

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO CUỐI KÌ MÔN HỌC  
PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG  
ỨNG DỤNG GHI CHÚ**

*Người hướng dẫn:* TS LÊ VĂN VANG

*Người thực hiện:* TRẦN HOÀI BẢO – 52000634

PHAN ANH KHOA – 32001031

Lớp : 20050201

Khoa : 24

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO CUỐI KÌ MÔN HỌC  
PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG  
ỨNG DỤNG GHI CHÚ**

Người hướng dẫn: **TS LÊ VĂN VANG**

Người thực hiện: **TRẦN HOÀI BẢO**

**PHAN ANH KHOA**

Lớp : **20050201**

Khoá : **24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## LỜI CẢM ƠN

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép nhóm chúng em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ nhóm chúng em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, nhóm chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của TS Lê Văn Vang

Với lòng biết ơn sâu sắc nhất, nhóm chúng em xin gửi lời cảm ơn đến TS Lê Văn Vang cùng tất cả thầy cô ở Khoa Công Nghệ Thông Tin đã truyền đạt vốn kiến thức quý báu cho nhóm chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của nhóm chúng em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn thầy Vang – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn nhóm chúng em hoàn thành tốt bài báo cáo này trong thời gian qua.

Nhóm chúng em xin chân thành cảm ơn!

## ĐỒ ÁN ĐƯỢC HOÀN THÀNH

### TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Nhóm chúng em xin cam đoan đây là sản phẩm đồ án của riêng nhóm chúng em và được sự hướng dẫn của TS Lê Văn Vang. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào nhóm chúng em xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do nhóm chúng em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 01 tháng 12 năm 2023

Tác giả

Trần Hoài Bảo

Phan Anh Khoa

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm

(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm

(kí và ghi họ tên)

## TÓM TẮT

Trong cuộc sống hiện nay, một ứng dụng ghi chép những công việc hàng ngày là rất cần thiết. NotNotes là một trong những ứng dụng ghi chép chúng ta có thể sử dụng. NotNotes là một ứng dụng ghi chép giúp người dùng có thể dễ dàng lên kế hoạch, ghi chú, lập thời gian biểu. Tất cả những dữ liệu trên sẽ được lưu trữ vào cùng một không gian để chúng ta có thể dễ dàng tìm kiếm lại khi cần.

Đồ án này tập trung vào việc phát triển một ứng dụng điện thoại di động trên Android quản lý thông tin về các ghi chú. Ứng dụng sẽ cung cấp cho người dùng một nền tảng trực tuyến để tạo ghi chú và nhắc nhở một cách thuận tiện và nhanh chóng.

Ứng dụng được xây dựng trên nền tảng Android sử dụng ngôn ngữ Kotlin, sử dụng cơ sở dữ liệu Firebase để lưu trữ thông tin thời gian thực về các ghi chú, tiến bộ công việc, nhắc nhở khi đến hạn. Ứng dụng có giao diện thân thiện, màu sắc chủ đạo là màu xanh biển làm dịu mắt người dùng, cho phép người dùng tìm kiếm và tạo ứng dụng ghi chú một cách dễ dàng.

Người dùng sẽ có thể đăng ký và đăng nhập để tạo ra một tài khoản cá nhân, lưu trữ thông tin về các ghi chú đã tạo và quản lý thông tin cá nhân. Ứng dụng cũng sẽ có tính năng sửa, xóa các ghi chú không cần thiết.

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<i>i</i>
<b>ĐỒ ÁN ĐƯỢC HOÀN THÀNH .....</b>	<i>ii</i>
<b>TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG .....</b>	<i>ii</i>
<b>PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....</b>	<i>iii</i>
<b>TÓM TẮT .....</b>	<i>iv</i>
<b>CHƯƠNG 1 – MÔ TẢ ĐỀ TÀI .....</b>	<b>6</b>
1.1    Lý do chọn đề tài .....	6
1.2    Phương pháp nghiên cứu .....	6
1.2.1 Tổng quan về đề tài .....	6
1.2.2 Tổng quan các công nghệ sử dụng .....	6
1.2.3 Tổng quan các chức năng của ứng dụng .....	9
1.3    Phạm vi đối tượng của đề tài .....	10
<b>CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT .....</b>	<b>11</b>
2.1    Giới thiệu về Android .....	11
2.2    Xây dựng giao diện ứng dụng .....	11
2.2.1 View và Viewgroup .....	11
2.2.2 XML Layout .....	12
2.2.3 Các widget cơ bản .....	13
2.3    Ngôn ngữ lập trình Kotlin .....	17
2.3.1 Kotlin là gì? .....	17
2.3.2 Ưu điểm nổi bật của ngôn ngữ lập trình Kotlin .....	18
2.3.3 Hạn chế của ngôn ngữ lập trình Kotlin .....	20
2.4    Firebase .....	20
2.4.1 Firebase là gì? .....	20
2.4.2 Tổng quan về các nhóm sản phẩm/dịch vụ của Firebase .....	22
2.4.3 Các tính năng chính của Firebase .....	26
2.4.4 Ưu điểm của Firebase .....	28
2.4.5 Nhược điểm của Firebase .....	29
<b>CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....</b>	<b>31</b>
3.1    Đặt vấn đề.....	31
3.2    Đặc tả hệ thống .....	31
3.2.1 Chức năng đăng nhập.....	31

3.2.2 Chức năng đăng ký .....	31
3.2.3 Chức năng đổi mật khẩu .....	31
3.2.4 Chức năng quên mật khẩu .....	31
3.2.5 Chức năng cập nhật thông tin .....	32
3.2.6 Chức năng nhắc nhở .....	32
3.2.7 Chức năng thay đổi ngôn ngữ .....	32
3.2.8 Chức năng về ghi chú .....	32
<b>3.3 Phân tích và thiết kế.....</b>	<b>32</b>
3.3.1 Mô hình usecase tổng quát .....	32
3.3.2 Đặc tả .....	33
3.3.3 ERD .....	34
3.3.4 Mô hình quan hệ .....	34
3.3.5 Luồng hoạt động .....	35
<b>CHƯƠNG 4 – CƠ SỞ THỰC TIỄN .....</b>	<b>43</b>
4.1 Bản thiết kế.....	43
4.2 Đăng ký người dùng .....	44
4.3 Đăng nhập người dùng .....	45
4.4 Màn hình chính .....	46
4.5 Màn hình thêm ghi chú .....	47
4.6 Màn hình sửa ghi chú .....	49
4.7 Màn hình xóa ghi chú .....	51
4.8 Màn hình cập nhật thông tin .....	52
4.9 Màn hình đổi mật khẩu .....	54
4.10 Màn hình quên mật khẩu .....	56

## **DANH MỤC CÁC BẢNG BIẾU, HÌNH VẼ, ĐỒ THỊ**

### **DANH MỤC HÌNH**

Hình 1. 1 - Hình mô tả file XML .....	7
Hình 1. 2 - Hình mô tả ngôn ngữ lập trình Kotlin .....	7
Hình 1. 3 - Hình mô tả Firebase.....	8
Hình 1. 4 - Hình mô tả Android Studio .....	9
Hình 2. 1 - Hình minh họa các đối tượng ViewGroup chứa đối tượng View.....	12
Hình 2. 2 - Hình mô tả hình dáng của View .....	12
Hình 2. 3 - Widget TextView .....	14
Hình 2. 4 - Widget Button .....	14
Hình 2. 5 - Widget ImageButton .....	14
Hình 2. 6 - Widget EditText .....	15
Hình 2. 7 - Widget Checkbox và RadioButton .....	16
Hình 2. 8 - Widget RecyclerView/ListView .....	16
Hình 2. 9 - Widget Switch .....	16
Hình 2. 10 - Hình so sánh Kotlin và Java .....	17
Hình 2. 11 - Kotlin vs Java .....	18
Hình 2. 12 - Hình dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây - Firebase .	21
Hình 2. 13 - Hình mô tả Firebase Cloud Firestore.....	22
Hình 2. 14 - Hình mô tả Firebase Authentication.....	22
Hình 2. 15 - Hình mô tả Firebase Extensions .....	23
Hình 2. 16 - Hình mô tả Firebase Crashlytics.....	23
Hình 2. 17 - Hình mô tả Firebase Performance .....	24
Hình 2. 18 - Hình mô tả Firebase Analytics .....	24
Hình 2. 19 - Hình mô tả Firebase Remote Config .....	25

Hình 2. 20 - Hình mô tả Firebase Cloud Messaging .....	25
Hình 2. 21 - Hình mô tả Firebase A/B Testing .....	26
Hình 2. 22 - Hình mô tả Firebase Realtime Database .....	26
Hình 2. 23 - Hình mô tả chi tiết Firebase Authentication.....	27
Hình 2. 24 - Hình mô tả Firebase Hosting.....	27
Hình 2. 25 - Hình minh họa ưu điểm Firebase .....	29
Hình 2. 26 - Hình minh họa nhược điểm Firebase.....	30
Hình 3. 1 - Hình mô tả usecase tổng quát.....	33
Hình 3. 2 - Hình sơ đồ ERD .....	34
Hình 3. 3 - Hình mô hình quan hệ .....	34
Hình 3. 4 - Luồng hoạt động màn đăng nhập .....	35
Hình 3. 5 - Luồng hoạt động màn đăng kí .....	36
Hình 3. 6 - Luồng hoạt động màn đổi mật khẩu .....	37
Hình 3. 7 - Luồng hoạt động màn hình quên mật khẩu .....	38
Hình 3. 8 - Luồng hoạt động màn hình cập nhật thông tin cá nhân.....	39
Hình 3. 9 - Luồng hoạt động màn hình tạo ghi chú mới.....	40
Hình 3. 10 - Luồng hoạt động màn hình cập nhật thông tin ghi chú đã tạo .....	41
Hình 3. 11 - Luồng hoạt động màn hình xóa ghi chú .....	42
Hình 4. 1 - Hình thiết kế giao diện bằng figma .....	43
Hình 4. 2 - Màn hình đăng kí thực tế của ứng dụng .....	44
Hình 4. 3 Hình thể hiện Firebase Authentication sau khi đăng kí .....	44
Hình 4. 4 – Hình thể hiện realtime database sau khi đăng kí .....	45
Hình 4. 5 - Màn hình đăng nhập thực tế của ứng dụng .....	45
Hình 4. 6 - Hình thể hiện màn hình chính của ứng dụng.....	46

Hình 4. 7 - Màn hình thêm ghi chú thực tế của ứng dụng .....	47
Hình 4. 8 - Firebase Readtime Database sau khi thêm note mới.....	48
Hình 4. 9 - Màn hình sửa ghi chú thực tế của ứng dụng.....	49
Hình 4. 10 - Firebase Realtime Database sau khi chỉnh sửa thông tin note đã được tạo .....	50
Hình 4. 11 - Màn hình xóa ghi chú thực tế của ứng dụng .....	51
Hình 4. 12 - Firebase Database Realtime sau khi xóa ghi chú .....	51
Hình 4. 13 - Màn hình thông tin và chỉnh sửa thông tin thực tế của ứng dụng .....	52
Hình 4. 14 - Màn hình thông tin cá nhân sau khi cập nhật thông tin.....	53
Hình 4. 15 - Firebase Realtime Database sau khi cập nhật thông tin người dùng.....	53
Hình 4. 16 - Màn hình đổi mật khẩu thực tế của ứng dụng .....	54
Hình 4. 17 - Thủ đăng nhập sau khi đổi mật khẩu.....	55
Hình 4. 18 - Màn hình quên mật khẩu thực tế của ứng dụng .....	56
Hình 4. 19 - Tạo mật khẩu mới cho tài khoản sau khi xác thực .....	57
Hình 4. 20 - Thủ đăng nhập sau khi reset mật khẩu .....	58

# CHƯƠNG 1 – MÔ TẢ ĐỀ TÀI

## 1.1 Lý do chọn đề tài

Đề cung cấp cho người dùng một ứng dụng tạo ghi chú trực tuyến thuận tiện và dễ dàng hơn. Thay vì tạo ghi chú một cách thủ công bằng cách viết ra trên giấy thì tạo ghi chú trên ứng dụng sẽ thuận tiện hơn vì điện thoại thì chúng ta hầu như luôn mang theo bên mình còn giấy và viết thì không phải lúc nào cũng có sẵn. Điều này giúp tiết kiệm thời gian và công sức cho người dùng và đồng thời giúp người dùng quản lý các ghi chú một cách hiệu quả hơn.

Ngoài ra, ứng dụng còn cung cấp cho người dùng nhiều tính năng nổi bật như chia sẻ tài nguyên với các ứng dụng khác, tạo lịch hẹn nhắc với google calendar hay tải ảnh lên từ thư viện google drive.

## 1.2 Phương pháp nghiên cứu

### 1.2.1 Tổng quan về đề tài

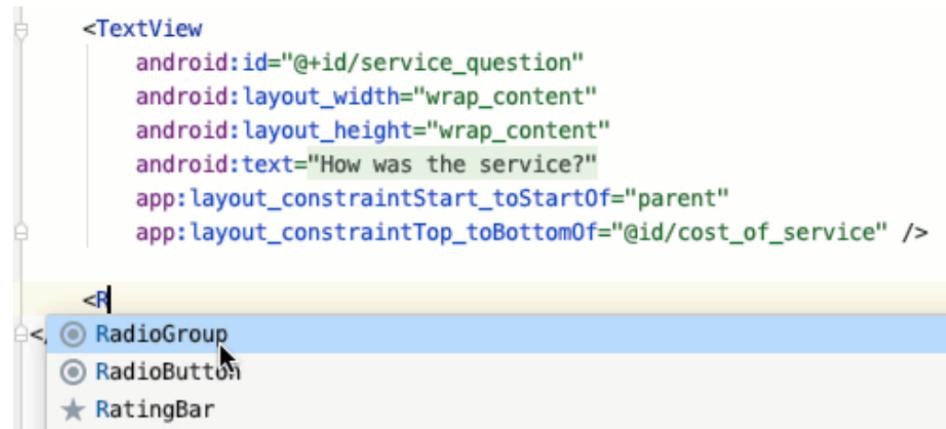
- Thiết kế giao diện người dùng (UI/UX): Khi thiết kế ứng dụng di động, chúng ta sẽ phải quan tâm đến việc tạo ra một giao diện thân thiện, dễ sử dụng và hấp dẫn người dùng, cần tạo ra nhiều giao diện để phù hợp với kích thước của nhiều loại điện thoại khác nhau. Đây là một lý thuyết quan trọng trong việc phát triển sản phẩm công nghệ.

- Quản lý cơ sở dữ liệu: Khi thiết kế ứng dụng ghi chú, chúng ta sẽ phải lưu trữ và quản lý thông tin các ghi chú, thông tin cá nhân của người dùng, cần phải thiết kế cơ sở dữ liệu gọn nhẹ để điện thoại có thể chạy mượt mà vì dung lượng điện thoại khá nhỏ.

- Thiết kế ứng dụng thời gian thực: Khi thiết kế ứng dụng di động, chúng ta sẽ phải tạo ra các ứng dụng lưu trữ dữ liệu thời gian thực, cho phép người dùng tương tác với các tính năng của ứng dụng như tạo ghi chú, cài đặt tiến độ, chỉnh sửa, xóa ghi chú, tạo lịch nhắc đồng bộ với các ứng dụng khác như google calendar.

### 1.2.2 Tổng quan các công nghệ sử dụng

- XML: Android cung cấp từ vựng XML (eXtensible Markup Language) là ngôn ngữ đánh dấu mở rộng cho các lớp ViewGroup và View, vì vậy hầu hết giao diện người dùng của được xác định trong các tệp XML.



Hình 1. 1 - Hình mô tả file XML

- Kotlin: Sử dụng ngôn ngữ lập trình kotlin để xây dựng ứng dụng di động dành cho các thiết bị android trên android studio. Kotlin là ngôn ngữ lập trình được phát triển bởi JetBrains.



Hình 1. 2 - Hình mô tả ngôn ngữ lập trình Kotlin

- Firebase: Đây là hệ quản trị cơ sở dữ liệu được sử dụng để lưu trữ thông tin thời gian thực. Firebase cung cấp các dịch vụ để lưu trữ dữ liệu và xác thực người dùng.



Hình 1. 3 - Hình mô tả Firebase

- Android Studio: Android Studio là Môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android. Nhờ có công cụ cho nhà phát triển và trình soạn thảo mã mạnh mẽ của IntelliJ IDEA, Android Studio cung cấp thêm nhiều tính năng giúp nâng cao năng suất khi xây dựng ứng dụng Android.

# android studio



### Hình 1. 4 - Hình mô tả Android Studio

#### **1.2.3 Tổng quan các chức năng của ứng dụng**

- **Các chức năng chính:**

- Chức năng liên quan tài khoản người dùng (cho phép người dùng đăng ký tài khoản mới, đăng nhập, xác thực email người dùng, quên mật khẩu, đổi mật khẩu,...)
  - Tạo mới ghi chú: Cho phép người dùng thêm các ghi chú mới vào danh sách.
  - Hiển thị danh sách các ghi chú đã được thêm vào, bao gồm cả tiêu đề và chi tiết của từng ghi chú, ngày, thời gian tạo, độ ưu tiên, trạng thái,...
  - Trạng thái ghi chú: gồm 3 trạng thái: Ready, In Progress, và Done
  - Xem chi tiết ghi chú: Khi người dùng chọn một công việc trong danh sách, ứng dụng hiển thị thông tin chi tiết về công việc đó, bao gồm tiêu đề, mô tả, ngày deadline, độ ưu tiên,...
  - Xóa công việc tạm thời ( thùng rác ): Khi người dùng xóa công việc thì sẽ cấm cờ cho công việc đó và để trong thùng rác, người dùng có thể restore hoặc xóa liền trong thùng rác nếu muốn, nếu không xóa thì sau 30 ngày công việc tự delete khỏi thùng rác
  - Sắp xếp công việc theo các danh mục: Ngày deadline, độ ưu tiên, mức độ quan trọng
  - Cập nhật công việc: Cho phép người dùng chỉnh sửa các thông tin trong công việc
  - Nhắc nhở cho người dùng trước 24h khi sắp đến hạn deadline
  - Nhắc nhở người dùng theo lịch mà người dùng đã cài đặt trong lúc tạo công việc
  - Tìm kiếm công việc theo tiêu đề hay nội dung
  - Đóng bộ hóa dữ liệu trên nhiều thiết bị khi đăng nhập cùng 1 tài khoản
- **Các chức năng nâng cao (sẽ tìm hiểu thêm):**
- Chia sẻ tài nguyên với các ứng dụng sắp xếp thời gian khác ( ví dụ cài đặt lịch cho google calendar)

- Dán nhãn cho từng loại ghi chú để người dùng dễ dàng tìm kiếm
- Đính kèm hình ảnh, âm thanh, file mp3, mp4 và các định dạng đặc biệt
- Đăng nhập bằng gmail hoặc facebook mà không cần tạo tài khoản
- Tính năng tạo công việc nhóm: cho phép người dùng tạo 1 công việc cho 1 nhóm và mọi người cùng chia sẻ tiến độ làm việc.

### **1.3 Phạm vi đối tượng của đề tài**

Phạm vi của ứng dụng là cho phép người dùng tạo các ghi chú nhanh chóng, dễ dàng, thuận tiện hơn. Đồng thời, ứng dụng cũng cho người dùng các tính năng đồng bộ dữ liệu, chia sẻ tài nguyên với các ứng dụng khác.

Đối tượng của ứng dụng là người dùng đang mong muốn có một ứng dụng nhỏ gọn trên điện thoại di động để lưu trữ những thông tin cần thiết, quan trọng một cách thuận tiện và có thể nhắc nhở người dùng khi sắp đến hạn. Ngoài ra, đối tượng cũng bao gồm những người tiết kiệm thời gian và tránh tình trạng muôn ghi chú mà không có sẵn giấy hay viết.

## CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

### 2.1 Giới thiệu về Android

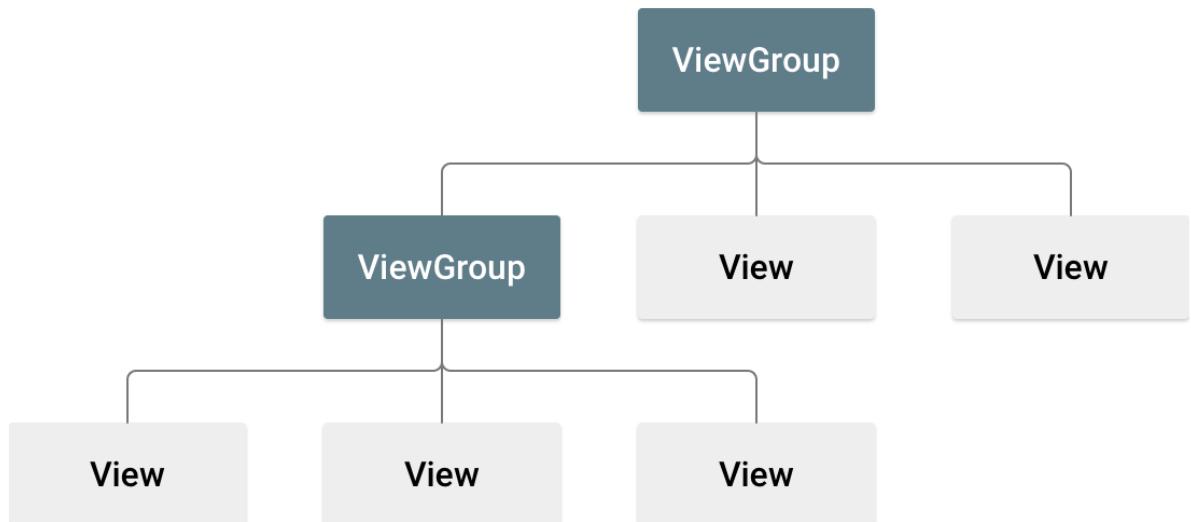
Android là hệ điều hành phát triển bởi Tổng công ty Android với sự hỗ trợ tài chính của Google từ năm 2003 và được Google chính thức mua lại vào năm 2005. Android là hệ điều hành dựa trên nền tảng Linux, chủ yếu dành cho các thiết bị có màn hình cảm ứng như: điện thoại, máy tính bảng.

Hiện nay Android không còn là hệ điều hành duy nhất trên thị trường nữa mà phải cạnh tranh với hệ điều hành iOS đến từ nhà Apple. Tuy nhiên, Android vẫn chiếm thị phần lớn trong ngành điện thoại.

### 2.2 Xây dựng giao diện ứng dụng

#### 2.2.1 View và ViewGroup

Giao diện người dùng (UI) cho ứng dụng Android được xây dựng dưới dạng phân cấp bô cục (layout) và tiện ích (widget). Bô cục là các đối tượng ViewGroup, vùng chứa kiểm soát cách định vị các chế độ xem con trên màn hình. Tiện ích là các đối tượng View, các thành phần giao diện người dùng như nút và hộp văn bản.



Hình 2. 1 - Hình minh họa các đối tượng ViewGroup chứa đối tượng View.

Lớp View là thành phần cơ bản nhất của Android mà từ đó có thể tạo giao diện người dùng. Nó hoạt động như một nơi chứa các phần tử có thể hiển thị. View chiếm một vùng hình chữ nhật trên màn hình và chịu trách nhiệm vẽ và xử lý sự kiện.



Hình 2. 2 - Hình mô tả hình dáng của View

Widget là các lớp con của View. Chúng được sử dụng để tạo các thành phần UI tương tác như nút, hộp kiểm, nhãn, trường văn bản, Layout là các vùng chứa có cấu trúc vô hình được sử dụng để chứa các View khác và các layout lồng nhau.

### **2.2.2 XML Layout**

Có 5 loại Layout cơ bản : FrameLayout, LinearLayout, RelativeLayout, TableLayout và AbsoluteLayout. Ngoài ra còn có ConstraintLayout mới được thêm vào các phiên bản sau và nó là Layout có tính đa dụng và tùy biến cao nhất. Trong đề tài sẽ sử dụng 3 layout chủ yếu là RelativeLayout, LinearLayout và ConstraintLayout.

- FrameLayout: là layout đơn giản nhất, một không gian trống trên screen mà có thể cho bất kì object nào lên đó. Ví dụ, 1 bức ảnh. Tất cả các thành phần con của FrameLayout được đính ở góc trái màn hình screen, có thể ko cần phải chỉ định rõ vị trí cho view con. Các views con kế tiếp sẽ đơn giản được vẽ lên cái cũ, từng phần hoặc tất cả.

- LinearLayout: xếp tất cả các View con thành một hướng ngang hoặc dọc dựa vào android:orientation. Tất cả các views con sẽ được xếp lần lượt kế tiếp: vertical sẽ chỉ có 1 con mỗi dòng, và horizontal cũng sẽ chỉ có mỗi con một cột. Một LinearLayout có margins giữa các con và gravity (trái, phải, center) cho mỗi con. Có thể thêm thuộc tính weight cho các con của LinearLayout. Mỗi Weight sẽ có một giá trị quan trọng đối với View, giá trị này cho phép mở rộng ra để điền lắp vào các phần trống thừa còn lại trong view cha
- TableLayout: cho view con vào trong hàng và cột. Container TableLayout sẽ không hiển thị line border. Một ví dụ về TableLayout hiển thị 2 hàng và 2 ô trong mỗi hàng. Border của các cell là dạng chấm chấm.
- RelativeLayout: Một layout cho phép views con chỉ định vị trí tương đối tới views cha hoặc đối với nhau. Có thể align 2 thành phần bằng right border hoặc below, centered v.v... Các thành phần được render theo thứ tự đã cho, cho nên nếu một thành phần được center trong màn hình, các thành phần khác sẽ align lại chính nó vậy nên những thành phần đó sẽ lại dc align tương đối đến trung tâm màn hình. Cũng bởi vì thứ tự sắp xếp, cho nên nếu sử dụng XML để chỉ định layout, các thành phần sẽ tham chiếu sẽ được list trong XML file trước khi tham chiếu nó từ views khác thông qua ID.
- AbsoluteLayout: Một layout cho phép chỉ định chính xác vị trí (x/y tọa độ) của views con. Absolute layouts thì ít linh hoạt và khó để maintain hơn là các loại layout mà ko có vị trí tuyệt đối khác.
- ConstraintLayout: Một layout tương tự relative layout cho phép views con chỉ định vị trí tới views cha hoặc đối với nhau nhưng khác ở chỗ ConstraintLayout linh hoạt hơn khi có thể ràng buộc 4 hướng (trên, dưới, trái, phải) của một view con với view cha hoặc view con khác.

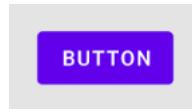
### **2.2.3 Các widget cơ bản**

TextView: thường để hiện thị text, ko thể chỉnh sửa được.



Hình 2. 3 - Widget TextView

Button: cho phép khởi tạo một cái nút bấm trong GUI và người dùng có thể click vào để kích hoạt sự kiện onclick. Button là một subclass của TextView. Do đó format một Buttons thì khá tương tự với TextView.



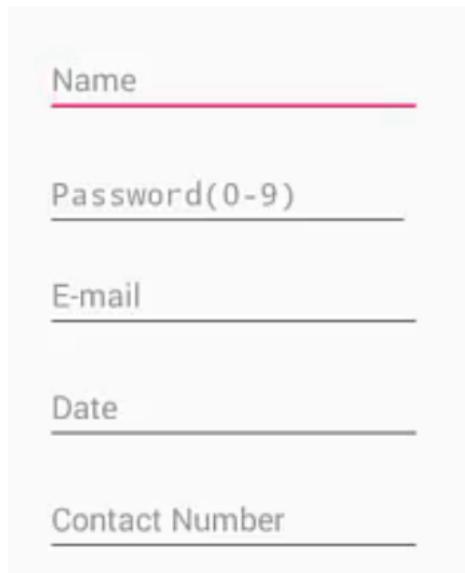
Hình 2. 4 - Widget Button

ImageView và ImageButton là 2 thành phần widget của Android cho phép tích hợp images vào ứng dụng. Cả 2 widget này đều khá tương đồng với TextView và Button. Mỗi một widget có thuộc tính android:src hoặc android:background trong file XML layout để chỉ định cần sử dụng ảnh nào. Picture sẽ thường nằm trong resource drawable. ImageButton là một subclass của ImageViews. Nó thêm một thuộc tính respond với click event của Button vào.



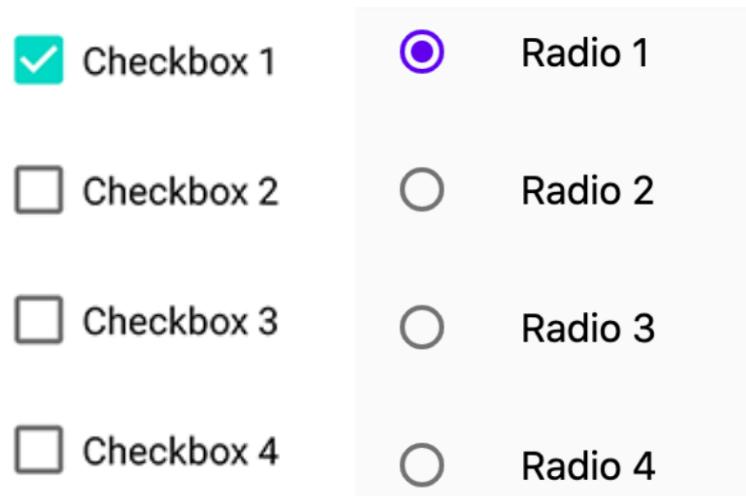
Hình 2. 5 - Widget ImageButton

EditText( hoặc là textBox) : là một mảng rộng của TextView cho phép chỉnh sửa, cập nhật nội dung văn bản.



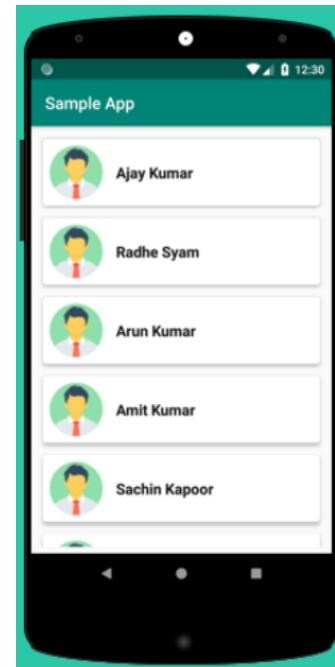
Hình 2. 6 - Widget EditText

Checkbox & RadioButtons: hay hộp kiểm, hộp lựa chọn hoặc hộp đánh dấu là một hộp tương tác nhỏ có thể được người dùng bật tắt để cho biết lựa chọn khẳng định hoặc phủ định.



Hình 2. 7 - Widget Checkbox và RadioButton

RecyclerView & ListView: là một widget cho phép thêm các phần tử như một danh sách.



Hình 2. 8 - Widget RecyclerView/ListView

Switch: Widget cho phép người dùng chuyển đổi trạng thái: bật/tắt



Hình 2. 9 - Widget Switch

Ngoài ra còn rất nhiều widget khác nhưng trên đây là các widget cơ bản và thường được sử dụng nhiều nhất.

### 2.3 Ngôn ngữ lập trình Kotlin

#### 2.3.1 Kotlin là gì?

Kotlin được phát triển bởi JetBrains. Về cơ bản, cũng như Java, C hay C++ , Kotlin cũng là "ngôn ngữ lập trình kiểu tĩnh" được sử dụng dành cho những ứng dụng thuộc dạng đa nền tảng hiện đại. Nghĩa là các biến không cần phải định nghĩa trước khi sử dụng. Kiểu tĩnh thực hiện việc kê khai nghiêm ngặt hoặc khởi tạo các biến trước khi chúng được sử dụng.

Sự xuất hiện của ngôn ngữ lập trình Kotlin đã khiến cho ngôn ngữ Java trong Android bị “soán ngôi”. Từ đó, Kotlin trở thành một trong những ngôn ngữ thông dụng hơn với Android. Kotlin đã được thiết kế với khả năng tương tác cùng với Java code cũng như có thể khai thác những gì có ở bên trong Java Class Library.



Hình 2. 10 - Hình so sánh Kotlin và Java

Sau khi trở thành ngôn ngữ lập trình cơ bản của Android vào năm 2019 thì Kotlin được Google hỗ trợ rất đầy đủ để có thể được tham gia đầy đủ vào các gói cài đặt của IDE để có thể thay thế cho trình biên dịch của Java dạng tiêu chuẩn.

### 2.3.2 *Ưu điểm nổi bật của ngôn ngữ lập trình Kotlin*

Kotlin có thể khắc phục được các yếu điểm của Java:



Hình 2. 11 - Kotlin vs Java

- **Code ngắn gọn và dễ hiểu:** Ngôn ngữ lập trình Kotlin được xây dựng bằng hệ thống code ít giúp lập trình viên dễ đọc, dễ viết và dễ làm việc cùng. Những người mới bắt đầu đều có thể tiếp thu dễ dàng các kiến thức đặc thù của loại ngôn ngữ này. Việc tối giản được số lượng code đã giúp cho Kotlin mang lại những trải nghiệm thú vị hơn cho người dùng so với các loại ngôn ngữ khác như Java.

- Kotlin không bị lỗi NullPointerException: NullPointerException là 1 trong những lỗi thường xuyên xuất hiện trong các dự án được viết bằng ngôn ngữ lập trình Java. Lỗi NullPointerException sẽ xuất hiện ngay khi gán giá trị null đến với một đối tượng nào đó, tuy nhiên khi truy xuất thì đối tượng này lại bị xuất hiện lỗi. Khi chạy trên Android, nếu như quên cập nhật đối tượng cho Java thì sẽ nhận được log crash (dừng đột ngột) đã được báo cáo về hệ thống. Những lỗi NullPointerException.Kotlin đều được thiết kế để có thể giảm thiểu cũng như loại bỏ được hầu hết các nguồn tham chiếu Null dựa vào cơ chế null-safety. Chính vì vậy, theo các chuyên gia thì việc sử dụng ngôn ngữ lập trình Kotlin sẽ trở nên an toàn hơn Java rất nhiều.
- Kotlin có khả năng tương tác cao: Có thể chạy trên máy ảo tương tự như Java. Tuy nhiên, khả năng tương tác cao giúp cho Kotlin có thể tương thích 100% với Java nên 1 dự án có thể sử dụng cả Java và Kotlin. Hiện nay, các developer Android đều có khả năng sử dụng Java class library ngay khi dùng Kotlin để có thể thực hiện viết code và ngược lại. Nhờ vậy, bạn có thể sử dụng ngôn ngữ này để có thể phát triển cũng như mở rộng cho việc phát triển các dự án Java bằng cũ mà không cần phải bắt lại. có thể sử dụng Java class library khi dùng Kotlin để viết code và ngược lại. Điều này có nghĩa là có thể sử dụng Kotlin để bạn mở rộng và phát triển các Với những lập trình viên đã quen làm việc với Java khi chuyển sang hợp tác cùng Kotlin thì sẽ không còn cảm thấy bỡ ngỡ hay xa lạ bởi cú pháp của nó bây giờ đều rất quen thuộc.
- Kotlin được hỗ trợ ưu tiên hỗ trợ trong Android Studio và IDE: Có thể dễ dàng tận dụng các IDE được tích hợp từ Android Studio 3.0. Những phiên bản Android Studio thấp hơn cần phải thực hiện cài thêm plugin và nó khiến cho cấu hình Kotlin trong dự án trở nên đơn giản hơn. Hiện tại, các IDE hỗ trợ cho Java đều sẽ có thể hỗ trợ cho cả Kotlin. Chính vì thế, hầu hết các Developer đều có thể tận dụng IDE trong số đó cả cả Android Studio. Kotlin cùng với tool làm việc thân thiện sẽ hỗ trợ lựa chọn Java IDE làm việc hoặc thực hiện làm việc cùng với command line.

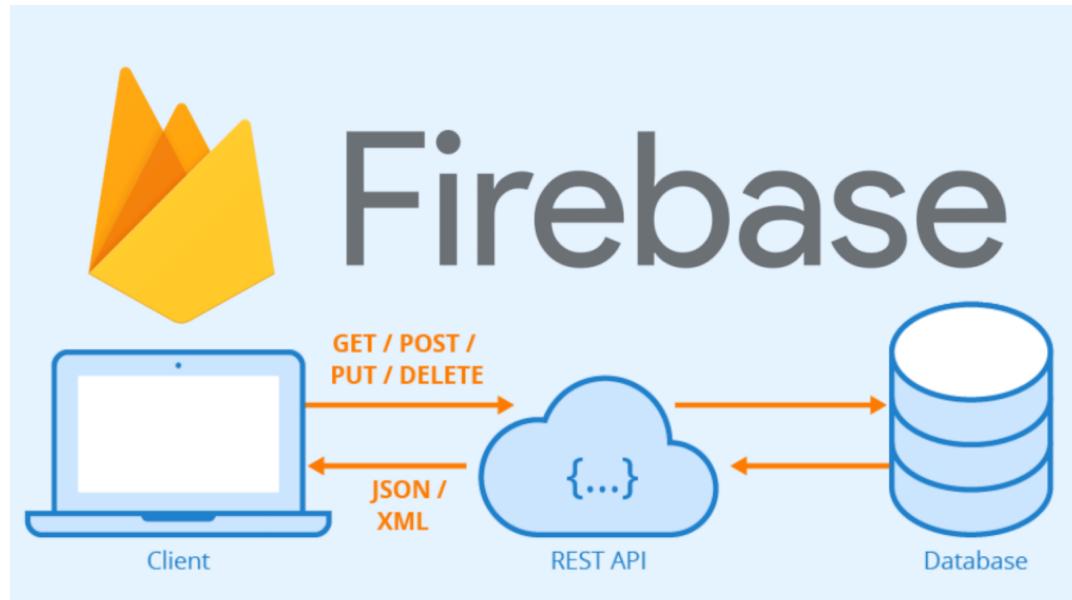
### **2.3.3 Hạn chế của ngôn ngữ lập trình Kotlin**

- Chưa có kiểu Aliases: Vì chưa có kiểu Aliases nên kiểu hàm vẫn còn phải viết thủ công nên phần mã nguồn sẽ bị thừa thãi.
- Các mặc định class trong Kotlin là final: Cần phải thêm từ khóa Open nếu như muốn class final trở thành class thông thường như trong Java. Đây chính là hạn chế có thể khiến cho các dự án có mã nguồn kết hợp giữa Kotlin và Java. Bởi vì một số Java Framework thường tự động bỏ qua từ khóa Final trong mã của Kotlin. Điều này sẽ khiến cho Kotlin có thể chạy không đúng với ý đồ của lập trình viên.
- Cộng đồng hỗ trợ hạn chế: Mặc dù là ngôn ngữ có thể sử dụng được toàn bộ cũng như thư viện của Java nhưng theo nhiều lập trình viên thì phiên bản chính chủ vẫn tốt hơn.
- Kotlin không tự ép kiểu dữ liệu: Kotlin sẽ không thực hiện tự động ép kiểu với những dữ liệu thuộc dạng nguyên thủy.

## **2.4 Firebase**

### **2.4.1 Firebase là gì?**

Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.



Hình 2. 12 - Hình dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây - Firebase

Firebase là một trong những BaaS (Backend as a Service), tức là một dịch vụ cung cấp các giải pháp backend cho các ứng dụng web và di động. Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

Firebase được ra đời vào năm 2011 bởi James Tamplin và Andrew Lee với tên gọi ban đầu là Evolve, một nền tảng cung cấp các API để tích hợp tính năng chat vào các trang web.

Sau đó, họ nhận ra rằng nền tảng này được sử dụng để truyền dữ liệu ứng dụng chứ không chỉ là chat. Họ đã phát triển Evolve thành Firebase và công bố nó vào tháng 4 năm 2012. Đến tháng 10 năm 2014, Firebase đã được Google mua lại và trở thành một phần của Google Cloud Platform.

### 2.4.2 Tổng quan về các nhóm sản phẩm/dịch vụ của Firebase

- Cloud Firestore: Một cơ sở dữ liệu NoSQL đám mây có thể mở rộng để lưu trữ và truy vấn dữ liệu cho ứng dụng.



Hình 2. 13 - Hình mô tả Firebase Cloud Firestore

- Authentication: Một dịch vụ xác thực người dùng hỗ trợ nhiều phương thức đăng nhập khác nhau, như email, mật khẩu, số điện thoại, Google, Facebook, Twitter...



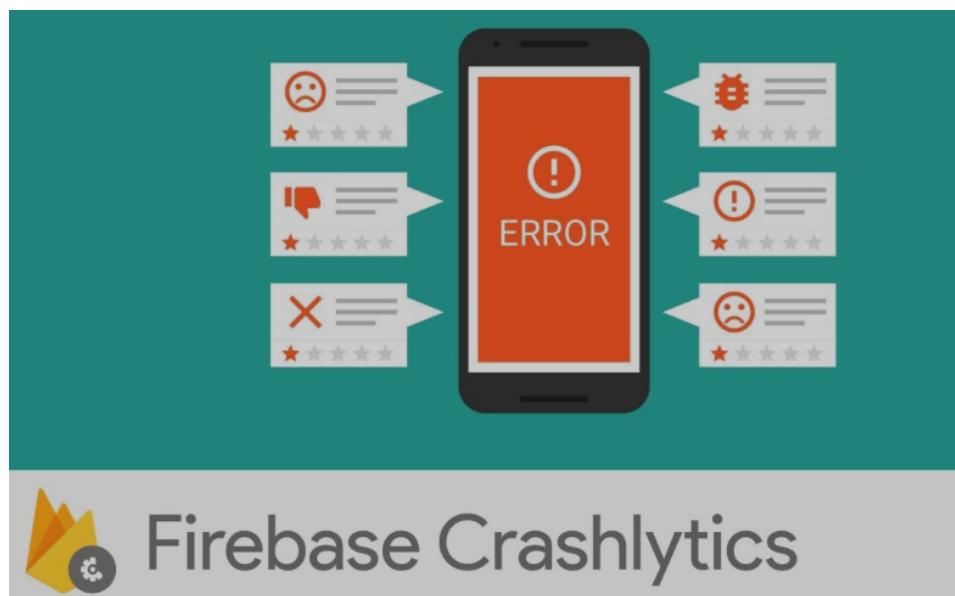
Hình 2. 14 - Hình mô tả Firebase Authentication

- Extensions: Những gói mã nguồn mở được đóng gói sẵn để tự động hóa các tác vụ phát triển thông thường, như đồng bộ hóa dữ liệu giữa Cloud Firestore và BigQuery, tích hợp với Algolia để tìm kiếm, gửi email khi có sự kiện xảy ra, xử lý thanh toán với Stripe...



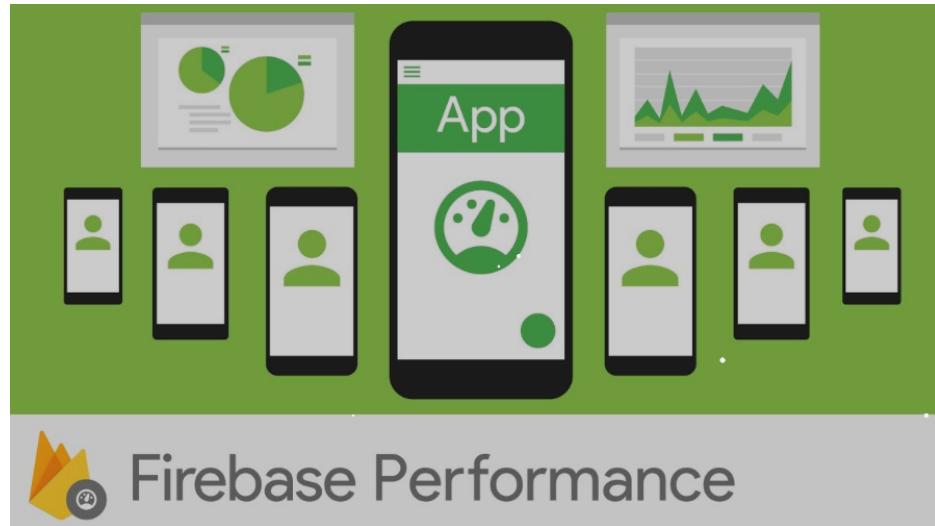
Hình 2. 15 - Hình mô tả Firebase Extensions

- Crashlytics: Một dịch vụ báo cáo sự cố giúp xác định, ưu tiên và khắc phục các lỗi gây ra sự cố trong ứng dụng.



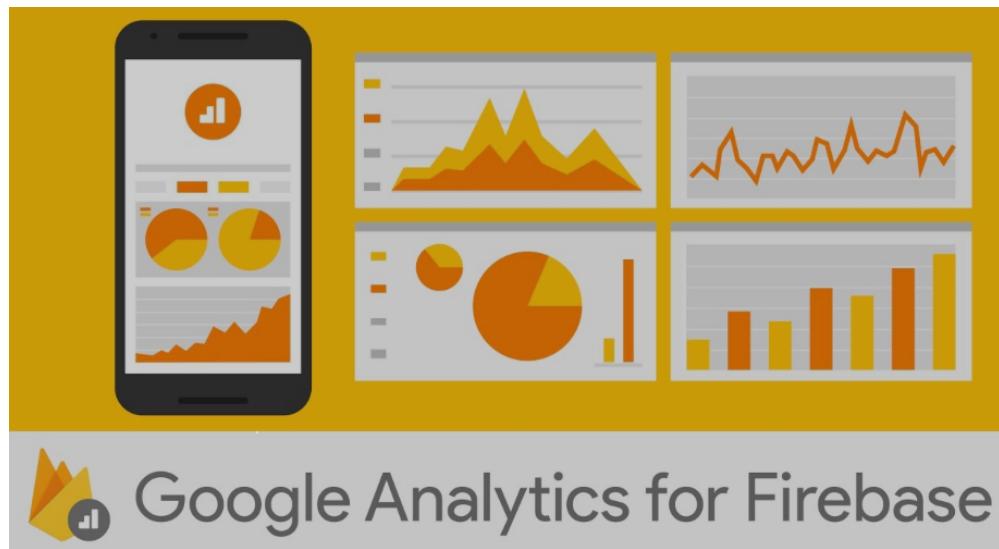
Hình 2. 16 - Hình mô tả Firebase Crashlytics

- Performance Monitoring: Một dịch vụ giám sát hiệu suất giúp đo lường và cải thiện hiệu suất của ứng dụng như thời gian tải ứng dụng, thời gian phản hồi của mạng, tiêu thụ pin và nhiều hơn nữa.



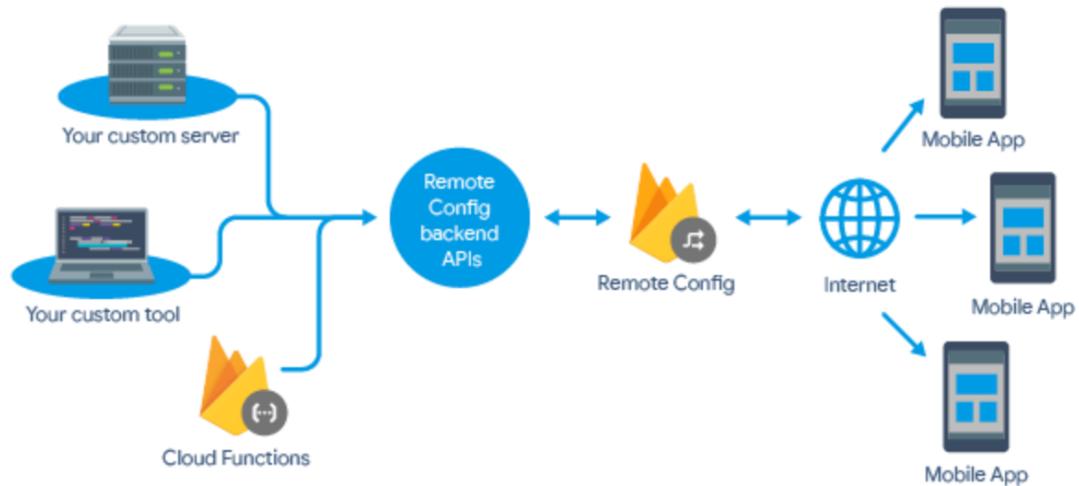
Hình 2. 17 - Hình mô tả Firebase Performance

- Firebase Analytic: Một dịch vụ phân tích hành vi người dùng giúp hiểu được người dùng là ai, họ làm gì và tại sao họ sử dụng ứng dụng.



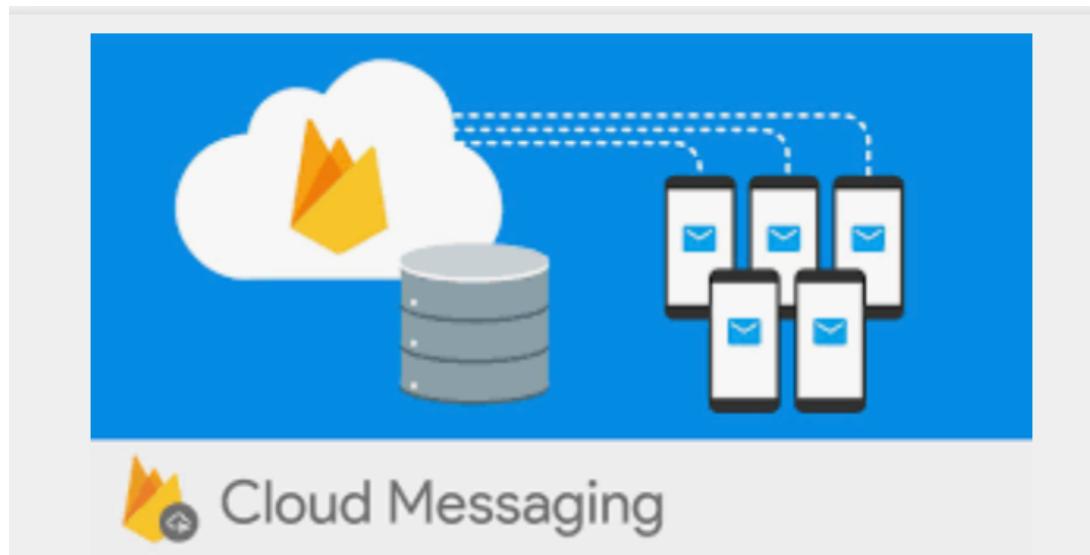
Hình 2. 18 - Hình mô tả Firebase Analytics

- Remote Config: Một dịch vụ cho phép thay đổi hành vi và giao diện của ứng dụng mà không cần phải phát hành phiên bản mới.



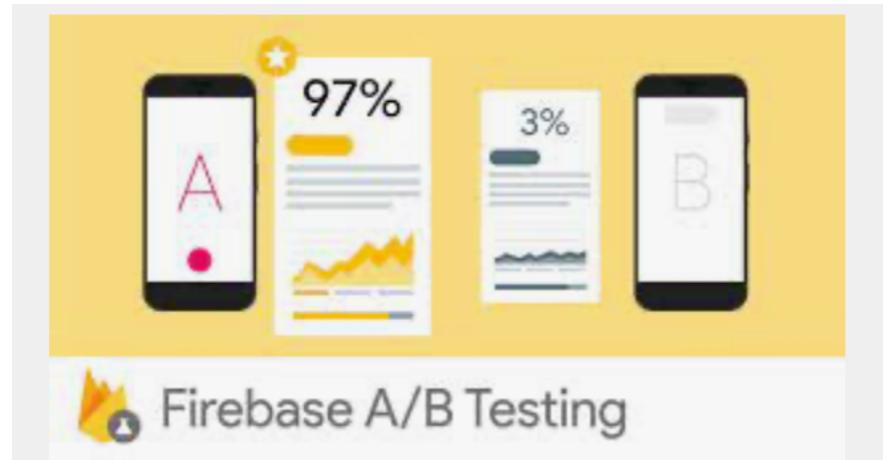
Hình 2. 19 - Hình mô tả Firebase Remote Config

- Cloud Messaging: Một dịch vụ cho phép gửi tin nhắn và thông báo đẩy cho người dùng, như thông báo về các sự kiện, khuyến mãi, cập nhật tin tức,...



Hình 2. 20 - Hình mô tả Firebase Cloud Messaging

- A/B Testing: Một dịch vụ cho phép chạy các thí nghiệm để so sánh hiệu quả của các phiên bản khác nhau của ứng dụng, như màu sắc, văn bản, tính năng và nhiều hơn nữa.



Hình 2. 21 - Hình mô tả Firebase A/B Testing

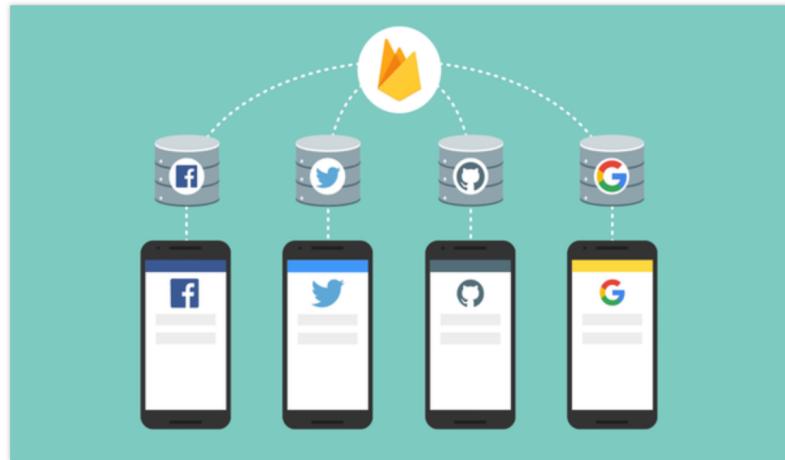
#### **2.4.3 Các tính năng chính của Firebase**

- **Realtime Database:** Cơ sở dữ liệu thời gian thực, lưu trữ dưới dạng JSON, đồng bộ hóa với mọi kết nối nhanh chóng. Realtime Database cho phép lưu trữ và truy vấn dữ liệu một cách hiệu quả, không cần thiết lập server hay viết code backend.



Hình 2. 22 - Hình mô tả Firebase Realtime Database

- **Authentication:** Hoạt động nổi bật của Firebase là xây dựng các bước xác thực người dùng bằng Email, Facebook, Twitter, GitHub, Google. Xác thực cho các ứng dụng. Hoạt động xác thực có thể giúp thông tin cá nhân của người sử dụng được an toàn và đảm bảo không bị đánh cắp tài khoản.



Hình 2. 23 - Hình mô tả chi tiết Firebase Authentication

- **Firebase Hosting:** Dịch vụ deploy trang web và app bằng thao tác đơn giản, có tính an toàn cao, phù hợp cho việc hiển thị các trang như điều khoản dịch vụ, chính sách bảo mật. Hosting cho phép deploy trang web lên một máy chủ ổn định và bảo mật.



Hình 2. 24 - Hình mô tả Firebase Hosting

- **Firebase Cloud Messaging:** Dịch vụ gửi nhận tin nhắn miễn phí, có thể sử dụng để push thông báo cho người dùng khi có tin nhắn mới, sự kiện mới... Firebase Cloud Messaging cho phép bạn gửi tin nhắn đến các thiết bị Android, iOS hoặc website của người dùng một cách đơn giản và hiệu quả, không cần quan tâm đến việc thiết lập server hay viết code backend. Bạn có thể sử dụng Firebase Cloud Messaging để tăng sự liên kết và thân thiện với người dùng, thông báo cho họ về các tin tức mới nhất hoặc các khuyến mãi hấp dẫn.

Ngoài ra, còn có những tính năng khác như: Firebase Analytics, Firebase Authentication, Firebase Storage, Firebase Crashlytics,... Tất cả các chức năng này đều được quản lý thông qua một giao diện đồ họa trực quan và dễ sử dụng, là Firebase Console.

#### **2.4.4 Ưu điểm của Firebase**

Firebase giúp tiết kiệm thời gian và chi phí phát triển ứng dụng, bằng cách cung cấp các dịch vụ và công cụ sẵn có và dễ sử dụng.

Firebase cho phép các nhà phát triển tập trung vào việc xây dựng giao diện và tính năng của ứng dụng, mà không cần lo lắng về việc xử lý và lưu trữ dữ liệu ở phía máy chủ.

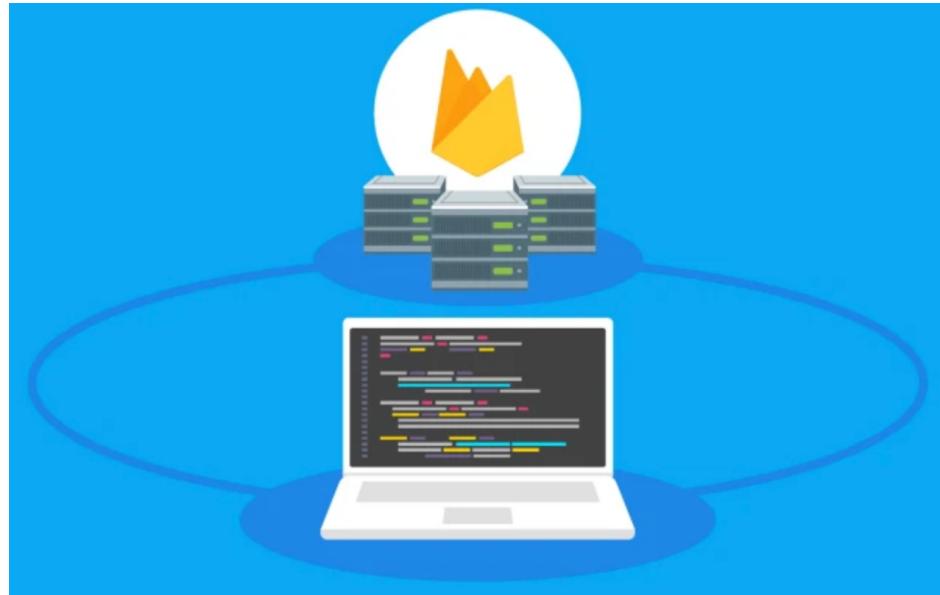
Firebase hỗ trợ nhiều nền tảng khác nhau, như web, Android, iOS, Unity, C++,...

Firebase có khả năng mở rộng tự động theo quy mô của ứng dụng, không cần phải quản lý cơ sở hạ tầng hay máy chủ.

Firebase được hưởng lợi từ hệ thống máy chủ mạnh mẽ và an toàn của Google, đảm bảo hiệu suất cao và độ tin cậy cao cho các ứng dụng.

Firebase hỗ trợ nhiều ngôn ngữ lập trình và nền tảng khác nhau, cho phép phát triển các ứng dụng đa nền tảng một cách hiệu quả và dễ dàng.

Firebase có nhiều tính năng hữu ích cho việc phát triển ứng dụng như xác thực người dùng, lưu trữ tệp tin, tích hợp thanh toán, thực hiện kiểm thử phần mềm và gỡ lỗi.



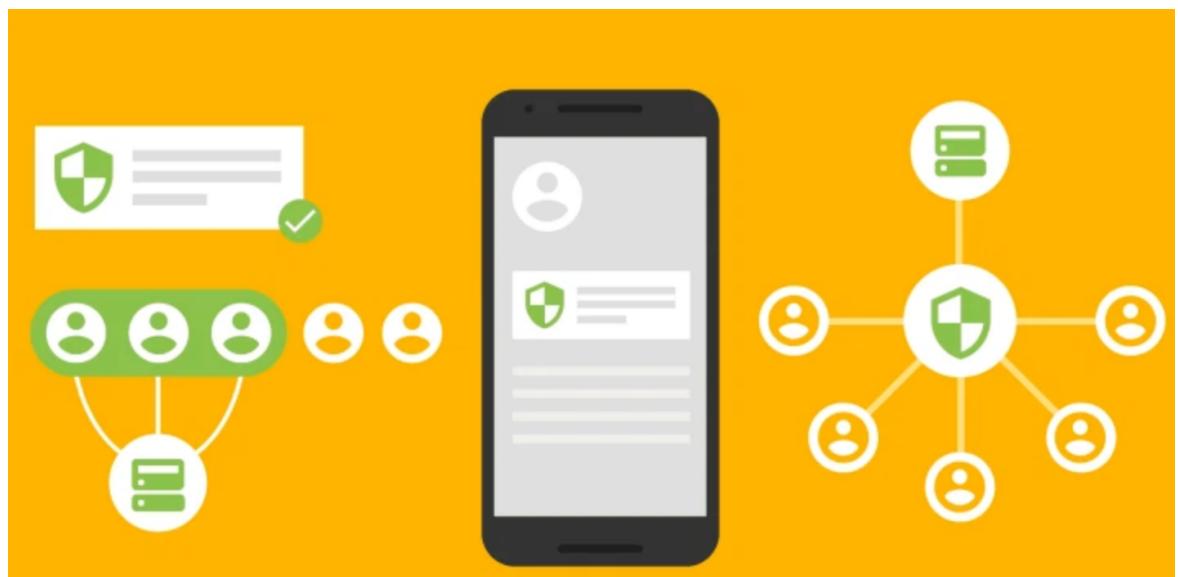
Hình 2. 25 - Hình minh họa ưu điểm Firebase

#### **2.4.5 Nhược điểm của Firebase**

Firebase có giới hạn về dung lượng lưu trữ và số lượng kết nối trong phiên bản miễn phí. Để sử dụng các tính năng nâng cao hoặc quy mô lớn hơn, các nhà phát triển phải trả phí theo mức sử dụng.

Firebase có thể không đáp ứng được một số yêu cầu đặc biệt hoặc phức tạp của các ứng dụng. Ví dụ, Firebase Realtime Database chỉ hỗ trợ cơ sở dữ liệu NoSQL, không hỗ trợ các truy vấn phức tạp hay các thao tác với dữ liệu liên quan.

Firebase có thể gặp khó khăn trong việc tích hợp với một số công nghệ hoặc dịch vụ khác. Ví dụ, Firebase không hỗ trợ GraphQL hay Apollo Client, hai công nghệ phổ biến trong việc xây dựng API hiện đại.



Hình 2. 26 - Hình minh họa nhược điểm Firebase

## CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1 Đặt vấn đề

Khi người dùng có ý tưởng sáng tạo vừa nghĩ ra cần một nơi để ghi chú lại mà không có sẵn bút hay viết. Điều này có thể dẫn đến sự mất mát về thời gian và ý tưởng khi người dùng phải đi tìm bút viết để ghi lại ý tưởng. Với sự phát triển về công nghệ cũng như nhu cầu của người dùng thì giao diện các ứng dụng hiện tại không thể đáp ứng được.

Vì vậy, nhóm chúng em muốn tạo ra ứng dụng có thiết kế giao diện thân thiện đẹp mắt, bao gồm các tone màu xanh biển giúp dịu mắt người dùng. Đồng thời ứng dụng cung cấp đầy đủ các chức năng cũng như bổ sung các chức năng report cho người dùng và báo lỗi cho hệ thống. Tất cả đều nhằm đem đến cho người dùng trải nghiệm tốt nhất.

### 3.2 Đặc tả hệ thống

#### 3.2.1 *Chức năng đăng nhập*

Sau khi tạo tài khoản, người dùng xác thực email sau đó người dùng có thể đăng nhập vào ứng dụng bằng email đã đăng ký và xác thực

#### 3.2.2 *Chức năng đăng ký*

Người dùng vào mục đăng ký điền họ tên, email, mật khẩu để đăng ký tài khoản mới. Sau khi đăng ký, người dùng vào email để xác thực tài khoản. Dữ liệu của đăng ký của người dùng sẽ được lưu trữ thời gian thực trên firebase.

#### 3.2.3 *Chức năng đổi mật khẩu*

Người dùng có thể đổi mật khẩu trong mục thông tin cá nhân. Người dùng cần nhập mật khẩu hiện tại để xác thực tài khoản người dùng, sau đó nhập mật khẩu mới để đổi mật khẩu.

#### 3.2.4 *Chức năng quên mật khẩu*

Nếu người dùng quên mật khẩu thì có thể vào mục này, điền email đã đăng kí tài khoản. Sau đó Firebase Authentication sẽ gửi mail xác thực qua cho người dùng, người dùng truy cập vào link đó để nhập mật khẩu mới.

### **3.2.5 Chức năng cập nhật thông tin**

Người dùng sau khi đăng kí thì trên cơ sở dữ liệu chỉ có thông tin họ và tên và email. Người dùng có thể cập nhật thêm các thông tin khác như địa chỉ, trang web, nghề nghiệp, hình ảnh avatar,...

### **3.2.6 Chức năng nhắc nhở**

Người dùng khi tạo ghi chú có thể cài đặt thêm lịch deadline. Khi gần đến deadline, ứng dụng sẽ gửi 1 notification đến cho người dùng.

### **3.2.7 Chức năng thay đổi ngôn ngữ**

Hiện tại ứng dụng hỗ trợ 2 ngôn ngữ là tiếng Việt và tiếng Anh, người dùng có thể thay đổi ngôn ngữ hiển thị trong mục cài đặt.

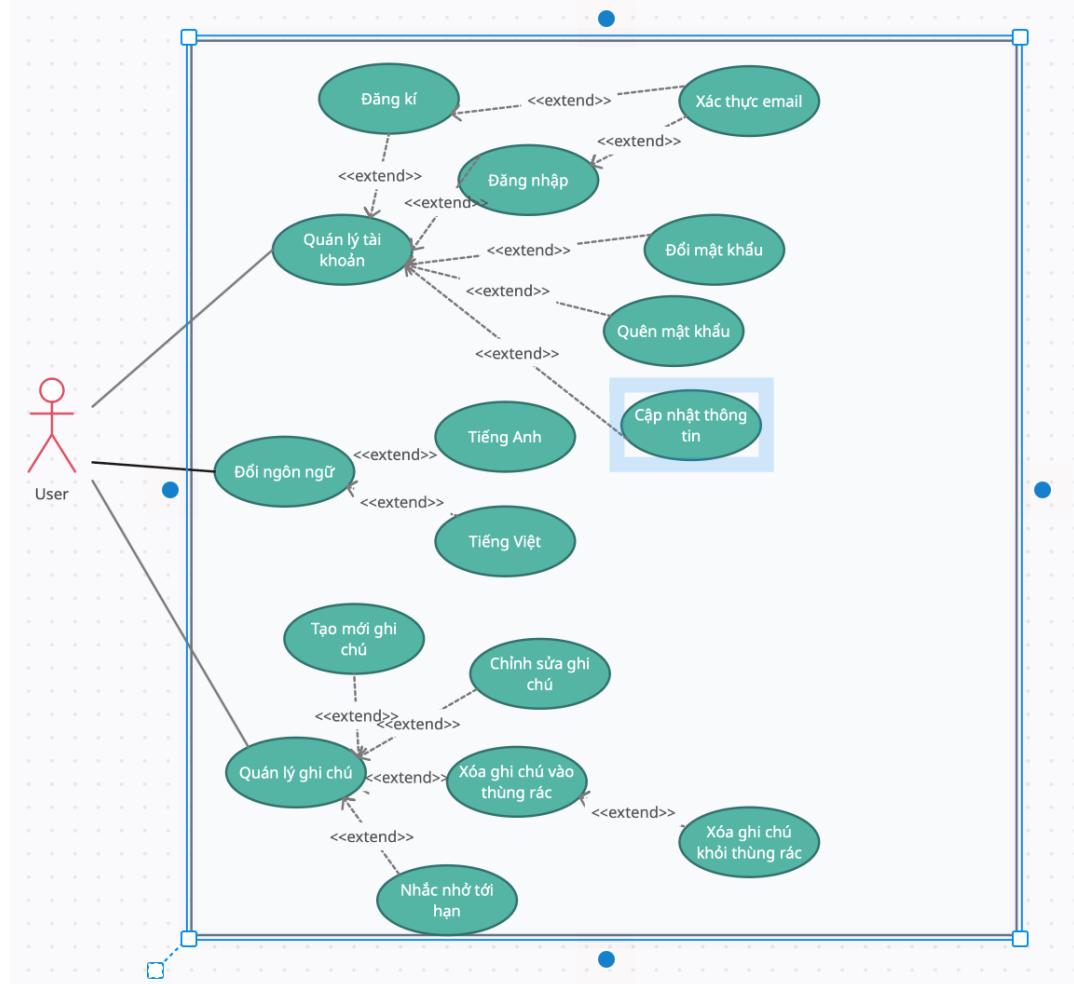
### **3.2.8 Chức năng về ghi chú**

Người dùng có thể tạo mới ghi chú, sửa nội dung thông tin ghi chú cũ đã tạo từ trước, xóa ghi chú vào thùng rác, xóa vĩnh viễn ghi chú ra khỏi cơ sở dữ liệu.

## **3.3 Phân tích và thiết kế**

### **3.3.1 Mô hình usecase tổng quát**

Mô hình use case tổng quát mô tả các chức năng của ứng dụng và sự tương tác của người dùng đối với ứng dụng.



Hình 3. 1 - Hình mô tả usecase tổng quát

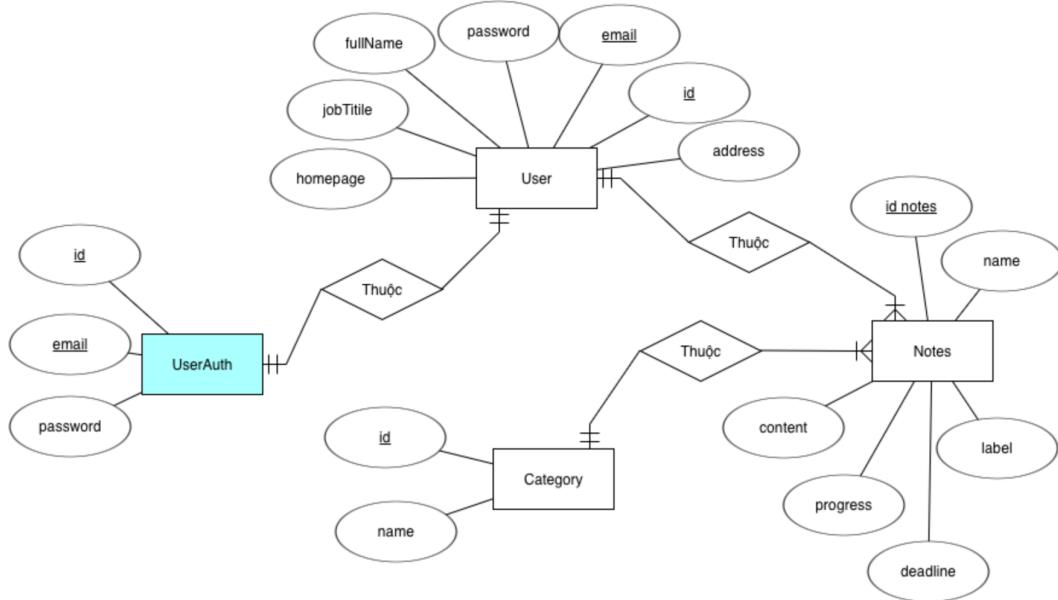
### 3.3.2 Đặc tả

Thông tin về người dùng sẽ bao gồm các thông tin như họ tên, email, id người dùng, nghề nghiệp, địa chỉ, avatar, link trang cá nhân. Những thông tin sẽ được lưu trữ thời gian thực trên Firebase Realtime Database. Ngoài ra email và mật khẩu của người dùng cũng sẽ được lưu trong Firebase Authentication để xác thực người dùng.

Thông tin về ghi chú bao gồm các thông tin như id người dùng, id của ghi chú, tên ghi chú, nội dung, tiến độ, nhãn, deadline. Thông tin id của người dùng sẽ được tham chiếu tới bảng người dùng. Mỗi ghi chú chỉ có thể có 1 id người dùng, nhưng 1 người

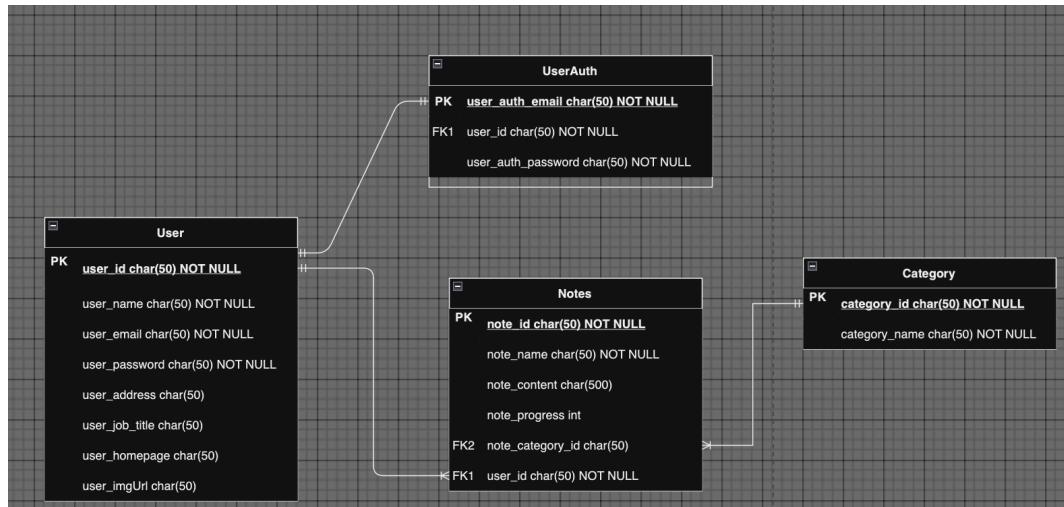
dùng có thẻ sở hữu nhiều ghi chú nên đây sẽ là mối quan hệ 1 – nhiều. Người dùng có thẻ phân loại ghi chú theo nhãn để dễ dàng tìm kiếm và lưu trữ.

### 3.3.3 ERD



Hình 3. 2 - Hình sơ đồ ERD

### 3.3.4 Mô hình quan hệ

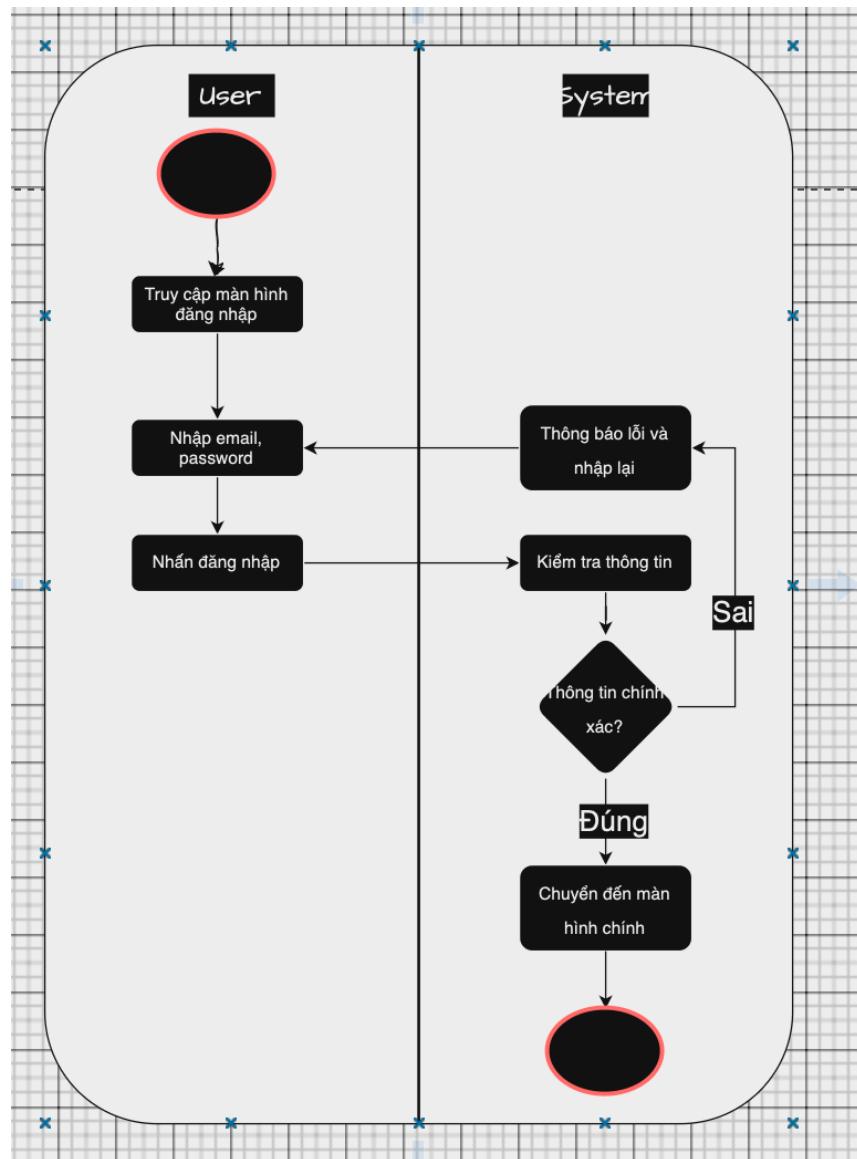


Hình 3. 3 - Hình mô hình quan hệ

### 3.3.5 Luồng hoạt động

- Hoạt động đăng nhập:

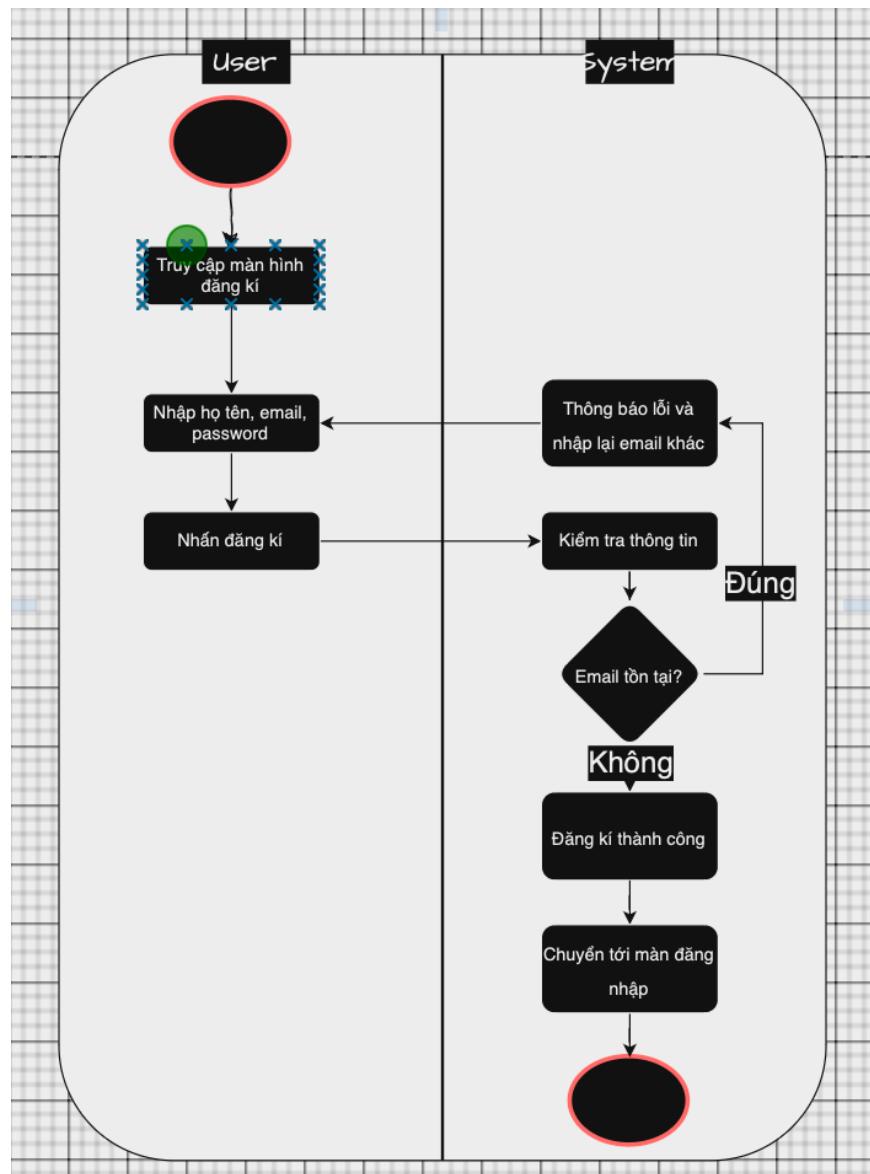
Mô tả: Khi muốn đăng nhập ta cần phải nhập thông tin theo yêu cầu, khi nhấn đăng nhập thì hệ thống sẽ kiểm tra thông tin, nếu thông tin lỗi thì sẽ thông báo lỗi và yêu cầu nhập lại. Nếu đúng thì sẽ chuyển đến màn hình chính của user.



Hình 3. 4 - Luồng hoạt động màn đăng nhập

- Hoạt động đăng ký:

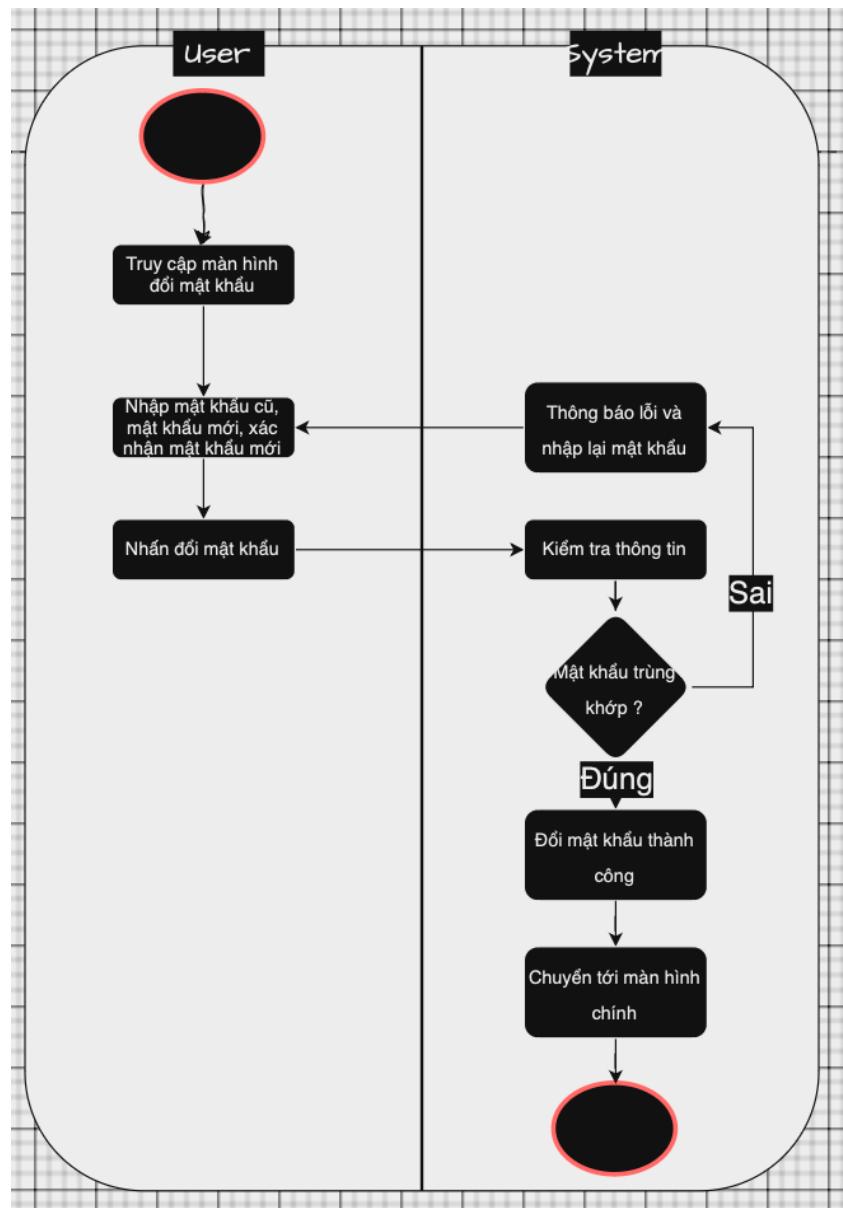
Mô tả: Khi muốn đăng kí ta cần phải nhập thông tin theo yêu cầu, khi nhấn đăng kí thì hệ thống sẽ kiểm tra thông tin, nếu email tồn tại thì sẽ thông báo lỗi và yêu cầu nhập lại email mới. Nếu chưa tồn tại thì đăng kí thành công và sẽ chuyển đến màn hình đăng nhập của



Hình 3. 5 - Luồng hoạt động màn đăng kí

- **Hoạt động đổi mật khẩu:**

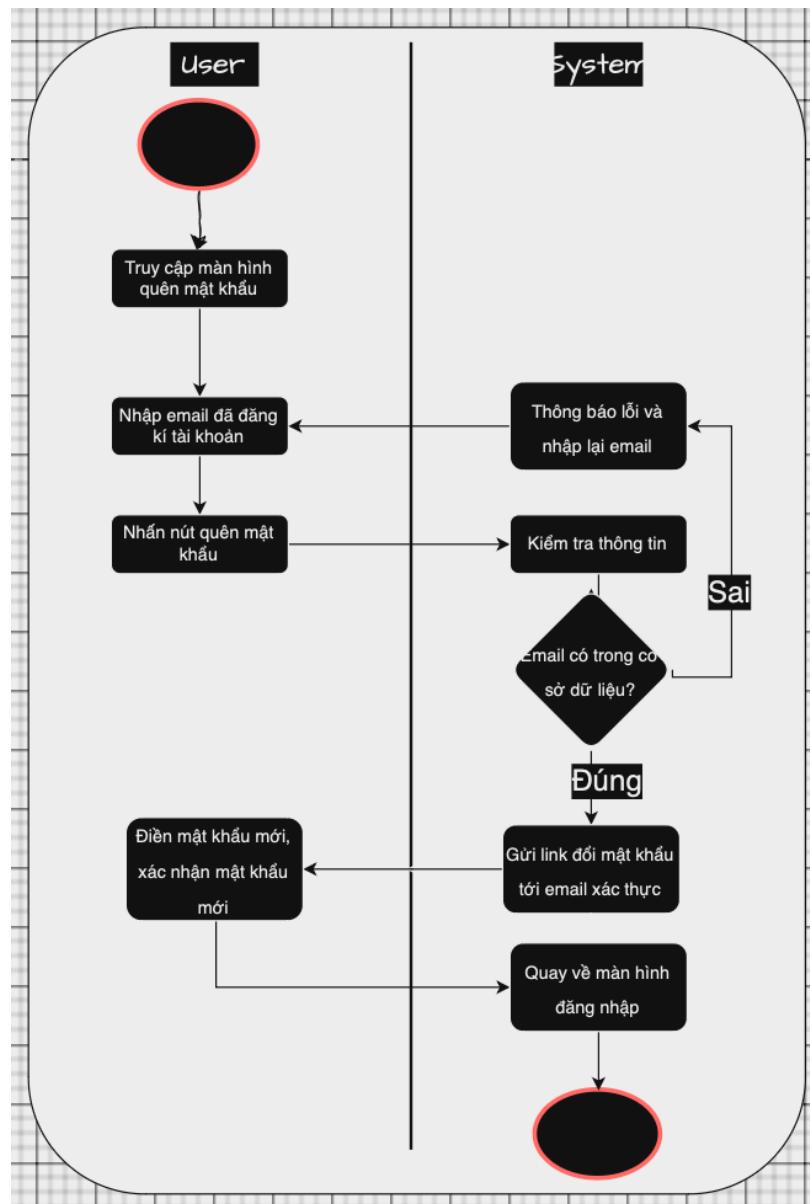
Mô tả: Khi muốn đổi mật khẩu ta cần phải nhập mật khẩu hiện tại, nhập tiếp mật khẩu mới, khi nhấn đổi mật khẩu thì hệ thống sẽ kiểm tra thông tin mật khẩu cũ, nếu trùng khớp thì tiến hành thay đổi mật khẩu, nếu sai thì sẽ thông báo lỗi và yêu cầu nhập lại.



Hình 3. 6 - Luồng hoạt động màn đổi mật khẩu

- Hoạt động quên mật khẩu:

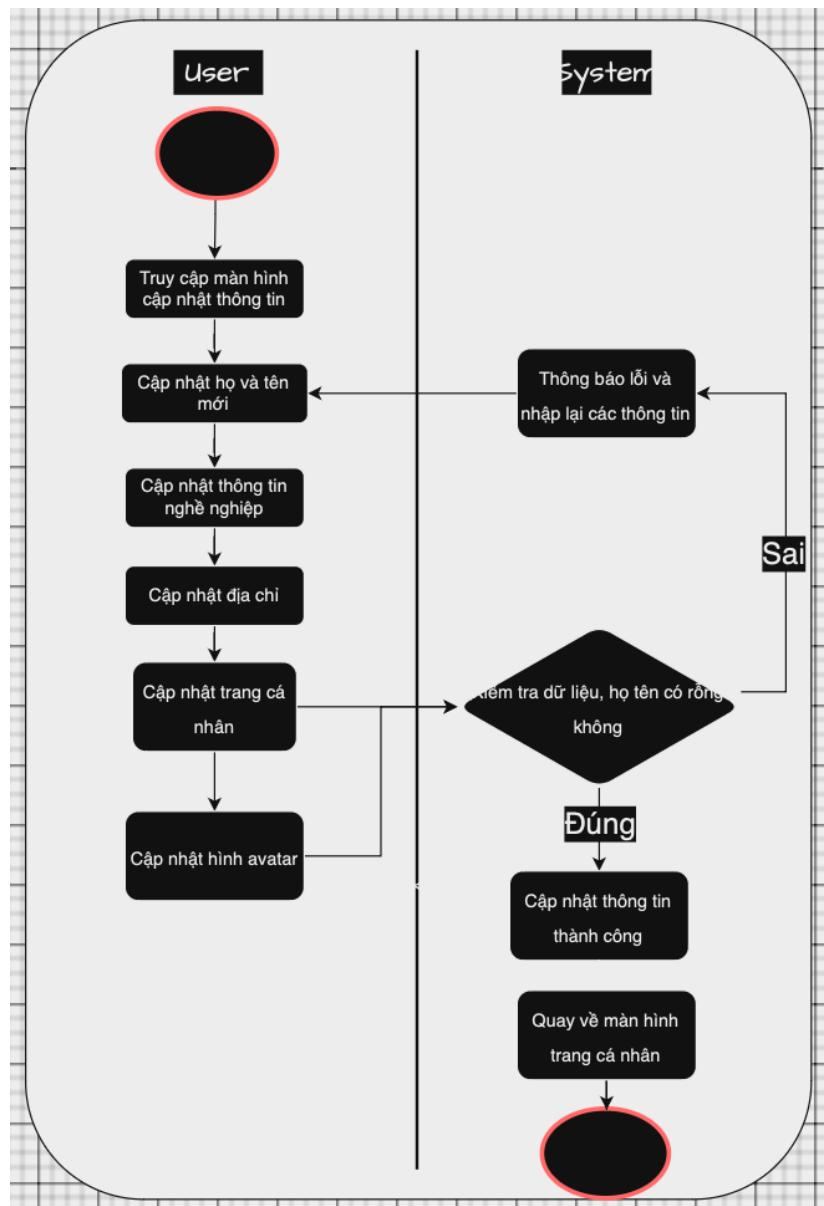
Mô tả: Khi người dùng quên mật khẩu ta cần phải nhập email đã đăng kí, khi nhấn nút quên mật khẩu thì hệ thống sẽ kiểm tra thông tin email và gửi link xác thực qua email đó nếu email đó tồn tại trong cơ sở dữ liệu, người dùng nhấp vào link để thay đổi mật khẩu mới, nếu sai thì sẽ thông báo lỗi và yêu cầu nhập lại email.



Hình 3. 7 - Luồng hoạt động màn hình quên mật khẩu

- Hoạt động cập nhật thông tin:

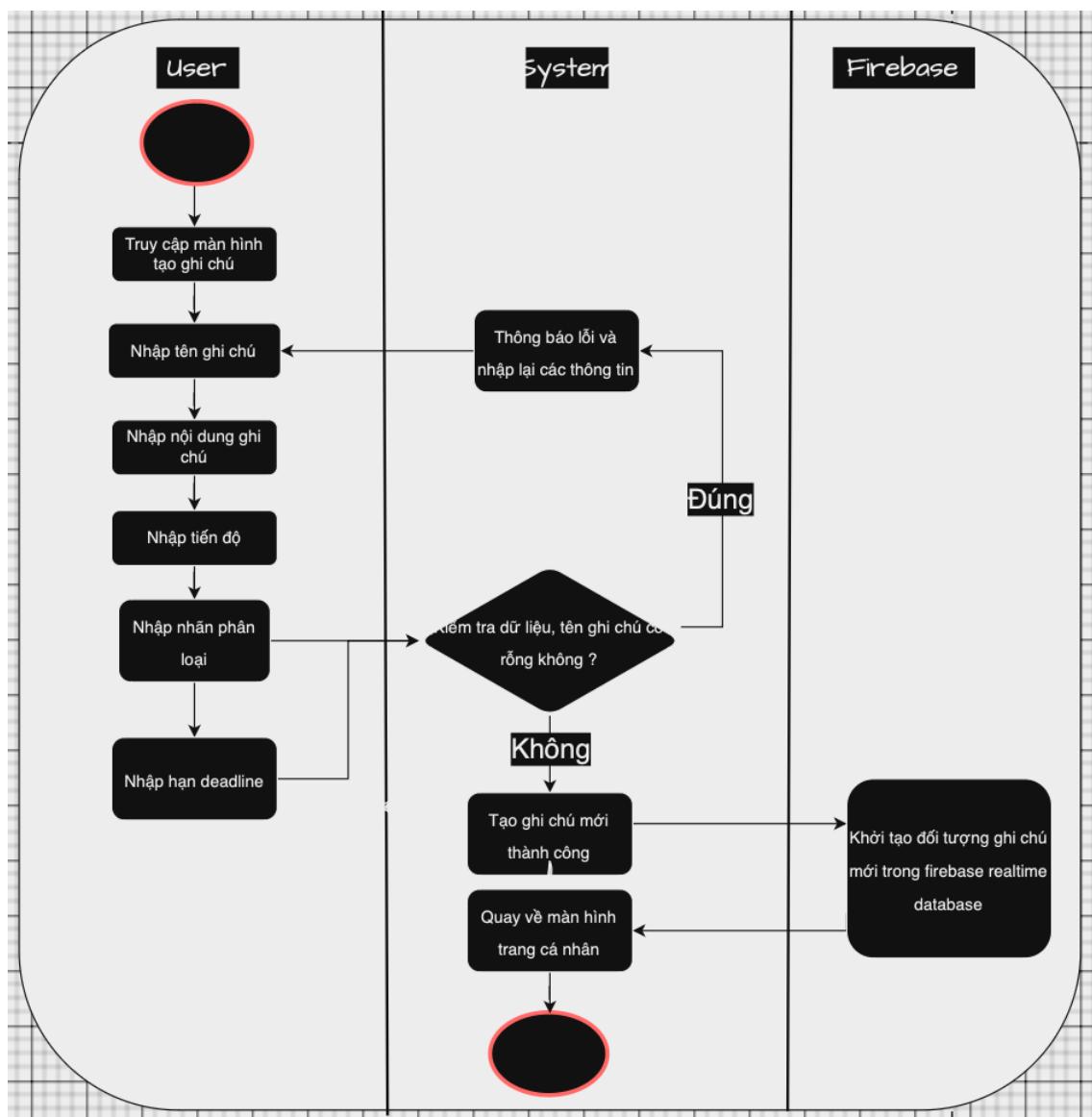
Mô tả: Khi người dùng cập nhật thông tin, người dùng điền hết thông tin vào các ô trống, riêng ô họ và tên không được để trống khi nhấn nút cập nhật thì hệ thống sẽ kiểm tra thông tin email và thông tin trong cơ sở dữ liệu, hệ thống sẽ cập nhật thông tin người dùng nếu dữ liệu trùng khớp, nếu sai sẽ thông báo lỗi và yêu cầu nhập lại.



Hình 3. 8 - Luồng hoạt động màn hình cập nhật thông tin cá nhân

- Hoạt động tạo ghi chú mới:

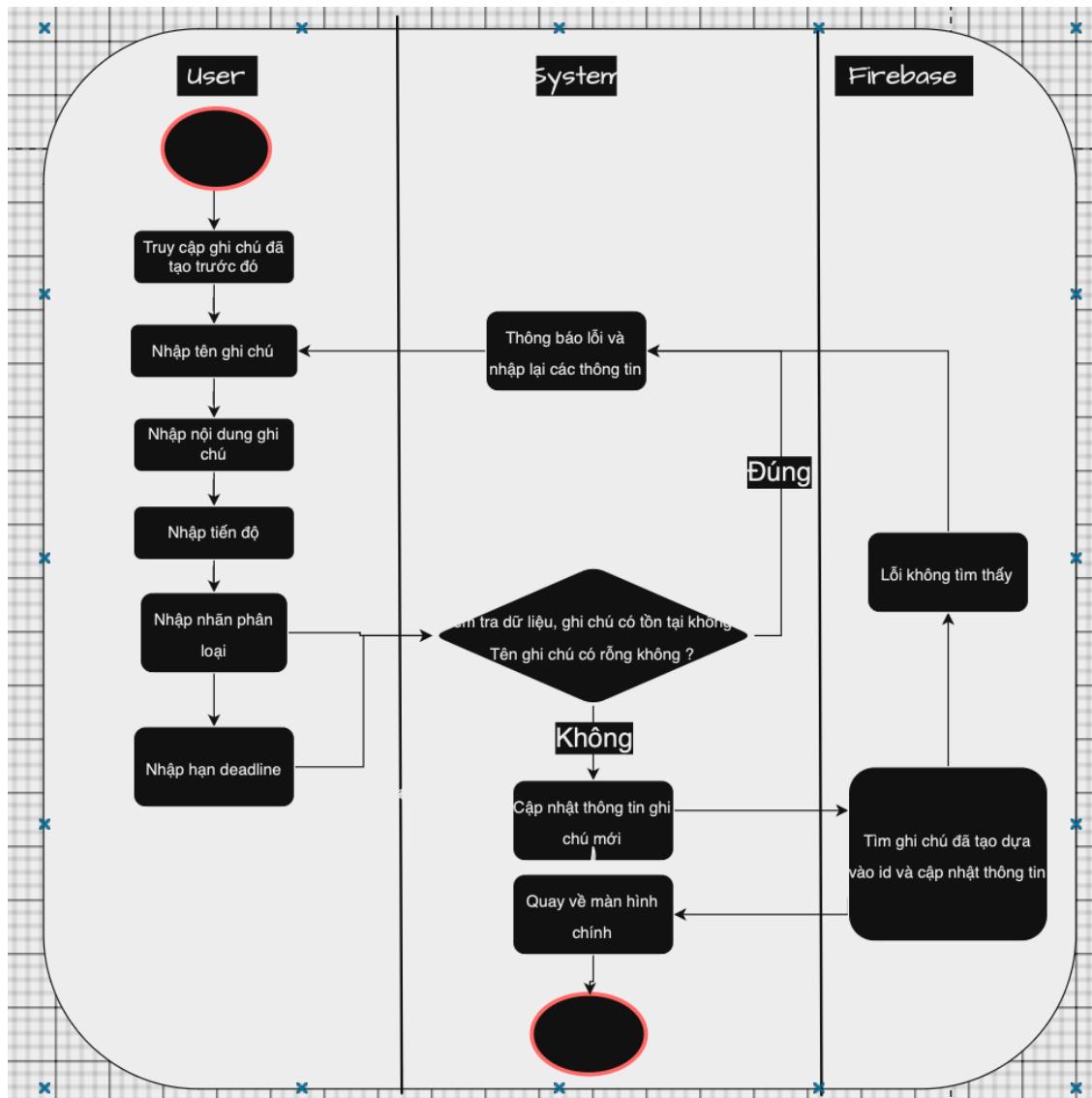
Mô tả: Khi người dùng nhấp vào nút tạo ghi chú mới, người dùng điền các thông tin như tên ghi chú, nội dung ghi chú, tiến độ, nhãn, các file đính kèm (nếu có). Hệ thống kiểm tra thông tin riêng ô tên ghi chú không được để trống. Nếu không có lỗi hệ thống tiến hành tạo ghi chú mới, ngược lại khi có lỗi thì hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại.



Hình 3. 9 - Luồng hoạt động màn hình tạo ghi chú mới

- Hoạt động chỉnh sửa thông tin ghi chú:

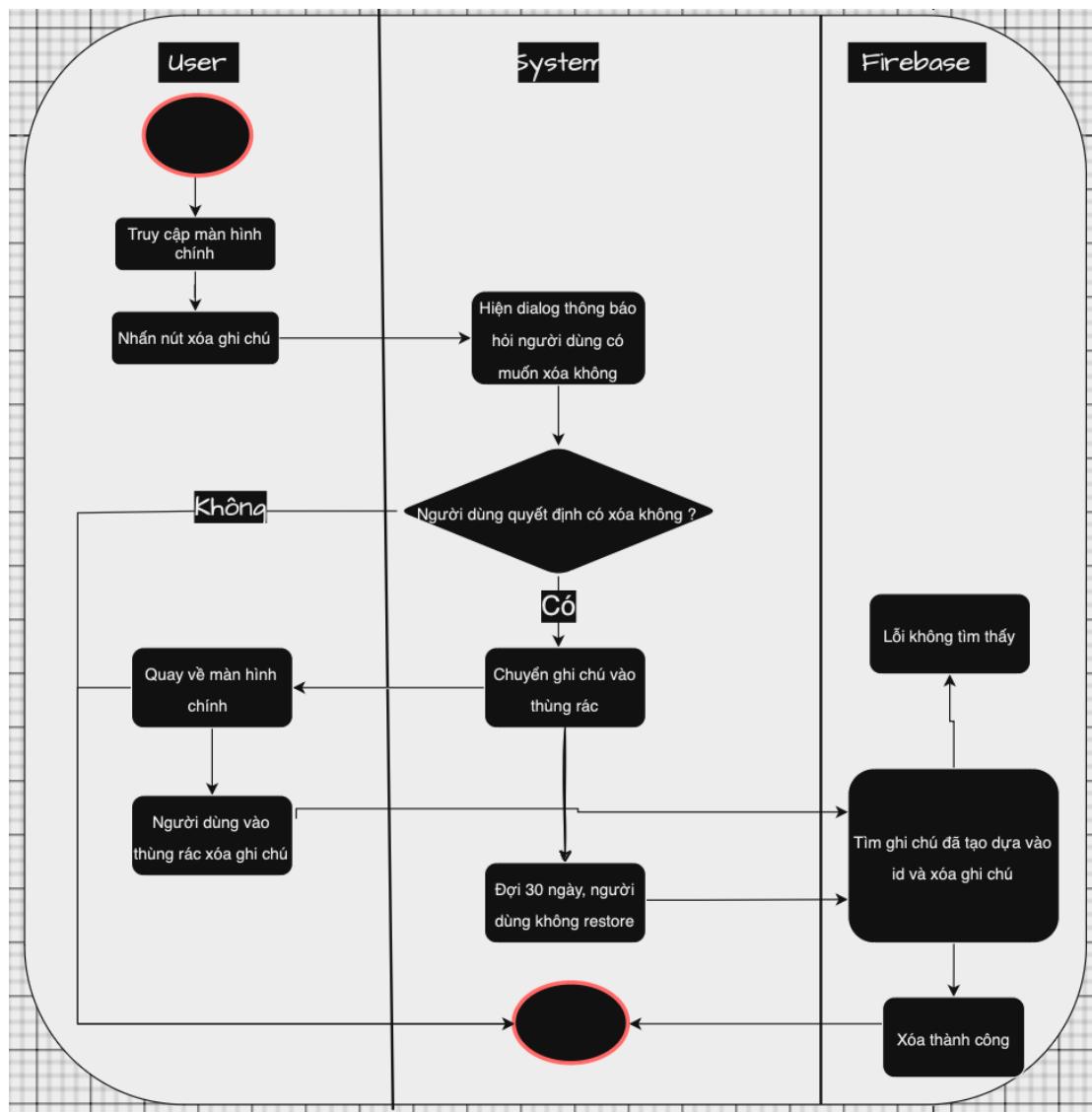
Mô tả: Khi người dùng nhấn vào ghi chú đã tạo, người dùng điền các thông tin như tên ghi chú, nội dung ghi chú, tiến độ, nhãn, các file đính kèm (nếu có). Hệ thống kiểm tra thông tin riêng ô tên ghi chú không được để trống. Nếu có ghi chú id tồn tại trong cơ sở dữ liệu thì hệ thống tiến hành cập nhật thông tin ghi chú, ngược lại khi không tìm thấy thì hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại.



Hình 3. 10 - Luồng hoạt động màn hình cập nhật thông tin ghi chú đã tạo

- Hoạt động xóa ghi chú:

Mô tả: Khi người dùng nhấp vào nút xóa ghi chú, hệ thống sẽ hiển thị một dialog để confirm người dùng có thật sự muốn xóa. Nếu người dùng chọn xóa thì ghi chú đó sẽ được chuyển vào thùng rác. Sau 30 ngày nếu không restore thì ghi chú sẽ tự động bị xóa vĩnh viễn khỏi cơ sở dữ liệu. Ngoài ra người dùng có thể vào thùng rác để xóa vĩnh viễn nhanh hơn mà không cần đợi 30 ngày.

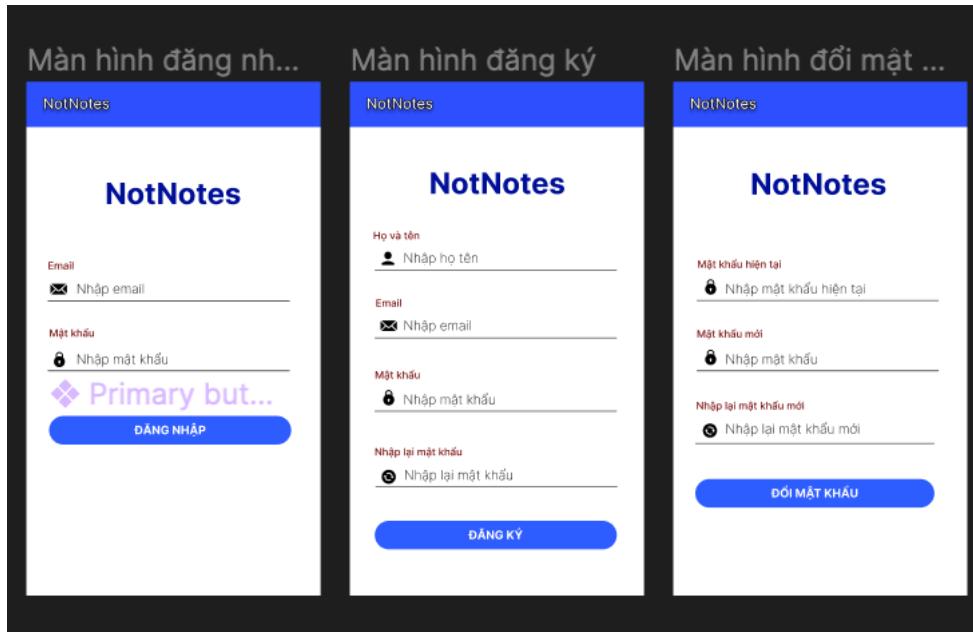


Hình 3. 11 - Luồng hoạt động màn hình xóa ghi chú

## CHƯƠNG 4 – CƠ SỞ THỰC TIỄN

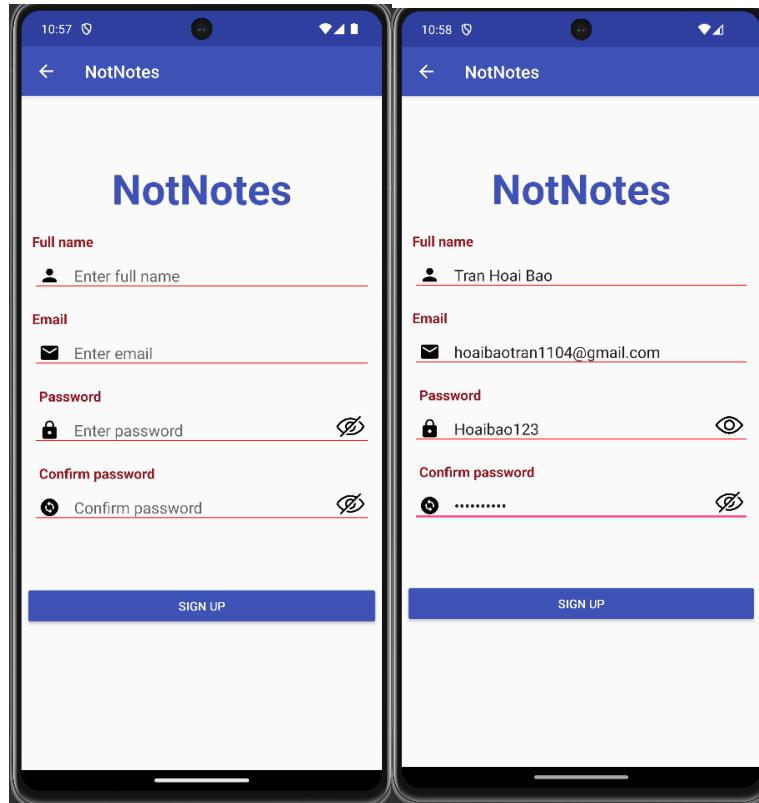
### 4.1 Bản thiết kế

Công cụ hỗ trợ thiết kế giao diện cho ứng dụng di động này là figma.



Hình 4. 1 - Hình thiết kế giao diện bằng figma

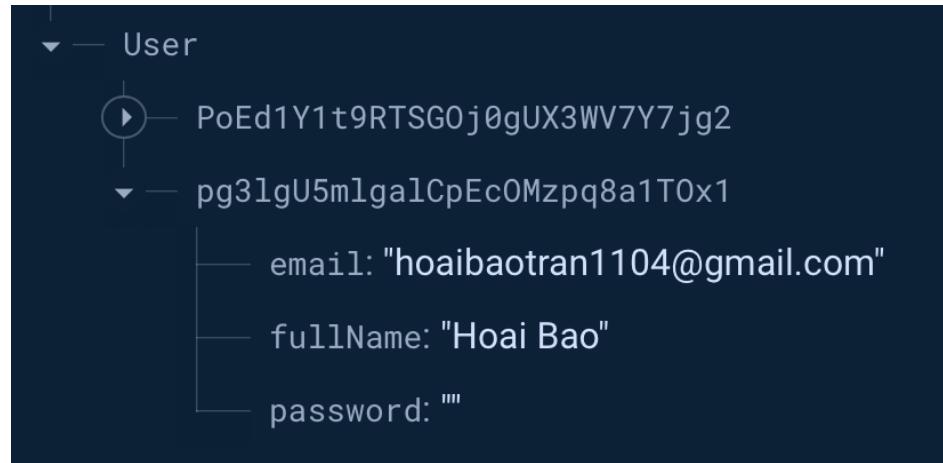
## 4.2 Đăng kí người dùng



Hình 4. 2 - Màn hình đăng kí thực tế của ứng dụng

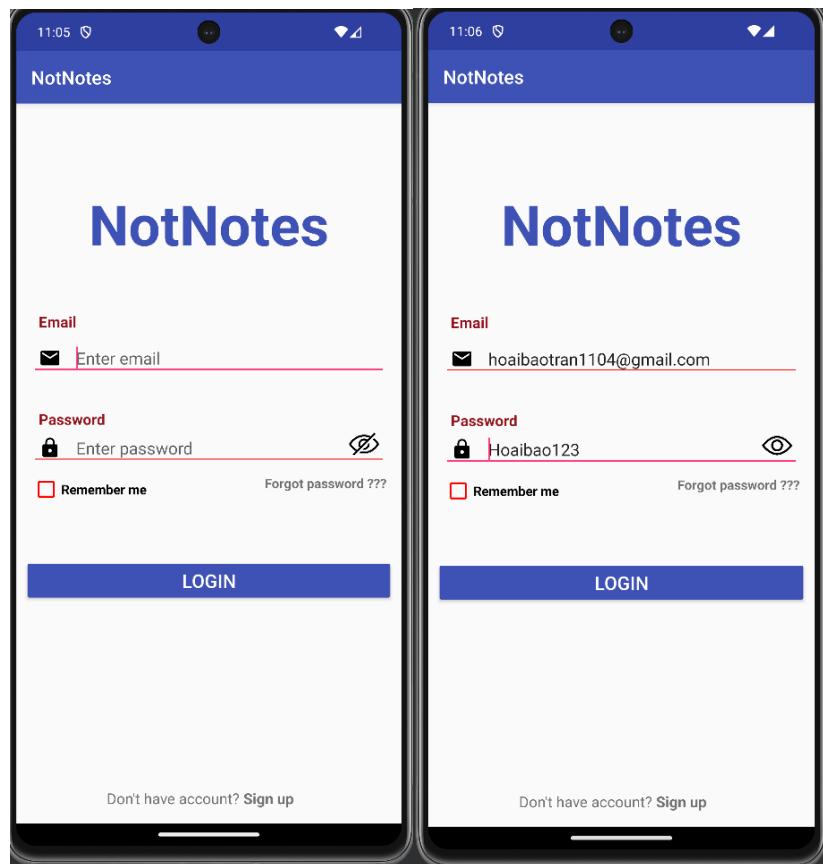
Authentication				
Users	Sign-in method	Templates	Usage	Settings
<input type="text"/> Search by email address, phone number, or user UID				<a href="#">Add user</a> <span style="margin-left: 10px;">C</span> <span style="margin-left: 10px;">⋮</span>
Identifier	Providers	Created	Signed In	User UID
anhkhoa5525@gmail.c...		Dec 20, 2023	Dec 20, 2023	PoEd1Y1t9RTSG0j0gUX3WV7...
hoaibaotran1104@gma...		Dec 20, 2023	Dec 21, 2023	pg3lgU5mgalCpEcOMzpq8a1...

Hình 4. 3 Hình ảnh hiện Firebase Authentication sau khi đăng kí



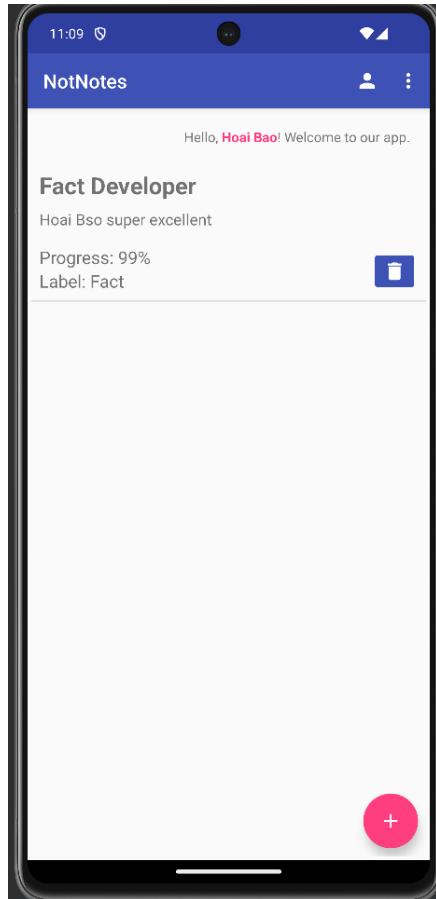
Hình 4. 4 – Hình thể hiện realtime database sau khi đăng kí

### 4.3 Đăng nhập người dùng



Hình 4. 5 - Màn hình đăng nhập thực tế của ứng dụng

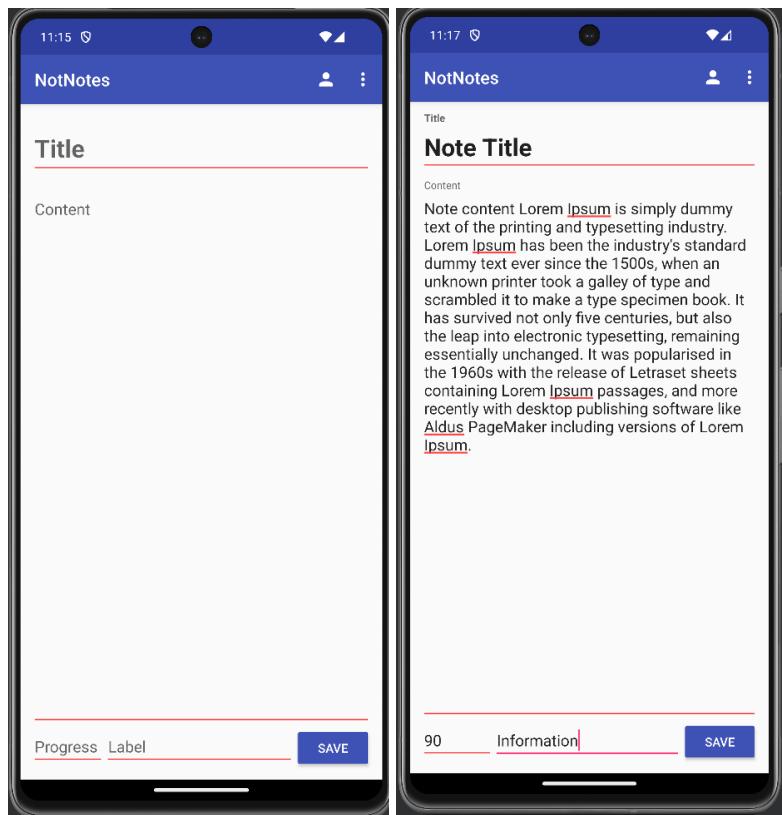
#### 4.4 Màn hình chính



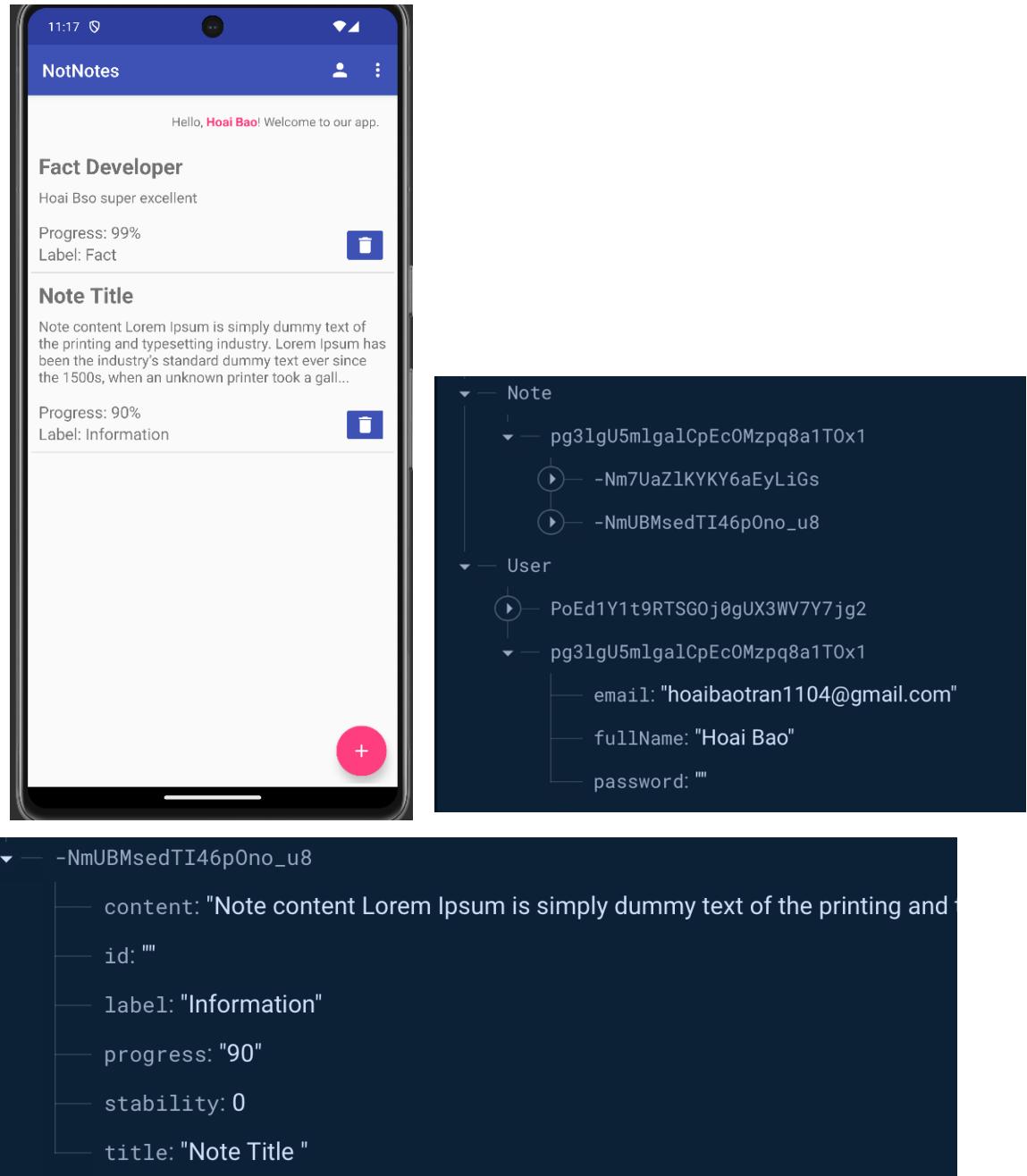
Hình 4. 6 - Hình thể hiện màn hình chính của ứng dụng

Màn hình này sẽ thể hiện 1 danh sách các ghi chú mà người dùng đã tạo ra trước đó. Khi nhấn nút dấu cộng ở góc dưới người dùng sẽ được dẫn tới màn hình thêm ghi chú mới.

## 4.5 Màn hình thêm ghi chú

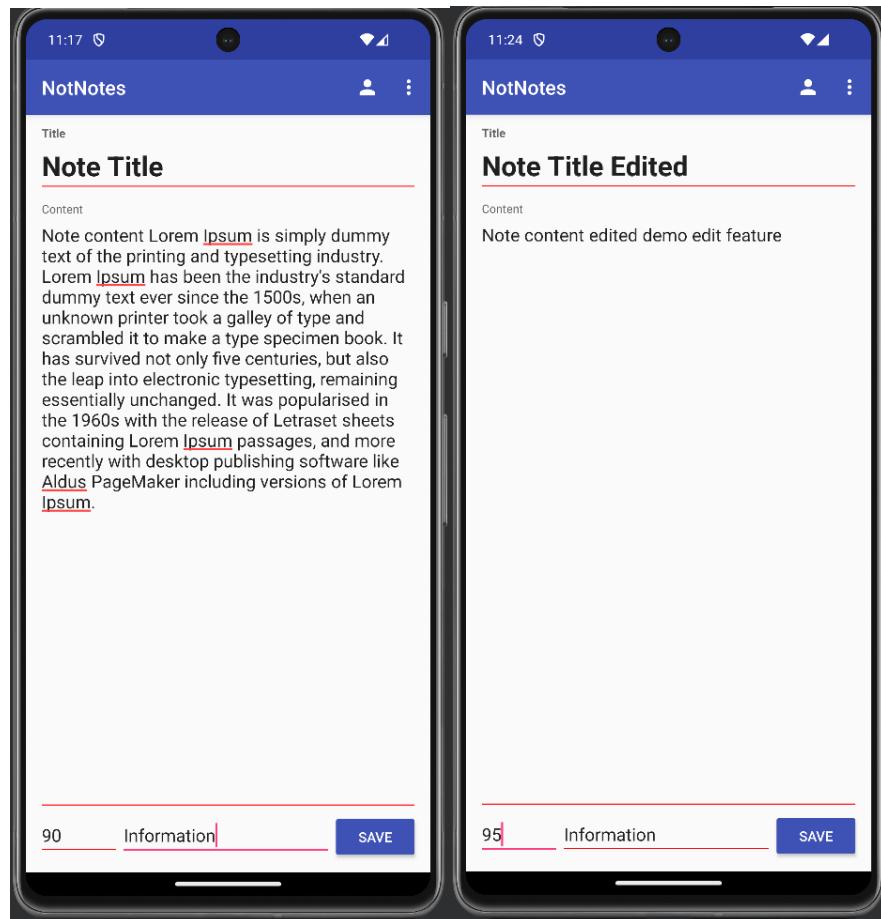


Hình 4. 7 - Màn hình thêm ghi chú thực tế của ứng dụng

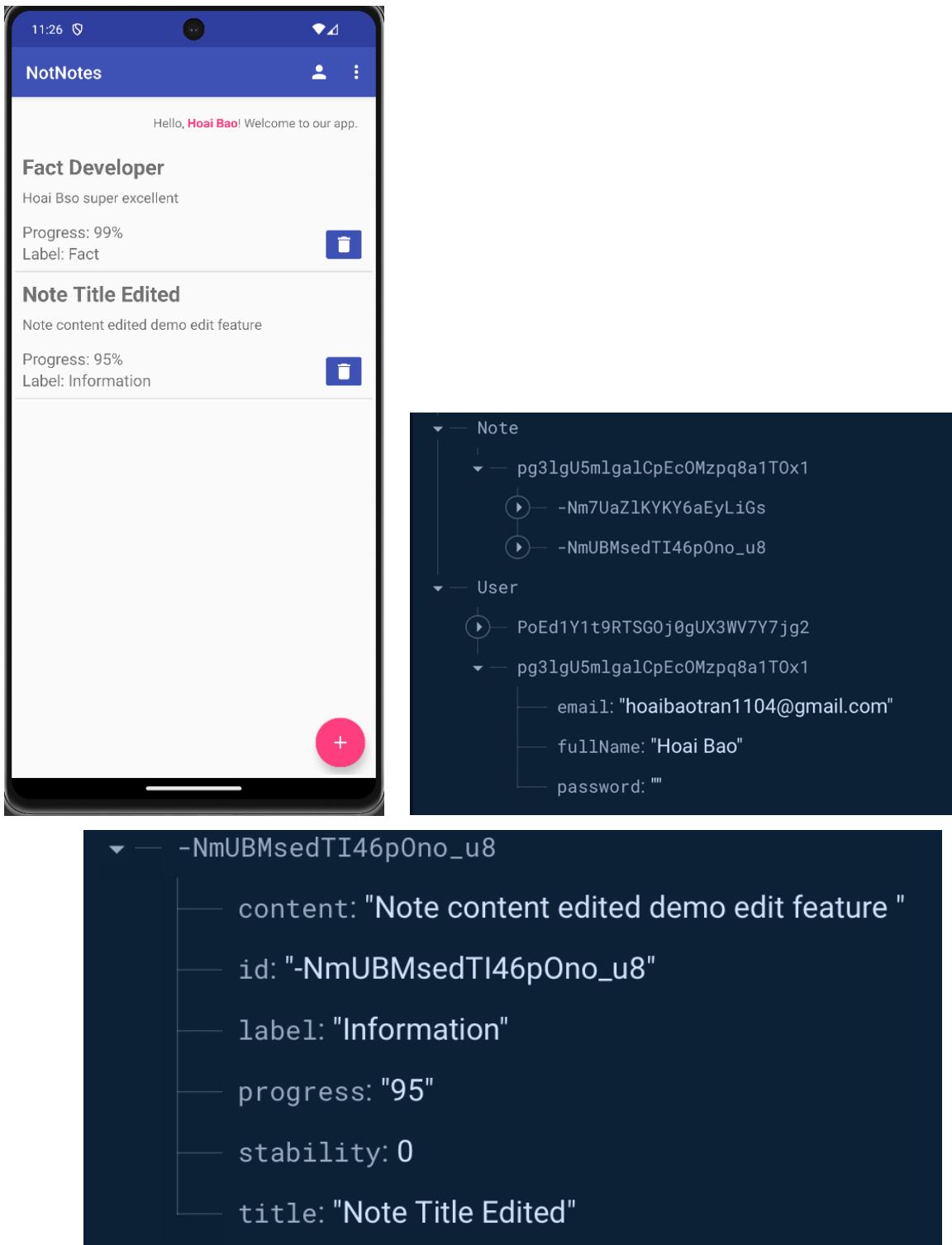


Hình 4. 8 - Firebase Readtime Database sau khi thêm note mới

## 4.6 Màn hình sửa ghi chú

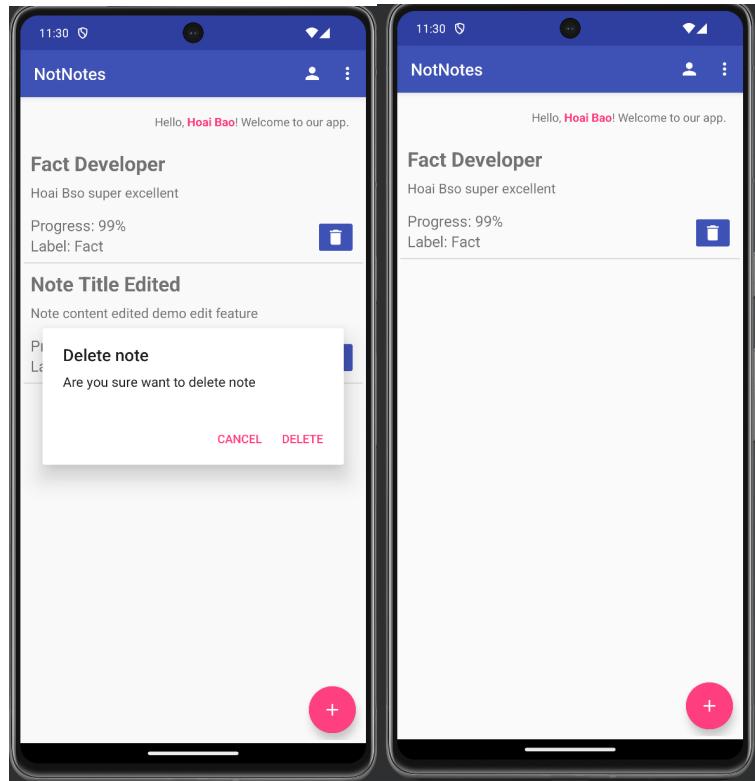


Hình 4. 9 - Màn hình sửa ghi chú thực tế của ứng dụng

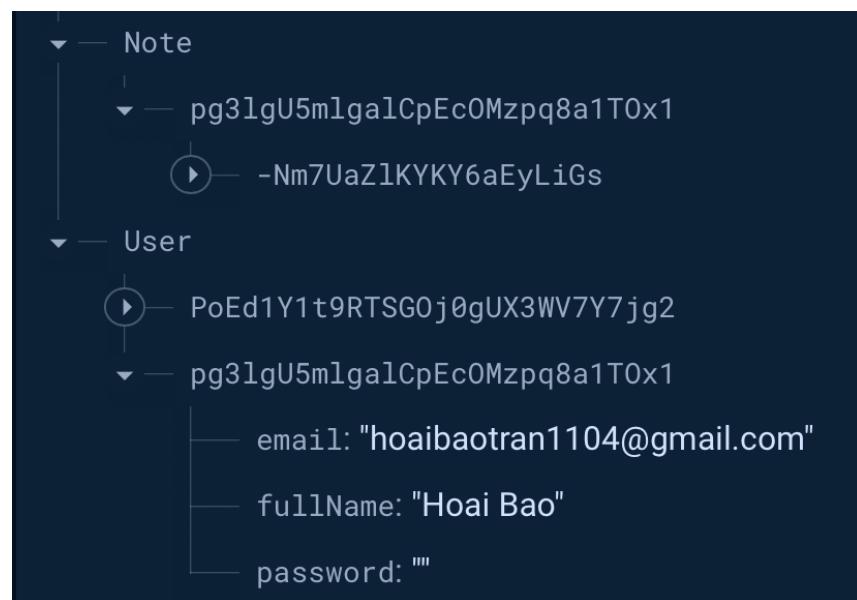


Hình 4. 10 - Firebase Realtime Database sau khi chỉnh sửa thông tin note đã được tạo

## 4.7 Màn hình xóa ghi chú

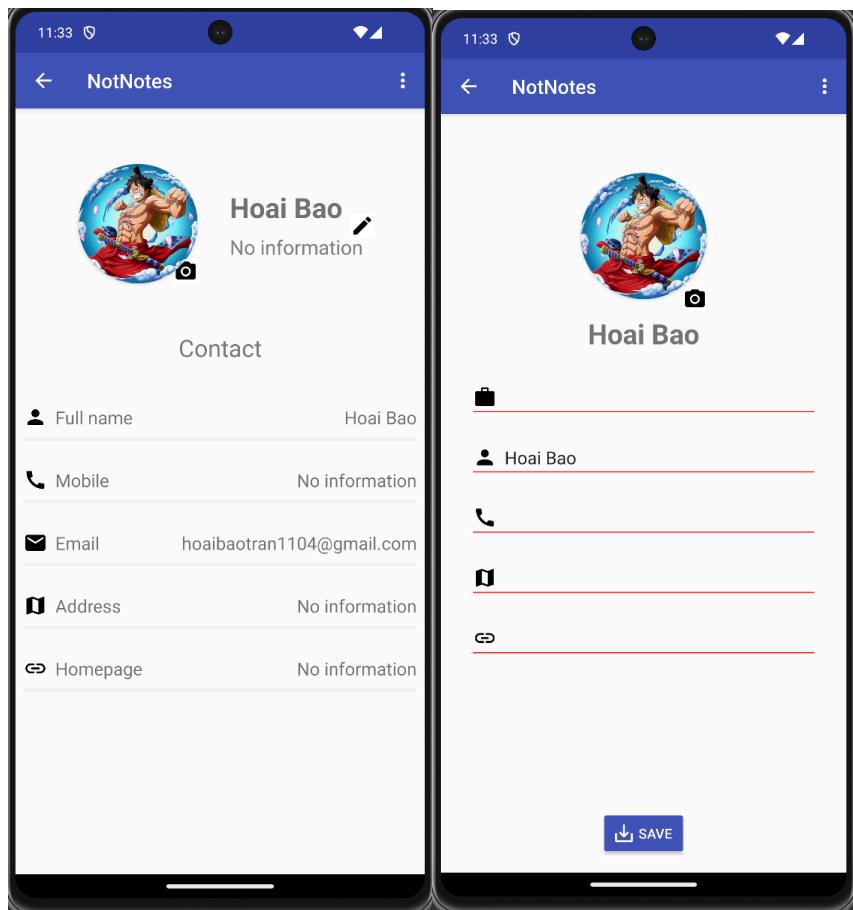


Hình 4. 11 - Màn hình xóa ghi chú thực tế của ứng dụng

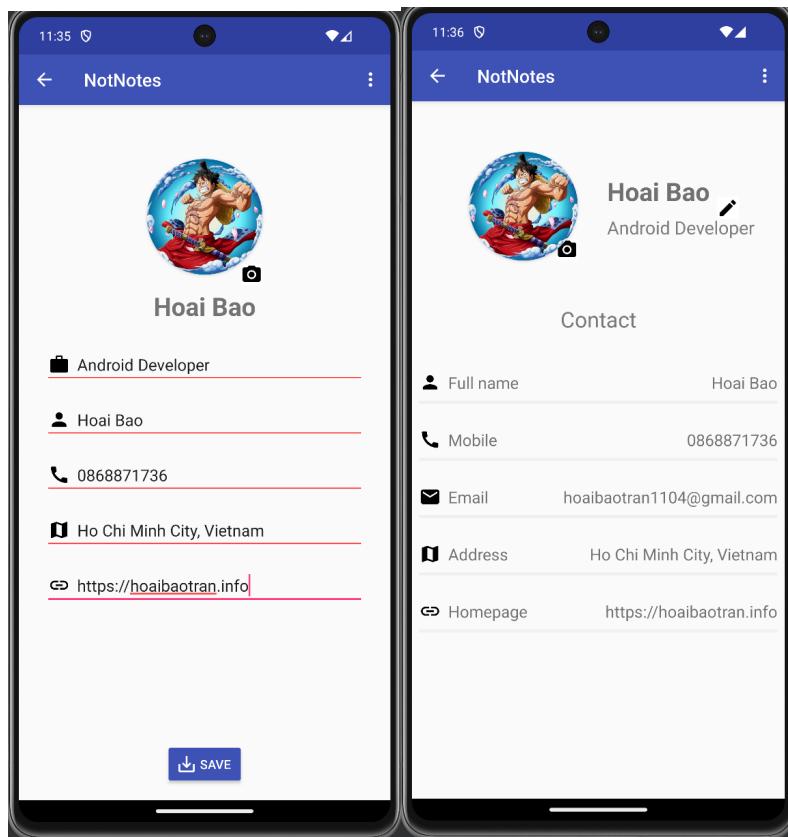


Hình 4. 12 - Firebase Database Realtime sau khi xóa ghi chú

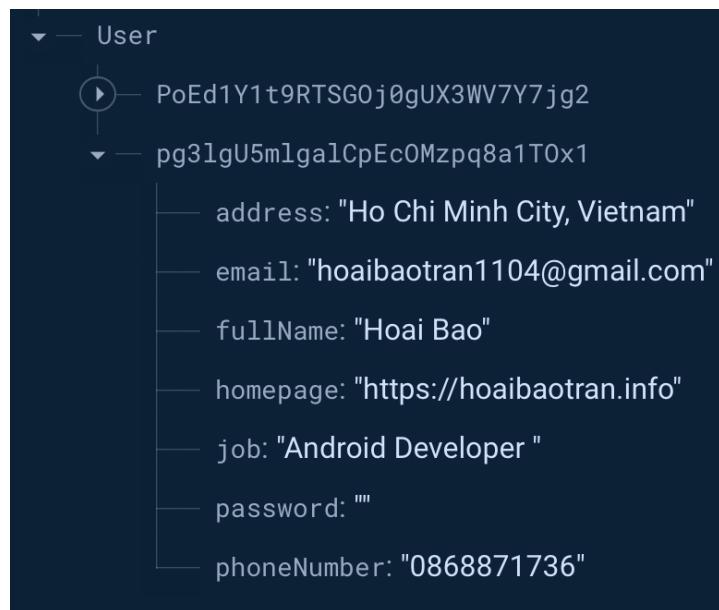
#### 4.8 Màn hình cập nhật thông tin



Hình 4. 13 - Màn hình thông tin và chỉnh sửa thông tin thực tế của ứng dụng

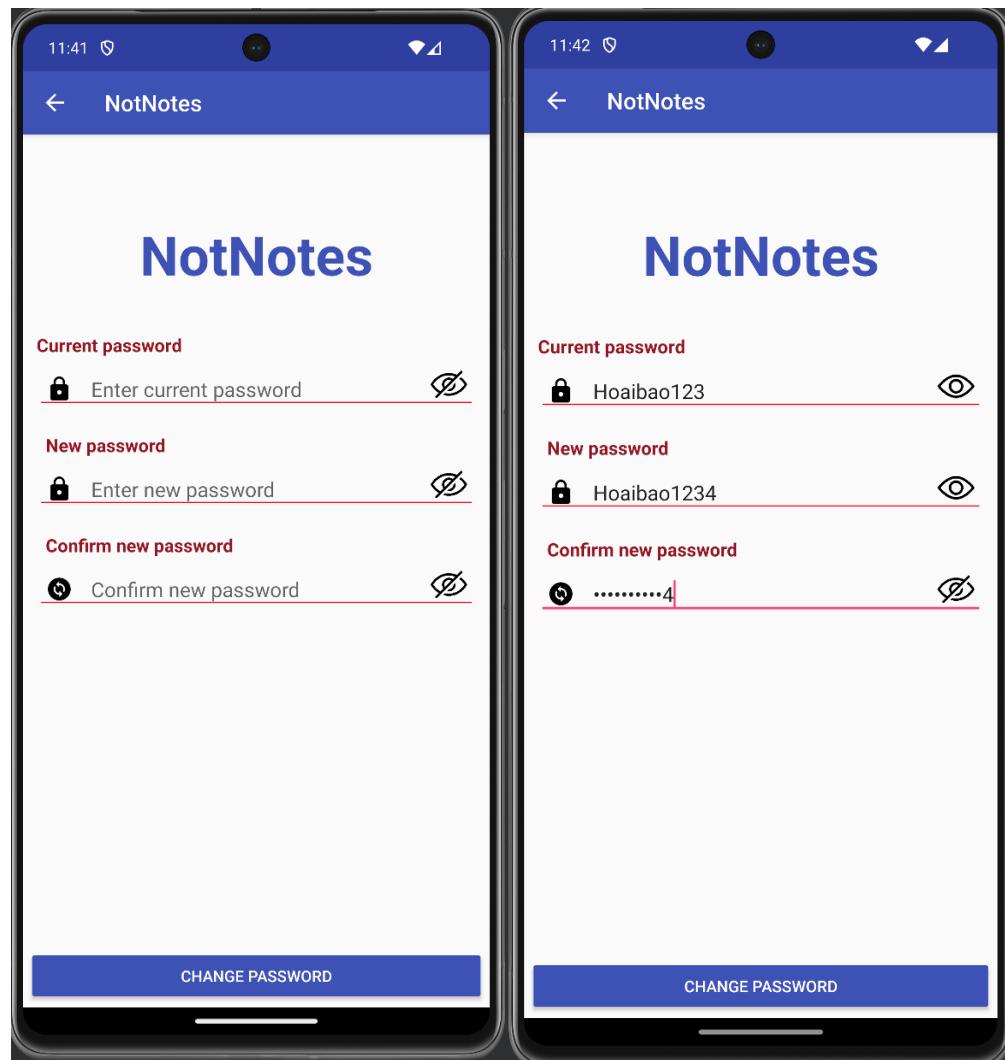


Hình 4. 14 - Màn hình thông tin cá nhân sau khi cập nhật thông tin

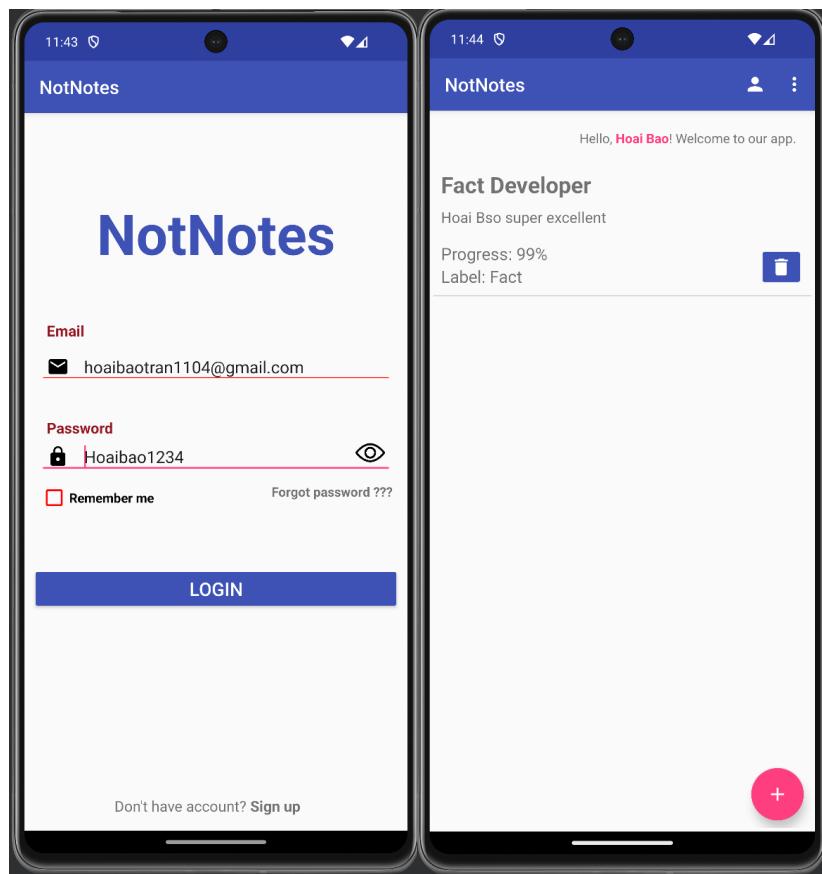


Hình 4. 15 - Firebase Realtime Database sau khi cập nhật thông tin người dùng

#### 4.9 Màn hình đổi mật khẩu

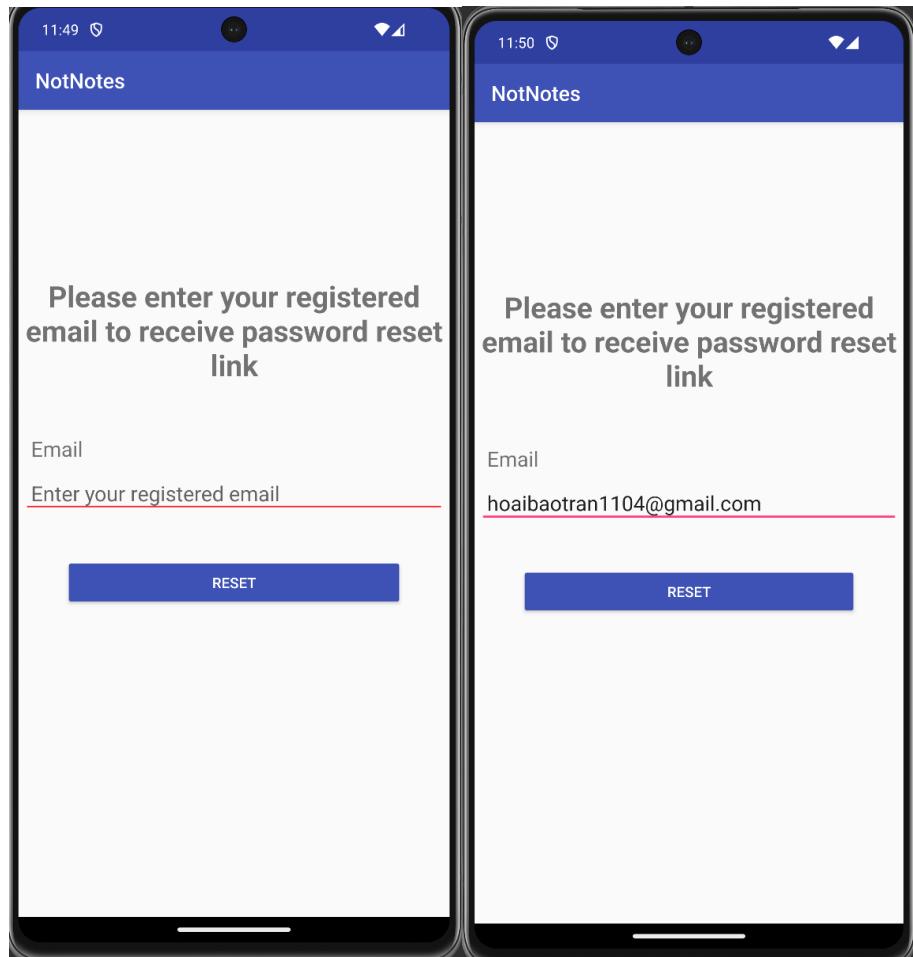


Hình 4. 16 - Màn hình đổi mật khẩu thực tế của ứng dụng

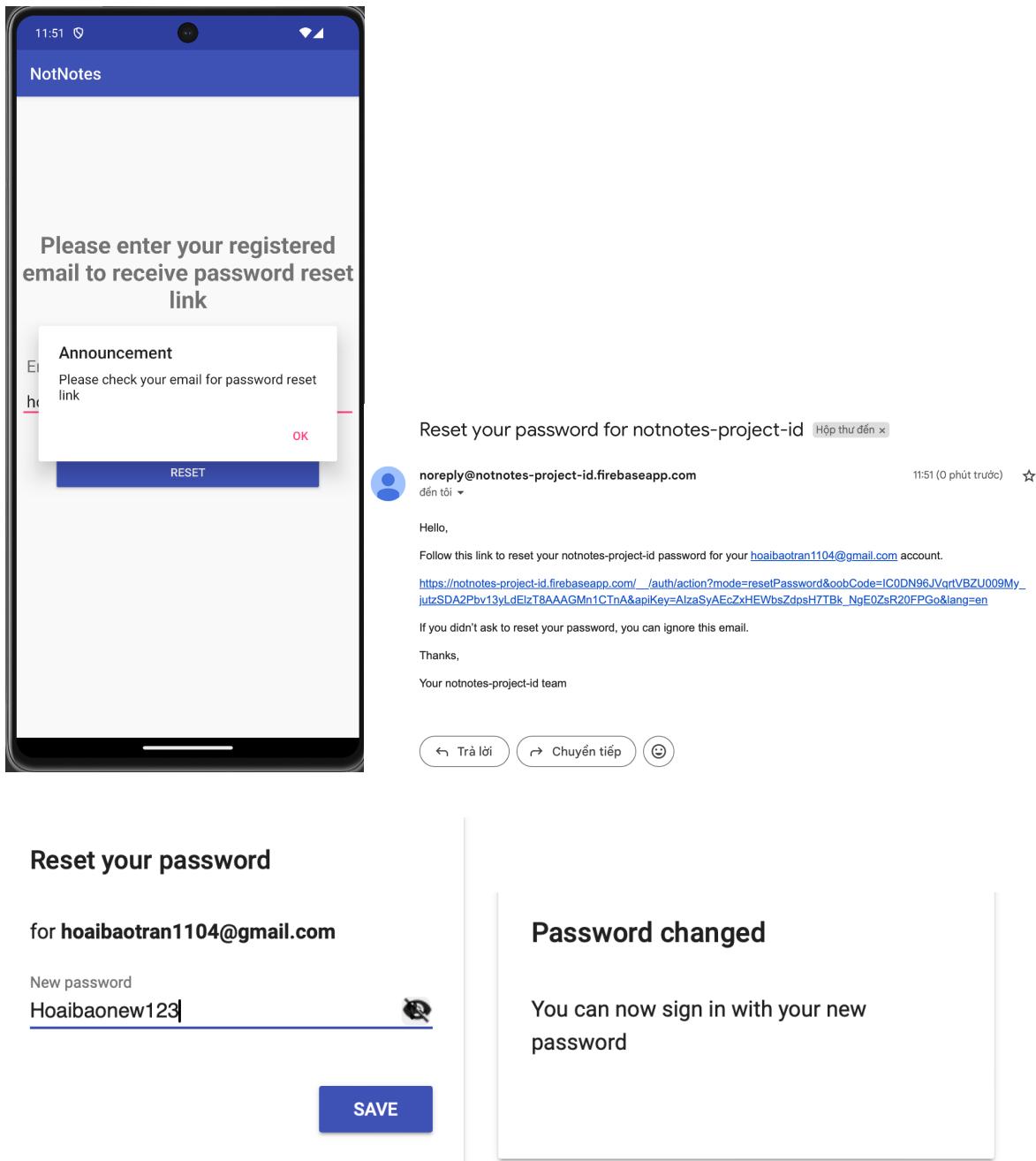


Hình 4. 17 - Thủ đăng nhập sau khi đổi mật khẩu

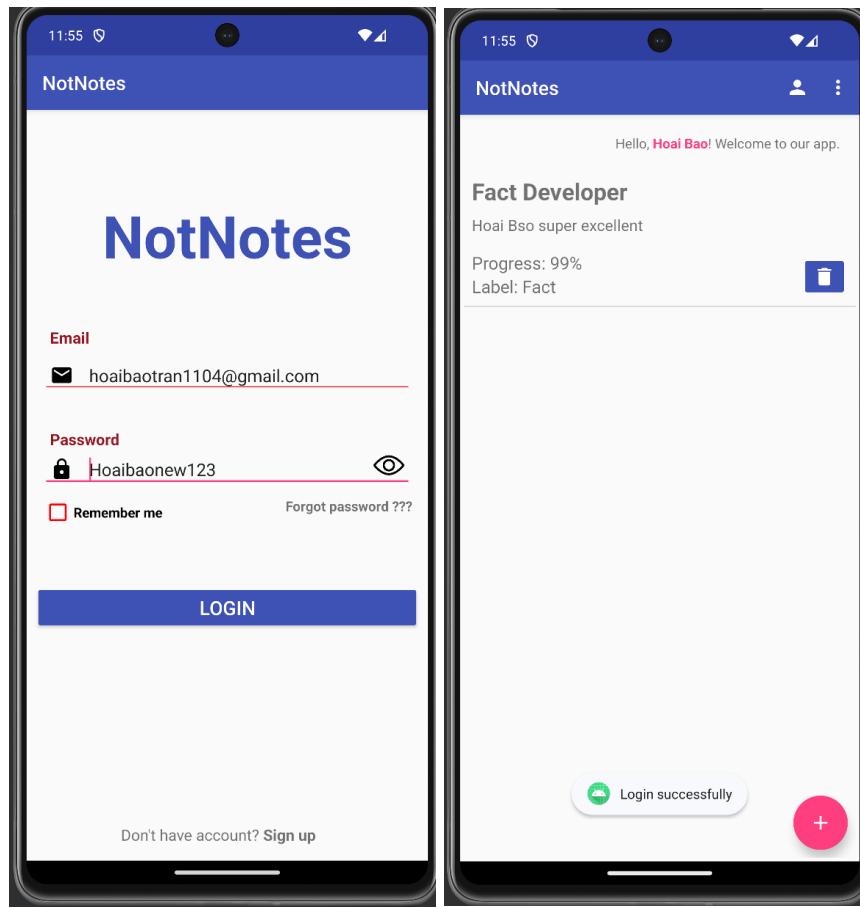
#### 4.10 Màn hình quên mật khẩu



Hình 4. 18 - Màn hình quên mật khẩu thực tế của ứng dụng



Hình 4. 19 - Tạo mật khẩu mới cho tài khoản sau khi xác thực



Hình 4. 20 - Thủ đăng nhập sau khi reset mật khẩu

## **TÀI LIỆU THAM KHẢO**

### **Tiếng Việt**

1. Hệ điều hành Android là gì? Những điều cần biết về Android, Thegioididong
2. Tổng quan về Kotlin và so sánh giữa Kotlin với Java, Viblo
3. Kotlin là gì? Ưu điểm nổi bật của ngôn ngữ lập trình Kotlin, ITNavi
4. Ngôn ngữ lập trình Kotlin là gì, Ưu nhược điểm của Kotlin, Codegym
5. Firebase là gì? Tìm hiểu tổng quan và hướng dẫn sử dụng Firebase, FPT
6. Tìm hiểu sơ lược về Firebase, Viblo
7. Firebase là gì? Ưu nhược điểm và các dịch vụ của Firebase, FPT Cloud

### **Tiếng Anh**

8. What is Android, Android, Anrdoid Developers
9. Android Overview, Tutorialspoint
10. Kotlin for Developers, Android, Android Developers
11. Kotlin Programming Language, Kotlin, kotlinlang.org
12. Kotlin Tutorial, W3S
13. Firebase for Apps | Build and grow apps users love, Firebase
14. Firebase | Google's Mobile and Web App Development Platform, Firebase
15. Firebase Console, Firebase
16. Login and Registration using Firebase in Android, Codes Easy
17. Login and Register Android App using Firebase Authentication with Profile Management and Realtime Database | Using Java | Android Studio, TanzTalksTech