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: …………………………………………..

Sinh viên thực hiện:

1. ………………………………………..
2. ………………………………………..
3. …..........................................................
4. ………………………………………..
5. ………………………………………..
6. ………………………………………..

Ngành: Công nghệ thông tin Lớp: …………… 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

## Giới thiệu tổng quan về React native

* 1. **Tổng quan về React native**

React Native cho phép bạn xây dựng ứng dụng dành cho thiết bị di động( cả Android và IOS) mà chỉ cần dùng JavaScript. Nó sử dụng cùng một thiết kế như React, bạn có thể tạo một giao diện người dùng (UI) phong phú từ các components được tối ưu cho thiết bị di động.

Component của React Native được kế thừa từ React, và nó đóng vai trò như hạt nhân thúc đẩy React và React Native phát triển, với khả năng tái sử dụng và quản lý giao diện, Component API được xem như là tương lai của thiết kế UI trên web và mobile.

Với React Native, nó cho phép bạn render các View của mình bằng chính **true native API**, bằng cách gọi chính sdk theo từng nền tảng để build ứng dụng.

Trong một số tính năng đặc thù cần phải tương tác nhiều với phần cứng của thiết bị như: tính năng camera, tính năng Bluetooth… thì bạn cần phải sử dụng code native(Java hoặc Swift). React Native cho bạn code native ngay chính trong project và có thể tương tác với code Javascript bên ngoài.

* **Ưu điểm ,nhược điểm**
* **Ưu điểm:**
* Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng nhanh chóng.
* Hiệu năng tương đối ổn định.
* Cộng đồng phát triển mạnh.
* Tiết kiệm tiền.
* Team phát triển nhỏ.
* Ứng dụng tin cậy và ổn định.
* Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
* Trải nghiệm người dùng tốt hơn là hybrid app.
* **Nhược điểm:**
* Vẫn đòi hỏi native code.
* Hiệu năng sẽ thấp hơn với app thuần native code.
* Bảo mật không cao do dựa trên JS.
* Quản lý bộ nhớ.
* Khả năng tùy biến cũng không thực sự tốt đối với một vài module

1. **Quy trình cài đặt react native ( windows)**

***Bước 1: Cài đặt Node và Java Development Kit***

* Đầu tiên cần cài đặt Chocolatey bằng cách sử dụng các hướng dẫn trên trang web [Chocolatey](https://chocolatey.org/).

choco install -y nodejs.install python2 jdk8

***Bước 2: React Native CLI***

* Sử dụng [Node Package Manager](https://www.npmjs.com/) (hay viết tắt là npm) để cài đặt công cụ React Native Command Line Interface (CLI). Tại thiết bị đầu cuối (Terminal hoặc Command Prompt hoặc shell), bạn gõ lệnh:

npm install -g react-native-cli

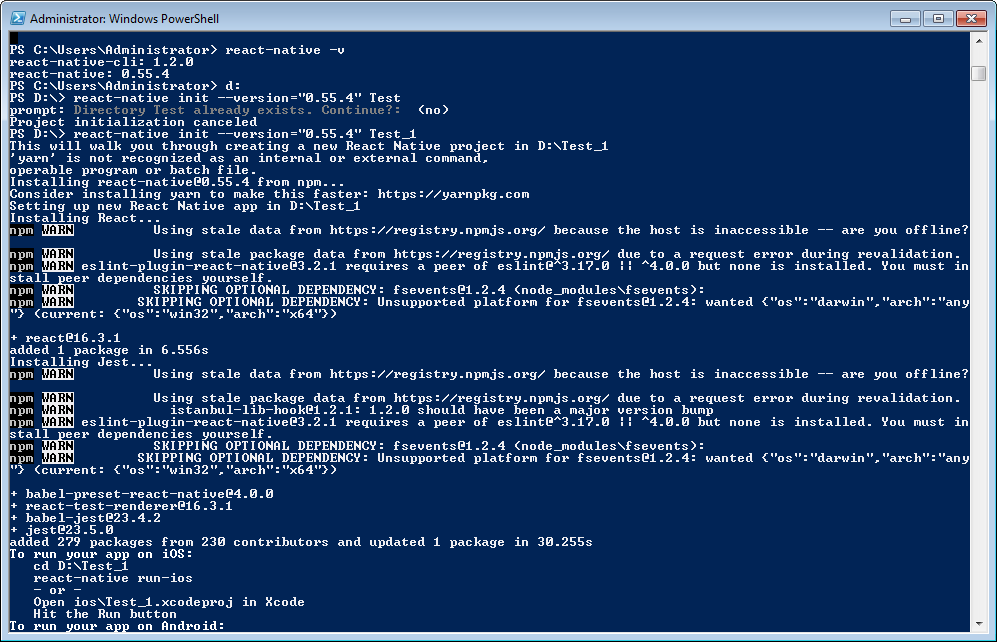
Với:

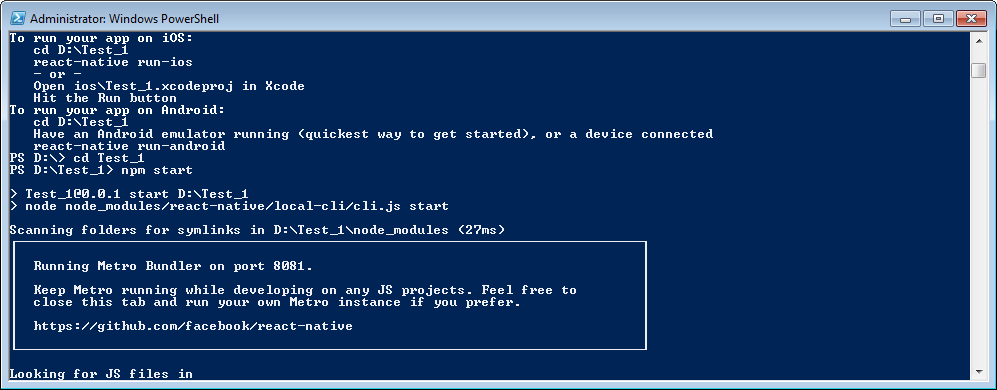
npm tìm nạp công cụ CLI và cài đặt nó. npm có chức năng tương tự như [JCenter](https://bintray.com/bintray/jcenter) và được đóng gói với Node.js.

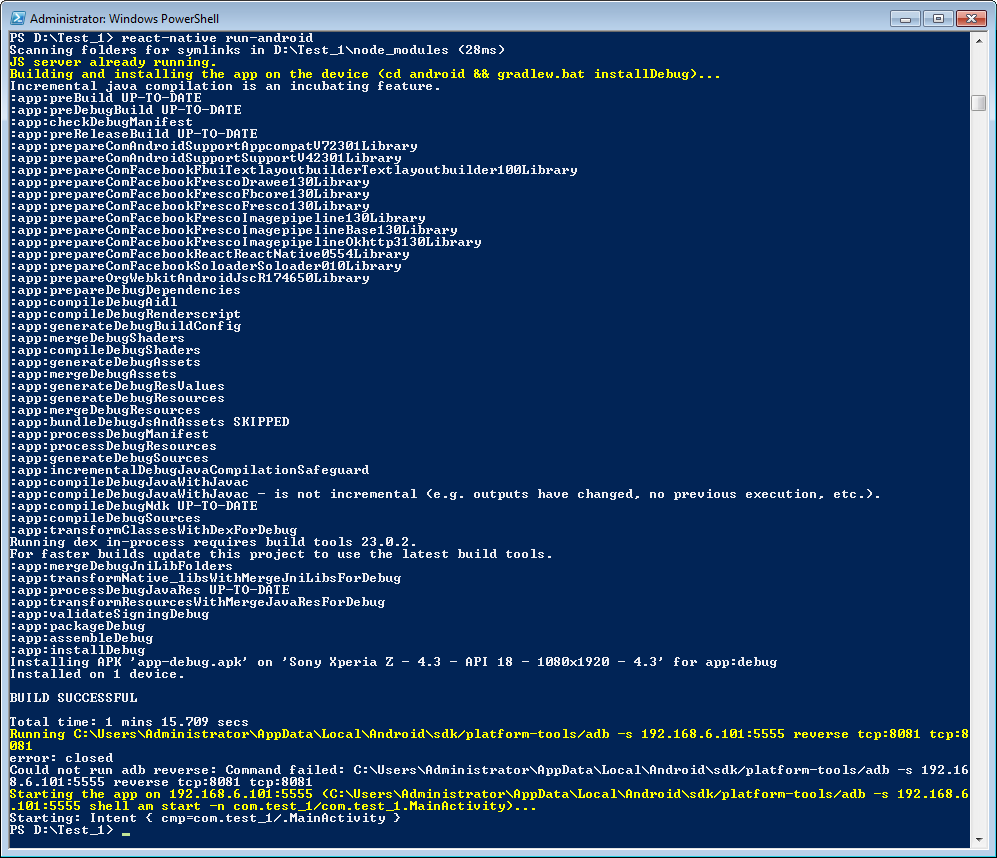
Sau đó, cài đặt Yarn bằng cách sử dụng các hướng dẫn trên trang web của [Yarn](https://yarnpkg.com/en/docs/install). Yarn là một client npm tốc độ cao.

***Bước 3: Chuẩn bị môi trường phát triển cho Android***

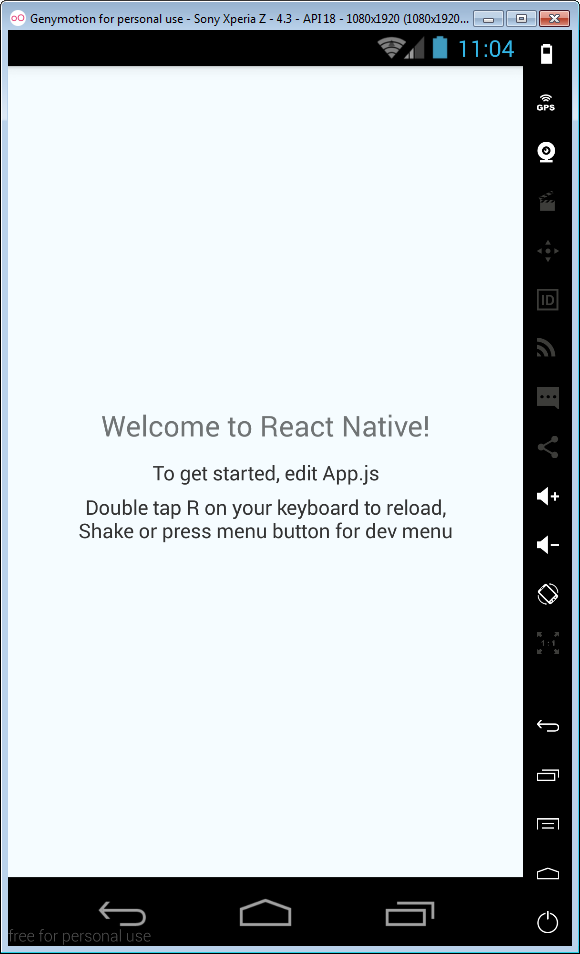
* Google APIs, Android 23
* Android SDK Platform 23
* Intel x86 Atom\_64 System Image
* Google APIs Intel x86 Atom\_64 System Image
  1. **Tìm hiểu một ứng dụng mẫu react native**
* Qúa trình cài đặt và run app



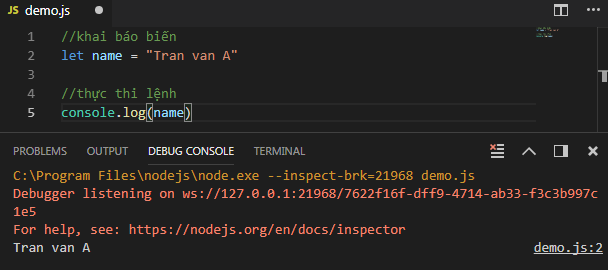




* ***Kết quả:***



## Kiến thức cơ bản về ES6 trong React native

* 1. **Tìm hiểu ES6**
  2. **Từ kháo Let, Var , Const**

**a/ Sử dụng từ khóa let**

**Cú pháp**: let var\_name = var\_value;

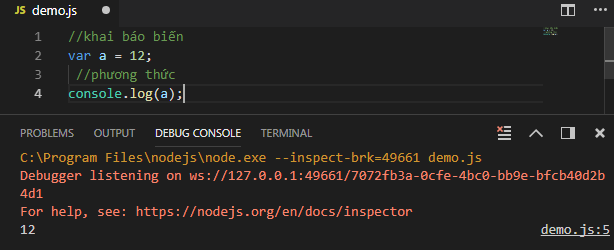
### **- Khi nào nên sử dụng biến Let**

### Với phạm vi hoạt động hẹp như vậy thì let thường dùng để khai báo các biến mang tính chất tạm thời, nghĩa là nó chỉ sống trong một phạm vi hoạt động của khối đó thôi, không sử dụng qua vị trí khác.

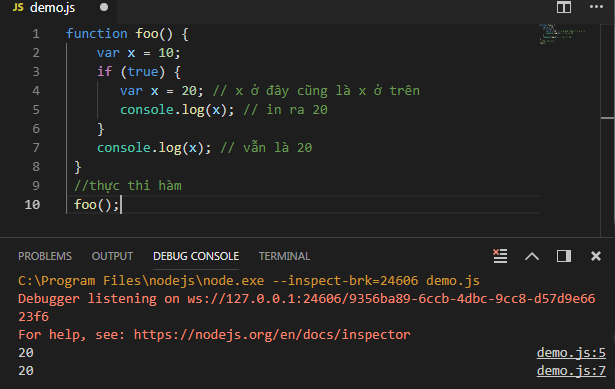
### **-Trường hợp sử dụng ngoài phạm vi của khối đã khai báo Let**

**b/ Sử dụng từ kháo Var**

Cú pháp:

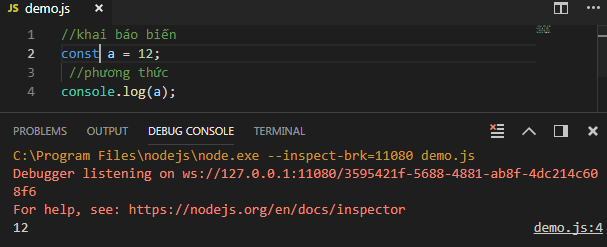


-Biến var có thể lưu dữ và thay đổi giá trị trong suốt quá trình

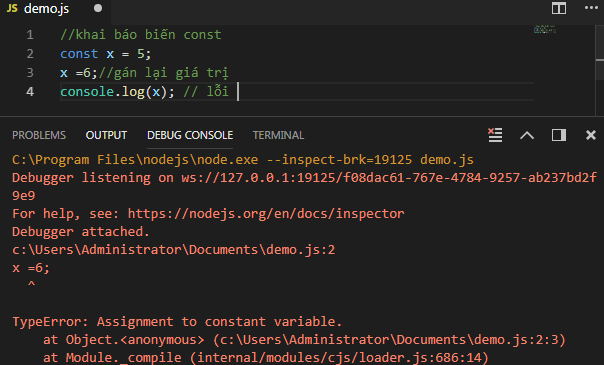


**c/ Sử dụng từ khóa Const**

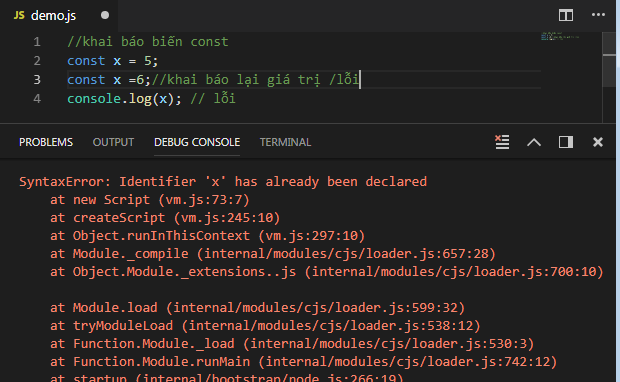
Cú pháp: Const const\_name = const\_value;



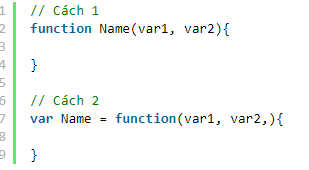
* **Biến const không thể gán lại giá trị**

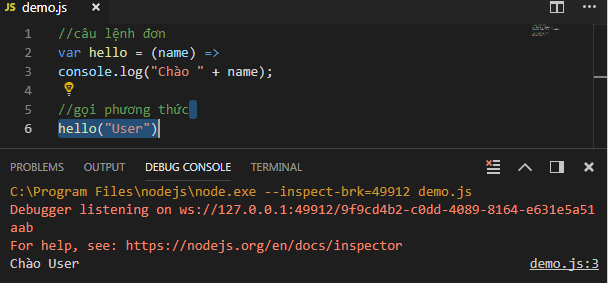


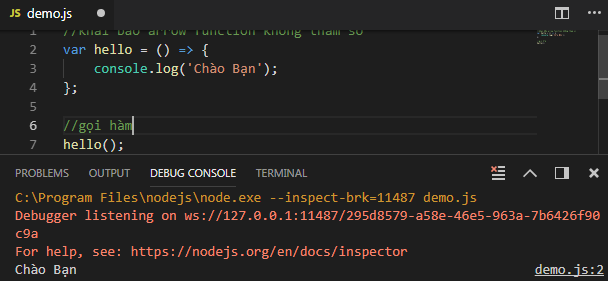
**-Biến const không thể được khai báo lại**



* 1. **Arrow function**

**-Cú pháp căn bản:**

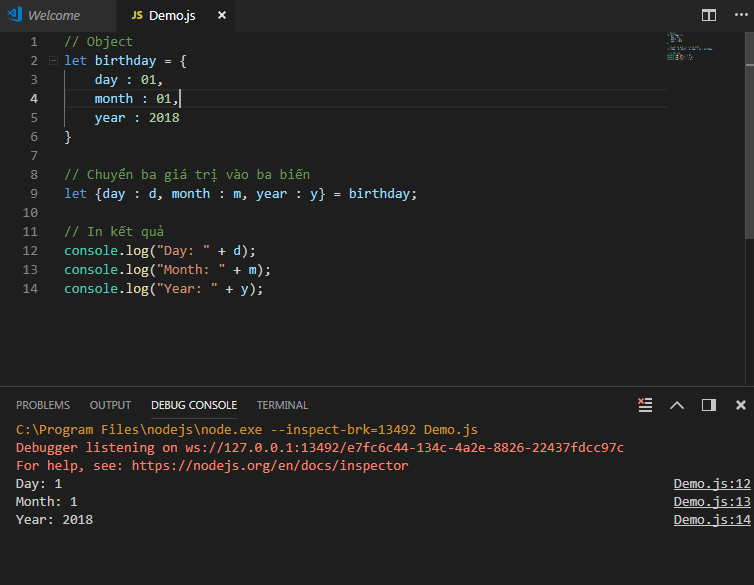
-Nội dung là 1 câu lệnh đơn

Trường hợp không có tham số

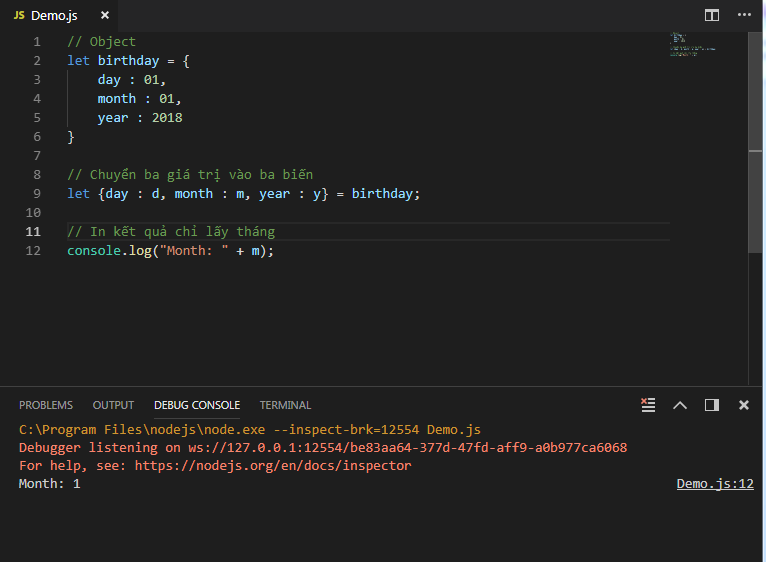
### Trường hợp arrow function có tham số

### 

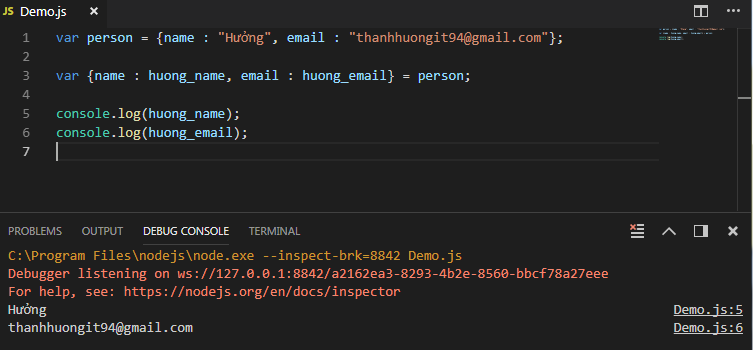
* 1. **Kiểu dữ liệu Object**
* Object



* Trường hợp khác thì bạn có thể lấy một phần tử bất kì chứ không bắt buộc theo thứ tự như bên mảng như ví dụ sau đây:



* **Lấy giá trị theo key Object**: Lấy giá trị dựa vào tên key của Object.



* 1. **Khai báo class trong ES6**

Trong ES6 đã hỗ trợ chúng ta khai báo một đối tượng theo chuẩn OOP, bằng cách sử dụng từ khóa class.

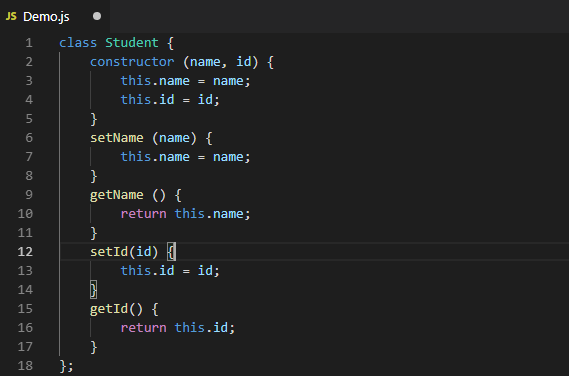
Để khởi tạo đối tượng được khai báo theo chuẩn ES6 thì các bạn sử dụng từ khóa newvới cú pháp như sau:

new ClassName; hoặc new ClassName();

Và với ES6, nó cũng hỗ trợ chúng ta một phương thức đặc biệt mà bất kỳ ngôn ngữ lập trình nào cũng có đối với class đó là constructor - phương thức khởi tạo. constructor trong ES6 cũng có tác dụng tương tự, nó sẽ tự động được gọi khi đối tượng được khởi tạo.

Để khai báo constructor trong ES6 thì các bạn chỉ cần khai báo một phương thức có tên là constuctor.

Khai báo constuctor cho đối tượng Student:



Với kiểu khai báo class trong ES6 chúng ta không thể khai báo trực tiếp thuộc tính cho nó được mà phải khởi tạo qua các phương thức và thường thì chúng ta sẽ đặt nó ở trong constructor luôn.

Lúc này khi khởi tạo đối tượng chúng ta có thể truyền luôn tham số cho nó như các ngôn ngữ khác.



Với ES6, nó cũng đã cung cấp cho chúng ta sử dụng từ khóa extends để kế thừa từ đối tượng khác.

Cú pháp:

class A extends B {

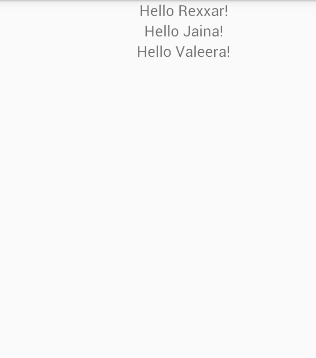
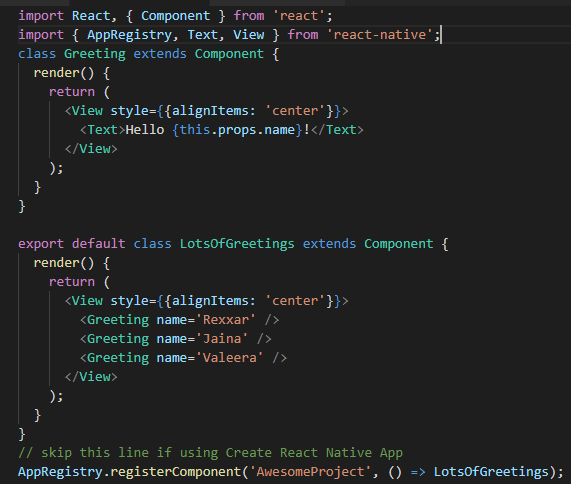
//code

}

## 3. Component trong react native

* 1. Tìm hiểu Props và State và Style

1. Props

* 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.
* Bạn có thể tạo ra component sử dụng props. Ý tưởng của props đó là việc trừu tượng hoá các component để có thể sử dụng được ở nhiều chỗ khác nhau trong app. Hãy xem một ví dụ đơn giản sau:
* Kết quả:

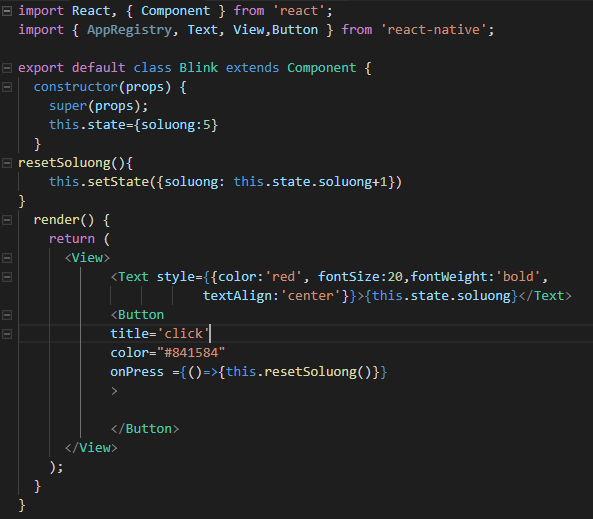
1. State

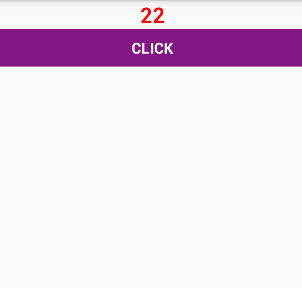
state thì hoạt động khác với props. state là dữ liệu nội bộ của một Component, trong khi propslà dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu **mutable**.

Tuy vậy, hãy nhớ rằng đừng bao giờ thay đổi trực tiếp biến this.state. Thay vào đó hãy dùng hàm setState để cập nhật giá trị. Sở dĩ chúng ta cần dùng hàm này là do nó sẽ kích hoạt việc render lại component và tất cả component con nằm trong nó, còn thay đổi this.state thì không. Còn một vấn đề nữa, đó là setState chạy bất đồng bộ, vậy nên nếu bạn tiến hành đọc ra giá trị state ngay sau khi setState thì chưa chắc giá trị sẽ được update lên mới nhất đâu.

Vậy thì khi nào nên sử dụng state? Bất cứ khi nào dữ liệu trong một component có thể thay đổi (bởi chính nó).

Người dùng tương tác với component là một ví dụ về cách mà state hoạt động. Ấn vào một button, checkbox, thêm text vào form,... là những trường hợp mà chúng ta có thể sử dụng state.Ví dụ :Khi bạn click vào button thì giá trị trên text xe thay đổi và sẽ tang lên 1



Kết quả

* 1. Tim hiểu Width ,Height,Flex

1. **Height & Width**

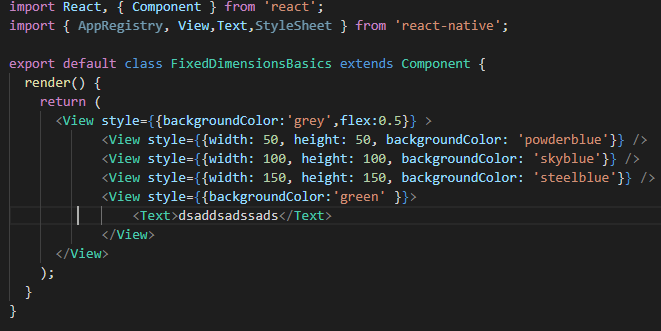
Chiều rộng và dài của các thành phần sẽ xác định kích thước của các thành phần giao diện trên màn hình.

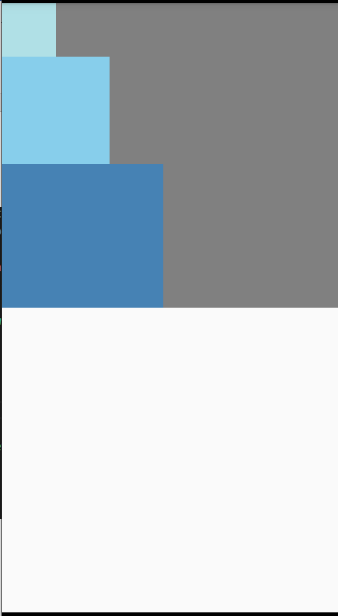
* **Kích thước cố định**

Cách đơn giản nhất để chỉ định kích thước của các thành phần là sử dụng các tham số kích thước cố định width và height trong style. Tất cả kích thước được sử dụng trong React Native đều tuân theo đơn vị pixel khi hiển thị lên trên màn hình.

* Kích thước linh động

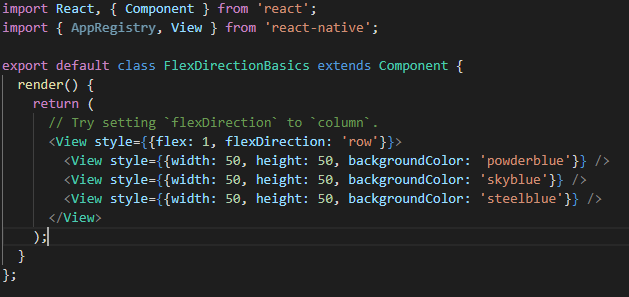
Sử dụng flex trong style của các thành phần để thanh đổi vùng hiển thị một cách linh động dựa trên vùng hiển thị có thể. Thông thường, bạn sẽ sử dụng flex: 1 khi mà bạn muốn một thành phần trải rộng hết mức có thể so với các thành phần khác có chung một view cha. Đây là giá trị tỷ lệ cao nhất mà flex cung cấp. Với các tỷ lệ khác thì nó sẽ so sánh với các thành phần ngang hàng với nó để đưa ra tỷ lệ hiển thị trong cùng một view cha.

* ****Một thành phần chỉ có thể mở rộng ra hết khoảng trống có thể nếu như view cha của nó có kích thước lớn hơn 0. Nếu như view cha không được gán các kích thước cố định như width và height hoặc flex, thì kích thước của view cha sẽ có kích thước là 0 và khi sử dụng flex tại view con thì nó cũng sẽ không được hiển thị.

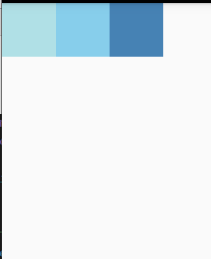
Kết quả:

* 1. Tìm hiểu Flexbox

Code :



Kết quả:



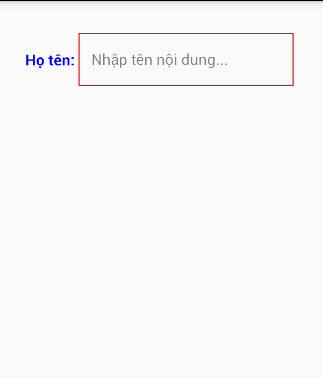
* 1. Tìm hiểu về Text,View ,TextInput

1. TextInput

TextInput là một thành phần cơ bản cho phép người dùng có thể nhập text. Nó có một tham số onChangeText dành cho việc cài đặt một function được gọi mỗi khi nội dung text trong khung nhập thay đổi

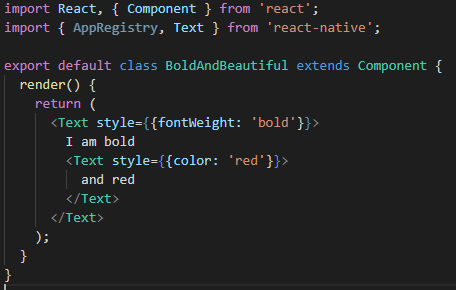
Code :

1. Kết quả:

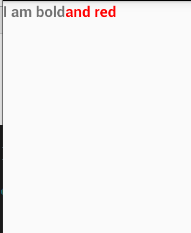


1. Text

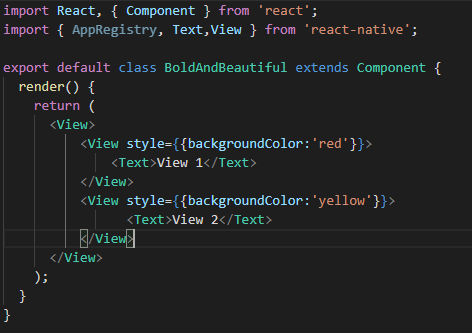
Code :

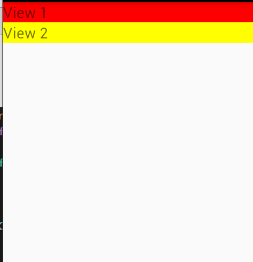


Kế quả:

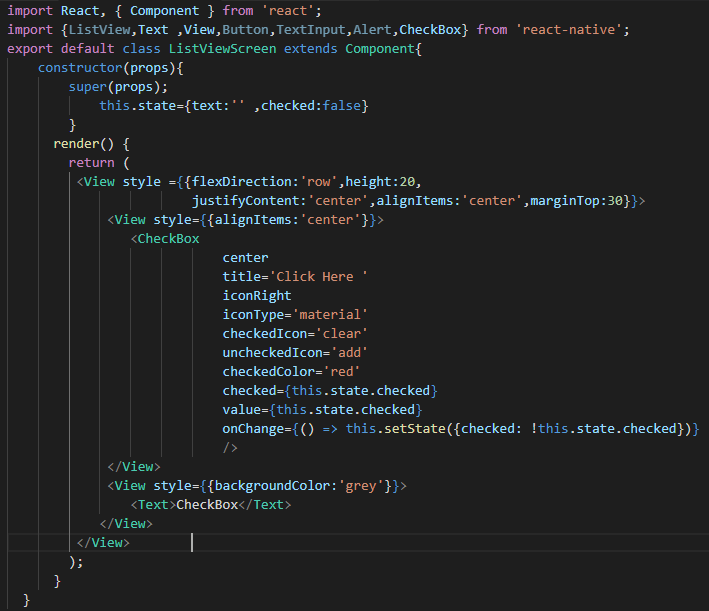


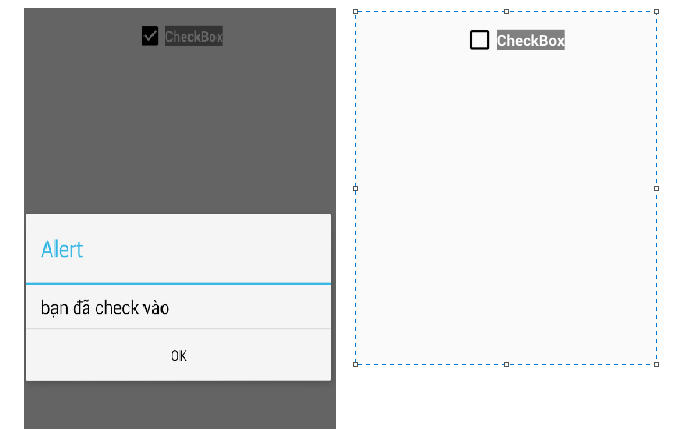
1. View

Code:

Kết quả:

* 1. CheckBox

Ví dụ: Tạo chương trình demo Checkbox, tạo sự kiện khi check vào sẽ hiển thị thông báo

Kết quả:

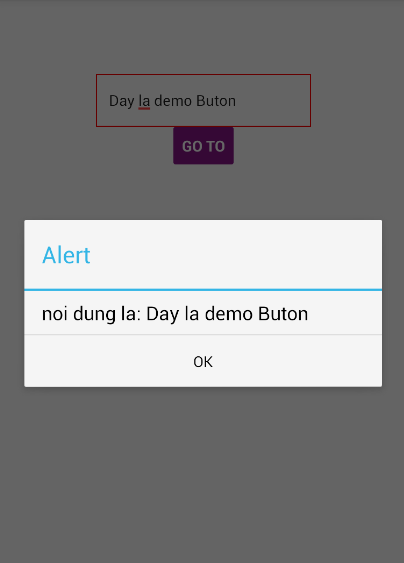
* 1. RadioButton
  2. Button

Trong React Native, ta có thể sử dụng component Button hoặc **TouchableOpacity** để hiển thị dạng nút bấm (button). Thêm vào đó ta sẽ sử dụng thuộc tính ***onPress*** để định nghĩa các hành động khi người dùng bấm vàoButton hoặc **TouchableOpacity** đó.

VD:Nhập nội dung trong ô textInput, sau khi ấn Go to -> thông báo hiển thị nội dung nhập trong ô TextInput.

Code:

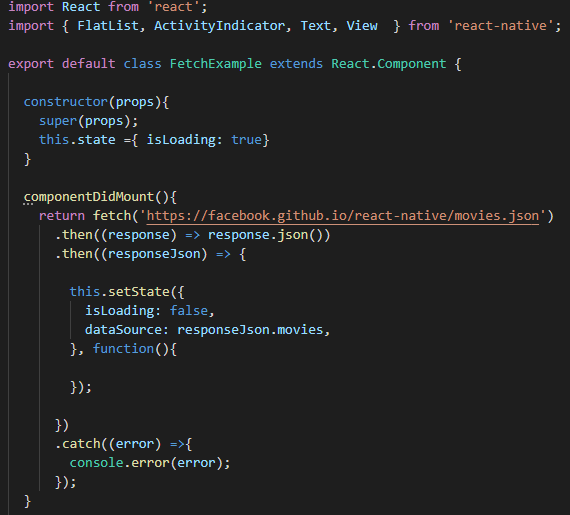
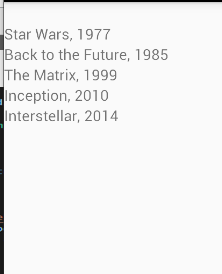
KẾt Quả:



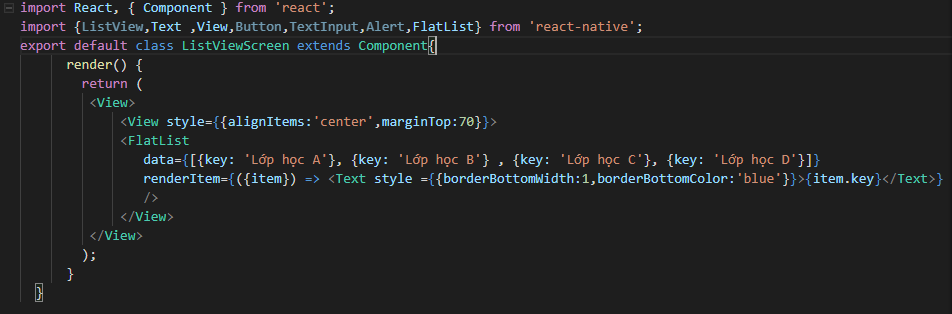
* 1. NetWorking

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.

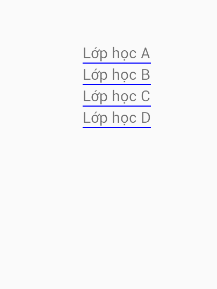
Networking bản chất là một hình thức bất đồng bộ (Lan man một chút, vì sự bất đồng bộ này mà ở Android từ API 11 trở lên hệ điều hành đã ngăn cản việc chạy Network trên main thread để ngăn cản độ trễ của chương trình trong thời gian chờ dữ liệu mạng được trả về dưới client).

Kết quả:

* 1. Flatlit

Code:

Kết quả:



## 4. API Trong React Native

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Phân tích hệ thống

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

#### 2.1.1.1 Giao diện Đăng Nhập



*Image1: Registration*

#### 2.1.1.2. Các yêu cầu Chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **Màn hình Log in** | Màn hình Login Gôm 3 Button và 2 InputText và 1 Image:   * Button:   + Đăng nhập  +Hủy   * TextInput :   +Nhập tài khoản  +Nhập mật khẩu   * Image:   + Hiển thị Logo nhóm | N/A | N/A |
| **<Button>**  **Đăng nhập** | Sẽ được dùng để Đăng nhập vào Hệ thống | Click trái vào nút Đăng nhập | Sau khi click trái vào , hệ thông sẽ kiểm tra 2 trường “Tài khoản” và “Password” , nếu hợp lệ sẽ chuyển sang giao diện Danh sách xe máy , nếu sai sẽ tiếp tục đăng nhập hoặc thoát ứng dụng |
| **<Button>**  **Hủy** | Sẽ được dùng để Thoát khỏi Ứng dụng | Click trái vào | Khi người dung click vào sẽ xác nhận có chắc chắn thoát, nếu ok sẽ thoát ứng dụng |
| **<Button>**  **Đăng ký** | Khi người dùng clịck vào sẽ chuyển sang giao diện màn hình “Đăng ký” | Click trái vào | Khi click trái vào sẽ chuyển sang giao diện đăng ký |
| **<TextInput>**  **Tài khoản** | Sử dụng để người dùng nhập thông tin User/Email để đăng nhập | Click trái vào | Hiển thị nội dung nhâp |
| **<TextInput>**  **Mật khẩu** | Sử dụng để người dùng nhập thông tin Mật khẩu để đăng nhập | Click trái vào | Hiển thị nội dung nhâp |
| **<Image>**  **Logo** | Chứa logo của nhóm | N/A | N/A |

#### 2.1.2.1 Giao diện Đăng ký Tài Khoản

#### 2.1.2.2 Yêu cầu Chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **tem** | **Description** | **Action** | **Response** |
| **Màn hình Đăng ký** | Màn hình Login Gôm 3 button và 3 InputText:   * Button:   + Đăng ký  + Hủy  + Trở lại   * TextInput   + Nhập email..  + Mật khẩu đăng ký  + Xác nhận mật khẩu | N/A | N/A |
| **<Button>**  **Đăng Ký** | Sẽ sử đụng để Thực hiện Đăng ký tài khoản | Cl ick trái vào | Nếu dữ liệu hợp lệ -> thông báo thành công->chuyển về màn hình Đâng nhập  -Nếu không->thông báo lỗi |
| **<Button>**  **Hủy** | Sẽ dử dụng để Thoát và quay trở lại màn hình “Đăng nhập” | Click trái vào | Quay về màn hình đăng nhâp |
| **<TextInput>**  **Nhập tài khoản** | Sử dụng để nhập Tài khoản đăng ký | Click trái vào | Hiển thị nội dung |
| **<TextInput>**  **Nhập Mật khẩu** | Sử dụng để nhập mật kkhẩu | Click trái vào | Hiển thị nội dung |
| **<TextInput>**  **Nhập Xác nhận mật khẩu** | Sử dụng để nhập mật khẩu xác nhận | Click trái vào | Hiển thị nội dung |

#### 2.1.3.1 Giao diện Danh sách Xe máy

#### 2.1.3.2 Yêu cầu Chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **Màn hình “Danh sách xe** | Màn hình Login Gôm 4 button và 1 InputText:   * Button:   + Thêm  +Sửa  +Xóa   * TextInput   + Tìm kiểm Xe | N/A | N/A |
| **<Button>**  **Thêm** | Sử dụng để Thêm 1 xe mới vào danh sách | Click trái vào | - Sẽ chuyển đến Màn hình “Thêm Xe mới” |
| **<Button>**  **Xóa** | Sử dụng để xóa 1 xe trong danh sách | Click trái vào | -Hiển thị thông báo xác nhận -> Click đồng ý ->Sẽ Xóa 1 Xe đã chọn trong danh sách và database |
| **<Button>**  **Sửa** | Sử dụng để sửa thông tin 1 xe đã chọn trong danh sách | Click trái vào | -Sẽ chuyển đến màn hinh “ Sửa Thông tin xe” |
| **<Button>**  **Tìm kiếm** | Tìm kiếm thông tin 1 xe máy | Click trái vào ->Nhập thông tin cần tìm | -Nếu có dữ liệu -> hiển thị kết quả |
| **<TextInput>**  **Nhập nội dung tìm kiếm** | Sẽ sử dụng để nhập thông tin cần tìm kiếm | Click trái và nhập thông tin | Hiển thị nội dung nhập |

#### 2.1.4.1 Giao diện Thêm Xe mới

#### 2.1.4.2 Yêu cầu Chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **Màn hình “Thêm xe mới”** | Màn hình gồm 3 Button và 5 TextInput và 1 Image   * <Button>   + Lưu  + Hủy  + Choice image   * <TextInput>   + Nhập tên xe  +Nhập biển số  + Nhập màu sơn  + Nhập dung tích Xilanh   * <Picker>   + Chứa danh mục Loại xe | N/A | N/A |
| **<Button>**  **Lưu** | Sử dụng để thực hiện Thêm xe mới vào danh sách | Click trái vào | Thêm 1 Xe vào danh sách -> quay về màn hình “Danh sách xe” |
| **<Button>**  **Hủy** | Sử dụng để thoát và quay trở lại giao diện trước đó | Click trái vào | Thoát ->màn hình “Danh sách xe” |
| **<Picker>** | Sử dụng để Chọn loại xe:  +Xe tay ga  +Xe so | Click trái vào ->Chọn | Hiển thị mục đã chọn |
| **<TextInput>**  **Nhập tên xe** | Sử dụng để nhập Tên xe | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |
| **<TextInput>**  **Nhập biển số** | Sử dụng để nhập biển số | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |
| **<TextInput>**  **Nhập màu xe** | Sử dụng để nhập màu xe | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |
| **<TextInput>**  **Nhập Phân khối** | Sử dụng để nhập phân khối xe | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |

#### 2.1.5.1 Giao diện Sửa thông tin Xe



#### 2.1.5.2 Yêu cầu Chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **Màn hình “Thêm xe mới”** | Màn hình gồm 3 Button và 5 TextInput và 1 Image   * <Button>   + Lưu  + Hủy  + Choice image   * <TextInput>   + Nhập tên xe  +Nhập biển số  + Nhập màu sơn  + Nhập dung tích Xilanh   * <Image>   + Chứa hình ảnh đại diện xe   * <Picker>   + Chứa danh mục Loại xe | N/A | N/A |
| **<Button>**  **Lưu** | Sử dụng để thực hiện Lưu/cập nhật thông tin xe đã chọn để sửa vào danh sách | Click trái vào | Lưu và cập nhật lại thông tin xe đã chọn và danh sách-> quay về màn hình “Danh sách xe” |
| **<Button>**  **Hủy** | Sử dụng để thoát và quay trở lại giao diện trước đó | Click trái vào | Thoát ->màn hình “Danh sách xe” |
| **<Picker>** | Sử dụng để Chọn loại xe:  +Xe tay ga  +Xe so | Click trái vào ->Chọn | Hiển thị mục đã chọn |
| **<TextInput>**  **Nhập tên xe** | Sử dụng để nhập Tên xe | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |
| **<TextInput>**  **Nhập biển số** | Sử dụng để nhập biển số | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |
| **<TextInput>**  **Nhập màu xe** | Sử dụng để nhập màu xe | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |
| **<TextInput>**  **Nhập Phân khối** | Sử dụng để nhập phân khối xe | Click trái vào->Nhập dữ liệu | Hiẻn thị dữ liệu đã nhập |

## 2.2 Thiết kế hệ thống

### 2.2.1. Màn hình Đăng Nhập

#### 2.2.1.1. Màn hình Đăng Nhập

#### 2.2.1.2. Các đối tượng và hành động cho màn hình Đăng nhập

Các nút:

* <Button> Đăng nhập
* <Button> Hủy
* <Button> Đăng ký
* <TextInput> Nhập email
* <TextInput> Nhập Mật khẩu

Hành động:

* Kiểm tra tài khoản và dữ liệu nhập -> Thông báo kết quả -> Màn hình “Danh sách xe máy”
* Thoát ứng dụng
* Chuyển đến Giao diện Đăng ký tài khoản
* Click vào ->Nhập tài khoản
* Click vào -> Nhập mật khẩu

### 2.2.2. Màn hình Đăng Ký

#### 2.2.2.1. Màn hình Đăng ký

#### 2.2 .2.2. Các đối tượng và hành động cho màn hình Đăng ký

Các nút:

* <Button> Đăng Ký
* <Button> Hủy
* <TextInput> Nhập email
* <TextInput> Nhập mật khẩu mới
* <TextInput> Nhập mật khẩu xác nhận

Hành động:

* Kiểm tra dữ liệu -> kiểm tra tài khoản đã tồn tại chưa-> Thông báo kết quả -> Màn hình “Đăng nhập” (Nếu hợp lệ) ngược lại tiếp tục ở lại đăng ký
* Click trái ->Quay lại màn hình Đăng nhập
* Click trái vào -> Nhập Email
* Click trái vào ->Nhập Mật khẩu mới
* Click trái -> Nhâp mật khẩu xác nhận

### 2.2.3. Màn hình Danh sách Xe Máy

#### 2.2.3.1. Màn hình Danh sách Xe Máy

#### 2.2.3.2. Các đối tượng và hành động cho màn hình Danh sách Xe Máy

Các nút:

*  (Thêm xe) -> Màn hình “Thêm Xe mới”
*  (Xóa ) -> xóa 1 xe trong danh sách và xóa trong database
*  (Sửa ) -> Màn hình “Thêm Xe mới” -> thiết lập thông tin của xe vào giao diện
*  (Tìm kiếm) -> Hiển thị danh sách kết quả tìm kiếm lên giao diện hiện tại
* <TextInput> Nhập thông tin cần tìm kiếm

Hành động:

* Click trái vào -> chuyển màn hình “Thêm xe mới”
* Click trái ->Thông báo xác nhận ->Xóa 1 Xe trong Danh sách hiển thị và database
* Chuyển sang Màn hình”Sửa thông tin xe” ->Hiển thị thông tin xe được chọn tại màn hình Sửa thông tin xe
* Click trái -> kiểm tra dữ liệu nhập ->Tìm kiếm thông tin xe theo tên xe và hiển thị trên bảng danh sách hiện tại

### 2.2.4. Màn hình Thêm Xe mới

#### 2.2.4.1. Màn hình Thêm Xe mới

#### 2.2.4.2. Các đối tượng và hành động cho màn hình Thêm Xe mới

Các nút:

* Lưu -> Thêm xe mới vào database
* Hủy -> Màn hình “Danh sách xe” trước đó
* <Picker> Loại xe
* <TextInput> Nhập tên xe
* <TextInput> Nhập Biển số
* <TextInput> Nhập màu xe
* <TextInput> Nhập phân khối

Hành động:

* Click vào -> Thêm thông tin Xe mới vào danh sách và database -> chuyển vè màn hình “Danh sách xe”
* Click trái vào ->Quay lại màn hình “Danh sách xe”
* Click trái vào ->Nhập Tên xe
* Click trái vào ->Nhập biển số
* Click trái vào ->Nhập Màu xe
* Click trái vào ->Nhập Phân khối

### Màn hình Sửa Thông tin Xe

#### 2.2.5.1. Màn hình Sửa Thông tin Xe

#### 2.2.5.2. Các đối tượng và hành động cho màn hình Sửa Thông tin Xe

Các nút:

* <Button>Lưu
* <Button>Hủy
* <Picker> Loại xe
* <TextInput> Nhập Tên xe
* <TextInput> Nhập Màu xe
* <TextInput> Nhập biển số
* <TextInput> Nhập phân khối

Hành động:

* Click trái ->Kiểm tra dữ liệu ->Lưu và thay đổi thông tin Xe được chọn trong danh sách và database
* Click trái 🡪Thoát vá quay lại màn hình “Danh sách xe”
* Hiển thị Danh sách Loại xe để người dùng chọn cố định giá trị->click chọn 1 loại
* Click Nhập -> Nhập thông tin Tên xe
* Click trái vào -> Nhập Màu xe
* Click trái vào -> Nhập Biển số xe
* Click trái vào -> Nhập phân khối xe

# 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