

**TRƯỜNG ĐẠI HỌC TRẦN ĐẠI NGHĨA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**LẬP TRÌNH DI ĐỘNG**

**ĐỀ TÀI: Xây dựng Whatsapp clone app**

**G/V hướng dẫn**: Nguyễn Kiên Cường

**Nhóm 4:**

Sinh viên thực hiện*:*

Lê Quang Khải MSSV: 18DDS0803112

Hồ Chí Minh, 06/2021

PHẦN MỞ ĐẦU

Mạng điện thoại di động xuất hiện tại Việt Nam từ đầu những năm 1990 và theo thời gian số lượng các thuê bao cũng như các nhà cung cấp dịch vụ đi động tại Việt Nam ngày càng tăng. Do nhu cầu trao đổi thông tin ngày càng tăng và nhu cầu sử dụng sản phẩm công nghệ cao nhiều tính năng, cấu hình cao, chất lượng tốt, kiểu dáng mẫu mà đẹp, phong phú nên nhà cung cấp phải luôn luôn cải thiện, nâng cao những sản phẩm của mình. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kĩ thuật. Cùng với sự phát triển của thị trường điện thoại di động là sự phát triển mạnh mẽ của xu hướng lập trình phần mềm ứng dụng cho các thiết bị di động. Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động cũng phát triển mạnh mẽ và đang thay đổi từ ngày. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web based Mobile Application đã có rất phát triển trên thị trường truyền thông di động.

Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay, đã được nhà phát triển công nghệ rất nổi tiếng hiện nay là Google. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất. Ngày nay, với sự phát triển nhanh chóng của xã hội, nhu cầu cập nhật tin tức xã hội mọi lúc mọi nơi là rất cần thiết, vì vậy em đã chọn đề tài “Xây dựng Whatsapp clone app” với mục đích nghiên cứu, tìm hiểu về quá trình xây dựng, cách hoạt động của một mạng xã hội là như thế nào, cùng đó để phát triển kĩ năng lập trình của bản thân để xây dựng càng nhiều ứng dụng hiệu quả, nhanh chóng và tiện lợi.

MỤC LỤC

[CHƯƠNG I. TỔNG QUAN 4](#_Toc74705385)

[1. KHẢO SÁT THỰC TRẠNG 4](#_Toc74705386)

[2. ĐỐI TƯỢNG VÀ PHẠM VI CỦA ỨNG DỤNG 4](#_Toc74705387)

[2.1 Đối tượng 4](#_Toc74705388)

[2.2 Phạm vi 4](#_Toc74705389)

[2.3 Rằng buộc tổng quan hệ thống 5](#_Toc74705390)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 6](#_Toc74705391)

[1. TỔNG QUAN VỀ JAVA 6](#_Toc74705392)

[1.1. Môi trường lập trình 6](#_Toc74705393)

[1.2 Ngôn ngữ lập trình Java 6](#_Toc74705394)

[2. TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU REALTIME DATABASE 7](#_Toc74705395)

[CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 8](#_Toc74705396)

[1. XÁC ĐỊNH YÊU CẦU 8](#_Toc74705397)

[1.1 Yêu cầu chức năng 8](#_Toc74705398)

[1.2 Yêu cầu hệ thống 8](#_Toc74705399)

[2. MÔ HÌNH HÓA 8](#_Toc74705400)

[3. CƠ SỞ DỮ LIỆU CỦA HỆ THỐNG 9](#_Toc74705401)

[4. SOURCE CODE CỦA CHƯƠNG TRÌNH 12](#_Toc74705402)

[5. THIẾT KẾ GIAO DIỆN HỆ THỐNG 18](#_Toc74705403)

[CHƯƠNG IV. KẾT LUẬN 25](#_Toc74705404)

[1. KẾT LUẬN ĐỀ TÀI 25](#_Toc74705405)

[2. LỜI KẾT 26](#_Toc74705406)

CHƯƠNG I. TỔNG QUAN

## 1. KHẢO SÁT THỰC TRẠNG

Trong thời đại công nghiệp hóa, hiện đại hóa phát triển như hiện nay công nghệ thông tin được coi là một trong những ngành quan trọng nhất của nhiều quốc gia trên toàn thế giới. Với sự phát triển đó thì những ứng dụng mạng xã hội ngày càng xuất hiện một nhiều hơn để đáp ứng nhu cầu sử dụng của con người. Việc chia sẽ những hình ảnh, video, trò chuyện hay lưu trữ lại những khoảnh khắc đáng nhớ của chúng ta là thực sự cần thiêt. Xuất phát từ những nhận định trên nhóm chúng em đã quyết định tìm hiểu và lựa chọn đề tài “Xây dựng lại ứng dụng Whatsapp….” Hy vọng sẽ đem tới một ứng dụng cho sự trải nghiệm đầy đủ các tiện ích, thuận tiện đáp ứng được nhu cầu của người sử dụng.

## 2. ĐỐI TƯỢNG VÀ PHẠM VI CỦA ỨNG DỤNG

### 2.1 Đối tượng

Hệ thống được xây dựng hướng đến các đối tượng:

+ Người quản trị hệ thống

+ Tất cả người dùng ứng dụng.

### 2.2 Phạm vi

a) Phát biểu vấn đề

Việc giao tiếp trực tiếp với nhau hằng ngày gần như đơn giản và bình thường như thế nhưng không phải như vậy là đủ, chúng ta cần bày tỏ cảm xúc của mình và bạn bè, chia sẻ với nhau những câu chuyện của bản thân. Còn gì tốt hơn khi chũng ta minh họa những điều đó bằng một bức ảnh hay vài dòng tin nhắn. Sự can thiệp ứng dụng sẽ mang đến điều đó.

b) Mục tiêu

+ Cho phép người dùng chia sẻ ảnh miễn phí, cùng với những tính năng chụp ảnh trên điện thoại, thêm bộ lọc hình ảnh và sau đó chia sẻ chúng trên nhiều mạng xã hội khác nhau

+ Mọi người có thể tự do gửi tin nhắn và tương tác với nhau qua các tính năng

+ Đảm bảo cơ sở dữ liệu bảo mật và có độ tin cậy cao.

c) Mô tả

+ Hệ thống mới sẽ thu thập tất cả các thông tin về người dùng hỗ trợ người dùng tăng tính bảo mật cá nhân

+ Lập một trang chủ cho phép nhìn thấy trạng thái của bạn bè, xem tin nhắn cuối cùng được public cũng như tương tác lên đó

d) Lợi ích mang lại

+ Xây dựng một cộng đồng gắn kết trên phương tiện truyền thông xã hội.

+ Cho phép người dùng trở thành một người có ảnh hưởng quan trọng. Khi có một số lượng lớn người theo dõi bất cứ điều gì, sẽ mở ra một số khả năng và cơ hội mới cho người dùng đó.

+ Nó giúp cho việc giao tiếp từ xa trở nên dễ dàng hơn, đáp ứng được nhu cầu được kết nối và sẻ chia

e) Các bước thực hiện để hoàn thành dự án

+ Lập kế hoạch phát triển hệ thống.

+ Phân tích hệ thống.

+ Thiết kế.

+ Cài đặt.

+ Kiểm tra.

+ Biên soạn tài liệu.

+ Huấn luyện sử dụng.

### 2.3 Ràng buộc tổng quan hệ thống

+ Phần mềm sau khi triển khai phải đáp ứng được nhu cầu giải trí của người dùng, tạo điều kiện thuật lợi cho mọi người tương tác kết bạn với nhau.

+ Dữ liệu phải được truy cập nhanh chống và phải cập nhật thường xuyên.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

## 1. TỔNG QUAN VỀ JAVA

### 1.1. Môi trường lập trình

Java Development Kit (JDK - Bộ công cụ cho người phát triển ứng dụng bằng ngôn ngữ lập trình Java) là một tập hợp những công cụ phần mềm được phát triển bởi Sun Microsystems dành cho các nhà phát triển phần mềm, dùng để viết những applet Java hay những ứng dụng Java.

### 1.2 Ngôn ngữ lập trình Java

Java là một ngôn ngữ hướng đối tượng, đa mục đích với các cú pháp rất giống với C và C++. Ban đầu thì đa số mọi người nghĩ là Java sẽ chủ yếu được sử dụng để lập trình nên những applet hay những chương trình nhỏ chạy trên các trình duyệt web, tuy nhiên đến giờ thì mọi người đã thay đổi quan điểm. Một số người vốn trước đây tin rằng applet chính là đất sống của Java thì nghĩ rằng Java đã chết do sự xuất hiện của các đoạn phim hoạt hình Flash. Nhưng Java đã thay đổi. Cái thời mà người ta nghĩ rằng ứng dụng chủ yếu của Java là làm các applet động trên các trang web đã qua. Ngày nay, Sun, IBM, BEA... và các công ty khổng lồ khác đã liên kết để phát triển Java thành một môi trường đa năng chứ không chỉ dừng lại là một thứ ngôn ngữ lập trình đa nền tảng nữa. Java đã có mặt ở khắp mọi nơi: từ những chiếc điện thoại di động nhỏ bé mang nhãn hiệu Nokia, Samsung, Motorola, Ericcson..., từ các thiết bị PDA dùng hệ điều hành Palm cho đến các con chíp điện tử dùng trên các tấm thẻ tín dụng, các thiết bị chẩn đoán và phân tích dùng trong y tế, khai thác năng lượng, điểu khiển và quản lý thiết bị.... từ các phần mềm trên server, các trang web động, cho đến các ứng dụng trên desktop. Bạn có biết rằng người máy Người tìm đường Sao Hỏa dùng phần mềm điều khiển bằng ngôn ngữ Java không? Nhưng điều có lẽ còn cuốn hút bạn hơn khi bạn biết rằng ngay từ năm 1997, năm mà Java còn chưa tốt như bây giờ, Trung tâm Vũ Trụ NASA đã chính thức công nhận ngôn ngữ Java là ngôn ngữ chính được sử dụng để lập trình cho các thiết bị và phần mềm dùng cho Trung tâm.

Giờ đây, khi nhắc đến Java, người ta cần phải hiểu đó là: thứ nhất: đó là một môi trường phát triển và triển khai ứng dụng; thứ hai: đó là một ngôn ngữ lập trình toàn năng. Sự xuất hiển phổ biến của Flash không hề đe dọa đến Java. Rõ ràng với sự đầu tư của Sun và các công ty hỗ trợ Java khác, chỉ trong vòng 5 năm, nó đã trở thành một ngôn ngữ toàn năng nhất trong các ngôn ngữ lập trình được sử dụng trên thế giới hiện nay. Điều người ta quan tâm nhất ở Java là khả năng viết một lần chạy mọi nơi nghĩa là bạn có thể viết chương trình trên một máy tính cài Window, chạy chip của Intel nhưng chương trình đó vẫn chạy tốt và cho cùng một kết quả hoạt động khi chạy nó trên Macitosh hay Unix. Điều này là không tưởng đối với C, C++, VB... Khả năng chuyển đổi nền tảng, dễ phân phối, đa tầng, hướng đối tượng chính là những gì mà Java chứng tỏ nó ưu việt hơn các ngôn ngữ khác.

Với C, C++ tức là cha mẹ đẻ của Java thì điều này là rõ ràng. Với Visual Basic, ngôn ngữ lập trình quan trọng nhất của Microsoft, ngoài những điểm vừa nói ở trên, Java còn được giới lập trình chuyên nghiệp trên thế giới trong đó các hacker thế hệ thứ nhất đánh giá rằng đây là một ngôn ngữ có cú pháp và cấu trúc tốt hơn nhiều. Sử dụng Java bạn có thể dễ dàng mở rộng dự án lập trình của mình với quy mô không giới hạn, việc quản lý cũng hết sức dễ dàng trong khi đó Visual Basic với cấu trúc thiết kế không thực sự tốt, nó chỉ thích hợp với các dự án nhỏ, ít có nhu cầu mở rộng hay quản lý.

## 2. TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU REALTIME DATABASE

Đây là một cơ sở dữ liệu NoSQL lưu và đồng bộ dữ liệu trên mây. Dữ liệu được đồng bộ trên tất cả clients trong thời gian thực, và vẫn khả dụng khi ứng dụng offline.

Firebase Realtime Database là cơ sở dữ liệu lưu trữ trên mây. Dữ liệu được lưu trữ và đồng bộ hóa theo thời gian thực với mỗi client được kêt nối. Khi bạn xây dựng ứng dụng đa nền tẩng với iOS, Android, và javascript SDK, tất cả các client của bạn chia sẽ một thể hiện Realtime Database và tự động tiếp nhận các thay đổi với dữ liệu mới nhất.

Các khả năng chính của Realtime Database:

+ Thời gian thực: Firebase Realtime Database sử dụng đồng bộ dữ liệu mối khi dữ liệu có thay đổi, mọi thiết bị được kết nối sẽ nhận được thay đổi trong vài mili giây.

+ Offline: Khi người dùng ngoại tuyến, dữ liệu sẽ được lưu trên bộ nhớ cache của thiết bị và tự động đồng bộ khi bạn trực tuyến. Tất cả là tự động

+ Có thể truy cập từ thiết bị khách: Firebase Realtime Database có thể truy cập từ một thiết bị mobile hoặc trình duyệt web. Nó không cần một ứng dụng server nào cả. Bảo mật và xác thực dữ liệu có thể thông qua các Rule bảo mật của Firebase Realtime Database, các rule được thực thi khi dữ liệu được đọc hoặc ghi.

CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 1. XÁC ĐỊNH YÊU CẦU

### 1.1 Yêu cầu chức năng

+ Hệ thống phải lưu trữ được tất cả các thông tin chi tiết về người dùng

+ Cho phép người dùng lưu ảnh miễn phí và chia sẽ nó

+ Cho phép tương tác với nhau thông qua các phòng chat kín hoặc phòng chat nhóm

+ Tìm kiếm bạn bè

+ Lưu lại các tương tác của người dùng

### 1.2 Yêu cầu hệ thống

+ Hệ thống sử dụng hệ quản trị cơ sở dữ liệu đủ lớn để đáp ứng số lượng người dùng ngày càng tăng.

+ Máy chủ có khả năng xử lý nhanh, chính xác, lưu trữ lâu dài, bảo mật.

+ Hệ thống mạng đáp ứng khả năng truy cập lớn.

+ Thông tin có tính đồng bộ, phân quyền quản lý chặt chẽ.

+ Bảo mật tốt cho người quản trị hệ thống.

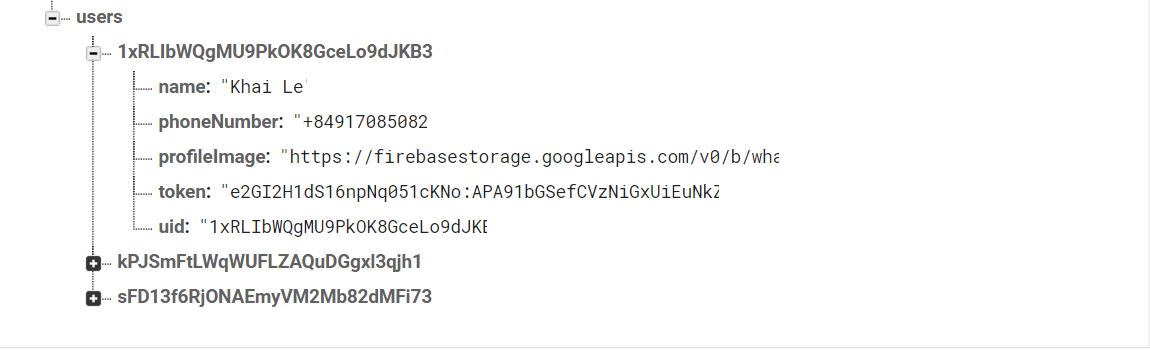
## 2. MÔ HÌNH HÓA

Biểu đồ phân rã chức năng (BFD)

## 3. CƠ SỞ DỮ LIỆU CỦA HỆ THỐNG

Để nắm được yêu cầu của bài toán, chúng ta cần hiểu dõ về cơ sở dữ liệu mà bài toán cần. Ở đây em xin đưa ra những thông tin mà phần mềm Whatsapp clone app cần có như sau:

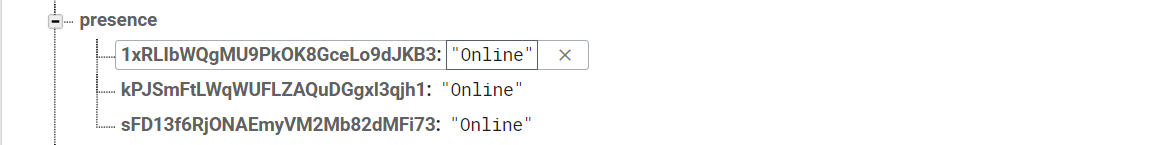
1. Nhánh User: Chứa các ID của user, nhánh nhỏ hơn gồm các thuộc tính name, phoneNumber, profileImage, token và uid(primary key) của ID user



1. Nhánh Story: Chứa các ID user có Story, nhánh nhỏ hơn là IDStory trong đó gồm lastUpdated, name, profileImage, statuses. Nhánh statuses lại tiếp tục chứa các storyId mà user từ nhánh gốc đã đăng, trong đó nhánh nhỏ hơn chứa imageUrl, timeStamp



1. Nhánh presence: Cho phép theo dõi trạng thái online hay offline của user , chứa ID user đã đăng ký, có value là 2 trạng thái online hay offline



1. Nhánh chats: Chứa ID của phòng chat, là sự kết hợp giữa 2 user chung 1 phòng, khi có sự kiện nhắn tin giữa cả hai, nhánh sẽ phân ra làm 2, nhánh thứ nhất có ID là ID người gửi + ID người nhận, nhánh thứ hai có ID là ID người nhận + ID người gửi nhằm phân biệt giữa người gửi và người nhận. Cả hai nhánh đều có các nhánh con như lastMsg, lastMgsTime, messages. Trong nhánh messages sẽ có các nhánh con là ID của message, chứa các nhánh con nhỏ hơn gồm: feeling, message, senderId, timeStamp, imageUrl (nếu có).



1. Nhánh public: Chứa các message ID có trong phòng chat nhóm, gồm các nhánh nhỏ hơn như: feeling, message, senderId, timeStamp, imageUrl (nếu có).



## 4. SOURCE CODE CỦA CHƯƠNG TRÌNH

+ Source code readUsers: Chức năng là đọc dữ liệu các user và hiển thị lên recycle view của user trong Main activity.

database.getReference().child("users").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 users.clear();  
 for(DataSnapshot snapshot1: snapshot.getChildren()){  
 User user = snapshot1.getValue(User.class);  
 if(!user.getUid().equals(FirebaseAuth.*getInstance*().getUid()) && user != null){  
 users.add(user);  
 }  
 }  
 binding.recyclerView.hideShimmerAdapter();  
 usersAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
});  
  
database.getReference().child("users").child(FirebaseAuth.*getInstance*().getUid())  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {  
 user = snapshot.getValue(User.class);  
 }  
  
 @Override  
 public void onCancelled(@NonNull @NotNull DatabaseError error) {  
  
 }  
 });

+ Source code readStory: Chức năng là đọc dữ liệu các story và hiển thị lên recycle view của story trong main activity

database.getReference().child("stories").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {  
 if(snapshot.exists()){  
 userStatuses.clear();  
 for(DataSnapshot snapshot1: snapshot.getChildren()){  
 UserStatus status = new UserStatus();  
 status.setName(snapshot1.child("name").getValue(String.class));  
 status.setProfileImage(snapshot1.child("profileImage").getValue(String.class));  
 status.setLastUpdated(snapshot1.child("lastUpdated").getValue(Long.class));  
  
 ArrayList<Status> statuses = new ArrayList<>();  
 for(DataSnapshot snapshot2: snapshot1.child("statuses").getChildren()){  
 Status status1 = snapshot2.getValue(Status.class);  
 statuses.add(status1);  
 }  
  
 status.setStatuses(statuses);  
 userStatuses.add(status);  
 }  
 binding.statusList.hideShimmerAdapter();  
 topStatusAdapter.notifyDataSetChanged();  
 }  
 else {  
 binding.statusList.hideShimmerAdapter();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull @NotNull DatabaseError error) {  
  
 }  
});

+ Source code checkPresence: Chức năng là kiểm tra trạng thái các user, nếu app còn hoạt động thì sẽ ghi vào database trạng thái online, ngược lại sẽ ghi offline

protected void onResume() {  
 super.onResume();  
 String currentId = FirebaseAuth.*getInstance*().getUid();  
 database.getReference().child("presence").child(currentId).setValue("Online");  
}

protected void onPause() {  
 String currentId = FirebaseAuth.*getInstance*().getUid();  
 database.getReference().child("presence").child(currentId).setValue("Offline");  
 super.onPause();  
}

+ Source code searchUsers: Chức năng thực hiện tìm kiếm khi textbox được thay đổi

private void searchUsers(String s){  
 Query query = FirebaseDatabase.*getInstance*("https://whatappsclone-335fb-default-rtdb.asia-southeast1.firebasedatabase.app").getReference("users").orderByChild("name")  
 .startAt(s.toUpperCase())  
 .endAt(s.toLowerCase()+"\uf8ff");  
 query.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 users.clear();  
 for(DataSnapshot snap: snapshot.getChildren()){  
 User user = snap.getValue(User.class);  
 if(!user.getUid().equals(FirebaseAuth.*getInstance*().getUid())){  
 users.add(user);  
 }  
 }  
 usersAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
}

+ Source code readPresence: Chức năng đọc và hiển thị trạng thái của user trong chat activity

database.getReference().child("presence").child(receiverUid).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {  
 if(snapshot.exists()){  
 String status = snapshot.getValue(String.class);  
 if(!status.isEmpty()){  
 if(status.equals("Offline")){  
 binding.status.setVisibility(View.*GONE*);  
 }  
 else{  
 if(binding.status.getVisibility() == View.*GONE*){  
 binding.status.setText(status);  
 binding.status.setVisibility(View.*VISIBLE*);  
 }  
 }  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull @NotNull DatabaseError error) {  
  
 }  
});

+ Source code readMessage: Đọc và hiển thị các message giữa 2 user

database.getReference().child("chats")  
 .child(senderRoom)  
 .child("messages")  
 .addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 messages.clear();  
 for(DataSnapshot snapshot1: snapshot.getChildren()){  
 Message message = snapshot1.getValue(Message.class);  
 message.setMessageId(snapshot1.getKey());  
 messages.add(message);  
 }  
  
 adapter.notifyDataSetChanged();  
 binding.recycleView.scrollToPosition(binding.recycleView.getAdapter().getItemCount() - 1);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });

+ Source code sendMessage: Chức năng gửi tin nhắn hoặc ảnh cho user được nhận

binding.sendBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String messageTxt = binding.messageBox.getText().toString();  
  
 Date date = new Date();  
 Message message = new Message(messageTxt, senderUid, date.getTime());  
 binding.messageBox.setText("");  
  
 String randomKey = database.getReference().push().getKey();  
  
 HashMap<String, Object> lastMsgObj = new HashMap<>();  
 lastMsgObj.put("lastMsg", message.getMessage());  
 lastMsgObj.put("lastMsgTime", date.getTime());  
 database.getReference().child("chats").child(senderRoom).updateChildren(lastMsgObj);  
 database.getReference().child("chats").child(receiverRoom).updateChildren(lastMsgObj);  
  
 database.getReference().child("chats")  
 .child(senderRoom)  
 .child("messages")  
 .child(randomKey)  
 .setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
 database.getReference().child("chats")  
 .child(receiverRoom)  
 .child("messages")  
 .child(randomKey)  
 .setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
 sendNotification(name, message.getMessage(), token);  
 }  
 });  
  
 }  
 });  
 }  
});

+ Source code uploadImage: Chức năng tải ảnh lên và lưu cho user

binding.attachment.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent();  
 intent.setAction(Intent.*ACTION\_GET\_CONTENT*);  
 intent.setType("image/\*");  
 startActivityForResult(intent, 25);  
 }  
});

+ Source code sendNotifications: Chức năng đọc dữ liệu và hiển thị lịch sử tương tác của người dùng

void sendNotification(String name, String message, String token){  
 try {  
 RequestQueue queue = Volley.*newRequestQueue*(this);  
 String url = "https://fcm.googleapis.com/fcm/send";  
  
 JSONObject data = new JSONObject();  
 data.put("title", name);  
 data.put("body", message);  
 JSONObject notificationData = new JSONObject();  
 notificationData.put("notification", data);  
 notificationData.put("to", token);  
 JsonObjectRequest request = new JsonObjectRequest(url, notificationData, new Response.Listener<JSONObject>() {  
 @Override  
 public void onResponse(JSONObject response) {  
 Toast.*makeText*(ChatActivity.this, "Success", Toast.*LENGTH\_SHORT*).show();  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.*makeText*(ChatActivity.this, error.getLocalizedMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 }){  
 @Override  
 public Map<String, String> getHeaders() throws AuthFailureError {  
 HashMap<String, String> map = new HashMap<>();  
 String key = "Key=AAAAVPa2v20:APA91bH2yidRU84ZnfGJNaiDn1TQeHlMdEGtqGvO\_0-TvueXgZSJzBHSIzsgUYDGy\_HOXW5x9BRPcKQ\_0PbM\_VMmxVsGgXnbbZpQGzEzc0sJZWIDfXdUlDMkDIJCeRUe1A4v9Ot\_uVpq";  
 map.put("Authorization", key);  
 map.put("Content-Type", "application/json");  
 return map;  
 }  
 };  
  
 queue.add(request);  
 }  
 catch (Exception exception){  
  
 }  
  
}

+ Source code setupProfile: Chức năng kiểm tra thông tin và thay đổi thông tin của user

binding.continueBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String name = binding.nameBox.getText().toString();  
  
 if(name.isEmpty()){  
 binding.nameBox.setError("Please type a name");  
 }  
  
 dialog.show();  
 if(selectedImage != null){  
 StorageReference storageReference = storage.getReference().child("Profiles").child(auth.getUid());  
 storageReference.putFile(selectedImage).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  
 if(task.isSuccessful()){  
 storageReference.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String imageUrl = uri.toString();  
  
 String uid = auth.getUid();  
 String phone = auth.getCurrentUser().getPhoneNumber();  
 String name = binding.nameBox.getText().toString();  
 User user = new User(uid, name, phone, imageUrl);  
 database.getReference()  
 .child("users")  
 .child(uid)  
 .setValue(user)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
 dialog.dismiss();  
 Intent intent = new Intent(SetupProfileActivity.this, MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 });  
 }  
 });  
 }  
 }  
 });  
 }  
 else{  
 String uid = auth.getUid();  
 String phone = auth.getCurrentUser().getPhoneNumber();  
 User user = new User(uid, name, phone, "No Image");  
 database.getReference()  
 .child("users")  
 .child(uid)  
 .setValue(user)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
 dialog.dismiss();  
 Intent intent = new Intent(SetupProfileActivity.this, MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 });  
 }  
 }  
});

+ Source code onResult: Chức năng đọc và hiển thị tất cả ảnh của user tải lên sau khi thực hiện chứa năng uploadImage

protected void onActivityResult(int requestCode, int resultCode, @Nullable @org.jetbrains.annotations.Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if(requestCode == 25){  
 if(data != null){  
 if(data.getData() != null){  
 Uri selectedImage = data.getData();  
 Calendar calendar = Calendar.*getInstance*();  
 StorageReference reference = storage.getReference().child("chats").child(calendar.getTimeInMillis() + "");  
 dialog.show();  
 reference.putFile(selectedImage).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onComplete(@NonNull @NotNull Task<UploadTask.TaskSnapshot> task) {  
 dialog.dismiss();  
 if(task.isSuccessful()){  
 reference.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 String filePath = uri.toString();  
  
 String messageTxt = binding.messageBox.getText().toString();  
  
 Date date = new Date();  
 Message message = new Message(messageTxt, senderUid, date.getTime());  
 message.setMessage("photo");  
 message.setImageUrl(filePath);  
 binding.messageBox.setText("");  
  
 String randomKey = database.getReference().push().getKey();  
  
 HashMap<String, Object> lastMsgObj = new HashMap<>();  
 lastMsgObj.put("lastMsg", message.getMessage());  
 lastMsgObj.put("lastMsgTime", date.getTime());  
 database.getReference().child("chats").child(senderRoom).updateChildren(lastMsgObj);  
 database.getReference().child("chats").child(receiverRoom).updateChildren(lastMsgObj);  
  
 database.getReference().child("chats")  
 .child(senderRoom)  
 .child("messages")  
 .child(randomKey)  
 .setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
 database.getReference().child("chats")  
 .child(receiverRoom)  
 .child("messages")  
 .child(randomKey)  
 .setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
  
 }  
 });  
  
 }  
 });  
  
 //Toast.makeText(ChatActivity.this, filePath, Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }  
 }  
 });  
 }  
 }  
 }  
}

+ Source code sendFeeling: Chức năng hiển thị popup feeling cho phép người dùng có thể gửi feeling qua người nhận về một bức ảnh hay tin nhắn

ReactionPopup popup = new ReactionPopup(context, config, (pos) -> {  
  
 if(pos < 0){  
 return false;  
 }  
  
 if(holder.getClass() == SentViewHolder.class){  
 SentViewHolder viewHolder = (SentViewHolder)holder;  
 viewHolder.binding.feeling.setImageResource(reactions[pos]);  
 viewHolder.binding.feeling.setVisibility(View.*VISIBLE*);  
 }  
 else{  
 ReceiveViewHolder viewHolder = (ReceiveViewHolder)holder;  
 viewHolder.binding.feeling.setImageResource(reactions[pos]);  
 viewHolder.binding.feeling.setVisibility(View.*VISIBLE*);  
 }  
  
 message.setFeeling(pos);  
  
 FirebaseDatabase.*getInstance*("https://whatappsclone-335fb-default-rtdb.asia-southeast1.firebasedatabase.app").getReference()  
 .child("chats")  
 .child(senderRoom)  
 .child("messages")  
 .child(message.getMessageId())  
 .setValue(message);  
  
 FirebaseDatabase.*getInstance*("https://whatappsclone-335fb-default-rtdb.asia-southeast1.firebasedatabase.app").getReference()  
 .child("chats")  
 .child(receiverRoom)  
 .child("messages")  
 .child(message.getMessageId())  
 .setValue(message);  
  
 return true; // true is closing popup, false is requesting a new selection  
});

## 5. THIẾT KẾ GIAO DIỆN HỆ THỐNG

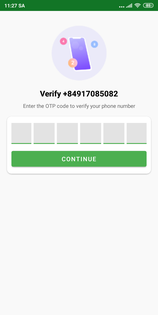
+ Form xác nhận số điện thoại



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Phone textView | Cho phép nhập thông tin về số điện thoại | Click và type |
| Continue button | Xác nhận thông tin số điện thoại và chuyển đến activity tiếp theo | Click |

+ Form xác nhận mã OTP



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| OTP view | Cho phép nhập mã OTP đã được gửi | Click và type |
| Continue button | Xác nhận thông tin mã OTP và chuyển đến activity tiếp theo | Click |

+ Form trang chủ



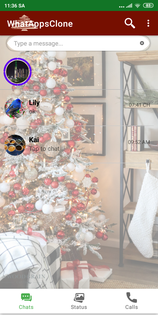


Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Story | Truy cập đến story cá nhân hoặc story của user khác | Click |
| Chats | Trở về trang chủ | Click |
| Status | Truy cập vào file Image và chọn hình để đăng story | Click |
| User | Cho phép vào phòng chat cùng user muốn gửi tin nhắn | Click |
| Search (a) | Hiển thị thanh tìm kiếm user | Click |
| More (b) | Hiển thị dialog có các chức năng khác | Click |

Kết quả của các sự kiện trong MainActivity Form:

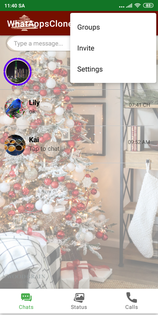
(a) Hiển thị thanh tìm kiếm



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| MessageBox | Cho phép tìm kiếm user khác | Click và type |
| Cancel Button | Bỏ tìm kiếm và ẩn thanh tìm kiếm | Click |

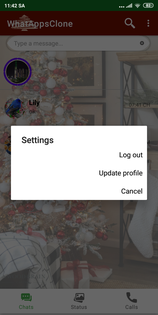
(b) Hiển thị dialog có các chức năng khác:



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Group Button | Cho phép vào phòng chat nhóm | Click |
| Settings (c) | Mở Dialog settings | Click |

(c) Mở Dialog settings



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Log out | Đăng xuất khỏi app | Click |
| Upload profile | Mở UpdateProfile activity | Click |
| Cancel | Đóng Dialog settings | Click |

+ From story



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Left click | Chuyển qua story tiếp theo | Left click screen |
| Right click | Trở về story trước đó | Right click screen |
| Cancel | Tắt form story | Click |

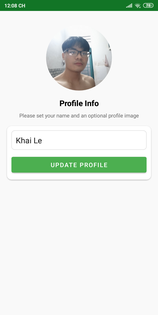
+ From tải ảnh lên



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Camera | Chọn camera hay thư viện ảnh | Click |
| Search | Tìm kiếm ảnh | Click |
| View type | Chọn chế độ xem | Click |

+From Thông tin cá nhân



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Profile image | Truy cập đến thư viện ảnh và load ảnh đại diện | Click |
| Name Box | Nhập thông tin tên tài khoản | Click |
| Update profile button | Cập nhật thông tin người dùng | Click |

+ From Chats



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Back | Quay lại màn hình chính | Click |
| MessageBox | Cho phép nhập nội dung tin nhắn | Click và type |
| Attachment | Cho phép upload ảnh hoặc tài liệu đính kèm | Click |
| Camera | Mở camera | Click |
| Send button | Gửi tin nhắn hoặc tài liệu đính kèm | Click |
| Message click (a) | Mở popup feeling | Click |
| Message long click (b) | Mở dialog Delete mesage | Click |

Kết quả của các sự kiện trong ChatActivity Form:

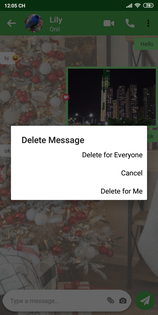
(a) Mở popup feeling



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Feeling icon | Cho phép gửi 1 trong các icon feeling | Click |
| Back | Hủy gửi feeling icon | Click |

(b) Mở dialog Delete message



Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Delete for Everyone | Xóa tin nhắn ở mọi phía | Click |
| Delete for me | Xóa tin nhắn ở phía bạn | Click |
| Cancel | Đóng dialog Delete message | Click |

CHƯƠNG IV. KẾT LUẬN

## 1. KẾT LUẬN ĐỀ TÀI

a) Đánh giá chung

Ưu nhược điểm của hệ thống mới

* Ưu điểm:

+ Việc chia sẽ các hình ảnh có thể truyền nhiều thông tin hơn một văn bản bằng văn bản. Hình ảnh có khả năng gợi lên cảm xúc và cảm xúc. Chúng cũng hấp dẫn hơn các hình thức tương tác khác. Theo cách này, một hệ thống truyền thông hiệu quả hơn nhiều được thiết lập giữa những người dùng.

+ Là nền tảng để kết nối giữa những người dùng với nhau dễ dàng hơn, mọi lúc mọi nơi miễn có kết nối internet.

+ Biểu đạt được cảm xúc cá nhân và thể hiện được những gì bản thân muốn chia sẻ

+ Cập nhật được trạng thái hiện tại thông qua chức năng story

* Nhược điểm:

+ Khi xuất bản hình ảnh chất lượng hoặc chuyên nghiệp trên mạng xã hội, có thể ai đó truy cập chúng và sử dụng chúng cho các vấn đề chuyên nghiệp có lợi cho họ về tài chính... Do đó, có khả năng "đánh cắp" hình ảnh từ người dùng mà không có sự đồng ý của họ.

+ Không thể định cấu hình riêng tư của từng ảnh.

+ Ứng dụng có thể trở thành nghiện đối với người dùng. Vì lý do này, nó phải được sử dụng một cách thận trọng.

b) Hướng phát triển và mở rộng đề tài

Để Whatsapp clone app góp phần mang lại lợi ích trong cuộc sống của con người, thì việc mở rộng đề tài, xem xét nhiều khía cạnh hơn nữa để phần mềm được hoàn thiện hơn là rất cần thiết. Trong đề tài này em chỉ mới có phân tích và xây dựng phần mềm đơn giản chưa có tính phức tạp. Vì vậy, hướng phát triển của đề tài này là:

+ Tích hợp thêm nhiều phương tiện giải trí cho người sử dụng

+ Kết nối nhiều hơn nữa những người sử dụng

+ Đưa sản phẩm thực hiện được trên đa nền tảng

+ Tiếp tục hoàn chỉnh các chức năng còn thiếu sót.

## 2. LỜI KẾT

Xây dựng phần mềm mạng xã hội nói chung, phần mềm “Whatsapp clone app” nói riêng không chỉ là việc xây dựng đơn thuần, mà nó đòi hỏi một cách có hệ thống các giai đoạn khảo sát, phân tích thiết kế phải được tiến hành trước đó.

Đứng trước xu thế phát triển của công nghệ thông tin như vũ bão hiện nay thì xây dựng phần mềm mạng xã hội là điều không thể thiếu, đây là một đề tài mang tính thực tế cao. Qua việc nghiên cứu xây dựng đề tài này, một phần đã củng cố cho em kiến thức về lập trình thì nó cũng cung cấp cho em thêm là làm thế nào có thể xây dựng được một phần mềm hoàn chỉnh, qua bài này em đã có thể tự tin xây dựng được các phần mềm tương tự, phát triển thêm một số tính năng đột phá và đã có những ý tưởng mới trong tương lai của mình.