**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU THÀNH PHỐ HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI:**

**NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG**

**CHAT APP VỚI FLUTTER**

Sinh viên thực hiện : VÕ TẤN ĐÀO

Mã sinh viên : 5951071011

Lớp : CQ.59.CNTT

Giảng viên hướng dẫn : ThS. TRẦN PHONG NHÃ

*TP Hồ Chí Minh, năm 2021*

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU THÀNH PHỐ HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI:**

**NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG**

**CHAT APP VỚI FLUTTER**

Sinhviên thực hiện : VÕ TẤN ĐÀO

Mã sinh viên : 5951071011

Lớp : CQ.59.CNTT

Giảng viên hướng dẫn : ThS. TRẦN PHONG NHÃ

*TP Hồ Chí Minh, năm 2021*

|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  **PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

# **NHIỆM VỤ THIẾT KẾ THỰC TẬP CHUYÊN MÔN**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:** 5951071011 **Họ tên SV:** Võ Tấn Đào

**Khóa:** 59 **Lớp:** CQ.59.CNTT

1. **Tên đề tài:**

NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG CHAT APP VỚI FLUTTER

1. **Mục đích, yêu cầu:**
   1. **Mục đích:**

Nhằm phục vụ cho nhu cầu giao tiếp giữa người với người. Công nghệ thông tin ngày càng xuất hiện nhiều kéo theo các cách giao tiếp cũng ngày một đa dạng, khả năng dữ liệu bị mất mát là khá cao. Chính vì thế, việc đòi hỏi khả năng bảo vệ dữ liệu của ứng dụng càng phải được nâng cao hơn nữa. Hệ thống được xây dựng trên nền tảng ứng dụng Android, IOS, Web.

* 1. **Yêu cầu:**
* **Yêu cầu công nghệ:**
  + Sử dụng ngôn ngữ lập trình dart.
  + Sử dụng công cụ Visual Studio Code.
  + Cơ sở dữ liệu thời gian thực Firebase.
* **Yêu cầu chức năng:**
  + Ứng dụng các chức năng chat, Profile.
  + Ứng dụng cho phép người dùng: login, chat, Profile.
* **Yêu cầu phi chức năng**
  + Tốc độ: Chương trình được lập trình để tối ưu về tốc độ, sử dụng băng thông hiệu quả. Tốc độ hiển thị dữ liệu đc tối ưu hiệu quả trên ứng dụng.
  + Giao diện: Thân thiện với người dùng và dễ dàng thao tác.

1. **Nội dung và phạm vi đề tài**
   1. **Nội dung:**

* Tổng quan bài toán.
* Tổng quan về các công nghệ đang sử dụng.
* Khảo sát bài toán.
* Phân tích và thiết kế hệ thống.
* Lập trình xây dựng.
* Kiểm thử và chạy thực nghiệm.
* Kết quả thu được
  1. **Phạm vi:**
     + Nghiên cứu công cụ Visual Studio Code và cơ sở dữ liệu Firebase.

1. **Công nghệ, công cụ và ngôn ngữ lập trình**
   * + Công cụ: Visual Studio Code.
     + Ngôn ngữ lập trình: Dart.
     + Cơ sở dữ liệu: Firebase.
2. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**
   * + Xây dựng ứng dụng CHAT APP.
     + Hiểu rõ hơn về Flutter.
3. **Giáo viên và cán bộ hướng dẫn**

Họ tên: ThS. Trần Phong Nhã

Đơn vị công tác: Phân hiệu Trường Đại học Giao thông Vận tải.

Điện thoại: Email:

**Ngày ...... tháng ...... năm 2021 Đã giao nhiệm vụ TTCM**

**Trưởng BM Công nghệ Thông tin Giảng viên hướng dẫn**

**ThS. Trần Phong Nhã**

Đã nhận nhiệm vụ TTCM

Sinh viên: Võ Tấn Đào Ký tên:

Điện thoại: 0394850516 Email: daovotan3@gmail.com

# **LỜI CẢM ƠN**

Lời đầu tiên em xin gửi lời cảm ơn chân thành đến quý thầy, cô giáo trong **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông vận tải.** Những người đã truyền dạy, đã trang bị cho em kho tàng kiến thức về bầu trời côngnghệ thông tin rộng lớn.

Ở đây, em không chỉ học được kiến thức về sách vở mà em còn học được các bài học, kỹ năng sống trước khi tạm biệt mái trường đại học thân yêu này và tiến ra biển đời mênh mông rộng lớn. Đặc biệt, em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy **Trần Phong Nhã**, người đã đồng hành cùng em trong suốt quá trình làm học tập, người đã bỏ thời gian quý báu. Em thật chẳng biết dùng lời nào để diễn tả được công lao của thầy “**CẢM ƠN THẦY VỀ TẤT CẢ**”.

Trong quá trình học tập và tìm hiểu em đã nỗ lực rất nhiều với mong muốn hoàn thành project một cách tốt nhất, nhưng đời người sẽ có những thiếu sót không thể tránh khỏi, và với những người chưa chững chạc và trưởng thành như em thì sai lầm là không thể không mắc phải. Em mong thầy, cô bộ môn có thể thông cảm và cho em những ý kiến, đóng góp để em có thể hoàn thành tốt hơn về đồ án của mình một cách trọn vẹn để có cơ hội sửa chữa.

Sau cùng, em xin kính chúc Quý Thầy Cô trong **Bộ môn Công nghệ thông tin** lời chúc sức khoẻ, luôn hạnh phúc và thành công hơn nữa trong công việc cũng nhưtrong cuộc sống.

Em xin chân thành cảm ơn!

# **NHẬN XÉT CỦA GIÁO VIÊN**

*Tp. Hồ Chí Minh, ngày … tháng … năm …*

Giảng viên hướng dẫn

MỤC LỤC:

[NHIỆM VỤ THIẾT KẾ THỰC TẬP CHUYÊN MÔN i](#_Toc75164566)

[LỜI CẢM ƠN iii](#_Toc75164567)

[NHẬN XÉT CỦA GIÁO VIÊN iv](#_Toc75164568)

[DANH MỤC HÌNH ẢNH vii](#_Toc75164569)

[MỞ ĐẦU viii](#_Toc75164570)

[1. Lý do chọn đề tài. viii](#_Toc75164571)

[2. Mục tiêu nghiên cứu: viii](#_Toc75164572)

[3. Đối tượng và phạm vi nghiên cứu: viii](#_Toc75164573)

[4. Phương pháp nghiên cứu viii](#_Toc75164574)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 1](#_Toc75164575)

[1.1 Tìm hiểu về Flutter. 1](#_Toc75164576)

[1.1.1 Giới thiệu. 1](#_Toc75164577)

[1.1.2 Khái niệm. 1](#_Toc75164578)

[1.1.3 Ưu và nhược điểm. 1](#_Toc75164579)

[1.2 Ngôn ngữ lập trình dart: 2](#_Toc75164580)

[1.2.1 Giới thiệu: 2](#_Toc75164581)

[1.2.2 Các đặc điểm của ngôn ngữ lập trình dart: 2](#_Toc75164582)

[1.3 Firebase. 3](#_Toc75164583)

[1.3.1 Giới thiệu. 3](#_Toc75164584)

[1.3.2 Khái niệm. 3](#_Toc75164585)

[1.3.3 Các chức năng chính của Firebase. 4](#_Toc75164586)

[1.3.4 Những lợi ích từ Firebase 8](#_Toc75164587)

[1.3.5 Nhược điểm của Firebase 9](#_Toc75164588)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ 10](#_Toc75164589)

[2.1 Đặc tả bài toán 10](#_Toc75164590)

[2.2 Biểu đồ use case 10](#_Toc75164591)

[2.3 Đăng ký thành viên 10](#_Toc75164592)

[2.4 Đăng nhập bằng Google 12](#_Toc75164593)

[2.5 Thiết kế dữ liệu Chat 13](#_Toc75164594)

[CHƯƠNG 3: THỬ NGHIỆM ỨNG DỤNG 15](#_Toc75164595)

[3.1 Môi trường cài đặt 15](#_Toc75164596)

[3.2 Triển khai chương trình 15](#_Toc75164597)

[3.2.1 Giao diện đăng nhập 15](#_Toc75164598)

[3.2.2 Giao diện chính 17](#_Toc75164599)

[3.2.3 Giao diện thêm bạn bè 18](#_Toc75164600)

[3.2.4 Giao diện Profile và Setting 19](#_Toc75164601)

[3.2.5 Giao diện nhắn tin 21](#_Toc75164602)

[3.2.6 Giao diện search 22](#_Toc75164603)

[3.2.7 Xoá tin nhắn 23](#_Toc75164604)

[3.2.8 Đăng xuất 24](#_Toc75164605)

[3.3 Kết quả và đánh giá 24](#_Toc75164606)

[CHƯƠNG 4: KẾT LUẬN 26](#_Toc75164607)

[TÀI LIỆU THAM KHẢO 28](#_Toc75164608)

# **DANH MỤC HÌNH ẢN****H**

[*Hình 1. Các chức năng chính của Firebase.* 4](#_Toc75164484)

[*Hình 2. Firebase Database.* 4](#_Toc75164485)

[*Hình 3. Firebase Authentication.* 5](#_Toc75164486)

[*Hình 4. Firebase Storage.* 6](#_Toc75164487)

[*Hình 5. Firebase Hosting.* 7](#_Toc75164488)

[*Hình 6. Cloud Messaging.* 7](#_Toc75164489)

[*Hình 7. Biểu đồ use case.* 10](#_Toc75164490)

[*Hình 8. Setup Firebase1.* 11](#_Toc75164491)

[*Hình 9. Setup Firebase2.* 11](#_Toc75164492)

[*Hình 10. Enable Google để sign in.* 12](#_Toc75164493)

[*Hình 11. Danh sách các user đã đăng kí bằng Google.* 12](#_Toc75164494)

[*Hình 12. Lưu user đã đã đăng nhập có id 61Bze9PRk9grKjs9O5HgAyqm0iO2.* 12](#_Toc75164495)

[*Hình 13. Lưu ChatRoom có id là daovotan3\_Đào Võ Tấn.* 13](#_Toc75164496)

[*Hình 14. Lưu chat có id là 5CUMlb48OtcNZHBcbk3Q.* 14](#_Toc75164497)

[*Hình 15. Giao diện đăng nhập.* 15](#_Toc75164498)

[*Hình 16. Giao diện chọn tài khoản để đăng nhập.* 16](#_Toc75164499)

[*Hình 17. Giao diện chính của app.* 17](#_Toc75164500)

[*Hình 18. Giao diện thêm bạn bè.* 18](#_Toc75164501)

[*Hình 19. Giao diện Profile.* 19](#_Toc75164502)

[*Hình 20. Giao diện setting.* 20](#_Toc75164503)

[*Hình 21. Giao diện chat giữa 2 user.* 21](#_Toc75164504)

[*Hình 22. Giao diện tìm kiếm user để chat.* 22](#_Toc75164505)

[*Hình 23. Xóa tin nhắn.* 23](#_Toc75164506)

[*Hình 24. Giao diện đăng xuất.* 24](#_Toc75164507)

# **MỞ ĐẦU**

1. **Lý do chọn đề tài.**

Khi chúng ta hầu như đều nhận ra rằng các ứng dụng nhắn tin đã thay đổi cách giao tiếp, thông qua việc sử dụng các smartphone. Thói quen sử dụng smartphone đang tăng dần trong tất cả lĩnh vực và nó cũng cho thấy rằng số lượng người dùng nhắn tin qua ứng dụng chat đang phát triển mạnh, lên đến 2,48 tỷ người dùng trong năm 2021. Những thống kê này hoàn toàn là một dấu hiệu tốt cho đối với việc phát triển ứng dụng chat.

**Hướng tiếp cận của đề tài:**

* Nghiên cứu và hiểu được cách thức hoạt động của database realtime.
* Thiết kế cơ sở dữ liệu sao cho phù hợp với ứng dụng đang phát triển.
* Nắm vững kiến thức nền tảng của ngôn ngữ lập trình dart, tiến đến phát triển ứng dụng.
* Nắm được cách xây dựng UI
* Tiến hành kiểm tra và chạy thử ứng dụng.

1. **Mục tiêu nghiên cứu:**

* Phát triển thành công ứng dụng Chat App.
* Hiểu được cách thức hoạt động của cơ sở dữ liệu.
* Khiến ứng dụng trở nên có ích hơn cho xã hội.
* Tìm hiểu về công nghệ Flutter.
* Tìm hiểu về ngôn ngữ dart.

1. **Đối tượng và phạm vi nghiên cứu:**

* Đối tượng: Mọi người có smartphone đều có thể sử dụng.
* Phạm vi ứng dụng: Mong muốn ứng dụng có thể tiếp cận được nhiều người sử dụng.

1. **Phương pháp nghiên cứu**

* Tài liệu: Tìm hiểu và khai thác các tài liệu liên quan đến cơ sở dữ liệu Firebase, các nền tảng kiến thức về dart cũng như công cụ Visual Studio Code.
* Thực hành: Tiến hành xây dựng UI cho ứng dụng sau đó kết nối cơ sở dữ liệu Firebase với Visual Studio Code và thao tác với dữ liệu.

# 

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

## Tìm hiểu về Flutter.

### Giới thiệu.

Flutter là bộ công cụ giao diện của Google để xây dựng ứng dụng đẹp, phù hợp với các nền tảng web, điện thoại và máy tính từ một codebase duy nhất – kiểu “Write once and run anywhere”.

Lịch sử phát triển:

* Vào năm 2015, Google tiết lộ Flutter – SDK mới dựa trên ngôn ngữ Dart, như là nền tảng mới để phát triển Android.
* Năm 2017, bản alpha (0.0.6) được public lần đầu tiên.
* Vào 04/12/2018, Flutter 1.0 được công bố tại sự kiện Flutter Live, đánh dấu phiên bản “stable” đầu tiên.
* Vào 11/12/2019, sự kiện Flutter Interact diễn ra và công bố rất nhiều sự nâng cấp cùng với sự ra đời của Flutter 1.12. Tính tới 15/6/2021, phiên bản mới nhất là 2.2

### Khái niệm.

Flutter là một nền tảng hỗ trợ phát triển cho các ứng dụng đa nền cho hệ iOS và Android do tập đoàn Google phát triển. Nó được sử dụng vô cùng phổ biến cho nhiệm vụ tạo ra các ứng dụng gốc dành cho Google.

### Ưu và nhược điểm.

* **Ưu điểm:**
* Flutter là bộ open-source SDK, tức là miễn phí và mở – cộng đồng developer có thể cùng tham gia phát triển.
* Giao diện đẹp, dễ nhìn và dễ sử dụng.
* Ngôn ngữ kiểu tĩnh nhưng với cú pháp hiện đại (tương tự JS, Python, Java), compiler linh động dùng AOT (cho sản phẩm cuối) và JIT (cho quá trình phát triển với hot reload).
* Có thể chạy được giả lập mobile ngay trên web, tiện cho việc phát triển. Các bộ đo lường chỉ số hiệu suất được hỗ trợ sẵn giúp lập trình viên kiểm soát tốt hiệu suất của ứng dụng.
* Viết code ứng dụng nhanh hơn, viết code 1 lần chạy trên cả IOS và Android.
* **Nhược điểm:**
* Bộ render UI được nhóm phát triển gần như viết lại, không liên quan tới UI có sẵn của Framework native, dẫn đến memory sử dụng khá nhiều.
* Phải học thêm ngôn ngữ Dart: lập trình viên biết về Dart không nhiều, cũng có rủi ro là học xong Dart sẽ dính liền luôn với Dart ở mảng phát triển ứng dụng mobile. Chứ không uyển chuyển như JS hay Python có thể nhảy qua lại giữa front, back hay AI…

## Ngôn ngữ lập trình dart.

### Giới thiệu:

Dart là một ngôn ngữ hướng đối tượng, được xác định theo lớp, với cơ chế garbage-collected, sử dụng cú pháp kiểu C để dịch mã tùy ý sang JavaScript. Nó hỗ trợ interface, mixin, abstract, generic, static typing và sound type (2 cái cuối có thể hiểu là type-safe).

### Các đặc điểm của ngôn ngữ lập trình dart:

* **Hướng đối tượng** – Trong Dart, mọi thứ đều là một Object. Dart có thể dễ dàng mở rộng và bảo trì vì nó được xây dựng dựa trên mô hình Object.
* *Tính trừu tượng (Abstraction)*: là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.
* *Tính đa hình (Polymorphism):* cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.
* *Tính kế thừa (Inheritance)*: Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
* *Tính đóng gói (Encapsulation)*: là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.
* Là ngôn ngữ lập trình đơn giản, dễ hiểu và dễ đến gần hơn.
* Là ngôn ngữ lập trình được sử dụng để build lên Flutter Framework.
* Nó xuất hiện vào năm 2011, hướng tới việc tạo ra các ứng dụng đa nền tảng – Website, mobile, desktop và IoT.
* Nó như 1 sự kết hợp giữa Java và JavaScript nên khi học nó nếu ai đã có nền tảng 1 trong 2 ngôn ngữ kia thì lúc đến gần hơn sẽ khá dễ.

## Firebase.

### Giới thiệu.

Trở lại những năm 2011, trước khi Firebase là Firebase bây giờ thì nó là một startup, được gọi là Envolve. Envolve cung cấp cho các nhà phát triển một API cho phép tích hợp chức năng trò chuyện trực tuyến vào trang web của họ. Điều thú vị là mọi người đã sử dụng Envolve để truyền dữ liệu ứng dụng chứ không chỉ là tin nhắn trò chuyện đơn thuần. Các nhà phát triển đang sử dụng Envolve để đồng bộ hóa dữ liệu ứng dụng chẳng hạn như trạng thái trò chơi trong thời gian thực trên người dùng của họ. Điều này đã dẫn dắt những người sáng lập Envolve, James Tamplin và Andrew Lee, tách riêng chat system và the real-time architecture. Vào tháng 4 năm 2012, Firebase được tạo ra dưới dạng công ty riêng biệt Backend-as-a-Service với chức năng thời gian thực. Sau khi được Google mua lại vào năm 2014, Firebase nhanh chóng phát triển thành ứng dụng đa năng của nền tảng di động và web hiện nay.

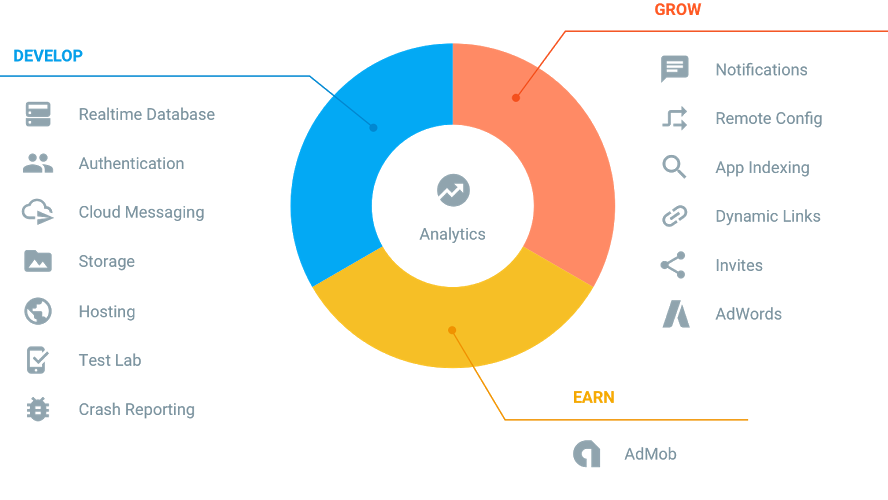
### Khái niệm.

**Firebase** là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Firebase là nền tảng phát triển ứng dụng dành cho thiết bị di động và web, cung cấp cho nhà phát triển nhiều công cụ và dịch vụ giúp họ phát triển các ứng dụng chất lượng cao, phát triển user base và tìm kiếm thêm lợi nhuận.

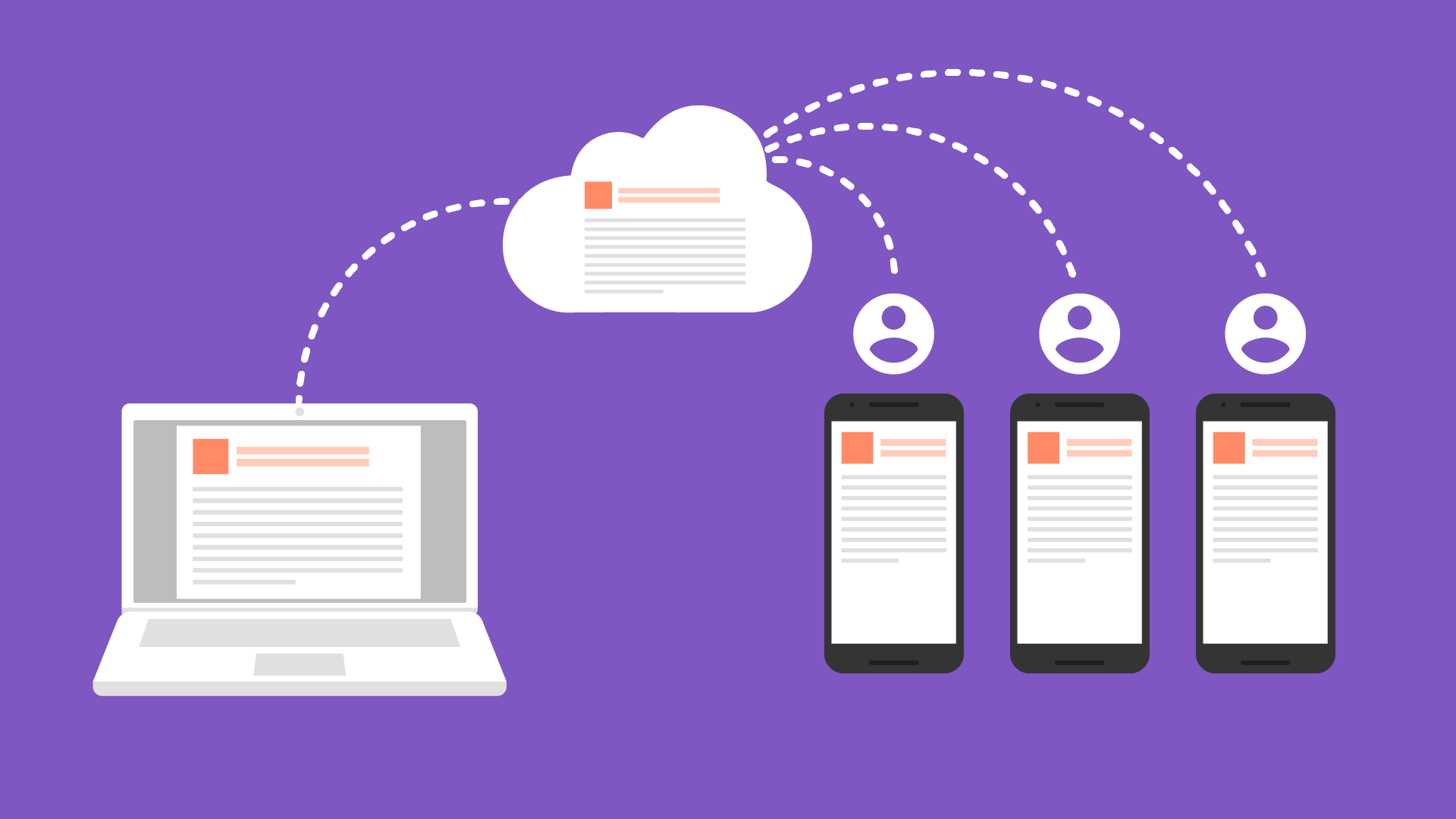
Firebase là sự kết hợp giữa nền tảng cloud với hệ thống máy chủ cực kì mạnh mẽ tới từ Google, để cung cấp cho chúng ta những API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database

### Các chức năng chính của Firebase.



*Hình 1. Các chức năng chính của Firebase.*

* **Realtime Database – Cơ sở dữ liệu thời gian thực.**



*Hình 2. Firebase Database.*

Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là bạn có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.

Ngoài ra Firebase còn cho phép bạn phân quyền một các đơn giản bằng cú pháp tương tự như javascript.

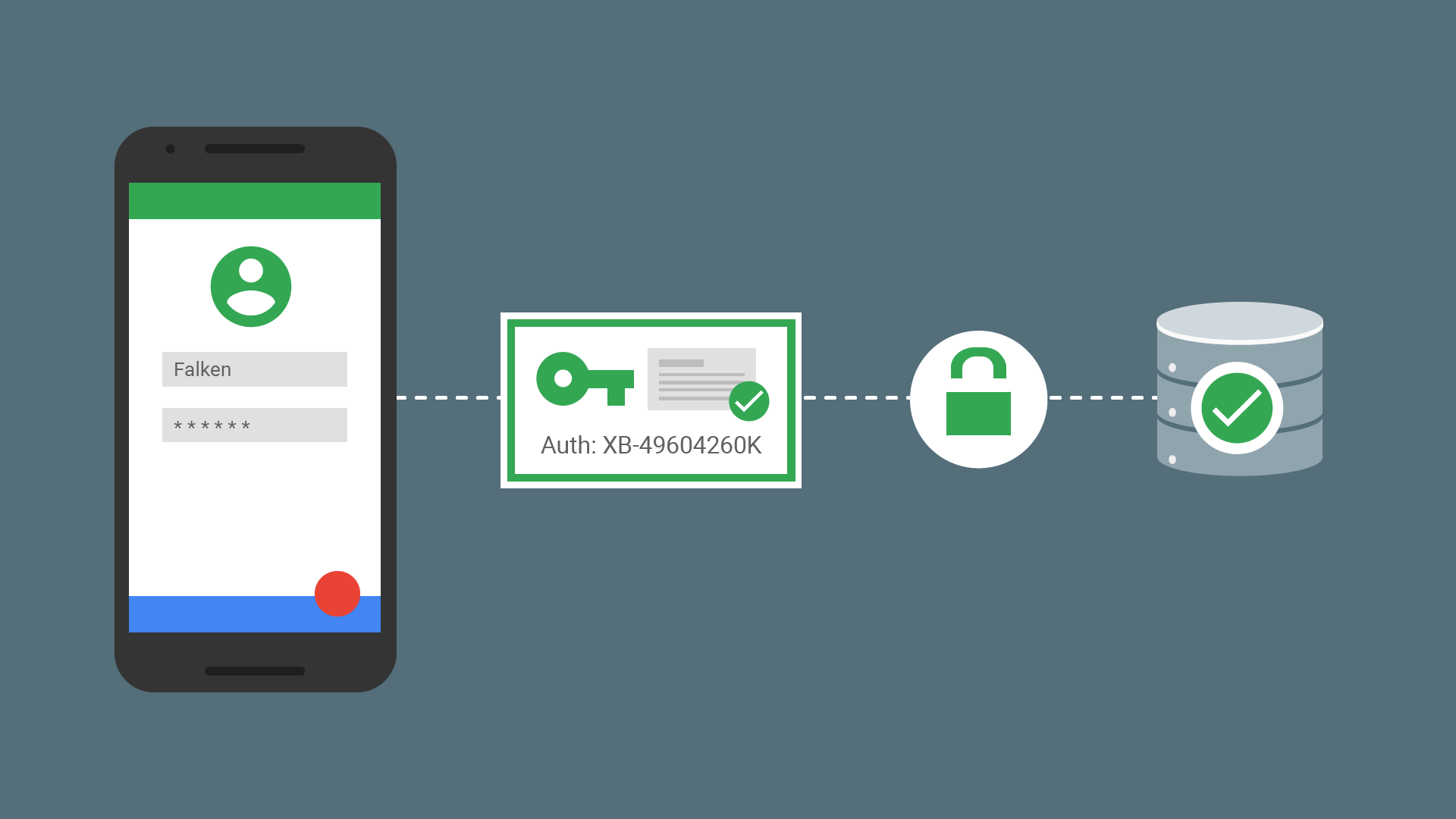
Đồng bộ hóa trong thời gian thực giúp người dùng dễ dàng truy cập dữ liệu của họ từ bất kỳ thiết bị nào, có thể là web hoặc thiết bị di động. Cơ sở dữ liệu thời gian thực cũng giúp người dùng có khả năng cộng tác với nhau.

Một lợi ích tuyệt vời khác của Cơ sở dữ liệu thời gian thực là nó chứa mobile và web SDKs, cho phép bạn tạo ứng dụng mà không cần máy chủ.

Khi người dùng của bạn ngoại tuyến, Realtime Database SDKs sử dụng bộ nhớ cache cục bộ trên thiết bị để phân phát và lưu trữ thay đổi. Khi thiết bị trực tuyến, dữ liệu cục bộ sẽ tự động được đồng bộ hóa.

Realtime Database cũng có thể tích hợp với Firebase Authentication để cung cấp quy trình xác thực đơn giản và trực quan.

**Firebase Authentication – Hệ thống xác thực của Firebase.**



*Hình 3. Firebase Authentication.*

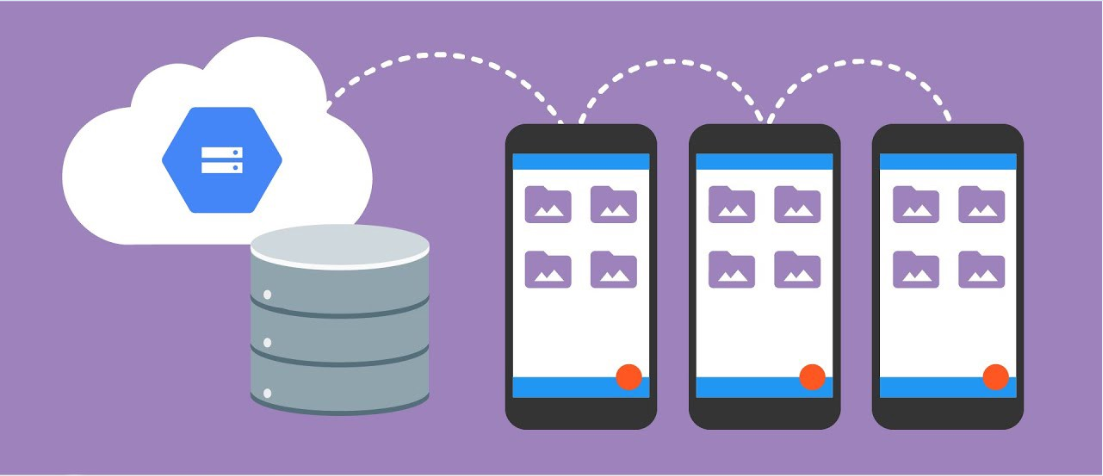
Với Firebase bạn có thể dễ dàng tích hợp các công nghệ xác thực của Google, Facebook, Twitter, … hoặc một hệ thống xác thực mà bạn tự mình tạo ra vào trong ứng dụng của bạn ở bất kì nền tảng nào như Android, iOS hoặc Web.

Firebase Authentication cung cấp backend services, easy-to-use SDKs, và ready-made UI libraries để xác thực người dùng cho ứng dụng.

Thông thường, sẽ mất vài tháng để thiết lập nên một hệ thống xác thực của riêng bạn. Và sau đó, bạn sẽ phải cần đến một đội ngũ chuyên viên để duy trì hệ thống đó. Nhưng nếu sử dụng Firebase, bạn có thể thiết lập toàn bộ hệ thống chỉ với dưới 10 dòng mã, bao gồm cả các hoạt động phức tạp như hợp nhất tài khoản.

Sử dụng Xác thực Firebase giúp việc xây dựng hệ thống xác thực an toàn dễ dàng hơn, đồng thời cải thiện sign-in và onboarding experience cho người dùng cuối.

* **Firebase storage.**



*Hình 4. Firebase Storage.*

Với firebase storage các lập trình viên có thể lưu trữ dữ liệu trực tiếp trên server của firebase như hình ảnh, video, tập tin một cách dễ dàng.  
Firebase còn bổ sung Google security để tải lên và tải về các ứng dụng firebase của bạn. Bạn có thể sử dụng nó để lưu trữ hình ảnh, âm thanh, video hoặc nội dung do người dùng tạo ra. Quả thật đây là một dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và hiệu quả chi phí.

* **Firebase Hosting.**



*Hình 5. Firebase Hosting.*

Firebase hosting cung cấp nhanh chóng và an toàn cho việc lưu trữ ứng dụng web của bạn. Với Hosting, bạn có thể nhanh chóng và dễ dàng triển khai các ứng dụng web và các nội dung tĩnh cho một mạng nội dung phân phối toàn cầu (CDN) với một lệnh duy nhất.

* **Firebase Cloud Messaging.**



*Hình 6. Cloud Messaging.*

Không chỉ vậy, hiện nay firebase còn cung cấp dịch vụ Cloud Messaging thay thế cho Google Cloud Messaging đã cũ kỹ. Một giải pháp đa nền tảng, đáng tin cậy cho phép bạn cung cấp các thông điệp miễn phí dạng tin nhắn đến người dùng.  
 Sử dụng FCM, bạn có thể thông báo cho một ứng dụng client email mới hoặc dữ liệu khác có sẵn để đồng bộ hóa. Bạn có thể gửi tin nhắn thông báo cho lái xe reengagement sử dụng và duy trì. Đối với trường hợp sử dụng như tin nhắn tức thời, tin nhắn có thể chuyển một tải trọng lên đến 4KB để một ứng dụng client.

### Những lợi ích từ Firebase

* **Triển khai ứng dụng cực kỳ nhanh**

Với Firebase bạn có thể giảm bớt rất nhiều thời gian cho việc viết các dòng code để quản lý và đồng bộ cơ sở dữ liệu, mọi việc sẽ diễn ra hoàn toàn tự động với các API của Firebase. Không chỉ có vậy Firebase còn hỗ trợ đã nền tảng nên bạn sẽ càng đỡ mất thời gian rất nhiều khi ứng dụng bạn muốn xây dựng là ứng dụng đa nền tảng.

Không chỉ nhanh chóng trong việc xây dựng database, Google Firebase còn giúp ta đơn giản hóa quá trình đăng kí và đăng nhập vào ứng dụng bằng các sử dụng hệ thống xác thực do chính Firebase cung cấp.

* + **Bảo mật**

Firebase hoạt động dựa trên nền tảng cloud và thực hiện kết nối thông qua giao thức bảo mật SSL, chính vì vậy bạn sẽ bớt lo lắng rất nhiều về việc bảo mật của dữ liệu cũng như đường truyền giữa client và server. Không chỉ có vậy, việc cho phép phân quyền người dùng database bằng cú pháp javascipt cũng nâng cao hơn nhiều độ bảo mật cho ứng dụng của bạn, bởi chỉ những user mà bạn cho phép mới có thể có quyền chỉnh sửa cơ sở dữ liệu.

Tính linh hoạt và khả năng mở rộng.

Sử dụng Firebase sẽ giúp bạn dễ dàng hơn rất nhiều mỗi khi cần nâng cấp hay mở rộng dịch vụ. Ngoài ra firebase còn cho phép bạn tự xây dựng server của riêng mình để bạn có thể thuận tiện hơn trong quá trình quản lý.

Việc Firebase sử dụng NoSQL, giúp cho database của bạn sẽ không bị bó buộc trong các bảng và các trường mà bạn có thể tùy ý xây dựng database theo cấu trúc của riêng bạn.

Firebase hoạt động dựa trên nền tảng cloud đến từ Google vì vậy hầu như bạn không bao giờ phải lo lắng về việc sập server, tấn công mạng như DDOS, tốc độ kết nối lúc nhanh lúc chậm, … nữa, bởi đơn giản là Firebase hoạt động trên hệ thống server của Google. Hơn nữa nhờ hoạt động trên nền tảng Cloud nên việc nâng cấp, bảo trì server cũng diễn ra rất đơn giản mà không cần phải dừng server để nâng cấp như truyền thống.

* **Giá thành**

Google Firebase có rất nhiều gói dịch vụ với các mức dung lượng lưu trữ cũng như băng thông khác nhau với mức giá dao động từ Free đến $1500 đủ để đáp ứng được nhu cầu của tất cả các đối tượng. Chính vì vậy bạn có thể lựa chọn gói dịch vụ phù hợp nhất với nhu cầu của mình. Điều này giúp bạn tới ưu hóa được vốn đầu tư và vận hành của mình tùy theo số lượng người sử dụng. Ngoài ra bạn còn không mất chi phí để bảo trì, nâng cấp, khắc phục các sự cố bởi vì những điều này đã có Firebase lo.

### Nhược điểm của Firebase

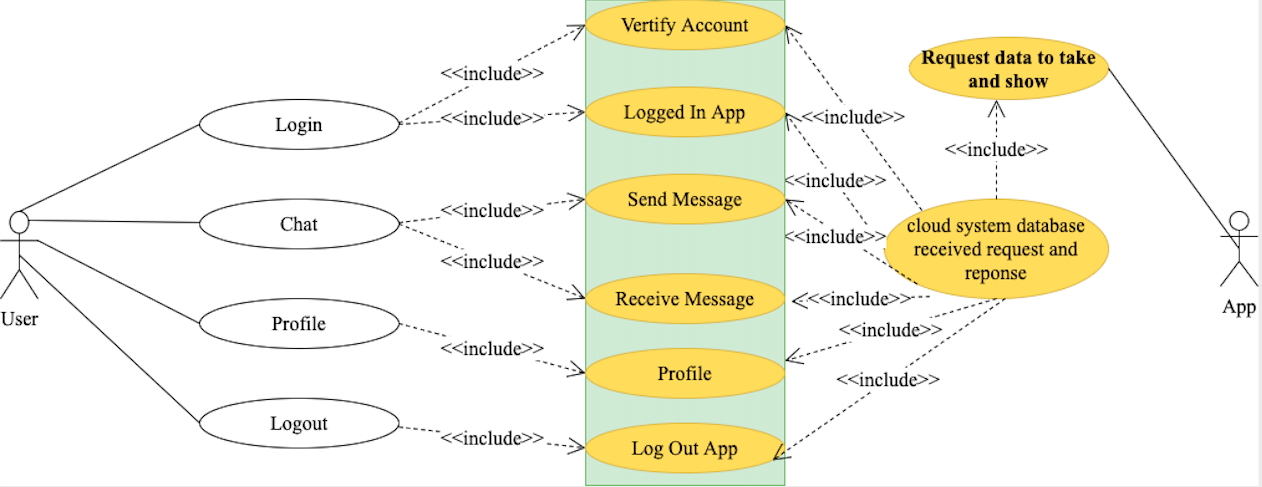
Điểm yếu duy nhất của “Firebase” chính là phần Realtime Database, mà đúng hơn chỉ là phần Database. Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn từ mức đôi chút tới khá nhiều.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ**

## Đặc tả bài toán

* Dữ liệu đầu vào: các user đăng nhập vào ứng dụng bằng tài khoản Google.
* Dữ liệu đầu ra:
* Sau khi người dùng đăng nhập thành công người dùng có thể search để tìm các user.
* Giao diện hiển thị đoạn chat giữa 2 user với nhau.
* Giao diện Profile và Setting.
* Giao diện đăng xuất để người dùng thoát khỏi ứng dụng.

## Biểu đồ use case

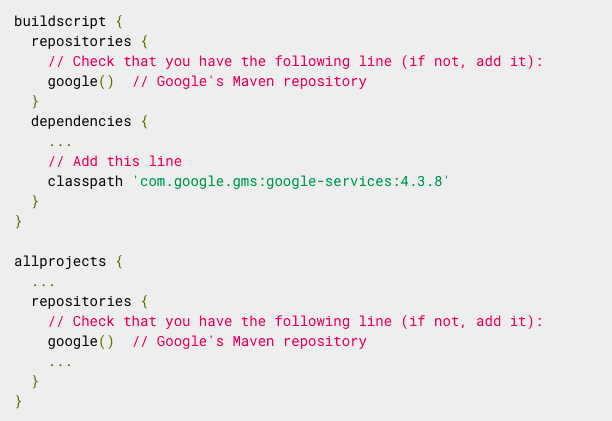


*Hình 7. Biểu đồ use case.*

## Đăng ký thành viên

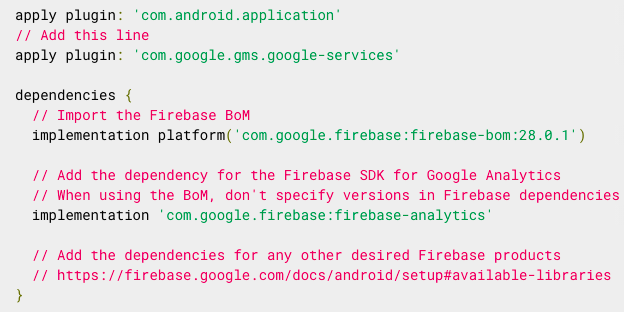
* **Bước 1:** Thêm chức năng xác thực vào project (Authentication).

Mở file **Project-level build.gradle** (<project>/build.gradle): và thêm Firebase dependency.



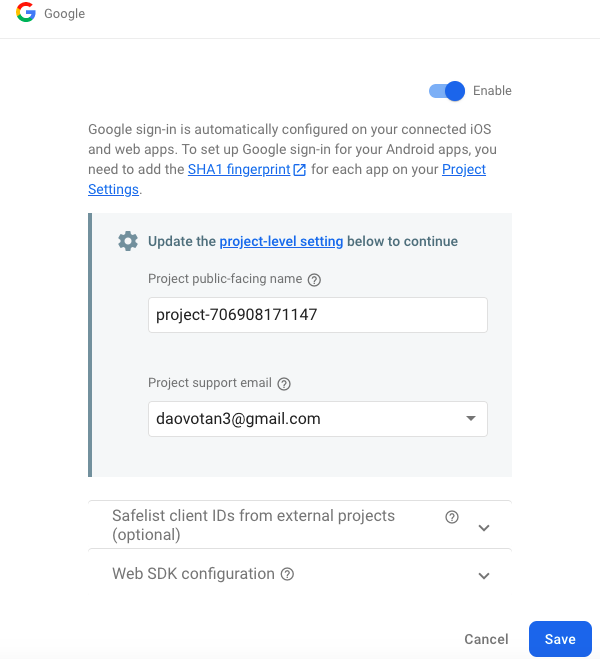
*Hình 8. Setup Firebase1.*

Mở file **App-level build.gradle** (<project>/<app-module>/build.gradle) và thêm **firebase auth dependency** và ở cuối thêm plugin **‘com.google.gms.google services’.**



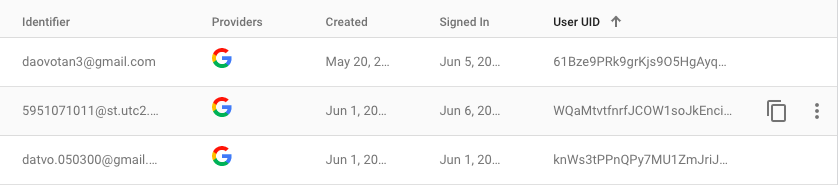
*Hình 9. Setup Firebase2.*

* **Bước 2:** Thêm Google để sign in

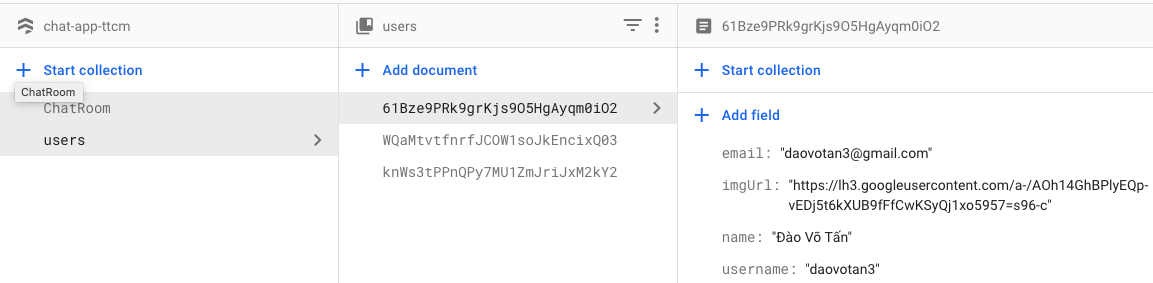


*Hình 10. Enable Google để sign in.*

## Đăng nhập bằng Google

****

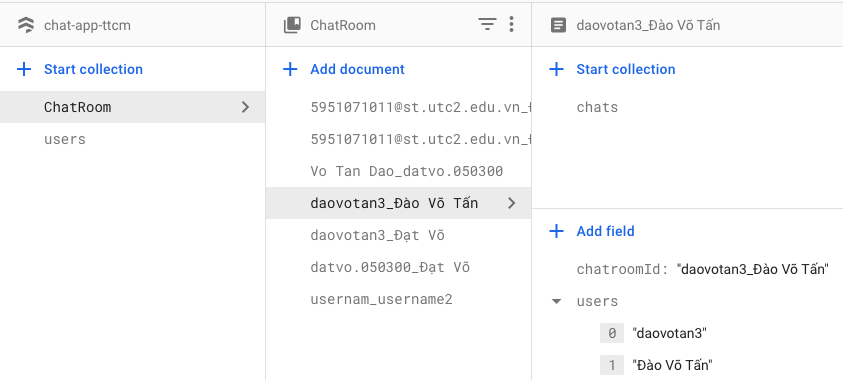
*Hình 11. Danh sách các user đã đăng kí bằng Google.*

****

*Hình 12. Lưu user đã đăng nhập có id 61Bze9PRk9grKjs9O5HgAyqm0iO2.*

Thêm dữ liệu Cloud Firestore lưu data trong Documents, các Documents được lưu trữ trong các Collections. Khi bạn thêm dữ liệu vào trong một document lần đầu tiên thì Cloud Firestore sẽ tự động tạo các collections và documents bạn sẽ không cần tạo collections và documents một cách tường minh nữa. Khi người dùng đăng nhập thì thông tin sẽ được lưu lên firebase có id là **61Bze9PRk9grKjs9O5HgAyqm0iO2**. Tương tự như các user đã đăng nhập vào có id lần lượt là: [W**QaMtvtfnrfJCOW1soJkEncixQ03**](https://console.firebase.google.com/u/0/project/chat-app-ttcm/firestore/data~2Fusers~2FWQaMtvtfnrfJCOW1soJkEncixQ03) và [**knWs3tPPnQPy7MU1ZmJriJxM2kY2**](https://console.firebase.google.com/u/0/project/chat-app-ttcm/firestore/data~2Fusers~2FknWs3tPPnQPy7MU1ZmJriJxM2kY2)**.**

## Thiết kế dữ liệu Chat

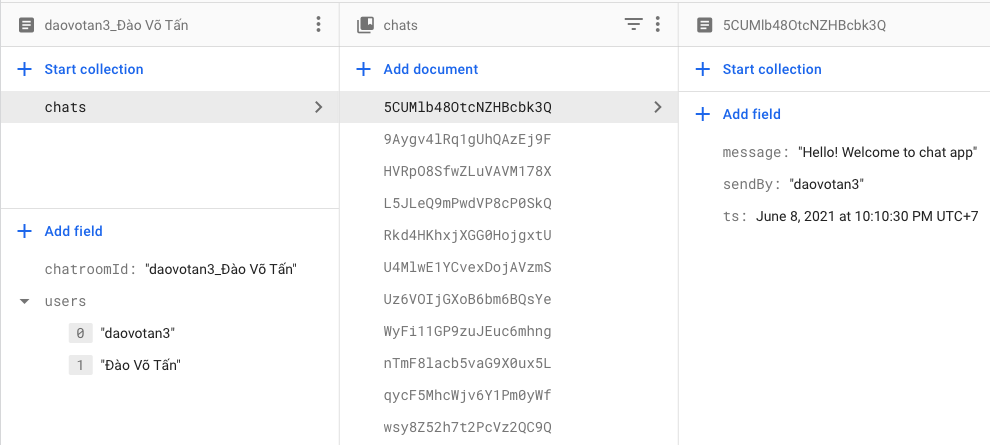
****

*Hình 13. Lưu ChatRoom có id là daovotan3\_Đào Võ Tấn.*

Giai đoạn này chỉ được thực hiện user khi các người dùng tìm kiếm người muốn nhắn tin. Cũng giống như các users mỗi tin nhắn gửi đi chúng ta cũng xem nó như một đối tượng. Tức nó cũng phải đảm bảo đầy đủ các thuộc tính cơ bản của một tin nhắn. Trong ChatRoom thì document có id ChatRoom được tạo từ 2 userName và colection sẽ bao gồm 2 field là chats và chatroomId, users. Các users là một mảng được tạo từ username, vị trí 0 là người gửi còn vị trí 1 là người nhận.

Vấn đề duy nhất cần đặt ra ở đây là nếu ứng dụng của bạn có nhiều người sử dụng và không phải họ chỉ nhắn tin với một người, họ có thể sẽ nhắn tin với nhiều người và đâu phải họ chỉ gửi một tin, họ sẽ gửi nhiều tin. Vậy làm cách nào để ta có thể quản lý vấn đề này?

Vấn đề này khiến em phải suy nghĩ khá nhiều vì do mới chỉ vừa hiểu cách thức hoạt động nên khả năng thiết kế cơ sở dữ liệu của em vẫn chưa hoàn thiện dẫn đến việc em đã lưu TẤT CẢ TIN NHẮN vào trong một node và em gọi nó là node “chats ”. Thoạt nhìn thì có vẻ đúng nhưng về sau số lượng tin nhắn quá nhiều dẫn đến việc không kiểm soát được. Sau một thời gian thì em sẽ thực hiện bằng cách tạo một node “chatRoom” sẽ có id được tạo từ 2 username mà 2 user thực hiện nhắn tin cho nhau và một collection bao gồm 2 trường là chats và chatroomId, users, thì rõ ràng ở đây ta đã quản lý được số lượng tin nhắn mà các user đã nhắn với nhau. Sau khi đã thiết kế hoàn tất việc lưu tin nhắn trên Realtime Database, chúng ta sẽ thu được kết quả như thế này:



*Hình 14. Lưu chat có id là 5CUMlb48OtcNZHBcbk3Q.*

Ở đây khi 2 user chat với nhau thì chatRoom sẽ tạo một id là [**daovotan3\_Đào Võ Tấn**](https://console.firebase.google.com/u/0/project/chat-app-ttcm/firestore/data~2FChatRoom~2Fdaovotan3_%C4%90%C3%A0o%20V%C3%B5%20T%E1%BA%A5n). Khi user có id là **daovotan3** thực hiện chat với id là **Đào Võ Tấn** sẽ tạo ra một id chat tự động là [**5CUMlb48OtcNZHBcbk3Q**](https://console.firebase.google.com/u/0/project/chat-app-ttcm/firestore/data~2FChatRoom~2Fdaovotan3_%C4%90%C3%A0o%20V%C3%B5%20T%E1%BA%A5n~2Fchats~2F5CUMlb48OtcNZHBcbk3Q). Cũng tương tự các “chatRoom” khác thì cũng có các id “chatRoom” được tạo và id chat được tạo tự động để quản lý tin nhắn một cách hợp lý và dễ dàng hơn.

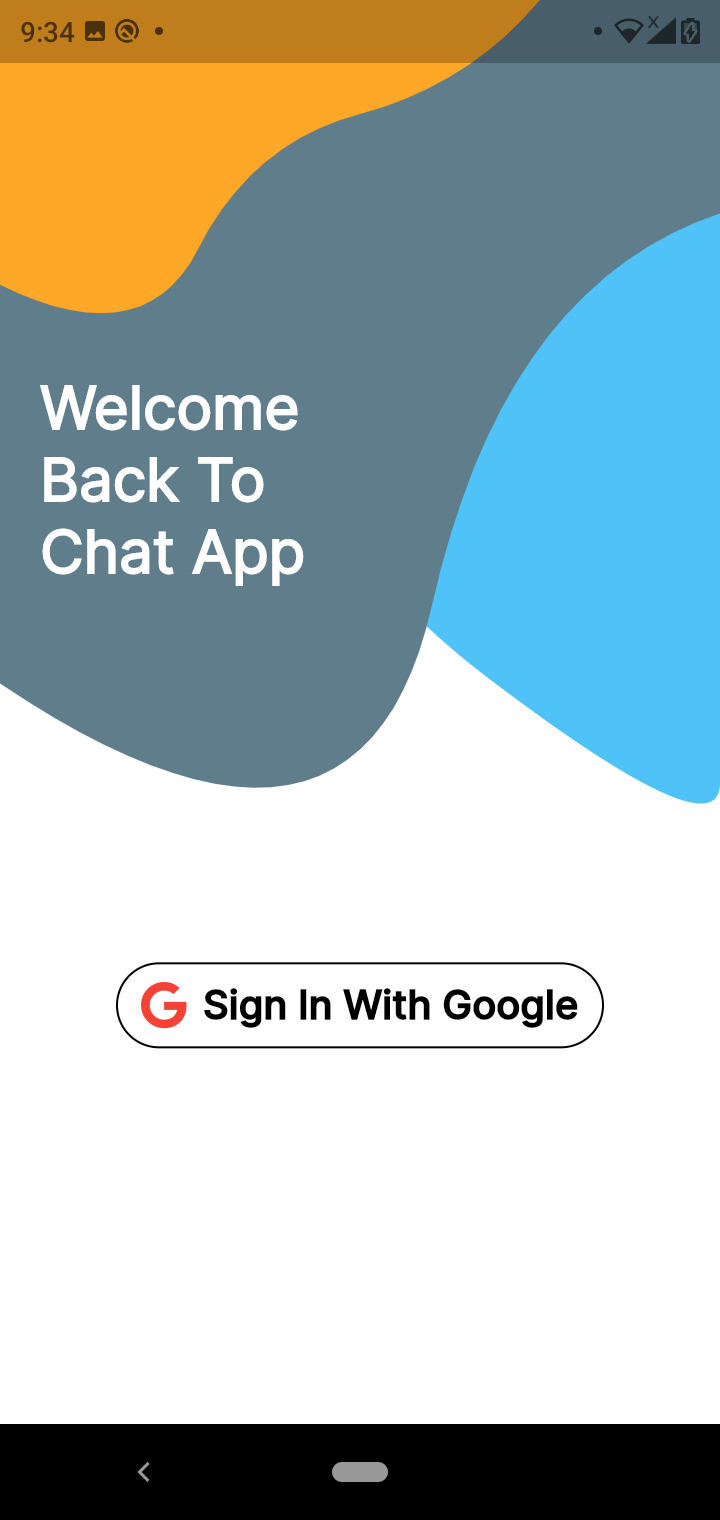
# **CHƯƠNG 3: THỬ NGHIỆM ỨNG DỤNG**

## Môi trường cài đặt

Từ những cơ sở lý thuyết, phương pháp phân tích và thiết kế được đề cập ở chương 2. Em đã xây dựng và phát triển thành công ứng dụng trên môi trường Visual Studio code, viết bằng ngôn ngữ lập trình dart.

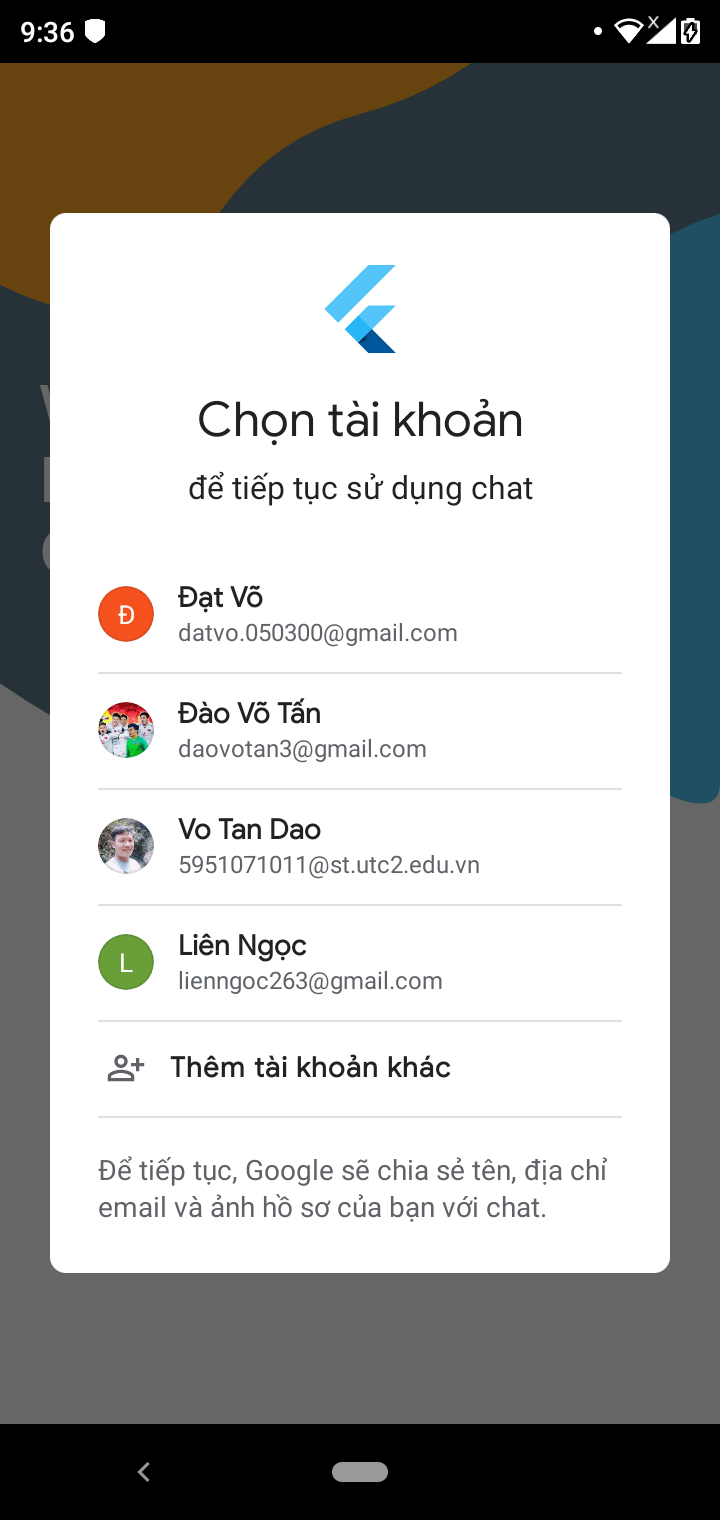
## Triển khai chương trình

### Giao diện đăng nhập



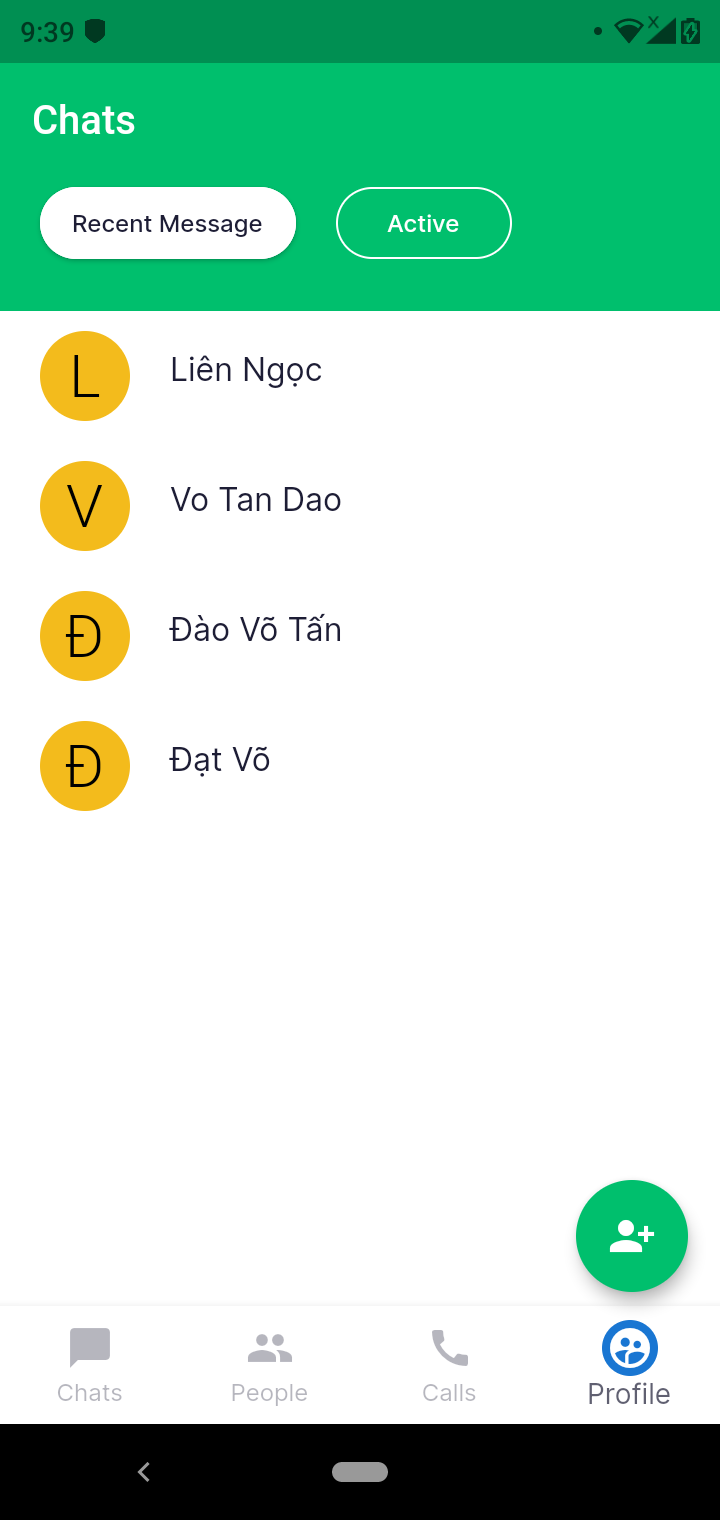
*Hình 15. Giao diện đăng nhập.*

Cũng như bao ứng dụng khác đầu tiên chúng ta phải tiến hành đăng ký thành viên trước. Chỉ khi nào đăng ký thành công thì toàn bộ các thông tin cá nhân đã đang ký sẽ được lưu vào Realtime Database, lúc này thì chúng ta có thể thực hiện trải nghiệm các chức năng vốn có của ứng dụng. Ở đây chúng ta đăng nhập bằng Google, khi người dùng đăng nhập thì màn hình sẽ show các tài khoản như sau:



*Hình 16. Giao diện chọn tài khoản để đăng nhập.*

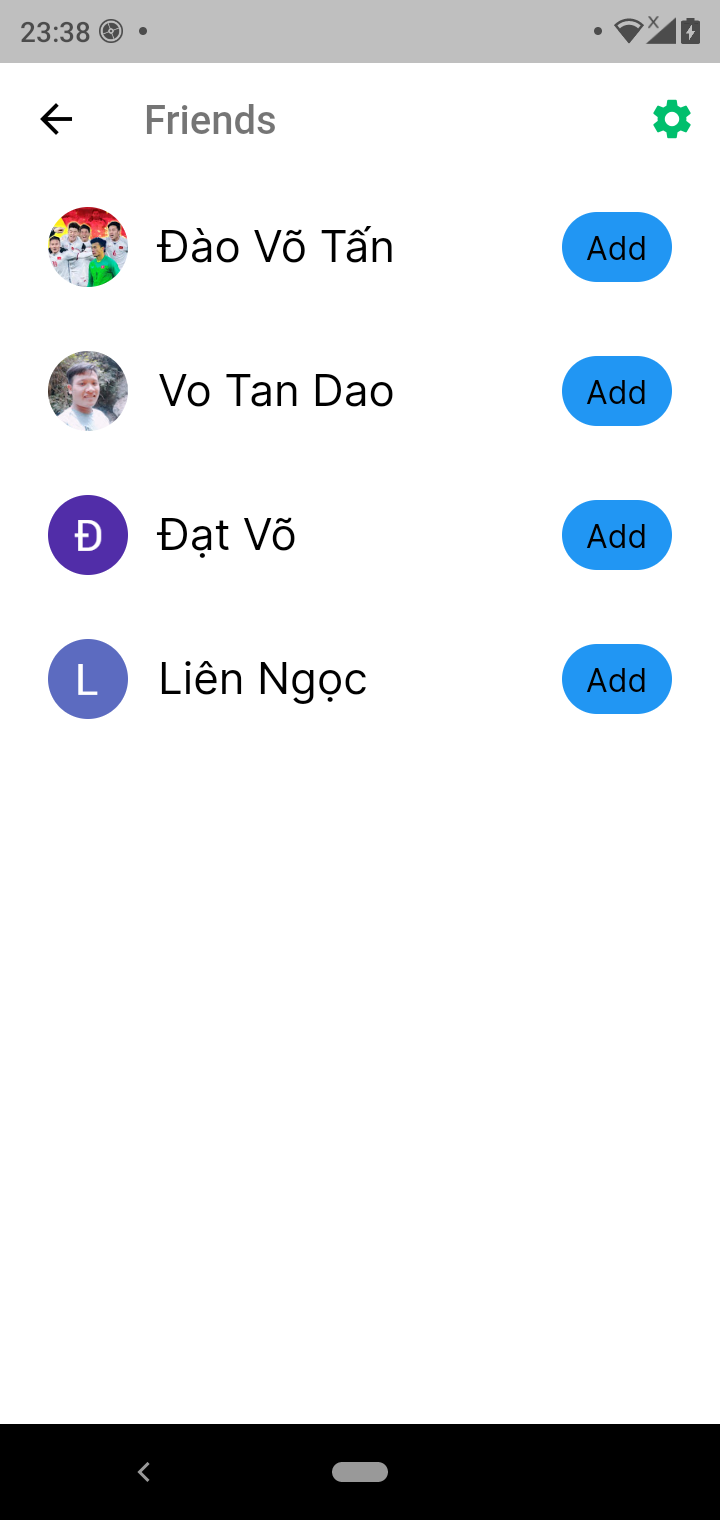
### Giao diện chính

****

*Hình 17. Giao diện chính của app.*

Sau khi đăng nhập thành công thì ứng dụng sẽ vào màn hình chính như trên có danh sách các người dùng đã chat. Khi người dùng đã chat với nhau thì trên giao diện chính sẽ hiển thị danh sách tin nhắn. Phía dưới là 4 Tabbar cho phép người dùng nhấn vào từng tab sẽ chuyển sang màn hình khác để thao tác.

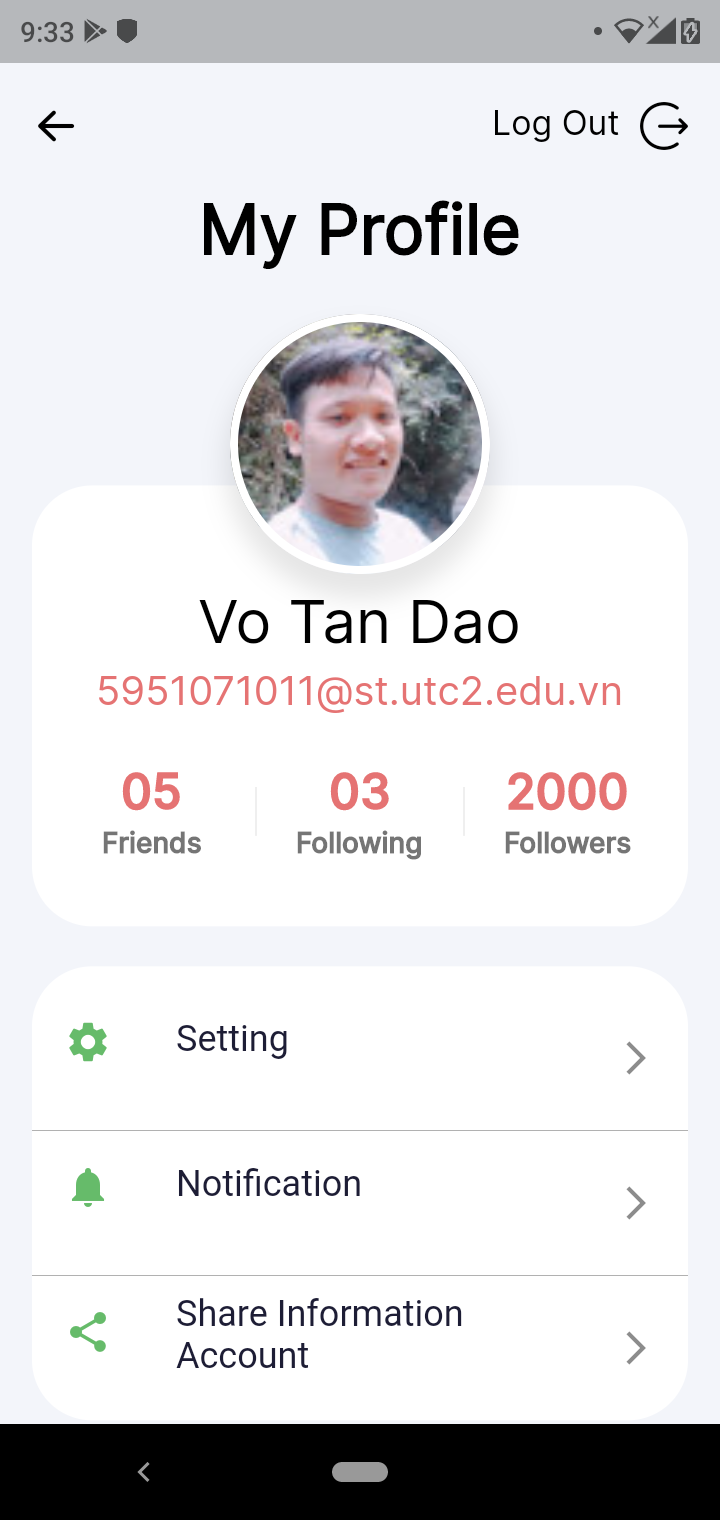
### Giao diện thêm bạn bè



*Hình 18. Giao diện thêm bạn bè.*

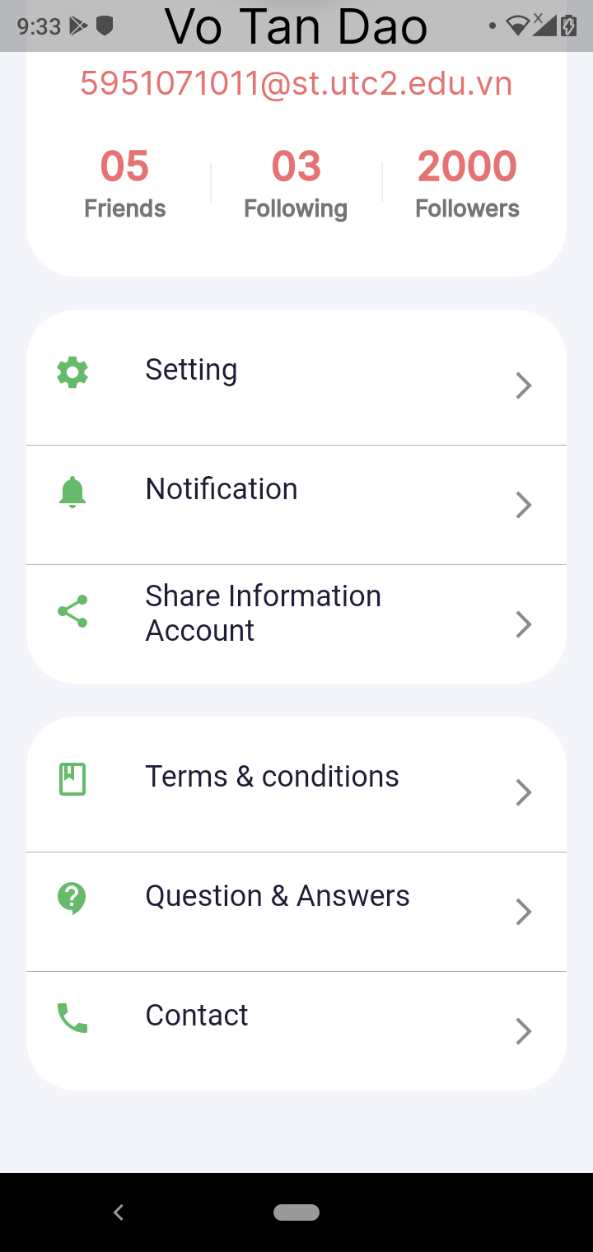
Đây là giao diện cho phép chúng ta có thể thêm bạn thông qua tên của họ, và từ đây chúng ta có thể gửi lời mời kết bạn đến họ và sẽ thật tuyệt nếu như họ chấp nhận lời mời, các bạn có thể tiến hành giao lưu với nhau. Tuy nhiên ở đây em mới chỉ dừng ở mức lấy những user từ firebase về nhưng chưa xử lý việc thêm bạn bè.

### Giao diện Profile và Setting



*Hình 19. Giao diện Profile.*

Như đã thấy, sau khi đăng nhập thành công thì bạn có thể xem được Profile của mình các thông tin sau khi đăng nhập sẽ được lưu lại trên firebase và lấy nó về để hiển thị lên Profile.



*Hình 20. Giao diện setting.*

Giao diện setting cho phép người dùng cài đặt các phương thức mà người dùng mong muốn, khi nhấn vào từng item sẽ xuất hiện lên màn hình 1 showModalButtonSheet để có thể lựa chọn các yêu cầu mà người dùng yêu cầu.

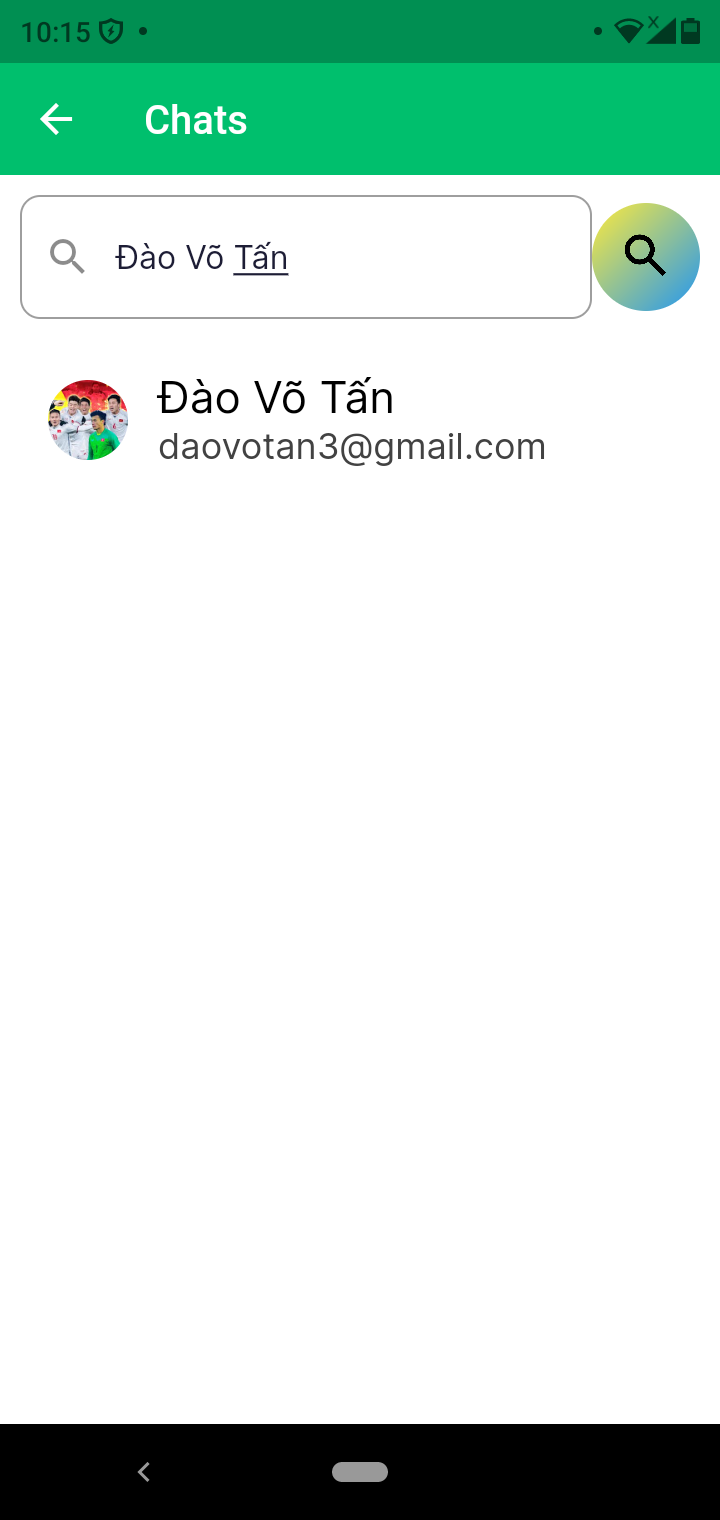
### Giao diện nhắn tin



*Hình 21. Giao diện chat giữa 2 user.*

Người dùng chat với nhau bằng cách nhập đoạn text hoặc icon trong TextField, sau đó nhấn vào button gửi thì đoạn text đó sẽ xuất lên màn hình đồng thời cũng sẽ được lưu lại trên firebase. Người gửi sẽ có giao diện màu xanh bên phải còn người nhận có giao diện màu xám bên trái.

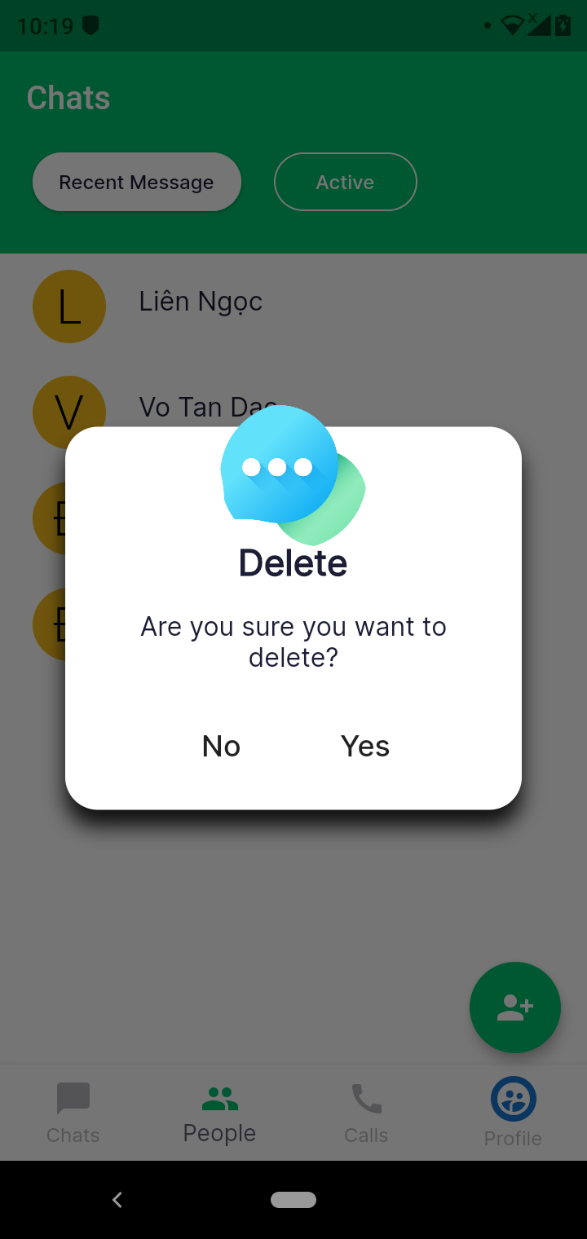
### Giao diện search



*Hình 22. Giao diện tìm kiếm user để chat.*

Giao diện search cho phép người dùng thực hiện tìm kiếm người bạn muốn chat thông qua tên người dùng. Khi người dùng thực hiện nhập tên người dùng muốn tìm kiếm thì trường đó sẽ được lấy từ firebase về và hiển thị lên listView.

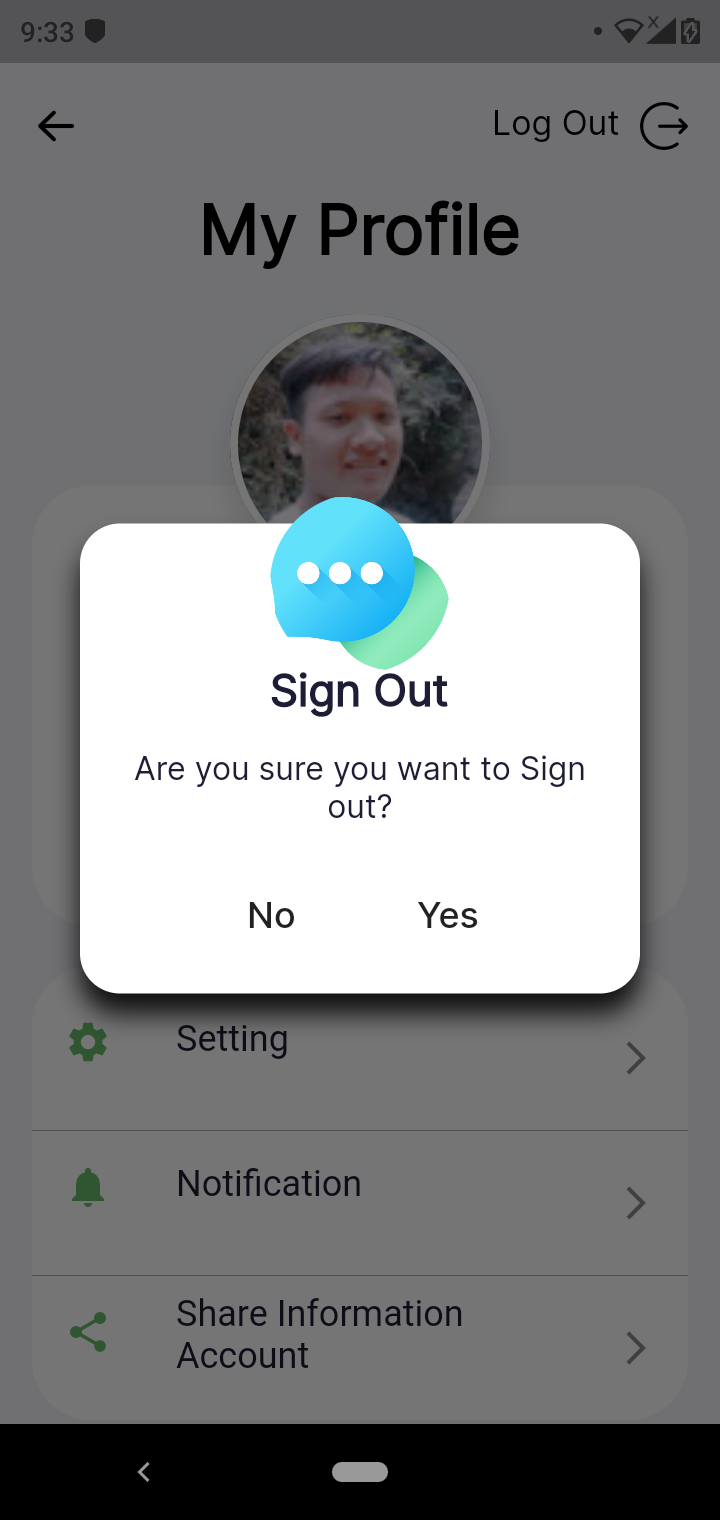
### Xoá tin nhắn

**

*Hình 23. Xóa tin nhắn.*

Khi người dùng chạm giữ 1 đối tượng trong danh sách các tin nhắn thì ứng dụng sẽ xuất lên màn hình 1 hộp thoại AlertDialog cho người dùng chọn xoá tin nhắn hoặc không. Nếu “yes” thì tin nhắn sẽ được xoá và danh sách tin nhắn sẽ xoá đối tượng đó đi, đồng thời dữ liệu trên firebase cũng được xoá.

### Đăng xuất



*Hình 24. Giao diện đăng xuất.*

Khi người dùng muốn đăng xuất thì ứng dụng sẽ xuất lên màn hình hộp thoại AlertDialog cho người dùng lựa chọn đăng xuất hay không. Nếu “Yes” thì ứng dụng sẽ đăng xuất và thoát ra màn hình đăng nhập, ngược lại “No” thì ứng dụng quay lại màn hình chính.

## Kết quả và đánh giá

* Kết quả
  + Ứng dụng đã được phát triển thành công. Đảm bảo đầy đủ các tính năng cơ bản của một ứng dụng nhắn tin.
  + Nhờ có Realtime Database mà dữ liệu tin nhắn, dữ liệu cá nhân liên tục được cập nhật ngay lập tức, tốc độ truy xuất dữ liệu nhanh và tối ưu.
  + Dữ liệu được lưu trữ trực tiếp trên server, tiện lợi trong việc load dữ liệu. Cải thiện được thời gian truy xuất dữ liệu.
  + Giao diện người dùng thân thiện, dễ sử dụng.
* Đánh giá

Ứng dụng hoạt động tốt trên nền tảng smartphone, giao diện thân thiện và dễ sử dụng. Em đã có phát triển thêm các icon để làm tăng thêm độ phong phú cho tin nhắn, giúp ứng dụng trở nên thú vị và có hồn hơn thay vì chỉ gửi đi những dòng tin nhắn chữ nhàm chán.

# **CHƯƠNG 4: KẾT LUẬN**

* + - 1. **Kết quả đạt được**
* Về tri thức:

Trong suốt khoảng thời gian 5 tuần, em đã học được rất nhiều về kiến thức nền tảng, các phương pháp tìm và đọc tại liệu cũng như kiến thức về lĩnh vực chuyên môn về đề tài mà em đang theo đuổi.

* Về ứng dụng:

Tính năng chat giữa 2 user đã được thực hiện đồng thời các user có thể add friend và xem profile của mình, tuy nhiên còn một số lỗi mà em chưa khắc phục được.

* Về con người:

Thật sự thì khoảng thời gian là khoảng thời gian em gặp rất nhiều khó khăn. Từ những khó khăn trong lúc tìm, đọc tài liệu vì em mới tìm hiểu về flutter chưa lâu. Nhưng cũng không vì thế mà em bỏ cuộc. Thay vì nản chí, em đã dùng chính những khó khăn, gian khổ đó như một ngòi nổ, một nguồn động lực to lớn để có thể tiếp tục con đường đang còn dang dở phía trước. Giúp em có thêm nghị lực và ý chí kiên cường, do có khó khăn thế nào thì em cũng sẽ vượt qua bằng mọi giá.

Tuy chỉ là 1 project báo cáo thực tập chuyên môn và quy mô thật sự là không lớn nhưng em đã học được rất nhiều thứ bổ ích từ đây: kỹ năng tìm đọc tài liệu, kỹ năng làm báo cáo, kỹ năng lập trình,… tất cả đều được cải thiện một cách rõ rệt. Đây sẽ là bước đệm để giúp em tiến ra biển lớn, tiến ra dòng đời xã hội khắc nghiệt. Và gặt hái được nhiều thành công trong cuộc sống. Trở thành một người thật sự có ích cho xã hội này.

Sau khi trải qua rất nhiều khó khăn và gian khổ, em đã hoàn thành đầy đủ các yêu cầu đã được đề ra trong nhiệm vụ thiết kế tốt nghiệm.

* + - 1. **Hạn chế tồn tại**

Vì thời gian thực hiện đề tài cũng như kiến thức của bản thân là có giới hạn nên vẫn còn vài thiếu sót như: chưa thể add friend được, phần UI/UX chưa thật sự hợp lý và một số lỗi chưa khắc phục được. Đây là một trong những sai sót lớn nhất của em.

* + - 1. **Hướng phát triển.**

Nếu có thời gian và điều kiện để tiếp tục thực hiện em sẽ cố gắng để hoàn thành ứng dụng của em một cách thật trọn vẹn. Em sẽ phát triển thêm nhiều tính năng ứng dụng như: video call, tìm bạn bè quanh đây thông qua thiết bị định vị GPS, chat bảo mật…

Mong muốn rằng một ngày nào ứng dụng của mình sẽ ngày một phát triển mở rộng khắp giống như zalo hay messenger. Trở thành một ứng dụng có ích cho xã hội và em tin rằng ứng dụng của mình sẽ là một ứng dụng phổ biến từ đó có thể phát triển thành 1 mạnh xã hội cho người việt.

# **TÀI LIỆU THAM KHẢO**

[1]. Alberto Miola, *Flutter complete reference*, 2020

[2]. Gilad Bracha, *Dart Programming Language Specification*, 2019

[3]. Gilad Bracha, The Dart Programming Language, 2015

[4]. *Cook book flutter.*

<https://flutter.dev/docs/cookbook> truy cập vào 9:23 23/5/2021

[5]. *Flutter TDD Clean Architecture Course [1] Explanation & Project Structure.*

<https://resocoder.com/2019/08/27/flutter-tdd-clean-architecture-course-1-explanation-project-structure> truy cập vào 23:24 15/5/2021

[6]. *Cloud Firestore.*

<https://firebase.flutter.dev/docs/firestore/usage/> truy cập vào lúc 8:30 16/5/2021

[7]. *Signing In with a Google Account in Flutter via Firebase.*

<https://www.youtube.com/watch?v=g3K8wgqE3XA>

[8]. Flutter Chat UI Tutorial.

<https://www.youtube.com/watch?v=h-igXZCCrrc> truy cập vào lúc 22:00 5/6/2021

[9]. Getting started with Firebase on Flutter – Firecasts.

<https://www.youtube.com/watch?v=EXp0gq9kGxI> truy cập vào lúc 11:32 16/6/2021

[10]. Shared Preferences.

<https://www.youtube.com/watch?v=uyz0HrGUamc> truy cập vào lúc 15:34 17/6/2021