

REACT NAVIGATION

Link tham khảo:

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

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

Content

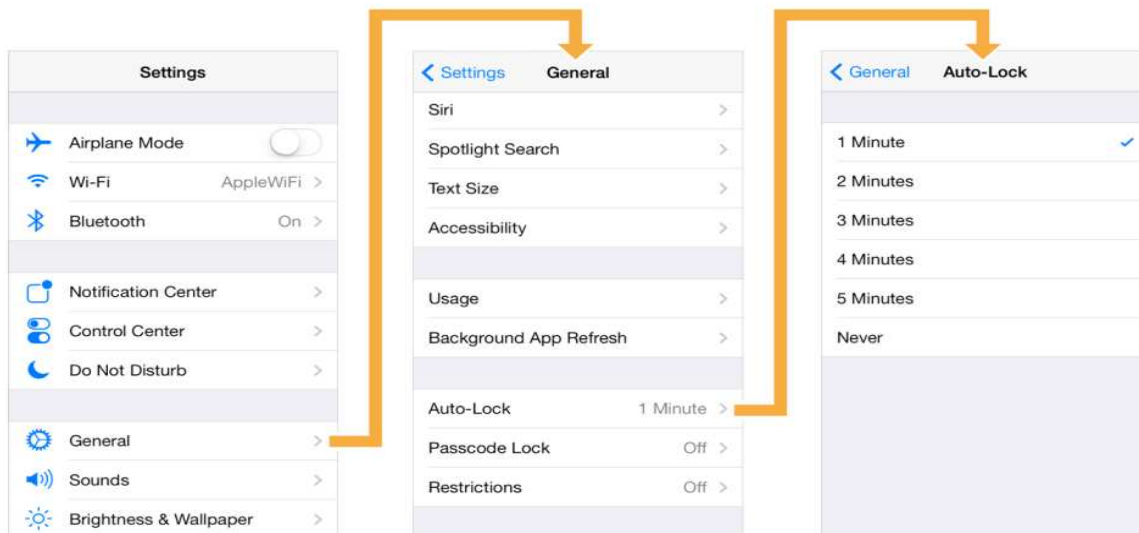
- Navigation in React Native
- Contact List
- Starting the project
- Contacts
- Profile
- React Navigation
- Stack navigation
- Tab navigation
- Drawer navigation
- Sharing state between screens

Navigation in React Native

Là việc điều hướng giữa các trang trong react native

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

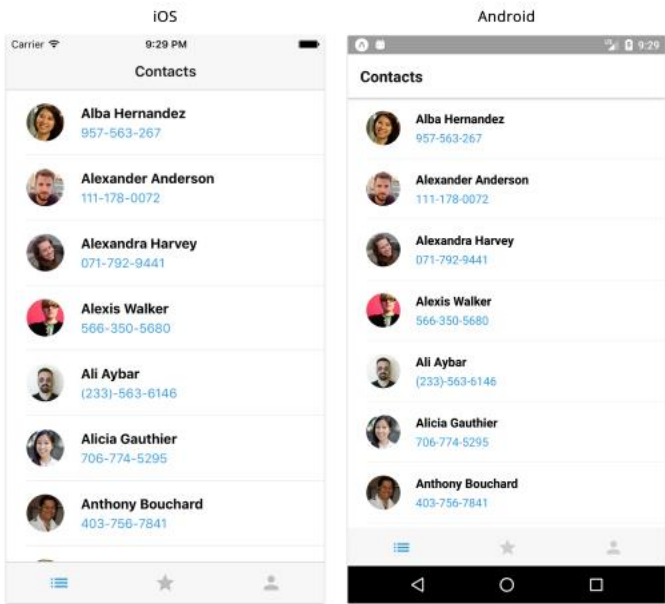
Link 2: <https://reactnative.dev/docs/navigation>



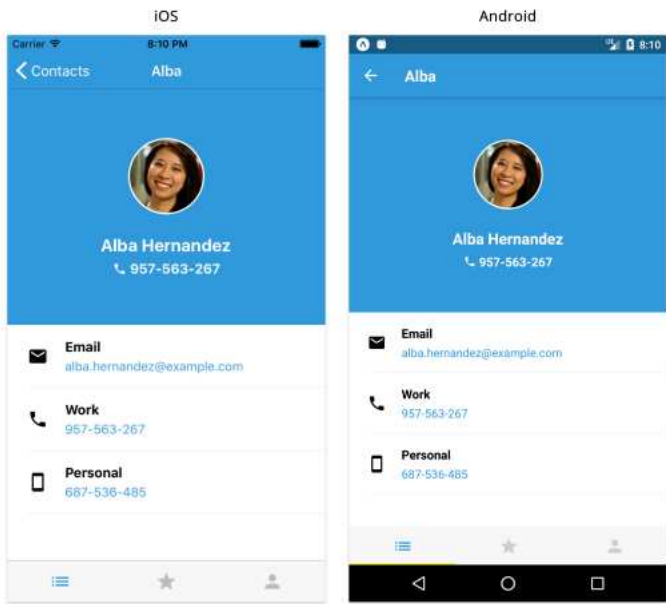
Example of a navigation controller (from Apple Developer Documentation - UINavigationController)

Contact List

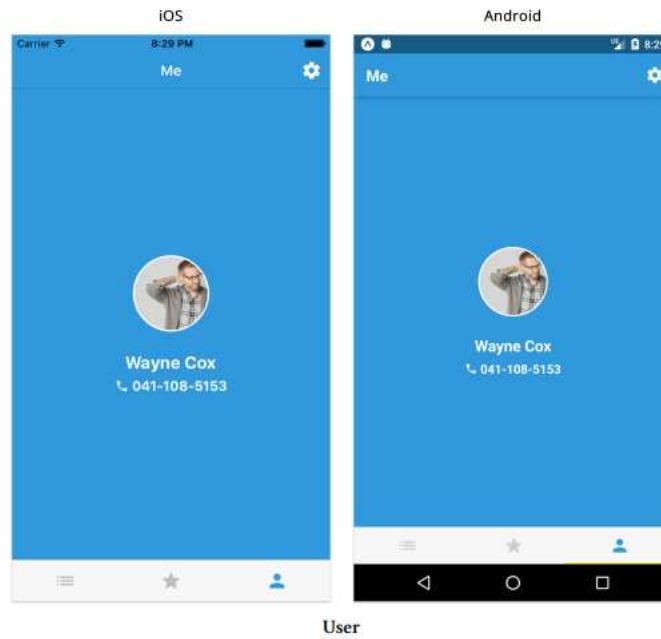
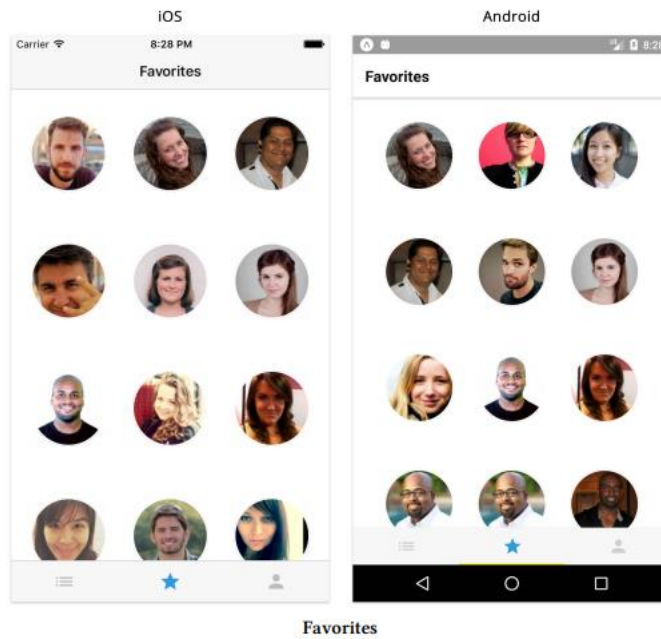
Trong chương này, chúng ta sẽ xây dựng một ứng dụng **Contact-List** cho phép người dùng xem thông tin liên hệ trên nhiều màn hình. Chúng ta sẽ xây dựng màn hình đầu tiên, **Contacts.js**, hiển thị danh sách thông tin liên hệ được tìm nạp từ một API từ xa.

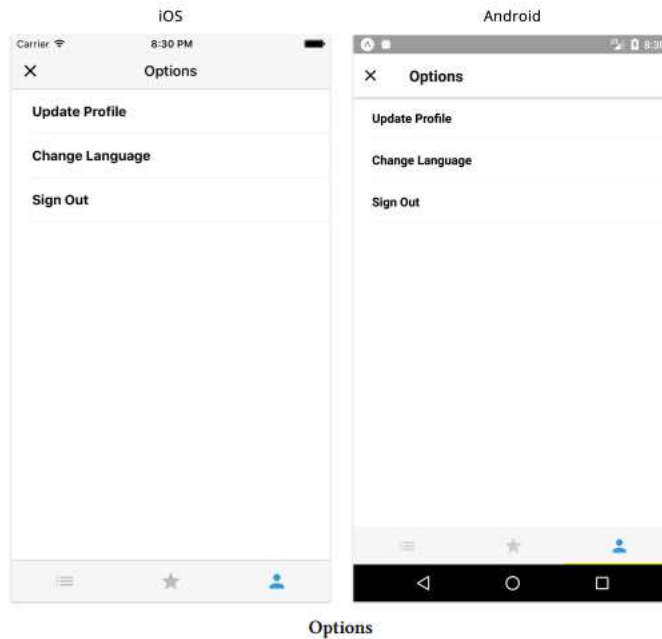


Contacts



Profile





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

```
2 import 'react-native-get-random-values';
3 import { v4 } from 'uuid';
4 const mapContact = contact => {
5   const {
6     name, picture, phone, cell, email,
7   } = contact;
8   return {
9     id: v4(),
10    name: name.first+ " " +name.last,
11    avatar: picture.large,
12    phone,
13    cell,
14    email,
15    favorite: Math.random() >= 0.5, // randomly generate favorite contacts
16  };
17 };
18 const fetchContacts = async () => {
19   const response = await fetch('https://randomuser.me/api/?results=100&seed=fullstackio');
20   const contactData = await response.json();
21   return contactData.results.map(mapContact);
22 };
23 const fetchUserContact = async () => {
24   const response = await fetch('https://randomuser.me/api/?seed=fullstackio');
25   const userData = await response.json();
26   return mapContact(userData.results[0]);
27 };
28 const fetchRandomContact = async () => {
29   const response = await fetch('https://randomuser.me/api/');
30   const userData = await response.json();
31
32   return mapContact(userData.results[0]);
33 };
34 export {fetchContacts, fetchUserContact, fetchRandomContact};
35
```

./utility/colors.js

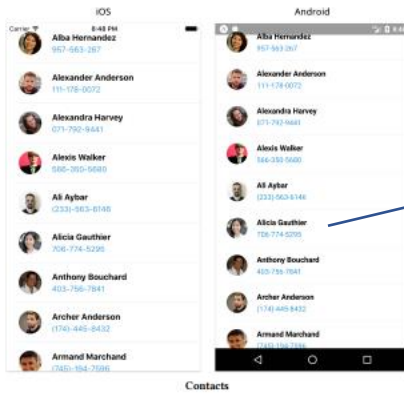
```
export default {
  blue: '#3498DB',
  greyLight: '#f7f7f7',
  grey: '#D8D8D8',
  greyDark: '#BBBBBB',
  black: '#000000',
};
```

./utility/getURLParams.js

RNLab2DesignUI > utils > JS getUrlParams.js > ...

```
1  export default url => {
2    const paramString = url.includes('?') ? url.split('?')[1].split('&') : [];
3    const params = {};
4
5    paramString.forEach(param => {
6      const paramSplit = param.split('=');
7      params[paramSplit[0]] = paramSplit[1];
8    });
9
10   return params;
11 };
12
```

Contacts Screen



ContactListItem.js

./components/ ContactListItem.js

```
1 import React from 'react';
2 import {StyleSheet,View,TouchableHighlight,Text,Image,} from 'react-native';
3 import PropTypes from 'prop-types';
4 import colors from '../utils/colors';
5 const ContactListItem=({
6   name, avatar, phone, onPress,
7 }) =>
8 {
9   return (
10     <TouchableHighlight
11       underlayColor={colors.grey}
12       style={styles.container}
13       onPress={onPress}
14     >
15       <View style={styles.contactInfo}>
16         <Image
17           style={styles.avatar}
18           source={{
19             uri: avatar,
20           }}
21         />
22
23         <View style={styles.details}>
24           <Text style={styles.title}>{name}</Text>
25           <Text style={styles.subtitle}>{phone}</Text>
26         </View>
27       </View>
28     </TouchableHighlight>
29   );
30 }
31
32 export default ContactListItem;
```

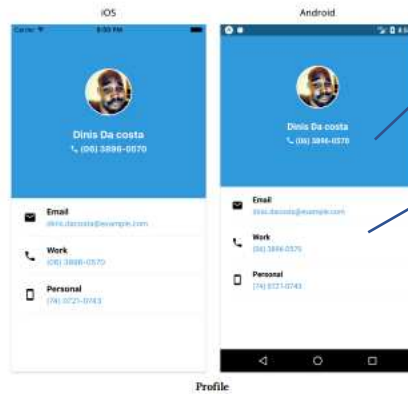
```
34 ContactListItem.propTypes = {
35   name: PropTypes.string,
36   avatar: PropTypes.string,
37   phone: PropTypes.string,
38   onPress: PropTypes.func,
39 };
40
41 const styles = StyleSheet.create({
42   container: {
43     paddingLeft: 24,
44   },
45   contactInfo: {
46     flex: 1,
47     flexDirection: 'row',
48     alignItems: 'center',
49     paddingTop: 16,
50     paddingBottom: 16,
51     paddingRight: 24,
52     borderBottomColor: colors.grey,
53     borderBottomWidth: StyleSheet.hairlineWidth,
54   },
55   avatar: {
56     borderRadius: 22,
57     width: 44,
58     height: 44,
59   },
60   details: {
61     justifyContent: 'center',
62     flex: 1,
63     marginLeft: 20,
64   },
65   title: {
66     color: colors.black,
67     fontWeight: 'bold',
68     fontSize: 16,
69   },
70   subtitle: {
71     color: colors.blue,
72     fontSize: 15,
73     marginTop: 4,
74   },
75 });
```


./screens/Contacts.js

```
1  import React,{useState, useEffect} from 'react';
2  import {StyleSheet, Text, View, FlatList, ActivityIndicator} from 'react-native';
3  import { fetchContacts } from '../utils/api';
4  import ContactListItem from '../components/ContactListItem';
5  const keyExtractor = ({ phone }) => phone;
6  const Contacts = ()=>
7  {
8    //state
9    const [contacts, setContacts] = useState([]);
10   const [loading, setLoading] = useState(false);
11   const [error, setError] = useState(false);
12   //Load du lieu
13   useEffect(()=>{
14     fetchContacts()
15     .then(
16       contacts=> {
17         setContacts(contacts);
18         setLoading(false);
19         setError(false);
20       }
21     )
22     .catch(
23       e=>{
24         console.log(e);
25         setLoading(false);
26         setError(true);
27       }
28     )
29   },[])
```

```
30   //sort
31   const contactsSorted = contacts.sort((a, b) =>a.name.localeCompare(b.name));
32   const renderContact = ({item}) => {
33     const { name, avatar, phone } = item;
34     return <ContactListItem
35       name={name}
36       avatar={avatar}
37       phone={phone}
38       onPress={()=>{}}
39     />;
40   };
41   //Render
42   return (
43     <View style={styles.container}>
44       {loading && <ActivityIndicator color="blue" size="large" />}
45       {error && <Text>Error...</Text>}
46       {!loading && !error && (
47         <FlatList
48           data={contactsSorted}
49           keyExtractor={keyExtractor}
50           renderItem={renderContact}
51         />
52       )}
53     </View>
54   );
55 }
56 const styles = StyleSheet.create({
57   container: {
58     backgroundColor: 'white',
59     justifyContent: 'center',
60     flex: 1,
61   },
62 });
63
64 export default Contacts;
```


Profile Screen



ContactThumbnail.js

ContactThumbnail.js

./components/ ContactThumbnail.js

```
1 import React from 'react';
2 import { StyleSheet, View,TouchableOpacity,Image,Text} from 'react-native';
3 import Icon from 'react-native-vector-icons/MaterialIcons';
4 import PropTypes from 'prop-types';
5
6 const ContactThumbnail= ({name,phone,avatar,textColor,onPress,})=>
7 {
8   const colorStyle = {
9     color: textColor,
10   };
11   const ImageComponent = onPress ? TouchableOpacity : View;
12
13   return (
14     <View style={styles.container}>
15       <ImageComponent onPress={onPress}>
16         <Image
17           source={{
18             uri: avatar,
19           }}
20           style={styles.avatar}
21         />
22       </ImageComponent>
23       {name !== '' && <Text style={[styles.name, colorStyle]}>{name}</Text>}
24
25       {phone !== '' && (
26         <View style={styles.phoneSection}>
27           <Icon name="phone" size={16} style={{ color: textColor }} />
28           <Text style={[styles.phone, colorStyle]}>{phone}</Text>
29         </View>
30       )}
31     </View>
32   );
33 }
34
35 export default ContactThumbnail;
```

```
37 ContactThumbnail.propTypes = {
38   name: PropTypes.string,
39   avatar: PropTypes.string,
40   phone: PropTypes.string,
41   onPress: PropTypes.func,
42 };
43
44 ContactThumbnail.defaultProps = {
45   name: '',
46   phone: '',
47   textColor: 'white',
48   onPress: null,
49 };
50
51 const styles = StyleSheet.create({
52   container: {
53     paddingVertical: 30,
54     marginHorizontal: 15,
55     alignItems: 'center',
56     justifyContent: 'center',
57   },
58   avatar: {
59     width: 90,
60     height: 90,
61     borderRadius: 45,
62     borderColor: 'white',
63     borderWidth: 2,
64   },
65   name: {
66     fontSize: 20,
67     marginTop: 24,
68     marginBottom: 2,
69     fontWeight: 'bold',
70   },
71   phoneSection: {
72     flexDirection: 'row',
73     marginTop: 4,
74     alignItems: 'center',
75     justifyContent: 'center',
76   },
77   phone: {
78     marginLeft: 4,
79     fontSize: 16,
80     fontWeight: 'bold',
81   },
82 });
```

./components/ DetailListItem.js

```
1 import React from 'react';
2 import { StyleSheet, View, Text } from 'react-native';
3 import Icon from 'react-native-vector-icons/MaterialIcons';
4 import PropTypes from 'prop-types';
5
6 import colors from '../utils/colors';
7
8 const DetailListItem = ({ icon, title, subtitle }) =>
9 {
10   return (
11     <View style={styles.borderContainer}>
12       <View style={styles.wrapper}>
13         <View style={styles.container}>
14           {icon && (
15             <Icon
16               name={icon}
17               size={24}
18               style={{
19                 color: colors.black,
20                 marginRight: 20,
21               }}
19             />
22           )}
23         <View style={styles.contentContainer}>
24           <Text style={styles.title}><{title}</Text>
25           {subtitle && <Text style={styles.subtitle}><{subtitle}</Text>}
26         </View>
27       </View>
28     </View>
29   );
30 }
31
32 export default DetailListItem;
```

```
36 DetailListItem.propTypes = {
37   icon: PropTypes.string,
38   title: PropTypes.string,
39   subtitle: PropTypes.string,
40 };
41
42 const styles = StyleSheet.create({
43   borderContainer: {
44     paddingLeft: 24,
45   },
46   wrapper: {
47     flexDirection: 'row',
48     paddingTop: 16,
49     paddingBottom: 16,
50     paddingRight: 24,
51     borderBottomColor: colors.grey,
52     borderBottomWidth: StyleSheet.hairlineWidth,
53   },
54   container: {
55     flex: 1,
56     flexDirection: 'row',
57     alignItems: 'center',
58   },
59   contentContainer: {
60     justifyContent: 'center',
61     flex: 1,
62   },
63   title: {
64     color: colors.black,
65     fontWeight: 'bold',
66     fontSize: 16,
67   },
68   subtitle: {
69     color: colors.blue,
70     fontSize: 15,
71     marginTop: 4,
72   },
73 });
```

./screens/Profile.js

```
1  import React,{useState,useEffect} from 'react';
2  import { StyleSheet, View , Text} from 'react-native';
3  import { fetchRandomContact } from '../utils/api';
4  import ContactThumbnail from '../components/ContactThumbnail';
5  import DetaillistItem from '../components/DetaillistItem';
6  import colors from '../utils/colors';
7  const Profile =()=>
8  {
9      const [contact, setContact] = useState({});
10     useEffect(()=>
11     {
12         fetchRandomContact().then(
13             contact => setContact(contact)
14         )
15     }
16     ,[]);
17     const {avatar, name, email, phone, cell}= contact;
18     return (
19         <View style={styles.container}>
20             <View style={styles.avatarSection}>
21                 <ContactThumbnail avatar={avatar} name={name} phone={phone} />
22             </View>
23             <View style={styles.detailsSection}>
24                 <DetaillistItem icon="mail" title="Email" subtitle={email} />
25                 <DetaillistItem icon="phone" title="Work" subtitle={phone} />
26                 <DetaillistItem icon="smartphone" title="Personal" subtitle={cell} />
27             </View>
28         </View>
29     );
30 }
31 export default Profile;
```

```
31  export default Profile;
32  const styles = StyleSheet.create({
33      container: {
34          flex: 1,
35      },
36      avatarSection: {
37          flex: 1,
38          alignItems: 'center',
39          justifyContent: 'center',
40          backgroundColor: colors.blue,
41      },
42      detailsSection: {
43          flex: 1,
44          backgroundColor: 'white',
45      },
46  });
47
```

React Navigation

Cài thư viện React Navigattion [link](#)

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 package.json

>npm install

Stack navigation

Cài thư viện Stack Navigattion [link](#)

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;

```

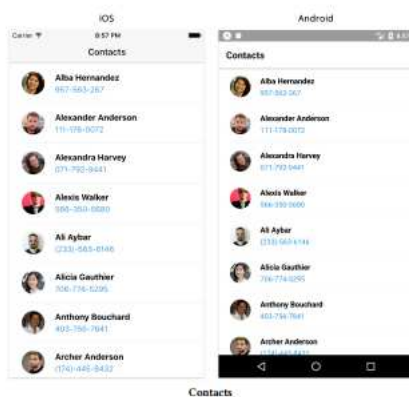
Navigation screen options

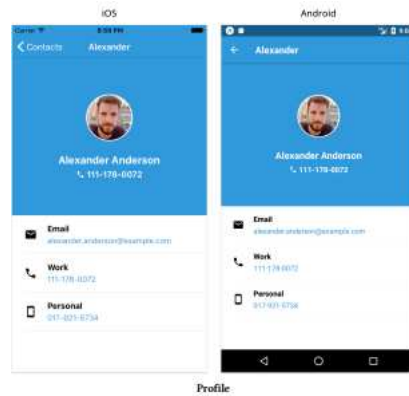
Cập nhật *contact-list/2/Route.js*

```

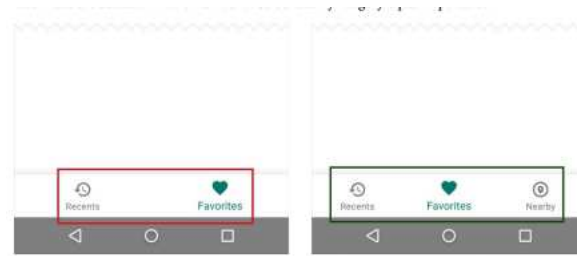
...
const StackNavigator = ()=>
{
  return (
    <NavigationContainer>
      <Stack.Navigator
        initialRouteName="Contacts"
        screenOptions={{
          headerTintColor: 'white',
          headerStyle: { backgroundColor: 'tomato' },
          headerTitleAlign:'center',
        }}
      >
        <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



Bottom navigation - (from Material Design documentation - Components)

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);

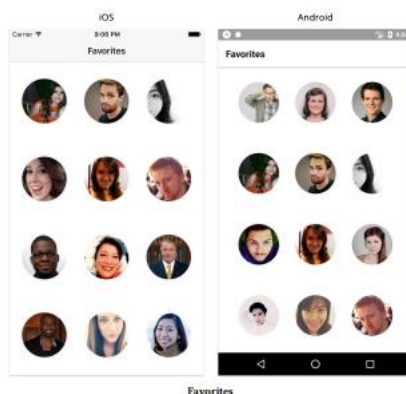
return (
    <View style={styles.container}>
        {loading && <ActivityIndicator size="large" />}
        {error && <Text>Error...</Text>}
        {!loading && !error && (
            <FlatList
                data={favorites}
                keyExtractor={keyExtractor}
                numColumns={3}
                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';

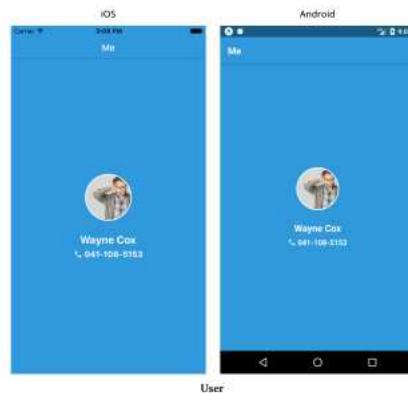
const User = ()=>
{
  const [user, setUser] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] =useState(false);
  //Load du lieu
  useEffect(()=>{
    fetchUserContact()
    .then(
      users=> {
        setUser(users);
        setLoading(false);
        setError(false);
      }
    )
    .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 packages của Material Bottom Tabs ([link](#))

Cập package.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.Navigator>
    </NavigationContainer>
  );
}

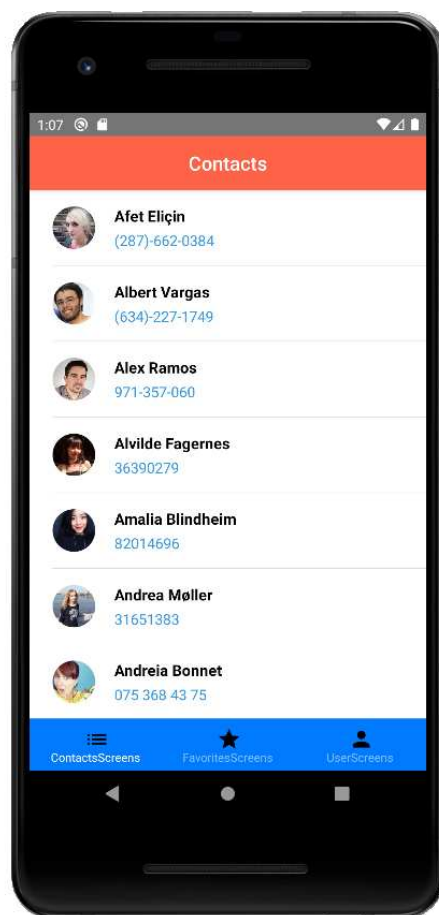
```

```

    <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

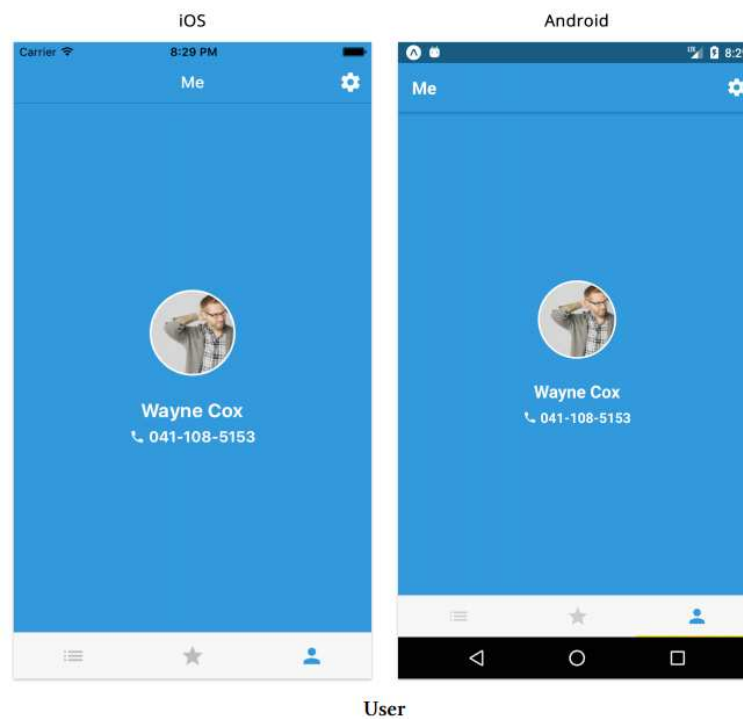
```

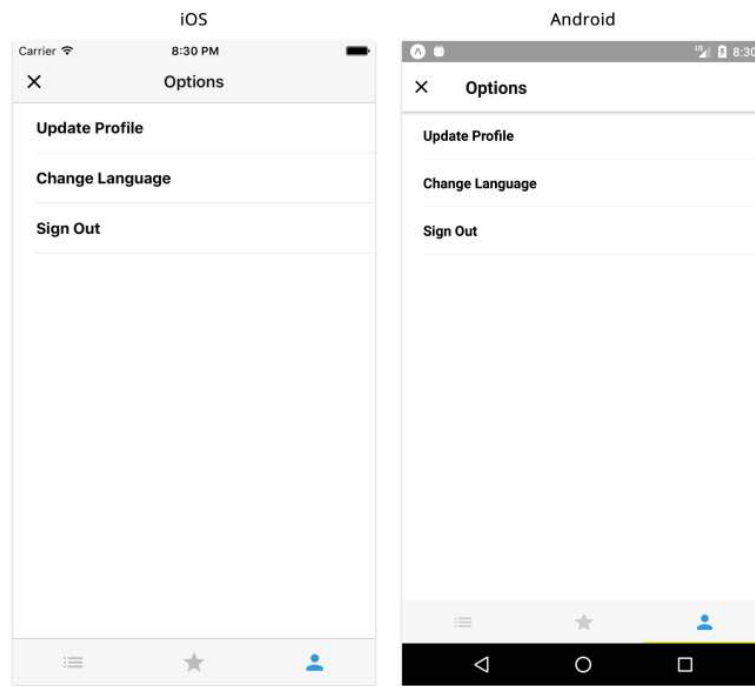
const UserScreens = ({navigation})=>
{
  return (
    <Stack.Navigator
      initialRouteName="User"
    >
      <Stack.Screen name='User' component={User}
        options={{
          headerTitle:"Me",
          headerTintColor: 'white',
          headerStyle: {
            backgroundColor: colors.blue,
          },
          headerRight: ()=>(
            <MaterialIcons
              name="settings"
              size={24}
              style={{ color: 'white', marginRight: 10 }}
              onPress={() => navigation.navigate('Options')}
            />
          ),
        }}
      />
    </Stack.Navigator>
  );
}

```

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

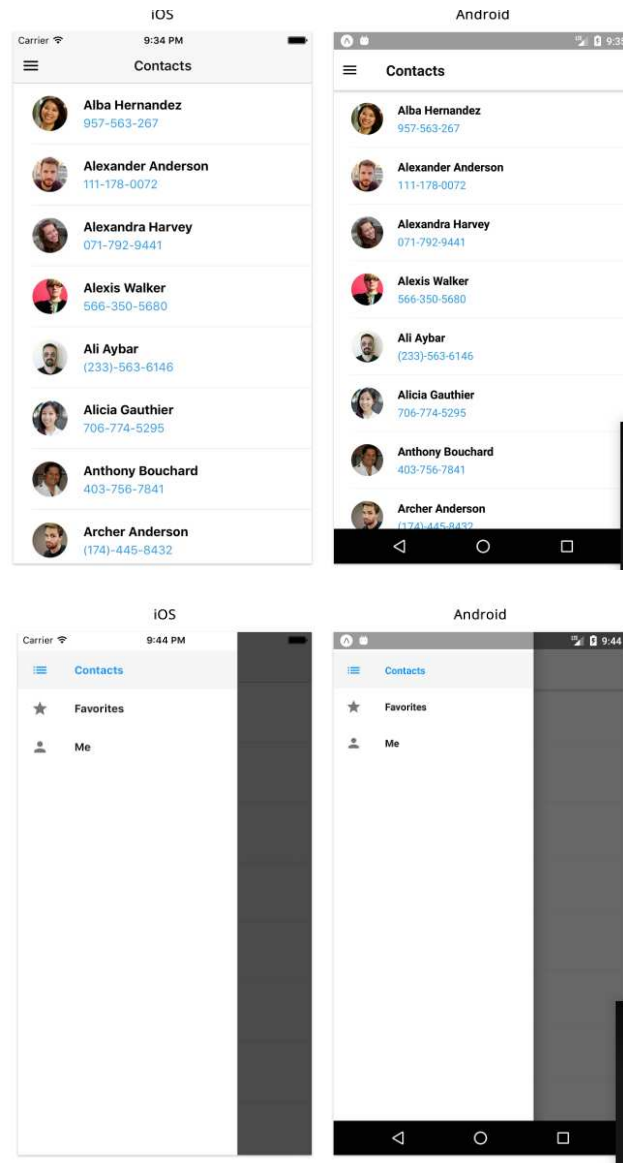
Kết quả thực thi





Options

Drawer navigation



Cài Package của Drawer [link](#)

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 nhật 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
        }
      }}
    >
      .....
    </Stack.Navigator>
  );
}

const FavoritesScreens = ()=>
{
  return (
    <Stack.Navigator
      initialRouteName="Favorites"
      screenOptions ={{
        {
          headerShown:false
        }
      }}
    >
      .....
    </Stack.Navigator>
  );
}

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>