

**TRƯỜNG ĐẠI HỌC PHÚ YÊN**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

****

**BÀI TIỂU LUẬN**

**LẬP TRÌNH DI ĐỘNG**

**Đề tài:**

**XÂY DỰNG ỨNG DỤNG THI TRẮC NGHIỆM TRÊN MOBILE**

**GVHD : Th.s Nguyễn Quốc Dũng**

**Sinh viên thực hiện :** **Cao Thị Út**

**Lớp : DC17CTT01**

***Phú Yên, 2022***

**MỤC LỤC**

[MỞ ĐẦU 4](#_Toc46858768)

[GIỚI THIỆU ĐỀ TÀI 5](#_Toc46858769)

[1. Lý do chọn đề tài 5](#_Toc46858770)

[2. Mục đích của đề tài 5](#_Toc46858771)

[3. Đối tượng và phương pháp nghiên cứu 5](#_Toc46858772)

[3.1. Đối tượng nghiên cứu 5](#_Toc46858773)

[3.2. Phương pháp nghiên cứu 6](#_Toc46858774)

[4. Phạm vi đề tài 6](#_Toc46858775)

[5. Nội dung thực hiện 6](#_Toc46858776)

[CHƯƠNG 1: HỆ ĐIỀU HÀNH ANDROID 7](#_Toc46858777)

[1.1. Giới thiệu về hệ điều hành Android 7](#_Toc46858778)

1.[2. Môi trường lập trình Android 8](#_Toc46858779)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc46858780)

[2.1. Flutter 9](#_Toc46858781)

[2.1.1. Giới thiệu về Flutter 9](#_Toc46858782)

[2.1.2 Mô hình cấu tạo nên một project Flutter 11](#_Toc46858783)

[2.1.3. Ưu điểm của Flutter 12](#_Toc46858784)

[2.1.4. Các thành phần quan trọng trong Flutter 13](#_Toc46858785)

[2.2. Dart 14](#_Toc46858786)

[2.2.1. Giới thiệu ngôn ngữ Dart 14](#_Toc46858787)

[2.2.2. Đặc điểm của ngôn ngữ Dart 16](#_Toc46858788)

[2.2.3. Những ưu điểm của ngôn ngữ Dart 16](#_Toc46858788)

[2.3. Firebase Realtime 17](#_Toc46858788)

[2.3.1. Giới thiệu 17](#_Toc46858788)

[2.3.2. Những thành phần trong Firebase Realtime 18](#_Toc46858788)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 20](#_Toc46858780)

[3.1. Thiết kế cơ sở dữ liệu 20](#_Toc46858781)

[3.1.1. Sơ đồ thực thể liên kết 20](#_Toc46858781)

[3.2. Thiết kế cơ sở dữ liệu 21](#_Toc46858781)

[3.2.1 Bảng questions 21](#_Toc46858781)

[3.2.2 Bảng users 21](#_Toc46858781)

[3.2.3 Bảng topics 21](#_Toc46858781)

[3.3. Quản trị hệ thống Firebase 22](#_Toc46858781)

[CHƯƠNG 4: CHƯƠNG TRÌNH DEMO APP THI TRẮC NGHIỆM 23](#_Toc46858789)

[4.1. Giao diện đăng ký 23](#_Toc46858790)

[4.2. Giao diện đăng nhập tài khoản 24](#_Toc46858791)

[4.3. Giao diện trang chủ 25](#_Toc46858792)

[4.4. Giao diện tạo bộ đề thi 26](#_Toc46858792)

[4.5. Giao diện tạo câu hỏi 27](#_Toc46858792)

[4.6. Giao diện thi trắc nghiệm 28](#_Toc46858792)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29](#_Toc46858793)

**DANH MỤC SƠ ĐỒ**

[Hình 2.1 Logo Flutter 9](#_Toc46858816)

[Hình 2.2 Mô hình cấu tạo nên một project Flutter 11](#_Toc46858817)

[Hình 2.3 Logo Dart 14](#_Toc46858818)

[Hình 2.4 Logo Firebase 17](#_Toc46858820)

[Hình 3.1 Lượt đồ EER Diagram 20](#_Toc46858820)

[Hình 4.1 Giao diện đăng ký người dùng 23](#_Toc46858820)

[Hình 4.2 Giao diện đăng nhập tài khoản 24](#_Toc46858820)

[Hình 4.3 Giao diện trang chủ 25](#_Toc46858820)

[Hình 4.4 Giao diện tạo bộ đề thi 26](#_Toc46858820)

[Hình 4.5 Giao diện tạo câu hỏi cho bộ đề thi 27](#_Toc46858820)

[Hình 4.6 Giao diện thi trắc nghiệm 28](#_Toc46858820)

MỞ ĐẦU

Ngày nay, công nghệ thông tin ngày càng phát triển thì nhu cầu cuộc sống cũng như việc ứng dụng Công nghệ thông tin vào công việc của con người được phát triển mạnh mẽ và ngày càng đa dạng hơn. Dễ thấy nhất là sự xuất hiện và phát triển của những chiếc smartphone trong nhiều năm gần đây. Smartphone ngày càng được sử dụng rất phổ biến trong cuộc sống, nó không chỉ được dùng với mục đích chính là nghe gọi nữa mà smartphone còn được sử dụng nhiều các ứng dụng hiện đại khác mà nó mang lại.

Cùng với đó ứng dụng thi trắc nghiệm phù hợp nhiều đối tượng khác nhau. Ứng dụng giúp việc kiểm tra trở nên đơn giản, thú vị hơn. Người dùng hoàn toàn có thể dễ dàng tạo dựng bộ câu hỏi của riêng mình để thử thách, khám phá cùng với mọi người. Chính vì thế chúng em đã chọn đề tài: *“Xây dựng ứng dụng thi trắc nghiệm trên mobile”* để giúp người dùng sử dụng dễ dàng tạo bộ trắc nghiệm đơn giản, dễ dàng trực quan và miễn phí.

Chúng em chân thành cảm ơn thầy Nguyễn Quốc Dũng đã tận tình hướng dẫn và chỉ bảo em có thể hoàn thành tốt tiểu luận này!

GIỚI THIỆU ĐỀ TÀI

1. Lý do chọn đề tài

Trong những năm gần đây, sự phát triển của công nghệ thông tin đã và đang làm thay đổi toàn bộ thế giới. Mọi ngành nghề, mọi lĩnh vực như kinh tế, y khoa, công nghiệp,… dần dần được tin học hóa và làm cho công việc trở nên dễ dàng, nhanh chóng, chính xác hơn. Đặc biệt trong công tác giáo dục, việc tin học hóa góp phần nâng cao chất lượng dạy và học. Trong khoa cử cũng vậy, để đảm bảo chất lượng của một kỳ thi, tính khách quan, chính xác và khoa học phải được đặt lên hàng đầu.

Sự kết hợp giữa phương pháp thi trắc nghiệm và tin học không những đáp ứng được các yếu tố đó mà còn giúp tiết kiệm chi phí tổ chức thi, thời gian và công sức đồng thời kết quả lại nhanh chóng và độ chính xác cao.

Chính vì vậy nhóm chúng em xin thực hiện đề tài “**Xây dựng ứng dụng thi trắc nghiệm trên mobile**”.

2. Mục đích của đề tài

Sử dụng ngôn ngữ lập trình Dart, Flutter và Firebase để xây dựng ứng dụng câu đố.

3. Đối tượng và phương pháp nghiên cứu

3.1. Đối tượng nghiên cứu

- Ngôn ngữ lập trình hướng đối tượng Dart, SDK Flutter, cơ sở dữ liệu Firebase.

- Github.

- Ứng dụng câu đố.

3.2. Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết, nghiên cứu tài liệu chuyên ngành liên quan đến đề tài thực hiện.

4. Phạm vi đề tài

Ứng dụng thi trắc nghiệm.

5. Nội dung thực hiện

Chương 1: Hệ điều hành Android

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích và thiết kế cơ sở dữ liệu

Chương 4: Chương trình demo app thi trắc nghiệm

CHƯƠNG 1: HỆ ĐIỀU HÀNH ANDROID

1.1. Giới thiệu về hệ điều hành Android

Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Các thành phần tạo nên một ứng dụng Android được chia làm 6 loại bao gồm:

- Activity:  là nền của 1 ứng dụng. Khi khởi động 1 ứng dụng Android nào đó thì bao giờ cũng có một main Activity được gọi, hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác.

- Service: là thành phần chạy ẩn trong Android. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.

- Content Provider: kho dữ liệu chia sẻ. Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng.

- Intent: nền tảng để truyền tải các thông báo. Intent được sử dụng để gửi các thông báo đi nhằm khởi tạo một Activity hay Service để thực hiện công việc bạn mong muốn.

- Broadcast Receiver: thành phần thu nhận các Intent bên ngoài gửi tới.

- Notification: đưa ra các cảnh báo mà không làm cho các Activity phải ngừng hoạt động.

Activity, Service, Broadcast Receiver và Content Provider mới là những thành phần chính cấu thành nên ứng dụng Android, bắt buộc phải khai báo trong file AndroidManifest.xml

1.2. Môi trường lập trình Android

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android studio dựa vào IntelliJ IDEA, là một IDE tốt cho nhất Java hiện nay. Do đó, Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho Android.

**Yêu cầu phần cứng của máy tính**

Microsoft® Windows® 8/7 (32 or 64-bit)

4 GB RAM. (Tốt nhất là 8GB)

400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator system images và caches

Độ phân giải tối thiếu 1280 x 800

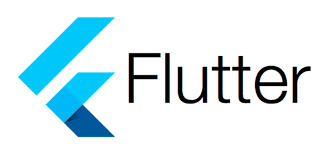
Java Development Kit ()

**Máy ảo Android**

Là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Trong Android Studio có cung cấp cho chúng ta một máy ảo Android mặc định là Android Virtual Device viết tắt là AVD.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## **2.1. Flutter**



Hình 2.1 Logo Flutter

## **2.1.1. Giới thiệu về Flutter**

Flutter là một Framework viết trên nền ngôn ngữ Dart. Flutter được sinh ra như một Cross-platform framework nhưng khác với những Cross-platform hiện tại, Flutter viết mã và build ra các đoạn mã thực thi tương ứng trên các thiết bị khác nhau thay vì build ra thành các đoạn mã được tối ưu và chạy trên những môi trường trung gian.

Flutter được viết chia làm hai tầng. Tầng ở trên sử dụng ngôn ngữ cung cấp các đoạn mã xây dựng lên một ứng dụng Flutter. Các đoạn mã này cung cấp phương tiện để có thể thay đổi và chỉnh sửa chúng. Từ đó giúp ứng dụng của lập trình viên có thể được tùy chỉnh theo mong muốn. Tầng Application này giúp lập trình viên thay đổi mã nguồn ứng dụng ở thời điểm compile time. Tầng thứ hai của Flutter nằm ở sâu bên dưới và được viết bằng C++. Tầng Shell này chứa các công tụ trợ giúp ứng dụng Flutter trong quá trình chạy. Nổi bật cần lưu ý ở tầng này là máy ảo Dart VM. Khái niệm máy ảo là khái niệm về một ứng dụng chạy song song với mã nguồn chính như một phần của ứng dụng.

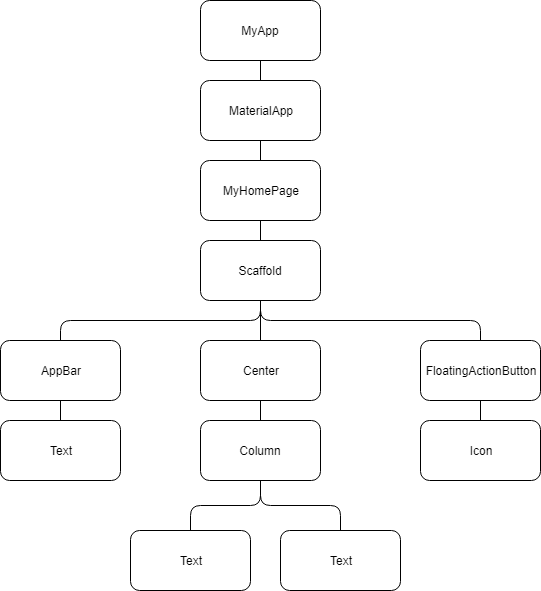
Flutter là SDK dành cho thiết bị di động của Google để tạo ra các giao diện native chất lượng cao trên iOS và Android trong thời gian ngắn. Flutter làm việc với source code có sẵn, được sử dụng bởi các nhà phát triển và các tổ chức trên khắp thế giới, đồng thời nó open-source và miễn phí.

Flutter có các công cụ và thư viện để dễ dàng đưa ý tưởng của mình vào cuộc sống trên iOS và Android. Nếu chưa có kinh nghiệm phát triển trên thiết bị di động, thì Flutter là một cách dễ dàng và nhanh chóng để xây dựng các ứng dụng di động tuyệt đẹp. Là một nhà phát triển iOS hoặc Android có kinh nghiệm, có thể sử dụng Flutter cho các View và tận dụng nhiều code Java / Kotlin / ObjC / Swift hiện có.

Flutter sử dụng Dart, một ngôn ngữ nhanh, hướng đối tượng với nhiều tính năng hữu ích như mixin, generic, isolate, và static type.

Flutter có các thành phần UI của riêng nó, cùng với một cơ chế để kết xuất chúng trên nền tảng Android và iOS. Hầu hết các thành phần giao diện người dùng, đều sẵn dùng, phù hợp với các nguyên tắc của Material Design.

## **2.1.2. Mô hình cấu tạo nên một project Flutter**



Hình 2.2 Mô hình cấu tạo nên một project Flutter

* My App: là widget do mình tạo và được xây dựng bởi một widget gốc

Flutter – MateralApp.

* Material App: có một thuộc tính home để chỉ định giao diện của trang chính và được xây dựng bởi một widget do mình tạo – MyHomePage.
* MyHomePage: được xây dựng bởi một widget gốc khác của flutter – Scaffold.
* Scafold: có hai thuộc tính body và appBar.
* Body: chứa giao diện chính và appBar chứa giao diện phần header.
* Header UI: là một widget gốc của flutter, appBar và Body UI sử dụng Center widger.
* Center widget có một thuộc tính child chứa nội dung chính là một Text widget.
* Column và Row: Column là một widget hiển thị các widget con của nó trên một cột. Còn Row hiển thị các widget con của nó trên một hàng.
* FloatingActionButton: là một nút biểu tượng hình tròn di chuột qua nội dung để thúc đẩy một hành động chính trong ứng dụng. Các nút tác vụ nổi thường được sử dụng nhất trong trường Scaffold.floatingActionButton.

## **2.1.3. Ưu điểm của Flutter**

Fast Development: Tính năng Hot Reload hoạt động trong milliseconds để hiện thị giao diện. Sử dụng tập hợp các widget có thể customizable để xây dựng giao diện trong vài phút. Ngoài ra Hot Reload còn giúp thêm các tính năng, fix bug tiết kiệm thời gian hơn mà không cần phải thông qua máy ảo, máy android hoặc iOS.

Expressive and Flexible UI: Có rất nhiều các thành phần để xây dựng giao diện của Flutter vô cùng đẹp mắt theo phong cách Material Design và Cupertino, hỗ trợ nhiều các APIs chuyển động, smooth scrolling...

Native Performance: Các widget của fluter kết hợp các sự khác biệt của các nền tảng ví dụ như scrolling, navigation, icons, font để cung cấp một hiệu năng tốt nhất tới iOS và Android.

## **2.1.4. Các thành phần quan trọng trong Flutter**

* **Widget**

Widget chịu trách nhiệm hình thành cấu trúc của Diagnosticable Tree (Cây chuẩn đoán là một cấu trúc dữ liệu dạng cây hỗ trợ việc định nghĩa cấu trúc giao diện được vẽ lên trong ứng dụng viết bằng Flutter) trong khi class element chịu trách nhiệm quản lý trạng thái của từng widget trên cái cây đó. Tư tưởng cơ bản của Flutter tương tự như một Web Application. Tất cả trong một. Flutter căng ra một mảnh vải là canvas và vẽ tất cả mọi thứ lên đó. Khi trạng thái của dữ liệu thay đổi ví dụ như người dùng chuyển màn hình hay thay đổi dữ liệu trên màn hình được phản ánh thông qua việc xóa đi các widget cũ và vẽ lên các widget mới. Chính vì vậy thay vì phải biết đến các StoryBoard (trong iOS) hay các Activity (trong Android), Flutter đưa tất cả các khái niệm liên quan đến giao diện về một khái niệm duy nhất gọi là widget. Việc quản lý trạng thái của các widget cũng được đưa về cho lập trình viên thực hiện. Điểm khác biệt này cũng đặc biệt hơn khi Dart là ngôn ngữ Component Based. Trong Flutter, khi muốn điều chỉnh trạng thái về kích thước hay vị trí của một widget đó, ta không thực hiện việc đó trong widget hiện tại mà được khuyến khích bọc widget đó trong một widget nào đó chỉ chuyên làm việc này. Điều này khiến cho tính chất "Trong Flutter thì mọi thứ đều là widget " lại càng chuẩn hơn bao giờ hết. Màn hình là widget, thành phần giao diện cũng là widget, thậm chí cả thông tin về layout cũng là widget...

* **StatefulWidget**

Là widget mà trạng thái của chúng có thể thay đổi. Để tạo state thì statefulwidget phải gọi method createState();

Khi tạo mới một StatefulWidget, ta quan tâm đến hai class là StatefulWidget và State của Widget đó. StatefulWidget là phần không thay đổi trong Widget Tree, thay vào đó, State là phần được thay đổi.

Mỗi khi Widget bị làm bẩn, State của Widget đó sẽ được khởi tạo lại thông qua việc gọi đến phương thức StatefulWidget. Thuộc tính này chính là instance của StatefulWidget đã tạo ra State đó. Chính nhờ vậy, từ trong State hoàn toàn có thể lấy được những giá trị được truyền vào bên trong StatefulWidget thay vì phải truyền vào thông qua hàm khởi tạo.

* **StatelessWidget**

Là các widget không chứa State. Tất cả các giá trị của StatelessWidget đều là final → không thể thay đổi trong thời gian chạy, do đó StatelessWidget chỉ hiển thị những gì được truyền vào trong contructor. Ngược lại, để thay đổi trạng thái của widget ta dùng **statefulwidget.**

## **2.2. Dart**



Hình 2.3 Logo Dart

## **2.2.1. Giới thiệu ngôn ngữ Dart**

Dart là ngôn ngữ lập trình dạng hướng đối tượng. Cũng là một ngôn ngữ viết ở một lần dùng ở nhiều nơi nhưng khác với Java, Dart thay vì tạo ra những môi trường trung gian giữa code thực thi và môi trường thiết bị thì Dart sử dụng những trình biên dịch khác nhau để biên dịch ra mã máy tương ứng. Hiện tại Dart đang hỗ trợ để tạo ra những ứng dụng trên iOS, Android, Fuchsia và Web.

Dart đồng thời cũng là một ngôn ngữ hỗ trợ asynchrony. Tức nghĩa là trong Dart chỉ có một dòng chảy tuần tự chảy từ trên xuống dưới. Code được đưa vào hàng đợi và hoàn toàn có thứ tự khi được thực thi.

* Dart là ngôn ngữ lập trình mới, được phát triển bởi Google, hiện đã được chấp thuận bởi tổ chức Ecma.
* Dart được sử dụng để xây dựng các loại ứng dụng: web, server, di động (IOS và Android với công cụ Flutter).
* Dart là ngôn ngữ hỗ trợ lập trình hướng đối tượng, cú pháp kiểu C, mã code Dart có thể biên dịch thành JavaSript để chạy trên trình duyệt. Nó hỗ trợ những khái niệm lập trình hiện đại như giao diện lớp, lớp trừu tượng ...
* Từ ngôn ngữ lập trình Dart, Google giới thiệu SDK Flutter để sử dụng ngôn ngữ Dart phát triển ứng dụng di động chạy đa nền tảng (Một lần viết code build ra ứng dụng cho ISO và Android).
* **Tại sao lại lựa chọn ngôn ngữ Dart?**

Các nhà phát triển tại Google và các nơi khác sử dụng Dart để tạo các ứng dụng chất lượng cao, quan trọng cho iOS, Android và web. Với các tính năng nhắm đến sự phát triển phía khách hàng, Dart rất phù hợp cho cả ứng dụng di động và web. Dart tạo ra những trải nghiệm đẹp, chất lượng cao trên tất cả các màn hình, với:

* Một ngôn ngữ được tối ưu hóa cho client
* Framework mạnh mẽ
* Công cụ linh hoạt

## **2.2.2. Đặc điểm của ngôn ngữ Dart**

* Dart đi kèm với trình soạn thảo nhẹ, có thể dung Dart Editor để viết, chạy và gỡ lỗi các ứng dụng Dart, Dart editor có thể phát hiện các lỗi tiềm năng.
* Viết code tốt và nhanh hơn.
* Dart được thiết kế để biên dịch sang JavaScript.

## **2.2.3. Những ưu điểm của ngôn ngữ Dart**

* + **Năng suất** Cú pháp Dart rõ ràng và súc tích, công cụ của nó đơn giản nhưng mạnh mẽ. Type-safe xác định sớm các lỗi tinh tế. Dart có các thư viện cốt lõi và một hệ sinh thái gồm hàng ngàn package.
  + **Nhanh** Dart cung cấp tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.
  + **Di động** Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript.
  + **Dễ gần** Dart quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên của nó. Nếu đã biết C ++, C # hoặc Java, có thể làm việc hiệu quả với Dart chỉ sau vài ngày.
  + **Reactive** Dart rất phù hợp với lập trình Reactive, với sự hỗ trợ để quản lý các đối tượng tồn tại trong thời gian ngắn, chẳng hạn như các widget UI, thông qua phân bổ đối tượng nhanh và GC. Dart hỗ trợ lập trình không đồng bộ thông qua các tính năng ngôn ngữ và API sử dụng các đối tượng Future và Stream.

## **2.3. Firebase Realtime**



Hình 2.4 Logo Firebase

## **2.3.1. Giới thiệu**

**Firebase** là một dịch vụ hệ thống backend được Google cung cấp sẵn cho ứng dụng Mobile, với Firebase có thể rút ngắn thời gian phát triển, triển khai và thời gian mở rộng quy mô của ứng dụng mobile mình đang phát triển. Hỗ trợ cả 2 nền tảng Android và IOS, Firebase mạnh mẽ, đa năng, bảo mật và là dịch vụ cần thiết đầu tiên để xây dưng ứng dụng với hàng triệu người sử dụng. Sử dụng Firebase sẽ có được hưởng các lợi ích sau:

* Xây dựng ứng dụng nhanh chóng mà không tốn thời gian, nhân lực để quản lý hệ thống và cơ sơ sở hạ tầng phía sau: Firebase cung cấp chức năng như phân tích, cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cố lỗi để có thể dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm đem lại các trải nghiệm tốt nhất cho họ.
* Uy tín chất lượng đảm bảo từ Google: Firebase được google hỗ trợ và cung cấp trên nền tảng phần cứng với quy mô rộng khắp thế giới, được các tập đoàn lớn và các ưng dụng với triệu lượt sử dụng từ người dùng.
* Quản lý cấu hình và trải nghiệm các ứng dụng của Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.

## **2.3.2. Những thành phần trong Firebase Realtime**

* [**Crashlytics**](https://firebase.google.com/products/crashlytics/)**:**

Hệ thống theo dõi và lưu trữ thông tin lỗi của ứng dụng đang chạy trên máy người dùng. Các thông tin lỗi này được thu thập một các toàn diện và ngay tức thời. Cách trình bày hợp lý với từng chu trình hoạt động đến khi xảy ra lỗi, các báo cáo trực quan giúp người phát triển có thể nắm bắt và xử lý kịp thời các lỗi chính của ứng dụng.

* [**Cloud Firestore:**](https://firebase.google.com/products/firestore/)

Lưu trữ và đồng bộ dữ liệu giữa người dùng và thiết bị – ở quy mô toàn cầu – sử dụng cơ sở dữ liệu noSQL được lưu trữ trên hạ tầng cloud. Cloud Firestore cung cấp tính năng đồng bộ hóa trực tuyến và ngoại tuyến cùng với các truy vấn dữ liệu hiệu quả. Tích hợp với các sản phẩm Firebase khác cho phép xây dựng các ứng dụng thực sự ngay cả khi kết nối internet bị gián đoạn.

* [**Authentication**](https://firebase.google.com/products/auth/):

Quản lý người dùng một cách đơn giản và an toàn. Firebase Auth cung cấp nhiều phương pháp để xác thực, bao gồm email và mật khẩu, các nhà cung cấp bên thứ ba như Google hay Facebook, và sử dụng trực tiếp hệ thống tài khoản hiện tại. Xây dựng giao diện của riêng mình hoặc tận dụng lợi thế của mã nguồn mở, giao diện người dùng tùy biến hoàn toàn.

* [**Cloud Functions**](https://firebase.google.com/products/functions/):

Mở rộng ứng dụng của mình bằng mã phụ trợ tùy chỉnh mà không cần quản lý và quy mô các máy chủ. Các chức năng có thể được kích hoạt bởi các sự kiện, được phát sinh ra bởi các sản phẩm Firebase, dịch vụ Google Cloud hoặc các bên thứ ba có sử dụng webhooks.

* [**Cloud Storage**](https://firebase.google.com/products/storage/):

Lưu trữ và chia sẻ nội dung do người dùng tạo ra như hình ảnh, âm thanh và video với bộ nhớ đối tượng mạnh mẽ, đơn giản và tiết kiệm chi phí được xây dựng cho quy mô của Google. Các Firebase SDK cho Cloud Storage thêm tính năng bảo mật của Google để tải lên và tải tệp cho các ứng dụng Firebase của mình, bất kể chất lượng mạng.

* [**Hosting**](https://firebase.google.com/products/hosting):

Đơn giản hóa lưu trữ web với các công cụ được thực hiện cụ thể cho các ứng dụng web hiện đại. Khi tải lên nội dung web, sẽ tự động đẩy chúng đến CDN toàn cầu và cung cấp chứng chỉ SSL miễn phí để người dùng có được trải nghiệm an toàn, đáng tin cậy, độ trễ thấp, dù họ ở đâu.

* [**Test Lab for Android**](https://firebase.google.com/products/test-lab/):

Chạy thử nghiệm tự động và tùy chỉnh cho ứng dụng trên các thiết bị ảo và vật lý do Google cung cấp. Sử dụng Firebase Test Lab trong suốt vòng đời phát triển để khám phá lỗi và sự không nhất quán để có thể cung cấp một trải nghiệm tuyệt vời trên nhiều thiết bị.

* [**Performance Monitoring**](https://firebase.google.com/products/performance/):

Chẩn đoán các vấn đề về hiệu suất ứng dụng xảy ra trên thiết bị của người dùng. Sử dụng dấu vết để theo dõi hiệu suất của các phần cụ thể trong ứng dụng và xem chế độ xem tổng hợp trong bảng điều khiển Firebase. Luôn cập nhật thời gian khởi động của ứng dụng và theo dõi các yêu cầu HTTP mà không cần viết bất kỳ mã nào.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU**

3.1. Thiết kế cơ sở dữ liệu

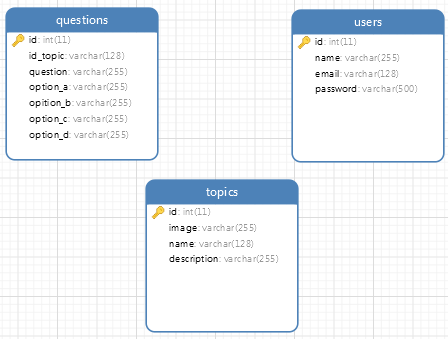
3.1.1. Sơ đồ thực thể liên kết

Cơ sở dữ liệu có 3 bảng gồm: questions, users, topics

questions (id, id\_topic, question, option\_a, option\_b, option\_c, option\_d)

topics (id, image, name, description)

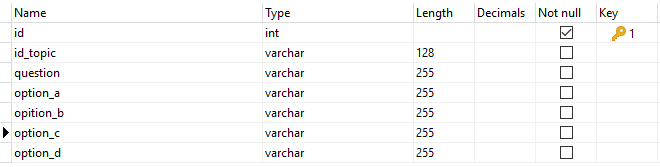
users (id, name, email, password)



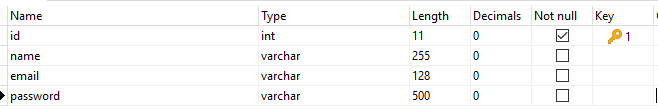
*Hình 3.1 Lược đồ EER Diagram*

3.2. Thiết kế cơ sở dữ liệu

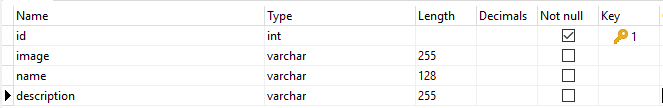
**3.2.1 Bảng questions**



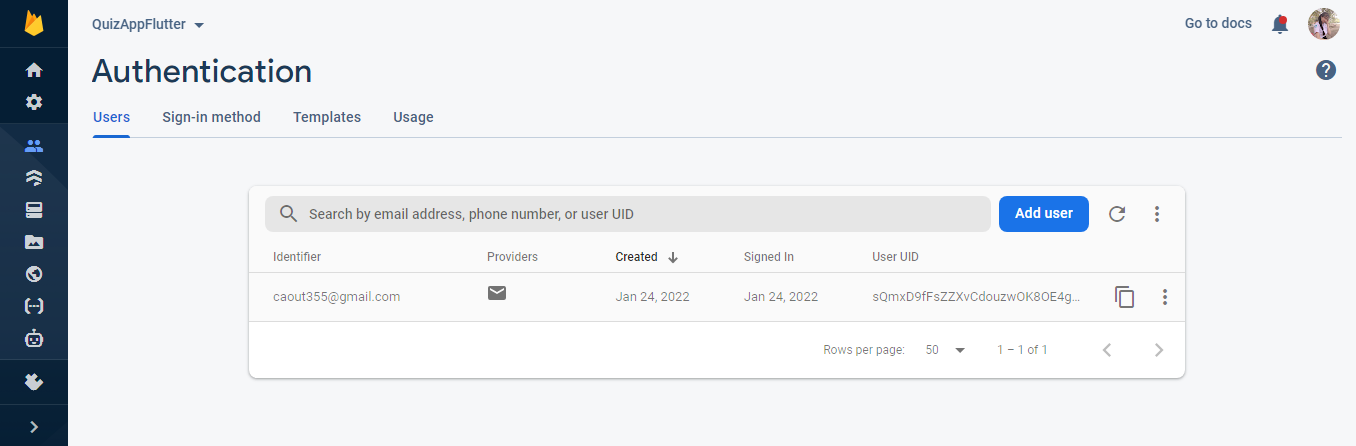
**3.2.2 Bảng users**



**3.2.3 Bảng topics**



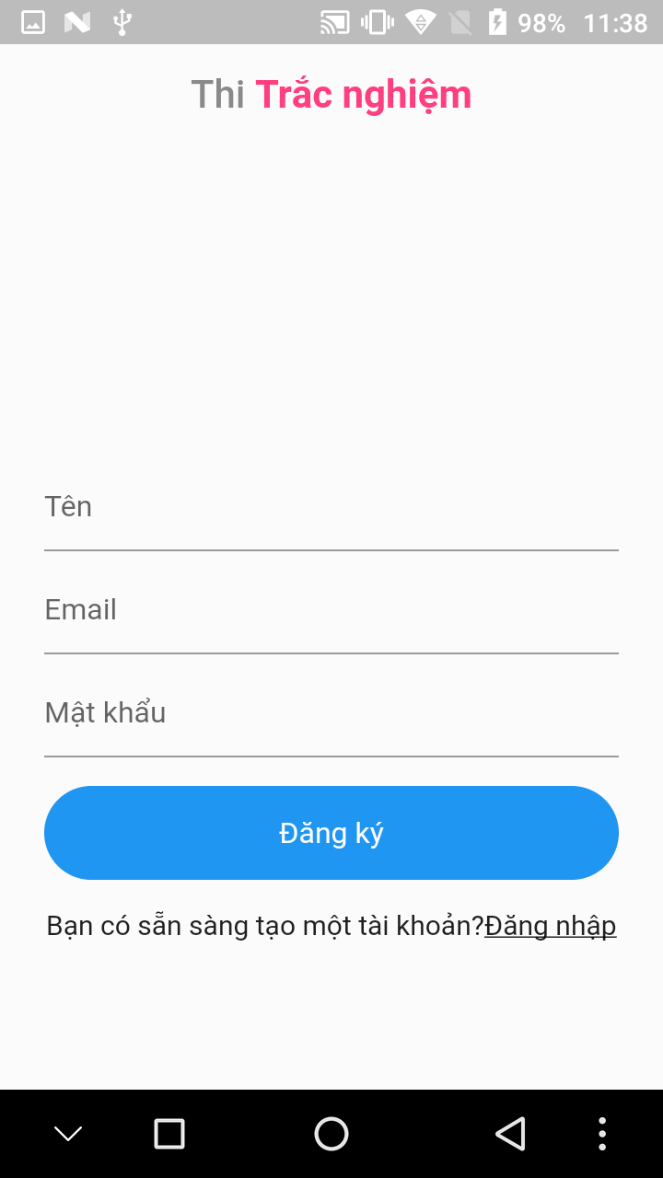
3.3. Quản trị hệ thống Firebase



Admin có thể xem thông tin người dùng, có thể thao tác lên các tài khoản như: Đặt lại mật khẩu, vô hiệu hóa tài khoản và xóa tài khoản đó đi.

CHƯƠNG 4: CHƯƠNG TRÌNH DEMO APP THI TRẮC NGHIỆM

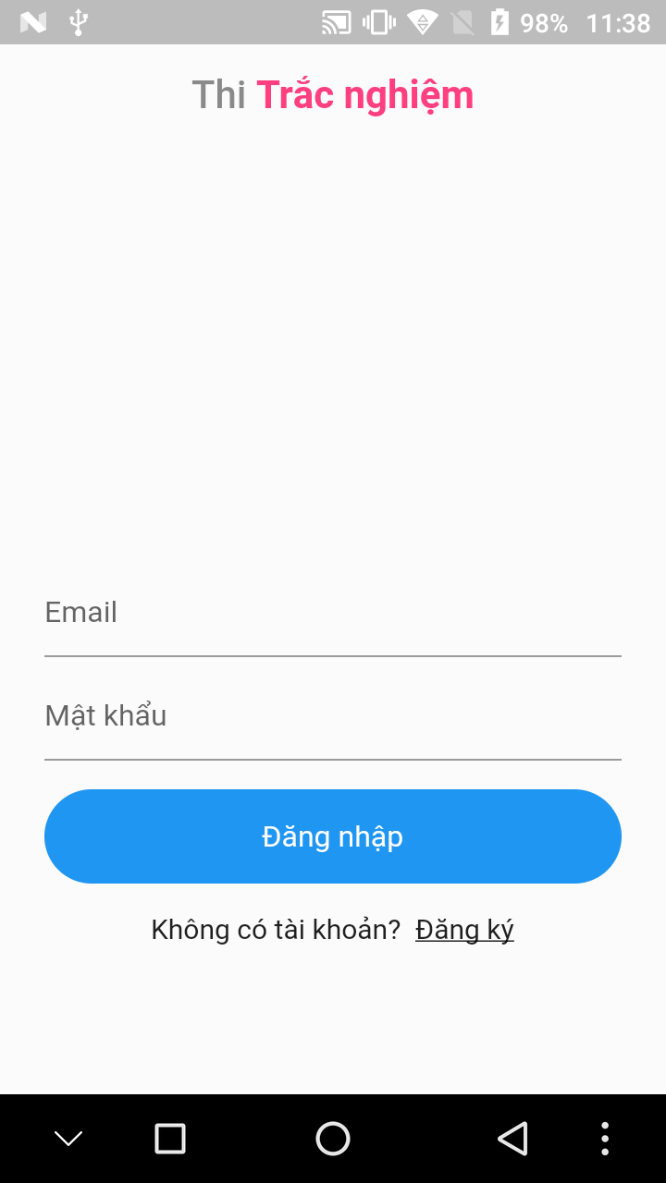
4.1. Giao diện đăng ký



*Hình 4.1 Giao diện đăng ký người dùng*

Tại màn hình Đăng ký nhập các thông tin như Tên người dùng, Email, Mật khẩu để đăng ký tài khoản. Lưu ý: Nhập đầy đủ thông tin yêu cầu, nếu thông tin thiếu hoặc sai thì hệ thống sẽ báo lỗi và không đăng ký tài khoản mới được. Sau khi nhập đầy đủ thông tin chọn “Đăng ký” để đăng ký tài khoản mới.

4.2. Giao diện đăng nhập tài khoản



*Hình 4.2 Giao diện đăng nhập tài khoản*

Tại giao diện màn hình Đăng nhập, điền thông tin tài khoản: Email và Mật khẩu để đăng nhập vào hệ thống. Lưu ý: Nhập đầy đủ thông tin tài khoản. Nếu nhập sai hoặc không đầy đủ thông tin thì hệ thống sẽ báo lỗi và không đăng nhập vào hệ thống được. Sau đó chọn “Đăng nhập” để vào hệ thống nếu thông tin tài khoản là đúng.

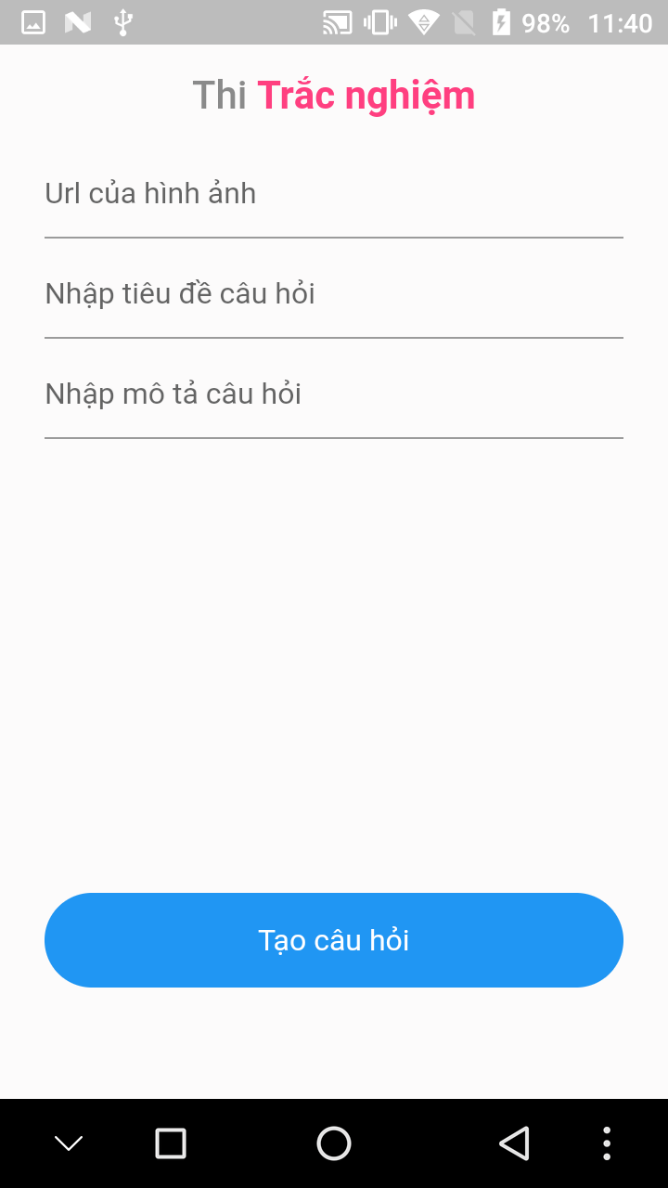
4.3. Giao diện trang chủ



*Hình 4.3 Giao diện trang chủ*

Giao diện này sẽ hiển thị các bộ đề thi đã tạo, người dùng sẽ chọn cho mình bộ đề thi để làm kiểm tra. Để tạo thêm bộ đề thi người dùng nhấn vào biểu tượng  để tạo.

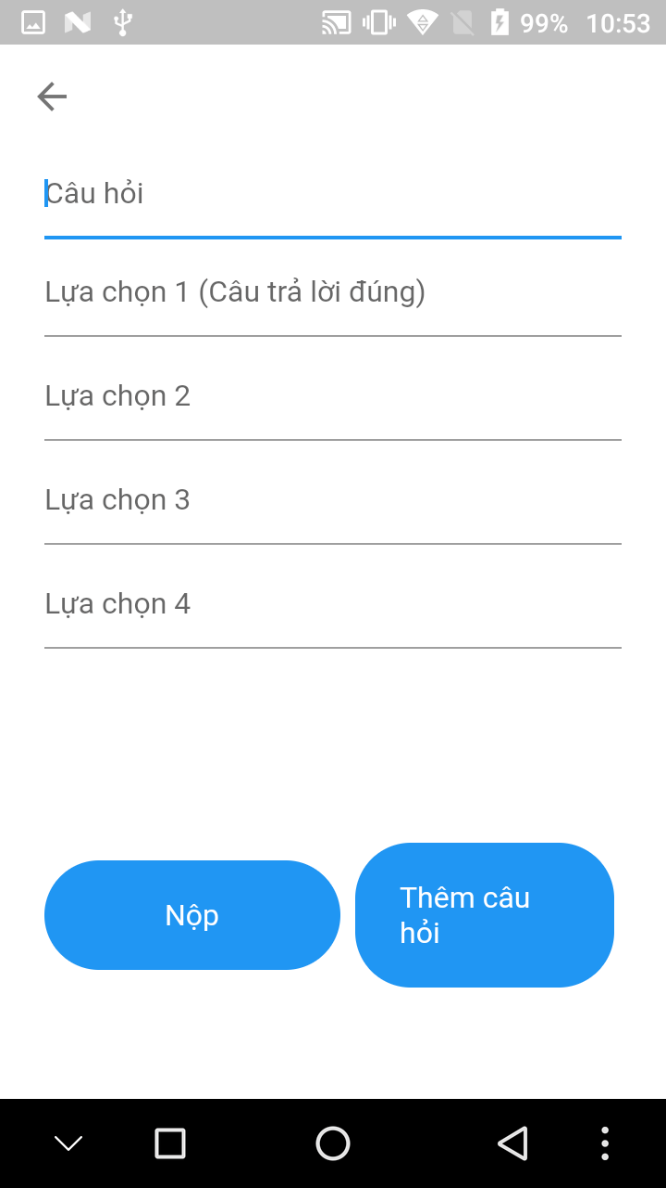
4.4. Giao diện tạo bộ đề thi



*Hình 4.4 Giao diện tạo bộ đề thi*

Tại giao diện này, người dùng thêm vào Url của hình ảnh hiển thị trên trang chủ của bộ đề thi, nhập tiêu đề và mô tả cho bộ đề thi.

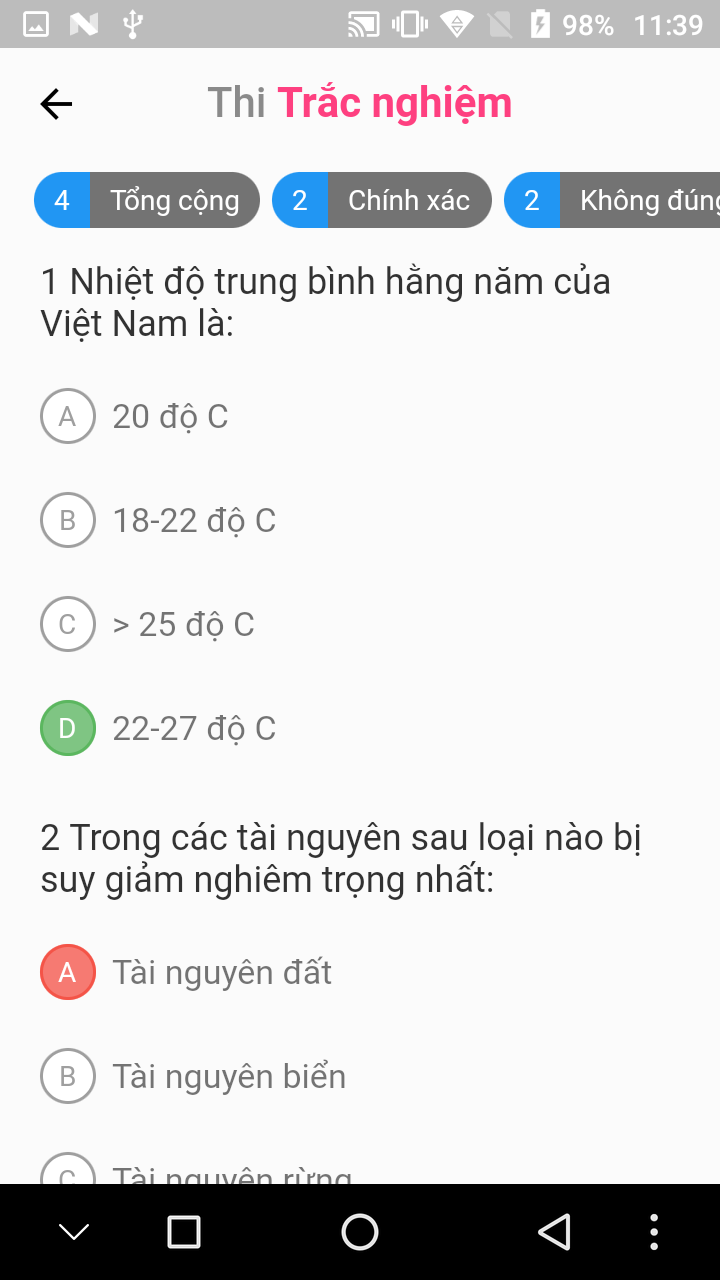
4.5. Giao diện tạo câu hỏi



*Hình 4.5 Giao diện tạo câu hỏi cho bộ đề thi*

Người dùng nhập vào câu hỏi và các câu trả lời, trong đó nhập vào 1 câu trả lời đúng sau đó nhấn “Nộp”. Để thêm nhiều câu hỏi cho bộ đề thi nhấn vào “Thêm câu hỏi” để tạo câu hỏi mới.

4.6. Giao diện thi trắc nghiệm



*Hình 4.6 Giao diện thi trắc nghiệm*

Giao diện hiển thị câu hỏi và câu trả lời, người dùng chọn câu trả lời của mình. Hệ thống sẽ cho ra kết quả số câu bạn trả lời chính xác và không đúng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Bài tiểu luận đã đạt được một số kết quả nhất định như:

- Đăng ký người dùng và đăng nhập tài khoản.

- Tạo bộ đề thi và các câu hỏi cho bộ đề kiểm tra.

- Hiển thị câu hỏi và câu trả lời, cho ra kết quả số câu trả lời chính xác và không đúng mà người dùng đã chọn.

2. Hướng phát triển

Một trong số những hướng phát triển tiếp theo của đề tài này là giữ thông tin chi tiết về thời gian bắt đầu, thời gian kết thúc của các cá nhân cho một bài kiểm tra cụ thể, lưu hồ sơ về kết quả của các cá nhân, cho phép nhiều phương thức xác thực như Gmail, số điện thoại di động, v.v.