TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

KHOA CÔNG NGHỆ THÔNG TIN

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

BÁO CÁO KẾT THÚC MÔN HỌC

Lập trình di động 3

**ỨNG DỤNG LƯU TRỮ CÁ NHÂN**

Giảng viên hướng dẫn: TRƯƠNG BÁ THÁI

Sinh viên thực hiện: LƯƠNG VĂN THUỘC

Ngành: Công nghệ thông tin Lớp: CD16TT3 Khoá: 16

*Tp. Hồ Chí Minh*, ngày 28 tháng 12 năm 2018

**NHẬT KÝ HOẠT ĐỘNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Họ và tên** | **Công việc đã thực hiện** | **Tự đánh giá** | **Nhóm đánh giá** | **Chữ ký** |
| 1 | Lương Văn Thuộc | 1. Chạy và viết báo cáo quá trình học  2. Thiết kế giao diện  3. Tìm hiểu, cài đặt và thiết kế navigation  4. Tìm hiểu về Database  5. Xây dựng Database truy xuất và chỉnh sửa dữ liệu giữa ứng dụng với Database  6. Viết báo cáo sản phẩm cuối cùng  7. Áp dụng, test và chỉnh sửa các chức năng vào đồ án  7. Áp dụng và cài đặt sản phẩm cuối cùng | 5 |  |  |

MỤC LỤC

GIAO DIỆN ỨNG DỤNG 4

CHƯƠNG 1. TỔNG QUAN VỀ REACT NATIVE 5

1. Giới thiệu tổng quan về React native 5

Tổng quan về React Native: 5

2. Kiến thức cơ bản về ES6 trong React native 7

Arrow function trong ES6: 8

Cú pháp căn bản nhất của arrow function như sau: 8

3. Component trong react native 10

Text component: 10

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 15

2.1 Phân tích hệ thống 15

2.1.1. Feature/Component #1: LOGIN screen 15

2.1.1.1 User Interfaces 15

2.1.1.2. Yêu cầu chức năng 16

2.1.2. Feature/Component #1: MENU screen 18

2.1.2.1 User Interfaces 18

2.1.2.2. Yêu cầu chức năng: 19

2.1.3.1 User Interfaces 21

2.1.3.2. Yêu cầu chức năng: 22

2.1.3. Feature/Component #1: THÊM GHI CHÚ screen 24

2.1.3.1 User Interfaces 24

2.1.3.2. Yêu cầu chức năng: 25

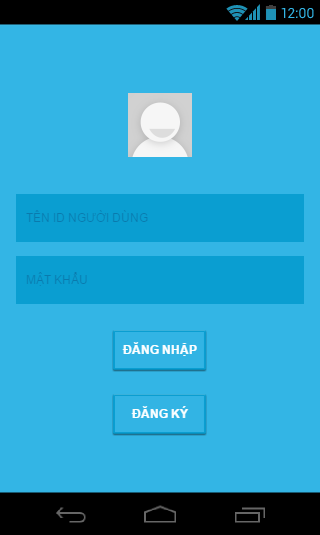
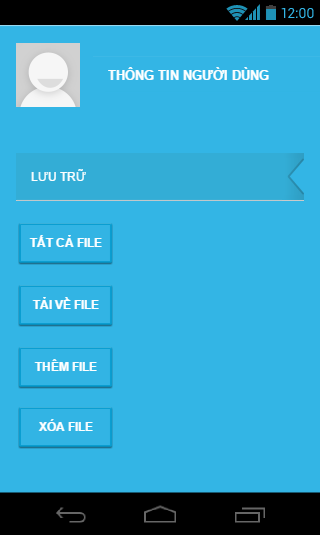
2.1.4. Feature/Component #1: XEM CẬP NHẬT screen 27

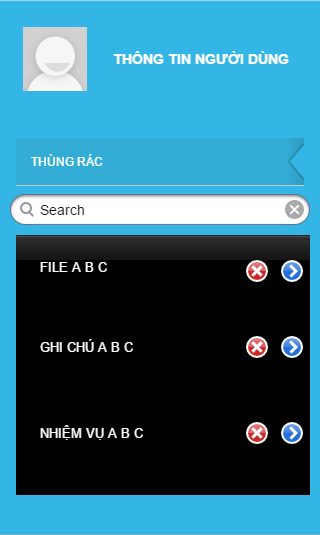
2.1.4.1 User Interfaces 27

2.1.4.2. Yêu cầu chức năng: 28

TÀI LIỆU THAM KHẢO 29

# GIAO DIỆN ỨNG DỤNG

# CHƯƠNG 1. TỔNG QUAN VỀ REACT NATIVE

## 1. Giới thiệu tổng quan về React native

# Tổng quan về React Native:

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…

**Ưu điểm:**

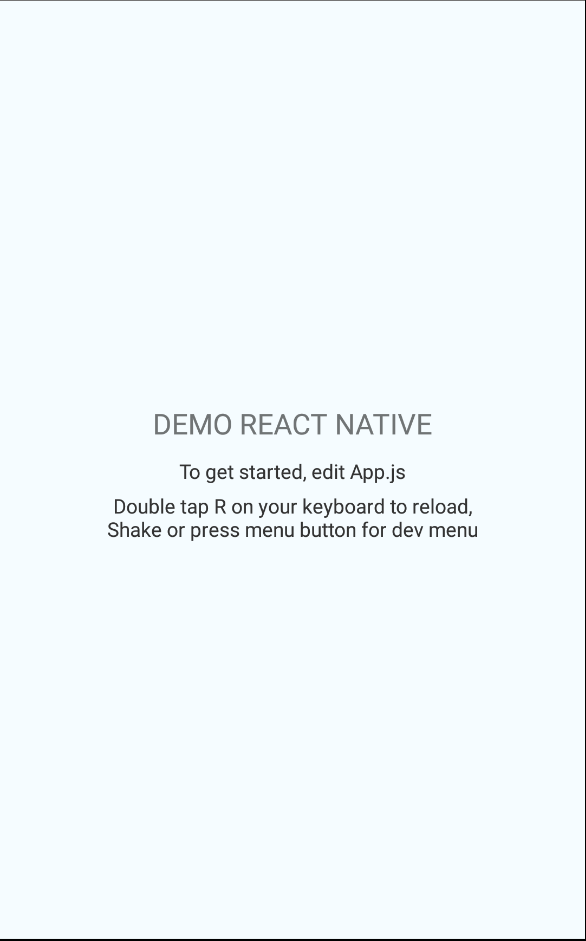
* Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng nhanh chóng.
* Hiệu năng tương đối ổn định.
* Cộng đồng phát triển mạnh.
* Tiết kiệm tiền.
* Team phát triển nhỏ.
* Ứng dụng tin cậy và ổn định.
* Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
* Trải nghiệm người 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.

**Cài đặt và khởi chạy thuận tiện:**

Cách cài đặt đã được nhà cung cấp hướng dẫn rất chi tiết ở link chủ: https://facebook.github.io/react-native/docs/getting-started



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

**Khái niệm ES6:**

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 do hiệp hội các nhà sản xuất máy tính Châu Âu đề xuất làm tiêu chuẩn của ngôn ngữ Javascript. Bạn cứ nghĩ xem hiện nay có khá nhiều trình duyệt Browser ra đời và nếu mỗi Browser lại có cách chạy Javascript khác nhau thì các trang web không thể hoạt động trên tất cả các trình duyệt đó được, vì vậy cần có một chuẩn chung để bắt buộc các browser phải phát triển dựa theo chuẩn đó.

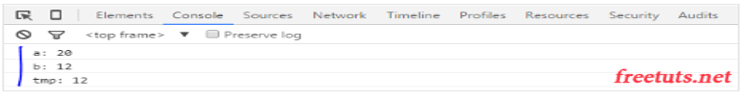
**Các nội dung được tìm hiểu trong ES6:**

**Biến Var, Let trong cs6:**

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.



Sử dụng firebug sẽ thấy kết quả như sau:

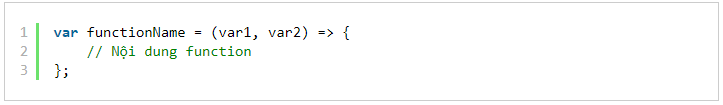


Kết luận:

Rõ ràng việc sử dụng từ khóa let có những ưu điểm và nhược điểm khác nhau, vì vậy không phải thấy nó mới được bổ sung vào là sử dụng đâu nhé :). Ví dụ nếu bạn sử dụng một biến cho toàn ứng dụng và biến đó thay đổi liên tục thì bắt buộc phải sử dụng var để khai báo.

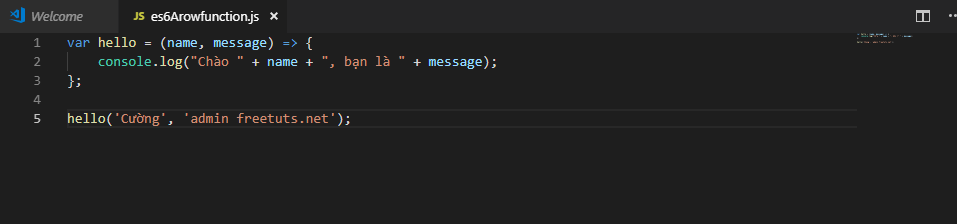
# Arrow function trong ES6:

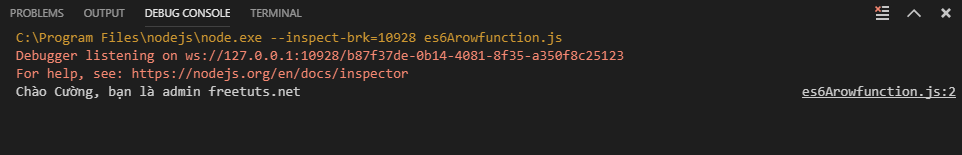
# Cú pháp căn bản nhất của arrow function như sau:



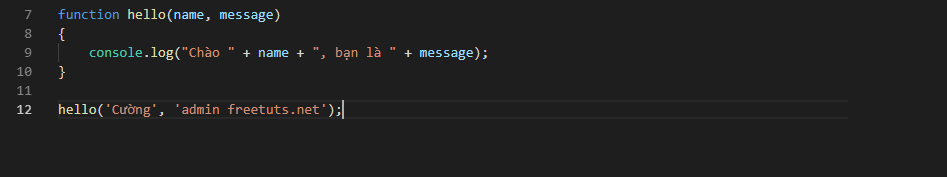
**Ví dụ**: Viết arrow function in ra câu chào và so sánh với cách tạo function thông thường.

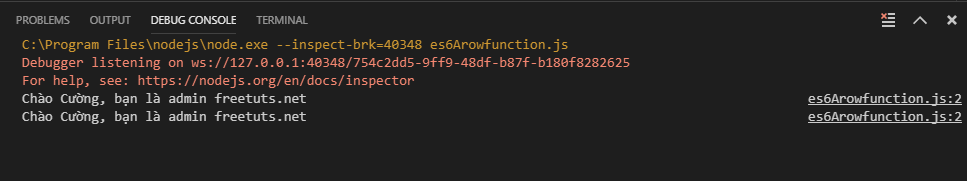
ARROW FUNCTION





NORMAL FUNCTION



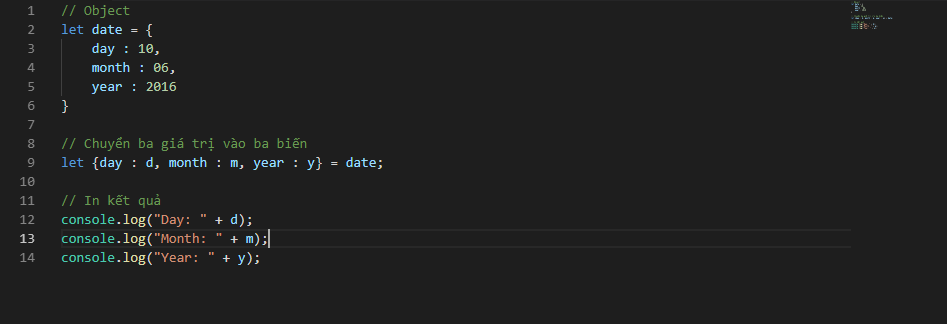


Kết luận:

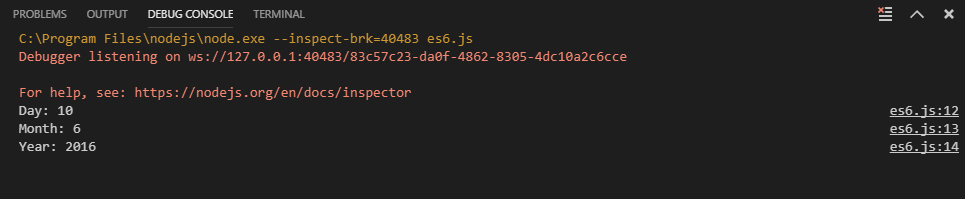
Nói là ES6 nhưng thực chất hầu hết các trình duyệt hiện nay đều đã hỗ trợ ES6 nên bạn có thẻ coi như đây là một tính năng mới của Javascript.

KIỂU DỮ LIỆU OBJECT TRONG ES6:

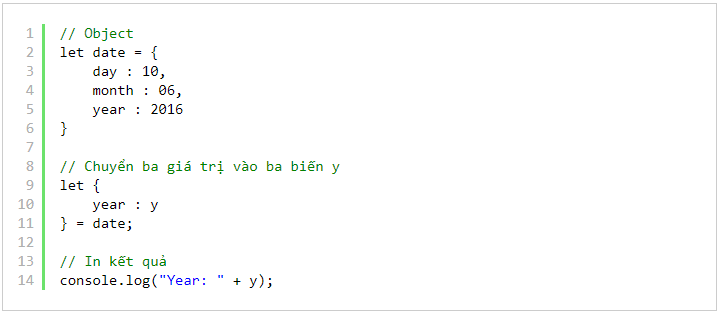
Ngoài mảng ra thì bạn có thể tách biến từ object:



Kết quả:

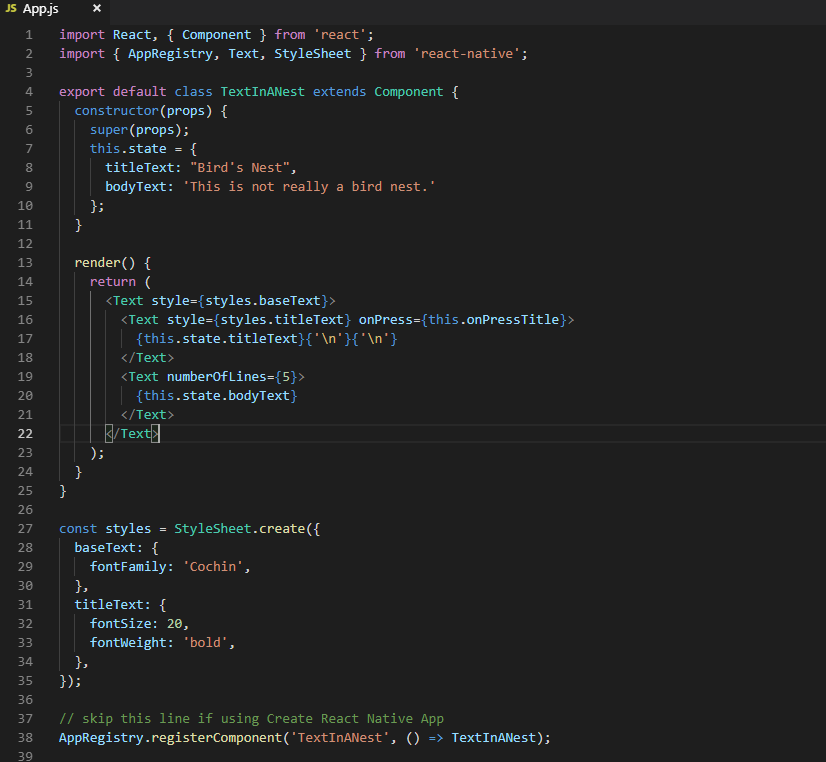


Với trường hợp này thì bạn có thể lấy một phần tử bất kì chứ không bắt buộc theo thứ tự như bên mảng:

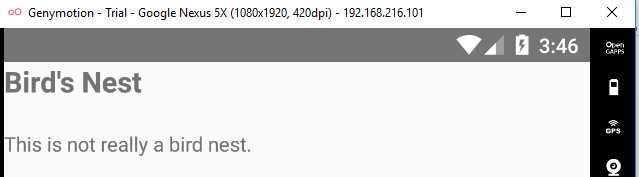


## 3. Component trong react native

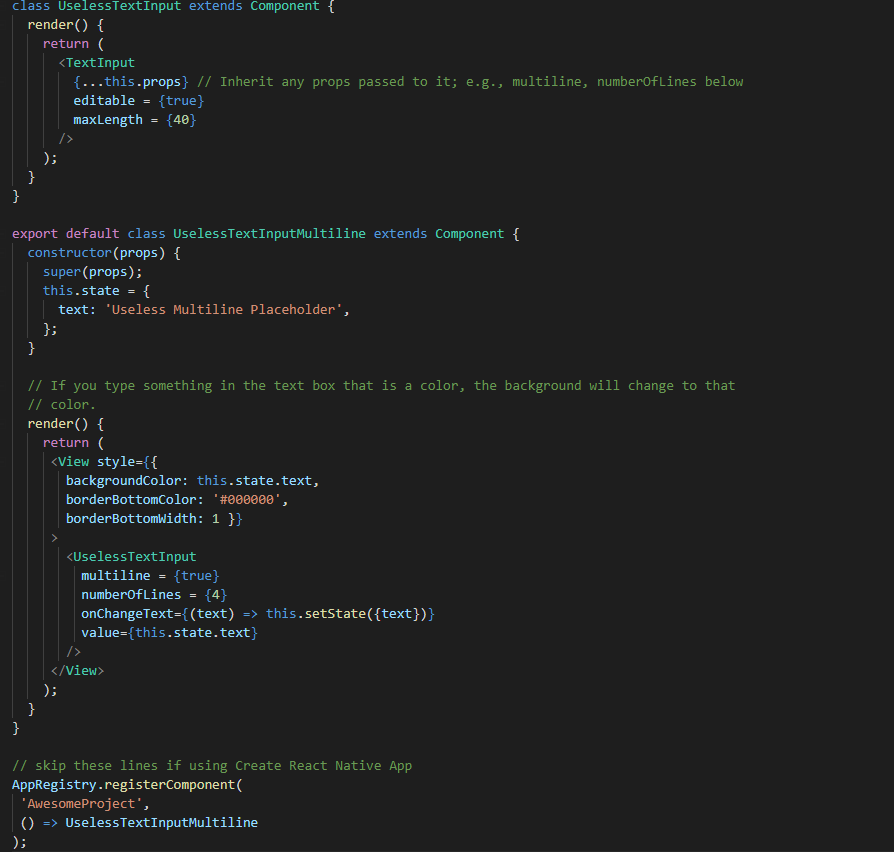
## Text component:



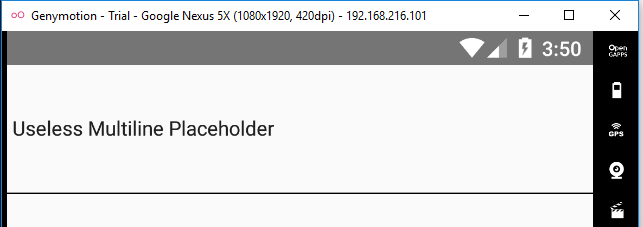
**Demo:**



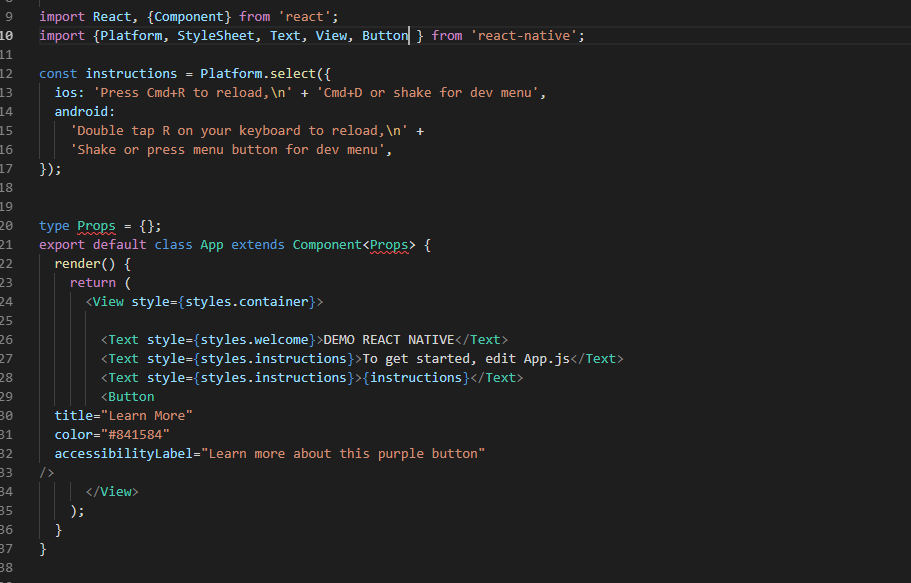
**TextInput:** là 1 thành phần nền tản cho phép nhập thông tin vào ứng dụng từ bàn phím, đạo cụ cung cấp khả năng sửa các tính năng tùy chỉnh cho các bàn phím khác nhau



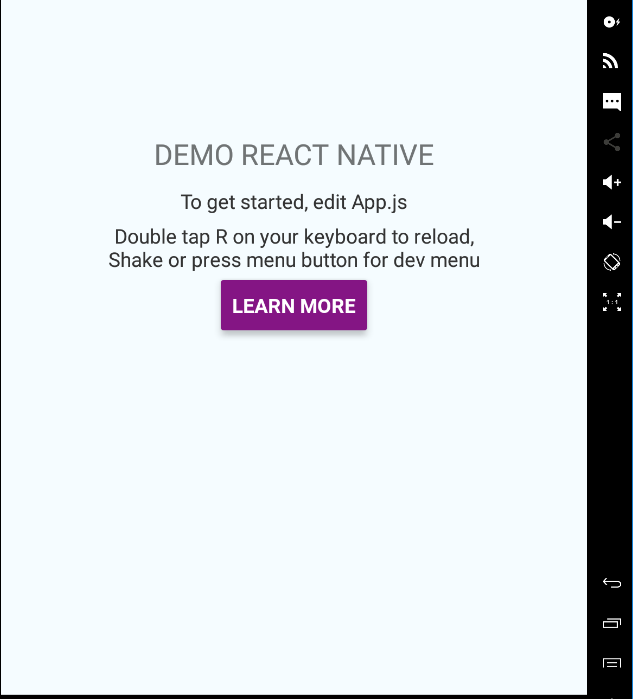
**Demo**:



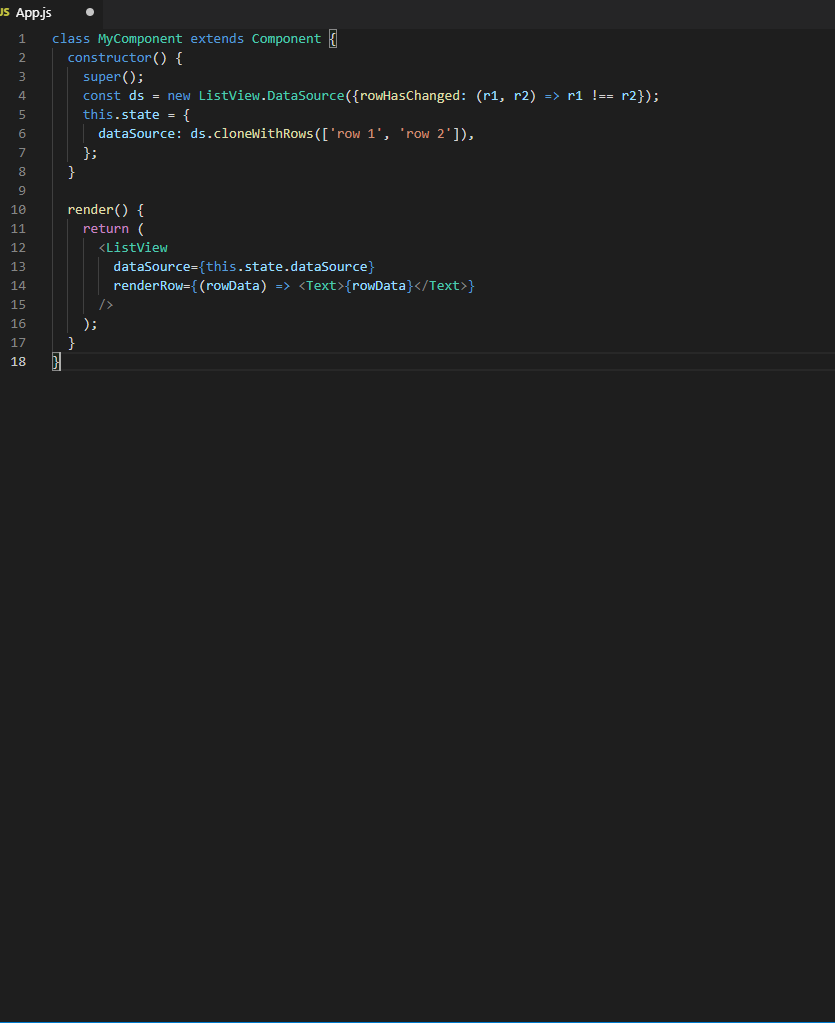
**Button:** một thành phần nút cơ bản sẽ hiển thị trên bất kỳ nền tản nào, hỗ trợ mức độ tùy biến tối thiểu



**Demo**



**ListViews:** một thành phần cốt lõi được thiết kế hiển thị hiệu quả các danh sách cuộn theo chiều dọc của dữ liệu



**FlatList:** giao diện biểu diễn để hiển thị danh sách đơn giản và hỗ trợ các tính năng tiện dụng nhất

Một số tính năng hay của Flatlist:

- Hoàn toàn đa nền tản

- Chế độ ngang tùy chọn

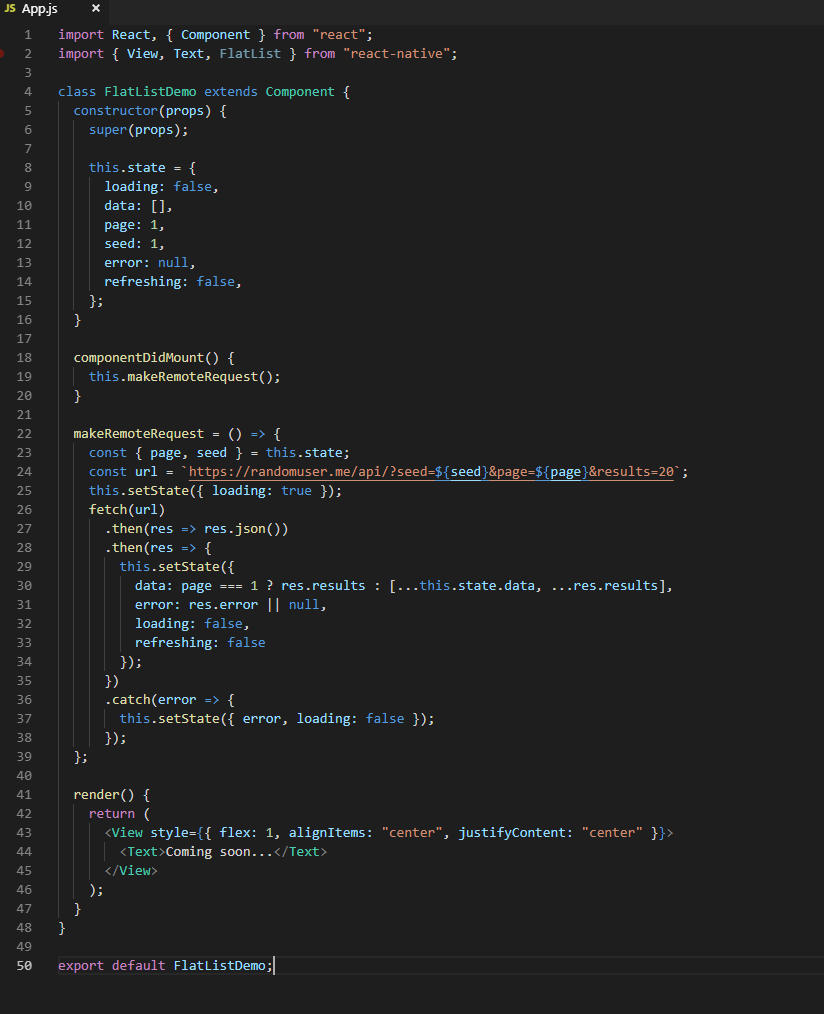
- Cấu hình gọi lại khả năng xem

- Hỗ trợ tiêu đề

- Hỗ trợ chân trang

- Cuộn xem

- Kéo để làm mới



# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Phân tích hệ thống

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

#### 2.1.1.1 User Interfaces

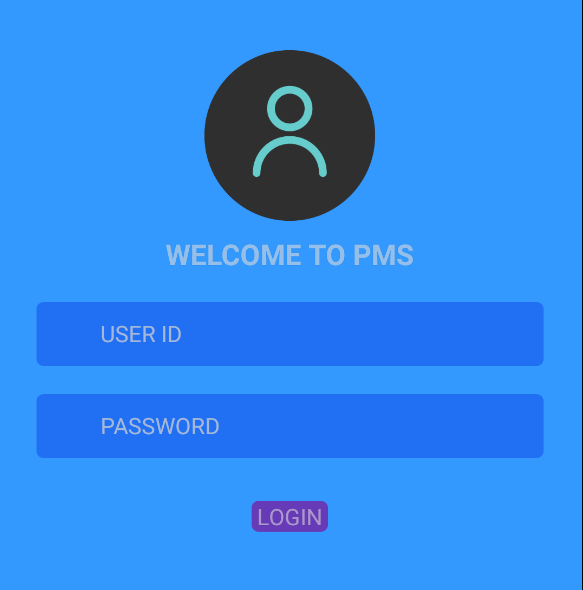


Image1: màng hình đăng nhập

#### 2.1.1.2. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **USER ID** | Phần này là phần cho phép người dùng nhập vào dữ liệu từ bàn phím theo trường dẫn cụ thể hơn là mục nhập tên id tài khoản của tài khoản đã đăng ký sử dụng ứng dụng | Chạm vào nó để bắt đầu nhập từ bàn phím | Bàn phím sẽ hiện lên |
| **PASSWORD** | Phần này cho phép người dùng nhập dữ liệu từ bàn phím theo đường dẫn cụ thể hơn là mục nhập mật khẩu của tài khoản đã đăng ký sử dụng ứng dụng | Chạm vào nó để bắt đầu nhập từ bàn phím | Bàn phím sẽ hiện lên |
| **NÚT LOGIN** | Nút này là nút chức năng cho phép ứng dụng kiểm tra và đăng nhập vào bên trong để sử dụng ứng dụng, khi click vào nó thì hệ thống sẽ cho phép đăng nhập vs điều kiện tài khoản được nhập trên 2 trường kia chính xác thì nó sẽ được chấp nhận cho người dùng vào trong ứng dụng | Chạm vào nút để đăng nhập | Bản thông báo sẽ hiện lên và ứng dụng chuyển tiếp tới giao diện chính của ứng dụng nếu tài khoản được chấp thuận và ngược lại thì màn hình đăng nhập vẫn giữ nguyên cho đến khi tài khoản chính xác được xác nhận |
| **GIAO DIỆN** | Hiển thị lời chào và tên ứng dụng |  |  |

**Code:**

**USER ID:**

**<View style={styles.inputContainer}>**

**<TextInput**

**style={styles.input}**

**placeholder={'USER ID'}**

**placeholderTextColor={'#A5B8D9'}**

**underLineColorAndroid='transparent'**

**/>**

**</View>**

**PASSWORD:**

**<View style={styles.inputContainer}>**

**<TextInput**

**style={styles.input}**

**placeholder={'PASSWORD'}**

**secureTextEntry={true}**

**placeholderTextColor={'#A5B8D9'}**

**underLineColorAndroid='transparent'**

**/>**

**</View>**

**NÚT LOGIN:**

**<TouchableOpacity style={styles.buttonLogin} onPress={() => this.props.navigation.navigate('Menu')}>**

**<Text style={styles.textLogin}> LOGIN </Text>**

**</TouchableOpacity>**

### 2.1.2. Feature/Component #1: MENU screen

#### 2.1.2.1 User Interfaces

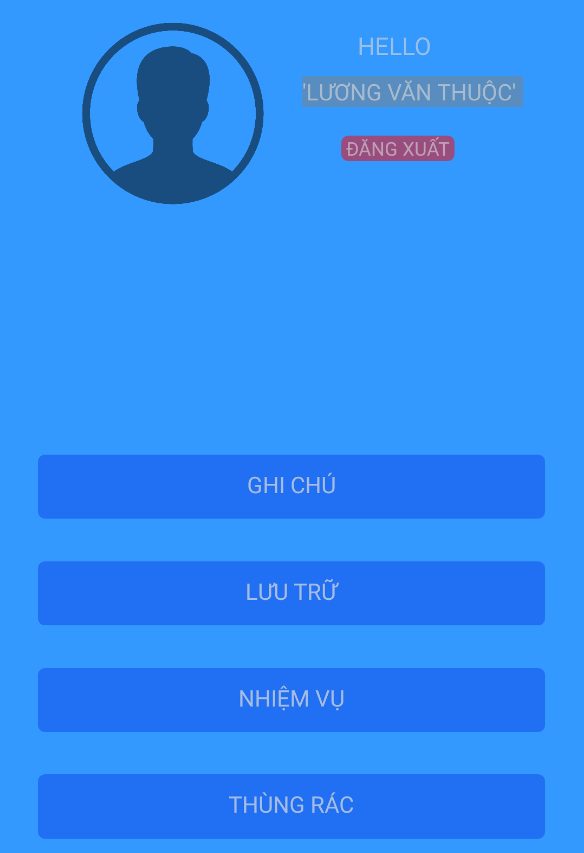


Image2: màng hình MENU

#### 2.1.2.2. Yêu cầu chức năng:

**MÀN HÌNH MENU:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **AVARTAR** | Mục này sẽ hiện ảnh của người dùng đặt cho họ hoặc ảnh mặc định nếu tài khoản chưa được thiết đặt hình người dùng |  | Hiện ảnh tùy theo người dùng đặt ở vị trí này |
| **USERNAME** | Mục này sẽ hiện tên tài khoản của người dùng giống với lúc người dùng đăng ký tài khoản |  | Hiện tên người dùng ở vị trí này |
| **NÚT ĐĂNG XUẤT** | Nó là một nút chức năng cho phép người dùng đăng xuất ra khỏi tài khoản của mình để ngắt quyền truy cập sử dụng và quay về màn hình đăng nhập | Nhấn vào đăng xuất | ứng dụng sẽ quay trở về màn hình đăng nhập ban đầu |
| **MỤC CHỨC NĂNG NHIỆM VỤ** | Mô tả cho một chức năng của ứng dụng cho phép người dùng nhấn vào nó để ứng dụng chuyển tiếp vào chức năng xem lịch hiện tại và đặt nhiệm vụ cho từng ngày | Nhấn vào nó | Ứng dụng sẽ chuyển tiếp vào giao diện của chức năng nhiệm vụ |
| **MỤC CHỨC NĂNG LƯU TRỮ** | Mô tả cho một chức năng của ứng dụng cho phép người dùng nhấn vào nó để ứng dụng chuyển tiếp vào chức năng quản lý các file lưu trữ của họ | Nhấn vào nó | Ứng dụng sẽ chuyển tiếp vào giao diện của chức năng lưu trữ |
| **MỤC CHỨC NĂNG GHI CHÚ** | Mô tả cho một chức năng của ứng dụng cho phép người dùng nhấn vào nó để ứng dụng chuyển tiếp vào chức năng quản lý ghi chú của họ | Nhấn vào nó | Ứng dụng sẽ chuyển tiếp vào giao diện của chức năng ghi chú |
| **MỤC CHỨC NĂNG THÙNG RÁC** | Mô tả cho một chức năng của ứng dụng cho phép người dùng nhấn vào nó để ứng dụng chuyển tiếp vào chức năng quản lý các tập tin các dữ liệu vừa được xóa | Nhấn vào nó | Ứng dụng sẽ chuyển tiếp vào giao diện của chức năng thùng rác |

Code:

**NÚT ĐĂNG XUẤT:**

**<TouchableOpacity style={styles.textLoginAccount} onPress={() => this.props.navigation.navigate('Login')}>**

**<Text style={styles.textLoginAccount1}> ĐĂNG XUẤT </Text>**

**</TouchableOpacity>**

**MỤC CHỨC NĂNG NHIỆM VỤ:**

**<TouchableOpacity onPress={() => this.props.navigation.navigate('GhiChu')} style={styles.buttonLogin}>**

**<Text style={styles.textLogin}> NHIỆM VỤ</Text>**

**</TouchableOpacity>**

**MỤC CHỨC NĂNG LƯU TRỮ:**

**<TouchableOpacity onPress={() => this.props.navigation.navigate('GhiChu')} style={styles.buttonLogin}>**

**<Text style={styles.textLogin}> LƯU TRỮ </Text>**

**</TouchableOpacity>**

**MỤC CHỨC NĂNG GHI CHÚ:**

**<TouchableOpacity onPress={() => this.props.navigation.navigate('GhiChu')} style={styles.buttonLogin}>**

**<Text style={styles.textLogin}> GHI CHÚ </Text>**

**</TouchableOpacity>**

**MỤC CHỨC NĂNG THÙNG RÁC:**

**<TouchableOpacity onPress={() => this.props.navigation.navigate('GhiChu')} style={styles.buttonLogin}>**

**<Text style={styles.textLogin}> THÙNG RÁC </Text>**

**</TouchableOpacity>**

**2.1.3. Feature/Component #1: GhiChu screen**

#### 2.1.3.1 User Interfaces



Image3: màng hình GHI CHÚ

#### 2.1.3.2. Yêu cầu chức năng:

**MÀN HÌNH GHI CHÚ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **NÚT QUAY LẠI** | Nút mũi tên quay lại trên cùng là nút chức năng cho phép người dùng quay lại màn hình trước đó là màn hình menu để sử dụng các chức năng khác | Nhấn vào nó | ứng dụng sẽ quay lại và hiển thị màn hình menu trước đó |
| **THANH THÔNG TIN** | Thanh thông tin trên cùng hay còn gọi là header , thanh này hiển thị tên tab chức năng cho phép người dùng biết đang ở trang chức năng nào |  | Hiển thị tên chức năng đang thực hiện |
| **MÀN HÌNH HIỂN THỊ** | Màn hình hiển thị này cho phép người dùng xem tất cả các thông báo đã tạo và lưu lại. Màn hình hiển thị tất nội dung và cho phép người dùng lướt lên nó để xem các nội dụng mới hoặc cũ hơn | Lướt vào nó nếu danh sách hiển thị không đủ | Tất cả thông tin đã được cập nhật sẽ được hiển thị |
| **NÚT THÊM MỚI GHI CHÚ** | Nút này cho phép người dùng tiến hành thêm các nội dung khác lên kho dữ liệu của mình | Nhấn vào nó | ứng dụng chuyển tiếp đến màn hình thêm ghi chú |
|  |  |  |  |

CODE:

**NÚT QUAY LẠI:**

Phần mặc định của navigation trong react nactive khi chúng ta không tùy biến nó

**THANH THÔNG TIN:**

**static navigationOptions = {**

**headerTitle: ' GHI CHÚ'**

**};**

**MÀN HÌNH HIỂN THỊ:**

**<View style= {styles.background}>**

**<View style={styles.flatList}>**

**<FlatList**

**data={this.state.dataSource}**

**renderItem={({ item }) => <Text style= {styles.TextFlat}>{item.stt} : {item.noidung}</Text>}**

**/>**

**</View>**

**<TouchableOpacity style={styles.buttonLogin} onPress={() => this.props.navigation.navigate('EditGC')}>**

**<Text style={styles.textLogin}> THÊM MỚI GHI CHÚ </Text>**

**</TouchableOpacity>**

**</View>**

**NÚT THÊM MỚI GHI CHÚ:**

**<TouchableOpacity style={styles.buttonLogin} onPress={() => this.props.navigation.navigate('EditGC')}>**

**<Text style={styles.textLogin}> THÊM MỚI GHI CHÚ </Text>**

**</TouchableOpacity>**

### 2.1.3. Feature/Component #1: THÊM GHI CHÚ screen

#### 2.1.3.1 User Interfaces

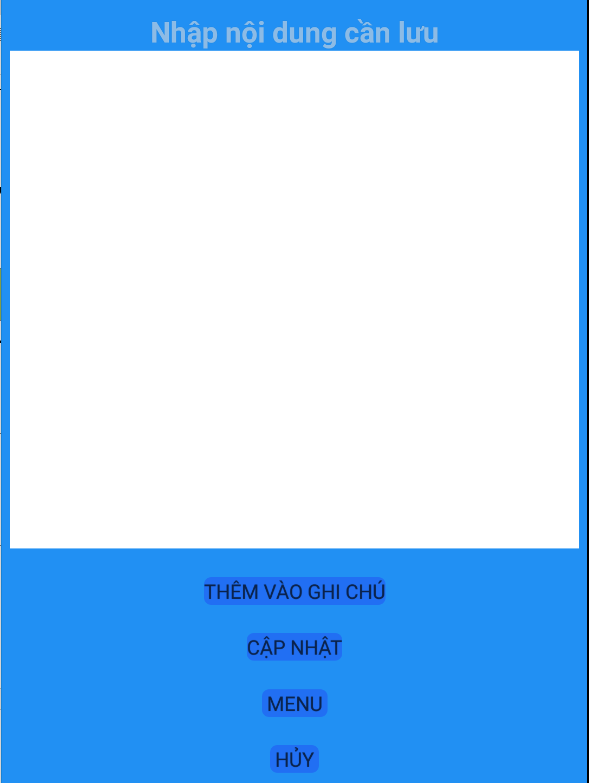
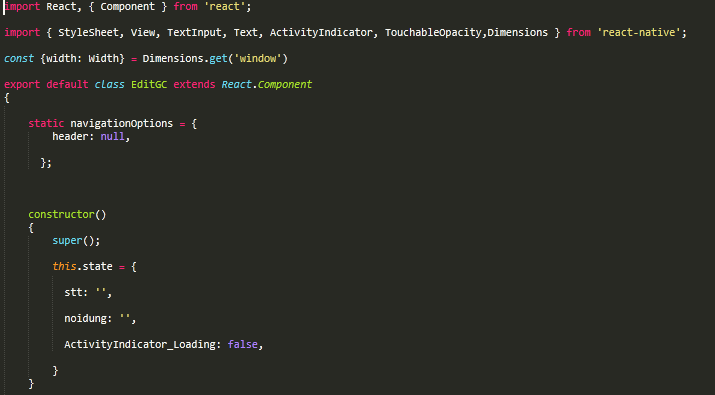


Image4: màng hình Thêm ghi chú

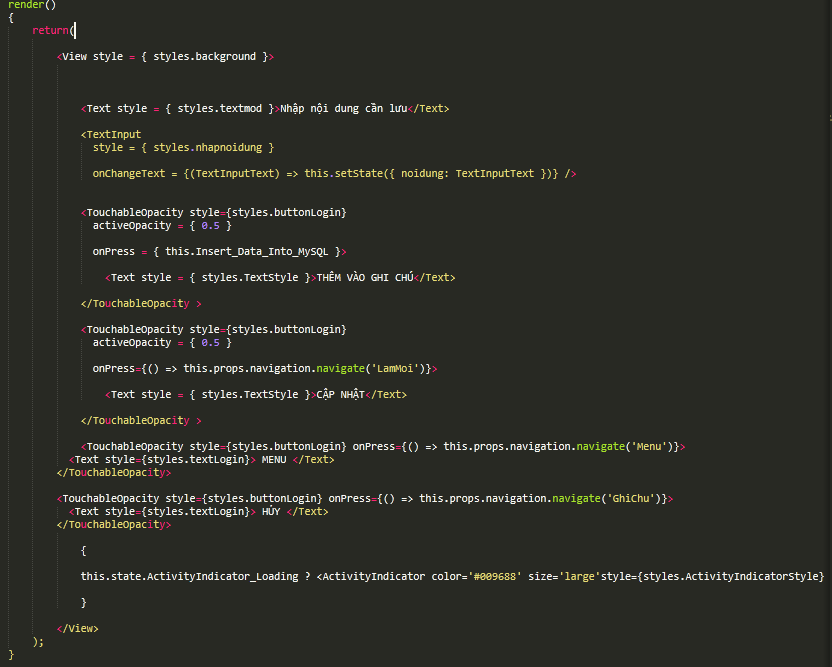
#### 2.1.3.2. Yêu cầu chức năng:

**MÀN HÌNH THÊM GHI CHÚ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **KHUNG NHẬP DỮ LIỆU** | Khung trống cho phép người dùng click vào nó và nhập các ghi chú hoặc dữ liệu mới vào trong vị trí này | Nhấn vào nó | Bàn phím hiện lên và cho phép người dùng nhập từ bàn phím |
| **NÚT THÊM GHI CHÚ** | Nút này là nút chức năng cho phép người dùng nhấn vào nó để thêm tất cả dữ liệu đã nhập ở trên khung dữ liệu bên trên vào cơ sở dữ liệu của người dùng | Nhấn vào nó | Thông báo hiện lên nhập thành công |
| **NÚT CẬP NHẬT** | Nút này là nút chức năng cho phép người dùng nhấn vào nó để xem cập nhật mới trên cơ sỡ dữ liệu người dùng | Nhấn vào nó | Màn hình sẽ hiện lên màn hình cập nhật |
| **NÚT MENU** | Nút này là nút chức năng cho phép người dùng nhấn vào nó để quay trở lại màn hình chính màn hình menu để thực hiện các chức năng khác | Nhấn vào nó | Màn hình menu hiện lên vì ứng dụng chuyển trở về màn hình menu |
| **NÚT HỦY** | Nút này là nút chức năng cho phép người dùng nhấn vào nó để quay trở lại màn hình trước đó là màn hình Ghi Chú | Nhấn vào nó | ứng dụng chuyển tiếp màn hình ghi chú trước đó hiện lên |







### 2.1.4. Feature/Component #1: XEM CẬP NHẬT screen

#### 2.1.4.1 User Interfaces

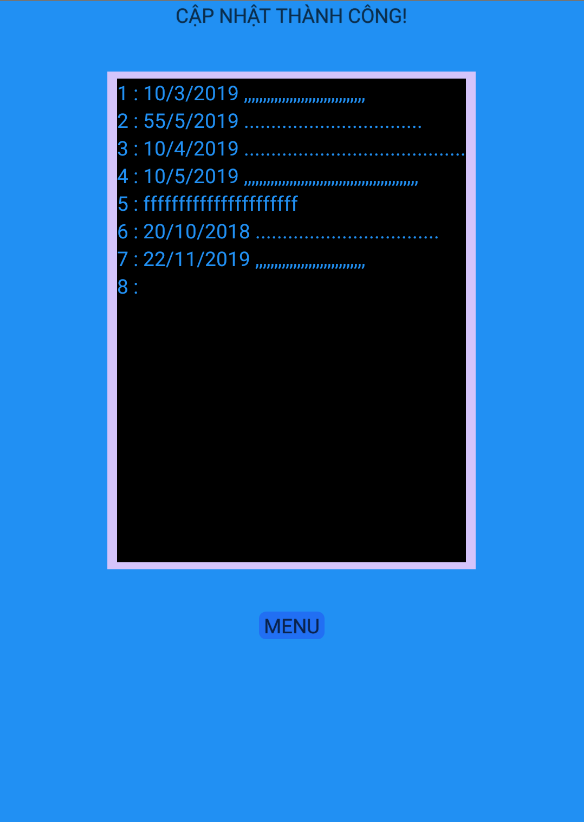


Image4: màng hình Thêm ghi chú

#### 2.1.4.2. Yêu cầu chức năng:

**MÀN HÌNH CẬP NHẬT:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **MỤC THÔNG BÁO** | Phần thông báo sẽ báo trạng thái cập nhật thành công hoặc không |  | Hiện thông tin trạng thái thành công hoặc thất bại |
| **MÀN HÌNH HIỂN THỊ** | Màn hình hiển thị này cho phép người dùng xem tất cả các thông báo đã tạo và lưu lại cập nhật mới nhất | Nhấn vào nó | Thông báo hiện lên nhập thành công |
| **NÚT MENU** | Nút này là nút chức năng cho phép người dùng nhấn vào nó để quay trở lại màn hình chính màn hình menu để thực hiện các chức năng khác | Nhấn vào nó | Màn hình sẽ hiện lên màn hình MENU |

# TÀI LIỆU THAM KHẢO

Các tài liệu tham khảo:

<https://facebook.github.io/react-native/docs/getting-started>

<https://reactnavigation.org/docs/en/getting-started.html>

<https://stacktips.com/tutorials/react/creating-login-screen-in-react-native>

<https://www.npmjs.com/package/react-native-mssql>

<https://www.youtube.com/watch?v=43HjPgfFAeE>

<https://tech.3si.vn/2015/08/16/web-api-tao-php-web-service-cho-ung-dung-mobile-client/>