

Spring 2016

# An Android Application for Remote Ringer

Ugendra Bhuvan Bobba  
*Governors State University*

Mallikharjuna Rao Kurapati  
*Governors State University*

Ramakrishna Pamarthi  
*Governors State University*

Follow this and additional works at: <http://opus.govst.edu/capstones>



Part of the [Software Engineering Commons](#)

---

## Recommended Citation

Bobba, Ugendra Bhuvan; Kurapati, Mallikharjuna Rao; and Pamarthi, Ramakrishna, "An Android Application for Remote Ringer" (2016). *All Capstone Projects*. 203.  
<http://opus.govst.edu/capstones/203>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to [http://www.govst.edu/Academics/Degree\\_Programs\\_and\\_Certifications/](http://www.govst.edu/Academics/Degree_Programs_and_Certifications/)

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact [opus@govst.edu](mailto:opus@govst.edu).

## **ABSTRACT**

At the time we set our mobile phones in silent mode, we may misplace it somewhere nearby and cannot find the mobile phone by ringing to it. It has happened to all of us. To solve this real world problem, we develop an application called Remote Ringer. Remote Ringer helps us to change mobile phone mode from silent to ringing mode . Once your phone is in ringing mode, you can ring it from any other mobile phones.

Remote Ringer needs to be configured before it can be used, The configuration allows maximum of five numbers that can be used to change the mode from silent to ring. Use one of the five phones to text the silent phone, the silent phone will be switched to a tone ring mode, then another call will enable user to locate the phone.

# Table of Content

<b>1. Introduction</b> .....	4
1.1. Problem definition.....	4
1.2. Methodology used.....	5
<b>2. Literature survey</b> .....	7
2.1. Proposed system.....	7
2.2. Advantages of proposed system.....	7
2.3. Working of proposed system.....	7
<b>3. System analysis and design</b> .....	9
3.1. Use case diagram.....	9
3.2. Software requirements.....	11
3.3. Hardware requirements.....	11
3.4. Class diagram.....	11
3.5. Sequence diagrams.....	12
<b>4.Implementation</b> .....	14
4.1. Android.....	14
4.2. Installing the android sdk.....	19
4.3. Creating .apk file for android devices.....	22
<b>5. App overview</b> .....	26
<b>6. Test plan report</b> .....	28
<b>7. Conclusion &amp; future scope</b> .....	31
<b>8.References</b> .....	32

## ***1.INTRODUCTION***

It's happened to all of us - You search all around your house for your smart phone only give up and resort to whipping out the old and busted landline phone and Call your number. Most of the time your phone is either tucked in your pants pocket that you have thrown in the hamper to be washed, or hiding in the cushions of your couch.

This method works probably 90% of the time. But what about that 10% of the time when your phone is either in silent or vibrate mode? The Remote Ringer app for Android can force your phone into normal mode by receiving a SMS text or Call from an approved number.

Remote Ringer works in three easy steps:

- Define the authorized contact number that can force your phone into normal mode
- Define your secret Password that will be sent in the text message or secret number that you can Call
- Press the back button to exit

As long as the text comes from an authorized number, anybody can send a text to your phone to force it into normal. So to the people who live alone and don't have another device in the home capable of sending text messages or to make a Call, you can authorize a friend or family member's number to force the change. Then you can Call your number and actually hear it ringing.

### ***1.1 PROBLEM DEFINITION***

Your Phone is in silent mode and you missed it somewhere nearby You search all around your house for your smart phone only give up and resort to whipping out the old and busted landline phone and Call your number. But you didn't find your phone because it is in silent mode

### ***1.2 METHODOLOGY USED:***

#### ***1.2.1 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC):***

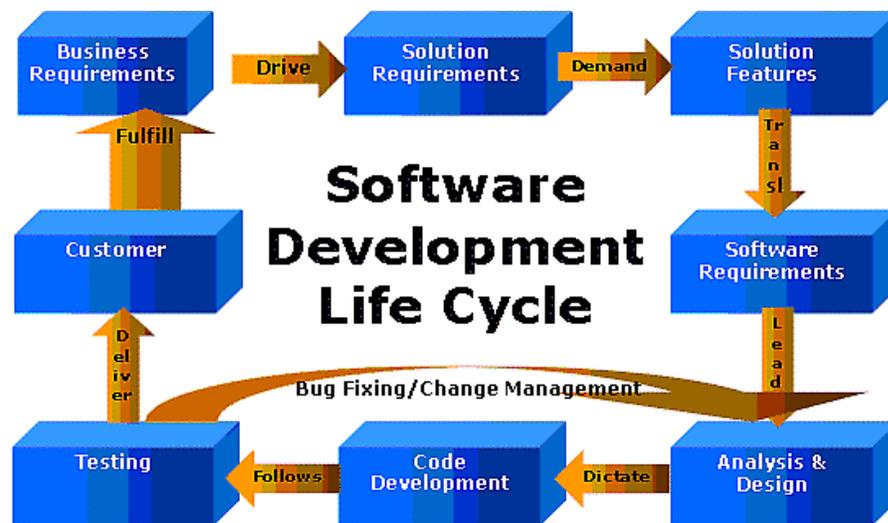
An effective Software Development Life Cycle (SDLC) enables us to provide cost-effective and efficient solutions. Project success is directly proportional to the project planning and the phases of software development life cycle.

***Phases of the SDLC:***

An effective SDLC has the following phases:

**i) Requirement Specification:**

Requirement Specification is the first and most important phase of the SDLC. During this phase Project Manager is in constant contact with the Customer to find out requirements of the project in detail. Main tasks in this phase include: Requirement Determination, Risk Analysis, Setting up Schedules, and deciding Deliverables



**Figure 1.2.1: Software Development Life Cycle**

**ii) Requirement Analysis and Design:**

Project Manager and System Analyst after reviewing the customers requirements, analyses the requirements and a System Requirement Specification Document (SRS) is prepared. This will be followed by the designing of the project. System Architecture, Database Design, Program Specifications and Test Scenarios are determined. A Detail Design Document is prepared at the end of analysis that can be used by the programmers to perform the coding.

**iii) Coding and Testing:**

Programmers begin programming in this phase using the Detail Design Document. Project Manager is in constant contact with the customer and provides updates on the progress of the project. The programmers follow coding standards decided by the company. Project Leader

helps the programmers with their coding problems and guides them to the solutions. Testing is done by the QA Team simultaneously for the finished modules and approval is given to the modules once they have passed their initial tests before integration.

**iv) Deployment and Support:**

This phase starts with Deployment of the Project. Initial hardware and software setup necessary to run the project is a very critical phase of the project. After project is completed Project Manager contacts the customer and prepares for the set-up. Software is handed over to the customer for acceptance testing only after complete internal testing. Support to the project is provided for a limited number of days during which any minor customer changes are finished.

## ***2.LITERATURE SURVEY***

There is no existing system. This proposed system rectifies most of the real world problems. This app helps you to change mobile phone mode from silent to ringing mode. Once your phone is in ringing mode, you can ring it from any another mobile.

### ***2.1 PROPOSED SYSTEM***

The proposed system needs to be configured before use, you can add maximum of five numbers that can be used to make a Call to your phone in case of emergency. You can also add a secret word that you can sent as an SMS to your phone from any other phone. It also provides Password protection to prevent unauthorized application changes.

### ***2.2 ADVANTAGES OF PROPOSED SYSTEM***

- Sent an SMS from any number.
- Call from a particular number.
- Password protection to prevent unauthorized application changes.

### ***2.3 WORKING OF PROPOSED SYSTEM***

Remote Ringer is used by all the users. Remote Ringer helps you to change mobile phone mode from silent to ringing mode. Once your phone is in ringing mode, you can ring it from any another mobile. User can install this application. After installing this application we specify message, number and Password.

User have to specify a secret word that you can sent as an SMS to your phone from any other phone or specify a maximum of five numbers that can be used to make a Call to your phone in case of emergency.

Any user can open this application and change the SMS settings and Call settings, So we provide Password protection to prevent unauthorized application changes.

## **Home Page**

Home page consists of three options:

- The SMS settings text view is used to specify a secret word.
- The Call settings text view is used to specify numbers.
- The Password settings text view is used to specify Password.

## **Login page for users**

This page is for users to login. Users can login into this application by filling the two fields i.e. user name and Password. Once authenticated he will be switched to his home page.

## **Home page of users**

The home page of user has options as follows:

- SMS settings
- Call settings
- Password settings

## **SMS settings**

This page helps the user to specify a secret word that you can sent as an SMS to your phone from any other phone. It also consists of checkbox, if this checkbox is enabled then message is compared for the incoming messages.

## **Call settings**

This page helps the user to specify numbers that that can be used to make a Call to your phone in case of emergency from any other phone. It also consists of checkbox, if this checkbox is enabled then number is compared for the incoming Calls.

## **Password settings**

This page helps the user to enter Password that can be used to open the application. It also consists of checkbox, if this checkbox is enabled then Password is compared with the specified Password.

### ***3.SYSTEM ANALYSIS AND DESIGN***

#### **3.1 USECASE DIAGRAM:**

A use case diagram shows a set of use cases, actors and their relationships. This diagram shows a static design view of system. These are important in organizing and modeling.

##### **Actors:**

Actor is something external to the system and interacts with the system. Actor may be a human being, device or some other software system.

For Remote Ringer, actors might be

- Users

##### **Use Cases:**

A use-case represents sequence of actions performed by the system that yields an observable result of value for a particular actor. Use-case represents a functional requirement of a system. For Remoter Ringer, we can find the following use cases:

- Set a Password
- Set a word
- Set a contact
- Enable Password
- Enable SMS
- Enable Call

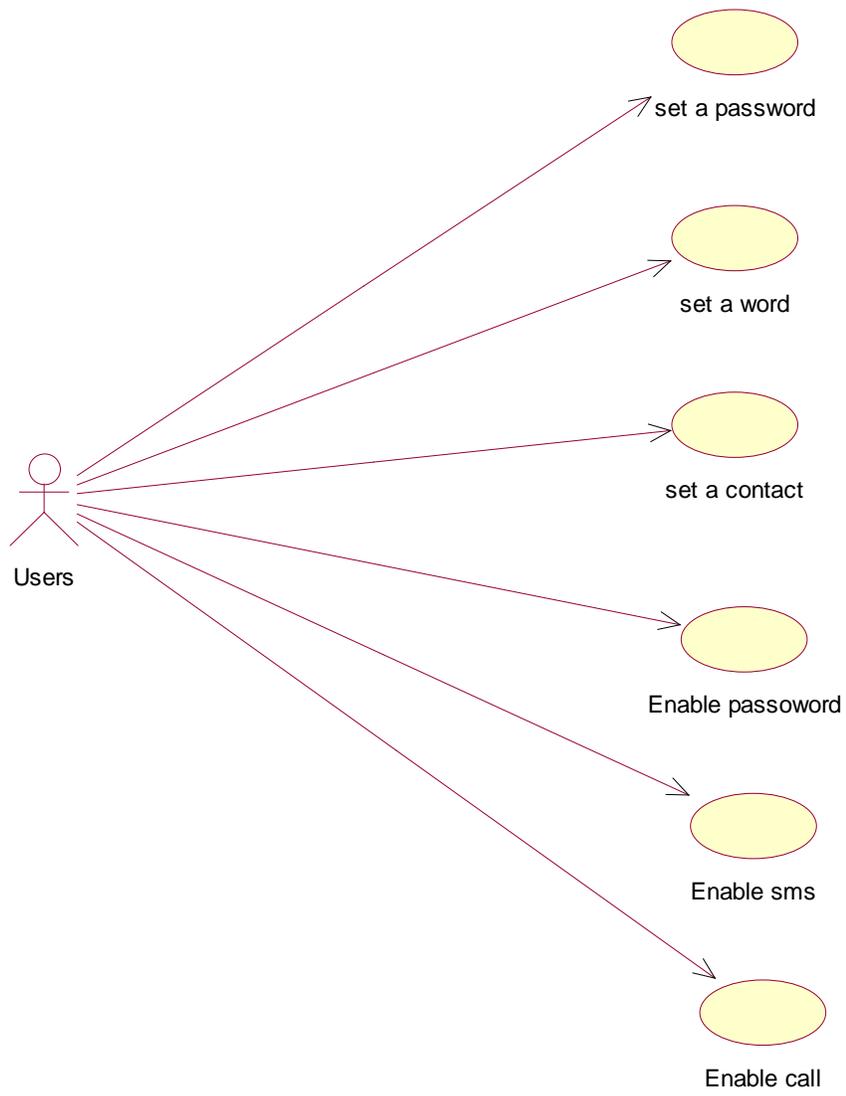


FIG 3.1.1 Usecase diagram for working of Remote Ringer

### **3.2 SOFTWARE REQUIREMENTS:**

- ✓ Android-SDK-Windows-1.5\_r1
- ✓ ECLIPSE- GANYMEDE
- ✓ JAVA
- ✓ A system with any OS

### **3.3 HARDWARE REQUIREMENTS:**

- ✓ Any android phone

### **3.4 CLASS DIAGRAM:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

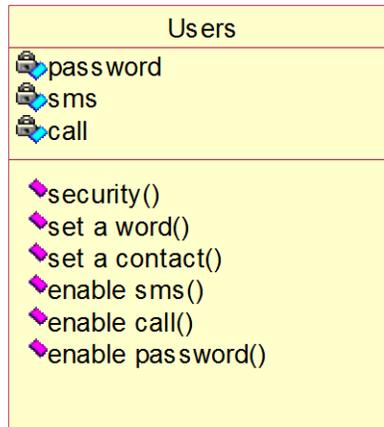


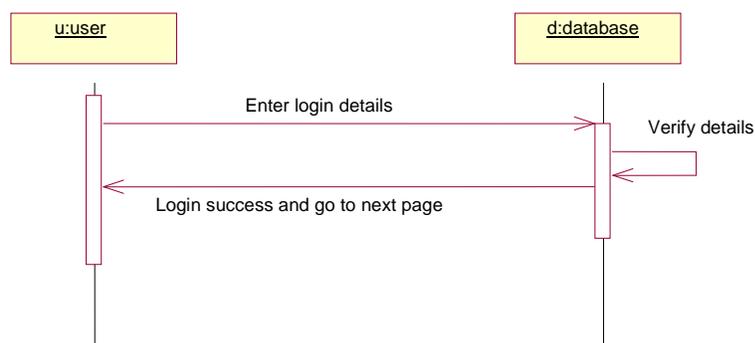
FIG 3.4.1 class diagram for working of Remote Ringer

### 3.5 SEQUENCE DIAGRAMS:

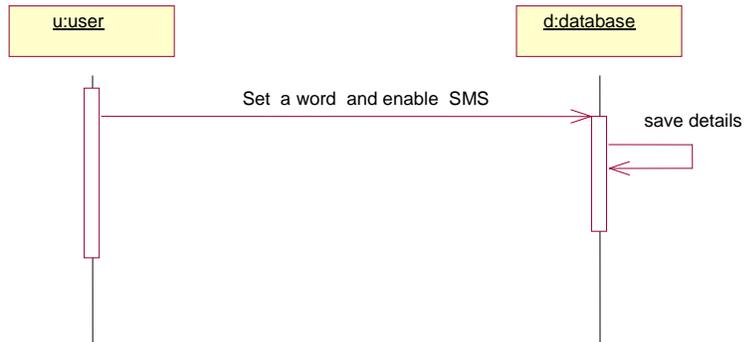
Sequence diagrams are a kind of interaction diagrams. An interaction diagram shows an interaction, consisting of set of objects and their relationships, including messages that may dispatch among them. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.

In our project we have 4 sequence diagrams. These are login, to set word, to set a Password and to set a contact.

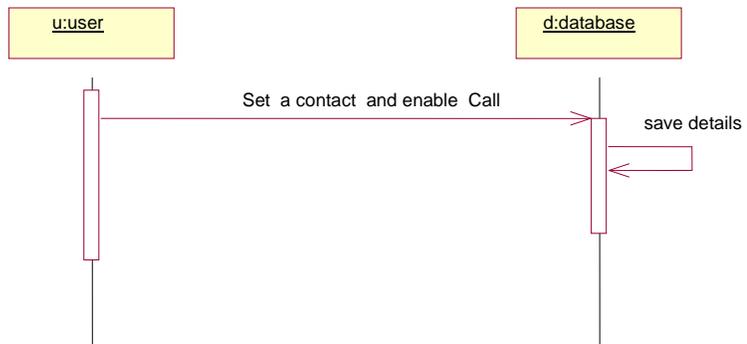
#### *Sequence diagram for login:*



***Sequence diagram to set SMS:***

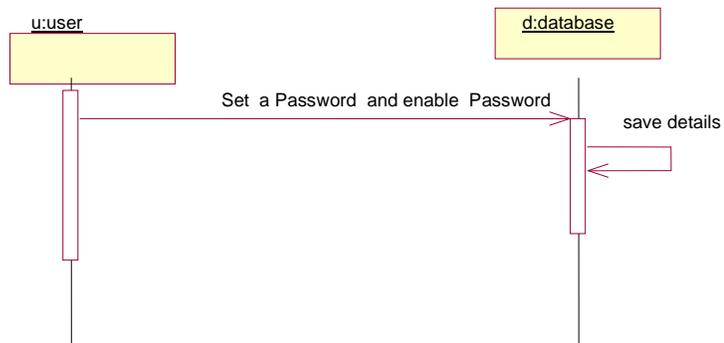


***Sequence diagram to set Call:***



---

***Sequence diagram to set Password:***



## ***4.IMPLEMENTATION***

### ***4.1 ANDROID:***

Mobile phones are miniature computers in these days and the applications that run on them are similar to web-applications. Android is a Linux-based operating system for mobile devices such as smart phones and tablet computers. It is developed by the Open Handset Alliance led by Google. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

### ***FEATURES:***

- **Application framework** enabling reuse and replacement of components.
- **Dalvik virtual machine** optimized for mobile devices.
- **Integrated browser** based on the open source WebKit engine.
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional).
- **SQLite** for structured data storage.
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, and GIF).
- **GSM Telephony** (hardware dependent).
- **Bluetooth, EDGE, 3G, and Wi-Fi** (hardware dependent).
- **Camera, GPS, compass, and accelerometer** (hardware dependent).
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plug-in for the Eclipse IDE.

The most commonly used words while developing an android application are:

### **Dalvik Virtual Machine**

Dalvik is open-source software. It was originally written by Dan Bornstein. Dalvik is the process virtual machine (VM) in Google's Android operating system. It is the software that runs the applications on Android devices. Dalvik is thus an integral part of Android, which is typically used on mobile devices such as mobile phones and tablet computers. Programs are

commonly written in Java and compiled to byte code. Then they are converted from Java Virtual Machine compatible *.class* files to Dalvik-compatible *.dex* (Dalvik Executable) files before installation on a device. The compact Dalvik Executable format is designed to be suitable for systems that are constrained in terms of memory and processor speed.

## **SQLite**

SQLite is an ACID compliant embedded relational database management system contained in a small C programming library.

SQLite implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity (Integrity constraints are used to ensure accuracy and consistency of data in a relational database). In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it. SQLite read operations can be multitasked, though writes can only be performed sequentially. The source code for SQLite is in the public domain.

## **Eclipse IDE**

Eclipse is an open-source community that develops open platforms and products. The community says its projects "are focused on building an open development platform consisting of extensible frameworks, tools and runtimes for building, deploying and managing software across the life cycle". The Eclipse Foundation is a non-profit corporation which acts as the steward of the Eclipse community.

However, as the Eclipse community states, Eclipse means a lot of different things to different people. To some Eclipse is Java development environment. To others, Eclipse is a flexible environment to experiment with new computer languages or extensions to existing languages. In the software world, a simple mention of "Eclipse" usually refers to the Eclipse software development kit (SDK). The Eclipse SDK consists of the Eclipse Platform, Java development tools and the Plug-in Development Environment. The Eclipse Platform is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java. By means of various plug-ins, it can be used to develop applications in various programming languages including Ada, C, C++, Java, PHP, Python and Android.

## Android Emulator

The Android SDK includes a mobile device emulator, a virtual mobile device that runs on your computer. It will let you to model an actual device by defining hardware and software options to be emulated. The emulator lets you develop and test Android applications without using a physical device.

## Broadcast Receiver

Base class for code that will receive intents sent by `sendBroadcast()`. You can either dynamically register an instance of this class with `Context.registerReceiver()` or statically publish an implementation through the `<receiver>` tag in your `AndroidManifest.xml`. **Note:** If registering a receiver in your `Activity.onResume()` implementation, you should unregister it in `Activity.onPause()`. (You won't receive intents when paused, and this will cut down on unnecessary system overhead). Do not unregister in `Activity.onSaveInstanceState()`, because this won't be called if the user moves back in the history stack.

There are two major classes of broadcasts that can be received:

- **Normal broadcasts** (sent with `Context.sendBroadcast`) are completely asynchronous. All receivers of the broadcast are run in an undefined order, often at the same time. This is more efficient, but means that receivers cannot use the result or abort APIs included here.
- **Ordered broadcasts** (sent with `Context.sendOrderedBroadcast`) are delivered to one receiver at a time. As each receiver executes in turn, it can propagate a result to the next receiver, or it can completely abort the broadcast so that it won't be passed to other receivers. The order receivers run in can be controlled with the `android:priority` attribute of the matching intent-filter; receivers with the same priority will be run in an arbitrary order.

Even in the case of normal broadcasts, the system may in some situations revert to delivering the broadcast one receiver at a time. In particular, for receivers that may require the creation of a process, only one will be run at a time to avoid overloading the system with new processes. In this situation, however, the non-ordered semantics hold: these receivers still cannot return results or abort their broadcast.

Note that, although the Intent class is used for sending and receiving these broadcasts, the Intent broadcast mechanism here is completely separate from Intents that are used to start Activities with Context.startActivity(). There is no way for a BroadcastReceiver to see or capture Intents used with startActivity(); likewise, when you broadcast an Intent, you will never find or start an Activity. These two operations are semantically very different: starting an Activity with an Intent is a foreground operation that modifies what the user is currently interacting with; broadcasting an Intent is a background operation that the user is not normally aware of.

The Broadcast Receiver class (when launched as a component through a manifest's <receiver> tag) is an important part of an application's overall lifecycle.

Topics covered here:

1. Receiver Lifecycle
2. Permissions
3. Process Lifecycle

### **Receiver Lifecycle**

A BroadcastReceiver object is only valid for the duration of the Call to onReceive(Context, Intent). Once your code returns from this function, the system considers the object to be finished and no longer active.

This has important repercussions to what you can do in an onReceive(Context, Intent) implementation: anything that requires asynchronous operation is not available, because you will need to return from the function to handle the asynchronous operation, but at that point the BroadcastReceiver is no longer active and thus the system is free to kill its process before the asynchronous operation completes.

In particular, you may *not* show a dialog or bind to a service from within a BroadcastReceiver. For the former, you should instead use the NotificationManager API. For the latter, you can use Context.startService() to send a command to the service.

### **Permissions**

Access permissions can be enforced by either the sender or receiver of an Intent.

To enforce a permission when sending, you supply a non-null *permission* argument to sendBroadcast(Intent, String) or sendOrderedBroadcast(Intent, String, BroadcastReceiver,

android.os.Handler, int, String, Bundle). Only receivers who have been granted this permission (by requesting it with the <uses-permission> tag in their AndroidManifest.xml) will be able to receive the broadcast.

To enforce a permission when receiving, you supply a non-null *permission* when registering your receiver -- either when calling registerReceiver(BroadcastReceiver, IntentFilter, String, android.os.Handler) or in the static <receiver> tag in your AndroidManifest.xml. Only broadcasters who have been granted this permission (by requesting it with the <uses-permission> tag in their AndroidManifest.xml) will be able to send an intent to the receiver.

See the Security and Permissions document for more information on permissions and security in general.

### **Process Lifecycle**

A process that is currently executing a BroadcastReceiver (that is, currently running the code in its onReceive(Context, Intent) method) is considered to be a foreground process and will be kept running by the system except under cases of extreme memory pressure.

Once you return from onReceive(), the BroadcastReceiver is no longer active, and its hosting process is only as important as any other application components that are running in it. This is especially important because if that process was only hosting the BroadcastReceiver (a common case for applications that the user has never or not recently interacted with), then upon returning from onReceive() the system will consider its process to be empty and aggressively kill it so that resources are available for other more important processes.

This means that for longer-running operations you will often use a Service in conjunction with a BroadcastReceiver to keep the containing process active for the entire time of your operation.

### **AUDIO MANAGER**

AudioManager provides access to volume and ringer mode control. Use Context.getSystemService(Context.AUDIO\_SERVICE) to get an instance of this class.

### **TELEPHONY MANAGER**

Provides access to information about the telephony services on the device. Applications can use the methods in this class to determine telephony services and states, as well as to access some types of subscriber information. Applications can also register a listener to receive notification of telephony state changes.

You do not instantiate this class directly; instead, you retrieve a reference to an instance through `Context.getSystemService(Context.TELEPHONY_SERVICE)`.

Note that access to some telephony information is permission-protected. Your application cannot access the protected information unless it has the appropriate permissions declared in its manifest file. Where permissions apply, they are noted in the methods through which you access the protected information.

## **4.2 INSTALLING THE ANDROID SDK**

### ***Step 1: Preparing your development computer***

Before getting started with the Android SDK, take a moment to confirm that your development computer meets the System Requirements (Windows XP (32-bit), Vista (32- or 64-bit), Windows 7 (32- or 64-bit) or Linux (tested on Ubuntu Linux, Lucid Lynx)). In particular, you might need to install the Java Development Kit (JDK).

If you will be developing in Eclipse with the Android Development Tools (ADT) Plug-in the recommended path if you are new to Android make sure that you have a suitable version of Eclipse installed on your computer as described in the System Requirements document. If you need to install Eclipse, you can download it from this location: <http://www.eclipse.org/downloads/>

The "Eclipse Classic" version is recommended. Otherwise, a Java or RCP version of Eclipse is recommended.

### ***Step 2: Downloading the SDK starter package***

The SDK starter package is not a full development environment it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform).

If you downloaded a .zip or .tgz package (instead of the SDK installer), unpack it to a safe location on your machine. By default, the SDK files are unpacked into a directory named android-sdk-<machine-platform>.

If you downloaded the Windows installer (.exe file), run it now and it will check whether the proper Java SE Development Kit (JDK) is installed (installing it, if necessary), then install the SDK Tools into a default location (which you can modify).

Make a note of the name and location of the SDK directory on your system you will need to refer to the SDK directory later, when setting up the ADT plug-in and when using the SDK tools from the command line.

### **Step 3: Installing the ADT Plug-in for Eclipse**

Android offers a custom plug-in for the Eclipse IDE, Called Android Development Tools (ADT) that is designed to give you a powerful, integrated environment in which to build Android applications.

ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application. In general, developing in Eclipse with ADT is a highly recommended approach and is the fastest way to get started with Android.

### **Step 4: Adding platforms and other components**

The last step in setting up your SDK is using the Android SDK and AVD Manager (a tool included in the SDK starter package) to download essential SDK components into your development environment.

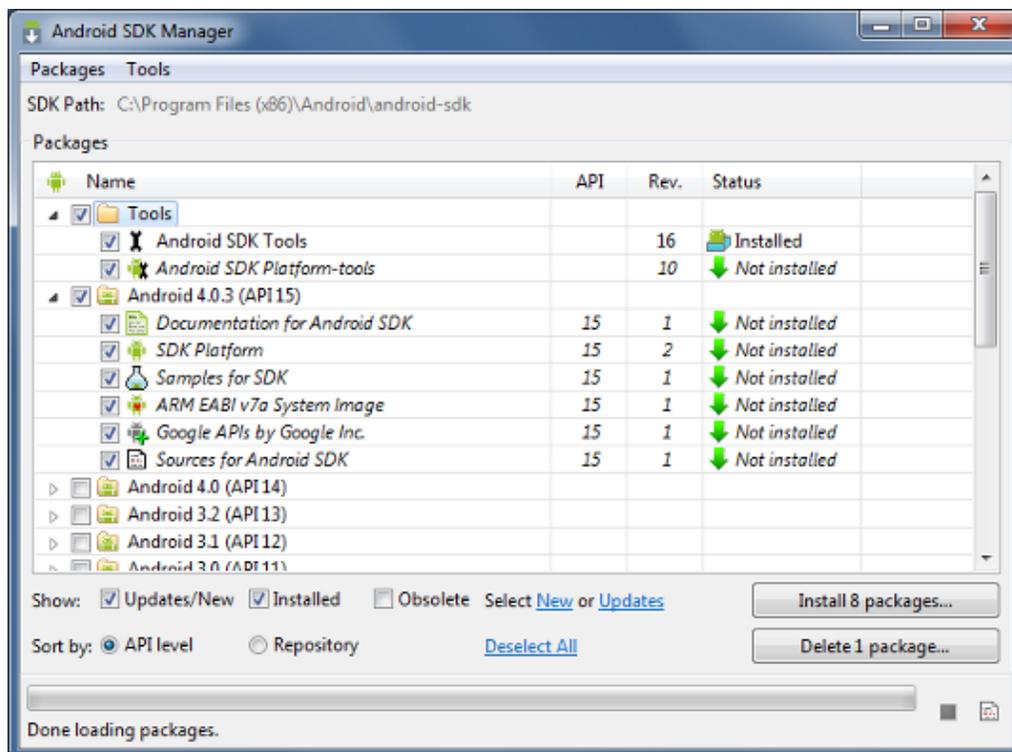
The SDK uses a modular structure that separates the major parts of the SDK Android platform versions, tools, samples, and documentation into a set of separately installable components. The SDK starter package, which you've already downloaded, includes only a single component: the latest version of the SDK Tools. To develop an Android application, you also need to download at least one Android platform and the associated platform tools. You can add other components and platforms as well, which is highly recommended.

If you used the Windows installer, when you complete the installation wizard, it will launch the Android SDK and AVD Manager with a default set of platforms and other components selected for you to install. Simply click **Install** to accept the recommended set of components and install them.

You can launch the Android SDK and AVD Manager in one of the following ways:

- From within Eclipse, select **Window => Android SDK and AVD Manager**.
- On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory.
- On Mac or Linux, open a terminal and navigate to the tools/ directory in the Android SDK, then execute.

To download components, use the graphical UI of the Android SDK and AVD Manager to browse the SDK repository and select new or updated components. The Android SDK and AVD Manager install the selected components in your SDK environment. The Android SDK and AVD Manager's **Available Packages** panel shows the SDK components that are available for you to download into your environment.



**Figure 4.1** Android SDK and AVD Manager's **Available Packages** panel

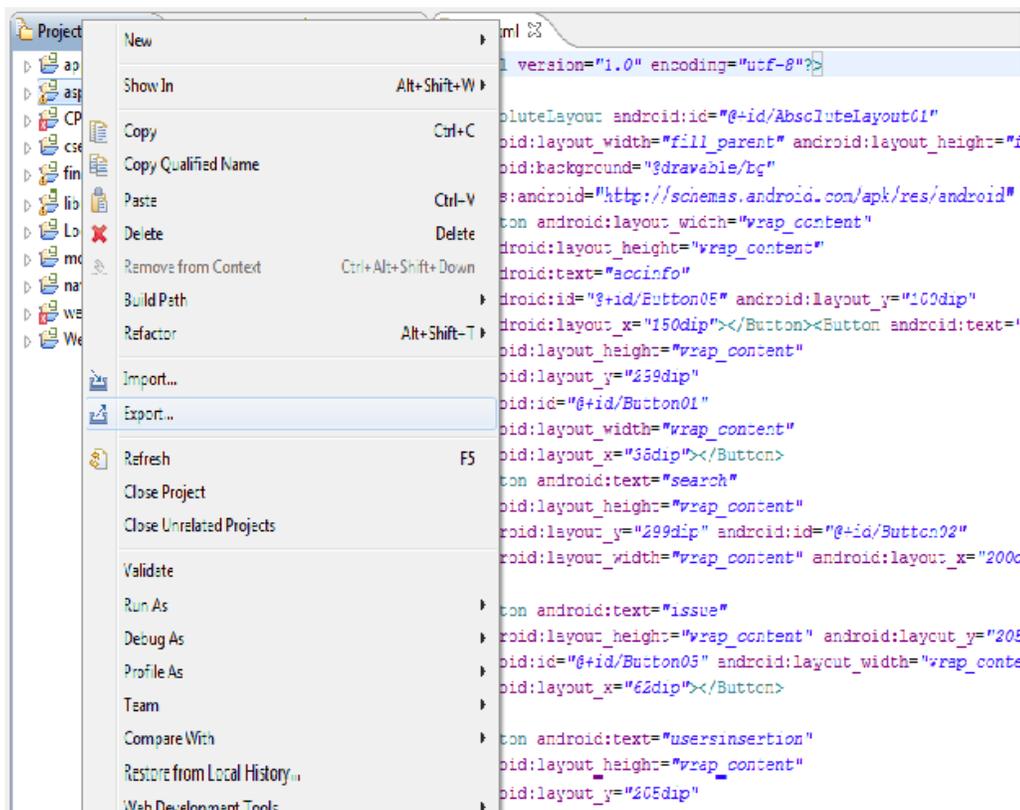
### 4.3 CREATING .APK FILE FOR ANDROID DEVICES

Android uses the .apk file extension for its applications. It denotes an Android Package Kit (APK) file, a variant of the JAR format that is used for distributing and installing bundled components onto the Android operating system. APK files and applications are two different things.

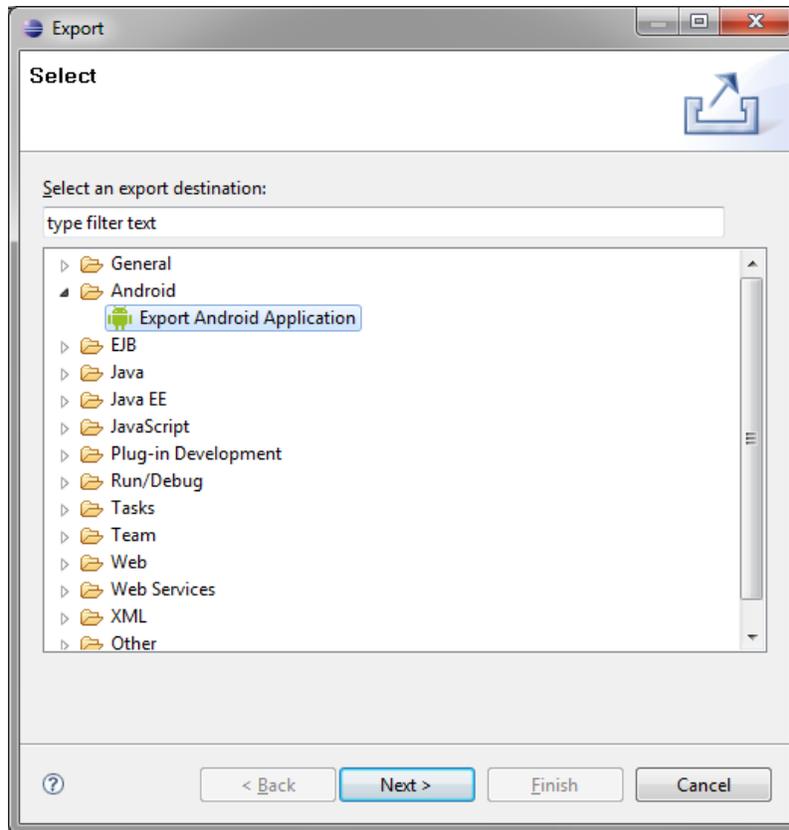
The APK files contain the information necessary for your device to install the program, but once this has been done the APK file is no longer necessary. If you copied APK files for installation onto your device, you can safely delete these copies once the apps are installed.

To create .apk file follow these steps:

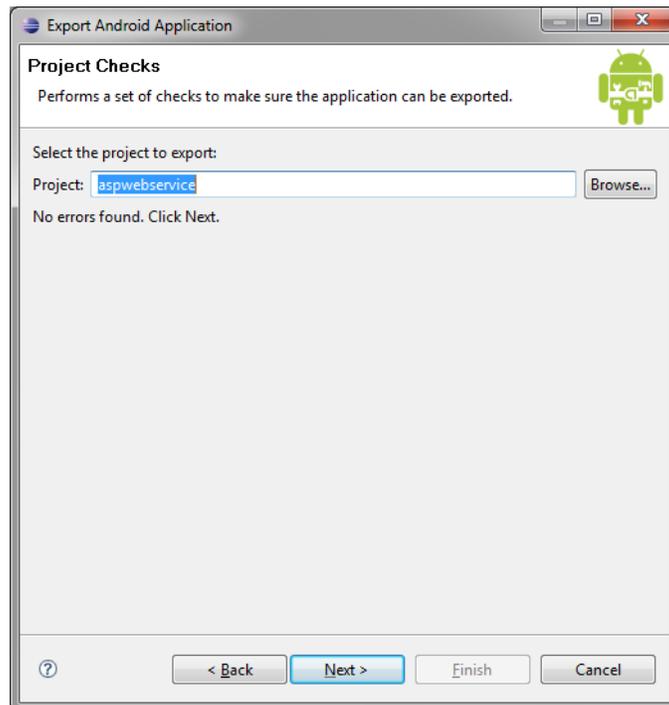
- 1) First click on **Export** (Right-click on Project name =>Export)



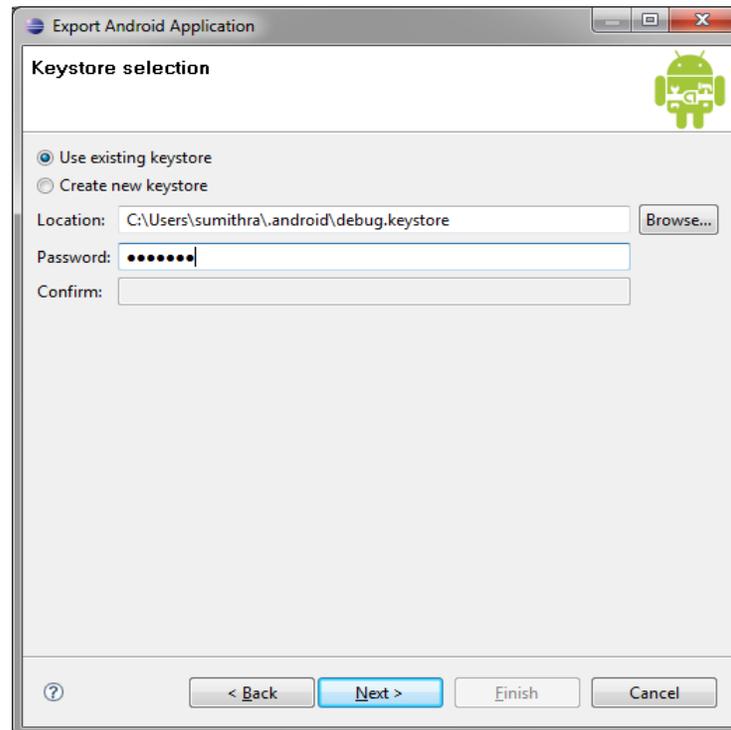
- 2) You will get a window as shown below. Click on “Android” and select “Export Android Application”.



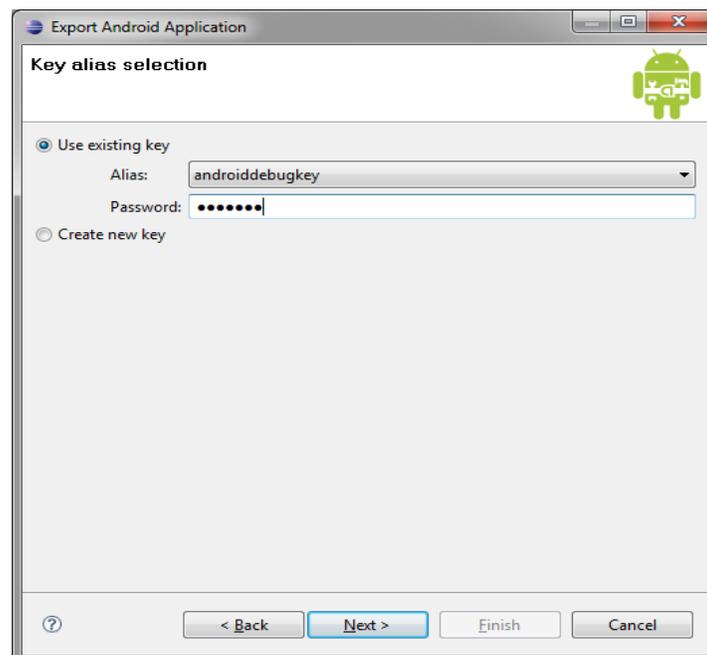
3) By clicking on “Next” you will get the project name to which we want to generate .apk file.



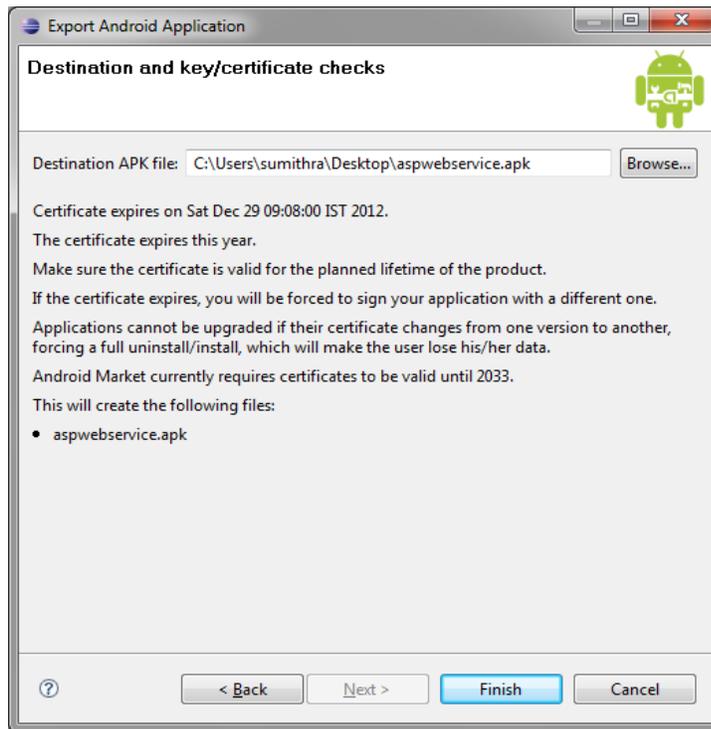
- 4) Select **Use existing keystore** and browse to the location where you have the key. Give the Password as *android* and click on “Next”.



- 5) Select alias as *androiddebugkey* and give Password as *android*. Click on “Next”.



6) Select the destination where we want to generate .apk file and click on



“Finish”. You can now install the .apk file on your Android enabled device

## 5. APP OVERVIEW

This chapter contains the screenshots of the entire project. The code for these forms and reports were provided

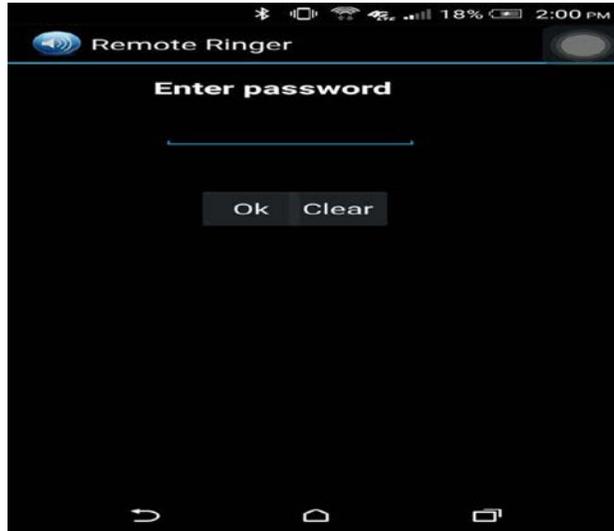


Figure 5.1 Security

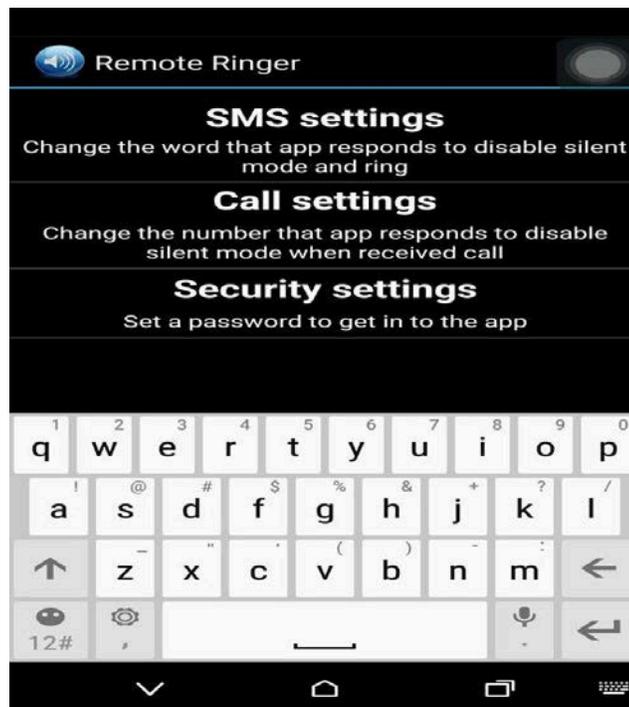


Figure 5.2 Home page

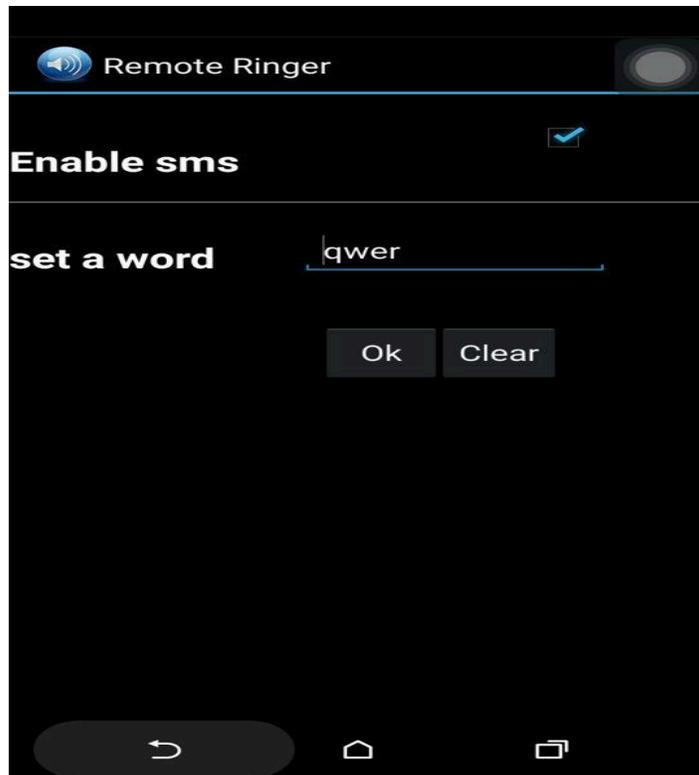


Figure 5.3 Setting the word

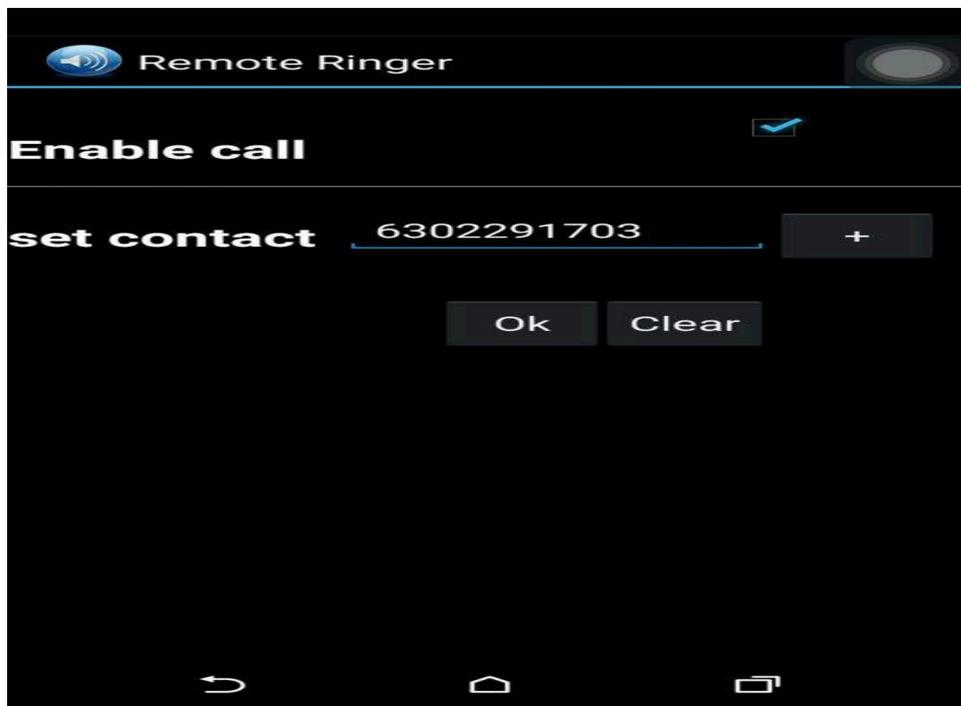


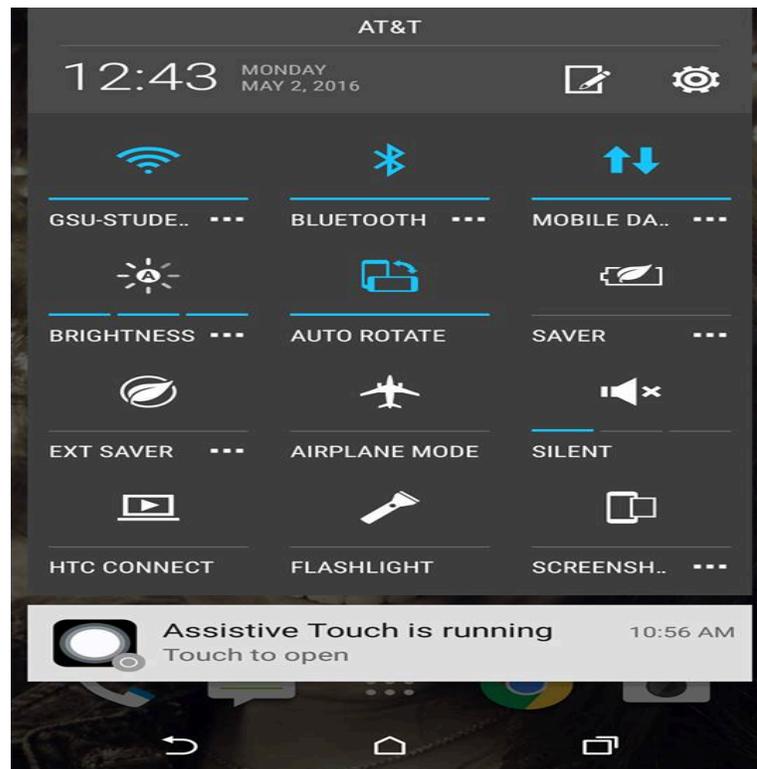
Figure 5.4 Setting the number

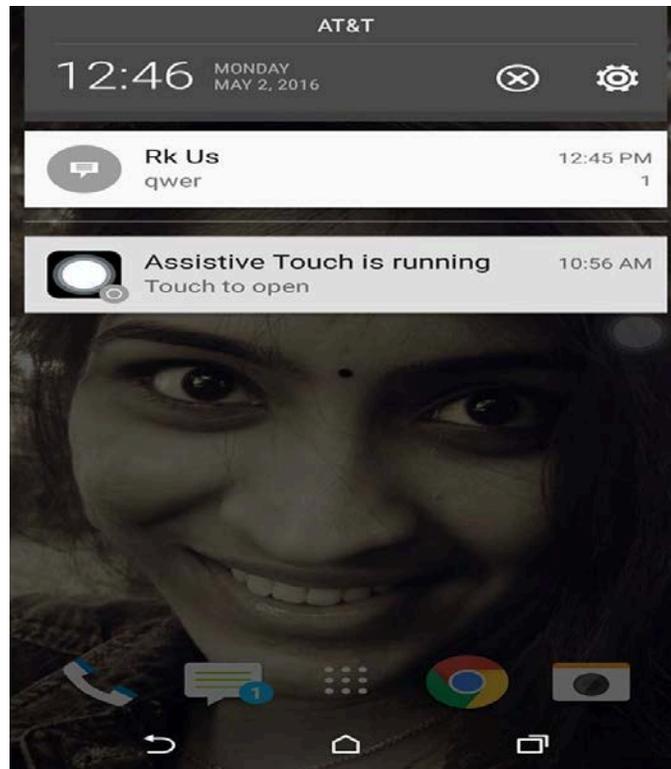
## 6. TEST PLAN REPORT

In this report, we are presenting the steps involved in testing our project. This will help the company to decide whether our project is executing correctly or not by testing our project through the steps given in this report.

### Enable SMS

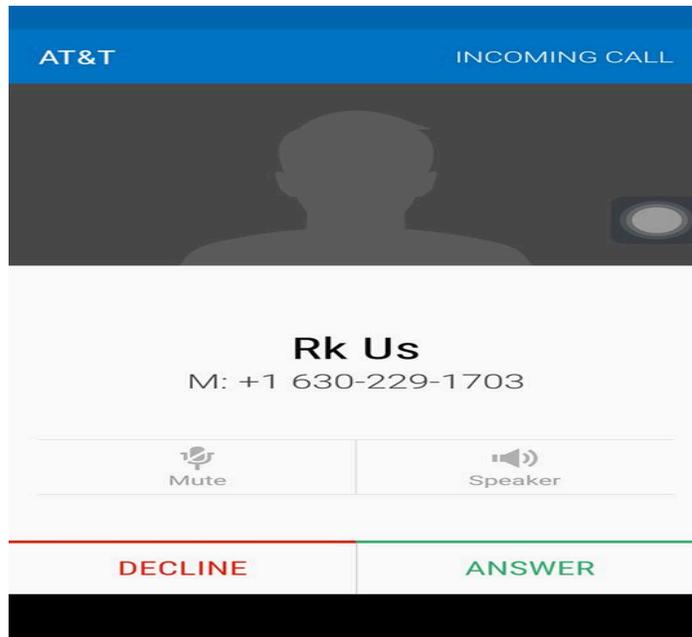
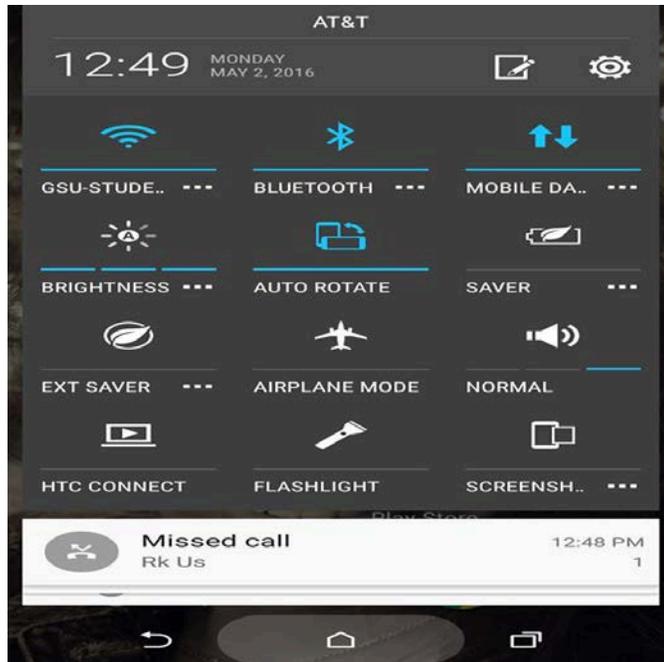
Check whether this checkbox is enabled or not. If it is enabled then only this app will check each and every incoming message. Our phone changes to general mode only if the word in this message matches with existing word.





## **Enable Call**

Check whether this checkbox is enabled or not. If it is enabled then only this app will check each and every incoming Call. Our phone changes to general mode only if the phone number matches with existing number.



## ***7.CONCLUSION & FUTURE SCOPE***

### ***7.1 CONCLUSION***

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements. This component can be easily plugged in many Android enabled devices.

### ***7.2 FUTURE SCOPE***

This module can be further enhanced by finding GPS location of the phone.

## ***8.REFERENCES***

### **Text Books:**

1. Mednieks, Dornin, Meike & Nakamura. (2011). Programming Android.
2. Reto Meier . (2009). Android Application Development.
3. Raghurama Krishnan. (2003). Database Management Systems.
4. Silberschatz, Korth. (2004). Database System Concepts.

### **Websites:**

1. <http://developer.android.com>
2. <http://www.eclipse.org/>