

California State University, San Bernardino

**CSUSB ScholarWorks**

---

Theses Digitization Project

John M. Pfau Library

---

2012

## Cell broadcast simulation in Android

Uzzal Chandra Das

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Digital Communications and Networking Commons](#)

---

### Recommended Citation

Das, Uzzal Chandra, "Cell broadcast simulation in Android" (2012). *Theses Digitization Project*. 4085.  
<https://scholarworks.lib.csusb.edu/etd-project/4085>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

# CELL BROADCAST SIMULATION IN ANDROID

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

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

---

by  
Uzzal Chandra Das  
March 2012

CELL BROADCAST SIMULATION IN ANDROID

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

by  
Uzzal Chandra Das  
March 2012

Approved by:

  
Dr. Kay Zermouden, Advisor, Computer  
Science And Engineering

  
Dr. Teng Yu

  
Dr. Haiyan Qiao

3/16/12

---

Date

## ABSTRACT

With broadcasting, a message can be sent to a large number of recipients instantly. As all the devices listen to a particular radio frequency, there is little data traffic. The Cell Broadcast Simulation App on Android platform is an attempt to make the cell broadcasting possible in the USA. Although broadcasting is not available for regular users in USA, this app will provide a chance to realize the usefulness of broadcasting in mobile devices. This app simulates important concepts of broadcasting using text messages or Short Message Service. The user of the app is not required to maintain or select contacts manually. The app maintains and updates the contacts itself. The user just installs the app and sends a message and whoever has the app already installed will receive the message. The app has a radius feature. The user has the option to select the radius of his/her message to send and receive. If user A's radius covers user B's current geographical position and B's radius covers A's position, then a message sent by A will be received by B. This feature gives complete control to the user over the broadcasting range. In addition, the app has an automatic message filtering and saving system. A user can set his/her own keyword to filter the incoming messages. Incoming messages with the filter keyword will be saved under that keyword and the user can access those messages at a later time. The user can modify, add, or delete filters.

## ACKNOWLEDGMENTS

It is a wonderful opportunity for me to study at the California State University, San Bernardino. I would like to thank all our faculty members of the Department of Computer Science and Engineering, who are always sincere, kind, patient and helpful. I would like to specially thank my committee members, Dr. Kay Zemoudeh, Dr. Tong Yu and Dr. Haiyan Qiao to guide me through this project with their invaluable advice and support. I would like to thank Dr. Josephine Mendoza, Graduate Coordinator, for all the guidance and advice to continue and complete my Master's degree. It is a great experience and I have learned a lot from all the dedicated and experience individuals at CSUSB.

## TABLE OF CONTENTS

ABSTRACT .....	iii
ACKNOWLEDGMENTS .....	iv
LIST OF TABLES .....	viii
LIST OF FIGURES .....	x
CHAPTER ONE: INTRODUCTION .....	1
1.1 Purpose of Project .....	2
1.2 Scope of Project .....	3
1.3 Significance of Project .....	4
1.4 Glossary .....	5
CHAPTER TWO: OVERALL DESCRIPTION	
2.1 Project Design .....	10
2.2 Project Perspective .....	10
2.3 System Environment .....	11
2.4 Software Interface .....	14
2.5 Communication Interface .....	14
2.6 Hardware Interface .....	14
2.7 Functional Requirements Specification .....	15
2.7.1 Sender's Use Case .....	15
2.7.2 Receiver's Use Case .....	16
2.7.3 Adding A New User .....	18
2.7.4 Deleting An Existing User .....	19

2.7.5 Selecting The GPS Radius .....	20
2.7.6 Sending A Message .....	20
2.7.7 Receiving A Message .....	21
2.7.8 Adding Filter .....	22
2.7.9 Accessing Filtered Messages .....	23
2.7.10 Deleting Filter Keyword .....	24
2.7.11 Deleting A Filtered Message .....	25
2.7.12 Notification Turn On Or Off .....	26
2.7.13 Deleting A Message .....	26
2.7.14 Deleting All Messages .....	27
2.7.15 Exiting The Application .....	27
2.7.16 Accessing Other Users List .....	28
2.8 Detailed Non-Functional Requirements .....	28
2.8.1 Logical Structure Of The Data .....	28
2.9 User Characteristics .....	32
2.10 Non-Functional Requirements .....	32
 <b>CHAPTER THREE:DESIGN</b>	
3.1 User Interface Design .....	33
3.2 Screen Descriptions .....	33
3.2.1 Main Screen .....	34
3.2.2 Other Users Screen .....	34
3.2.3 Filter Screen .....	35

3.2.4 Saved Filtered Message Screen .....	35
3.2.5 Class Diagrams .....	36
<b>CHAPTER FOUR: IMPLEMENTATION</b>	
4.1 Implementation Outlines .....	42
4.2 Interaction .....	43
4.3 Functionality .....	43
<b>CHAPTER FIVE: TESTING .....</b>	<b>57</b>
<b>CHAPTER SIX: MAINTENANCE .....</b>	<b>58</b>
<b>CHAPTER SEVEN: CONCLUSION AND ENHANCEMENT</b>	
7.1 Conclusion .....	59
7.2 Enhancements .....	59
<b>REFERENCES .....</b>	<b>60</b>



## LIST OF TABLES

Table 1. Adding A New User .....	18
Table 2. Deleting An Existing User .....	19
Table 3. Selecting The GPS Radius .....	20
Table 4. Sending A Message .....	20
Table 5. Receiving A Message .....	21
Table 6. Adding Filter .....	22
Table 7. Accessing Filtered Message .....	23
Table 8. Deleting Filter Keyword .....	24
Table 9. Deleting A Filtered Message .....	25
Table 10. Notification Turn On Or Off .....	26
Table 11. Deleting A Message .....	26
Table 12. Deleting All Messages .....	27
Table 13. Exiting The Application .....	27
Table 14. Accessing Other Users List .....	28
Table 15. Contact Database .....	29
Table 16. Message Database .....	29
Table 17. Temporary Message Saver Database .....	30
Table 18. Filter Keyword Database .....	31
Table 19. Filtered Messages Database .....	31
Table 20. Main Screen .....	34
Table 21. Other Users Screen .....	35

Table 22. Filter Screen .....	35
Table 23. Saved Filter Screen .....	36

## LIST OF FIGURES

Figure 1. System Environment .....	11
Figure 2. Adding A New User .....	12
Figure 3. Deleting A Particular User .....	13
Figure 4. Send Message Use Case .....	15
Figure 5. Receiving Message Use Case .....	16
Figure 6. Filtering Message Use Case .....	17
Figure 7. CellBroadcast Class .....	37
Figure 8. Receiver Class .....	38
Figure 9. UsersCBapp Class .....	39
Figure 10. Filter Class .....	40
Figure 11. SavedFilter Class .....	40
Figure 12. Class Diagram .....	41
Figure 13. Main Screen After Opening The App .....	44
Figure 14. Radius Dialog To Set The Radius In Miles .....	45
Figure 15. Option Menu .....	46
Figure 16. Other Users Screen .....	48
Figure 17. Filter Screen .....	49
Figure 18. Saved Filter Message .....	50
Figure 19. Normal Message And Filtered Message .....	51
Figure 20. Enlarged Edit Message Dialog .....	52
Figure 21. Deleting A Message On The Main Screen .....	54

Figure 22. Deleting A User By The Root User ..... 55

## CHAPTER ONE

### INTRODUCTION

Cell phones use radio frequencies to communicate. Two different radio frequencies create a channel to make communication possible between cell phones. In general, this type of communication is a point-to-point connection and each connection uses a single channel to make a call or send a text message. Point-to-point has many limitations. Only two users can use it to communicate. So, if many users want to communicate with each other through text messages at the same time, then point-to-point communication is not very efficient. For a large group, many point-to-point connections have to be established for proper communication. There is a more efficient solution to this problem: Broadcasting. Only a single channel is required to broadcast data or messages among the cell phones (Mobile phones). In broadcasting, many cell phones can listen to a particular channel provided by the Service Provider. Broadcasting is a simple way to reach many users by using only a single channel with less traffic and overhead. Since cell phones have become a common utility and most people are using smart phone nowadays, it is easier to reach an individual using cell phone. In the time of emergency, when it is required to reach a large number of people, cell broadcasting is the most effective way. Social communication at the local level with less traffic is another important application of cell broadcasting and there are many other uses.

## 1.1 Purpose of Project

Since no promising application of Broadcasting exists, service providers do not provide broadcast channels in the US[14]. A prototype can be very helpful to investigate the applications of broadcasting, and Cell Broadcast Simulation app will serve this purpose. It simulates all the capabilities of SMS broadcasting. Using this app a large number of users can be easily reached. GPS feature and message filtering gives an indication of the many features that can be integrated with this promising smart phone application. Cell Broadcast Simulation has these basic features of the broadcasting along with other features. This app will act as a proof of concept that there are lots of practical uses of cell broadcast and upon availability of the broadcast channels this app will become the standard with which one can perform many real life activities all with little overload or traffic. The app is simple yet has many integrated features. The main focus of this app is to make cell broadcasting possible by giving the user a positive experience and expose them to its many potential uses.

For example, one can use this app to solicit the advice of his/her neighbors on a good local handyman. She will simply set the radius to one mile and sends a broadcast message such as "does anyone know a good handyman?" Those who have already installed the app will receive the message and if they are inclined they will respond with the phone number of a handyman that they had a good experience with. The sender of the message could wait for the responses by monitoring his/her phone, or set a new filter "handyman" to

save away any message containing the word handyman and view them at a later time.

As another example, one could solicit the traffic status from other drivers to find out the source of the traffic jam up the road and approximately how long it would take for the traffic to open up. All this happens anonymously without the parties involved knowing each other.

## 1.2 Scope of Project

The Cell Broadcast Simulation app will be developed on Android platform. The app will simulate the functionality of a cell broadcast system. Cell Broadcast Simulation app will investigate different features that can be integrated with the broadcasting message capability and will act as a prototype for the development of the cell broadcast app.

- The Cell Broadcast Simulation app will use default SMS system of the Android platform.
- It will have a radius feature, which will use GPS system to let a user choose how far to send a message.
- It will have its own contact list with cell phone numbers of other users who have installed the app.
- It will have a filtering system, by which a user will be able to automatically filter and save messages she is interested in.

The user will have a real experience of cell broadcasting on Android platform through this simulation app.

### 1.3 Significance of Project

In this age of mobile communication, a large part of the population uses cell phone. Most of the cell phones are smartphones and are capable of performing many functions. Smartphones use radio frequency to communicate. The carrier or provider has many broadcasting channels, but currently they are only used for the national Emergency Alert System. One reason for not making broadcast channels available to private users is that there does not exist a profitable business model for carriers.

Using regular point-to-point channel to reach many people via text messages involves heavy traffic. But using a broadcast channel, it can be very easily done. All cell phones will listen to a particular radio frequency or broadcast channel. When a text message is sent on this channel, all those listening will receive the message at the cost of sending one message. Different level of broadcasting is possible. It can be limited to a particular radius. It can be limited to a cell area. It can be limited to a particular purpose. For example, it is very easy to discard unrelated messages, so that unauthorized users will not be able to see a message. In that way, the broadcasting can be used in a secure way. Social communication is another important application of text message broadcasting. Emergency broadcast is another important application.

Cell Broadcast Simulation app is implemented in a way so that the user can experience the cell broadcasting. When a broadcasting channel is available, only the channel has to be integrated and nothing else needs to be changed in



the app. Then the simulation app will be a fully functional broadcast app. Until then, this simulation app performs its functionality through the SMS system. Therefore, currently the broadcast is simulated using send and receive through SMS. It is important to note that users do not know how the broadcast is performed. The app will maintain the contact, so that only by installing the app, users can communicate with each other. The app has a radius feature which is the broadcasting area. With filtering and message-saving feature, users have total control over how they want to see their message. This app exposes the users to the functionality of the cell broadcasting and illustrates how promising the application would be. This app will be distributed to different service providers so that they can see the real application of the cell broadcasting and may agree to use their broadcast channel so that cell broadcasting can be implemented.

#### 1.4 Glossary

Term	Definition
SMS	Short Message Service (SMS) is a method of that exchange text between mobile phones for communication. The maximum no of character is 160 characters. [1]
Message	An SMS message
Radius	Preferred radius based on user's location to send and receive messages.

Contact List	User's own contacts where cell phone numbers of other users will be stored.
Filter	Monitor and save messages which include a specified keyword.
User	User of a cell phone with Android platform
GPS	Global Positioning System.
Android	"Android™ delivers a complete set of software for mobile devices: an operating system, middleware and key mobile applications." [2]
App	Application or software that can run in a device especially on a mobile platform.
SDK	A Software Development Kit (SDK) is provided for developing application or software in a particular platform. [3]
Android SDK	The software development kit is for developing applications to run on Android devices. "The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications. Applications are written

	using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use which runs on top of a Linux kernel.”[4]
Eclipse	The IDE for android application development provided by the Eclipse open source community. “Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle.”[5]
IDE	“A programming environment integrated into a software application that provides a GUI builder, a text or code editor, a compiler and/or interpreter and a debugger.”[6]
Java	Java is a high level object oriented programming language for developing application developed and maintained by Sun Microsystem.[7]
API	Application Programming Interface (API)

	<p>perform special function or task for an application. API is already provide by SDK or can be used which is developed by other. [8]</p>
Cell Broadcast	<p>“Cell Broadcast (CB) is a mobile technology that allows messages (currently of up to 15 pages of up to 93 characters) to be broadcast to all mobile handsets and similar devices within a designated geographical area. The broadcast range can be varied, from a single cell to the entire network.”[9]</p>
ADT	<p>Android Development Tools (ADT) is a Eclipse IDE plugin to provide a powerful environment to develop Android Application.[10]</p>
Latitude & Longitude	<p>Latitude and Longitude is measuring unit to get any location on earth. “Latitude is measured from the equator, with positive values going north and negative values going south. Longitude is measured from the Prime Meridian (which is the longitude that runs through Greenwich, England), with positive values going east and negative values going west. So, for example, 65 degrees west longitude, 45</p>

	degrees north latitude is -65 degrees longitude, +45 degrees latitude.”[11] [12]
OOP	Object Oriented Programming (OOP) is a programming language model involved around object and data which represent real world scenarios. [13]

## CHAPTER TWO

### OVERALL DESCRIPTION

#### 2.1 Project Design

The app is developed on Android platform using the Android SDK and Eclipse IDE. At first, initial design and analysis was done with various tools such as UML and in particular Use Cases. Then using an object oriented approach it was analyzed for better coding techniques and performance improvement. Testing and deployment were done on real Android devices.

#### 2.2 Project Perspective

The main perspective of this project is to give the user a taste of text message cell broadcasting. The user interface is designed carefully so that the users can easily operate with a positive experience. Practical features of a real broadcasting system are implemented in a natural way. Although this app is a simulation, it feels and look like a real cell broadcast app and the internals of how it operates is hidden from the user.

## 2.3 System Environment

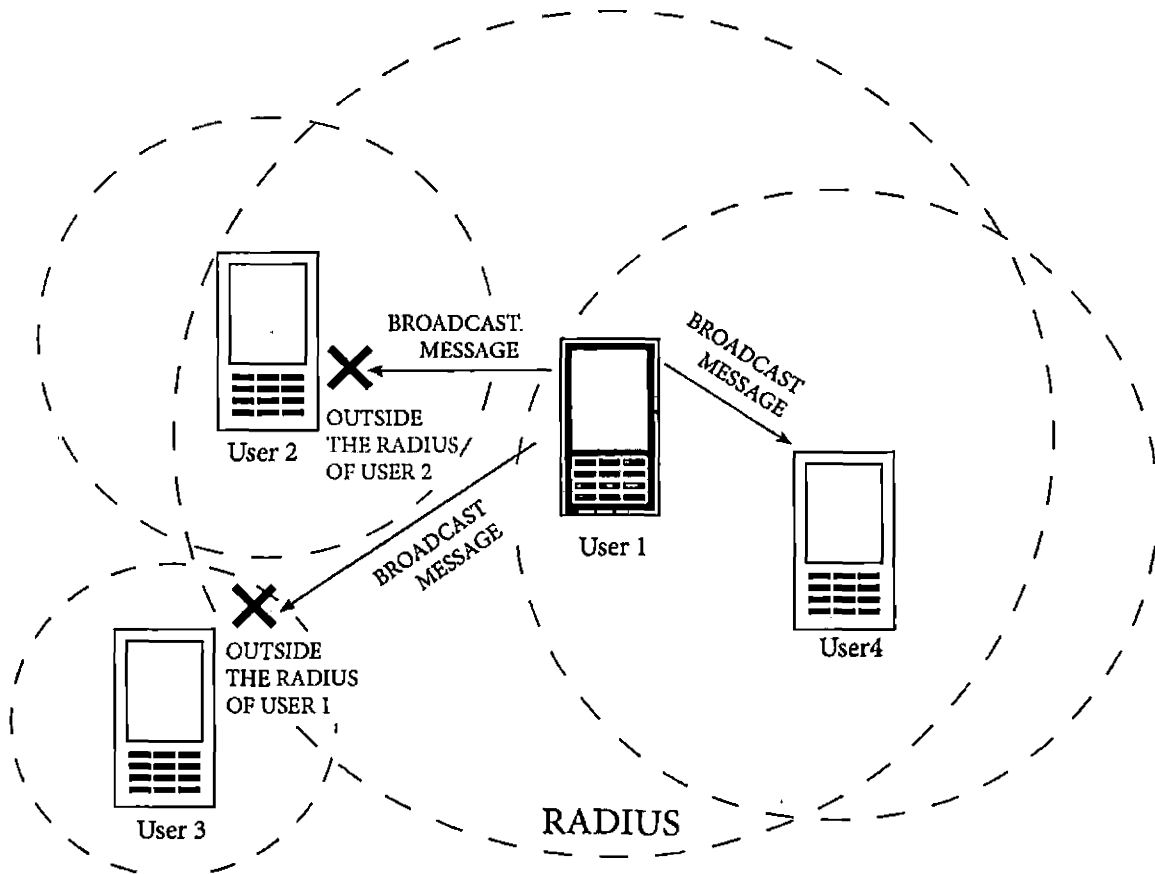


Figure 1. System Environment

Using this app on an Android cell phone, a user will be able to send an SMS to all the users who have also installed the app. She can also select a radius. When she sends an SMS, her current GPS position and her selected radius are added to the message. The receiver app will receive the message and calculates the distance with the sender's GPS location and receiver's GPS location. If the distance falls within both the sender's and the receiver's radius,

then the message will be displayed on the receiver app. As shown in Figure 1, only user 4 receives the message because both users cover each other's radius. If the message has any word that matches the receiver's filter keyword, the message will be also saved and the receiver will be able to recall and see it later.

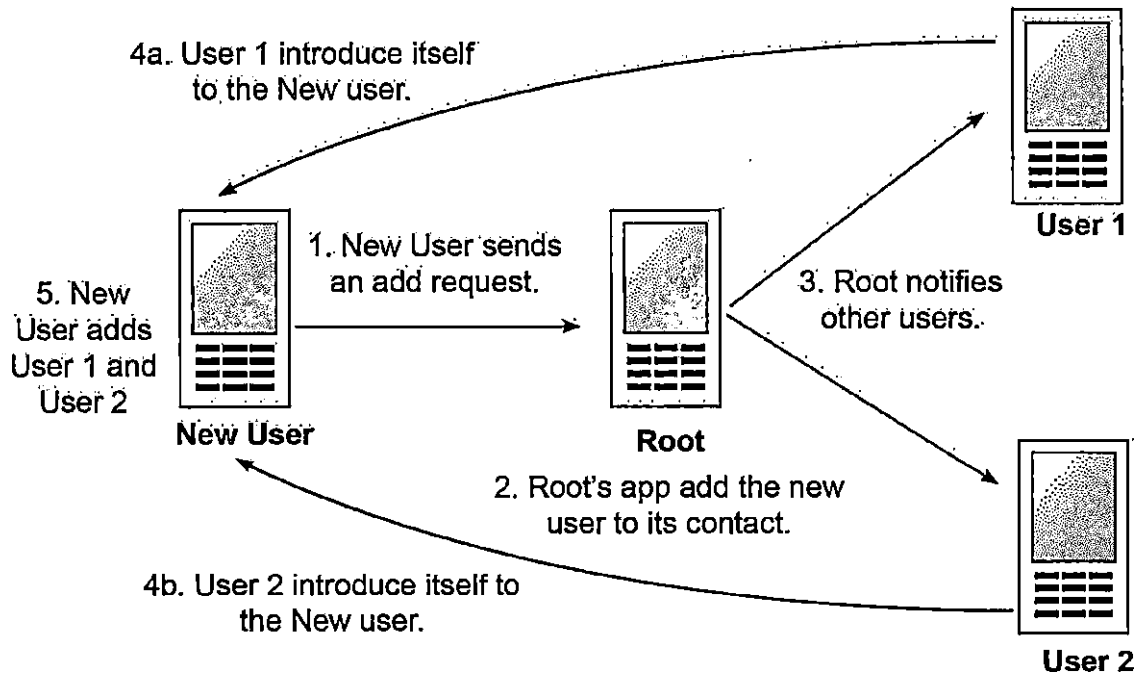


Figure 2. Adding A New User

The process of adding a new user is shown in Figure 2. There will be a root user for this app. Root's mobile phone number will be used to introduce a new user to others. When a new user installs this app, the new user's phone will automatically send a message to the root to add her in the contact list. After



receiving the add request message from the new user, the root updates its contact with the new user's mobile number and sends an add request to all other existing users. After receiving the add request from the root, existing user apps will automatically include the new user in their contact and send an add request with their own mobile number to the new user. The new user will receive those add requests and adds all of them to its contact list. All these processes happen in the background automatically without user involvement.

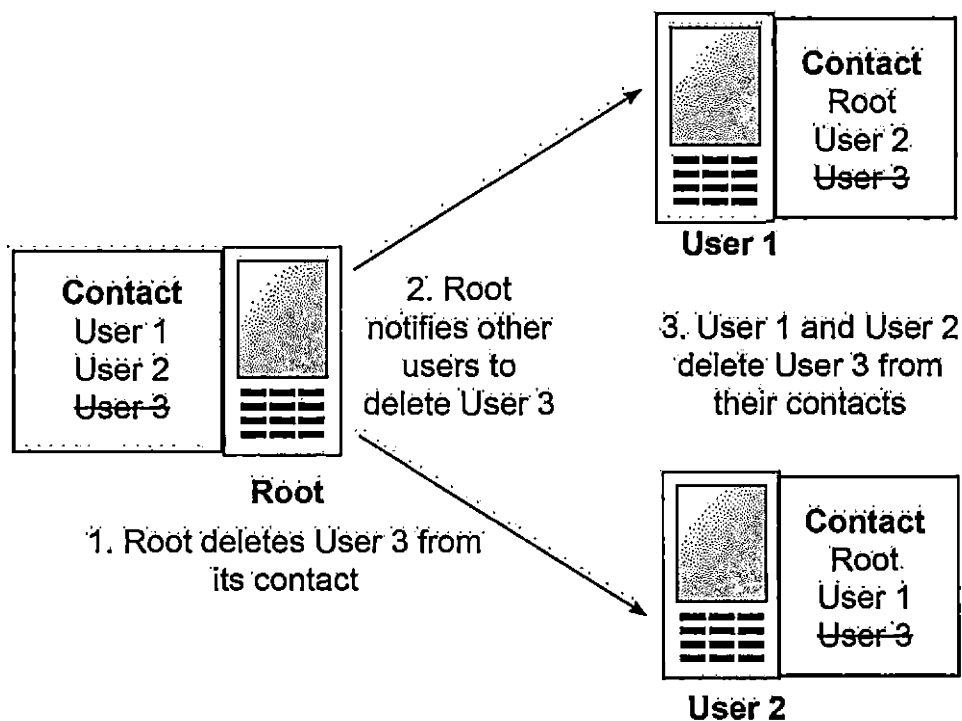


Figure 3. Deleting A Particular User

Also the root user can delete a user from the contact. The deletion process is shown in Figure 3. To delete a user, the root's app will notify all other users about the deletion. Upon receiving the delete request from the root, all the other users will delete that particular user from their contact. The deletion process is also automatic without user involvement.

## 2.4 Software Interface

The Cell Broadcasting Simulation app is designed using the SMS system to simulate broadcasting. It is implemented on Android platform using Java, XML, SQLite, Eclipse IDE with Android Emulator.

## 2.5 Communication Interface

The app is made for any Android smart phone device with Android 2.1 software platform and up. There should be GPS and SMS system available for the proper functioning of the app.

## 2.6 Hardware Interface

The built-in GPS system on an Android smart phone is a requirement for this app. Normally, Android smart phones support SMS system with built-in radio, but the SMS system may need to be activated by the carrier.

## 2.7 Functional Requirements Specification

This section outlines the use cases for the sender and receiver.

### 2.7.1 Sender's Use Case

#### 2.7.1.1 Use Case: Send Message

##### 2.7.1.1.1 Diagram

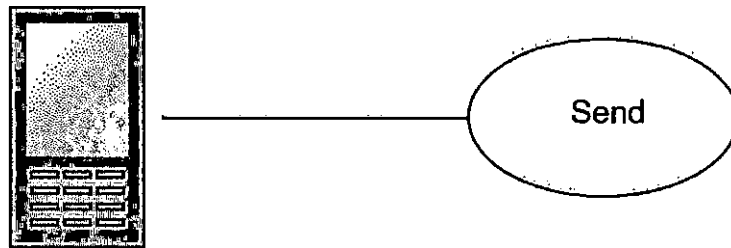


Figure 4. Send Message Use Case

2.7.1.1.2 Brief Description. The sender will broadcast a message to other users – Figure 4.

2.7.1.1.3 Initial Step-By-Step Description. The sender runs the app. The app waits for GPS system to initiate and to get the correct location of the sender.

1. The sender chooses the radius in terms of miles for the first time. The same radius will be used for all messages unless sender inputs a new radius.
2. The sender types the message using the cell phone's keyboard.
3. The sender presses send button to send the message.

4. The app gets all the contacts.
5. The app appends the GPS location and radius of the sender with the message.
6. The app sends the message to all other users.

## 2.7.2 Receiver's Use Case

### 2.7.2.1 Use Case: Receiving Message

#### 2.7.2.1.1 Diagram

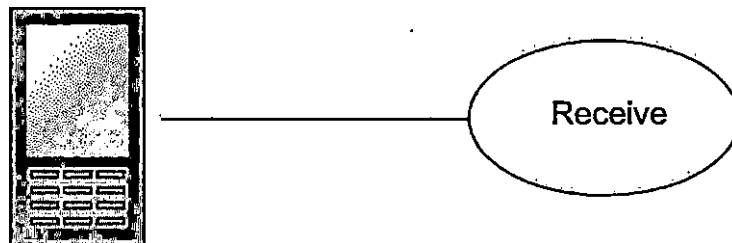


Figure 5. Receiving Message Use Case

2.7.2.1.2 Brief Description. The receiver receives a message broadcast by a user – Figure 5.

2.7.2.1.3 Initial Step-By-Step Description. The app listens for any incoming message. When a message arrives, the app decides whether the message will be displayed based on the sender's and the receiver's selected radius.

1. The app receives the message.
2. Gets the GPS location and radius of the sender from the message.

3. Gets the current GPS location and radius of itself.
4. Calculates the distance between the sender and the receiver.
5. If the distance is covered by both the sender's and the receiver's radius, then the message is displayed.

### 2.7.2.2 Use Case: Filtering Message

#### 2.7.2.2.1 Diagram

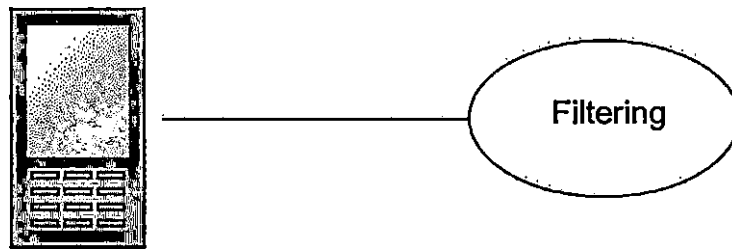


Figure 6. Filtering Message Use Case

2.7.2.2.2 Brief Description. The receiver wants to save particular messages with particular words to find them later if she wasn't reading messages in real time – Figure 6.

2.7.2.2.3 Initial Step-By-Step Description. The app keeps listening for any incoming message. When a message passes the distance check, it will be checked for filter keywords.

1. User inputs a filter keyword.
2. The keyword will be saved by the app.

3. When a message comes and passes the distance check, the app will look for the filter keyword in the message.
4. If there is a match, then the message will be saved.
5. User can recall filtered messages at a future time.

### 2.7.3 Adding A New User

Table 1. Adding A New User

Trigger	The user installs the app.
Precondition	The root user is already selected. A dedicated mobile phone number will act as a root.
Basic Path	<ol style="list-style-type: none"> <li>1. As soon as the app is installed, an SMS is sent to the root user.</li> <li>2. The root checks whether the user already exists or not. The root stores the new user phone number in its database.</li> <li>3. The root sends an SMS request to the existing users to add the new user to their directory.</li> </ol>
Post condition	Other existing users add the new users.
Exception Paths	The user reinstalls the app. The root always checks whether a user exists in the directory

	or not.
--	---------

#### 2.7.4 Deleting An Existing User

Table 2. Deleting An Existing User

Trigger	The user doesn't want to receive messages or the root wants to delete a user.
Precondition	Only root can delete any user.
Basic Path	<ol style="list-style-type: none"> <li>1. From the option menu, the root selects "Other Users".</li> <li>2. The root long presses on the particular user to be deleted.</li> <li>3. From the Delete dialog, the root presses the Delete button.</li> </ol>
Post condition	<ol style="list-style-type: none"> <li>1. The particular user gets deleted from the root's database.</li> <li>2. The root's app sends an SMS to other users automatically with a deletion request.</li> <li>3. Other users' app receives the SMS and deletes the user from the database automatically.</li> </ol>
Exception Paths	None

### 2.7.5 Selecting The GPS Radius

Table 3. Selecting The GPS Radius

Trigger	The user selects "GPS Radius" option menu to set the distance to receive and send a message.
Precondition	Any user can set his/her own preferred radius to send and receive messages.
Basic Path	<ol style="list-style-type: none"><li>1. The user selects the "GPS Radius" option menu.</li><li>2. GPS Radius dialog comes up.</li><li>3. The user inputs the preferred radius.</li><li>4. Presses OK button.</li></ol>
Post condition	No message is sent or received outside the radius. The selected radius will be displayed at the top right corner of the main user interface.
Exception Paths	None.

### 2.7.6 Sending A Message

Table 4. Sending A Message

Trigger	The user types a message and presses the
---------	--



	Send button.
Precondition	The GPS should be on and correct location is displayed at the top right corner of the main user interface.
Basic Path	1. The user types the message. 2. Presses Send button.
Post condition	The message is sent to all the users.
Exception Paths	None.

### 2.7.7 Receiving A Message

Table 5. Receiving A Message

Trigger	Some user broadcasts a message.
Precondition	None.
Basic Path	1. The message reaches to the receiver. 2. The radius is compared with the receiver's radius and sender's radius. 3. If the message covers the radius, then it will be displayed otherwise discarded. 4. The receiver gets notification for the new message.
Post condition	The new message gets displayed with the

	sender's phone number, longitude, latitude, distance and the time.
<b>Exception Paths</b>	If user is not currently using the app, then user can select the notification from the notification bar to open and read the message.

### 2.7.8 Adding Filter

Table 6. Adding Filter

<b>Trigger</b>	The user selects the "Filter" option menu or presses the Filter bar above the Send button.
<b>Precondition</b>	None.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The user selects the Filter option menu.</li> <li>2. Filter screen appears.</li> <li>3. The user types the new filter keyword.</li> <li>4. Presses the add button.</li> </ol>
<b>Alternate Path</b>	<ol style="list-style-type: none"> <li>1. The user presses the Filter bar.</li> <li>2. Filter screen appears.</li> <li>3. The user types the new filter keyword.</li> <li>4. Presses the add button.</li> </ol>
<b>Post condition</b>	New filter is added to the database and

	whenever a message arrives the app searches for the keyword. If a message has the keyword, the message background changes to light blue.
Exception Paths	None.

### 2.7.9 Accessing Filtered Messages

Table 7. Accessing Filtered Message

Trigger	The user selects the "Filter" option menu or presses the Filter bar above the Send button to see the filtered messages.
Precondition	Filtered Keyword is already added.
Basic Path	<ol style="list-style-type: none"> <li>1. The user selects the "Filter" option menu.</li> <li>2. Filter Screen appears.</li> <li>3. User presses the filtered keyword.</li> </ol>
Alternate Path	<ol style="list-style-type: none"> <li>1. The user presses the Filter bar.</li> <li>2. Filter screen appears.</li> <li>3. Presses the filtered keyword.</li> </ol>
Post condition	If any message has arrived with that keyword, it is displayed.
Exception Paths	None.

## 2.7.10 Deleting Filter Keyword

Table 8. Deleting Filter Keyword

Trigger	The user selects the Filter option menu or presses the Filter bar above the Send button to delete the filtered keyword.
Precondition	Filtered keyword is already added.
Basic Path	<ol style="list-style-type: none"><li>1. The user selects the "Filter" option menu</li><li>2. Filter Screen appears.</li><li>3. User presses the filtered keyword.</li><li>4. Filtered messages with that keyword appears.</li><li>5. By long pressing a particular message to be deleted.</li><li>6. Delete dialog screen appears.</li><li>7. Selects OK button.</li></ol>
Alternate Path	<ol style="list-style-type: none"><li>1. The user presses the Filter bar.</li><li>2. Filter screen appears.</li><li>3. User presses the filtered keyword.</li><li>4. Filtered messages with that keyword appear.</li><li>5. Long presses a particular message to be</li></ol>

	<p>deleted.</p> <p>6. Delete dialog appears.</p> <p>7. Selects OK button.</p>
Post condition	The filtered keyword and all the messages that are saved under that keyword are deleted.
Exception Paths	None.

### 2.7.11 Deleting A Filtered Message

Table 9. Deleting A Filtered Message

Trigger	The user selects the "Filter" option menu or presses the Filter bar above the Send button to delete the filtered message.
Precondition	Filtered Keyword is already added.
Basic Path	<ol style="list-style-type: none"> <li>1. The user selects the "Filter" option menu</li> <li>2. Filter screen appears.</li> <li>3. User presses the delete button on the right side of the filtered keyword.</li> </ol>
Alternate Path	<ol style="list-style-type: none"> <li>1. The user presses the Filter bar.</li> <li>2. Filter screen appears.</li> <li>3. Presses the delete button on the right side</li> </ol>

	of the filtered keyword.
Post condition	Only the particular messages that are saved under that keyword are deleted.
Exception Paths	None.

### 2.7.12 Notification Turn On Or Off

Table 10. Notification Turn On Or Off

Trigger	The user selects the "Notification On/Off" option menu.
Precondition	None.
Basic Path	The user selects the Notification On/Off option menu.
Post condition	If the notification was on, it is turned off or vice versa.
Exception Paths	None.

### 2.7.13 Deleting A Message

Table 11. Deleting A Message

Trigger	The user wants to delete a message.
Precondition	None.

Basic Path	<ol style="list-style-type: none"> <li>1. The user long presses on the message.</li> <li>2. The Delete dialog appears.</li> <li>3. Selects Delete button.</li> </ol>
Post condition	The particular message is deleted.
Exception Paths	None.

### 2.7.14 Deleting All Messages

Table 12. Deleting All Messages

Trigger	The user selects the "Delete All" option menu to delete all the messages.
Precondition	None.
Basic Path	The user selects the Delete All option menu.
Post condition	All messages in the database are deleted.
Exception Paths	None.

### 2.7.15 Exiting The Application

Table 13. Exiting The Application

Trigger	The user selects the "Exit" option menu to stop and exit the app.
Precondition	None.

Basic Path	The user selects the Exit option menu.
Post condition	The app shutdowns.
Exception Paths	None.

### 2.7.16 Accessing Other Users List

Table 14. Accessing Other Users List

Trigger	The user selects the "Other Users" option menu to see other users list.
Precondition	None.
Basic Path	The user selects the "Other Users" option menu.
Post condition	All the users are listed.
Exception Paths	None.

## 2.8 Detailed Non-Functional Requirements

### 2.8.1 Logical Structure Of The Data

Several databases and files are written to make the application to operate properly. The last known GPS location is saved in a file to help to get a message correctly when the user is not using the app. GPS radius is also saved in a file. There are four databases to save contacts, messages, filter keyword and filtered messages.



### 2.8.1.1 Contact Database

Table 15. Contact Database

<b>Data Item</b>	<b>Type</b>	<b>Description</b>	<b>Comment</b>
Contact	Varchar	Stores all user's mobile phone numbers	Only the root can delete any contact.

### 2.8.1.2 Messages Database

Table 16. Message Database

<b>Data Item</b>	<b>Type</b>	<b>Description</b>	<b>Comment</b>
id	Integer	Increments automatically	
Message	Varchar	Saves a message	
Phone	Varchar	Saves the sender's phone Number	
Time	Varchar	message sent time	
Filter Status	Varchar	Message has any filter	

		keyword or not	
Distance	Varchar	The distance in miles between the sender and receiver	Calculated from sender's longitude and latitude.
Latitude	Varchar	Sender's Latitude	
Longitude	Varchar	Sender's Longitude	

2.8.1.3 Temporary Message Saver Database.

Table 17. Temporary Message Saver Database

Data Item	Type	Description	Comment
Message	Varchar	temporarily saves a message.	When the user opens the app, all messages are fetched and discarded.

#### 2.8.1.4 Filter Keyword Database

Table 18. Filter Keyword Database

<b>Data Item</b>	<b>Type</b>	<b>Description</b>	<b>Comment</b>
Filter	Varcar	Save the filter keyword	

#### 2.8.1.5 Filtered Messages Database

Table 19. Filtered Messages Database

<b>Data Item</b>	<b>Type</b>	<b>Description</b>	<b>Comment</b>
id	Integer	Increments automatically	
Message	Varchar	Saves a message	
Phone	Varchar	Saves the sender's phone Number	
Filter Key	Varchar	Saves the filter keyword	

## 2.9 User Characteristics

The user needs to have basic knowledge about how to type and send an SMS and how to operate cell phones with Android platform. Also, the knowledge of how GPS works can help a user to use this app more efficiently.

## 2.10 Non-Functional Requirements

Sending SMS may incur extra cost depending on the user's cell phone plan and service provider. Activating and running GPS can drain the battery faster than normal operation. User has to be aware of these conditions to use this application efficiently.

## CHAPTER THREE

### DESIGN

The basic focus of the design was to implement simple user interface enriched with broadcasting feature. The Android platform provides a better programming architecture which separates user layouts, UI elements, icon and other resources from programming. Since Java programming language is used to write the application, the design process is object oriented which makes the development and maintenance more efficient.

#### 3.1 User Interface Design

There are several different screens with option menu and dialogs. They are created with XML files with great care to make it simple for the users. All the icons are custom made. The navigation between different screens is also easy and similar. Users can easily understand what they have to do to make this app function properly.

#### 3.2 Screen Descriptions

With the initial design, four basic screens are designed. The main focus is to keep the operation simple and similar, so that the user doesn't get frustrated.

### 3.2.1 Main Screen

As soon as a user opens the app, this screen appears. The screen is designed to display all the necessary information that is required to operate the app properly. There is a place to input the message. There is a button to send the message. The main screen also displays radius in terms of mile and the current location in terms of latitude and longitude. It will also display a few filtered keywords that the user has already entered.

Table 20. Main Screen

<b>Button</b>	<b>Function</b>
Send	This will send the message to all other users.
Enlarge	It will open a bigger input box where a user can see large part of the message. It will be much easier to see and edit a message here.
Filter Bar	it will display few recent filter keywords. Pressing on it, the filter screen will open and all the filters will be displayed.

### 3.2.2 Other Users Screen

All the users who have installed the app are displayed in this screen. The root user has a special permission to delete any users from here. Any regular user can see the other users, but he/she will not be able to perform any operation.

Table 21. Other Users Screen

Button	Function (For root user only)
Delete	A particular user will be deleted and all the other users will be notified to delete that particular users.

### 3.2.3 Filter Screen

This screen displays all the filter keywords. In this screen, users are able to add or delete their keywords. Users input the keyword in the input field and then press the add button. After that, the keyword will be added to the database and will be displayed in the list.

Table 22. Filter Screen

Button	Function
Add	Adds a new filter keyword under which any message with that keyword will be saved.
Deleted	Deletes a particular keyword and all messages saved under it.

### 3.2.4 Saved Filtered Message Screen

If a message is received with a filtered keyword, the message will be saved under that keyword. By pressing the keyword in the Filter Screen, the user can see the messages that are saved under that keyword. The user can delete any saved message under any particular filtered keyword by long pressing it.

Table 23. Saved Filter Screen

Button	Function
Delete	Delete the particular saved message.

### 3.2.5 Class Diagrams

Using object oriented programming approach, several classes are designed for proper operation of the app – Figure12. Each class is named for the most important function it performs. They have distinct relationship with each other and they interact with each other through different functions or methods. The most important classes are: CellBroadcast – Figure 7, Receiver – Figure 8, UsersCBapp – Figure 9, Filter – Figure 10, SavedFilter – Figure 11. The detailed description of these classes and their relationships follows.



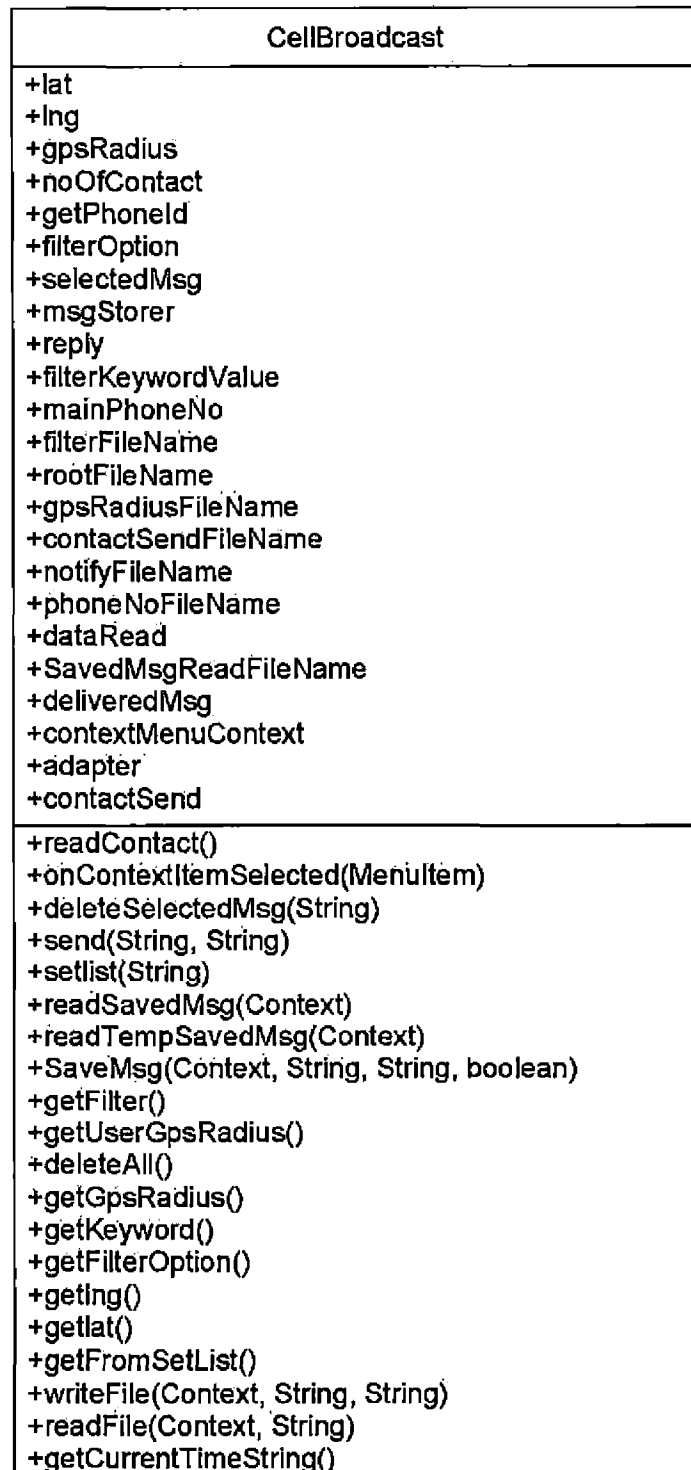


Figure 7. CellBroadcast Class

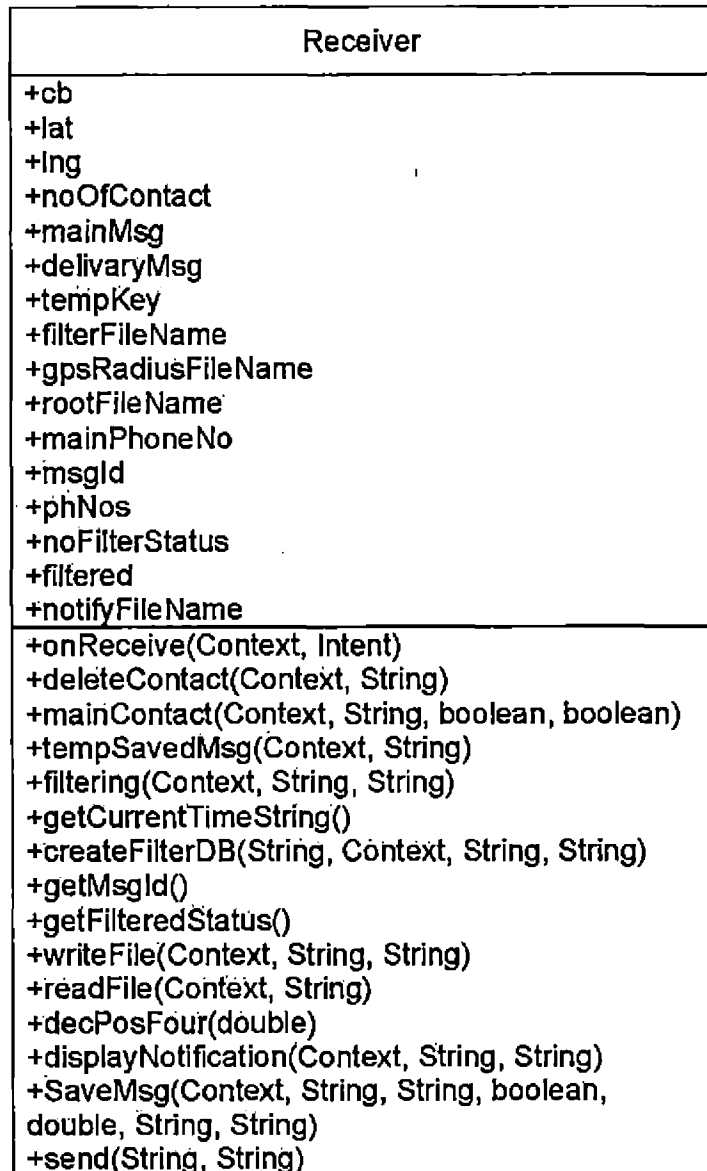


Figure 8. Receiver Class

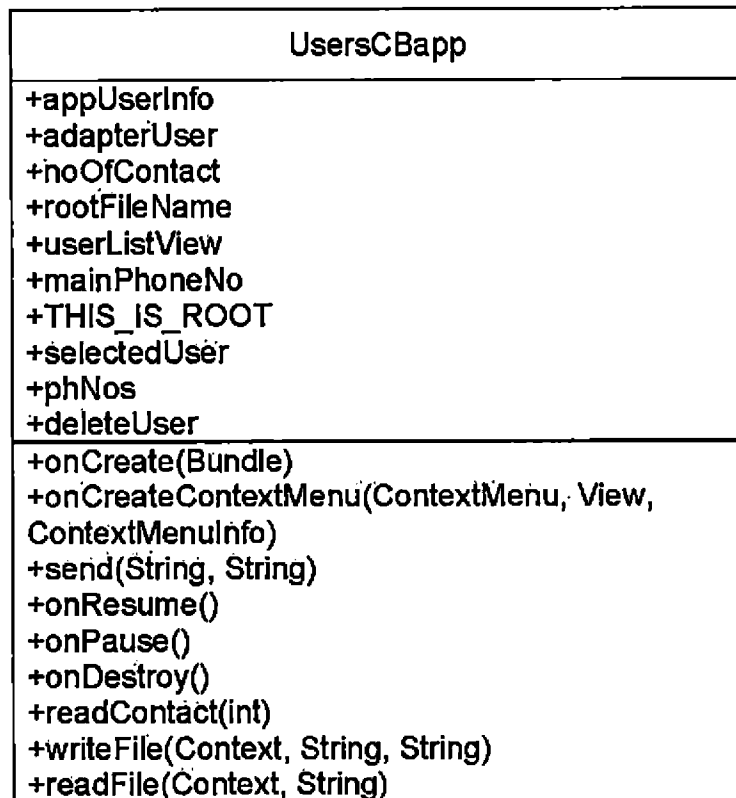


Figure 9. UsersCBapp Class

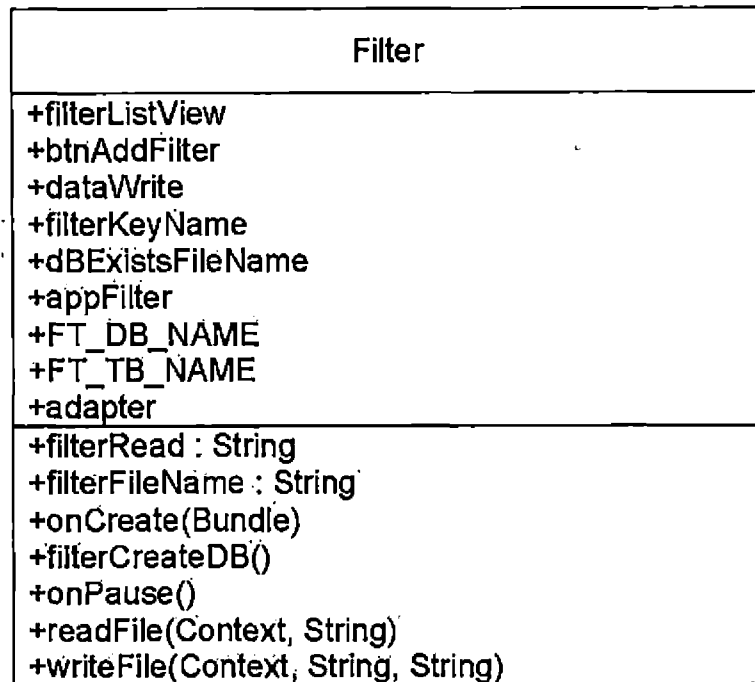


Figure 10. Filter Class

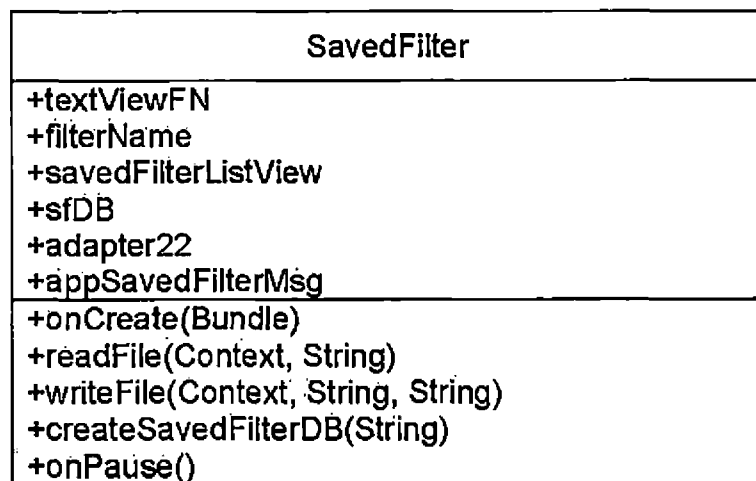


Figure 11. SavedFilter Class

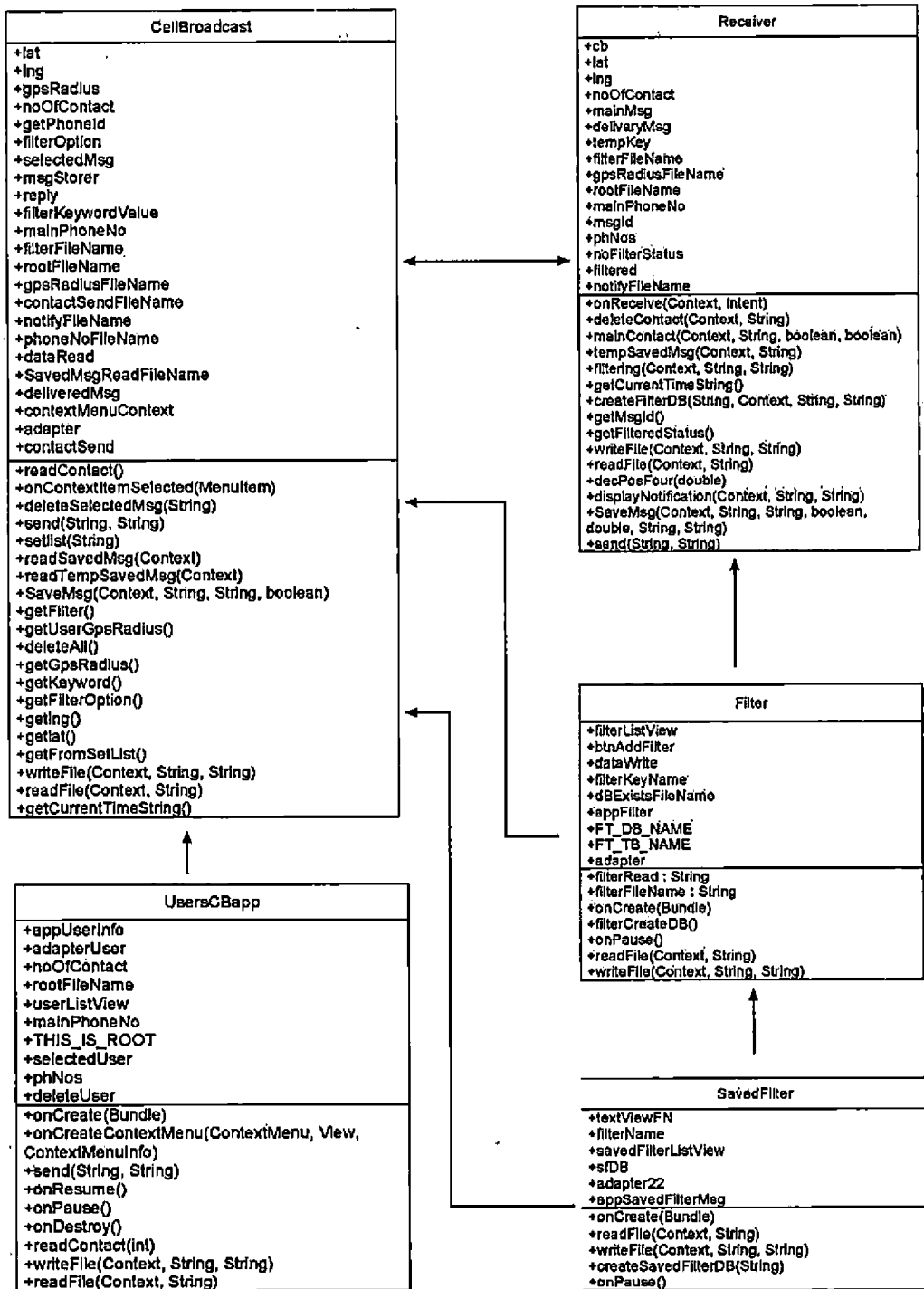


Figure 12. Class Diagram

## CHAPTER FOUR

### IMPLEMENTATION

Agile Software development method was applied to implement this app with Android SDK in Eclipse IDE. Main key classes were implemented first with basic feature and then tested to verify the result. Additional features were added and tested sequentially.

#### 4.1 Implementation Outlines

First the CellBroadcast class was created. This is the main class from which all activity will start. All the user layouts were done in XML. Each layout has individual names which are integrated with different screens to display the views. The CellBroadcast class integrates the main view and main option menu. This is the view the user sees after he/she opens the app. Receiver class is the important class which handles the SMS system. This class will work with CellBroadcast class to get the SMS message and display it in the main window. The Filter class handles filter screen and it creates and displays filter keyword. The SavedFilter handles the filter message screen which displays the saved messages under a particular keyword. The UserCBapp class handles the Other User screen which shows other users who have installed the app.

## 4.2 Interaction

The Receiver runs in the background and waits for an SMS. When an SMS arrives, the Receiver parses it and checks for filter keywords saved using Filter class. It checks for radius, latitude and longitude saved in the CellBroadcast. If the distance falls within the radius of both the sender and receiver, the Receiver passes the message to the CellBroadcast. CellBroadcast displays the message in the main screen. If the SMS falls outside of the radius of sender and receiver then the SMS is discarded. If a message has one or more keyword in it, the Receiver saves the message under those particular keyword's database. When a new user installs the app, the CellBroadcast sends a special message with add request to the root user's app. When the Receiver sees this special message, it saves and notifies other users about the new users. Also, if the root user deletes a user from the Other Users screen which is handled by the UserCBapp, the app sends a delete request to all the other users to delete that particular users. The Receiver handles this delete operation. The GPS system and radius is handled by the CellBroadcast and the Receiver always access these values to check the incoming SMS.

## 4.3 Functionality

The final implemented app works in an integrated environment to provide a simple user experience. It provides a positive experience of Cell Broadcasting to the users. Users just have to type the message and press the send button.

Also, the users can optionally set their own preferred filter keywords and radius.

The default radius is one mile.

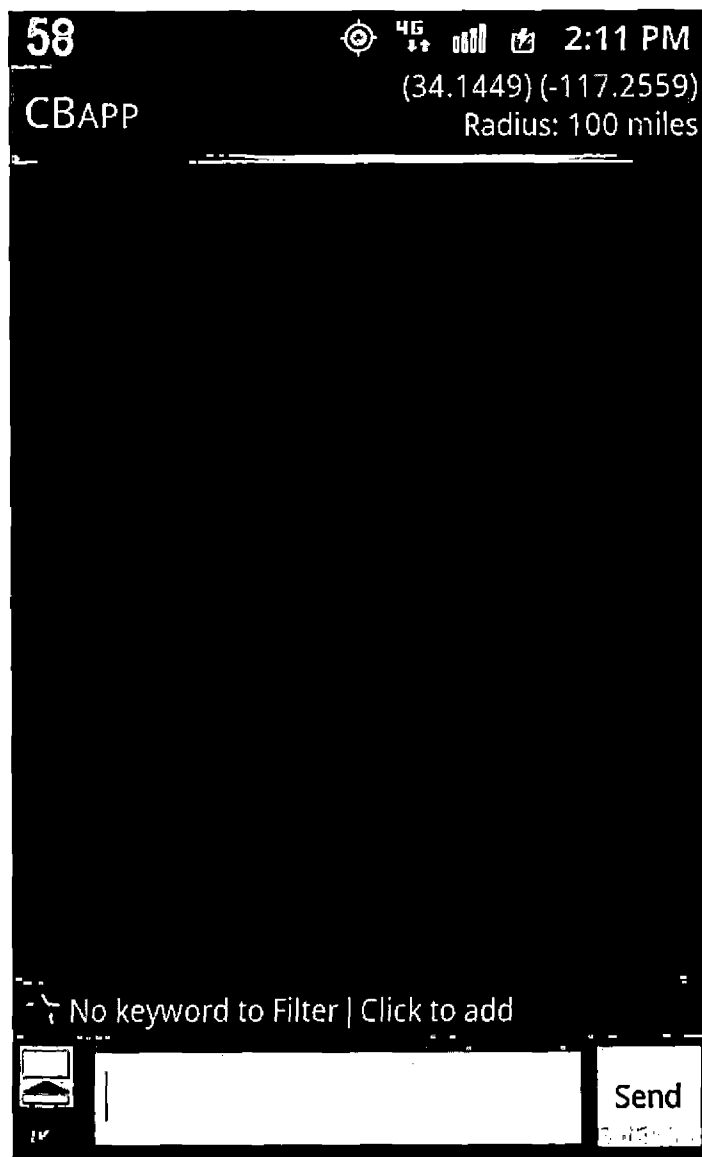


Figure 13. Main Screen After Opening The App



Figure 13 illustrates the main screen. At top right corner, latitude, longitude and radius are displayed. At the bottom, input field, send button, enlarge input field button and filter bar appear. The user will not be able to send a message before the GPS is working and latitude and longitude is displayed in the top right corner field.

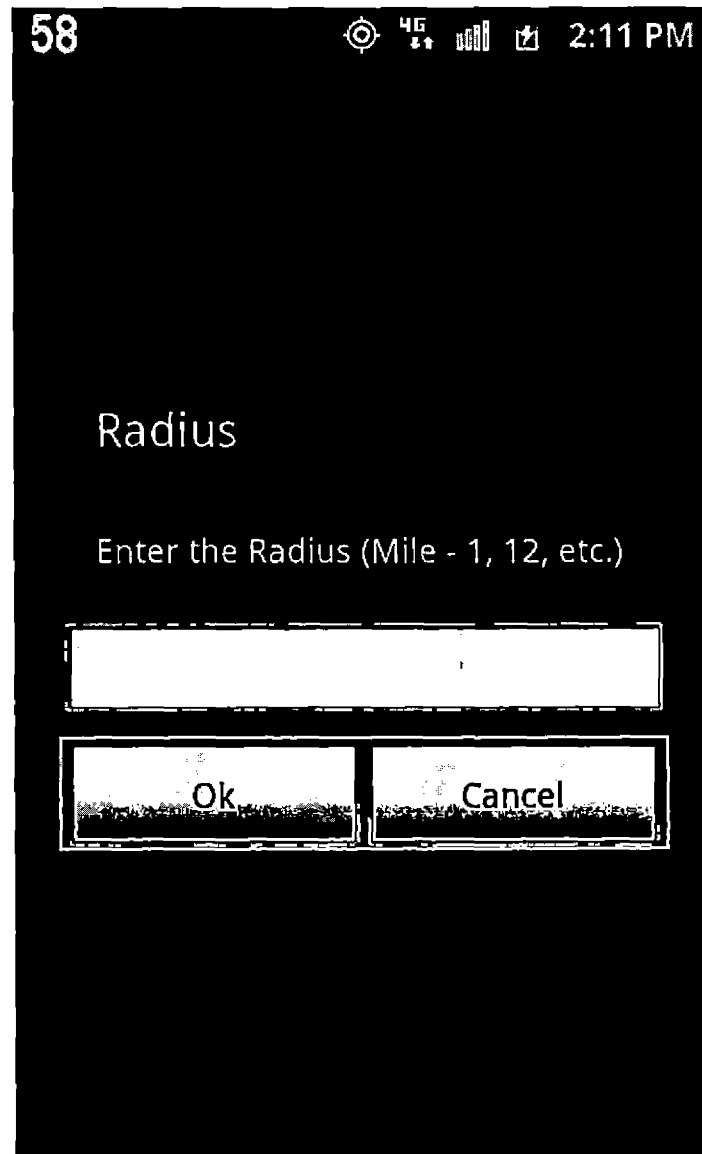


Figure 14. Radius Dialog To Set The Radius In Miles

After installing the app, the user has to set the radius from the dialog shown in Figure 14. If the user doesn't set the radius after installing the app, then she will not receive any message beyond one mile.

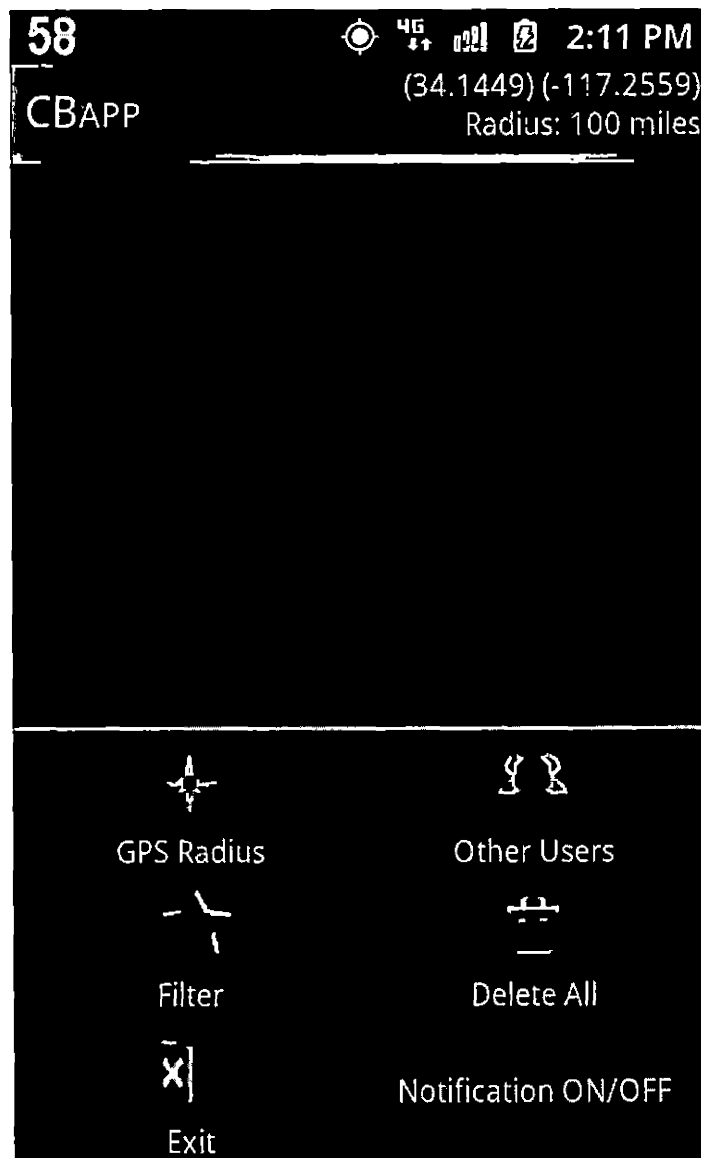


Figure 15. Option Menu

By clicking the menu button of the Android device, the user can get the option menu shown in Figure 15. The menu has six options to select. The "GPS Radius" option is for setting the GPS radius. The "Other Users" option opens up a new screen and displays the other users list. "Filter" option is for setting filter keyword and also for accessing saved filtered messages. "Delete All" option deletes all the messages in the main screen. "Exit" option shutdowns the app and the app will no longer use any resource of the device such as GPS service. When a message arrives, the app will notify the user through the notification service. It will make a sound and display the message in the notification bar. This feature can be turned off through the Notification ON/OFF option. If the notification service of the app is on, this option turns off the notification. If it is off, it will turn on notification.

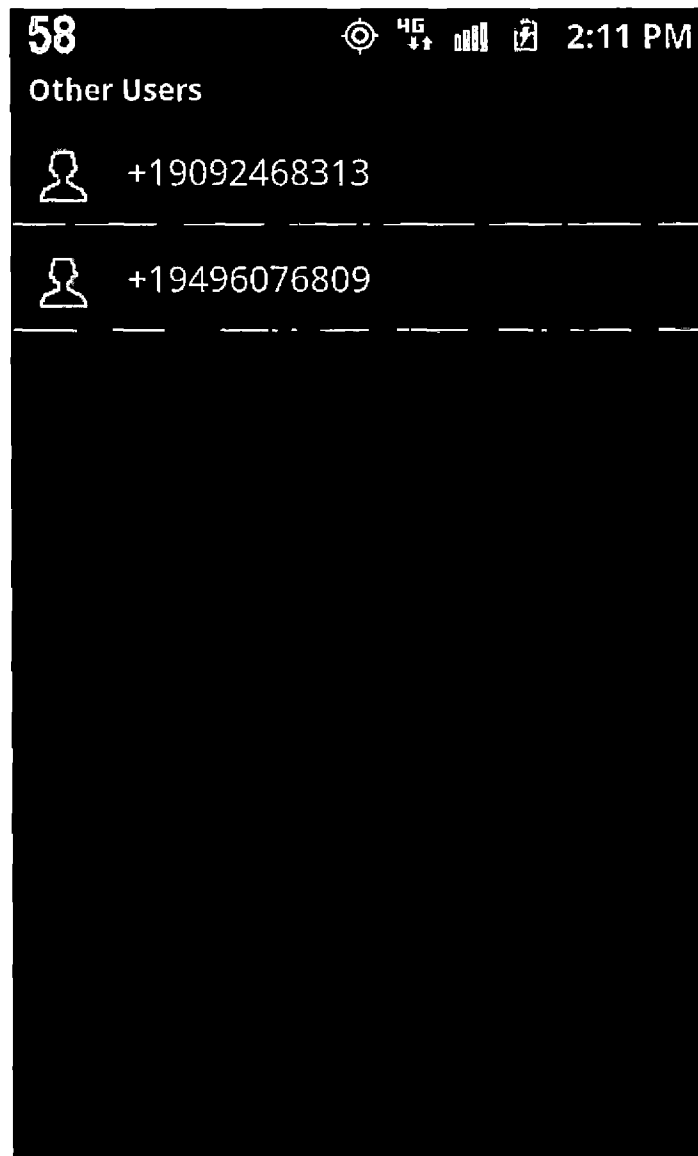


Figure 16. Other Users Screen

Figure 16 displays the “Other Users” screen. Mobile phone numbers of all the users, who have installed the app are displayed on this screen. The user can go to option menu by pressing the menu key of the device and press the “Other Users” option to see this screen. The user’s number will not be displayed on the screen.

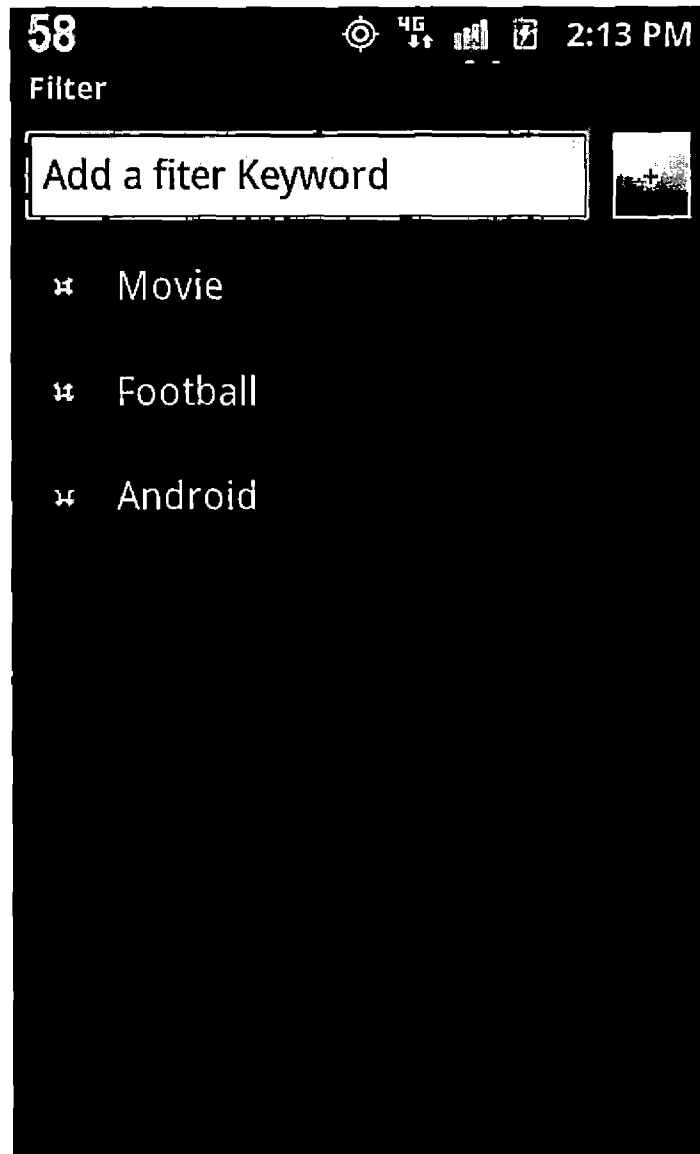


Figure 17. Filter Screen

Figure 17 displays the "Filter" screen. When the user presses the "Filter" option from the option menu, this screen is displayed. One can put the filter keyword in the input field and press the add button to add the filter keyword. Messages with specified keywords will be saved under that keyword. One can see those messages by clicking on the displayed keywords. One can also delete

any keyword from this screen by pressing the delete button at the left of the keyword. All the messages saved under the deleted keyword will be also deleted.

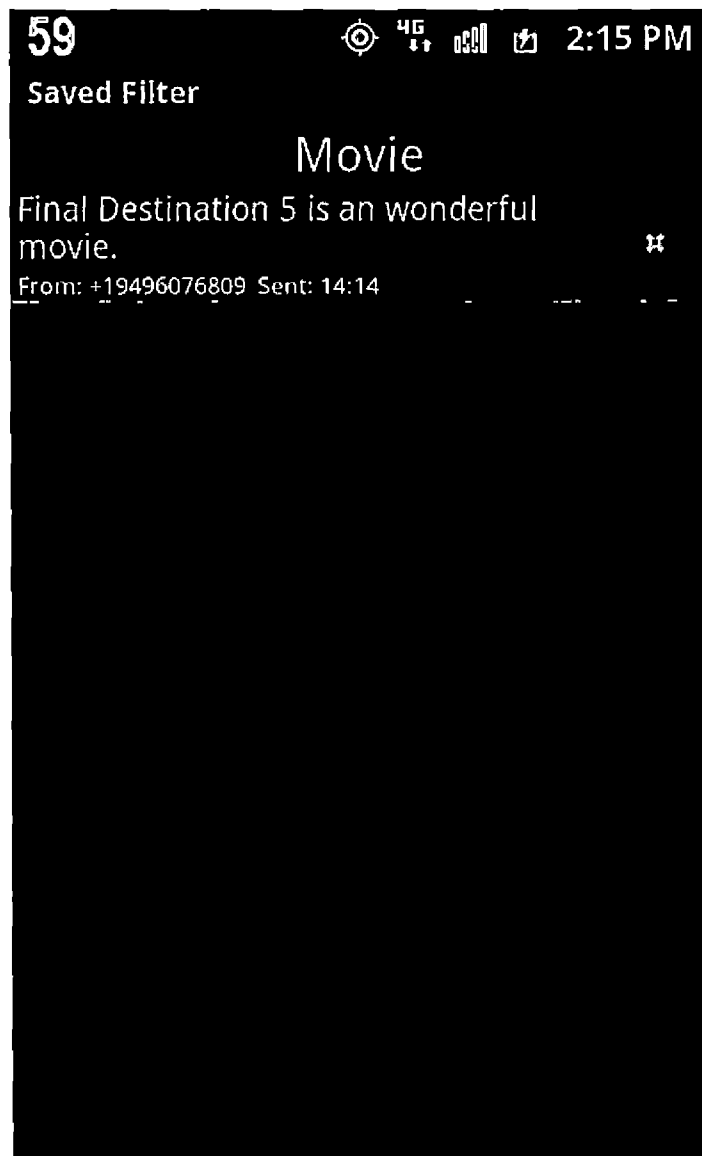


Figure 18. Saved Filter Message

Under each filter keyword, the filtered message with that keyword will be saved. The user can access these saved messages anytime from the filter option in option menu or from the filter bar – Figure 18.

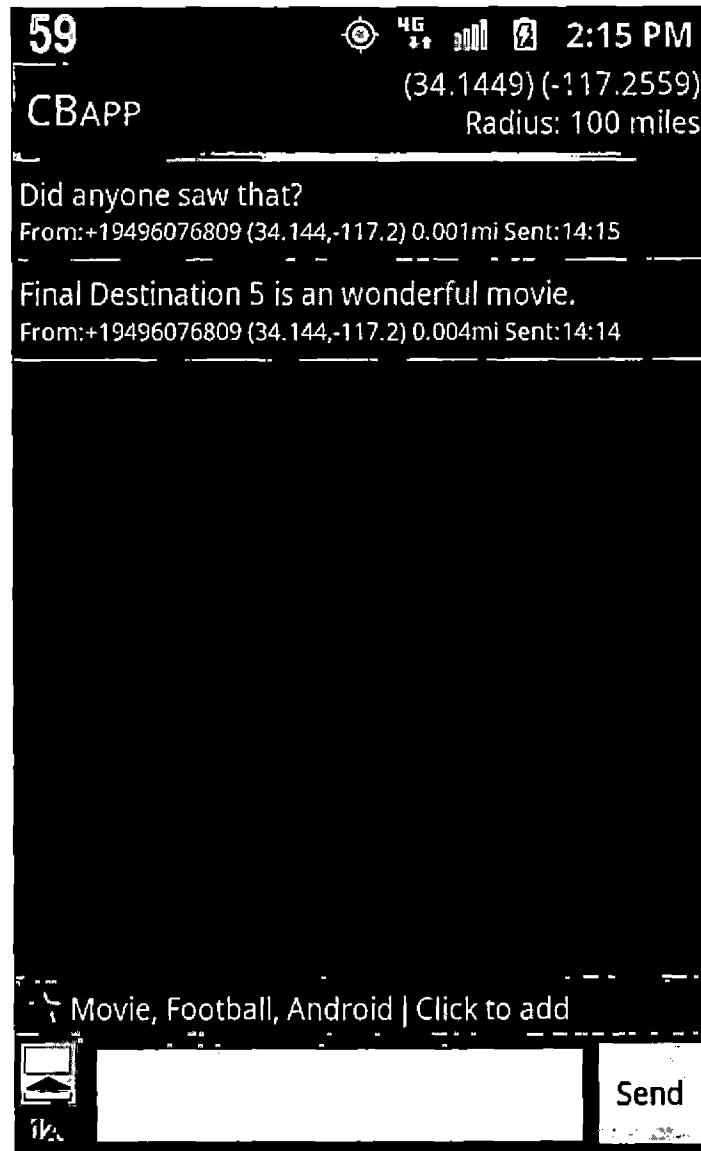


Figure 19. Normal Message And Filtered Message

A normal message doesn't have a background color. It will be displayed with the sender phone number, latitude, longitude, distance from receiver and the

time. But a filtered message has a light blue background color. The user will instantly understand that a message has a filter keyword that she has set –

Figure 19.

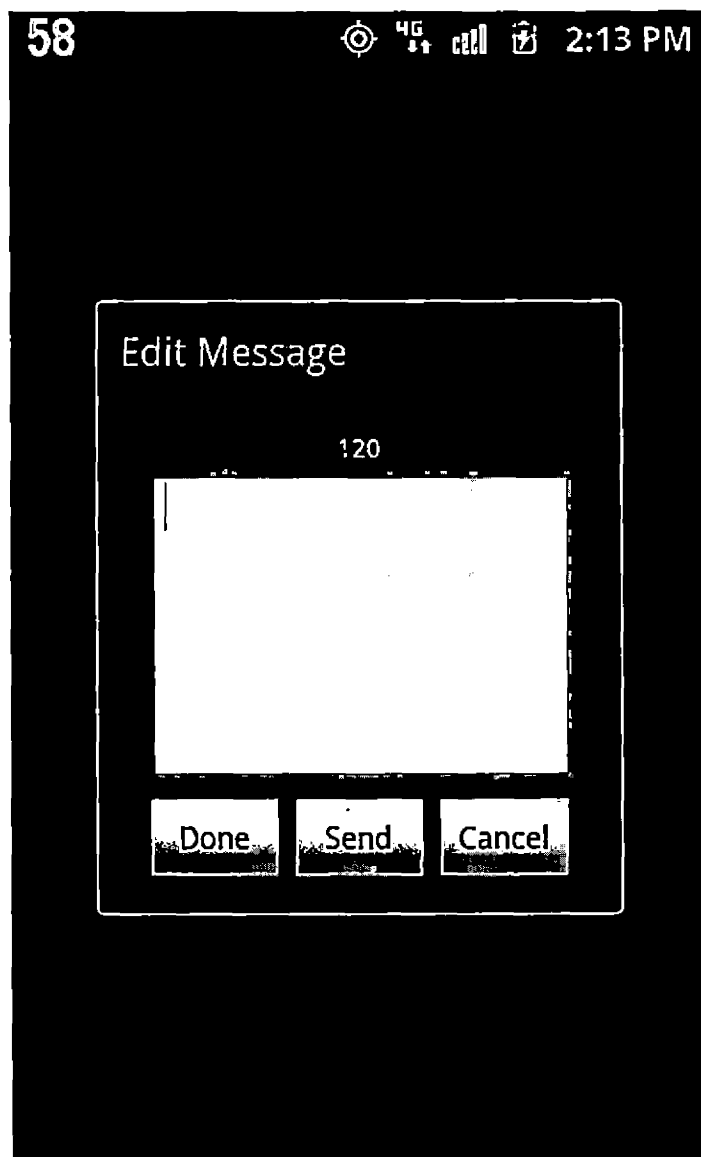


Figure 20. Enlarged Edit Message Dialog



By pressing the Enlarged Edit Message button beside the message input field in the main screen, the user can open Edit Message screen, Figure 20. This will give the user more space to see the message. By pressing the "Done" button, the user will go back to the main screen with the edited message. By pressing the "Send" button, the message will be directly sent and the user will go back to the main screen with an empty input field.

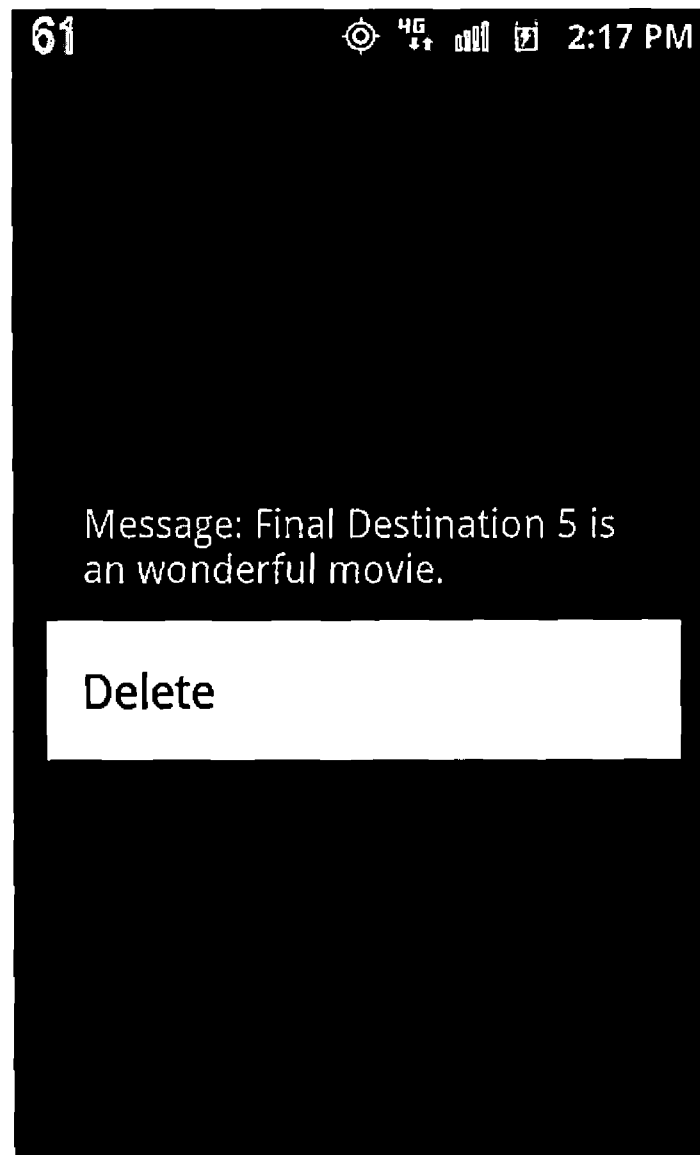


Figure 21. Deleting A Message On The Main Screen

By long pressing on an individual message, the user can delete the message from the main window – Figure 21. A context menu will be displayed with Delete option and the message will be deleted if the option is selected.



Figure 22. Deleting A User By The Root User

Only the root has the ability to delete a user. By long pressing on the mobile phone number of the user, the root will get the Delete dialog box, Figure 22. If the delete button is pressed, the user will be deleted from the root's contact list and the root's app will notify all the other users to delete the user. After

receiving the delete request from the root, all the other users' app will delete the contact.

## CHAPTER FIVE

### TESTING

Since Agile Software Development methodology has been used in the development of the app, testing was an integral part of the development process. Based on Agile method, every time a new feature was implemented, it was immediately unit tested followed by regression test. The app has gone through numerous iterations of testing. The approach has greatly helped with integrating new feature into the app. Consequently and at the same time, components of the app are strongly encapsulated. This in turn has contributed to the easy maintenance of the app.

Each new feature has been tested with the Android emulator in Eclipse followed by testing on multiple physical devices or cell phones. At least five cell phones have been used simultaneously to test the app in a number of settings. Testing has also included multiple carriers such as Verizon and T-Mobile.

## CHAPTER SIX

### MAINTENANCE

The app has a main root user, who receives all the add and delete requests and notifies other users to add or delete a user. The root user has the authority to delete any user. Without the root user the app will not work. So maintaining the root user is very important. The root user should not be disconnected. The root should be functioning properly all the time. It is not hard to maintain the root user. The app has the built-in features to handle all the requests. The only requirements are: a connection from the carrier must exist and there must be enough battery power.

## CHAPTER SEVEN

### CONCLUSION AND ENHANCEMENT

#### 7.1 Conclusion

The app is developed to give the user the best experience to understand the usefulness of the cell broadcasting. Although the real broadcasting channel doesn't exist, the user will not be able to notice the difference easily. This app has lot of potentialities. It serves as a prototype of a real broadcasting app. Users doesn't have to maintain a contact list. One can easily reach many users with this app.

#### 7.2 Enhancements

The main enhancement of this app is to integrate a real broadcast channel from the provider. A broadcast channel can replace the current SMS system. This will reduce traffic and overload. In future, the same app will be developed for iPhone.

## REFERENCES

- [1] HowStuffWorks Inc, "How SMS Works", [howstuffworks.com](http://howstuffworks.com), [Online]. Available: <http://communication.howstuffworks.com/sms.htm> [Accessed: 12 Nov. 2011].
- [2] Open Handset Alliance, "Android", [openhandsalliance.com](http://openhandsalliance.com), [Online]. Available: [http://www.openhandsalliance.com/android\\_overview.html](http://www.openhandsalliance.com/android_overview.html) [Accessed: 14 Nov, 2011].
- [3] Wikimedia Foundation Inc, "Software Development Kit", [wikipedia.org](http://wikipedia.org), [Online]. Available: [http://en.wikipedia.org/wiki/Software\\_development\\_kitn](http://en.wikipedia.org/wiki/Software_development_kitn) [Accessed: 4 Dec. 2011].
- [4] QuinStreet Inc, "Android SDK", [webopedia.com](http://webopedia.com), [Online]. Available: [http://www.webopedia.com/TERM/A/Android\\_SDK.html](http://www.webopedia.com/TERM/A/Android_SDK.html) [Accessed: 17 Dec. 2011].
- [5] The Eclipse Foundation, "About the Eclipse Foundation", [eclipse.org](http://eclipse.org), [Online]. Available: <http://www.eclipse.org/org/> [Accessed: 17 Dec. 2011].
- [6] QuinStreet Inc, "Integrated Development Environment" [webopedia.com](http://webopedia.com), [Online]. Available: [http://www.webopedia.com/TERM/I/integrated\\_development\\_environment.html](http://www.webopedia.com/TERM/I/integrated_development_environment.html) [Accessed: 23 Dec. 2011].



- [7] About.com, "What is Java?", about.com, [Online]. Available: <http://java.about.com/od/gettingstarted/a/whatisjava.htm> [Accessed: 27 Dec. 2011].
- [8] Computerworld Inc, "QuickStudy: Application Programming Interface (API)", computerworld.com, [Online]. Available: [http://www.computerworld.com/s/article/43487/Application\\_Programming\\_Interface](http://www.computerworld.com/s/article/43487/Application_Programming_Interface) [Accessed: 28 Dec. 2011].
- [9] Cell Broadcast Forum, "What is Cell Broadcast", cellbroadcastforum.org, [Online]. Available: <http://www.cellbroadcastforum.org/whatisCB/index.html> [Accessed: 2 Jan. 2012].
- [10] Google, "ADT Plugin for Eclipse", android.com, [Online]. Available: <http://developer.android.com/sdk/eclipse-adt.html> [Accessed: 2 Jan. 2012].
- [11] NASA, "What is the definition of latitude and longitude?", nasa.gov, [Online]. Available: <http://jwocky.gsfc.nasa.gov/teacher/latlonarchive.html> [Accessed: 2 Jan. 2012].
- [12] NASA, "Latitude and Longitude", nasa.gov, [Online]. Available: <http://www-istp.gsfc.nasa.gov/stargaze/Statlong.htm> [Accessed: 2 Jan, 2012].
- [13] TechTarget, "object-oriented programming (OOP)", techTarget.com, [Online]. Available: <http://searchsoa.techtarget.com/definition/object-oriented-programming> [Accessed: 3 Jan, 2012]

- [14] Hillebrand. F et al., "SHORT MESSAGE SERVICE (SMS): THE CREATION OF PERSONAL GLOBAL TEXT MESSAGING", Chichester, West Sussex: Wiley, 2010.