TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

Bộ môn Mạng & Truyền Thông

Tel. (84-511) 3736949, Fax. (84-511) 3842771

Website: itf.dut.udn.vn, E-mail: cntt@dut.udn.vn



BÁO CÁO

ĐỒ ÁN 4: Hệ điều hành & Mạng máy tính

**ĐỀ TÀI:**

**Xây dựng ứng dụng BKZalo cho phép người dùng trò chuyện, nhắn tin,gửi hình ảnh**

SINH VIÊN : Nguyễn Trung Đức 102200206

Mai Anh Quân 102200227

LỚP : 20TCLC\_DT5

CBHD : PGS.TS. Nguyễn Tấn Khôi

Đà Nẵng, 12/2022

MỤC LỤC

[CHƯƠNG 1: Cơ sở lý thuyết 3](#_Toc27847085)

[1.1. BỘ GIAO THỨC TCP/IP 3](#_Toc27847086)

[1.1.1. Khái niệm bộ giao thức TCP/IP 3](#_Toc27847087)

[1.1.2. Tầng liên kết 3](#_Toc27847088)

[1.1.3. Tầng mạng 4](#_Toc27847089)

[1.1.4. Tầng giao vận 4](#_Toc27847090)

[1.1.5. Tầng ứng dụng 4](#_Toc27847091)

[1.2. GIAO THỨC TCP 5](#_Toc27847092)

[1.2.1. Khái niệm giao thức TCP 5](#_Toc27847093)

[1.2.2. Chu trình làm việc của TCP 5](#_Toc27847094)

[1.3. GIAO THỨC HTTP 6](#_Toc27847095)

[1.3.1. Khái niệm giao thức HTTP 6](#_Toc27847096)

[1.3.2. Kết nối HTTP 6](#_Toc27847097)

[1.3.3. Các thành phần chính 7](#_Toc27847098)

[1.4. GIỚI THIỆU VỀ GOOGLE FIREBASE 8](#_Toc27847099)

[1.4.1. Giới thiệu 8](#_Toc27847100)

[1.4.2. Cơ sở dữ liệu NoSQL của Firebase 9](#_Toc27847101)

[1.5. GIỚI THIỆU VỀ ANDROID 10](#_Toc27847102)

[1.5.1. Hệ điều hành Android 10](#_Toc27847103)

[1.5.2. Lập trình phần mềm bằng Android Studio 10](#_Toc27847104)

[1.5.3. Lập trình ứng dụng Android 11](#_Toc27847105)

[CHƯƠNG 2: Phân tích và thiết kế hệ thống 13](#_Toc27847106)

[2.1. YÊU CẦU CHỨC NĂNG CỦA ỨNG DỤNG 13](#_Toc27847107)

[2.2. PHÂN TÍCH VÀ THIẾT KẾ CHƯƠNG TRÌNH 13](#_Toc27847108)

[2.2.1. Biểu đồ phân rã chức năng 13](#_Toc27847109)

[2.2.2. Sơ đồ thực thể quan hệ 14](#_Toc27847110)

[2.2.3. Biểu đồ ca sử dụng (Use Case) 15](#_Toc27847111)

[2.2.4. Thiết kế cơ sở dữ liệu 17](#_Toc27847112)

[2.2.5. Sử dụng API Firebase 18](#_Toc27847113)

[CHƯƠNG 3: Kết luận và hướng phát triển 27](#_Toc27847114)

[3.1. KẾT LUẬN 27](#_Toc27847115)

[3.1.1. Môi trường cài đặt 27](#_Toc27847116)

[3.1.2. Kết quả đạt được 27](#_Toc27847117)

[3.1.3. Đánh giá kết quả 37](#_Toc27847118)

[3.2. HƯỚNG PHÁT TRIỂN 38](#_Toc27847119)

**DANH SÁCH HÌNH ẢNH**

Hình 1. Các tầng trong bộ giao thức TCP/IP 3

Hình 2. Cấu trúc dữ liệu trong TCP/IP 5

Hình 3. Giao thức HTTP 6

Hình 4. Sơ đồ hoạt động giao thức HTTP 7

Hình 5. Giới thiệu Google Firebase 9

Hình 6. Vòng đời Activity trong Android 12

Hình 7. Biểu đồ phân rã chức năng 13

Hình 8. Sơ đồ thực thể quan hệ 14

Hình 9. Biểu đồ ca sử dụng cho xác thực 15

Hình 10. Biểu đồ ca sử dụng cho thiết lập thông tin cá nhân 15

Hình 11. Biểu đồ ca sử dụng cho gửi nhận tin nhắn 16

Hình 12. Mô hình cơ sở dữ liệu 16

Hình 13. Firebase Sign-up 18

Hình 14. User 19

Hình 15. Firebase Database chứa đường dẫn ảnh từ Storage 19

Hình 16. Thư mục lưu trữ file tin nhắn trong Firebase Storage 20

Hình 17. Tin nhắn giữa các user lưu trong Firebase Database 21

Hình 18. Thông báo tin nhắn 21

Hình 19. Giao diện đăng ký đăng nhập 22

Hình 20. Giao diện màn hình chính 23

Hình 21. Giao diện bạn bè 24

Hình 22. Giao diện trò chuyện nhắn tin - 1 25

Hình 23. Giao diện trò chuyện nhắn tin – 2 26

Hình 24: Chỉnh sửa thông tin cá nhân. 27

DANH SÁCH TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| IP | Internet Protocol |
| ICMP | Internet Control Message Protocol |
| ARP | Address Resolution Protocol |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| FTP | File Transfer Protocol |
| SSH | Secure Shell |
| SYN | The Synchronous Idle Character |
| ACK | Acknowledgement |
| HTTP | Hypertext Transfer Protocol |

MỞ ĐẦU

**1.**  **Tổng quan về đề tài**

Đồ án các môn học chính của Khoa Công nghệ thông tin nói chung và môn học Hệ điều hành và Mạng máy tính nói riêng nhằm tạo điều kiện cơ bản giúp cho sinh viên:

* *Đi sâu và nắm vững một cách có hệ thống kiến thức đã thu nhận được trong quá trình học tập bằng việc nghiên cứu, làm việc thực tế.*
* *Từng bước làm quen với các công tác khoa học có định hướng của giáo viên hướng dẫn và hình thành hành vi nghiên cứu độc lập có sự trợ giúp của tài liệu tham khảo.*
* *Nắm bắt và nghiên cứu các kỹ thuật, công nghệ mới để phục vụ cho công việc và học tập*.

Dưới sự hướng dẫn của thầy Nguyễn Tấn Khôi, giảng viên khoa Công nghệ thông tin, Trường Đại học Bách khoa – Đại học Đà Nẵng, thông qua đồ án môn học, em đã nghiên cứu, tìm hiểu và thực hiện đề tài “Xây dựng ứng dụng BKZalo cho phép người dùng trò chuyện, nhắn tin.”

Vì còn sự hạn chế về kiến thức và thời gian nên trong thời gian nghiên cứu, thực hiện dự án này sẽ không thể tránh khỏi những sai sót. Xin cảm ơn sự hỗ trợ, hướng dẫn nhiệt tình từ thầy Nguyễn Tấn Khôi và rất mong nhận được sự góp ý từ quý thầy cô.

**2. Mục đích và ý nghĩa đề tài**

Ngày nay, công nghệ thông tin ngày càng phát triển thì nhu cầu cuộc sống cũng như việc ứng dụng Công nghệ thông tin vào phục vụ con người cũng được phát triển mạnh mẽ và ngày càng đa dạng hơn, đặc biệt là vấn đề giao tiếp, liên lạc một cách thuận tiện và nhanh chóng. Cùng với sự phát triển mạnh mẽ của các thiết bị di động thông minh nói chung và điện thoại thông minh nói riêng thì việc áp dụng các kỹ thuật, phát triển các ứng dụng trên nền tảng này là hết sức cần thiết. Một ứng dụng đơn giản, có thể sử dụng trên các loại điện thoại di động thông minh phổ biến hiện nay, cùng với các tính năng trao đổi thông tin thời gian thực có thể giúp giải quyết vấn đề đặt ra. Vì vậy, ý tưởng xây dựng ứng dụng BKZalo cho phép người dùng trò chuyện, nhắn tin trên nền tảng thiết bị Android phổ biển hiện nay được ra đời.

**3. Bố cục của đồ án**

Đồ án bao gồm các nội dung sau:

*Chương 1: trình bày cơ sở lý thuyết*

*Chương 2: trình bày quá trình phân tích và thiết kế hệ thống, thiết kế chương trình*

*Chương 3: trình bày kết quả thực thi chương trình, đánh giá kết quả và hướng phát triển*

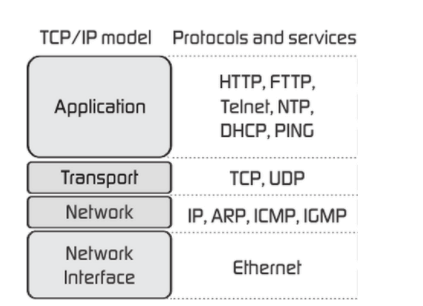
# Cơ sở lý thuyết

## BỘ GIAO THỨC TCP/IP

### Khái niệm bộ giao thức TCP/IP

TCP/IP là một tập hợp các giao thức (protocol) điều khiển truyền thông giữa tất cả các máy tính trên Internet. Cụ thể hơn, TCP/IP chỉ rõ cách thức đóng gói thông tin (hay còn gọi là gói tin), được gửi và nhận bởi các máy tính có kết nối với nhau. TCP/IP được phát triển vào năm 1987 bởi Bob Kahn và Vint Cerf.

Giống như nhiều bộ giao thức khác TCP/IP cũng tập hợp các dữ liệu được phân cấp theo từng tầng khác nhau trong đấy mỗi tầng đóng vai trò khác nhau, chúng xử lý dữ liệu thô từ những tầng dưới cùng sau đó vận chuyển lên trên để các giao thức khác dễ làm việc hơn.



1. Các tầng trong bộ giao thức TCP/IP

### Tầng liên kết

Tầng liên kết (còn được gọi là tầng liên kết dữ liệu hay là tầng giao tiếp mạng) là tầng thấp nhất trong mô hình TCP/IP, bao gồm các thiết bị giao tiếp mạng và chương trình cung cấp các thông tin cần thiết để có thể hoạt động, truy nhập đường truyền vật lý qua thiết bị giao tiếp mạng đó.

### Tầng mạng

Giống như tầng mạng của mô hình OSI. Tại đây, nó cũng được định nghĩa là một giao thức chịu trách nhiệm truyền tải dữ liệu một cách logic trong mạng. Các phân đoạn dữ liệu sẽ được đóng gói (Packets) với kích thước mỗi gói phù hợp với mạng chuyển mạch mà nó dùng để truyền dữ liệu. Lúc này, các gói tin được chèn thêm phần Header chứa thông tin của tầng mạng và tiếp tục được chuyển đến tầng tiếp theo. Các giao thức chính trong tầng là IP, ICMP và ARP.

### Tầng giao vận

Tầng giao vận phụ trách luồng dữ liệu giữa hai trạm thực hiện các ứng dụng của tầng trên. Tại đây, dữ liệu sẽ được phân đoạn, mỗi đoạn sẽ không bằng nhau nhưng kích thước phải nhỏ hơn 64KB. Cấu trúc đầy đủ của một Segment lúc này là Header chứa thông tin điều khiển và sau đó là dữ liệu.

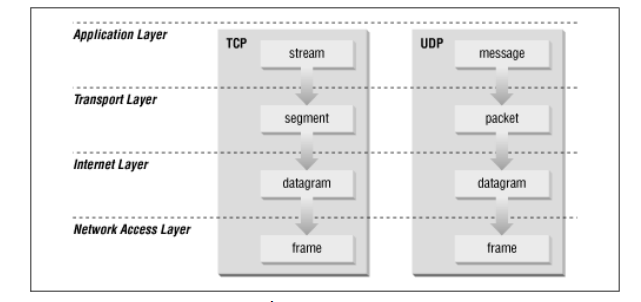
Tầng này có hai giao thức chính: TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).

TCP cung cấp một luồng dữ liệu tin cậy giữa hai trạm, nó sử dụng các cơ chế như chia nhỏ các gói tin của tầng trên thành các gói tin có kích thước thích hợp cho tầng mạng bên dưới, báo nhận gói tin, đặt hạn chế thời gian time-out để đảm bảo bên nhận biết được các gói tin đã gửi đi. Do tầng này đảm bảo tính tin cậy, tầng trên sẽ không cần quan tâm đến nữa.

UDP cung cấp một dịch vụ đơn giản hơn cho tầng ứng dụng. Nó chỉ gửi các gói dữ liệu từ trạm này tới trạm kia mà không đảm bảo các gói tin đến được tới đích. Các cơ chế đảm bảo độ tin cậy cần được thực hiện bởi tầng trên.

### Tầng ứng dụng

Đây là lớp trên cùng của mô hình. Đúng với tên gọi, tầng ứng dụng đảm nhận vai trò giao tiếp dữ liệu giữa hai máy khác nhau thông qua các dịch vụ mạng khác nhau (duyệt web, chat, gửi email, một số giao thức trao đổi dữ liệu: SSH, FTP….). Dữ liệu khi đến đây sẽ được định dạng theo kiểu Byte nối Byte, cùng với đó là các thông tin định tuyến giúp xác định đường đi đúng của một gói tin.



1. Cấu trúc dữ liệu trong TCP/IP

## GIAO THỨC TCP

### Khái niệm giao thức TCP

TCP là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các kết nối với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự.

### Chu trình làm việc của TCP

Chu trình làm việc của giao thức TCP gồm ba bước:

#### Thiết lập liên kết (quy trình bắt tay ba bước xảy ra)

* Client yêu cầu mở cổng dịch vụ bằng cách gởi gói tin SYN tới Server, trong gói tin này, tham số sequence number được gán cho một giá trị ngẫu nhiên x.
* Server hồi đáp bằng cách gởi lại phía Client bản tin SYN-ACK, trong gói tin này, tham số acknowledgment number được gán bằng giá trị x + 1, tham số sequence number được gán ngẫu nhiên một giá trị y.
* Để hoàn tất quy trình bắt tay ba bước, Client tiếp tục gửi tới Server bản tin ACK, trong bản tin này, tham số sequence number được gán giá trị bằng x+1 còn tham số acknowledgment number được gán bằng giá trị y + 1.

#### Truyền nhận dữ liệu

#### Đóng liên kết

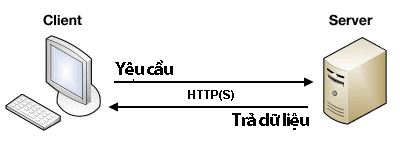
Để kết thúc kết nối hai bên sử dụng quá trình bắt tay 4 bước và chiều của kết nối kết thúc độc lập với nhau. Khi một bên muốn kết thúc, nó gửi đi một gói tin FIN và bên kia gửi lại tin báo nhận ACK. Vì vậy, một quá trình kết thúc tiêu biểu sẽ có 2 cặp gói tin trao đổi.

Một kết nối có thể tồn tại ở dạng nửa mở: một bên đã kết thúc gửi dữ liệu nên chỉ nhận thông tin, bên kia vẫn tiếp tục gửi.

## GIAO THỨC HTTP

### Khái niệm giao thức HTTP

Hypertext Transfer Protocol (HTTP) là giao thức thuộc lớp ứng dụng web. Sử dụng kết nối TCP cổng 80. HTTP hoạt động dựa trên mô hình client – server. Trình duyệt thực hiện yêu cầu, nhận và hiển thị đối tượng web ( gồm dữ liệu HTML, hình ảnh, video, âm thanh,…). Trong khi web server sẽ gửi trả lời khi nhận được yêu cầu từ client.



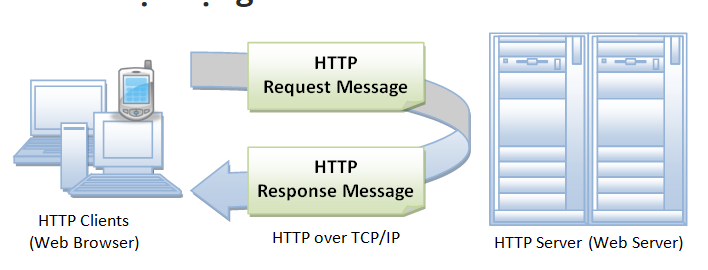
1. Giao thức HTTP

### Kết nối HTTP

Có hai loại kết nối HTTP là kết nối bền vững và kết nối không bền vững.

Kết nối không bền vững: sau khi server gửi đi một đối tượng thì kết nối TCP sẽ được đóng. Như vậy, mỗi kết nối TCP chỉ truyền được duy nhất một yêu cầu từ client và nhận lại một thông điệp trả lời từ server.

Kết nối bền vững: server sẽ duy trì kết nối TCP cho việc gửi nhiều đối tượng. Như vậy, sẽ có nhiều yêu cầu từ client được gửi đến server trên cùng một kết nối. Thông thường kết nối TCP này sẽ được đóng lại trong khoảng thời gian định trước.



1. Sơ đồ hoạt động giao thức HTTP

### Các thành phần chính

**HTTP Request Method:** Là phương thức để chỉ ra hành động mong muốn được thực hiện trên tài nguyên đã xác định.

Một HTTP Request gồm có: phương thức + URI-Request + Phiên bản HTTP. Giao thức HTTP định nghĩa một tập các giao thức GET, POST, PUT,… Client có thể sử dụng một trong các phương thức đó để gửi request lên server

Một số HTTP Request method thường dùng:

- GET: được sử dụng để lấy lại thông tin từ Server một tài nguyên xác định.

- POST: yêu cầu máy chủ chấp nhận thực thể được đính kèm trong request được xác định bởi URI.

- PUT: nếu URI đề cập đến một tài nguyên đã có, nó sẽ bị sửa đổi; nếu URI không trỏ đến một tài nguyên hiện có, thì máy chủ có thể tạo ra tài nguyên với URI đó.

- PATCH: áp dụng cho việc sửa đổi một thành phần của tài nguyên được xác định.

- DELETE: xóa bỏ tất cả các đại diện của tài nguyên được chỉ định bởi URI.

**HTTP Responses:** là tín hiệu phản hồi từ Server cho Client khi nhận được 1 HTTP Request từ Client.

Một HTTP Responses gồm: Phiên bản HTTP + Mã trạng thái + Trạng thái

Trong đó:

- Mã trạng thái thông báo về kết quả khi nhận được yêu cầu xử lí bên server cho client.

- Các kiểu trạng thái:

+ 1xx: Thông tin (100 -101). Ví dụ: 100 (Continue)…

+ 2xx: Thành công (200 -> 206). Ví dụ: 200 (OK), 201 (CREATED),…

+ 3xx: Sự điều hướng lại (300 -> 307). Ví dụ: 305 (USE PROXY),…

+ 4xx: Lỗi phía Client (400 -> 417). Ví dụ: 404 (NOT FOUND),…

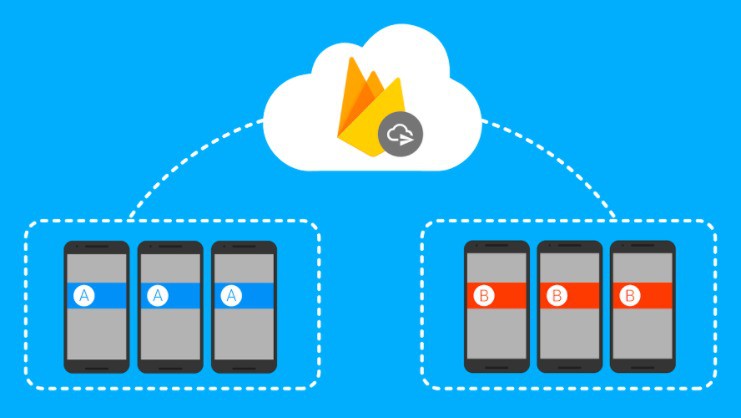
+ 5xx: Lỗi phía Server (500 -> 505). Ví dụ: 500 (INTERNAL SERVER ERROR),…

## GIỚI THIỆU VỀ GOOGLE FIREBASE

### Giới thiệu

Firebase Realtime Database là một cơ sở dữ liệu lưu trữ đám mây hỗ trợ đa nền tảng: Android, IOS, và Web. Các dữ liệu được lưu trữ trong hệ thống cơ sở dữ liệu hỗ trợ NoSQL và được đặt trên nền tảng máy chủ Cloud, dữ liệu được ghi và đọc với thời gian thấp nhất tính bằng mili giây. Firebase sử dụng phương thức HTTPS với cổng 443 và REST API để thực hiện giao tiếp. Nền tảng này hỗ trợ đồng bộ háo dữ liệu của người dùng kể cả khi không có kết nối mạng, tạo nên sự trải nghiệm xuyên suốt bất chấp tình trạng kết nối internet của người sử dụng.

Firebase Auth cung cấp nhiều phương pháp để xác thực, bao gồm email và mật khẩu, các nhà cung cấp bên thứ ba như Google…



1. Giới thiệu Google Firebase

### Cơ sở dữ liệu NoSQL của Firebase

Cơ sở dữ liệu NoSQL là cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn. Các cơ sở dữ liệu này sử dụng nhiều mô hình dữ liệu đa dạng, trong đó có văn bản, đồ thị, khóa – giá trị, trong bộ nhớ và tìm kiếm. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng lớn và có độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.

Cơ sở dữ liệu NoSQL là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ liệu cực ỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao để đem đến cho người dùng trải nghiệm tuyệt vời như:

- Linh hoạt: Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.

- Khả năng thay đổi quy mô: Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém. Một số nhà cung cấp dịch vụ đám mây xử lý các hoạt động này một cách không công khai dưới dạng dịch vụ được quản lý đầy đủ.

- Hiệu năng cao: Cơ sở dữ liệu NoSQL được tối ưu hóa theo mô hình dữ liệu ( ví dụ như văn bản, khóa – giá trị và đồ thị) và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.

- Cực kỳ thiết thực: Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.4

## GIỚI THIỆU VỀ ANDROID

### Hệ điều hành Android

Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành riêng cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, hệ điều hành này được phát triển bởi công ty Android, với sự hỗ trợ tài chính từ Google, sau đó chính Google đã mua lại công ty này và tiếp tục phát triển Android trở thành một nền tảng hiệu quả hơn.

### Lập trình phần mềm bằng Android Studio

Android Studio là một phần mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet,.. Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống biuld/deploy cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp. Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc lapop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiết bị mobile (hoặc tablet) để tiến hành cài đặt ứng dụng và chạy kiểm thử. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.

### Lập trình ứng dụng Android

Activities: Thông thường trong một ứng dụng sẽ có một hoặc nhiều Activity. Mỗi một Activity này sẽ có một vòng đời riêng độc lập hoàn toàn với các Activity khác. Việc hiểu rõ vòng đời của Activity là rất quan trọng trong việc xử lý thông tin.

**Vòng đời một Activity trong Android (Life Cycle States):**

- active (running): Activity đang hiển thị trên màn hình. Activity này tập trung vào những thao tác của người dùng trên ứng dụng.

- paused: Activity được tạm dừng, tuy nhiên vẫn hiển thị nhưng không thể tương tác. Có nghĩa là một Activity mới ở trên nó, nhưng không bao phủ đầy màn hình. Một Activity tạm dừng là còn sống, nhưng có thể bị kết thúc bởi hệ thống trong trường hợp thiếu vùng nhớ.

- stop: Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop. Nó vẫn còn trạng thái và thông tin thành viên trong nó. Người dùng không thấy nó và thường bị loại bỏ khi hệ thống cần vùng nhớ cho các tác vụ khác.

- killed: Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái stop hoặc paused cũng có thể bị giải phóng và khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục hồi lại trạng thái trước đó.

**Các hàm thực thi:**

- onCreate(): hàm này được gọi khi lớp Activity được khởi tạo, dùng để thiết lập giao diện ứng dụng và thực thi những thao tác cơ bản.

- onStart(): hàm này được gọi khi lớp ứng dụng xuất hiện trên màn hình.

- onResume(): hàm này được gọi ngay sau onStart hoặc khi người dùng focus ứng dụng, hàm này sẽ đưa ứng dụng lên top màn hình.

- onStop(): hàm được gọi khi một Activity khác được khởi động và focus.

- onRestart(): được gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().



1. Vòng đời Activity trong Android

# Phân tích và thiết kế hệ thống

## YÊU CẦU CHỨC NĂNG CỦA ỨNG DỤNG

- Cho phép người dùng đăng ký tài khoản, đăng nhập và đăng xuất khỏi ứng dụng.

- Cho phép người dùng có thể chỉnh sửa thông tin cá nhân như tên sử dụng, các status, hình ảnh đại diện.

- Ứng dụng cho phép tìm kiếm bạn bè, kết bạn với những người dùng khác.

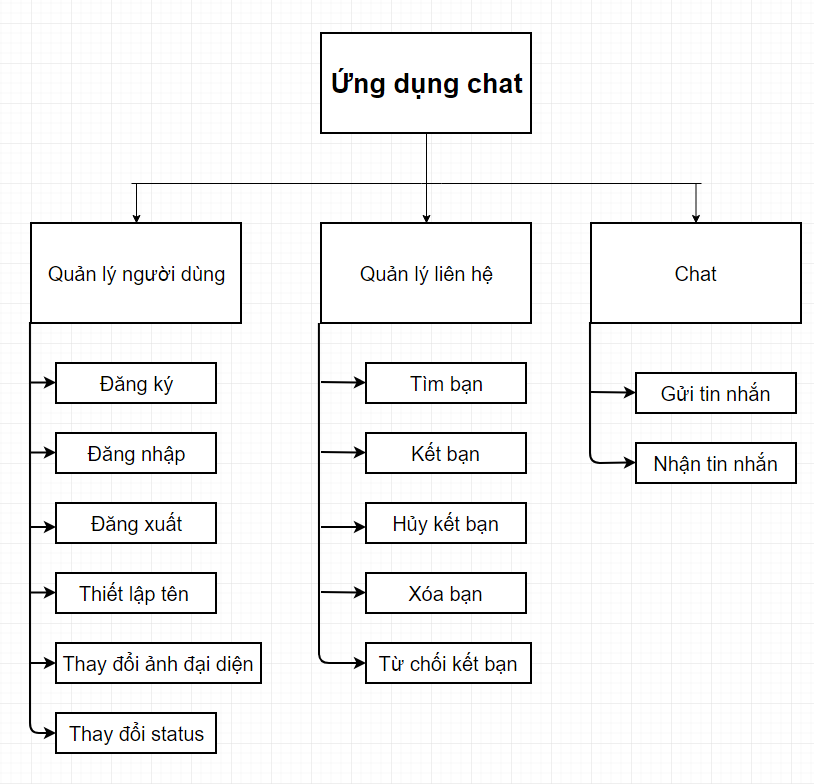
- Hiển thị danh sách các lời mời kết bạn đến từ các người dùng khác.

- Ứng dụng cho phép người dùng có thể trò chuyện, nhắn tin theo thời gian thực, cho phép người dùng trao đổi hình ảnh, các file dữ liệu cho nhau.

- Ứng dụng lưu trữ được nội dung của từng cuộc trò chuyện.

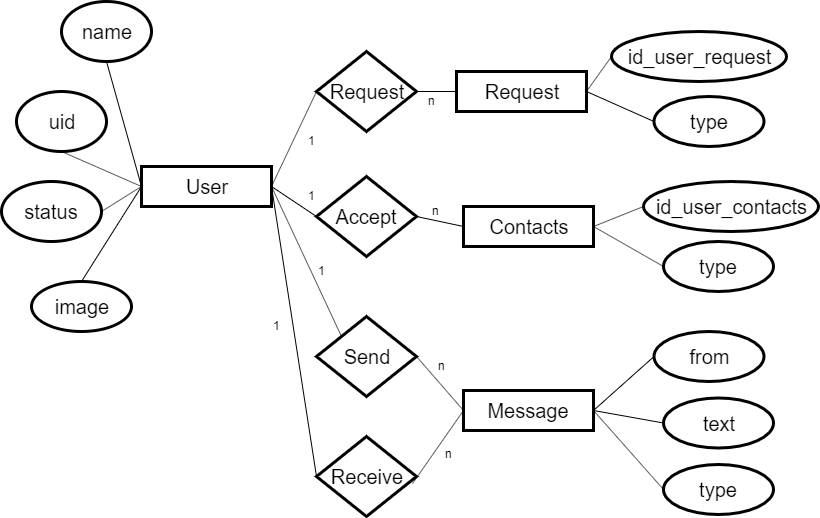
## PHÂN TÍCH VÀ THIẾT KẾ CHƯƠNG TRÌNH

### Biểu đồ phân rã chức năng

****

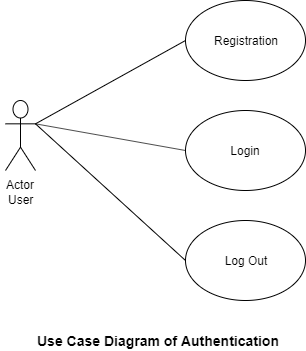
1. Biểu đồ phân rã chức năng

### Sơ đồ thực thể quan hệ

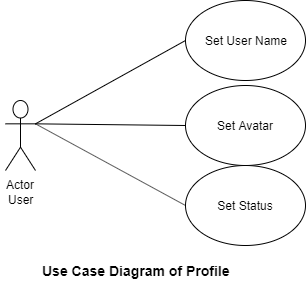
****

1. Sơ đồ thực thể quan hệ

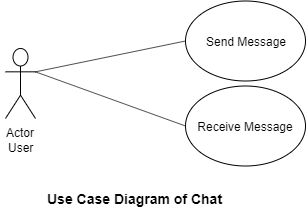
### Biểu đồ ca sử dụng (Use Case)

****

1. Biểu đồ ca sử dụng cho xác thực

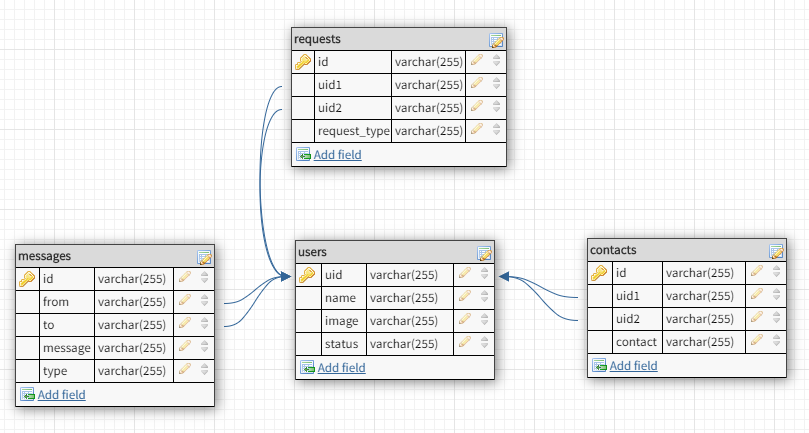
****

1. Biểu đồ ca sử dụng cho thiết lập thông tin cá nhân

****

1. Biểu đồ ca sử dụng cho gửi nhận tin nhắn

### Thiết kế cơ sở dữ liệu



1. Mô hình cơ sở dữ liệu

### Sử dụng API Firebase

Đầu tiên, để thực hiện bất kỳ xử lý nào trên database cần phải tham chiếu đến database bằng cách sử dụng:

FirebaseFirestore database = FirebaseFirestore.*getInstance*();

1. Thêm dữ liệu (C):

Để thêm dữ liệu, sử dụng phương thức add() cho node cần thêm dữ liệu.

1. Đọc dữ liệu (R)

Bất cứ khi nào cần đọc dữ liệu ta đều cần gọi phương thức datasnapshot để kiểm tra dữ liệu cần đọc có đang tồn tại trên database hay không.Để đọc dữ liệu, ta sử dụng phương thức getValue() và sử dụng một trong 3 listener là addValueEventListener, addListenerForSingleValueEvent, addChildEventListener.

Trong đó, addValueEventListener và addListenerForSingleValueEvent là hai listener giúp ứng dụng phát hiện những thay đổi của dữ liệu. Điểm khác biệt là addListenerForSingleValueEvent phát hiện những thay đổi dữ liệu một lần duy nhất. Những thay đổi của dữ liệu tiếp theo sẽ không được lăng nghe. Đối với addChildEventListener được sử dụng khi làm việc với danh sách dữ liệu (list of data).

1. Thay đổi dữ liệu (U)

Để thay đổi dữ liệu ta vẫn sử dụng phương thức setValue() như trong thao tác thêm dữ liệu với giá trị truyền vào là giá trị mới.

Xóa dữ liệu (D) Phương thức removeValue() được sử dụng để xóa dữ liệu trong Firebase Database.

**Đăng ký tài khoản:**

- Get 4 tham số nhận được từ ứng dụng là image, name,email and password

- Bắt sự kiện OnClick trên button đăng ký.

- Nếu tạo tài khoản thành công thì sẽ được lưu vào FireStore, trong database sẽ tạo nhánh con mang tên “users” ( nếu chưa từng có tài khoản nào trong hệ thống) và truyền vào dữ liệu là Id tài khoản vừa đăng ký

Text

Description automatically generated

1. Firebase Sign-up

Graphical user interface, text, application

Description automatically generated

1. User

**Đăng nhập:**

- Get 2 tham số nhận được từ ứng dụng là email và password.

- Bắt sự kiện OnClick trên button đăng nhập

- Sau đó, ta sẽ gọi với phương thức SignIn() để kiểm tra tài khoản đã có trên hệ thống hay chưa - trong đó Email và Password tương ứng với 2 tham số get được ở trên. Nếu tài khoản đã có, đăng nhập thành công vào ứng dụng.

**Thiết lập thông tin tài khoản:**

- Đối với ảnh đại diện: Sử dụng encodeImage để chọn ảnh trong thiết bị

**Text

Description automatically generated**

1. Firebase Database chứa đường dẫn ảnh từ Storage

- Đối với tên sử dụng của tài khoản:

+ Ta get 2 tham số là username và status từ Edit Text của ứng dụng

+ Ta dùng HashMap<String, Object> profile sau đó truyền các tham số sau vào: put(“uid”, currentUser), put(“name”,userName)

**Gửi tin nhắn:** Phân loại tin nhắn thành các kiểu: text, image, document (bao gồm file pdf và docx). Ta xây dựng lớp Messages với các trường: from, to, type, message, messageID.

+ from: user gửi

+ to: user nhận

+ type: loại tin nhắn (text, image, document)

+ message: chứa thông điệp, lời nhắn

+ messageID: id của tin nhắn

- Đối với các tin nhắn dạng text ta truyền thông điệp, lời nhắn vào biến message của lớp Message và gọi FirebaseDatabase để lưu trực tiếp vào hệ thống.

Graphical user interface, text, application, email

Description automatically generated

1. Thư mục lưu trữ file tin nhắn trong Firebase Storage

Graphical user interface, text, application

Description automatically generated

1. Tin nhắn giữa các user lưu trong Firebase Database

**Thông báo tin nhắn:** Sử dụng phương thức sendNotification() để thông báo tin nhắn được gửi tới user.

Text

Description automatically generated

1. Thông báo tin nhắn

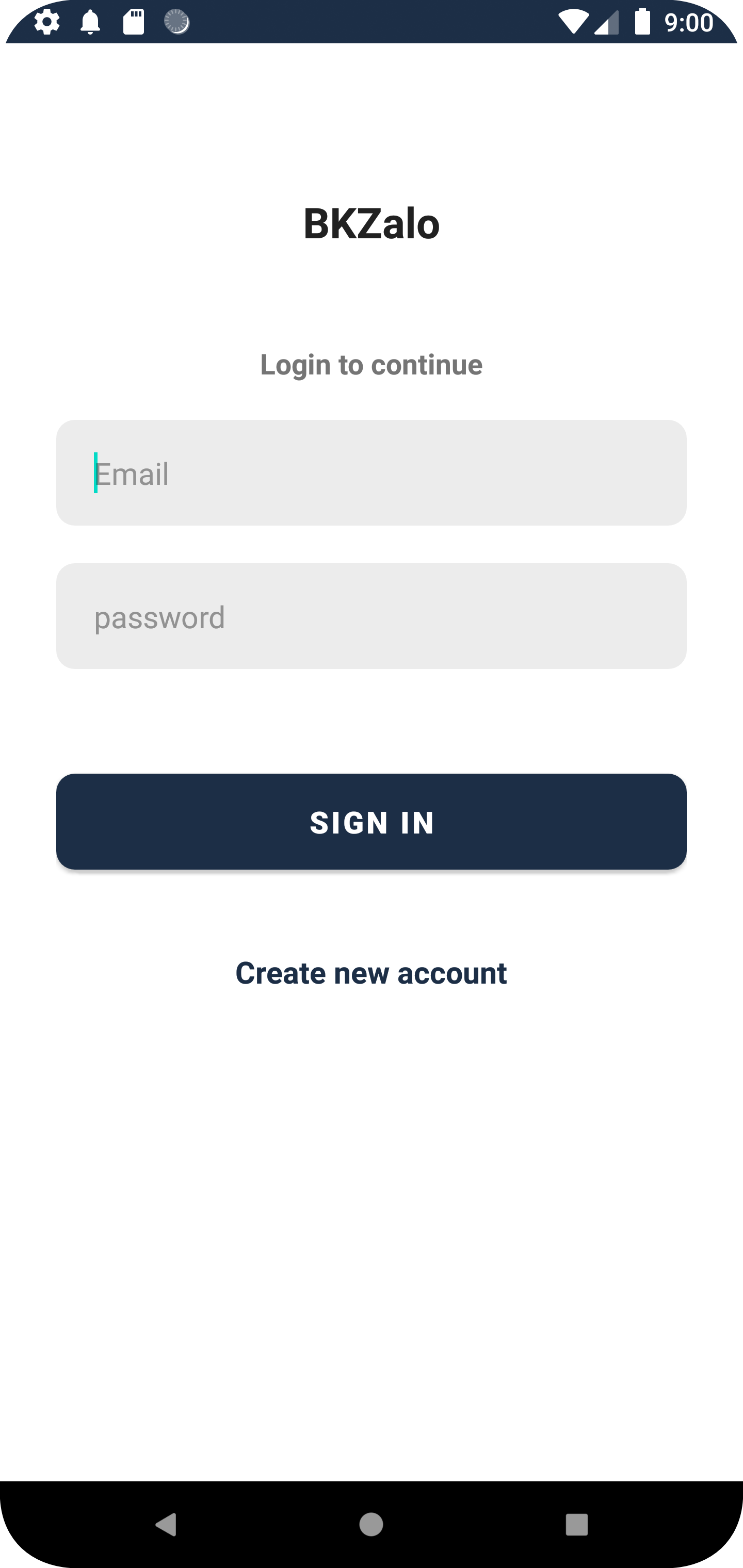
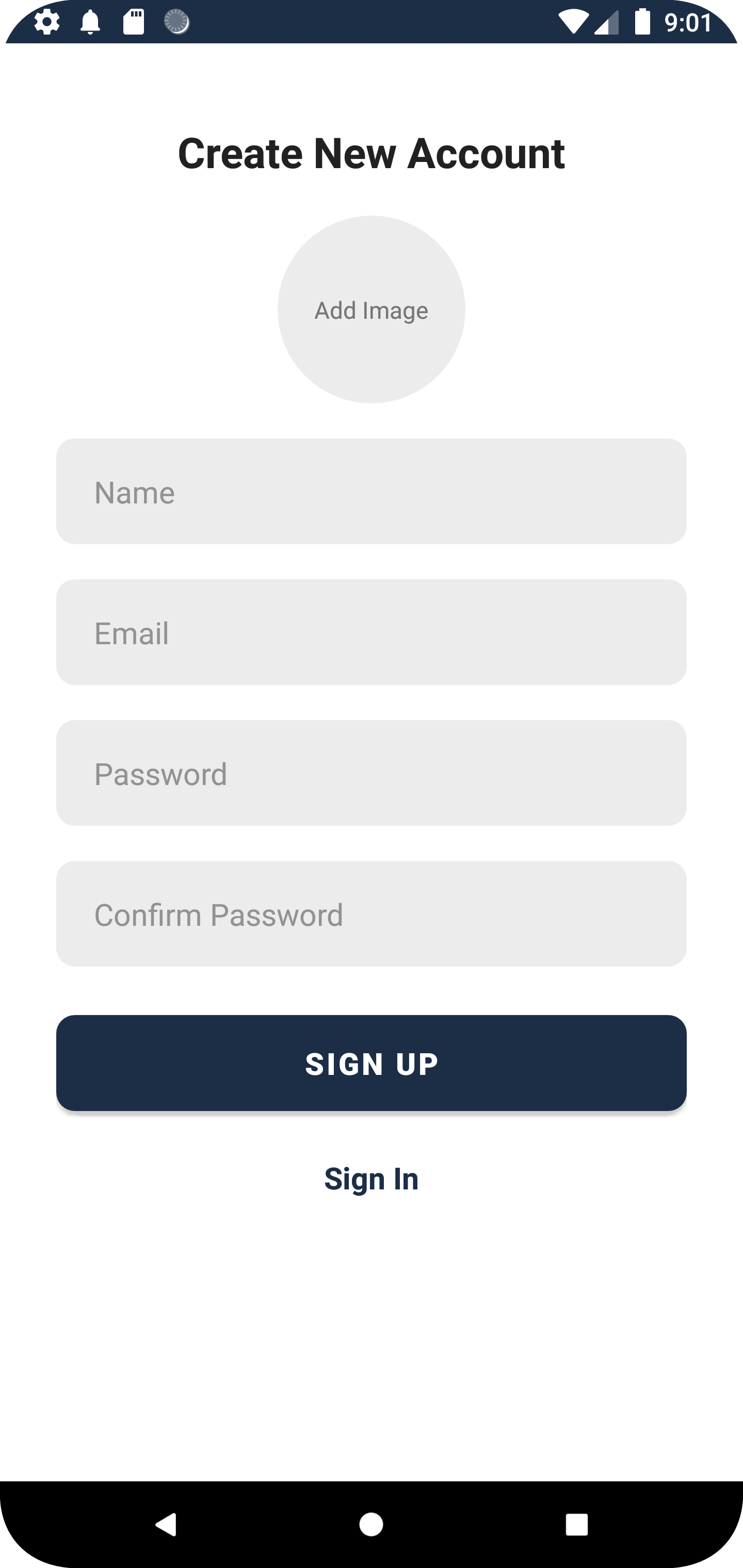
# Kết luận và hướng phát triển

## KẾT LUẬN

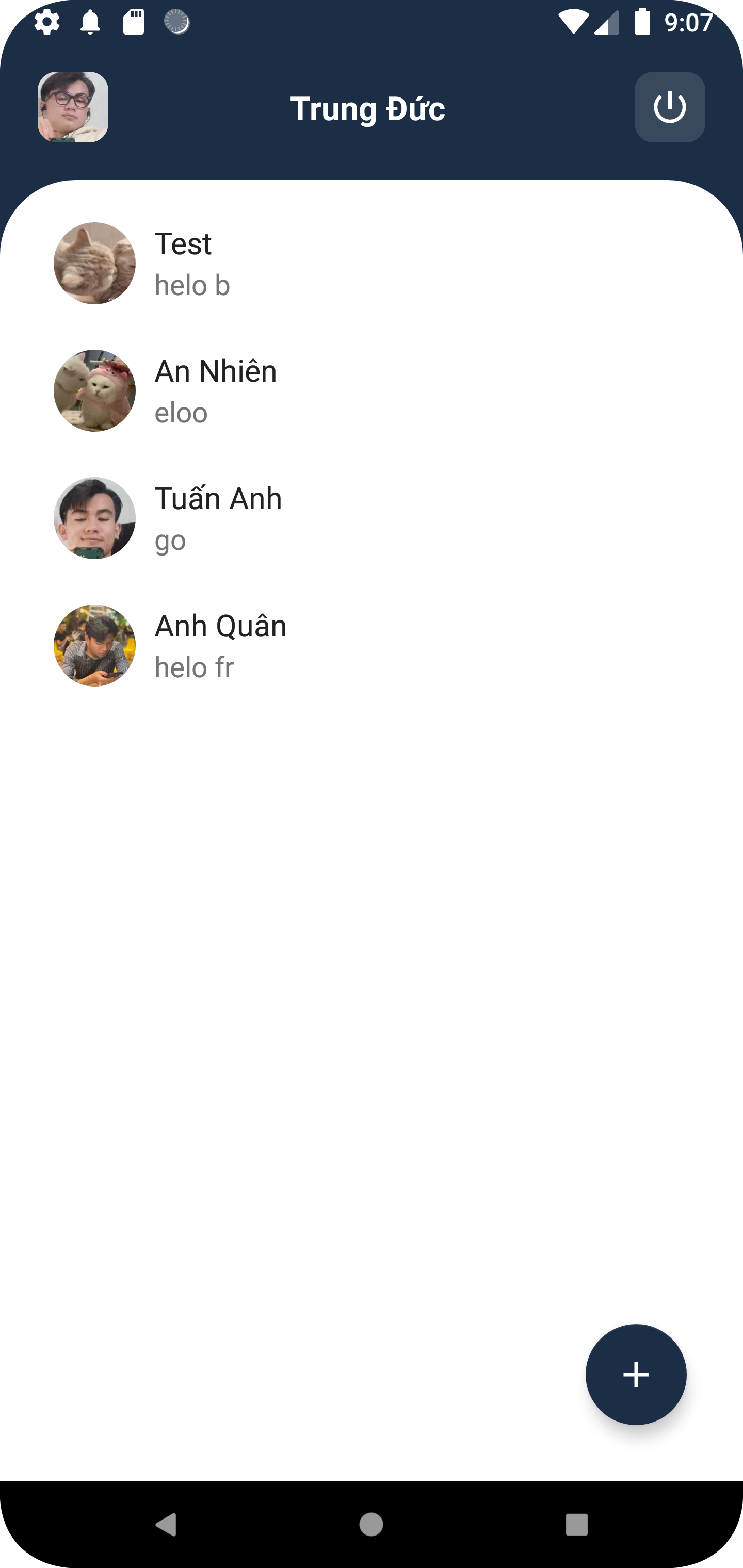
### Môi trường cài đặt

Ứng dụng BKZalo được viết bằng ngôn ngữ JAVA và XML trên nền tảng hệ điều hành Android. Các thiết bị để có thể chạy được ứng dụng cần phải có hệ điều hành Android với API level từ 19 trở lên (tương đương với phiên bản Android 4.4 trở lên).

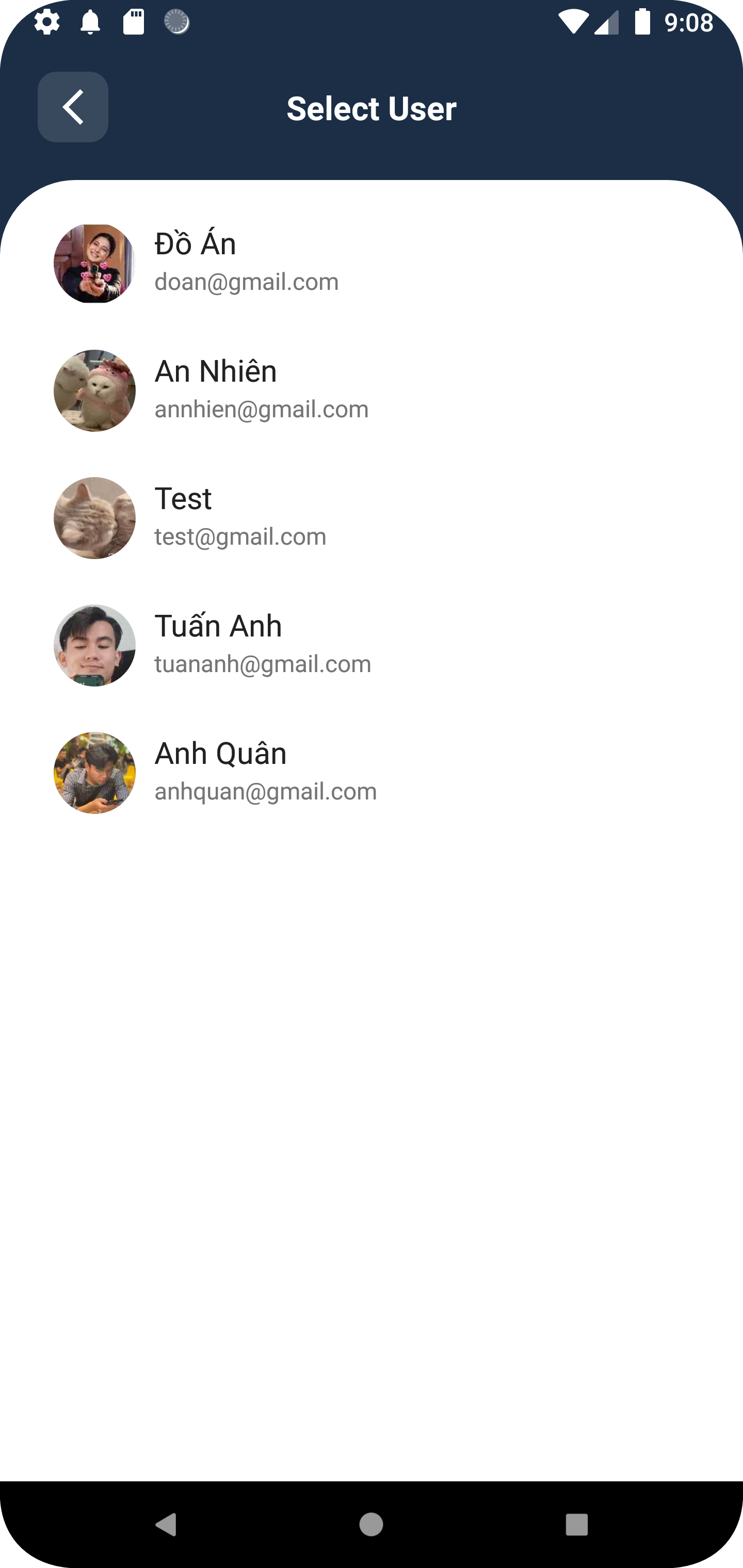
### Kết quả đạt được

** **

1. Giao diện đăng ký đăng nhập



1. Giao diện màn hình chính

****

1. Giao diện bạn bè



1. Giao diện trò chuyện nhắn tin - 1

1. Giao diện trò chuyện nhắn tin – 2

A screenshot of a cell phone

Description automatically generated with medium confidence

Hình 24: Chỉnh sửa thông tin cá nhân.

### Đánh giá kết quả

Ứng dụng đã đáp ứng được những yêu cầu cơ bản đặt ra như trò chuyện, nhắn tin, kết bạn với những người dùng khác, lưu được lịch sử các cuộc trò chuyện,hiện thị các thông báo tin nhắn.

Tuy nhiên, ứng dụng vẫn còn tồn tại một số hạn chế, bất cập như:

* Việc load dữ liệu ban đầu chưa nhanh
* Một số tính năng về thay đổi ảnh đại diện chưa tối ưu
* Chưa nghiên cứu áp dụng vào trong môi trường nội bộ đề sử dụng

## HƯỚNG PHÁT TRIỂN

Trong tương lai, để có thể cải thiện về chức năng, cũng như tối ưu hệ thống ứng dụng là điều cần thiết để đảm bảo quá trình sử dụng ứng dụng được tốt hơn. Do đó, một số hướng phát triển ứng dụng trong thời gian sắp tới là:

* Cải thiện về thiết kế giao diện người dùng cho đẹp mắt hơn.
* Tối ưu hóa để ứng dụng có thể load nhanh hơn.
* Sửa đổi, sửa chữa một số tính năng còn chưa tốt để cải thiện quá trình sử dụng của người dùng.
* Cải tiến để có thể sử dụng trong nội bộ.

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Nguyễn Phương Lan & Hoàng Đức Hải (2001), *JAVA – Lập trình mạng*, Nhà xuất bản Lao động Hà Nội.
2. Nguyễn Linh, *Phân tích chức năng hệ thống*
3. Mai Văn Hà, *Bài giảng Lập trình mạng,* Khoa Công nghệ thông tin, Đại học Bách Khoa – Đại học Đà Nẵng

**Tiếng Anh**

1. Barry A. Burd, *“Java Programming for Android Developers”*.

**Internet**

1. *https://* *viblo.asia*
2. [*http://android-er.blogspot.com*](http://android-er.blogspot.com)
3. [*https:/ /stackoverflow.com*](https://stackoverflow.com)
4. *https:// square.github.io/picasso/*
5. *https:// firebase.google.com/docs*
6. *https://* *youtube.com*

PHỤ LỤC

Toàn bộ source code ứng dụng BKZalo cho phép người dùng trò chuyện, nhắn tin được upload tại : https://github.com/TrungDuc2803 /BKZalo.git

**-------------------------**