**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙞🕮🙜**

****

**TÀI LIỆU ĐẶC TẢ PHẦN MỀM**

**Lập trình Di Động 3**

**App Manga Reader Online**

**Giảng viên hướng dẫn: (Thầy) Trương Bá Thái**

**Nhóm thực hiện: Nhóm 3**

1. **Ngô Quốc Cường**
2. **Lê Quốc Việt**

***TP. Hồ Chí Minh, Ngày 19 tháng 9 năm 2019***

**LỊCH SỬ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả** | **Tác giả** | **Người sửa đổi** |
| **19/09/2019** | **1.0** | **Viết bản SRS** | **Ngô Quốc Cường** | **Ngô Quốc Cường** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **MỤC LỤC**

[**MỤC LỤC** 3](#_Toc23401467)

[CHƯƠNG 1. TỔNG QUAN,GIỚI THIỆU VỀ ỨNG DỤNG 1](#_Toc23401468)

[1.1. Tổng quan về ứng dụng 1](#_Toc23401469)

[1.2. Tẩm hoạt động của ứng dụng 1](#_Toc23401470)

[1.3. Feature/Component #1: Màn hình chính/ trang chủ: 2](#_Toc23401471)

[1.3.1. Giao diện người dùng 2](#_Toc23401472)

[1.3.2. Yêu cầu chức năng 3](#_Toc23401473)

[1.4. Feature/Component #2: Màn hình chi tiết từng truyện 4](#_Toc23401474)

[1.4.1. Giao diện người dùng 4](#_Toc23401475)

[1.4.2. Yêu cầu chức năng 5](#_Toc23401476)

[1.5. Feature/Component #3: Màn hình chương truyện 5](#_Toc23401477)

[1.5.1. Giao diện người dùng 5](#_Toc23401478)

[1.5.2. Yêu cầu chức năng 6](#_Toc23401479)

[CHƯƠNG 2. TÌM HIỂU VỀ ES6 7](#_Toc23401480)

[CHƯƠNG 3. CÁC COMPONENT 8](#_Toc23401481)

[CHƯƠNG 4. PROJECT 24](#_Toc23401482)

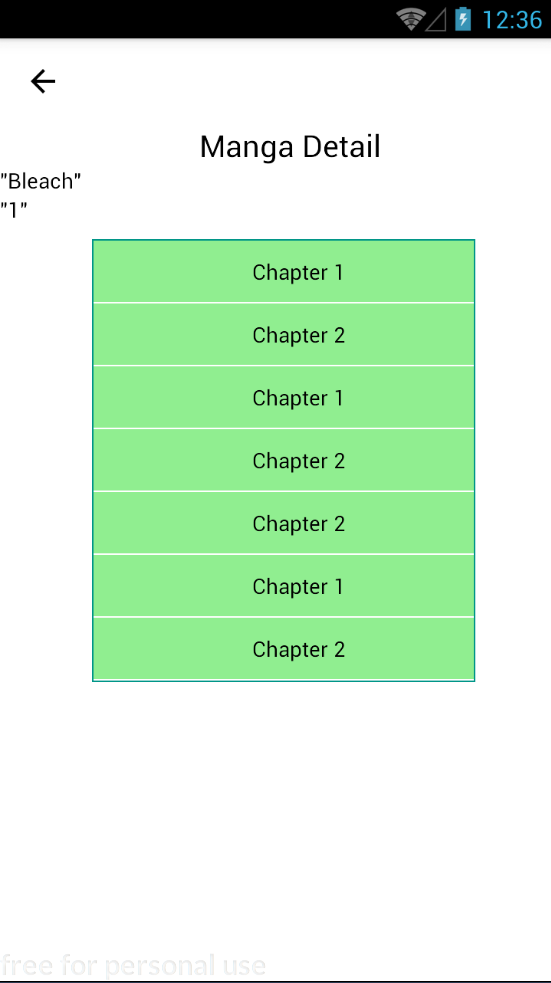
[4.1: Màn hình đăng nhập (Việt làm) 24](#_Toc23401483)

[4.2: Màn hình đăng kí (Cường làm) 29](#_Toc23401484)

[4.3: Màn hình trang chủ (Cường làm) 32](#_Toc23401485)

[4.4: Màn hình tìm kiếm (Việt làm) 38](#_Toc23401486)

[4.5: Màn hình chi tiết trang truyện ( Cường làm) 44](#_Toc23401487)

[ 45](#_Toc23401488)

[*MangaDetail.js* 45](#_Toc23401489)

[CHƯƠNG 5. BÁO CÁO KIẾN NGHỊ 49](#_Toc23401490)

1. TỔNG QUAN,GIỚI THIỆU VỀ ỨNG DỤNG
   1. Tổng quan về ứng dụng

* App hỗ trợ việt đọc truyện tranh online của giới trẻ
  1. Tẩm hoạt động của ứng dụng

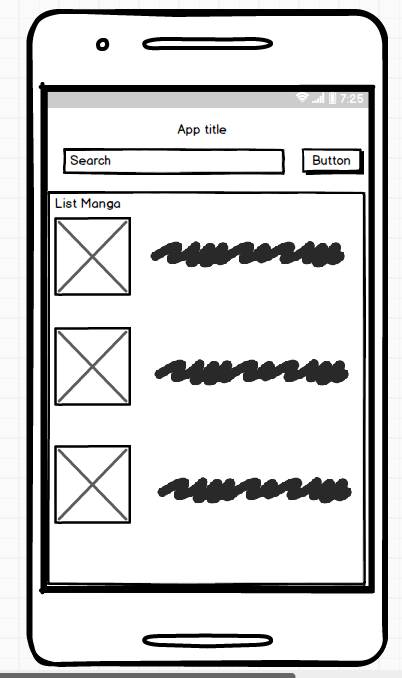
Android OS: Hỗ trợ các thiết bị có Android 26 , API 23 trở lên

* Android test devices: Oppo,Xiaomi,Sony Xperia Z

Android screen resolutions: 720x1520

Màn hình chỉ hỗ trợ chạy theo hàng dọc

* 1. Feature/Component #1: Màn hình chính/ trang chủ:
     1. Giao diện người dùng

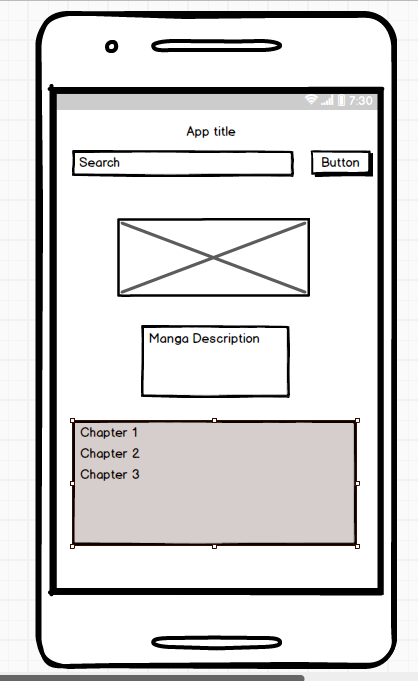


Hình 1 : Giao diện màn hình chính/trang chủ

* + 1. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Đối tượng** | **Mô tả** | **Thao tác** | **Phản hồi** |
| **Màn hình gồm có** | * Text AppTitle * Button Search * TextInput Search * FlatList Truyen | N/A | N/A |
| **Button Search** | Tìm kiếm truyện | Nhấp vào button | Hiện ra danh sách truyện cần tìm |
| **TextInput** | Nhập truyện cần tìm kiếm | Nhấp vào TextInput | N/A |
| **FlatList Truyen** | Gồm các itiem hiển thị danh sách các truyện hiện có | Nhấp vào itiem muốn xem | Chuyển sang màn hình**“Chi tiết từng truyện”** |

* 1. Feature/Component #2: Màn hình chi tiết từng truyện
     1. Giao diện người dùng

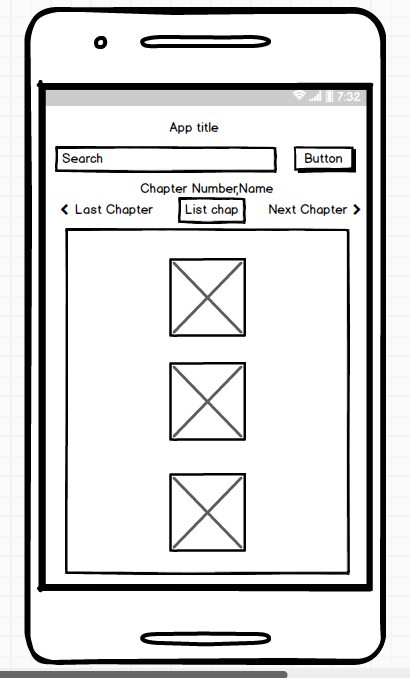


Hình 2 : Màn hình chi tiết từng truyện

* + 1. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Đối tượng** | **Mô tả** | **Thao tác** | **Phản hồi** |
| Màn hình chọn nhân vật gồm có | * Text AppTitle * Button Search * TextInput Search * FlatList ChuongTruyen * Image Truyen * Text Mota | N/A | N/A |
| **Button Search** | Tìm kiếm truyện | Nhấp vào button | Hiện ra danh sách truyện cần tìm |
| **TextInput Search** | Nhập truyện cần tìm kiếm | Nhấp vào TextInput | N/A |
| **Image Truyen** | Hiển thị giao diện bìa của truyện | N/A | N/A |
| **Text Mota** | Mô tả sơ lược về truyện | N/A | N/A |
| **FlastList ChuongTruyen** | Chứa các itiem hiển thị danh sách các chương truyện hiện có của 1 truyện | Nhấp vào itiem muốn xem | Chuyển sang màn hình**“Chương truyện”** |

* 1. Feature/Component #3: Màn hình chương truyện
     1. Giao diện người dùng



Hình 3 : Giao diện màn hình từng trang truyện

* + 1. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Đối tượng** | **Mô tả** | **Thao tác** | **Phản hồi** |
| **Màn hình hướng dẫn chơi gồm có** | Text App Title  TextInput Search  Button Search  Text Chap Title  Button Back  Button Next  ComboBox ChuongTruyen  FlatList AnhTruyen | N/A | N/A |
| **Button Search** | Tìm kiếm truyện | Nhấp vào button | Hiện ra danh sách truyện cần tìm |
| **TextInput Search** | Nhập truyện cần tìm kiếm | Nhấp vào TextInput | N/A |
| **Button Next** | Cho phép người dùng chuyển chương kế tiếp | Nhấp vào button | Chuyển sang chương tiếp theo |
| **Button Back** | Cho phép người dùng chuyển chương kế trước đó | Nhấp vào button | Chuyển sang chương trước đó |
| **ComboBox ChuongTruyen** | Cho phép người dùng chọn chương truyện tùy ý | Nhấp vào combo box rồi chọn chương tùy ý | Chuyển sang chương truyện đã chọn |
| **FlatList AnhTruyen** | Gồm các item hiển thị nội dung chương truyện | N/A | N/A |

CHƯƠNG 2. TÌM HIỂU VỀ ES6

*Ngô Quốc Cường tìm hiểu về Let,var, const ,arrow function , kiểu dữ liệu object trong es6*

**const**

const dùng để khai báo một hằng số - là một giá trị không thay đổi được trong suốt quá trình chạy.

Ví dụ:

const A = 5;

A = 10; *// Lỗi Uncaught TypeError: Assignment to constant variable*

Sử dụng var:

function foo() {

var x = 10;

if (true) {

var x = 20; *// x ở đây cũng là x ở trên*

console.log(x); *// in ra 20*

}

console.log(x); *// vẫn là 20*

}

Sử dụng let:

function foo() {

let x = 10;

if (true) {

let x = 20; *// x này là x khác rồi đấy*

console.log(x); *// in ra 20*

}

console.log(x); *// in ra 10*

}

ArrowFunction

Arrow function - còn được gọi là "fat arrow", là cú pháp được mượn từ CoffeeScript (một ngôn ngữ chuyển tiếp), cú pháp này là cách ngắn gọn hơn dùng để viết function. Ở đây sử dụng kí tự =>, trông giống như một mũi tên "béo". Arrow function là một hàm vô danh và nó thay đổi cách this bind đến function. Arrow function làm code của ta trông ngắn gọn hơn, giúp đơn giản hóa function scoping cũng như từ khóa this. Arrow function hoạt động tương tự như Lambdas trong các ngôn ngữ khác như C # hay Python. Bằng cách sử dụng arrow function, chúng ta tránh được việc phải gõ từ khoá function, return và dấu ngoặc nhọn.

//ES5

var phraseSplitterEs5 = function phraseSplitter(phrase) {

return phrase.split(' ');

};

//ES6

var phraseSplitterEs6 = phrase => phrase.split(" ");

console.log(phraseSplitterEs6("ES6 Awesomeness")); // ["ES6", "Awesomeness"]

Kiểu dữ liệu object

Object trong JavaScript có thể được tạo theo một số cách khác nhau. Object literal là kiểu cú pháp tạo object sử dụng cặp dấu ngoặc {} và bên trong đó là danh sách các **property** (thuộc tính) của object.

**let** name = "John"; **let** myObj = { name, sayHi() { console.log("Hi, my nam is " + **this**.name) } }; myObj.sayHi() *// Hi, my nam is John*

*Lê Quốc Việt tìm hiểu về class,mảng,set,map trong es6*

Class

class Employee {

setName (name) {

this.name = name;

}

getName () {

return this.name;

}

setAge (age) {

this.age = age;

}

getAge () {

return this.age;

}

};

Array trong es6

đây là cách lấy biến trong mảng mà ta thường dùng để gán vào biến khác Tuy nhiên ES6 có 1 cách đơn giản ngăn hơn

let [a, b, c, d] = [1, 2, 3, 4];

hoặc với object thông thường có 1 cách lấy giá trị như:

var luke = { occupation: 'jedi', father: 'anakin' };

var occupation = luke.occupation; // 'jedi'

var father = luke.father; // 'anakin'

bây giờ có thể được thay thế bằng:

let luke = { occupation: 'jedi', father: 'anakin' };

let {occupation, father} = luke;

map trong es6

// Tạo một map gồm 3 thông tin

 let map = new Map([

     ["Name", "Nguyen Van Cuong"],

     ["Email", "thehalfheart@gmail.com"],

     ["Website", "freetuts.net"]

]);

console.log(map);

// Map {"Name" => "Nguyen Van Cuong", "Email" => "thehalfheart@gmail.com", "Website" => "freetuts.net"}

Set trong es6

Set là tập hợp các giá trị không bị trùng lặp, nghĩa là trong một set không thể có hai giá trị bằng nhau.

const s = new Set()

set

.add('red')

.add('blue')

.add('sweet')

.add('you')

s.size // 4

Bạn cũng có thể truyền một mảng vào hàm dựng của Set.

const s = new Set(['red', 'blue', 'sweet', 'red', 'you'])

console.log(s) // Set (4) {'red', 'blue', 'sweet', 'you'}

Bạn cũng có thể thấy giá trị 'red' bị trùng lặp đã được loại bỏ. Chúng ta có thể áp dụng Set để tạo ra một mảng chứa những phần tử duy nhất.

const a = ['red', 'blue', 'sweet', 'red', 'you']

const b = [...new Set(a)]

console.log(b) // [ 'red', 'blue', 'sweet', 'you' ]

Cũng tương tự như Map, Set sử dụng SameZeroValue để so sánh các phần tử với nhau.

const obj = {}

const s = new Set([NaN, {}, obj])

s.has(NaN) // true

s.has(obj) // true

s.has({}) // false

Để duyệt qua các phần tử của Set, bạn có thể dùng các phương thức như với Map.

const s = new Set([1, 2, 3, 4, 5])

// Vì Set không có khái niệm keys nên kết quả của `s.keys()` và `s.values()` là như nhau.

s.keys()

s.values()

s.entries()

s.forEach(function(value, key, setReference) {

}, thisArg)

for (let el of s) {

console.log(el)

}

Một số thao tác khác trên Set.

const s = new Set([1, 2, 3, 4, 5])

// Xóa một phần tử trong set

s.delete(3) // Set (4) {1, 2, 4, 5}

// Xóa hết phần tử trong set

s.clear()

Code demo ES6 của Lê Quốc Việt

var x = 20;

var fib = [];

fib[0] = 0;

fib[1] = 1;

for (let i = 2; i <= x; i++) {

fib[i] = fib[i - 2] + fib[i - 1];

}

for (i = 1; i < fib.length ; i++) {

console.log(fib[i]);

}

Code demo ES6 của Ngô Quốc Cường về arrow function, map

var arraySanPham = [

{masp: 'sp001' , tensp : 'LapTop '},

{masp: 'sp002' , tensp : 'Tivi '},

{masp: 'sp003' , tensp : 'Tủ Lạnh'},

];

Console.log(arraySanPham.map(arraySanPham => arraySanPham.tensp));

CHƯƠNG 3. CÁC COMPONENT

Phần làm của Lê Quốc Việt

Props :

import React, { Component } from 'react';

import { Text, View } from 'react-native';

class Greeting extends Component {

  render() {

    return (

      <View style={{alignItems: 'center'}}>

        <Text>Hello {this.props.name}!</Text>

      </View>

    );

  }

}

export default class LotsOfGreetings extends Component {

  render() {

    return (

      <View style={{alignItems: 'center', top: 50}}>

        <Greeting name='Rexxar' />

        <Greeting name='Jaina' />

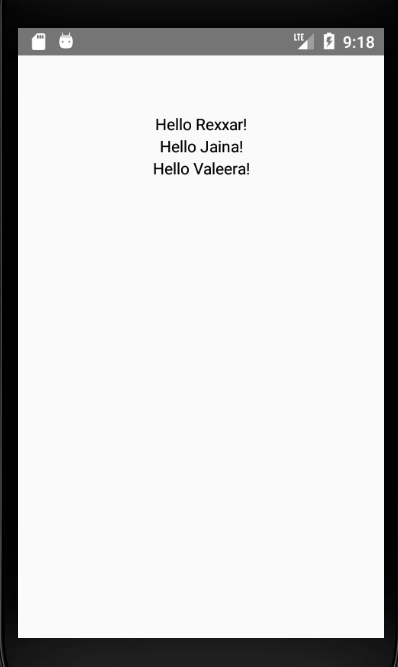
        <Greeting name='Valeera' />

      </View>

    );

  }

}



State:

import React, { Component } from 'react';

import { Text, View } from 'react-native';

class Blink extends Component {

  componentDidMount(){

    // Toggle the state every second

    setInterval(() => (

      this.setState(previousState => (

        { isShowingText: !previousState.isShowingText }

      ))

    ), 1000);

  }

  //state object

  state = { isShowingText: true };

  render() {

    if (!this.state.isShowingText) {

      return null;

    }

    return (

      <Text>{this.props.text}</Text>

    );

  }

}

export default class BlinkApp extends Component {

  render() {

    return (

      <View>

        <Blink text='I love to blink' />

        <Blink text='Yes blinking is so great' />

        <Blink text='Why did they ever take this out of HTML' />

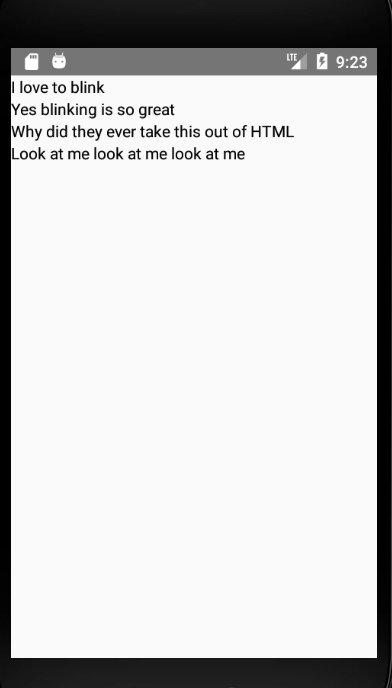
        <Blink text='Look at me look at me look at me' />

      </View>

    );

  }

}



Style trong react:

import React, { Component } from 'react';

import { StyleSheet, Text, View } from 'react-native';

const styles = StyleSheet.create({

  bigBlue: {

    color: 'blue',

    fontWeight: 'bold',

    fontSize: 30,

  },

  red: {

    color: 'red',

  },

});

export default class LotsOfStyles extends Component {

  render() {

    return (

      <View>

        <Text style={styles.red}>just red</Text>

        <Text style={styles.bigBlue}>just bigBlue</Text>

        <Text style={[styles.bigBlue, styles.red]}>bigBlue, then red</Text>

        <Text style={[styles.red, styles.bigBlue]}>red, then bigBlue</Text>

      </View>

    );

  }

}



Width và Height:

import React, { Component } from 'react';

import { View } from 'react-native';

export default class FixedDimensionsBasics extends Component {

  render() {

    return (

      <View>

        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />

        <View style={{width: 100, height: 100, backgroundColor: 'skyblue'}} />

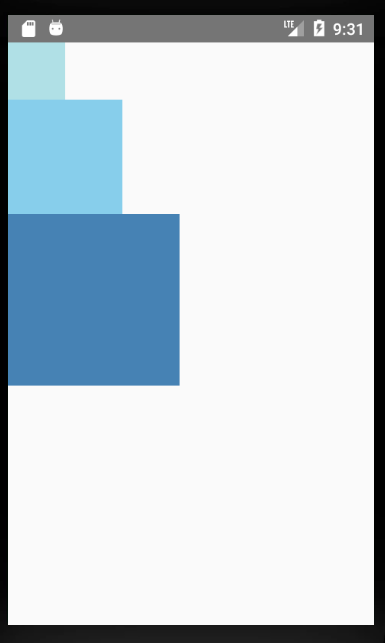
        <View style={{width: 150, height: 150, backgroundColor: 'steelblue'}} />

      </View>

    );

  }

}



Flex:

import React, { Component } from 'react';

import { View } from 'react-native';

export default class FlexDimensionsBasics extends Component {

  render() {

    return (

      // Try removing the `flex: 1` on the parent View.

      // The parent will not have dimensions, so the children can't expand.

      // What if you add `height: 300` instead of `flex: 1`?

      <View style={{flex: 1}}>

        <View style={{flex: 1, backgroundColor: 'powderblue'}} />

        <View style={{flex: 2, backgroundColor: 'skyblue'}} />

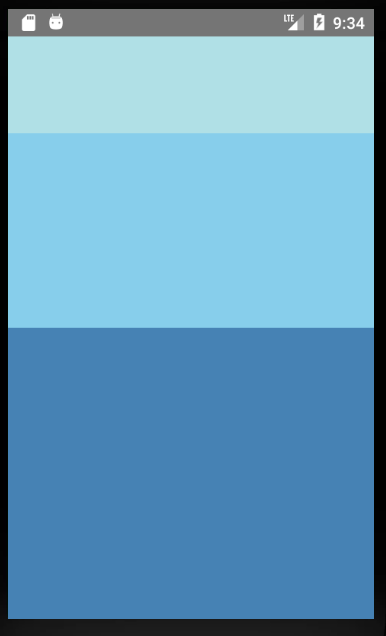
        <View style={{flex: 3, backgroundColor: 'steelblue'}} />

      </View>

    );

  }

}



FlexBox:

import React, { Component } from 'react';

import { View } from 'react-native';

export default class FlexDirectionBasics extends Component {

  render() {

    return (

      // Try setting `flexDirection` to `column`.

      //<View style={{flex: 1, flexDirection: 'row'}}>

      //<View style={{flex: 1, flexDirection: 'row-reverse'}}>

        <View style={{flex: 1, flexDirection: 'column'}}>

        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />

        <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />

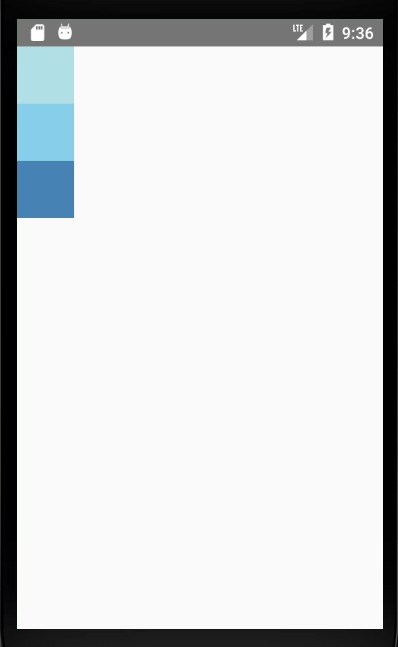
        <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />

      </View>

    );

  }

};



Text Component:

import React, { Component } from 'react';

import { Text, StyleSheet } from 'react-native';

export default class TextInANest extends Component {

  constructor(props) {

    super(props);

    this.state = {

      titleText: "Bird's Nest",

      bodyText: 'This is not really a bird nest.'

    };

  }

  render() {

    return (

      <Text style={styles.baseText}>

        <Text style={styles.titleText} onPress={this.onPressTitle}>

          {this.state.titleText}{'\n'}{'\n'}

        </Text>

        <Text numberOfLines={5}>

          {this.state.bodyText}

        </Text>

      </Text>

    );

  }

}

const styles = StyleSheet.create({

  baseText: {

    fontFamily: 'Cochin',

  },

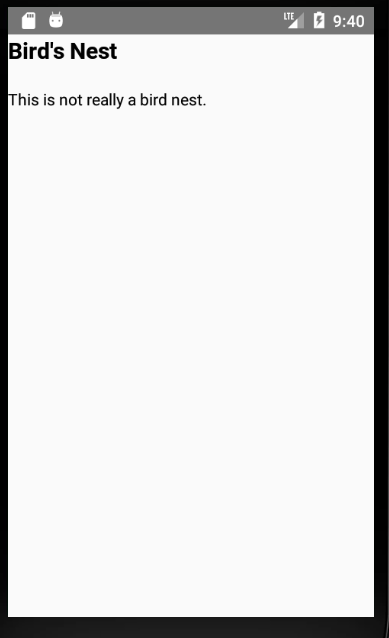
  titleText: {

    fontSize: 20,

    fontWeight: 'bold',

  },

});



Text input Component:

import React, { Component } from 'react';

import { Text, TextInput, View } from 'react-native';

export default class PizzaTranslator extends Component {

  constructor(props) {

    super(props);

    this.state = {text: ''};

  }

  render() {

    return (

      <View style={{padding: 10}}>

        <TextInput

          style={{height: 40}}

          placeholder="Type here to translate!"

          onChangeText={(text) => this.setState({text})}

          value={this.state.text}

        />

        <Text style={{padding: 10, fontSize: 42}}>

          {this.state.text.split(' ').map((word) => word && '🍕').join(' ')}

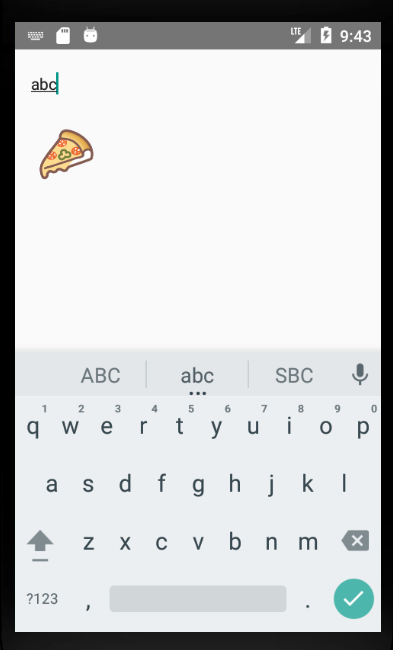
        </Text>

      </View>

    );

  }

}



Button Component:

import React, { Component } from 'react';

import { Text, TextInput, View, Button, Alert } from 'react-native';

export default class PizzaTranslator extends Component {

  constructor(props) {

    super(props);

    this.state = {text: ''};

  }

  render() {

    return (

      <View style={{padding: 10}}>

        <TextInput

          style={{height: 40}}

          placeholder="Type here to translate!"

          onChangeText={(text) => this.setState({text})}

          value={this.state.text}

        />

         <Button

          title="Press me"

          onPress={() => Alert.alert('Simple Button pressed')}

        />

        <Text style={{padding: 10, fontSize: 42}}>

          {this.state.text.split(' ').map((word) => word && '🍕').join(' ')}

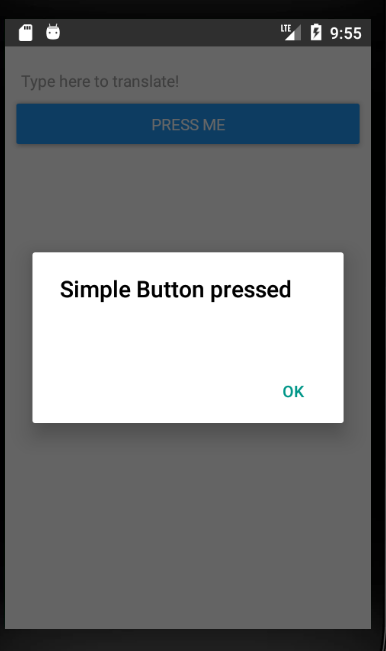
        </Text>

      </View>

    );

  }

}



CheckBox Component:

export default class PRNRadioForm extends Component {

    \_onSelect = ( item ) => {

      console.log(item);

    };

  render() {

    return (

      <View style={styles.container}>

          <View style={{ marginVertical: 10, backgroundColor: "#E7E7E7" }} >

              <CheckboxFormX

                  style={{ width: 350 - 30 }}

                  dataSource={mockData}

                  itemShowKey="label"

                  itemCheckedKey="RNchecked"

                  iconSize={16}

                  formHorizontal={true}

                  labelHorizontal={false}

                  onChecked={(item) => this.\_onSelect(item)}

              />

          </View>

     </View>

    );

  }

}

Radio Button:

import RadioForm, {RadioButton, RadioButtonInput, RadioButtonLabel} from 'react-native';

var radio\_props = [

  {label: 'param1', value: 0 },

  {label: 'param2', value: 1 }

];

var RadioButtonProject = React.createClass({

  getInitialState: function() {

    return {

      value: 0,

    }

  },

  render: function() {

    return (

      <View>

        <RadioForm

          radio\_props={radio\_props}

          initial={0}

          onPress={(value) => {this.setState({value:value})}}

        />

      </View>

    );

  }

});

**Phần làm của Ngô Quốc Cường**

ProgressBar

VD:

import React, {Component} from 'react';

import {ProgressBarAndroid, StyleSheet, View} from 'react-native';

export default class progress extends Component {

render() {

return (

<View style={styles.container}>

<ProgressBarAndroid />

<ProgressBarAndroid styleAttr="Horizontal" />

<ProgressBarAndroid styleAttr="Horizontal" color="#2196F3" />

<ProgressBarAndroid

styleAttr="Horizontal"

indeterminate={false}

progress={0.5}

/>

</View>

);

}

}

const styles = StyleSheet.create({

container: {

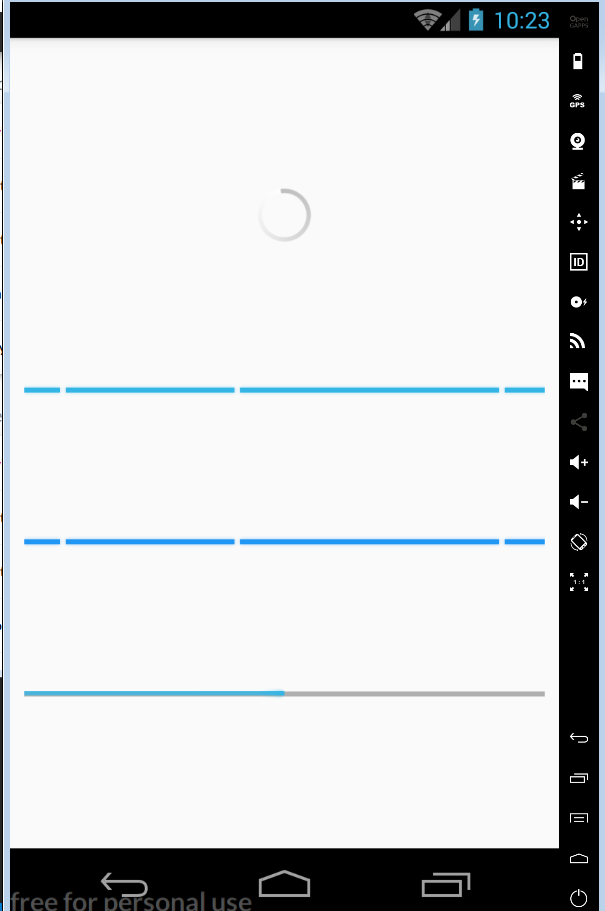
flex: 1,

justifyContent: 'space-evenly',

padding: 10,

},

});



ScrollView

VD:

import React, { Component } from 'react';

import { ScrollView, Image, Text } from 'react-native';

export default class IScrolledDownAndWhatHappenedNextShockedMe extends Component {

render() {

return (

<ScrollView>

<Text style={{fontSize:96}}>Scroll me plz</Text>

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}}/>

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Text style={{fontSize:96}}>If you like</Text>

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Text style={{fontSize:96}}>Scrolling down</Text>

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Text style={{fontSize:96}}>What's the best</Text>

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Text style={{fontSize:96}}>Framework around?</Text>

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

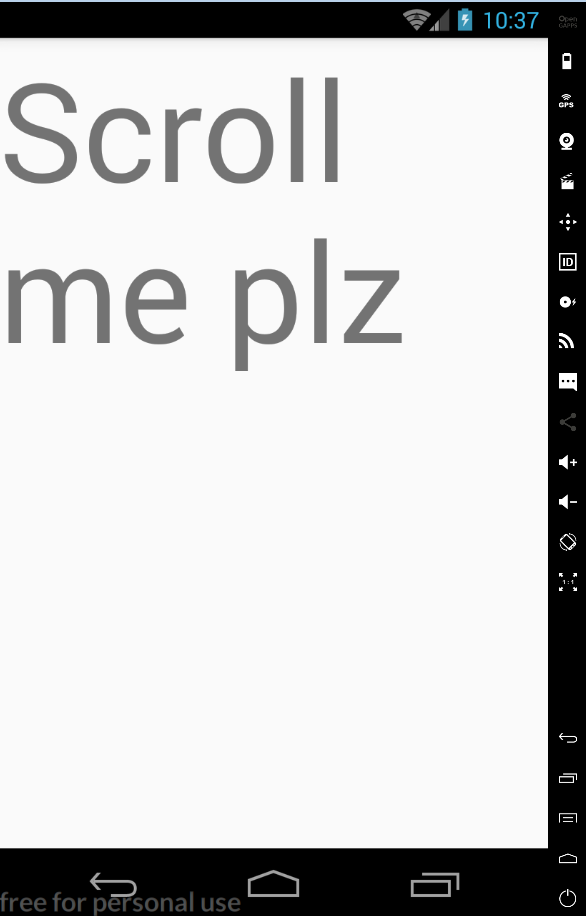
<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Image source={{uri: "https://facebook.github.io/react-native/img/tiny\_logo.png", width: 64, height: 64}} />

<Text style={{fontSize:80}}>React Native</Text>

</ScrollView>



ListView

VD:

import React, { Component } from 'react';

import { FlatList, StyleSheet, Text, View } from 'react-native';

export default class FlatListBasics extends Component {

render() {

return (

<View style={styles.container}>

<FlatList

data={[

{key: 'Devin'},

{key: 'Dan'},

{key: 'Dominic'},

{key: 'Jackson'},

{key: 'James'},

{key: 'Joel'},

{key: 'John'},

{key: 'Jillian'},

{key: 'Jimmy'},

{key: 'Julie'},

]}

renderItem={({item}) => <Text style={styles.item}>{item.key}</Text>}

/>

</View>

);

}

}

const styles = StyleSheet.create({

container: {

flex: 1,

paddingTop: 22

},

item: {

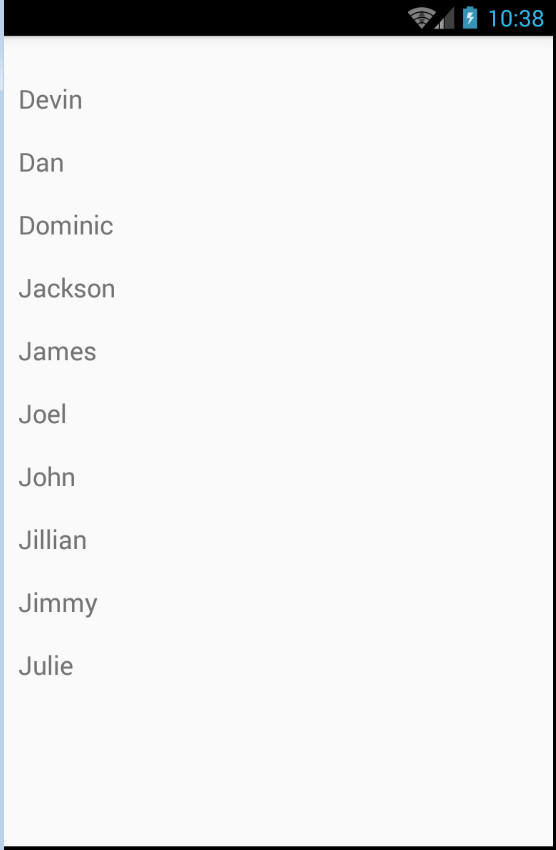
padding: 10,

fontSize: 18,

height: 44,

},

})



Picker

VD:

import React, { Component } from 'react'

import {StyleSheet,View, Text,Picker} from 'react-native'

export default class SwitchExample extends Component {

state = {

choosenIndex: 0

};

render() {

return (

<View style={styles.container}>

<Text style={styles.textStyle}>Swith Example</Text>

<Picker style={styles.pickerStyle}

selectedValue={this.state.language}

onValueChange={(itemValue, itemPosition) =>

this.setState({language: itemValue, choosenIndex: itemPosition})}

>

<Picker.Item label="Java" value="java" />

<Picker.Item label="JavaScript" value="js" />

<Picker.Item label="React Native" value="rn" />

</Picker>

<Text style={styles.textStyle}> {"Index ="+this.state.choosenIndex}</Text>

</View>

);

}

}

const styles = StyleSheet.create ({

container: {

flex: 1,

alignItems: 'center',

justifyContent: 'center',

},

textStyle:{

margin: 24,

fontSize: 25,

fontWeight: 'bold',

textAlign: 'center',

},

pickerStyle:{

height: 150,

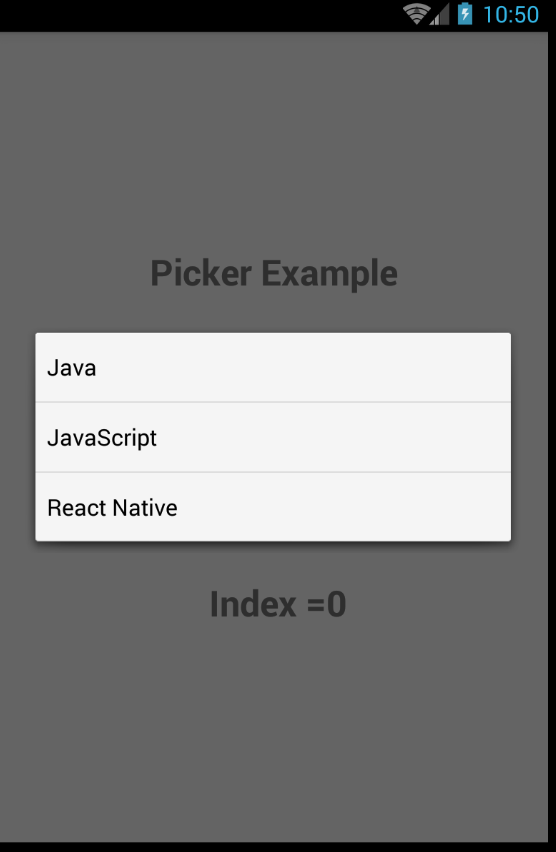
width: "80%",

color: '#344953',

justifyContent: 'center',

}

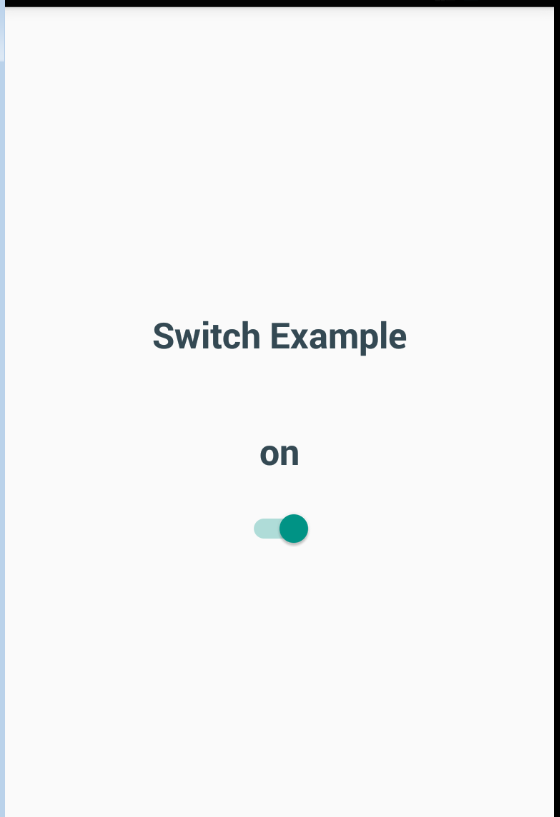
})



Switch

VD:

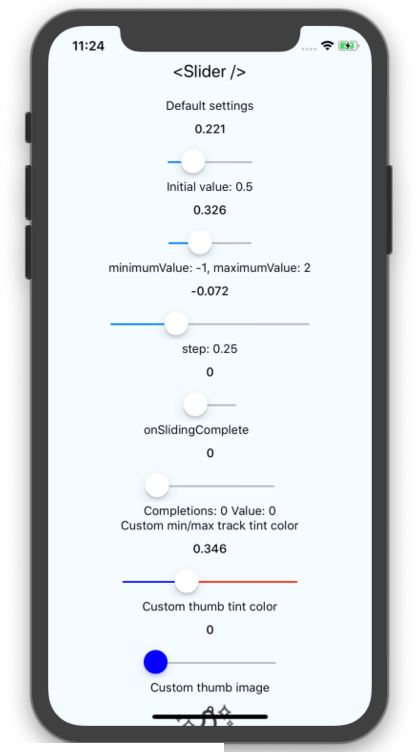
1. **import** React, { Component } from 'react'
2. **import** {StyleSheet, Switch, View, Text} from 'react-native'
4. export **default** **class** SwitchExample **extends** Component {
5. state = {
6. switchValue: **false**
7. };
9. render() {
10. **return** (
11. <View style={styles.container}>
12. <Text style={styles.textStyle}>Switch Example</Text>
13. <Text style={styles.textStyle}>{**this**.state.switchValue ? 'on' :'off'}</Text>
14. <Switch
15. value={**this**.state.switchValue}
16. onValueChange ={(switchValue)=>**this**.setState({switchValue})}/>
17. </View>
18. );
19. }
20. }
21. **const** styles = StyleSheet.create ({
22. container: {
23. flex: 1,
24. alignItems: 'center',
25. justifyContent: 'center',
26. },
27. textStyle:{
28. margin: 24,
29. fontSize: 25,
30. fontWeight: 'bold',
31. textAlign: 'center',
32. color: '#344953'
33. }
34. })



Slider

VD:

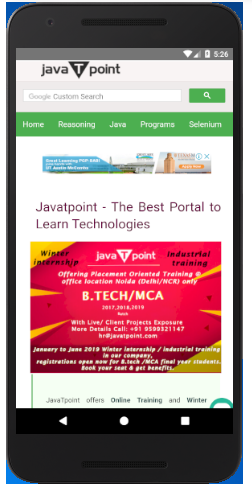
|  |
| --- |
| /\*\* |
|  | \* Copyright (c) Facebook, Inc. and its affiliates. |
|  | \* |
|  | \* This source code is licensed under the MIT license found in the |
|  | \* LICENSE file in the root directory of this source tree. |
|  | \* |
|  | \* @format |
|  | \* @flow |
|  | \*/ |
|  |  |
|  | 'use strict'; |
|  |  |
|  | const React = require('react'); |
|  | const {Text, StyleSheet, View} = require('react-native'); |
|  | const Slider = require('@react-native-community/slider'); |
|  |  |
|  | class SliderExample extends React.Component<$FlowFixMeProps, $FlowFixMeState> { |
|  | static defaultProps = { |
|  | value: 0, |
|  | }; |
|  |  |
|  | state = { |
|  | value: this.props.value, |
|  | }; |
|  |  |
|  | render() { |
|  | return ( |
|  | <View> |
|  | <Text style={styles.text}> |
|  | {this.state.value && +this.state.value.toFixed(3)} |
|  | </Text> |
|  | <Slider |
|  | {...this.props} |
|  | onValueChange={value => this.setState({value: value})} |
|  | /> |
|  | </View> |
|  | ); |
|  | } |
|  | } |
|  |  |
|  | class SlidingStartExample extends React.Component< |
|  | $FlowFixMeProps, |
|  | $FlowFixMeState, |
|  | > { |
|  | state = { |
|  | slideStartingValue: 0, |
|  | slideStartingCount: 0, |
|  | }; |
|  |  |
|  | render() { |
|  | return ( |
|  | <View> |
|  | <SliderExample |
|  | {...this.props} |
|  | onSlidingStart={value => |
|  | this.setState({ |
|  | slideStartingValue: value, |
|  | slideStartingCount: this.state.slideStartingCount + 1, |
|  | }) |
|  | } |
|  | /> |
|  | <Text> |
|  | Starts: {this.state.slideStartingCount} Value:{' '} |
|  | {this.state.slideStartingValue} |
|  | </Text> |
|  | </View> |
|  | ); |
|  | } |
|  | } |
|  |  |
|  | class SlidingCompleteExample extends React.Component< |
|  | $FlowFixMeProps, |
|  | $FlowFixMeState, |
|  | > { |
|  | state = { |
|  | slideCompletionValue: 0, |
|  | slideCompletionCount: 0, |
|  | }; |
|  |  |
|  | render() { |
|  | return ( |
|  | <View> |
|  | <SliderExample |
|  | {...this.props} |
|  | onSlidingComplete={value => |
|  | this.setState({ |
|  | slideCompletionValue: value, |
|  | slideCompletionCount: this.state.slideCompletionCount + 1, |
|  | }) |
|  | } |
|  | /> |
|  | <Text> |
|  | Completions: {this.state.slideCompletionCount} Value:{' '} |
|  | {this.state.slideCompletionValue} |
|  | </Text> |
|  | </View> |
|  | ); |
|  | } |
|  | } |
|  |  |
|  | const styles = StyleSheet.create({ |
|  | text: { |
|  | fontSize: 14, |
|  | textAlign: 'center', |
|  | fontWeight: '500', |
|  | margin: 10, |
|  | }, |
|  | }); |
|  |  |
|  | exports.title = '<Slider>'; |
|  | exports.displayName = 'SliderExample'; |
|  | exports.description = 'Slider input for numeric values'; |
|  | exports.examples = [ |
|  | { |
|  | title: 'Default settings', |
|  | render(): React.Element<any> { |
|  | return <SliderExample />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Initial value: 0.5', |
|  | render(): React.Element<any> { |
|  | return <SliderExample value={0.5} />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'minimumValue: -1, maximumValue: 2', |
|  | render(): React.Element<any> { |
|  | return <SliderExample minimumValue={-1} maximumValue={2} />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'step: 0.25', |
|  | render(): React.Element<any> { |
|  | return <SliderExample step={0.25} />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'onSlidingStart', |
|  | render(): React.Element<any> { |
|  | return <SlidingStartExample />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'onSlidingComplete', |
|  | render(): React.Element<any> { |
|  | return <SlidingCompleteExample />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Custom min/max track tint color', |
|  | render(): React.Element<any> { |
|  | return ( |
|  | <SliderExample |
|  | minimumTrackTintColor={'blue'} |
|  | maximumTrackTintColor={'red'} |
|  | value={0.5} |
|  | /> |
|  | ); |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Custom thumb tint color', |
|  | render(): React.Element<any> { |
|  | return <SliderExample thumbTintColor={'blue'} />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Custom thumb image', |
|  | platform: 'ios', |
|  | render(): React.Element<any> { |
|  | return <SliderExample thumbImage={require('./uie\_thumb\_big.png')} />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Custom track image', |
|  | platform: 'ios', |
|  | render(): React.Element<any> { |
|  | return <SliderExample trackImage={require('./slider.png')} />; |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Custom min/max track image', |
|  | platform: 'ios', |
|  | render(): React.Element<any> { |
|  | return ( |
|  | <SliderExample |
|  | minimumTrackImage={require('./slider-left.png')} |
|  | maximumTrackImage={require('./slider-right.png')} |
|  | /> |
|  | ); |
|  | }, |
|  | }, |
|  | { |
|  | title: 'Inverted slider direction', |
|  | platform: 'ios', |
|  | render(): React.Element<any> { |
|  | return <SliderExample value={0.6} inverted />; |
|  | }, |
|  | }, |
|  | ]; |



WebView

VD:

1. **import** React, { Component } from 'react'
2. **import** {
3. View,WebView,StyleSheet,AppRegistry
4. } from 'react-native'
6. export **default** **class** ActivityIndicatorDemo **extends** Component {
7. render() {
8. **return** (
9. <View style = {styles.container}>
10. <WebView
11. source = {{ uri:'https://www.javatpoint.com' }}
12. />
13. </View>
14. )
15. }
16. }
17. **const** styles = StyleSheet.create({
18. container: {
19. flex: 1,
20. }
21. })
23. AppRegistry.registerComponent('App', () => ActivityIndicatorDemo)



ViewPager

VD:

import ViewPager from "@react-native-community/viewpager";

class MyPager extends React.Component {

render() {

return (

<ViewPager

style={styles.viewPager}

initialPage={0}>

<View key="1">

<Text>First page</Text>

</View>

<View key="2">

<Text>Second page</Text>

</View>

</ViewPager>

);

}

}

const styles = StyleSheet.create({

viewPager: {

flex: 1

},

})

Code demo của Lê Quốc Việt

*danhsachSV.js*

import React, { Component } from 'react';

import {FlatList, StyleSheet, Text, View, TextInput, Button, Alert } from 'react-native';

import SinhVien from './SinhVien';

var arraySinhVien = [];

// class user {

// constructor(name, phone){

// this.name = name;

// this.phone = phone;

// }

// }

// class user {

// constructor (name, phone)

// {

// this.name = name;

// this.phone = phone;

// }

// setName (name){

// this.name = name;

// }

// getName () {

// return this.name;

// }

// setPhone (phone) {

// this.phone = phone;

// }

// getPhone (){

// return this.phone;

// }

// };

export default class DemoFlatlist extends Component {

constructor(props) {

super(props);

this.state = {name: ""};

this.state = {phone: ""};

this.state = {reload: null};

}

reloadFlat = (keyname) =>{

this.setState((prevState) => {

return {

reload: keyname

};

});

}

varTest(text) {

let x = ""

if (x== text || text == null) {

return true;

}

return false

}

render() {

return (

<View style={styles.container}>

<Text style={{fontSize: 20, marginLeft:130}}>Quản lý sinh viên</Text>

<TextInput

style={{height: 50, marginLeft: 20, marginRight: 20}}

placeholder="Name"

onChangeText={(text) => this.setState({ name: text })}

value={this.state.name}

/>

<TextInput

style={{height: 50, marginLeft: 20, marginRight: 20}}

placeholder="Phone"

onChangeText={(text) => this.setState({ phone: text })}

value={this.state.phone}

/>

<Button

style={{height: 30, margin: 20}}

title="Save"

onPress={()=>{

if(this.varTest(this.state.phone) == true || this.varTest(this.state.name) == true){

alert("Vui lòng điền đầy đủ thông tin");

return;

}

const newSV = {name: this.state.name , phone : this.state.phone}

arraySinhVien.push(newSV)

//alert("Hello "+ this.state.name);

this.reloadFlat(this.state.name);

}}

/>

<Text style={{fontSize: 16, marginLeft:130}}>Danh sách sinh viên</Text>

<FlatList

data={arraySinhVien}

renderItem={({item, index}) =>

<SinhVien name = {item.name} phone= {item.phone} parentFlatlist={this}></SinhVien>

}

/>

</View>

);

}

}

const styles = StyleSheet.create({

container: {

flex: 1,

paddingTop: 22

},

item: {

padding: 10,

fontSize: 18,

height: 44,

},

})

*SinhVien.js*

import React,{Component} from 'react';

import { TouchableOpacity, StyleSheet, Text, View, Image} from 'react-native';

var icon;

export default class SanPham extends Component{

render(){

return (

<View style={styles.product}>

<View style={styles.containerContent}>

<Text> Họ và tên : {this.props.name} </Text>

<Text> Số điện thoại : {this.props.phone} </Text>

</View>

</View>

);

}

}

const styles = StyleSheet.create({

product: {

justifyContent: 'center',

//alignItems: 'center',

backgroundColor: 'lightgreen',

width: 410,

height: 70,

marginBottom : 10

},

containerContent: {

marginTop:-20,

marginLeft: 20

},

});

Code demo các component của Ngô Quốc Cường

*DBConfig.php*

<?php

//Define your host here.

$HostName = "localhost";

//Define your database name here.

$DatabaseName = "mangareader";

//Define your database username here.

$HostUser = "root";

//Define your database password here.

$HostPass = "";

?>

*Select\_manga.php*

<?php

include 'DBConfig.php';

$conn = new mysqli($HostName, $HostUser, $HostPass, $DatabaseName);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT \* FROM manga";

$result = $conn->query($sql);

if ($result->num\_rows >0) {

while($row[] = $result->fetch\_assoc()) {

$tempt = $row;

$json = json\_encode($tempt);

}

} else {

echo "No Results Found.";

}

echo $json;

$conn->close();

?>

*Homepage.js*

import React, { Component } from 'react';

import {

StyleSheet, Text, View, Image,FlatList,

TouchableWithoutFeedback, StatusBar,

TextInput, SafeAreaView, Keyboard, TouchableOpacity,

KeyboardAvoidingView, Button, ToolbarAndroid

} from 'react-native';

import Manga from './itemManga';

export default class homepage extends Component {

// static navigationOptions = {

// header: null

// };

constructor(props) {

super(props);

this.state = {

keyword: "",

isLoading: true

};

}

componentDidMount(){

return fetch('http://10.0.3.2/Select\_Manga.php')

.then((response) => response.json())

.then((responseJson) => {

this.setState({

isLoading: false,

dataSource: responseJson,

}, function(){

});

})

.catch((error) =>{

console.error(error);

});

}

render() {

return (

<View style={styles.container}>

<Text style={{fontSize: 20, marginLeft:130}}>Manga Home</Text>

<TextInput

style={styles.textInput}

placeholder="Name"

onChangeText={(text) => this.setState({ keyword: text })}

value={this.state.keyword}

/>

<TouchableOpacity

style = { styles.TouchableOpacityStyle }

onPress={() => this.goSearch()}>

<Text>Search</Text>

</TouchableOpacity>

<FlatList style={{marginTop:10}}

data={this.state.dataSource}

renderItem={({item}) => <Manga tenMG={item.manga\_name} desMG={item.manga\_des} img={item.image}/>}

/>

</View>

)

}

goSearch()

{

alert("go to search");

}

}

const styles = StyleSheet.create({

container:{

flex: 1,

padding: 20,

},

textInput:{

height: 40,

marginLeft: 15,

marginRight: 15,

marginTop: 10,

borderColor: '#009688',

borderWidth: 1,

borderRadius: 20

},

TouchableOpacityStyle:

{

marginLeft:310,

marginTop:-40,

width: '10%'

},

})

*Itemmanga.js*

import React, { Component } from 'react';

import {

TouchableOpacity,

StyleSheet,

Text,

View,

Image,

} from 'react-native';

export default class itemManga extends Component{

constructor(props){

super(props);

this.state={}

}

render() {

return (

<View style={styles.manga}>

<View style={styles.containerImg}>

<View>

<Image style={styles.img} source={{uri: this.props.img}} />

</View>

</View>

<View style={styles.containerTenMG}>

<Text style={{fontWeight: 'bold'}}> {this.props.tenMG} </Text>

</View>

<View style={styles.containerDesMG}>

<Text > {this.props.desMG} </Text>

</View>

</View>

);

}

}

const styles = StyleSheet.create({

manga: {

justifyContent: 'center',

//alignItems: 'center',

backgroundColor: '#DDDDDD',

width: 300,

height: 110,

marginBottom : 10,

borderRadius: 4,

borderWidth: 0.5,

},

idMG:{

marginTop: 300,

},

img: {

marginTop: -50,

width: 70,

height:90,

},

containerImg: {

width: 100,

height: 150,

marginLeft: 10

},

containerTenMG: {

marginTop: -200,

marginLeft: 90,

},

containerDesMG: {

marginTop:0,

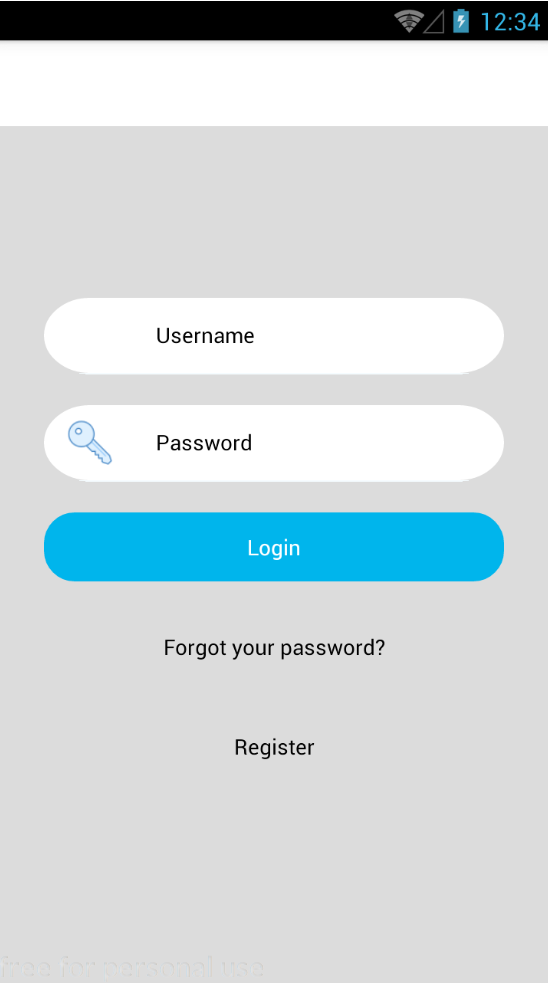
marginLeft: 90

},

});

CHƯƠNG 4. PROJECT

4.1: Màn hình đăng nhập (Việt làm)



*Login.js*

import React, { Component } from 'react';

import {

    StyleSheet, Text, View, Image,

    TouchableWithoutFeedback, StatusBar,

    TextInput, SafeAreaView, Keyboard, TouchableOpacity,

    KeyboardAvoidingView,TouchableHighlight,AsyncStorage,

} from 'react-native';

export default class LoginScreen extends Component {

    constructor(props) {

        super(props);

        this.state = {

            password: "",

            username: "",

        };

    }

        render() {

      return (

          <View style={styles.container}>

        <View style={styles.inputContainer}>

          <Image style={styles.inputIcon} source={{uri: 'https://image.flaticon.com/icons/svg/1828/1828453.svg'}}/>

          <TextInput style={styles.inputs}

              placeholder="Username"

              keyboardType="email-address"

              underlineColorAndroid='transparent'

              onChangeText={(username) => this.setState({username})}/>

        </View>

        <View style={styles.inputContainer}>

        <Image style={styles.inputIcon} source={{uri: 'https://png.icons8.com/key-2/ultraviolet/50/3498db'}}/>

        <TextInput style={styles.inputs}

            placeholder="Password"

            secureTextEntry={true}

            underlineColorAndroid='transparent'

            onChangeText={(password) => this.setState({password})}/>

      </View>

      <TouchableHighlight style={[styles.buttonContainer, styles.loginButton]} onPress={() => this.login()}>

          <Text style={styles.loginText}>Login</Text>

        </TouchableHighlight>

        <TouchableHighlight style={styles.buttonContainer} onPress={() => this.onClickListener('restore\_password')}>

            <Text>Forgot your password?</Text>

        </TouchableHighlight>

        <TouchableHighlight style={styles.buttonContainer} onPress={() => this.props.navigation.navigate('SignUp')}>

            <Text>Register</Text>

        </TouchableHighlight>

      </View>

      );

    }

    login() {

        fetch('http://10.0.3.2/login.php', {

            method: 'POST',

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json',

            },

            body: JSON.stringify({

                username: this.state.username,

                password: this.state.password,

            }),

        })

            .then((response) => response.json())

            .then((responseJson) => {

                if (responseJson == "true") {

                    alert("Successfully Login");

                    this.props.navigation.navigate('Home');

                } else {

                    alert("Login Failed");

                    return (false);

                }

            })

            .catch((error) => {

                console.log(error);

            })

    }

}

const styles = StyleSheet.create({

    container: {

      flex: 1,

      justifyContent: 'center',

      alignItems: 'center',

      backgroundColor: '#DCDCDC',

    },

    inputContainer: {

        borderBottomColor: '#F5FCFF',

        backgroundColor: '#FFFFFF',

        borderRadius:30,

        borderBottomWidth: 1,

        width:300,

        height:50,

        marginBottom:20,

        flexDirection: 'row',

        alignItems:'center'

    },

    inputs:{

        height:50,

        marginLeft:16,

        borderBottomColor: '#FFFFFF',

        flex:1,

    },

    inputIcon:{

      width:30,

      height:30,

      marginLeft:15,

      justifyContent: 'center'

    },

    buttonContainer: {

      height:45,

      flexDirection: 'row',

      justifyContent: 'center',

      alignItems: 'center',

      marginBottom:20,

      width:300,

      borderRadius:20,

    },

    loginButton: {

      backgroundColor: "#00b5ec",

    },

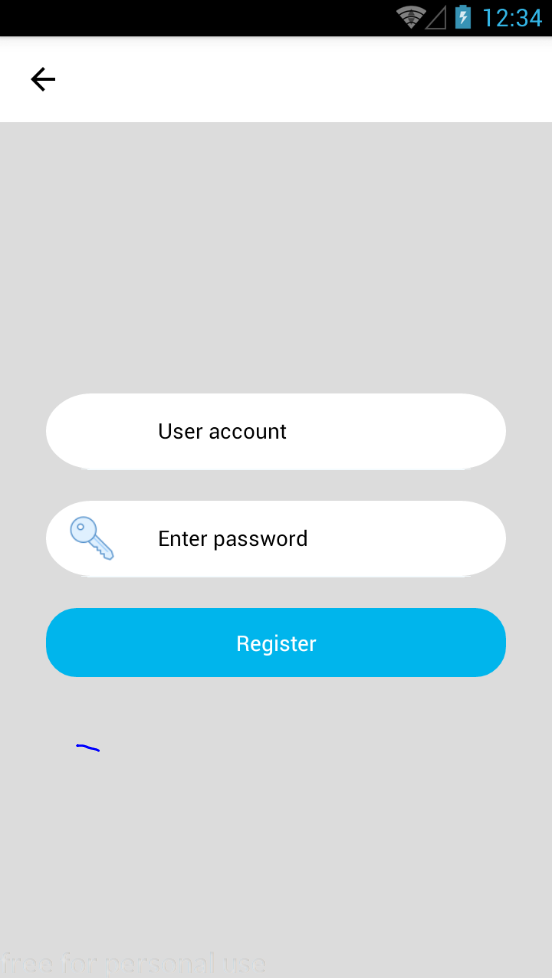
    loginText: {

      color: 'white',

    }

  });

4.2: Màn hình đăng kí (Cường làm)



*SignUp.js*

import React, { Component } from 'react';

import {

    StyleSheet,

    Text,

    View,

    TextInput,

    Button,

    TouchableHighlight,

    Image,

    Alert

  } from 'react-native';

export default class SignUp extends Component {

    constructor(props) {

        super(props);

        state = {

            username: '',

            password: '',

        }

      }

      onClickListener = (viewId) => {

        Alert.alert("Alert", "Button pressed "+viewId);

      }

    render() {

      return (

          <View style={styles.container}>

        <View style={styles.inputContainer}>

          <Image style={styles.inputIcon} source={{uri:'https://image.flaticon.com/icons/svg/1828/1828453'}}/>

          <TextInput style={styles.inputs}

              placeholder="User account"

              keyboardType="email-address"

              underlineColorAndroid='transparent'

              onChangeText={(username) => this.setState({username})}/>

        </View>

        <View style={styles.inputContainer}>

        <Image style={styles.inputIcon} source={{uri: 'https://png.icons8.com/key-2/ultraviolet/50/3498db'}}/>

        <TextInput style={styles.inputs}

            placeholder="Enter password"

            secureTextEntry={true}

            underlineColorAndroid='transparent'

            onChangeText={(password) => this.setState({password})}/>

      </View>

      <TouchableHighlight style={[styles.buttonContainer, styles.loginButton]} onPress={() => this.register()}>

          <Text style={styles.loginText}>Register</Text>

        </TouchableHighlight>

      </View>

      );

    }

    register= () => {

        {

            fetch('http://10.0.3.2/SignUp.php',

            {

                method: 'POST',

                headers:

                {

                    'Accept': 'application/json',

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify(

                {

                  username : this.state.username,

                  password : this.state.password,

                })

            }).then((response) => response.json()).then((responseJsonFromServer) =>

            {

                alert(responseJsonFromServer);

            }).catch((error) =>

            {

                console.error(error);

            });

        }

    }

}

  const styles = StyleSheet.create({

    container: {

      flex: 1,

      justifyContent: 'center',

      alignItems: 'center',

      backgroundColor: '#DCDCDC',

    },

    inputContainer: {

        borderBottomColor: '#F5FCFF',

        backgroundColor: '#FFFFFF',

        borderRadius:30,

        borderBottomWidth: 1,

        width:300,

        height:50,

        marginBottom:20,

        flexDirection: 'row',

        alignItems:'center'

    },

    inputs:{

        height:50,

        marginLeft:16,

        borderBottomColor: '#FFFFFF',

        flex:1,

    },

    inputIcon:{

      width:30,

      height:30,

      marginLeft:15,

      justifyContent: 'center'

    },

    buttonContainer: {

      height:45,

      flexDirection: 'row',

      justifyContent: 'center',

      alignItems: 'center',

      marginBottom:20,

      width:300,

      borderRadius:20,

    },

    loginButton: {

      backgroundColor: "#00b5ec",

    },

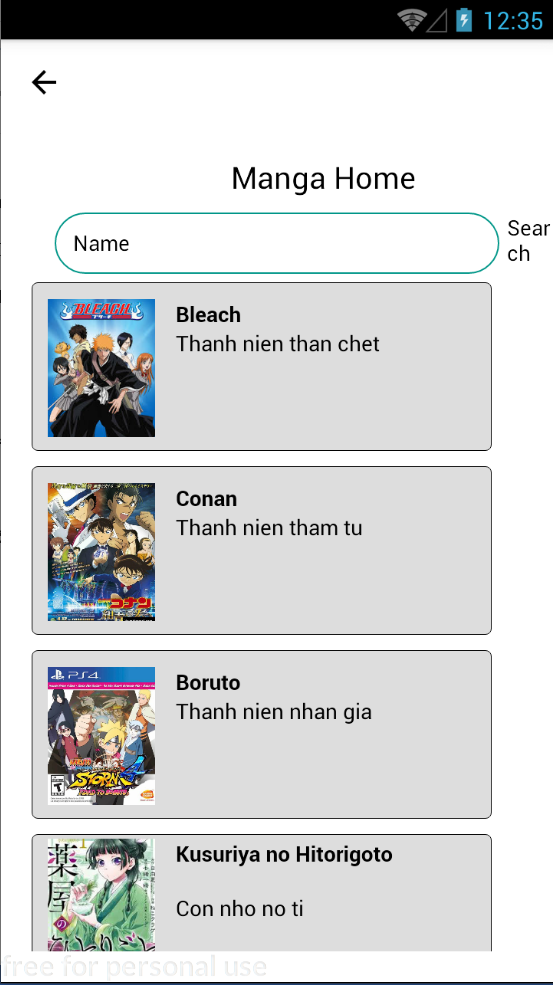
    loginText: {

      color: 'white',

    }

  });

4.3: Màn hình trang chủ (Cường làm)



*Homepage.js*

import React, { Component } from 'react';

import {

    StyleSheet, Text, View, Image,FlatList,

    TouchableWithoutFeedback, StatusBar,

    TextInput, SafeAreaView, Keyboard, TouchableOpacity,

    KeyboardAvoidingView, Button, ToolbarAndroid

} from 'react-native';

import Manga from './itemManga';

import itemManga from './itemManga';

export default class homepage extends Component {

    // static navigationOptions = {

    //     header: null

    // };

    constructor(props) {

        super(props);

        this.state = {

            keyword: "",

            isLoading: true

        };

    }

    componentDidMount(){

        return fetch('http://10.0.3.2/Select\_Manga.php')

          .then((response) => response.json())

          .then((responseJson) => {

            this.setState({

              isLoading: false,

              dataSource: responseJson,

            }, function(){

            });

          })

          .catch((error) =>{

            console.error(error);

          });

      }

    render() {

        const {navigation} = this.props;

        return  (

            <View style={styles.container}>

                <Text style={{fontSize: 20, marginLeft:130}}>Manga Home</Text>

                <TextInput

                    style={styles.textInput}

                    placeholder="Name"

                    onChangeText={(text) => this.setState({ keyword: text })}

                    value={this.state.keyword}

                />

                <TouchableOpacity

                  style = { styles.TouchableOpacityStyle }

                  onPress={() => this.goSearch()}>

                    <Text>Search</Text>

                </TouchableOpacity>

                <FlatList style={{marginTop:10}}

                data={this.state.dataSource}

                renderItem={({item}) => (

                    <TouchableOpacity onPress={() => this.props.navigation.navigate('Detail',{

                        img: item.image,

                        des: item.manga\_des,

                        name: item.manga\_name,

                        id: item.manga\_id

                    })}>

                <Manga tenMG={item.manga\_name} desMG={item.manga\_des} img={item.image} mangaid={item.manga\_id} />

                </TouchableOpacity>)}

                keyExtractor={item => item.id}

                />

            </View>

        )

    }

    goSearch()

    {

        fetch('http://10.0.3.2/Select\_search.php', {

            method: 'POST',

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json',

            },

            body: JSON.stringify({

                keyword: this.state.keyword,

            }),

        })

            .then((response) => response.json())

            .then((responseJson) => {

                this.setState({

                  isLoading: false,

                  dataSource: responseJson,

                }, function(){

                });

              })

              .catch((error) =>{

                console.error(error);

              });

    }

    }

const styles = StyleSheet.create({

    container:{

        flex: 1,

        padding: 20,

    },

    textInput:{

        height: 40,

        marginLeft: 15,

        marginRight: 15,

        marginTop: 10,

        borderColor: '#009688',

        borderWidth: 1,

        borderRadius: 20

    },

    TouchableOpacityStyle:

    {

        marginLeft:310,

        marginTop:-40,

       width: '10%'

     },

})

*itemManga.js*

import React, { Component } from 'react';

import {

  TouchableOpacity,

  StyleSheet,

  Text,

  View,

  Image,

} from 'react-native';

export default class itemManga extends Component{

    constructor(props){

      super(props);

      this.state={};

    }

    render() {

        return (

          <View style={styles.manga}>

            <View style={styles.containerImg}>

              <View>

                <Image style={styles.img} source={{uri: this.props.img}} />

              </View>

            </View>

            <View style={styles.containerTenMG}>

              <Text style={{fontWeight: 'bold'}}> {this.props.tenMG} </Text>

            </View>

            <View style={styles.containerDesMG}>

              <Text > {this.props.desMG} </Text>

            </View>

          </View>

        );

      }

}

const styles = StyleSheet.create({

    manga: {

      justifyContent: 'center',

      //alignItems: 'center',

      backgroundColor: '#DDDDDD',

      width: 300,

      height: 110,

      marginBottom : 10,

      borderRadius: 4,

      borderWidth: 0.5,

    },

    idMG:{

      marginTop: 300,

    },

    img: {

      marginTop: -50,

      width: 70,

      height:90,

    },

    containerImg: {

      width: 100,

      height: 150,

      marginLeft: 10

    },

    containerTenMG: {

      marginTop: -200,

      marginLeft: 90,

    },

    containerDesMG: {

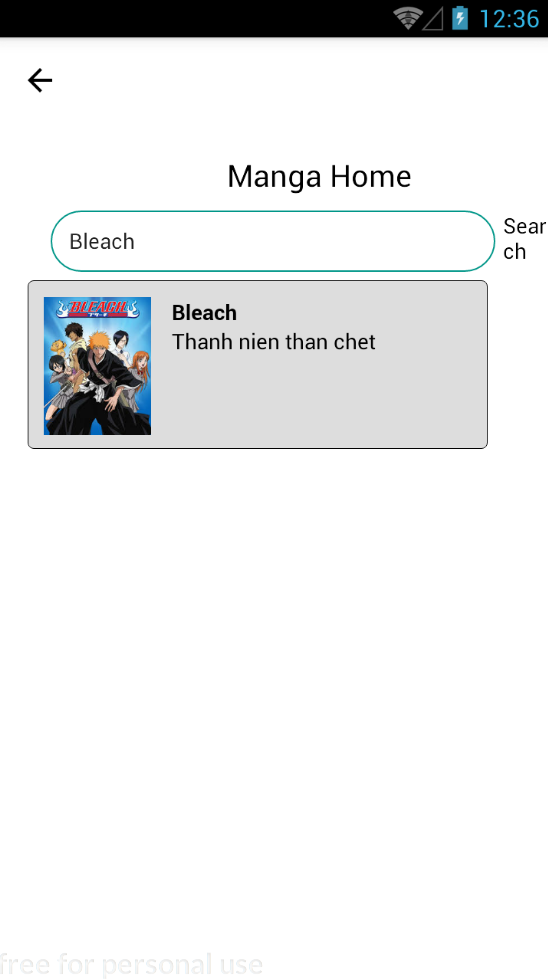
      marginTop:0,

      marginLeft: 90

    },

  });

4.4: Màn hình tìm kiếm (Việt làm)



*Homepage.js*

import React, { Component } from 'react';

import {

    StyleSheet, Text, View, Image,FlatList,

    TouchableWithoutFeedback, StatusBar,

    TextInput, SafeAreaView, Keyboard, TouchableOpacity,

    KeyboardAvoidingView, Button, ToolbarAndroid

} from 'react-native';

import Manga from './itemManga';

import itemManga from './itemManga';

export default class homepage extends Component {

    // static navigationOptions = {

    //     header: null

    // };

    constructor(props) {

        super(props);

        this.state = {

            keyword: "",

            isLoading: true

        };

    }

    componentDidMount(){

        return fetch('http://10.0.3.2/Select\_Manga.php')

          .then((response) => response.json())

          .then((responseJson) => {

            this.setState({

              isLoading: false,

              dataSource: responseJson,

            }, function(){

            });

          })

          .catch((error) =>{

            console.error(error);

          });

      }

    render() {

        const {navigation} = this.props;

        return  (

            <View style={styles.container}>

                <Text style={{fontSize: 20, marginLeft:130}}>Manga Home</Text>

                <TextInput

                    style={styles.textInput}

                    placeholder="Name"

                    onChangeText={(text) => this.setState({ keyword: text })}

                    value={this.state.keyword}

                />

                <TouchableOpacity

                  style = { styles.TouchableOpacityStyle }

                  onPress={() => this.goSearch()}>

                    <Text>Search</Text>

                </TouchableOpacity>

                <FlatList style={{marginTop:10}}

                data={this.state.dataSource}

                renderItem={({item}) => (

                    <TouchableOpacity onPress={() => this.props.navigation.navigate('Detail',{

                        img: item.image,

                        des: item.manga\_des,

                        name: item.manga\_name,

                        id: item.manga\_id

                    })}>

                <Manga tenMG={item.manga\_name} desMG={item.manga\_des} img={item.image} mangaid={item.manga\_id} />

                </TouchableOpacity>)}

                keyExtractor={item => item.id}

                />

            </View>

        )

    }

    goSearch()

    {

        fetch('http://10.0.3.2/Select\_search.php', {

            method: 'POST',

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json',

            },

            body: JSON.stringify({

                keyword: this.state.keyword,

            }),

        })

            .then((response) => response.json())

            .then((responseJson) => {

                this.setState({

                  isLoading: false,

                  dataSource: responseJson,

                }, function(){

                });

              })

              .catch((error) =>{

                console.error(error);

              });

    }

    }

const styles = StyleSheet.create({

    container:{

        flex: 1,

        padding: 20,

    },

    textInput:{

        height: 40,

        marginLeft: 15,

        marginRight: 15,

        marginTop: 10,

        borderColor: '#009688',

        borderWidth: 1,

        borderRadius: 20

    },

    TouchableOpacityStyle:

    {

        marginLeft:310,

        marginTop:-40,

       width: '10%'

     },

})

*ItemManga.js*

import React, { Component } from 'react';

import {

  TouchableOpacity,

  StyleSheet,

  Text,

  View,

  Image,

} from 'react-native';

export default class itemManga extends Component{

    constructor(props){

      super(props);

      this.state={};

    }

    render() {

        return (

          <View style={styles.manga}>

            <View style={styles.containerImg}>

              <View>

                <Image style={styles.img} source={{uri: this.props.img}} />

              </View>

            </View>

            <View style={styles.containerTenMG}>

              <Text style={{fontWeight: 'bold'}}> {this.props.tenMG} </Text>

            </View>

            <View style={styles.containerDesMG}>

              <Text > {this.props.desMG} </Text>

            </View>

          </View>

        );

      }

}

const styles = StyleSheet.create({

    manga: {

      justifyContent: 'center',

      //alignItems: 'center',

      backgroundColor: '#DDDDDD',

      width: 300,

      height: 110,

      marginBottom : 10,

      borderRadius: 4,

      borderWidth: 0.5,

    },

    idMG:{

      marginTop: 300,

    },

    img: {

      marginTop: -50,

      width: 70,

      height:90,

    },

    containerImg: {

      width: 100,

      height: 150,

      marginLeft: 10

    },

    containerTenMG: {

      marginTop: -200,

      marginLeft: 90,

    },

    containerDesMG: {

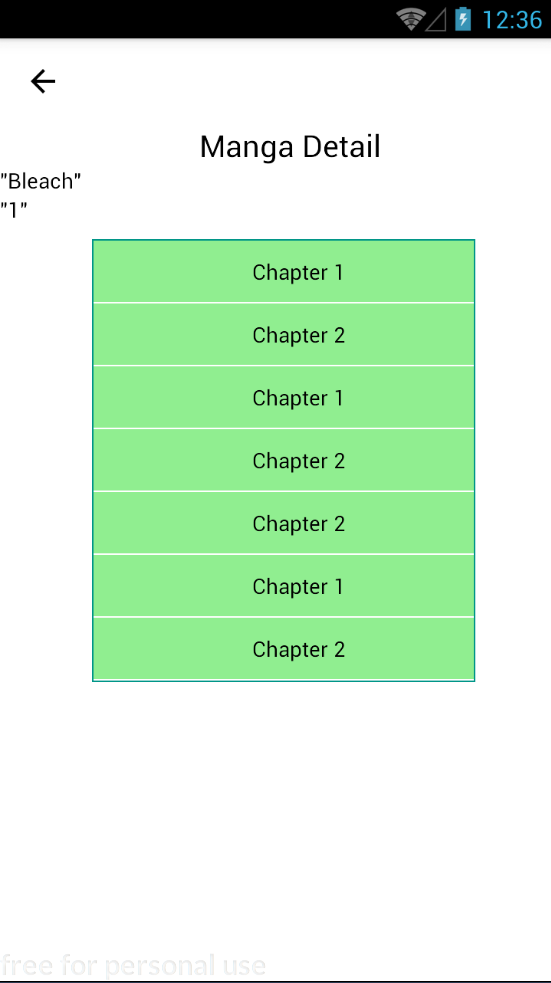
      marginTop:0,

      marginLeft: 90

    },

  });

4.5: Màn hình chi tiết trang truyện ( Cường làm)



*MangaDetail.js*

import React, { Component } from 'react';

import {FlatList, StyleSheet, Text, View, TextInput, Button, Alert, Image, TouchableOpacity } from 'react-native';

import Chapter from './itemChapter';

var arrayChapter = ["Chapter 1", "Chapter 2", "Chapter 1", "Chapter 2", "Chapter 2", "Chapter 1", "Chapter 2"]

export default class MangaDetail extends Component {

    constructor(props) {

        super(props);

        this.state = {

            search: "",

            dataSource: {}

            };

          })

          .catch((error) =>{

            console.error(error);

          });

          console.log('dataSource',this.state.dataSource)

      }

    render(){

        const{ navigation} = this.props;

        return (

            <View style={styles.Container}>

                <Text style={{fontSize: 20, marginLeft:130}}>Manga Detail</Text>

                <View style={styles.search}>

                </View>

                <Text >

                 {JSON.stringify(navigation.getParam('name'))}

                </Text>

                <Text >

                 {JSON.stringify(navigation.getParam('id'))}

                </Text>

                <FlatList style={styles.chaplist}

                    data={arrayChapter}

                    renderItem={({item, index}) =>

                    <Chapter Chapter = {item}></Chapter>

          }

        />

            </View>

        );

    }

}

const styles = StyleSheet.create({

    Container:{

        justifyContent: 'center',

    },

    img:{

        marginTop:20,

        marginLeft: 50,

        width:300,

        height:300,

    },

    dec:{

        marginLeft: 30,

        marginRight: 30,

        marginTop: 10,

    },

    search:{

    },

    chaplist:{

        marginLeft: 60,

        marginTop: 10,

        marginRight: 50,

        marginBottom:10,

        borderWidth: 1,

        borderColor: '#009688'

    },

    textInput:{

        height: 40,

        marginLeft: 15,

        marginRight: 15,

        marginTop: 10,

        borderColor: '#009688',

        borderWidth: 1,

        borderRadius: 20

    },

    TouchableOpacityStyle:

    {

        marginLeft:310,

        marginTop:-40,

       width: '10%'

     },

})

*itemChapter.js*

import React,{Component} from 'react';

import { TouchableOpacity, StyleSheet, Text, View, Image} from 'react-native';

var icon;

export default class Chapter extends Component{

    render(){

        return (

            <View style={styles.itemChap}>

                <View style={styles.containerContent}>

                    <Text> {this.props.Chapter} </Text>

                </View>

            </View>

        );

    }

}

const styles = StyleSheet.create({

    itemChap: {

      justifyContent: 'center',

      //alignItems: 'center',

      backgroundColor: 'lightgreen',

      width:300,

      height: 40,

      marginBottom : 1

    },

    containerContent: {

      marginLeft: 100

    },

  });

CHƯƠNG 5. BÁO CÁO KIẾN NGHỊ