ảnh theo ngày. ( Được dùng API ) | Bấm vào button | Navigation các được hiện ra, hiển thị các nội dung Sentences theo ngày |

## 2.2 Thiết kế hệ thống

### 2.2.1. Dictionary Mobile Application Main Screen

#### 2.2.1.1. Screen Shot for Dictionary Mobile Application

****

****

#### 

#### 2.2.1.2. Objects and actions for Dictionary Mobile Application

Objects:

* Original Languages
* Languages translate
* Words translated
* Button change interface
* Save Word
* Daily Sentences

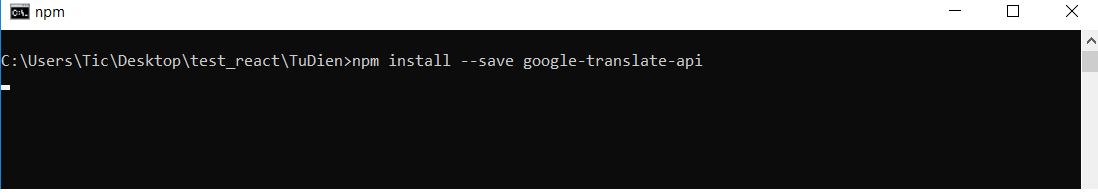
Actions:

* Translate words
* Save words
* Show words already
* Daily Sentences

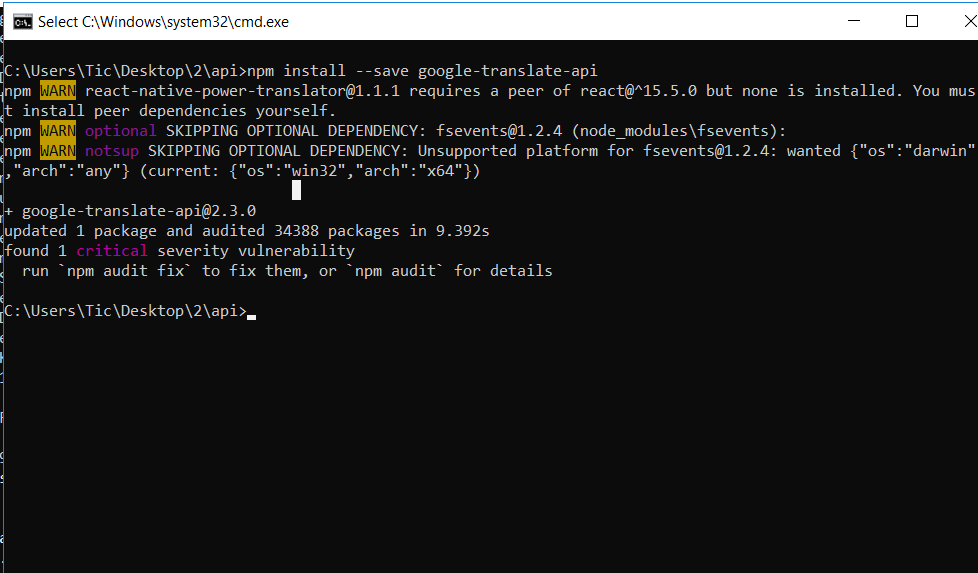
# CHƯƠNG 3. CÀI ĐẶT VÀ SỬ DỤNG GOOGLE TRANSLATE API

**3.1. Cài đặt Google Translate Api:**

Cần cài đặt packed vào thư mục react-native đã được cài đặt bằng lệnh:



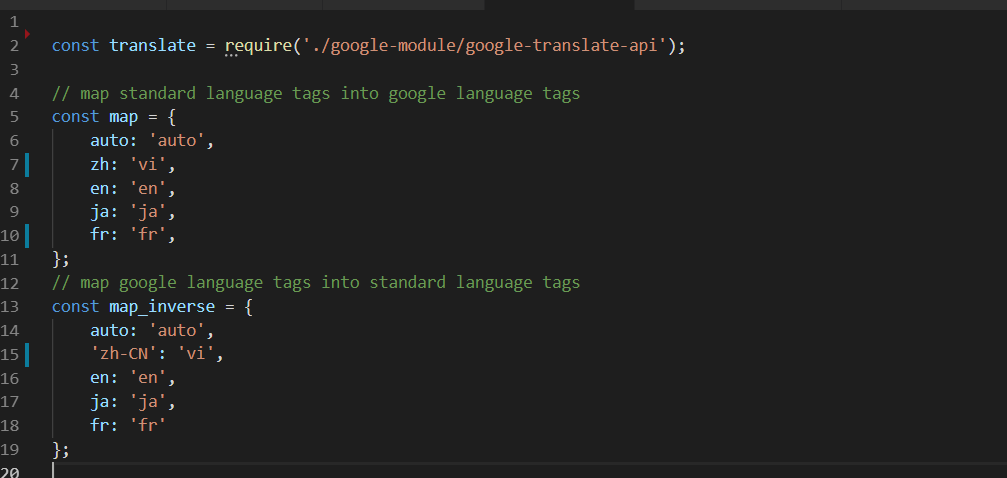
Hệ thống sẽ tiến hành cài đặt các gói mở rộng để có thể sử dụng google-translate-api.



**3.2. Khai báo và sử xây dựng hàm xử lý dịch**

Sau khi cài đặt, chúng ta có thể sử dụng module của Google-translate api một cách dễ dàng.

Khai báo để sử dụng module, và chọn những loại ngôn ngữ muốn chuyển đổi, dịch:

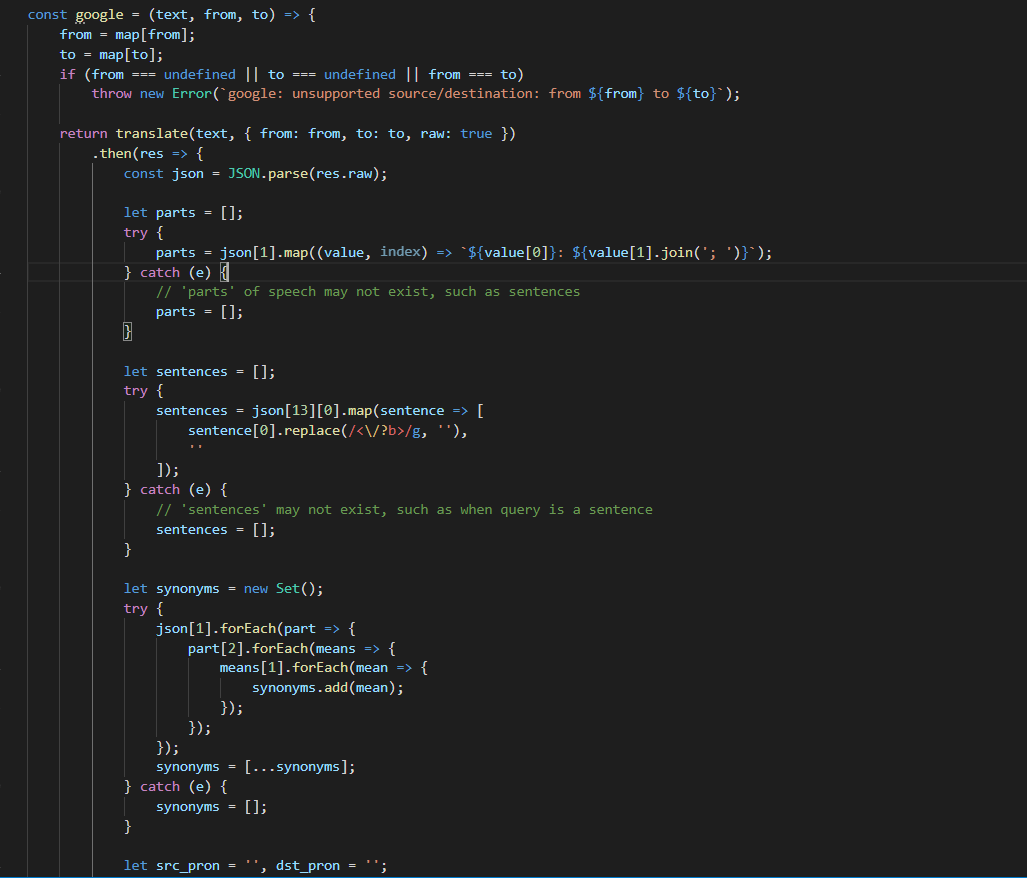


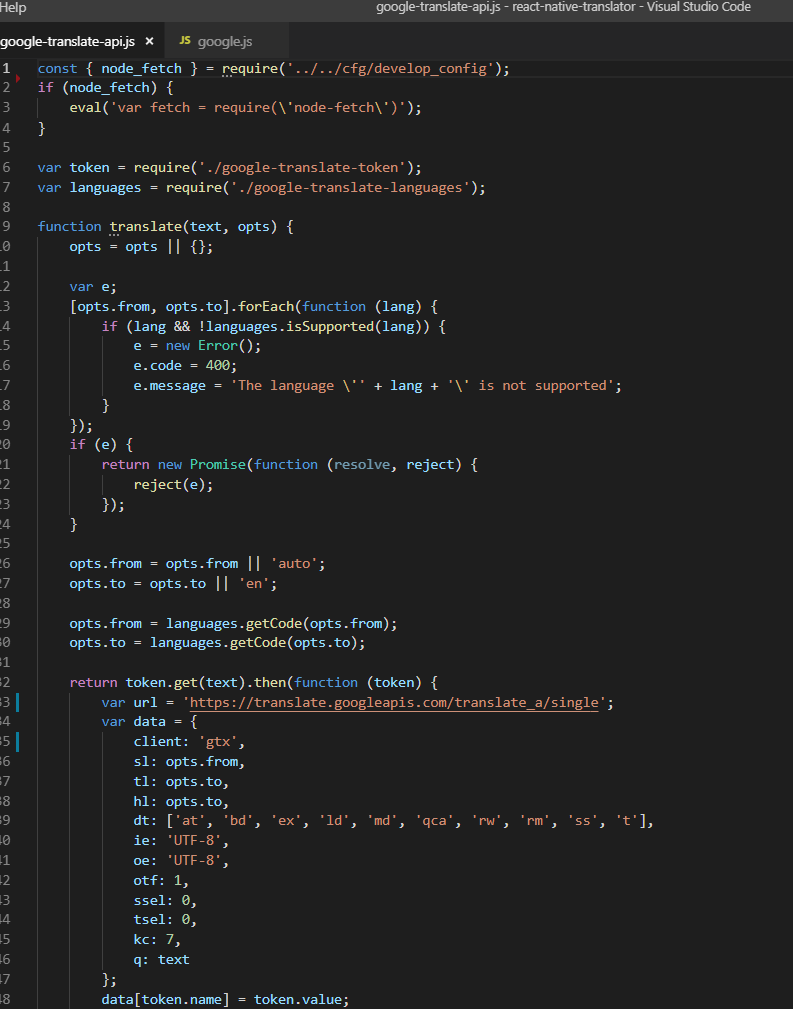
# 

Có rất nhiều loại ngôn ngữ có thể dịch sang, API của Google đã hỗ trợ rất nhiều việc đó, chúng ta có thể thêm nhiều loại ngôn ngữ, và dịch sang ở trong module này.

Vì đây là một loại API có trả phí, khi sử dụng link get translate của google, cần phải lọc và replace những ký tự có trong chuỗi trả về ( replace <br>, replace \r\n, …)

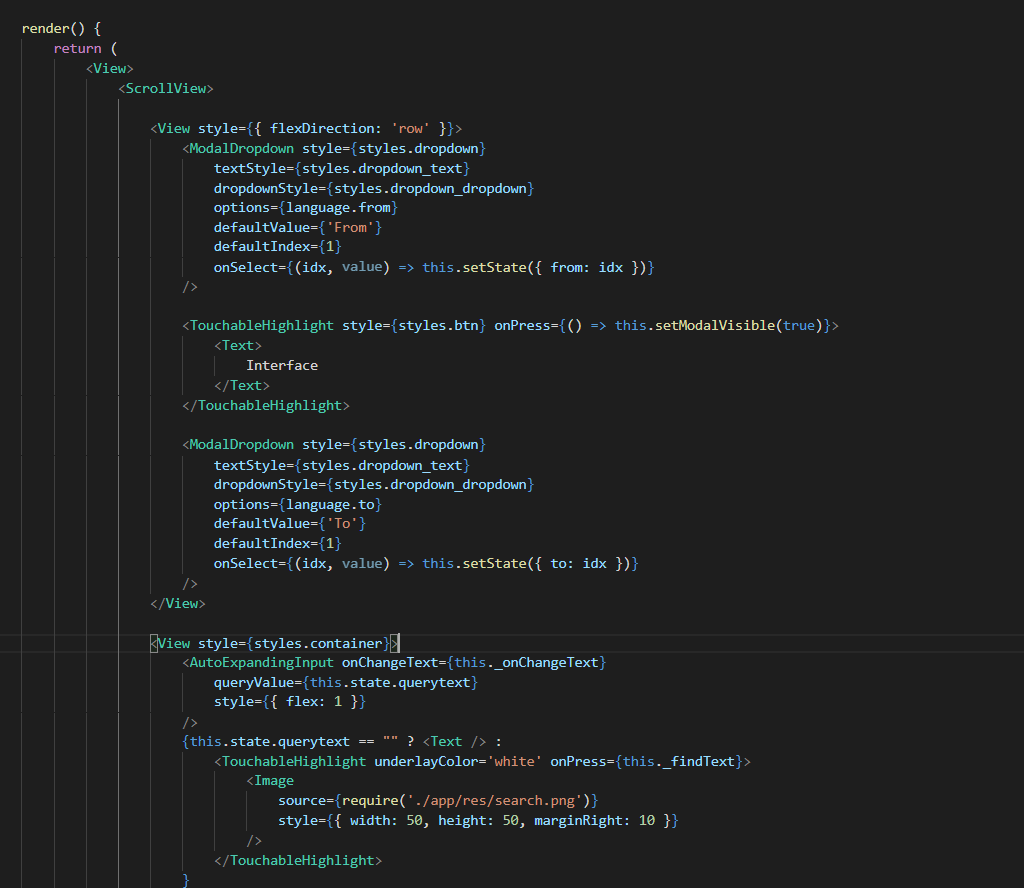
Xây dựng hàm chuyển đổi giữa các ngôn ngữ, và sử lý các chuỗi json được trả về từ API



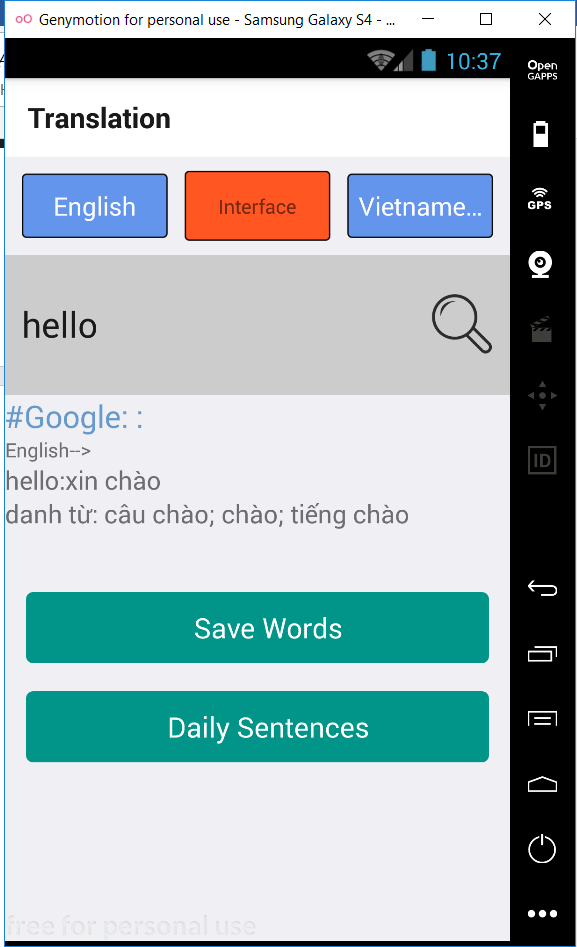


**3.3. Test chương trình khi dùng Google-translate-API:**

Tiến hành add các hàm trong API vào các button:



Chạy thử chương trình:



# CHƯƠNG 4. CÀI ĐẶT VÀ KIỂM THỬ

## 4.1 Cài đặt

- Sử dụng Genymotion để cài đặt và phát triển

- Cài đặt react-native

- Thiết kế hệ thống, giao diện

- Cài đặt modules google-api-translate

- Triển khai code phần mềm

- Sửa lỗi, debug

- Tìm hiểu các components

- Áp dụng, sử dụng các components vào project

- Chỉnh sửa giao diện

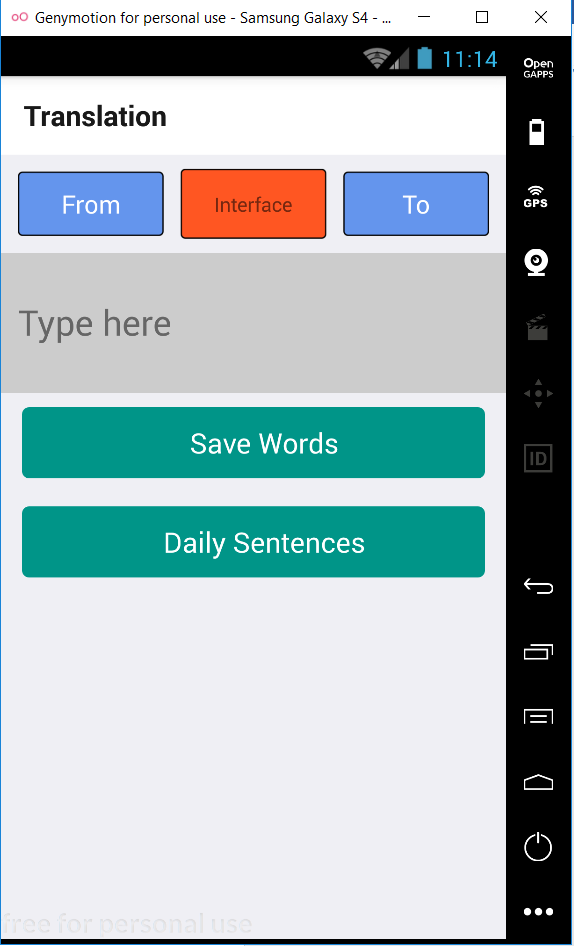
- Kiểm thử

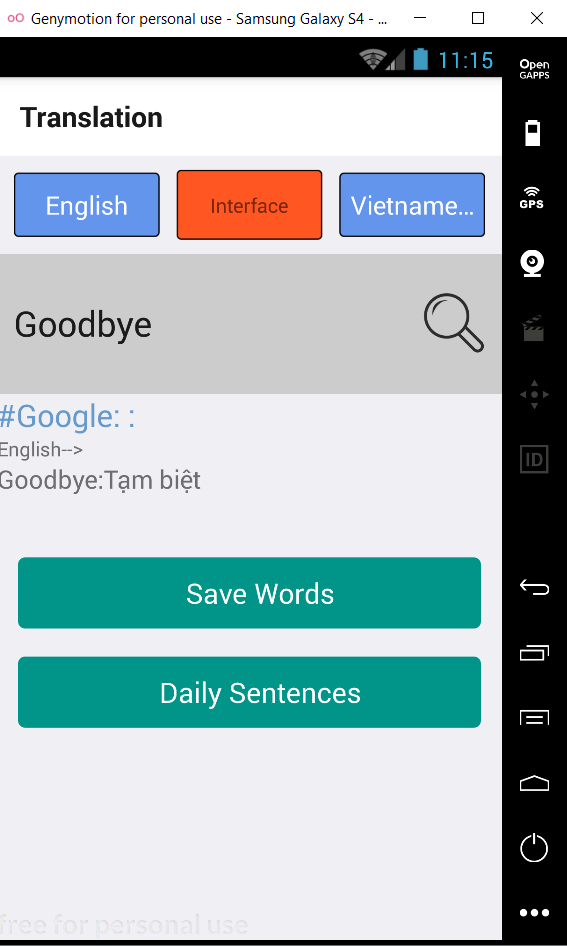
## 4.2 Kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Actual Result** | **Status (Pass/Fail)** | **Note** |
| 1 | Test Interface action | 1.Nhấp vào interface 2.Chọn google 3.Done | Pass |  |
| 2 | Test From Action | 1.Nhấp From 2.Chọn 1 ngôn ngữ trong 4 ngôn ngữ  3.Button From chuyển từ From sang ngôn ngữ ấy | Pass |  |
| 3 | Test To Action | 1.Nhấp To 2.Chọn 1 ngôn ngữ trong 4 ngôn ngữ  3.Button To chuyển từ To sang ngôn ngữ ấy | Pass |  |
| 4 | Test Text Input | 1.Click vào Text Input 2.Nhập từ cần dịch vào text input | Pass |  |
| 5 | Test button Save Words | Tap on button | Pass |  |
| 6 | Test button interface | Tap on button | Pass |  |
| 7 | Test button Daily Sentences | Tap on button | Pass |  |
| 8 | Test From | 1.Nhấp From 2. Hiện thị những ngôn ngữ | Pass |  |
| 9 | Test To | 1.Nhấp To 2. Hiện thị những ngôn ngữ | Pass |  |
| 10 | Test find action | 1.Chọn ngôn ngôn ngữ cần dịch và ngôn ngữ dịch 2.Nhập từ vào Text Input 3.Xuất hiện từ đã được dịch | Pass |  |
| 11 | Test Save Words action | 1.Dịch thành công từ cần dịch  2.Nhấp save words 3.Save data successfully | Pass |  |
| 12 | Test Daily Sentences | 1.Nhấp daily sentences 2.Xuất hiện Daily sentences layout | Pass |  |
| 13 | Test the same language | 1.Ví dụ chọn 2 ngôn ngữ là english và english 2. Xuất hiện thông báo the two languages are the same | Pass |  |
| 14 | Test different languages | 1.Ví dụ chọn 2 ngôn ngữ là english và french 2. Từ nhập "hello" 3.Hiện ra là "Boujour" | Pass |  |

# CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC

## 5.1 Kết quả đạt được

 Các kết quả đạt được khi xây dựng ứng dụng:



## 

## 5.2 Các kết luận và kiến nghị

- Dịch được và chuyển đổi được từ vựng sang ngôn ngữ khác

- Sử dụng và chuyển đổi được 4 loại ngôn ngữ

- Dịch ngược các loại văn bản

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

+ Sử dụng thêm nhiều loại API hơn

+ Lưu những từ vựng được dịch vào database

# TÀI LIỆU THAM KHẢO

1. Github – website & source code react native: https://github.com/facebook/react-native

2. Google – website: https://google.com/

3. Wikipedia – website: <https://en.wikipedia.org/wiki/React_(JavaScript_library)>

4. React-native from facebook – website: <https://facebook.github.io/react-native/>