**Android application For free Text Messaging service**

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY-INFORMATION TECHNOLOGY(B.TECH-IT)**

**Submitted by**

**Name of the student: M.A. MOHIDEEN ABDUL KHADER**

**Reg:No:U14IT010**



**Under the Guidance of**

**Mrs.A.RAMA**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**BHARATH UNIVERSITY (Declared Under Section 3 of UGC Act, 1956) Selaiyur, Chennai -600 073. TamilNadu, India.**

**MAY 2017**



**DEPARTMENT OF COMPUTER APPLICATIONS**

**BHARATH UNIVERSITY**

**(Declared Under Section 3 of UGC Act, 1956) Selaiyur, Chennai -600 073, Tamil Nadu, India.**

Certified that this project report titled **“Message Communication Using  
Android Application”** is the bonafide work of **“M.A.MOHIDEEN ABDUL KHADER(U14IT010)”** of the third Year B.tech-Information technology who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of my other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

This dissertation is submitted in partial fulfillment of the requirements for the award of the degree of department of information technology during academic year 2014 -2017.

**Mrs.A.RAMA**  **Dr.KUMARAVEL**

**Project Guide Head of the Department**

**Submitted for the Viva-Voce Examination held on ………………………………………… at Bharath University.**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMET**

First I wish to thank the Almighty who gave me good health and success throughout my project work. I thank my parents without whom nothing would have been possible this support was the foundation of my efforts.

I wish to express my heart full thanks to our **Honorable Founder Dr.S. Jagathrakshagan,** **Chancellor Dr.J.Sundeep Aanand**, **Bharath University**, for this encouragement that we are receiving our academic career.

I devote my thanks to esteemed **Dr.V.Kanagasabai Vice-Chancellor., Dr.M.Ponnavaiko Pro.Chancellor., Dr.K.P.Thooyamani Pro.Vice-chancellor and Dr.J. Hameed Hussain, Dean- Engineering, Bharath University.**

I devote my thanks to esteemed “**MR.A.RAMA” Department of Information Technology** for their encouragement and the valuable guidance throughout the project work.

I express my sincere thanks to all the Third-Year IT students by your own capability, efficient and worthy guidance, which was quite encouraging and support throughout the project to get it successful.

I also thank people who directly or indirectly gave me encouragement and support throughout the project. Last but not least, I offer my cardinal thanks to my friends, and other staff members and well-wishers for their continuous encouragement throughout my career.

**M.A.MOHIDEEN ABDUL KHADER**



**DECLARATION**

I hereby declare that this project report entitled **“Message Communication Using Android Application”** embodies the report of my project work carried out during Final Semester having Main Project Phase –II in Department of Information Technology under the guidance of **Mrs.A.RAMA** and has not been submitted previously for the award of other degrees by me or others to any institute or any university to the best of my knowledge.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **DESCRIPTION** | **PAGE NO** |
| **1.** | **INTRODUCTION**  **1.1. Abstract**  **1.2. Objectives of the project** |  |
| **2.** | **SYSTEM CONFIGURATION**  **2.1. Hardware Requirements**  **2.2. Software Requirements**  **2.3. Gantt Chart** |  |
| **3.** | **SYSTEM ANLAYSIS**  **3.1. Existing System**  **3.2. Proposed System** |  |
| **4.** | **SYSTEM DESIGN**  **4.1. Architectural Design Specification**    **4.2. Form Design** |  |
| **5.** | **SYSTEM TESTING**  **5.1. Logical Testing**  **5.2. Functional Testing**  **5.3. System Testing** |  |
| **6.** | **SYSTEM IMPLEMENTATION** |  |
| **7.** | **SYSTEM MAINTENANCE** |  |
| **8.** | **APPENDIX**  **8.1. Screen shots**  **8.2. Source Code**  **8.3. Dataflow Diagram** |  |
| **9.** | **CONCLUSION** |  |
| **10.** | **FUTURE ENHANCEMENTS** |  |
| **11.** | **BIBLIOGRAPHY** |  |

**1. INTRODUCTION**

**1.1. Abstract**

In this project, android message communication between various phones is implemented. The fundamental problem of key agreement among the parties communicating over an in Message public network. Over the years, a number of solutions to this problem have been proposed with varying degrees of complexity. However, there seems to have been no previous systematic look at the problem of key agreement over combined phone consisting of low-power mobile devices. It is efficient key agreement scheme well suited for this networking environment. Key agreement is intuitively simple, and yet offers a scalable solution to the problem.

The project” android Message communication” contains of two module. They are

1. Sender

2. Receiver

**1.2. Project Description**

In this project, the Message communication between the sender and receiver trout Phones messaging process to encrypt the message providing Message key agreement. Communication complexity is quantified as both the number of rounds of communication among users and the number of messages sent/received by users, while computational complexity is mostly concerned with the number of public-key cryptography operations that users have to perform.

Although mobile devices represent an already large and growing percentage of the computing population, security is still a major gating factor for their full adoption. Despite all the work conducted over many decades, the implementation of strong protection in a mobile environment is non-trivial. Security solutions targeted for more traditional networks are often networks due to a marked difference in computing resource.

In contributory key agreement protocols, a correctly behaving party is assured that as long as his contribution is chosen at random, even a coalition of all other parties will not be able to have any means of controlling the final value of the session key.

Moreover, most key transport protocols are focused on minimizing the cost of the rekeying operations associated with group updates, lack at least one of the important security properties: perfect forward secrecy or known key security.

**2. SYSTEM CONFIGURATION**

**2.1. Hardware Requirements**

  Processor       :   Intel Core i3/Amd A4 Processor

  RAM            :  4 GB

  Hard disk        :  120GB or more

  Monitor         :  10inch Touch monitor

**2.2. Software Requirements**

* Operating System  :  Android Jellybean os& Above version
* Technology Used     : Android studio 2.3.0 For Windows, The Official IDE for Android.
* Database               :  SQLite
* Environment : The Official IDE for Android

**Client:**

* Operating System :  Android Jellybean os& Above version

**3. SYSTEM ANLAYSIS**

Communication over a phone is one field where this tool finds wide ranging application. Chat application establishes a connection between 2 connected over the mobile networks. This tool can be used for large scale communication and conferencing in an organization or campus of vast size, thus increasing the standard of security. In addition it converts the complex concept of sockets to a user friendly environment. This software can have further potentials, such as file transfer and voice chatting options that can be worked upon later.

**3.1. Existing System**

Sometimes, the confidential information like password, pass code, banking details and private identity to our friends, family members and service providers through an SMS. But the traditional SMS service offered by various mobile operators surprisingly does not provide information security of the message being sent over the network. In order to protect such confidential information, it is strongly required to provide end-to-end Message communication between end users. SMS usage is threatened with security concerns, such as we present the related work on SMS Spam filtering problem.SMS disclosure, man-in-the-middle attack, replay attack and impersonation attack. There are some more issues related to the open functionality of SMS which can incapacitate all voice communications in a metropolitan area, and SMS-based mobile bottelnet as Android bottlnet . SMS messages are transmitted as plaintext between mobile user (MS) and the SMS center (SMSC) SMS contents are stored in the systems of network operators and can be read by their personnel.

**3.2. Proposed System**

**Aspects**:

Firstly, the solution’s ability to provide peer- to-peer SMS security. Secondly, the security services which are provided by the solution, in this project give attention to four main security services; confidentiality, integrity, authentication and non-repudiation. In its usual ORM(object-oriented systems), algebraic cryptanalysis only requires one plain text /cipher text pair.

**Attack model** :

An attack model describes different scenarios for the possibilities of various attacks where a malicious MS can access the authentic information, or misguide the legitimate MS. Since, the SMS is sent as plaintext, thus network operators can easily access the content of SMS during the transmission at SMSC. This leads to SMS disclosure attack. The attacker may fraudulently delay the conversation between both Sender and Receiver and can capture or reuse the authenticated information (during the protocol execution) contain in previous messages which results in the form of replay attack. Later, the attacker may send the captured information to the server or token. An attacker can also perform a man-in- the-middle attack when an MS (Mobile service) is connected to a BTS through wireless network and eavesdrops the session In order to overcome the above stated attacks, various cipher algorithms are implemented with the proposed authentication protocol. The recommend that the cipher algorithms should be stored on to the SIM (part of MS) as well as in the AS. Since providing security needs to do some extra effort which is measured in terms of cost, thus providing or adding extra security means increasing more cost. Here it includes one more service as ‘Message Message’ in the menu.

**Protocol:**

In this section, it is propose d that a new protocol named SMS with two different scenarios which provide end-to-end Message transmission of information in the cellular networks. First scenario is illustrated in where both MS belong to the same AS, in other words share the same Home Location Register (HLR) while the second scenario is presented. In non-server architecture mobile security systems, all the cryptographic operations will be achieved in the user mobile phone device. This section focuses on the attack model, system and communication model, basic assumption and initiated by legitimate MS.

The above requirements can be accomplished by proposing a protocol called Easy SMS which provides end-to-end security during the transmission of SMS over the network. The Easy SMS protocol prevents the SMS information from various attacks including SMS disclosure, over the air (OTA) modification, replay attack, man-in-the-middle attack, and impersonation attack. This Easy SMS sends lesser number of transmitted bits, generates less computation overhead, and reduces bandwidth consumption and message exchanged as compare to SMS Seconds.

**4. SYSTEM DESIGN**

**4.1. Block Diagram for System Process**

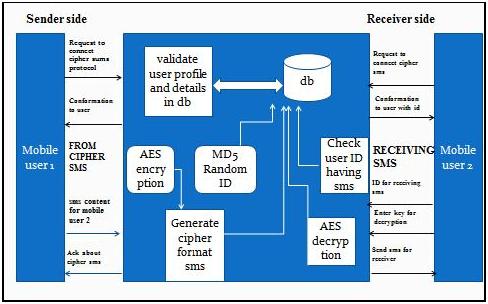
The chat application works in two forms.

* **List form:**

In this form, all the names of the systems connected to a network are enlisted. These names can later be used for communication with the help of mouse event, or in simple language: a click or a double click.

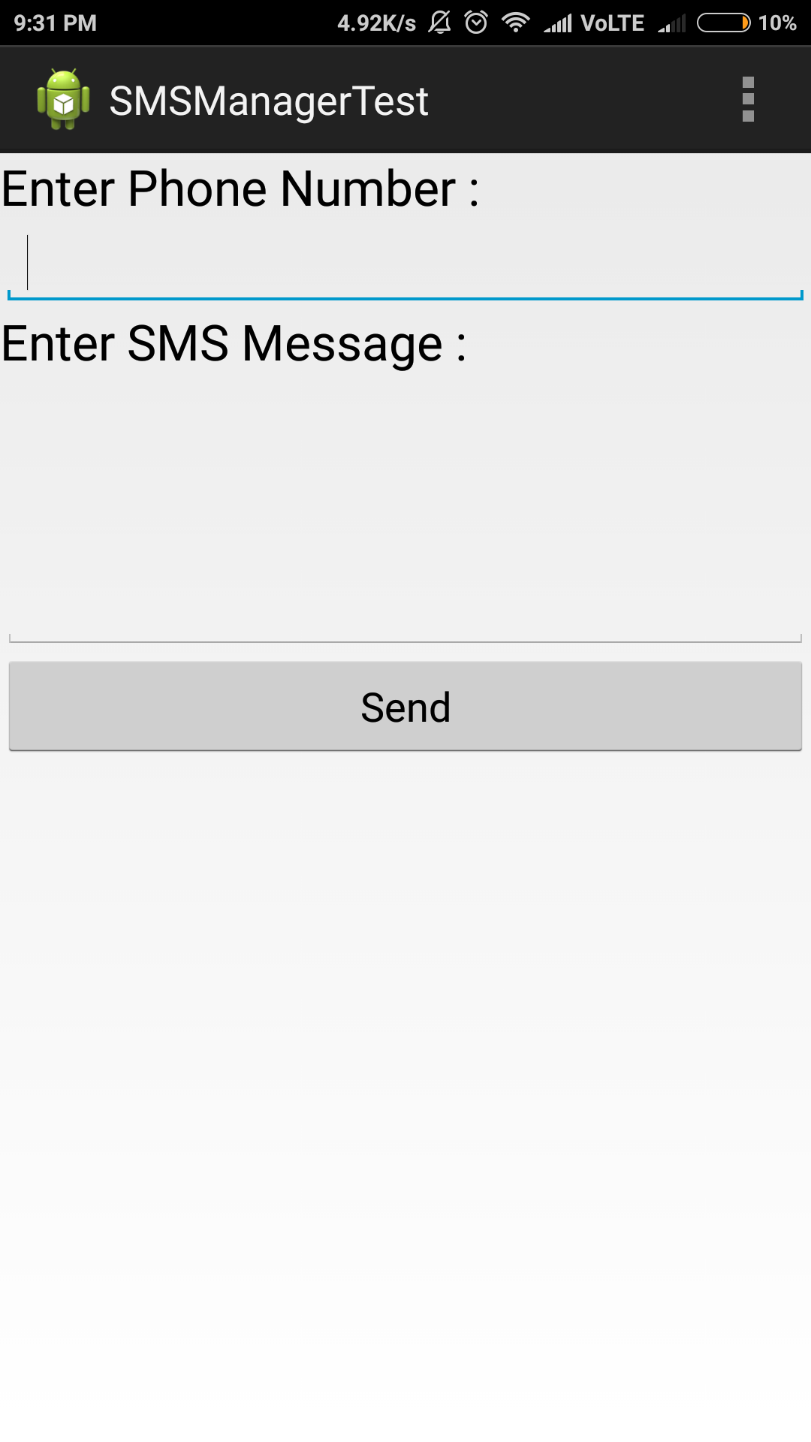
* **Chat form**

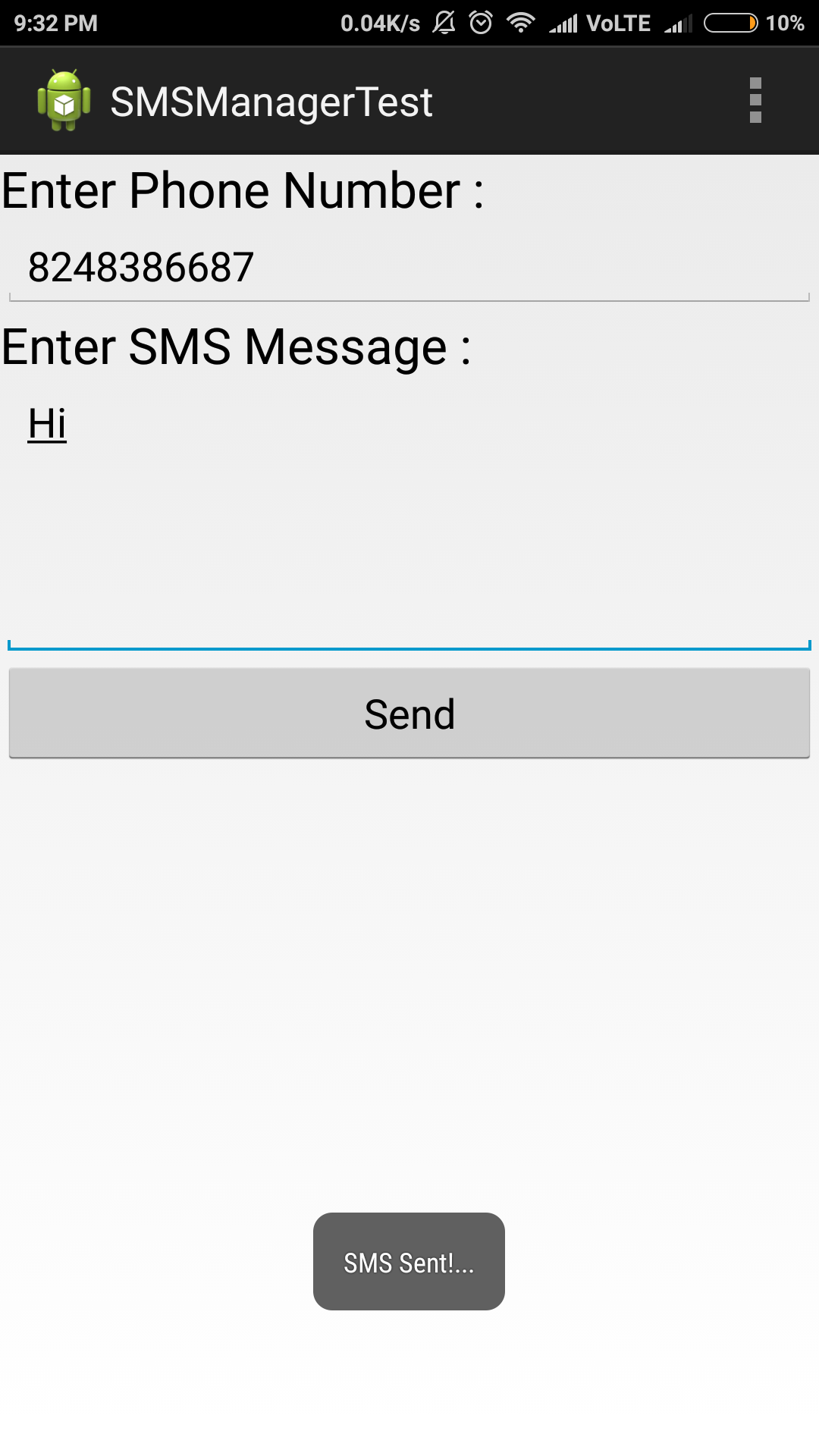
This form is called only when an element is selected from the List form. In this form, a connection is created between the host system and the selected system with the help of a socket.



**4.2. Form Design**

SENDER PAGE





**5. SYSTEM TESTING**

After the unit testing they have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. After the modules are connected they have perform the total testing.

Testing is often used in association with the terms verification and validation .Verification is the checking of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

The test strategies will include five different types of testing as describes below:-

**5.1. Logical Testing**:

This will be used to test every aspect of both modes, report and query as soon as it is implemented, using valid, invalid and extreme data test data will be added to test each code module and results compared with the expected results. Sufficient data will be added to ensure that there is at least one entry in each category. Subsequent tests will often involve adding new data, which will be deleted when the test works satisfactorily. As per our requirement we have also included some field such as character size etc. And then queries were performed after those results were tabulated and then the module was free from extra field.

**5.2. Functional Testing**:

In this menu items were tested to ensure no functions has been missed out. This is done for the smooth working of the project.

**5.3. System Testing**:

This is done after the completion of system; all the queries were carried out again to ensure that no errors have been introduced.

**6. SYSTEM IMPLEMENTATION**

**Modules:**

The project” Message communication” contains of five module. They are

1. Sender
2. Receiver

**Modules Description:**

**Sender:**

This module allows only the sender to send message.

**Receiver:**

This module allows only a receiver to get the message.

**7. SYSTEM MAINTENANCE**

**Android:**

Before learning all topics of android, it is required to know what is android.

**Android** is a software package and Linux based operating system for mobile devices such as tablet computers and smart phones. It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used. The goal of android project is to create a successful real-world product that improves the mobile experience for end users. There are many code names of android such as Lollipop, Kit Kat, Jelly Bean, Ice cream Sandwich, Forgo, Éclair, Donut etc which is covered in next page.

## What is Open Handset Alliance (OHA)

It's a consortium of 84 companies such as Google, Samsung, AKM, synaptic, KDDI, Garmin, Telecom, EBay, Intel etc.

It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Plate form.

## Features of Android

After learning what is android, let's see the features of android. The important features of android are given below:

1) It is open-source.

2) Anyone can customize the Android Platform.

3) There are a lot of mobile applications that can be chosen by the consumer.

4) It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.

It provides support for messaging services (SMS and MMS), web browser, storage (SQ Lite), connectivity (GSM, CDMA, Blue Tooth, and Wi-Fi etc.), media, handset layout etc.

## Categories of Android applications

There are many android applications in the market. The top categories are:

* Entertainment
* Tools
* Communication
* Productivity
* Personalization
* Music and Audio
* Social
* Media and Video
* Travel and Local etc.

# History of Android

The history and versions of android are interesting to know. The code names of **Aestro**, **Blender**, **Cupcake**, **Donut**, **Eclair**, **Froyo**, **Gingerbread**, **Honeycomb**,**Ice Cream Sandwich**, **Jelly Bean**, **Kit Kat** and **Lollipop**. Let's understand the android history in a sequence.

1) Initially, **Andy Rubin** founded Android Incorporation in Palo Alto, California, United States in October, 2003.

2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.

3) The key employees of Android Incorporation are **Andy Rubin**, **Rich Miner**, **Chris White** and **Nick Sears**.

4) Originally intended for camera but shifted to smart phones later because of low market for camera only.

5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.

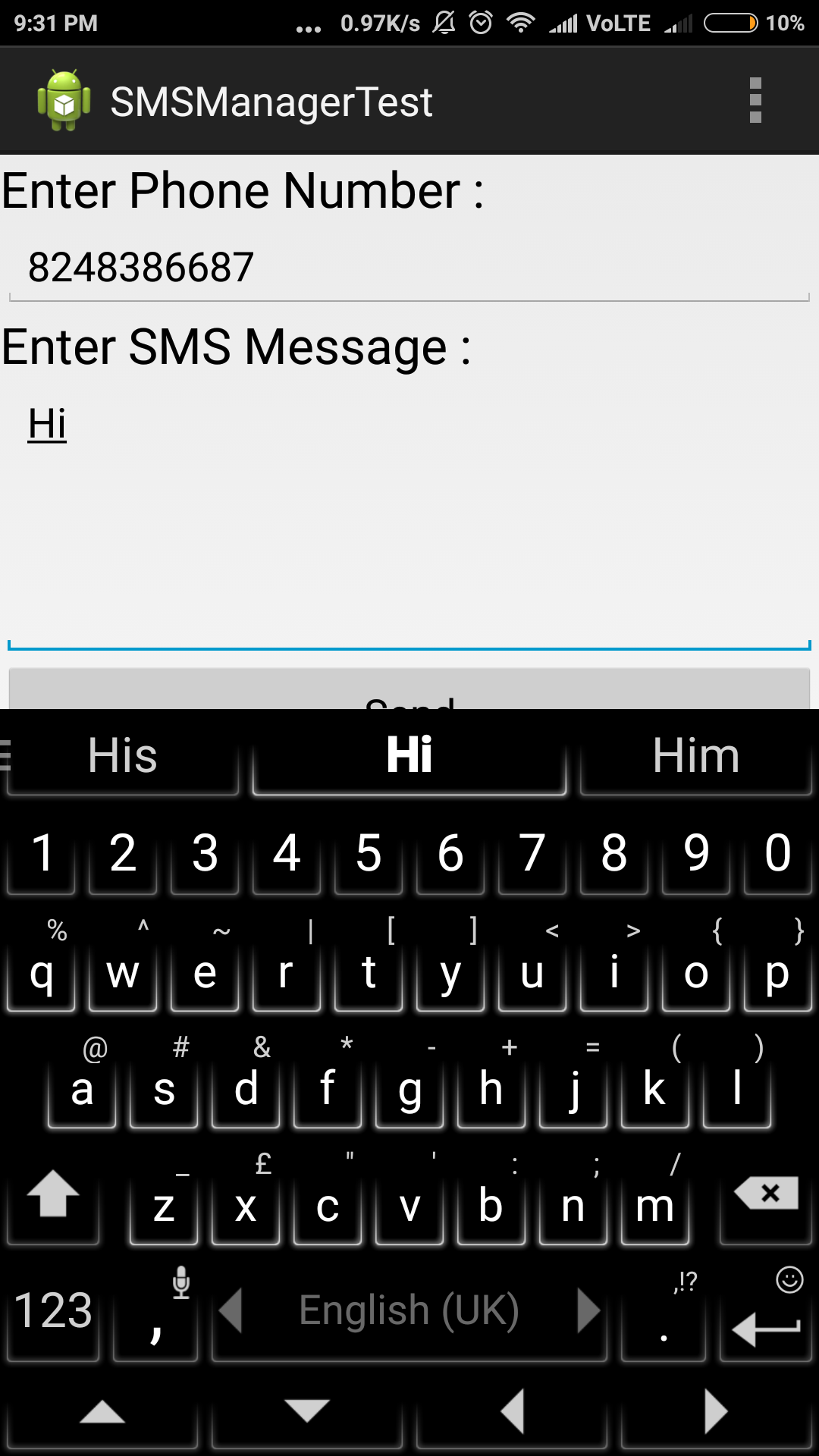
6) In 2007, Google announces the development of android OS.

7) In 2008, HTC launched the first android mobile.

**8. APPENDIX**

**8.1. Screen shots**

SENDER PAGE



**8.2. Source Code**

**MainActivity.java**

**package**com.example.you.chat8;

**import**android.Manifest;

**import**android.app.Activity;

**import**android.app.PendingIntent;

**import**android.content.BroadcastReceiver;

**import**android.content.Context;

**import**android.content.Intent;

**import**android.content.IntentFilter;

**import**android.net.Uri;

**import**android.os.Bundle;

**import**android.support.v4.app.ActivityCompat;

**import**android.telephony.SmsManager;

**import**android.view.View;

**import**android.view.View.OnClickListener;

**import**android.widget.AdapterView;

**import**android.widget.ArrayAdapter;

**import**android.widget.Button;

**import**android.widget.EditText;

**import**android.widget.ListView;

**import**android.widget.Toast;

**import**com.google.android.gms.appindexing.Action;

**import**com.google.android.gms.appindexing.AppIndex;

**import**com.google.android.gms.common.api.GoogleApiClient;

**import**java.util.ArrayList;

**public class** MainActivity**extends**Activity{

Button **buttonSend**;

EditText**textPhoneNo**;

EditText**textSMS**;

**private static** MainActivity*inst*;

ArrayList<String>**smsMessagesList**= **new** ArrayList<String>();

ListView**smsListView**;

ArrayAdapter**arrayAdapter**;

**public static** MainActivity instance() {

**return***inst*;

}

@Override

**public void** onStart() {

**super**.onStart();

*inst*= **this**;

}

*/\*\**

*\* ATTENTION: This was auto-generated to implement the App Indexing API.*

*\* See https://g.co/AppIndexing/AndroidStudio for more information.*

*\*/*

**private**GoogleApiClient**client**;

@Override

**public void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

**buttonSend**= (Button) findViewById(R.id.***buttonSend***);

**textPhoneNo**= (EditText) findViewById(R.id.***editTextPhoneNo***);

**textSMS**= (EditText) findViewById(R.id.***editTextSMS***);

*//access permission*

ActivityCompat.*requestPermissions*(**this**,

**new**String[]{Manifest.permission.***RECEIVE\_SMS***},1);

*// Recieve SMS*

**smsListView**= (ListView) findViewById(R.id.***SMSList***);

**arrayAdapter**= **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***, **smsMessagesList**);

**smsListView**.setAdapter(**arrayAdapter**);

*// smsListView.setOnItemClickListener(this);*

*// ATTENTION: This was auto-generated to implement the App Indexing API.*

*// See https://g.co/AppIndexing/AndroidStudio for more information.*

**client**= **new** GoogleApiClient.Builder(**this**).addApi(AppIndex.***API***).build();

}

**public void** updateList(**final** String smsMessage) {

**arrayAdapter**.insert(smsMessage, 0);

**arrayAdapter**.notifyDataSetChanged();

}

**public void** sendSMS(View v) {

*//Step 1 :*

String SENT = **"SMS SENT"**; *// Any name*

String DELIVERED = **"SMS\_DELIVERED"**;

ActivityCompat.*requestPermissions*(**this**,

**new**String[]{Manifest.permission.***SEND\_SMS***}, 1);

PendingIntentsentPI = PendingIntent.*getBroadcast*(**this**, 0,

**new**Intent(SENT), 0);

PendingIntentdeliveredPI = PendingIntent.*getBroadcast*(**this**, 0,

**new**Intent(DELIVERED), 0);

*//—when the SMS has been sent—*

registerReceiver(**new** BroadcastReceiver() {

@Override

**public void** onReceive(Context arg0, Intent arg1) {

**switch**(getResultCode()) {

**case**Activity.***RESULT\_OK***:

Toast.*makeText*(getBaseContext(), **"SMS sent"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

**case**SmsManager.***RESULT\_ERROR\_GENERIC\_FAILURE***:

Toast.*makeText*(getBaseContext(), **"Generic failure"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

**case**SmsManager.***RESULT\_ERROR\_NO\_SERVICE***:

Toast.*makeText*(getBaseContext(), **"No service"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

**case**SmsManager.***RESULT\_ERROR\_NULL\_PDU***:

Toast.*makeText*(getBaseContext(), **"Null PDU"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

**case**SmsManager.***RESULT\_ERROR\_RADIO\_OFF***:

Toast.*makeText*(getBaseContext(), **"Radio off"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

}

}

}, **new** IntentFilter(SENT));

*//—when the SMS has been delivered—*

registerReceiver(**new** BroadcastReceiver() {

@Override

**public void** onReceive(Context arg0, Intent arg1) {

**switch**(getResultCode()) {

**case**Activity.***RESULT\_OK***:

Toast.*makeText*(getBaseContext(), **"SMS delivered"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

**case**Activity.***RESULT\_CANCELED***:

Toast.*makeText*(getBaseContext(), **"SMS not delivered"**,

Toast.***LENGTH\_SHORT***).show();

**break**;

}

}

}, **new** IntentFilter(DELIVERED));

SmsManagersms = SmsManager.*getDefault*();

Toast.*makeText*(MainActivity.**this**, **textSMS**.getText().toString() +**"-"** + **textPhoneNo**.getText().toString(), Toast.***LENGTH\_LONG***).show();

sms.sendTextMessage(**textPhoneNo**.getText().toString(), **null**, **textSMS**.getText().toString(), sentPI, deliveredPI);

}

}

**AndroidManifist.xml**

***<?*xml version="1.0" encoding="utf-8"*?>***

**<manifest xmlns:android="http://schemas.android.com/apk/res/android"**

**package="com.example.you.chat8">**

**<uses-permission android:name="android.permission.SEND\_SMS" />**

**<uses-permissionandroid:name="android.permission.RECEIVE\_SMS"/>**

**<application**

**android:allowBackup="true"**

**android:icon="@drawable/images1"**

**android:label="@string/app\_name"**

**android:supportsRtl="true"**

**android:theme="@style/AppTheme">**

**<activity android:name=".Login">**

**<intent-filter>**

**<action android:name="android.intent.action.MAIN" />**

**<category android:name="android.intent.category.LAUNCHER" />**

**</intent-filter>**

**</activity>**

**<activity android:name=".Recive">**

**</activity>**

**<activity android:name=".MainActivity">**

**</activity>**

**<receiver android:name=".SmsBroadcastReceiver" android:exported="true" android:enabled="true">**

**<intent-filter>**

**<action android:name="android.provider.Telephony.SMS\_RECEIVED" />**

**</intent-filter>**

**</receiver>**

***<!-- ATTENTION: This was auto-generated to add Google Play services to your project for***

***App Indexing. See https://g.co/AppIndexing/AndroidStudio for more information. -->***

**<meta-data**

**android:name="com.google.android.gms.version"**

**android:value="@integer/google\_play\_services\_version" />**

**</application>**

**</manifest>**

**LoginActivity.xml**

***<?*xml version="1.0" encoding="utf-8"*?>***

**<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"**

**android:orientation="vertical" android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent">**

**<TextView**

**android:id="@+id/tv\_Username"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_marginTop="69dp"**

**android:text="UserName"**

**android:layout\_alignParentTop="true"**

**android:layout\_alignParentLeft="true"**

**android:layout\_alignParentStart="true" />**

**<EditText**

**android:id="@+id/et\_Username"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_alignBaseline="@+id/tv\_Username"**

**android:layout\_alignBottom="@+id/tv\_Username"**

**android:layout\_marginLeft="30dp"**

**android:layout\_toRightOf="@+id/tv\_Username"**

**android:ems="10" >**

**<requestFocus/>**

**</EditText>**

**<TextView**

**android:id="@+id/tv\_Password"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_marginTop="106dp"**

**android:text="Password"**

**android:layout\_below="@+id/et\_Username"**

**android:layout\_alignLeft="@+id/tv\_Username"**

**android:layout\_alignStart="@+id/tv\_Username" />**

**<EditText**

**android:id="@+id/et\_Password"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_alignBaseline="@+id/tv\_Password"**

**android:layout\_alignBottom="@+id/tv\_Password"**

**android:layout\_alignLeft="@+id/et\_Username"**

**android:ems="10"**

**android:inputType="textPassword" />**

**<Button**

**android:id="@+id/bt\_SignIn"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="signin"**

**android:layout\_below="@+id/et\_Password"**

**android:layout\_alignRight="@+id/et\_Password"**

**android:layout\_alignEnd="@+id/et\_Password"**

**android:layout\_marginTop="151dp" />**

**</RelativeLayout>**

****

**8.3. Dataflow Diagram**

SENDER

SERVER

****

****

RECEIVER

STOP PROCESSOR

**9. CONCLUSION**

This Project has given a broad confidence to the developer and people the software in terms of the different modules used. It also gives us an idea about the Security to which each module performs related tasks. Project also get an idea about the interdependence of the public.

In this paper we created a sample app to show the usage and level of protection of permissions in android. We have studied various permissions and their protection levels. The importance of in self-created permissions was analyzed and was studied in detailed manner. These permission protection levels when used at development level can guard the other application and data from hacking and malicious apps. The apps should not over privileged to access the content from the phone.

**10. FUTURE ENHANCEMENTS**

There is always a room for improvements in any software package, however good and efficient it may be done. But the most important thing should be flexible to accept further modification. Right now we are just dealing with text communication. In future this software may be extended to include features such as:

**File transfer:**

This will enable the user to send files of different formats to others via the chat application.

**Voice chat:**

This will enhance the application to a higher level where communication will be possible via voice calling as in telephone.

**Video chat:**

This will further enhance the feature of calling into video communication.

**11. BIBLIOGRAPHY**

**References:**

1. Computer Science 2008, An Interim Revision of CS

2001( )

2. Android Developer’s Guide. <http://developer.android.com/guide/index.html>

3. Abelson, W.F., Collins, C., Sen, R. *Unlocking Android – A Developer’s Guide*.

Manning Pub. April 2009.

4. Victor Matos, Rebecca Grasser, Building Applications for the Android OS Mobile

Platform: A Primer and Course Materials, *Journal of Computing Sciences in Colleges*,

Volume 26 Issue 1, pp: 23-29, October 2010.