**REACT** **NAVIGATION**

**Link** **tham** **khảo:**

**Link** **1:** **<https://reactnative.dev/docs/getting-started>**

**Link** **2:** **<https://reactnative.dev/docs/navigation>**

**Content**

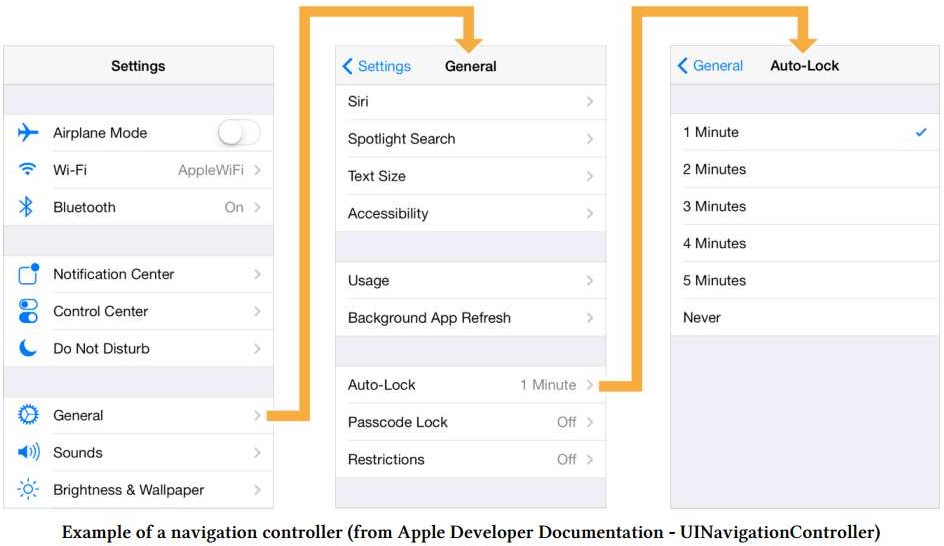
* **Navigation** **in** **React** **Native**
* **Contact** **List**
* **Starting** **the** **project**
* **Contacts**
* **Profile**
* **React** **Navigation**
* **Stack** **navigation**
* **Tab** **navigation**
* **Drawer** **navigation**
* **Sharing** **state** **between** **screens**

**Navigation** **in** **React** **Native**

Là việc điều hướng giữa các trang trong react native

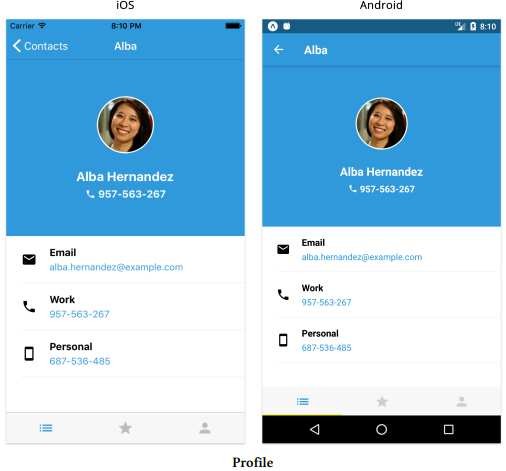
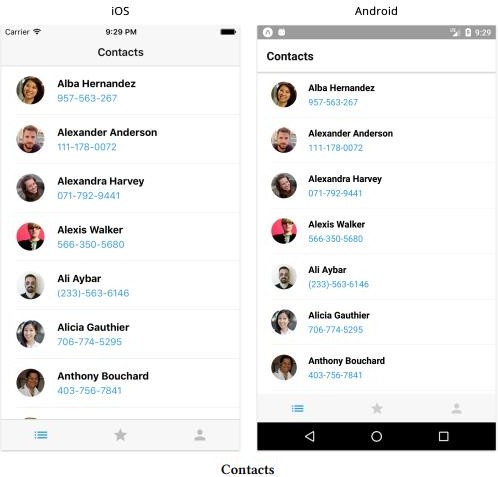
**Link** **1:** **<https://reactnative.dev/docs/getting-started>**

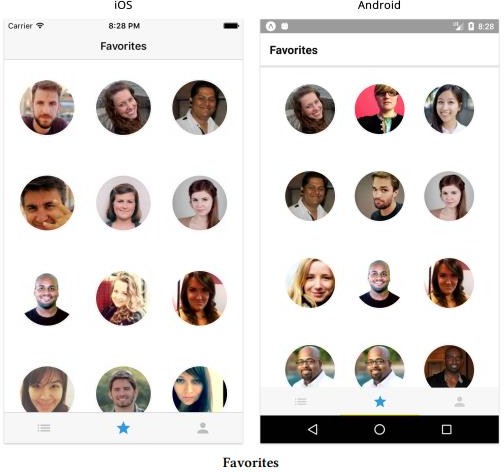
**Link** **2:** **<https://reactnative.dev/docs/navigation>**

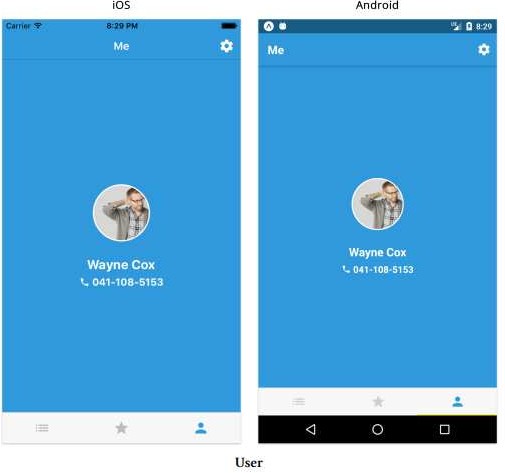
****

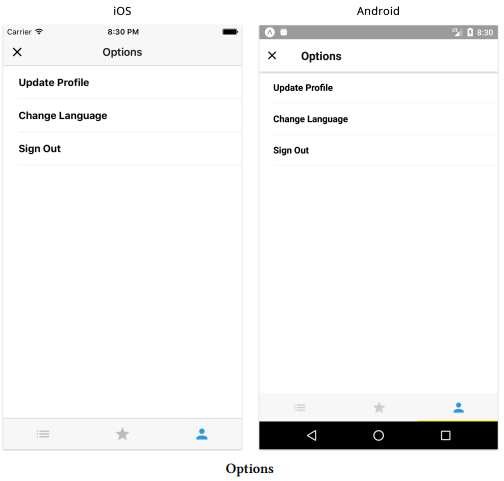
**Contact** **List**

Trong chương này, chúng ta sẽ xây dựng một ứng dụng **Contact-List** cho phép người dùng xem thông tin liên hệ trên nhiều màn hình. Chúng ta sẽ xây dựng màn hình đầu tiên, **Contacts.js,** hiển thị danh sách thông tin liên hệ được tìm nạp từ một API từ xa.









**Starting** **the** **project**

**>>npx** **create-expo-app** **RNLab2-ContactLists**

**Tạo** **các** **folder** **sau:**

* **Thư mục** **asset:** chứa ảnh project, link

# **Thư mục** **utility:** chứa các file java script tiện tích (api.js, colors.js, getURLParams.js)

* **Thư mục** **components:** chứa 3 component chính của ứng dụng.

**Folder** **utility**

**./utility/api.js**

**Các** **API** **của** **ứng** **dụng**

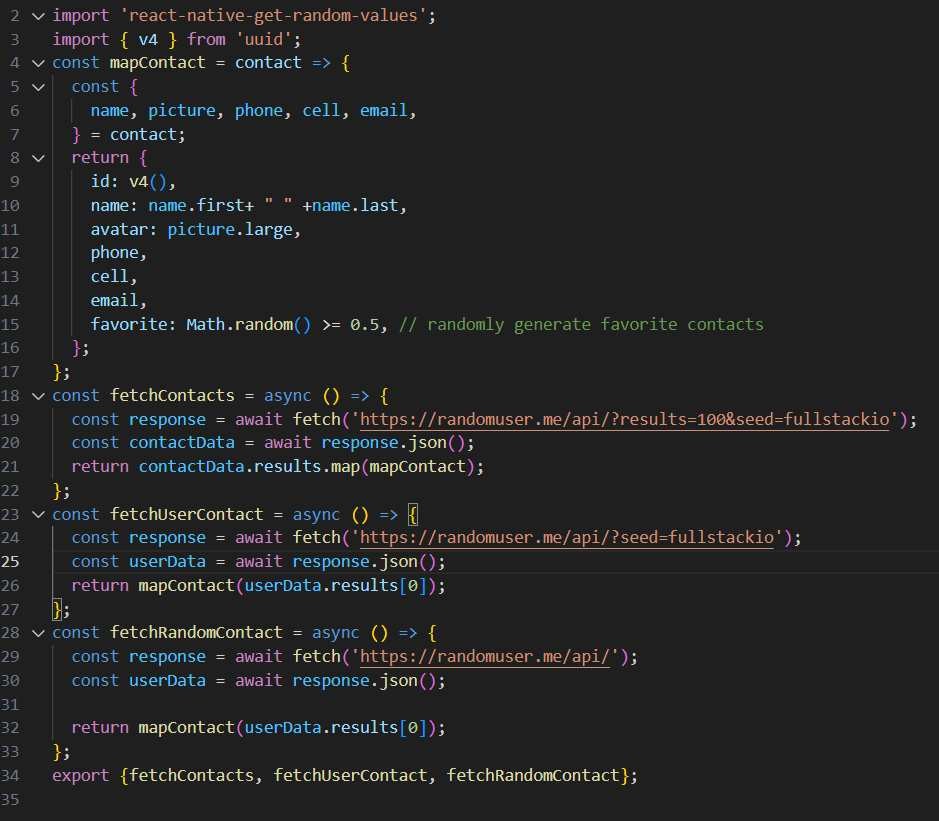
* **fetchContacts:** **<https://randomuser.me/api/?results=100&seed=fullstackio>**
* **fetchUserContact:** **<https://randomuser.me/api/?seed=fullstackio>**
* **fetchRandomContact** **<https://randomuser.me/api/>**

**UUID** **là** **một** **giá** **trị mã định** **danh** **duy** **nhất** **với 128 bit, được** **gọi** **là** **Universally** **Unique**

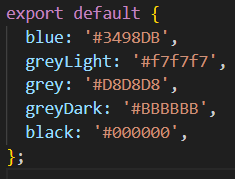
**Identifier. Đây là một** **công** **cụ quan** **trọng** **cho** **việc** **nhận** **dạng** **duy** **nhất** **và** **không** **trùng** **lặp.**

* <https://github.com/uuidjs/uuid>
* **npm** **install** **uuid**
* React Native / Expo: **npm** **Install** **react-native-get-random-values**

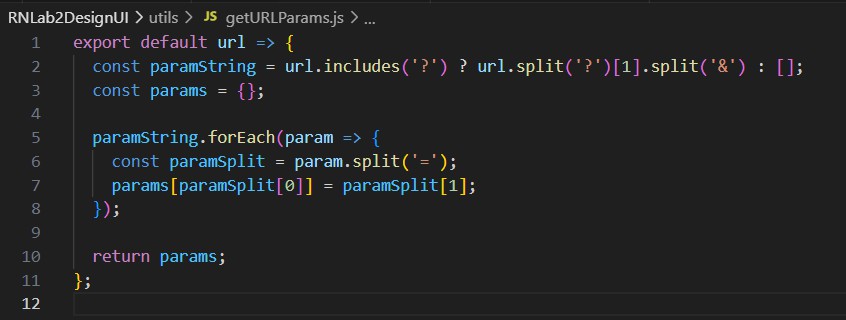
**./utility/api.js**

****

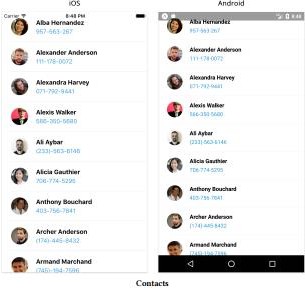
**./utility/colors.js**

****

**./utility/getURLParams.js**

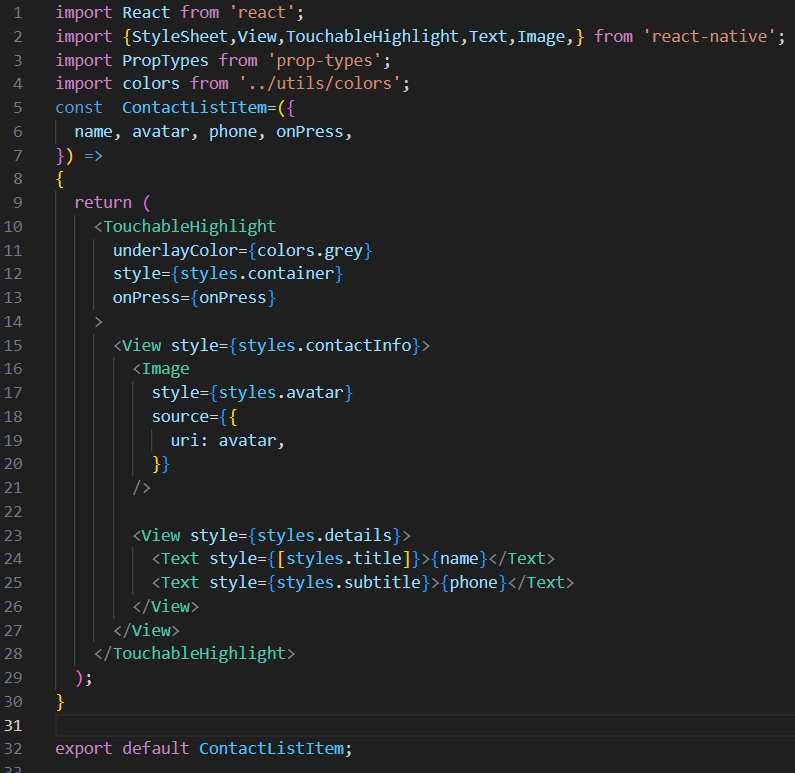


**Contacts** **Screen**

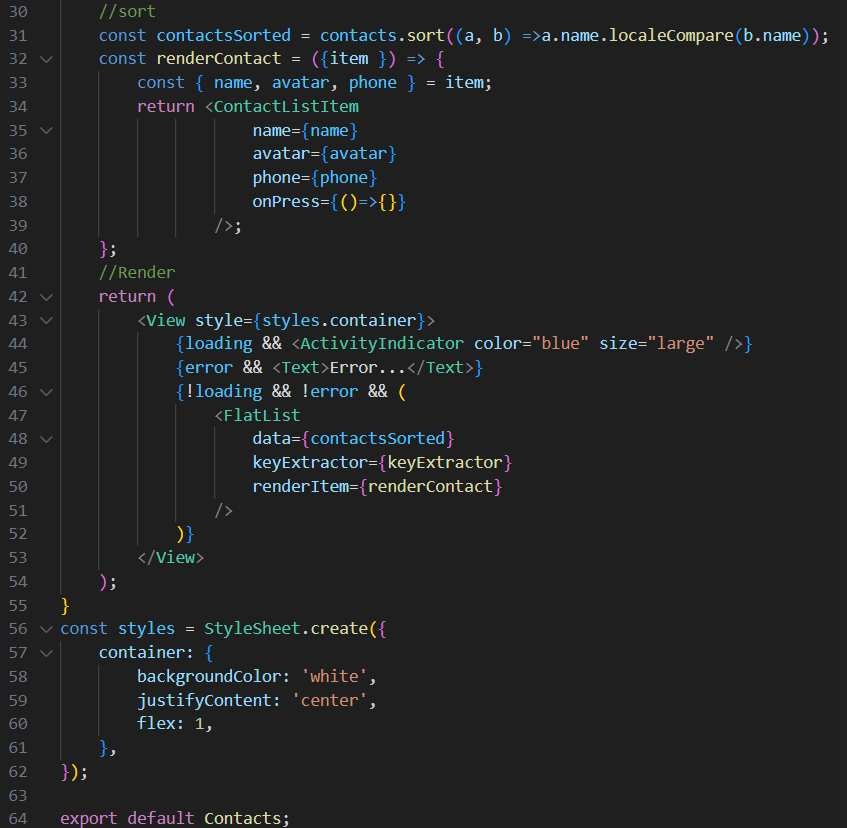
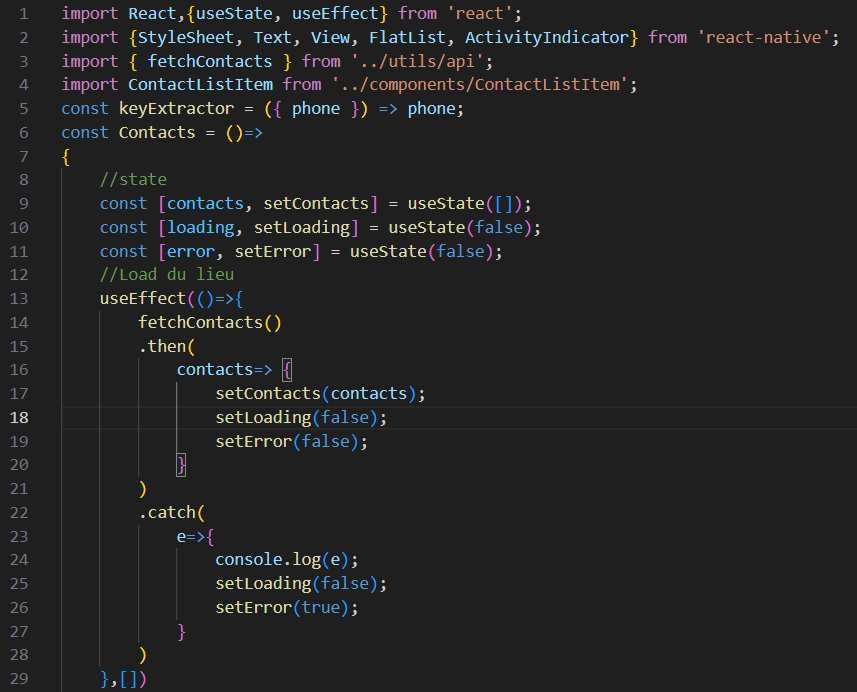


**ContactListItem.js**

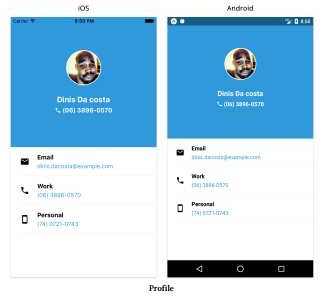
**./components/** **ContactListItem.js**

****

***./screens/Contacts.js***

******

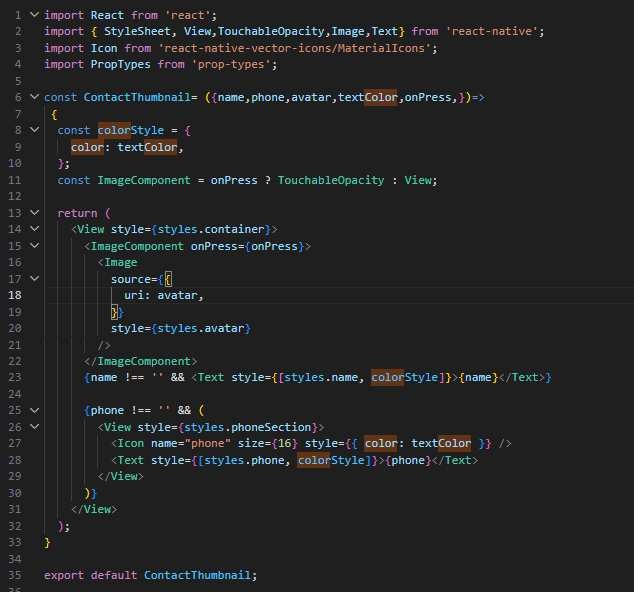
**Profile** **Screen**

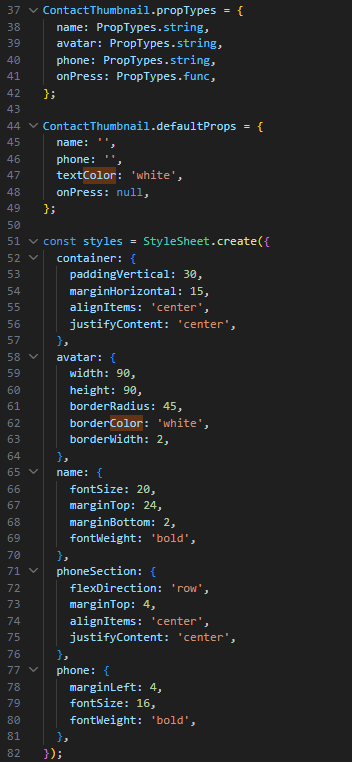


**ContactThumbnail.js**

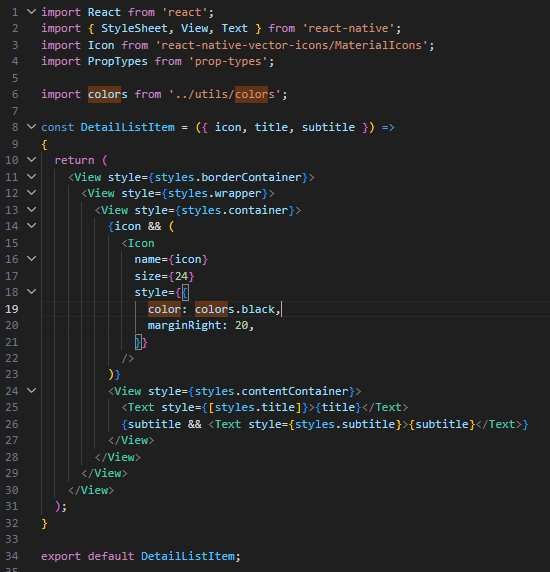
**ContactThumbnail.js**

**./components/** **ContactThumbnail.js**

****

****

**./components/** **DetailListItem.js**

****

****

***./screens/Profile.js***

******

**React** **Navigation**

*Cài thư viện* *React* *Navigattion* *[link](https://reactnavigation.org/docs/getting-started)* *Cập* *nhật* *Package.json*

{

…

"dependencies": {

…

"react-native-safe-area-context": "3.3.2", "react-native-screens": "~3.8.0",

..

"@react-navigation/native":"6.0.6",

..

},

…

}

*Cài thư viện* *bổ sung* *trong* *pakage.json*

*>npm* *install*

**Stack** **navigation**

*Cài thư viện* *Stack* *Navigattion* *[link](https://reactnavigation.org/docs/getting-started)* *Cập* *nhật* *Package.json*

"@react-navigation/stack":"6.0.11"

*Tạo* *component* ***Routes*** **với** *Stack* *navigation*

***contact-list/2/routes.js***

import React from 'react';

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

import {NavigationContainer} from '@react-navigation/native';

import {createNativeStackNavigator} from '@react-navigation/native-stack'; import {createStackNavigator} from '@react-navigation/stack';

import Contacts from './screens/Contacts'; import Profile from './screens/Profile'; import colors from '../utils/colors';

const Stack = createStackNavigator(); const StackNavigator = ()=>

{

return (

<NavigationContainer>

<Stack.Navigator

initialRouteName="Contacts"

>

<Stack.Screen name='Contacts' component={Contacts} options={{title:"Contacts"}}/>

<Stack.Screen name='Profile' component={Profile} options={{title:

"Profile"}}

/>

</Stack.Navigator>

</NavigationContainer>

);

}

export default StackNavigator;

**Cập** **nhật** ***Contacts***

*contact-list/2/screens/Contacts.js*

const Contacts = ({navigation})=>

{

…

const renderContact = ({item}) => {

const { name, avatar, phone } = item; return <ContactListItem

name={name} avatar={avatar} phone={phone}

onPress={() => navigation.navigate("Profile",{ contact: item })}

/>;

};

…

}

**Cập** **nhật** **Profile**

***contact-list/2/screens/Profile.js***

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

import ContactThumbnail from '../../components/ContactThumbnail'; import DetailListItem from '../../components/DetailListItem'; import { fetchRandomContact } from '../../utils/api';

import colors from '../../utils/colors';

const Profile =({ route})=>

{

const {contact} = route.params;

const {avatar, name, email, phone, cell}= contact; return (

<View style={styles.container}>

<View style={styles.avatarSection}>

<ContactThumbnail avatar={avatar} name={name} phone={phone} />

</View>

<View style={styles.detailsSection}>

<DetailListItem icon="mail" title="Email" subtitle={email} />

<DetailListItem icon="phone" title="Work" subtitle={phone} />

<DetailListItem icon="smartphone" title="Personal" subtitle={cell} />

</View>

</View>

);

}

const styles = StyleSheet.create({ container: {

flex: 1,

},

avatarSection: { flex: 1,

alignItems: 'center', justifyContent: 'center', backgroundColor: colors.blue,

},

detailsSection: { flex: 1,

backgroundColor: 'white',

},

});

export default Profile;

**Navigation**

const Contacts = ({navigation})=>

onPress={() => navigation.navigate("Profile")}

**Navigation** **parameters**

onPress={() => navigation.navigate("Profile",{ contact: item })}

const Profile =({ route})=>

{

const {contact} = route.params;

**Navigation** **screen** **options**

Cập nhật *contact-list/2/Route.js*

…..

const StackNavigator = ()=>

{

return (

<NavigationContainer>

<Stack.Navigator initialRouteName="Contacts" screenOptions={{

headerTintColor: 'white',

headerStyle: { backgroundColor: 'tomato' }, headerTitleAlign:'center',

}}

>

<Stack.Screen name='Contacts' component={Contacts}

options={{title:"Contacts"}}/>

<Stack.Screen

name='Profile' component={Profile} options={({route})=>

{

const {contact} = route.params; const {name}= contact;

return {

title: name.split(' ')[0], headerTintColor: 'white', headerStyle: {

backgroundColor: colors.blue,

}

};

}

}

/>

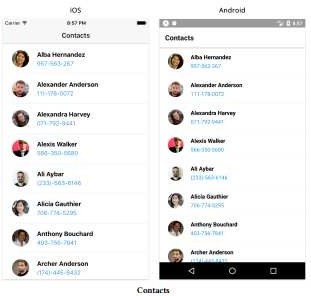
</Stack.Navigator>

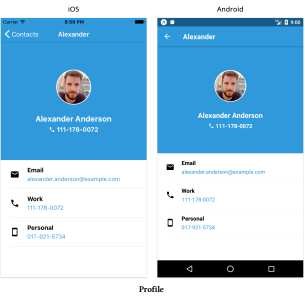
</NavigationContainer>

);

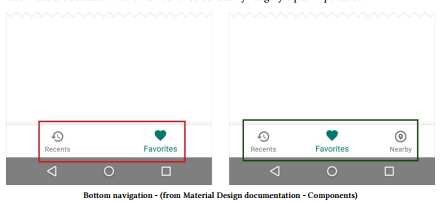
}

export default StackNavigator;





**Tab** **navigation**

****

**Favorites**

**contact-list/3/screens/Favorites.js**

import React,{useState, useEffect} from 'react'; import {

StyleSheet, Text,

View, FlatList,

ActivityIndicator,

} from 'react-native';

import { fetchContacts } from '../../utils/api';

import ContactThumbnail from '../../components/ContactThumbnail';

const keyExtractor = ({ phone }) => phone; const Favorites = ({navigation})=>

{

//state

const [contacts, setContacts] = useState([]); const [loading, setLoading] = useState(false); const [error, setError] = useState(false);

//Load du lieu useEffect(()=>{

fetchContacts()

.then(

contacts=> {

setContacts(contacts);

setLoading(false); setError(false);

}

)

.catch(

e=>{

setLoading(false); setError(true);

}

)

})

const renderFavoriteThumbnail = ({ item }) => { const { avatar } = item;

return (

<ContactThumbnail avatar={avatar}

onPress={() => navigation.navigate('Profile', { contact: item })}

/>

);

};

const favorites = contacts.filter(contact => contact.favorite);

return (

<View style={styles.container}>

{loading && <ActivityIndicator size="large" />}

{error && <Text>Error...</Text>}

{!loading && !error && (

<FlatList

data={favorites} keyExtractor={keyExtractor} numColumns={3} contentContainerStyle={styles.list} renderItem={renderFavoriteThumbnail}

/>

)}

</View>

);

}

const styles = StyleSheet.create({ container: {

backgroundColor: 'white', justifyContent: 'center', flex: 1,

},

list: {

alignItems: 'center',

},

});

export default Favorites;



**User** **screen**

**contact-list/3/screens/User.js**

import React, {useState, useEffect} from 'react';

import { StyleSheet, Text, View, ActivityIndicator } from 'react-native'; import ContactThumbnail from '../../components/ContactThumbnail';

import colors from '../../utils/colors';

import { fetchUserContact } from '../../utils/api';

const User = ()=>

{

const [user, setUser] = useState([]);

const [loading, setLoading] = useState(true); const [error, setError] =useState(false);

//Load du lieu useEffect(()=>{

fetchUserContact()

.then(

users=> {

setUser(users); setLoading(false); setError(false);

}

)

.catch(

e=>{

setLoading(false); setError(true);

}

)

});

const { avatar, name, phone } = user;

return (

<View style={styles.container}>

{loading && <ActivityIndicator size="large" />}

{error && <Text>Error...</Text>}

{!loading && !error (

<ContactThumbnail avatar={avatar} name={name} phone={phone} />

)}

</View>

);

}

const styles = StyleSheet.create({ container: {

flex: 1,

alignItems: 'center', justifyContent: 'center', backgroundColor: colors.blue,

},

});

export default User;

****

**Nested** **navigators**

**Cài** **pakages** **của** **Material** **Bottom** **Tabs** **([link](https://reactnavigation.org/docs/material-bottom-tab-navigator))** **Cập** **pakage.json**

"react-native-vector-icons":"9.0.0",

"@react-navigation/material-bottom-tabs": "6.0.9", "react-native-paper":"4.10.0",

"@expo/vector-icons":"12.0.5"

**>>npm** **install**

**Cập** **Routes**

*Constact-list/3/routes.js*

import React from 'react';

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

import {NavigationContainer} from '@react-navigation/native';

import {createNativeStackNavigator} from '@react-navigation/native-stack'; import Contacts from './screens/Contacts';

import Profile from './screens/Profile';

import Favorites from '../3/screens/Favorites'; import User from './screens/User';

import { MaterialIcons } from '@expo/vector-icons'; import colors from '../utils/colors';

import { createMaterialBottomTabNavigator } from '@react-navigation/material-bottom-tabs';

const getTabBarIcon = icon => ({ tintColor }) => (

<MaterialIcons name={icon} size={26} style={{ color: tintColor }} />

);

const Stack = createNativeStackNavigator(); const ContactsScreens = ()=>

{

return (

<Stack.Navigator initialRouteName="Contacts" screenOptions={{

headerTintColor: 'white',

headerStyle: { backgroundColor: 'tomato' }, headerTitleAlign:'center',

}}

>

<Stack.Screen name='Contacts' component={Contacts}

options={{title:"Contacts"}}/>

<Stack.Screen

name='Profile' component={Profile} options={({route})=>

{

const {contact} = route.params; const {name}= contact;

return {

title: name.split(' ')[0], headerTintColor: 'white', headerStyle: {

backgroundColor: colors.blue,

}

};

}

}

/>

</Stack.Navigator>

);

}

const FavoritesScreens = ()=>

{

return (

<Stack.Navigator initialRouteName="Favorites"

>

<Stack.Screen name='Favorites' component={Favorites}

options={{title:"Favorites"}}/>

<Stack.Screen name='Profile' component={Profile} options={{title:"Profile"}}/>

</Stack.Navigator>

);

}

const UserScreens = ({navigation})=>

{

return (

<Stack.Navigator initialRouteName="User"

>

<Stack.Screen name='User' component={User} />

</Stack.Navigator>

);

}

const Tab = createMaterialBottomTabNavigator(); const TabNavigator= ()=>

{

return(

<NavigationContainer>

<Tab.Navigator

initialRouteName= 'ContactsScreens'

barStyle= {{backgroundColor: colors.blue}} labeled={false} activeTintColor={colors.greyLight} inactiveColor = {colors.greyDark}

>

<Tab.Screen name="ContactsScreens" component={ContactsScreens} options={{

tabBarIcon: getTabBarIcon('list'),

}}

/>

<Tab.Screen name="FavoritesScreens" component={FavoritesScreens} options={{

tabBarIcon: getTabBarIcon('star'),

}}

/>

<Tab.Screen name="UserScreens" component={UserScreens} options={{

tabBarIcon: getTabBarIcon('person'),

}}

/>

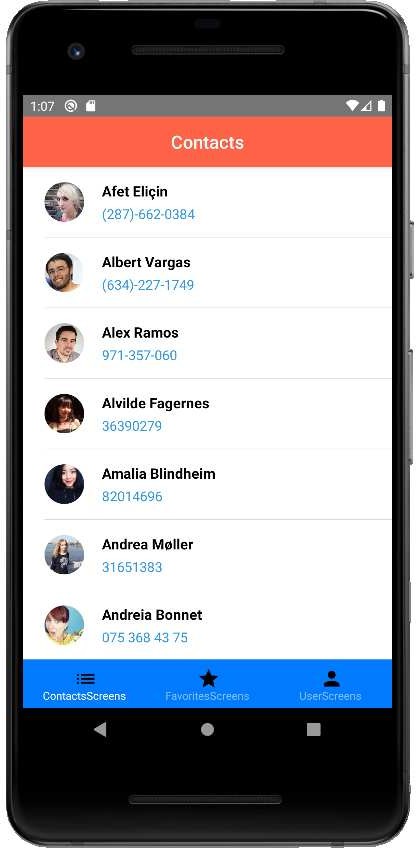
</Tab.Navigator>

</NavigationContainer>

);

}

export default TabNavigator;



**Modal** **navigation**

***contact-list/3/screens/Options.js***

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

import DetailListItem from '../../components/DetailListItem';

const Options =()=>

{

return (

<View style={styles.container}>

<DetailListItem title="Update Profile" />

<DetailListItem title="Change Language" />

<DetailListItem title="Sign Out" />

</View>

);

}

const styles = StyleSheet.create(

{

container:{

flex:1, backgroundColor:'white',

},

}

);

export default Options;

**Cập** **nhật** **UserScreens**

*Constact-list/3/routes.js*

const UserScreens = ({navigation})=>

{

return (

<Stack.Navigator initialRouteName="User"

>

<Stack.Screen name='User' component={User} options={{

headerTitle:"Me", headerTintColor: 'white', headerStyle: {

backgroundColor: colors.blue,

},

headerRight: ()=>(

<MaterialIcons name="settings" size={24}

style={{ color: 'white', marginRight: 10 }} onPress={() => navigation.navigate('Options')}

/>

),

}}

/>

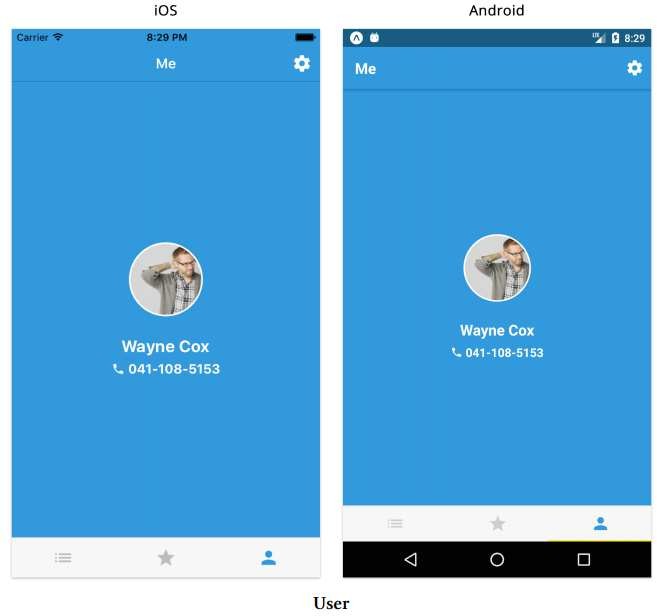
<Stack.Screen name='Options' component={Options} options={{title:"Options"}}/>

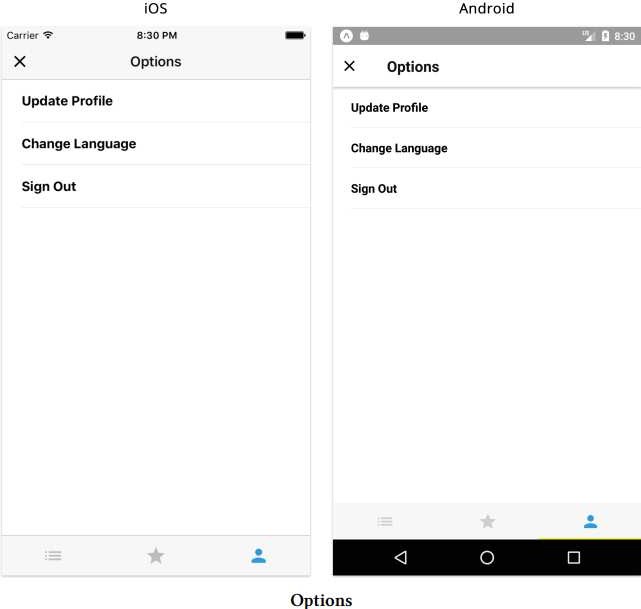
</Stack.Navigator>

);

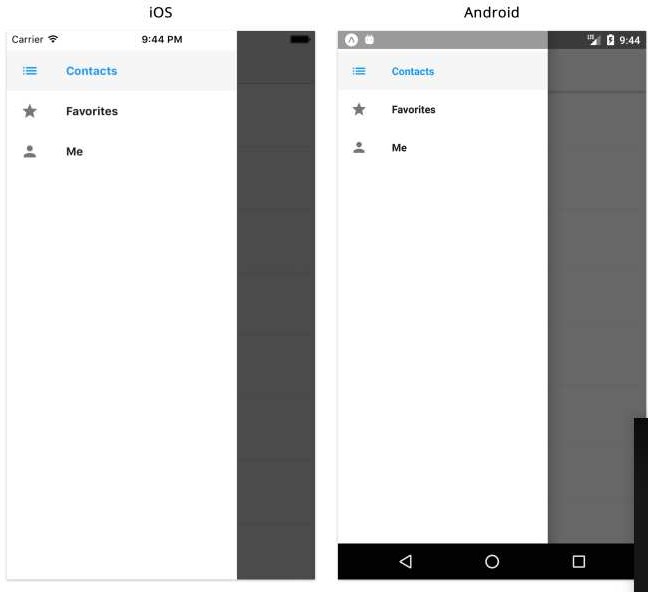
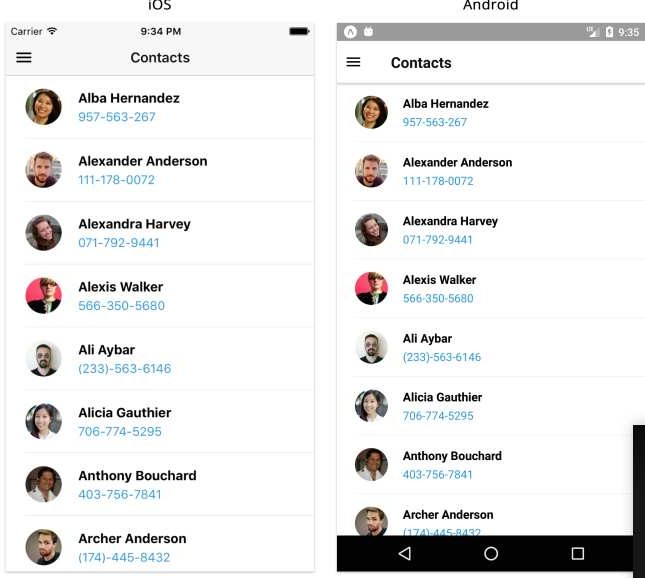
}

**Kết** **quả thực** **thi**

****



**Drawer** **navigation**

****

**Cài** **Package** **của** **Drawer** **[link](https://reactnavigation.org/docs/drawer-navigator)** **Cập** **nhật** **package.json**

"react-native-gesture-handler": "~1.10.2",

"react-native-reanimated": "~2.2.0", "@react-navigation/drawer":"6.1.8"

**>>npm** **install**

**Cập** **nhập** **routes.js**

***contact-list/4/routes.js***

import React from 'react';

import {NavigationContainer} from '@react-navigation/native';

import {createNativeStackNavigator} from '@react-navigation/native-stack'; import { createDrawerNavigator } from '@react-navigation/drawer';

import Contacts from './screens/Contacts'; import Profile from './screens/Profile';

import Favorites from '../3/screens/Favorites'; import User from './screens/User';

import { MaterialIcons } from '@expo/vector-icons'; import colors from '../utils/colors';

import Options from './screens/Options';

const getDrawerItemIcon = icon => ({ tintColor }) => (

<MaterialIcons name={icon} size={22} style={{ color: tintColor }} />

);

const Stack = createNativeStackNavigator(); const ContactsScreens = ()=>

{

return (

<Stack.Navigator initialRouteName="Contacts" screenOptions ={

{

headerShown:false

}

}

>

……….

);

}

const FavoritesScreens = ()=>

{

return (

<Stack.Navigator initialRouteName="Favorites" screenOptions ={

{

headerShown:false

}

}

>

…….

);

}

const Drawer = createDrawerNavigator(); const DrawerNavigator= ()=>

{

return(

<NavigationContainer>

<Drawer.Navigator

initialRouteName= 'ContactsScreens'

>

<Drawer.Screen name="ContactsScreens" component={ContactsScreens} options={{

drawerIcon: getDrawerItemIcon('list'),

}}

/>

<Drawer.Screen name="FavoritesScreens" component={FavoritesScreens} options={{

drawerIcon: getDrawerItemIcon('star'),

}}

/>

<Drawer.Screen name="UserScreens" component={UserScreens} options={{

drawerIcon: getDrawerItemIcon('person'),

}}

/>

</Drawer.Navigator>

</NavigationContainer>

);

}

export default DrawerNavigator;

**Sharing** **state** **between** **screens** **Update** **package.json**

"@reduxjs/toolkit":"1.6.2", "react-redux": "7.2.6"

**>** **npm** **install**

**Create** **store.js**

import { createSlice, configureStore } from '@reduxjs/toolkit'; const contactsSlice = createSlice({

name: 'counter', initialState: {

contacts: [], loading: false, error: false,

},

reducers: {

fetchContactsLoading: (state, action) =>

{

state.loading = true; state.loading = false;

// return ({

// ...state,

// loading: true,

// error: false,

// });

},

fetchContactsSuccess: (state, action) =>

{

state.contacts = action.payload; state.loading = false; error.loading = false;

// return ({

// contacts: action.payload,

// loading: false,

// error: false,

// })

},

fetchContactsError: (state, action) =>

{

return ({

...state, loading: false, error: true,

});

},

}

})

export const {fetchContactsLoading, fetchContactsSuccess, fetchContactsError} = contactsSlice.actions;

export default Store = configureStore({ reducer: contactsSlice.reducer,

})

**Cập** **nhật** **Contact.js**

import {fetchContactsLoading, fetchContactsSuccess, fetchContactsError} from '../store'; import { useDispatch, useSelector } from 'react-redux';

const Contacts = ({navigation})=>

{

const {contacts,loading,error} = useSelector((state) =>state); const dispatch = useDispatch();

//Load du lieu useEffect(()=>{

dispatch(fetchContactsLoading()); fetchContacts()

.then(

contacts=> {

dispatch(fetchContactsSuccess(contacts));

}

)

.catch(

e=>{

dispatch(fetchContactsError());

}

)

},[])

//sort

const contactsSorted = contacts.slice().sort((a, b) => a.name.localeCompare(b.name));

……………

# Cập nhật Favorites.js

import { useDispatch, useSelector } from 'react-redux';

const Favorites = ()=>

{

//state

const {contacts,loading,error} = useSelector((state) =>state);

……………

Cập nhật App.js

import React from 'react';

import {View, Text} from 'react-native'; import Contacts from './screens/Contacts'; import Profile from './screens/Profile'; import DrawerNavigator from './routes'; import Favorites from './screens/Favorites'; import User from './screens/User';

import Options from './screens/Options'; import Store from './store';

import { Provider } from 'react-redux';

const App=()=>

{

return(

<Provider store={Store}>

<DrawerNavigator/>

</Provider>

);

}

export default App;

Link tham khảo: <https://redux.js.org/introduction/getting-started>

Link React Redux: <https://react-redux.js.org/introduction/getting-started> Link redux toolkit : <https://redux-toolkit.js.org/introduction/getting-started>