

# ỨNG DỤNG GOOGLE MAP API VÀ GOOGLE FIREBASE XÂY DỰNG PHẦN MỀM GIÁM SÁT VÀ CẢNH BÁO TRƯỢT LỞ ĐẤT

**PHẠM THỊ THANH LOAN, NGUYỄN THẾ LỰC**  
*Trường Đại học Mỏ-Địa chất*  
**VÕ CẨM THỦY** - *Trường Đại học Công nghiệp Hà Nội*  
 Email: thanhloanbkhn@gmail.com

## 1. Mở đầu

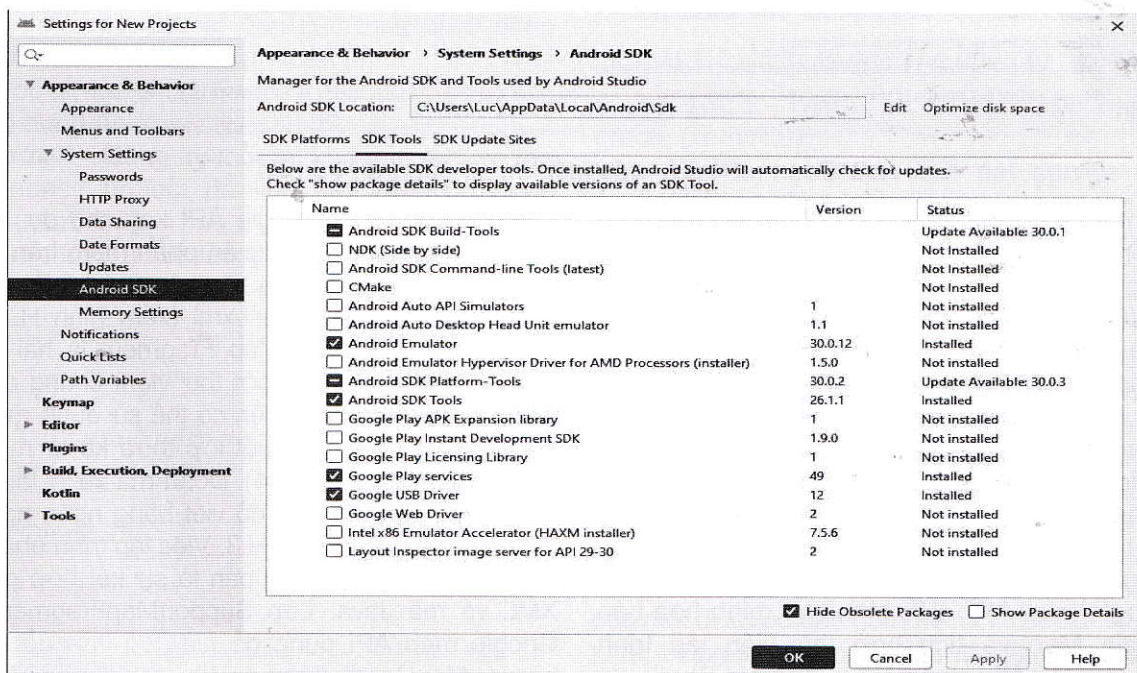
Trong những năm gần đây, các sự cố trượt lở đất thường xuyên xuất hiện tại các khu vực miền núi phía Bắc gây ảnh hưởng nghiêm trọng đến sản xuất nông, lâm nghiệp, các công trình công nghiệp, giao thông và cuộc sống sinh hoạt của người dân [1]. Thực trạng đó đặt ra một yêu cầu cấp bách trong việc xây dựng một hệ thống giám sát và cảnh báo trượt lở đất giúp cho nhà quản lý và người dân chủ động ứng phó trong các tình huống thiên tai có thể xảy ra.

Hệ thống giám sát các thông số nhiệt độ môi trường, độ ẩm đất, lượng mưa và cảnh báo sớm trượt lở đất dựa trên ứng dụng nền tảng Android của nhóm tác giả đã được thiết kế và phát triển thành công [2]. Hệ thống cho phép theo dõi các thông số về nhiệt độ, độ ẩm đất, lượng mưa... tại các vị trí có nguy cơ trượt lở dựa trên sự tích hợp Google map API. Hệ thống cũng

cho phép người dùng nhận được các thông tin cảnh báo trượt lở đất cung cấp bởi hệ thống phân tích dữ liệu dựa trên trí tuệ nhân tạo. Sản phẩm của dự án cung cấp thông tin chính xác và đảm bảo an toàn cho người dùng với sự trợ giúp của “Cơ sở dữ liệu thời gian thực-Firebase”. Các tin nhắn cảnh báo sẽ được tự động gửi tới thiết bị di động bằng cơ sở dữ liệu thời gian thực Firebase tích hợp trong phần mềm.

## 2. Tích hợp Google map API vào ứng dụng Android

Google map android API là một thư viện bản đồ được cung cấp bởi Google cho phép nhúng bản đồ số vào trong ứng dụng Android. Người dùng có thể xem các thông tin điểm đo, các dữ liệu đo theo thời gian thực tại các vị trí được cung cấp trên bản đồ. Các thông tin dưới dạng chữ viết, hình ảnh được gắn kèm với các marker trên bản đồ [3].



H.1. Tích hợp Google Play services SDK vào dự án Android

```

<permission
    android:name="com.example.gisclient.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />
<permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.example.gisclient.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AlzaSyBLC1k8LrhdaMbiMN84hTZocnumUUvu6jA" />
<uses-feature android:glEsVersion="0x00020000" android:required="true" />
<uses-feature android:name="android.hardware.camera"
    android:required="true" />

```

## H.2. Thiết đặt ứng dụng

```

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    //Get data from ViewModel to bind data to the google map
    mViewModel.getMeasurementLocations().observe(getViewLifecycleOwner(), (Observer) (measurementLocations) -> {
        //Add markers to map
        for(int i=0;i<measurementLocations.size();i++){
            MeasurementLocation mLocation=measurementLocations.get(i);
            LatLng location=new LatLng(mLocation.getLatitude(),mLocation.getLongitude());
            MarkerOptions markerOptions=new MarkerOptions();
            markerOptions.position(location);
            markerOptions.title(mLocation.getLocationName());
            markerOptions.icon(BitmapDescriptorFactory.fromBitmap(getMarkerBitmapFromView(mLocation.getMeasureValue())));
            Marker mk=mMap.addMarker(markerOptions);
            mMap.put(mk,mLocation.getLocationID());
        }
        //Custom Map view
        mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        mMap.setMyLocationEnabled(true);
        mMap.setOnMarkerClickListener(markerClickListener);
        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng( v: 22.3862, v1: 103.4819), v: 15.0f));
    });
}

```

## H.3. Đoạn mã xử lý việc hiển thị dữ liệu trên bản đồ

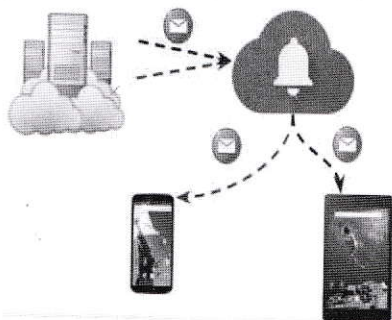
Các nhà phát triển có thể sử dụng Google map API phiên bản 2 nhúng vào một activity dưới dạng một fragment tạo bởi đoạn mã XML. Bản đồ Google (*Google Map*) cung cấp các tính năng hiển thị dữ liệu hình ảnh 3D, bản đồ vệ tinh, bản đồ địa hình hoặc bản đồ hybrid. Ngoài ra thư viện Google map api còn cung cấp các hiệu ứng và xử lý các tương tác cơ bản như phóng to thu nhỏ bản đồ, định vị tọa độ người dùng, tìm kiếm địa điểm... Ngoài các tính năng được cung cấp sẵn, nhà phát triển có thể tự tạo ra các tính năng riêng biệt cho từng ứng dụng sử dụng các công cụ hỗ trợ bởi thư viện Google map API. Các hình H.1 minh họa bản đồ Google Map và tích hợp thư viện lập trình Google map API trên phần mềm Android Studio để lập trình ứng dụng kết hợp công nghệ Web GIS.

Để nhận được Google API key cho dự án, nhà phát triển cần đăng ký dự án phần mềm trên Google Cloud Platform Console và thêm các thiết đặt cần thiết trong *AndroidManifest.xml* (xem hình H.2). *Map Fragment* và *Support Map Fragment* là hai lớp đối tượng được cung cấp bởi thư viện, chúng có chứa *GoogleMap*, nhà phát triển có thể sử dụng một trong hai lớp này, hoặc viết một lớp thừa kế từ một trong hai lớp đó (xem hình H.4). Đoạn mã xử lý tải bản đồ và gắn dữ liệu lên bản đồ viết bằng ngôn ngữ Java được thể hiện trên hình H.3. Việc lập trình phần mềm nền tảng Android về cơ bản là sự kết hợp giữa mã XML và mã Java. Lớp *SupportMapFragment* trong Google map API dùng để hiển thị bản đồ Google, hàm *onMapReady* được gọi khi ứng dụng hoàn thành việc tải bản đồ. Trong hàm này các tác giả thực hiện

việc xử lý thêm lớp dữ liệu hiển thị các thông số của các trạm đo tại các vị trí cụ thể được đánh dấu trên bản đồ (xem hình H.12).

```
<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_below="@id/layout_search"/>
```

H.4. Mã XML nhúng bản đồ Google vào ứng dụng Android



H.5. Hoạt động của Firebase Cloud Messaging

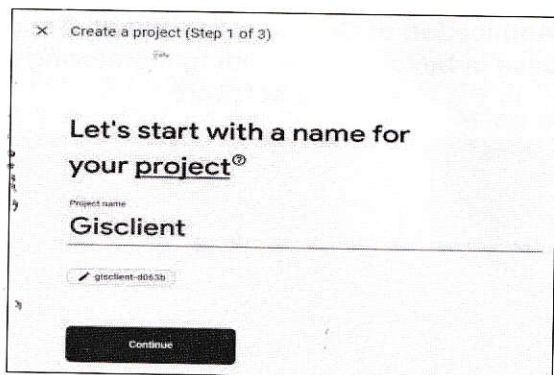
Trên hình H.5 mô tả hoạt động của dịch vụ tin nhắn cung cấp bởi Google. Hệ thống thông báo kết nối dữ liệu tin nhắn từ đám mây và phân phối đến thiết bị cầm tay của người dùng. Đây là một tính năng quan trọng và cần thiết cho ứng dụng cảnh báo trượt lở đất ở khu vực giám sát.

**3. Tích hợp Google Firebase Cloud Messaging**

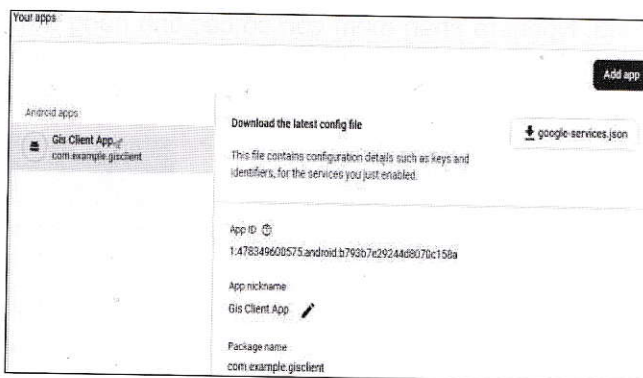
Firebase Cloud Messaging là một dịch vụ miễn phí của Google có chức năng gửi các thông báo hoặc tin nhắn từ Server đến các thiết bị di động một cách nhanh chóng và an toàn [4].

Các bước tích hợp Firebase Cloud Messaging vào ứng dụng Android như sau [5]:

- Bước 1 - Đăng ký một tài khoản Firebase tạo một Firebase project để liên kết với ứng dụng android (xem H.6);
- Bước 2 - Tải file Google-services.json sau đó thêm file cấu hình vào dự án android (xem hình H.8);



H.6. Giao diện Tạo Firebase project



H.7. Tải file Google-services.json

```
implementation 'com.google.android.gms:play-services-maps:17.0.0'
implementation 'com.google.firebase:firebase-analytics:17.4.3'
implementation 'com.google.firebase:firebase-messaging:20.2.1'
```

H.8. Cấu hình Firebase messaging plugin

```
<service android:name=".api.MyFirebaseMessagingService" android:exported="false" >
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
</service>
```

H.9. Cài đặt messaging service

```
@Override
public void onNewToken(String token) {
    Log.d(TAG, "msg: " + "Refreshed token: " + token);

    // If you want to send messages to this application instance or
    // manage this apps subscriptions on the server side, send the
    // Instance ID token to your app server.

    DatabaseHelper databaseHelper=new DatabaseHelper(StartActivity.context);
    UserAccount userAccount=databaseHelper.getLoggedInAccount();
    if(userAccount!=null){
        databaseHelper.updateFirebaseToken(userAccount.getUserID(), token);
    }else{
        databaseHelper.updateFirebaseToken( userID: 1,token);
    }
    sendRegistrationToServer(token);
}
```

H.10. Đăng ký nhận tin nhắn từ Firebase

- Bước 3 - Thêm các cài đặt vào file app/build.gradle (xem H.8);
- Bước 4 - Thêm thiết đặt service vào file AndroidManifest.xml (xem H.9);
- Bước 5 - Viết mã Java để cho phép thiết bị đăng ký và nhận tin nhắn từ Firebase (xem H.10).

**4. Kết quả đạt được**

Phần mềm giám sát và cảnh báo trượt lở đất với giao diện trực quan, kết hợp công nghệ Web GIS được xây dựng có tên gọi GIS Client sẽ sớm được nhóm tác giả đăng ký trên kho ứng dụng Play để mọi người có thể cài đặt và sử dụng. Hình H.11 là một số hình ảnh về phần mềm đã được xây dựng. Hình H.11.a là giao diện hiển thị dữ liệu điểm đo trên bản đồ. Hình H.11.b là giao diện hiển thị thông tin chi tiết của một điểm đo bao gồm hiển thị các dữ liệu đo mới nhất, thông tin nhà cung cấp dữ liệu đo và thống kê lịch sử dữ liệu đo. Hình H.11.c là danh mục các chức năng cài đặt ứng dụng, đăng nhập, đăng xuất ứng

dụng. Ngoài ra phần mềm còn có các tính năng hữu ích khác như: cho phép người dùng upload hình ảnh

tự chụp hay chức năng dành cho người quản lý, thống kê dữ liệu và xuất file báo cáo.



H.11. Phần mềm giám sát và cảnh báo nguy cơ trượt lở đất

5. Kết luận

Phần mềm giám sát và cảnh báo trượt lở đất được phát triển trên nền tảng hệ điều hành Android sử dụng các công cụ phát triển và thư viện được hỗ trợ bởi Google. Hai thư viện chính được sử dụng để xây dựng nên ứng dụng là Google map API và Firebase Cloud Messaging được cung cấp miễn phí bởi Google. Phần mềm ứng dụng góp phần quan trọng trong việc theo dõi và cảnh báo trượt lở đất tại một số khu vực của tỉnh Lai Châu. Thông tin về các giá trị đo được cập nhật nhanh chóng giúp người sử dụng lập kế hoạch sản xuất cũng như có những biện pháp phòng tránh chủ động trước các sự cố trượt lở có thể xảy ra. Ứng dụng sẽ được tiếp tục hoàn thiện và có thể áp dụng cho các địa bàn khác cũng như trong công nghiệp mỏ. □

TÀI LIỆU THAM KHẢO

1. Viện Khoa học Địa chất và Khoáng sản. “Điều tra, đánh giá và phân vùng cảnh báo nguy cơ trượt lở đất đá các vùng miền núi Việt Nam”. Đề tài cấp bộ. 2017.
2. Phạm Thị Thanh Loan, Nguyễn Thế Lực, “Thiết kế hệ thống giám sát, cảnh báo trượt lở đất dựa trên nền tảng Android”. Hội nghị khoa học môi trường địa chất-ERSD”. Trường Đại học Mở-Địa chất, 2020.
3. Google Map Platform documentation
4. Google Firebase Cloud Messaging documentation
5. Developer guides documentation online

Ngày nhận bài: 01/06/2020

Ngày gửi phản biện: 12/08/2020

Ngày nhận phản biện: 15/10/2020

Ngày chấp nhận đăng: 10/12/2020

Từ khóa: hệ thống giám sát, cảnh báo, trượt lở đất, google map API, Google firebase.

Trách nhiệm pháp lý của các tác giả bài báo: Các tác giả hoàn toàn chịu trách nhiệm về các số liệu, nội dung công bố trong bài báo theo Luật Báo chí Việt Nam.

Tóm tắt: Mục đích của nghiên cứu được giới thiệu trong bài báo này là đưa tin hiệu cảnh báo dưới dạng tin nhắn, âm thanh đến thiết bị di động của người dùng để giảm thiểu thiệt hại do thiên tai gây ra.

Application of Google map API and Google firebase in building land sliding monitoring and warning software

SUMMARY

The purpose of the research presented in this paper is to send warning signals in the form of messages and sounds to human mobile devices to minimize the damage caused by natural disasters.