**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**Viện Kỹ thuật – Công nghệ**

****

**TIỂU LUẬN MÔN HỌC**

**THỰC TẬP DOANH NGHIỆP**

**ỨNG DỤNG DI ĐỘNG SỬ DỤNG**

**REACT NATIVE**

**GVHD: Th.S TRẦN VĂN HỮU**

**SVTH:**

**NGUYỄN PHẠM MINH TRIẾT 1924801030011**

**TRẦN ĐÌNH PHƯƠNG 1924801030242**

**NGUYỄN TIẾN ĐẠT 1924801030035**

**LỚP: D19PM02**

**BÌNH DƯƠNG – 12/2020**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**Viện Kỹ thuật – Công nghệ**

****

**TIỂU LUẬN MÔN HỌC**

**THỰC TẬP DOANH NGHIỆP**

**ỨNG DỤNG DI ĐỘNG SỬ DỤNG**

**REACT NATIVE**

**GVHD: Th.S TRẦN VĂN HỮU**

**SVTH:**

**NGUYỄN PHẠM MINH TRIẾT 1924801030011**

**TRẦN ĐÌNH PHƯƠNG 1924801030242**

**NGUYỄN TIẾN ĐẠT 1924801030035**

**LỚP: D19PM02**

**BÌNH DƯƠNG – 12/2020**

MỤC LỤC

[MỤC LỤC ii](#_Toc59014987)

[DANH MỤC HÌNH ii](#_Toc59014988)

[MỞ ĐẦU 1](#_Toc59014989)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ 2](#_Toc59014990)

[1.1. CÔNG NGHỆ SỬ DỤNG 2](#_Toc59014991)

[1.1.1. Giới thiệu về JavaScript 2](#_Toc59014992)

[1.1.2. Giới thiệu về React Native 2](#_Toc59014993)

[1.1.3. Giới thiệu về Firebase 3](#_Toc59014994)

[1.1.4. Giới thiệu về Firebase Realtime Database 3](#_Toc59014995)

[CHƯƠNG 2. GIỚI THIỆU ỨNG DỤNG 5](#_Toc59014996)

[2.1. Giao diện ứng dụng 5](#_Toc59014997)

[2.1.1. Giao diện trang đăng nhập 5](#_Toc59014998)

[2.1.2. Giao diện trang đăng ký 5](#_Toc59014999)

[2.1.3. Giao diện trang TodoList 6](#_Toc59015000)

[2.1.4. Giao diện thông báo nhập sai 6](#_Toc59015001)

[2.2. Các chức năng chính của ứng dụng 6](#_Toc59015002)

[CHƯƠNG 3. THIẾT KẾ CHƯƠNG TRÌNH 7](#_Toc59015003)

[3.1. Thiết kế CSDL 7](#_Toc59015004)

[3.2. Code chương trình 7](#_Toc59015005)

[KẾT LUẬN 14](#_Toc59015006)

[**1.** **Kết quả đạt được** 14](#_Toc59015007)

[**2.** **Hướng phát triển của đề tài** 14](#_Toc59015008)

[TÀI LIỆU THAM KHẢO 15](#_Toc59015009)

DANH MỤC HÌNH

[Hình 2.1 Trang đăng nhập 5](#_Toc59000553)

[Hình 2.2 Trang đăng ký 5](#_Toc59000554)

[Hình 2.3 Trang TodoList 6](#_Toc59000555)  
[Hình 2.4 Thông báo nhập sai 6](#_Toc59000555)

MỞ ĐẦU

Hiện nay việc sử dụng điện thoại di động ngày càng tăng cao, sự phát triển của cả hai nền tảng là Android và IOS. React Native cho phép phát triển ứng dụng chạy được trên hai nền tảng này nên hiện tại đang rất được ưa chuộng.

Ứng dụng TodoList được phát triển bằng React Native gồm các giao diện đơn giản với các trang: đăng nhập, đăng ký và TodoList. Các chức năng thiết kế đơn giản, dễ thao tác. Dữ liệu của TodoList được quản lý bằng Firebase nên các thông tin của TodoList được lưu trữ trên Firebase theo thời gian thực.

1. TỔNG QUAN VỀ CÔNG NGHỆ
   1. CÔNG NGHỆ SỬ DỤNG
      1. Giới thiệu về JavaScript

JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

JavaScript được biết đến đầu tiên với tên Mocha, và sau đó là LiveScript, nhưng công ty Netscape đã đổi tên của nó thành JavaScript, bởi vì sự phổ biến như là một hiện tượng của Java lúc bấy giờ. JavaScript xuất hiện lần đầu trong Netscape 2.0 năm 1995 với tên LiveScript. Core đa năng của ngôn ngữ này đã được nhúng vào Netscape, IE, và các trình duyệt khác.

ECMA-262 Specification định nghĩa một phiên bản chuẩn của ngôn ngữ JavaScript như sau:

* JavaScript là một ngôn ngữ chương trình thông dịch, nhẹ.
* Được thiết kế để tạo các ứng dụng mạng trung tâm.
* Bổ sung và tích hợp với Java.
* Bổ sung và tích hợp với HTML.
* Mở và đa nền tảng.

JavaScript hoặc JS sẽ giúp tăng tính tương tác trên website. Script này chạy trên các trình duyệt của người dùng thay vì trên server và thường sử dụng thư vuiên của bên thứ 3 nên có thể tăng thêm chức năng cho website mà không phải code từ đầu.

* + 1. Giới thiệu 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 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 trên cả hai nền tảng.

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…

* + 1. Giới thiệu về Firebase

Firebase là một nền tảng để phát triển ứng dụng di động và trang web, bao gồm các API đơn giản và mạnh mẽ mà không cần backend hay server.

Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

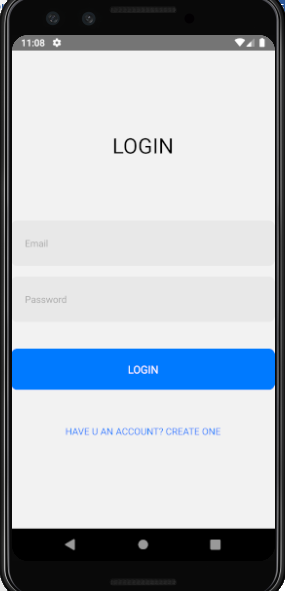
* + 1. Giới thiệu về Firebase Realtime Database

Khi đăng ký một tài khoản trên Firebase để tạo ứng dụng, bạn đã có một cơ sở dữ liệu thời gian thực. Dữ liệu bạn nhận được dưới dạng JSON. Đồng thời nó cũng luôn được đồng bộ thời gian thực đến mọi kết nối client.

Đối với các ứng dụng đa nền tảng, tất cả các client đều sử dụng cùng một cơ sở dữ liệu. Nó được tự động cập nhật dữ liệu mới nhất bất cứ khi nào các lập trình viên phát triển ứng dụng. Cuối cùng, tất cả các dữ liệu này được truyền qua kết nối an toàn SSL có bảo mật với chứng nhận 2048 bit.

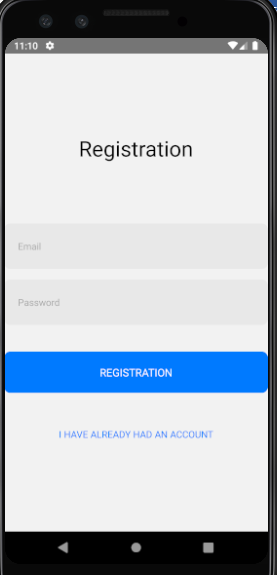
Trong trường hợp bị mất mạng, dữ liệu được lưu lại ở local, vì thế khi có mọi sự thay đổi nào đều được tự động cập nhật lên Server của Firebase. Bên cạnh đó, đối với các dữ liệu ở local cũ hơn với Server thì cũng tự động cập nhật để được dữ liệu mới nhất.

2. GIỚI THIỆU ỨNG DỤNG
   1. Giao diện ứng dụng
      1. Giao diện trang đăng nhập



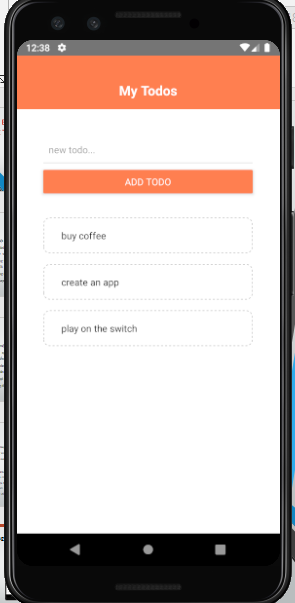
Hình 2.1 Trang đăng nhập

* + 1. Giao diện trang đăng ký



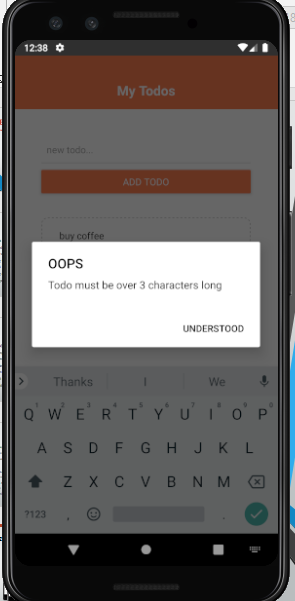
Hình 2.2 Trang đăng ký

* + 1. Giao diện trang TodoList



Hình 2.3 Trang TodoList

* + 1. Giao diện thông báo nhập sai



Hình 2.4 Thông báo nhập sai

* 1. Các chức năng chính của ứng dụng
* Đăng ký và đăng nhập
* Thêm/xóa hoạt động trong TodoList
* Báo lỗi khi người dùng nhập danh mục trong TodoList ít hơn 3 kí tự.

1. THIẾT KẾ CHƯƠNG TRÌNH
   1. Thiết kế CSDL

Dữ liệu nhập vào Firebase dưới dạng chuỗi theo thời gian thực.

Firebase

TodoList

* 1. Code chương trình

//Trang đăng nhập

import React from 'react';

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

import {Heading} from '../components/Heading';

import {Input} from '../components/Input';

import {FilledButton} from '../components/FilledButton';

import {TextButton} from '../components/TextButton';

import {Error} from '../components/Error';

export function LoginScreen({navigation}) {

return (

<View style={styles.container}>

<Heading style={styles.title}>LOGIN</Heading>

<Error error={''} />

<Input

style={styles.input}

placeholder={'Email'}

keyboardType={'email-address'}

/>

<Input

style={styles.input}

placeholder={'Password'}

secureTextEntry

/>

<FilledButton

title={'Login'}

style={styles.loginButton}

onPress={() => {

navigation.navigate('TodoList');

}}

/>

<TextButton

title={'Have u an account? Create one'}

onPress={() => {

navigation.navigate('Registration');

}}

/>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

paddingTop:120,

alignItems:'center',

},

title: {

marginBottom: 48,

},

input: {

marginVertical: 8,

},

loginButton: {

marginVertical: 32,

},

});

//Trang đăng ký

import React from 'react';

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

import {Heading} from '../components/Heading';

import {Input} from '../components/Input';

import {FilledButton} from '../components/FilledButton';

import {TextButton} from '../components/TextButton';

import {Error} from '../components/Error';

export function RegistrationScreen({navigation}) {

return (

<View style={styles.container}>

<Heading style={styles.title}>Registration</Heading>

<Error error={''} />

<Input

style={styles.input}

placeholder={'Email'}

keyboardType={'email-address'}

/>

<Input

style={styles.input}

placeholder={'Password'}

secureTextEntry

/>

<FilledButton

title={'Registration'}

style={styles.loginButton}

onPress={() => {

navigation.navigate('Login');

}}

/>

<TextButton

title={'I have already had an account'}

onPress={() => {

navigation.navigate('Login');

}}

/>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

paddingTop:120,

alignItems:'center',

},

title: {

marginBottom: 48,

},

input: {

marginVertical: 8,

},

loginButton: {

marginVertical: 32,

},

});

//Trang TodoList

import React, { useState } from 'react';

import { StyleSheet, View, FlatList, Alert, TouchableWithoutFeedback, Keyboard } from 'react-native';

import Header from '../components/header';

import TodoItem from '../components/todoItem';

import AddTodo from '../components/addTodo';

import database from '@react-native-firebase/database';

var i =0

const setData = (tt) => {

i++;

database().ref('text' + i).set({ tt }).then(() => {

console.log('Data set.')

}).catch(e => {

console.log('Set data error: ', e);

});

}

export function TodoListScreen() {

const [todos, setTodos] = useState([

{ text: 'buy coffee', key: '1' },

{ text: 'create an app', key: '2' },

{ text: 'play on the switch', key: '3' },

]);

const pressHandler = (key) => {

setTodos(prevTodos => {

return prevTodos.filter(todo => todo.key != key);

});

};

const submitHandler = (text) => {

if(text.length > 3){

setData(text);

setTodos(prevTodos => {

return [

{ text, key: Math.random().toString() },

...prevTodos

];

});

} else {

Alert.alert('OOPS', 'Todo must be over 3 characters long', [

{text: 'Understood', onPress: () => console.log('alert closed') }

]);

}

};

return (

<TouchableWithoutFeedback onPress={() => {

Keyboard.dismiss();

console.log('dismissed');

}}>

<View style={styles.container}>

<Header />

<View style={styles.content}>

<AddTodo submitHandler={submitHandler} />

<View style={styles.list}>

<FlatList

data={todos}

renderItem={({ item }) => (

<TodoItem item={item} pressHandler={pressHandler} />

)}

/>

</View>

</View>

</View>

</TouchableWithoutFeedback>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: '#fff',

},

content: {

padding: 40,

},

list: {

marginTop: 20,

},

});

KẾT LUẬN

## **Kết quả đạt được**

Triển khai được ứng dụng với trang đăng ký, đăng nhập, và trang TodoList, lưu được những công việc người dùng muốn làm theo thời gian thực.

## **Hướng phát triển của đề tài**

* Tạo cơ sở dữ liệu cho người dùng .
* Thêm các mô tả, thời gian thực hiện cho phần TodoList

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Lê Hoàng Sơn ,Nguyễn Thọ Thông(2018), Giáo trình Lập trình Android , NXB Xây dựng

**Website**

https://reactnative.dev/. Ngày truy cập 15/11/2020.

1. <https://firebase.google.com/docs>. Ngày truy cập 12/12/2020.