Active Monitoring System for Android Devices using Cloud

[1]Sagar Bhalodia, [2]K. Navin
[1]Student, [2]Assistant Professor
Department of Information Technology, Faculty of Engineering and Technology, SRM University, Chennai, India
[1]snbhalodia@gmail.com, [2]navin.k@ktr.srmuniv.ac.in

***Abstract*— the idea of tracking android devices is explained and implemented in this paper using the cloud service. The android devices can be tracked and monitored on the basis of call logs, call duration whether incoming, outgoing or missed calls and text messages. With the help of GPS (Global Positioning System) technology the location of the user device can also be monitored and traced according to their geographical location on the basis of longitude-latitude coordinates. This system is designed specially on android devices as it is one of the most popular smart phone operating system developed using Java programming technology. For the ease, the detailed information about the device is stored on the Cloud. Cloud storage is used over here to access the details about the device globally, whenever one wants to view the details and the information will be permanently stored. This android based active monitoring mobile application using cloud technology can come out to be very useful in the institutions like colleges, big companies or hospitals for keeping track on students, teachers; companies and banks to keep a track on their employees. By the use of this system unnecessary us of the android mobile device can be controlled and restricted. For certain restrictions parents, managers or admins of the system can even block the numbers i.e. both incoming and outgoing calls to stop the unnecessary phone calls during particular time. This system is more useful as one can keep monitoring and tracking remotely.**

***Index Terms*— Keywords- Android, GPS, cloud, monitoring, location, Eclipse.**

I. INTRODUCTION

Android is one of the world's most popular and widely used operating system. Google, who developed this mobile platform kept it as an open source operating system whose source code is available for development and modification of any application according to open source license agreement to improve the usability, functionality and performance of the application. From the developer's perspective, android is a Linux based operating system for developing applications for smart phones and tablets.

The Android framework is licensed under the Apache License, with android application development, having the right to distribute their application under their license.

The bottom layer of the Android framework is the Linux kernel which provides with the basic system functionality like process management, memory management, device management like camera, keypad, display, etc. Linux is considered good for networking and support to device drivers which gives easy interface to peripheral hardware. On top of the Linux kernel libraries used by android application like web kit, libc and SQLite database is present which the default database for android operating system is. Android used Dalvik Virtual Machine (DVM) like java uses Java Virtual Machine (JVM) for converting the java class files to .dex files for the application to run on the Android devices. The Application Framework layers provides high level services to the android applications which are java classes.

The top layer is the Application which a user can install and run in the device and which is the only visible layer to the users e.g. games, contacts, web browser which is an android package (.apk) file. So this is the overall architecture of android Framework shown in the figure.

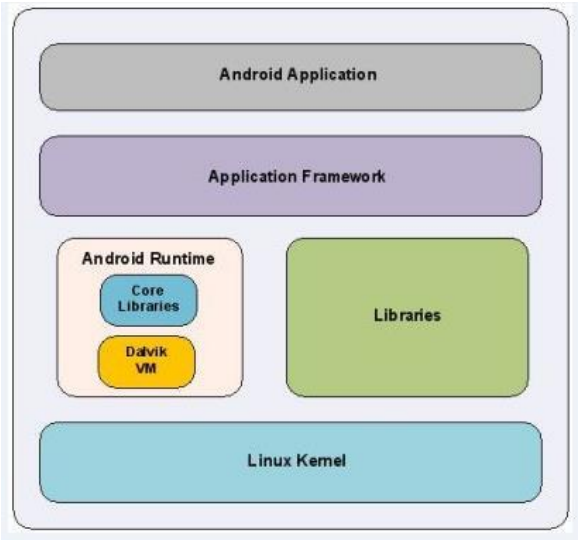

Fig. 1: Android Framework

II. EXISTING SYSTEM

The existing system is implemented with the help of the Bluetooth and wireless LAN by installing tags at various points to track the location of the user. Here, the system is not secure as a whole and the communication is done through the wireless LAN. The tracking can be generally done for a limited area because the range of Bluetooth is not high. Moreover the data stored and accessed is over a server which is not always easily accessible. In the existing system the activities of the user device can't be traced on the basis of text messages, calls and current location. To overcome all this drawbacks the proposed system is designed and implemented.

III. PROPOSED SYSTEM

In the proposed system we have implemented that the client user can be tracked by the admin with the details like call logs, call duration, recent incoming, outgoing, and missed calls with the current location of the person. This is possible with the telephony manager and GPS technology which locates the position of the user device where a GPS chip is inserted in the device which will send the data to the satellite in the form of longitude and latitudes coordinates and we are able to trace the current location or position of the user device or person with the help of Google Maps. This is made fast with the help of 3G connectivity for sending and retrieving the data to and fro in a high speed. Over here we have used Cloud as a centralized server where all the details of the device can be stored easily and can be viewed later anytime, anywhere from a personal computer. The monitoring system is well secured with the help of Web Security Service (WSS). Thus the encryption algorithm is used to send the messages between the client device and the admin. The phone details will be retrieved with the help of SOAP (Simple Object Access Protocol) protocol from the user's device.

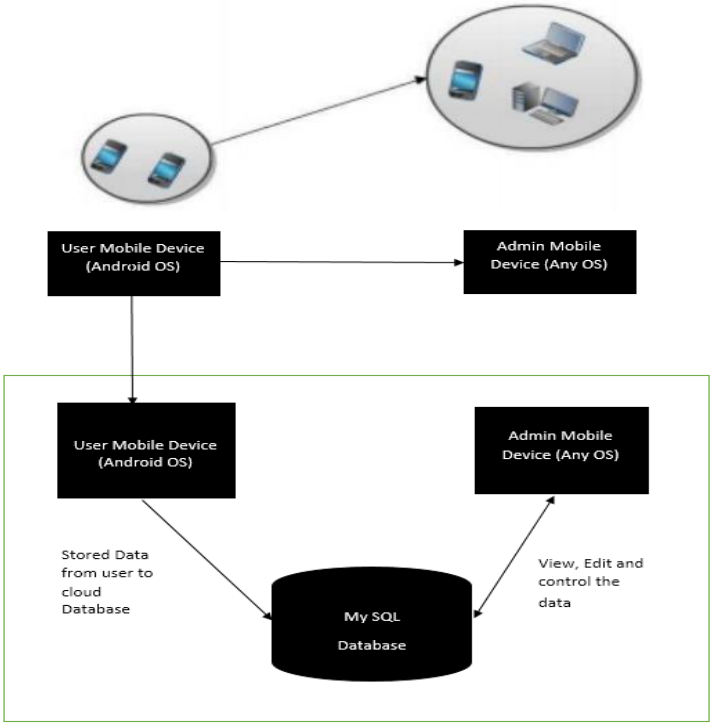

Fig. 2: Architecture of Proposed System

As shown in the figure is the architecture of the monitoring system over cloud. Here the user device should have android OS and the admin device can have any OS. The calls made and the user location can be tracked and stored on the cloud database so that the admin can view and edit anytime he wants. This is a portable monitoring system where the tracking can be done from anywhere irrespective of the physical location; authorized person can access and view the user details from the cloud.

IV. IMPLEMENTATION

The android application is a package of .apk file where java programming language is used to develop the application. The online monitoring system for tracking the android devices is completely a new generation tracking/monitoring system with the help of the cloud technology where two applications are designed and developed. One is installed in the user device to retrieve all the information like calls, messages and location; and the other in the admin device to view the traced calls, messages and location of the same. Here the details of the user device is fetched with the help of IMEI (International Mobile Equipment Identity) number which is permanent for any device and can be traced through it. Even if the SIM (Subscriber's Identity Module) card of the user device gets changed, the user can still be traced as the IMEI number doesn't change and can't be modified by anyone. The call information is retrieved by the telephony manager from which the contacts of user are fetched and stored on the cloud with the help of SOAP protocol which is used to exchange the information through the network. With the help of XML in message format, the data is transmitted by HTTP or SMTP from device to cloud. SOAP allows any programming model to transmit messages over SMTP, HTTP or FTP providing a message framework over the secure web services. The envelope of the SOAP identifies the XML document as a SOAP message, the header part of the SOAP contains the header information of the message and the body part contains call – response information. The current location of the user can be tracked with the help of GPS technology by their

longitude – latitude coordinates and Google Maps which shows the exact position of the user with the help of satellites. Users can be traced if they go beyond the particular area or location and an alert message will be sent automatically by the system to the admin device. The same alert message is sent if the user calls or receives a blocked number so that they can't make the unnecessary use of the mobile and can be restricted in an organization, institute, hospitals, etc. New feature added in the monitoring system is that the admin can control the calls and messages sent by the user by blocking a particular number into the user device so that they can't call or receive the incoming and outgoing call, messages from a particular number. By this way the user can be fully controlled, tracked and monitored remotely anywhere, anytime with the help of this cloud based monitoring system and can view and edit later from the cloud storage. Here the data transfer and communication is done with the 3G technology so no delay in sending and receiving the message is observed.

The cloud computing technology is used in this application for its various advantages over a remote server or a computer as it maximizes the effectiveness of the shared resources, low cost, use on demand, always accessible, low maintenance, high performance and virtualization and secured. No data is lost or redundant with new generation cloud technology. Every big organization have started working with the cloud technology because of their advantages like cost effectiveness, convenience and continuous available, scalable, easy deployment and integration, location independence and high storage capacity.

V. ANALYSIS AND DESIGN OF THE SYSTEM

A. Hardware and Software:

The platform development and design for the application system is done with the help of Eclipse IDE installed with JDK 6. The Android SDK (Software Development Kit) and ADT (Android Development Toolkit) bundle is installed for testing and running android application in the AVD (Android Virtual Device) called as emulator which is same as the android device. Then the required packages for the development of the system are installed necessary for the application to run with the help of cloud technology. The android version used in this system is targeted to android 2.3.6 Ginger Bread so that it works on every android device. The system is developed on Windows 7 operating system with 4 GB RAM and 2.40 GHz i3 Intel processor.

B. Functional Requirements:

- User Registration: Users should register with their user name, password, mobile number, email id, etc.
- User Login: The user should login with the authenticated username and password to successful login the system.
- Location: In this function the admin can view the current location of the registered user.
- Call Logs: Here, the admin can view all the call history in the form of call logs of the incoming, outgoing and the missed calls with date and time duration.
- Blocked Numbers: Here the admin can restrict a particular user by blocking the number.

C. Overall Design:

Here, in this system the structure of the system is divided into three parts: Admin, Client and Database. Here the admin can monitor and track on the basis of the user device IMEI number and can view call logs, location and edit blocking number module. The client user has only to install the client application into their android mobile device for getting monitored. Database is deployed on cloud can store the files in the text format in the form of table.

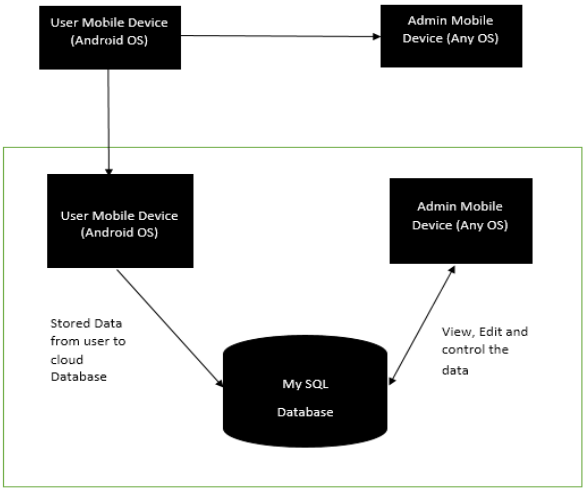

Fig. 3: Architecture of Proposed System

VI. DETAILED DESIGN OF MONITORING SYSTEM

A. Client Design:

In order to track the user device, the client application is installed in the user android mobile device to track them. The application will automatically fetch the IMEI number of the device and will show all the details about call logs and current location.

B. Admin Design:

Here the admins who wants to keep a track on the users have to register before getting the view of the system. Only the authorized and authenticated admins can login in to the system and can monitor and track the system with their secure user ID and password. After logging into the system they can view call logs of the particular users through their IMEI numbers and can know their current location and manage the call numbers by blocking them. The coding for this module is shown below.

- Registration Activity: In this module the admin has to register with their username, password, email-id, phone number to login into the system at the later stage to view the details of the user.

```
package andr.develop.mobiletracker;

import org.ksoap2.SoapEnvelope;

public class RegisterActivity extends Activity
{
public void call()
{ try {
String SOAP_ACTION = "http://tempuri.org/regService";
String METHOD_NAME = "regService";
String NAMESPACE = "http://tempuri.org/";
String URL = "http://cyberstudents.in/android/mob_track/Service.asmx";

SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
request.addProperty("name",txtbox_reg_user.getText().toString());
request.addProperty("pass",txtbox_reg_pass.getText().toString());
request.addProperty("phone",txtbox_reg_phone.getText().toString());
request.addProperty("mail",txtbox_reg_mail.getText().toString());
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.dotNet = true;
envelope.setOutputSoapObject(request);
HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
//HttpTransportSE androidHttpTransport = new HttpTransportSE(URL,6000);
androidHttpTransport.call(SOAP_ACTION, envelope);
Object result = (Object) envelope.getResponse();
```

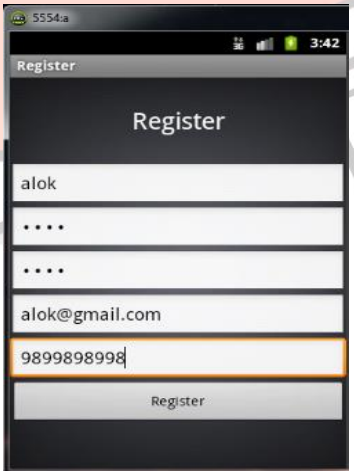

Fig. 4: Register

- *Login Activity:* In this activity the registered admins can login into the system and can view and monitor on the users by location, call logs. The code for this activity is as follows :

```
package andr.develop.mobiletracker;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;
```

```
btn_login=(Button) findViewById(R.id.btn_login);
btn_login.setOnClickListener(new OnClickListener()
{ @Override
public void onClick(View v)
{ if(txtbox_login_user.getText().length()!=0 && txtbox_login_pass.getText().length()!=0)
{
public void call() {
try {
String SOAP_ACTION = "http://tempuri.org/loginService";
String METHOD_NAME = "loginService";
String NAMESPACE = "http://tempuri.org/";
String URL = "http://cyberstudents.in/android/mob_track/Service.asmx";

SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
request.addProperty("name",txtbox_login_user.getText().toString());
request.addProperty("pass",txtbox_login_pass.getText().toString());
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.dotNet = true;
envelope.setOutputSoapObject(request);
HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
androidHttpTransport.call(SOAP_ACTION, envelope);
Object result = (Object) envelope.getResponse();
if(result.toString().equals("Login Success"))
{ handler.post(new Runnable()
{
@Override
```

Fig. 5: Login

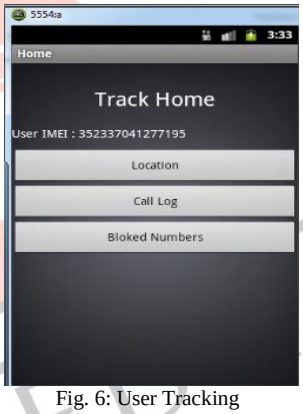

Fig. 6: User Tracking

- *Tracking Call Log Activity:* After logging into the monitoring system, admin can view the details of each and every user on the basis of IMEI number and can view the call logs of the user with the call duration, call type and call timings, etc.

```
package andr.develop.mobiletracker;

import java.util.ArrayList;
import org.ksoap2.SoapEnvelope;

android.R.layout.simple_list_item_1, R.id.tvNameMain, conNames));
AsyncCallWS_list task = new AsyncCallWS_list();
task.execute(); }
private class MyAdapter extends ArrayAdapter<String>
{ public MyAdapter(Context context, int resource, int textViewResourceId, ArrayList<String> conNames)
{ super(context, resource, textViewResourceId, conNames); }
@Override
public View getView(int position, View convertView, ViewGroup parent) {
View row = setList(position, parent);
return row; }
private View setList(int position, ViewGroup parent) {
LayoutInflater inf = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```
public void call()
{ try {
/** Called when the activity is first created. */
String SOAP_ACTION = "http://tempuri.org/getcalllogService";
String METHOD_NAME = "getcalllogService";
String NAMESPACE = "http://tempuri.org/";
String URL = "http://cyberstudents.in/android/mob_track/Service.asmx";
SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME); Void
request.addProperty("imei",uid );
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.dotNet = true;
envelope.setOutputSoapObject(request);
HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
//HttpTransportSE androidHttpTransport = new HttpTransportSE(URL,6000);
androidHttpTransport.call(SOAP_ACTION, envelope);

SoapObject res= (SoapObject) envelope.getResponse();
StringBuffer str_buf=new StringBuffer();
for(int i=0;i<res.getPropertyCount();i++)
{ SoapObject res_in=(SoapObject) res.getProperty(i);
//str_buf.append(res_in.getProperty("Name")+","+res_in.getProperty("ID")+"\n");
conNames.add(res_in.getPropertyAsString("d_contact"));
conNumbers.add(res_in.getPropertyAsString("d_num"));
conTime.add(res_in.getPropertyAsString("d_call_duration"));
conDate.add(res_in.getPropertyAsString("d_call_time"));
conType.add(res_in.getPropertyAsString("d_call_type")); }
//tv.setText(str_buf.toString()); }
catch (Exception e) {
//tv.setText(e.getMessage());
e.printStackTrace(); }  } }
```

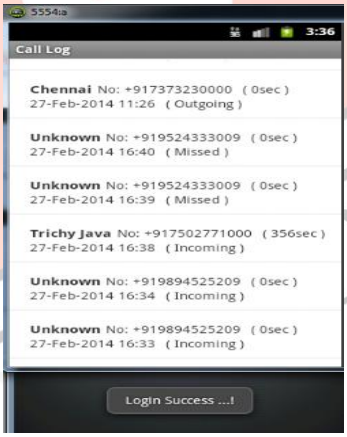

Fig. 7: Call Tracking

VII. FUTURE ENHANCEMENTS

In future, the application can be extended by adding features like keeping a track on the multimedia messages and can store the recordings of incoming and outgoing calls i.e. audio and video calls made by the user with the accurate time span. The audio and video format files should be easily stored on the cloud for the future access. Moreover the GPS locating can also be improved by providing the accurate position of the user device in the specific area. A limited accessibility can be given to the users as per their need and use according to the organization. It can be developed for the cross platform operating systems like IOS, Windows and Blackberry.

VIII. CONCLUSION

The Active Monitoring System for Android devices with the help of cloud technology is implemented successfully to monitor the users irrespective of any physical location. The application is working smooth with the use of cloud storage and the user events are automatically stored into the cloud to make the task easy. As the application is only designed to work on android devices, its usability and functionality can be improved in the future.

IX. REFERENCES

[1] Android Developers - Advanced Training. http://developer.android.com/training/cloudsync.html.
[2] Atushi Ito, Yoshiaki Kakuda Tomoyuki Ohta and Shinji Inoue, "New safety support system for children on school routes using mobile ad hoc networks", IEICE Transaction.
[3] Reto Meier , " Professional Android Application Development" Wrox publication.
[4] http://www.theappguruz.com/blog/android-call-logs-code-sample/
[5] Muhammad Fahim Iram Fatima, Sungyoung Lee, Young-Koo Lee,"Daily life /activity Tracking Application for Smart Homes using Android Smartphone"