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 |  |  |  |  |  |
| 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

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 React 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.

React Native đó chính là chúng ta chỉ cần sử dụng JS để phát triển được một ứng dụng di động hoàn chỉnh.

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.

## Kiến thức cơ bản về ES6 trong React native

**ES6 l**à phiên bản mới nhất của bộ tiêu chuẩn **ECMAScript** – một bộ đặc tả tiêu chuẩn dành cho Javascript do Hiệp hội các nhà sản xuất máy tính Châu Âu (European Computer Manufacturers Association – ECMA) đề xuất.

## CÁC CHỨC NĂNG MỚI CỦA ES6

### ARROW

Arrow là một dạng viết tắt của các function sử dụng dấu =>, tương tự như trong C#, Java 8,…

// Expression bodies

var odds = evens.map(v => v + 1);

var nums = evens.map((v, i) => v + i);

var pairs = evens.map(v => ({even: v, odd: v + 1}));

// Statement bodies

nums.forEach(v => {

if (v % 5 === 0)

fives.push(v);

});

// Lexical this

var bob = {

\_name: "Bob",

\_friends: [],

printFriends() {

this.\_friends.forEach(f =>

console.log(this.\_name + " knows " + f));

}

}

**CLASS**

Đối với Javascript truyền thống, để sử khai báo và kế thừa các class, chúng ta phải thiết kế theo hướng sử dụng Prototype (prototype-based OO). Việc khai báo và kế thừa các class trong ES6 dễ hơn rất nhiều với cú pháp gần giống với Java và C++, ngoài ra, class trong ES6 cũng hỗ trợ kế thừa thông qua prototype, các static method, constructor,…

class SkinnedMesh extends THREE.Mesh {

constructor(geometry, materials) {

super(geometry, materials);

this.idMatrix = SkinnedMesh.defaultMatrix();

this.bones = [];

this.boneMatrices = [];

//...

}

update(camera) {

//...

super.update();

}

get boneCount() {

return this.bones.length;

}

set matrixType(matrixType) {

this.idMatrix = SkinnedMesh[matrixType]();

}

static defaultMatrix() {

return new THREE.Matrix4();

}

}

### XỬ LÝ CHUỖI

Xử lý chuỗi trong ES6 đã trở nên dễ dàng và tiện dụng hơn rất nhiều, mang hơi hướng của các ngôn ngữ như Python, Perl,… đặc biệt, hỗ trợ chuỗi nhiều dòng, đây có lẽ là một cải tiến khiến rất nhiều người cảm thấy thích thú.

// Basic literal string creation

`In JavaScript 'n' is a line-feed.`

// Multiline strings

`In JavaScript this is

not legal.`

// String interpolation

var name = "Bob", time = "today";

`Hello ${name}, how are you ${time}?`

### GIÁ TRỊ DEFAULT CHO THAM SỐ

Ở phiên bản này, Javascript đã có thể sử dụng các giá trị mặc định cho tham số truyền vào các hàm như những ngôn ngữ lập trình khác như C++, C#.

function f(x, y=12) {

// y = 12 nếu không truyền giá trị cho nó (hoặc truyền undefined)

return x + y;

}

f(3) == 15

### TRUYỀN THAM SỐ KHÔNG XÁC ĐỊNH SỐ LƯỢNG

Việc này trước đây có thể thực hiện thông qua biến arguments có trong từng hàm, nhưng giờ đây chúng ta có thể sử dụng nó một cách linh hoạt hơn rất nhiều.

function f(x, ...y) {

// y là một mảng

return x \* y.length;

}

f(3, "hello", true) == 6

### TRUYỀN THAM SỐ THÔNG QUA TỪNG PHẦN TỬ CỦA MẢNG

Với kĩ thuật này, bạn có thể truyền một mảng hoặc một đối tượng vào một hàm, các phần tử của mảng/đối tượng này sẽ được tự động truyền vào thành các tham số của hàm đó

function f(x, y, z) {

return x + y + z;

}

// Pass each elem of array as argument

f(...[1,2,3]) == 6

### TỪ KHOÁ LET VÀ CONST

const, đúng như tên gọi của nó, là cách khai báo hằng số, một hằng số thì không thể thay đổi giá trị được.

const x = 10;

x = 5; // Lỗi

let cũng là một dạng khai báo biến giống như var, tuy nhiên, biến được định nghĩa bằng từ khoá let có phạm vi truy cập khép kín trong khối lệnh chứa nó.

function testLet() {

// a \*không\* truy cập được tại đây

for( let a = 0; a < 5; a++ ) {

// a chỉ truy cập được trong này

};

// a \*không\* truy cập được tại đây

};

### MODULES

// lib/math.js

export function sum(x, y) {

return x + y;

}

export var pi = 3.141593;

// app.js

import \* as math from "lib/math";

alert("2π = " + math.sum(math.pi, math.pi));

// otherApp.js

import {sum, pi} from "lib/math";

alert("2π = " + sum(pi, pi));

### MAP, SET, WEAKMAP, WEAKSET

ES6 còn giới thiệu thêm một số kiểu dữ liệu mới để hỗ trợ chúng ta thực hiện các thuật toán phức tạp hơn.

// Sets

var s = new Set();

s.add("hello").add("goodbye").add("hello");

s.size === 2;

s.has("hello") === true;

// Maps

var m = new Map();

m.set("hello", 42);

m.set(s, 34);

m.get(s) == 34;

// Weak Maps

var wm = new WeakMap();

wm.set(s, { extra: 42 });

wm.size === undefined

// Weak Sets

var ws = new WeakSet();

ws.add({ data: 42 });

## Component trong react native

chính là 1 trong những lí do khiến ứng dụng viết bằng React/React Native có khả năng tái sử dụng và tăng khả năng mở rộng. Đây là điểm mà tôi rất thích ở React, vì nó buộc bạn phải nghĩ đến việc thiết kế và lọc ra những thành phần chung trong một ứng dụng thay vì ném mọi thứ vào 1 cái container view.

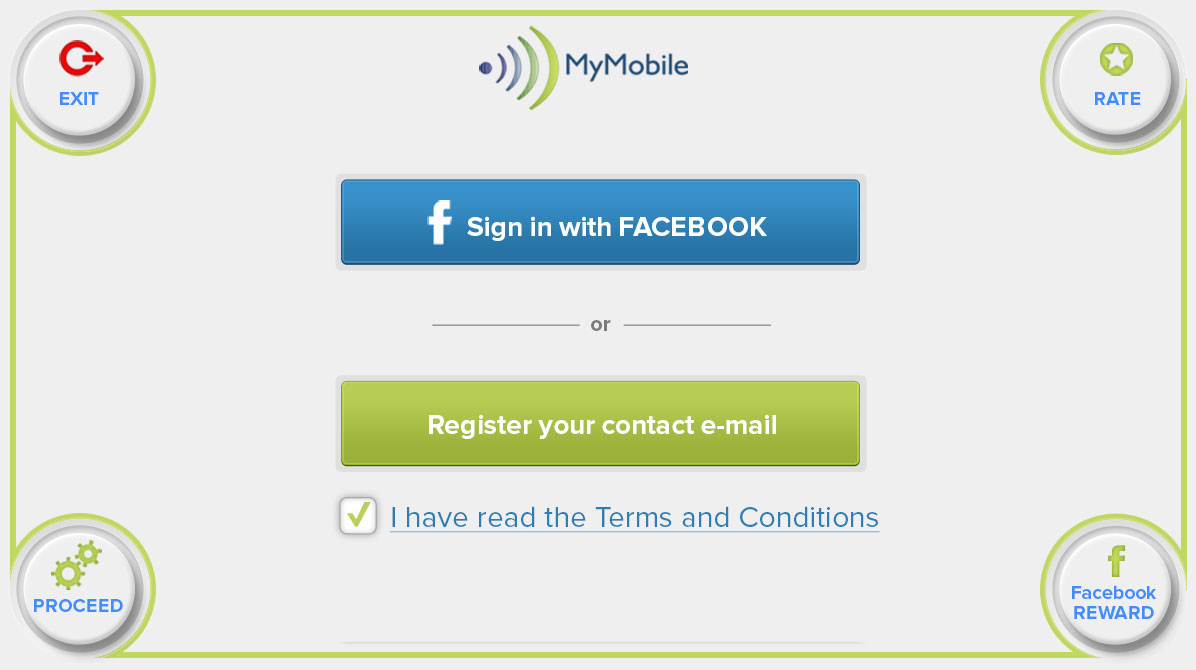
## 4. API Trong React Native

# 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 User Interfaces



*Image1: Registration*

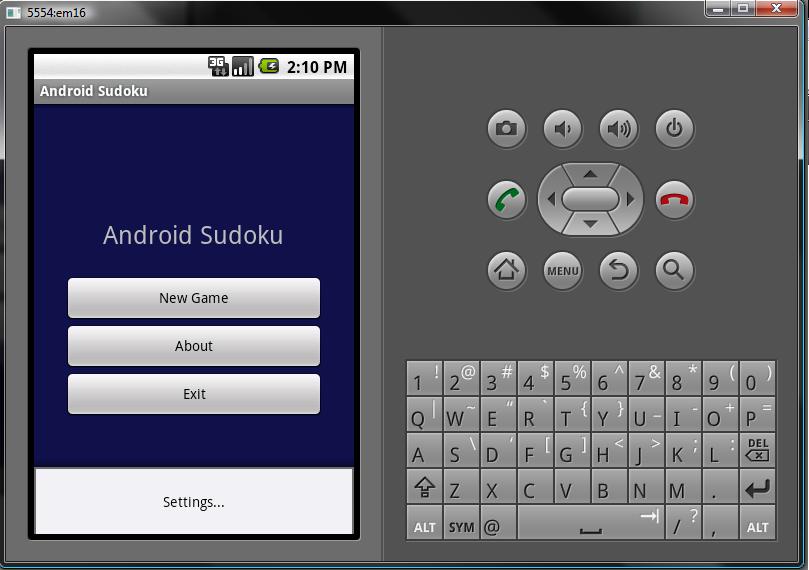
#### 2.1.1.2. Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Action** | **Response** |
| **MyMobile**  **Registration screen** | MyMobile Register Screen has 6 buttons:   * Sign in with Facebook * Register your contact e-mail * Proceed * Rate * Exit * Facebook REWARD   Screen has a "I have read the Terms and Conditions" link and a check box | N/A | N/A |
| **“I have read the Terms and Conditions”Link** | Use to display a pop up and have a check box | Tap on the Link | A pop up is showed with " Standard Terms and Conditions outlining Device Asset Management System role as an introducer with no lianbity here, plus prices and values are for guide purposes only." |
| A Check box as a condition to navigate to valuation screen | Tap on the check box | The check box is ticked |
| **Sign in with Facebook** | If the check box of "I have read the Terms and Conditions” link is ticked, a pop up come up with "In full version, this function links app to users Facebook account, allowing app to communicate directly with user via Facebook and user to post activity to Facebook." and a proceed button is embedded in pop up. | Tap on Sign in with Facebook button | A pop up is showed |
| Proceed button to navigate to valuation screen | Tap on the button | Navigate to Valuation screen |
| If the check box of "I have read the Terms and Conditions" link is not ticked, a pop up come up with “Please check Terms and Conditions first.” with OK button |  |  |
| **Register your contact e-mail** | If the check box of "I have read the Terms and Conditions" link is ticked, A pop up is opened for user to enter email With:   * E-mail field * Confirm E-mail field * Proceed button * Cancel Button | Tap on the button | A pop up is showed for user to enter |
| If user does not enter email or confirm email shows a pop up “ E-mail or confirm E-mail is empty, please fill out them!”  If E-mail or confirm E-mail is incorrect, A pop up is showed “E-mail or confirm E-mail is incorrect, please try again!” | N/A | N/A |
| Proceed button to navigate to valuation screen | Tap on the button | Navigate to Valuation screen |
| Cancel button to navigate to valuation screen | Tap on the button | Pop up is closed |
| If the check box of "I have read the Terms and Conditions" link is not ticked, a pop up come up with “Terms and Conditions Box with I agree.” | Tap on Sign in with Facebook button | A pop up is showed.  Note: then back to the main screen with the Terms & Conditions box now ticked |
| **Proceed** | Using to open a pop up comes up with "Before you use the great functions of this app, please either register a e-mail address or connect with Facebook so we can send you shopping voucher rewards in the future and stay in touch." and Cancel button | Tap on Proceed button | A pop up is showed |
| Cancel button to navigate to valuation screen | Tap on the button | Pop up is closed |
| **Rate** | Using to open a pop up comes up with "Before you use the great functions of this app, please either register a e-mail address or connect with Facebook so we can send you shopping voucher rewards in the future and stay in touch." and Cancel button | Tap on Rate button | A pop up is showed |
| Cancel button to navigate to valuation screen | Tap on the button | Pop up is closed |
| **Exit** | Closed application | Tap on Exit button | Application is closed |
| **Facebook REWARD** | Using to open a pop up comes up with "Before you use the great functions of this app, please either register a e-mail address or connect with Facebook so we can send you shopping voucher rewards in the future and stay in touch." | Tap on Facebook RE-WARD button | A pop up is showed |
| **Login policy** | Each time the App is started is goes to the Registration Screen | N/A | N/A |

## 2.2 Thiết kế hệ thống

### 2.2.1. Sudoku Main Screen

#### 2.2.1.1. Screen Shot for Sudoku Main Screen

****

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

Objects:

* New Game
* About
* Exit
* Settings

Actions:

* Start a new game
* View the about page
* Exit to the Home Screen
* Set setting for music and hints

# 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