SWE599 Project, Spring 2014
Advisor: Fatih ALAGÖZ

Final Report

LittleRedButton
An Android Based Mobile Emergency Declaration Application

Revision 1.0
09.05.2014

By: Mustafa Göksu GÜRKAŞ
Student Id: 2011719225

Revision History

Author	Date	Explanation	Revision
M.Göksu GÜRKAŞ	09.05.2014	Initial version of the document is created.	1.0

Table of Contents

Revision History 2
Table of Contents 3
Table of Figures 4
1. Introduction 5
2. Purpose 5
3. Environment Information 7
3.1 Eclipse ADT 7
3.2 Android SDK 7
3.3 AVD Manager & Emulators 9
3.4 Android SharedPreferences 10
3.5 SQLite 10
3.6 Google Maps Android API v2 10
3.7 Facebook SDK for Android 3.0 11
3.8 Twitter4j 12
4. Design Decisions 12
4.1 Class Design 13
4.2 Business Workflow Design 14
    4.2.1 Facebook Activities Workflow 14
    4.2.2 Twitter Activities Workflow 14
    4.2.3 Recipients Activities Workflow 15
    4.2.4 Display Location on Maps Activities Workflow 15
    4.2.5 Get Help Activity Workflow 16
4.3 Page Navigation 17
5. Application Features 18
5.1 Main Screen 18
5.2 Settings Screen 19
5.3 Facebook Enable/Disable Screen 21
5.4 Twitter Enable/Disable Screen 22
5.5 Recipient Organization Screen 23
5.6 Nearest Places Screen 23
5.7 Emergency Declaration Screen 25
6. Conclusions and Future Work 29
7. References 29

Table of Figures

Figure 1: Use Case Diagram 6
Figure 2: Android SDK - Build components 8
Figure 3: Android SDK - Build tools and processes 8
Figure 4: Android application development phases 9
Figure 5: Android device emulator 10
Figure 6: Facebook SDK protocol flow 11
Figure 7: LittleRedButton class diagram 13
Figure 8: Facebook Activity Diagram 14
Figure 9: Twitter Activity Diagram 14
Figure 10: Recipients Activity Diagram 15
Figure 11: Display Location on Maps Activity Diagram 15
Figure 12: Get Help Activity Diagram 16
Figure 13: Page Navigation Map 17
Figure 14: Main Screen 18
Figure 15: Settings Screen-1 19
Figure 16: Settings Screen-2 20
Figure 17: Settings Screen-3 20
Figure 18: Facebook Login & Permissions Screen 21
Figure 19: Main Screen after Facebook Login 21
Figure 20: Twitter Login & Redirection Screen 22
Figure 21: Main Screen after Twitter Login 22
Figure 22: All Contacts Screen Figure 23: Recipient Display and Deletion Screen 23
Figure 24: User’s Location on the Google Maps 24
Figure 25: Nearest Help Points on Google Maps 24
Figure 26: Emergency Declaration Workflow 26
Figure 27: Recipient's SMS Inbox 27
Figure 28: Recipient's Mail Inbox 27
Figure 29: Recipient's Facebook Wall 28
Figure 30: Recipient's Twitter Timeline 28

1. Introduction

The LittleRedButton project is a mobile application design and development project for users who need urgent help in some cases. The goal is firstly shoot a picture of the scene where the user is, then to locate the user who declares an emergency occasion and finally to share his/her GPS location coordinates (with the photo taken as an attachment) through various selected notification channels to the people whom the user selected as recipients of the application previously. The recipient list is created and the notification channels are selected formerly by the user. When the user needs to notify his friends that he/she is in an urgent situation, he/she simply just presses the red button. The LittleRedButton sends either a Facebook post, a tweet, an e-mail or a SMS to its recipients immediately.

In latter sections, application development environment and application features leveraged by mockups from the application itself shall be described.

2. Purpose

The purpose of this software project is to design and develop a new mobile application on Android devices. The mobile application is to be implemented with JAVA for Android and by using a database system like MySQL. There will be only one actor in this mobile application which is the user himself/herself. One aims to accomplish the following tasks by using this system.

1. Users shall be able to locate himself/herself by using his smart phone's GPS/Network location services.
2. Users shall be able to add or delete his/her contacts in his/her smart phone as recipients.
3. Users shall be able to display his/her contact list that takes place in his/her smart phone memory.
4. Users shall be able to login to/logout from his/her Facebook account by using this application.
5. Users shall be able to notify his/her friends with his/her current GPS location coordinates as an emergency message via his/her Facebook profile.
6. Users shall be able to login to/logout from his/her Twitter account by using this application.
7. Users shall be able to notify his/her friends with his/her current GPS location coordinates as an emergency message via his/her Twitter profile.
8. Users shall be able to notify his/her selected recipients with his/her current GPS location coordinates as an emergency message via SMS.
9. Users shall be able to notify his/her selected recipients with his/her current GPS location coordinates as an emergency message via e-mail.
10. Users shall be able to let the application track himself/herself by sending emergency messages with his/her current (updated) GPS location coordinates to above channels every 60 seconds.
11. Users shall be able to display his/her current GPS location coordinates on Google Maps.
12. Users shall be able to display nearest police stations and health institutions to his/her current GPS location coordinates on Google Maps.
13. Users shall be able to configure the application's behavior by arranging the settings of the application.

When the user declares an emergency occasion, the GPS location coordinates are extracted by using the mobile platform's location services. For sharing through different channels (such as SMS, e-mail, Facebook and Twitter), the mobile platform APIs and frameworks will be used.

The design is based on LRB Requirements Specification Document, Revision 1.0, in file Gurkas-MustafaGoksu-LittleRedButton-RSD-2014-03-09-Rev-1.0.pdf [1]. The design approach of this project consists of standard activity lifecycle of Android mobile operating systems.

The use case diagram of is shown below to depict the main functionalities of LittleRedButton application.

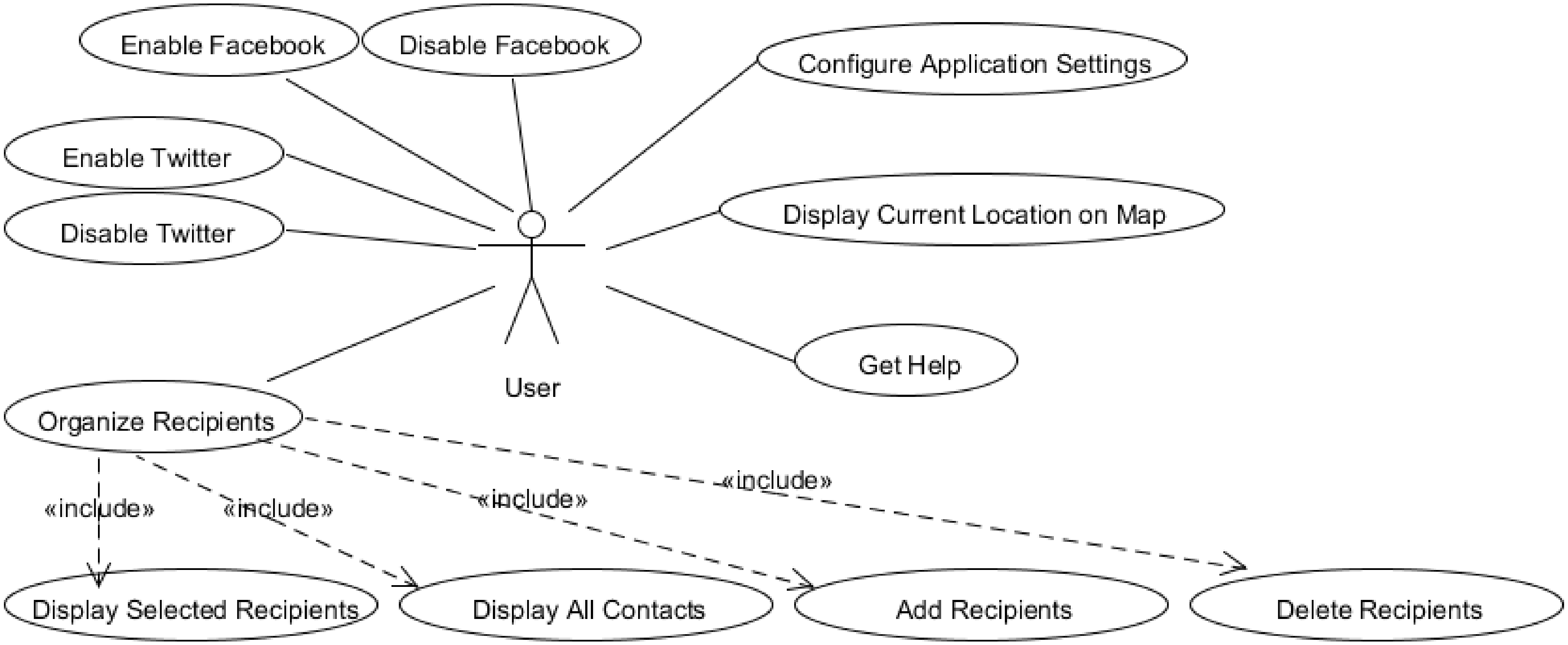


Figure 1: Use Case Diagram

3. Environment Information

The LittleRedButton application has been developed on Eclipse ADT IDE in JAVA platform with Android Framework. Testing of the application has been handled on both Android Emulator on personal computer and with Android device Samsung Galaxy SIII (Model: GT-I9300) which has an Android OS version as 4.3 and Kernel version as 3.0.31-2168382.

3.1 Eclipse ADT

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Fortran, Haskell, Perl, PHP, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, and Scheme. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

ADT (Android Developer Tools) is a plugin for Eclipse that provides a suite of tools that are integrated with the Eclipse IDE. It offers you access to many features that help you develop Android applications quickly. ADT provides GUI access to many of the command line SDK tools as well as a UI design tool for rapid prototyping, designing, and building of your application's user interface.
Because ADT is a plugin for Eclipse, you get the functionality of a well-established IDE, along with Android-specific features that are bundled with ADT. The following describes important features of Eclipse and ADT [2]:

- **Integrated Android project creation, building, packaging, installation, and debugging**: ADT integrates many development workflow tasks into Eclipse, making it easy for you to rapidly develop and test your Android applications.
- **SDK Tools integration:**Many of the SDK tools are integrated into Eclipse's menus, perspectives, or as a part of background processes ran by ADT.
- **Java programming language and XML editors:**The Java programming language editor contains common IDE features such as compile time syntax checking, auto-completion, and integrated documentation for the Android framework APIs. ADT also provides custom XML editors that let you edit Android-specific XML files in a form-based UI. A graphical layout editor lets you design user interfaces with a drag and drop interface.
- **Integrated documentation for Android framework APIs:** You can access documentation by hovering over classes, methods, or variables.

3.2 Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. The Android software development kit (SDK) includes a comprehensive set of development tools. These include adebugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later; for the moment one can develop Android software on Android itself by using [AIDE - Android IDE - Java, C++] app and [Android java editor] app [3]. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely) [4]. An Android SDK of minimum API Level: 11 (Platform Version: 3.0.x – Version Code: HONEYCOMB) has been used for developing LittleRedButton.

During the build process, your Android projects are compiled and packaged into an .apk file, the container for your application binary. It contains all of the information necessary to run your application on a device or emulator, such as compiled .dex files (.class files converted to Dalvik byte code), a binary version of the AndroidManifest.xml file, compiled resources (resources.arsc) and uncompiled resource files for your application.

The following diagram depicts the components involved in building and running an application [5]:

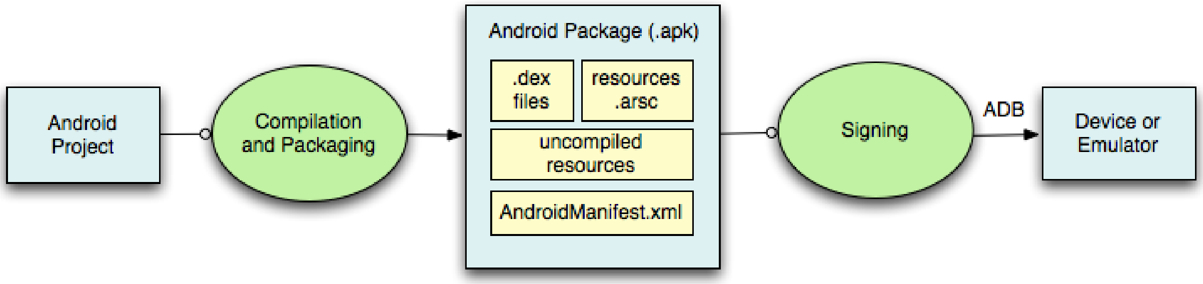


Figure 2: Android SDK - Build components

The build process involves many tools and processes that generate intermediate files on the way to producing an .apk. The following diagram depicts the different tools and processes that are involved in a build:

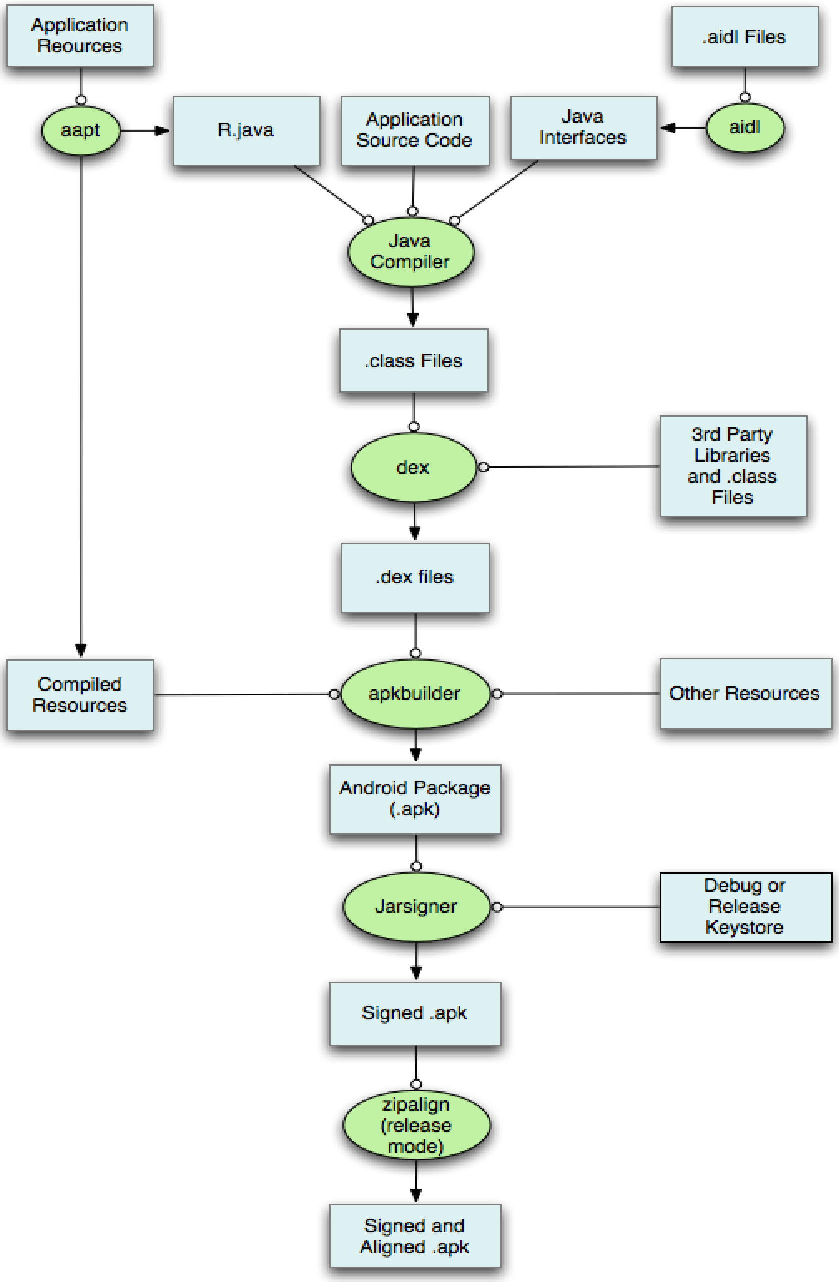


Figure 3: Android SDK - Build tools and processes

The basic steps for developing applications are shown in figure [6]:

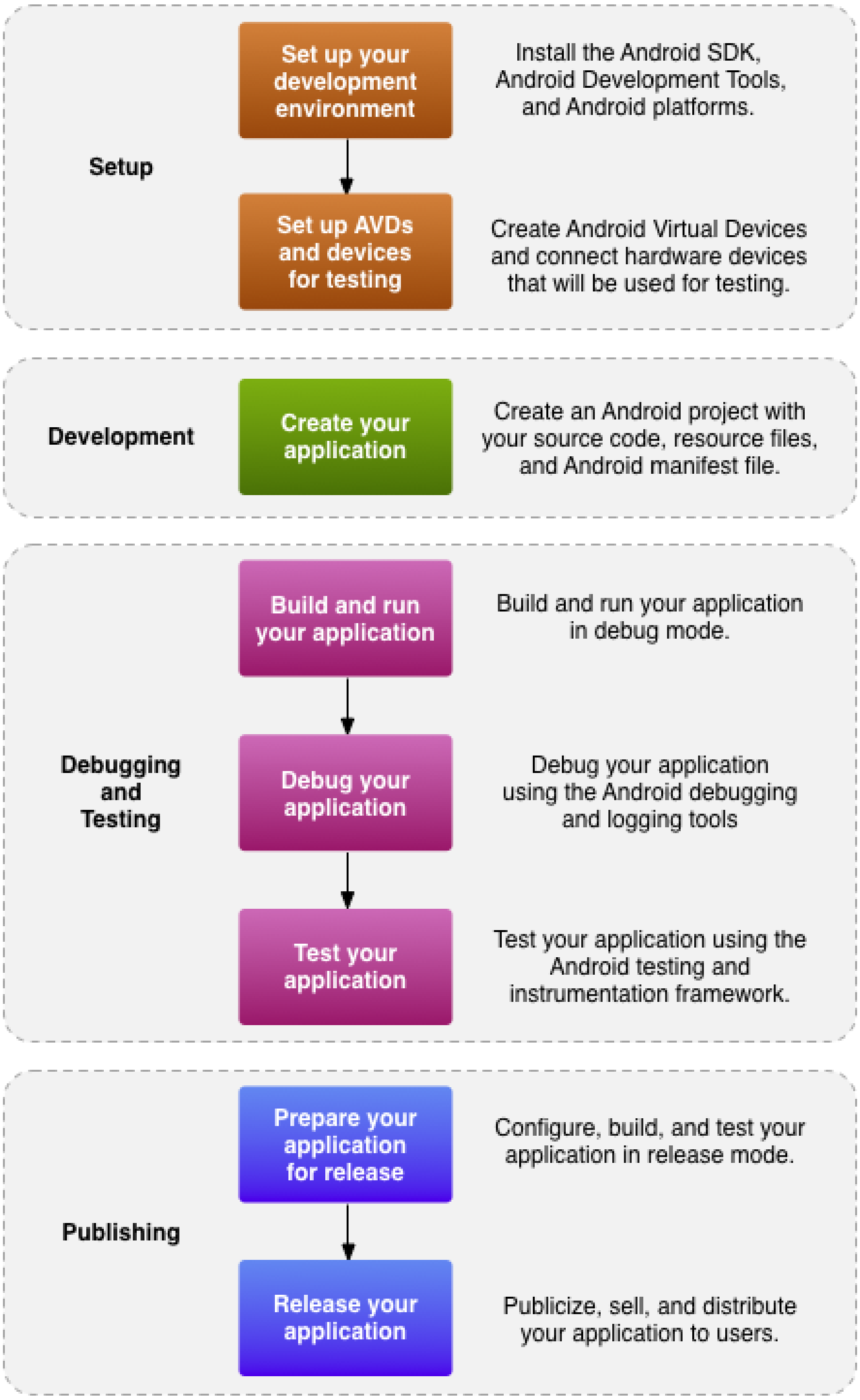

Setup
Set up your development environment
Install the Android SDK, Android Development Tools, and Android platforms.
Set up AVDs and devices for testing
Create Android Virtual Devices and connect hardware devices that will be used for testing.
Development
Create your application
Create an Android project with your source code, resource files, and Android manifest file.
Debugging and Testing
Build and run your application
Build and run your application in debug mode.
Debug your application
Debug your application using the Android debugging and logging tools
Test your application
Test your application using the Android testing and instrumentation framework.
Publishing
Prepare your application for release
Configure, build, and test your application in release mode.
Release your application
Publicize, sell, and distribute your application to users.

Figure 4: Android application development phases

3.3 AVD Manager & Emulators

The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs), which are required by the Android Emulator. The Android SDK includes a mobile device emulator — a virtual mobile device that runs on your computer. The emulator lets you develop and test Android applications without using a physical device. The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot place actual phone calls. It provides a variety of navigation and control keys, which you can "press" using your mouse or keyboard to generate events for your application. It also provides a screen in which your application is displayed, together with any other active Android applications [7]. A sample of an Android device emulator is shown below:

Figure 5: Android device emulator

3.4 Android SharedPreferences

If you have a relatively small collection of key-values that you'd like to save, you should use the SharedPreferences APIs. A SharedPreferences object points to a file containing key-value pairs and provides simple methods to read and write them. Each SharedPreferences file is managed by the framework and can be private or shared. For the user preferences that are set via application settings by the user shall be stored by Android SharedPreferences component.

3.5 SQLite

A database is an organized collection of data, today typically in digital form. The data are typically organized to model relevant aspects of reality (for example, the availability of rooms in hotels), in a way that supports processes requiring this information (for example, finding a hotel with vacancies).

SQLite is a relational database management system contained in a C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it. SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming languages [8].

LittleRedButton application has the SQLite light weighted SQL engine to keep the track of recipients list. SQLite comes with the application internally via application's apk file. There is no need for external SQLite installation. The database consists of only one table which keeps the related attributes of Recipient class. The column names are self-explanatory and the column types are as simple as possible: They are recipientname, recipientemailaddress, recipientphonenumber.

3.6 Google Maps Android API v2

With the Google Maps Android API, you can add maps based on Google Maps data to your application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. You can also use API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map. The API allows you to add these graphics to a map [9]:

- Icons anchored to specific positions on the map (Markers).
- Sets of line segments (Polylines).
- Enclosed segments (Polygons).
- Bitmap graphics anchored to specific positions on the map (Ground Overlays).
- Sets of images which are displayed on top of the base map tiles (Tile Overlays).

During the development of LittleRedButton’s “Where am I?” functionality, Google Maps Android v2 has been used in a collaboration with JSON files. In order to be able to use Google Maps API, project target is set to Google Inc.:Google APIs:19.

3.7 Facebook SDK for Android 3.0

You can incorporate Facebook functionality into your own applications. From the mobile perspective, the Facebook Platform supports APIs for mobile web applications, and mobile SDKs for native mobile applications for the iPhone, iPad, and Android platforms.

The core Facebook Platform API is the Graph API that allows you to read and write data to and from Facebook. Facebook also has what is called the Old Rest API. The newer Graph API changes the API paradigm from a method-oriented way of reading and writing data to and from Facebook to a new way that uses objects (think user profiles, friends, posts, photos, likes, and so on) and their relationships or connections with each other. This approach simplifies the Facebook API and makes it more consistent when working with objects. Note that while the Graph API is the preferred Facebook API, the Old REST API is still active and supported. Both the Graph and the REST APIs are applicable to mobile applications, both native and mobile web applications, including mobile web content within native applications through the use of WebViews.

With the Graph API, you can retrieve an object, delete an object, and publish objects. You can search, update objects, filter results, and even dynamically discover the connections/relationships of an object. By default, applications have access to the user's public data. To access private data, applications must first request the user's permissions, which are called extended permissions [10].

The Facebook SDK for Android is a Java programming language wrapper to the Facebook Graph and old REST APIs. This SDK is open source, and it is hosted at github's facebook / facebook-android-sdk repository.

The following items are use-cases for a typical Facebook application:

- Installing the Facebook SDK for Android
- Registering your application
- Creating your application
- Authorizing the user
- Making API requests

LittleRedButton makes use of Facebook SDK for Android on logging in & out a user to/from Facebook and posting feeds to user’s wall on emergency situations. A simple abstract flow for using Facebook through LittleRedButton is shown below:

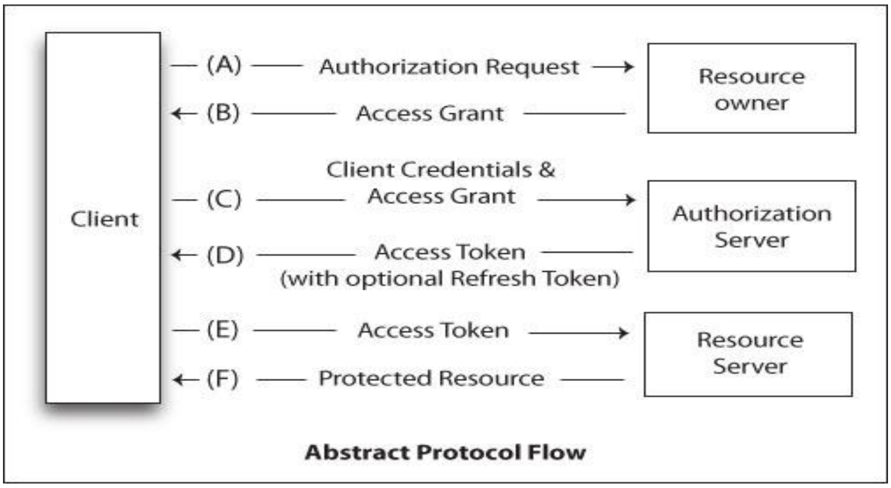

Client
— (A) — Authorization Request →
← (B) — Access Grant —
Resource owner
— (C) — Client Credentials & Access Grant →
← (D) — Access Token (with optional Refresh Token) —
Authorization Server
— (E) — Access Token →
← (F) — Protected Resource —
Resource Server
Abstract Protocol Flow

Figure 6: Facebook SDK protocol flow

3.8 Twitter4j

Twitter4j is a core java library which makes use of the Twitter API. In previous version, it was possible to use Basic Authentication but basic authentication has been removed in August 16th 2010. The new authentication protocol is OAuth.

OAuth (Open Authorization) is an open standard for authorization. It allows users to share their private resources (e.g. photos, videos, contact lists, Twitter account) stored on one site with another site without having to hand out their credentials, typically username and password.

OAuth allows users to hand out tokens instead of credentials to their data hosted by a given service provider. Each token grants access to a specific site (e.g. a video editing site) for specific resources (e.g. just videos from a specific album) and for a defined duration (e.g. the next 2 hours) if needed. This allows a user to grant a third party site access to their information stored with another service provider, without sharing their access permissions or the full extent of their data.

LittleRedButton makes use of Twitter4j library in order to simplify the integration with Twitter servers. By using this library, LittleRedButton shall be able to login/logout to/from Twitter and update user’s status with the emergency message being sent by the application.

4. Design Decisions

In this section, the design decisions that are used to develop LittleRedButton shall be explained in detail.

4.1 Class Design

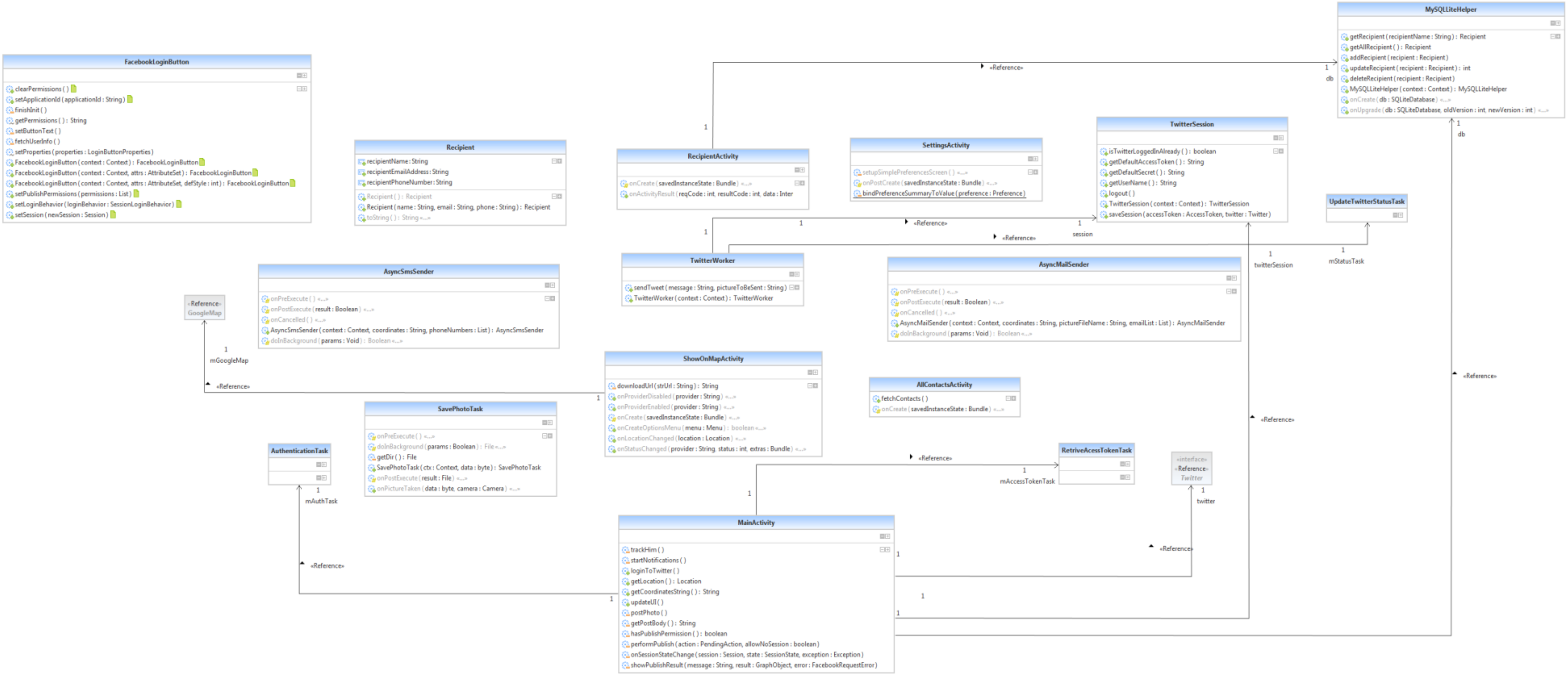

FacebookLoginButton
clearPermissions ( )
setApplicationId (applicationId : String)
finishInit ( )
getPermissions ( ) : String
setButtonText ( )
fetchUserInfo ( )
setProperties (properties : LoginButtonProperties)
FacebookLoginButton (context : Context) : FacebookLoginButton
FacebookLoginButton (context : Context, attrs : AttributeSet) : FacebookLoginButton
FacebookLoginButton (context : Context, attrs : AttributeSet, defStyle : int) : FacebookLoginButton
setPublishPermissions (permissions : List)
setLoginBehavior (loginBehavior : SessionLoginBehavior)
setSession (newSession : Session)
Recipient
recipientName : String
recipientEmailAddress : String
recipientPhoneNumber : String
Recipient ( ) : Recipient
Recipient (name : String, email : String, phone : String) : Recipient
toString ( ) : String «...»
RecipientActivity
onCreate (savedInstanceState : Bundle) «...»
onActivityResult (reqCode : int, resultCode : int, data : Inter
SettingsActivity
setupSimplePreferencesScreen ( ) «...»
onPostCreate (savedInstanceState : Bundle) «...»
bindPreferenceSummaryToValue (preference : Preference)
TwitterSession
isTwitterLoggedInAlready ( ) : boolean
getDefaultAccessToken ( ) : String
getDefaultSecret ( ) : String
getUserName ( ) : String
logout ( )
TwitterSession (context : Context) : TwitterSession
saveSession (accessToken : AccessToken, twitter : Twitter)
MySQLLiteHelper
getRecipient (recipientName : String) : Recipient
getAllRecipient ( ) : Recipient
addRecipient (recipient : Recipient)
updateRecipient (recipient : Recipient) : int
deleteRecipient (recipient : Recipient)
MySQLLiteHelper (context : Context) : MySQLLiteHelper
onCreate (db : SQLiteDatabase) «...»
onUpgrade (db : SQLiteDatabase, oldVersion : int, newVersion : int) «...»
UpdateTwitterStatusTask
«Reference»
1
db
session
twitterSession
mStatusTask
«Reference»
GoogleMap
mGoogleMap
AsyncSmsSender
onPreExecute ( ) «...»
onPostExecute (result : Boolean) «...»
onCancelled ( ) «...»
AsyncSmsSender (context : Context, coordinates : String, phoneNumbers : List) : AsyncSmsSender
doInBackground (params : Void) : Boolean «...»
TwitterWorker
sendTweet (message : String, pictureToBeSent : String)
TwitterWorker (context : Context) : TwitterWorker
AsyncMailSender
onPreExecute ( ) «...»
onPostExecute (result : Boolean) «...»
onCancelled ( ) «...»
AsyncMailSender (context : Context, coordinates : String, pictureFileName : String, emailList : List) : AsyncMailSender
doInBackground (params : Void) : Boolean «...»
ShowOnMapActivity
downloadUrl (strUrl : String) : String
onProviderDisabled (provider : String) «...»
onProviderEnabled (provider : String) «...»
onCreate (savedInstanceState : Bundle) «...»
onCreateOptionsMenu (menu : Menu) : boolean «...»
onLocationChanged (location : Location) «...»
onStatusChanged (provider : String, status : int, extras : Bundle) «...»
AllContactsActivity
fetchContacts ( )
onCreate (savedInstanceState : Bundle) «...»
SavePhotoTask
onPreExecute ( ) «...»
doInBackground (params : Boolean) : File «...»
getDir ( ) : File
SavePhotoTask (ctx : Context, data : byte) : SavePhotoTask
onPostExecute (result : File) «...»
onPictureTaken (data : byte, camera : Camera) «...»
AuthenticationTask
mAuthTask
RetriveAcessTokenTask
mAccessTokenTask
«interface»
«Reference»
Twitter
twitter
MainActivity
trackHim ( )
startNotifications ( )
loginToTwitter ( )
getLocation ( ) : Location
getCoordinatesString ( ) : String
updateUI ( )
postPhoto ( )
getPostBody ( ) : String
hasPublishPermission ( ) : boolean
performPublish (action : PendingAction, allowNoSession : boolean)
onSessionStateChange (session : Session, state : SessionState, exception : Exception)
showPublishResult (message : String, result : GraphObject, error : FacebookRequestError)

Figure 7: LittleRedButton class diagram

4.2 Business Workflow Design

4.2.1 Facebook Activities Workflow

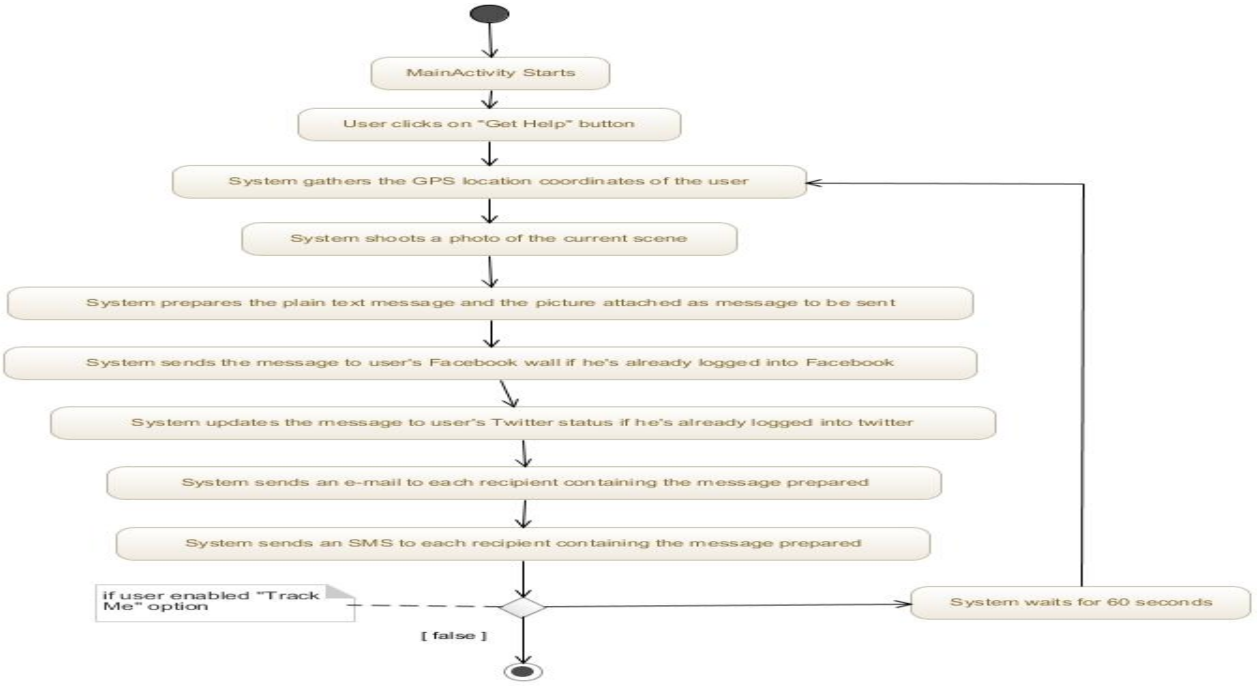

MainActivity Starts
User clicks on "Get Help" button
System gathers the GPS location coordinates of the user
System shoots a photo of the current scene
System prepares the plain text message and the picture attached as message to be sent
System sends the message to user's Facebook wall if he's already logged into Facebook
System updates the message to user's Twitter status if he's already logged into twitter
System sends an e-mail to each recipient containing the message prepared
System sends an SMS to each recipient containing the message prepared
if user enabled "Track Me" option
System waits for 60 seconds
[false]

Figure 8: Facebook Activity Diagram

4.2.2 Twitter Activities Workflow

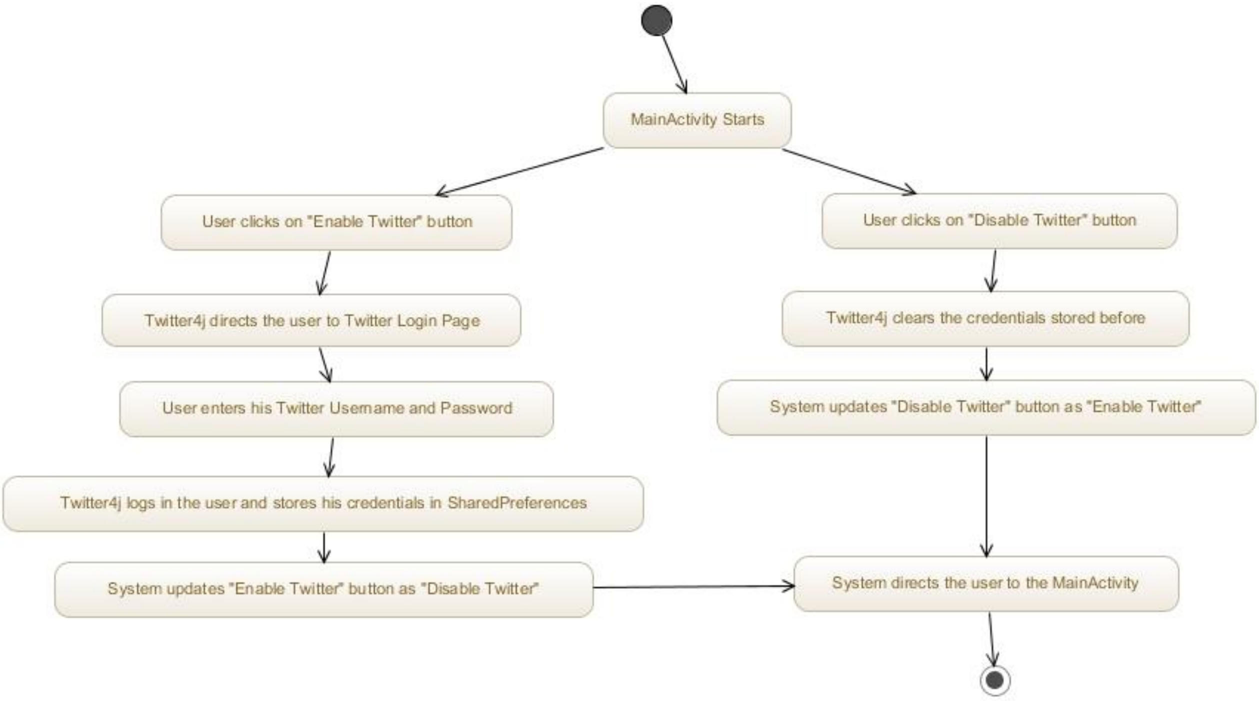

MainActivity Starts
User clicks on "Enable Twitter" button
User clicks on "Disable Twitter" button
Twitter4j directs the user to Twitter Login Page
Twitter4j clears the credentials stored before
User enters his Twitter Usemame and Password
System updates "Disable Twitter" button as "Enable Twitter"
Twitter4j logs in the user and stores his credentials in SharedPreferences
System updates "Enable Twitter" button as "Disable Twitter"
System directs the user to the MainActivity

Figure 9: Twitter Activity Diagram

4.2.3 Recipients Activities Workflow

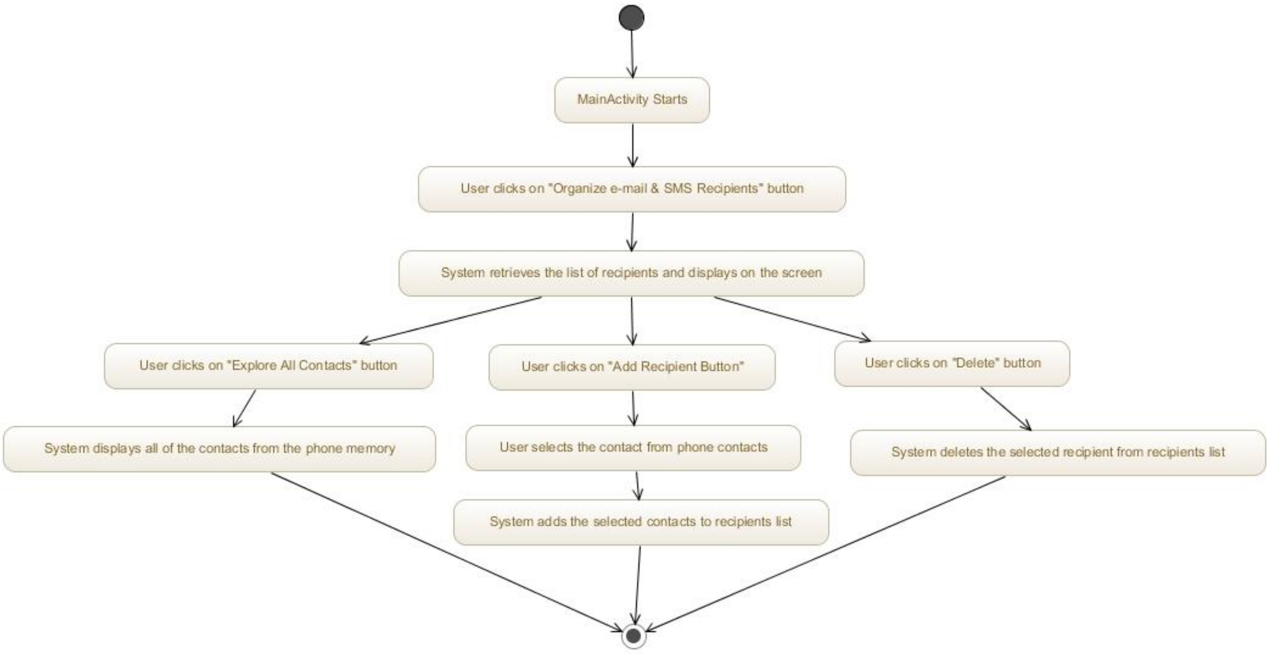

MainActivity Starts
User clicks on "Organize e-mail & SMS Recipients" button
System retrieves the list of recipients and displays on the screen
User clicks on "Explore All Contacts" button
User clicks on "Add Recipient Button"
User clicks on "Delete" button
System displays all of the contacts from the phone memory
User selects the contact from phone contacts
System deletes the selected recipient from recipients list
System adds the selected contacts to recipients list

Figure 10: Recipients Activity Diagram

4.2.4 Display Location on Maps Activities Workflow

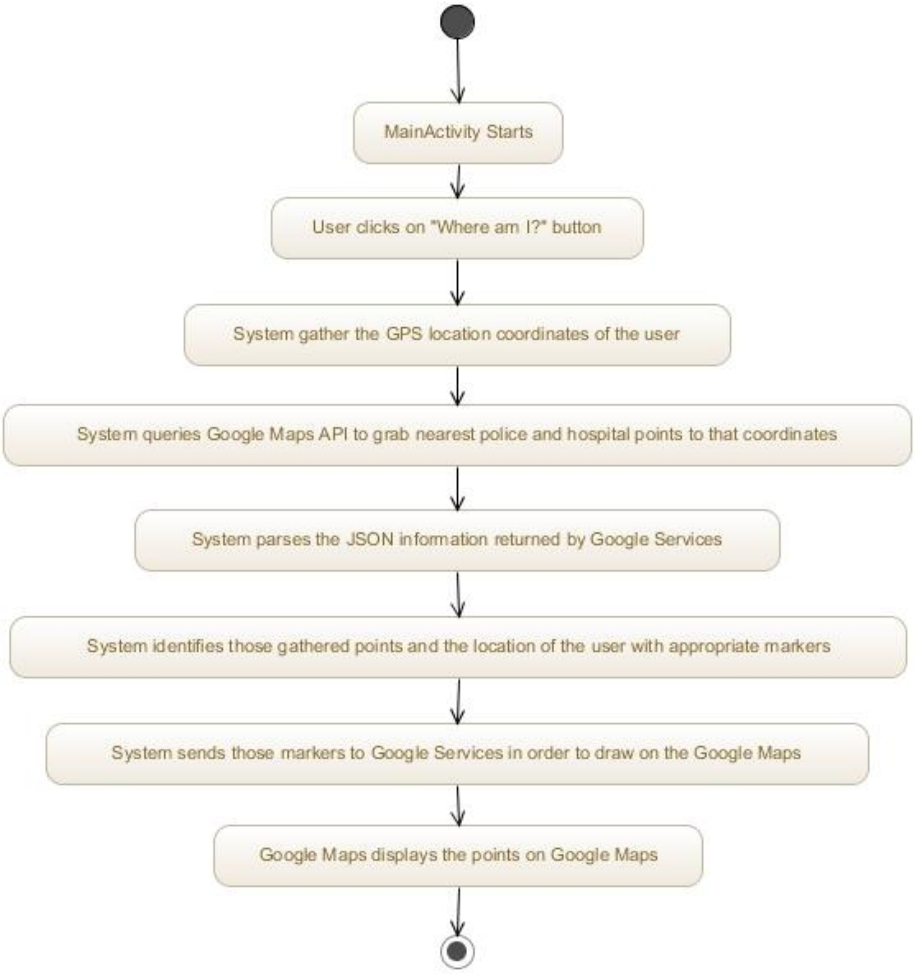

MainActivity Starts
User clicks on "Where am I?" button
System gather the GPS location coordinates of the user
System queries Google Maps API to grab nearest police and hospital points to that coordinates
System parses the JSON information returned by Google Services
System identifies those gathered points and the location of the user with appropriate markers
System sends those markers to Google Services in order to draw on the Google Maps
Google Maps displays the points on Google Maps

Figure 11: Display Location on Maps Activity Diagram

4.2.5 Get Help Activity Workflow

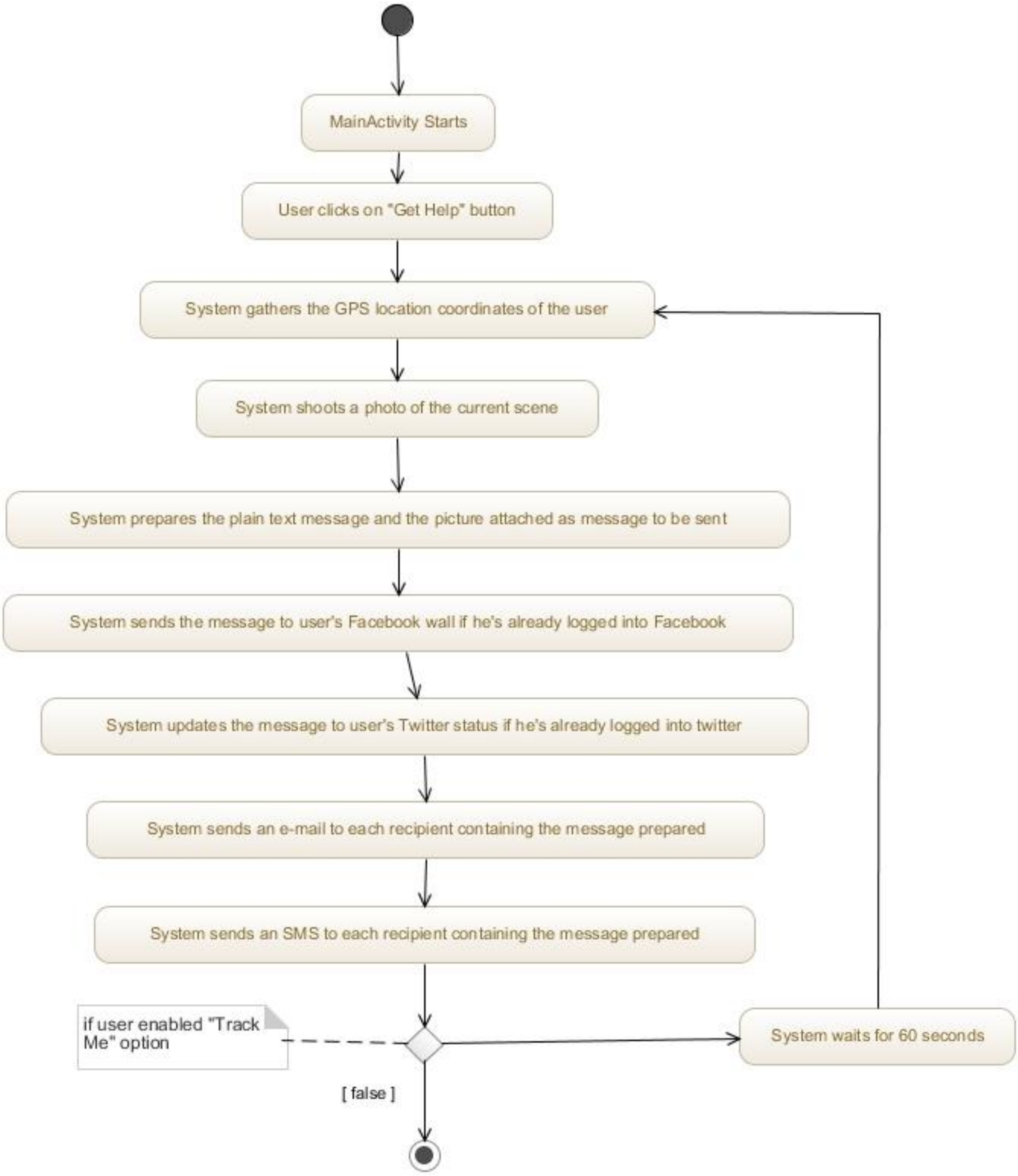

MainActivity Starts
User clicks on "Get Help" button
System gathers the GPS location coordinates of the user
System shoots a photo of the current scene
System prepares the plain text message and the picture attached as message to be sent
System sends the message to user's Facebook wall if he's already logged into Facebook
System updates the message to user's Twitter status if he's already logged into twitter
System sends an e-mail to each recipient containing the message prepared
System sends an SMS to each recipient containing the message prepared
if user enabled "Track Me" option
System waits for 60 seconds
[false]

Figure 12: Get Help Activity Diagram

4.3 Page Navigation

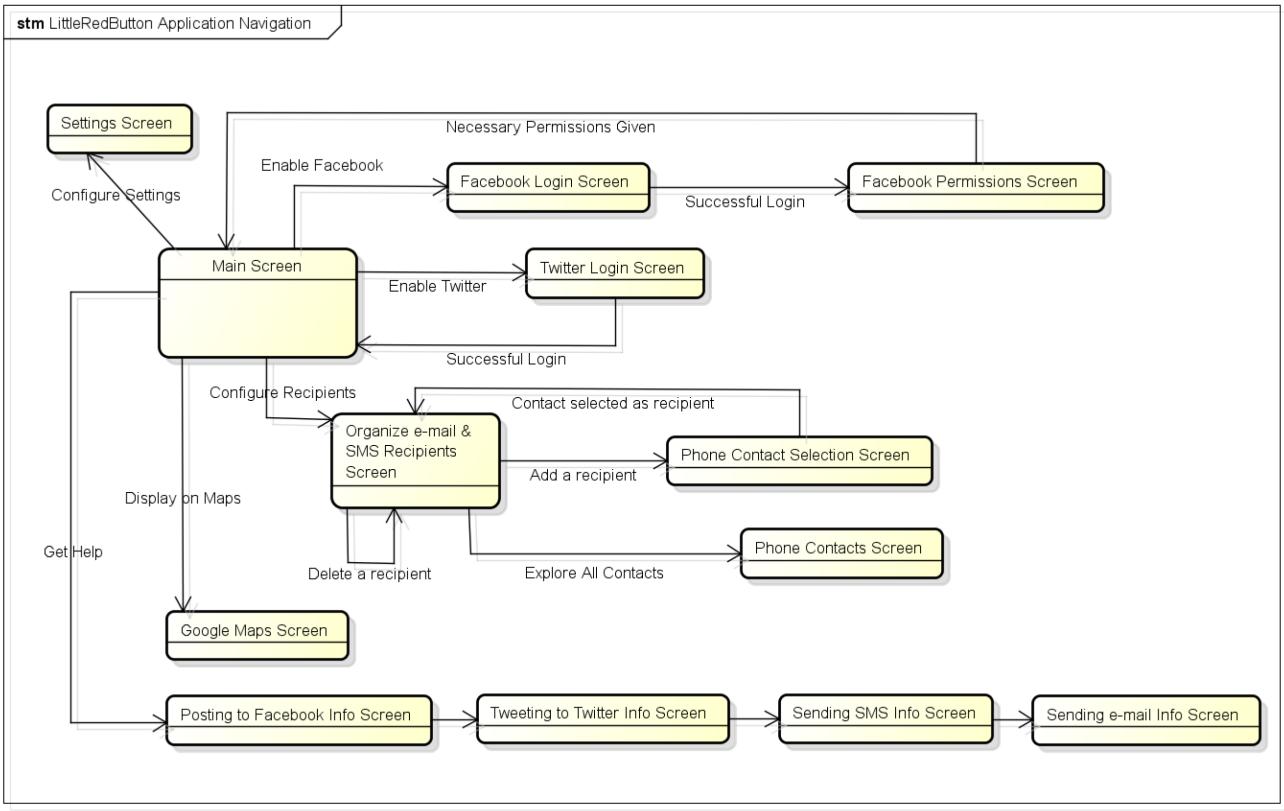


Figure 13: Page Navigation Map

5. Application Features

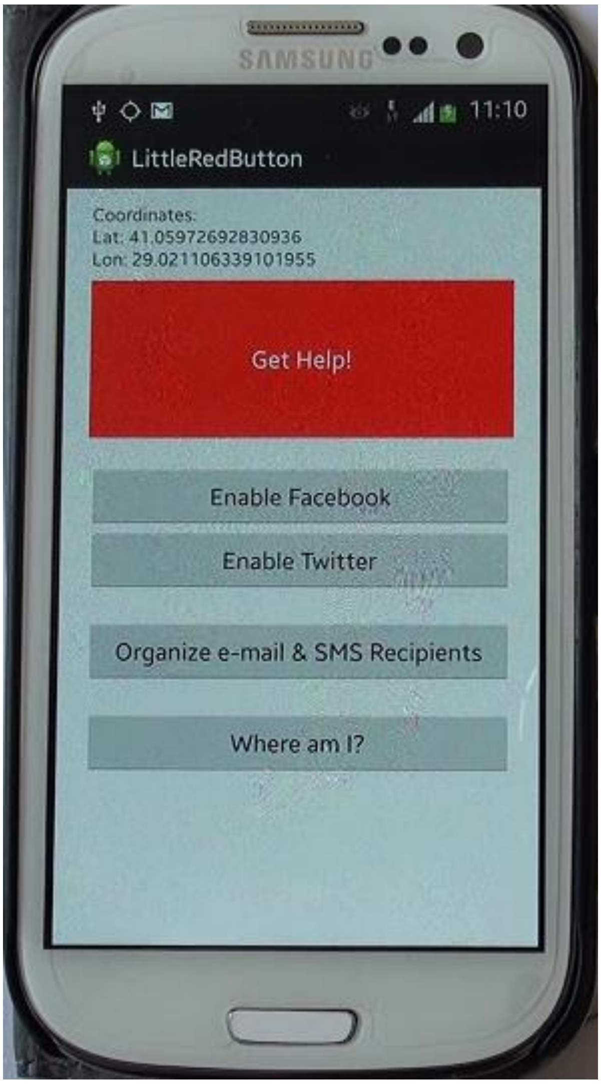
By using LittleRedButton, the user shall be able to locate himself/herself and send his/her GPS location coordinates in an emergency message to his previously created recipient list through selected channels.

The emergency message includes GPS location coordinates of the user and also Google Maps URL that will directly represent the user’s location on the Google Maps on receiver’s device. For the messages that is sent to Facebook, Twitter and e-mail, a picture of the scene that the user is currently at shall be attached. SMS message shall only contain the message and the coordinates URL data.

The application user interfaces and the user experience sequence are described in latter sections.

5.1 Main Screen

When the user launches LittleRedButton the following screen shall be displayed. On this screen, user shall be able to get his current GPS location coordinates values and display help button, Facebook enable/disable button, Twitter enable/disable button, recipient organization button and the button that is used to display his current coordinates on Google Maps.

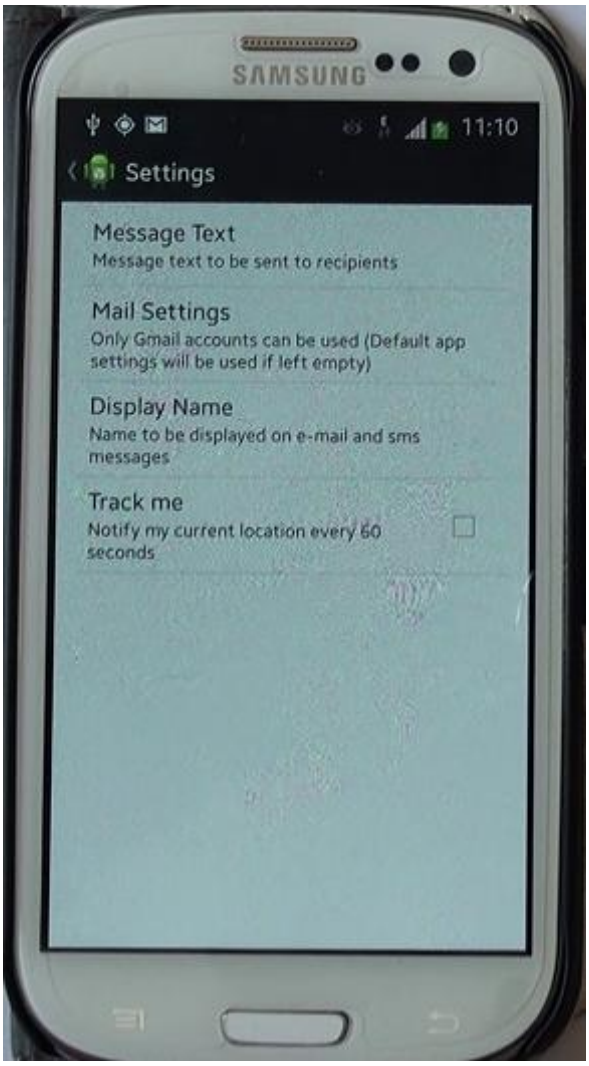


Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Figure 14: Main Screen

5.2 Settings Screen

From any page of LittleRedButton, the user shall be able to tap on the “Settings” button of his/her Android device and open Settings screen of the application. By using this screen, the user shall be able to set his/her desired text to be sent with emergency message, set his Gmail account information that would be used to send e-mail on behalf of the user (LittleRedButton shall use its default Gmail settings if those settings are left empty), set the display name which is used to represent the user’s name on the sent e-mails and to enable/disable tracking facility of LittleRedButton. If the user enables “Track Me” option from the settings screen LittleRedButton shall send a new picture with the updated current GPS location coordinates of the user every 60 seconds via selected channels. The settings screen and an example of its usage is depicted below.



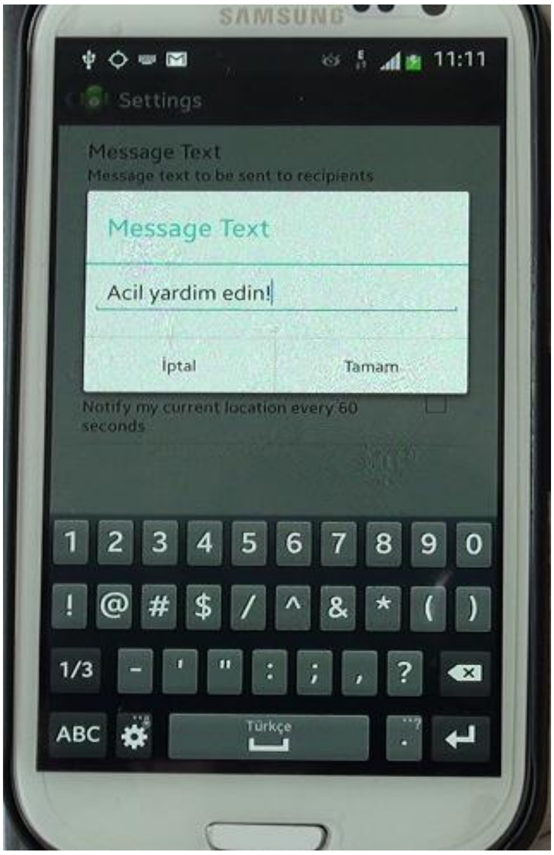
Message Text
Message text to be sent to recipients

Mail Settings
Only Gmail accounts can be used (Default app settings will be used if left empty)

Display Name
Name to be displayed on e-mail and sms messages

Track me
Notify my current location every 60 seconds □

Figure 15: Settings Screen-1



Message Text

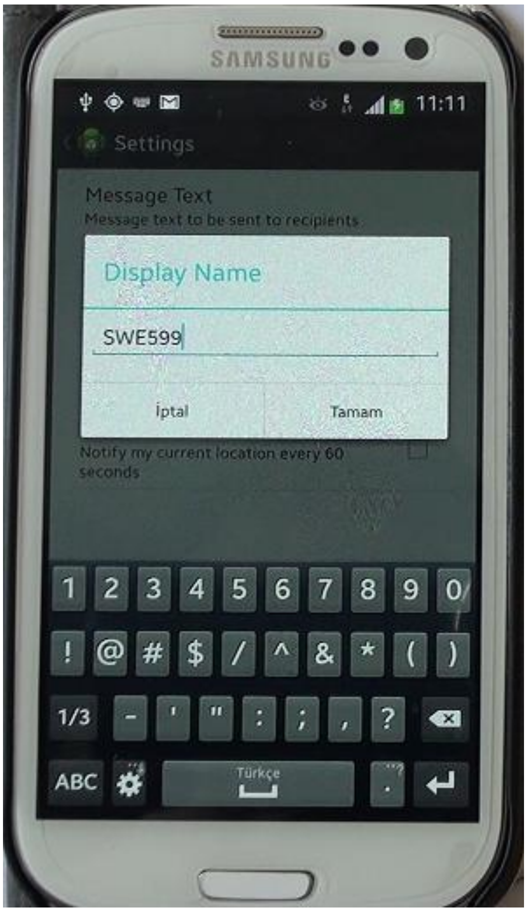
Message text to be sent to recipients

Message Text

Acil yardim edin!

Notify my current location every 60 seconds □

Figure 16: Settings Screen-2



Message Text

Message text to be sent to recipients

Display Name

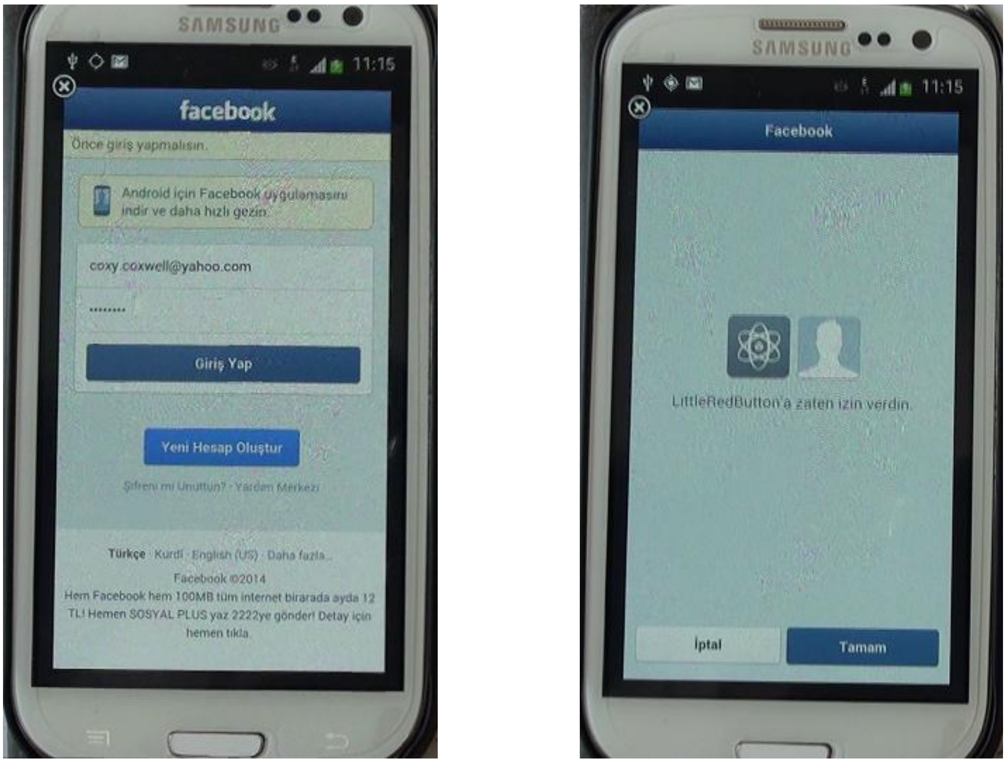
SWE599

Notify my current location every 60 seconds □

Figure 17: Settings Screen-3

5.3 Facebook Enable/Disable Screen

When the user wants to enable his/her Facebook account in order to use Facebook as an emergency notification channel, he/she has to click on “Enable Facebook” button that takes place on the main screen. After that, LittleRedButton directs the user to Facebook login page in order to let the user give his/her credentials to Facebook servers and get an access token to Facebook servers. LittleRedButton does not want user’s Facebook account information nor stores those values to device memory. The following screenshots show the workflow for Facebook activation via LittleRedButton.



Önce giriş yapmalısın.

Android için Facebook uygulamasını indir ve daha hızlı gezin.

coxy.coxwell@yahoo.com

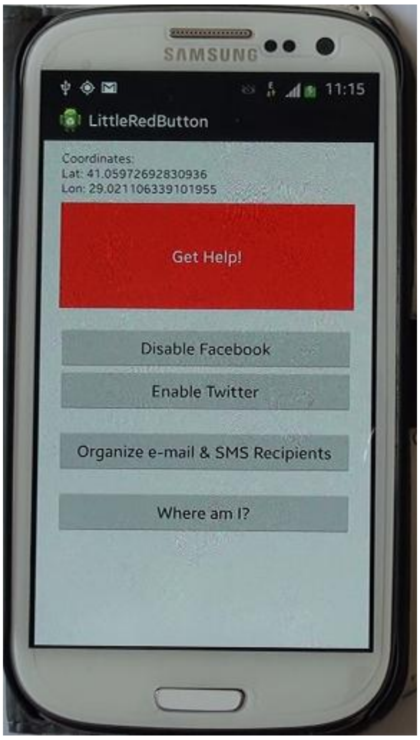
Facebook ©2014

Hem Facebook hem 100MB tüm internet birarada ayda 12 TL! Hemen SOSYAL PLUS yaz 2222ye gönder! Detay için hemen tıkla.

LittleRedButton'a zaten izin verdin.

SAMSUNG
11:15
facebook
Giriş Yap
Yeni Hesap Oluştur
Şifreni mi Unuttun? · Yardım Merkezi
Türkçe · Kurdî · English (US) · Daha fazla...
Facebook
İptal
Tamam

Figure 18: Facebook Login & Permissions Screen

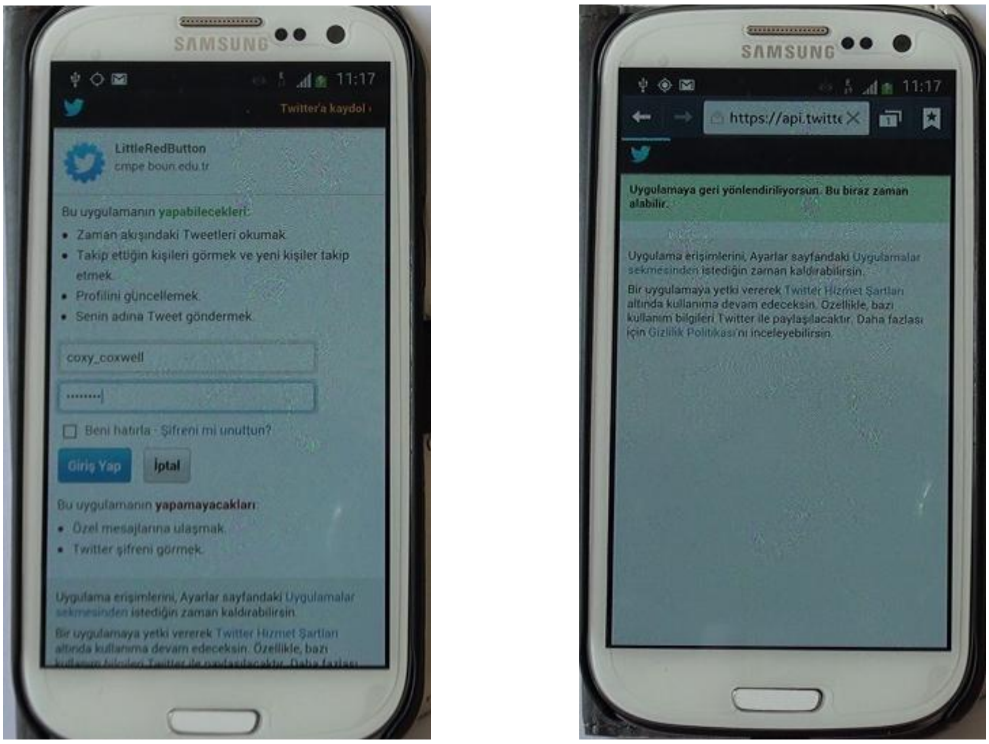


Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Figure 19: Main Screen after Facebook Login

5.4 Twitter Enable/Disable Screen

When the user wants to enable his/her Twitter account in order to use Twitter as an emergency notification channel, he/she has to click on "Enable Twitter" button that takes place on the main screen. After that, LittleRedButton directs the user to Twitter login page in order to let the user give his/her credentials to Twitter servers and get an access token to Twitter servers. LittleRedButton does not want user's Twitter account information nor stores those values to device memory. The following screenshots show the workflow for Twitter activation via LittleRedButton.



LittleRedButton
cmpe.boun.edu.tr

Bu uygulamanın **yapabilecekleri**:

- Zaman akışındaki Tweetleri okumak.
- Takip ettiğin kişileri görmek ve yeni kişiler takip etmek.
- Profilini güncellemek.
- Senin adına Tweet göndermek.

coxy_coxwell

••••••••

□ Beni hatırla · Şifreni mi unuttun?

Giriş Yap   İptal

Bu uygulamanın **yapamayacakları**:

- Özel mesajlarına ulaşmak.
- Twitter şifreni görmek.

Uygulama erişimlerini, Ayarlar sayfandaki Uygulamalar sekmesinden istediğin zaman kaldırabilirsin.

Bir uygulamaya yetki vererek Twitter Hizmet Şartları altında kullanıma devam edeceksin. Özellikle, bazı kullanım bilgileri Twitter ile paylaşılacaktır. Daha fazlası

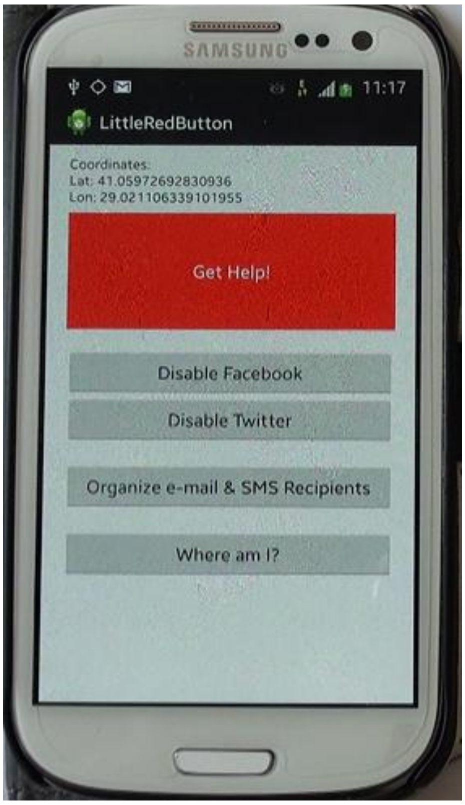
https://api.twitte

Uygulamaya geri yönlendiriliyorsun. Bu biraz zaman alabilir.

Uygulama erişimlerini, Ayarlar sayfandaki Uygulamalar sekmesinden istediğin zaman kaldırabilirsin.

Bir uygulamaya yetki vererek Twitter Hizmet Şartları altında kullanıma devam edeceksin. Özellikle, bazı kullanım bilgileri Twitter ile paylaşılacaktır. Daha fazlası için Gizlilik Politikası'nı inceleyebilirsin.

Figure 20: Twitter Login & Redirection Screen

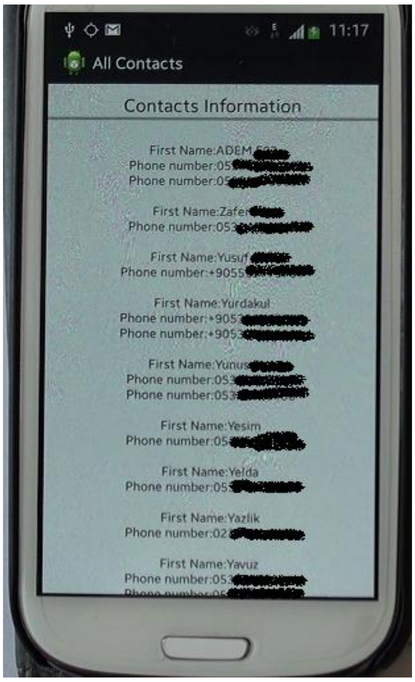


Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Figure 21: Main Screen after Twitter Login

5.5 Recipient Organization Screen

The user shall be able to view all contacts that take place on the phone memory (only the information composed of contact’s name, phone number (if any) and e-mail address (if any)) and add/delete those contacts as recipients to LittleRedButton. The following screens are part of recipient organization workflow. When the user clicks on “Add Recipient” button the phonebook of the phone shall be opened and user shall be able to select one of the contacts as recipient. The selected contact shall be stored and displayed on “Organize Recipients” screen with a “Delete” button next to his/her name.



Contacts Information

First Name:ADEM [redacted]
Phone number:05[redacted]
Phone number:05[redacted]

First Name:Zafer[redacted]
Phone number:053[redacted]

First Name:Yusuf[redacted]
Phone number:+9055[redacted]

First Name:Yurdakul
Phone number:+9053[redacted]
Phone number:+9053[redacted]

First Name:Yunus[redacted]
Phone number:053[redacted]
Phone number:053[redacted]

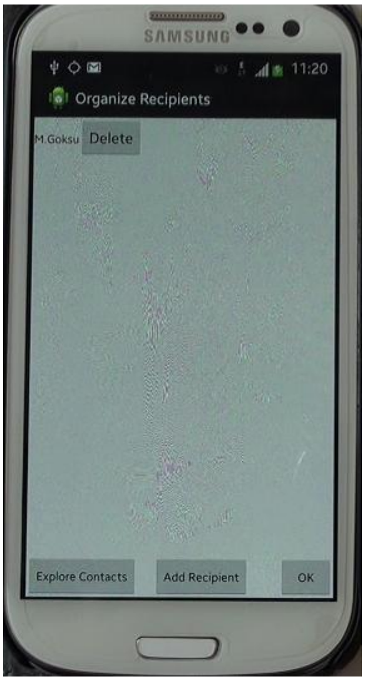
First Name:Yesim
Phone number:05[redacted]

First Name:Yelda
Phone number:05[redacted]

First Name:Yazlik
Phone number:02[redacted]

First Name:Yavuz
Phone number:053[redacted]
Phone number:05[redacted]

Figure 22: All Contacts Screen



M.Goksu

Figure 23: Recipient Display and Deletion Screen

5.6 Nearest Places Screen

The user shall be able to locate himself/herself on Google Maps by clicking on “Where am I?” button that is on the main screen of the application. Besides his/her location the nearest police and health points to that GPS location coordinates shall also be displayed to the user. User shall be green, police stations shall be blue and health points shall be displayed by red markers on the map. When the user clicks on a marker, LittleRedButton shall tell the user the name and the address of the place he/she selected. The following screens show a few example of that map view.

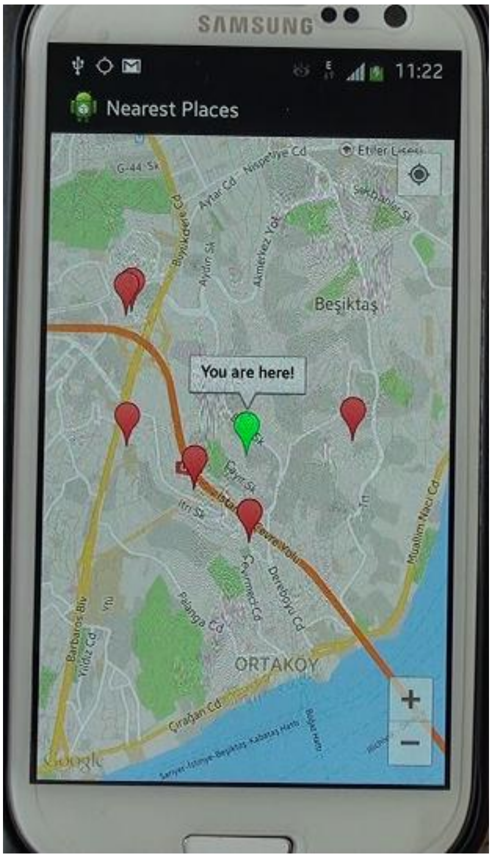


Figure 24: User's Location on the Google Maps

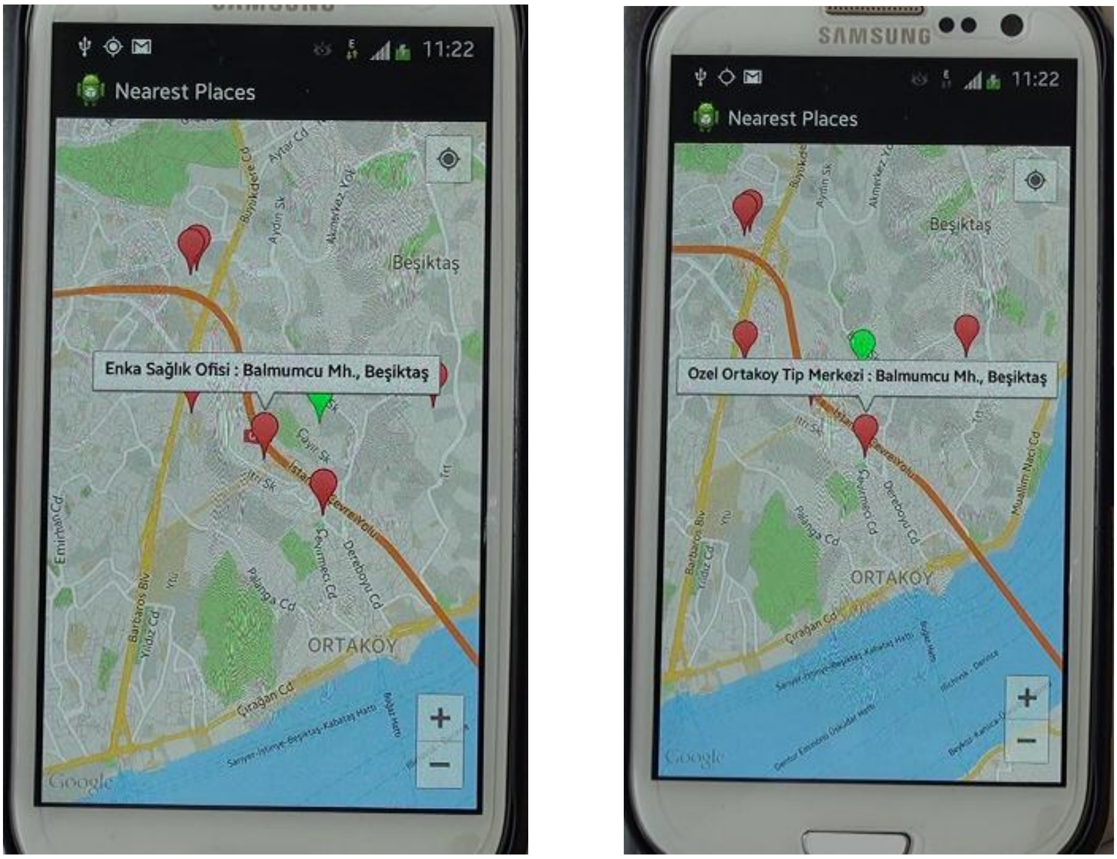
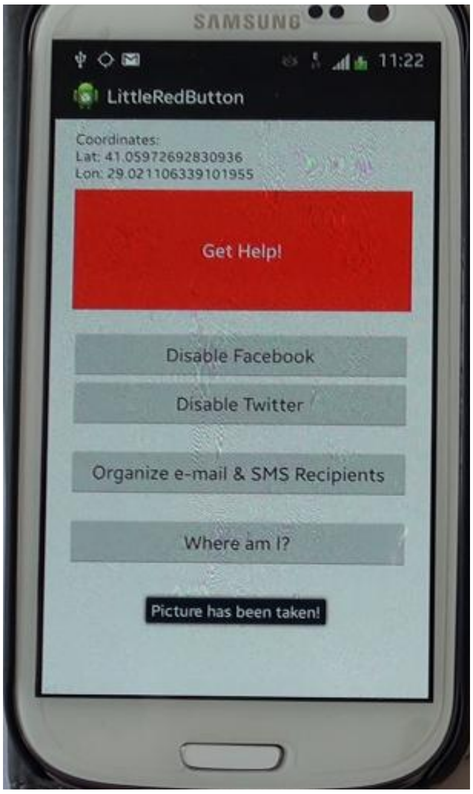

11:22
Nearest Places
Enka Sağlık Ofisi : Balmumcu Mh., Beşiktaş
Beşiktaş
ORTAKÖY
SAMSUNG
11:22
Nearest Places
Ozel Ortakoy Tip Merkezi : Balmumcu Mh., Beşiktaş
Beşiktaş
ORTAKÖY
+
−

Figure 25: Nearest Help Points on Google Maps

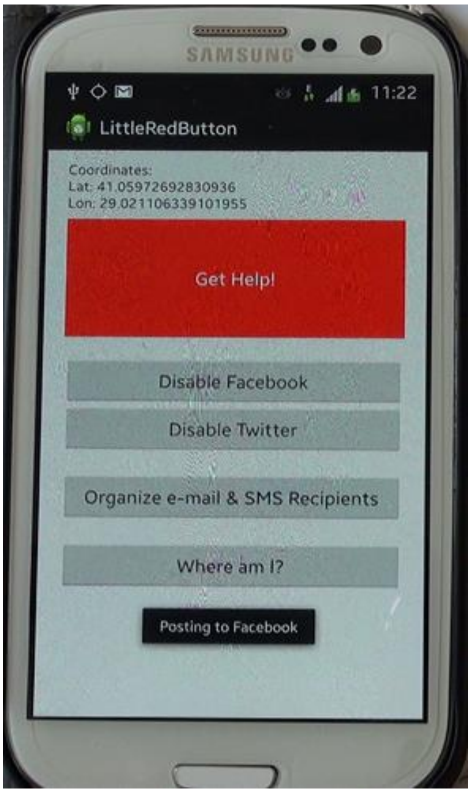
5.7 Emergency Declaration Screen

When the user declares an emergency situation, LittleRedButton shall immediately take a picture of the scene and attach it to the emergency message to be sent. After that the message shall be sent through selected channels. The following screens shows the flow of emergency declaration process and the resulting recipient’s PC and phone screen shots. Note that the photos attached to messages are Android’s emulated camera’s artifacts. On the real device, the real camera display shall be used.



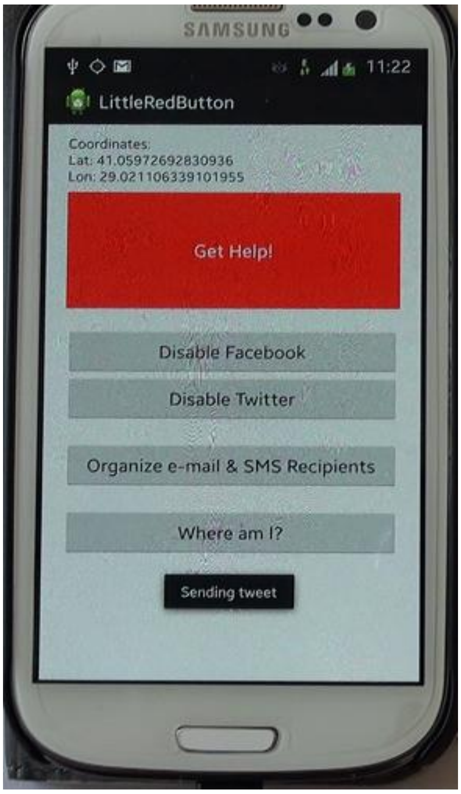
Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Picture has been taken!



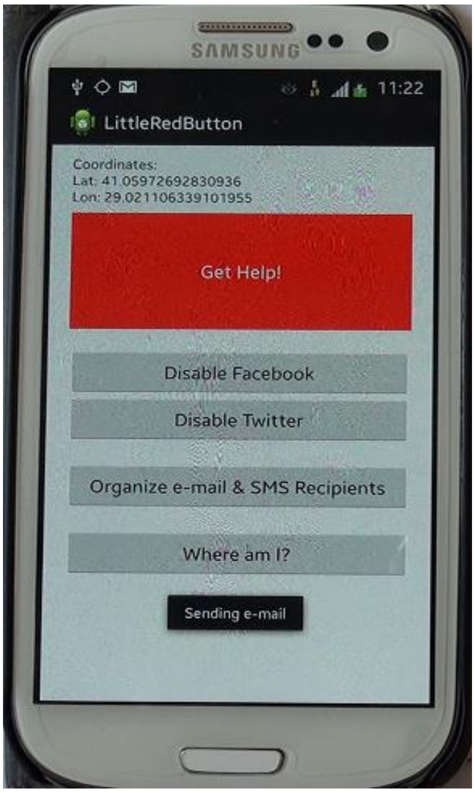
Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Posting to Facebook



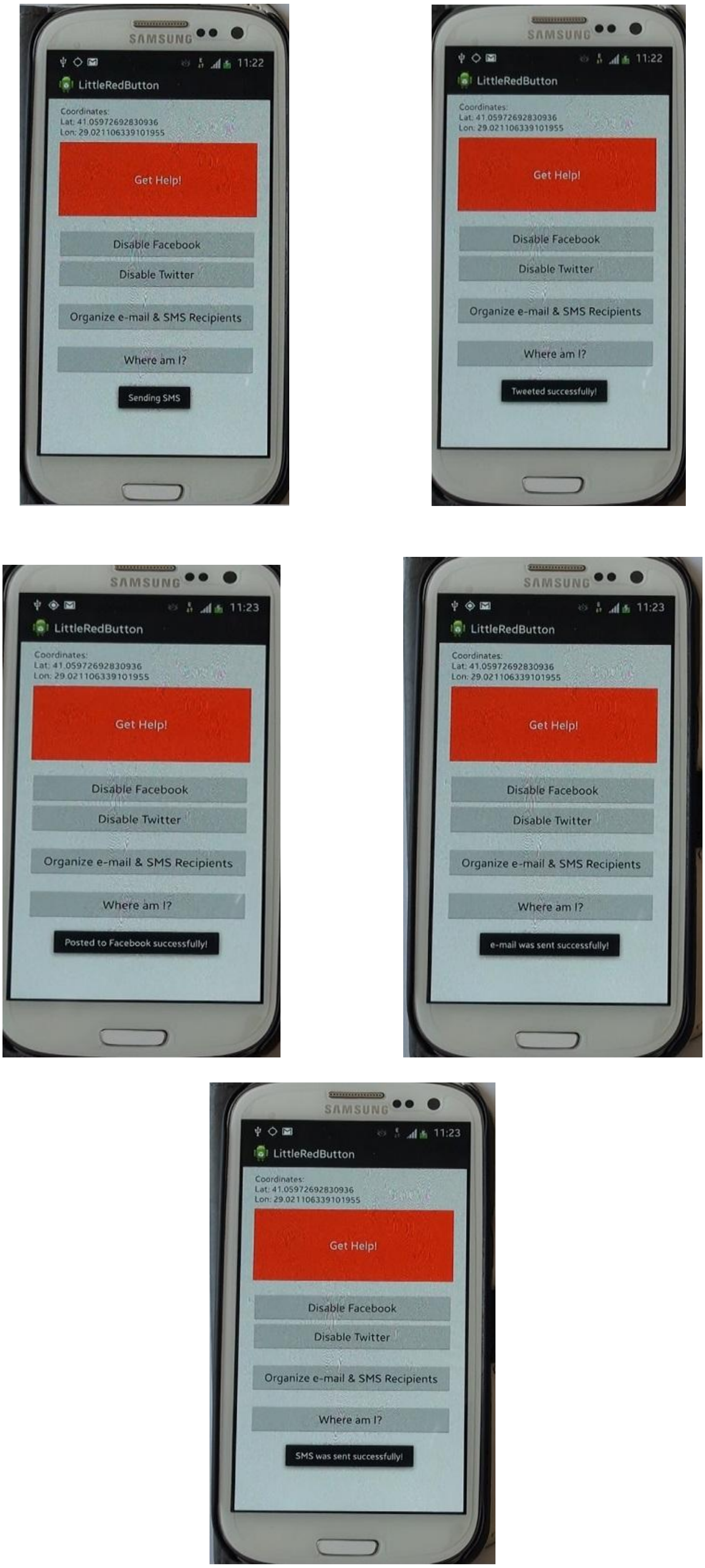
Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Sending tweet



Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Sending e-mail



Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Sending SMS

Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Tweeted successfully!

Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

Posted to Facebook successfully!

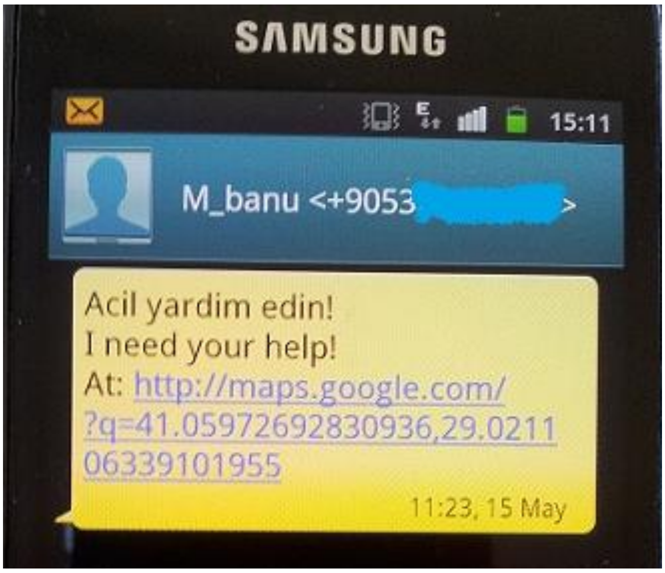
Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

e-mail was sent successfully!

Coordinates:
Lat: 41.05972692830936
Lon: 29.021106339101955

SMS was sent successfully!

Figure 26: Emergency Declaration Workflow

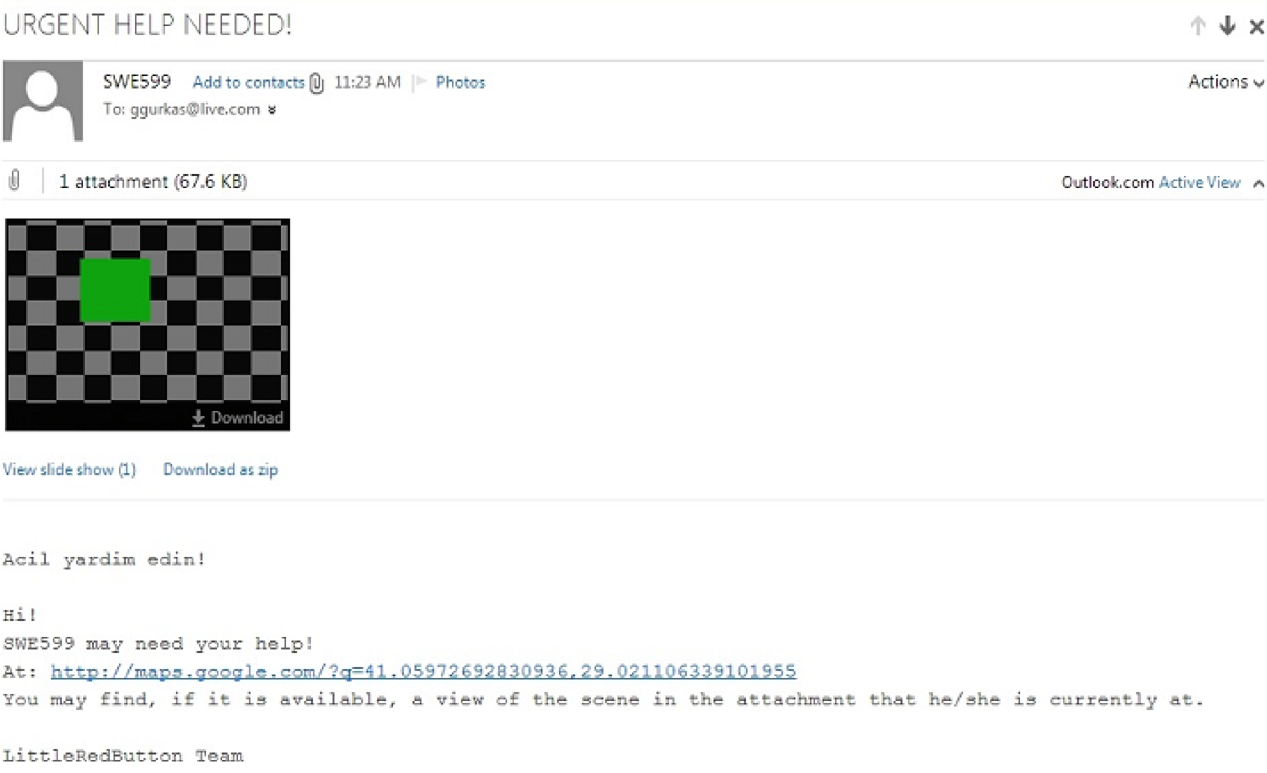


M_banu <+9053[redacted]>

Acil yardim edin!
I need your help!
At: http://maps.google.com/?q=41.05972692830936,29.021106339101955

11:23, 15 May

Figure 27: Recipient's SMS Inbox



URGENT HELP NEEDED!

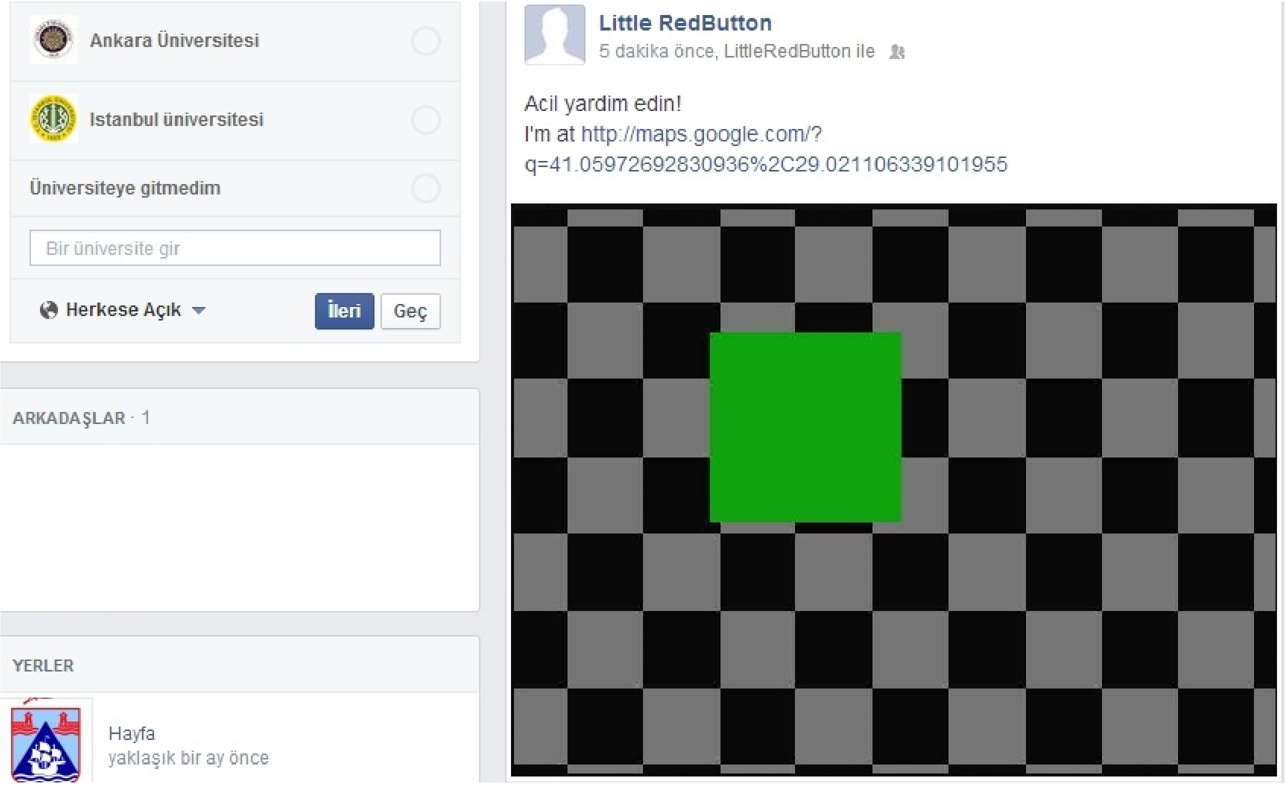
SWE599 Add to contacts 11:23 AM Photos
To: ggurkas@live.com

```
Acil yardim edin!

Hi!
SWE599 may need your help!
At: http://maps.google.com/?q=41.05972692830936,29.021106339101955
You may find, if it is available, a view of the scene in the attachment that he/she is currently at.

LittleRedButton Team
```

Figure 28: Recipient's Mail Inbox

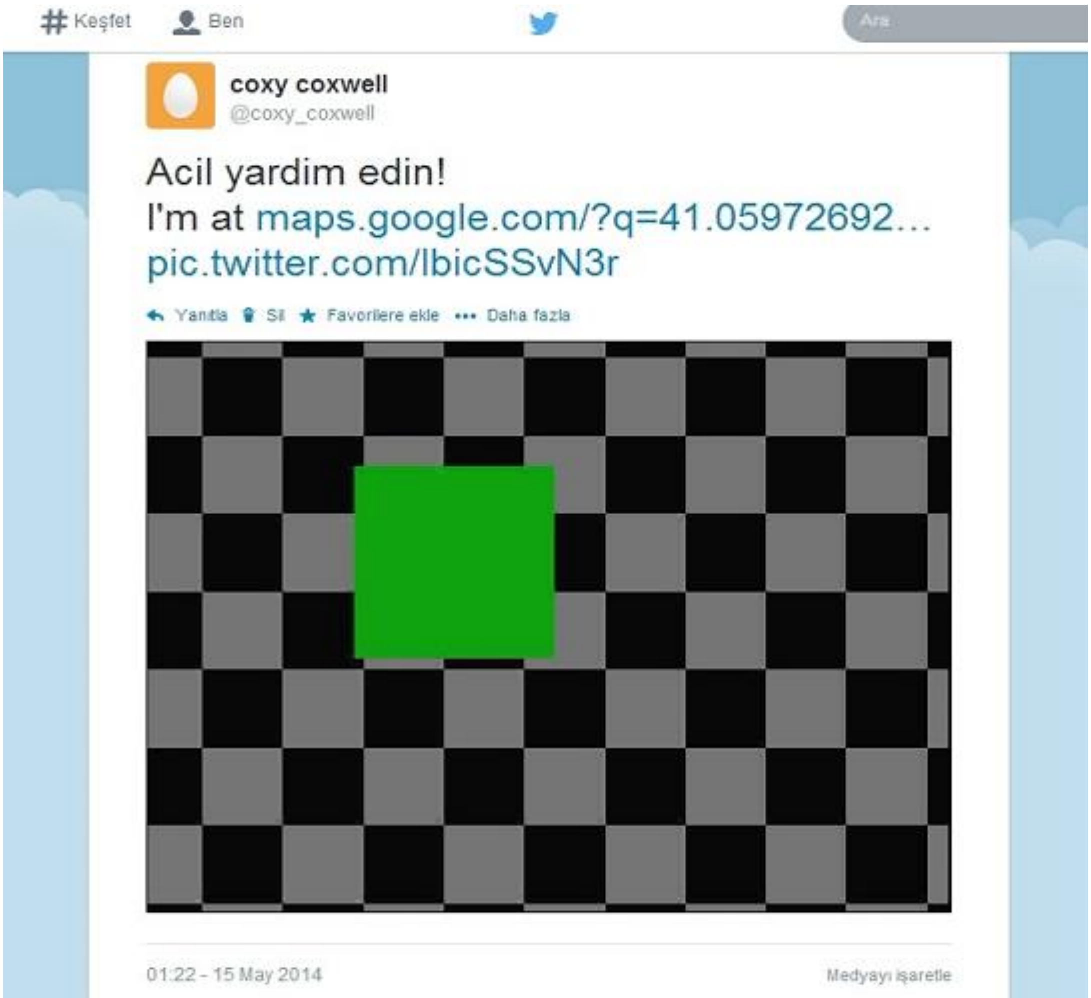


Little RedButton

5 dakika önce, LittleRedButton ile

Acil yardim edin!
I'm at http://maps.google.com/?q=41.05972692830936%2C29.021106339101955

Figure 29: Recipient's Facebook Wall



coxy coxwell
@coxy_coxwell

Acil yardim edin!
I'm at maps.google.com/?q=41.05972692…
pic.twitter.com/IbicSSvN3r

01:22 - 15 May 2014

Figure 30: Recipient's Twitter Timeline

6. Conclusions and Future Work

In conclusion, the LittleRedButton does what it was intended to do in the first place. It was a great experience for me to work with a different platform, Android, various frameworks, libraries and mobile application APIs in the same project with a knowledge level from scratch.

In the future work, for enhancement purposes, LittleRedButton may have features such as;

- Recording for a defined length of audio/video data and send them in attachment as well.
- Calling the emergency services such as 112, 155 etc. by a text message or by voice.
- Displaying the shortest route to the selected nearest place on Google Maps.

7. References

1. LRB Requirements Specification Document, Revision 1.0, in file Gurkas-MustafaGoksu-LittleRedButton-RSD-2014-03-09-Rev-1.0.pdf
2. Eclipse ADT Plugin (http://developer.android.com/tools/sdk/eclipse-adt.html)
3. Android SDK (http://developer.android.com/sdk/index.html?hl=sk)
4. Android SDK (http://en.wikipedia.org/wiki/Android_SDK#SDK)
5. Android Building and Running (http://developer.android.com/tools/building/index.html)
6. Android SDK Workflow (http://developer.android.com/tools/workflow/index.html)
7. Android Emulator (http://developer.android.com/tools/devices/emulator.html)
8. SQLite (http://en.wikipedia.org/wiki/SQLite)
9. Google Maps API v2 (https://developers.google.com/maps/documentation/android/start#getting_the_google_maps_android_api_v2)
10. Android Facebook API (http://www.ibm.com/developerworks/library/x-androidfacebookapi/)