



BÀI TẬP

Chuyên đề

React Native cơ bản





Bài 1 Giới thiệu và ứng dụng mở đầu

1.1 Xây dựng ứng dụng Mở đầu

- Yêu cầu: Xây dựng màn hình lời chào:





- Hướng dẫn:

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

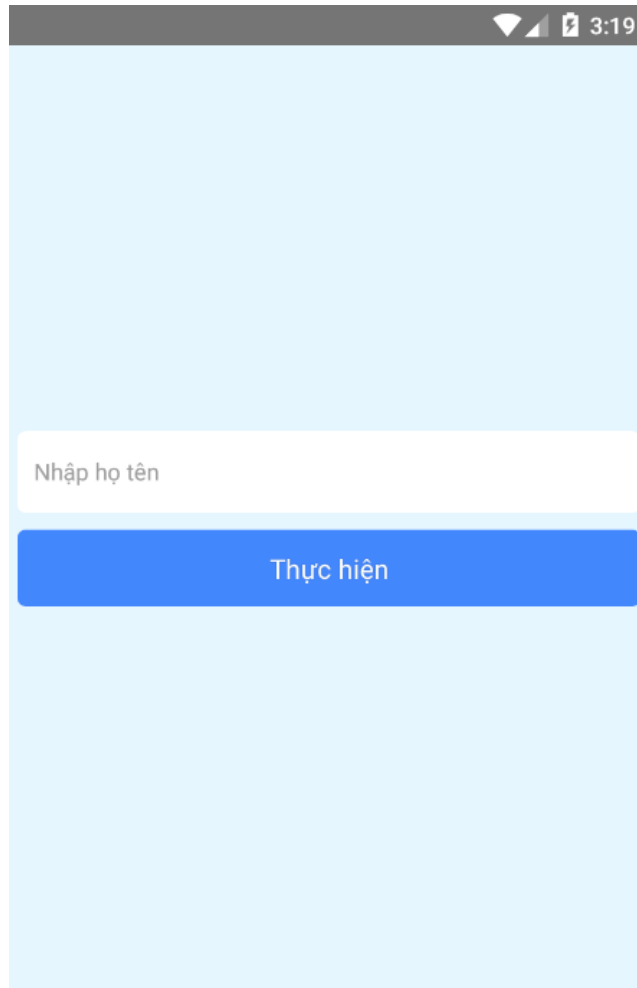
export default class App extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.welcome}>Chào bạn đến với React Native!</Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  welcome: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  }
});
```

1.2 Xây dựng màn hình xuất lời chào

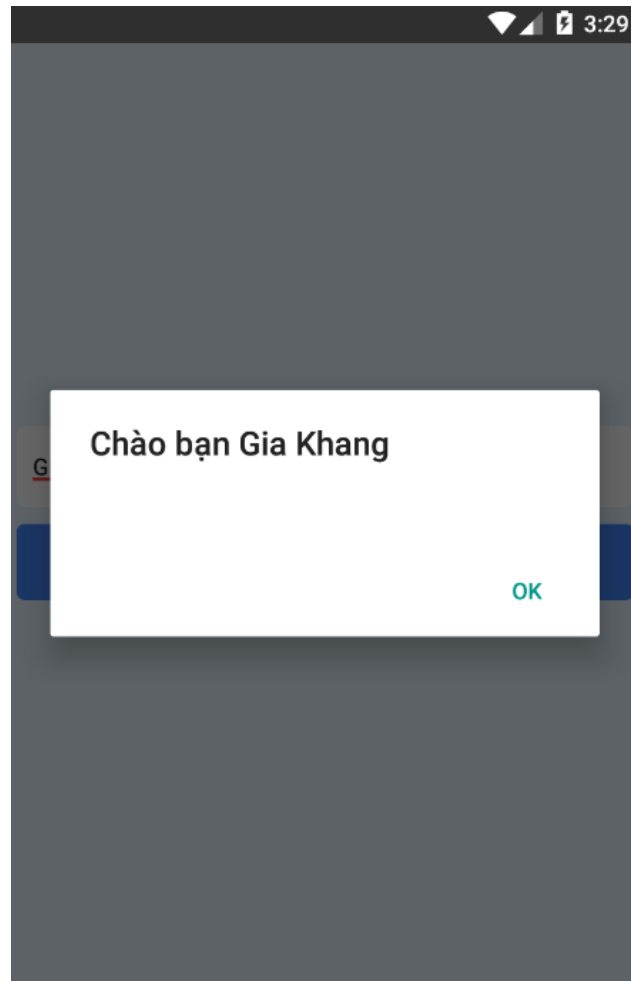
- Yêu cầu:

- Xây dựng màn hình xuất lời chào:

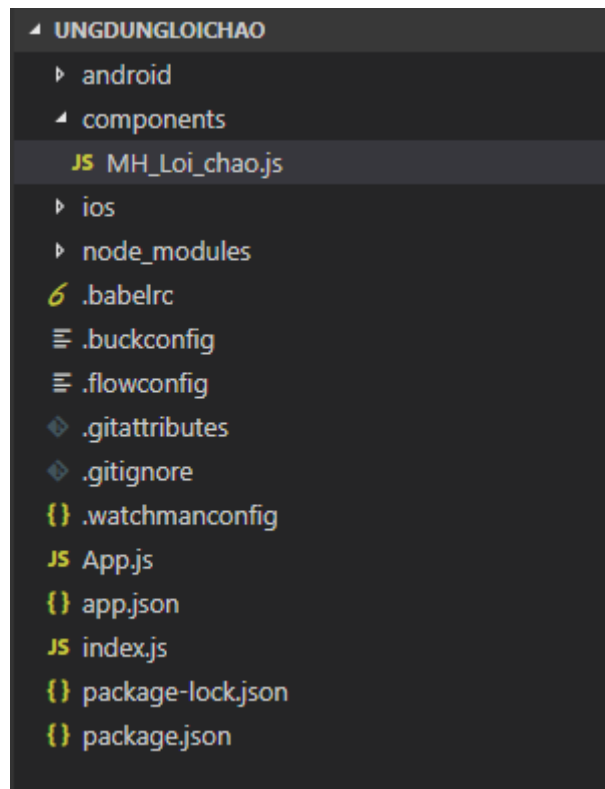


The image shows a mobile application interface. At the top, there is a dark grey status bar with icons for Wi-Fi, cellular signal, and battery, along with the time 3:19. Below this is a large light blue rectangular area. In the center of this area is a white rounded rectangular input field with the placeholder text "Nhập họ tên". Directly below the input field is a solid blue rounded rectangular button with the white text "Thực hiện".

- Nhập họ tên người dùng
- Khi click vào nút “Thực hiện” ứng dụng sẽ xuất lời chào đến người dùng



- Hướng dẫn:
 - Tạo thư mục components và tập tin MH_Xuat_loi_chao.js



- Thiết kế và cài đặt sự kiện khi người dùng nhấn vào “nút thực hiện”



```
import React, {Component} from 'react';
import {View,TouchableOpacity,Text, TextInput,StyleSheet, Alert} from 'react-native';

var Ho_ten="";
export default class MH_Loi_chao extends Component{
  XL_Nhan(){
    Alert.alert("Chào bạn " + Ho_ten);
  }
  render(){
    return(
      <View style={style.container}>
        <TextInput style={style.input} placeholder='Nhập họ tên' onChangeText={(value)->Ho_ten=value}/>
        <TouchableOpacity onPress={this.XL_Nhan.bind(this)} activeOpacity={0.5}>
          <View style={style.button}>
            <Text style={style.text}>Thực hiện</Text>
          </View>
        </TouchableOpacity>
      </View>
    )
  }
}

const style= StyleSheet.create({
  container:{
    flex:1,
    backgroundColor:'#e5f6ff',
    justifyContent:'center'
  },
  input:{
    paddingHorizontal:10,
    backgroundColor:'#ffffff',
    margin:5,
    borderRadius:5
  },
  button:{
    height:46,
    borderRadius:5,
    margin:5,
    backgroundColor:'#4387fd',
    justifyContent:'center',
    alignItems:'center'
  },
  text:{
    color:'#ffffff',
    fontSize:16
  }
})
})
```



Bài 2 Lập trình JavaScript (ES6)

2.1 Tạo tập tin dữ liệu

- Yêu cầu:
 - Tạo tập tin dữ liệu Danh_sach_Tivi.js theo cấu trúc sau:

```
[
  {
    "Ten": "Smart Tivi Cong Samsung 49 inch",
    "Ma_so": "TIVI_1",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_1.png",
    "Don_gia": 22900000,
    "Danh_sach_Ban_hang": [
      {
        "Ma_so": "PB_1",
        "Ngay": "2018-10-10",
        "So_luong": 2,
        "Don_gia": 22900000
      },
      {
        "Ma_so": "PB_2",
        "Ngay": "2018-10-10",
        "So_luong": 1,
        "Don_gia": 22900000
      }
    ]
  },
  {
    "Ten": "Internet Tivi Sony 4K 43 inch",
    "Ma_so": "TIVI_4",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_4.png",
    "Don_gia": 14500000,
```




```

        "Danh_sach_Ban_hang":[]
    },
    {
        "Ten":"Internet Tivi Samsung 32 inch",
        "Ma_so":"TIVI_8",
        "Hinh":"http://10.0.3.2:8081/images/TIVI_8.png",
        "Don_gia":6890000,
        "Danh_sach_Ban_hang":[]
    },
    {
        "Ten":"Internet Tivi Sony 40 inch",
        "Ma_so":"TIVI_10",
        "Hinh":"http://10.0.3.2:8081/images/TIVI_10.png",
        "Don_gia":9990000,
        "Danh_sach_Ban_hang":[]
    },
    {
        "Ten":"Smart Tivi Samsung 40 inch",
        "Ma_so":"TIVI_11",
        "Hinh":"http://10.0.3.2:8081/images/TIVI_11.png",
        "Don_gia":8640000,
        "Danh_sach_Ban_hang":[]
    },
    {
        "Ten":"Smart Tivi LED 4K SONY 49 inch",
        "Ma_so":"TIVI_18",
        "Hinh":"http://10.0.3.2:8081/images/TIVI_18.png",
        "Don_gia":10420000,
        "Danh_sach_Ban_hang":[]
    },
    {

```



```

    "Ten": "Smart Tivi LED 4K SONY 75 Inch",
    "Ma_so": "TIVI_20",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_20.png",
    "Don_gia": 7340000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "Smart Tivi SONY Ultra HD 75 Inch",
    "Ma_so": "TIVI_22",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_22.png",
    "Don_gia": 11480000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "Internet Tivi LG 55 Inch",
    "Ma_so": "TIVI_32",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_32.png",
    "Don_gia": 11480000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "TV Philips",
    "Ma_so": "TIVI_34",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_34.png",
    "Don_gia": 10420000,
    "Danh_sach_Ban_hang": []
  }
]

```

2.2 Thực hiện truy vấn xem thông tin

- Yêu cầu:



- Tìm Tivi có mã số "TIVI_32", thông tin trả về gồm: Tên, Mã số, Hình, Đơn giá
- Lọc danh sách các Tivi 49 inch, thông tin trả về gồm: Tên, Mã số, Hình, Đơn giá, Danh sách bán hàng
- Lọc danh sách các Tivi có đơn giá ≤ 3000000 , thông tin trả về gồm: Tên, Mã số, Hình, Đơn giá. Sắp xếp theo đơn giá giảm dần
- Lọc danh sách các Tivi được bán trong ngày 10/10/2018, thông tin trả về gồm: Tên, Mã số, Hình, Đơn giá, Danh sách bán hàng. Sắp xếp theo tên Tivi
- Lọc danh sách các Tivi không bán được trong ngày 10/10/2018, thông tin trả về gồm: Tên, Mã số, Hình, Đơn giá, Danh sách bán hàng. Sắp xếp tên Tivi
- Lọc danh sách các Tivi 49 inch được bán từ ngày 10/10/2018 đến 20/10/2018, thông tin trả về gồm: Tên, Mã số, Hình, Đơn giá, Danh sách bán hàng. Sắp xếp theo mã số Tivi
- Tính tổng số lượng bán của các Tivi trong ngày 10/10/2018, thông tin gồm: Tên, Mã, Hình, Đơn giá, Tổng số lượng bán
- Cho biết 3 Tivi có doanh thu cao nhất trong ngày 10/10/2018

2.3 Thực hiện truy vấn cập nhật thông tin

- Yêu cầu:
 - Thêm mới Tivi vào trong danh sách Tivi, thông tin gồm:
 Tên: "Internet Tivi SONY Ultra HD 55 inch"
 Mã số: "TIVI_24"
 Hình: "http://10.0.3.2:8081/images/TIVI_24.png"
 Đơn giá: 8640000
 Danh sách Bán hàng: []
 - Cập nhật đơn giá Tivi có mã số "TIVI_24" thành 8600000
 - Xóa Tivi có mã số "TIVI_24"
 -

2.4 Xây dựng hàm

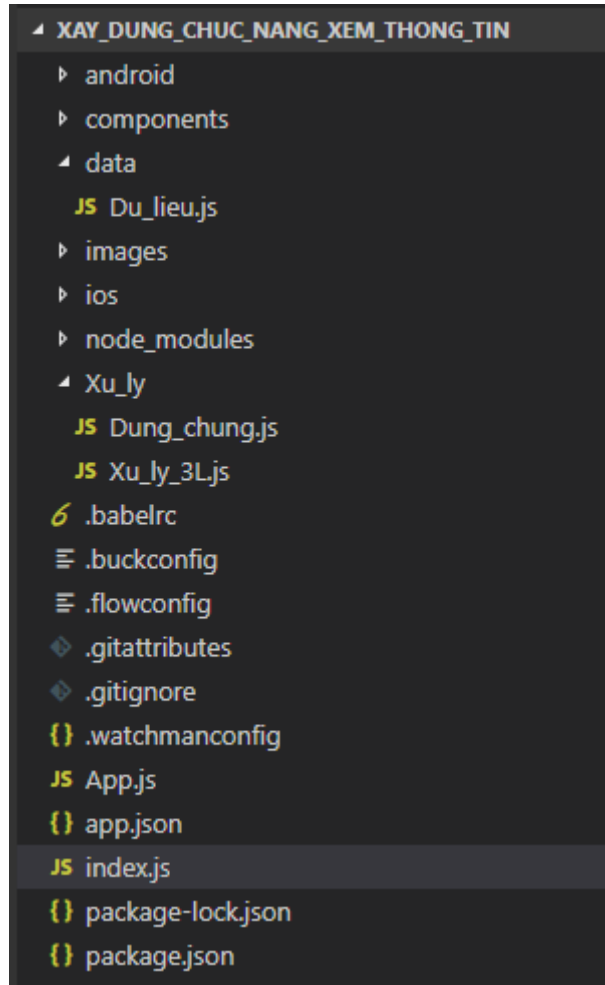
- Yêu cầu:
 - Xây dựng hàm Xóa Tivi trong danh sách Tivi với tham số nhận vào là Mã số Tivi cần xóa.
 - Xây dựng hàm Thêm mới Tivi vào trong danh sách Tivi với tham số nhận vào là đối tượng Tivi cần thêm.
 - Xây dựng hàm Cập nhật Tivi (dựa vào Mã số Tivi) trong danh sách Tivi với tham số nhận vào là đối tượng Tivi cần cập nhật.



Bài 3 Chức năng Xem thông tin

3.1 Tạo tập tin dữ liệu

- Yêu cầu:
 - Tổ chức cây thư mục



- Tạo tập tin dữ liệu Du_lieu.js theo cấu trúc sau:

```
var Du_lieu = {
  Nguai_dung:{
    "Ho_ten": "Tô thị bé Bé",
    "Ma_so": "NV_2",
    "Ten_Dang_nhap": "NVBH",
    "Mat_khau": "NVBH",
    "Nhom_Nguoi_dung": {
```



```

        "Ten": "Nhân viên Bán hàng",
        "Ma_so": "NHAN_VIEN_BAN_HANG"
    }
},
Danh_sach_Tivi: [
    {
        "Ten": "Smart Tivi Cong Samsung 49 inch",
        "Ma_so": "TIVI_1",
        "Hinh": "http://10.0.3.2:8081/images/TIVI_1.png",
        "Don_gia": 22900000,
        "Danh_sach_Ban_hang": [
            {
                "Ma_so": "PB_1",
                "Ngay": "2018-10-10",
                "So_luong": 2,
                "Don_gia": 22900000
            },
            {
                "Ma_so": "PB_2",
                "Ngay": "2018-10-10",
                "So_luong": 1,
                "Don_gia": 22900000
            }
        ]
    }
],
{
    "Ten": "Internet Tivi Sony 4K 43 inch",
    "Ma_so": "TIVI_4",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_4.png",
    "Don_gia": 14500000,
    "Danh_sach_Ban_hang": []
}

```



```

    },
    {
      "Ten": "Internet Tivi Samsung 32 inch",
      "Ma_so": "TIVI_8",
      "Hinh": "http://10.0.3.2:8081/images/TIVI_8.png",
      "Don_gia": 6890000,
      "Danh_sach_Ban_hang": []
    },
    {
      "Ten": "Internet Tivi Sony 40 inch",
      "Ma_so": "TIVI_10",
      "Hinh": "http://10.0.3.2:8081/images/TIVI_10.png",
      "Don_gia": 9990000,
      "Danh_sach_Ban_hang": []
    },
    {
      "Ten": "Smart Tivi Samsung 40 inch",
      "Ma_so": "TIVI_11",
      "Hinh": "http://10.0.3.2:8081/images/TIVI_11.png",
      "Don_gia": 8640000,
      "Danh_sach_Ban_hang": []
    },
    {
      "Ten": "Smart Tivi LED 4K SONY 49 inch",
      "Ma_so": "TIVI_18",
      "Hinh": "http://10.0.3.2:8081/images/TIVI_18.png",
      "Don_gia": 10420000,
      "Danh_sach_Ban_hang": []
    },
    {
      "Ten": "Smart Tivi LED 4K SONY 75 Inch",

```



```

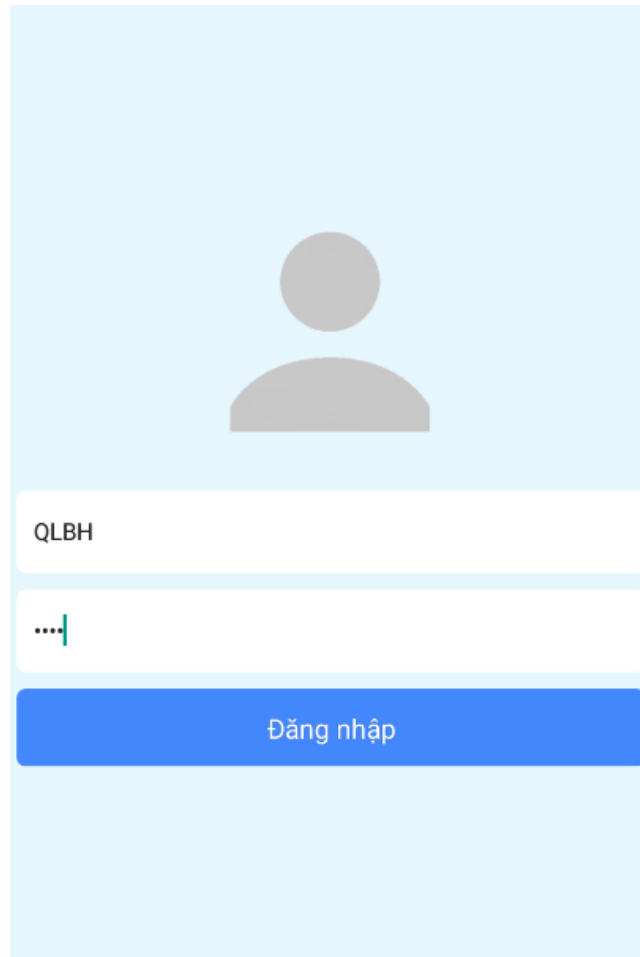
    "Ma_so": "TIVI_20",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_20.png",
    "Don_gia": 7340000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "Smart Tivi SONY Ultra HD 75 Inch",
    "Ma_so": "TIVI_22",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_22.png",
    "Don_gia": 11480000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "Internet Tivi LG 55 Inch",
    "Ma_so": "TIVI_32",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_32.png",
    "Don_gia": 11480000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "TV Philips",
    "Ma_so": "TIVI_34",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_34.png",
    "Don_gia": 10420000,
    "Danh_sach_Ban_hang": []
  }
]
}

module.exports = Du_lieu;

```

3.2 Xây dựng màn hình Đăng nhập

- Yêu cầu:
 - Xây dựng màn hình:



A mockup of a user login form. It features a light blue background. At the top center is a gray silhouette of a person's head and shoulders. Below this are two white input fields. The first field contains the text 'QLBH'. The second field contains four dots '....' followed by a vertical cursor line. Below the input fields is a solid blue button with the white text 'Đăng nhập'.

- Khi click nút “Đăng nhập” kiểm tra nếu thông tin Tên đăng nhập và mật khẩu đúng với thông người dùng trong dữ liệu `Du_lieu.js` thì xuất hộp thoại thông báo đăng nhập thành công. Ngược lại thông báo đăng nhập thất bại
- Hướng dẫn:
 - Sử dụng đối tượng state để lưu trữ thông tin Tên đăng nhập và mật khẩu mà người dùng nhập trong constructor
 - Xây dựng phương thức render



```

constructor(props){
  super(props);
  this.state={Ten_Dang_nhap:'',Mat_khau:''}
}

render(){
  return(
    <View style={style.container}>
      <View style={{flex:1, alignItems:"center",justifyContent:"flex-end"}}>
        <Image style={style.image} source={require('../images/user_login.png')}></Image>
      </View>
      <View style={{flex:1, justifyContent:"flex-start"}}>
        <TextInput onChangeText={({Ten_Dang_nhap})=>this.setState({Ten_Dang_nhap})}
          value={this.state.Ten_Dang_nhap} placeholder='Tên đăng nhập' style={style.input}/>
        <TextInput onChangeText={({Mat_khau})=>this.setState({Mat_khau})}
          value={this.state.Mat_khau} placeholder='Mật khẩu' style={style.input} secureTextEntry/>
        <TouchableOpacity onPress={this.XL_Nhan.bind(this)} activeOpacity={0.5}>
          <View style={style.button}>
            <Text style={style.text}>Đăng nhập</Text>
          </View>
        </TouchableOpacity>
      </View>
    </View>
  );
}

```

- Sử dụng đối tượng StyleSheet để định dạng



```
const style= StyleSheet.create({
  container:{
    flex:1,
    backgroundColor:'#0082c8',
    justifyContent:'center'
  },
  input:{
    height:44,
    paddingHorizontal:6,
    backgroundColor:'#ffffff',
    margin:5,
    borderRadius:5
  },
  button:{
    height:46,
    borderRadius:5,
    margin:5,
    backgroundColor:'#036fa9',
    justifyContent:'center',
    alignItems:'center'
  },
  text:{
    color:'#ffffff',
    fontSize:16,
  },
  image:{
    width: 140, height: 140,
    marginBottom:20,
    borderTopLeftRadius:10,
    borderTopRightRadius:10,
    borderBottomLeftRadius:10,
    borderBottomRightRadius:10
  }
})
```

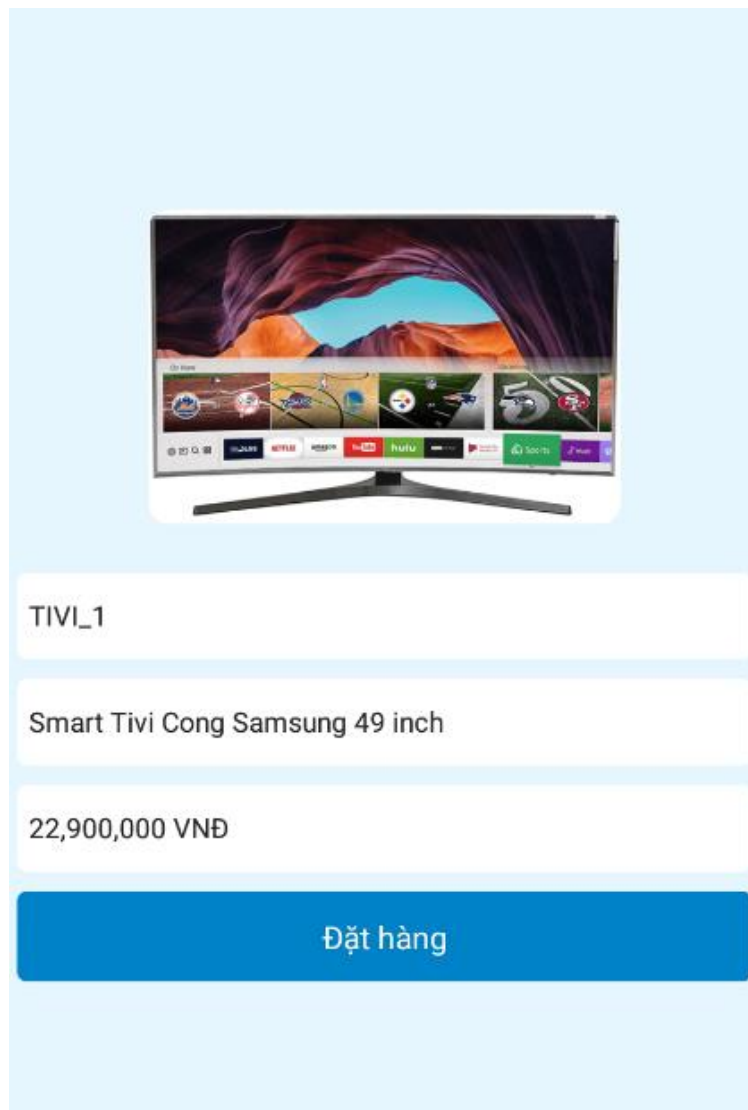
- Cài đặt sự kiện khi người dùng click nút “Đăng nhập”

```
XL_Nhan(){
  if(this.state.Ten_Dang_nhap==Du_lieu.Nguoi_dung.Ten_Dang_nhap && this.state.Mat_khau==Du_lieu.Nguoi_dung.Mat_khau)
    Alert.alert("Đăng nhập thành công");
  else
    Alert.alert("Đăng nhập thất bại");
}
```

3.3 Xây dựng màn hình Thông tin Tivi

- Yêu cầu:

- Xây dựng màn hình:



- Xuất thông của 1 Tivi trong danh sách Tivi từ dữ liệu `Du_lieu.js` khi biết mã số của Tivi.
- Khi click nút “Đặt hàng” xuất hộp thoại hiển thị thông tin Tivi đặt
- Hướng dẫn:
 - Sử dụng phương thức `find` để tìm đối tượng Tivi trong danh sách Tivi, nếu tìm thấy sử dụng dụng đối tượng `state` để lưu trữ thông tin Tivi.

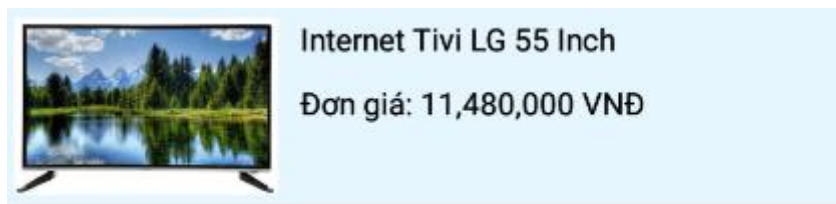
```
constructor(props){
  super(props);
  let Tivi= Du_lieu.Danh_sach_Tivi.find((item)=>{return item.Ma_so=="TIVI_1"});
  this.state={
    Ma_so:Tivi.Ma_so,
    Ten:Tivi.Ten,
    Hinh:Tivi.Hinh,
    Don_gia:Tivi.Don_gia,
  }
}
```

- Xây dựng phương thức render
- Sử dụng đối tượng StyleSheet để định dạng
- Cài đặt sự kiện khi người dùng click nút “Đặt hàng”

```
XL_Nhan(){
  //Minh họa xuất đơn giá
  if(this.state.Don_gia != ""){
    let Don_gia=parseFloat(this.state.Don_gia);
    Alert.alert("Đơn giá Tivi: " + Don_gia);
  }
}
```

3.4 Xây dựng màn hình Thông tin Tivi dạng 2

- Yêu cầu:
 - Xây dựng màn hình:



- Xuất thông của 1 Tivi trong danh sách Tivi từ dữ liệu Du_lieu.js khi biết mã số của Tivi.
- Khi click tên Tivi xuất hộp thoại hiển thị thông tin Tivi chọn
- Hướng dẫn:
 - Sử dụng phương thức find để tìm đối tượng Tivi trong danh sách Tivi, nếu tìm thấy sử dụng dụng đối tượng state để lưu trữ thông tin Tivi.
 - Xây dựng phương thức render



```
<View style={{flex:1, flexDirection:"column"}}>
  <View style={{flex:1, flexDirection:"row", backgroundColor:"#e5f6ff"}}>
    <Image style={{height:80,width:120, margin:5}} source={{uri:this.state.Hinh}}></Image>
    <View style={{flex:1, flexDirection:"column", height:30}}>
      <Text style={styles.itemText}>{this.state.Ten}</Text>
      <Text style={styles.itemText}>Đơn giá: {format_number(this.state.Don_gia.toString()) + ' VNĐ'}</Text>
    </View>
  </View>
  <View style={{height:1, backgroundColor:"#e6e5e5"}}></View>
</View>
```

- Sử dụng đối tượng StyleSheet để định dạng

```
const styles= StyleSheet.create({
  itemText:{
    color:"#000000",
    padding:5,
    fontSize:14
  }
})
```

- Cài đặt sự kiện khi người dùng click tên Tivi



Bài 4 Chức năng Tra cứu thông tin

4.1 Chuẩn bị dữ liệu

- Yêu cầu:
 - Bổ sung 2 danh sách: Gio_hang và Danh_sach_Nguoi_dung vào tập tin dữ liệu Du_lieu.js theo cấu trúc sau:

```
var Du_lieu = {
  Nguoi_dung:{
    // Dữ liệu được tạo trong bài 3
  },
  Danh_sach_Tivi: [
    // Dữ liệu được tạo trong bài 3
  ],
  Gio_hang: [
    {
      "Ten":"Smart Tivi Cong Samsung 49 inch",
      "Ma_so":"TIVI_1",
      "Hinh":"http://10.0.3.2:8081/images/TIVI_1.png",
      "Don_gia":22900000,
      "So_luong":1
    },
    {
      "Ten":"Internet Tivi Sony 4K 43 inch",
      "Ma_so":"TIVI_4",
      "Hinh":"http://10.0.3.2:8081/images/TIVI_4.png",
      "Don_gia":14500000,
      "So_luong":2
    },
    {
      "Ten":"Internet Tivi Samsung 32 inch",
      "Ma_so":"TIVI_8",
      "Hinh":"http://10.0.3.2:8081/images/TIVI_8.png",
```



```

        "Don_gia":6890000,
        "So_luong":1
    }
],
Danh_sach_Nguoi_dung:[
    {
        "Ho_ten": "Nguyễn thị bé Nhỏ",
        "Ma_so": "NV_1",
        "Ten_Dang_nhap": "NV_1",
        "Mat_khau": "NV_1",
        "Nhom_Nguoi_dung": {
            "Ten": "Nhân viên Nhập hàng",
            "Ma_so": "NHAN_VIEN_NHAP_HANG"
        }
    },
    {
        "Ho_ten": "Tô thị bé Bé",
        "Ma_so": "NV_2",
        "Ten_Dang_nhap": "NV_2",
        "Mat_khau": "NV_2",
        "Nhom_Nguoi_dung": {
            "Ten": "Nhân viên Bán hàng",
            "Ma_so": "NHAN_VIEN_BAN_HANG"
        }
    },
    {
        "Ho_ten": "Nguyễn văn bé Xíu",
        "Ma_so": "NV_3",
        "Ten_Dang_nhap": "NV_3",
        "Mat_khau": "NV_3",
        "Nhom_Nguoi_dung": {

```




```

        "Ten": "Nhân viên Bán hàng",
        "Ma_so": "NHAN_VIEN_BAN_HANG"
    }
},
{
    "Ho_ten": "Trần văn bé Lớn",
    "Ma_so": "NV_4",
    "Ten_Dang_nhap": "QL_1",
    "Mat_khau": "QL_1",
    "Nhom_Nguoi_dung": {
        "Ten": "Quản lý Nhập hàng",
        "Ma_so": "QUAN_LY_NHAP_HANG"
    }
},
{
    "Ho_ten": "Trần thị bé Cao",
    "Ma_so": "NV_5",
    "Ten_Dang_nhap": "QL_2",
    "Mat_khau": "QL_2",
    "Nhom_Nguoi_dung": {
        "Ten": "Quản lý Bán hàng",
        "Ma_so": "QUAN_LY_BAN_HANG"
    }
},
{
    "Ho_ten": "Lê văn sáng Rực",
    "Ma_so": "NV_6",
    "Ten_Dang_nhap": "QL",
    "Mat_khau": "QL",
    "Nhom_Nguoi_dung": {
        "Ten": "Quản lý Cửa hàng",

```



```

    "Ma_so": "QUAN_LY_CUA_HANG"
  }
}
]
}
module.exports = Du_lieu;

```

4.2 Xây dựng màn hình tìm Người dùng theo Họ tên

- Yêu cầu:
 - Xây dựng màn hình:

Nhập họ tên

Nguyễn thị bé Nhỏ

>

Tô thị bé Bé

>

Nguyễn văn bé Xíu

>

Trần văn bé Lớn


>

Trần thị bé Cao

>

Lê văn sáng Rực

>

- Khi nhập tên người dùng và click biểu tượng , thực hiện tìm và xuất danh sách người dùng có họ tên chứa nội dung cần tìm.
- Hướng dẫn:
 - Sử dụng đối tượng state để lưu trữ thông tin Danh sách người dùng và nội dung tìm trong constructor

```
constructor(props){
  super(props);
  this.state = {
    Gia_tri_tim: '',
    Danh_sach_Nguoi_dung: Du_lieu.Danh_sach_Nguoi_dung
  };
}
```

- Xây dựng phương thức render

```
<View>
  <View style={styles.inputContainer}>
    <TextInput
      style={styles.inputs}
      placeholder="Nhập họ tên"
      value={this.state.Gia_tri_tim}
      onChangeText={value => this.setState({Gia_tri_tim: value})}
      onSubmitEditing={this.Tim_Nguoi_dung.bind(this)}
      placeholderTextColor="#ffffff"
    />
    <TouchableOpacity onPress={this.Tim_Nguoi_dung.bind(this)} activeOpacity={0.5}>
      <Image style={styles.inputIcon} source={require('../images/search.png')}/>
    </TouchableOpacity>
  </View>
  <FlatList
    data={this.state.Danh_sach_Nguoi_dung}
    keyExtractor={(item) => item.Ma_so}
    renderItem={({item, index})=>{
      return(
        <FlatListItem item={item} index={index}></FlatListItem>
      );
    }}
  </FlatList>
</View>
```

- Xây dựng đối tượng thể hiện người dùng



```
class FlatListItem extends Component{
  render(){
    return(
      <View style={{flex:1, flexDirection:"column"}}>
        <View style={{flex:1, flexDirection:"row"}}>
          <Image
            style={{height:30,width:30, margin:10, borderRadius:20}}
            source={require('../images/avatar-40.png')}>
          </Image>
          <View style={{flex:1,justifyContent:'center', height:50}}>
            <Text style={styles.itemText}>{this.props.item.Ho_ten}</Text>
          </View>
          <View style={{justifyContent:'center', height:50}}>
            <Image
              style={{height:20,width:20, margin:5, tintColor:'#A7A7A7'}}
              source={require('../images/greater-than-50.png')}>
            </Image>
          </View>
        </View>
        <View style={{height:1, backgroundColor:"#e6e5e5"}}></View>
      </View>
    );
  }
}
```

- Sử dụng đối tượng StyleSheet để định dạng

```
const styles= StyleSheet.create({
  itemText:{
    color:"#000000",
    padding:5,
    fontSize:14
  },
  inputContainer: {
    borderColor: '#b9b7b7',
    backgroundColor: '#4387fd',
    borderBottomWidth: 1,
    height:50,
    flexDirection: 'row',
    alignItems:'center', |
  },
  inputs:{
    height:40,
    marginRight:5,
    borderBottomColor: '#FFFFFF',
    flex:1,
    color:'#ffffff',
    fontSize:14
  },
  inputIcon:{
    width:26,
    height:26,
    tintColor:'#ffffff',
    marginRight:5,
    justifyContent: 'center'
  }
})
```

- Cài đặt sự kiện khi người dùng click biểu tượng









```
Tìm_Nguoi_dung(){
  let Gia_tri_tim = this.state.Gia_tri_tim;
  let Danh_sach_Nguoi_dung=Du_lieu.Danh_sach_Nguoi_dung.filter(n=>{
    return n.Ho_ten.toLowerCase().includes(Gia_tri_tim.toLowerCase())
  })
  this.setState({Danh_sach_Nguoi_dung:Danh_sach_Nguoi_dung})
}
```

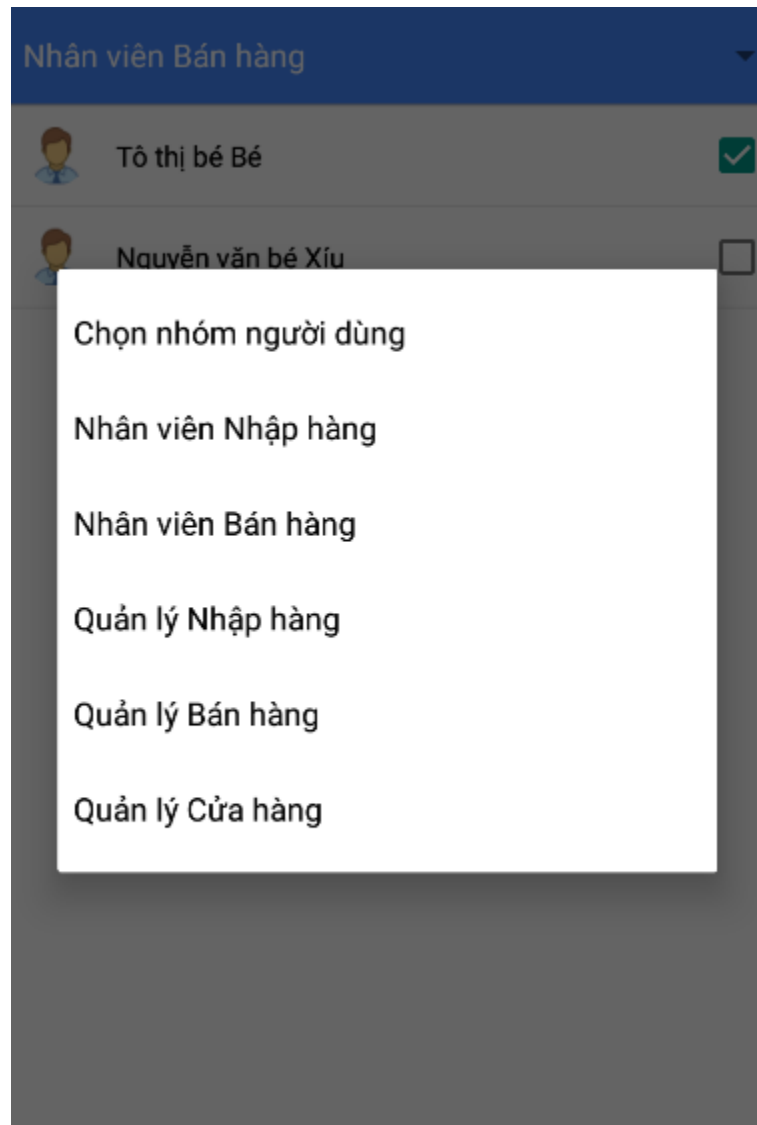
4.3 Xây dựng màn hình tra cứu Người dùng theo nhóm

- Yêu cầu:
 - Xây dựng màn hình:

Chosen group of people

	Nguyễn thị bé Nhỏ	<input checked="" type="checkbox"/>
	Tô thị bé Bé	<input checked="" type="checkbox"/>
	Nguyễn văn bé Xíu	<input type="checkbox"/>
	Trần văn bé Lớn	<input type="checkbox"/>
	Trần thị bé Cao	<input type="checkbox"/>
	Lê văn sáng Rực	<input type="checkbox"/>

- Khi chọn nhóm người dùng từ danh sách chọn, thực hiện lọc và xuất danh sách người dùng thuộc nhóm chọn



- Hướng dẫn:
 - Sử dụng đối tượng state để lưu trữ thông tin Danh sách người dùng và nội dung tìm trong constructor
 - Xây dựng phương thức render: sử dụng Picker component để tạo thể hiện danh sách nhóm người dùng



```
<View style={styles.inputContainer}>
  <Picker
    style={{width:"100%", color:'#ffffff'}}
    selectedValue={this.state.Gia_tri_tim}
    onChange={this.Tim_Nguoi_dung(item,index)}>
    <Picker.Item label="Chọn nhóm người dùng" value="" />
    <Picker.Item label="Nhân viên Nhập hàng" value="NHAN_VIEN_NHAP_HANG" />
    <Picker.Item label="Nhân viên Bán hàng" value="NHAN_VIEN_BAN_HANG" />
    <Picker.Item label="Quản lý Nhập hàng" value="QUAN_LY_NHAP_HANG" />
    <Picker.Item label="Quản lý Bán hàng" value="QUAN_LY_BAN_HANG" />
    <Picker.Item label="Quản lý Cửa hàng" value="QUAN_LY_CUA_HANG" />
  </Picker>
</View>
```

- Xây dựng đối tượng thể hiện người dùng: sử dụng CheckBox component để tạo thể hiện chọn người dùng

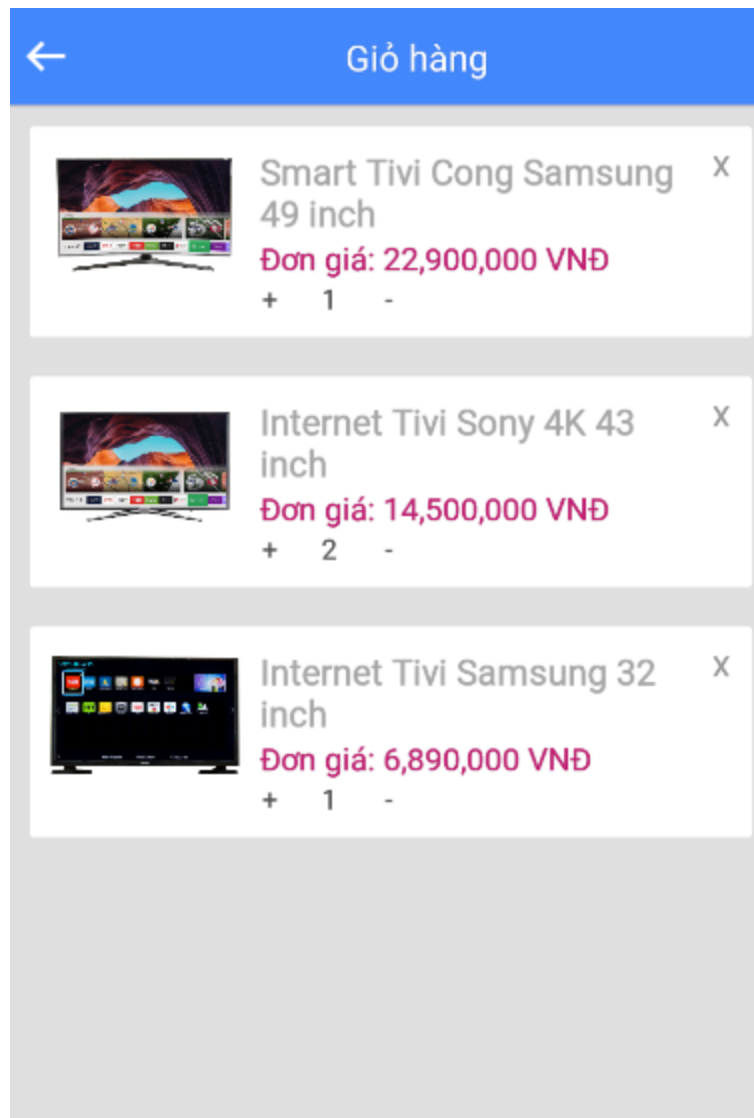
```
<View style={{justifyContent:'center', height:50}}>
  <CheckBox value={this.state.Chon} onChange={()=>this.XL_Chon()} />
</View>
```

- Sử dụng đối tượng StyleSheet để định dạng
- Cài đặt sự kiện khi người dùng click chọn nhóm người dùng từ danh sách nhóm người dùng

```
Tim_Nguoi_dung(item,index){
  this.setState({Gia_tri_tim:item});
  let Danh_sach_Nguoi_dung=Du_lieu.Danh_sach_Nguoi_dung.filter(n=>{
    return n.Nhom_Nguoi_dung.Ma_so.includes(item)
  })
  this.setState({Danh_sach_Nguoi_dung:Danh_sach_Nguoi_dung})
}
```

4.4 Xây dựng màn hình Giỏ hàng

- Yêu cầu:
 - Xây dựng màn hình:



- Xuất thông tin Giỏ hàng.
- Chọn + tăng số lượng thêm 1
- Chọn – giảm số lượng xuống 1
- Hướng dẫn:
 - Sử dụng đối tượng state để lưu trữ thông tin Danh sách Tivi từ Giỏ hàng
 - Xây dựng phương thức render



```
<View style={styles.container}>
  <View style={styles.inputContainer}>
    <TouchableOpacity>
      <Image style={styles.inputIcon} source={require('../images/back.png')}/>
    </TouchableOpacity>
    <Text style={styles.text}>Giỏ hàng</Text>
  </View>
  <FlatList
    contentContainerStyle={styles.containerList}
    data={this.state.Danh_sach_Tivi}
    keyExtractor={({item}) => item.Ma_so}
    renderItem={({item, index})=>{
      return(
        <FlatListItem item={item} index={index}></FlatListItem>
      );
    }}
  </FlatList>
</View>
```

- Sử dụng đối tượng state để lưu trữ thông tin Tivi:

```
componentDidMount(){
  this.setState({
    Ma_so:this.props.item.Ma_so,
    Ten:this.props.item.Ten,
    Hinh:this.props.item.Hinh,
    Don_gia:this.props.item.Don_gia,
    So_luong:this.props.item.So_luong
  })
}
```

- Xây dựng đối tượng thể hiện Tivi trong giỏ hàng



```

<View style={styles.productItem}>
  <Image style={styles.productImage} source={{uri:this.state.Hinh}}></Image>
  <View style={styles.containerInfo}>
    <View style={{ justifyContent: 'space-between', flexDirection: 'row' }}>
      <Text style={styles.txtName}>{this.state.Ten}</Text>
      <TouchableOpacity onPress={() => this.Xoa_San_pham()}>
        <Text style={{ fontFamily: 'Avenir', color: '#969696', width:10 }}>X</Text>
      </TouchableOpacity>
    </View>
    <View>
      <Text style={styles.txtPrice}>
        Đơn giá: {format_number(this.state.Don_gia.toString()) + ' VNĐ'}
      </Text>
    </View>
    <View style={styles.productController}>
      <View style={styles.number}>
        <TouchableOpacity onPress={() => this.Them_So_luong()}>
          <Text style={{width:30, textAlign:"center"}}>+</Text>
        </TouchableOpacity>
        <Text style={{width:30, textAlign:"center"}}>{this.state.So_luong}</Text>
        <TouchableOpacity onPress={() => this.Giam_So_luong()}>
          <Text style={{width:30, textAlign:"center"}}>-</Text>
        </TouchableOpacity>
      </View>
    </View>
  </View>
</View>

```

- Sử dụng đối tượng StyleSheet để định dạng



```
const { width } = Dimensions.get('window');
const imageWidth = width / 4;
const imageHeight = (imageWidth * 34) / 55;
const styles = StyleSheet.create({
  container: {
    flex: 1, backgroundColor: '#DFDFDF'
  },
  inputContainer: {
    borderColor: '#b9b7b7', backgroundColor: '#4387fd', borderBottomWidth: 1,
    height: 50, flexDirection: 'row', alignItems: 'center',
  },
  text: {
    marginRight: 5, borderBottomColor: 'FFFFFF', flex: 1,
    color: 'ffffff', fontSize: 18, textAlign: 'center', justifyContent: 'center'
  },
  inputIcon: {
    width: 26, height: 26, tintColor: 'ffffff',
    marginLeft: 5, justifyContent: 'center'
  },
  containerList: {
    width, backgroundColor: '#DFDFDF'
  },
  productItem: {
    flexDirection: 'row', margin: 10, padding: 10,
    backgroundColor: 'FFFFFF', borderRadius: 2, shadowColor: '#3B5458',
    shadowOffset: { width: 0, height: 3 }, shadowOpacity: 0.2
  },
  productImage: {
    width: imageWidth, height: imageHeight,
    marginTop: 5, resizeMode: 'center'
  },
  containerInfo: {
    flex: 1, justifyContent: 'space-between'
  },
  productController: {
    flexDirection: 'row'
  },
  number: {
    flex: 1, flexDirection: 'row'
  },
  txtName: {
    flex: 1, paddingLeft: 10, paddingRight: 10, color: '#A7A7A7',
    fontSize: 18, fontWeight: '400', fontFamily: 'Avenir'
  },
  txtPrice: {
    paddingLeft: 10, paddingRight: 10, color: '#C21C70',
    fontSize: 16, fontWeight: '400', fontFamily: 'Avenir'
  }
});
```













- Cài đặt sự kiện khi người dùng click nút +, -

```
Them_So_luong(){
    if(this.state.So_luong != ""){
        this.setState({So_luong: this.state.So_luong + 1});
    }
}
Giam_So_luong(){
    if(this.state.So_luong != ""){
        this.setState({So_luong: this.state.So_luong - 1});
    }
}
```

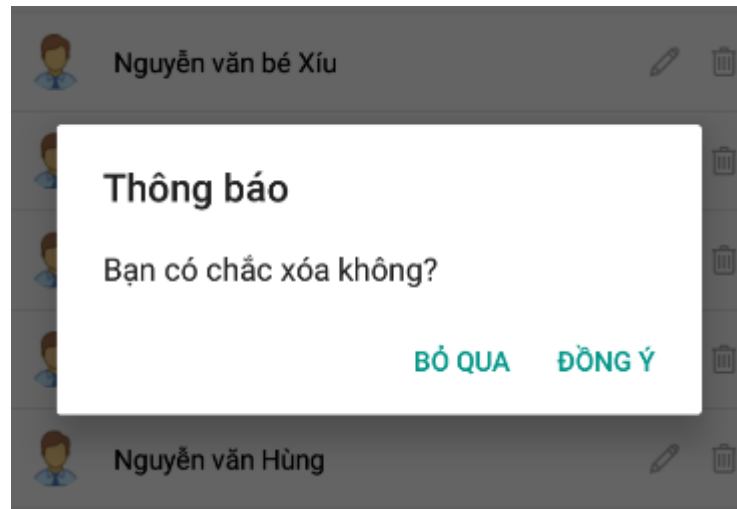
Bài 5 Chức năng Cập nhật thông tin

5.1 Xây dựng màn hình cập nhật thông tin người dùng

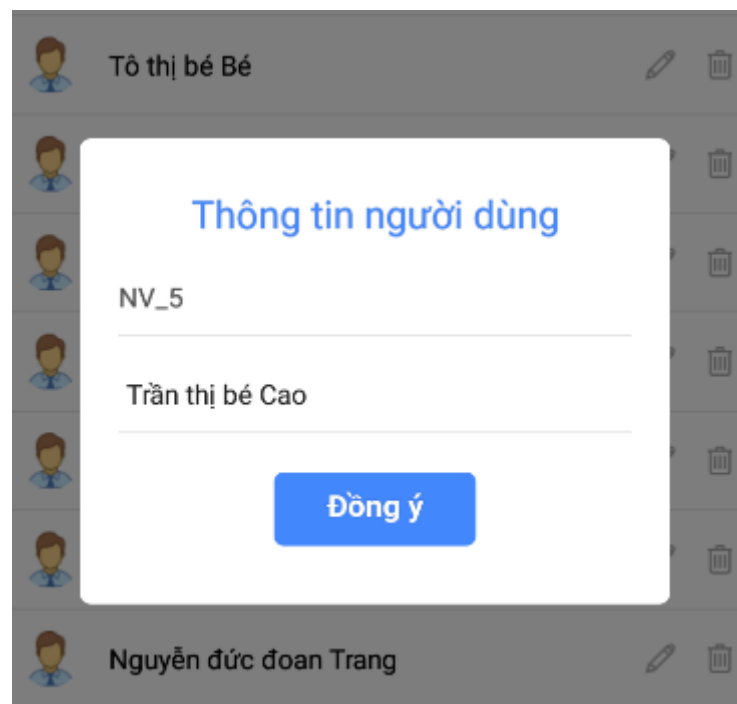
- Yêu cầu:
 - Xây dựng màn hình:


Danh sách người dùng		+
	Nguyễn thị bé Nhỏ	...
	Tô thị bé Bé	...
	Nguyễn văn bé Xíu	<div>Cập nhật</div> <div>Xóa</div>
	Trần văn bé Lớn	...
	Trần thị bé Cao	...
	Lê văn sáng Rực	...
	Nguyễn văn Hùng	...
	Nguyễn đức đoan Trang	...
	Nguyễn thanh quỳnh Châu	...
	Nguyễn trọng Tuyển	...

- Chức năng Xóa: xuất hộp thoại xác nhận, nếu đồng ý thì xóa người dùng chọn trong danh sách người dùng



- Chức năng Cập nhật: hiển thị thông tin người dùng chọn dạng Modal, cho phép chỉnh sửa họ tên.



- Chức năng Thêm mới  : hiển thị hộp thoại, cho phép nhập thông tin người dùng.



- Hướng dẫn:
 - Sử dụng react-native-popup-menu component tạo context menu cho đối tượng người dùng

```
<Menu style={{marginRight:5}} onSelect={value => {this.XL_Chon(value)}}>
  <MenuTrigger>
    <Image
      style={{ width:24, tint_color:'#60616c', }}
      source={require('../images/more-24.png')}/>
    </MenuTrigger>
    <MenuOptions optionsContainerStyle={{width:120, fontSize:16}}>
      <MenuOption value="0" text="Cập nhật" />
      <MenuOption value="1" text="Xóa" />
    </MenuOptions>
  </Menu>
```

- Xây dựng chức năng Cập nhật người dùng



```
<Modal ref={'Th_Hop_thoai'} style={{
  borderRadius: Platform.OS==='ios'?20:6, shadowRadius: 10,
  width:width - 80, height:240
}}
position='center'
backdrop={true}
>

  <Text style={{
    fontSize:20, textAlign:'center',
    marginTop:25,marginBottom:20, color:'#4387fd'}}>
    Thông tin người dùng
  </Text>
  <Text style={{
    height:30, borderBottomColor:'#e6e5e5',
    fontSize:14, borderBottomWidth:1,
    marginBottom:10, marginLeft:20, marginRight:20}}>
    {this.state.Ma_so}
  </Text>
  <TextInput style={{
    height:40, borderBottomColor:'#e6e5e5',
    fontSize:14, borderBottomWidth:1,
    marginBottom:10, marginLeft:20, marginRight:20}}
    placeholder='Họ tên'
    value={this.state.Ho_ten}
    onChangeText={(value) => {this.setState({Ho_ten:value})}}
  >
  </TextInput>
  <Button style={{fontSize:16, color:'#ffffff'}}
    containerStyle={{
      padding:6, marginLeft:100,
      marginRight:100, height:38,
      borderRadius:5, backgroundColor:'#4387fd', marginTop:10}}
    onPress={this.Xu_ly_Cap_nhat.bind(this)}
  >
    Đồng ý
  </Button>
</Modal>
```

- Tạo thể hiện chức năng cập nhật người dùng

```
<Cap_nhat_Nguoi_dung ref={'Th_Cap_nhat'} parentFlatList={this}></Cap_nhat_Nguoi_dung>
```

- Xây dựng các phương thức xử lý khi chọn chức năng (xóa người dùng/ cập nhật người dùng) trong chức năng người dùng:



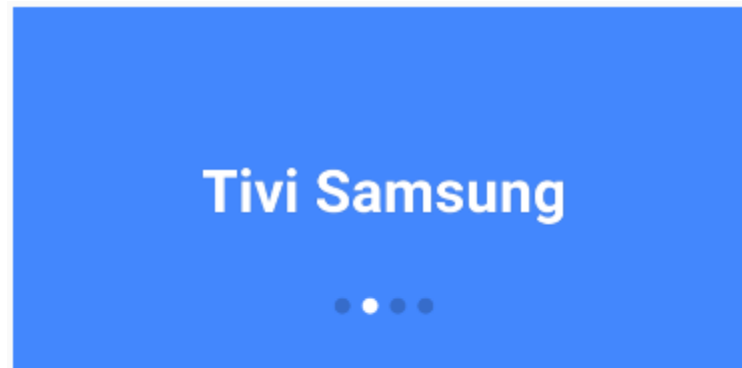
```
Cap_nhat_Nguoi_dung(){
    this.props.parentFlatList.refs.Th_Cap_nhat.Mo_Hop_thoai(this.props.item,this);
}
XL_Chon(value){
    if(value=="1"){
        this.Xoa_Nguoi_dung();
    }
    else{
        this.Cap_nhat_Nguoi_dung();
    }
}
}
```

- Xây dựng các phương thức xử lý hiển thị và cập nhật thông tin người dùng trong chức năng cập nhật người dùng:

```
Mo_Hop_thoai(Nguoi_dung,Th_Nguoi_dung){
    this.setState({
        Ma_so:Nguoi_dung.Ma_so,
        Ho_ten:Nguoi_dung.Ho_ten,
        Th_Nguoi_dung:Th_Nguoi_dung
    });
    this.refs.Th_Hop_thoai.open();
}
Xu_ly_Cap_nhat(){
    if(this.state.Ma_so==' ' || this.state.Ho_ten==' '){
        alert('Bạn phải nhập đủ thông tin');
        return;
    }
    let Vi_tri=Danh_sach_Nguoi_dung.findIndex(n=>n.Ma_so==this.state.Ma_so);
    if(Vi_tri<0){
        return;
    }
    Danh_sach_Nguoi_dung[Vi_tri].Ho_ten=this.state.Ho_ten;
    this.state.Th_Nguoi_dung.refresh_Nguoi_dung();
    this.refs.Th_Hop_thoai.close();
}
```

5.2 Xây dựng màn hình danh sách nhóm Tivi

- Yêu cầu:
 - Xây dựng màn hình:



- Danh sách nhóm Tivi thể hiện dạng slider
- Hướng dẫn:
 - Sử dụng react-native-swiper component

```
const swiper= (
  <Swiper showsPagination activeDotColor='#ffffff'>
    {
      Du_lieu.Danh_sach_Nhom_Tivi.map((item,index) => {
        return (
          <View style={styles.slide} key={item.Ma_so}>
            <TouchableOpacity onPress={() => alert(item.Ma_so)} >
              <Text style={styles.text}>{item.Ten}</Text>
            </TouchableOpacity>
          </View>
        );
      })
    }
  </Swiper>
);
return (
  <View style={styles.wrapper}>
    { Du_lieu.Danh_sach_Nhom_Tivi.length>0? swiper : null }
  </View>
);
```


- Định dạng thể hiện danh sách nhóm Tivi



```
const { width } = Dimensions.get('window');
const imageWidth = width;
const imageHeight = imageWidth/2;
var styles = {
  wrapper: {
    height:imageHeight,
    padding:5
  },
  slide: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#4387fd',
    padding: 10,
    shadowColor: '#01458e',
    shadowOffset: {
      width: 0,
      height: 3
    },
    shadowRadius: 5,
    shadowOpacity: 1.0
  },
  text: {
    color: '#fff',
    fontSize: 30,
    fontWeight: 'bold'
  }
}
```

5.3 Xây dựng màn hình thông tin Tivi

- Yêu cầu:
 - Xây dựng màn hình:



TIVI_1

Smart Tivi Cong Samsung 49 inch

22,900,000 VNĐ

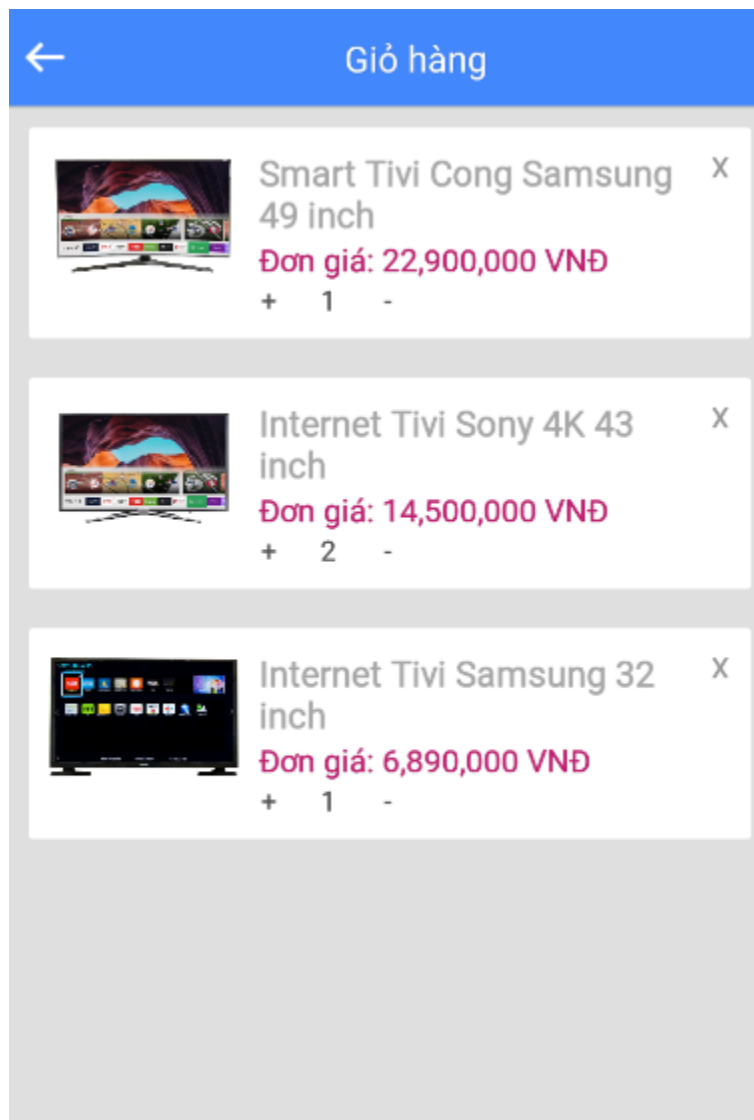
Đặt hàng

- Khi chọn chức năng Đặt hàng: Kiểm tra Tivi chọn đã được đặt trước đó trong Giỏ hàng chưa?
 - Nếu đã đặt thì cập nhật lại số lượng tăng thêm 1
 - Nếu chưa thì thêm Tivi chọn vào trong Giỏ hàng
- Hướng dẫn:
 - Thực hiện tìm Tivi chọn trong Giỏ hàng (mảng chứa danh sách Tivi đặt)
 - Nếu tìm thấy thì cập nhật lại số lượng
 - Nếu không tìm thấy thì thêm Tivi mới vào trong Giỏ hàng

5.4 Xây dựng màn hình Giỏ hàng

- Yêu cầu:

- Xây dựng màn hình:














- Xuất thông tin danh sách Tivi trong Giỏ hàng.
 - Chọn + cập nhật số lượng của Tivi trong giỏ hàng tăng thêm 1
 - Chọn – cập nhật số lượng của Tivi trong giỏ hàng giảm xuống 1
 - Chọn x xóa Tivi chọn trong giỏ hàng
- Hướng dẫn:

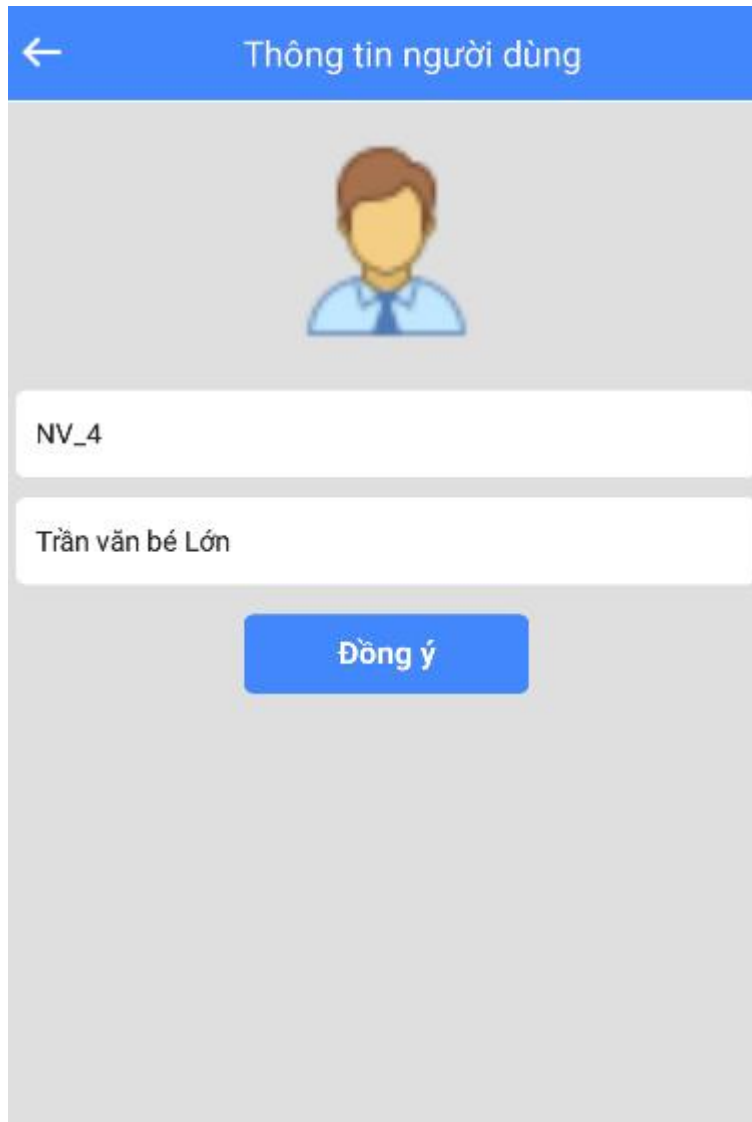
Bài 6 Màn hình Chính và Thực đơn


6.1 Xây dựng màn hình danh sách người dùng

- Yêu cầu:
 - Xây dựng màn hình:

Danh sách người dùng		
	Nguyễn thị bé Nhỏ	>
	Tô thị bé Bé	>
	Nguyễn văn bé Xíu	>
	Trần văn bé Lớn	>
	Trần thị bé Cao	>
	Lê văn sáng Rực	>
	Nguyễn văn Hùng	>
	Nguyễn đức đoan Trang	>
	Nguyễn thanh quỳnh Châu	>
	Nguyễn trọng Tuyển	>

- Chức năng  : chuyển sang màn hình xem thông tin người dùng tương ứng



- Chức năng : trở về màn hình xem danh sách người dùng
- Hướng dẫn:
 - Khởi tạo đối tượng StackNavigator



```
const Man_hinh = StackNavigator({
  "Man_hinh_Danh_sach_Nguoi_dung": {
    screen: Danh_sach_Nguoi_dung
  },
  "Man_hinh_Thong_tin_Nguoi_dung": {
    screen: Thong_tin_Nguoi_dung
  }
},
{
  headerMode: 'none',
  navigationOptions: {
    headerVisible: false,
  }
});
```

- Xây dựng phương thức tạo thể hiện danh sách người dùng

```
<View>
  <View style={styles.inputContainer}>
    <Text style={styles.text}>Danh sách người dùng</Text>
  </View>
  <FlatList
    data={Du_lieu.Danh_sach_Nguoi_dung}
    keyExtractor={({item}) => item.Ma_so}
    renderItem={({item, index})=>{
      return(
        <FlatListItem item={item} index={index} navigation={this.props.navigation}>
        </FlatListItem>
      );
    }}>
  </FlatList>
</View>
```

- Xây dựng phương tạo thể hiện người dùng trong danh sách người dùng



```
<View style={{flex:1, flexDirection:"column"}}>
  <View style={{flex:1, flexDirection:"row"}}>
    <Image
      style={{height:30,width:30, margin:10, borderRadius:20, borderWidth:0.5}}
      source={require('../images/avatar-40.png')}>
    </Image>
    <View style={{flex:1,justifyContent:'center', height:50}}>
      <Text style={styles.itemText}>{this.props.item.Ho_ten}</Text>
    </View>
    <View style={{justifyContent:'center', height:50}}>
      <TouchableOpacity onPress={this.XL_Chon.bind(this)} activeOpacity={0.5}>
        <Image
          style={{height:20,width:20, margin:5, tintColor:'#A7A7A7'}}
          source={require('../images/greater-than-50.png')}>
        </Image>
      </TouchableOpacity>
    </View>
  </View>
  <View style={{height:1, backgroundColor:"#e6e5e5"}}></View>
</View>
```

- Xử lý chọn người dùng

```
XL_Chon(){
  this.props.navigation.navigate("Man_hinh_Thong_tin_Nguoi_dung",{ "Nguoi_dung":this.props.item})
}
```

- Xử lý nhận trị từ tham số người dùng

```
constructor(props){
  super(props);
  let Tivi= this.props.navigation.state.params.Tivi;
  this.state={
    Ma_so:Tivi.Ma_so,
    Ten:Tivi.Ten,
    Hinh:Tivi.Hinh,
    Don_gia:Tivi.Don_gia
  }
}
```

- Xây dựng phương thức tạo thể hiện người dùng



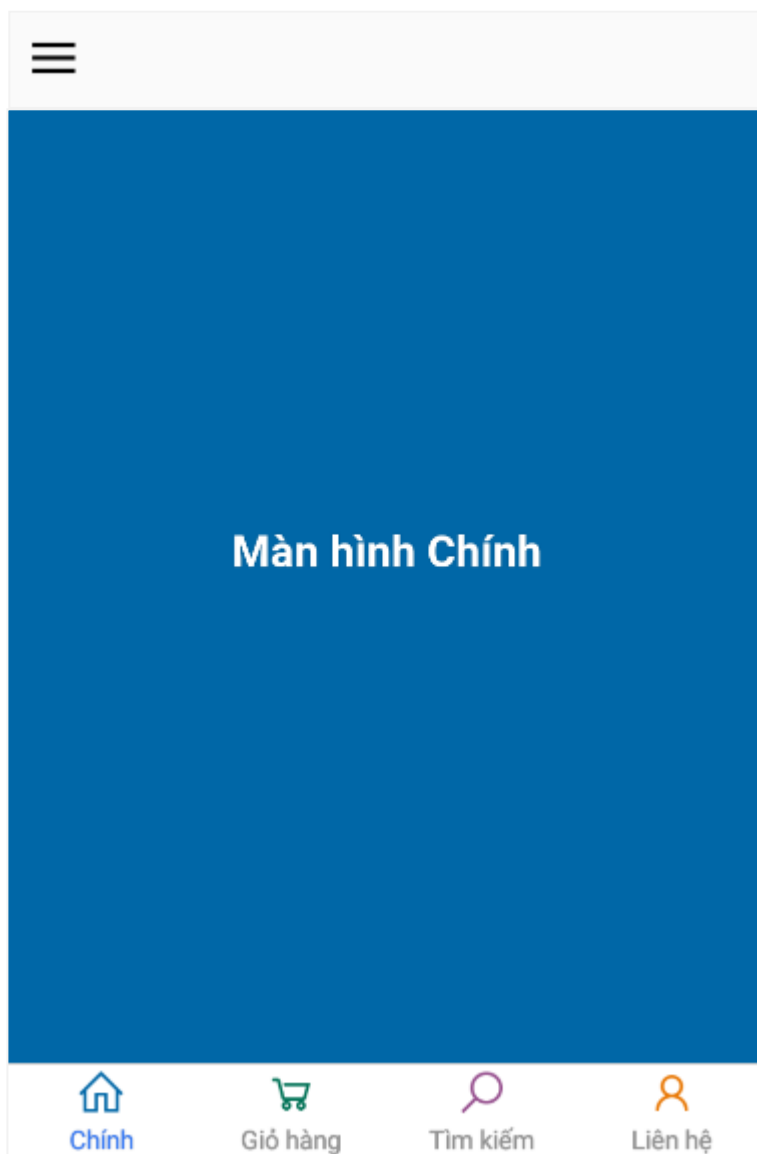
```
<View style={style.container}>
  <View style={style.inputContainer}>
    <TouchableOpacity onPress={this.XL_Tro_ve.bind(this)}>
      <Image style={style.inputIcon} source={require('../images/back.png')}/>
    </TouchableOpacity>
    <Text style={style.text}>Thông tin người dùng</Text>
  </View>
  <View style={{alignItems:"center",marginTop:20}}>
    <Image style={style.image} source={require('../images/avatar-40.png')}></Image>
  </View>
  <View style={{flex:1, justifyContent:"flex-start"}}>
    <TextInput
      onChangeText={({Ma_so})=>this.setState({Ma_so})} value={this.state.Ma_so}
      placeholder='Mã số' style={style.input}/>
    <TextInput
      onChangeText={({Ho_ten})=>this.setState({Ho_ten})} value={this.state.Ho_ten}
      placeholder='Họ tên' style={style.input}/>
    <Button style={{fontSize:16, color:'#ffffff'}}
      containerStyle={{
        padding:8, marginLeft:120,
        marginRight:120, height:40,
        borderRadius:5, backgroundColor:'#4387fd', marginTop:10}}
      onPress={this.XL_Nhan.bind(this)}
    >
      Đồng ý
    </Button>
  </View>
</View>
```

- Xử lý trở về

```
XL_Tro_ve(){
  this.props.navigation.goBack();
}
```


6.2 Xây dựng màn hình chính với hệ thống thực đơn

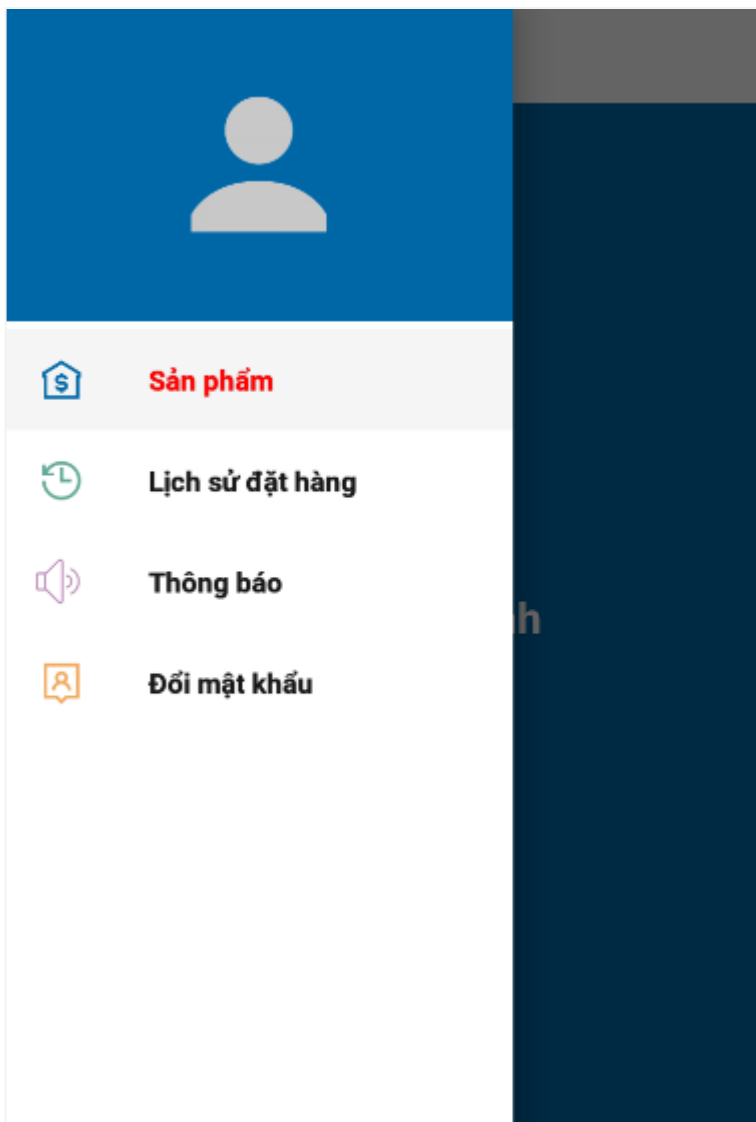
- Yêu cầu:
 - Xây dựng màn hình:



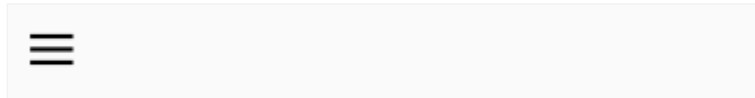
- Khi chọn danh sách chức năng trong Tab, chuyển màn hình tương ứng



- Khi chọn  hiển thị danh sách chức năng



- Khi chọn danh sách chức năng trong Slider, chuyển màn hình tương ứng
- Hướng dẫn:
 - Xây dựng chức năng thể hiện Thực đơn, dùng chung cho các màn hình trong thực đơn Tab



```
export default class MH_Thuc_don extends Component{
  render(){
    return(
      <View style={{
        flexDirection:"row", height:50,
        justifyContent:'flex-start', alignItems:'center'}}>
        <TouchableHighlight
          style={{marginLeft:10}}
          onPress={()=>{this.props.navigation.navigate("DrawerOpen")}}>
          <Image source={require("../icons/menu.png")} style={{width:26, height:26}}/>
        </TouchableHighlight>
      </View>
    );
  }
}
```

- Kết hợp TabNavigator và DrawerNavigator



Bài 7 Dịch vụ Dữ liệu

7.1 Chuẩn bị dữ liệu

- Yêu cầu:
 - Tạo tập tin danh sách Tivi và người dùng lưu trong thư mục dữ liệu:
 - Du_lieu\Nguoi_dung.json

```
{
  "Ho_ten": "Tô thị bé Bé",
  "Ma_so": "NV_2",
  "Ten_Dang_nhap": "NVBH",
  "Mat_khau": "NVBH",
  "Nhom_Nguoi_dung": {
    "Ten": "Nhân viên Bán hàng",
    "Ma_so": "NHAN_VIEN_BAN_HANG"
  }
}
```

- Du_lieu\Danh_sach_Tivi.json

```
[
  {
    "Ten": "Smart Tivi Cong Samsung 49 inch",
    "Ma_so": "TIVI_1",
    "Hinh": "http://media/images/TIVI_1.png",
    "Don_gia": 22900000,
    "Danh_sach_Ban_hang": [
      {
        "Ma_so": "PB_1",
        "Ngay": "2018-10-10",
        "So_luong": 2,
        "Don_gia": 22900000
      },
      {
        "Ma_so": "PB_2",
```




```

        "Ngày":"2018-10-10",
        "So_luong":1,
        "Don_gia":22900000
    }
]
},
{
    "Ten":"Internet Tivi Sony 4K 43 inch",
    "Ma_so":"TIVI_4",
    "Hinh":"http://media/images/TIVI_4.png",
    "Don_gia":14500000,
    "Danh_sach_Ban_hang":[]
},
{
    "Ten":"Internet Tivi Samsung 32 inch",
    "Ma_so":"TIVI_8",
    "Hinh":"http://media/images/TIVI_8.png",
    "Don_gia":6890000,
    "Danh_sach_Ban_hang":[]
},
{
    "Ten":"Internet Tivi Sony 40 inch",
    "Ma_so":"TIVI_10",
    "Hinh":"http://media/images/TIVI_10.png",
    "Don_gia":9990000,
    "Danh_sach_Ban_hang":[]
},
{
    "Ten":"Smart Tivi Samsung 40 inch",
    "Ma_so":"TIVI_11",
    "Hinh":"http://media/images/TIVI_11.png",

```



```

    "Don_gia":8640000,
    "Danh_sach_Ban_hang":[]
  },
  {
    "Ten":"Smart Tivi LED 4K SONY 49 inch",
    "Ma_so":"TIVI_18",
    "Hinh":"http://media/images/TIVI_18.png",
    "Don_gia":10420000,
    "Danh_sach_Ban_hang":[]
  },
  {
    "Ten": "Smart Tivi LED 4K SONY 75 Inch",
    "Ma_so": "TIVI_20",
    "Hinh":"http://media/images/TIVI_20.png",
    "Don_gia": 7340000,
    "Danh_sach_Ban_hang":[]
  },
  {
    "Ten": "Smart Tivi SONY Ultra HD 75 Inch",
    "Ma_so": "TIVI_22",
    "Hinh":"http://media/images/TIVI_22.png",
    "Don_gia": 11480000,
    "Danh_sach_Ban_hang":[]
  },
  {
    "Ten":"Internet Tivi LG 55 Inch",
    "Ma_so":"TIVI_32",
    "Hinh":"http://media/images/TIVI_32.png",
    "Don_gia":11480000,
    "Danh_sach_Ban_hang":[]
  },
  },

```



```
{
  "Ten": "TV Philips",
  "Ma_so": "TIVI_34",
  "Hinh": "http://media/images/TIVI_34.png",
  "Don_gia": 10420000,
  "Danh_sach_Ban_hang": []
}
```

7.2 Xây dựng phương thức đăng nhập

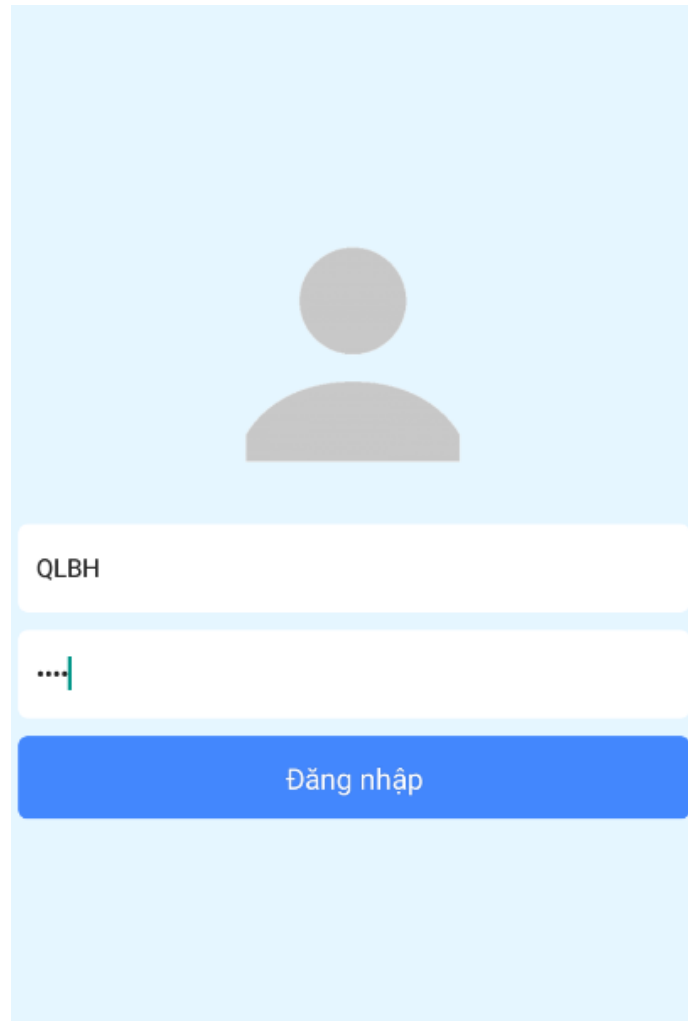
- Yêu cầu:
 - Sử dụng phương thức POST
 - Tham số vào: Tên đăng nhập (string), Mật khẩu (string)
 - Kiểm tra Tên đăng nhập và Mật khẩu với thông tin trong dữ liệu người dùng
 - Nếu đúng: trả về đối tượng người dùng
 - Nếu sai: trả về false
- Hướng dẫn:
 - Tạo đối tượng người dùng đọc từ tập tin Nguoi_dung.json
 - Kiểm tra thông tin nhận từ tham số vào và thông tin từ đối tượng người dùng

7.3 Xây dựng phương thức cập nhật phiếu bán

- Yêu cầu:
 - Sử dụng phương thức POST
 - Tham số vào: đối tượng Tivi chọn bán
 - Kiểm tra thông tin hợp lệ thực hiện:
 - Thêm phiếu bán vào danh sách bán hàng của Tivi tương ứng
- Hướng dẫn:
 - Tìm Tivi bán từ danh sách Tivi
 - Phát sinh tự động Mã số phiếu bán
 - Thêm phiếu bán vào danh sách bán hàng của Tivi

7.4 Xây dựng màn hình đăng nhập







- Yêu cầu:
 - Đăng nhập từ dịch vụ đăng nhập (bài 7.2)
 - Đăng nhập với vai trò nhân viên bán hàng:
 - Nếu thành công thì chuyển sang màn hình danh sách Tivi (bài 7.5), cho phép cập nhật phiếu bán
 - Nếu thất bại thì xuất hộp thoại thông báo tương ứng



A mockup of a user login form on a light blue background. At the top center is a gray silhouette of a person's head and shoulders. Below this are two white input fields. The first field contains the text 'QLBH'. The second field contains four dots '....' followed by a vertical cursor line. Below the input fields is a solid blue button with the white text 'Đăng nhập'.

7.5 Xây dựng màn hình danh sách Tivi


- Yêu cầu:
 - Hiển thị danh sách Tivi được đọc từ dịch vụ danh sách Tivi
 - Khi chọn Tivi cần bán, chuyển sang màn hình thông tin Tivi (bài 7.6) với thông tin Tivi chọn

Danh sách Tivi	
	Smart Tivi Cong Samsung 49 inch Đơn giá: 22,900,000 VND
	Internet Tivi Sony 4K 43 inch Đơn giá: 14,500,000 VND
	Internet Tivi Samsung 32 inch Đơn giá: 6,890,000 VND
	Internet Tivi Sony 40 inch Đơn giá: 9,990,000 VND
	Smart Tivi Samsung 40 inch Đơn giá: 8,640,000 VND
	Smart Tivi LED 4K SONY 49 inch Đơn giá: 10,420,000 VND

7.6 Xây dựng màn hình thông tin Tivi

- Yêu cầu:
 - Hiển thị thông tin Tivi tương ứng khi chọn từ danh sách Tivi
 - Cho phép sửa số lượng bán
 - Khi chọn Bán hàng thì cập nhật phiếu bán từ dịch vụ (bài 7.3)

[←](#) Internet Tivi Sony 4K 43 inch





Bài 8 Đồ án môn học