**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**Nghiên cứu và XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ NHÂN SỰ**

Giảng viên hướng dẫn: Trần Thị Dung

Sinh viên thực hiện: Nguyễn Thị Mỹ Anh

Đặng Bình Phước

Trần Lê Xuân Sơn

Lớp: CQ.59.CNTT

Khoá: 59

TP. Hồ Chí Minh, năm 2020

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**Nghiên cứu và XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ NHÂN SỰ**

Giảng viên hướng dẫn: Trần Thị Dung

Sinh viên thực hiện: Nguyễn Thị Mỹ Anh

Đặng Bình Phước

Trần Lê Xuân Sơn

Lớp: CQ.59.CNTT

Khoá: 59

TP. Hồ Chí Minh, năm 2020

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH Độc lập – Tự do – Hạnh phúc**

# **NHIỆM VỤ THIẾT KẾ CƠ SỞ DỮ LIỆU**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

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

**Mã sinh viên:** 5951071001 **Họ tên SV:** Nguyễn Thị Mỹ Anh

**Mã sinh viên:** 5951071059 **Họ tên SV:** Đặng Bình Phước

**Mã sinh viên:** 59510710 **Họ tên SV:** Trần Lê Xuân Sơn

**Khóa:** 59 **Lớp:** CQ.59.CNTT

1. **Tên đề tài**

NGHIÊN CỨU VÀ PHÁT TRIỂN CƠ SỞ DỮ LIỆU QUẢN LÝ NHÂN SỰ

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

Nhằm phục vụ cho nhu cầu quản lý nhân viên được hiệu quả.

* 1. **Yêu cầu:**
* **Yêu cầu công nghệ**
  + Sử dụng ngôn ngữ truy vấn dữ liệu.
  + Sử dụng công cụ SQL Sever.
* **Yêu cầu chức năng**
* **Yêu cầu phi chức năng**

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.
      * Lập trình xây dựng.
      * Kiểm thử và chạy thực nghiệm.
      * Kết quả thu được.
   2. **Phạm vi:**
      * Nghiên cứu công cụ Sql Sever.
2. **Công nghệ, công cụ và ngôn ngữ lập trình**
   * + Công cụ Sql Sever.
     + Ngôn ngữ truy vấn dữ liệu: SQL.
3. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**
   * + Xây dựng thành công cơ sở dữ liệu phù hợp thực tế đã nếu trên.

# **LỜI CẢM ƠN**

Lời đầu tiên chúng 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 chúng em kho tàng kiến thức về bầu trời công nghệ thông tin rộng lớn. Ở đây, chúng em không chỉ học được kiến thức về sách vở mà còn học được các bài học, kỷ năng sống. Đặc biệt, chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến cô **Trần Thị Dung**, người đã giúp chúng em trong quá trình làm bài tập lớn, người đã bỏ thời gian quý báu, thậm chí là thời gian nghỉ ngơi để hướng dẫn, để giúp chúng em có thêm kinh nghiệm để quản lý cơ sở dữ liệu một cách hiệu quả nhất. Chúng em thật chẳng biết dùng lời nào để diễn tả được công lao của cô. “**CẢM ƠN CÔ**. **CẢM ƠN CÔ VỀ TẤT CẢ**”.

Trong quá trình học tập và tìm hiểu chúng em đã nổ lực rất nhiều với mong muốn hoàn thành bài tập lớn 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 chúng em thì sai lầm là không thể không mắc phải. Chúng em mong thầy, cô bộ môn có thể thông cảm và cho chúng em những ý kiến, đóng góp để chúng em có thể hoàn thành đồ án của mình một cách tọn vẹn nhất trước khi rời xa ngôi trường thân yêu này.

Sau cùng, chúng 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 như trong cuộc sống.

Chúng em xin chân thành cảm ơn!

# **NHẬN XÉT CỦA GIÁO VIÊN**

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

Giáo viên hướng dẫn

MỤC LỤC

[**NHIỆM VỤ THIẾT KẾ CƠ SỞ DỮ LIỆU** i](#_Toc11394917)

[**LỜI CẢM ƠN** iii](#_Toc11394918)

[**NHẬN XÉT CỦA GIÁO VIÊN** iv](#_Toc11394919)

[**MỞ ĐẦU** 1](#_Toc11394921)

[**1.** **Lý do chọn đề tài.** 1](#_Toc11394922)

[**2.** **Hướng tiếp cận của đề tài.** 1](#_Toc11394923)

[**3.** **Mục tiêu nghiên cứu.** 1](#_Toc11394924)

[**4.** **Đối tượng và phạm vi nghiên cứu.** 2](#_Toc11394925)

[**5.** **Phương pháp nghiên cứu.** 2](#_Toc11394926)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 3](#_Toc11394927)

[**1.1** **Hệ điều hành Android.** 3](#_Toc11394928)

[**1.1.1** **Giới thiệu.** 3](#_Toc11394929)

[**1.1.2** **Khái niệm.** 3](#_Toc11394930)

[**1.1.3** **Kiến trúc của hệ điều hành Android.** 4](#_Toc11394931)

[**1.1.4** **Ưu và nhược điểm.** 5](#_Toc11394932)

[**1.2** **Ngôn ngữ lập trình Java.** 6](#_Toc11394933)

[**1.2.1** **Giới thiệu.** 6](#_Toc11394934)

[**1.2.2** **Khái niệm.** 6](#_Toc11394935)

[**1.2.3** **Các đặc điểm của ngôn ngữ lập trình java**. 7](#_Toc11394936)

[**1.3** **Firebase.** 8](#_Toc11394937)

[**1.3.1** **Giới thiệu.** 8](#_Toc11394938)

[**1.3.2** **Khái niệm.** 9](#_Toc11394939)

[**1.3.3** **Các chức năng chính của Firebase.** 9](#_Toc11394940)

[**1.3.4** **Những lợi ích từ Firebase.** 14](#_Toc11394941)

[**1.3.5** **Nhược điểm của Firebase.** 16](#_Toc11394942)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ** 17](#_Toc11394943)

[**2.1** **Đặc tả bài toán.** 17](#_Toc11394944)

[**2.2** **Những khó khăn gặp phải.** 17](#_Toc11394945)

[**2.3** **Đăng ký thành viên.** 19](#_Toc11394946)

[**2.4** **Kết bạn.** 20](#_Toc11394947)

[**2.5** **Giới thiệu bạn bè cho nhau.** 23](#_Toc11394948)

[**2.6** **Nhắn tin giữa 2 user với nhau.** 26](#_Toc11394949)

[**2.7** **Nâng cấp bảo mật.** 30](#_Toc11394950)

[**CHƯƠNG 3: THỬ NGHIỆM ỨNG DỤNG** 31](#_Toc11394951)

[**3.1** **Môi trường cài đặt.** 31](#_Toc11394952)

[**3.2** **Giao diện đăng ký thành viên.** 31](#_Toc11394953)

[**3.3** **Giao diện đăng nhập tự động.** 33](#_Toc11394954)

[**3.4** **Giao diện thêm bạn.** 34](#_Toc11394955)

[**3.5** **Giao diện danh sách bạn bè khi đã kết bạn thành công.** 36](#_Toc11394956)

[**3.6** **Giao diện giới thiệu bạn bè.** 37](#_Toc11394957)

[**3.7** **Giao diện thiết lập mật mã giữa 2 user.** 38](#_Toc11394958)

[**3.8** **Giao diện nến người dùng muốn thay đổi khóa.** 39](#_Toc11394959)

[**3.9** **Giao diện nếu người dùng muốn xóa mật mã.** 40](#_Toc11394960)

[**3.10** **Kết quả và đánh giá.** 41](#_Toc11394961)

[Kết quả. 41](#_Toc11394962)

[Đánh giá. 41](#_Toc11394963)

[**KẾT LUẬN VÀ KHUYẾN NGHỊ** 42](#_Toc11394964)

[Kết quả đạt được 42](#_Toc11394965)

[Tồn tại. 42](#_Toc11394966)

[Hướng phát triển. 42](#_Toc11394967)

[**TÀI LIỆU THAM KHẢO** 43](#_Toc11394968)

# **MỞ ĐẦU**

1. **Lý do chọn đề tài.**

Chúng ta, thế hệ đang sống trong thời đại công nghiệp hóa, hiện đại hóa. Thời đại mà công nghệ thông tin đang ngày một phát triển như vũ bão. Nó có mặt hầu hết trong tất cả các lĩnh vực mà chúng ta đang sinh hoạt, làm việc và cũng như giải trí….Nó giúp đỡ cho chúng ta từ nông nghiệp, công nghiệp, kinh tế, giáo dục, y tế. Việc giao tiếp giữa con người với nhau cũng theo đó mà dần phát triển. và dưới sự giúp đỡ của công nghệ thông tin hiện nay chúng ta đã có giữ liên lạc với nhau một cách dễ dàng hơn thông qua các trang mạng xã hội, các ứng dụng chat với nhau. Nó vừa nhanh lại vừa tiết kiệm hơn rất nhiều so với việc viết thư,….

Nhưng theo ý kiến của em vậy là chưa đủ. Sẽ có những lúc nào đó chúng ta sẽ phải đối mặt với những tình huống, những trường hợp như bị xem trộm tin nhắn, rơi điện thoại, và sẽ rất là phiền phức nếu như thông tin cá nhân của mình bị tuồn ra ngoài một cách không kiểm soát như vậy. Chẳng hạn như trong vấn đề yêu đương những tin nhắn sến súa, mùi mẫn là không thể tránh khỏi. Và sẽ càng tệ hơn khi đoạn hội thoại đó lên quan đến một cuộc giao dịch. Cụ thể là số tài khoản ngân hàng hay là những giấy tờ quan trọng nào khác,….có thể gây thiệt hại về vật chất và tinh thần.

Chính vì thế mà em đã nảy ra ý tưởng là kết hợp giữa một ứng nhắn tin với một thuật toán bảo mật và cho ra đời một sản phẩm mới, một ứng dụng mới. Và em gọi nó là ứng dụng tin nhắn bảo mật.

1. **Hướng tiếp cận của đề tài.**

* Thiết kế cơ sở dữ liệu sao cho phù hợp với thực tế doanh nghiệp.
* Nắm vững kiến thức nền tảng của ngôn ngữ truy xuất dữ liệu SQL, tiến đến phát triển cơ sở dữ liệu.
* Tiến hành kiểm tra và chạy thử.

1. **Mục tiêu nghiên cứu.**

* Phát triển thành công cơ sở dữ liệu.
* Hiểu được cách thức hoạt động của cơ sở dữ liệu.
* Khiến ứng dụng trở nên có ích cho xã hội, phù hợp với doanh nghiệp.

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

* Đối tượng: Mọi người đều có thể sử dụng.
* Phạm vi ứng dụng: Mong muốn ứng dụng có thể có mặt hầu hết các công ty.

1. **Phương pháp nghiên cứu.**

* Tài liệu: Tìm hiểu và khai thác các tài liệu liên quan đến cơ sở dữ liệu, các nền tảng kiến thức về SQL cũng như công cụ SQL Server.
* Thực hành: Tiến hành thiết kế cơ sở dữ liệu từ thực tế. Tiến hành thao tác với dữ liệu.

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. **Hệ điều hành Android.**
2. **Giới thiệu.**

**Hệ điều hành Android** đang chiếm hơn 80% thị trường thiết bị di động hiện đại ngày nay. Nhu cầu sử dụng ứng dụng di động cũng ngày càng cao. Nhu cầu việc làm về lập trình trên hệ điều hành mobile phổ biến nhất thế giới này cũng vì thế mà tăng mạnh do tính mở và dễ tiếp cận của nó.

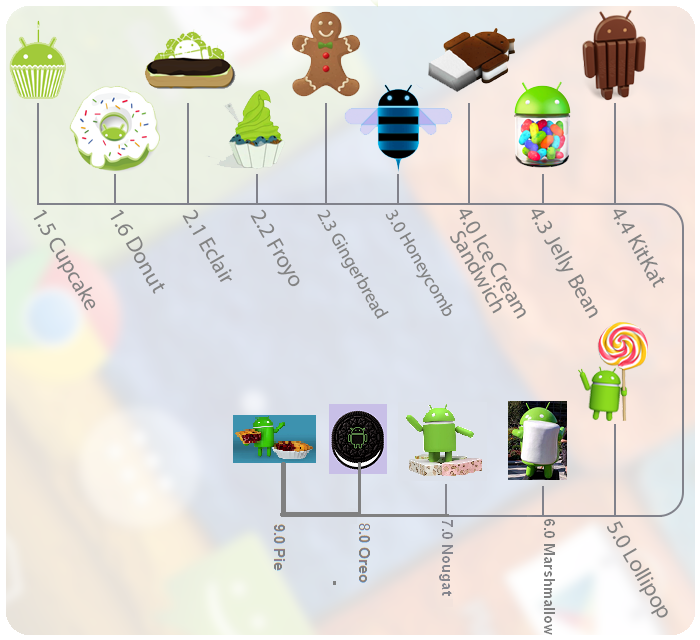
1. **Khái niệm.**

Android là một HỆ ĐIỀU HÀNH MÃ NGUỒN MỞ, ban đầu hệ điều hành này được xây dựng hướng đến việc sử dụng trên các điện thoại di động (sau này nó tiếp tục được phát triển để sử dụng rộng rãi trên các máy tính bảng, đầu phát HD, TV, Wearable, ..). Hệ điều hành Android được phát triển bởi Google và dựa trên nền tảng Linux.

Có nghĩa là Android không chỉ giới hạn trong phạm vi một hệ điều hành cho điện thoại! Nó có thể được nhà sản xuất cài đặt lên đồng hồ, máy nghe nhạc, thiết bị định vị GPS, thậm chí là ô tô (các thiết bị Android Auto).

Android Có khả năng tùy biến rất cao và có thể chạy trên nhiều thiết bị, nhiều kiến trúc vi xử lý (ARM / x86). Tính đến nay, Android đã có các phiên bản (kèm tên mã) lần lượt là:

* Android 1.5 Cupcake
* Android 1.6 Donut
* Android 2.1 Eclair
* Android 2.2 Froyo
* Android 2.3 Gingerbread
* Android 3.2 Honeycomb – Phiên bản Android đầu tiên được thiết kế cho máy tính bảng.
* Android 4.0 Ice Cream Sandwich
* Android 4.1 Jelly Bean
* Android 4.2 Jelly Bean
* Android 4.3 Jelly Bean
* Android 4.4 KitKat
* Android 5.0 Lollipop
* Android 6.0 Marshmallow
* Android 7.0 Nougat
* Android 8.0 Oreo

****

*Hình 1. 1 - Các phiên bản của hệ điều hành Android.*

1. **Kiến trúc của hệ điều hành Android.**



*Hình 1. 2 - Kiến trúc hệ điền hành Android*

**Tầng Applications:** Là tầng chứa các ứng dụng Danh bạ, Gọi điện, Trình duyệt, Nghe nhạc,… các ứng dụng này thường mua máy về chúng ta đã có sẵn rồi.

**Tầng Framework:**Là tầng chứa các API để làm việc với hệ điều hành như lấy thông tin danh bạ, quản lý các Activity (Activity là gì thì giờ chúng ta chưa cần quan tâm, các bài sau sẽ giải thích kĩ), quản lý địa điểm, quản lý các View (cũng chưa cần quan tâm).

**Tầng Libraries:** Chứa các thư viện, API gần như là cốt lõi của Android, bao gồm bộ quản lý bề mặt cảm ứng (Surface Manager), OpenGL (phục vụ cho việc dựng đồ họa phức tạp),…

**Tầng Android Runtime:**Chứa các thư viện lõi của Android và máy ảo Dalvik Virtual Machine (từ Android 4 trở lên chúng ta có thêm máy ảo ART).

**Tầng Kernel:**Là nhân lõi của hệ điều hành, chứa các tập lệnh, driver giao tiếp giữa phần cứng và phần mềm của Android.

1. **Ưu và nhược điểm.**

* **Ưu điểm:**
* Là hệ điều hành có mã nguồn mở nên khả năng tuỳ biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.
* Đa dạng sản phẩm, rất nhiều hãng điện thoại, thiết bị công nghệ đã ưu ái chọn Android cho thiết bị của họ, giá cả thì hợp lý từ bình dân đến cao cấp.
* Kho ứng dụng Google Play Store đồ sộ.
* Thân thiện và dễ sử dụng.
* Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.
* **Nhược điểm:**
* Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.
* Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.
* Sự phân mảnh lớn. Trong khi một số thiết bị Android xuất sắc đã trình làng như [**Galaxy S5**](https://www.dienmayxanh.com/dien-thoai/samsung-galaxy-s5-g900)**,**[**Galaxy Note 4**](https://www.dienmayxanh.com/dien-thoai/samsung-galaxy-note-4)**,**[**Xperia Z3**](https://www.dienmayxanh.com/dien-thoai/sony-xperia-z3)…, vẫn còn rất nhiều sản phẩm giá rẻ bình thường khác.
* Cập nhật không tự động với tất cả thiết bị. Khi một phiên bản hệ điều hành mới ra mắt, không phải tất cả sản phẩm đều được cập nhật, thậm chí nếu muốn trải nghiệm bạn thường xuyên phải mua mới thiết bị.

1. **Ngôn ngữ lập trình Java.**
2. **Giới thiệu.**

Ngôn ngữ [lập trình Java](http://hoclaptrinhweb.org/hoc-lap-trinh/hoc-lap-trinh-java.html) ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995 (Java 1.0 [J2SE]). Với ưu thế về đa nền tảng (multi platform) Java càng lúc càng được ứng dụng rộng rãi trên nhiều thiết bị từ máy tính đến mobile và nhiều thiết bị phần cứng khác...

1. **Khái niệm.**

Java là một trong những ngôn ngữ lập trình hướng đối tượng, được thiết kế độc lập với hệ điều hành, cho phép người lập trình viết chương trình một lần và có thể sử dụng tại bất kỳ đâu. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

1. **Các đặc điểm của ngôn ngữ lập trình java**.

* **Hướng đối tượng** – Trong Java, mọi thứ đều là một Object. Java 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.
* **Nền tảng độc lập** – Không giống nhiều ngôn ngữ lập trình khác bao gồm cả C và C ++, khi Java được biên dịch, nó không được biên dịch thành ngôn ngữ máy nền tảng cụ thể, thay vào mã byte – nền tảng độc lập. Mã byte này được thông dịch bởi máy ảo (JVM) trên nền tảng nào đó mà nó đang chạy.
* **Đơn giản** – Java được thiết kế để dễ học. Nếu bạn hiểu khái niệm cơ bản về OOP Java, sẽ rất dễ để trở thành master về java.
* **Bảo mật** – Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virut, giả mạo. Các kỹ thuật xác thực dựa trên mã hoá khóa công khai.
* **Kiến trúc – trung lập** – Trình biên dịch Java tạo ra định dạng tệp đối tượng kiến trúc trung lập, làm cho mã biên dịch được thực thi trên nhiều bộ vi xử lý, với sự hiện diện của hệ điều hành Java.
* **Portable** – Là kiến trúc tập trung và không có khía cạnh thực hiện phụ thuộc của đặc tả này làm cho Java khả chuyển. Trình biên dịch trong Java được viết bằng ANSI C, đó là một tập con POSIX.
* **Mạnh mẽ** – Java làm nỗ lực để loại trừ các tình huống dễ bị lỗi bằng cách kiểm tra lỗi tại thời gian biên dịch và kiểm tra lỗi tại runtime.
* **Đa luồng** – Với tính năng đa luồng của Java có thể viết các chương trình có thể thực hiện nhiều tác vụ đồng thời. Tính năng thiết kế này cho phép các nhà phát triển xây dựng các ứng dụng tương tác có thể chạy trơn tru hơn.
* **Thông dịch** – Mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.
* **Hiệu năng cao** – Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.
* **Phân tán** – Java được thiết kế cho môi trường phân tán của Internet.
* **Năng động** – Java là năng động hơn C hoặc C++ vì nó được thiết kế để thích nghi với môi trường đang phát triển. Các chương trình Java có thể mang một lượng lớn thông tin tại runtime mà có thể được sử dụng để xác minh và giải quyết các truy cập vào các đối tượng tại runtime.

1. **Firebase.**
2. **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.

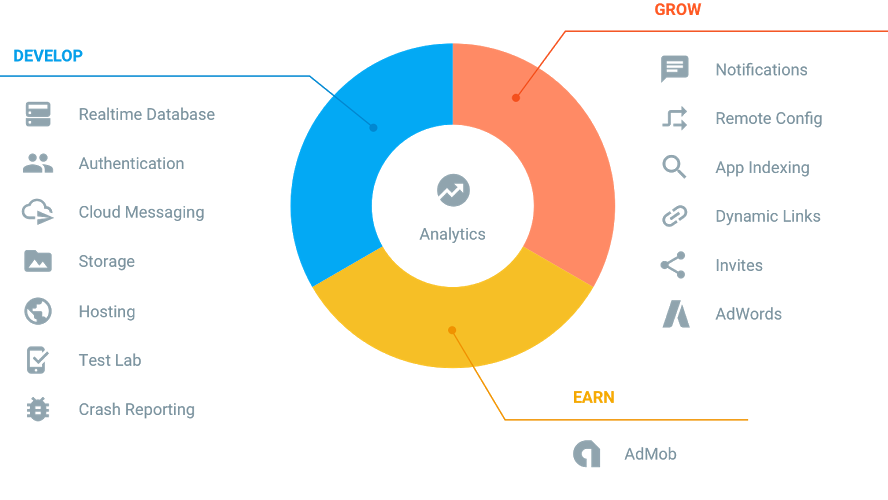
1. **Khái niệm.**

**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.

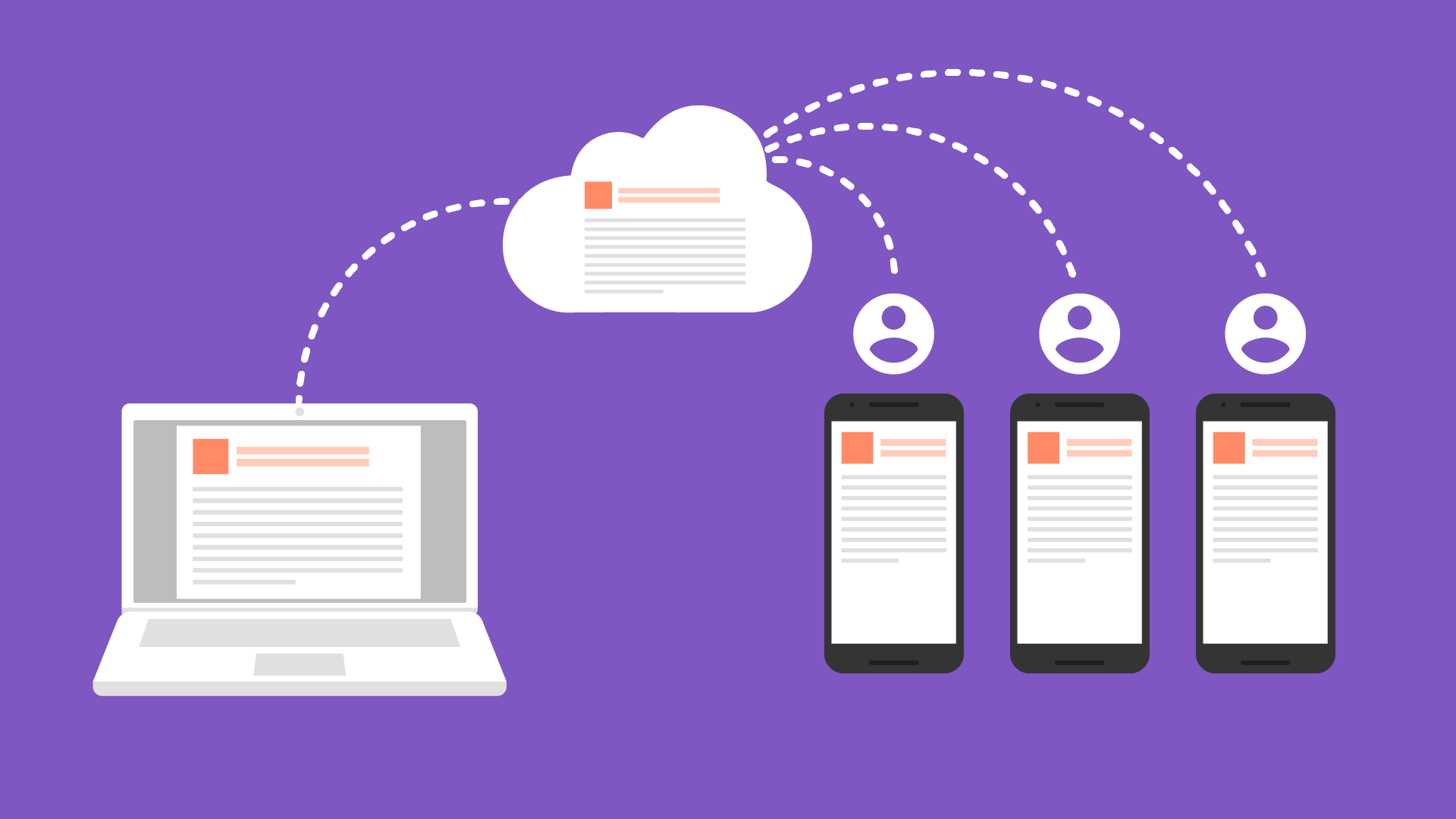
Firebase là sự kết hợp giữa nền tảng cloud với hệ thống máy chủ cực kì mạnh mẽ tới từ Google, để cung cấp cho chúng ta những API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database

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



*Hình 1. 3 - Các chức năng chính của Firebase.*

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



*Hình 1. 4 - 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.

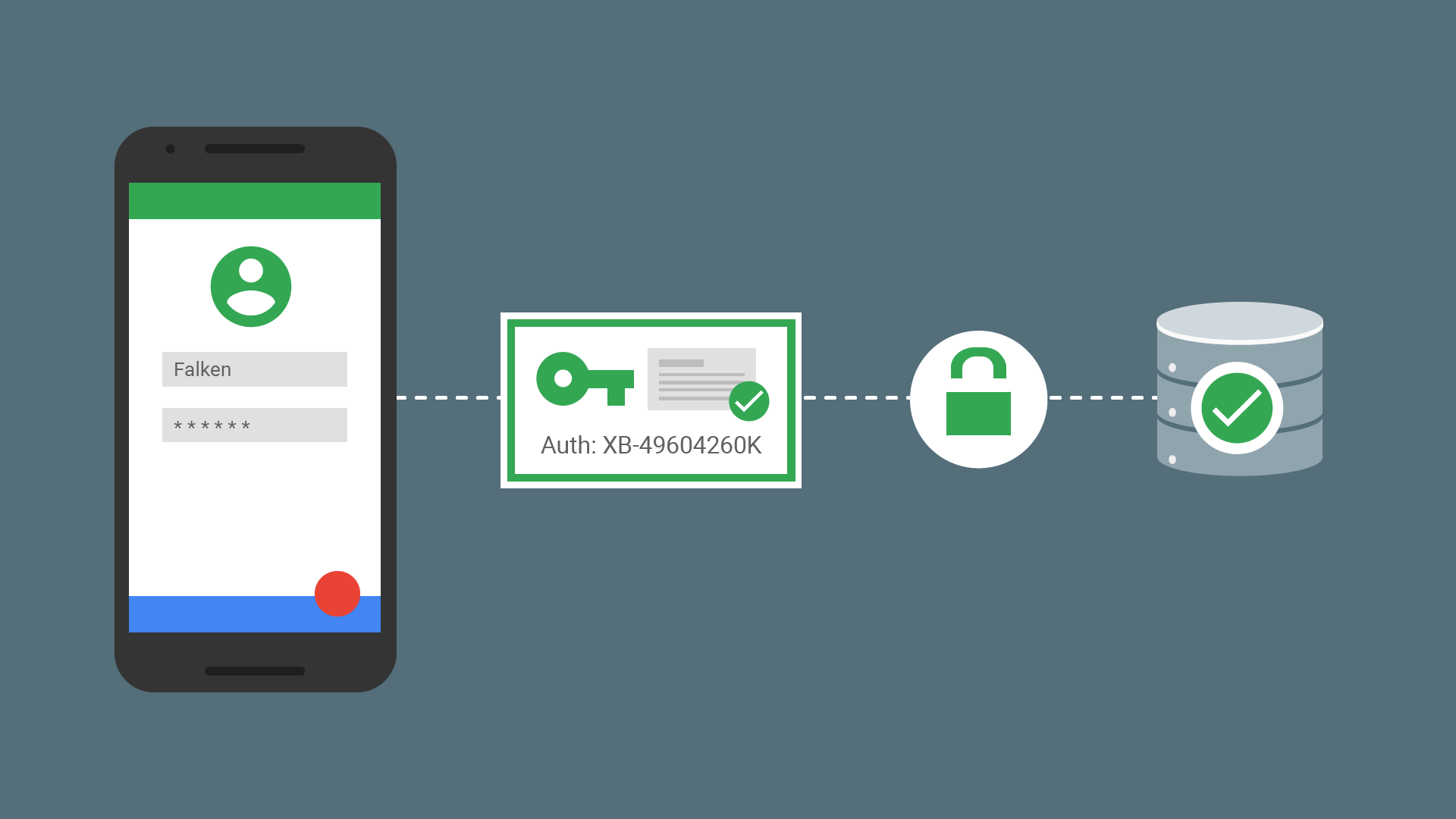
Đồng bộ hóa trong thời gian thực giúp người dùng dễ dàng truy cập dữ liệu của họ từ bất kỳ thiết bị nào, có thể là web hoặc thiết bị di động. Cơ sở dữ liệu thời gian thực cũng giúp người dùng có khả năng cộng tác với nhau.

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 1. 5 - Firebase Authentication.*

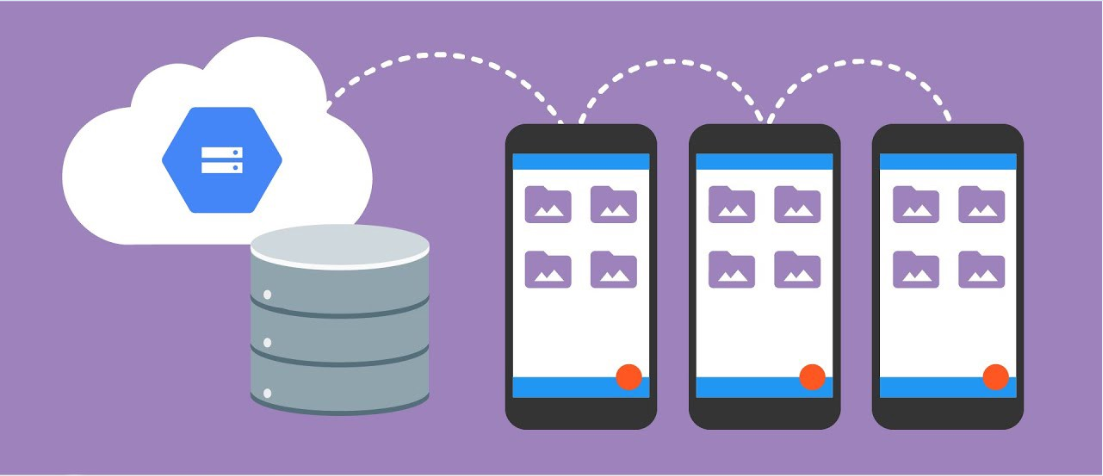
Với Firebase bạn có thể dễ dàng tích hợp các công nghệ xác thực của Google, Facebook, Twitter, … hoặc một hệ thống xác thực mà bạn tự mình tạo ra vào trong ứng dụng của bạn ở bất kì nền tảng nào như Android, iOS hoặc Web.

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 1. 6 - 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 . Qủa thật đây là một dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và hiệu quả chi phí.

**Firebase Hosting.**



*Hình 1. 7 - 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.  
Điểm nổi bật:

|  |  |
| --- | --- |
| Server với kết nối an toàn | Cấu hình SSL được thêm vào Firebase Hosting   nên nội dung luôn được phân phối một cách an toàn. |
| Tốc độ cao | Mỗi tập tin bạn tải lên sẽ được lưu trữ trên ổ SSD tại CDN. Chính vì vậy mà nội dung được cung cấp nhanh chóng. |
| Triển khai nhanh chóng | Sử dụng Firebase CLI, bạn có thể có được ứng dụng và chạy trong vài giây. Chỉ với một vài lệnh command line để thêm vào ứng dụng của bạn. |
| One-click rollbacks | Triển khai nhanh chóng là rất tốt, nhưng có thể lùi lại sai lầm thậm chí còn tốt hơn. Firebase Hosting cung cấp versioning đầy đủ cho phép rollbacks một cú nhấp chuột. |

Các ứng dụng sẽ được cấp 1 tên miền dạng \*.firebaseapp.com hoặc bạn có thể trả tiền để sử dụng tên miền của riêng mình.

**Firebase Cloud Messaging.**



*Hình 1. 8 - 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.

1. **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.

1. **Nhược điểm của Firebase.**

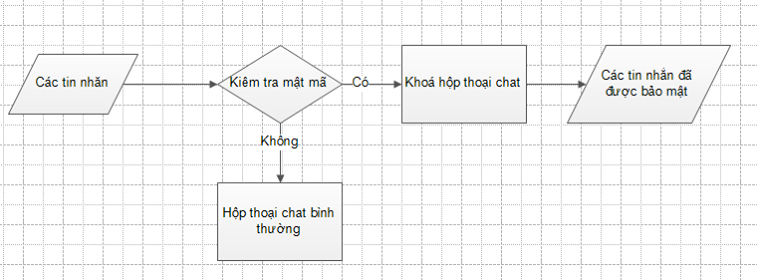
Điểm yếu duy nhất của “Firebase” chính là phần Realtime Database, mà đúng hơn chỉ là phần Database. Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn từ mức đôi chút tới khá nhiều.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ**

1. **Đặc tả bài toán.**

* Dữ liệu đầu vào: các dòng tin nhắn ( chữ, video, hình ảnh, audio, sticker,…)
* Dữ liệu đầu ra:
* Nếu giữa 2 người dùng chưa thiệt lập tính năng bảo mật tin nhắn thì ứng dụng sẽ giống như messenger hay zalo. Chỉ là một ứng dụng nhắn tin bình thường.
* Nhưng nếu người dùng đã kích hoạt lớp bảo mật tin nhắn thì chỉ những ai biết đc sự tồn tại của mật mã thì mới có tư cách vào hộp thư thoại để xem tin nhắn. Và tất nhiên mật mã ở đây được thiết lập cho cả 2 người dùng đang nhắn tin chung với nhau. Còn lại những ai không biết mật mã nhưng vẫn cố tình vào thì sau khi nhập sai đến lần thứ 3, ứng dụng sẽ tự kích hoạt khả năng bảo vệ của nó bằng cách đăng xuất tài khoản của người dùng hiện tại nếu có phát hiện các hành vi như muốn đọc trộm tin nhắn.

Sơ đồ mô tả hệ thống



*Hình 2. 1 Quy trình thực hiện của ứng dụng*

1. **Những khó khăn gặp phải.**

Trước khi bắt đầu học kỳ 2 năm 4 ( tức học kỳ làm đồ án ). Em đã gặp rất nhiều khó khăn trong quá trình lên ý tưởng cho đồ án của mình. Vì là thế hệ sau nên tất cả các ý tưởng mà em nghỉ ra đa phần đều trùng mới những thế hệ trước. Và thật không may các ý tưởng đó đều đã họ nghiên cứu, trình bày và họ đã đạt được những kết quả, những thành tựu vô cùng lớn lao. Và cho dù có làm lại theo ý tưởng của họ đi chăng nữa thì em cũng không thể nào bì kiệp được, không có gì mới mẻ. Tất cả chỉ là làm lại những thành tựu nghiên cứu của người khác, sẽ chẳng có ý nghĩa gì nếu như chúng ta không đổi mới nghiên cứu của người khác theo một hướng phát triển mới, hướng phát triển mang dấu ấn của riêng mình. Lúc đó em thật sự rất bế tắc, em bắt đầu lâm vào tình trạng bí ý tưởng cho đề tài tốt nghiệp. Nổi sợ đó dần dần nuốt lấy em, đã biết bao lần em muốn bỏ đề tài tốt nghiệp, đã biết bao lần em tự nói với mình rằng: “ Có lẽ mình đã chọn sai ngành ?”. “ CÔNG NGHỆ THÔNG TIN ” nghe thôi cũng đã biết được sức nặng và lợi ích đóng góp của nó trong xã hội ngày nay. Nếu chúng ta không phát triển, nếu như chúng ta không đổi mới, thì cuối cũng tự tay chúng ta sẽ đào thải chính mình ra khỏi xã hội mà khoa học công nghệ và cụ thể ở đây là công nghệ thông tin đang phát triển 1 cách chóng mặt nếu như không muốn nói là “ điên cuồng ”. “Vậy làm sao để mình có thể tồn tại được ? Làm sao để ý tưởng của mình là mới nhất, là đặc biệt, là độc quyền ? ” Đây mới chính và vấn đề nản giải. Ngay khoảnh khắc em định từ bỏ thì đột nhiên một dòng ký ức ùa về. “ Nhưng nó là cái gì mới được ? “. Ngồi nhớ lại một hồi, cuối cùng em cũng đã nhớ ra đó là ý tưởng mà em đã nghỉ ra trong lúc học môn Andoird. Ý tưởng làm một ứng dụng nhắn tin nhưng khác ở chỗ ứng dụng này sẽ cho phép người dùng mã hóa các tin nhắn thành các ký tự đặc biệt. Nhưng vì trình độ chuyên môn lẫn hiểu biết về các công cụ còn hạn chế, ý tưởng của em đã mau chóng thất bại. Bây giờ em đã quyết định thử làm lại ý tưởng này một lần nữa, và nhờ sự giúp đỡ từ thầy Trần Phong Nhã em đã có thể định hình được chính xác những gì em cần làm, những gì đang thừa, những gì còn thiếu…

Thoạt đầu khi bắt tay vào thực hiện em đã gặp rất nhiều khó khăn. Đặc biệt là với công cụ Firebase. Nó gần như là một phạm trù cơ sở dữ liệu mới hoàn toàn so với những cơ sở dữ liệu trước đây em đã được học như SQL Server, Oracle,….Đối với SQL Server hay Oracle dữ liệu đề được lưu ở dạng bảng ( table ). Còn với Firebase

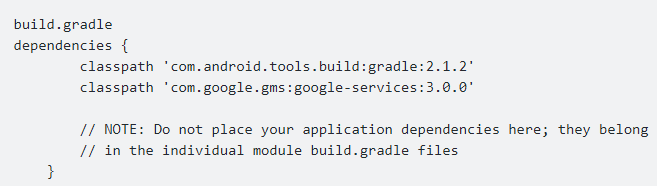
Database Realtime dữ liệu sẽ được lưu ở dạng cây ( tree ) do cách thức hoạt động vô cùng mới mẻ em đã không thể kiểm soát được dữ liệu lưu vào dẫn đến mất kiểm soát dữ liệu. Đó là sai lầm lớn nhất mà em đã mắc phải trong suốt khoảng thời gian làm đồ án. Nhưng cũng không phải vì vậy mà em nản long. Nhờ các sai lầm liên tiếp như vậy, em đã bắt đầu học được cách thức vận hành của Firebase Database Realtime. Và vấn đề mất kiểm soát dữ liệu cũng đã được giải quyết một cách ổn thỏa.

Sau khi giải quyết xong vấn đề về mất kiểm soát dữ liệu cũng đã hiều được cách thức hoạt động của Firebase Database Realtime em bắt đầu tiến hành quá trình phân tích và thiết kế cơ sở dữ liệu.

1. **Đăng ký thành viên.**

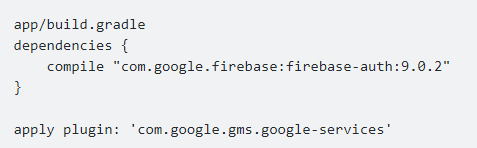
* Bước 1 Thêm chức năng xác thực vào project ( Authentication ).

Mở file **build.gradle** và thêm Firebase dependency.



*Hình 2. 2 – Thêm Authentication vào project Android Studio.*

Mở file **app/build.gradle** và thêm **firebase auth dependency** và ở cuối thêm plugin **‘com.google.gms.google-services’.**



*Hình 2. 3 – Thêm Authentication vào project Android Studio.*

Mở file **AndroidManifest.xml** và sau đó thêm quyền truy cập Internet.



*Hình 2. 4 – Thêm Authentication vào project Android Studio.*

* Bước 2 Thêm user vừa đăng ký vào Database.

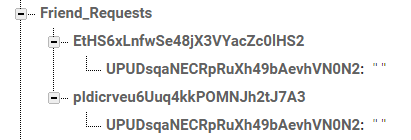


*Hình 2. 5 – Lưu tất cả user vào một node và đặt tên là Users.*

Đây là node dùng để chứa tất cả các dữ liệu của user đã đăng ký.

1. **Kết bạn.**

* Bước 1 Gửi lời mời kết bạn



*Hình 2. 6 – Danh sách lời mời kết bạn.*

Khác với các cơ sở dữ liệu mà chúng ta đã từng tiếp xúc. Realtime Database là cơ sở dữ liệu theo dạng cây (tree). Ở đây ta sẽ hiểu như sau: Khi thực hiện hành động gửi lời mời kết bạn, Realtime Database sẽ tạo ra 1 node đặt tên là Friend\_Resquests. Node này sẽ lưu tất cả các lời mời kết bạn mà các user đã gửi cho nhau. Ví dụ ở đây chúng ta có tất cả 2 lời mời kết bạn.

Ở node thứ nhất, user có id là [***UPUDsqaNECRpRuXh49bAevhVN0N2***](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/Friend_Requests/EtHS6xLnfwSe48jX3VYacZc0lHS2/UPUDsqaNECRpRuXh49bAevhVN0N2)đã gửi lời mời kết bạn cho user có id là [**EtHS6xLnfwSe48jX3VYacZc0lHS2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/Friend_Requests/EtHS6xLnfwSe48jX3VYacZc0lHS2) ( ở đây chúng ta dùng id của user để thuận tiện trong quá trình truy xuất dữ liệu ).

Tương tự, user có id là[**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/Friend_Requests/EtHS6xLnfwSe48jX3VYacZc0lHS2/UPUDsqaNECRpRuXh49bAevhVN0N2)đã gửi lời mời kết bạn cho user có id là [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/Friend_Requests/pIdicrveu6Uuq4kkPOMNJh2tJ7A3).

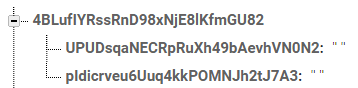
* Bước 2 Thêm bạn.



*Hình 2. 7 – Danh sách bạn bè.*

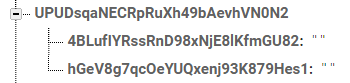
Cũng tương tự như gửi lời mời kết bạn, nếu người đó chấp nhận yêu cầu kết bạn thì Database sẽ tạo ra 1 node gọi là List\_Friends. Node này sẽ chịu trách nhiệm lưu toàn bộ dữ liệu danh sách bạn bè của tất cả các user. Lấy ví dụ trên hình 23, ở đây chúng ta có tất cả 5 danh sách bạn bè.

Ở node thứ nhất, chúng ta hiểu như sau: Đây là danh sách bạn bè của user có id là [**4BLufIYRssRnD98xNjE8lKfmGU82**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82) và user này có 2 người bạn có id lần lượt là [**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/UPUDsqaNECRpRuXh49bAevhVN0N2) và [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3).



*Hình 2. 8 – Danh sách bạn bè của user 4BLufIYRssRnD98xNjE8lKfmGU82.*

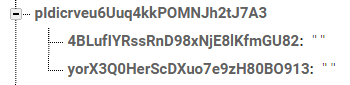
Và vì danh sách bạn bè của mỗi user là hoàn toàn độc lập với nhau nên chúng ta bắt buộc phải tạo ra thêm 2 danh sách bạn bè nữa. Danh sách của user có id là [**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/UPUDsqaNECRpRuXh49bAevhVN0N2) và [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3).



*Hình 2. 9 – Danh sách bạn bè của user UPUDsqaNECRpRuXh49bAevhVN0N2.*

Như các bạn đã thấy bạn bè của user [**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/UPUDsqaNECRpRuXh49bAevhVN0N2) có chứa 1 người bạn là user có id là [**4BLufIYRssRnD98xNjE8lKfmGU82**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82) ở node thứ nhất. Nhưng khác ở chỗ user này còn có thêm 1 người bạn nữa. Đó là user có id là [**hGeV8g7qcOeYUQxenj93K879Hes1**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/UPUDsqaNECRpRuXh49bAevhVN0N2/hGeV8g7qcOeYUQxenj93K879Hes1). Và cũng tương tự thế Database sẽ tạo một danh sách bạn bè nữa của user có id là [**hGeV8g7qcOeYUQxenj93K879Hes1**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/UPUDsqaNECRpRuXh49bAevhVN0N2/hGeV8g7qcOeYUQxenj93K879Hes1).

Quay trở lại node thứ nhất, chúng ta còn thiếu danh sách bạn bè của user có id là [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3).



*Hình 2. 10 – Danh sách bạn bè của user pIdicrveu6Uuq4kkPOMNJh2tJ7A3*

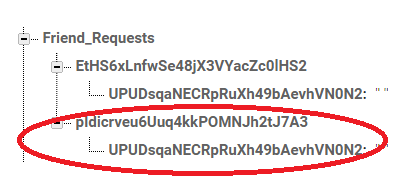
Và cũng có thể dễ dàng thấy được ngoài 1 người bạn là user có id là [**4BLufIYRssRnD98xNjE8lKfmGU82**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82), user còn có thêm 1 người bạn nữa đó là [**yorX3Q0HerScDXuo7e9zH80BO913**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/yorX3Q0HerScDXuo7e9zH80BO913).

Còn một điều nữa, đó là khi đã chấp nhận lời mời kết bạn rồi chúng ta bắt buộc phải xóa lời mời kết bạn đi để đảm bảo được tính logic của một ứng dụng nhắn tin.

Như vậy là chúng ta đã hoàn thành quá trình thêm bạn giữa các user với nhau và vẫn đảm bảo được tính độc lập danh sách bạn bè của mỗi user.

1. **Giới thiệu bạn bè cho nhau.**

Quay trở lại ví dụ đầu tiên. Chúng ta có danh sách bạn bè như thế này. Nhưng hãy để ý đến node thứ hai vì bây giờ chúng ta sẽ bắt đầu thực hiện giới thiệu bạn bè cho 2 user có id là [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3) và user có id là [**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/UPUDsqaNECRpRuXh49bAevhVN0N2)**.**



*Hình 2. 11 – Danh sách lời mời kết bạn.*

Và chúng ta cũng có 2 danh sách bạn bè của 2 user như sau:

* User có id [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3)**.**



*Hình 2. 12 – Danh sách bạn bè của user pIdicrveu6Uuq4kkPOMNJh2tJ7A3.*

* User có id [**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/UPUDsqaNECRpRuXh49bAevhVN0N2)**.**



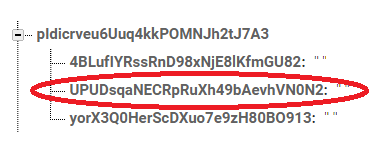
*Hình 2. 13 – Danh sách bạn bè của user UPUDsqaNECRpRuXh49bAevhVN0N2.*

Sau khi quan sát hai *hình 29* và *hình 30* chúng ta có 4 trường hợp xảy ra.

* **Trường hợp 1**: “Không giới thiệu mình cho chính mình”

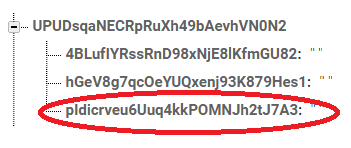
Thoạt đầu nghe có vẻ khó hiểu thật nhưng hãy thử tưởng tượng xem khi 2 user trên trở thành bạn bè thì sẽ như thế nào ? Chính xác là danh sách bạn bè của mỗi bên sẽ được cập nhật lần lượt như sau:

* User có id [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3)**.**



*Hình 2. 14 – Danh sách bạn bè của user pIdicrveu6Uuq4kkPOMNJh2tJ7A3 đã cập nhật.*

User có id [**UPUDsqaNECRpRuXh49bAevhVN0N2**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/UPUDsqaNECRpRuXh49bAevhVN0N2)**.**



*Hình 2. 15 – Danh sách bạn bè của user UPUDsqaNECRpRuXh49bAevhVN0N2 đã cập nhật.*

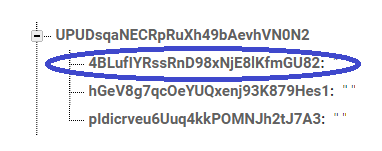
Và điều này đã chứng tỏ rằng 2 user này đã trở thành bạn bè của nhau, nhưng đó không phải là mục đích chính. Giới thiệu bạn bè cho nhau, nếu như hiểu một cách đơn giản hơn đó là ta sẽ lấy dach sách bạn bè của người này để giới thiệu cho người kia và ngược lại. Nhưng quan sát kỹ *hình 31* va *hình 32* user id của bên bị trùng với nhau. Điều đó có nghĩa là khi lấy danh sách bạn bè của user này để giới thiệu cho user kia thì đồng thời sẽ “giới thiệu luôn cả mình cho chính mình”. Việc này là hoàn toàn bất hợp lý nên chúng ta sẽ loại trừ trường hợp này.

* **Trường hợp 2**: “Không giới thiệu bạn chung cho nhau”.

Đến đây thì có lẽ mọi người cũng đã mượn tượng ra được điều gì đó r phải không ? Để làm rõ điều đó chúng ta sẽ thông qua 2 hình sau:



*Hình 2. 16 – Ví dụ bạn bè chung.*

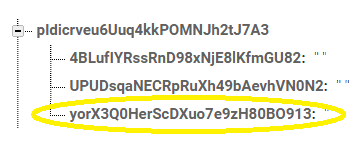


*Hình 2. 17 – Ví dụ bạn bè chung.*

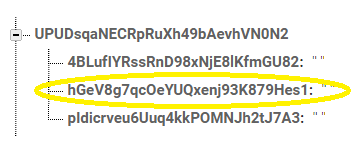
Bây giờ thì mọi thứ đã quá rõ ràng rồi phải không ? Vì 2 user trên đã có một người bạn chung là user có id là [**4BLufIYRssRnD98xNjE8lKfmGU82**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82). Nên chúng ta sẽ không với thiệu user này cho 2 user kia nữa.

* **Trường hợp 3**: “Giới thiệu bạn bè cho nhau ngoại trừ 2 trường hợp trên”.

Để có thể thấy rõ hơn chúng ta sẽ xem 2 hình sau:

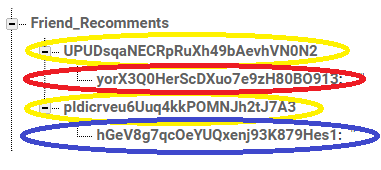


*Hình 2. 18 – Ví dụ bạn bè riêng.*



*Hình 2. 19 – Ví dụ bạn bè riêng.*

Và cuối cùng thì sau khi loại được 2 trường hợp trên, chúng ta sẽ tiến hành giới thiệu bạn bè của 2 user cho nhau. Rất đơn giản, chúng ta sẽ tạo ra 1 node nữa có tên là Friend\_Recomments. Node này sẽ chịu trách nhiệm lưu tất cả các lời gợi ý kết bạn. Sau khi qua quá trình sàn lọc chúng ta sẽ thu được kết quả như thế này:



*Hình 2. 20 – Giới thiệu bạn bè thành công.*

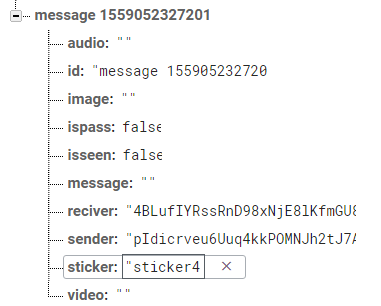
Ở đây chúng ta có thể thấy ràng là đã giới thiệu thành công cho bạn bè của 2 user cho nhau.

* **Trường hợp 4**: “ Nếu bạn bè đã được giới thiệu một lần thì sẽ không giới thiệu lần thứ hai nữa ”

Em cứ ngỡ là sau khi giới thiệu bạn bè thông qua 3 trường hợp trên thì dự án của em sẽ phần nào được hoàn thiện. Nhưng sau khi kiểm tra và thử nghiệm chức năng này nhiều lần thì em đã phát hiện thêm một trường hợp mới, nếu như nói chính xác thì đó là một ngoại lệ mới. Đó là nếu như một người đã được giới thiệu cho người khác thì sẽ không giới thiệu người đó lại lần hai. Lấy ví dụ như: A có 1 người bạn là C, A tiến hành kết bạn với B ( giả sử B không có bạn bè ) thì bạn bè của A sẽ được giới thiệu cho B. Ở đây tức là C sẽ được giới thiệu cho B. Thoạt nhìn thì có vẻ ổn nhưng nếu B kết bạn thêm một người thì đó lại là một câu chuyện hoàn toàn khác. Cũng lấy ví dụ như vừa rồi nhưng lần này chúng ta sẽ cho B kết bạn với D và cũng giả sử như bạn bè của D là C vậy chuyện gì sẽ xảy ra tiếp theo ? Chính xác ! nếu thực thi theo thuật toán cũ thì C sẽ đc giới thiệu cho B một lẫn nữa, như thế danh sách giới thiệu bạn bè của B chỉ có 1 người nhưng lại được giới thiệu tới 2 lần. Điều này là hoàn toàn thiếu logic. Vậy nên chúng ta sẽ thực hiện thêm một bước lọc nữa, bước này sẽ lọc ra danh sách những người đã được giới thiệu cho B và tiến hành kiểm tra đối với danh sách bạn bè mà B đã tiến hành kết bạn. Trong trường hợp này là A và D, nếu không có ai trùng nhau thì không có gì bất ổn. Còn nếu như bị trùng thì sẽ chỉ giới thiệu một lần.

1. **Nhắn tin giữa 2 user với nhau.**

Giai đoạn này chỉ được thực hiện khi 2 user đã trở thành bạn của nhau. Cũng giống như các user 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.



*Hình 2. 21 – Các thuộc tính cơ bản của một tin nhắn.*

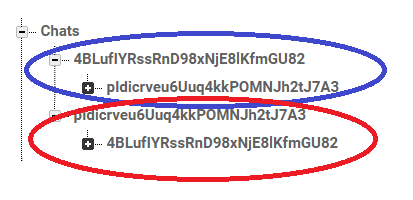
Vấn đề duy nhất cần đặt ra ở đây là nếu ứng dụng của bạn có nhiều người sử dụng và đâu phải họ chỉ nhắn tin với một người, họ có thể sẽ nhắn tin với nhiều người. Và đâu phải họ chỉ gửi một tin, họ sẽ gửi nhiều tin 100, 200 và con số đó thậm chí có thể lên đến 500 là chuyện rất bình thường đối với những cặp đôi đang yêu nhau. Vậy thì làm thế nào để ta có thể quản lý hàng nghìn tin nhắn như thế ?

Vấn đề này khiến em đau đầu thậm chí là mất ngủ mấy đêm liền. Vì do mới chỉ vừa hiểu cách thức hoạt động nên khả năng thiết kế cơ sở dữ liệu của em vẫn chưa thoàn thiện dẫn đến việc e đã lưu TẤT CẢ TIN NHẮN vào trong một node và em gọi nó là node “ Chats ”. Thoạt nhìn thì có vẻ là đúng nhưng khi số lượng tin nhắn ngày càng tăng cùng với số lượng người dùng nhắn tin ngày càng nhiều, em đã bị mất kiểm soát dữ liệu. Lúc này em cảm thấy mình trông thật ngớ ngẩn. Cả một trời tin nhắn thế kia mà lưu vào 1 node thì có mà quản lý bằng mắt. Nhưng cũng nhờ bị đẩy vào đường cùng như thế đã làm kích thích khả năng tư duy logic của em. Thoáng chốc em đã nghỉ 2 hướng giải quyết vấn đề.

Hướng đi thứ nhất, đó là trong node “ Chats “ ban đầu thay vì lưu tất cả tin nhắn của tất cả mọi người vào thì lúc này em sẽ phân chia các tin nhắn theo nhóm người chat với nhau. Lấy ví dụ như: A và B nhắn tin cho nhau thì trong node “ Chats ” ta sẽ tạo ra 2 node nữa 1 node là A, node còn lại là B. trong 2 node này em sẽ lưu tất cả các tin nhắn giữa 2 người nhằm đảm bảo rằng cả 2 người dùng đều có hộp thư thoại riêng của họ => Vấn đề mất kiểm soát nếu số lượng tin nhắn tăng lên đã được giải quyết. Nhưng lại nảy sinh ra tiếp vấn đề thứ 2. Đó là nếu số lượng người mà tăng lên thì sẽ như thế nào ? Lấy ví dụ như A không chỉ nhắn tin cho mỗi B mà bây giờ A muốn nhắn tin cho C và D thì sẽ ra sao tin nhắn giữa A với B, A với C, A với D sẽ lưu vào một node A nữa => mất kiểm soát dữ liệu. Nhờ vậy mà em đã nảy ra thêm 1 hướng đi mới có thể giải quyết được bài toán nản giải này.

Hướng đi thứ 2, nếu chúng ta lưu tất cả tin nhắn vào một node thì sẽ cực kỳ khó quản lý và nếu số lượng cứ ngày càng tăng thì việc mất kiểm soát là chắc chắn xảy ra. Chưa kể tin nhắn thì sẽ có 2 chiều, lấy ví dụ như có 2 user đang nhắn tin với nhau. Nếu như người đó muốn xóa một tin nhắn nào đó thì khi xóa tin nhắn đó thì chỉ xóa mỗi tin nhắn bên người muốn xóa thôi, còn dữ liệ tin nhắn đó bên người kia thì vẫn còn. Lưu dữ liệu như vậy sẽ đảm bảo được tính độc lập của mỗi hộp thoại. Mỗi một tin nhắn được gửi đi thì khi lưu dữ liệu chúng ta phải lưu tin nhắn đó 2 lần nhưng trong 2 node khác nhau (có thể hiểu node ở đây là hộp thoại nhắn tin).

Sau khi đã thiết kết hoàn tất việt lưu tin nhắn trên Realtime Database, chúng ta sẽ thu được kết quả như thế này.

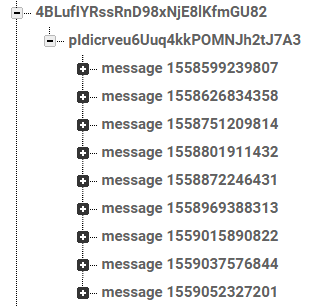


*Hình 2. 22 – Hộp thoại tin nhắn của 2 user.*

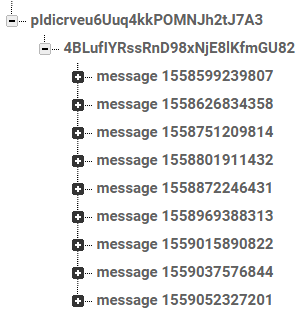
Ví dụ *hình 39,* khi user có id là [**4BLufIYRssRnD98xNjE8lKfmGU82**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82) nhắn một tin nhắn cho user có id là [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3).

Đối với user [**4BLufIYRssRnD98xNjE8lKfmGU82**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82) thì sẽ tạo ra một node có tên là chính id của user đó, bên trong node này sẽ tạo thêm 1 node nữa. Tên của node này chính là id của user mà user kia đã gửi tin nhắn. Trong node này sẽ lưu tất cả các tin nhắn giữa 2 user với nhau.

Đối với user có id là [**pIdicrveu6Uuq4kkPOMNJh2tJ7A3**](https://console.firebase.google.com/project/chat-3d2ec/database/chat-3d2ec/data/List_Friends/4BLufIYRssRnD98xNjE8lKfmGU82/pIdicrveu6Uuq4kkPOMNJh2tJ7A3) cũng tương tự như trên, khi nhận được tin nhắn cũng sẽ ngay lập tạo ra 1 node có tên là id của chính user đó, bên trong sẽ tạo thêm 1 node nữa chính là id của user đã nhắn tin với user này. Và tất nhiên bên trong node này cũng sẽ lưu nội dung tin nhắn y hệt trên kia. Đảm bảo được độc lập và toàn vẹn của tin nhắn.



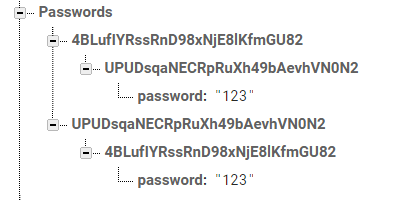
*Hình 2. 23 – Hộp thoại tin nhắn của user 4BLufIYRssRnD98xNjE8lKfmGU82.*



*Hình 2. 24 Hộp thoại tin nhắn của user pIdicrveu6Uuq4kkPOMNJh2tJ7A3.*

1. **Nâng cấp bảo mật.**

Đây chính là ý chính khiến ứng dụng nhắn tin này khác hoàn toàn so với các ứng dụng nhắn tin như: zalo, messenger….



*Hình 2. 25 – Tạo mật khẩu với người mà bạn không muốn cho ai biết.*

Thoạt nhìn thì trông có vẻ rối nhưng nếu để ý kỹ chúng ta có thể dễ dàng đoán ra được 2 user trên đã tạo một mối quan hệ bí mật với nhau thông qua 1 mật mã. Và chỉ những người nhập đúng mật mã này mới có thể xem được tin nhắn giữa 2 user này.

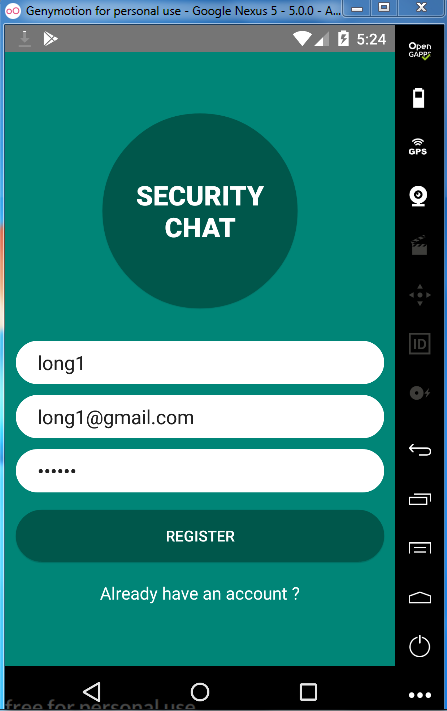
# 

# **CHƯƠNG 3: THỬ NGHIỆM ỨNG DỤNG**

1. **Môi trường cài đặt**

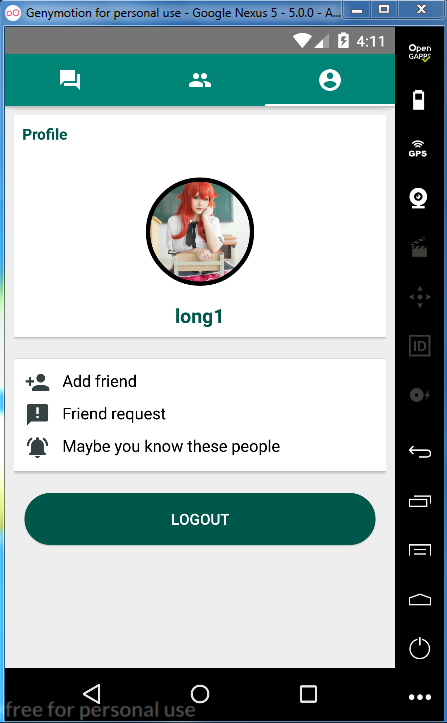
Từ những cơ sở lý thuyết, phương pháp phân tích và thiết kế được đề cập ở chương 2. Em đã xây dựng và phát triển thành công ứng dụng trên môi trường Android Studio 3.3.2, viết bằng ngôn ngữ lập trình Java.

1. **Giao diện đăng ký thành viên**



*Hình 3. 1 – Giao diện đăng ký thông tin cá nhân.*

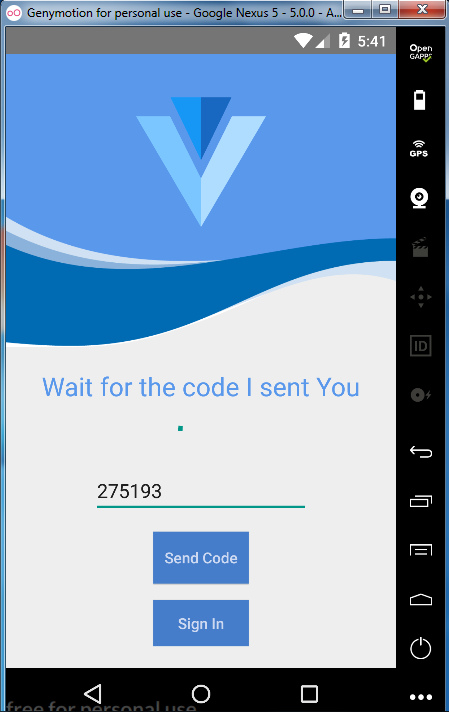
Cũng như bao ứng dụng tin nhắn khác. Đầu tiên chúng ta phải tiến hành đăng ký thành viên trước. Chỉ khi nào đăng ký thành công thì toàn bộ các thông tin cá nhân đã đang ký sẽ được lưu vào Realtime Database. Lúc này thì chúng ta có thể thực hiện trải nghiệm các chức năng vốn có của ứng dụng.



*Hình 3. 2 – Giao diện chính sau khi đăng ký thành công.*

Đây chính là giao diện cá nhân của chúng ta sau khi đăng ký tài khoản thành công. Có cả chức năng cập nhật ảnh đại diện, kết bạn,… và còn nhiều chức năng khác nữa.

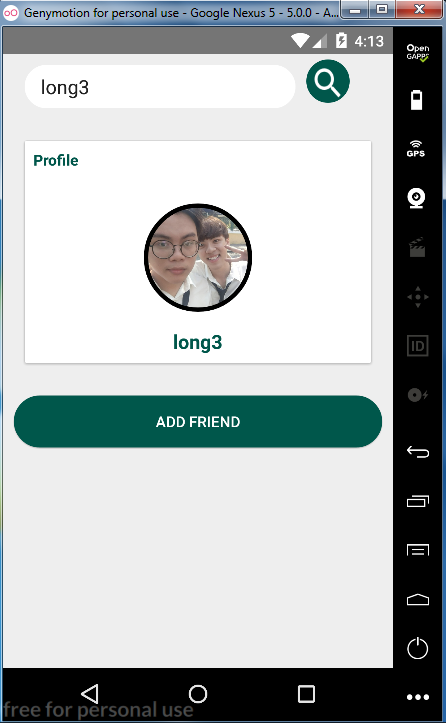
1. **Giao diện đăng nhập tự động**



*Hình 3. 3 – Giao diện gửi mã opt mỗi khi thực hiện phiên đăng nhập tự động.*

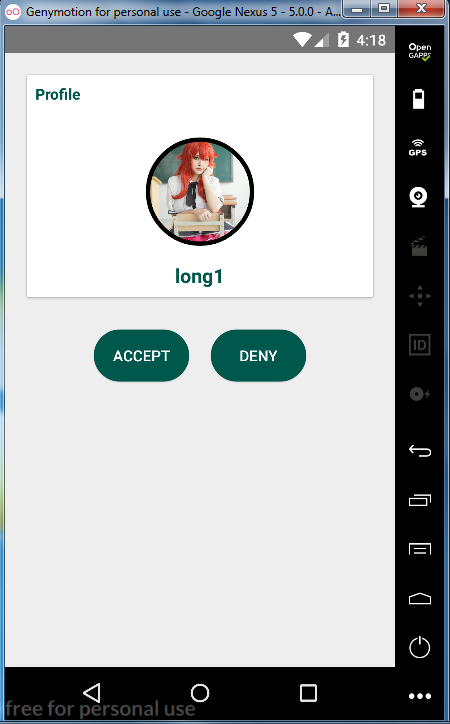
Đây cũng là một bức phát triển lớn trong lĩnh vực bảo mật của ứng dụng, mỗi lần thực hiện đăng nhập tự động server sẽ tiến hành gửi một mã OTP về smartphone của người dùng và chỉ khi đúng mã thì ứng dụng mới cho phép đăng nhập tự động. Về việc gửi tin nhắn từ server đã được Firebase hổ trợ rất mạnh mẽ.

1. **Giao diện thêm bạn**



*Hình 3. 4 – Giao diện tìm bạn bè.*

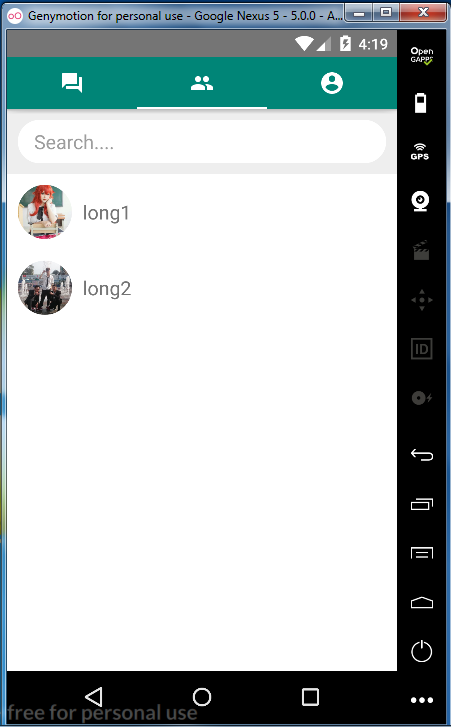
Đây là giao diện cho phép chúng ta có thể tìm bạn thông qua tên của họ, và từ đây chúng ta có thể gửi lời mời kết bạn đến họ và sẽ thật tuyệt nếu như họ chấp nhận lời mời, các bạn có thể tiến hành giao lưu với nhau thông qua những tin nhắn, những đoạn ghi âm hay những clip vui nhộn tất cả đều được hổ trợ tất.



*Hình 3. 5 – Giao diện hiển thị chi tiết người đã gửi lời mời kết bạn.*

Bây giờ chúng ta tiến hành đăng nhập vào tài khoản của user vừa nhận được lời mời kết bạn để xem thử cách hoạt động của nó như thế nào. Như trên thì chúng ta có thể dễ dàng thấy được thông tin cá nhân của user vừa gửi lời mời kết bạn, bên dưới đó là 2 nút “Accept ” và “ Deny ”. Để cho người dùng có cơ hội lựa chọn có nên kết bạn hay là không.

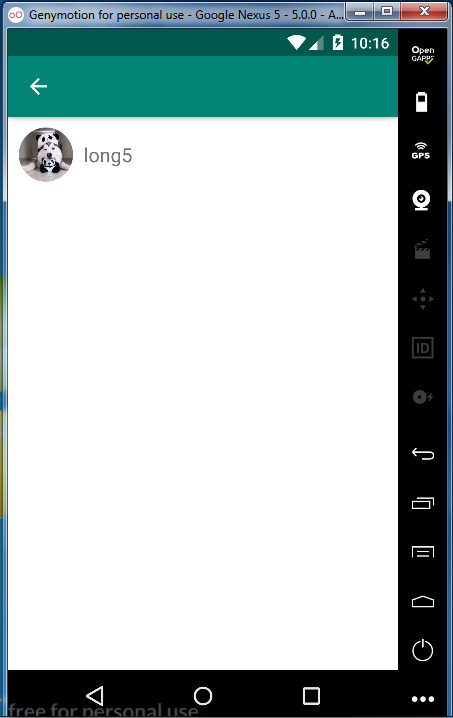
1. **Giao diện danh sách bạn bè khi đã kết bạn thành công**



*Hình 3. 6 – Giao diện danh sách bạn bè.*

Như đã thấy, sau khi bấm vào nút “Accept” thì 2 người dùng trên đã chính thức trở thành bạn bè của nhau, danh sách bạn bè trên đã thêm được bạn mới ( long1 ).

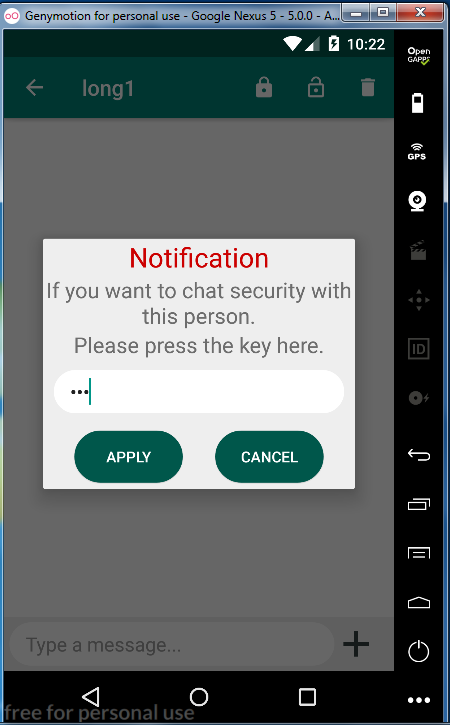
1. **Giao diện giới thiệu bạn bè**



*Hình 3. 7 – Bạn bè được giới thiệu cho nhau.*

Đây là một trong những tính năng nổi bật nhất của ứng dụng này, sau khi kết bạn thành công hệ thống sẽ tự động giới thiệu bạn bè cho lẫn nhau. Điều này không chỉ giúp cho các người dùng có thể quen biết nhau nhiều hơn mà nó càng thúc đẩy quá trình mở rộng quy mô của ứng dụng nữa.

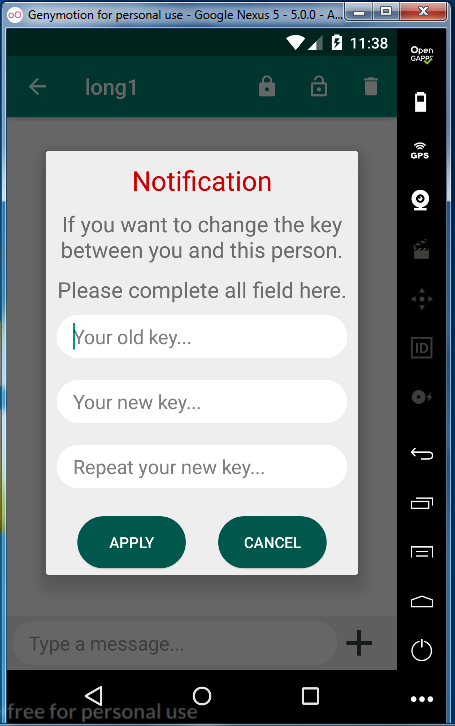
1. **Giao diện thiết lập mật mã giữa 2 user**



*Hình 3. 8 – Giao diện thiết lập mật mã.*

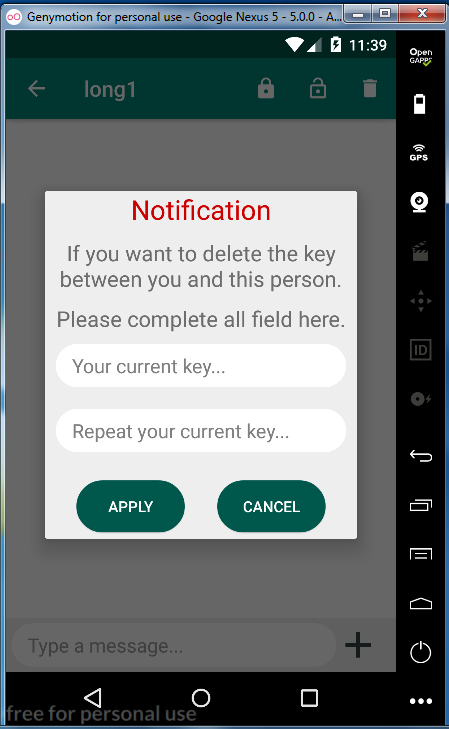
Đây chính là tính năng độc quyền mà chỉ ứng dụng này mới có, tính năng thiết lập mật mã bảo mật tin nhắn. Tính năng này cho phép 2 người dùng đang nhắn tin với nhau có thể thiết lập mật mã đã được thống nhất với nhau từ trước. Và chỉ khi nhập đúng mật mã thì cả 2 người dùng mới có thể xem tin nhắn, nhắn tin,….

1. **Giao diện nến người dùng muốn thay đổi khóa**



*Hình 3. 9 – Giao diện nếu người dùng muốn thay đổi mật mã.*

1. **Giao diện nếu người dùng muốn xóa mật mã**



*Hình 3. 10 – Giao diện nếu người dùng muốn xóa mật mã.*

1. **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 như: Messenger, mocha, zalo,…..

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 như hình ảnh, âm thanh, video,…. Được lưu trực tiếp trên server, tiện lợi trong việc load dữ liệu. Cải thiện được thời gian truy xuất dữ liệu.

Chế độ xác thực người dùng khiến cho độ bảo mật của ứng dụng được nâng lên một bậc.

### 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 sticker 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. Không chỉ vậy ứng dụng còn cho phép gửi cả hình ảnh, âm thanh thậm chí là cả video.

Đặc biệt là khả năng thiết lập mật mã bảo mật đã được em đầu tư rất nhiều thời gian và công sức. Đảm bảo rằng chức năng này hoạt động tốt nhất có thể. Đem lại lợi ích không nhỏ cho người dùng về vấn đề an toàn và bảo mật thông tin.

# **KẾT LUẬN VÀ KHUYẾN NGHỊ**

## Kết quả đạt được

* Về tri thức:

Trong suốt khoảng thời gian 4 tháng làm đồ án tốt nghiệp, 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:

Cuối cùng là tính năng đặc biệt của ứng dụng. Tính năng bảo mật thông qua mật mã đã được thống nhất từ trước giữa 2 người. Khiến cho mức độ bảo mật của ứng dụng đã tốt nay lại càng tốt hơn.

Đồng thời ứng dụng cũng đã giải quyết được vấn đề mà messenger hay zalo không làm được. Đó chính là vấn đề nếu có 2 tài khoản được lưu trên cùng 1 thiết bị thì thiết bị đó hoàn toàn có thể xem dữ liệu tin nhắn của cả 2 tài khoản như vậy là không đảm bảo yêu cầu về bảo mật. Nhất là trong xã hội ngày nay, dữ liệu chính là tài nguyên vô cùng quan trọng.

* Về con người:

Thật sự thì khoảng thời gian là khoảng thời gian em gặp rất nhiều khó khăn và mệt mỏi. Từ những khó khăn về hoàn cảnh gia đình cho đến những khó khăn trong lúc tìm, đọc tài liệu. 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 đồ án tốt nghiệp 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.

## 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: thông báo tin nhắn trực quan, hay trong lúc nhắn tin thay vì hộp thoại của mỗi người dùng có thể di chuyển lên xuống tùy thuộc vào tin nhắn đã tới hay đã gửi thì em lại không làm được. Đây là một trong những sai sót lớn nhất của em.

## 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,…

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 mà người người, nhà nhà ai cũng biết đến, ai cũng tin dùng.

# **TÀI LIỆU THAM KHẢO**

[1]. [Robin Wieruch](https://www.google.com/search?q=the+road+to+react+with+firebase:+your+journey+to+master+advanced+react+for+business+web+applications+robin+wieruch&stick=H4sIAAAAAAAAAC2NsQ7CMAxEJyQWGJgZLEaWUKli6M8gJ3FJaImL4xD6O3wpRbCcdNK7d-vtbmOupmnC-TT6cZgPv9q30-vRJj3uB9cZyzyYKlGV0qWyDB0WDSzvlWggEEYPyiCETqFGDdBHIYuZOpi5CNyWSDR_oTtmJQH0T0yO_H_Us4AtOSbKGSpZwGkao0ONnPJyYGNaxCTFhQ-jBR2eswAAAA&sa=X&ved=2ahUKEwi0zs6c8-XiAhWZA4gKHd1wCEEQmxMoATAPegQIDRAH), *The Road to React with Firebase*, Independently, 2019

[2]. [Neil Smyth](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Neil+Smyth%22), *Firebase Essentials - Android Edition*, Payload Media, 2017

[3]. [Bill Stonehem](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Bill+Stonehem%22), *Google Android Firebase: Learning the Basics*, First Rank, 2016

[4]. [Reto Meier](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Reto+Meier%22), [Ian Lake](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Ian+Lake%22), *Professional Androird*, John Wiley & Sons, 2018

[5]. [Erik Hellman](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Erik+Hellman%22), *Android Programming: Pushing the limits* , John Wiley & Sons, 2013

[6]. [Greg Nudelman](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Greg+Nudelman%22), *Android Design Patterns: Interaction Design Solutions for Developers*, John Wiley & Sons, 2013

[7]. [Allen B. Downey](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Allen+B.+Downey%22), [Chris Mayfield](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Chris+Mayfield%22), *Hink Java: How to think like a computer scientist*, O'Reilly Media, 2016

[8]. [Tim Peierls](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Tim+Peierls%22), [Brian Goetz](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Brian+Goetz%22), [Joshua Bloch](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Joshua+Bloch%22), [Joseph Bowbeer](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Joseph+Bowbeer%22), [Doug Lea](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Doug+Lea%22), [David Holmes](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22David+Holmes%22), *Java Concurrency In Practice*, Pearson Education, 2006

[9]. [Cay S. Horstmann](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Cay+S.+Horstmann%22), *Java SE 8 for the Really Impatient*, ddison-Wesley Professional, 2014

[10]. *Chat application with Firebase,*

<https://www.youtube.com/watch?v=fJWFeW09qeE&list=PLzLFqCABnRQftQQETzoVMuteXzNiXmnj8&index=1> truy cập vào 9:32 23/4/2019

[11]. *Upload image on Firebase*,

<https://stackoverflow.com/questions/48377935/how-to-upload-image-on-firebase-without-button-android-studio> truy cập vào 14:32 25/4/2019

[12]. *Upload video on firebase*,

<https://stackoverflow.com/questions/44746549/how-to-upload-video-to-firebase-on-android> truy cập vào 17:55 25/4/2019

[13]. *Học lập trình android Khoa Phạm*, <https://www.youtube.com/watch?v=CKsRo6ksvio>, truy cập vào 19:53 14/5/2019