

Firebase Authentication

Thành viên: Lê Xuân Nam 106220226
Nguyễn Nam Phương 106220266

GVHD: TS. Nguyễn Duy Nhật Viễn

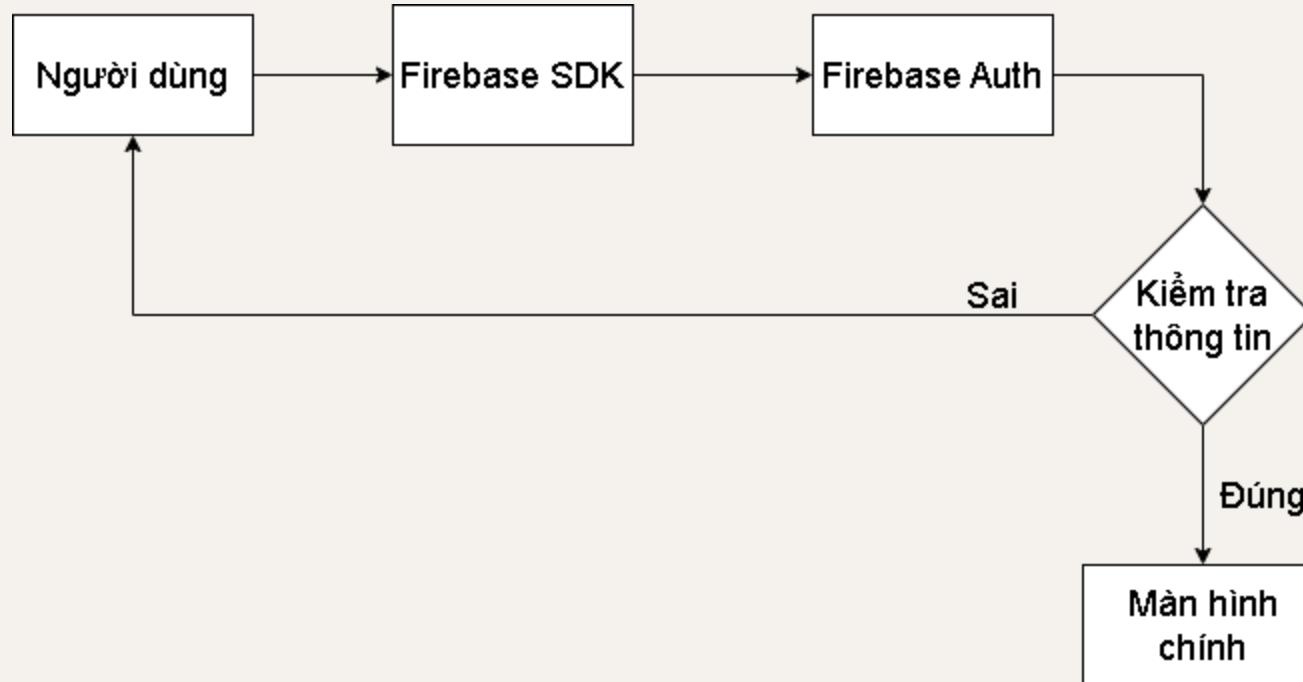
Mục lục

1. Firebase Authentication là gì?
2. Một số phương thức đăng nhập với Firebase Authentication
3. User Management & Profile
4. Các nguyên tắc bảo mật

1. Firebase Authentication là gì?

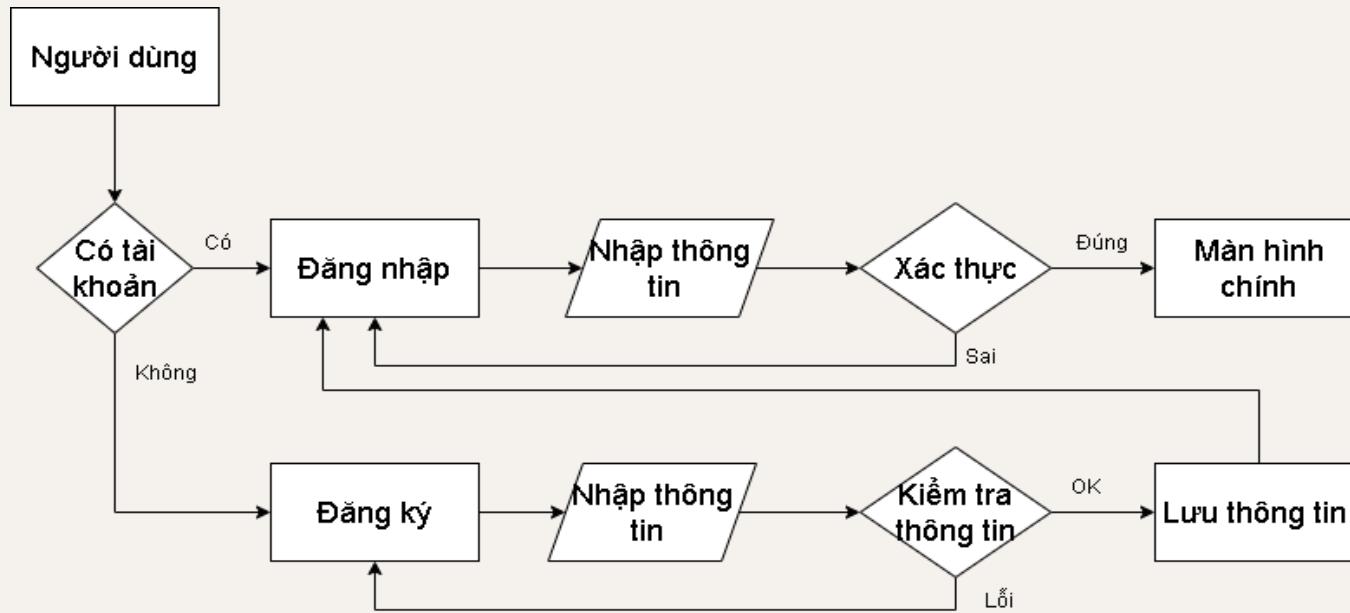
- Là một dịch vụ xác thực người dùng do Google cung cấp trong nền tảng Firebase
- Giúp chúng ta quản lý nhiều người dùng mà không cần thiết kế hệ thống phức tạp
- Hỗ trợ các phương thức đăng nhập: Email/password, Google, Facebook, Anonymous,....

Firebase Authentication hoạt động như thế nào?



2. Một số phương thức đăng nhập với Firebase Authentication

- a) Đăng nhập bằng Email/ password



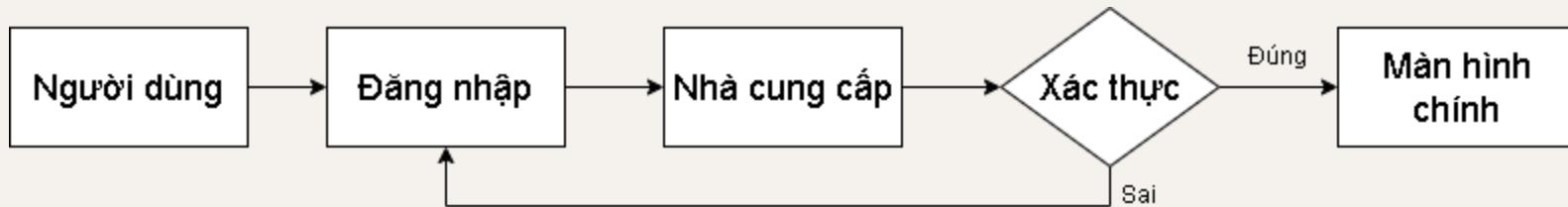
* Đăng nhập :

```
try {
  await FirebaseAuth.instance.signInWithEmailAndPassword(
    email: emailCtrl.text.trim(),
    password: passCtrl.text,
  );
} catch (e) {
  if (!mounted) return;
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(_error(e))));
}
```

* Đăng ký:

```
try {
  await FirebaseAuth.instance.createUserWithEmailAndPassword(
    email: emailCtrl.text.trim(),
    password: passCtrl.text,
  );
  if (!mounted) return;
  ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Đăng ký thành công!")));
} catch (e) {
  if (!mounted) return;
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(_error(e))));
}
```

b) Đăng nhập bằng tài khoản mạng xã hội



Ví dụ: đăng nhập với Google

- * Mở popup và lấy thông tin người dùng

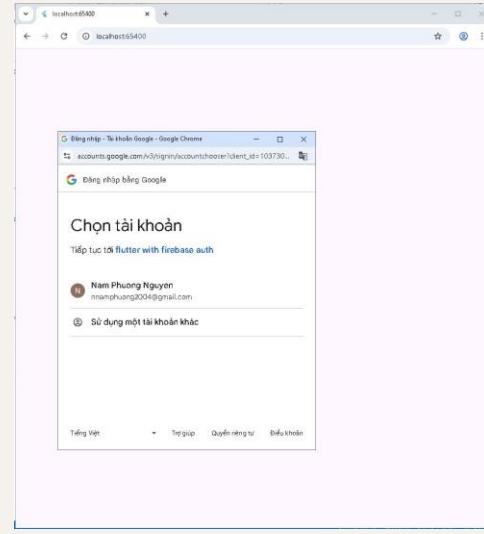
```
final googleUser = await googleSignIn.signIn();
```

- * Lấy id Token

```
final googleAuth = await googleUser.authentication;
```

- * Firebase SDK đưa id Token cho server để xác thực

```
final credential = GoogleAuthProvider.credential(  
    accessToken: googleAuth.accessToken,  
    idToken: googleAuth.idToken,  
);  
  
await FirebaseAuth.instance.signInWithCredential(credential);
```



3. User Management & Profile

- Sau khi đăng nhập người dùng có thể xem và quản lý thông tin của mình
 - Người dùng có thể:
 - Xem thông tin tài khoản hiện tại (ID, Email, Tên hiển thị, Ảnh đại diện)
 - Cập nhật tên hiển thị
 - Cập nhật ảnh đại diện
 - Thay đổi địa chỉ Email
 - Thay đổi mật khẩu đăng nhập
 - Đăng xuất khỏi tài khoản hiện tại

3.1 Khai báo TexteditingController và lấy thông tin user

-Cách khai báo:

```
User? get currentUser => FirebaseAuth.instance.currentUser;
```

- Đây là một **getter**. Dùng *Firebase* để lấy tên, email, UID, ảnh. User trả về đối tượng người dùng đang đăng nhập
- Dùng để gọi user.displayName, user.updatePassword(...), v.v.

3.1 Khai báo TexteditingController và lấy thông tin user

-Cách khai báo:

```
class ProfilePageState extends State<ProfilePage> {  
    final displayNameCtrl = TextEditingController();  
    final photoUrlCtrl = TextEditingController();  
    final emailCtrl = TextEditingController();  
    final newPassCtrl = TextEditingController();  
    final curPassCtrl = TextEditingController();
```

- Tạo 5 đối tượng TextEditingController, giúp quản lý nội dung nhập liệu trong các TextField
- Từng controller tương ứng với 1 thông tin

3.2 Cập nhật tên hiển thị và ảnh đại diện

-Cách khai báo:

```
Future<void> updateProfile() async {
    final user = currentUser;
    if (user == null) {
        showSnackBar("Chưa đăng nhập");
        return;
    }
    await user.updateDisplayName(displayNameCtrl.text.trim());
    await user.updatePhotoURL(photoUrlCtrl.text.trim());
    await user.reload();
    setState(() {});
    showSnackBar("Cập nhật hồ sơ thành công");
}
```

-Hàm này giúp cập nhật thông tin hồ sơ Firebase của người dùng hiện tại, bao gồm:

- Tên hiển thị mới (displayName)
- Ảnh đại diện mới (photoURL)

3.2 Cập nhật tên hiển thị và ảnh đại diện

Hồ sơ người dùng

DEBUG

UID:
gjGkrBAY3WfJa71bQcUWaq8sg643
Email: 123@gmail.com
Display Name: Nam



Cập nhật hồ sơ

Tên hiển thị

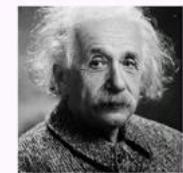
URL Hình ảnh

Cập nhật hồ sơ

Hồ sơ người dùng

DEBUG

UID:
gjGkrBAY3WfJa71bQcUWaq8sg643
Email: 123@gmail.com
Display Name: Nam Phương



Cập nhật hồ sơ

Tên hiển thị

Nam Phương

URL Hình ảnh

../d/d3/Albert_Einstein_Head.jpg

Cập nhật hồ sơ thành công

Sau khi ấn cập nhật hồ sơ. Hình ảnh đại diện và Tên sẽ được thay đổi

3.3 Hàm xác thực người dùng

-Khi đổi email, mật khẩu, Firebase yêu cầu phải xác thực để đảm bảo an toàn.

-Hàm reauthenticate() có chức năng xác minh lại danh tính của người dùng bằng cách yêu cầu nhập lại mật khẩu hiện tại và email đã đăng ký.

```
Future<void> reauthenticate() async {
    final user = currentUser;
    final email = user?.email;
    final password = curPassCtrl.text.trim();

    if (email == null || password.isEmpty) {
        throw Exception("Chưa nhập email hoặc mật khẩu hiện tại");
    }

    final xacthuc = EmailAuthProvider.credential(
        email: email,
        password: password,
    );
    await user!.reauthenticateWithCredential(xacthuc);
}
```

3.3 Thay đổi mật khẩu

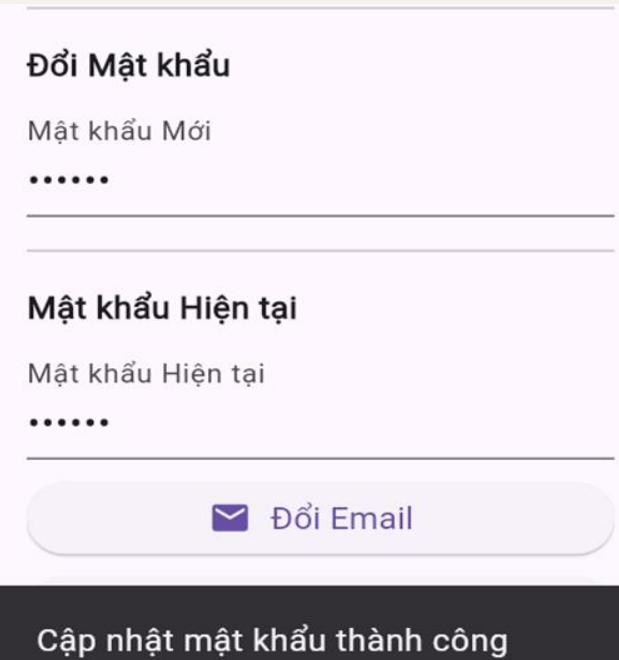
- Hàm updatePassword() có chức năng chính là thay đổi mật khẩu .
- Yêu cầu người dùng nhập mật khẩu hiện tại và mật khẩu mới
- Khai báo: `Future<void> updatePassword() async {`

```
    final user = currentUser;
    if (user == null) return;
    final newPassword = newPassCtrl.text.trim();
    if (newPassword.length < 6) {
        showSnackBar("Mật khẩu mới phải có ít nhất 6 ký tự");
        return;
    }
    await reauthenticate();
    await user.updatePassword(newPassword);
    showSnackBar("Cập nhật mật khẩu thành công");
}
```

3.3 Thay đổi mật khẩu

- Hàm updatePassword() có chức năng chính là thay đổi mật khẩu .
- Yêu cầu người dùng nhập mật khẩu hiện tại và mật khẩu mới
- Khai báo và kết quả:

```
Future<void> updatePassword() async {  
    final user = currentUser;  
    if (user == null) return;  
    final newPassword = newPassCtrl.text.trim();  
    if (newPassword.length < 6) {  
        showSnackBar("Mật khẩu mới phải có ít nhất 6 ký tự");  
        return;  
    }  
    await reauthenticate();  
    await user.updatePassword(newPassword);  
    showSnackBar("Cập nhật mật khẩu thành công");  
}
```



3.3 Thay đổi Email

- Hàm updateEmail() có chức năng chính là thay đổi địa chỉ email đăng nhập.
- Yêu cầu người dùng nhập mật khẩu hiện tại , xác minh từ đường dẫn được gửi tới email mới.
- Khai báo :*

```
Future<void> updateEmail() async {  
    final user = currentUser;  
    if (user == null) return;  
    await reauthenticate();  
    await user.verifyBeforeUpdateEmail(emailCtrl.text.trim());  
    showSnackBar("Email xác minh đã được gửi.");  
}
```

3.3 Thay đổi mật khẩu

- Nhập email mới và mật khẩu hiện tại sau đó nhấn đổi Email.
- Một link xác nhận sẽ được gửi về email mới.

Verify your email for project-374592173250

Unknown Sender

Từ: noreply@assistant-fe83c.firebaseio.com

6 giờ tới

Tới: test123@tempmail.id.vn

Hello Nam Phuong,

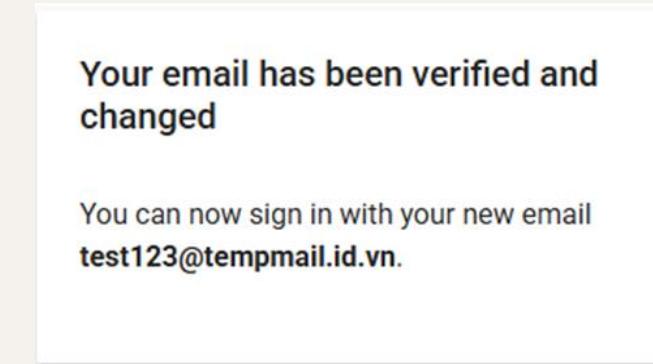
Follow this link to verify your email address.

https://assistant-fe83c.firebaseio.com/_auth/action?mode=verifyAndChangeEmail&oobCode=f3oalR5wcERWZoYxbE3OtqCmhtyEkQG4uKajQ2PG4-sAAAGabKz8bA&apiKey=AIzaSyAHwnm-1ar63uQsvthBQTfDcUn43ONUp1s&lang=en

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your project-374592173250 team



Hồ sơ người dùng

Cập nhật hồ sơ

Đổi Email

Email Mới
test123@tempmail.id.vn

Đổi Mật khẩu

Mật khẩu Mới

Mật khẩu Hiện tại

Mật khẩu Hiện tại

Đổi Email

Email xác minh đã được gửi.

4. Các Nguyên Tắc Bảo Mật

Biện pháp bảo mật	Mô tả
Kiểm tra người dùng đã đăng nhập	Trước khi cập nhật hồ sơ như tên, ảnh, email hay mật khẩu, mã luôn kiểm tra if (user == null)
Xác thực	Khi đổi email hoặc mật khẩu , Firebase yêu cầu xác thực lại bằng mật khẩu cũ
Gửi email xác minh	Người dùng cần xác nhận qua email trước khi hệ thống chính thức thay đổi email mới
Đặt quy tắc cho mật khẩu	Thêm các quy tắc để tránh đặt mật khẩu yếu
Ẩn mật khẩu khi nhập	Sử dụng obscureText: true trong TextField nhập mật khẩu giúp ẩn ký tự , tránh bị nhìn trộm khi đang nhập liệu.

Tài liệu tham khảo

- [1] Google, *Firebase Authentication Documentation*. Accessed: Nov. 10, 2025. [Online]. Available: <https://firebase.google.com/docs/auth>
- [2] Google, *Firebase Authentication - Manage Users*. Accessed: Nov. 10, 2025. [Online]. Available: <https://firebase.google.com/docs/auth/users>
- [3] Google, *Firebase Authentication - Reauthenticate a User Before Sensitive Operations*. Accessed: Nov. 10, 2025. [Online]. Available: https://firebase.google.com/docs/auth/web/manage-users#re-authenticate_a_user
- [4] Google, *Firebase Authentication - Update User Email and Password*. Accessed: Nov. 10, 2025. [Online]. Available: https://firebase.google.com/docs/auth/web/manage-users#update_a_users_email_address
- [5] Flutter Team, *FlutterFire Documentation - Firebase for Flutter*. Accessed: Nov. 10, 2025. [Online]. Available: <https://firebase.flutter.dev/docs/overview>
- [6] Flutter Team, *Handling Forms and User Input in Flutter*. Accessed: Nov. 10, 2025. [Online]. Available: <https://docs.flutter.dev/cookbook/forms/validation>
- [7] Dart SDK, *TextEditingController Class Reference*. Accessed: Nov. 10, 2025. [Online]. Available: <https://api.flutter.dev/flutter/widgets/TextEditingController-class.html>
- [8] Google, *Firebase Security Best Practices Guide*. Accessed: Nov. 10, 2025. [Online]. Available: <https://firebase.google.com/support/guides/security-checklist>
- [9] Nguyễn Duy Nhật Viễn, *Lập Trình Đa Nền Tảng*, Khoa Điện Tử Viễn Thông – Đại học Bách Khoa, Trường Đại học Đà Nẵng, 2025. [Lecture Slides].
- [10] Flutter Community, “Authentication in Flutter with Firebase Auth,” Medium, Nov. 2023. [Online]. Available: <https://medium.com/flutter-community/firebase-authentication-in-flutter-dc4f7f7a8c26>

Bảng Phân Công Nhiệm Vụ

STT	HỌ VÀ TÊN	NHIỆM VỤ	KHỐI LƯỢNG
01	Lê Xuân Nam	<ul style="list-style-type: none">- User management và profile- Security best practices	50%
02	Nguyễn Nam Phương	<ul style="list-style-type: none">- Email/password authentication- Social login (Google, Facebook)	50%