**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**KHOA CÔNG NGHỆ THÔNG TIN**

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



**BÁO CÁO MÔN HỌC**

Lập trình di động 3

**Đề Tài: Tạo App Bán Đồng Hồ Online**

Giảng viên hướng dẫn: Trương Bá Thái

Sinh viên thực hiện:

1. Mai Thị Quanh\_16211TT1403
2. Lưu Tuấn Tài\_16211TT1147
3. Đoàn Thị Yến Nhi\_16211TT0966

Ngành: Công nghệ thông tin

*Tp. Hồ Chí Minh*, ngày 24 tháng 10 năm 2019

**NHẬT KÝ HOẠT ĐỘNG NHÓM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Họ và tên** | **Công việc đã thực hiện** | **Tự đánh giá** | **Nhóm đánh giá** | **Chữ ký** |
| 1 | Mai Thị Quanh | -Màn hình đăng nhập  -Màn hình đăng ký  -Viết báo cáo  -Vẽ mockup  -Viết testcase |  |  |  |
| 2 | Lưu Tuấn Tài | -Màn hình trang chủ  -Thêm sản phẩm |  |  |  |
| 3 | Đoàn Thị Yến Nhi | -Màn hình trang chủ  -Mô tả SRS |  |  |  |

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 7

Thao tác cơ bản 15

Khởi tạo 15

Thêm phần tử 16

Xóa phần tử 16

Kiểm tra phần tử tồn tại 17

Đếm số phần tử 17

Xóa toàn bộ phần tử 17

Thao tác cơ bản 18

Khởi tạo 18

Thêm phần tử 18

Xóa phần tử 18

Kiểm tra phần tử tồn tại 18

Tính số phần tử 19

Vòng lặp với Set 19

Chuyển sang Array 19

Xóa toàn bộ phần tử 20

3. Component trong react native 20

4. API Trong React Native 21

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 25

CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ 59

3.1 Cài đặt 59

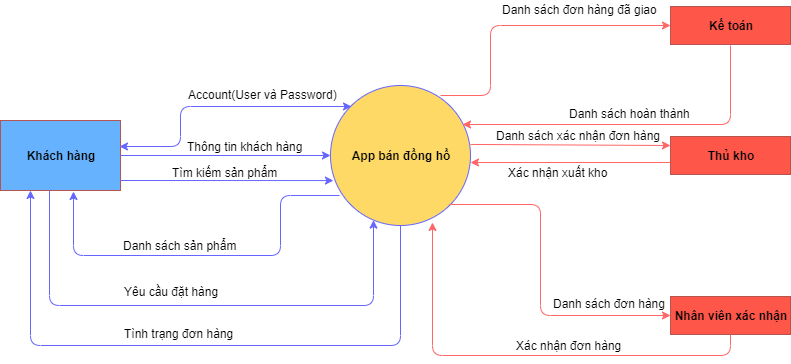
3.2 Kiểm thử 59

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC 67

4.1 Kết quả đạt được 67

68

# DANH MỤC BẢNG BIỂU, HÌNH VẼ, SƠ ĐỒ



# CHƯƠNG 1. TỔNG QUAN VỀ REACT NATIVE

## 1. Giới thiệu tổng quan về React native

***1.1 Quy trình cài đặt React Native (Windows)***

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

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

choco install -y nodejs.install python2 jdk8

***Bước 2: React Native CLI***

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

npm install -g react-native-cli

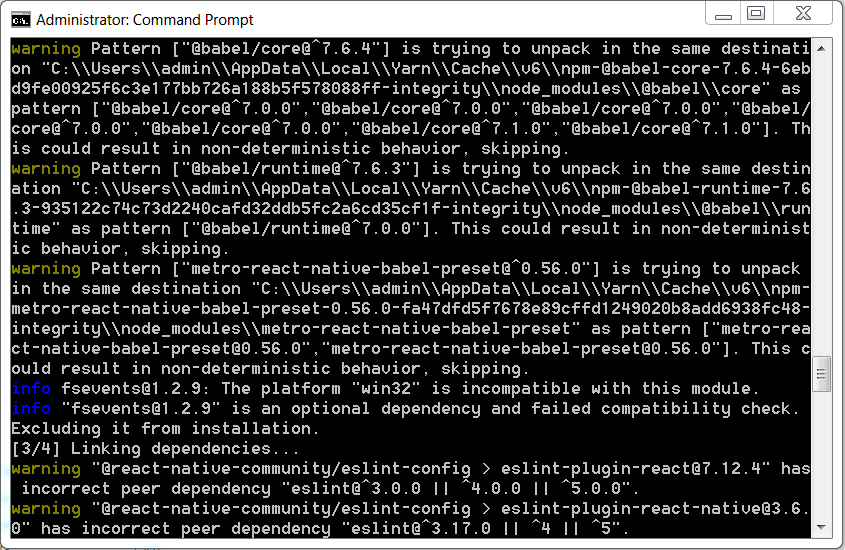
***Bước 3: Chuẩn bị môi trường phát triển cho Android***

Google APIs, Android 23

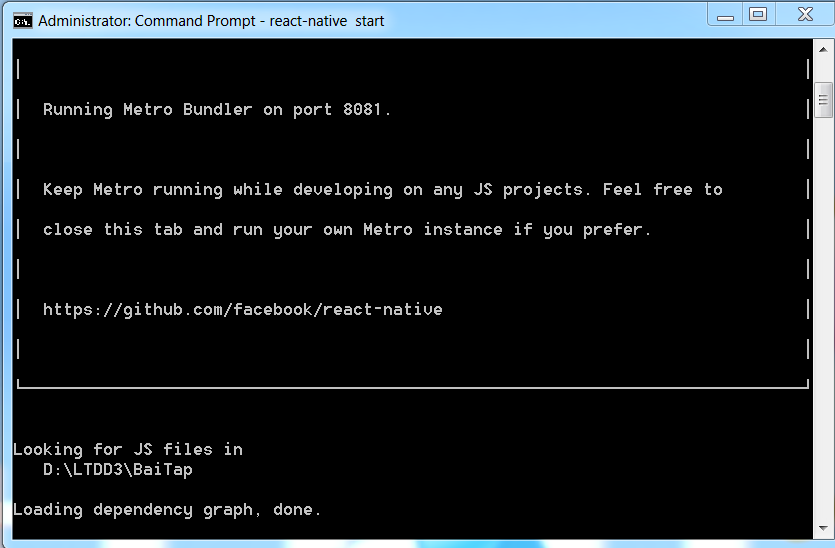
Android SDK Platform 23

Intel x86 Atom\_64 System Image

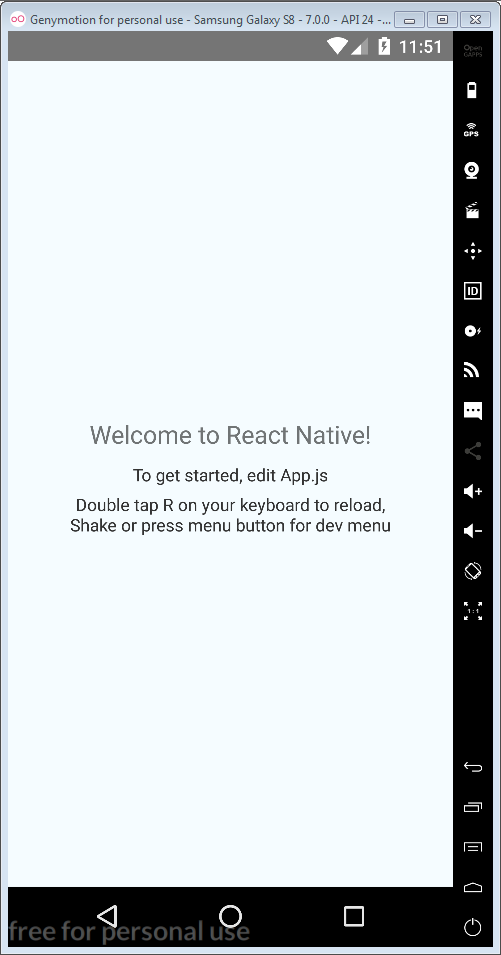
Google APIs Intel x86 Atom\_64 System Image

***1.2 Tìm hiểu một ứng dụng mẫu React Native*** 

*Hình 1: Quá trình cài đặt React Native*



*Hình 2: Chạy React Native trên Android*



*Hình 3: Kết quả chạy React Native trên máy ảo Genymotion*

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

***2.1 Biến Let, Var, Conts trong ES6***

***2.1.1 Biến Let***

- Tạo ra một biến chỉ có thể truy cập được trong block bao quanh nó

Ví dụ:

function foo() {

let x = 10;

if (true) {

let x = 20; // x này là x khác

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

}

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

}

- Ngoài ra, khi ở global scope (tức là không nằm trong một function nào cả), từ khóa var tạo ra thuộc tính mới cho global object (this), còn let thì không

Ví dụ:

var x = 'global';

let y = 'global';

console.log(this.x); // "global"

console.log(this.y); // undefined

- Let rất hiệu quả đó là sử dụng callback trong một vòng lặp

***2.1.2 Biến Var***

- Biến var trong ES6 có thể tạo ra 1 biến có phạm vi truy cập xuyên suốt 1 function chứa nó.

Ví dụ:

function khoiTao()

{

var x = 5;

if(true)

{

var x = 10;// x khởi tạo ở đây giống với x đã khởi tạo trước đó

console.log(x);// in ra giá trị là 10

}

Console.log(x); // vẫn in ra giá trị 10.

}

***2.1.3 Biến Const***

- Dùng để khai báo một hằng số, nếu một biến được khai báo là hằng số thì phải gán giá trị lúc khai báo luôn và sẽ không thể thay đổi giá trị cho biến đó được nửa.

Ví dụ:

const A = 2;

A = 4;

Chạy lên sẽ nhận thông báo lỗi Uncaught TypeError: Assignment to constant variable

***2.2 Arrow Function***

- Là kiểu cú pháp giúp đơn giản hoá việc việc định nghĩa hàm

- Cách dùng arrow function

- Trường hợp nhiều tham số:

+ Từ khoá function được bỏ đi

+ Thêm ký tự => (gần giống với mũi tên) đặt giữa dấu ngoặc kết thúc danh sách tham số ) và dấu ngoặc bắt đầu logic của hàm {

+ Thay từ khoá var bằng const để quy định rằng giá trị của sumNumbers sẽ không thay đổi

Ví dụ:

- Trường hợp 1 tham số:

//ES5

var docLogEs5 = function docLog() {

console.log(document);

};

//ES6

var docLogEs6 = () => { console.log(document); }

docLogEs6();

- Trường hợp không tham số:

//ES5

var phraseSplitterEs5 = function phraseSplitter(phrase) {

return phrase.split(' ');

};

//ES6

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

console.log(phraseSplitterEs6("ES6 Awesomeness"));

// ["ES6", "Awesomeness"]

***2.3 Kiểu dữ liệu Object***

- Trong Javascript, ta có 5 kiểu dữ liệu cơ bản và 1 kiểu dữ liệu phức hợp. 5 kiểu dữ liệu cơ bản bao gồm: Number, String, Boolean, Undefined và Null. Kiểu dữ liệu phức hợp là kiểu dữ liệu Object.   
- Kiểu dữ liệu Object được sử dụng rất thường xuyên trong Js vì nó rất biến hoá và mạnh mẽ, đây là khái niệm cơ sở cho hàng loạt các đặc điểm nổi bật khác của Javascript

Về mặt định nghĩa, một đối tượng (một object) là một **danh sách các item**, mỗi item là một cặp name-value, trong đó value có thể là: các kiểu dữ liệu cơ bản, function, hay cũng có thể là một object khác (kiểu dữ liệu phức hợp).Ta gọi mỗi item là một property(thuộc tính) của object nếu value của item đó có kiểu dữ liệu là kiểu phức hợp hoặc các kiểu dữ liệu cơ bản, ngược lại nếu value của item nó là một hàm (một function) thì ta gọi nó là method của object (phương thức của object).

Ví dụ:

var myFirstObject = {

firstName: "Nam",

favoriteFood: "Fish"

};

- Cách truy cập đến các thuộc tính của Object:

+ Để lấy được các giá trị của thuộc tính trong object, ta có thể truy cập tới tên của thuộc tính bằng toán tử ngoặc vuông [ ] hoặc dấu chấm ., minh hoạ như sau:

myFirstObject.firstName; //Nam

myFirstObject['favoriteFood']; //Fish

Truy cập tới các thuộc tính không tồn tại của object sẽ cho ta giá trị là undefined.

Có một trường hợp nên chú ý, nếu tên của thuộc tính là số (tức là number), thì ta chỉ có thể truy cập bằng cách dùng dấu ngoặc vuông [ ] mà không thể dùng dấu chấm ., xem minh hoạ sau để hiểu rõ:

var myObject = {10: 'test1'}

myObject.10; *//Lỗi*

myObject['10']; *//test1*

- Cách để tạo ra 1 Object:

Thông thường, ta có 2 cách để tạo ra một object: dùng object literals và dùng Object constructor

Cách 1: Dùng Object literals

Mình không rõ tiếng Việt gọi literal là gì, nhưng đại khái là sẽ dùng cặp ngoặc nhọn { } để tạo một object. Literal có thể hiểu theo ngữ cảnh nào đó là việc “sử dụng chuỗi thuần tuý”. Đoạn code sau minh hoạ cho điều này:

var myBook = {10: 'test1'}; *//đối tượng có 1 thuộc tính*

*//đối tượng có 1 thuộc tính và 1 phương thức (method)*

var myCar = {

brand: 'Toyota',

run: function(){

console.log('running');

}

};

Cách 2: Dùng Object constructor

Cách này sẽ sử dụng phương thức khởi tạo (constructor) của kiểu dữ liệu Object để tạo ra các object. Phương thức khởi tạo này là một hàm để tạo ra các object mới, ta dùng kèm từ khoá new:

*//Tạo 1 đối tượng mới*

var myApple = new Object();

*//Thêm các thuộc tính cho đối tượng*

myApple.color = 'red';

myApple.shape = 'round';

myApple.howSweet = function(){

console.log('I am sweet');

};

Trong Js, thì các thuộc tính của object có thể là các kiểu dữ liệu cơ bản, các phương thức, hay cũng có thể là các object con. Kiểu dữ liệu Object cho ta một cách sử dụng rất linh hoạt và mạnh mẽ.

***2.4 Khai báo Class trong ES6***

***2.4.1 Class Definition***

Với ES6 thì bạn không thể khai báo các thuộc tính như bình thường được mà bạn chỉ có thể gán nó vào các phương thức trong đối tượng được thôi.

Ví dụ 1: Khai báo một biến có kiểu dữ liệu là object

var employee = {

name: null;

age: null,

setName: function (name) {

this.name = name;

},

getName: function () {

return this.name;

},

setAge: function (age) {

this.age = age;

},

getAge: function () {

return this.age;

},

Ví dụ 2: Chuyển sang dạng ES6

class Employee {

setName (name) {

this.name = name;

}

getName () {

return this.name;

}

setAge (age) {

this.age = age;

}

getAge () {

return this.age;

}

};

Để khởi tạo đối tượng được khai báo theo chuẩn ES6 thì các bạn sử dụng từ khóa new

**Cú pháp : new ClassName; hoặc new ClassName();**

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

Ví dụ

class Employee {

constructor (name, age) {

this.name = name;

this.age = age;

}

setName (name) {

this.name = name;

}

getName () {

return this.name;

}

setAge (age) {

this.age = age;

}

getAge () {

return this.age;

}

};

***2.4.2* Class Inheritance**

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

**Cú Pháp**: class A extends B {}

**Trong đó**: A là class đang được khai báo, và nó kế thừa lại các thuộc tính và phương thức từ class B.

Ví dụ

class MaleEmployee extends Employee {

constructor (name,age,wifeName) {

super(name,age);

this.wifeName = wifeName;

}

setWifeName (wifeName) {

this.setWifeName;

}

getWifeName () {

return this.wifeName;

}

}

***2.4.3 Base class access***

Trong ES6, để thực hiện gọi các phương thức trong lớp cha khi đang ở lớp con, mà phương thức đó đã bị rewrite trong lớp con rồi thì các bạn sử dụng keyword super với cú pháp như sau:

super.methodName();

Trong đó, methodName là phương thức của lớp cha mà bạn muốn gọi.

Ví dụ

class Employee {

getClassName () {

return "Class Employee";

}

};

class MaleEmployee extends Employee {

getClassName () {

return "Class MaleEmployee";

}

classClassName () {

return super.getClassName();

}

}

var employee = new MaleEmployee();

console.log(employee.classClassName());

***2.4.4 Static Members***

Trong ES6 cũng hỗ trợ chúng ta khai báo các thành phần tĩnh cho đối tượng, bằng cách sử dụng từ khóa static ở trước tên phương thức.

Ví dụ:

class Employee {

static defaultEmployee () {

return "Đây là phương thức defaultEmployee";

}

};

Và khi một phương thức được khai báo là static methods thì chúng ta sẽ không thể gọi phương thức đó một cách thông thường được nữa, mà chúng ta sẽ gọi theo cú pháp sau:

className.staticMethodName();

Trong đó:

* className là tên của class chứa static methods mà bạn muốn gọi..
* staticMethodName là tên của static methods mà bạn muốn gọi.

Ví dụ: Mình sẽ gọi phương thức defaultEmployee của class Employee ở trên.

console.log(Employee.defaultEmployee());

***2.4.5*  *Setter and Getter.***

Để khai báo các setter và getter trong ES6 chúng ta sử dụng keyword set và get trước tên các phương thức mà bạn muốn thiết lập nó là setter hoặc getter.

Ví dụ

class Employee {

constructor (name, age) {

this.name = name;

this.age = age;

}

set employeeName (name) {

this.name = name;

}

get employeeName () {

return this.name;

}

set employeeAge (age) {

this.age = age;

}

get employeeAge () {

return this.age;

}

};

Hoặc bạn cũng có thể thực hiện setter, getter bằng cách truy cập trực tiếp vào class mà không cần khởi tạo.

***2.5 Cách xử lý mảng trong ES6***

***2.6 Cách sử dụng Map trong ES6***

Map là một kiểu dữ liệu tương tự như Set, tuy nhiên với Map thì có cấu trúc dạng key => value còn với Set thì chỉ có value. Và đây cũng chính là giải pháp cho vấn để ở đầu bài viết

Key có thể là bất cứ thứ gì string, number, NaN thậm trí là Object như (class, function, json object)

## Thao tác cơ bản

* Khởi tạo: let map = new Map()
* Thêm phần tử: map.set('Name', 'Nguyen Van A');
* Xóa phần tử: map.delete("Name");
* Kiểm tra phần tử tồn tại: map.has('Name')
* Đếm tổng số phần tử: map.size
* Xóa toàn bộ phần tử: map.clear();

### Khởi tạo

let blankMap = new Map(); // Map(0) {}

let thiToiec = new Map([

["lan\_1", "failed"],

["lan\_2", "failed"],

["lan\_3", "failed"]

]); // Map(3) {"lan\_1" => "failed", "lan\_2" => "failed", "lan\_3" => "failed"}

Có thể khởi tạo Map rỗng hoặc Map kèm dữ liệu.

Khi khởi tạo sẽ có một tham số truyền vào, và giá trị của tham số này là một mảng của một mảng con, trong đó mảng con chứa hai phần tử đại diện cho key và value. Nghe có vẻ hack não nhưng nhìn ví dụ bạn sẽ thấy dễ hiểu hơn.

**Lưu ý: key trong map là unique nên bạn truyền 2 mảng con có key giống nhau thì giá trị mảng con trước sẽ bị ghi đè bằng giá trị mảng con sau.**

### Thêm phần tử

let thiToiec = new Map([

["lan\_1", "failed"],

["lan\_2", "failed"],

["lan\_3", "failed"]

]);

thiToiec.set('lan\_4', 'failed');

// Map(4) {"lan\_1" => "failed", "lan\_2" => "failed", "lan\_3" => "failed", "lan\_4" => "failed"}

thiToiec.set('lan\_4', 'passed');

// Map(4) {"lan\_1" => "failed", "lan\_2" => "failed", "lan\_3" => "failed", "lan\_4" => "passed"}

Nếu thêm phần tử đã có key thì value sẽ được ghi đè.

### Xóa phần tử

let thiToiec = new Map([

["lan\_1", "failed"],

["lan\_2", "failed"],

["lan\_3", "failed"]

]);

thiToiec.delete('lan\_3'); //true

thiToiec.delete('lan\_4'); //false

### Kiểm tra phần tử tồn tại

let thiToiec = new Map([

["lan\_1", "failed"],

["lan\_2", "failed"],

["lan\_3", "failed"]

]);

thiToiec.has('lan\_3'); //true

thiToiec.has('lan\_4'); //false

//

### Đếm số phần tử

let thiToiec = new Map([

["lan\_1", "failed"],

["lan\_2", "failed"],

["lan\_3", "failed"]

]);

thiToiec.size; // 3

### Xóa toàn bộ phần tử

let thiToiec = new Map([

["lan\_1", "failed"],

["lan\_2", "failed"],

["lan\_3", "failed"]

]);

thiToiec.clear();

// Map(0) {}

***2.7 Cách sử dụng Set trong ES6***

Set là tập hợp các phần tử khác nhau và N/A key, vì vậy nếu bạn thêm 2 giá trị giống nhau vào set thì nó chỉ lưu 1 lần thôi.

## Thao tác cơ bản

* Khởi tạo: let set = new Set();
* Thêm phần tử: set.add(value);
* Xóa phần tử: set.delete(value);
* Kiểm tra tồn tại giá trị: set.has(value);
* Đếm tổng số phần tử: set.size;
* Xóa toàn bộ phần tử: set.clear();

### Khởi tạo

let blankSet = new Set();

let set = new Set(['a', 'b', 'c', 'd', 1, 2, 3]);

Khi khởi tạo bạn có thể khai báo một set rỗng, hoặc khai báo luôn giá trị

**Lưu ý**: **giá trị truyền vào phải mà một mảng**

### Thêm phần tử

let char = new Set(['a', 'b']);

char.add('c'); //char = Set(3) {"a", "b", "c"}

### Xóa phần tử

let char = new Set(['a', 'b', 'c']);

char.delete('c'); //char = Set(2) {"a", "b"}

### Kiểm tra phần tử tồn tại

let char = new Set(['a', 'b', 'c']);

char.has('c'); // true

char.has('d'); // false

### Tính số phần tử

let char = new Set(['a', 'b', 'c']);

char.size; // 3

### Vòng lặp với Set

Để lặp qua các phần tử ta dùng cú pháp **for...of**

let chars = new Set(['a', 'b', 'c']);

for (let char of chars){

console.log(char);

}

// Output:

// a

// b

// c

### Chuyển sang Array

let char = new Set(['a', 'b', 'c']);

let arr\_char = [...char]; // arr\_char = ['a', 'b', 'c']

Rất nhanh mà không sợ nóng, để chuyển sang array ta dùng toán tử 3 chấm [...set](Spread Operator), do vậy có thể dùng map và fillter của array bằng cú pháp

let set = new Set([1, 2, 3]);

let arr = [...set].filter(function(x){

return (x % 2) == 0;

});

//arr = [2]

Chuyển set sang array được, vậy mình tự hỏi chuyển ngược lại từ array sang set thì làm sao? Hỏi xong thấy mình ngu luôn, câu trả lời ở phần khởi tạo nhé

**Lưu ý: nếu array có nhiều phần tử giống nhau, thì chỉ lấy 1 phần tử thôi nhé.**

### Xóa toàn bộ phần tử

let set = new Set([1, 2, 3]);

set.clear();

//Set(0) {}

## 3. Component trong react native

Component là từ được dùng trong công nghệ thông tin mang ý nghĩa là bộ phận, thành phần hoặc yếu tố tổ chức của một hệ thống hoặc phần mềm.

Trong phần mềm, Component được gọi ra từ các chương trình khác, cho phép liên kết và sử dụng như một bộ phận trong chương trình.

Việc chúng ta xây dựng trang web sử dụng những bộ phận nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, bên trong mỗi React Component lại có thể chứa các thành phần khác.

Mỗi component trong React Native có thể thay đổi, được cập nhật component dựa trên những thay đổi của trạng thái.

Trong React, chúng ta điều khiển component bằng cách sử dụng props và state.\ React hỗ trợ 3 cách viết component đó là:

* Functional Component
* React.Component
* React.PureComponent

Functional Component:

Nó đơn giản chỉ là một function return về React Element mà thôi.Không implement [Lifecycle hook](https://github.com/nguyenvanhoang26041994/dev-experiences/blob/master/React/lifecycle_hook.md) nên nói về tốc độ render nó là nhanh nhất.

Ví dụ

* React.Component sử dụng bạn cần extends React.Component.Nó khác ở React.PureComponent một chổ duy nhất là shouldComponentUpdate.

+ shouldComponentUpdate mặc định là reference compare. Nghĩa là nếu nextProps === prevProps && nextState === prevState thì return false. Return false thì sẽ không re-render. Tuy nhiên bạn có thể override lại được để chống render không cần thiết, từ đó ang performance.

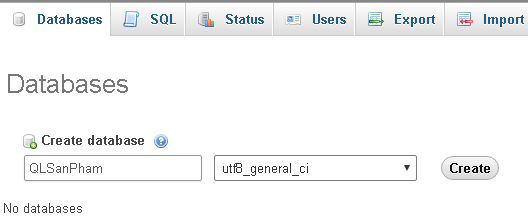
* React.PureComponent

+ sử dụng bạn cần extends React.PureComponent.

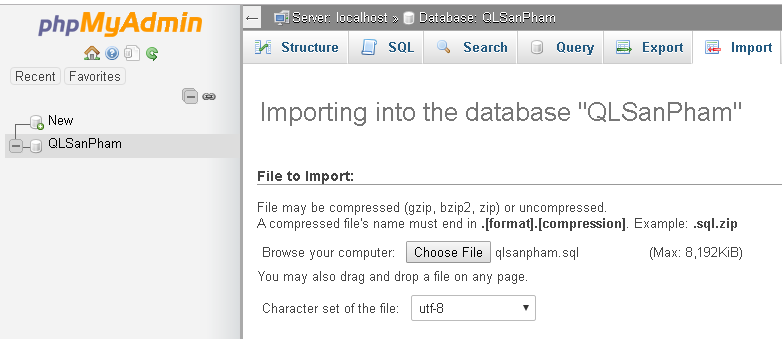
Nó là anh em sinh đôi nhưng thông minh hơn một chút nhờ sự khác biệt tại shouldComponentUpdate.

Bạn không thể override lại shouldComponentUpdate và nó là shallow compare. Nghĩa là các props con của nextProps === các props con của prevProps và các state con của nextState === các state con của prevState thì return false, return false thì sẽ không re-render nữa.

## 4. API Trong React Native



*Hình 4: Cơ sở dữ liệu API*



*Hình 5: Tạo cơ sở dữ liệu quản lý sản phẩm*

DBConfig.php

<?php

//Define your host here.

$HostName ="localhost";

//Define your database name here.

$DatabaseName = "QLSanPham";

//Define your database username here.

$HostUser = "root";

//Define your database password here.

$HostPass = "mysql";

mysql\_query('SET NAMES utf8');

?>

Select\_SanPham.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 tbSanPham";

$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();

?>

Update\_SanPham.php

<?php

include 'DBConfig.php';

$con = mysqli\_connect($HostName,$HostUser,$HostPass,$DatabaseName);

$json = file\_get\_contents('php://input');

$obj = json\_decode($json,true);

$masp = $obj["masp"];

$tensp=$obj["tensp"];

$hinhanh=$obj["hinhanh"];

$Sql\_Query = "UPDATE tbSanPham SET tensp = '$tensp',hinhanh= '$hinhanh' WHERE masp = '$masp'";

if(mysqli\_query($con,$Sql\_Query)){

$MSG = 'Sửa dữ liệu thành công!' ;

$json = json\_encode($MSG);

echo $json ;

}

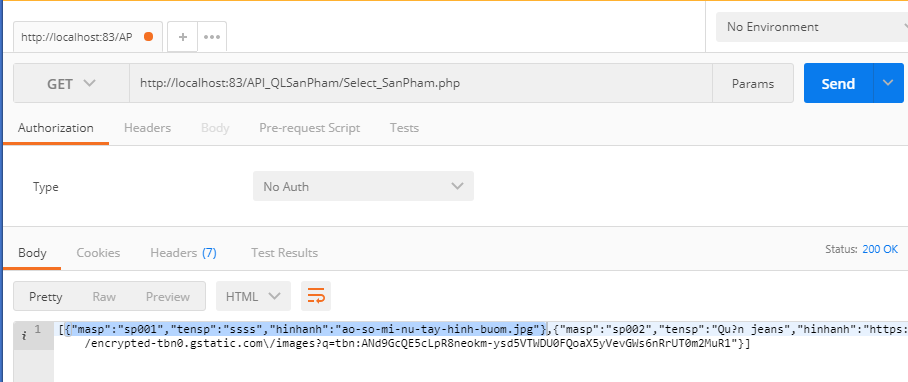
else{

echo 'Wrong';

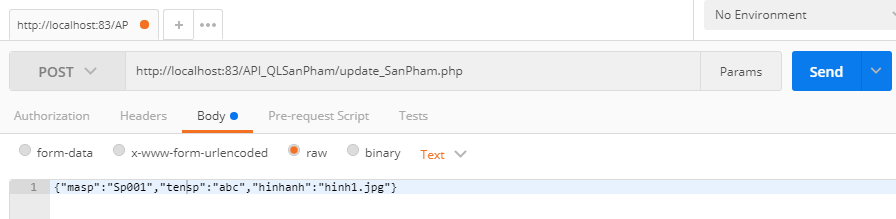
}

mysqli\_close($con);

?>



*Hình 6: Chạy cơ sở dữ liệu*



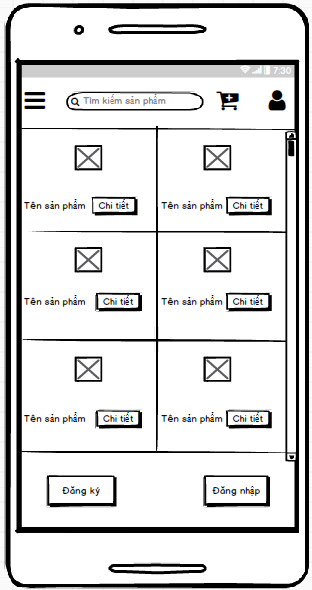
*Hình 7: Kết quả*

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

**2.1.**       **Functional RequirementsYêu cầu chức năng**

**2.11.****1 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **chính**

**2.1.11.1.1** **User InterfacesGiao diện màn hình** **chính**



*Hình 1: Màn hình chính*

**2.1.21.1.2** **Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemMục** | **DescriptionMiêu tả** | **ActionHoạt động** | **ResponsePhản ứng** |
| **Màn hình chính** | Màn hình chính gồm có:   * Slide Menu * Icon Button giỏ hàng * Textbox tìm kiếm * Icon button Tìm kiếm * ListView Item * Image Item * Button Đăng ký * Button Đăng nhập | N/A | N/A |
| Slide Menu | Khi người dùng muốn tìm sản phẩm theo phân loại | Chạm vào Slide Menu trên màn hình chính | Sử dụng biến Intent để thiết lập điều hướng khi khách hàng chạm vào slider menu, màn hình sẽ chuyển sang màn hình có Slide Menu với các chức năng |
| Icon Button giỏ hàng | Khi người dùng muốn xem giỏ hàng | Chạm vào button giỏ hàng trên màn hình chính | Sử dụng biến Intent để thiết lập điều hướng khi khách hàng chạm vào icon giỏ hàng, màn hình sẽ chuyển sang màn hình đơn hàng(dữ liệu từ database DSĐH) |
| Textbox tìm kiếm | Cho phép người dùng gõ vào món hàng cần tìm | Chạm vào textbox tìm kiếm và gõ kí tự | So sánh dữ liệu nhập của khách hàng với database, nếu trùng khớp sẽ hiển thị sản phẩm khách hàng cần tìm, nếu sai sẽ báo lỗi “N/A sản phẩm phù hợp" |
| Icon button Tìm kiếm | Khi người dùng muốn tìm món hàng | Chạm vào Icon Button Tìm kiếm | Lấy dữ liệu từ databse(DS sản phẩm) và hiển thị bằng ListView cùng với từ khóa mà khách hàng đã nhập trước đó |
| ListView Item | Hiển thị danh sách các món hàng | N/A | N/A |
| Image Item | Hiển thị hình ảnh sản phẩm | N/A | N/A |
| Button Đăng nhập | Khi người dùng muốn đăng nhập tài khoản | Chạm vào button đăng nhập | Sử dụng biến Intent để thiết lập điều hướng khi khách hàng chạm vào link đăng nhập, màn hình sẽ chuyển sang màn hình đăng nhập |
| Button Đăng ký | Khi người dùng muốn đăng ký tài khoản | Chạm vào button đăng ký | Sử dụng biến Intent để thiết lập điều hướng khi khách hàng chạm vào link đăng kí, màn hình sẽ chuyển sang màn hình đăng kí |

**2.11.2 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **đăng nhập**

**2.1.11.2.1    User InterfacesGiao diện màn đăng nhập**

# 

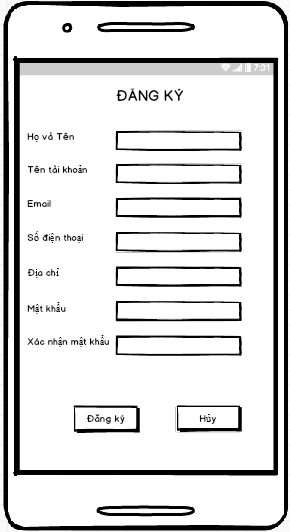
*Hình 2: Màn hình đăng nhập*

**2.1.21.2.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemMục** | **DescriptionSự miêu tả** | **ActionHoạt động** | **ResponsePhản ứng** |
| **MyMobileGame Recommendation ScreenMàn hình đăng nhập** | Màn hình đăng nhập gồm có:   * Label đăng nhập * Image Logo * Label Tên tài khoản * Textbox Tên tài khoản * Label Mật khẩu * Textbox Mật khẩu * Link Quên mật khẩu * Button Đăng nhập * Button Hủy | N/A | N/AN/A |
| Label Đăng nhập | Hiển thị cho người dùng biết đó là trang Đăng nhập | N/A | N/A |
| Image Logo | Hiển thị logo của shop |  |  |
| Label Tên Tài Khoản | Hiển thị cho người dùng xem vị trí đó cần gõ Tên tài khoản | N/A | N/A |
| Textbox Tên Tài Khoản | Cho phép người dùng gõ Tên tài khoản đã đăng ký vào ô textbox | Chạm vào textbox Tên tài khoản và gõ kí tự | Hiển thị các kí tự vừa gõ |
| Label Mật khẩu | Hiển thị cho người dùng xem vị trí đó cần gõ mật khẩu | N/A | N/A |
| Textbox Mật khẩu | Cho phép người dùng gõ Mật khẩu đã đăng ký vào ô textbox | Chạm vào ô textbox và gõ kí tự | Hiển thị dấu \* thay cho kí tự vừa gõ |
| Button Đăng nhập | Khi người dùng muốn đăng nhập vào tài khoản của mình | Chạm vào button Đăng nhập | Username và Password của khách hàng sẽ được so sánh trong Database(danh sách tài khoản), nếu trùng khớp sẽ cho phép đăng nhập, còn nếu sai sẽ thông báo “Tài khoản hoặc mật khẩu không đúng” |
| Button Hủy | Khi người dùng không muốn đăng nhập vào tài khoản cá nhân của mình | Chạm vào button Hủy | Sử dụng biến Intent để thiết lập điều hướng khi khách hàng chạm vào button hủy, màn hình sẽ quay về màn hình chính |

**2.11.3 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **đăng ký**

**2.1.11.3.1    User InterfacesGiao diện màn đăng ký**



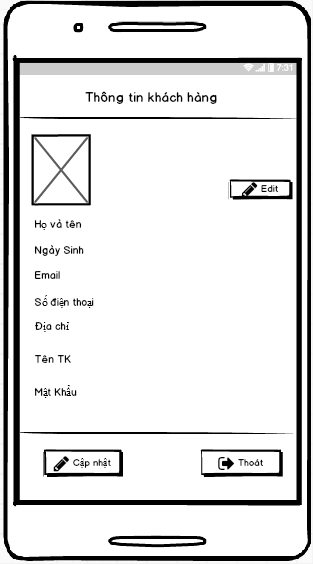
*Hình 3: Màn hình đăng ký*

**2.1.21.3.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemMục** | **DescriptionMiêu tả** | **ActionHoạt động** | **ResponsePhản ứng** |
| **Màn hình đăng kí** | Màn hình đăng kí gồm có:   * Label Đăng ký * Label họ và tên * Textbox họ và tên * Label Email * Textbox Email * Label số điện thoại * Textbox số điện thoại * Label tên tài khoản * Textbox tên tài khoản * Label mật khẩu * Textbox mật khẩu * Label xác nhận lại mật khẩu * Textbox xác nhận lại mật khẩu * Button đăng kí * Button Hủy | N/A | N/A |
| Label Đăng ký | Hiển thị trên màn hình cho người dùng biết đó là trang đăng ký | N/A | N/A |
| Label Họ và tên | Hiển thị cho người dùng xem vị trí đó cần gõ họ và tên | N/A | N/A |
| Textbox Họ và tên | Cho phép người dùng nhập Họ và tên vào | Chạm vào ô textbox và gõ họ và tên người dùng | Kiểm tra nếu khách hàng nhập đúng sẽ xuất hiện dấu ✓, nếu nhập sai sẽ thông báo “Họ và tên không thể chứa kí tự đặc biệt hoặc số” |
| Label Email | Hiển thị cho người dùng xem vị trí đó cần gõ Email | N/A | N/A |
| Textbox Email | Cho phép người dùng nhập thông tin Email | Chạm vào Textbox Email và gõ kí tự | Kiểm tra nếu khách hàng nhập đúng sẽ xuất hiện dấu ✓, nếu nhập sai sẽ thông báo “Email không thể chứa kí tự đặc biệt” |
| Label Số điện thoại | Hiển thị cho người dùng xem vị trí đó cần gõ Số điện thoại | N/A | N/A |
| Textbox Số điện thoại | Cho phép người dùng nhập thông tin Số điện thoại | Chạm vào textbox Số điện thoại và gõ kí tự | Kiểm tra nếu khách hàng nhập đúng sẽ xuất hiện dấu ✓, nếu nhập sai sẽ thông báo “Số điện thoại không thể chứa kí tự đặc biệt hoặc chữ” |
| Label Tên Tài Khoản | Hiển thị cho người dùng xem vị trí đó cần gõ Tên Tài Khoản | N/A | N/A |
| Textbox Tên Tài Khoản | Cho phép người dùng nhập thông tin Tên Tài Khoản | Chạm vào textbox và gõ tên tài khoản | Kiểm tra nếu khách hàng nhập đúng sẽ xuất hiện dấu ✓, nếu tên tài khoản đã có người đăng kí sẽ xuất hiện câu “Tên tài khoản đã tồn tại” |
| Label mật khẩu | Hiển thị cho người dùng xem vị trí đó cần gõ mật khẩu | N/A | N/A |
| Textbox Mật khẩu | Cho phép người dùng nhập Mật khẩu | Chạm vào textbox Mật khẩu và gõ kí tự | Kiểm tra nếu khách hàng nhập đúng sẽ xuất hiện dấu ✓, nếu mật khẩu nhập không đủ yêu cầu sẽ xuất câu “Mật khẩu phải có ít nhất 1 kí tự”. |
| Label Nhập lại mật khẩu | Hiển thị cho người dùng xem vị trí đó cần gõ lại mật khẩu | N/A | N/A |
| Textbox Nhập lại mật khẩu | Cho phép người dùng Nhập Lại Mật Khẩu | Chạm vào textbox Nhập Lại Mật khẩu và gõ kí tự | Kiểm tra nếu khách hàng nhập đúng sẽ xuất hiện dấu ✓, nếu mật khẩu nhập không khớp sẽ xuất câu “Mật khẩu không khớp” |
| Button Đăng ký | Khi người dùng muốn đăng kí tài khoản | Chạm vào button Đăng kí | Dữ liệu của khách hàng khi nhập xong tất cả thông tin sẽ được lưu vào Database(thông tin khách hàng) và điểu hướng san màn hình chính |
| Button Hủy | Khi người dùng muốn hủy, không đăng ký tài khoản | Chạm vào button Hủy | Sử dụng biến Intent để thiết lập điều hướng khi khách hàng chạm vào button hủy, màn hình sẽ quay về màn hình chính |

**2.11.4 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **thông tin khách hàng**

**2.1.11.4.1    User InterfacesGiao diện thông tin khách hàng**



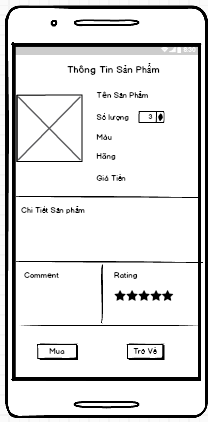
*Hình 4: Màn hình thông tin khách hàng*

**2.1.21.4.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemMục** | **DescriptionMiêu tả** | **ActionHoạt động** | **ResponsePhản ứng** |
| **Registration screenMàn hình thông tin khách hàng** | Hiển thị thông tin khách hàng đã login  Màn hình gồm:  +Label Thồng tin khách hàng  +Label Họ và Tên  + TextView Họ và Tên  +Label Ngày Sinh  +TextView Ngày Sinh  +Lable Email  +TextView Email  +Label Số điện thoại  +TextView Số điện thoại  +Label Địa chỉ  +TextView Địa chỉ  +Label Tên tài khoản  +TextView Tên tài khoản  +Label Mật khẩu  +TextView Mật khẩu  +Image Khách hàng  + Button Cập nhật  +Button Thoát  +Button EditSign in with FacebookScreen has a"I have read the Terms and Conditions" link anda checkbox | Không có | Không có |
| Label Thông tin khách hàng | Hiển thị để người dùng biết màn hình hiện tại là màn hình “Thông tin khách hàng” | Không có | Không có |
| Label Họ và Tên | Hiện thị cho người biết thông tin hiện thị là họ và tên |  |  |
| TextView Họ và Tên | -Hiện thị thông tin họ và tên của người dùng  -Cho phép người dùng chỉnh sửa thông tin họ và tên | Chạm vào textview | Khi người dùng thay đổi thông tin họ và tên thì ký tự thay đổi sẽ hiện thị trong textview |
| Label Ngày sinh | Hiện thị cho người dùng biết thông tin hiển thị là ngày sinh | Không có | Không có |
| TextView Ngày sinh | -Hiện thị thông tin ngày sinh của khách hang  -Cho phép người dùng chỉnh sửa thông tin ngày sinh | Chạm vào textview | Khi người dùng thay đổi thông tin ngày sinh thì ký tự thay đổi sẽ hiện thị tròn textview |
| Lable Email | Hiện thị cho người biết thông tin hiện thị là Email | Không có | Không có |
| TextView Email | -Hiện thị thông tin Email của khách hang  -Cho phép người dùng chỉnh sửa thông tin Email | Chạm vào textview | Khi người dùng thay đổi thông tin Emai thì ký tự thay đổi sẽ hiện thị trong textview |
| Label Số điện thoại | Hiện thị cho người dùng biết thông tin hiện thị là Số điện thoại | Không có | Không có |
| TextView Số điện thoại | -Hiện thị thông tin số điện thoại của khách hang  -Cho phép người dùng chỉnh sửa thông tin Số điện thoại | Chạm vào textview | Khi người dùng thay đổi thông tin Số điện thoại thì ký tự thay đổi sẽ được hiển thị trong textview |
| Label Địa chỉ | Hiện thị cho người dùng biết thông tin hiên thị là Địa chỉ | Không có | Không có |
| TextView Địa chỉ | -Hiện thị thông tin Địa chỉ của khách hang  -Cho phép người dùng chỉnh sửa thông tin Địa chỉ | Chạm vào textview | Khi người dùng thay đổi thông tin Địa chỉ thì ký tự thay đổi sẽ hiển thị trong textview |
| Label Tên tài khoản | Hiện thị cho người dùng biết thông tin hiện thị là Tên tài khoản | Không có | Không có |
| TextView Tên tài khoản | -Hiện thị thông tin Tên tài khoản của khách hang  -Cho phép người dùng chỉnh sửa thông tin Tên tài khoản | Chạm vào textview | Khi người dùng thay đổi thông tin Tên tài khoản thì ký tự thay đổi sẽ hiện thị trong textview |
| Label Mật khẩu | Hiện thị cho người dùng biết thông tin hiện thị là Mật khẩu của khách hàng | Không có | Không có |
| TextView Mật khẩu | -Hiển thị thông tin Mật khẩu của khách hang  -Cho phép người dùng chỉnh sửa thông tin Mật khẩu | Chạm vào textview | Khi người dùng thay đổi thông tin |
|  |  |  |  |
| **Image Khách Hàng** | Hiện thị hình ảnh đại diện của khách hàng | Không có | Không có |
| **Button chỉnh sửa** | Cho phép khách hàng thay đổi hình ảnh cá nhân | Chạm vào nút chỉnh sửa | -Sẽ chuyển sang màn hình mới để người dùng lựa chọn hình ảnh.  -Hình ảnh sau khi thay đổi sẽ hiện thị trong image Hình ảnh |
| **Button cập nhật** | Cho phép khách hàng cập nhật lại thông tin đã nhập trên màn hình | Chạm vào nút cập nhật | Tất cả thông tin thay đổi sẽ được lưu lại và cập nhật lại vào bảng dữ liệu tài khoản |
| **Button Thoát** | Khi người dùng muốn thoát và đăng xuất khỏi màn hình | Chạm vào nút đăng xuất | Trở về màn hình login |

**2.11.5 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **thông tin sản phẩm**

**2.1.11.5.1    User InterfacesGiao diện màn hình thông tin sản phẩm**



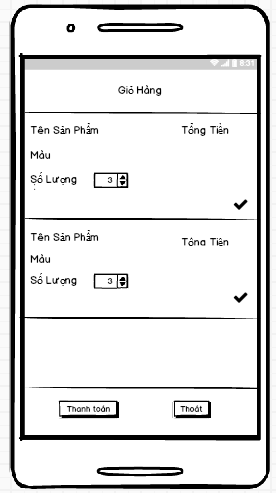
*Hình 5: Màn hình thông tin sản phẩm*

**2.1.21.5.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemMục** | **DescriptionMiêu tả** | **ActionHoạt động** | **ResponsePhản ứng** |
| Màn hình thông tin sản phẩm | Màn hình thông tin sản phẩm gồm có:   * Label Thông tin sản phẩm * Label Tên sản phầm * Image sản phẩm * Label Số lượng * Combo box Số lượng * Label Màu sắc * Label Giá tiền * Label Chi tiết sản phẩm * Label Comment * Label Rating * Button Mua * Button Trở về | N/A | N/A |
| Label Thông tin sản phẩm | Hiển thị cho người dùng biết đó là Hiện thị cho người dùng biết màn hình hiện tại là màn hình “Thông tin sản phẩm” | N/A | N/A |
| Label Tên sản phẩm | Hiển thị cho người dùng biết thông tin hiển thị là Tên sản phẩm | Không có | Không có |
| Image sản phẩm | Hiển thị hình ảnh của sản phẩm | Không có | Không có |
| Label Số lượng | Hiển thị cho người dùng biết thông tin hiển thị là Số lượng của sản phẩm | Không có | Không có |
| Combo box Số lượng | -Hiện thị thông tin số lượn của sản phẩm  -Cho phép người dùng chỉnh sửa số lượng sản phẩm | Chạm vào mũi tên trên combobox và chạm chọn số lượng người dùng muốn chọ | Khi người dùng thay đổi thông tin số lượng sản phẩm thì số lượng mới sẽ được hiển thị trong combobox |
| Label Màu sắc | Hiển thị cho người biết thông tin hiển thị là máu sắc của sản phẩm | Không có | Không có |
| Label Giá tiền | Hiển thị cho người dùng biết thông tin hiển thị là Gía tiền của sản phẩm | Không có | Không có |
| Label Chi tiết sản phẩm | Hiển thị cho người dùng biết thông tin hiển thị là mô tả Chi tiết của sản phẩm | Không có | Không có |
| Label Comment | Hiển thị cho người dùng biết thông tin hiển thị là commen của khách hàng | Không có | Không có |
| Label Rating | Hiển thị cho người dùng biết thông tin hiển thị là rating của sản phẩm do | Không có | Không có |
| Button Mua | Người dùng sử dụng khi muốn mua sản phẩm | Chạm vào button Mua | -Thoát khỏi màn hình “Thông tin sản phẩm”  -Chuyển sang màn hình “Thanh toán” |
| Button Trở về | Khi người dùng muốn trở về trang trước đó(màn hình chính) | Chạm vào button Trở về | -Thoát khỏi màn hình “ Thông tin sản phẩm”  -Chuyển sang màn hình chính khi khách hàng đã đăng nhập vào tài khoản |

**2.11.6 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **giỏ hàng**

**2.1.11.6.1    User InterfacesGiao diện màn hình giỏ hàng**



*Hình 6: Màn hình giỏ hàng*

**2.1.21.6.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemMục** | **DescriptionMiêu tả** | **ActionHoạt động** | **ResponsePhản ứng** |
| **Màn hình Giỏ hàng** | Màn hình **Giỏ hàng** gồm :  Checkbox sản phẩm muốn mua  Label Giỏi Hàng  Label Tên Sản Phẩm  Label Màu  Label Số lượng  Label Tổng tiền  Button Thanh toán  Button Thoát  Combobox Số lượng  ListView Sản Phẩm | Không có | Không có |
| **Hình ảnh sản phẩm** | Hiện thị hình ảnh mô tả sản phẩm | Không có | Không có |
| **Label Gio hàng** | Hiện thị cho người dùng biết màn hình hiện tại là màn hình “Giỏ hàng” | Không có | Không có |
| **Label Tên Sản Phẩm** | Hiện thị cho người dùng biết thông tin hiển thị là Tên sản phẩm | Không có | Không có |
| **Label Màu sắc** | Hiển thị cho người dùng biết thông tin hiển thị là Màu của sản phẩm | Không có | Không có |
| **Label Tổng tiền** | Hiện thị cho người dùng biết thông tin hiển thị là Tổng tiền của các sản phảm mà người dùng mua | Không có | Không có |
| **Combobox số lượng** | -Hiện thị số lượng sản phẩm muốn mua  -Người dùng sử dụng khi muốn tăng giảm số lượng sản phẩm muốn mua | - Chạm vào biểu tượng mủi tên trong combobox Số lượng và lựa chọn số lượng sản phẩm | Khi người dùng thay đổi số lượng, số lượng mới sẽ được hiển thị trong combobox |
| **ListView Sản phẩm** | Hiện thị danh sách các sản phẩm người dùng đã lựa chọn để thanh toán gồm các thuộc tính: tên sản phẩm, số lượng sản phẩm, màu sắc sản phẩm và hình ảnh sản phẩm | Không có | Không có |
| **Button Thoát** | Người dùng sử dụng khi muốn trở về màn hình trước đó | Chạm vào button | -Thoát khỏi màn hình “Giỏ hàng”  -Chuyển về màn hình trước đó |
| **Button Thanh toán** | Người dùng sử dung khi muốn thanh toán sản phẩm đã lựa chọn | Chạm vào button | -Thoát khỏi màn hình “Giỏ hàng”  - Chuyển sang màn hình “Chi tiết đơn hàng” |
| **Checkbox sản phẩm** | Người dùng sử dụng khi muốn chọn hoặc hủy bỏ sản phẩm lựa chọn để thanh toán | Chạm vào checkbox | -Khi Sản phẩm được người dùng lựa chọn sẽ hiện thị dấu tích  - Sản phẩm không được người người lựa chọn thì check box không hiện thị dấu tích |

**2.11.7 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **chi tiết đơn hàng**

**2.1.11.7.1    User InterfacesGiao diện màn hình chi tiết đơn hàng**



*Hình 7: Màn hình chi tiết đơn hàng*

**2.1.21.7.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình Chi tiết đơn hàng** | Màn hình **Chi tiết đơn hàng** gồm :   * Button Tiếp tục * Button Trở về * Date picker ngày đặt hàng * Lable Chi tiết đơn hàng * Lable Mã đơn hàng * Lable Tên đơn hàng * Lable Thông tin giao hàng * Lable Tình trạng vận chuyển * Lable Tổng tiền * Lable Địa chỉ cửa hàng * Lable Liên hệ sự cố | Không có | Không có |
| **Button Tiếp tục** | Người dùng sử dụng khi muốn đặt hàng sản phẩm đã được lựa chọn | Chạm vào button | -Thoát khỏi màn hình”Chi tiết đơn hang”  -Chuyển sang giao diện màn hình “Danh sách đơn hàng xác nhận” |
| **Lable Chi tiết đơn hàng** | Hiện thị cho người dùng biết màn hình hiện tại là màn hình “Chi tiết đơn hàng” | Không có | Không có |
| **Lable Mã đơn hàng** | Hiện thị cho người dùng biết thông tin hiển thị là Mã đơn hàng mà người dùng mua | Không có | Không có |
| **Lable Tên đơn hàng** | Hiện thị cho người dùng biết thông tin hiển thị là Tên đơn hàng mà người dùng mua | Không có | Không có |
| **Lable Thông tin giao hàng** | Hiện thị cho người dùng biết thông tin hiển thị là Thông tin giao hàng mà người dùng mua | Không có | Không có |
| **Lable Tình trạng vận chuyển** | Hiện thị cho người dùng biết thông tin hiển thị là Tình trạng vận chuyển mà người dùng mua | Không có | Không có |
| **Lable Tổng tiền** | Hiện thị cho người dùng biết thông tin hiển thị là Tổng tiền của các sản phảm mà người dùng mua | Không có | Không có |
| **Lable Địa chỉ cửa hàng** | Hiện thị cho người dùng biết thông tin hiển thị là Địa chỉ cửa hàng của sản phẩm mà người dùng mua | Không có | Không có |
| **Lable Liên hệ sự cố** | Hiện thị cho người dùng biết thông tin hiển thị là Liên hệ sự cố khi gặp vấn đề với sản phẩm mà người dùng mua | Không có | Không có |
| **Button Trở về** | Khi người sử dụng muốn quay lại màn hình trước đó | Chạm vào button | -Thoát khỏi màn hình “Chi tiết đơn hàng”  -Chuyên về màn hình trước đó |
| **Date picker ngày đặt hàng** | Người dùng sử dụng khi muốn thay đổi thời gian đặt hàng | Chạm vào data picker | Hiện thị thời gian người dùng lựa chọn |

**2.11.8 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **danh sách đơn hàng**

**2.1.11.8.1    User InterfacesGiao diện màn hình danh sách đơn hàng**

# 

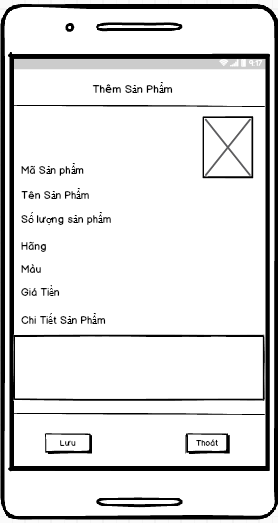
*Hình 8: Màn hình danh sách đơn hàng*

**2.1.21.8.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình Danh sách đơn hàng** | Màn hình Danh sách sản phẩm gồm :  Button Xong  Label Danh sách sản phẩm  Label Mã sản phẩm  Label Tên sản phẩm  Label Giá tiền  Label Số lượng  Label Màu sắc  .Lable Chi tiết  Button Trở lại  Button Thêm sản phẩm  Search box Tìm sản phẩm | Không có | Không có |
| **Button Xong** | Người dùng sử dụng khi muốn kết thúc việc hàng sản phẩm | Chạm vào button | -Thoát khỏi màn hình “Danh sách sản phẩm”  -Chuyển sang màn hình thêm sản phẩm |
| **Label Danh sách sản phẩm** | Hiện thị cho người dùng biết màn hình hiện tại là màn hình “Danh sách sản phẩm” | Không có | Không có |
| **Label Mã sản phẩm** | Hiện thị cho người dùng biết thông tin hiển thị là Mã sản phẩm | Không có | Không có |
| **Label Tên sản phẩm** | Hiện thị cho người dùng biết thông tin hiển thị là Tên sản phẩm | Không có | Không có |
| **Label Giá tiền** | Hiện thị cho người dùng biết thông tin hiển thị là Giá tiền | Không có | Không có |
| **Label Số lượng** | Hiện thị cho người dùng biết thông tin hiển thị là Số lượng | Không có | Không có |
| **Label Màu sắc** | Hiện thị cho người dùng biết thông tin hiển thị là Màu sắc | Không có | Không có |
| **.Lable Chi tiết** | Hiện thị cho người dùng biết thông tin hiển thị là Chi tiết | Không có | Không có |
| **Button Trở lại** | Khi người sử dụng khi muốn quay lại màn hình trước đó | Chạm vào button | -Thoát khỏi màn hình “Danh sách sản phẩm”  -Chuyên về màn hình trước đó |
| **Button Thêm sản phẩm** | Người dùng sử dụng khi muốn thêm sản phẩm vào danh sách | Chạm vào button | -Thoát khỏi màn hình “Danh sách sản phẩm”  -Chuyển sang màn hình “Thêm sản phẩm” |
| **Search box Tìm sản phẩm** | Người dùng sử dụng khi muốn tìm sản phẩm trong danh sách | -Chạm vào search box  -Gõ tên sản phẩm người dùng cần tìm  Chạm vào icon “tìm kiếm” | -Sản phẩm cần tìm sẽ hiện thị trong danh sách nếu có sản phẩm  -Hiện thị thông báo “Không có sản phẩm nào trong danh sách” khi sản phẩm cần tìm khôn có trong danh sách |

**2.11.9 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **thêm sản phẩm**

**2.1.11.9.1    User InterfacesGiao diện màn hình thêm sản phẩm**



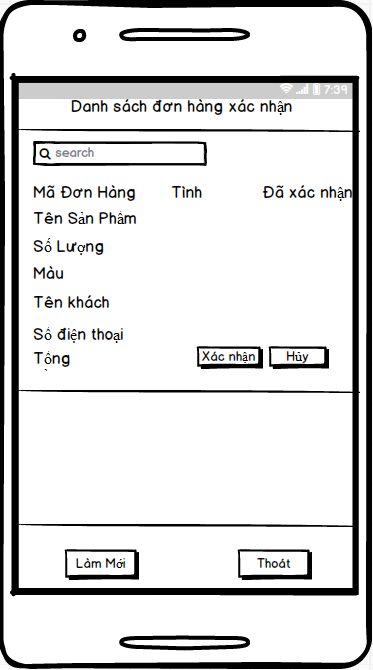
*Hình 9: Màn hình thêm sản phẩm*

**2.1.21.9.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình Thêm sản phẩm** | Màn hình Danh sách đơn hàng gồm :  Label Mã sản phẩm  TextView Mã sản phẩm  Label Tên sản phẩm  TextView Tên sản phẩm  Label Giá tiền  TextView Giá tiền  Label Số lượng  TextView Số lượng  Label Màu sắc  TextView Màu sắc  Lable Chi tiết  Button Lưu  Button Thoát  Image Sản phẩm  Text area Chi tiết sản phẩm | Không có | Không có |
| **Button Lưu** | Người dùng sử dụng khi lưu sản phẩm vừa tạo vào danh sách | Chạm vào button | -Thoát khỏi màn hình “Thêm sản phẩm”  -Chuyển sang màn hình “Danh sách sản phẩm” và hiển thị sản phẩm vừa thêm |
| **Label Mã sản phẩm** | Hiện thị cho người dùng biết thông tin hiển thị là Mã sản phẩm | Không có | Không có |
| **TextView Mã sản phẩm** | Người dùng sử dụng khi muốn thêm thông tin về Mã sản phẩm | Chạm vào textview và gõ ký tự | Hiện thị ký tự người dùng đã nhập vào. |
| **TextView Tên sản phẩm** | Người dùng sử dụng khi muốn thêm thông tin về Tên sản phẩm | Chạm vào textview và gõ ký tự | Hiện thị ký tự người dùng đã nhập vào. |
| **TextView Giá tiền** | Người dùng sử dụng khi muốn thêm thông tin về Gía tiền sản phẩm | Chạm vào textview và gõ ký tự | Hiện thị ký tự người dùng đã nhập vào. |
| **TextView Số lượng** | Người dùng sử dụng khi muốn thêm thông tin về Số lượng sản phẩm | Chạm vào textview và gõ ký tự | Hiện thị ký tự người dùng đã nhập vào. |
| **TextView Màu sắc** | Người dùng sử dụng khi muốn thêm thông tin về Màu sản phẩm | Chạm vào textview và gõ ký tự | Hiện thị ký tự người dùng đã nhập vào. |
| **Label Tên sản phẩm** | Hiện thị cho người dùng biết thông tin hiển thị là Tên sản phẩm | Không có | Không có |
| **Label Giá tiền** | Hiện thị cho người dùng biết thông tin hiển thị là Giá tiền | Không có | Không có |
| **Label Số lượng** | Hiện thị cho người dùng biết thông tin hiển thị là Số lượng | Không có | Không có |
| **Label Màu sắc** | Hiện thị cho người dùng biết thông tin hiển thị là Màu sắc | Không có | Không có |
| **.Lable Chi tiết** | Hiện thị cho người dùng biết thông tin hiển thị là Chi tiết | Không có | Không có |
| **Button Thoát** | Khi người sử dụng khi muốn quay lại màn hình trước đó | Chạm vào button | -Thoát khỏi màn hình “Thêm sản phẩm”  -Chuyên về màn hình trước đó |
| **Image Sản phẩm** | -Hiển thị hình ảnh mô tả của sản phẩm  -Người dùng sử dụng khi muốn thêm ảnh mô tả sản phẩm vào danh sách | -Chạm vào button và lựa chọn hình ảnh từ thư viện | Hiện thị hình ảnh người dùng vừa lựa chọn |
| **Text area Chi tiết sản phẩm** | Người dùng sử dụng khi muốn thêm mô tả thông tin sản phẩm trong danh sách | -Chạm vào ô chữ  -Thêm ký tự mô tả sản phẩm n | Hiển thị ký tự mô tả sản phẩm người dùng vừa thêm |

**2.12. 1 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **danh sách đơn hàng xác nhận**

**2.1.12.1.1    User InterfacesGiao diện màn hình danh sách đơn hang xác nhận**



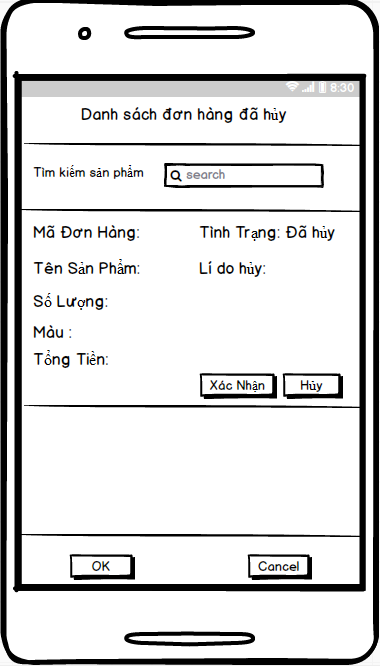
*Hình 10: Màn hình danh sách đơn hang xác nhận*

**2.1.22.1.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình xác nhận đơn hàng đã xác nhận** | Màn hình danh sách của đơn hàng của khách hàng đã xác nhận gồm có:  - Label Danh sách đơn hàng xác nhận  - Label Tên sản phẩm  - Label Số lượng  - Label Màu sắc  - Label Mã đơn hàng  - Label Tỉnh  - Label Đã xác nhận  - Label Tên khách hàng  - Label Số điện thoại  - Label Tổng tiền  - TextView Tên sản phẩm  - TextView Số lượng  - TextView Màu sắc  - TextView Tỉnh  - TextView Mã đơn hàng  - TextView Tên khách hàng  - TextView Số điện thoại  - TextView Tổng tiền  - Searchbox Đơn hàng  - Button Xác nhận  - Button Hủy  - Button Làm mới  - Button Thoát | Không có | Không có |
| **SearchBox** Đơn hàng | Người dùng sử dụng khi muốn tìm kiếm sản phẩm trong danh sách đã xác nhận | - Chạm vào thanh tìm kiếm  - Gõ ký tự tên sản phẩm cần tìn  - Chạm vào biểu tượng “ tìm kiếm” | - Hiển thị ra thông tin sản phẩm tìm kiếm theo từ khóa đã nhập |
| **TextView Tên sản phẩm** | Hiện thị thông tin Tên sản phẩm | Không có | Không có |
| **TextView Số lượng** | Hiện thị thông tin Số lượng sản phẩm | Không có | Không có |
| **TextView Màu sắc** | Hiện thị thông tin Màu sắc sản phẩm | Không có | Không có |
| **TextView Mã đơn hàng** | Hiện thị thông tin Mã đơn hàng của sản phẩm | Không có | Không có |
| **TextView Tên khách hàng** | Hiện thị thông tin Tên khách hàng mua sản phẩm | Không có | Không có |
| **TextView Số điện thoại** | Hiện thị thông tin Số điện thoại của khách hàng mua sản phẩm | Không có | Không có |
| **TextView Tổng tiền** | Hiện thị thông tin Tổng tiền các sản phẩm khách hàng đã mua | Không có | Không có |
| **Button xác nhận** | Người dùng sử dụng khi muốn xác nhận trạng thái tình trạng thành xác nhận đơn hàng | Chạm vào button xác nhận | Đơn hàng của người dùng đã được xác nhận thành công |
| **Button hủy** | Hủy đơn hàng của khách hành khi khách hàng không muốn mua sản phẩm đó nữa | Chạm vào button hủy | Đơn hàng lựa chọn hủy sẽ bị xóa khỏi danh sách |
| **Button làm mới** | Hiển thị thêm các sản phẩm mới của khách hàng vào danh sách | Chạm vào button làm mới | Cập nhật lại danh sách đơn hàng |
| **Button thoát** | Người dùng sử dụng khi muốn thoát khỏi màn hình “danh sách đơn hàng” | Chạm vào button thoát | - Thoát khỏi màn hình “ Danh sách đơn hàng”  - Trở về màn hình trước đó |

**2.12. 2 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **danh sách đơn hang hủy**

**2.1.12.2.1    User InterfacesGiao diện màn hình danh sách đơn hang hủ**



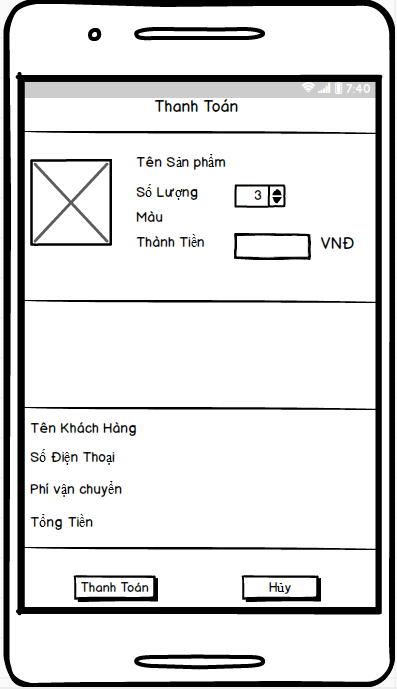
*Hình 11: Màn hình danh sách đơn hàng hủy*

**2.1.22.2.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình xác nhận đơn hàng đã hủy** | Màn hình danh sách của đơn hàng đã hủy có:  - Textview Lý do hủy  - TextView Mã đơn hàng  - TextView Tên sản phẩm  - TextView Số lượng  - TextView Màu sắc  - TextView Tổng tiền  - Searchbox Đơn hàng  - Button Xác nhận  - Button Hủy  - Button OK  - Button Cancel | N/A | N/A |
| **TextView Tên sản phẩm** | Hiện thị thông tin Tên sản phẩm | Không có | Không có |
| **TextView Số lượng** | Hiện thị thông tin Số lượng sản phẩm | Không có | Không có |
| **TextView Màu sắc** | Hiện thị thông tin Màu sắc sản phẩm | Không có | Không có |
| **TextView Mã đơn hàng** | Hiện thị thông tin Mã đơn hàng của sản phẩm | Không có | Không có |
| **Textview Lý do hủy** | Hiện thị thông tin Lý do hủy đơn hàng của sản phẩm khách hàng đã mua | Không có | Không có |
| **TextView Tổng tiền** | Hiện thị thông tin Tổng tiền của các sản phẩm khách hàng đã mua | Không có | Không có |
| **SearchBox Đơn hàng** | Người dùng sử dụng khi muốn tìm kiếm sản phẩm trong danh sách đã hủy | - Chạm vào thanh tìm kiếm  - Gõ ký tự thông tin đơn hàng đã hủy  - Chạm vào biểu tượng “tìm kiếm” | Hiển thị ra thông tin sản phẩm tìm kiếm theo từ khóa đã nhập |
| **Button xác nhận** | Người dùng sử dụng khi muốn thay đổi trạng thái tình trạng đơn hàng thành đã hủy | Chạm vào button xác nhận | Đơn hàng mà người dùng lựa chọn để hủy đã xác nhận hủy thành công |
| **Button hủy** | Người dùng sử dụng khi muốn thay đổi trạng thái tình trạng đơn hàng thành chưa hủy | Chạm vào button hủy | Tình trạng đơn hàng mà người dùng lựa chọn sẽ được xác nhận là bỏ hủy đơn hàng |
| **Button làm mới** | Hiển thị thêm các sản phẩm mới của khách hàng vào danh sách | Chạm vào button làm mới | Cập nhật lại danh sách đơn hàng |
| **Button thoát** | Người dùng sử dụng khi muốn thoát khỏi màn hình danh sách đơn hàng | Chạm vào button thoát | - Thoát khỏi màn hình “Danh sách đơn hàng hủy”  - Trở về màn hình trước đó |

**2.12. 3 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **Thanh toán**

**2.1.12.3.1    User InterfacesGiao diện màn hình Thanh Toán**



*Hình 12: Màn hình Thanh Toán*

**2.1.22.3.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình thanh toán** | Màn hình “Thanh toán “ gồm có:  - TextView Tên sản phẩm  - Combobox Số lượng  - TextView Màu sắc  - TextView Giá tiền  - TextView Tên khách hàng  - TextView Số điện thoại  - TextView Tổng tiền  - TextView Phí vận chuyển  - Image Sản phầm  - Button Hủy  - Button Thanh toán | Không có | Không có |
| **TextView Tên sản phẩm** | Hiện thị thông tin Tên sản phẩm | Không có | Không có |
| **Combobox Số lượng** | Hiện thị thông tin Số lượng sản phẩm | Không có | Không có |
| **TextView Màu sắc** | Hiện thị thông tin Màu sắc sản phẩm | Không có | Không có |
| **TextView Giá tiền sản phẩm** | Hiện thị thông tin Giá tiền sản phẩm | Không có | Không có |
| **TextView Tên khách hàng** | Hiện thị thông tin Tên khách hàng mua sản phẩm | Không có | Không có |
| **TextView Số điện thoại** | Hiện thị thông tin Số điện thoại của khách hàng mua sản phẩm | Không có | Không có |
| **TextView Tổng tiền** | Hiện thị thông tin Tổng tiền các sản phẩm khách hàng đã mua | Không có | Không có |
| **TextView Phí vận chuyển** | Hiện thị thông tin Phí vận chuyển  Đơn hàng của khách hàng | Không có | Không có |
| **Imgae Sản phẩm** | Hiển thị hình ảnh của sản phẩm mà khách hàng chuẩn bị thanh toán | Không có | Không có |
| **Button thanh toán** | Người dùng sử dụng khi muốn thanh toán các sản phẩm đã mua | Chạm vào button thanh toán | - Các sản phẩm được thanh toán thành công - Thoát khỏi màn hình “Thanh toán”  - Quay lại màn hình “Danh sách đơn hàng” để xem các sản phẩm đã mua |
| **Button hủy** | Người dùng sử dụng khi muốn hủy thanh toán đơn hàng nếu khách hàng muốn đổi sản phẩm khác | Chạm vào button hủy | - Thoát khỏi màn hình “Thanh toán”  - Trở về màn hình “Thông tin sản phẩm” |

**2.12. 4 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **Danh sách vận chuyển**

**2.1.12.4.1    User InterfacesGiao diện màn hình Danh sách vận chuyển**



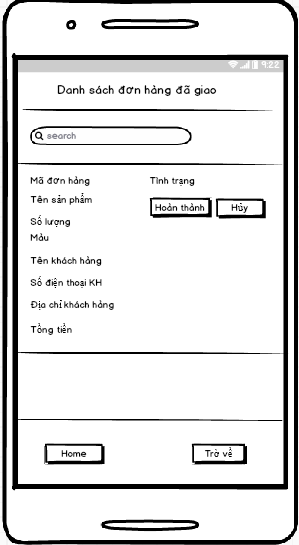
*Hình 13: Màn hình Danh sách vận chuyển*

**2.1.22.4.2            Functional RequirementsYêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình danh sách vận chuyển** | Màn hình “ danh sách vận chuyển**”** gồm có:  - TextView Tên sản phẩm  - TextView Số lượng  - TextView Màu sắc  - TextView Tổng tiền  - TextView Mã đơn hàng  - TextView Tình trạng  - Button Giao hàng  - Button Hủy  - Button Hoàn thành  - Button Thoát  - SeachrBox Sản Phẩm | Không có | Không có |
| **TextView Tên sản phẩm** | Hiện thị thông tin Tên sản phẩm | Không có | Không có |
| **TextView Số lượng** | Hiện thị thông tin Số lượng sản phẩm | Không có | Không có |
| **TextView Tổng tiền** | Hiện thị thông tin Tổng tiền các sản phẩm khách hàng đã mua | Không có | Không có |
| **TextView Mã đơn hàng** | Hiện thị thông tin Mã đơn hàng của sản phẩm | Không có | Không có |
| **TextView Màu sắc** | Hiện thị thông tin Màu sắc sản phẩm | Không có | Không có |
| **TextView Tình trạng đơn hàng** | Hiện thị thông tin Tình trạng đơn hàng  Của khách hàng | Không có | Không có |
| **SearchBox Sản Phẩm** | Người dùng sử dụng khi muốn tìm kiếm sản phẩm trong danh sách vận chuyển | - Chạm vào thanh tìm kiếm  - Gõ ký tự thông tin đơn hàng vận chuyển  - Chạm vào biểu tượng “tìm kiếm” | Hiển thị ra thông tin sản phẩm tìm kiếm theo từ khóa đã nhập |
| **Button hủy** | Người dùng sử dụng khi muốn hủy giao hàng nếu khách hàng không muốn mua sản phẩm | Chạm vào button hủy | Đơn hàng sẽ được xác nhận trạng thái hủy giao đơn hàng |
| **Button Giao hàng** | Người dùng sử dụng khi muốn giao hàng nếu khách hàng muốn mua sản phẩm | Chạm vào button hủy | Đơn hàng sẽ được xác nhận trạng thái giao đơn hàng |
| **Button hoàn thành** | Người dùng sử dụng khi muốn xác nhận tất cả các đơn hàng chuẩn bị giao. | Chạm vào button hoàn thành | Đơn hàng sẽ được lưu trữ các thông tin để chuẩn bị giao |
| **Button thoát** | Người dùng sử dụng khi muốn thoát khỏi màn hình danh sách vận chuyển | Chạm vào button thoát | - Thoát khỏi màn hình “Danh sách vẫn chuyển”  - Trở về màn hình danh sách đơn hàng xác nhận |

**2.12. 5 Feature/Component #1:MyMobileRegistrationscreenTính năng / Thành phần # 1:** **Màn hình** **Danh sách đơn hàng đã giao**

**2.1.12.5.1    User InterfacesGiao diện màn hình Danh sách đơn hàng đã giao**



*Hình 14: Màn hình Danh sách đơn hàng đã giao*

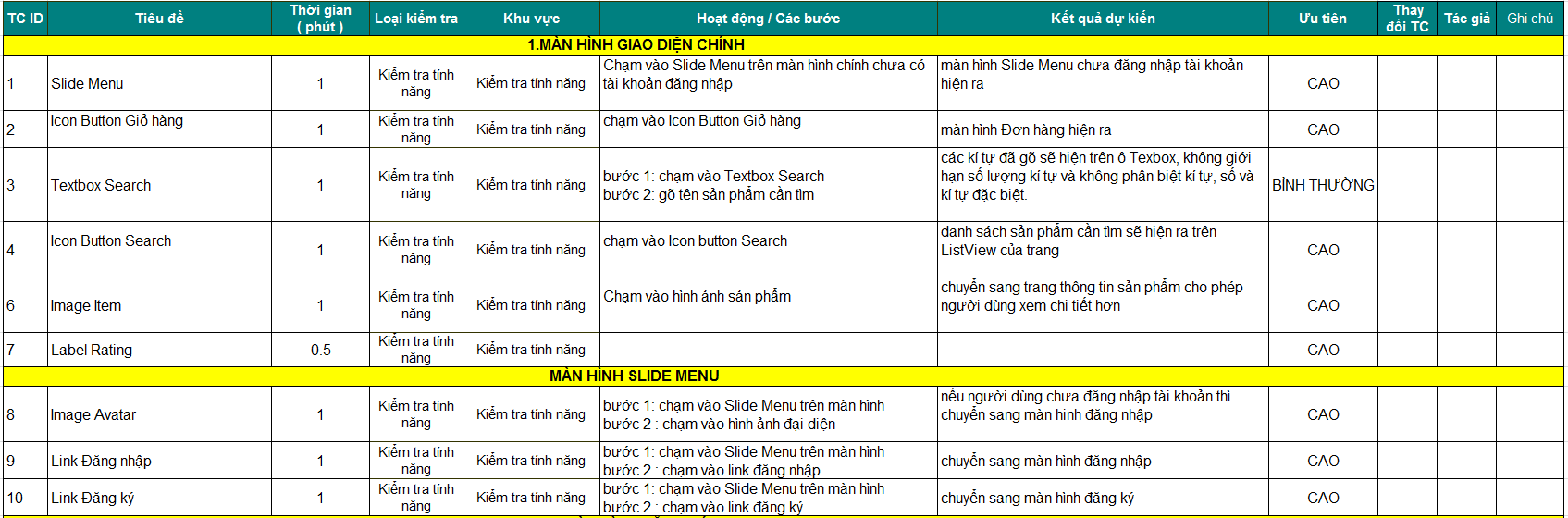
**2.1.22.5.2            Functional RequirementsYêu cầu chức năng**

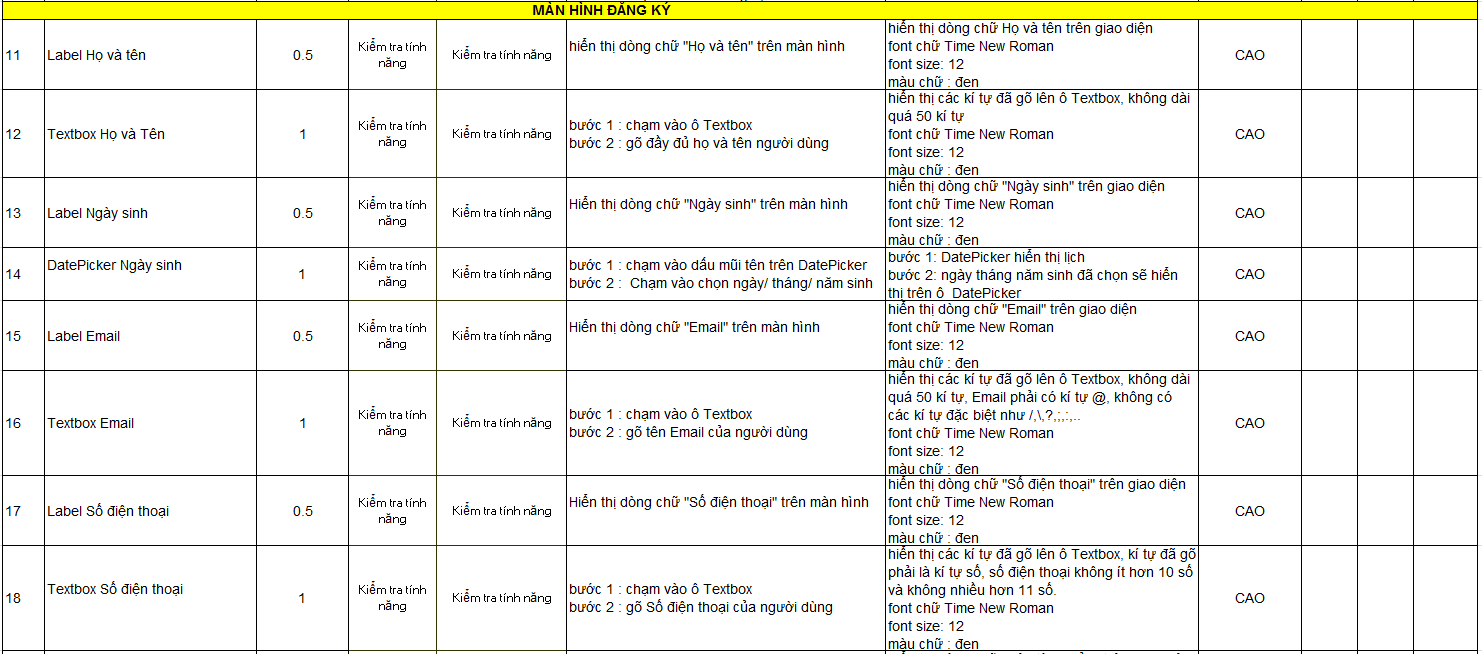
|  |  |  |  |
| --- | --- | --- | --- |
| **Màn hình** | **Mô tả** | **Hoạt động** | **Phản ứng** |
| **Màn hình danh sách đơn hàng hoàn thành** | Màn hình “danh sách đơn hàng hoàn thành” gồm có:  - TextView Tên sản phẩm  - TextView Số lượng  - TextView Màu sắc  - TextView Tên khách hàng  - TextView Số điện thoại  - TextView Tổng tiền  - TextView Mã đơn hàng  - TextView Địa chỉ  - TextView Tình trạng  - Searchbox Sản Phẩm  - Button Home  - Button Trở về  - Button Hoàn thành  - Button Hủy | N/A | N/A |
| **TextView Tên sản phẩm** | Hiện thị thông tin Tên sản phẩm | Không có | Không có |
| **TextView Số lượng** | Hiện thị thông tin Số lượng sản phẩm | Không có | Không có |
| **TextView Tổng tiền** | Hiện thị thông tin Tổng tiền các sản phẩm khách hàng đã mua | Không có | Không có |
| **TextView Mã đơn hàng** | Hiện thị thông tin Mã đơn hàng của sản phẩm | Không có | Không có |
| **TextView Màu sắc** | Hiện thị thông tin Màu sắc sản phẩm | Không có | Không có |
| **TextView Tình trạng đơn hàng** | Hiện thị thông tin Tình trạng đơn hàng  Của khách hàng | Không có | Không có |
| **TextView Tên khách hàng** | Hiện thị thông tin Tên khách hàng mua sản phẩm | Không có | Không có |
| **TextView Số điện thoại** | Hiện thị thông tin Số điện thoại của khách hàng mua sản phẩm | Không có | Không có |
| **TextView Địa chỉ** | Hiện thị thông tin Địa chỉ của khách hàng mua sản phẩm | Không có | Không có |
| **SearchBox Sản phẩm** | Người dùng sử dụng khi muốn tìm kiếm sản phẩm trong danh sách vận chuyển | - Chạm vào thanh tìm kiếm  - Gõ ký tự thông tin đơn hàng đã giao  - Chạm vào biểu tượng “tìm kiếm” | Hiển thị ra thông tin sản phẩm tìm kiếm theo từ khóa đã nhập |
| **Button hoàn thành** | Người dùng sử dụng khi muốn xác nhận đơn hàng đã được giao cho khách hành thành công | Chạm vào button hoàn thành | Thay đổi trạng thái thành hoàn thành |
| **Button hủy** | Người dùng sử dụng khi muốn hủy đơn hàng chưa được giao | Chạm vào button hủy | Đơn hàng sẽ cập nhật trạng thái là hủy đơn hàng |
| **Button Home** | Thoát khỏi màn hình danh sách đơn hàng hoàn thành | Chạm vào button home | - Thoát khỏi màn hình “danh sách đơn hàng hoàn thành”  - Trở về màn hình chính |
| **Button trở về** | Người dùng sử dụng khi muốn thoát khỏi màn hình danh sách đơn hàng hoàn thành | Chạm vào button thoát | - Thoát khỏi màn hình “danh sách đơn hàng hoàn thành”  - Trở về màn hình “Danh sách vận chuyển” |
| **Button Giao hàng** | Người dùng sử dụng khi muốn xác nhận giao đơn hàng | Chạm vào button hủy | Đơn hàng sẽ cập nhật trạng thái là xác nhận vận chuyển đơn hàng |

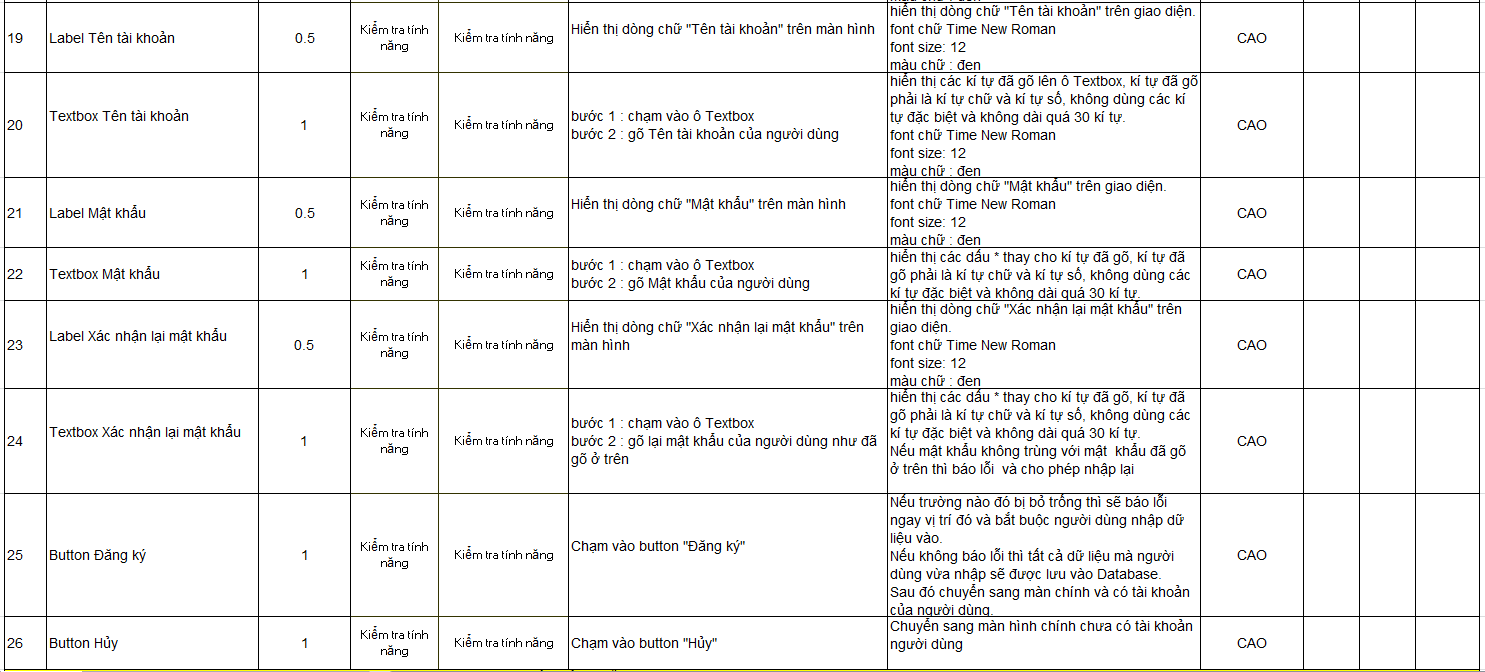
# CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ

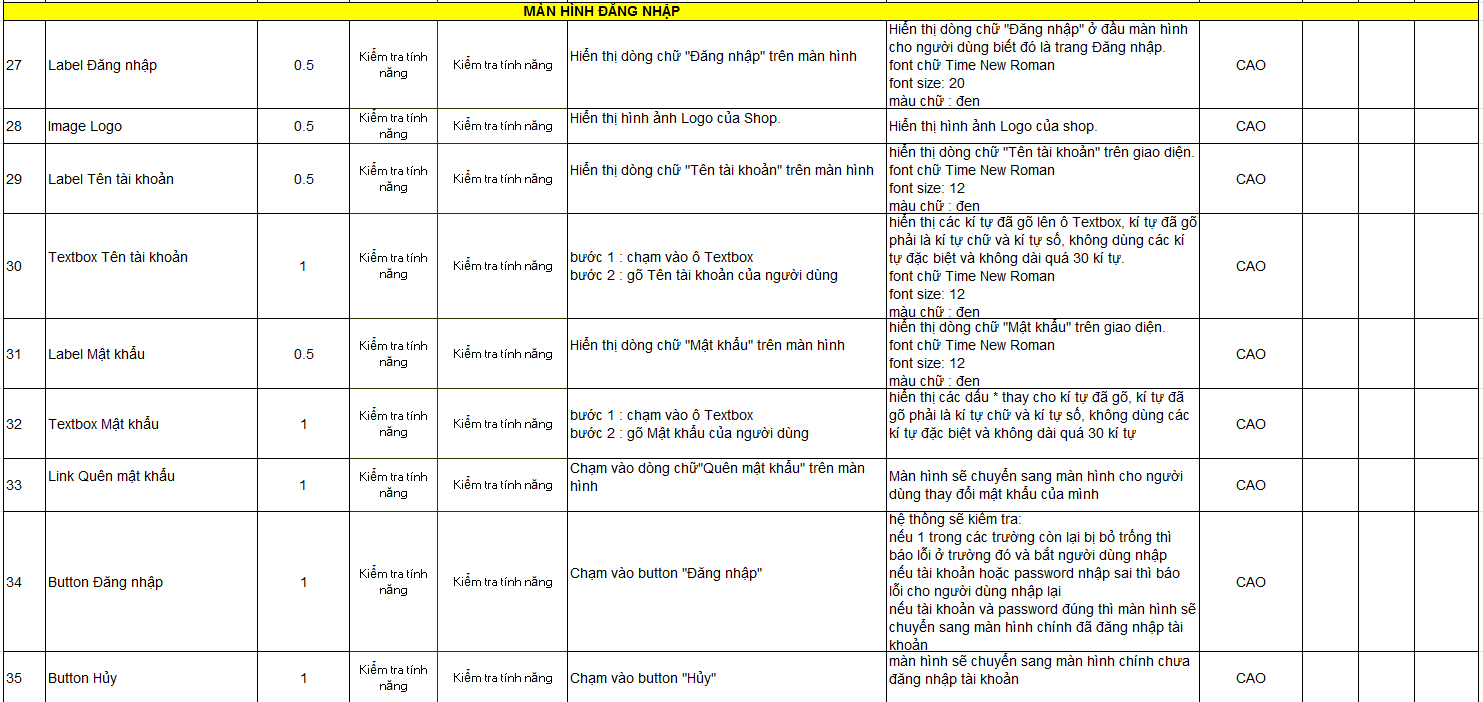
## 3.1 Cài đặt

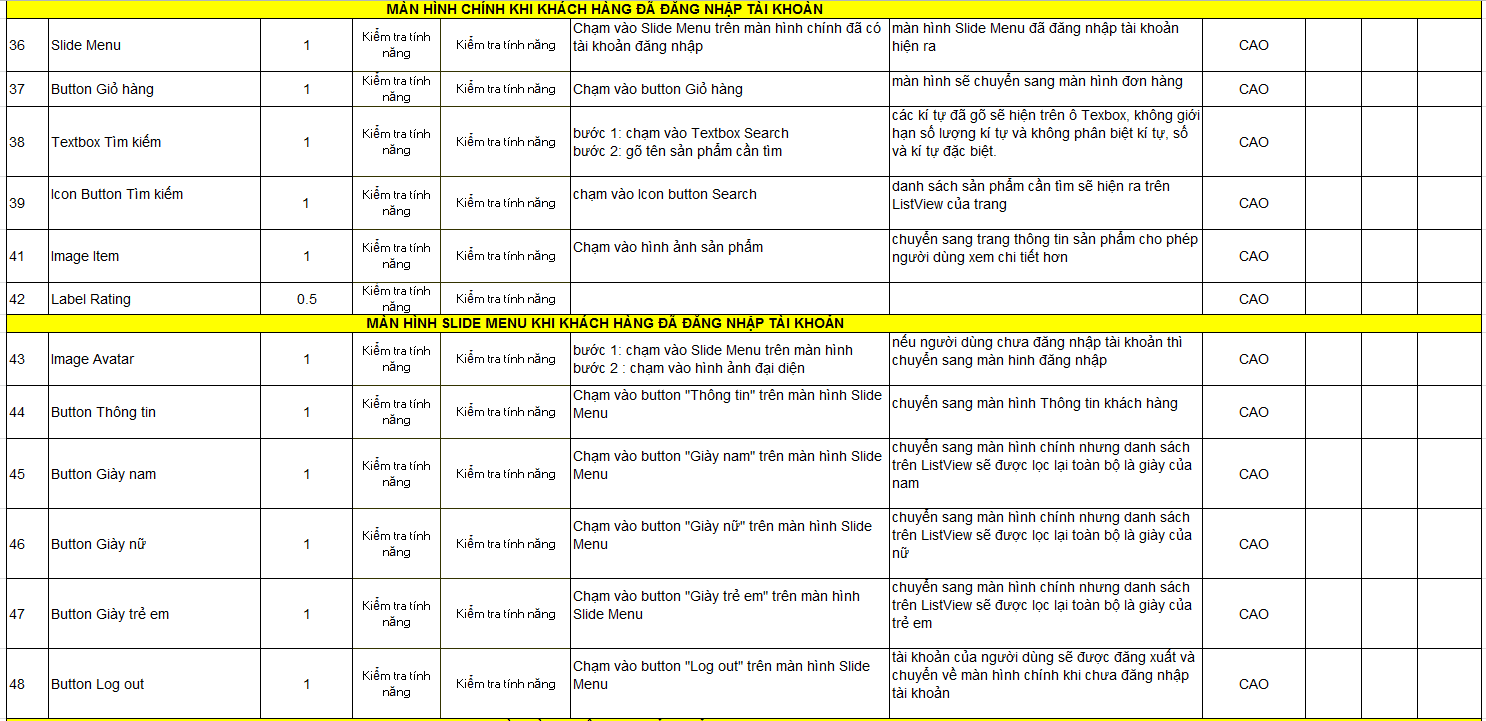
## 3.2 Kiểm thử

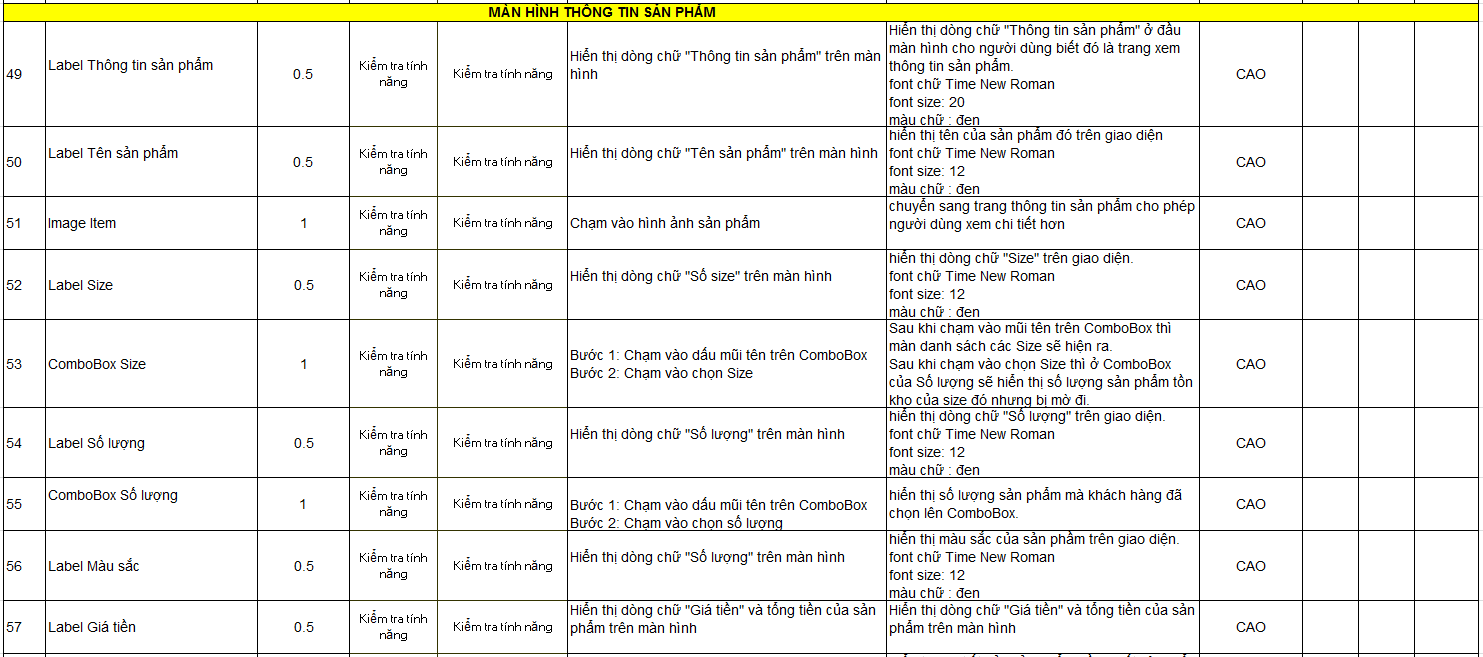


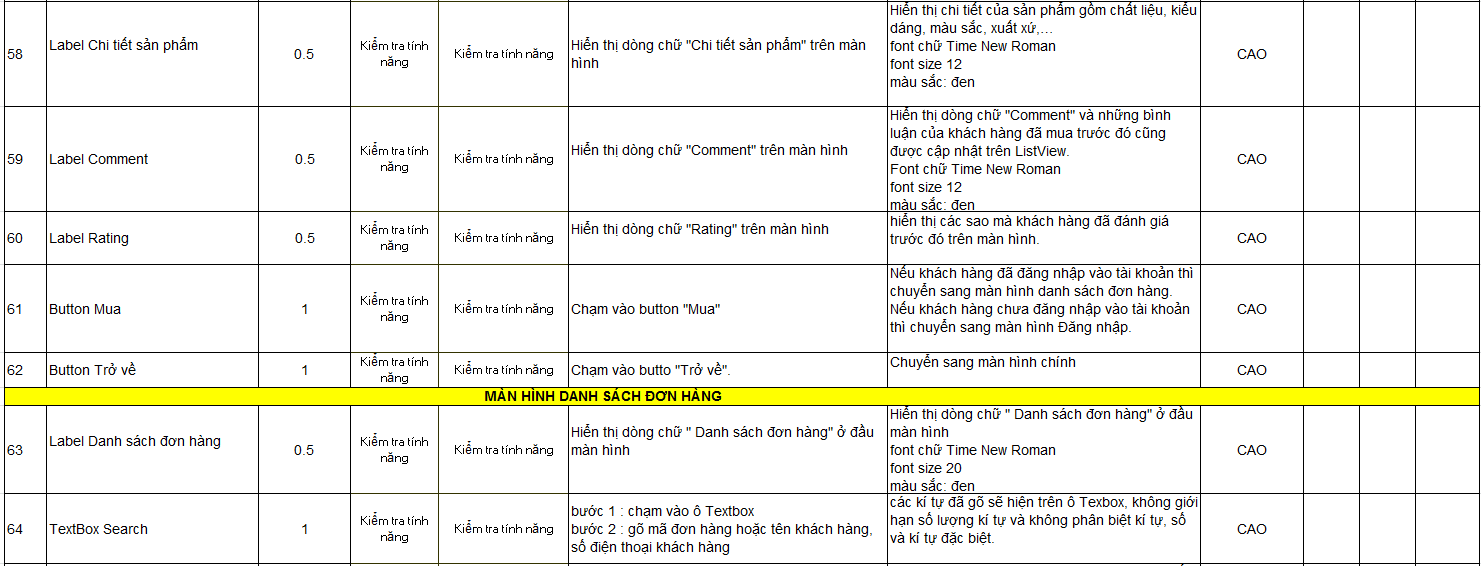


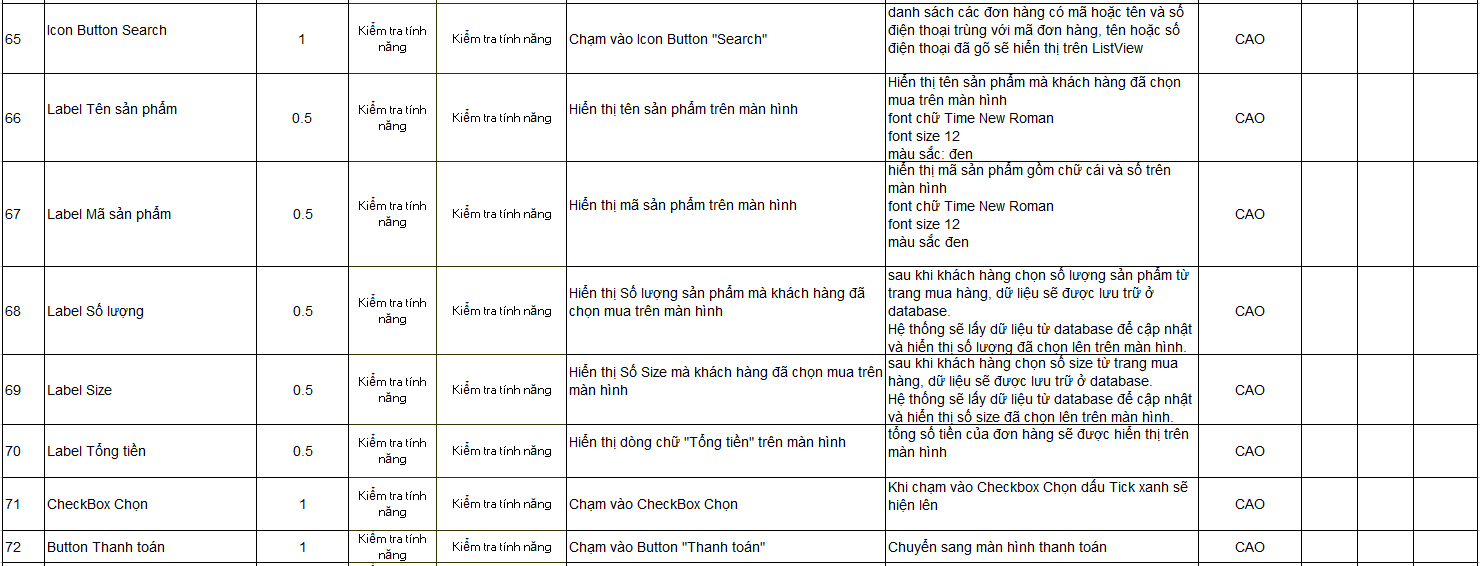


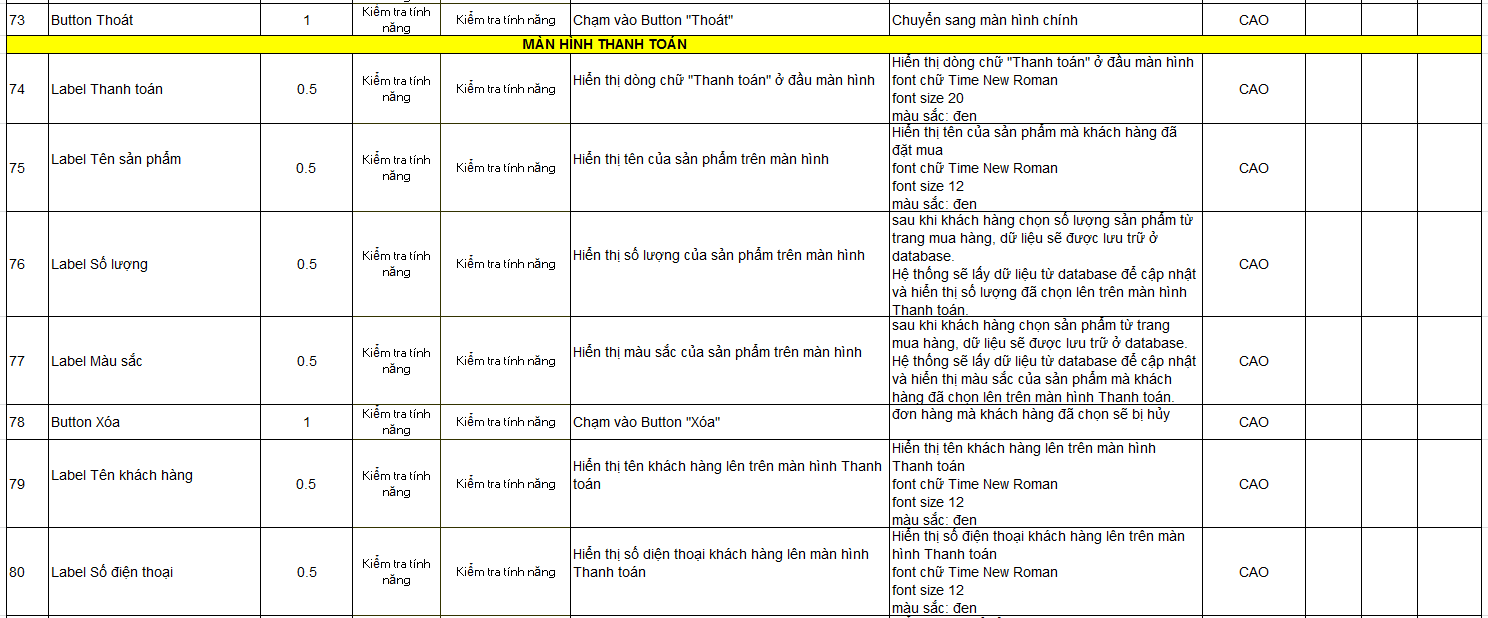


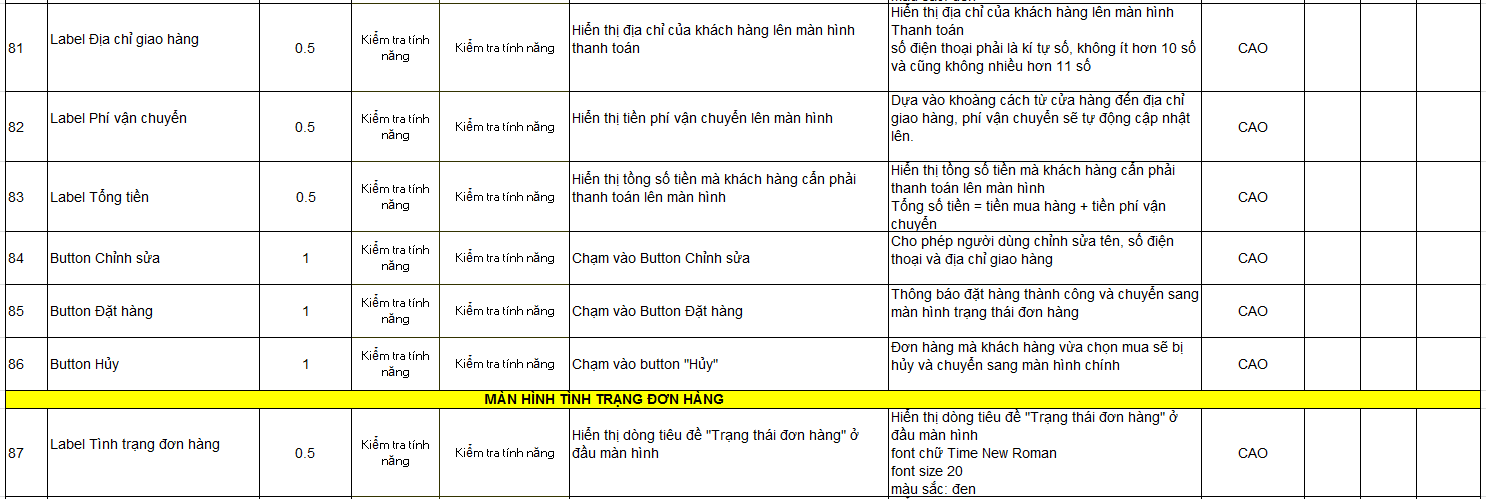


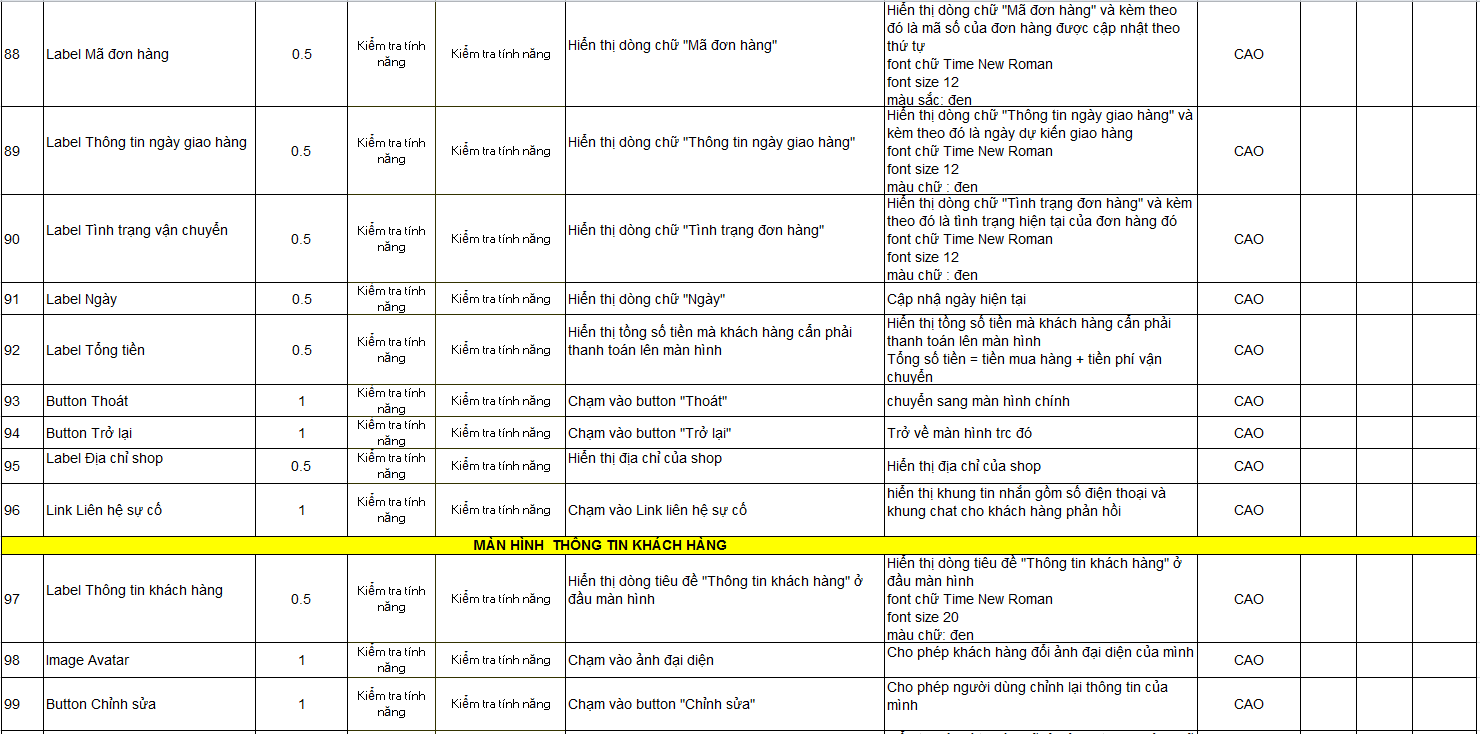




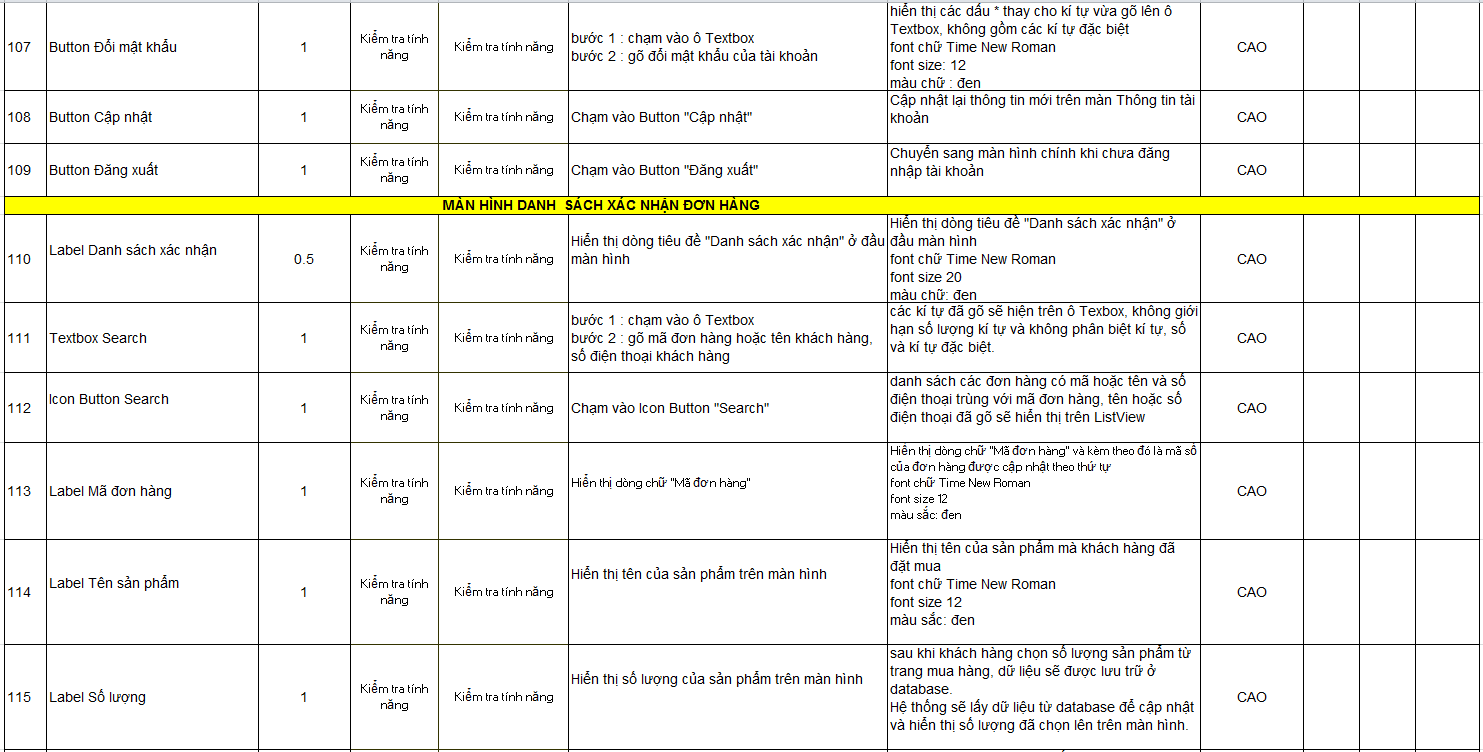


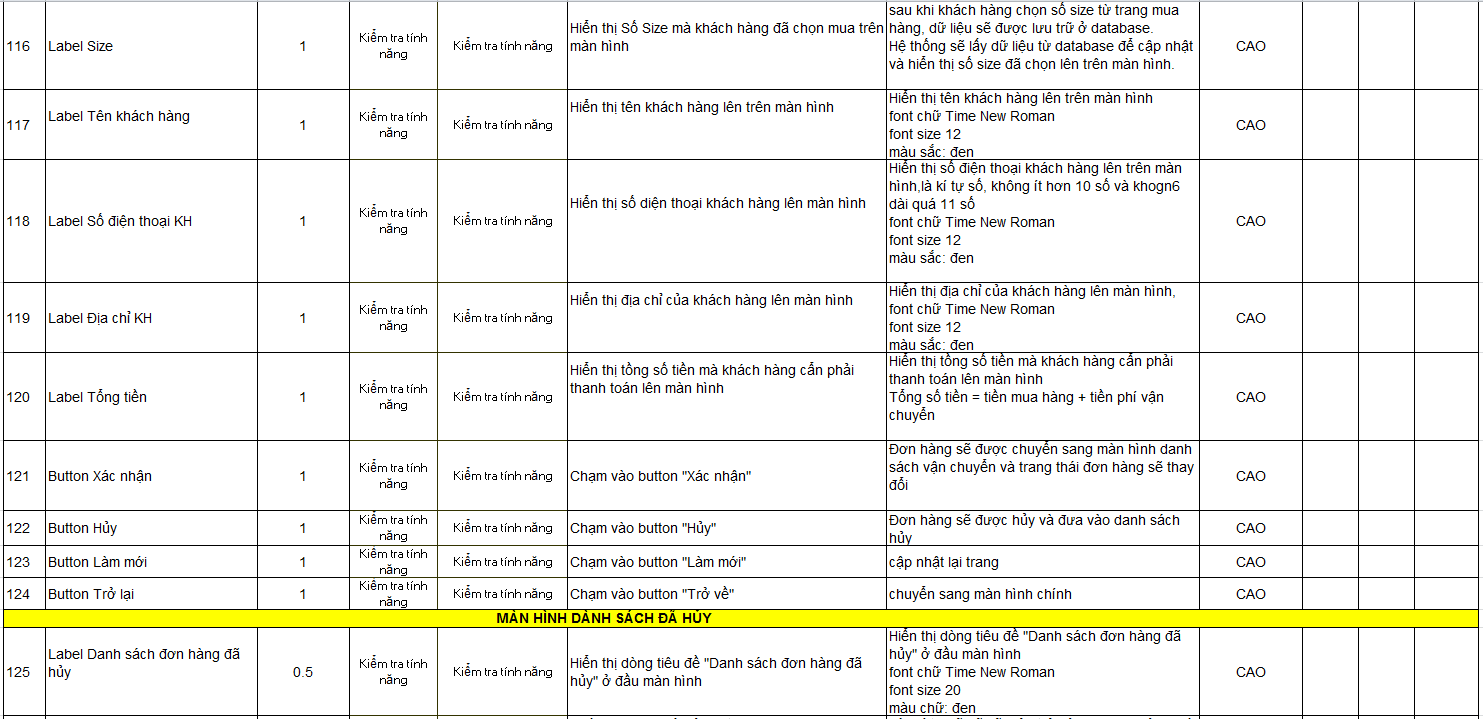


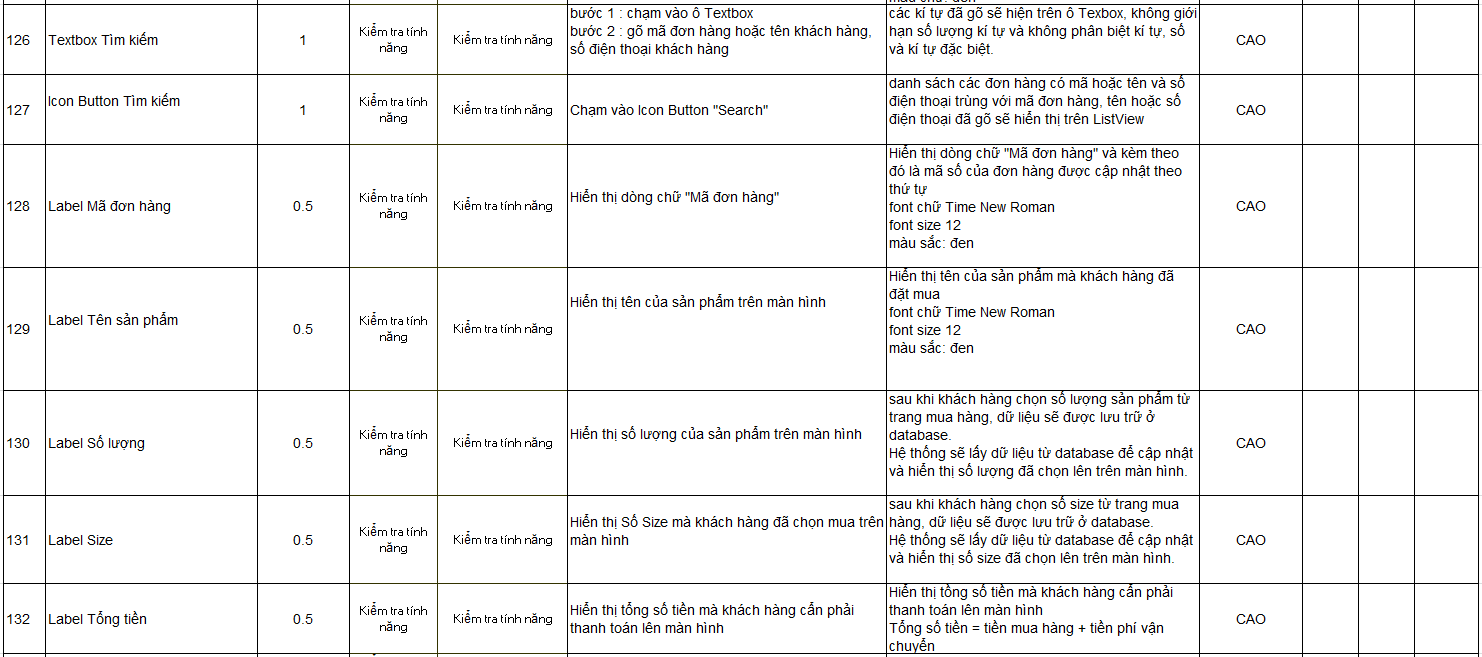


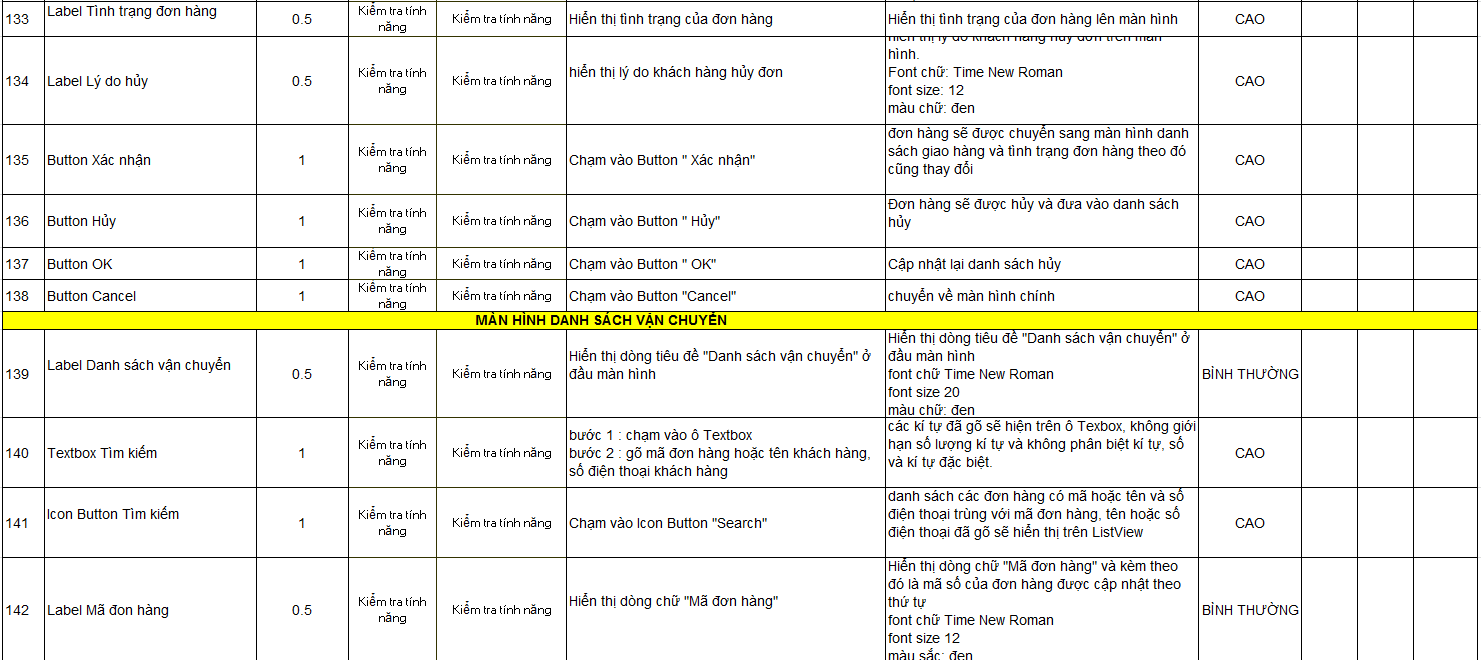


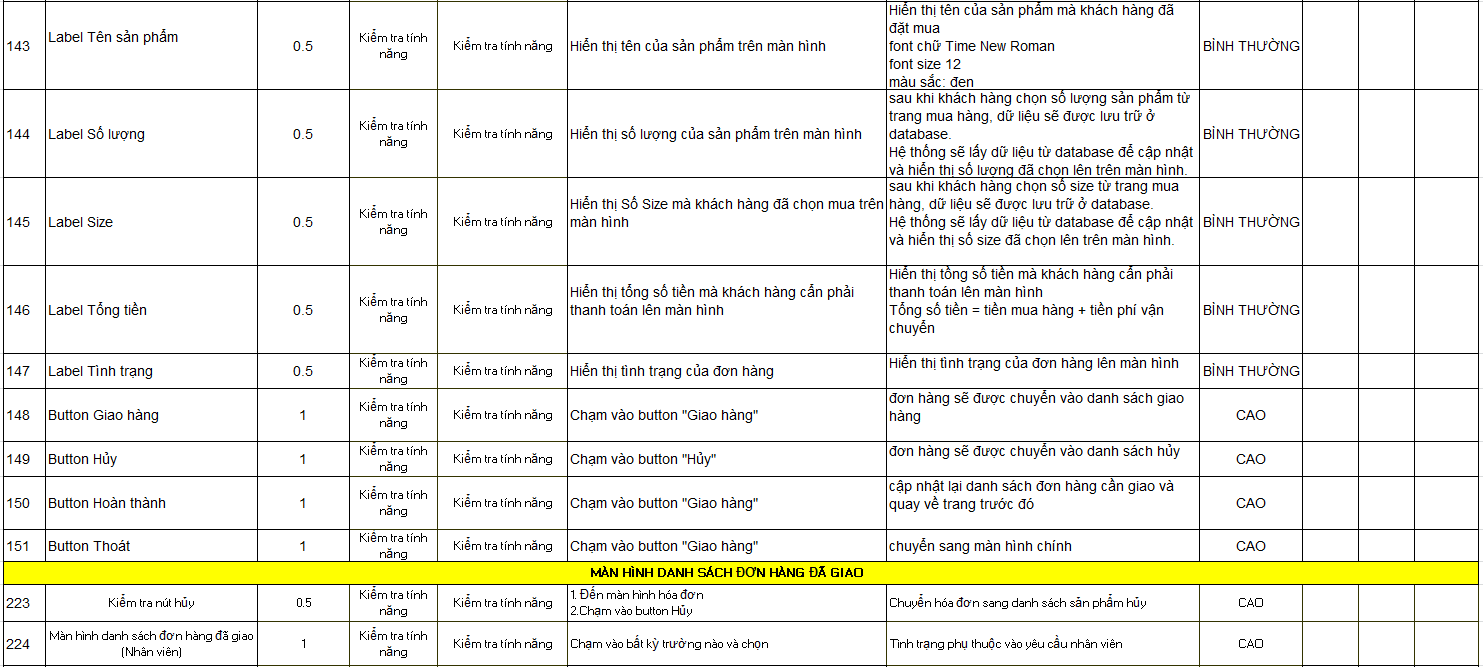


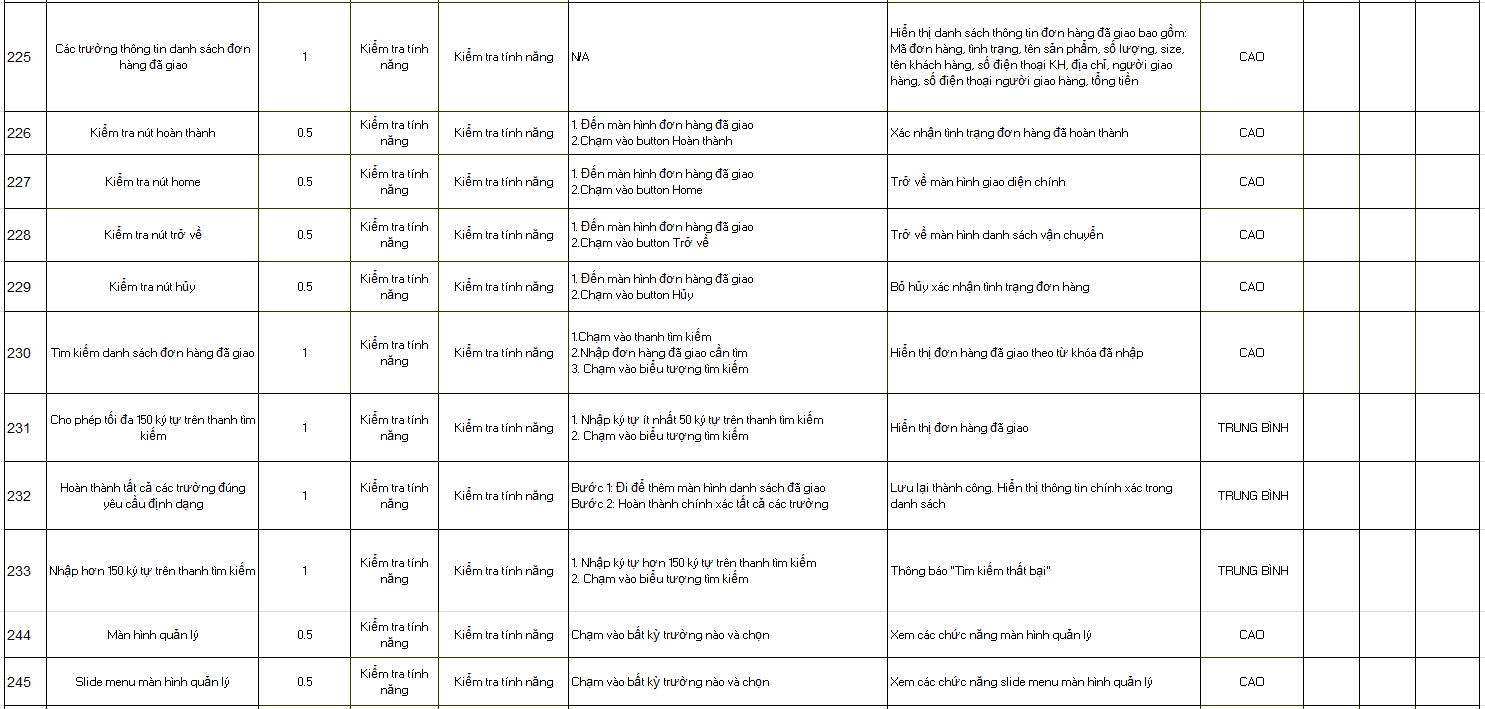


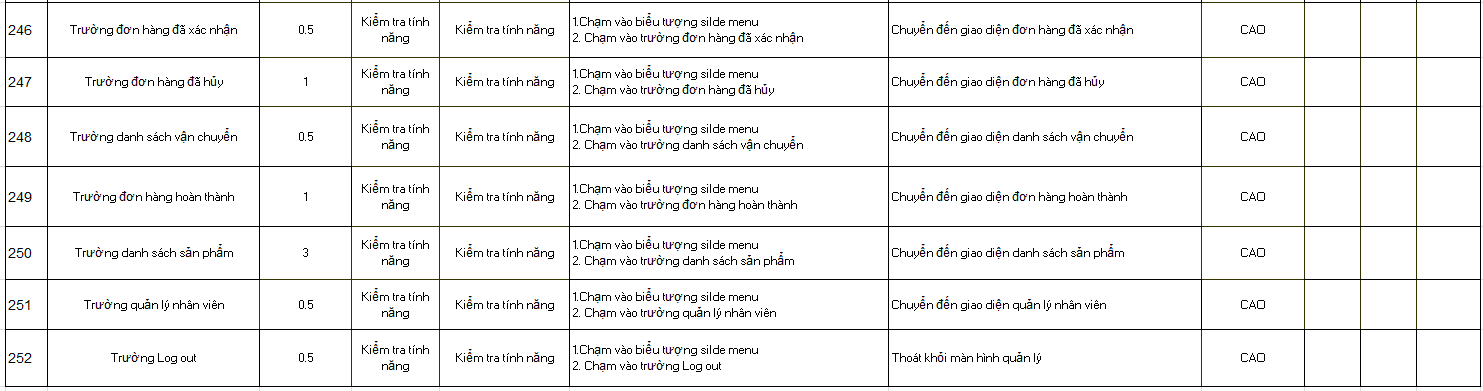












# 

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

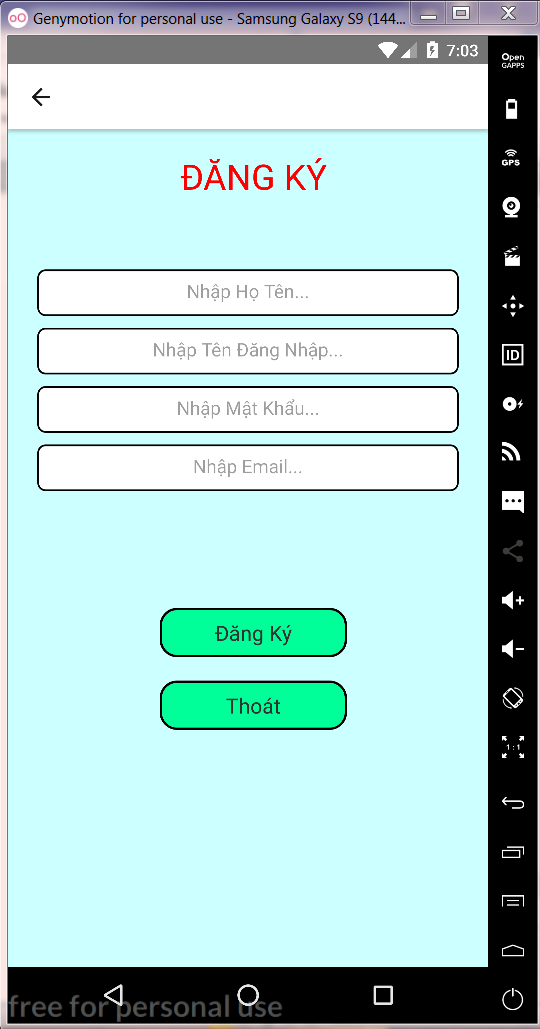
## 4.1 Kết quả đạt được

*Hình 1: Màn hình danh sách sản phẩm*

## 

## 

*Hình 2:Màn hình đăng nhập*



*Hình 3: Màn hình đăng ký*

4.2 Các kết luận và kiến nghị

- Những điểm đã làm được

* Sử dụng được flatlist fetch dữ liệu từ webservice đổ vào
* Hiểu được một số tính năng, các component của react native

- Những điểm chưa làm được

* Kết nối các chức năng chưa được logic
* Còn giao diện chưa xử lý kịp

- Các chức năng bổ sung nếu có thêm thời gian…

Thiết kế lại giao diện và thêm đầy đủ tính năng của 1 App Bán Đồng hồ

# 

TÀI LIỆU THAM KHẢO

* <https://facebook.github.io/react-native/>
* <https://facebook.github.io/react-native/docs/navigation#react-navigation>