

CAO ĐẲNG THỰC HÀNH FPT POLYTECHNIC

DỰ ÁN 2 – LẬP TRÌNH MOBILE

ĐỀ TÀI: ỨNG DỤNG ĐỌC TRUYỆN TRANH



MANGA READING APP

GVHD: Thầy Lê Phạm Tuấn Kiệt

NHÓM: 3

SINH VIÊN THỰC HIỆN:

1. Trịnh Quang Đồng - PS09349
2. Ngô Trọng Phúc – PS09279
3. Trần Công Khanh - PS09331
4. Phạm Hoàng Huy - PS09353
5. Đặng Lê Duy Nghĩa - PS07407

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày ... tháng ... năm

GIẢNG VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày ... tháng ... năm

HỘI ĐỒNG

MỤC LỤC

1	<i>Giới thiệu đề tài.....</i>	7
2	<i>Khảo sát yêu cầu.....</i>	8
2.1	Chức năng của người dùng (User).....	8
2.2	Chức năng của quản trị viên (Admin)	9
2.3	Yêu cầu về môi trường công nghệ	9
2.4	Mô tả thông tin thực thể quản lý	10
3	<i>Phân tích hiện trạng.....</i>	16
4	<i>Thiết kế hệ thống</i>	17
4.1	Sơ đồ use case	17
4.1.1	Người dùng (User).....	17
4.1.2	Quản trị viên (Admin)	17
4.2	Sơ đồ giao diện.....	18
4.2.1	Người dùng (User).....	18
4.2.2	Quản trị viên (Admin)	18
4.3	Giao diện ứng dụng.....	19
4.3.1	Người dùng (User).....	19
4.3.2	Quản trị viên (Admin)	20
4.4	Sơ đồ ERD.....	21
4.5	Sơ đồ tổ chức thực thể	22
5	<i>Thực hiện dự án</i>	23
5.1	Phân công dự án	23

5.2	Tổ chức dự án	24
5.3	Mã nguồn	24
5.3.1	Front-end	24
5.3.2	Back-end	24
5.4	Thư viện sử dụng.....	24
6	<i>Viết mã cho ứng dụng.....</i>	26
6.1	Ứng dụng	26
6.1.1	Màn hình đăng nhập.....	26
6.1.2	Màn hình đăng ký	31
6.1.3	Màn hình chính	37
6.1.4	Màn hình chi tiết	40
6.1.5	Màn hình tìm kiếm.....	48
6.2	Quản trị viên.....	51
6.2.1	Chức năng đăng nhập.....	51
6.2.2	Chức năng thêm, sửa truyện và chương	52
6.2.3	Chức năng tải truyện từ firebase.....	54
7	<i>Kiểm thử.....</i>	55
5.5	Kiểm thử đăng ký.....	55
5.6	Kiểm thử đăng nhập	55
5.7	Kiểm thử cập nhật thông tin người dùng	55
5.8	Kiểm thử đổi mật khẩu.....	56
5.9	Kiểm thử cập nhật thông tin truyện.....	56
6	<i>Triển khai và sử dụng.....</i>	57
6.1	Hướng dẫn và cài đặt.....	57
6.2	Hướng dẫn sử dụng phần mềm	57

7	<i>Kết luận</i>	58
7.1	Thuận lợi	58
7.2	Khó khăn	58
7.3	Nhưng phần chưa làm	58
8	<i>Tài liệu tham khảo</i>	58
-	Tài liệu tham khảo: GitHub, Stack Overflow, Google . . .	59

1 Giới thiệu đề tài

Trong xã hội công nghiệp hóa, hiện đại hóa hiện nay nhu cầu giải trí của mọi người ngày càng cao nhất là đối với giới trẻ. Bên cạnh các hình thức giải trí phổ biến như là chơi game, nghe nhạc,... Một số bạn trẻ lại yêu thích đọc truyện tranh. Nhận thấy nhu cầu này, đồng thời nhận thấy sự phổ biến cũng như phát triển của các thiết bị di động, chúng tôi quyết định xây dựng một ứng dụng đọc truyện tranh trên nền tảng di động android. Với tiêu chí đơn giản nhưng hiện đại, ứng dụng hướng tới sự tiện dụng cũng như trải nghiệm của người dùng.

2 Khảo sát yêu cầu

2.1 Chức năng của người dùng (User)

- Đăng ký tài khoản.
- Đăng nhập tài khoản.
- Đổi mật khẩu.
- Quên mật khẩu.
- Thay đổi ảnh đại diện.
- Cập nhật thông tin cá nhân.
- Điều hướng các màn hình.
- Bố cục, giao diện.
- Xem danh sách truyện.
- Xem thông tin truyện.
- Đọc truyện.
- Bình luận truyện.
- Thích truyện.
- Lưu truyện vào bộ sưu tập.
- Lưu lịch sử đọc truyện.
- Tăng lượt đọc khi đọc truyện.
- Tìm kiếm truyện.
- Xem danh sách thể loại.
- Sắp xếp ngẫu nhiên truyện.
- Thống kê truyện theo lượt đọc.
- Thống kê truyện theo lượt yêu thích.

2.2 Chức năng của quản trị viên (Admin)

- Đăng nhập tài khoản quản trị viên được cấp sẵn.
- Lưu mật khẩu.
- Quên mật khẩu.
- Bố cục giao diện.
- Thêm, xóa, sửa truyện.
- Thêm chương của truyện.
- Xem danh sách truyện.
- Tìm kiếm truyện.
- Xem danh sách thể loại.
- Thêm, xóa thể loại.
- Xem danh sách tài khoản người dùng và quản trị viên.
- Sắp xếp truyện.
- Xem danh sách chương truyện.
- Xem chương truyện.
- Tính năng darkmode.
- Hiệu ứng, trang trí.

2.3 Yêu cầu về môi trường công nghệ

- ✓ Ứng dụng được cài đặt trên điện thoại chạy hệ điều hành Android hoặc IOS.
- ✓ Hệ thống quản trị CSDL được sử dụng là Firebase.
- ✓ Web admin quản lý dữ liệu là HTML, CSS.

2.4 Mô tả thông tin thực thể quản lý

NGƯỜI DÙNG (USER)			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Họ và tên		Họ tên người dùng (không được để trống).
2	ID		Tên đăng nhập của người dùng (không được để trống - duy nhất, không được trùng nhau. Nếu trùng sẽ báo lỗi đã tồn tại).
3	Mật khẩu		Mật khẩu đăng nhập vào ứng dụng (không được để trống, bắt buộc mật khẩu có ít nhất 6 ký tự).
5	Email		Email người dùng
7	Ảnh đại diện		Ảnh đại diện người dùng
8	Ngày tạo		Ngày tài khoản được tạo lần đầu
9	Loại tài khoản		Nomal hoặc admin

PHÂN LOẠI TRUYỆN			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã loại truyện		Mã phân loại truyện tự sinh ngẫu nhiên khi có phân loại truyện mới. (Bắt buộc).
2	Loại truyện		Loại món ăn

TRUYỆN			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã truyện		Mã truyện tự sinh ngẫu nhiên khi có thể loại mới. (Bắt buộc – duy nhất).
2	Tên truyện		Tên của truyện
3	Tác giả		Tác giả (Không được để trống).
4	Ảnh đại diện		Ảnh đại diện mô tả truyện (không được để trống).
5	Mô tả		Mô tả truyện
6	Tình trạng		Ngày tháng cập nhật truyện
7	Lượt thích		Lượt thích truyện
8	Lượt đọc		Lượt đọc truyện tự tăng khi người dùng đọc truyện được 30s
9	Ngày đăng truyện		Ngày truyện được đăng lần đầu
10	Ngày cập nhật		Ngày truyện được cập nhật lần cuối
11	Thể loại		Thể loại của truyện

THÍCH TRUYỆN			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã truyện		Mã của truyện (không được để trống - duy nhất).
2	Mã người dùng		Mã của người dùng (không được để trống - duy nhất).
2	Trạng thái thích		True là thích, false là chưa thích

CHƯƠNG TRUYỆN			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã chương		Mã chương tăng dần khi có chương mới. (Bắt buộc).
2	Ngày tạo		Ngày chương được tạo lần đầu
3	Ảnh của chương		Số lượng ảnh của 1 chương

BÌNH LUẬN			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã người dùng		Mã người dùng (Bắt buộc).
2	Thời gian bình luận		Thời gian bình luận
3	Nội dung		Nội dung của bình luận
4	Tên người dùng		Tên người dùng.
5	Ảnh đại diện		Ảnh đại diện
6	Thời gian bình luận		Thời gian bình luận

Lịch sử			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã người dùng		Mã người dùng (Bắt buộc).
2	Mã truyện		Mã truyện
3	Lần đọc cuối		Lần đọc cuối (Chương).
4	Tập mới nhất		Tập mới nhất
5	Ảnh bìa		Ảnh bìa của truyện
6	Trạng thái		Đang cập nhật/ Đã hoàn thành
7	Tên truyện		Tiêu đề của truyện

Trạng thái lưu truyện			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã truyện		Mã của truyện (không được để trống - duy nhất).
2	Mã người dùng		Mã của người dùng (không được để trống - duy nhất).
2	Trạng thái lưu		True là lưu, false là chưa lưu

Bộ sưu tập			
STT	Thuộc tính	Ràng buộc	Mô tả
1	Mã người dùng		Mã người dùng (Bắt buộc).
2	Mã truyện		Mã truyện
3	Tác giả		Tác giả của truyện
4	Mô tả		Mô tả của truyện
5	Thể loại		Thể loại của truyện
6	Trạng thái		Đang cập nhật/ Đã hoàn thành
7	Tên truyện		Tiêu đề của truyện
8	Ảnh bìa		Ảnh bìa của truyện
9	Lượt đọc		Lượt đọc của truyện
10	Lượt thích		Lượt thích của truyện

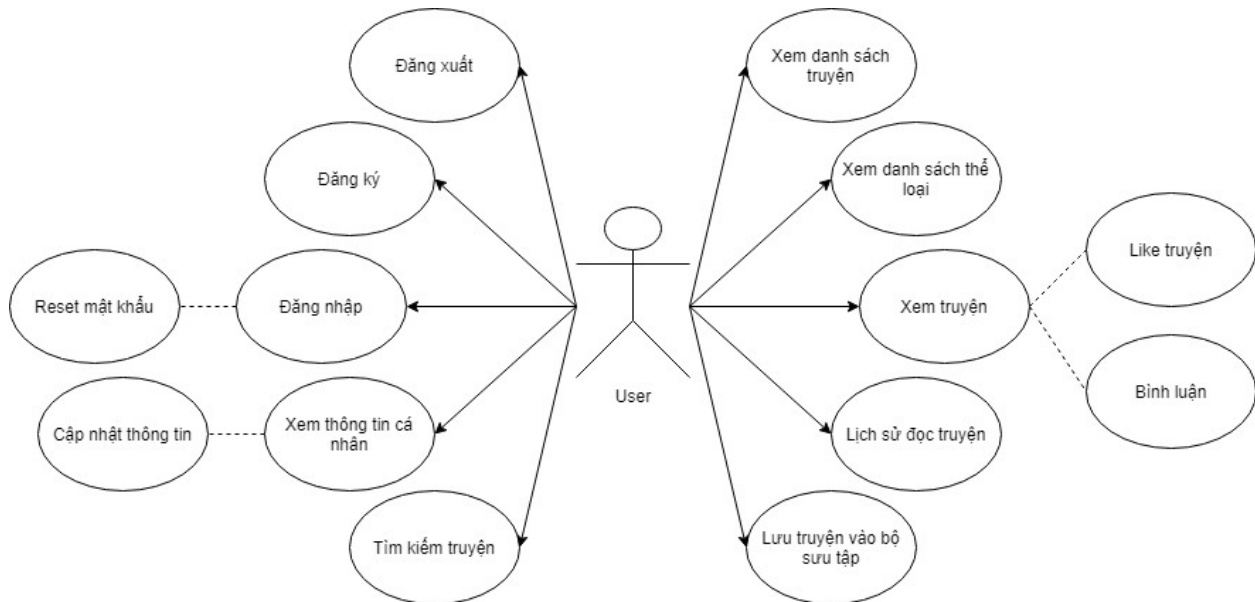
3 Phân tích hiện trạng

Hiện nay, nhu cầu giải trí của giới trẻ ngày một gia tăng, bên cạnh nhiều hình thức giải trí như xem phim, chơi game, nghe nhạc,... một số bạn trẻ còn tỏ ra khá thích thú với việc đọc truyện tranh. Nhận thấy nhu cầu này, đồng thời nhận thấy sự phổ biến cũng như phát triển của các thiết bị di động, chúng tôi quyết định xây dựng một ứng dụng đọc truyện tranh trên nền tảng di động android. Với tiêu chí đơn giản nhưng hiện đại, ứng dụng hướng tới sự tiện dụng cũng như trải nghiệm của người dùng.

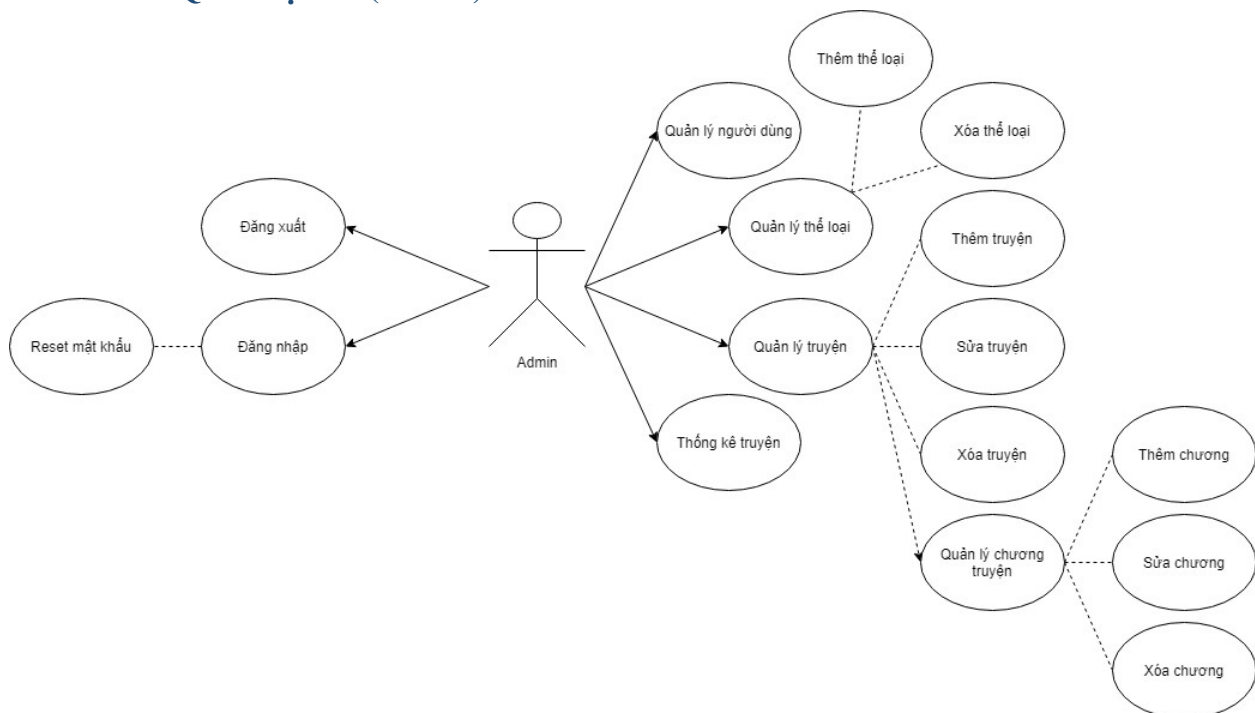
4 Thiết kế hệ thống

4.1 Sơ đồ use case

4.1.1 Người dùng (User)

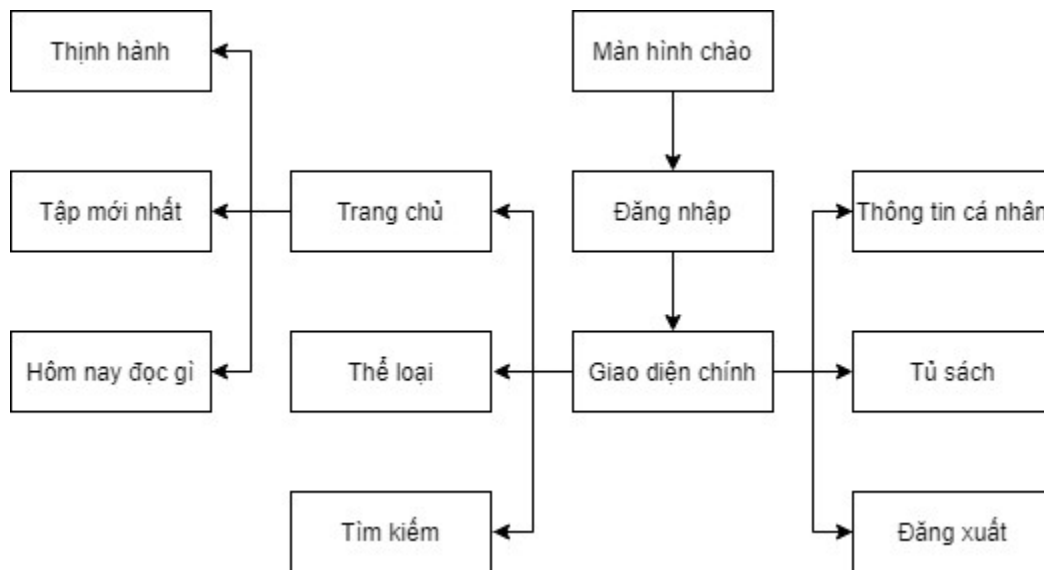


4.1.2 Quản trị viên (Admin)

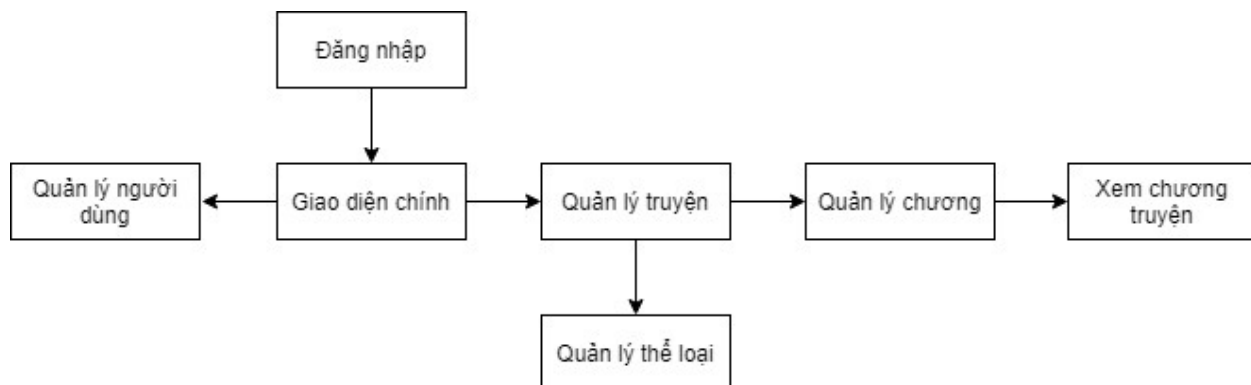


4.2 Sơ đồ giao diện

4.2.1 Người dung (User)

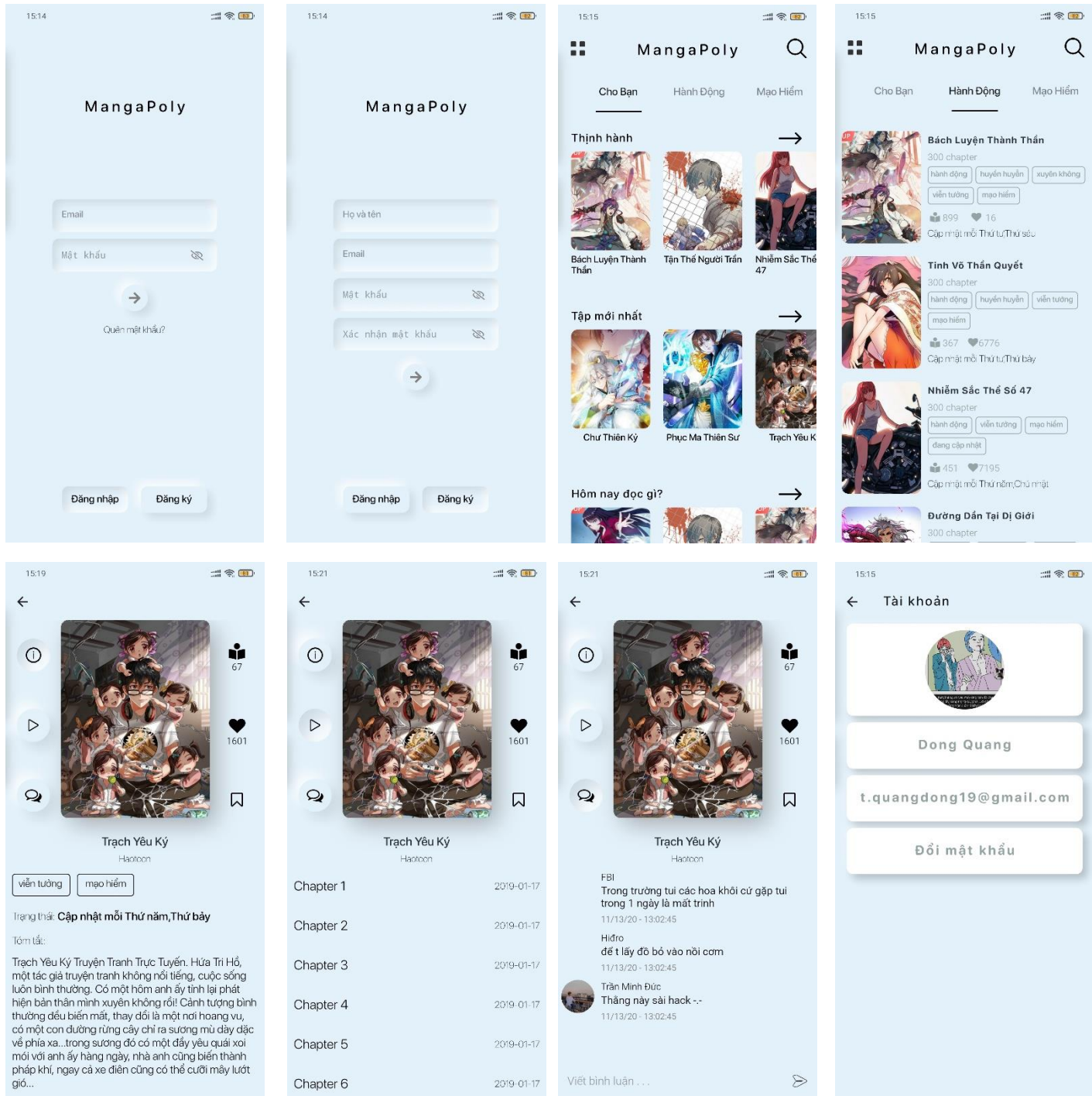


4.2.2 Quản trị viên (Admin)

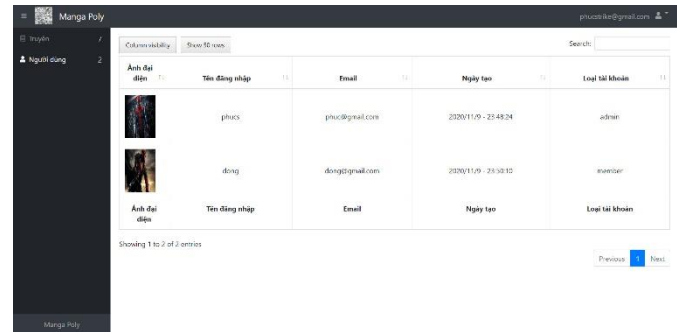
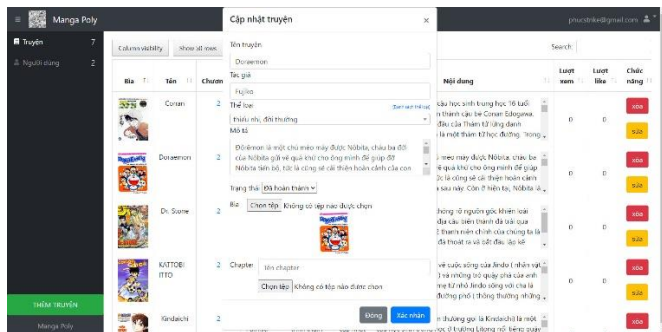
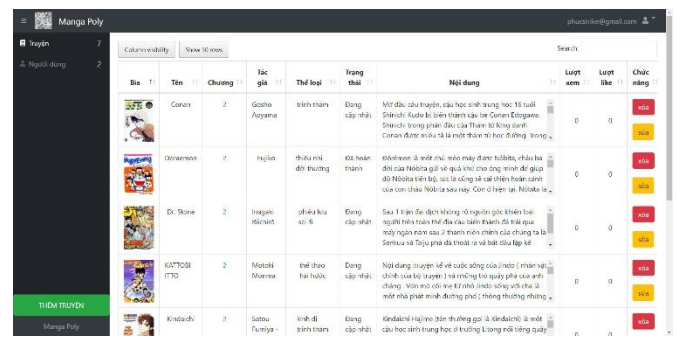
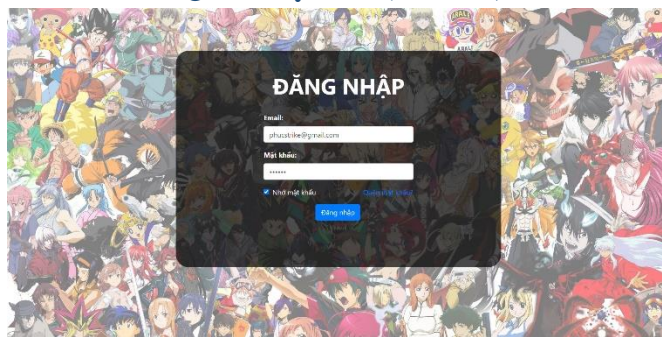


4.3 Giao diện ứng dụng

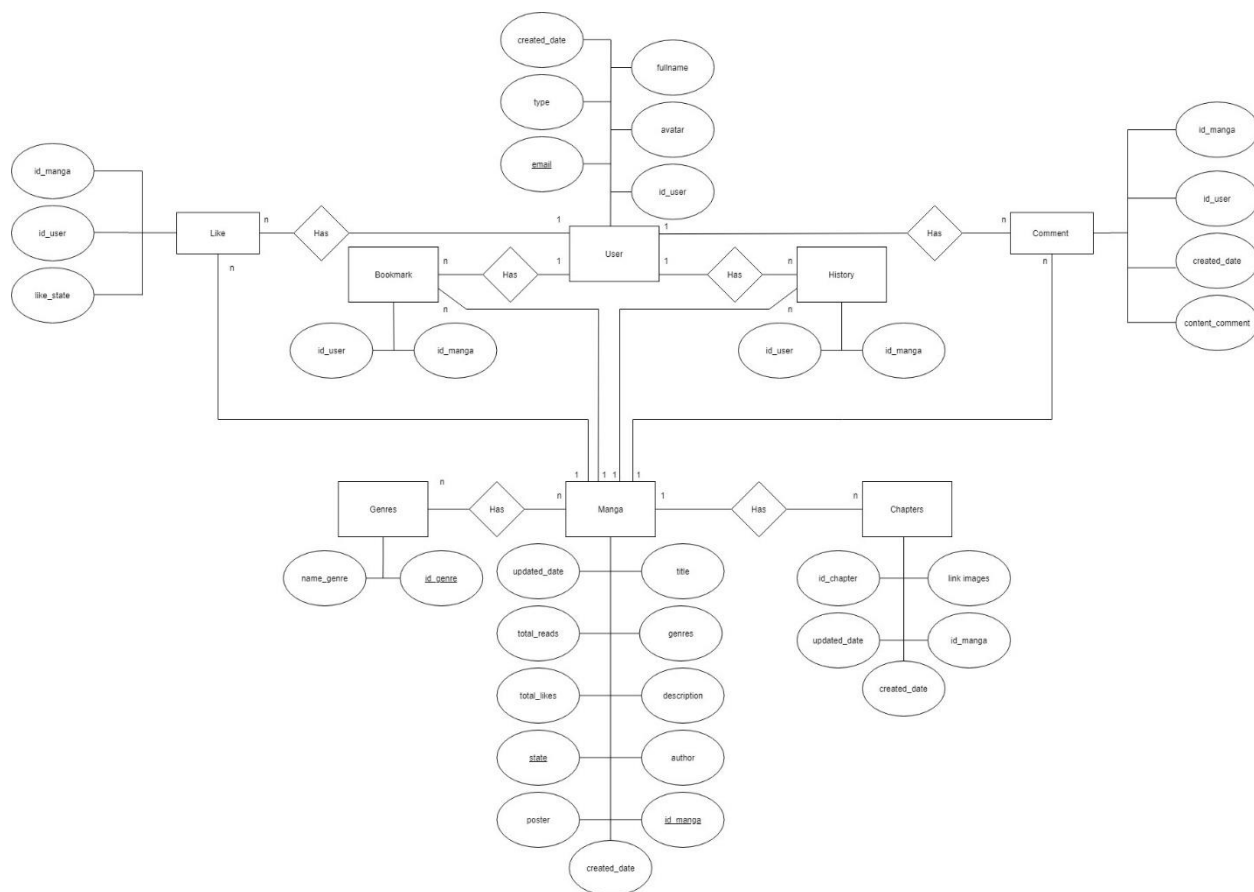
4.3.1 Người dùng (User)



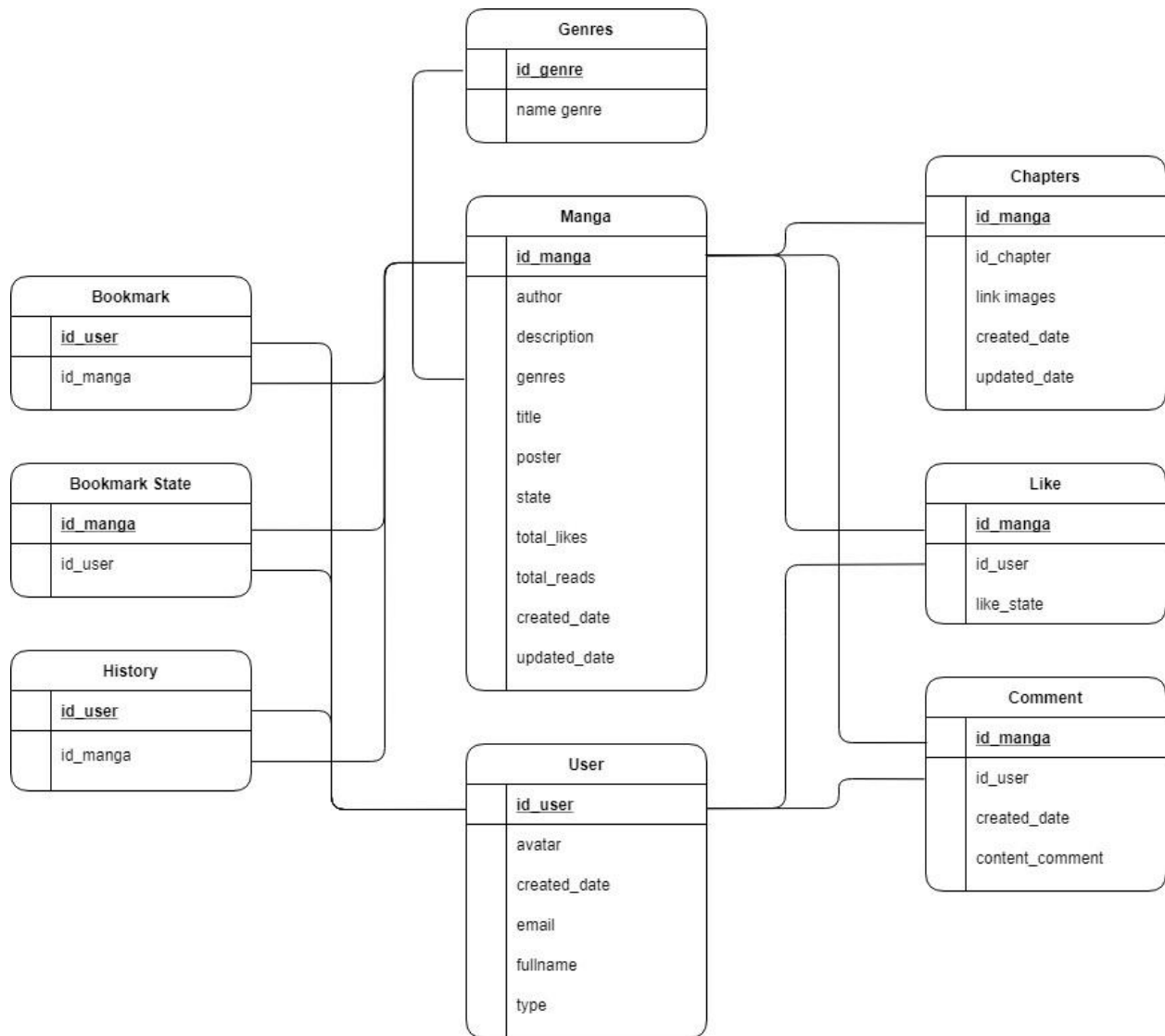
4.3.2 Quản trị viên (Admin)



4.4 Sơ đồ ERD



4.5 Sơ đồ tổ chức thực thể



5 Thực hiện dự án

5.1 Phân công dự án

- Trưởng nhóm Trịnh Quang Đồng:
 - Điều hướng các màn hình (User).
 - Bố cục giao diện (User).
 - Xem danh sách truyện (User).
 - Xem thông tin truyện (User).
 - Đọc truyện (User).
 - Bình luận truyện (User).
 - Thích truyện (User).
 - Lưu truyện vào bộ sưu tập (User).
 - Lưu lịch sử đọc truyện (User).
 - Tăng lượt đọc khi đọc truyện (User).
- Thành viên Ngô Trọng Phúc:
 - Thay đổi ảnh đại diện (User).
 - Cập nhật thông tin cá nhân (User).
 - Thêm, xóa, sửa truyện (Admin).
 - Thêm, xóa, sửa chương của truyện (Admin).
 - Hiệu ứng, trang trí (Admin).
- Thành viên Phạm Hoàng Huy:
 - Đăng ký tài khoản (User).
 - Đăng nhập tài khoản (User).
 - Đổi mật khẩu (User).
 - Quên mật khẩu (User).
 - Tìm kiếm truyện (User).
 - Sắp xếp ngẫu nhiên truyện (User).
 - Sắp xếp truyện theo lượt đọc (User).
 - Sắp xếp truyện theo lượt yêu thích (User).

- Thành viên Trần Công Khanh:
 - Bố cục giao diện (Admin).
 - Xem danh sách truyện (Admin).
 - Tìm kiếm truyện (Admin).
 - Thêm, xóa thể loại (Admin).
 - Xem danh sách thể loại (Admin).
 - Sắp xếp truyện (Admin).
 - Tính năng darkmode (Admin).
- Thành viên Đặng Lê Duy Nghĩa:
 - Đăng nhập (Admin).
 - Lưu mật khẩu (Admin).
 - Quên mật khẩu (Admin).
 - Xem danh sách tài khoản người dùng và quản trị viên (Admin).
 - Xem danh sách chương truyện (Admin).
 - Xem chương truyện (Admin).

5.2 Tổ chức dự án

- Quản lý source code bằng GitHub

5.3 Mã nguồn

5.3.1 Front-end

- Phát triển ứng dụng trên nền tảng React Native
- Công cụ phát triển Visual Studio Code
- Ngôn ngữ: Javascript

5.3.2 Back-end

- Cơ sở dữ liệu: Firebase
- Ngôn ngữ: HTML, CSS, Javascript.

5.4 Thư viện sử dụng

- UI (material, design, circle image view, glide, ...)

- Firebase (Firebase Real-time Database, Authentication)
- Datatable.

6 Viết mã cho ứng dụng

6.1 Ứng dụng

6.1.1 Màn hình đăng nhập

```
19 const SignIn = ({navigation}) => {
20   const [hidePass, setHidePass] = useState(true);
21   const [err, setErr] = useState(null);
22   const [showLoading, setShowLoading] = useState(true);
23   const [email, setEmail] = useState('');
24   const [password, setPassword] = useState('');
25
26   const genreRef = firebaseApp.database().ref('Genres');
27
28   var genreList = [];
29
30   const getGenres = async () => {
31     genreRef.once('value', (snapshot) => {
32       snapshot.forEach((child) => {
33         const key = child.key;
34         const genres = child.val();
35         genreList.push({key, genres});
36       });
37     });
38     await firebaseApp.auth().onAuthStateChanged((user) => {
39       if (user) {
40         console.log(user.email);
41         navigation.replace('Home', {data: genreList});
42       } else {
43         setShowLoading(false);
44       }
45     });
46   };
```

```
48   useEffect(() => {
49     |   getGenres();
50   }, []);
51
52   const signIn = () => {
53     |   Keyboard.dismiss();
54     |   setShowLoading(true);
55     |   firebaseApp
56     |     .auth()
57     |     .signInWithEmailAndPassword(email, password)
58     |     .then(() => {
59       |       setErr('');
60       |       setShowLoading(false);
61       |       navigation.replace('Home', {data: genreList});
62     |     })
63     |     .catch((error) => {
64       |       setShowLoading(false);
65       |       var errorCode = error.code;
66       |       var errorMessage = error.message;
67       |       if (errorCode === 'auth/user-not-found') {
68         |         |   setErr('Tài khoản không tồn tại!');
69       |       } else if (errorCode === 'auth/user-disabled') {
70         |         |   setErr('Tài khoản đã bị vô hiệu hóa!');
71       |       } else if (errorCode === 'auth/invalid-email') {
72         |         |   setErr('Email không hợp lệ!');
73       |       } else if (errorCode === 'auth/wrong-password') {
74         |         |   setErr('Sai mật khẩu!');
75       |       } else {
76         |         |   setErr(errorMessage);
77       |     })
78     |   }
```

```

78         console.log(error);
79     });
80 };
81
82     return (
83         <SafeAreaView style={styles.container}>
84             {/* START - loading screen */}
85             {showLoading ? (
86                 <View style={styles.loading}>
87                     <ActivityIndicator animating={true} size="large" color="black" />
88                 </View>
89             ) : null}
90             {/* END - loading screen */}
91             <Text style={styles.txtError}>{err}</Text>
92             {/* START - TextInput Layout Email */}
93             <TextInputLayout>
94                 <TextInput
95                     style={styles.txtInputEmail}
96                     autoCapitalize="none"
97                     placeholder="Email"
98                     autoCompleteType="email"
99                     keyboardType="email-address"
100                     contentType="emailAddress"
101                     value={email.trim()}
102                     onChangeText={(val) => setEmail(val.trim())}
103                 />
104             </TextInputLayout>
105             {/* END - TextInput Layout Email */}

```

```
107      {/ * START - TextInput Layout Password */}
108      <TextInputLayout type={true}>
109          <TextInput
110              style={styles.txtInputPassword}
111              placeholder="Mật khẩu"
112              autoCapitalize="none"
113              autoCompleteType="password"
114              contentType="password"
115              secureTextEntry={hidePass ? true : false}
116              value={password.trim()}
117              onChangeText={(val) => setPassword(val.trim())}
118          />
119          <Icon
120              name={hidePass ? 'eye-slash' : 'eye'}
121              size={15}
122              color="grey"
123              onPress={() => setHidePass(!hidePass)}
124          />
125      </TextInputLayout>
126      {/ * END - TextInput Layout Password */}
127
128      {/ * START - Button SignIn */}
129      <TouchableOpacity
130          onPress={() => {
131              signIn();
132          }}>
133          <Neomorph style={styles.btn}>
134              <Icon name="arrow-right" size={20} color="grey" />
135          </Neomorph>
136      </TouchableOpacity>
```

```
137     {/* END - Button SignIn */}
138
139     {/* START - ForgotPassword */}
140     <TouchableOpacity
141     onPress={() =>
142         navigation.navigate('ForgotPassword', {title: 'Quên mật khẩu'})
143     }>
144     <Text style={styles.txtForgotPass}>Quên mật khẩu?</Text>
145     </TouchableOpacity>
146     {/* END - ForgotPassword */}
147 </SafeAreaView>
148 );
149 };
150
151 export default SignIn;
```

6.1.2 Màn hình đăng ký

```
19  const SignUp = ({navigation}) => {
20    const [isShowLoading, setIsShowLoading] = useState(false);
21    const [hidePass, setHidePass] = useState(true);
22    const [err, setErr] = useState(null);
23    const [fullName, setFullName] = useState('');
24    const [email, setEmail] = useState('');
25    const [password, setPassword] = useState('');
26    const [rePassword, setRePassword] = useState('');
27
28    const signUp = () => {
29      //ấn bàn phím.
30      Keyboard.dismiss();
31      //show loading screen
32      setIsShowLoading(true);
33
34      if (password.trim() !== rePassword.trim()) {
35        setErr('Mật khẩu không trùng khớp!');
36        setIsShowLoading(false);
37      } else if (fullName.trim().length < 3) {
38        setErr('Họ và tên trống!');
39        setIsShowLoading(false);
40      } else {
41        //push user data lên firebase.
42        firebaseApp
43          .auth()
44          .createUserWithEmailAndPassword(email.trim(), password.trim())
45          .catch((error) => {
46            setIsShowLoading(false);
47            const errorCode = error.code;
48            const errorMessage = error.message;
```

```
49         if (errorCode === 'auth/weak-password') {
50             |   setErr('Mật khẩu ngắn dưới 6 kí tự!');
51         } else if (errorCode === 'auth/email-already-in-use') {
52             |   setErr('Email đã tồn tại!');
53         } else if (errorCode === 'auth/invalid-email') {
54             |   setErr('Email không hợp lệ!');
55         } else {
56             |   setErr(errorMessage);
57         }
58         console.log(error);
59     });
60
61     //lắng nghe nếu người dùng đăng ký thành công.
62     //Nên dùng 'return' để trả về kết quả cuối để tránh rò rỉ ram.
63     return firebaseApp.auth().onAuthStateChanged(async (user) => {
64         |   setIsShowLoading(true);
65         |   if (user) {
66             |       await firebaseApp.database().ref('Users').child(user.uid).set({
67                 |       id: user.uid,
68                 |       ava: '',
69                 |       email: user.email.trim(),
70                 |       type: 'normal',
71                 |       fullName: fullName.trim(),
72                 |       created: Date.now(),
73             |   });
74
75             |   user.updateProfile({
76                 |   displayName: fullName.trim(),
77             |   });
78         } else {
```



```

79         setErr('');
80     }
81     });
82 }
83 };
84
85 return (
86     <SafeAreaView style={styles.container}>
87         {/* START - loading screen */}
88         {isShowLoading ? (
89             <View style={styles.loading}>
90                 <ActivityIndicator animating={true} size="large" color="black" />
91             </View>
92         ) : null}
93         {/* END - loading screen */}
94
95         <Text style={styles.txtError}>{err}</Text>
96
97         {/* START - TextInput Layout Full name */}
98         <TextInputLayout>
99             <TextInput
100                 style={styles.txtInput}
101                 placeholder="Họ và tên"
102                 autoCompleteType="name"
103                 autoCapitalize="words"
104                 keyboardType="name-phone-pad"
105                 contentType="name"
106                 value={fullName}
107                 onChangeText={({val}) => setFullName(val)}
108                 />

```

```
109     </TextInputLayout>
110     { /* END - TextInput Layout Full name */ }
111
112     { /* START - TextInput Layout Email */ }
113     <TextInputLayout>
114     <TextInput
115         style={styles.txtInput}
116         placeholder="Email"
117         autoCompleteType="email"
118         autoCapitalize="none"
119         keyboardType="email-address"
120         textContentType="emailAddress"
121         value={email}
122         onChangeText={(val) => setEmail(val.trim())}
123     />
124 </TextInputLayout>
125 { /* END - TextInput Layout Email */ }
126
127 { /* START - TextInput Layout password */ }
128 <TextInputLayout type={true}>
129     <TextInput
130         style={styles.txtInputPass}
131         placeholder="Mật khẩu"
132         autoCompleteType="password"
133         textContentType="password"
134         autoCapitalize="none"
135         secureTextEntry={hidePass ? true : false}
136         value={password}
137         onChangeText={(val) => setPassword(val.trim())}
138     />
```

```

139         <Icon
140             name={hidePass ? 'eye-slash' : 'eye'}
141             size={15}
142             color="grey"
143             onPress={() => setHidePass(!hidePass)}
144         />
145     </TextInputLayout>
146     { /* END - TextInput Layout password */ }
147
148     { /* START - TextInput Layout confirm password */ }
149     <TextInputLayout type={true}>
150         <TextInput
151             style={styles.txtInputPass}
152             placeholder="Xác nhận mật khẩu"
153             autoCompleteType="password"
154             textContentType="password"
155             autoCapitalize="none"
156             secureTextEntry={hidePass ? true : false}
157             value={rePassword}
158             onChangeText={(val) => setRePassword(val.trim())}
159         />
160         <Icon
161             name={hidePass ? 'eye-slash' : 'eye'}
162             size={15}
163             color="grey"
164             onPress={() => setHidePass(!hidePass)}
165         />
166     </TextInputLayout>
167     { /* END - TextInput Layout confirm password */ }
168
    
```

```
169      {/* START - TextInput Layout button sign up */}  
170      <TouchableOpacity onPress={() => signUp()}>  
171      <Neomorph style={styles.btnSignUp}>  
172        <Icon name="arrow-right" size={20} color="grey" />  
173      </Neomorph>  
174    </TouchableOpacity>  
175    {/* END - TextInput Layout button sign up */}  
176  </SafeAreaView>  
177  );  
178  };  
179  
180  export default SignUp;
```

6.1.3 Màn hình chính

```
18  const Item = (props) => {
19    const [data, setData] = useState();
20    const {title} = props;
21
22    const db = firebaseApp.database().ref('Mangas');
23
24    const a = title === 'Thịnh hành' ? 'totalReads' : b;
25    const b = title === 'Tập mới nhất' ? 'updated' : 'id';
26
27    useEffect(() => {
28      db.orderByChild(a || b).once('value', (snapshot) => {
29        var li = [];
30
31        snapshot.forEach((child) => {
32          const data1 = child.val();
33          li.push(data1);
34        });
35        li.reverse();
36
37        setData(li);
38      });
39    }, [a, b]);
40    return (
41      <SafeAreaView style={styles.container}>
42        {/* START - tiêu đề */}
43        <TouchableOpacity
44          onPress={() => {
45            props.navigation.navigate('ListManga', {title, data});
46          }}

```

```

47         style={styles.titleContainer}>
48         <Text style={styles.titleText}>{title}</Text>
49         <FastImage
50         |   source={arrow_right}
51         |   style={{width: wp(40), height: hp(40)}}
52         />
53     </TouchableOpacity>
54     {/* END - tiêu đề */}
55
56     {/* nội dung - show ra list ảnh*/}
57     <FlatList
58     |   data={data} //dữ liệu - dữ liệu này sẽ dùng để gán vào giao diện
59     |   horizontal //set cho list view nằm ngang
60     |   showsHorizontalScrollIndicator={false} // ẩn thanh cuộn
61     |   renderItem={({item}) => (
62     |       <TouchableOpacity
63     |       |   key={item.id}
64     |       |   style={styles.contentContainer}
65     |       |   onPress={() => {
66     |       |       props.navigation.navigate('DetailItem', {
67     |       |       |   mangaID: item.id,
68     |       |       |   });
69     |       |       console.log(item.id);
70     |       |   }}>
71     |       <FastImage
72     |       |   source={{uri: item.poster}}
73     |       |   resizeMode={FastImage.resizeMode.cover}
74     |       |   style={styles.contentImg}
75     |       />
76     |       <View style={styles.contentTextView}>

```

```
77 |         <Text style={styles.contentText}>{item.name}</Text>
78 |       </View>
79 |     </TouchableOpacity>
80 |   )} // 'item' là dữ liệu,
81 |   //hàm renderItem là gán giao diện cho item
82 |   keyExtractor={({item}) => item.id} //mỗi item trong list phải
83 |   />
84 | </SafeAreaView>
85 | );
86 | };
87 |
88 | export default Item;
89 |
90 | const styles = StyleSheet.create({
```

6.1.4 Màn hình chi tiết

```
26 const DetailItem = (props) => {
27   const [isPressed, setIsPressed] = useState(null);
28   const [like, setLike] = useState(false);
29   const [bookmark, setBookmark] = useState(false);
30   const [countLikes, setCountLikes] = useState(mangaLikes);
31   const [uID, setUID] = useState();
32
33   const [mangaPoster, setMangaPoster] = useState(null);
34   const [mangaName, setMangaName] = useState('');
35   const [mangaAuthor, setMangaAuthor] = useState('');
36   const [mangaState, setMangaState] = useState('');
37   const [mangaDes, setMangaDes] = useState('');
38   const [mangaLikes, setMangaLikes] = useState('');
39   const [mangaReads, setMangaReads] = useState('');
40   const [mangaGenre, setMangaGenre] = useState('');
41
42   const mangaRef = firebaseApp.database().ref('Mangas');
43   const likeStateRef = firebaseApp.database().ref('LikeState');
44   const bookmarkStateRef = firebaseApp.database().ref('BookmarkState');
45   const bookmarkRef = firebaseApp.database().ref('Bookmark');
46
47   const user = firebaseApp.auth().currentUser;
48
49   useEffect(() => {
50     getLikeState();
51     getBookmarkState();
52   });
```



```
55   useEffect(() => {
56     setIsPressed('Description');
57     getMangas();
58   }, [getMangas]);
59
60   const {mangaID} = props.route.params;
61
62   const getLikeState = () => {
63     if (user) {
64       setUID(user.uid);
65
66       // Lấy trạng thái đã like hay chưa để đưa lên UI
67       likeStateRef
68         .child(mangaID)
69         .child(user.uid)
70         .on('value', (snapshot) => {
71           const isLiked = snapshot.val();
72           setLike(isLiked);
73         });
74
75       // Ở thông tin truyện, lấy lượt like hiện có của truyện
76       mangaRef
77         .child(mangaID)
78         .child('totalLikes')
79         .on('value', (snapshot) => {
80           const Likes = snapshot.val();
81           setCountLikes(Likes);
82         });
83     }
84   };
```

```
86   const getBookmarkState = () => {
87     bookmarkStateRef
88       .child(mangaID)
89       .child(user.uid)
90       .on('value', (snapshot) => {
91         const isBookmark = snapshot.val();
92         setBookmark(isBookmark);
93       });
94
95     setTimeout(() => {
96       bookmark
97         ? bookmarkRef.child(user.uid).child(mangaID).set({
98           name: mangaName,
99           poster: mangaPoster,
100          author: mangaAuthor,
101          description: mangaDes,
102          state: mangaState,
103          genres: mangaGenre,
104          totalLikes: mangaLikes,
105          totalReads: mangaReads,
106          id: mangaID,
107        })
108        : bookmarkRef.child(user.uid).child(mangaID).remove();
109     }, 3000);
110   };
111
112   const getMangas = useCallback(() => {
113     console.log(mangaID);
114     mangaRef.child(mangaID).once('value', (snapshot) => {
115       setMangaName(snapshot.child('name').val());
```

```
116         setMangaPoster(snapshot.child('poster').val());
117         setMangaAuthor(snapshot.child('author').val());
118         setMangaDes(snapshot.child('description').val());
119         setMangaState(snapshot.child('state').val());
120         setMangaGenre(snapshot.child('genres').val());
121         setMangaLikes(snapshot.child('totalLikes').val());
122         setMangaReads(snapshot.child('totalReads').val());
123     });
124 }, [mangaID, mangaRef]);
125
126 const pressLike = () => {
127     likeStateRef.child(mangaID).child(uID).set(!like);
128
129     //Nhờ vào 'lượt like hiện có' đã lấy ở trên để tăng thêm 1 nếu true, fal
130     mangaRef
131         .child(mangaID)
132         .update(
133             like ? {totalLikes: countLikes - 1} : {totalLikes: countLikes + 1},
134         );
135 };
136
137 const pressBookmark = () => {
138     bookmarkStateRef.child(mangaID).child(uID).set(!bookmark);
139 };
140
141 //tại sao phải chạy cái này mà ko set cũng ở trên useState?
142 //Bởi vì set cũng làm giao diện của button Description bị lỗi
143 const descriptionJSX =
144     isPressed === 'Description' ? (
145         <Description state={mangaState} des={mangaDes} genres={mangaGenre} />
```

```
146 |   ) : null;
147 |   const chaptersJSX =
148 |     isPressed === 'Chapter' ? (
149 |       <Chapters
150 |         id={mangaID}
151 |         navigation={props.navigation}
152 |         state={mangaState}
153 |         title={mangaName}
154 |         poster={mangaPoster}
155 |       />
156 |     ) : null;
157 |   const commentsJSX =
158 |     isPressed === 'Comment' ? (
159 |       <Comments id={mangaID} navigation={props.navigation} />
160 |     ) : null;
161 |   return (
162 |     <SafeAreaView style={styles.container}>
163 |       { /* START - Title */ }
164 |       <View style={styles.titleContainer}>
165 |         { /* START - Image view */ }
166 |         <View style={styles.titleViewImg}>
167 |           { /* START - viewLeft */ }
168 |           <View style={styles.titleViewLeftRight}>
169 |             { /* START - btnDescription */ }
170 |             <Pressable
171 |               onPressIn={() => {
172 |                 setIsPressed('Description');
173 |               }}>
174 |               <Neomorph
175 |                 inner={isPressed === 'Description' ? true : false}
```

```
176         swapShadows // <- change zIndex of each shadow
177         style={styles.titleNeomorph}>
178         <Image
179         |         source={ic_des}
180         |         style={styles.titleViewLeftRightIconDes}
181         />
182     </Neomorph>
183 </Pressable>
184 {/* START - btnChapters */}
185 <Pressable
186 |     onPressIn={() => {
187 |         setIsPressed('Chapter');
188 |     }}>
189     <Neomorph
190     |         inner={isPressed === 'Chapter' ? true : false}
191     |         swapShadows
192     |         style={styles.titleNeomorph}>
193         <Ionicons name="ios-play-outline" size={25} />
194     </Neomorph>
195 </Pressable>
196 {/* START - btnComments */}
197 <Pressable
198 |     onPressIn={() => {
199 |         setIsPressed('Comment');
200 |     }}>
201     <Neomorph
202     |         inner={isPressed === 'Comment' ? true : false}
203     |         swapShadows
204     |         style={styles.titleNeomorph}>
205         <FontAwesome name="comments-o" size={25} />
```

```

206 |         </Neomorph>
207 |     </Pressable>
208 | </View>
209 | /*END - viewLeft*/}
210 | <Shadow useArt style={styles.titleShadow}>
211 |     <FastImage
212 |         source={{uri: mangaPoster || mangaPoster}}
213 |         resizeMode={FastImage.resizeMode.contain}
214 |         style={styles.titleImg}
215 |     />
216 | </Shadow>
217 | /*START - viewRight*/}
218 | <View style={styles.titleViewLeftRight}>
219 |     <View style={styles.titleViewLeftRightItem}>
220 |         <FontAwesome5 name="book-reader" size={25} />
221 |         <Text>{mangaReads}</Text>
222 |     </View>
223 |     <TouchableOpacity
224 |         onPress={() => {
225 |             pressLike();
226 |         }}
227 |         style={styles.titleViewLeftRightItem}>
228 |         <FontAwesome name={!like ? 'heart-o' : 'heart'} size={25} />
229 |         <Text>{countLikes}</Text>
230 |     </TouchableOpacity>
231 |     <TouchableOpacity
232 |         onPress={() => {
233 |             pressBookmark();
234 |         }}>
235 |         <View style={styles.titleViewLeftRightItem}>
    
```

```

236         <FontAwesome
237             name={!bookmark ? 'bookmark-o' : 'bookmark'}
238             size={25}
239         />
240     </View>
241 </TouchableOpacity>
242 </View>
243 { /*END - viewRight*/}
244 </View>
245 { /* END - Img*/}
246 { /* START - name */}
247 <View style={styles.titleViewMangaName}>
248     <Text style={styles.titleMangaName}>{mangaName}</Text>
249     <Text style={styles.titleAuthor}>{mangaAuthor}</Text>
250 </View>
251
252 { /* END - name */}
253 </View>
254 { /* END - Title */}
255 { /* START - Content */}
256 <View style={styles.contentContainer}>
257     {descriptionJSX}
258     {chaptersJSX}
259     {commentsJSX}
260 </View>
261 { /* END - Content */}
262 </SafeAreaView>
263 );
264 };
265
    
```

6.1.5 Màn hình tìm kiếm

```

14  const InfiniteHits = ({hits, hasMore, refine, navigation}) => (
15    <FlatList
16      data={hits}
17      keyExtractor={({item}) => item.objectID}
18      onEndReached={() => hasMore && refine()}
19      renderItem={({item}) => {
20        const genre = item.genres.split(',');
21        return (
22          // START - Item
23          <TouchableOpacity
24            key={item.id}
25            onPress={() => {
26              navigation.navigate('DetailItem', {
27                mangaID: item.id,
28              });
29            }}
30            style={styles.items}>
31            {/* START - Poster */}
32            <FastImage
33              source={{uri: item.poster}}
34              resizeMode={FastImage.resizeMode.cover}
35              style={{
36                height: hp(170),
37                width: wp(120),
38                borderRadius: wp(10),
39              }}
40            />
41            {/* END - Poster */}
42            {/* START - Right content */}
43            <View style={styles.rightContent}>
44              {/* START - Title */}

```



```

45 <Highlight attribute="name" hit={item} />
46 {/ * END - Title */}
47 {/ * START - Author */}
48 <Text style={styles.iconTxt}>{item.author}</Text>
49 {/ * END - Author */}
50 {/ * START - Genre */}
51 <View style={styles.genreContainer}>
52   {/ * Lấy từng thể loại trong mảng thể loại ra để gán vào UI */}
53   {genre
54     ? genre.map((e) => (
55       <Text key={e} style={styles.genreItem}>
56         {e}
57       </Text>
58     ))
59     : null}
60 </View>
61 {/ * END - Genre */}
62 {/ * START - Icons */}
63 <View style={styles.iconsContainer}>
64   {/ * START - Icon reads */}
65   <View style={styles.iconView}>
66     <FontAwesome5 name="book-reader" size={15} color="#95a5a6" />
67     <Text style={styles.iconTxt}>{item.totalReads}</Text>
68   </View>
69   {/ * END - Icon reads */}
70   {/ * START - Icon likes */}
71   <View style={styles.iconView}>
72     <FontAwesome5 name="heart" size={15} color="#95a5a6" />
73     <Text style={styles.iconTxt}>{item.totalLikes}</Text>
74   </View>

```

```
75 | | | | | | { /* END - Icon likes */}
76 | | | | | | </View>
77 | | | | | | { /* END - Icons */}
78 | | | | | | { /* START - State */}
79 | | | | | | <View style={styles.stateView}>
80 | | | | | | | <Text style={styles.stateTxt}>{item.state}</Text>
81 | | | | | | </View>
82 | | | | | | { /* END - State */}
83 | | | | | | </View>
84 | | | | | | { /* END - Right content */}
85 | | | | | | </TouchableOpacity>
86 | | | | | | // END - Item
87 | | | | | | );
88 | | | | | | }}
89 | | | | | | />
90 | | | | | | );
91 |
92 | InfiniteHits.propTypes = {
93 |   hits: PropTypes.arrayOf(PropTypes.object).isRequired,
94 |   hasMore: PropTypes.bool.isRequired,
95 |   refine: PropTypes.func.isRequired,
96 | };
97 |
98 | export default connectInfiniteHits(InfiniteHits);
99 |
```

6.2 Quản trị viên

6.2.1 Chức năng đăng nhập

```
27  function signIn() {
28      var email = document.getElementById("email");
29      var password = document.getElementById("password");
30
31      firebase.auth().signInWithEmailAndPassword(email.value, password.value)
32      .then(function (firebaseUser) {
33          // Success
34          //set email into localStorage
35          localStorage.setItem("emailAdminMangaPoly", email.value);
36          //redirect to index
37          window.location.href = "mangas.html";
38          let user = firebase.auth().currentUser;
39          console.log(user.email + " / " + user.metadata.lastSignInTime);
40      })
41      .catch(function (error) {
42          // Error Handling
43          alert("Đăng nhập thất bại!");
44      });
45  }
--
```

6.2.2 Chức năng thêm, sửa truyện và chương

```

33     //add poster
34     async function uploadPoster() {
35         const name = 'poster';
36         const metadata = {
37             contentType: poster.type
38         };
39         const task = posterRef.child(name).put(poster, metadata);
40         await task
41             .then(snapshot => snapshot.ref.getDownloadURL())
42             .then(url => {
43                 urlPoster = url;
44             })
45             .catch(console.error);
46     }
47
48     //add/update chapter
49     async function uploadChapter() {
50         var arrayImgs = []
51         for (i = 0; i < chapterInput.files.length; i++) {
52             const file = document.querySelector("#imgsChapter").files[i];
53             const name = file.name;
54             const metadata = {
55                 contentType: file.type
56             };
57             const task = chapterRef.child(name).put(file, metadata);
58             await task
59                 .then(snapshot => snapshot.ref.getDownloadURL())
60                 .then(url => {
61                     arrayImgs.push(url);
62                 })
63                 .catch(console.error);
64         }
65         firebaseRef.child('chapters').child(id).child(chapterTitle.value.trim()).set(arrayImgs);
66     }
--

```

```
68     //add info
69     async function uploadInfo() {
70         var manga = {
71             id: id,
72             title: title.value.trim(),
73             author: author.value.trim(),
74             category: stringSelectedCate,
75             description: description.value.trim(),
76             state: state,
77             poster: urlPoster,
78             totalViews: 0,
79             totalLikes: 0,
80             createdAt: dateTime,
81             updatedAt: ""
82         }
83         // for(var i = 0; i < arrSelectedCate.length; i++){
84         //     manga[arrSelectedCate[i]] = arrSelectedCate[i];
85         // }
86         firebaseRef.child('mangas').child(id).set(manga);
87     }
88
89     //update info
90     async function updateInfo() {
91         var manga = {
92             title: title.value.trim(),
93             author: author.value.trim(),
94             category: stringSelectedCate,
95             description: description.value.trim(),
96             state: state,
97             poster: urlPoster,
98             updatedAt: dateTime
99         }
100         firebaseRef.child('mangas').child(id).update(manga);
101     }
```

6.2.3 Chức năng tải truyện từ firebase

```

266 //load all mangas
267 var totalManga;
268 var totalUser;
269 var listManga = [];
270 await firebase.database().ref('mangas').once('value', function (snapshot) {
271   snapshot.forEach(function (childSnapshot) {
272     var childKey = childSnapshot.key;
273     var childData = childSnapshot.val();
274     //count total manga
275     totalManga = snapshot.numChildren();
276
277     firebase.database().ref('chapters/' + childKey).once('value', function (snapshot) {
278       var totalChapters = snapshot.numChildren();
279       var listCategory = childData.category.split(",");
280       var strCategory=listCategory[0]+'<br>';
281       for (i = 1; i < listCategory.length; i++) {
282         strCategory += listCategory[i]+'<br>';
283       }
284       console.log(strCategory);
285
286       listManga.push([
287         'img width="70px" height="100px" src="'+ childData.poster + '"',
288         'div style="height:100px;display:flex;justify-content:center;overflow:auto;">' + childData.title + '</div>',
289         'div style="height:100px;display:flex;justify-content:center;"><a href="#">void(0)" onclick="showModalListChapter(' + "" + childKey + "" + "', " + "" +
290         'div style="height:100px;display:flex;justify-content:center;overflow:auto;">' + childData.author + '</div>',
291         'div style="width:100px;height:100px;display:flex;justify-content:center;overflow:auto;">' + strCategory + '</div>',
292         'div style="height:100px;display:flex;justify-content:center;">' + childData.state + '</div>',
293         'div style="width:400px;height:100px;display:flex;justify-content:center;overflow:auto;">' + childData.description + '</div>',
294         'div style="height:100px;display:flex;justify-content:center;align-items:center;">' + childData.totalViews + '</div>',
295         'div style="height:100px;display:flex;justify-content:center;align-items:center;">' + childData.totalLikes + '</div>',
296         'button class="btn btn-danger" style="margin-bottom:20px" onclick="deleteManga(' + "" + childKey + "" + ')">xóa</button>' +
297         'button class="btn btn-warning" onclick="showModalUpdateManga(' + "" + childKey + "" + ')">sửa</button>',
298         'div style="height:100px;display:flex;justify-content:center;align-items:center;">' + childData.createdAt + '</div>',
299         'div style="height:100px;display:flex;justify-content:center;align-items:center;">' + childData.updatedAt + '</div>',
300       ]);
301     //do code không đồng bộ và datatable chỉ đăng ký được 1 lần nên phải check nếu đã add xong hết manga vào listManga mới set datatable
302     if (listManga.length == totalManga) {
303       $('#dataTable').DataTable({
304         dom: 'Bfrtip',
305         buttons: ['colvis', 'pageLength'],
306         data: listManga,
307         pageLength: 50,
308       });
309     }
310   });
311 });
312
313

```

7 Kiểm thử

5.5 Kiểm thử đăng ký

Thuộc tính	Thao tác		
	Thêm mới	Cập nhật	Xóa
Email đăng nhập	Không để trống Không trùng		
Mật khẩu	Ít nhất 6 ký tự		
Xác nhận mật khẩu	Trùng khớp mật khẩu trên		

5.6 Kiểm thử đăng nhập

Thuộc tính	Thao tác		
	Thêm mới	Cập nhật	Xóa
Email		Không để trống	
Mật khẩu	Ít nhất 6 ký tự		

5.7 Kiểm thử cập nhật thông tin người dùng

Thuộc tính	Thao tác
	Cập nhật
Tên	Không để trống
Email	Không để trống

5.8 Kiểm thử đổi mật khẩu

Thuộc tính	Thao tác
	Cập nhật
Mật khẩu mới	Không để trống
Xác nhận mật khẩu	Mật khẩu mới và xác nhận mật khẩu phải giống nhau

5.9 Kiểm thử cập nhật thông tin truyện

Thuộc tính	Thao tác
	Cập nhật
Tên truyện	Không để trống
Ảnh đại diện	Không để trống
Tác giả	Không để trống
Thể loại	Không để trống
Mô tả	Không để trống
Trạng thái	Không để trống
Chương	Không để trống

6 Triển khai và sử dụng

6.1 Hướng dẫn và cài đặt

- Cài đặt file APK vào điện thoại.
- APK: https://j2c.cc/manga_poly
- Hệ điều hành Android từ 5.0 trở lên.

6.2 Hướng dẫn sử dụng phần mềm

- Sau khi cài đặt ứng dụng trên điện thoại, mở ứng dụng lên và thực hiện phần đăng ký, điền đầy đủ thông tin sau đó nhấn đăng ký.
- Sau khi đăng ký thành công, màn hình sẽ tự động chuyển vào trang chủ.
- Sau khi vào màn hình trang chủ, có thể nhấn chọn 1 truyện bất kỳ trên màn hình để vào thông tin truyện.
- Sau khi vào thông tin truyện, nếu thấy hợp thì có thể vào chương để đọc truyện.
- Sau đó nhấn tìm hoặc lưu truyện vào bộ sưu tập, menu ở phía trên bên trái màn hình có chứa tủ sách nơi chứa lịch sử đọc truyện, và bộ sưu tập – nơi truyện được lưu.
- Có thể bình luận ở truyện nếu muốn.

7 Kết luận

7.1 Thuận lợi

- Có kiến thức cơ bản về React Native.
- Khả năng giải quyết vấn đề nhanh.
- Hỗ trợ lẫn nhau khi gặp khó khăn.
- Khả năng làm việc nhóm tốt.
- Nhiều framework hỗ trợ tốt.

7.2 Khó khăn

- Tất cả các thành viên hiện đang đi làm nên không có nhiều thời gian tập trung để làm dự án.
- Chuyên môn của mỗi thành viên khác nhau.
- Vấn đề giao tiếp chưa thực sự tốt.

7.3 Nhưng phần chưa làm

- Gửi thông báo cho người dùng.
- Đăng nhập bằng google, facebook.

7.4 Mô hình SWOT của ứng dụng

- ❖ Điểm mạnh:
 - Giao diện hiện đại đơn giản.
 - Dễ dàng sử dụng.
 - Không chứa quảng cáo.
- ❖ Điểm yếu:
 - Quảng bá sản phẩm chưa có.
 - Ứng dụng chưa được tối ưu tốt.
- ❖ Cơ hội:
 - Ứng dụng có tiềm năng.
 - Cộng đồng manga và anime đông đảo.
- ❖ Thách thức
 - Cạnh tranh cao với các ứng dụng đọc truyện khác.

8 Tài liệu tham khảo

- Tài liệu tham khảo: GitHub, Stack Overflow, Google . . .