

Chat Firebase Android

1. Tổng quan

Xây dựng một ứng dụng chat cơ bản với Firebase và Android Studio

Tương tác với một cơ sở dữ liệu của Firebase từ một ứng dụng Android

Sử dụng Firebase Authentication trong một ứng dụng Android để xác thực người dùng.

2. Những gì bạn cần :

Android Studio phiên bản 1.0 trở lên

Một thiết bị kiểm tra hoặc giả lập với Android 4.1 trở lên

Thiết bị phải có quyền truy cập internet để truy cập vào máy chủ Firebase

3. Tạo một ứng dụng trên Firebase

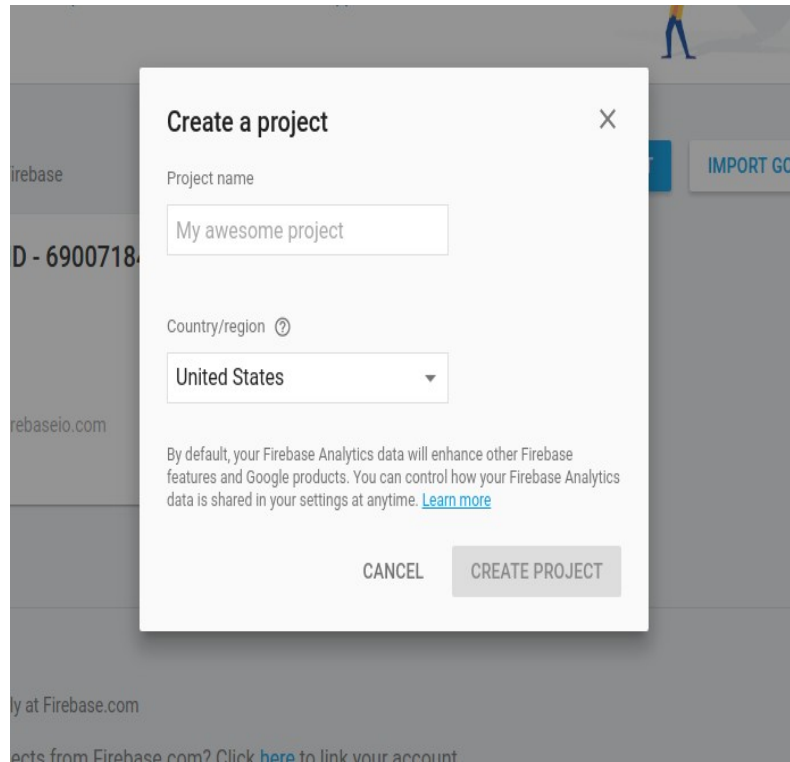
Bước đầu tiên là tạo ra một ứng dụng Firebase . Đây sẽ là cài đặt phía máy chủ Firebase để có thể tích hợp vào app trên Android Studio :

+ Hãy vào trang web Firebase : <https://firebase.google.com/>

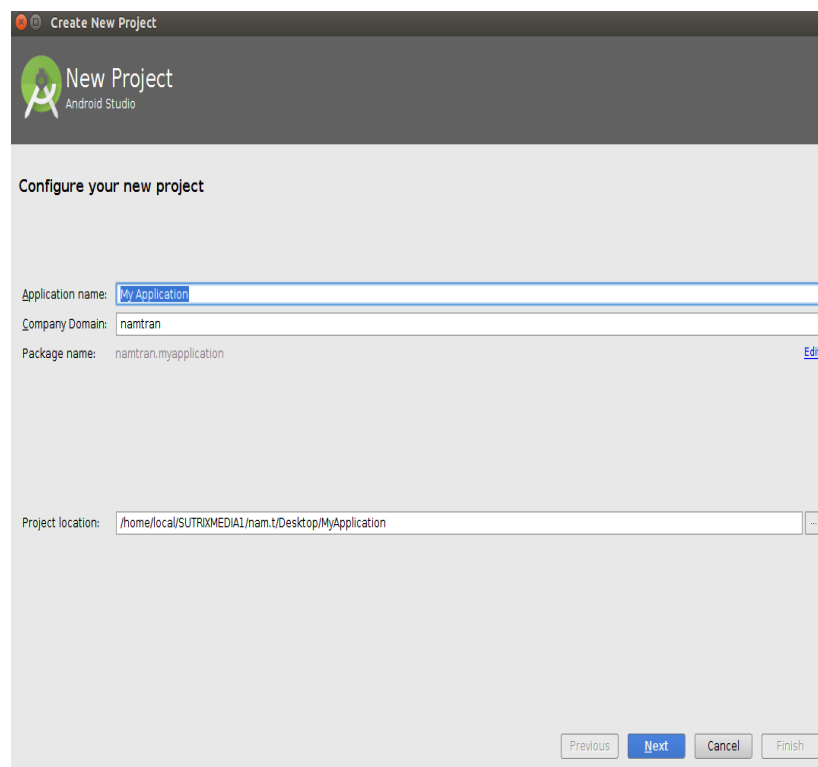
+ Đăng nhập hoặc đăng ký

+ Nhấp vào **GET STARTED FOR FREE** → **CREATE NEW PROJECT**

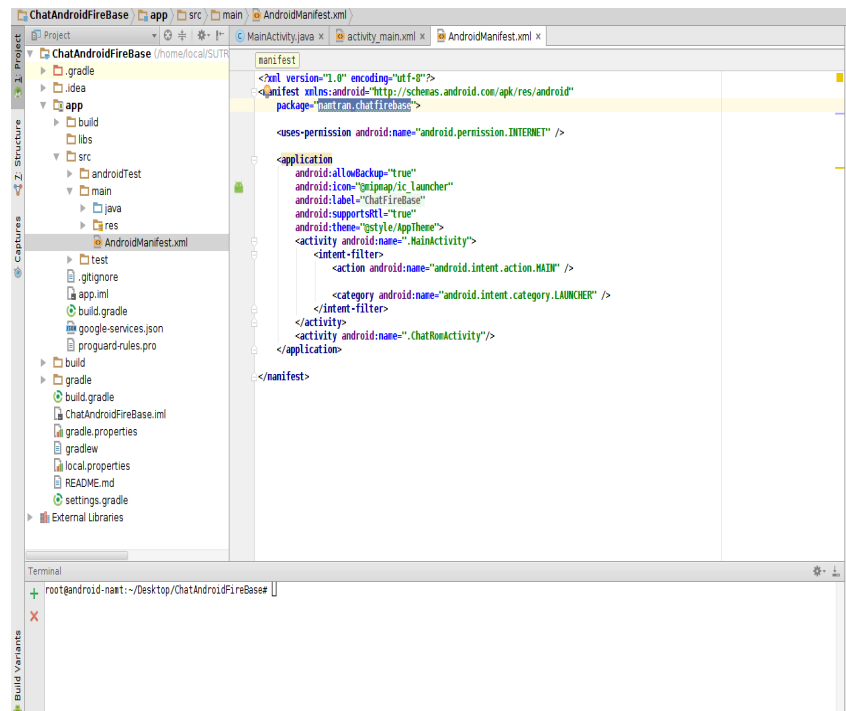
The image shows a sequence of screenshots from the Firebase website and console. The top screenshot is the Firebase homepage with the headline "App success made simple" and a "GET STARTED FOR FREE" button. Below this is a diagram illustrating the Firebase ecosystem, categorized into "DEVELOP" (Realtime Database, Authentication, Cloud Messaging, Storage, Hosting, Test Lab, Crash Reporting), "ANALYTICS" (Analytics), "GROW" (Notifications, Remote Config, App Indexing, Dynamic Links, Inlines, AdWords), and "EARN" (AdMob). The bottom screenshot is the "Welcome back to Firebase" page, which prompts the user to continue building apps and provides links for documentation, sample code, API reference, and support. It also displays a list of projects using Firebase, including "AdMob App ID - 6900718466" and "ChatAndroidFireBase", each with its own AdMob App ID and a link to the project page.



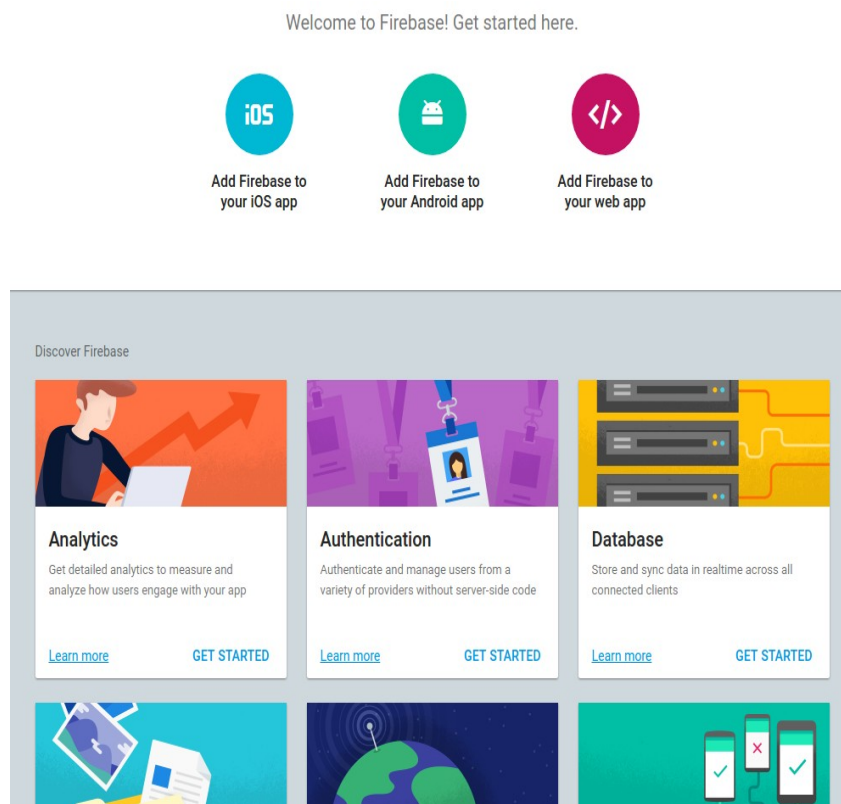
+ Mở Android Studio và tạo một ứng dụng



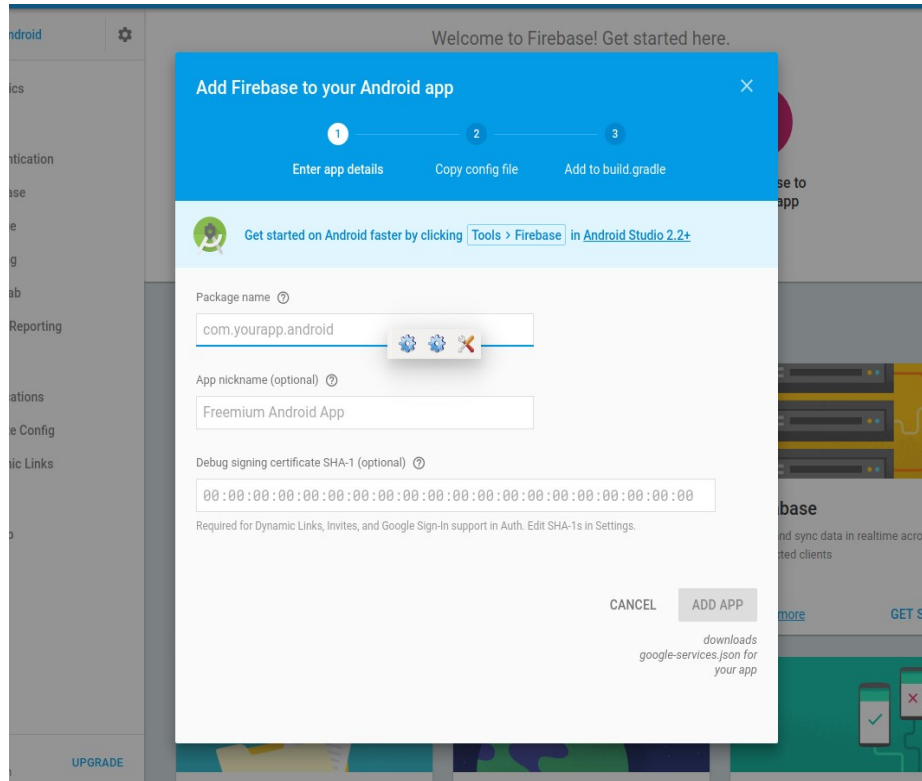
+ Sau khi tạo xong mở file Android Mainifesh và copy package của ứng dụng :



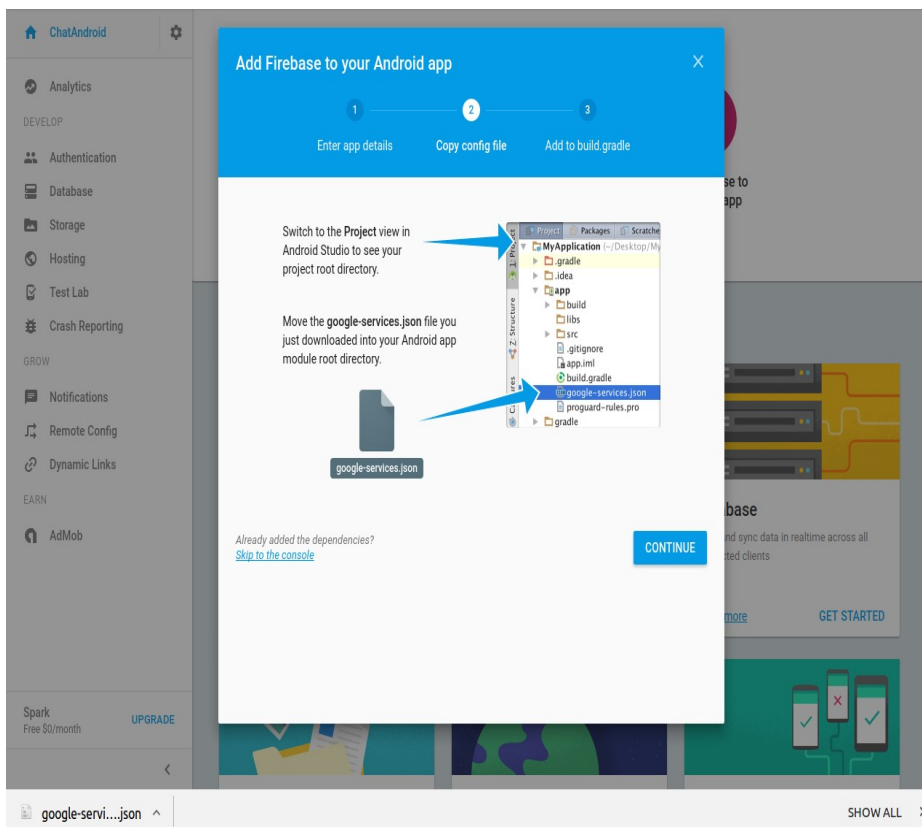
+ MỞ lại tab firebase vừa này . Sau khi tạo project trên FireBase nhấp tiếp vào Add Firebase to your Android app



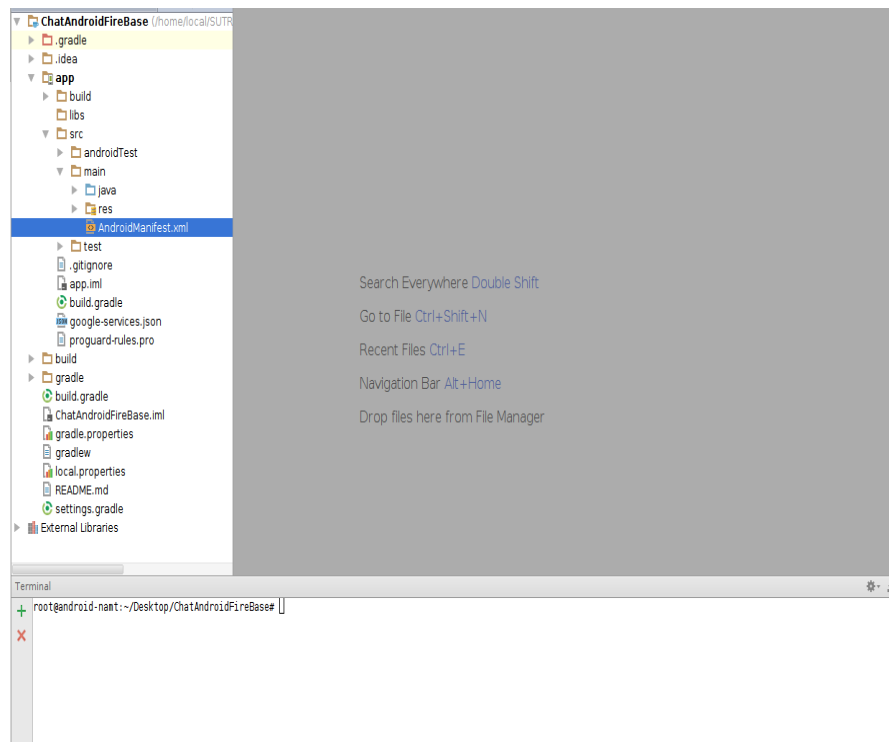
+ Dán package bạn vừa copy ở Mainifesh vào và ấn Add App



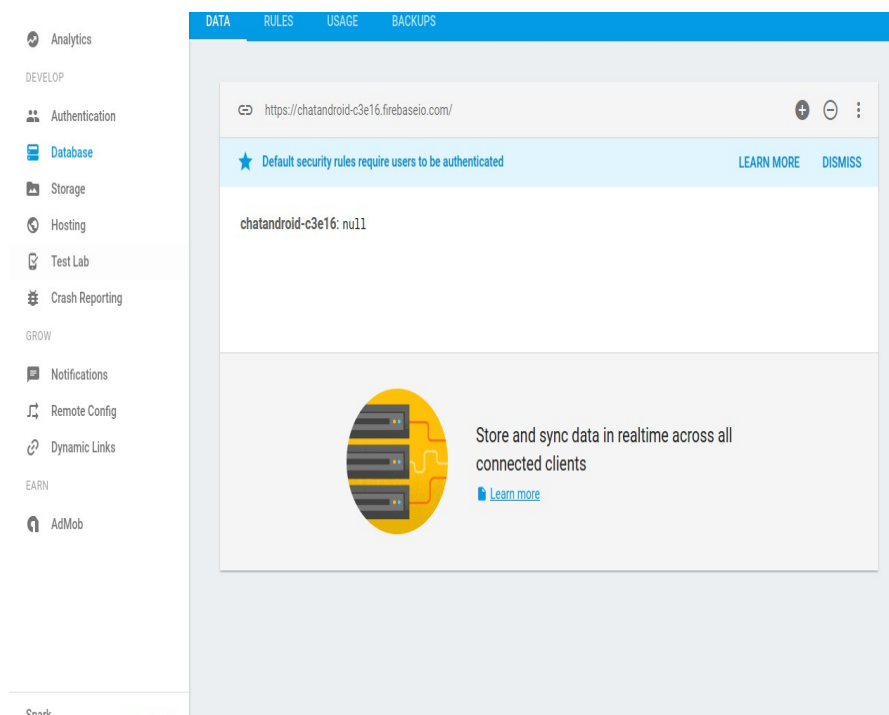
+ FireBase sẽ tự sinh ra một file google-service.json



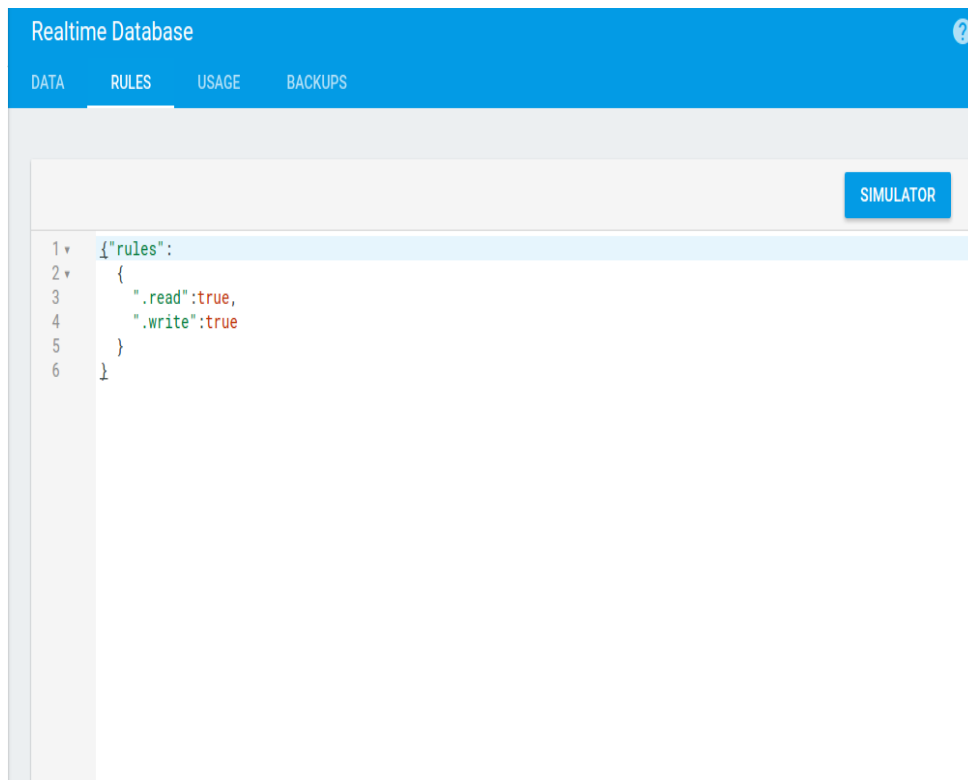
+ Bạn copy file nào bỏ vào ứng dụng



+ Bạn mở lại tab firebase trên web nhấp vào DataBase



+ Chọn qua tab Rules và chỉnh sửa thành



==> Hoàn tất quá trình cấu hình trên máy chủ Firebase . Bây giờ bắt đầu code trong Android Studio

4 . Import FireBase in Android Studio

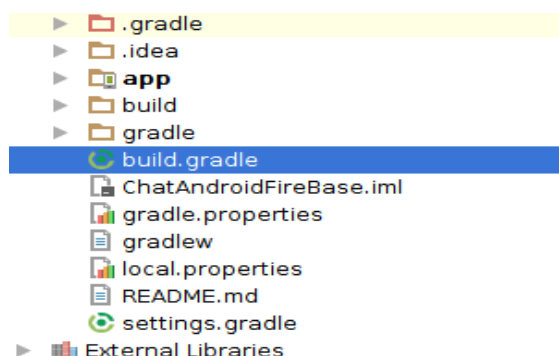
Trước khi chúng ta có thể bắt đầu code tương tác với cơ sở dữ liệu Firebase , Chúng ta cần import sdk firebase vào project :

+ Mở Gradle Scripts > build.gradle (Module: app). File này chứa các thư viện mà Android Studio sử dụng để xây dựng ứng dụng .

+Thêm dòng sau vào dependencies ở phía dưới:

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:24.2.1'
    compile 'com.google.firebase:firebase-database:9.0.0'
    compile 'com.google.firebase:firebase-auth:9.0.0'
    compile 'com.google.firebase:firebase-messaging:9.0.0'
}
```

+ Mở Gradle của Android studio



+ Thêm dòng sau bên dưới dependencies :

```
}  
dependencies {  
    classpath 'com.android.tools.build:gradle:2.1.0'  
    classpath 'com.google.gms:google-services:3.0.0'  
  
    // NOTE: Do not place your application dependencies here; they belong  
    // in the individual module build.gradle files  
}
```

+ Mở lại Gradle của app và thêm dòng sau trên cùng :

```
apply plugin: 'com.android.application'  
apply plugin: 'com.google.gms.google-services'
```

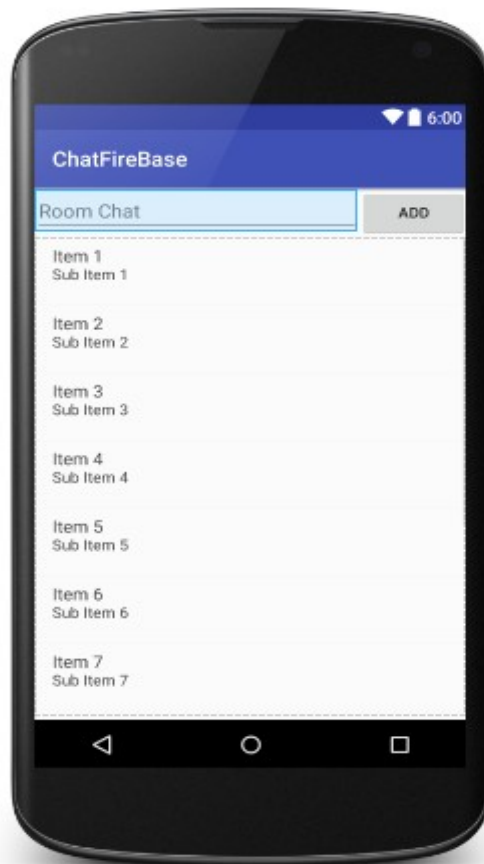
+ Mở Android Mainifest và thêm vào câu

“<uses-permission android:name="android.permission.INTERNET" /> “ để xin quyền sử dụng Internet cho ứng dụng

==> Hoàn tất quá trình import thư viện . Bây giờ chính thức code .

5. Activity Room

Tạo layout cơ bản với listview chứa các room edittext để nhập tên room và button để click add room chat



Khi mở app sẽ có hàm requestUser() để yêu cầu người dùng nhập tên (Dùng Dialog để hiện thi

và dùng sharedPreferences để lưu lại trạng thái đã nhập tên hay chưa cho những lần mở app sau)

Gọi đối tượng DatabaseReference từ firebase :

```
DatabaseReference root = FirebaseDatabase.getInstance().getReference().getRoot();
```

Một tham chiếu FireBase đại diện cho một vị trí đặc biệt trong cơ sở dữ liệu của bạn và có thể được sử dụng cho việc đọc hoặc ghi dữ liệu vào vị trí đó cơ sở dữ liệu.

Khi click button add room sẽ truyền text từ EditText vào đối tượng Map<String,Object> và sau đó đối tượng này sẽ được truyền vào hàm “updateChildren” của DatabaseReference (Cập nhật các Key con cụ thể cho các giá trị quy định. Đi qua null trong một bản đồ để updateChildren () sẽ loại bỏ các giá trị tại vị trí quy định.)

```
mBtnAdd.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Map<String,Object> map = new HashMap<>();  
        map.put(mEdtRoom.getText().toString(), "");  
        root.updateChildren(map);  
        mEdtRoom.setText("");  
    }  
});
```

Sau đó chúng ta gọi interface “ValueEventListener”

```
root.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        Set<String> set = new HashSet<>();  
        Iterator item = dataSnapshot.getChildren().iterator();  
        while (item.hasNext()){  
            set.add(((DataSnapshot)item.next()).getKey());  
        }  
  
        mListRoom.clear();  
        mListRoom.addAll(set);  
        mAdapterRoom.notifyDataSetChanged();  
    }  
  
    @Override  
    public void onCancelled(DatabaseError databaseError) {  
    }  
});
```

ValueEventListener : Class thực hiện giao diện này có thể được sử dụng để nhận các sự kiện về những thay đổi dữ liệu tại một địa điểm. Gắn việc lắng nghe một addValueEventListener vị trí người dùng (ValueEventListener)

public abstract void onCancelled (DatabaseError error) : Phương pháp này sẽ được kích hoạt trong trường hợp người nghe này, hoặc không thành công tại các máy chủ, hoặc được lấy ra như là kết quả của sự an toàn và Firebase quy định cơ sở dữ liệu.

public abstract void **onDataChange** (**DataSnapshot** snapshot) : Phương pháp này sẽ được gọi với a snapshot của các dữ liệu tại vị trí này. Nó cũng sẽ được gọi mỗi khi dữ liệu thay đổi.

(Nguồn :

<https://firebase.google.com/docs/reference/android/com/google/firebase/database/ValueEventListener>)

==> Sự kiện lắng nghe khi database thay đổi (Đối với Room)

DataSnapshot : Một **DataSnapshot** instance chứa dữ liệu từ một vị trí Cơ sở dữ liệu FireBase . Bất cứ lúc nào bạn đọc dữ liệu Cơ sở dữ liệu, bạn nhận được dữ liệu như một **DataSnapshot** . Chúng được hiệu quả tạo ra các bản sao bất biến của các dữ liệu tại một vị trí Cơ sở dữ liệu FireBase Họ không thể được sửa đổi và sẽ không bao giờ thay đổi. Để sửa đổi dữ liệu tại một địa điểm, sử dụng một **DatabaseReference**

dataSnapshot.getChildren().iterator() : Trả về danh sách các phòng . Dùng while để đọc các giá trị nhận được sau đó update lại adapter cho listview .

Sau khi tạo phòng chat , chúng ta xử lý việc nhấp vào phòng chat với **OnItemClickListener** của **listview**

```
mLvRoom.setOnItemClickListener((adapterView, view, i, l) -> {  
    Intent intent = new Intent(getApplicationContext(), ChatRomActivity.class);  
    intent.putExtra("room_name", ((TextView)view).getText().toString());  
    intent.putExtra("user_name", mUserName);  
    startActivity(intent);  
});
```

Truyền vào tên User và tên Room qua extra của intend == > qua activity Chat.

6. Activity Chat

Tạo layout đơn giản gồm edittext để nhập message . Button sent message tới server và TextView nằm trong ScrollView để show các message



Dùng getIntent để lấy username và roomname bên MainActivity

```
mUserName = getIntent().getExtras().getString("user_name").toString();  
mRoomName = getIntent().getExtras().getString("room_name").toString();
```

Vẫn gọi DatabaseReference giống bên MainActivity nhưng lần này sẽ chỉ rõ ra là đang ở room nào

```
root = FirebaseDatabase.getInstance().getReference().child(mRoomName);
```

Cũng giống như khi push room Name lên server database của FireBase . Chúng ta tạo ra một object Map<String,Object> . Nhưng lần này chúng ta truyền thêm vào roomname vào Object để xác định phòng chat và tạo ra Map<String,Object> thứ để chứa username và message từ edittext sau đó gửi đến FireBase with phương thức updateChildren(Map<String,Object>)

```
mBtnSend.setOnClickListener((view) -> {  
    Map<String,Object> map = new HashMap<>();  
    temp_key = root.push().getKey();  
    root.updateChildren(map);  
  
    DatabaseReference message_root = root.child(temp_key);  
  
    Map<String,Object> map2 = new HashMap<>();  
    map2.put("user_name", mUserName);  
    map2.put("message", mEdtMessage.getText().toString());  
  
    message_root.updateChildren(map2);  
    mEdtMessage.setText("");  
});
```

Chúng ta hứng các đoạn chat với phương thức addChildEventListener(ChildEventListener())

```

root.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        appendChatConversation(dataSnapshot);
    }

    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {
        appendChatConversation(dataSnapshot);
    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {
    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});

```

ChildEventListener : Interface này có thể được sử dụng để nhận các sự kiện về những thay đổi trong các địa điểm con của một DatabaseReference chỉ định. Gắn việc lắng nghe một vị trí bằng addChildEventListener (ChildEventListener) và các phương pháp thích hợp sẽ được kích hoạt khi thay đổi xảy ra.

onCancelled(DatabaseError error) : Phương thức này sẽ được kích hoạt trong trường hợp người nghe này, hoặc không thành công tại các máy chủ, hoặc được lấy ra như là kết quả của các quy tắc bảo mật và Firebase.

onChildAdded(DataSnapshot snapshot, String previousChildName) : Phương thức này được kích hoạt khi một đứa trẻ mới được bổ sung vào vị trí mà người nghe này đã được bổ sung.

onChildChanged(DataSnapshot snapshot, String previousChildName) : Phương thức này được kích hoạt khi dữ liệu tại một vị trí con đã thay đổi.

onChildMoved(DataSnapshot snapshot, String previousChildName) : Phương thức này được kích hoạt khi một vị trí của đứa trẻ thay đổi ưu tiên.

onChildRemoved(DataSnapshot snapshot) : Phương thức này được kích hoạt khi một đứa trẻ được lấy ra từ các vị trí mà người nghe này đã được bổ sung.

(Nguồn :

<https://firebase.google.com/docs/reference/android/com/google/firebase/database/ChildEventListener>)

Chúng ta sẽ show các đoạn chat với hàm appendChatConversation(DataSnapshot dataSnapshot) và được truyền vào trong onChildAdded và onChildChanged của interface ChildEventListener

```

root.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        appendChatConversation(dataSnapshot);
    }

    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {
        appendChatConversation(dataSnapshot);
    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {

    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});
}

private String mNameUser, mMessage;
private void appendChatConversation(DataSnapshot dataSnapshot) {

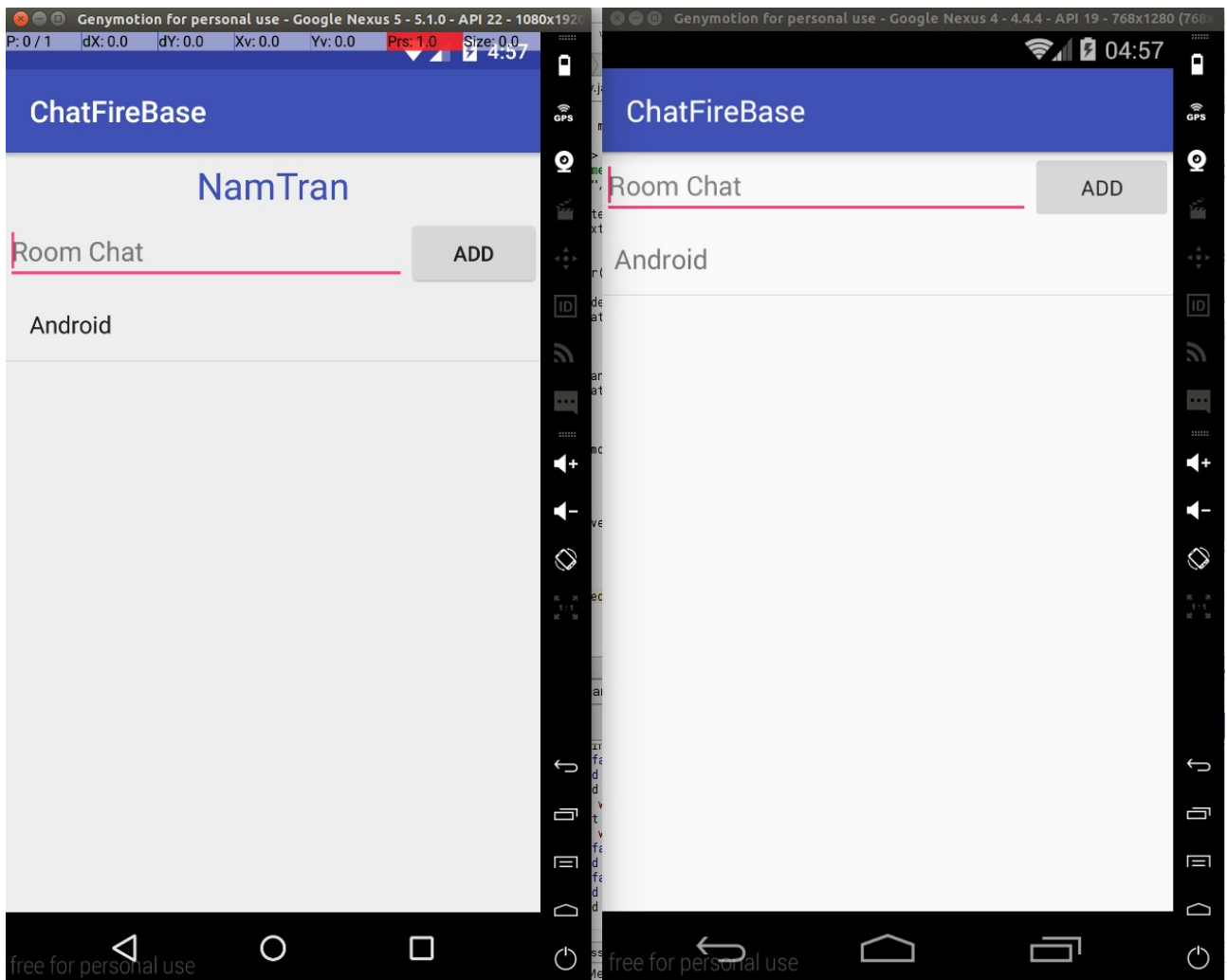
    Iterator i = dataSnapshot.getChildren().iterator();
    while (i.hasNext()){
        mMessage = ((DataSnapshot)i.next()).getValue().toString();
        mNameUser = ((DataSnapshot)i.next()).getValue().toString();

        mTxtConversation.append(mNameUser + " : " + mMessage + "\n");
    }
}
}

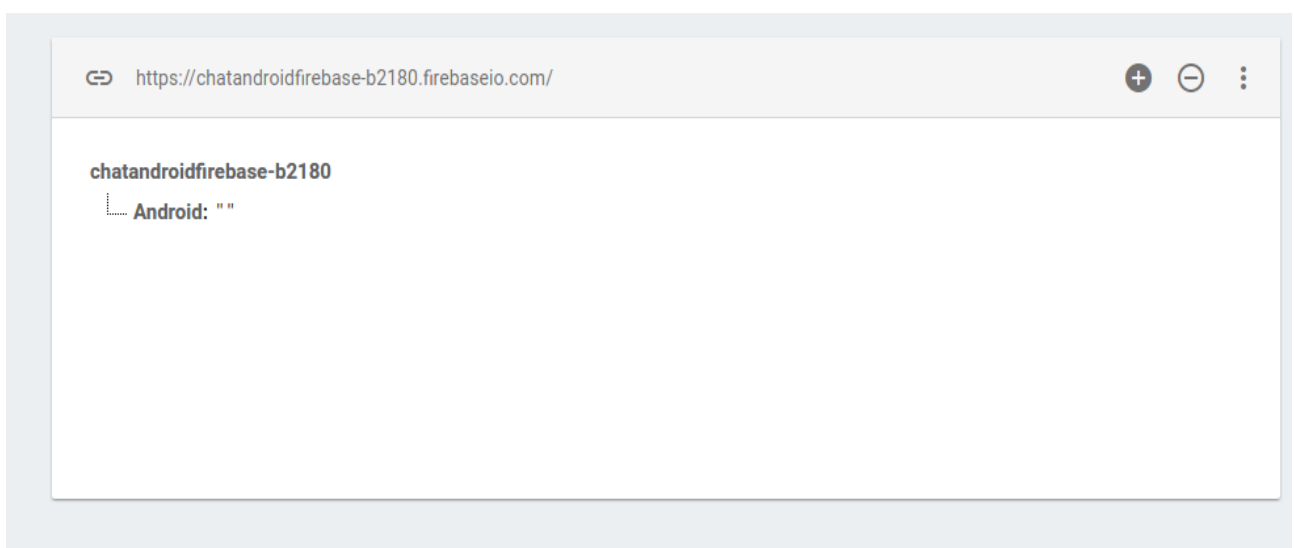
```

7. Demo

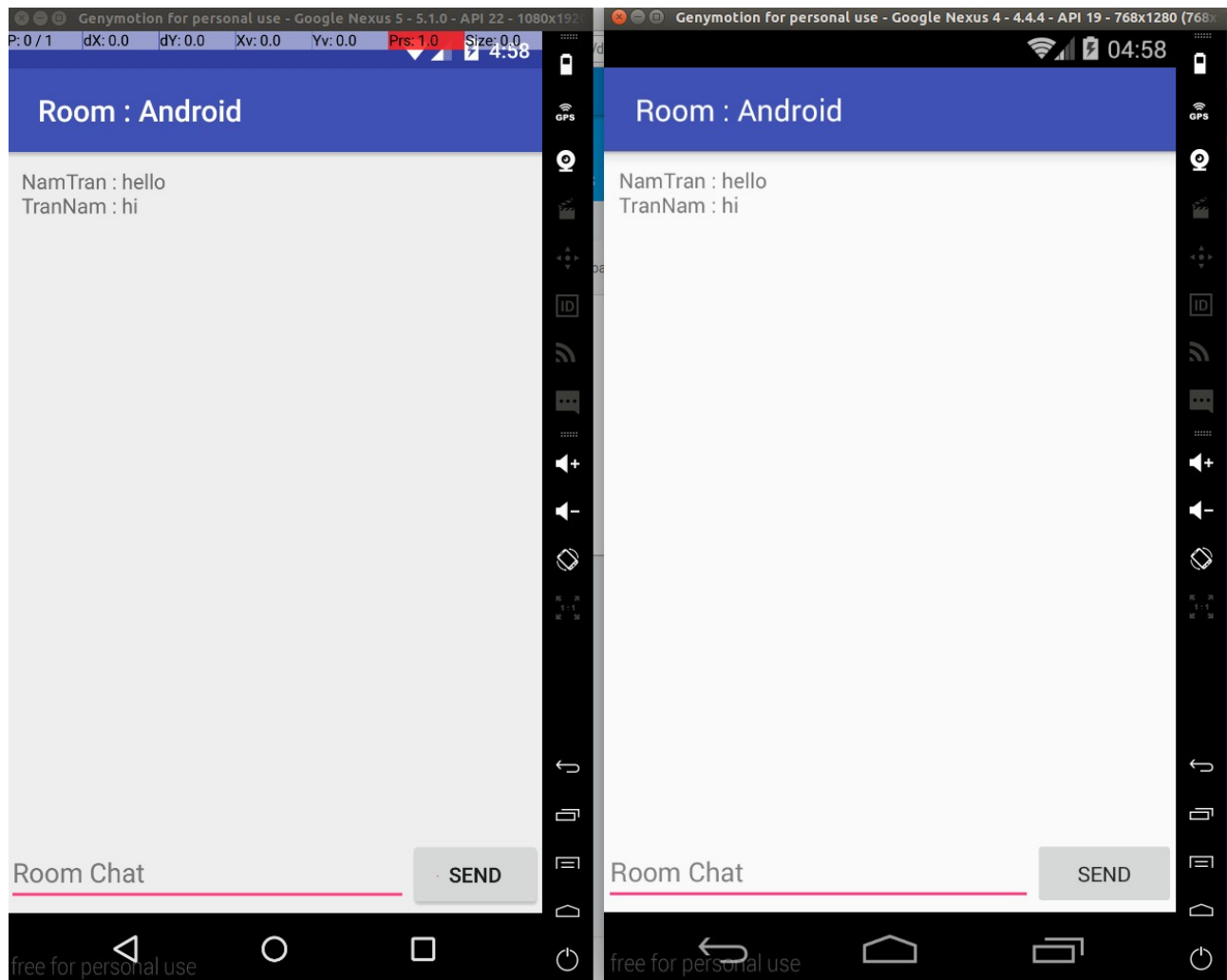
Cài đặt app trên 2 máy ảo . Máy ảo thứ nhất tạo phòng chat Android



Trên firebase database sẽ cập nhật ngay lập tức phòng chat



Sau đó cả hai máy ảo cùng vào phòng chat “Android” và chat với nhau



Trên firebase cũng sẽ cập nhật ngay dữ liệu

