

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN**

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG ĐỌC TRUYỆN CHỮ TRÊN ANDROID SỬ DỤNG FIREBASE**

**Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng**

**Sinh viên thực hiện:**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	2251061849	Hoàng Đình Nghĩa	26/01/2004		
2	2251061869	Vũ Minh Quang	07/10/2004		
3	2251061895	Nguyễn Tiến Toàn	14/09/2004		
4	2251061927	Bùi Hoàng Vũ	17/12/2004		

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2025**

## LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, các ứng dụng di động ngày càng trở nên phổ biến và đóng vai trò quan trọng trong đời sống hàng ngày. Trong đó, nhu cầu đọc truyện, tiểu thuyết trên điện thoại thông minh cũng ngày càng tăng cao nhờ tính tiện lợi, nhanh chóng và dễ tiếp cận. Việc xây dựng một ứng dụng đọc truyện chữ không chỉ giúp người dùng có thể đọc truyện mọi lúc, mọi nơi mà còn là cơ hội để sinh viên thực hành và áp dụng những kiến thức đã học vào một sản phẩm thực tế.

Bài tập lớn này được thực hiện nhằm mục tiêu xây dựng một ứng dụng đọc truyện chữ trên nền tảng Android, sử dụng cơ sở dữ liệu Realtime Database của Firebase. Qua quá trình thực hiện, em có cơ hội rèn luyện kỹ năng lập trình Android, thiết kế giao diện, xử lý dữ liệu thời gian thực và tư duy phát triển phần mềm theo mô hình hiện đại.

Báo cáo này trình bày đầy đủ quá trình thiết kế, xây dựng và triển khai ứng dụng, cũng như những khó khăn gặp phải và hướng giải quyết trong quá trình thực hiện.

## MỤC LỤC

LỜI NÓI ĐẦU.....	2
MỤC LỤC.....	3
DANH MỤC HÌNH ẢNH.....	5
DANH MỤC BẢNG BIỂU .....	5
DANH MỤC CÁC TỪ VIẾT TẮT .....	6
Chương 1. TỔNG QUAN VỀ ĐỀ TÀI.....	7
1.1. Giới thiệu về đề tài.....	7
1.2. Mục tiêu của đề tài.....	7
1.3. Phạm vi của đề tài.....	8
1.4. Phân chia nhiệm vụ .....	8
Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ.....	9
2.1. Kiến trúc hệ thống.....	9
2.2. Giới thiệu về Công nghệ phát triển .....	9
2.2.1. Ngôn ngữ lập trình: Java .....	9
2.2.2 Android Studio .....	9
2.2.3. Firebase Realtime Database .....	9
Chương 3. XÂY DỰNG ỨNG DỤNG.....	10
3.1. Thiết kế Figma .....	10
3.2. Thiết kế CSDL.....	10
3.3. Giao diện ứng dụng .....	11
3.3.1. Màn hình đăng nhập.....	11
3.3.2. Màn hình đăng ký .....	11
3.3.3. Màn hình quên mật khẩu.....	12

3.3.4. Tài khoản admin .....	13
3.3.5. Tài khoản user.....	20
3.4. Code minh họa các chức năng cốt lõi .....	26
3.4.1. Chức năng đăng ký .....	26
3.4.2. Chức năng đăng nhập .....	28
3.4.3. Chức năng đăng xuất .....	30
3.4.4. Chức năng theo dõi truyện .....	30
3.4.5. Chức năng thêm truyện .....	32
3.4.6. Chức năng cập nhật truyện .....	33
3.4.7. Chức năng xóa truyện .....	34
3.4.8. Chức năng tìm truyện.....	35
3.4.9. Chức năng đổi mật khẩu .....	35
3.4.10. Chức năng quên mật khẩu .....	36
KẾT LUẬN .....	38
1. Kết quả đạt được.....	38
2. Nhược điểm .....	38
3. Hướng phát triển.....	38
TÀI LIỆU THAM KHẢO .....	39

## DANH MỤC HÌNH ẢNH

Ảnh 1: Thiết kế CSDL .....	10
Ảnh 2: Màn hình đăng nhập .....	11
Ảnh 3: Màn hình đăng ký.....	12
Ảnh 4: Màn hình quên mật khẩu .....	13
Ảnh 5: Màn hình giao diện chính (admin) .....	14
Ảnh 6: Màn hình thông tin truyện (admin) .....	15
Ảnh 7: Màn hình nội dung chapter (admin) .....	16
Ảnh 8: Màn hình thêm chapter .....	17
Ảnh 9: Màn hình thêm truyện .....	18
Ảnh 10: Màn hình sửa chapter .....	19
Ảnh 11: Màn hình sửa truyện .....	20
Ảnh 12: Màn hình giao diện chính (user) .....	21
Ảnh 13: Màn hình thông tin truyện (user).....	22
Ảnh 14: Màn hình nội dung chapter (user) .....	23
Ảnh 15: Màn hình danh sách theo dõi .....	24
Ảnh 16: Màn hình đổi mật khẩu .....	25

## DANH MỤC BẢNG BIỂU

Bảng 1: Danh mục các từ viết tắt.....	6
Bảng 2: Phân chia nhiệm vụ .....	8

## DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	IDE	Integrated Development Environment
2		

Bảng 1: Danh mục các từ viết tắt

## Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Giới thiệu về đề tài

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và các thiết bị di động, nhu cầu giải trí của con người ngày càng đa dạng và tiện lợi hơn. Một trong những hình thức giải trí phổ biến hiện nay là **đọc truyện chữ trên thiết bị di động**, thay thế cho việc đọc truyện giấy truyền thống. Người dùng có thể dễ dàng truy cập, lựa chọn và đọc hàng ngàn đầu truyện chỉ với một chiếc điện thoại thông minh có kết nối Internet.

Nhằm đáp ứng nhu cầu đó, đề tài "**Xây dựng ứng dụng đọc truyện chữ trên Android sử dụng Firebase**" được lựa chọn và thực hiện. Ứng dụng giúp người dùng:

- Duyệt danh sách các truyện có sẵn.
- Xem danh sách các chương tương ứng của từng truyện.
- Đọc nội dung chi tiết từng chương theo cách đơn giản và thuận tiện.

Việc sử dụng **Firestore Realtime Database** giúp đồng bộ hóa dữ liệu nhanh chóng và hiệu quả giữa ứng dụng và hệ thống lưu trữ, đồng thời giảm thiểu thời gian xây dựng backend riêng.

Đề tài không chỉ giúp nâng cao khả năng lập trình Android mà còn tạo điều kiện để người thực hiện làm quen với kiến trúc ứng dụng thời gian thực, quản lý dữ liệu trên cloud, từ đó ứng dụng vào các sản phẩm thực tế trong tương lai.

### 1.2. Mục tiêu của đề tài

Mục tiêu của đề tài là xây dựng một ứng dụng đọc truyện chữ đơn giản trên nền tảng **Android**, sử dụng **Firestore Realtime Database** làm hệ thống lưu trữ dữ liệu. Cụ thể:

- Thiết kế giao diện thân thiện, dễ sử dụng cho người dùng.
- Hiển thị danh sách các truyện lấy từ cơ sở dữ liệu Firestore.
- Khi người dùng chọn một truyện, hiển thị danh sách các chương tương ứng.
- Cho phép người dùng đọc nội dung từng chương truyện với trải nghiệm mượt mà.
- Tìm hiểu, áp dụng các công nghệ hiện đại như Firestore, RecyclerView, ViewBinding,...

Thông qua đề tài, người thực hiện cũng mong muốn rèn luyện và nâng cao các kỹ năng:

- Phân tích và thiết kế ứng dụng di động.
- Làm việc với cơ sở dữ liệu thời gian thực.

**1.3. Phạm vi của đề tài**

Đề tài tập trung vào xây dựng ứng dụng đọc truyện chữ với các chức năng cơ bản. Các chức năng được giới hạn trong phạm vi sau:

- Ứng dụng chỉ hỗ trợ trên **thiết bị Android**.
- Dữ liệu được lấy từ **Firebase Realtime Database** đã được thêm sẵn (quản trị viên có thể thêm truyện thông qua ứng dụng hoặc thêm trực tiếp trên database).
- Giao diện đọc truyện ở mức đơn giản, không có tính năng nâng cao như thay đổi font, kích thước chữ, chế độ nền tối, v.v.

Phạm vi được giới hạn nhằm đảm bảo đề tài phù hợp với thời gian, nguồn lực và mục tiêu học tập của bài tập lớn.

**1.4. Phân chia nhiệm vụ**

Công việc, chức năng	Thành viên thực hiện
Thiết kế CSDL	Nghĩa, Quang, Toàn, Vũ
Thiết kế giao diện (Figma)	Nghĩa, Quang, Toàn, Vũ
Đăng nhập, đăng ký, màn hình chính (tìm kiếm, danh sách truyện).	Nghĩa
Thêm, sửa, xóa truyện và chapter. Xác thực email, quên/ đổi mật khẩu.	Toàn
Nội dung chapter, thông tin truyện, viết báo cáo.	Quang
Theo dõi truyện, danh sách theo dõi, viết báo cáo.	Vũ

*Bảng 2: Phân chia nhiệm vụ*



## Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

### 2.1. Kiến trúc hệ thống

Hệ thống ứng dụng đọc truyện chữ được xây dựng theo mô hình **Client – Server**, trong đó:

- **Client (Ứng dụng Android):** Là nơi người dùng tương tác trực tiếp. Giao diện ứng dụng được xây dựng bằng Android (Java), có nhiệm vụ hiển thị danh sách truyện, danh sách chương và nội dung chương. Các thao tác như chọn truyện, chọn chương, và đọc nội dung đều diễn ra tại client.
- **Server (Firebase Realtime Database):** Là nơi lưu trữ toàn bộ dữ liệu truyện, chương và nội dung. Firebase cung cấp một hệ thống backend thời gian thực, cho phép ứng dụng Android lấy và đồng bộ dữ liệu một cách nhanh chóng mà không cần viết server backend riêng.

### 2.2. Giới thiệu về Công nghệ phát triển

Để xây dựng ứng dụng đọc truyện chữ, nhóm đã sử dụng các công nghệ và công cụ hiện đại, phổ biến trong phát triển ứng dụng di động:

#### 2.2.1. Ngôn ngữ lập trình: Java

- Java là ngôn ngữ chính thức hỗ trợ phát triển ứng dụng Android.
- Có cộng đồng phát triển lớn, dễ học và tài liệu phong phú.

#### 2.2.2 Android Studio

- Là môi trường phát triển (IDE) chính thức do Google phát hành cho Android.
- Hỗ trợ công cụ thiết kế giao diện, giả lập, debug và tích hợp Firebase dễ dàng.

#### 2.2.3. Firebase Realtime Database

- Là hệ thống cơ sở dữ liệu NoSQL do Google phát triển.
- Hỗ trợ lưu trữ và đồng bộ dữ liệu theo thời gian thực.
- Tích hợp tốt với Android thông qua Firebase SDK.

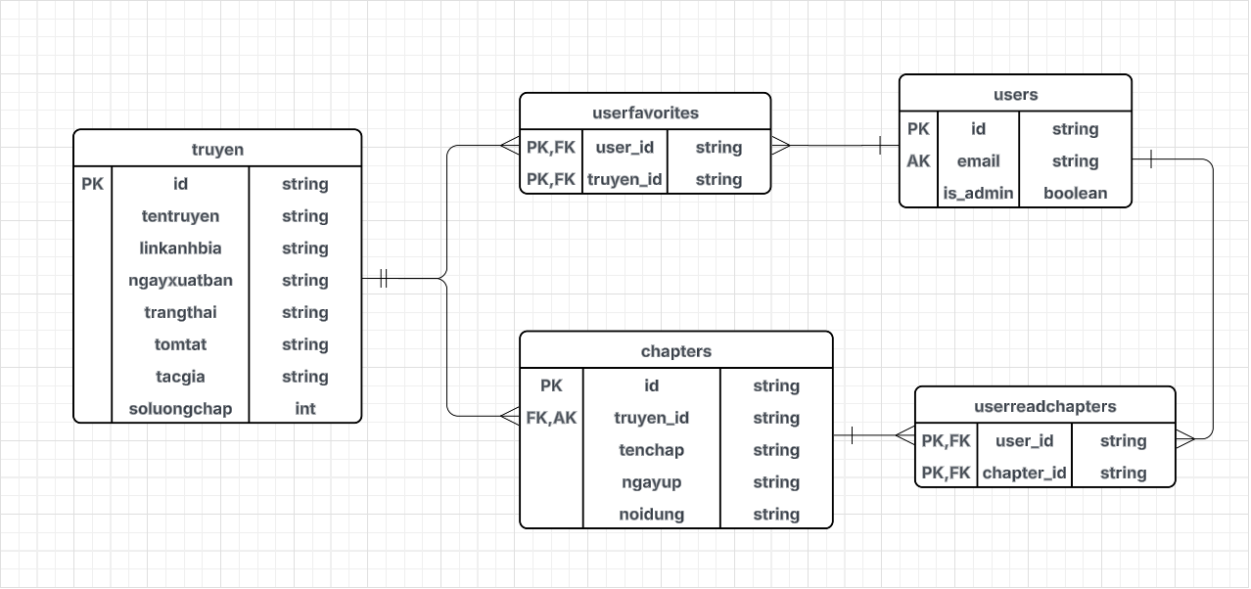
## Chương 3. XÂY DỰNG ỨNG DỤNG

### 3.1. Thiết kế Figma

Github: [toanv12345/AppDocTruyen](https://github.com/toanv12345/AppDocTruyen)

Figma: [App đọc truyện chữ – Figma](#)

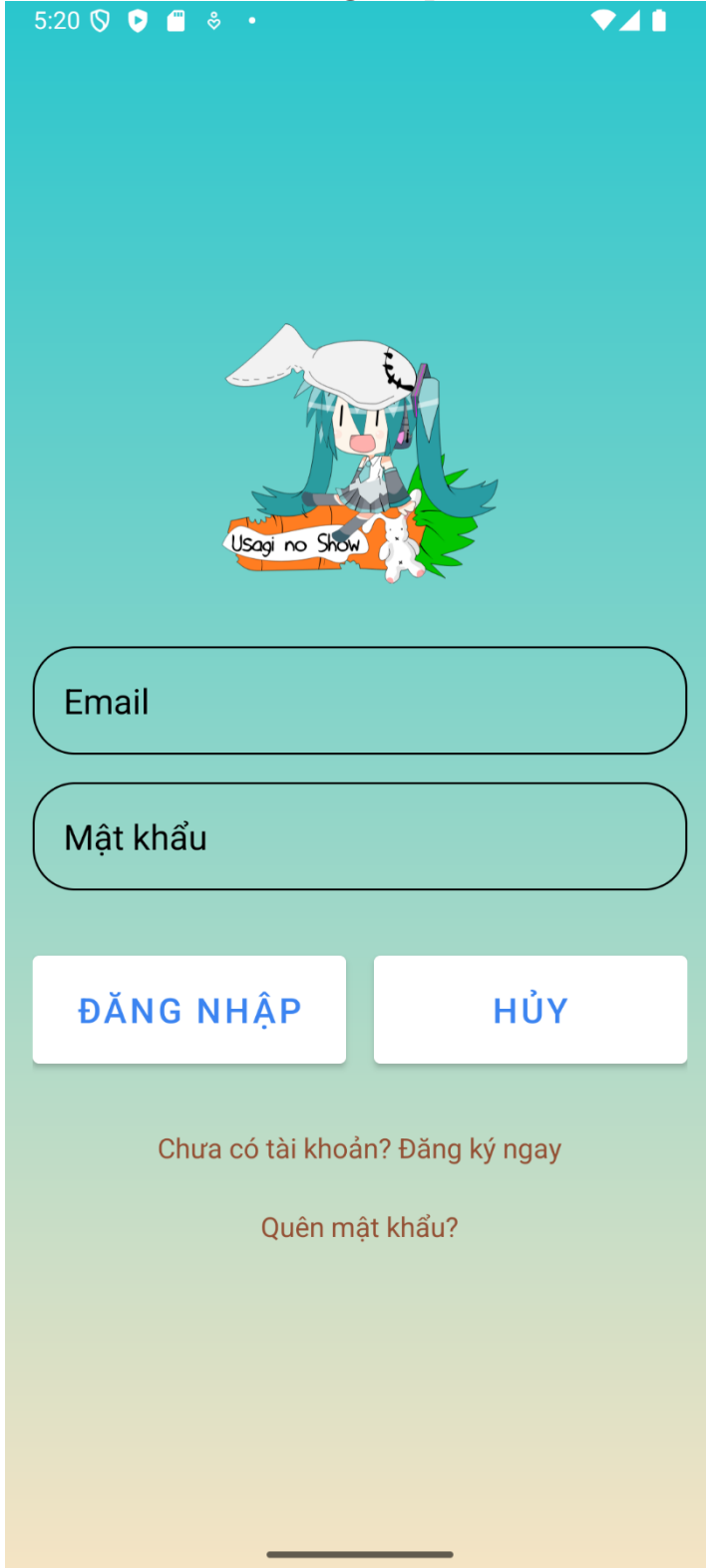
### 3.2. Thiết kế CSDL



Ảnh 1: Thiết kế CSDL

### 3.3. Giao diện ứng dụng

#### 3.3.1. Màn hình đăng nhập



5:20

Usagi no Show

Email

Mật khẩu

ĐĂNG NHẬP

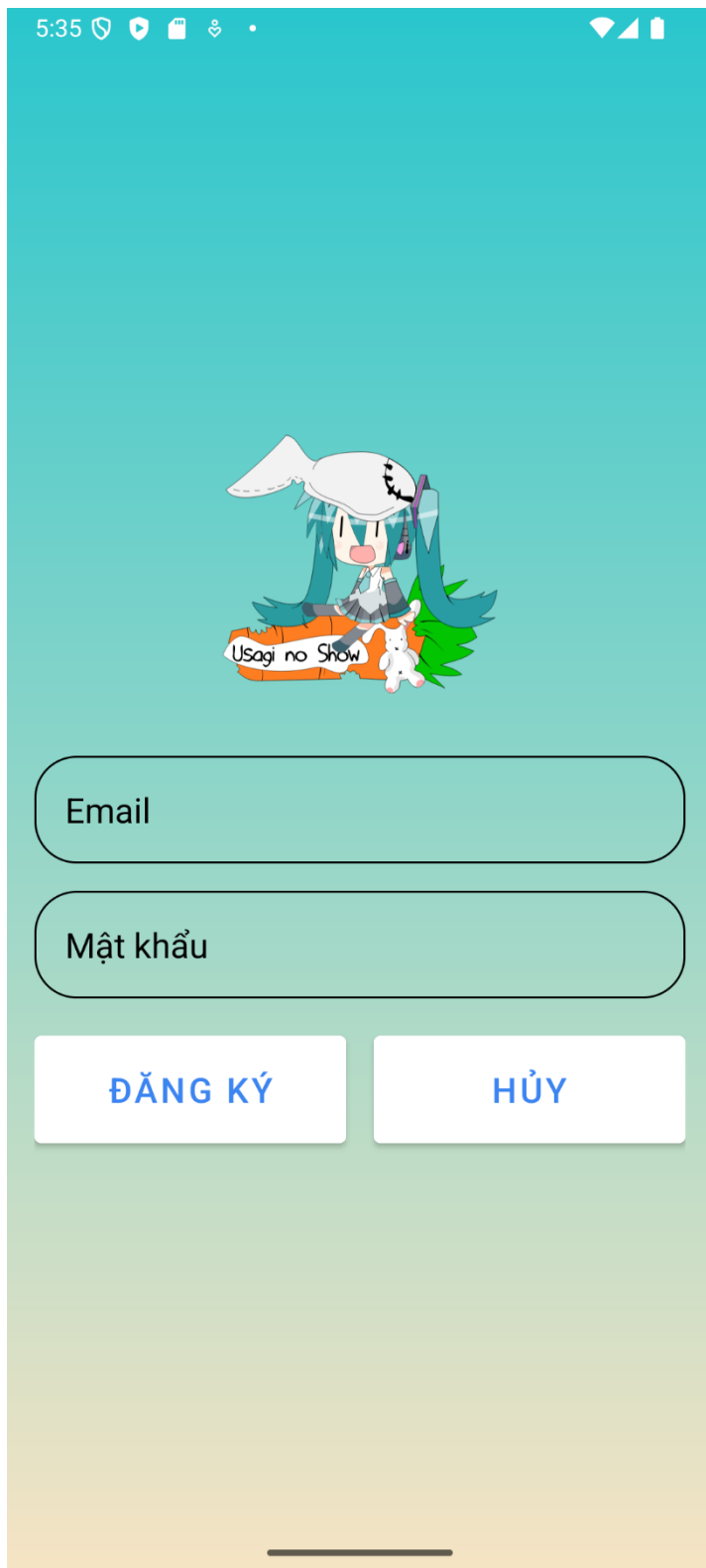
HỦY

Chưa có tài khoản? Đăng ký ngay

Quên mật khẩu?

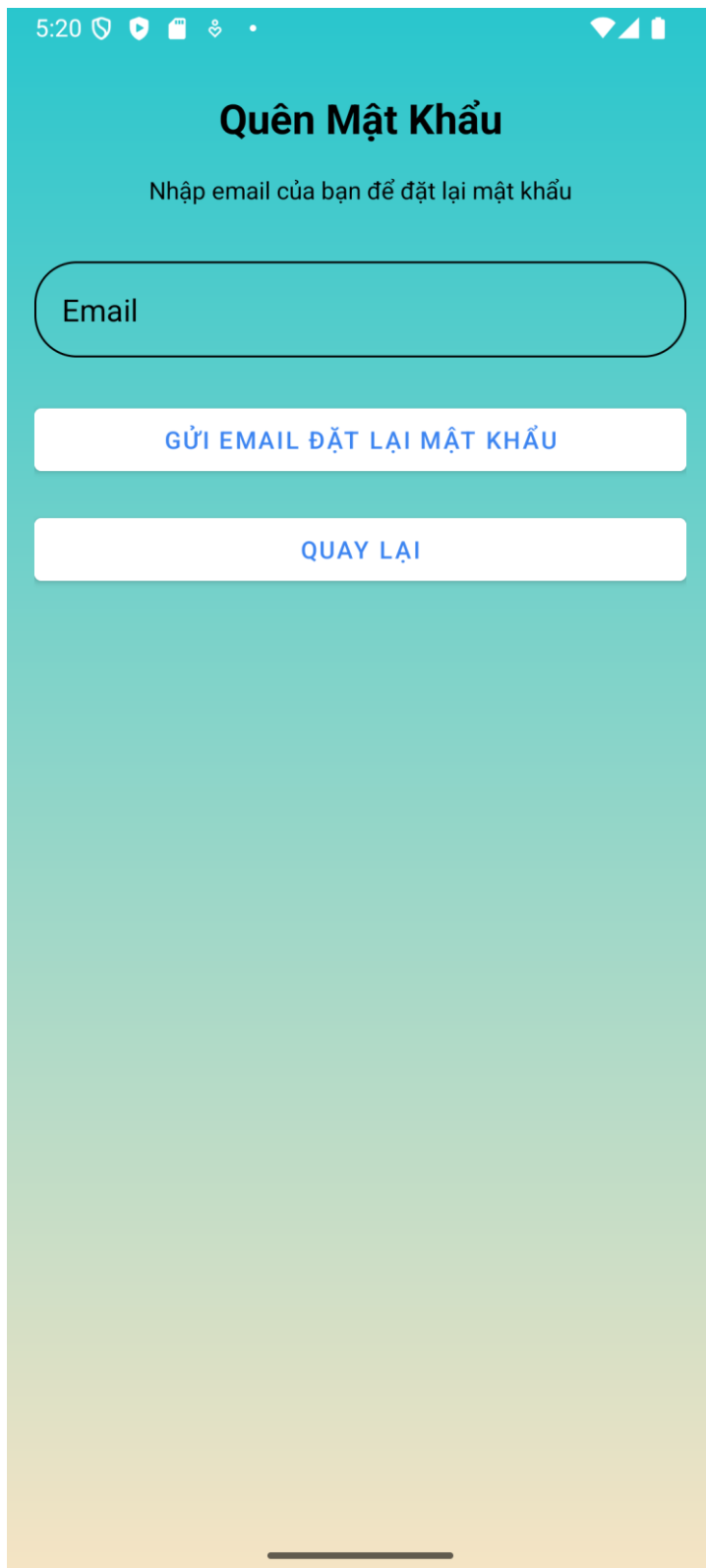
Ảnh 2: Màn hình đăng nhập

#### 3.3.2. Màn hình đăng ký



Ảnh 3: Màn hình đăng ký

### 3.3.3. Màn hình quên mật khẩu



Ảnh 4: Màn hình quên mật khẩu

### 3.3.4. Tài khoản admin

### 3.3.4.1. Màn hình giao diện chính



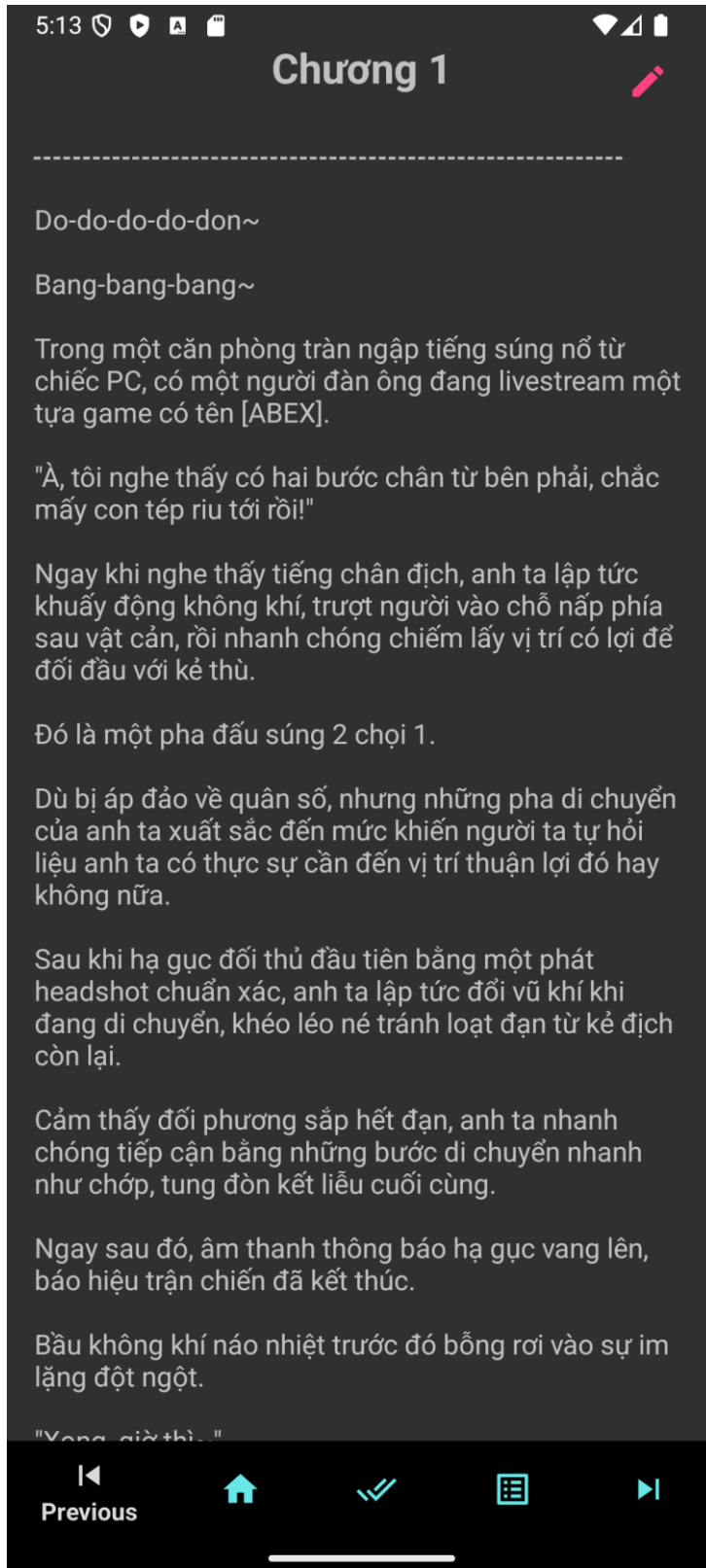
Ảnh 5: Màn hình giao diện chính (admin)

3.3.4.2. Màn hình thông tin truyện



Ảnh 6: Màn hình thông tin truyện (admin)

### 3.3.4.3. Màn hình nội dung chapter



Ảnh 7: Màn hình nội dung chapter (admin)



#### 3.3.4.4. Màn hình thêm chương truyện mới

Thêm chapter mới

Tên chapter

Ngày đăng

Nội dung chapter

THÊM CHAPTER

Ảnh 8: Màn hình thêm chapter

### 3.3.4.5. Màn hình thêm truyện mới

15:27 0.03 Kbps 4G 4G 56

## Thêm truyện mới

### Ảnh bìa

URL ảnh bìa

XEM TRƯỚC ẢNH

### Thông tin cơ bản

Tên truyện

Tác giả

Thể loại

Ảnh 9: Màn hình thêm truyện

### 3.3.4.6. Màn hình sửa chương truyện

15:28

0.00 Kbps

4G

4G

57

Tên chapter

Chương 7

Ngày đăng

2025/04/12

Nội dung chapter

<p id="1">Thương nhân Rudy tự nhận mình là kẻ được những vị thần ưu ái.</p><p id="2">Là người con thứ 3 được sinh ra trong gia đình Tử tước Maggio – gia đình có truyền thống làm thương nhân. Mặc dù không được thừa kế tài sản nhưng cậu đã rất may mắn khi được sống trong một môi trường tốt đẹp, không phải chịu sự phân biệt đối xử, được nuôi nấng dưới bàn tay của cha mẹ cùng với 2 người anh của mình. Cậu cũng nhận được nền giáo dục cao cấp như cả 2 người anh trai và đã sẵn sàng để sống độc lập từ thuở bé, vì cậu không

CẬP NHẬT

XÓA NỘI DUNG CHAPTER

Ảnh 10: Màn hình sửa chapter

### 3.3.4.7. Màn hình sửa truyện

15:28

2.00 KB/s 4G 4G 57

Sửa Truyện

Tên truyện

The Death Mage Who Doesn't Want a Fourth Time

Tác giả

Densuke

Thể loại

Action, Drama, Ecchi, Fantasy, Mature, Tragedy

Ngày xuất bản

10/10/2025


Tình trạng: Tạm ngưng

Số lượng chap: 7

Tóm tắt

<p>Amamiya Hiroto thật xui xẻo. Cuộc đời của cậu tràn ngập bất hạnh này đến bất hạnh khác và đỉnh điểm là cậu đã chết trong một chuyến đi chơi của trường trong khi cố gắng cứu một cô gái mà cậu còn không quen biết. Sau cái

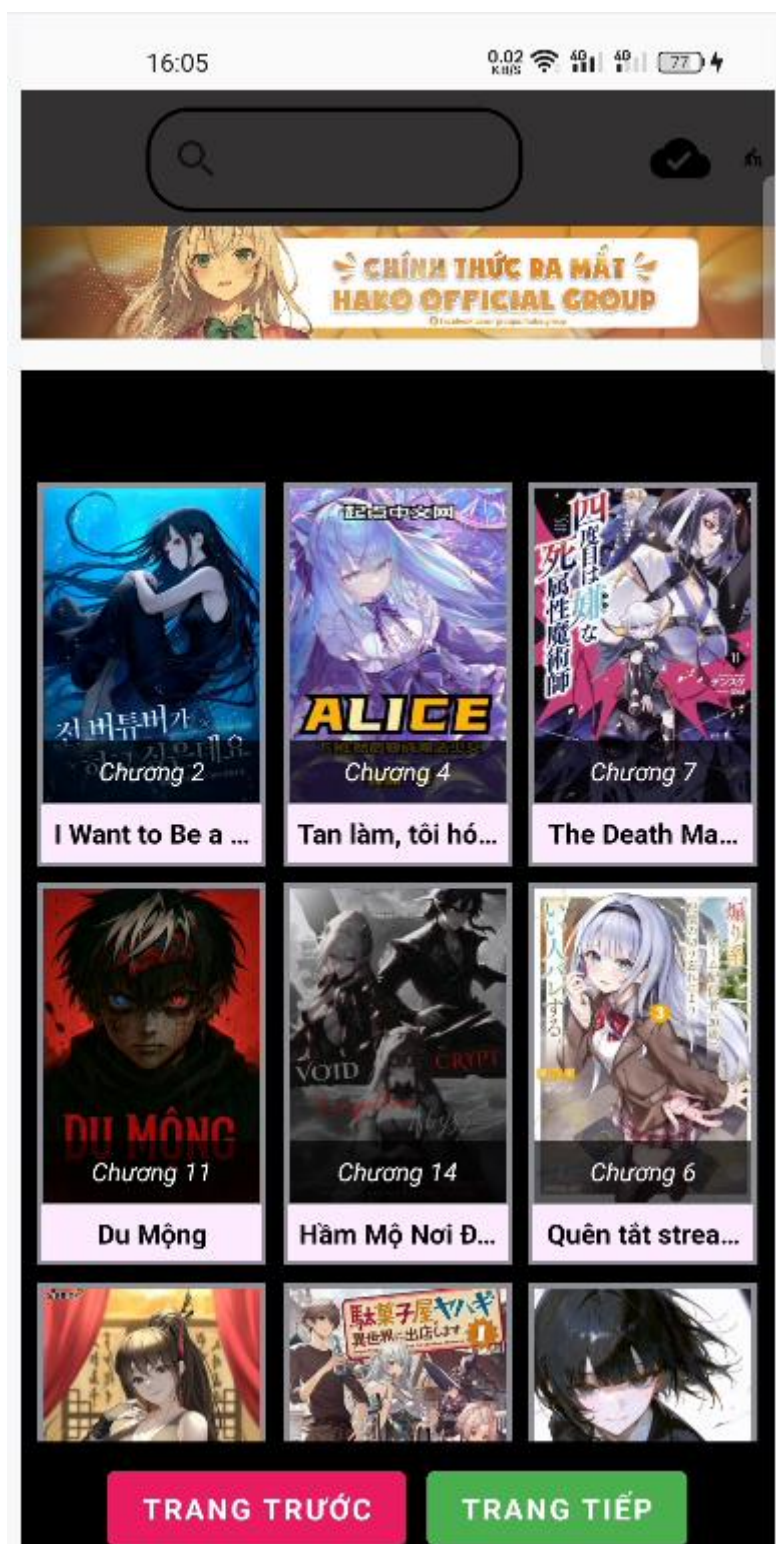
Ảnh bìa



Ảnh 11: Màn hình sửa truyện

### 3.3.5. Tài khoản user

### 3.3.5.1. Màn hình giao diện chính



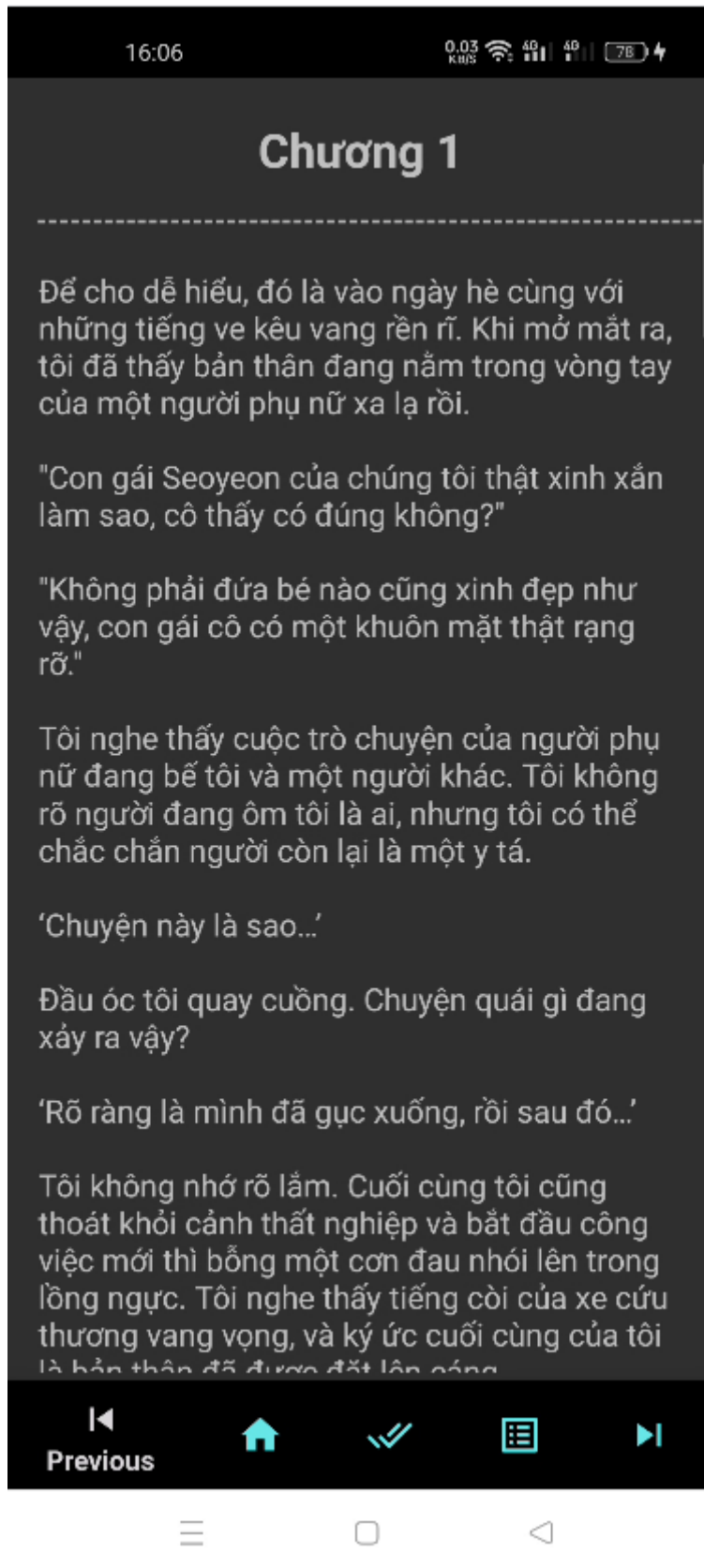
Ảnh 12: Màn hình giao diện chính (user)

### 3.3.5.2. Màn hình thông tin truyện



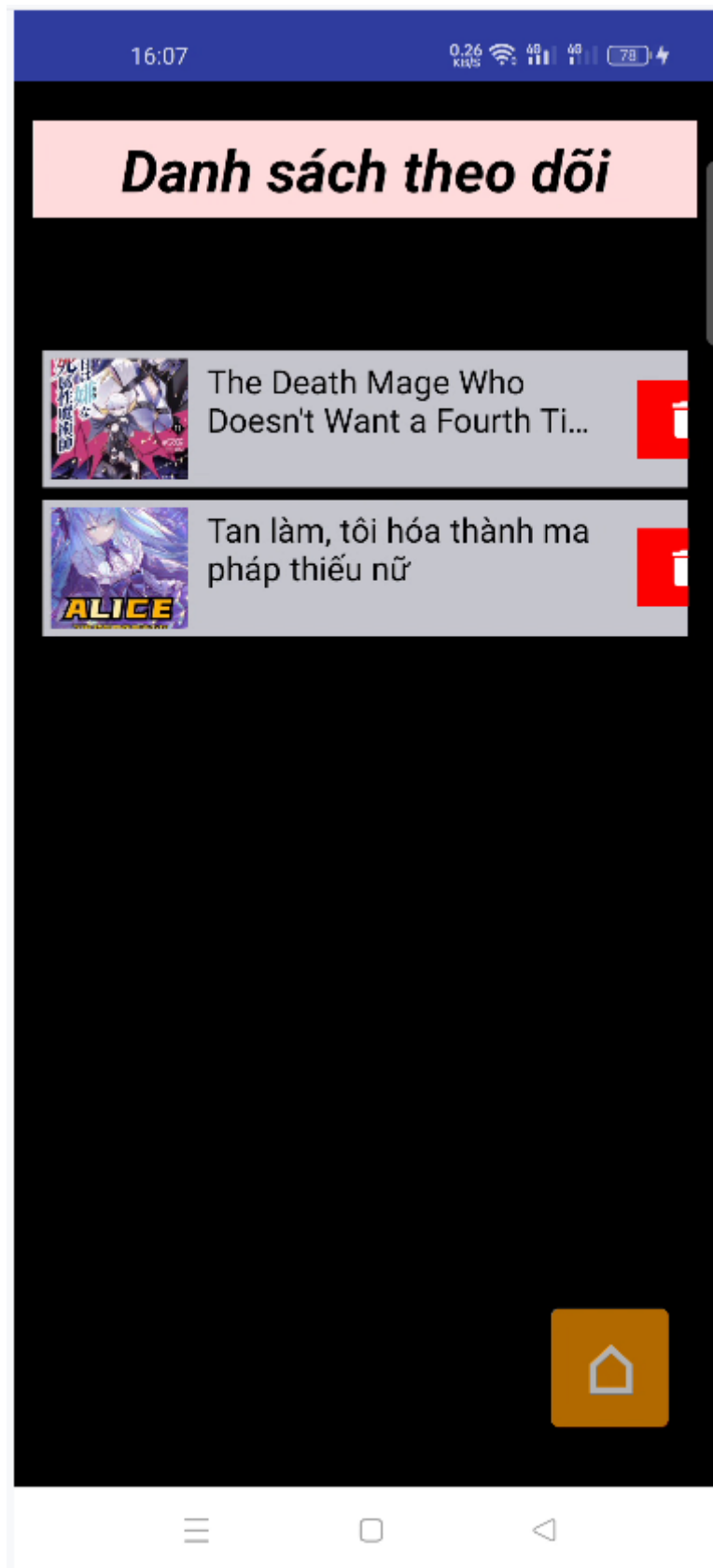
Ảnh 13: Màn hình thông tin truyện (user)

### 3.3.5.3. Màn hình nội dung chapter



Ảnh 14: Màn hình nội dung chapter (user)

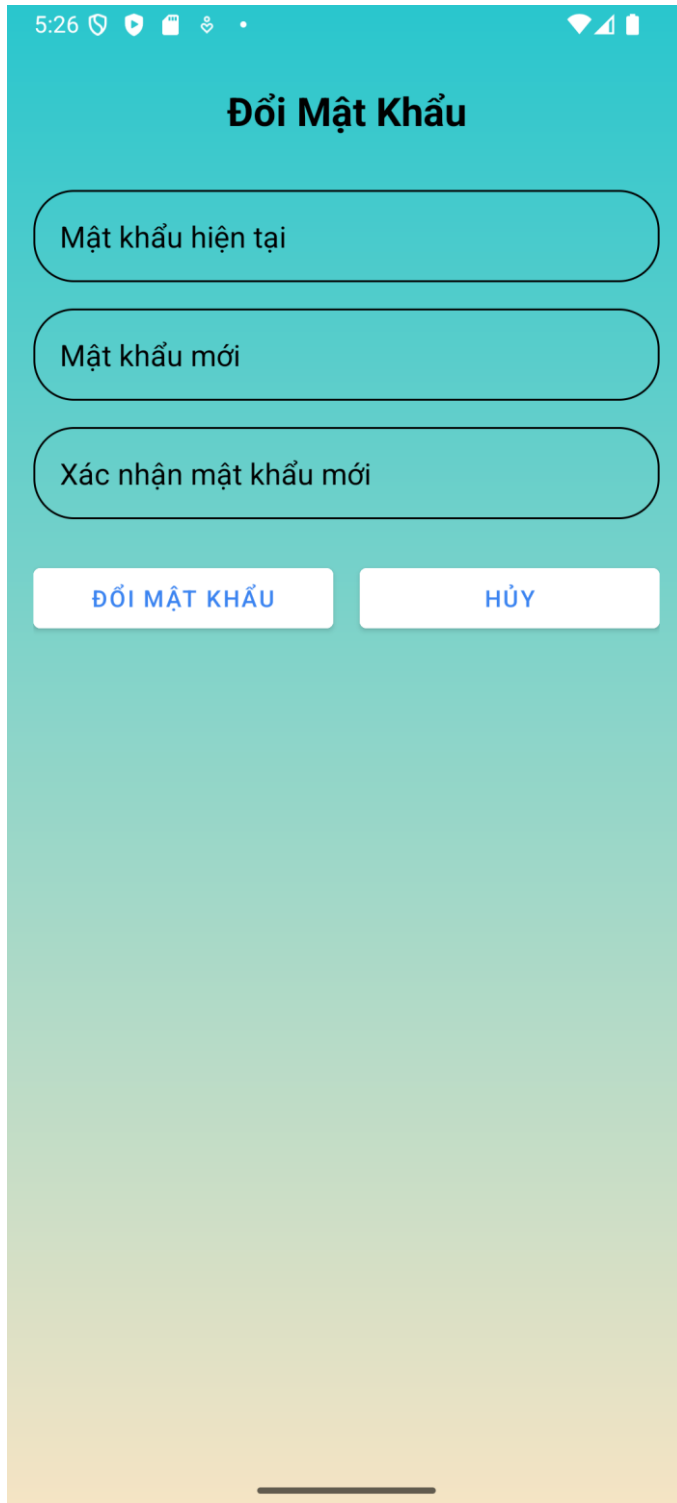
#### 3.3.5.4. Màn hình danh sách theo dõi



Ảnh 15: Màn hình danh sách theo dõi



### 3.3.5.5. Màn hình đổi mật khẩu



5:26

**Đổi Mật Khẩu**

Mật khẩu hiện tại

Mật khẩu mới

Xác nhận mật khẩu mới

ĐỔI MẬT KHẨU HỦY

Ảnh 16: Màn hình đổi mật khẩu

### 3.4. Code minh họa các chức năng cốt lõi

#### 3.4.1. Chức năng đăng ký

```
btnSignUp.setOnClickListener(v -> {  
    String email = editMail.getText().toString();  
    String pass = editPass.getText().toString();  
  
    // Kiểm tra thông tin đăng ký  
    if (email.isEmpty() || pass.isEmpty()) {  
        Toast.makeText(SignUpActivity.this, "Vui lòng điền đầy đủ thông tin!",  
            Toast.LENGTH_SHORT).show();  
        return;  
    }  
  
    Auth.createUserWithEmailAndPassword(email, pass).addOnCompleteListener(task ->  
    {  
        if (task.isSuccessful()) {  
            Toast.makeText(SignUpActivity.this, "Đăng ký thành công!",  
                Toast.LENGTH_SHORT).show();  
  
            FirebaseUser currentUser = Auth.getCurrentUser();  
            if (currentUser != null) {  
                // Gửi email xác thực  
                currentUser.sendEmailVerification().addOnCompleteListener(verifyTask -> {  
                    if (verifyTask.isSuccessful()) {  
                        // Lưu thông tin người dùng vào Firestore  
                        Map<String, Object> user = new HashMap<>();  
                        user.put("email", email);  
                        user.put("favorite", new ArrayList<String>());  
  
                        db.collection("users").document(currentUser.getUid())  
                            .set(user)  
                            .addOnSuccessListener(aVoid -> {
```

```

// Lưu thông tin người dùng vào Realtime Database
DatabaseReference userRef =
dbRef.child("users").child(currentUser.getUid());

Map<String, Object> userData = new HashMap<>();
userData.put("email", email);
userData.put("isAdmin", false); // Mặc định không phải admin
userData.put("emailVerified", false); // Thêm trạng thái xác thực
email

userRef.setValue(userData).addOnCompleteListener(task1 -> {
    if (task1.isSuccessful()) {
        // Hiển thị thông báo nhưng vẫn giữ phiên đăng nhập hiện tại
        showVerificationDialog(email);
    } else {
        Toast.makeText(SignUpActivity.this, "Lỗi khi lưu thông tin
người dùng!", Toast.LENGTH_SHORT).show();
    }
});

})
.addOnFailureListener(e -> {
    Toast.makeText(SignUpActivity.this, "Lỗi khi lưu thông tin người
dùng!", Toast.LENGTH_SHORT).show();
});

} else {
    Toast.makeText(SignUpActivity.this, "Không thể gửi email xác thực.",
Toast.LENGTH_SHORT).show();
}

});

} else {
    Toast.makeText(SignUpActivity.this, "Lỗi khi lấy thông tin người dùng!",
Toast.LENGTH_SHORT).show();
}
}

```

```

    } else {
        String errorMessage = task.getException() != null ?
task.getException().getMessage() : "Đăng ký thất bại!";
        Toast.makeText(SignUpActivity.this, errorMessage,
Toast.LENGTH_SHORT).show();
    }
});
});

```

### 3.4.2. Chức năng đăng nhập

```

btnLogin.setOnClickListener(v -> {
    String email = editMail.getText().toString();
    String pass = editPass.getText().toString();

    if (email.isEmpty() || pass.isEmpty()) {
        Toast.makeText(LoginActivity.this, "Vui lòng nhập đầy đủ thông tin!",
Toast.LENGTH_SHORT).show();
        return;
    }
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        Toast.makeText(LoginActivity.this, "Địa chỉ email không hợp lệ!",
Toast.LENGTH_SHORT).show();
        return;
    }
    Log.d(TAG, "Attempting to log in with email: " + email);

    // thực hiện đăng nhập
    Auth.signInWithEmailAndPassword(email, pass)
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                Log.d(TAG, "Login successful");
                FirebaseUser user = Auth.getCurrentUser();
                if (user != null) {

```

```

// Đầu tiên, kiểm tra xem người dùng có phải admin không
checkIfAdmin(user, isAdmin -> {
    if (isAdmin) {
        // Nếu là admin, không cần xác thực email
        checkAdminAndProceed(user);
    } else {
        // Nếu không phải admin, kiểm tra xác thực email
        EmailVerificationChecker.checkEmailVerificationStatus(new
EmailVerificationChecker.VerificationCallback() {
            @Override
            public void onVerified() {
                // Đã xác thực email, tiếp tục đăng nhập bình thường
                checkAdminAndProceed(user);
            }

            @Override
            public void onNotVerified() {
                // Hiển thị thông báo yêu cầu xác thực email
                Toast.makeText(LoginActivity.this,
                    "Một số tính năng sẽ bị hạn chế cho đến khi bạn xác
thực email.",
                        Toast.LENGTH_LONG).show();
                checkAdminAndProceed(user);
            }
        });
    }
});
} else {
    Log.e(TAG, "Login failed: " + task.getException().getMessage());
    Toast.makeText(LoginActivity.this, "Đăng nhập thất bại!",
        Toast.LENGTH_SHORT).show();
}

```

```

    }
});
});

```

### 3.4.3. Chức năng đăng xuất

```

// Xử lý đăng xuất
new AlertDialog.Builder(MainActivity.this)
    .setTitle("Đăng xuất")
    .setMessage("Bạn có chắc chắn muốn đăng xuất?")
    .setPositiveButton("Có", (dialog, which) -> {
        auth.signOut();
        checkLoginStatus();
        btnAdd.setVisibility(View.GONE);
        Toast.makeText(MainActivity.this, "Đăng xuất thành công",
Toast.LENGTH_SHORT).show();
    })
    .setNegativeButton("Không", null)
    .show();
return true;

```

### 3.4.4. Chức năng theo dõi truyện

```

private void updateFollowStatus(String novelId, boolean isFollowed) {
    FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();
    if (currentUser != null) {
        DatabaseReference userRef = FirebaseDatabase.getInstance()
            .getReference("users")
            .child(currentUser.getUid())
            .child("favorite");

        if (isFollowed) {
            userRef.child(novelId).setValue(true)
                .addOnSuccessListener(aVoid -> {
                    Toast.makeText(NovelsInfoActivity.this, "Đã theo dõi truyện",

```

```

Toast.LENGTH_SHORT).show();

    })

    .addOnFailureListener(e -> {

        Toast.makeText(NovelsInfoActivity.this, "Lỗi cập nhật trạng thái theo dõi:
" + e.getMessage(), Toast.LENGTH_SHORT).show();

        });

    } else {

        userRef.child(novelId).removeValue()

        .addOnSuccessListener(aVoid -> {

            Toast.makeText(NovelsInfoActivity.this, "Đã bỏ theo dõi truyện",
Toast.LENGTH_SHORT).show();

            })

        .addOnFailureListener(e -> {

            Toast.makeText(NovelsInfoActivity.this, "Lỗi cập nhật trạng thái theo dõi:
" + e.getMessage(), Toast.LENGTH_SHORT).show();

            });

        }

    }

}

```

### 3.4.5. Chức năng thêm truyện

```
databaseReference.child(novelId).setValue(novelData)
    .addOnSuccessListener(aVoid -> {
        // Tạo node chapter trống
        databaseReference.child(novelId).child("chapter").setValue(null)
        .addOnSuccessListener(aVoid1 -> {
            loadingDialog.dismiss();
            Toast.makeText(AddNovelActivity.this, "Thêm truyện thành công!",
                Toast.LENGTH_SHORT).show();

            // Quay về MainActivity
            Intent intent = new Intent(AddNovelActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        })
        .addOnFailureListener(e -> {
            loadingDialog.dismiss();
            Toast.makeText(AddNovelActivity.this, "Lỗi khi tạo chương: " +
                e.getMessage(), Toast.LENGTH_SHORT).show();
        });
    })
    .addOnFailureListener(e -> {
        loadingDialog.dismiss();
        Toast.makeText(AddNovelActivity.this, "Lỗi khi thêm truyện: " +
            e.getMessage(), Toast.LENGTH_SHORT).show();
    });
```



### 3.4.6. Chức năng cập nhật truyện

```
Map<String, Object> novelUpdates = new HashMap<>();
    novelUpdates.put("tentruyen", title);
    novelUpdates.put("tacgia", author);
    novelUpdates.put("theloai", genre);
    novelUpdates.put("ngayxuatban", publishDate);
    novelUpdates.put("tomtat", summary);
    novelUpdates.put("tinhtrang", status);

    // Thêm URL ảnh vào dữ liệu cập nhật nếu có
    if (!coverUrl.isEmpty()) {
        novelUpdates.put("linkanh", coverUrl);
    }

    // Cập nhật dữ liệu
    updateNovelData(novelUpdates, loadingDialog);
}

private void updateNovelData(Map<String, Object> novelUpdates, AlertDialog
loadingDialog) {
    novelRef.child(novelId).updateChildren(novelUpdates)
        .addOnSuccessListener(aVoid -> {
            loadingDialog.dismiss();
            Toast.makeText(EditNovelActivity.this, "Cập nhật thành công",
Toast.LENGTH_SHORT).show();
            finish();
        })
        .addOnFailureListener(e -> {
            loadingDialog.dismiss();
            Toast.makeText(EditNovelActivity.this, "Lỗi: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
        });
}
```

### 3.4.7. Chức năng xóa truyện

```
private void deleteNovel() {
    AlertDialog loadingDialog = showLoadingDialog();

    // Xóa truyện và tất cả chapter
    novelRef.child(novelId).removeValue()
        .addOnSuccessListener(aVoid -> {
            // Cập nhật truyện yêu thích của người dùng
            removeFromUserFavorites(loadingDialog);
        })
        .addOnFailureListener(e -> {
            loadingDialog.dismiss();
            Toast.makeText(EditNovelActivity.this, "Lỗi khi xóa truyện: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
        });
}

novelRef.child(novelId).removeValue()
    .addOnSuccessListener(aVoid -> {
        // Cập nhật truyện yêu thích của người dùng
        removeFromUserFavorites(loadingDialog);
    })
    .addOnFailureListener(e -> {
        loadingDialog.dismiss();
        Toast.makeText(EditNovelActivity.this, "Lỗi khi xóa truyện: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
    });
}
```

### 3.4.8. Chức năng tìm truyện

```
private void filterNovels(String query) {
    ArrayList<Novel> filteredList = new ArrayList<>();
    for (Novel novel : novelArrayList) {
        String title = novel.getTentruyen() != null ? novel.getTentruyen().toLowerCase() :
"";
        String author = novel.getTacgia() != null ? novel.getTacgia().toLowerCase() : "";
        String status = novel.getTinhtrang() != null ? novel.getTinhtrang().toLowerCase() :
"";
        String genre = novel.getTheloai() != null ? novel.getTheloai().toLowerCase() : "";

        if (title.contains(query.toLowerCase()) ||
            author.contains(query.toLowerCase()) ||
            status.contains(query.toLowerCase()) ||
            genre.contains(query.toLowerCase())) {
            filteredList.add(novel);
        }
    }
    novelAdapter.updateList(filteredList);
}
```

### 3.4.9. Chức năng đổi mật khẩu

```
private void changePassword(String currentPassword, String newPassword) {
    // Xác thực lại người dùng trước khi đổi mật khẩu
    AuthCredential credential =
EmailAuthProvider.getCredential(currentUser.getEmail(), currentPassword);

    currentUser.reauthenticate(credential)
        .addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    // Xác thực thành công, đổi mật khẩu
                    currentUser.updatePassword(newPassword)
                        .addOnCompleteListener(new OnCompleteListener<Void>() {
                            @Override
                            public void onComplete(@NonNull Task<Void> task) {
```

```

        progressBar.setVisibility(View.GONE);
        if (task.isSuccessful()) {
            Toast.makeText(ChangePasswordActivity.this,
                "Đổi mật khẩu thành công",
                Toast.LENGTH_SHORT).show();
            finish();
        } else {
            Toast.makeText(ChangePasswordActivity.this,
                "Đổi mật khẩu thất bại: " +
task.getException().getMessage(),
                Toast.LENGTH_SHORT).show();
        }
    }
});
} else {
    progressBar.setVisibility(View.GONE);
    Toast.makeText(ChangePasswordActivity.this,
        "Mật khẩu hiện tại không đúng",
        Toast.LENGTH_SHORT).show();
}
}
});
}

```

#### **3.4.10. Chức năng quên mật khẩu**

*// Xử lý nút đặt lại mật khẩu*

```

btnResetPassword.setOnClickListener(v -> {
    String email = editEmail.getText().toString().trim();

    if (email.isEmpty()) {
        Toast.makeText(ForgotPasswordActivity.this, "Vui lòng nhập email",
            Toast.LENGTH_SHORT).show();
        return;
    }

```

```

    progressBar.setVisibility(View.VISIBLE);

```

*// Kiểm tra nếu email là tài khoản admin*

```

if (email.equalsIgnoreCase(ADMIN_EMAIL)) {
    // Đây là tài khoản admin, không cho phép đặt lại mật khẩu

```

```

        progressBar.setVisibility(View.GONE);
        txtStatus.setText("Tài khoản admin không thể đặt lại mật khẩu thông qua ứng
dụng.");
        Toast.makeText(ForgotPasswordActivity.this,
            "Tài khoản admin không thể đặt lại mật khẩu thông qua ứng dụng",
            Toast.LENGTH_LONG).show();
    } else {
        // Không phải tài khoản admin, tiến hành gửi email đặt lại mật khẩu
        mAuth.sendPasswordResetEmail(email)
            .addOnCompleteListener(task -> {
                progressBar.setVisibility(View.GONE);
                if (task.isSuccessful()) {
                    txtStatus.setText("Đã gửi email đặt lại mật khẩu.\nVui lòng kiểm tra hộp
thư của bạn.");
                    editEmail.setEnabled(false);
                    btnResetPassword.setEnabled(false);
                    Toast.makeText(ForgotPasswordActivity.this,
                        "Email đặt lại mật khẩu đã được gửi!",
                        Toast.LENGTH_SHORT).show();
                } else {
                    String errorMessage = task.getException() != null ?
                        task.getException().getMessage() :
                        "Không thể gửi email đặt lại mật khẩu.";
                    Toast.makeText(ForgotPasswordActivity.this, errorMessage,
Toast.LENGTH_LONG).show();
                }
            });
    }
});

```

## KẾT LUẬN

### 1. Kết quả đạt được

Dự án tạo ứng dụng Android đọc truyện chữ đã đạt được những kết quả khả quan, qua đó thể hiện năng lực và sự sáng tạo trong việc xây dựng một sản phẩm số đáp ứng nhu cầu đọc truyện trực tuyến. Cụ thể, nhóm đã hoàn thành việc xây dựng ứng dụng với đầy đủ các chức năng cơ bản theo như thiết kế ban đầu, giúp người dùng dễ dàng tra cứu và đọc truyện chữ một cách thuận tiện.

### 2. Nhược điểm

Trong quá trình phát triển, nhóm cũng nhận ra một số nhược điểm cần được cải thiện. Đầu tiên, giao diện hiện tại của ứng dụng chưa tối ưu cho việc hiển thị trên nhiều kích thước màn hình khác nhau, dẫn đến trải nghiệm người dùng không đồng nhất. Ngoài ra, ứng dụng còn thiếu một số chức năng quan trọng nhằm tạo nên một hệ sinh thái đọc truyện hoàn chỉnh như tính năng bình luận, đánh giá, chia sẻ truyện, báo lỗi hoặc hỗ trợ người dùng kịp thời.

### 3. Hướng phát triển

Nhìn về tương lai, dự án có nhiều hướng phát triển đáng chú ý. Nhóm dự kiến sẽ tiếp tục bổ sung và hoàn thiện các tính năng phụ trợ nhằm mang lại trải nghiệm toàn diện cho người dùng. Đặc biệt, việc phát triển các chức năng như bình luận, đánh giá, chia sẻ truyện và các kênh hỗ trợ người dùng sẽ góp phần nâng cao tính tương tác và trải nghiệm đọc. Bên cạnh đó, cải tiến giao diện người dùng để tối ưu hiển thị trên nhiều loại thiết bị với các kích thước màn hình khác nhau cũng là một mục tiêu quan trọng, hướng tới việc đem lại sự tiện lợi và trực quan cho khách hàng.

## TÀI LIỆU THAM KHẢO

- [1] [Trang chủ - Cổng Light Novel - Đọc Light Novel](#)
- [2] [Android studio - Hướng dẫn làm app đọc truyện - Phần 1](#)
- [3] [Android studio - Hướng dẫn làm app đọc truyện - Phần 2](#)
- [4] [Android studio - Hướng dẫn làm app đọc truyện - Phần 3](#)