Implementation of an Android Application to Retrieve Information from a Lost Android Device

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Pratyusha Adusumalli
Spring 2014

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. Longzhuang Li
Committee Member

Dr. David Thomas
Committee Member
ABSTRACT

The Android mobile platform has developed from its first phone in October 2008 to being the most popular smart phone operating system in the world by 2012. The explosive growth of the platform has been a significant win for consumers with respect to competition and features. The market has been booming in the past few years that, there are now over 1,195,932 applications on the Android market. Due to the wide usage, it is necessary to provide users with security applications to manage the data in their personal smart phones.

If a user has misplaced the mobile or forgot the mobile somewhere and wanted to know the call, SMS, GPS locations etc, by using this application the user can always keep in touch with the lost phone. The security solution provided by this app requires the user to install the application with security codes for call logs, SMS and GPS tracking. User has to send an SMS with these secret codes to the mobile in order to retrieve the call logs, messages, GPS locations to the mobile from which the SMS was sent. User can also manage personal information like, deleting the call logs or messages. If the SIM card has been changed, the user will receive a notification with that information to the alternate number. Managing the personal information remotely will be of great use to the users.
# TABLE OF CONTENTS

Abstract........................................................................................................................................ ii

Table of Contents............................................................................................................................... iii

List of Figures ....................................................................................................................................... iv

List of Tables ....................................................................................................................................... viii

1. Background and Rationale............................................................................................................. 1
   1.1 Introduction................................................................................................................................. 2
   1.2 Existing Applications..................................................................................................................... 4
       1.2.1 SMS Tracker ....................................................................................................................... 4
       1.2.2 SMS Alert .......................................................................................................................... 5
   1.3 Proposed Solution....................................................................................................................... 5

2. Narrative.......................................................................................................................................... 7
   2.1 Motivation.................................................................................................................................... 7
   2.2 Proposed System......................................................................................................................... 8
   2.3 Project Scope ............................................................................................................................. 8
       2.3.1 Functionality ....................................................................................................................... 9
       2.3.2 Android OS Architecture ................................................................................................. 9

3. Project Implementation Details....................................................................................................... 11
3.1 Android Components ........................................................................................................11
3.2 Location based services used for System ..........................................................................12
3.3 Software and Hardware Requirements .............................................................................14
3.4 Functional Requirements .................................................................................................14
3.5 Android Application ..........................................................................................................16
  3.5.1 Connectivity of the Application .................................................................................16
  3.5.2 Set Secret Code and Alternate Number .......................................................................17
  3.5.3 SMS Retriever ............................................................................................................18
  3.5.4 Location Tracker ........................................................................................................20
  3.5.5 Call Log Retriever ......................................................................................................22
  3.5.6 Delete Requested SMS .............................................................................................26
  3.5.7 Delete Requested Call logs .........................................................................................28
4. System Design .....................................................................................................................31
  4.1 Design Rationale ............................................................................................................31
  4.2 Object Oriented Analysis & Design ..............................................................................32
    4.2.1 Use Case Diagrams .................................................................................................33
    4.2.2 Sequence Diagram ..................................................................................................43
5. System Implementation .......................................................................................................45
6. Evaluation and Expected Results .....................................................................................47
  6.1 Compatibility ..................................................................................................................47
  6.2 Portability .......................................................................................................................47
  6.3 Security ...........................................................................................................................47
  6.4 Scalability .......................................................................................................................47
6.5   Reliability..................................................................................................................48
       6.5.1 Attack scenario 1..........................................................................................48
       6.5.2 Attack Scenario 2.......................................................................................49

6.6   Testing......................................................................................................................49

7.    Deployment...............................................................................................................51

8.    Conclusion and Future Work..................................................................................52
       8.1 Conclusion .........................................................................................................52
       8.2 Future Work .....................................................................................................52

Bibliography and References........................................................................................53
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>System Architecture</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Structure of Android Components</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Structure of Network Service Location</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>Set 4-Digit Secret Code</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Set Alternate Phone Number</td>
<td>18</td>
</tr>
<tr>
<td>3.5</td>
<td>Receive Secret Code through SMS</td>
<td>19</td>
</tr>
<tr>
<td>3.6</td>
<td>Send Recent SMS to Requesting device</td>
<td>20</td>
</tr>
<tr>
<td>3.7</td>
<td>Receive Secret Code through SMS</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>Send Location Details</td>
<td>22</td>
</tr>
<tr>
<td>3.9</td>
<td>Receive Secret Code through SMS</td>
<td>23</td>
</tr>
<tr>
<td>3.10</td>
<td>Send Recent Missed Call Logs</td>
<td>24</td>
</tr>
<tr>
<td>3.11</td>
<td>Send Recent Dialed Call Logs</td>
<td>25</td>
</tr>
<tr>
<td>3.12</td>
<td>Send Recent Received Call Logs</td>
<td>26</td>
</tr>
<tr>
<td>3.13</td>
<td>Receive Secret Code and SMSId through SMS</td>
<td>27</td>
</tr>
<tr>
<td>3.14</td>
<td>Delete SMS and Send Success Notification</td>
<td>28</td>
</tr>
<tr>
<td>3.15</td>
<td>Receive Secret Code and Call log Id through SMS</td>
<td>29</td>
</tr>
<tr>
<td>3.16</td>
<td>Delete Call Log and Send Success Notification</td>
<td>30</td>
</tr>
<tr>
<td>4.1</td>
<td>Overall System design</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Sample Codes to Retrieve Information</td>
<td>32</td>
</tr>
<tr>
<td>4.3</td>
<td>Use Case for Complete Application</td>
<td>33</td>
</tr>
<tr>
<td>4.4</td>
<td>Use Case for Setting 4-Digit Code</td>
<td>35</td>
</tr>
</tbody>
</table>
Figure 4.5.   Use Case for Retrieving Call Logs.................................................................36
Figure 4.6.   Use Case for Retrieving SMS.........................................................................37
Figure 4.7.   Use Case for Retrieving Location .................................................................38
Figure 4.8.   Use Case for Notifying SIM Card Change.........................................................39
Figure 4.9.   Use Case for Deleting SMS.............................................................................40
Figure 4.10. Use Case for Deleting Call Logs .................................................................42
Figure 4.11. Sequence Diagram for Complete Application.................................................43
Figure 4.12. Class Diagram for Complete Application.........................................................44
Figure 5.1.   Settings Page ..................................................................................................45
Figure 5.2.   About Page ..................................................................................................45
LIST OF TABLES

Table 4.1 Use Case Description for TrackMyPhone App ...........................................34
Table 4.2 Use Case Description to Set 4-Digit Code and Alternate Number ...............35
Table 4.3 Use Case Description for Retrieving Call Logs .........................................36
Table 4.4 Use Case Description for Retrieving SMS ..................................................37
Table 4.5 Use Case Description for Retrieving Current Location ..............................38
Table 4.6 Use Case Description for Notifying change in SIM card .............................40
Table 4.7 Use Case Description for Deleting SMS .....................................................41
Table 4.8 Use Case Description for Deleting Call Logs .............................................42
Table 6.1 Test Cases .................................................................................................49
1. BACKGROUND AND RATIONALE

Due to the recent advances in the technology, the smart phones are quickly replacing the traditional mobile phones. According to Mobile OS Market Share as of 2\textsuperscript{nd} quarter 2013 surveys reveal that 79.0\% of the consumers use smart phones with Android continuing its domination in the space, nearly accounting for 48.5\% of all the smart phones handsets.

According to Stirparo, P (2013), the market reports, 90\% of the mobiles purchased in 2013 were captured by different smart phones. Smart phone is the combination of Personal Digital Assistant (PDA) and cellular phone. It is a smarter version of traditional mobile phone which is built on a mobile operating system with advanced connectivity and computing capabilities.

It helps the users to perform multiple tasks simultaneously. A smart phone usually helps the users, to perform their duties easily, instead of carrying PC’s or a PDA. They are able to perform various operations due to the different applications embedded in it. For example, the users can compose an email, send an SMS and at the same time they can make a call to a person.

Most popular mobile operating systems include Google’s Android, Apple’s iOs, Nokia’s Symbian and RIM’s BlackBerry OS. Android is a Linux based operating system, designed primarily for touch screen mobile devices such as smart phones and tablets. Users can customize the user interface such that they can place the shortcuts to various applications on the home screen, like displaying live content, for example, weather
information, emails, message data etc. Users also receive updates and notifications from
the applications, such as text messages, missed calls, new emails etc (Android OS, Wiki,
2012). Android is compatible with multiple hardware and supports various features like
Web browser, Email, Java, Video calling, Media streaming, Bluetooth, Multitasking,
External storage, Screen capturing,

We focus on Android in our project as:

1. It accounts for a huge market of smart phones, according to the survey as of July
   2013; it has the highest number of applications available for download on Google
   Play.

2. It is popular among technology organizations which focus on readymade, low-
   cost and customizable operating system.

3. It is an open source project.

Android operating system helps the different third party developers to develop
their own applications to their own enterprises, with the help of Android SDK. Over 71%
of the developers use Android software. An application in the smart phones helps the
users to do their work efficiently like e-books, managing personal information on notes,
performing bank transactions, attending the conferences etc. In September 2013, there
were more than 1 million apps available for Android, and the estimated number of
applications downloaded from Google Play was 48 billion.
1.1 Introduction

“Mobile phone security” plays a major role in the present scenario, as it involves personal and business information. There are many concerns as most of the businesses and individuals have their information stored on a smart phone, for example, planning of business meetings, organizing the work and private life etc. The information stored on a smart phone can include highly secure information ranging from personal account passwords to simple reminders. These technologies are changing the organizations systems that they highly depending on smart devices which have become the new source of threats.

Indeed, smart phones collect and execute an increasing amount of sensitive information to which access must be controlled to protect the privacy of the user and the intellectual property of the company. According to ABI Research the Mobile Security Services market will total around $1.88 billion by the end of 2013.

Most of the smart phones are targeted by attackers, as computers, since they account to sensitive information. They mainly target on weaknesses related to communication services on smart phones, for example, Short Message Services(SMS), Multimedia messaging, Wi-Fi networks, Bluetooth and GSM etc.

Cell phones present special challenges especially Android smart phones as it supports multitude of hardware, as it has no standard way to copy data out, no standardized connectors or cables or extraction protocols. The following are the few important reasons to consider securing the information in smart phone:
• **Data**: smart phones are devices for data management, therefore they may contain sensitive data like credit card numbers, authentication information, private information, activity logs (calendar, call logs) etc.

• **Identity**: smart phones are highly customizable, so the device or its contents are associated with a specific person. For example, every mobile device can transmit information related to the owner of the mobile phone contract, and an attacker may want to steal the identity of the owner of a smart phone to commit other offenses.

• **Availability**: by attacking a smart phone one can limit access to it and deprive the owner of the service.

A typical security threat to a lost phone can be caused by the above reasons. Hence, a user cannot trust any individual to obtain information from the lost or misplaced phone. Thus a security application requires controlling access to sensitive information.

### 1.2 Existing Applications

#### 1.2.1 SMS Tracker:

This application allows a user to secretly track information on a mobile like text messages, multimedia messages, GPS locations and browsing information. It is mainly useful for parents to track their children or a teenager. This application has a very narrow use i.e., only used to view the sensitive information and the user has to
pay for each service they require from the app. Moreover, misusing the application can lead to serious threats, because user is not aware of the installed application on the mobile. As a result, it can’t be used to retrieve details of all the calls, messages and locations by remotely logging into a browser. (GooglePlay, 2013)

1.2.2 SMS Alert:
This application allows the user to make the phone ring even if it is in a silent mode. This is achieved by sending a message with a keyword such that the application filters the text and allows the phone to ring, based on the received text. It also allows others to send these texts, texts can be set up initially during installation like “asap”, “urgent”, or a phone number. But this app doesn’t serve the purpose of security as it only allows the phone to ring, but not to retrieve data or lock the phone. (GooglePlay, 2013)

1.3 Proposed Solution:
It was described previously that there has been no solution to solve the security problem yet. No solution allows the user to access the content from a remote device. No solution exists to send information using a message service. The intent and main purpose of this project is to come up with an application which effectively and efficiently solves the problem. Using the application:
- User can set the pass-code and emergency phone number contacts, in the application settings
- Use this pass-code in all SMS’s for authentication and get required information
- Get Call logs when away from the cell phone
- Get text messages when away from the cell phone
- Get the position of their cell phone (in terms of latitude and longitude)
- Get Alert message when SIM card is replaced.

The background of the application involves a service which should start at boot up and continuously be running to check the incoming SMSs. When the device receives any SMS, Service will read and parse the SMS and checks the SMS syntax for the desired keywords for an authentication purpose. Once the authentication is done the service will start fetching information from the device, based on the data provided by the incoming SMS and will send it to the appropriate sender. For example, the information can be any contact details, Missed call information if any, Position of the device etc.
2. NARRATIVE

Thanks to the advent in the communication and mobile phone technology, almost everyone is carrying a smart phone these days. Several measures of digital technology are improving at exponential rates related to Moore's law (Moore, 2006, p.11), including the size, cost, density and speed of components. Due to this exponential improvement, the digital electronics has already reached to an extent where most of the daily activities can be stored on the device. For example, the modern smart phones support unlimited call entries and unlimited SMS logs, and other storage like pictures, social networking profiles, emails etc.

2.1 Motivation:

It is natural that people are curious about any kind of information. But, when it is in the case of a lost mobile discovery, this curiosity can lead to violation of personal information and exposure to sensitive data. A study according to (7) states that a total of 89 percent of devices showed attempts to access personal apps or data, 72% to private pictures app, 43% to online banking applications, 57% to saved passwords and data in text messages. From this it is clear that a person can access as much as personal information from a lost phone and can lead to serious threats to the user.
Because of the daily events being stored on the digital electronics, one can easily figure out the person’s activities. So, there is a high chance that the thief would collect information of the logs stored on the device, important messages exchanged etc. To overcome these situations and find out the way to reduce the threat to the user caused by a lost or misplaced phone.

2.2 Proposed System:

As observed from the previous studies it is very necessary for an application which can provide the user with facilities to secure personal information. Therefore, this application is being developed based on android operating system to receive and manipulate all the call log details and the Text messages that are sent and received from the smart phone. In addition this application also allows the user to receive the GPS location of the mobile and alert messages to the alternate numbers set up during the initialization of the app when the SIM card has been changed. This application is used in the perspective of mobile owner.

2.3 Project Scope:

The project consists of an Android Application which involves a service to run in background and it starts immediately at boot up and continuously be running to check the incoming SMS’s. When the device receives any SMS, Service will send the requested information to the appropriate sender. For example, the information can be any
contact details, Missed call information if any, Position of the device etc. In addition it sends alert message to the alternate number if the SIM has been changed.

The application can be distributed via email as a .apk file. The users can download the application, double click on it and tap on the downloaded application icon in order to use it. Once this is done, the tasks can be accomplished by setting up the security code and alternate numbers.

2.3.1 Functionality:

1. Read SMS
2. Parse and decode it
3. Fetch information based on request
4. Calculate Geo-position of current location
5. Send SMS

2.3.2 System Architecture:

Fig. 2.1 presents the proposed system architecture. It comprises of the following four main modules.

1. Service
2. Broadcast Receiver
3. Telephony Manager
4. Location listeners
The service module is responsible to send and receive messages. The Broadcast receiver in the android framework involves in sending device notifications, for example monitoring incoming messages etc. Telephony manager and Location listeners give information regarding the call logs and GPS location of the device.

![Figure 2.1 Architecture of Android OS](image-url)

**Figure 2.1 Architecture of Android OS**
3. PROJECT IMPLEMENTATION DETAILS

3.1 Android Components:

For the easy implementation of the Android application, the project makes use of 3 components of Android, as shown in fig 3.1.

![Figure 3.1: Structure of Android Components](image)

**Broadcast Receiver**

Broadcast Receiver is a type of component that can receive and respond to any broadcast announcements. It continuously checks for the incoming notifications on the device. For this application it is used to check for the incoming SMS and Calls.

**Service**

A Service is a body of code that runs in the background. It can run in its own process, or in the context of another application's process, depending on its needs. Other components "bind" to a Service and invoke methods on it via remote procedure calls. For example, a Service in this application is; even when the user quits the “TrackMyPhone”
application UI, user probably still intends for the application to run in the background. A Service keeps the application working even when the UI has completed.

**Content Provider**

Content Provider is a data storehouse that provides access to data on the device; the best example is the Content Provider that's used to access the user's list of contacts, SMS etc. This application can access data that other applications have exposed via a Content Provider. In other terms it acts like database storage inside the android system itself, rather than using an external database.

3.2 Location based services in Android used for the application:

Android's Network Location Provider determines user location using cell tower and Wi-Fi signals. The purpose of location-based services is to find the Physical location of the device. Access to the location-based services is handled by the LocationManager system Service. To access the Location Manager, request an instance of the LOCATION_SERVICE using the get System Service() method. To get access to GPS hardware of android, it is requested using following statement

`LocationManager.GPS_PROVIDER;`

In order to use a Network Service Location, it uses the current cell ID to locate the Base Transceiver Station (BTS) that the mobile phone is interacting with and the location of that BTS. It is the most basic and cheapest method, as the advantage is that no additional cost is attached to the handset or to the network to enable this service. To get
access to Network Provider on android, it is requested using following statement

```java
LocationManager.NETWORK_PROVIDER;
```

![Figure 3.2: Structure of Network Service Location [11]](image-url)

**Geocoding and Reverse Geocoding**

Geocoding allows to translate between street addresses and longitude/latitude map coordinates. The Geocoding lookups are done on the server. The Geocoder class provides access to two Geocoding functions:

- **Forward Geocoding**
  
  Forward Geocoding converts the address into latitude and longitude.

- **Reverse Geocoding**
  
  Reverse Geocoding converts latitude and longitude to corresponding address
3.3 Software and Hardware requirements:

- Eclipse
- Android SDK (Software Development Kit)
- Windows OS
- Android device (Version > 2.1)

3.4 Functional Requirements:

For easier explanation of the project, the project is divided into 6 different Module’s, the Call log Retriever is again divided into 3 sub-modules as follows

1. Set Secret Code and Alternate Number
2. SMS Retriever
3. Location Tracker
4. Delete SMS
5. Delete Call log
6. Call Log Retriever
   - Retrieve Missed Calls
   - Retrieve Dialed Calls
   - Retrieve Received Calls
1. Set Secret Code and Alternate Number

- Be able to let user set the 4 digit secret code
- Be able to let user set an alternate phone number

2. SMS Retriever

- Be able to recognize the secret code received through SMS.
- Be able to send recent SMS on the mobile through SMS.

3. Location Tracking

- Be able to detect the current location of Android device.
- Be able to retrieve the device, SIM card & location details.
- Be able to send retrieved details through SMS.

4. Call Log Retriever

- Be able to recognize the secret code received through SMS.
- Be able to send recent dialed call logs list on the mobile through SMS.
- Be able to send recent missed call logs list on the mobile through SMS.
- Be able to send recent received call logs list on the mobile through SMS.

5. Delete SMS

- Be able to recognize the secret code and SMS ID received through SMS.
• Be able to delete the SMS with requested ID on the mobile and send success notification through SMS.

6. **Delete Call log**

• Be able to recognize the secret code and Call log ID received through SMS.
• Be able to delete the Call log with requested ID on the mobile and send success notification through SMS.

7. **Notify when SIM Card is Changed**

• Be able to detect the change in mobile phone SIM card and send Notification to the Alternate Number.

### 3.5 Android Application:

Due to inexperience and lack of knowledge in Android SDK, the initial phase was met with complications because of a limited amount of knowledge of android application and its programming pattern.

#### 3.5.1 Connectivity of the Application:

The connectivity of the application deals with one main component:

a. Establish connection between Android Phone and the Application.

**a. Establish connection between Android Phone and the Application:**

In first stage, connection should be established between the Android Phone and the application. Because, as soon as the connection is established, the application should start continuously capturing the history (message and call details).
There will be an automated pooling, defined in an application. As soon as a new event (Call or SMS) occurs, the android system sends a notification to the application with details through the android application component **Broadcast Receiver**. The application, as soon as it receives the notification through the receiver, collects the information and stores using the inbuilt database i.e. **Content Provider**.

### 3.5.2 Set Secret Code and Alternate Number:

This has two functions:

- a. Set 4-digit Secret Code
- b. Set an Alternate Phone Number

#### a. Set 4-digit Secret Code:

User will be able to set a 4-digit secret code, which can be later used to retrieve data like call logs, SMS, location when the phone is lost or misplaced.

The details are shown in Figure 3.3.
b. Set an Alternate Phone Number:

User will be able to set an alternate phone number, which can be later used to receive notification when the phone stolen and replaced with a new SIM card. The details are shown in Figure 3.4.
3.5.3 SMS Retriever:

SMS Retriever has two functions:

a. Receive the secret code through SMS

b. Send the recent SMS on the phone to requesting device.

a. Receive the Secret Code through SMS:

The application should continuously pool the incoming messages with the desired format and secret code. The details are shown in Figure 3.5
b. **Send the Recent SMS on the Phone to Requesting device:**

Application sends the most recent SMS on the lost mobile to the requesting device in a SMS format with an ID for each SMS. The details are shown in Figure 3.6.
3.5.4 Location Tracker:

The Location Tracker has four functions:

a. Receive the secret code through SMS

b. Detect the current location of Android device.

c. Retrieve the device, SIM card & location details.

d. Send retrieved details through SMS.
a. **Receive the Secret Code through SMS:**

The application should continuously poll for the incoming messages with the desired format and secret code. The details are shown in Figure 3.7

![Receive Secret Code through SMS](image)

**Figure 3.7 Receive Secret Code through SMS**

b. **Send Retrieved Details through SMS:**

The application verifies the requested format, and retrieves the current location of the android device. It sends the retrieved location details through SMS to the requesting device. The details are shown in Figure 3.8
3.5.5 Call Log Retriever:

Call Log Retriever has four functions:

a. Receive the secret code through SMS

b. Send the 5 recent missed calls on the phone to requesting device.

c. Send the 5 recent dialed calls on the phone to requesting device.

d. Send the 5 recent received calls on the phone to requesting device.

Figure 3.8 Send Location Details
a. **Receive the Secret Code through SMS:**

The application should continuously pool the incoming messages with the desired format and secret code. The details are shown in Figure 3.9

![Image of SMS messages](image)

**Figure 3.9 Receive Secret Code through SMS**

b. **Send the 5 Recent Missed Calls on the Phone to Requesting device:**

Application sends the most recent missed call logs on the lost mobile to the requesting device in a SMS format with an ID for each log. The details are shown in Figure 3.10
c. **Send the 5 Recent Dialed Calls on the Phone to Requesting device:**

Application sends the most recent dialed call logs on the lost mobile to the requesting device in a SMS format with an ID for each log. The details are shown in Figure 3.11
d. **Send the 5 Recent Received Calls on the Phone to Requesting device:**

Application sends the most recent received call logs on the lost mobile to the requesting device in a SMS format with an ID for each log. The details are shown in Figure 3.12.
3.5.6 Delete Requested SMS

Delete SMS has two functions:

a. Receive the secret code and SMS ID through SMS

b. Delete SMS with specified ID and send the “Successfully Deleted SMS” message to requesting device.

a. Receive the secret code and SMS ID through SMS:

The application should continuously pool the incoming messages with the desired format, SMS ID and secret code. The details are shown in Figure 3.13
b. Delete SMS with specified ID and send the “Successfully Deleted SMS” message to requesting device:

Application deletes the SMS with received ID and sends a “Successfully deleted SMS” message to the requesting device. The details are shown in Figure 3.14
3.5.7 Delete Requested Call log

Delete Call log has two functions:

a. Receive the secret code and Call log ID through SMS

c. Delete Call log with specified ID and send the “Successfully Deleted SMS” message to requesting device.

Figure 3.14 Delete SMS and send Success notification to Requesting device
a. **Receive the secret code and Call log ID through SMS:**

The application should continuously pool the incoming messages with the desired format, Call log ID and secret code. The details are shown in Figure 3.15

![Figure 3.15 Receive Secret Code and Call log ID through SMS](image)

b. **Delete Call log with specified ID and send the “Successfully Deleted SMS” message to requesting device:**

Application deletes the Call log with received ID and sends a “Successfully deleted SMS” message to the requesting device. The details are shown in Figure 3.16
Figure 3.16 Delete Call log and send Success notification to Requesting device
4. SYSTEM DESIGN

This chapter discusses the design of the architecture of the overall system. For efficiency and usability, the project is strictly adhered to the standards of Object Oriented programming. The following section discusses the design in brief and analyse the use cases of the system

4.1 Design Rationale

![Android Application Design](image)

**Figure 4.1 Overall System Design**

After the initialization of the app, it starts continuously pooling for the incoming data and looks for SMS with an application specified format. For this the application registers with various components like Broadcast Receivers, SMS, Telephony Manager, Content Provider, Location and Services. The codes used to retrieve this information are shown below in Figure 4.2.
Code to retrieve SMS :- XXXX::SMS

Code to delete SMS:- XXXX::SMSDEL::SMSId

Code to retrieve Missed Call logs :- XXXX::CALLS::MISSED

Code to retrieve Dialed Call logs :- XXXX::CALLS::DIALED

Code to retrieve Received Call logs :- XXXX::CALLS::RECEIVED

Code to delete a particular Call log :- XXXX::CALLDEL::ID

Code to delete all Call logs :- XXXX::ALLDEL

Code to retrieve Location :- XXXX::LOC

Code to retrieve IMEI :- XXXX::IMEI

Figure 4.2 Sample codes to retrieve information

In case of valid code it further checks for the instructions like LOC, SMS, Missed calls etc. Based on the instruction in the SMS, application uses relevant services and performs the operation. For example, if it receives “LOC” in SMS, then it uses Location manager service to get the device location. If it receives “MISSED CALLS” in SMS, then it uses Telephony services to get the call details.

4.2 Objected Oriented Analysis and Design

This application is designed following all UML design guidelines and meets the standards of Object Oriented programming. The use case diagrams and sequence diagram are explained below.
4.2.1 Use-case diagrams

Use case diagrams describe a system which involves a set of use cases and a set of actors. It gives a complete description of who uses the application or system and what actions can the users perform on the system.

Use Case for TrackMyPhone Application:

![Use case diagram for TrackMyPhone application]

Figure 4.3 Use case diagram for the TrackMyPhone application
Use Case Description for Figure 4.3:

Table 4.1

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>TrackMyPhone Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows users to set a 4-digit secret code and alternate number which can be used to retrieve SMS, Call logs and Location later.</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application User</td>
</tr>
<tr>
<td>Precondition:</td>
<td>User must initialize for the first time and run the application.</td>
</tr>
<tr>
<td>Trigger:</td>
<td>User clicks the “TrackMyPhone” app on the home screen of the Android device.</td>
</tr>
<tr>
<td>Main Successful Scenario:</td>
<td>The app displays the information such as:</td>
</tr>
<tr>
<td></td>
<td>1. Set 4-digit secret code and Set alternate phone number</td>
</tr>
<tr>
<td></td>
<td>2. Send recent SMS to the requested number</td>
</tr>
<tr>
<td></td>
<td>3. Send recent call logs to the requested number</td>
</tr>
<tr>
<td></td>
<td>4. Send current location of the device</td>
</tr>
</tbody>
</table>
Use Case to Set 4-digit Code and Alternate Number:

![Use case diagram for setting 4-digit code and Alternate Number](image)

**Figure 4.4 Use case diagram for setting 4-digit code and Alternate Number**

**Use Case Description for Figure 4.4:**

<table>
<thead>
<tr>
<th><strong>Table 4.2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TITLE:</strong> Set 4-dit Code and Alternate Number</td>
</tr>
<tr>
<td><strong>Summary:</strong> This use case allows users to set a 4-digit code and alternate number, which are later used to retrieve information.</td>
</tr>
<tr>
<td><strong>Actors:</strong> Application User</td>
</tr>
<tr>
<td><strong>Precondition:</strong> User must initialize for the first time and run the application.</td>
</tr>
<tr>
<td><strong>Trigger:</strong> User will be able to set the code and alternate number successfully.</td>
</tr>
<tr>
<td><strong>Main Successful Scenario:</strong> The App stores information such as:</td>
</tr>
<tr>
<td>1. Save the 4-digit Code</td>
</tr>
<tr>
<td>2. Save Alternate Number</td>
</tr>
</tbody>
</table>
Use Case for Retrieving Call Logs:

Figure 4.5 Use case diagram for Retrieving Call Logs

Use Case Description for Figure 4.5:

Table 4.3

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Retrieve Call logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows users to send a 4-digit code to the lost phone. Then the application identifies the code and sends recent missed calls, dialed calls and received calls list in an SMS format to the requested number.</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application User and TrackMyPhone App</td>
</tr>
<tr>
<td>Precondition:</td>
<td>User must initialize for the first time and run the application. Then set the 4-digit code and alternate number</td>
</tr>
<tr>
<td>Trigger:</td>
<td>User receives recent call log list to the number from which 4-digit code SMS</td>
</tr>
</tbody>
</table>
Main Successful Scenario: The App sends information such as:
3. Send recent Received calls list
4. Send recent Missed calls list
5. Send recent Dialed calls list

Use Case for Retrieving SMS:

![Use case diagram for retrieving SMS](image)

Figure 4.6 Use case diagram for retrieving SMS

Use Case Description for Figure 4.6:

Table 4.4

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Retrieve recent SMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows user to send a 4-digit code to the lost phone. Then the application identifies the code and sends recent SMS to the requested number.</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application User and TrackMyPhone App</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>User must initialize for the first time and run the application. Then set the 4-digit code and alternate number</td>
</tr>
</tbody>
</table>
**Trigger:** User receives recent SMS list to the number from which 4-digit code SMS was sent

**Main Successful Scenario:** The App sends information such as:

1. Send recent Sent and Received SMS list

---

**Use Case for Retrieving Location:**

![Use Case Diagram for Retrieving Location](image)

**Figure 4.7 Use Case diagram for Retrieving Location**

---

**Use Case Description for Figure 4.7:**

**Table 4.5**

<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th>Retrieve Current Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>This use case allows user to send a 4-digit code to the lost phone. Then the application identifies the current location and sends it to the requested number.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Application User and TrackMyPhone App</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>User must initialize for the first time and run the application. Then set the 4-digit code and alternate number</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Trigger:</td>
<td>User receives current location of lost device to the phone that sent the 4-digit code.</td>
</tr>
<tr>
<td>Main Successful Scenario:</td>
<td>The App sends information such as:</td>
</tr>
<tr>
<td></td>
<td>1. Send current location of the device in an SMS format</td>
</tr>
</tbody>
</table>

Use Case to Notify SIM Card Change:

![Use case diagram to Notify SIM Card Change](image)

Figure 4.8 Use case diagram to Notify SIM Card Change
Use Case Description for Figure 4.8:

Table 4.6

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Notify SIM Card change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case notifies the user with an alert message. Then the application identifies the current location and sends it to the requested number.</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application User and TrackMyPhone App</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>User must initialize for the first time and run the application. Then set the 4-digit code and alternate number</td>
</tr>
<tr>
<td>Trigger:</td>
<td>User receives current location of lost device to the phone that sent the 4-digit code.</td>
</tr>
<tr>
<td>Main Successful Scenario:</td>
<td>The App sends information such as:</td>
</tr>
<tr>
<td>1.</td>
<td>Send current location of the device in SMS format</td>
</tr>
</tbody>
</table>

Use Case for Deleting SMS:

Figure 4.9 Use case diagram for Deleting SMS
Use Case Description for Fig 4.9:

### Table 4.7

<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th>Delete SMS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>This use case receives the 4-digit secret code and SMS ID. It deletes the specified SMS and sends successful deletion SMS to the requested number.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Application User and TrackMyPhone App</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>User must initialize for the first time and run the application. Then set the 4-digit code and alternate number.</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>User receives successful deletion SMS and also the requested SMS gets deleted.</td>
</tr>
<tr>
<td><strong>Main Successful Scenario:</strong></td>
<td>The App sends information such as:</td>
</tr>
<tr>
<td></td>
<td>1. Receives SMS ID of the SMS that has to be deleted.</td>
</tr>
<tr>
<td></td>
<td>2. “Successfully deleted SMS” to the requested number.</td>
</tr>
</tbody>
</table>
Use Case for Deleting Call Logs:

Figure 4.10 Use case diagram for Deleting Call Logs

Use Case Description for Fig 4.10:

Table 4.8

<table>
<thead>
<tr>
<th>TITLE</th>
<th>Delete Call Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case receives the 4-digit secret code and Call log ID. It deletes the specified Call log and sends successful deletion SMS to the requested number.</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application User and TrackMyPhone App</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>User must initialize for the first time and run the application. Then set the 4-digit code and alternate number.</td>
</tr>
<tr>
<td>Trigger:</td>
<td>User receives successful deletion SMS and also the requested Call log gets deleted.</td>
</tr>
<tr>
<td>Main Successful Scenario:</td>
<td>The App sends information such as:</td>
</tr>
<tr>
<td></td>
<td>1. Receives Call log ID of the Call log that has to be deleted.</td>
</tr>
</tbody>
</table>
2. “Successfully deleted SMS” to the requested number.

4.2.2 Sequence Diagram:

A Sequence diagram describes how a set of objects communicate with each other to perform a task. Fig 4.9 gives graphical representation of the processes involved in the application.

Figure 4.11 Sequence diagram for TrackMyPhone Application
4.2.3 Class Diagram:

Figure 4.12 Class Diagram for TrackMyPhone Application
5 SYSTEM IMPLEMENTATION

The application is written for Android using Android SDK which is open source toolkit released by Google. There are a total of 15 different classes with the total number of lines of code crossing over 2000 lines and about 30 methods for accessing the information from the lost Android device. The jar files used for the app are “android4.2.jar” and “android-support-4.jar”. The name of the package which is used to deploy the application is “Com.android.trackmyphone” and the final file format that has to be downloaded by the user is “TrackMyPhone.apk”. The total size occupied by the application is about 200 kb, which is very cheap compared to present storage capabilities.

*The app is built on my laptop which has Intel i3 processor and 4 GB of RAM.*

In the app, the interface has been designed to look simple yet powerful with user friendly interface. The elements of the user interface are built in elements and system components. The interface of the application consists of:

1. Settings
2. About

The other functionalities of the application run on the background, which helps the user to retrieve desired information. Please find the figures for Retrieving Call Logs, Retrieving SMS, Retrieving Location in the previous section at Figures 3.6, 3.8, 3.10, 3.11, 3.12 respectively. Home Settings and About screenshots are shown at Figures 5.1 and 5.2 respectively.
Figure 5.1 Settings Page
In the perspective of user, the app is installed after purchasing the device before the user even starts using it. But, in the present scenario, for testing purposes, it has been installed on an already used phone which has some existing call logs and SMS logs. So, it retrieves the most recent SMS and Call Logs on the device. All the requested logs are sent to the requesting device once it receives the specified code, when the app is launched for the first time.

Figure 5.2 About Page
6: EVALUATION AND EXPECTED RESULTS

6.1 Compatibility

The app was tested on the emulator while developing the app. Once finished, it was tested on a couple of live devices running various versions. Also, different people tested the app on variety of mobile devices. Based on the working, some features are modified so as to make it more elegant and useful. The app was compatible with all the versions. Although due to the variation of read and write access permissions on different versions, few changes are done on the device, to make it compatible. The app is tested successfully in different localities since the code depends on the phone numbers which are in different format in different places.

6.2 Portability:

The project is built using JAVA and can be run on any device which uses android OS.

6.3 Security:

The application will prompt the user for upgrading and downloading new features updated by the developer.

6.4 Scalability

Since all the call records and SMS records and location details are sent in an SMS format, the scalability was considered an issue. Because in the initial stages of the
project, all the SMS and Call logs were retrieved which can highly cost the user and occupy lot of memory space on the retrieved device. But, later on the retrieval was limited to recent 5 SMS and 5 Call logs, which is pretty less storage compared to the present day storage capability. And also it does not look messed up to the user as limited data is retrieved. This number can be incremented in later versions.

6.5 Reliability

The main aim is to create an application which helps the user to retrieve information when the phone is lost or misplaced. The user should be able to achieve this using any basic device, irrespective of it being only android or only smart phones, because, it is not always possible to have access to the internet, laptop or a smart phone. As already stated, the app automatically records all phone calls and SMS messages without user intervention to the content provider and also updates with new records. Few attack scenarios to change the secret code and not let the user to retrieve data and notification on SIM card change, the methods followed to safeguard against them are listed below.

6.5.1 Attack Scenario 1:

Since the app has information on how to set a secret code and alternate number, if a person gets access to the lost mobile, and gets aware of an application running in the background, this attacker can change the secret code and alternate number. This may cause the use from not being able to retrieve the information.

Method to safeguard:

This is not yet taken care in the project and is one of the only shortcoming to the
app's reliability as of now. But if a system provider enables a security lock to the device in order to open it, this issue can be resolved.

6.5.2 Attack Scenario 2:

From the Android version 4.0, OS has provided an option to disable system apps. If the user disables your app, what happens?

Method to safeguard:

This is not yet taken care in the project and is one of the only shortcoming to the app's reliability as of now. But if a system provider pre-installs it, he can disable user disabling the app and thus will be foolproof.

6.6 Testing:

Table 6.1, gives details of the unit tests applied to the system, in order to check if each function of the system works correct individually.

<table>
<thead>
<tr>
<th>SIN</th>
<th>SCENARIOS</th>
<th>EXPECTED RESULT</th>
<th>ACTUAL RESULT</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Install TrackMyPhone.apk file on Android phone</td>
<td>Installation Successful</td>
<td>Installation Successful</td>
<td>Success</td>
</tr>
<tr>
<td>2</td>
<td>Check whether UI Is Displaying On screen</td>
<td>Display UI</td>
<td>Display UI</td>
<td>Success</td>
</tr>
<tr>
<td>3</td>
<td>Set 4-digit Secret Code</td>
<td>Code set</td>
<td>Code set</td>
<td>Success</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Status</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------</td>
<td>-------------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Set Alternate Phone Number</td>
<td>Alternate Number set</td>
<td>Alternate Number set</td>
<td>Success</td>
</tr>
<tr>
<td>5</td>
<td>Send SMS to Android device</td>
<td>SMS sent</td>
<td>SMS Sent</td>
<td>Success</td>
</tr>
<tr>
<td>6</td>
<td>Receive SMS on Android device</td>
<td>SMS Received</td>
<td>SMS Received</td>
<td>Success</td>
</tr>
<tr>
<td>7</td>
<td>Read Contents Of SMS</td>
<td>Contents Read</td>
<td>Contents Read</td>
<td>Success</td>
</tr>
<tr>
<td>8</td>
<td>Send recent SMS list</td>
<td>5 recent SMS sent</td>
<td>5 recent SMS sent</td>
<td>Success</td>
</tr>
<tr>
<td>9</td>
<td>Send recent Call logs list</td>
<td>5 recent dialed, missed and received logs sent</td>
<td>5 recent dialed, missed and received logs sent</td>
<td>Success</td>
</tr>
<tr>
<td>10</td>
<td>Delete requested SMS</td>
<td>SMS deleted</td>
<td>SMS deleted</td>
<td>Success</td>
</tr>
<tr>
<td>11</td>
<td>Delete requested Call logs</td>
<td>Call log deleted</td>
<td>Call log deleted</td>
<td>Success</td>
</tr>
<tr>
<td>12</td>
<td>Notify if SIM card is changed</td>
<td>Notification received</td>
<td>Notification received</td>
<td>Success</td>
</tr>
<tr>
<td>13</td>
<td>Retrieve Latitude And Longitude</td>
<td>Latitude And Longitude</td>
<td>Latitude And Longitude</td>
<td>Success</td>
</tr>
</tbody>
</table>
7. DEPLOYMENT

Deployment encompasses all the processes involved in order to make a new software or hardware up and running properly in its environment, including installation, configuration, running, testing, and making required modifications.

**Software deployment** is all of the activities that make a software system available for use.

Android application can be deployed multiple ways:

1. This application is developed using eclipse, hence for deploying click run application in eclipse's launch menu. For this enable "USB Debugging Mode" on android phone, which is available in the application menu.
2. Download the .apk file on to the Android device in order to install the application.
8. CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

The TrackMyPhone mobile application is a unique and efficient application, which is used to retrieve basic information from a misplaced or lost Android phone. All the features work on SMS basis. Therefore, incoming SMS format plays a vital role. This application running in the mobile continuously monitors all the incoming messages. If the SMS matches the format which the application is designed with, it parses the SMS and performs the desired task, for example, retrieving SMS, Call logs, Location etc. It has various features, which will be useful to enhance the existing application. Application remains different from the existing systems as it is not only the GPS value it makes use of but also works on GSM/ text messaging services which makes the application a simple and unique one.

8.2 FUTURE WORK

1. Lock device remotely, wipe memory to keep mobile private data safe.

2. Control Android mobile remotely via a web-based interface through TrackMyPhone itself, which includes a client application also along with the server application

3. Connect the application to an external database to store the logs and data with secured features.
4. Increment the retrieval capacity of the application.

5. Provide a lock screen to the application, such that the thief should not be able to alter the code inside the application.

6. Instead of sending multiple text messages to retrieve and delete data, allow user to delete the data with only sending a single text message.


