CSE 535 Mobile Computing
MOBILE MONITORING APPLICATION

Ankit Goyal
Arizona State University
agoyal22@asu.edu

Shruti Mahajan
Arizona State University
smahaja7@asu.edu

Abhishek Rao
Arizona State University
arao23@asu.edu

Abstract— We have developed a mobile monitoring application which intends to facilitate social order and culture by providing to the parent a descriptive report of the locations, calls and message transfer from and to the child mobile. Application provides a Call, Location and Message Tracking to enforce shadowing and control.

***Keywords*—Mobile Monitoring, SQLite Database, Authentication, Call Tracking, Message Tracking GPS, Tracking**

I. INTRODUCTION

Sometimes a need arises to monitor a person, profile or a role owing to some legitimate reason. Hence what can be of more use than a smart device which has become an essential part of our life-style. Hence we have come up with this application which monitors three elementary activities of our daily routine namely, Calls, texts and mobility.

This monitoring android application is developed to provide the essential information about the mobile holder activities. The main purpose of this application is to provide all call/text/location records to the person who is monitoring and also implement call/message blocking from a particular number. We are implementing single sign -In system, i.e the parent signs in and sets the username and password for the first time whenever the app is installed. As a result of which the child's access to this app will be restricted as nobody else other than parent can sign up and change the password.

II. CHALLENGES AND CONSIDERATIONS

A. *Android Permissions*

In the latest Android Versions, Mobile App needs to take permissions explicitly from the user. We are taking permissions to read Call Logs, Message, GPS Location, Contact List and to receive and send SMS.

B. *Cloud Server*

Since there was no cloud server to upload to, we decided to implement this application in child phone Itself. The problem is that the data will stay only in that particular phone.

C. *Database Storage Consideration.*

In the sqlite database it is not possible to store texts since it might contains vivid characters like ',', ':', '|', etc. and datatypes like varchar2 and nvarchar2 also doesn't support such characters. Therefore, to store messages, text file has been used.

III. IMPLEMENTATION AND ALGORITHMS

Previous Call Records and Text Messages:

This Feature reads the call log and messages from the mobile and stores them in a sqlite database and text file respectively. It stores the relevant details viz. time, duration body and date which fulfils objective of presenting the detailed analysis to the monitor agent. The deleted records from the phone will also be displayed since it has been stored intact.

Blocking a call/text & SMS Notification:

Whenever there is a call/text, application will receive it as well through broadcast receivers implemented. It then matches the contacting number from the list of blocked no. and checks if it is blocked or not. If it is a blocked no. then it will disconnect the call using telephony Manager and for a message a notification to the monitor agent. This service runs in background incessantly. Additionally it will send SMS to the no. which are already stored in the app to notify that Call Attempt to Blocked: Contact no. This is the notification sent using SMSManager.

```
String serviceManagerName =
"android.os.ServiceManager";
String serviceManagerNativeName =
"android.os.ServiceManagerNative";
String telephonyName =
"com.android.internal.telephony.ITelephony";
Class<?> telephonyClass;
Class<?> telephonyStubClass;
Class<?> serviceManagerClass;
Class<?> serviceManagerNativeClass;
Method telephonyEndCall;
Object telephonyObject;
Object serviceManagerObject;
telephonyClass = Class.forName(telephonyName);
telephonyStubClass =
telephonyClass.getClasses()[0];
serviceManagerClass =
```

```
Class.forName(serviceManagerName);
serviceManagerNativeClass =
Class.forName(serviceManagerNativeName);
Method getService = // getDefaults[29];

serviceManagerClass.getMethod("getService",
String.class);
Method tempInterfaceMethod =
serviceManagerNativeClass.getMethod("asInterface",
IBinder.class);
Binder tmpBinder = new Binder();
tmpBinder.attachInterface(null, "fake");
serviceManagerObject =
tempInterfaceMethod.invoke(null, tmpBinder);
IBinder retbinder = (IBinder)
getService.invoke(serviceManagerObject, "phone");
Method serviceMethod =
telephonyStubClass.getMethod("asInterface",
IBinder.class);
telephonyObject = serviceMethod.invoke(null,
retbinder);
telephonyEndCall =
telephonyClass.getMethod("endCall");
telephonyEndCall.invoke(telephonyObject);
```

Call Analysis:

User can view detailed analysis of all the incoming and outgoing calls which has taken place in a specific duration of time. It will display the list in descending order based on the time of call. It will display the caller or receiver name, Type of the call, Time & Date, Duration etc. Al the information is stored in the database.

Blocking a no. & adding Notification no:

Parent can block any no. he wants to by adding the no. in the list of blocked contact list. Whenever there is any incoming or outgoing call to that no. it will be disconnected without the User control. Additionally, SMS notification is sent to the no. already added to the list of Notification Contact no. User can add as many as numbers in the Notification List. Notification is sent to all the no.

Content Filtering

When the Messages with the specified keywords inserted, are exchanged, the monitor agent is sent a notification. The receiver receives the messages and checks through the body and sends a sms notification.

Blocking text messages from specified contact number.

The receiver implemented has the highest priority set which is of maximum level of 2147483647. Hence it receives the text first and then checks with the address. If it happens to be one of the blocked number then the message broadcast in aborted.

LOCATION TRACKER

Whenever the user clicks on start tracking location, location tracker service which implements location listener is started. We are using both GPS and Network localization for tracking the user's location. As GPS is more accurate, we urge user to use GPS to track location. The GPS will be tracking the location updates of the user and consequently stores the longitude and latitude coordinates into the SQL database whenever there is a minimum distance change of 30 meters and minimum time lapse of 3 minutes. These filter values are chosen such that the database is not overloaded with more redundant same location coordinates.

Displaying Recent Locations

When the user clicks on display recent locations, the database values are retrieved and latitudinal and longitudinal coordinates of the locations are converted to a physical address using Geocoder class. There are chances that the user has not moved much from his current location in 3 minutes and we may get same location coordinates in the next location update. So, while displaying the recent locations we are implementing this algorithm: The address of the location of the first entry of database is compared with address of next consecutive entries and if the values are different, the corresponding location's address is added to an array list. If address is same, the corresponding location's address is discarded and the loop iterates to the next value and comparisons continue until we reach end of loop and all the addresses which are different from one another, are retrieved from an array list and displayed on the screen using a list view.

The user can also choose to display the locations on the map. Google Maps API is used for displaying recent the locations on a map.

IV. DATABASE CREATION

For both instances where data storage is required, the SQLite Database and a text file has used as the backend. Multiple tables and two textiles are created in a single database.

All the Call records are stored in the callLogs table (id INTEGER, contactno VARCHAR, tmdate VARCHAR, blocked BOOL, type VARCHAR, duration INTEGER). Contact no. to which notification will be sent is stored in notification table (id INTEGER, contactno VARCHAR). User Login information is stored in the Login table (id INTEGER username VARCHAR, password password). All the blocked numbers are stored in BLOCKEDNOS table (ID INTEGER,NUMBER VARCHAR). Number is the respective blocked no which the user wants to block.

All latitude and longitude coordinates are stored in recent-locations-table
(id INTEGER,LATITUDEDOUBLE,LONGITUDE DOUBLE) .

V. TASK COMPLETION STATUS

All the tasks are completed. In addition to these tasks, we made Login, SignUp and Notification No. Activities.

Task No.	Task	Assignee
1	Recent Call and previous Records - Acquisition: Retrieving the Incoming and Outgoing call and read the previous call records too.	Ankit Goyal
2	Call History - Delivery and Reception: Storing the complete call records in the Sqlite.	Ankit Goyal
3	Calls Analysis - Action: Providing the Detailed Call analysis for the user.	Ankit Goyal
4	Call blocking - Action: Disconnecting the call if there is an incoming call from the blocked number or if an attempt is made for an outgoing call and sending a notification to the user in the form of a text message.	Ankit Goyal
5	Call history of blocked calls - Delivery and Reception: Storing the call history of blocked calls in sqlite.	Ankit Goyal
6	Retrieving Text Messages	Shruti Mahajan
7	Content Filtering	Shruti Mahajan
8	Blocking Text Messages from specified number.	Shruti Mahajan
9	Retrieving messages for specified duration	Shruti Mahajan
10	Notifying the Monitor agent	Shruti Mahajan
11	Location using GPS : Acquisition: The current location of the user is retrieved using GPS.	Abhishek Rao
12	Location history: Delivery and Reception: The user's recent locations are stored in SQLite database.	Abhishek Rao
13	Recent Locations: Action: Detailed analysis on the current location and the history of recent locations is displayed to the parent user.	Abhishek Rao
14	Blocked Numbers :Acquisition : Retrieving the list of blocked numbers as an input from the user.	Abhishek Rao
15	Storing Blocked Numbers: Delivery and Reception : Storing certain phone numbers as blocked phone numbers in sqlite.	Abhishek Rao

VI. CONCLUSION

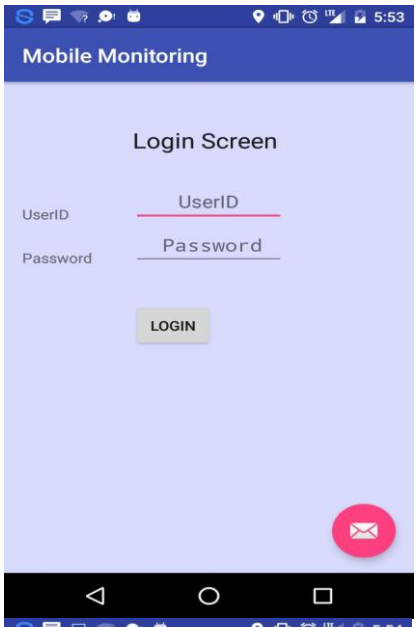
This application is developed to deploy in the mobile which you want to monitor. This application can monitor the Calls, SMS, and GPS Location precisely. Future scalability is to implement Cloud server from which person who wants to monitor other mobile will receive all the information in his mobile only through cloud.

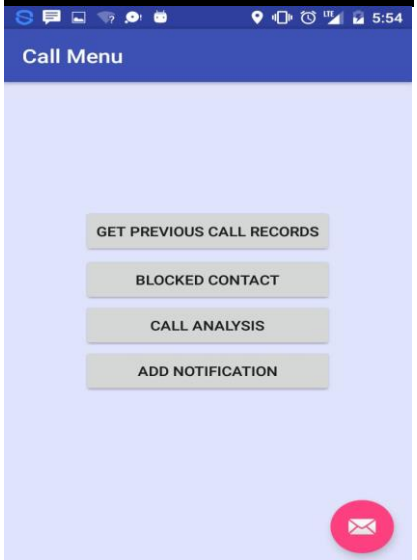
ACKNOWLEDGMENT

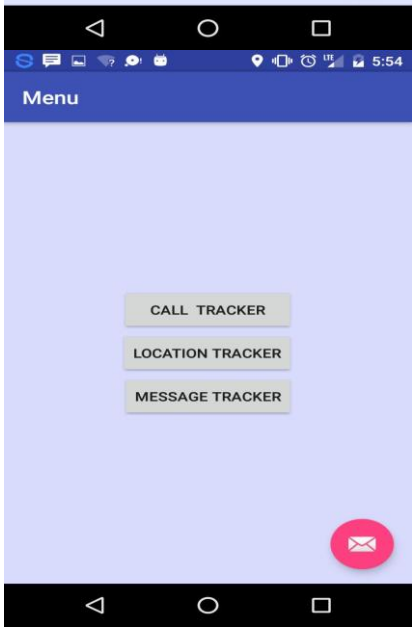
We thank Dr. Ayan Banerjee and our TA Junghyo Lee for providing this course and this android programming project. This project helped us in various ways and made us understand the mobile computing solutions better.

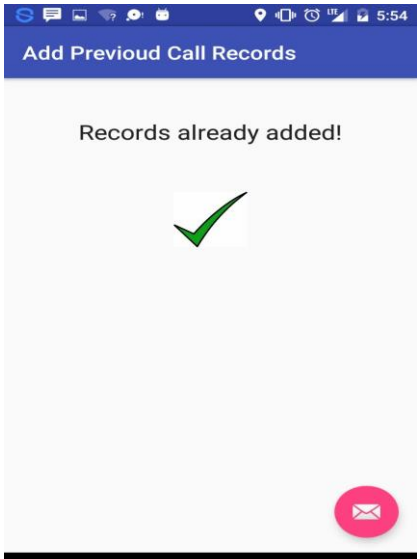
REFERENCES

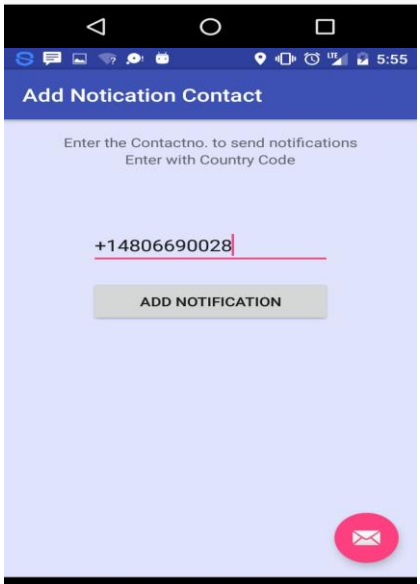
1. http://developer.android.com/reference/android/provider/CallLog.Calls.html
2. http://stackoverflow.com/questions/848728/how-can-i-read-sms-messages-from-the-inbox-programmatically-in-android
3. http://www.tutorialspoint.com/android/
4. http://www.codeproject.com/Articles/665527/A-GPS-Location-Plotting-Android-Application

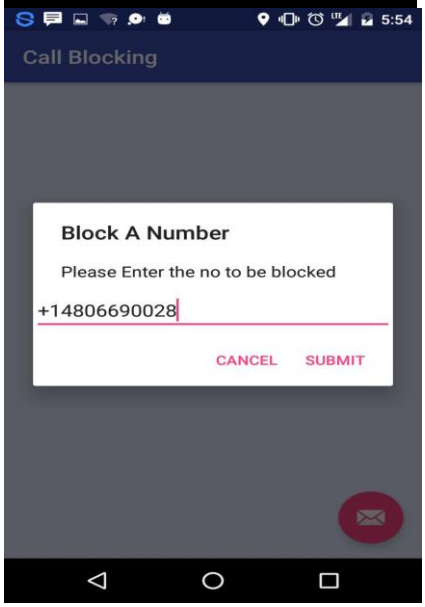

5:53
Mobile Monitoring
Login Screen
UserID
UserID
Password
Password
LOGIN

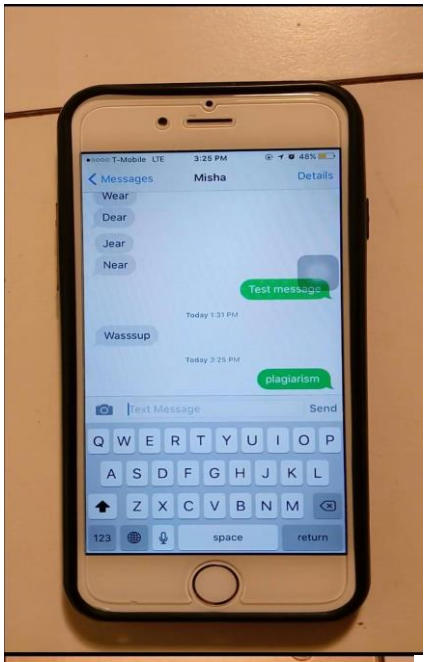

5:54
Call Menu
GET PREVIOUS CALL RECORDS
BLOCKED CONTACT
CALL ANALYSIS
ADD NOTIFICATION

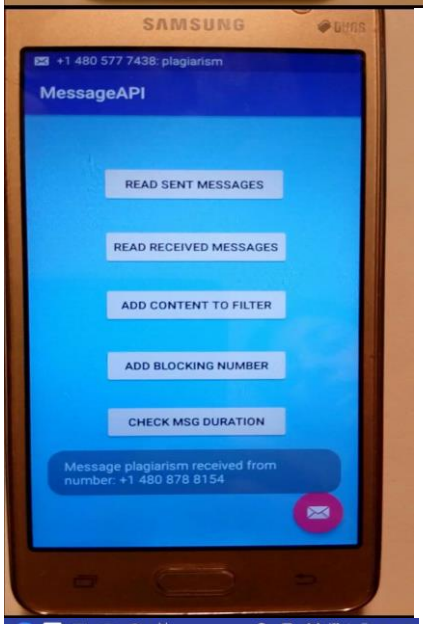

5:54
Menu
CALL TRACKER
LOCATION TRACKER
MESSAGE TRACKER


5:54
Add Previoud Call Records
Records already added!

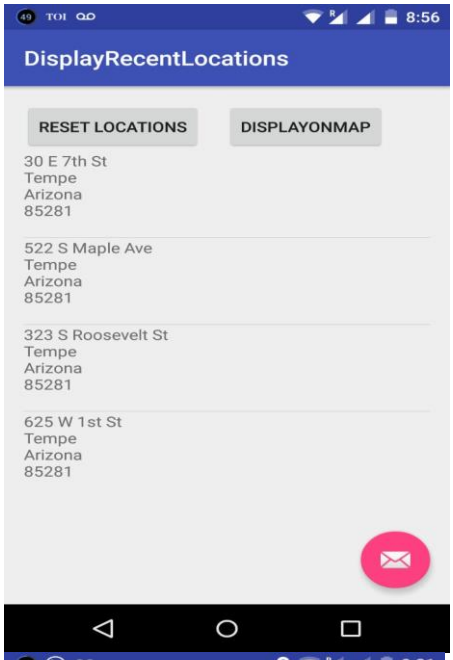

5:55
Add Notication Contact
Enter the Contactno. to send notifications
Enter with Country Code
+14806690028
ADD NOTIFICATION

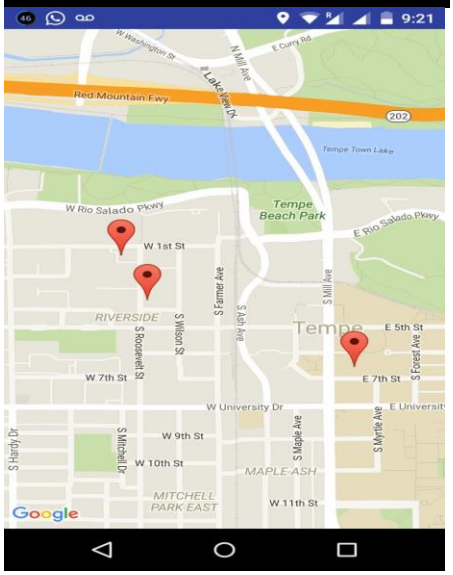

5:54
Call Blocking
Block A Number
Please Enter the no to be blocked
+14806690028
CANCEL
SUBMIT


T-Mobile LTE
3:25 PM
48%
Messages
Misha
Details
Wear
Dear
Jear
Near
Test message
Today 1:31 PM
Wasssup
Today 3:25 PM
plagiarism
Text Message
Send
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return


SAMSUNG
+1 480 577 7438: plagiarism
MessageAPI
READ SENT MESSAGES
READ RECEIVED MESSAGES
ADD CONTENT TO FILTER
ADD BLOCKING NUMBER
CHECK MSG DURATION
Message plagiarism received from number: +1 480 878 8154


5:54
Call Records Analysis
ALL CALLS
MOST FREQUENT CALLS
Enter the count
GET RECORDS
Contact No.
Date/Time
Duration
Asu Shreyas
2016-05-02 17:50:06
00:00:00
Asu Sanjans
2016-05-02 16:46:54
00:00:14
Asu Sanjans
2016-05-02 16:46:33
00:00:05
Asu Jaharsh
2016-05-02 16:45:38
00:00:29
Asu Jaharsh
2016-05-02 16:40:42
00:00:00
Asu Sanjans
2016-05-02 14:16:59
00:00:00
+14086699951
2016-05-02 13:59:21
00:00:15
+14086699951
2016-05-02 12:09:11


8:56
DisplayRecentLocations
RESET LOCATIONS
DISPLAYONMAP
30 E 7th St
Tempe
Arizona
85281
522 S Maple Ave
Tempe
Arizona
85281
323 S Roosevelt St
Tempe
Arizona
85281
625 W 1st St
Tempe
Arizona
85281


9:21
Red Mountain Fwy
202
Tempe Town Lake
Tempe Beach Park
W Rio Salado Pkwy
E Rio Salado Pkwy
W 1st St
RIVERSIDE
S Roosevelt St
S Wilson St
S Farmer Ave
S Ash Ave
S Mill Ave
Tempe
E 5th St
E 7th St
S Forest Ave
W 7th St
W University Dr
W 9th St
W 10th St
S Mitchell Dr
S Maple Ave
S Myrtle Ave
MAPLE-ASH
MITCHELL PARK EAST
W 11th St
Google