

**RESEARCH PAPER**

Available Online at [www.jgrcs.info](http://www.jgrcs.info)

## SOCIAL NETWORKING IN SMARTPHONE THROUGH A PROTOTYPE IMPLEMENTATION USING ANDROID

Anil Kumar<sup>\*1</sup>, Prem Mithilesh.M<sup>2</sup>, Chandra Kiran.Y<sup>3</sup>, Vinay Gautam<sup>4</sup>, S Jaya Kumar<sup>12345</sup>  
Department of Computer Science, SRM University, Chennai, India  
<sup>1</sup>kumar.anil.342011@gmail.com, <sup>2</sup>premmithilesh@gmail.com, <sup>3</sup>chandrayelagam@gmail.com,  
<sup>4</sup>vinaygautam2010@gmail.com

**Abstract:** In the paper "Social Networking in Smartphone through a prototype implementation using Android", we are going to introduce and develop a system which brings user's across various corner's together, and let's them connect, share media, documents, and chat. The system is entirely based on Android O.S; Instant messaging service will be available for users through which a user can send and receive messages from one device to another. It track's the location of an user using GPS service and allows the user to share files of any format. The main objective of developing a social network application is to provide an easy accessibility and communication for users in a secure network. Thereby our aim is to provide them a room through which they can easily be in touch with their friends. This paper gives some basic concepts about Instant Messaging System and tracking location of a user using GPS service. It describes how the documents can be shared between two android devices.

**Keywords:** Android, Instant Messaging, Sharing, GPS Service.

### INTRODUCTION

Today, mobile is beyond a device that was merely used for talking to someone like in the past. It is a PC in pocket. A tablet or Smartphone installed with Android OS can be geared up with a variety of applications to provide a global connectivity. Smartphone's installed with Android OS just require an app to get connected with any social-networking services. They were designed to enhance flexibility, usability, and functionality of the communication system. In an instant messaging architecture, when a server process is running an user can send messages to other user's using web server on some unique port. According to Wikipedia pages, Android architecture consists of Linux kernels, libraries and APIs written in C and inbuilt java compatible libraries for developers and an inbuilt server, a free type of software and open source license, aspect that makes it very attractive among developers. Thus we are going to use this Android architecture to develop social networking application.

The Smartphone's are the major resources of the present system for speedy and smart activities. So we are going to develop a system which allows an user to chat, share data, media and documents.

- Instant Messaging: Transferring of messages from one device to the other instantly.
- Location Tracking: User's footprints are tracked and updated in real-time in Google maps.
- Sharing: Sharing of documents and media.

#### Features of proposed system:

- Being able to share files of larger data than existing applications.
- Allows the user to share files in any format and a wide range of data exchange is made possible.
- It allows you to share documents with your friends.  
Ex: PDF, text files, photos, videos.

- It keeps in tracks of the location of user's friends and notifies when the user's friends get close to the user.
- It allows the user to login with their Facebook account.
- Without downloading the documents the user can open the documents and view them within the app through cloud.
- User password will be in encrypted form in database which prevents the intruders to access the users account.

The rest of the paper is organised as follows: Section II describes the architecture of proposed system, security and existing client-server architecture. Section III discusses about the technologies used to implement this system. Section IV describes about the implementation using web server and various technologies. In the last section there will be conclusion and future work.

### SYSTEM ARCHITECTURE

In this client server architecture through which user interact with other user's through web server using the REST, SOAP and JSON web services which is shown in Figure 2.1. Client sends XML data using HTTP POST method with HTTP-REQUEST to the server in Figure 2.2, but before reaching the server this XML data will be converted to JSON format which is nothing but the KEY-VALUE pair and server receives the request using php GET method. After getting request, The server side script runs and checks for the data with the given request and if it exists, then it responses back using POST method and it converts JSON format to XML format at client side and with this HTTP-RESPONSE client get data using HTTP GET method.

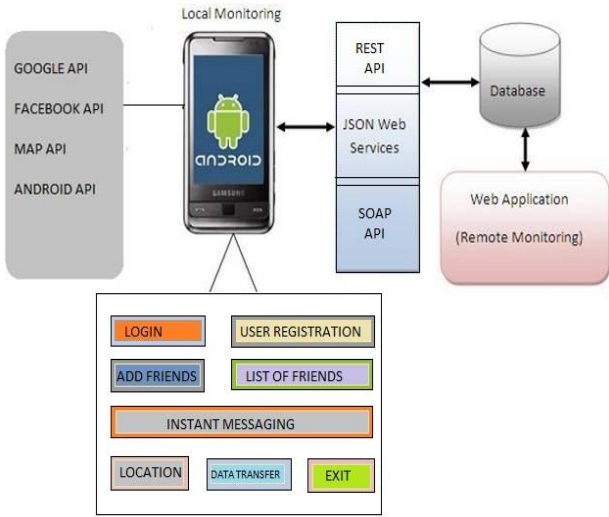


Figure 2.1 System Architecture

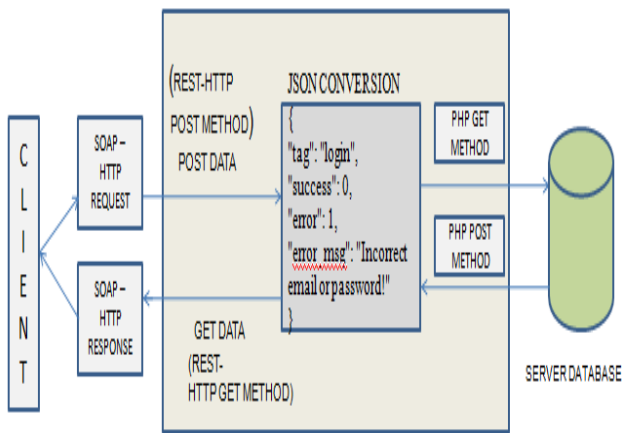


Figure 2.2 Client-Server Model

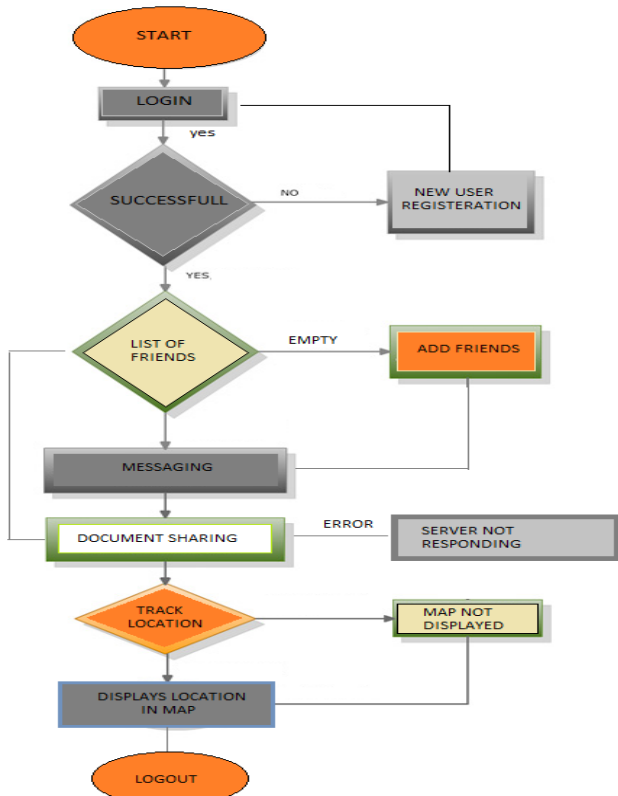


Figure 2.3 Working Flow Diagram

**TECHNOLOGIES USED:**

**Google map api:**

To access the Google Maps servers you have to add a GOOGLE Maps API key to your application which will be available on Google developer’s account. The key is free, you can use it with any of your applications that call the Maps API, and it supports an unlimited number of users. You can obtain a Map API key from the Google API’s Console by providing your application’s signing certificate and its package name. Add the key to AndroidManifest.xml file in your application.

**Facebook api:**

Facebook provides an API key that enable Facebook members to log onto third-party applications, mobile devices with their Facebook identity. For authenticating it uses OAuth2.0 authentication method for logging into facebook account. Beside the regular registration using the user-id our application users are able to login with their Facebook account

**Rest:**

To develop client-server network applications REST architecture will be used. REST represents REpresentational State Transfer. REST is easy to implement as compared to other architectures like SOAP, CORBA etc., it works on HTTP protocol and support most commonly used HTTP methods (GET, POST, PUT and DELETE)

- GET To fetch a resource
- POST To create a new resource
- PUT To update existing resource
- DELETE To delete a resource

**Soap:**

Simple Object Access Protocol is a way of transferring data between two applications by using the World Wide Web’s Hypertext Transfer Protocol and its Extensible Markup Language as information exchange mechanism. HTTP and XML provide an easy way of communication with each network. SOAP encodes an HTTP header and XML file so that a program in one device can call a program in another device and pass the information. It also explains the mechanism of http-request and response.

**OTHER ARCHITECTURES**

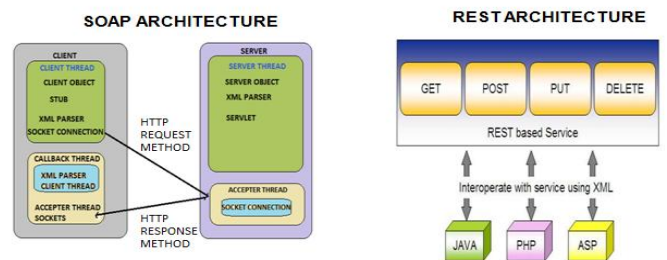


Figure 3.1 SOAP and REST Architectures

**MODULES**

**Login:**

Users will login into their account with registered user id and password. If not they can use their facebook account to login into application.

**Registration:**

For new users there will a registration process in which user has to create a new user id , password and enter his/her email id. After registration process user can go to login page to login.

**Adding of friends:**

Successful logged in users can add friends by entering their user name and send them request to add them.

**List of friends:**

When the approval of friend request then user name will be added to the list of members.

**Instant messaging:**

Users can send message to their friends when they are in online and even when they are in offline. Private chat and group chat will also be available for registered users.

**Document sharing:**

It allows the user's to share media and documents among friends like .doc files, .pdf, .txt files etc and media files.

**Tracking of location:**

Current location of users can be identified using GOOGLE maps. It tracks the location of all added members in the list.

**Authentication:**

Users login id details will be mailed to their mail id from server. It authenticates and restricts the user to transfer files of large size.

**Logout:**

Users can logout from the application.

**IMPLEMENTATION**

This section provides information on the software preferences and protocols used on the implementation of Social Networking system on an android development environment. The hardware platform chosen was Android SDK 2.2 ,ANDROID SDK 3.0,ANDROID SDK 4.0 and ANDROID SDK because applications developed on a lower target are compatible on higher devices too. To develop this Social Networking application we are using JAVA programming as client side and PHP on server side.For using inbuilt Android APIs and other packages we are using Eclipse for application development. At the initial stage of development of the application, a webserver was used for the testing of the authentication, login, registration and instant messaging aspect of the android messaging application.

For implementing web services in our project we are using JAX-WS(JAVA API for XML). JAX-WS is a widespread open source platform, which is an XML encoded protocol. JSON[14] (JavaScript Object Notation) was used to transport the messages from client to server via HTTP by converting XML data to JSON format which is nothing but KEY-VALUE pair. In order to work on mobile devices, Micro edition of Java was used and Android SDK (Software Development Kit) was installed and run the application from an AVD (Android Virtual Device). For testing the application a device having Wifi and GPS enabled on it, is used.

**Setting up of database:**

```
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(45) NOT NULL DEFAULT "",
  `password` varchar(32) NOT NULL DEFAULT "",
  `email` varchar(45) NOT NULL DEFAULT "",
  `date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `status` tinyint(3) unsigned NOT NULL DEFAULT '0',
  `authenticationTime` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `userKey` varchar(32) NOT NULL DEFAULT "",
  `IP` varchar(45) NOT NULL DEFAULT "",
  `port` int(10) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `Index_2` (`username`),
  KEY `Index_3` (`authenticationTime`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
AUTO_INCREMENT=15 ;
```

id	username	password	email	date	status	authenticationTime	userKey	IP	port
12	anil123	e10ad39430a5f8bbe89e05720803e	a@gmail.com	0000-00-00 00:00:00	0	2014-02-01 02:22:11		122.164.183.55	15145
13	icoder	827cc0eee8a706c4c34e1839184e7b	ai@gmail.com	0000-00-00 00:00:00	0	2014-01-29 10:44:02		122.164.183.53	15145
14	vinay	827cc0eee8a706c4c34e1839184e7b	vinaygautam2010@gmail.com	0000-00-00 00:00:00	0	2014-02-01 02:14:58		122.164.183.55	15145

Figure 5.1 Users Table

**For storing messages in database:**

```
CREATE TABLE IF NOT EXISTS `messages` (
  `id` int(255) NOT NULL AUTO_INCREMENT,
  `fromuid` int(255) NOT NULL,
  `toid` int(255) NOT NULL,
  `sentsdt` datetime NOT NULL,
  `read` tinyint(1) NOT NULL DEFAULT '0',
  `readdt` datetime DEFAULT NULL,
  `messagetext` longtext CHARACTER SET utf8
  COLLATE utf8_bin NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
AUTO_INCREMENT=34 ;
```

id	fromuid	toid	sentsdt	read	readdt	messagetext
22	12	13	2014-01-29 09:18:00	1	2014-01-29 09:22:00	hello anil123
23	12	13	2014-01-29 09:21:00	1	2014-01-29 09:22:00	hi icoder
24	12	13	2014-01-30 11:40:00	0		NULL hi dude
25	12	13	2014-02-01 02:01:00	0		NULL hii
26	14	12	2014-02-01 02:04:00	1	2014-02-01 02:06:00	hello
27	14	12	2014-02-01 02:05:00	1	2014-02-01 02:06:00	hello

Figure 5.2 Message Table

Encrypted Password – The password entered by user will be stored using md5 Hash method. This method is used to encrypt the password for security. When user entered his password the password will be compared to hash of user's password that stored in database.. It can be implemented by php code below

```
$password = isset($_REQUEST['password']) ?
md5($_REQUEST['password']) : NULL;
```

#### **Instant Messaging:**

Our application for Social Networking provides an user interface for sending messages to your friends who are registered on the app. The received messages can be read at any time by your friend when you are offline also. For sending messages from one device to other device we are using HTTP POST [2] method on client side and they are received using server side using php GET method at server. After receiving message at server your message will be delivered to the requested user using HTTP GET method and status will be set as one in database which means your message is received by your friend.

#### **Tracking location of a user:**

When user launches application his current location will be displayed to him and asks for his permission to send these details to the server for showing it friends. After storing his location lat, long [3] and address his location will be displayed to his friends. To get the current location of user create [7] LocationClient object, connect it to Location Services using connect() method, and then call its getLastLocation() method [6],[7]. To have location based functionality in your activity, you will have to implement two interfaces

- a. GooglePlayServicesClient.ConnectionCallbacks
- b. GooglePlayServicesClient.OnConnectionFailedListener

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version"
/>
```

With the Google Maps Android API, you can add maps based on Google Maps data to your application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. You can also use API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map.

For using Location services app must request permissions. Android has two location permissions: ACCESS\_COARSE\_LOCATION and ACCESS\_FINE\_LOCATION. The permission chosen controls the accuracy of the current location. Requesting ACCESS\_FINE\_LOCATION implies a request for ACCESS\_COARSE\_LOCATION. For example, to add ACCESS\_COARSE\_LOCATION, insert the following as a child element of the <manifest> element:

```
<uses-permission
    android:name="android.permission.ACCESS_COARSE_L
OCATION"/>
```

Current location of a user sent to web server database using post method in java and sending data to php. TO get the address from coordinates GEOCODER method is used which is defined in a package of google play services.

#### **Php code to get location of user from Latitude and longitude (server side)**

```
function getLocation($latitude, $longitude)
{
    $geolocation=file_get_contents('http://maps.googleapis.com
/maps/api/geocode/json?latlng='
    . $latitude.'.'.$longitude.'&sensor=false');
    $result= json_decode($geolocation);
    echo $result->results[0]->formatted_address;
}
```

#### **Document Sharing:**

1. To securely share a file from one device to another device. The Android FileProvider component which is available from android library generates content URIs for files, based on specifications provided in XML.

```
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="com.example.myapp.fileprovider"
        android:grantUriPermissions="true"
        android:exported="false">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/filepaths" />
</provider>
```

To send file from android to server first Create HttpURLConnection and using FILEINPUTSTREAM method get the path of the file. After getting the path then read the content of file byte by byte. Post the content of file to server using url of web server. [1] Set the property to multipart/form-data in order to send data of file and write the content of file using DATAOUTPUTSTREAM using buffer.

- a. Declare the variables for connection string and web server url.

```
uploadServerUri = "WEB SERVER URL ";
String fileName = sourceFileUri;
HttpURLConnection conn = null;
DataOutputStream dos = null;
```

- b. For sharing Image file .get the image path from default path and get data of image and convert them to bytes using buffer. Selected image will be decoded to bitmap type using BitmapFactory method.

```
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
```

```
if (requestCode == 1 && resultCode == RESULT_OK) {
    //Bitmap photo = (Bitmap) data.getData().getPath();
```

```
Uri selectedImageUri = data.getData();
```

```

    imagepath = getPath(selectedImageUri);
    Bitmap
bitmap=BitmapFactory.decodeFile(imagepath);
    imageview.setImageBitmap(bitmap);
    messageText.setText("Uploading file path:"
+imagepath);
}
}
public String getPath(Uri uri) {
    String[] projection = {
MediaStore.Images.Media.DATA };
    Cursor cursor = managedQuery(uri, projection,
null, null, null);
    int column_index =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media
.DATA);
    cursor.moveToFirst();
    return cursor.getString(column_index);
}

```

- c. For sending files like text,doc,docx,pdf and various other format ,get the source of files using FILE and send the file to webserver by opening HttpURLConnection .

File sourceFile = new File(sourceFileUri);

- d. sending file to web server  
try {

```

    // open a URL connection
    FileInputStream fileInputStream = new
FileInputStream(sourceFile);
    URL url = new URL(upLoadServerUri);

    // Open a HTTP connection to the URL
    conn = (HttpURLConnection)
url.openConnection();
    conn.setRequestMethod("POST");
    conn.setRequestProperty("ENCTYPE",
"multipart/form-data");
    conn.setRequestProperty("Content-Type",
"multipart/form-data;boundary=" + boundary);
    dos = new
DataOutputStream(conn.getOutputStream());

    dos.writeBytes("Content-Disposition: form-data;
name=\"uploaded_file\";filename=\""
+ fileName + "\" + lineEnd);

    dos.writeBytes(lineEnd);

    // read file and write it into form...
    bytesRead = fileInputStream.read(buffer, 0,
bufferSize);

    while (bytesRead > 0) {

        dos.write(buffer, 0, bufferSize);
        bytesAvailable = fileInputStream.available();
        bufferSize = Math.min(bytesAvailable,
maxBufferSize);
        bytesRead = fileInputStream.read(buffer, 0,
bufferSize);

```

```

    }
}
// send multipart form data necessary after file data...
dos.writeBytes(lineEnd);
// Responses from the server (code and message)
serverResponseCode =
conn.getResponseCode();
String serverResponseMessage =
conn.getResponseMessage();

```

- e. After sending file from Android we will store the file on our web server using server side scripting PHP language .[2] To submit files to a server, two methods of data transmission are well known, Post and Get methods. Get method transmit data directly through URL (universal resource locator) but is limited to ASCII characters and do not accept binary data. Post method open a direct connection to server of which type depends on supplementary parameters specifies. For binary data (as if pictures, archives, word or excel files) combined (or not) with regular ASCII data, always ENCTYPE='multipart/form-data',

Php code for storing files on server

```

<?php
$target_path = "./";
$target_path = $target_path . basename(
$_FILES['uploadedfile']['name']);
if(move_uploaded_file($_FILES['uploadedfile']['tmp_name']
,$target_path)) {
    echo "The file ". basename(
$_FILES['uploadedfile']['name']).
" has been uploaded";
}
?>

```

There are dozens of file sharing services that allow you to share files with your friends, but the process is often a long one because you have to upload a file, set security and visibility settings. IM clients are one way of transferring files, but again, if you're transferring files to someone at work, you might not have access to a messenger, or you may not use the same messenger service as your friend. For transferring files of large size we are using third party web application in our Android application which allow user to transfer the file of large amount. Using **JustBeamIt** [13] that allows file transfer. It offers simple Peer to Peer (P2P) file transfer that gives you a download link to share with anyone. The service creates a connection between you and whoever you share the link with so your file is not uploaded anywhere; rather, the link serves only as a gateway for transferring the file from application to the recipient.

## CONCLUSION

Our paper "Social Networking in Smartphone through a prototype implementation using Android" is an application development on Android platform. It is classified into three main phases i.e. Instant Messaging, Document Sharing and tracking of location which is a client server architecture using Json, Rest and Soap Technologies. The document/ media/ files sharing work as the user sends a document to their friends and the files get uploaded to the server and the other end user get a link automatically generated by the server and the user will be able to view the document through downloading it. This application uses Google Map

API and get's the user tagged along the map through his footprints and lets the user's friend view the real time updated foot prints of the user's friend.

### FUTURE WORK

The Future work on the paper would be developing the application so that the documents shared between the user would be viewed on cloud without actually downloading it. The future work also concentrates on encrypting the messages and data exchanged between user's and improve the security of the Application and maintain user privacy.

### ACKNOWLEDGEMENT

The paper was developed under the supervision of Dr. J.Jagadeesan, M.Tech(Ph.D) A.P(S.G)&HOD, Mr. Augustian Issac.R M.E.,A.P(Sr.G) and guidance of Mr. S.Jaya Kumar M.E A.P(O.G).The Authors thank their support, navigation and help at different stages of the project.

### REFERENCES

- [1]. Upload Files to Servers. Case Study and Implementation By Lorentz JÄNTSCHI/issn 1583-0233
- [2]. <http://getablogger.blogspot.in/2008/01/android-how-to-post-file-to-php-server.html>
- [3]. <http://googlesystem.blogspot.in/2009/02/google-latitude-share-your-locations.html> /February 4, 2009 Google Latitude: Share Your Location with Friends

- [4]. <http://google.about.com/od/mapsanddirections/tp/10-Quick-Google-Maps-Tricks.htm/10> Quick Google Maps Tricks By Marziah Karch
- [5]. <http://developer.android.com/guide/topics/location/strategies.html>
- [6]. <https://developer.android.com/training/location/retrieve-current.html>
- [7]. [http://www.tutorialspoint.com/android/android\\_location\\_based\\_services.htm](http://www.tutorialspoint.com/android/android_location_based_services.htm)
- [8]. [http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)
- [9]. [http://en.wikipedia.org/wiki/SOAP\\_with\\_Attachments\\_API\\_for\\_Java](http://en.wikipedia.org/wiki/SOAP_with_Attachments_API_for_Java)
- [10]. <https://restful-api-design.readthedocs.org/en/latest/methods.html>
- [11]. <https://developers.facebook.com/docs/reference/dialogs/oauth/>
- [12]. <https://developers.google.com/maps/documentation/imagemapapis/>
- [13]. <http://www.maketecheasier.com/use-justbeemit-for-easy-single-file-transfer/>
- [14]. <http://en.wikipedia.org/wiki/JSON>

### Short Bio Data for the Authors Author:



**Anil Kumar**<sup>\*1</sup>: I am Anil kumar, an Independent Web Developer. I am Interested in Web Application Development and Android Programming.