1. INTRODUCTION

Android has security features built into the operating system that significantly reduce the frequency and impact of application security issues. The system is designed so you can typically build your apps with default system and file permissions and avoid difficult decisions about security. The most common security concern for an application on Android is whether the data that you save on the device is accessible to other apps. It is important that you be familiar with the Android security best practices in this document. Following these practices as general coding habits will reduce the likelihood of inadvertently introducing security issues that adversely affect your user.

Android is a modern mobile platform that was designed to be truly open. Android applications make use of advanced hardware and software, as well as local and served data, exposed through the platform to bring innovation and value to consumers. To protect that value, the platform must offer an application environment that ensures the security of users, data, applications, the device, and the network.

Securing an open platform requires robust security architecture and rigorous security programs. Android was designed with multi-layered security that provides the flexibility required for an open platform, while providing protection for all users of the platform.

Android was designed with developers in mind. Security controls were designed to reduce the burden on developers. Security-savvy developers can easily work with and rely on flexible security controls. Developers less familiar with security will be protected by safe defaults.

Gotya is an android application which is mainly based on security and systems. Gotya takes a silent snapshot through the front facing camera of the criminal who is attempting to use your device, whenever the screen lock is entered incorrectly. Then, Gotya gets the location of your device, forms a Google maps link, and then send it along with the picture to your email. Gotya is the best anti-theft solution you need for your device. Gotya help as to set our mobile phone secure, means no unknown person can ever access our personal data in our smart phone. Losing one’s smartphone is somewhat of a catastrophe, and that’s putting it mildly. The fact that our phones have become so smart can work against us, in the sense that we tend to store an enormous amount of important personal information on them, or else they can be used to access personal information. We are talking accessing emails, calendars, contacts, you name it—and that’s not even the scariest of scenarios.

There are quite a few apps floating around that were developed to help the rightful owner in worst-case scenario type situations. Gotya is an example of such an app, but with a unique feature: it will take a photo of the thief (or person who finds your lost phone)! You can also choose under which circumstances a photo should be taken with the device’s front facing camera: every time the display is activated or only when an unsuccessful attempt to unlock the screen has been made.

1. PROBLEM DEFINITION/OBJECTIVES

Existed system is less user friendly. They only locate/track your ‘lost’ phone on a map through our web-based mobile phone tracking feature. So they only give idea about where our phone is. The proposed system help us not only to track the phone but also give an instant idea about who stolen our phone with a photograph.

Proposed system not only help to find who took our phone, but also help as to keep our mobile phone secure from unknown persons. It help us to keep our personal data away from unknown persons. Proposed system will be a great security application for smart phones.

Gotya will be a great security application which help as to find which unknown person took our smart phone. Gotya will be an android application. Which will run in an android OS. This application will work when someone enter a wrong swipe in any gotya installed smart phone. Gotya will take a silent photo of the person and will access the phone location and gotya will mail this two things to our e-mail id. Gotya also helps us to erase all our personal data from the phone when some thief try to access our smart phone. It also helps us to encrypt our personal data in our phone. Gotya help us to set a strong password or a strong swipe pattern for our smart phone. It will also provide password expire technology. The minimum password requirements like valid characters, numbers, symbols, non-letters can be dynamically entered by the user. To provide more security a new feature like password history depth is included in this application. A user who is not having a mail id can also have access to the system as the system provides an e-mail facility in which a user can create a mail id.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1. INTRODUCTION

**3.1.1 Purpose of this Document**

This SRS describes the function and the performance allocated to our product. It provides a reference for the validation of the final product. SRS provides an overview of the product including functional and nonfunctional requirements, abbreviations used, product and functions etc.

**3.1.2 Scope of the Development Project**

The name of our product is “GoTyA”. This application is mainly based on security and systems. Provides better protection for the smartphones. User gets a solid security solution for his device. Gotya is the best anti-theft solution need for user’s device.

**3.1.3Overview of Document**

This document provides a description of the requirements of the product. Section 2 of the SRS gives detailed description of the product including the data requirements. Section 3 provides specific functional requirements of the different components of the product and the performance criteria. Section 4 provides idea about the software and hardware requirements.

3.2 GENERAL DESCRIPTION

**3.2.1User Characteristics**

The user may be the owner of the device. The owner can view the trapped face details. Owner has the provision to customize the settings such as enabling or disabling the application, encryption/decryption, email settings, password characteristics, etc.

**3.2.2 Product Perspective**

The software on been installed can be implemented on any android system or a device which has a front camera.

**3.2.3General Constraints, Assumptions, Dependencies and Guidelines**

The system will work only if there is a properly configured network connection. The language must be English. First-off, it’s the infrastructure, which includes the type of the internet connection (Wi-Fi, 2G or 3G), the transfer bandwidth as well as the storage capacity of our (client and server) databases that determine transmission speed and the amount of data we obtain. Another crucial aspect is the resource reservoir. Battery consumption, computing power and financial restrictions.

3.3 SPECIAL REQUIREMENTS

**3.3.1 User Interface**

The user can interact with the system using graphical user interface. Using the screen, user can customize all settings the application.

**3.3.2 Detailed Description of Functional Requirements**

This section provides a requirement overview of the product. The project will be developed on java, and will run in an android operating system.

**3.3.3 Functional Requirements**

* Login only for valid user.
* Owner can change the password after login to application.
* Owner can customize the application settings.
* View all the trapped faces.

**3.3.4 User Input Validation**

If the user leaves a mandatory field blank, he will be prompted to enter valid data in that particular field.

**3.3.5 Performance Requirement**

The performance of our product is at its best if accessed via internet.

3.4 REQUIREMENTS

**3.4.1 Software Requirements**

Operating system : Android

Platform : Eclipse

Client Side : Android

Server Side : JSP

**3.4.2 Hardware Requirements**

Processor : 4.1.6 GHz or faster processor

Memory size : 1GB RAM

Storage : 10 GB Hard Disk

Keyboard : Virtual keyboard with 102 keys

Device : Android Mobile

4. SYSTEM STUDY/REVIEW OF LITERATURE

4.1 SURVEY OF LITERATURE/ EXISTING SYSTEM

The available anti-theft applications will only trace the location of our lost phone, and inform as through message. They will find your phone on a map through geolocation using both GPS and Wi-Fi triangulation. In all the existing systems no one provides an email face trapping. And no one provides encryption/decryption on data. If all the above services are provided, there is no room for wiping. Existing applications provides only the default phone settings for setting a password. So the user cannot define his own password by minimum requirements like minimum numbers of alphabets, numeric, symbols, and non-letters. Existing applications do not provides password quality, protection, etc. The user can set the maximum depth of the password. So that if the maximum exceeds, the data are wiped from the device. Existing applications do not provide camera options. So the existing application cannot provide better security services.

4.2 E-R DIAGRAM

4.3 PROPOSED SYSTEM

Proposed system not only help to find who took our phone, but also help as to keep our mobile phone secure from unknown persons. It help us to keep our personal data away from unknown persons. Proposed system will be a great security application for smart phones.

Gotya will be a great security application which help as to find which unknown person took our smart phone. Gotya will be an android application. Which will run in an android OS. This application will work when someone enter a wrong swipe in any gotya installed smart phone. Gotya will take a silent photo of the person and will access the phone location and gotya will mail this two things to our e-mail id. Gotya also helps us to erase all our personal data from the phone when some thief try to access our smart phone. It also helps us to encrypt our personal data in our phone. Gotya help us to set a strong password or a strong swipe pattern for our smart phone. It will also provide password expire technology. The minimum password requirements like valid characters, numbers, symbols, non-letters can be dynamically entered by the user. To provide more security a new feature like password history depth is included in this application. A user who is not having a mail id can also have access to the system as the system provides an e-mail facility in which a user can create a mail id.

4.4 FEASIBILITY STUDY

The main objective of this study is to determine whether the proposed system is feasible or not. Mainly there are three types of feasibility study to which the proposed system is subjected as described below.

Three key considerations involved in the feasibility are:

 Economic Feasibility  Technical Feasibility  Behavioural Feasibility. The proposed system must be evaluated from a technical viewpoint first, and if technically feasible, their impact on the organization must be assessed. If compatible, the behavioural system can be devised. Then those must be tested for economic feasibility.

4.4.1 ECONOMIC FEASIBILITY

Economic analysis is used for evaluating the cost effectiveness of a proposed system. In an existing system people go to a retailer and assemble. But in proposed system, provides a single shop for assembling. This will make the user more comfortable.

4.4.2 TECHNICAL FEASIBITITY

The technical requirements of the proposed system are highly affordable. There is no difficulty in migrating from the existing system to the proposed system. Since J2EE is platform independent the application will run smoothly on any kernel without causing much trouble to the end user.

4.4.3 BEHAVIOURAL FEASIBILITY

The system is intended for use by web administrator and user. Since the software is simple and provides good Graphical User Interface, it provides high level user convenience.

5. SYSTEM DESIGN

5.1 MODULES

There are 7 modules in this project.

**5.1.1 General**

* Enable/disable the application.
* Enable/disable device cameras.

**5.1.2 Password Quality**

* Set password for the application.
* Setting password quality
* Unspecified
* Something
* Numeric
* Alphabetic
* Alphanumeric
* Complex

**5.1.3 Password Expiration**

* Setting screen lock password
* Password expiration timeout

**5.1.4 Lock Screen**

* Setting screen lock
* Maximum time to screen lock

**5.1.5 Wipe**

* Maximum password failures for local wipe
* Confirming the erasing of data.

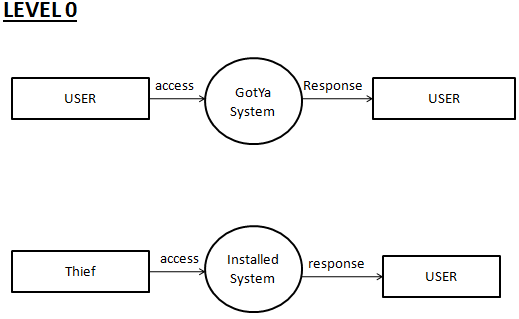
**5.1.6 Encryption**

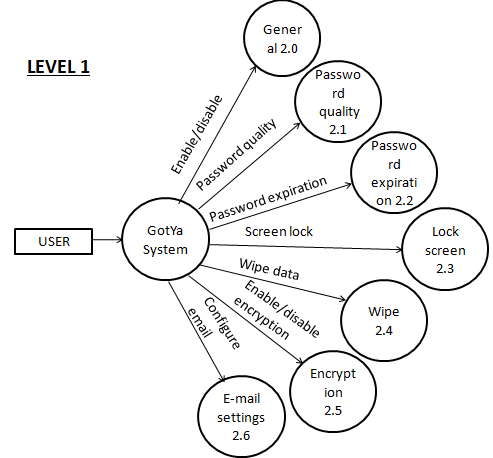
* Enabling encryption.
* Activating Encryption (if the device supports).

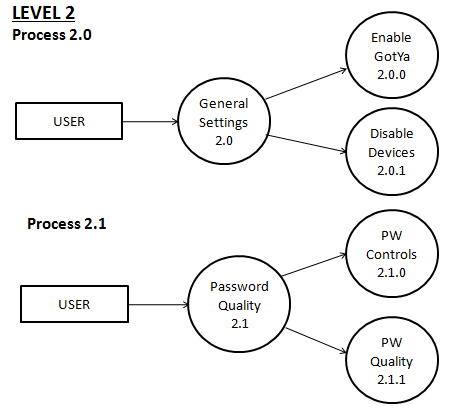
**5.1.7 Email Settings**

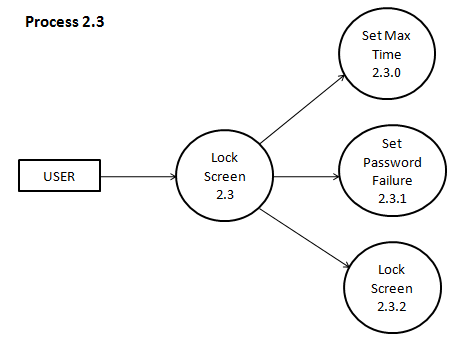
* Enable sending images via Gmail.
* Email address which the image have to be sent.
* Creating Gmail account for the user.
* Username and password of owner.

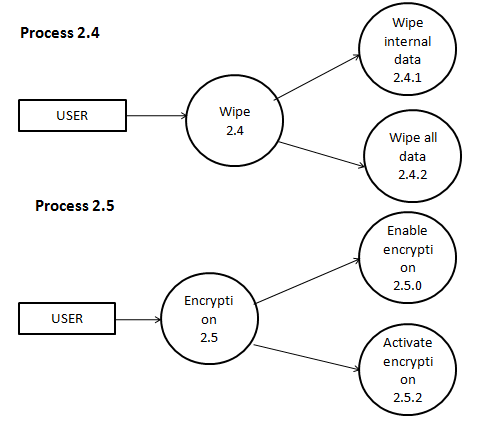
5.2 DATA FLOW DIAGRAM

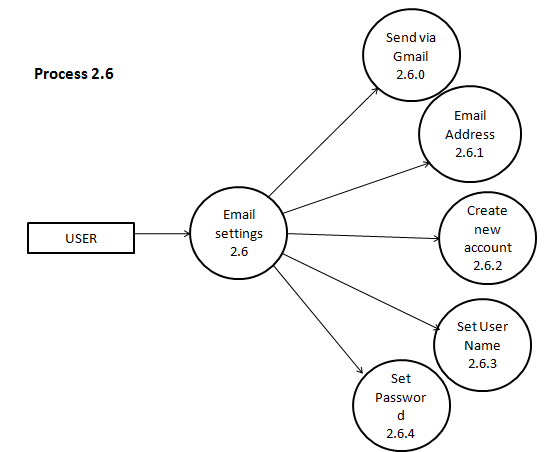


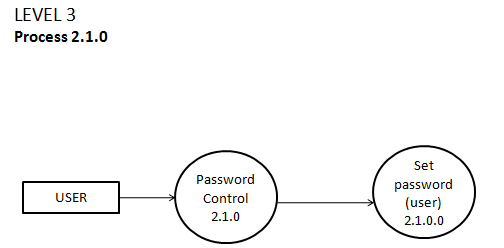


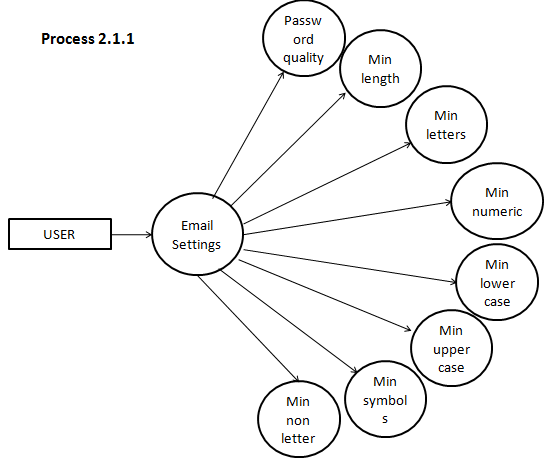


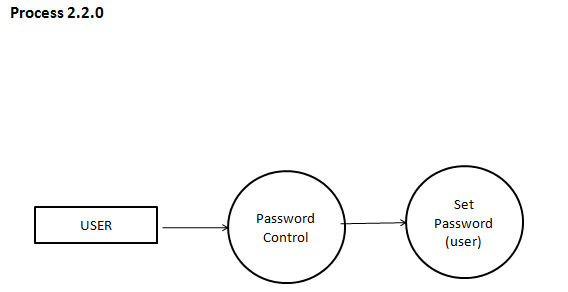


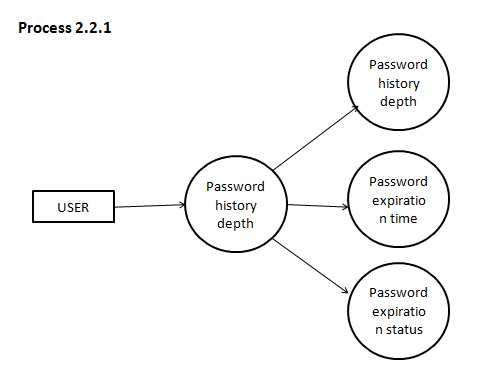












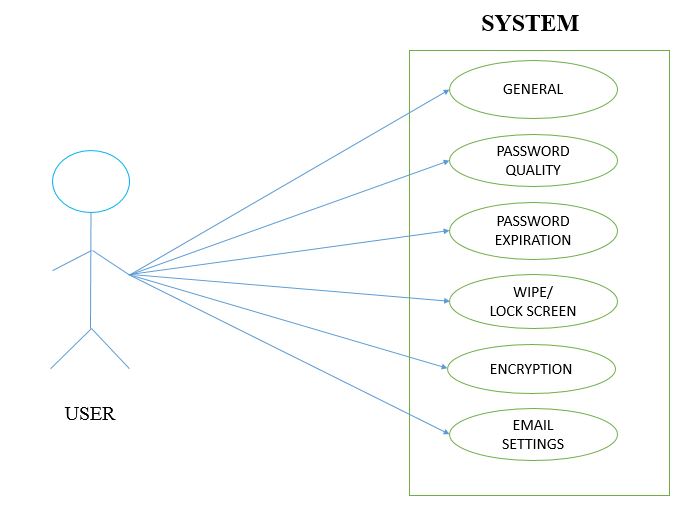
6. USE CASE DESIGN

Use case is a set of scenarios that describing an interaction between a user and a system. The two main component of use case diagram are use cases and actors. They are helpful in exposing requirements and planning the projects. During the initial stage of project, most use cases should be defined. But as the project continues more might become visible, the use case diagram for showing the project is that it measures the size of the software. It reduces users need for training and operating the system.

In use case diagram, we all know, the main two things are the actors and the actions. Here the actors is the user. Apart from the actors there are actions to be performed by these actors. These actions are shown in oval.

Use case diagrams are relatively easy UML diagram. They are helpful in exposing requirements and planning of the project. During the initial stage of a project most use cases should be defined, but as the project continues more might become visible. A use case is an external view of the system that represents some action the user might perform in order to complete a task. A use case diagram displays the relationship among actors and use cases.

6.1 UML DIAGRAM



7. DETAILED DESIGN

7.1 DESIGN

Design of the system includes mainly two steps: System design and detailed design. In system design a structural framework for the entire system is created. It is done in such a way that related part come under particular groups. Thus after the system design, a network of different groups is obtained. It is the high-level strategy for solving the problem and building a solution. It includes the decision about the organization of the system into subsystems, the allocation of subsystems to hardware and software components, and major conceptual and policy decisions that form the framework for the detailed design. System design is the first stage in which the basic approach to solving problem is selected. During system design, the overall style and structure are decided.

In detailed design, each group is studied in detail and the internal operations are decided. Based on this, the data structures and the programming language to be used are decided. Apart from detailed design, the system design can be grouped into physical design and structural design. The physical design maps out the details of the physical system and plans the system implementation and specifies the hardware and software requirements. Structured design is an attempt to minimize the complexity and make a problem manageable by subdividing into smaller segments, which is called modularization or decomposition.

7.2 SPECIFICATION

The following are the specifications that are used to develop the proposed system: based upon the levels of the product, the project had been divided to 7 modules which are as follows.

MODULE 1: GENERAL

The user can enable gotya here in this module. the application should be enabled so as to get activated in the phone else the application will be activated but cannot be used.only after enabling the application the gotya services will be visible.the user can enable/disable the front camera of the Smartphone.

MODULE 2: PASSWORD QUALITY

Using this module the user can set his password. Input options like, length, letters, numeric, lowercase, uppercase, symbols, non letter are user defined. The minimum password requirement is 4.the user can externally edit the minimum limit of the requirement. Normally to change the already registered password the user have to confirm the password a two step verification is required but using API’s we can set the pin for the system with which the further entries to the system will be entertained.

MODULE 3: PASSWORD EXPIRATION

Using this module the user can set the password expiration time. the user can also check the status of the password whether expired or not. Another feature is password history depth. to provide accurate security a set of last used passwords say 2 cannot be used again.

MODULE 4: LOCK SCREEN

Using this module the user can set the minimum time limit in which the phone gets locked.

MODULE 5: WIPE

To provide advance security feature a new feature called wipe. Once the thief enters a wrong input pattern till the specified limit the phone’s internal datas will be wiped off.to wipe the external datas the application have the privilege.

MODULE 6: ENCRYPTION

Using this module we can enable/disable the encryption services only if the phone provides this service. Android disk encryption is based on dm-crypt, which is a kernel feature that works at the block device layer. Because of this, encryption works with Embedded Multi Media Card **(**eMMC) and similar flash devices that present themselves to the kernel as block devices. Encryption is not possible with YAFFS, which talks directly to a raw NAND flash chip.

The encryption algorithm is 128 AdvanceICE.d Encryption Standard (AES) with cipher-block chaining (CBC) and ESSIV:SHA256. The master key is encrypted with 128-bit AES via calls to the OpenSSL library. You must use 128 bits or more for the key (with 256 being optional). OEMs can use 128-bit or higher to encrypt the master key.

In the Android 5.0 release, there are four kinds of encryption states:

* Default
* Pin
* Password
* Pattern

When the user sets the PIN/pass or password on the device, only the 128-bit key is re-encrypted and stored. Encryption is managed by init and vold. init calls vold, and vold sets properties to trigger events in init. Other parts of the system also look at the properties to conduct tasks such as report status, ask for a password, or prompt to factory reset in the case of a fatal error. To invoke encryption features in vold, the system uses the command line tool vdc’s cryptfs commands: checkpw, restart, enablecrypto, changepw, cryptocomplete, verifypw, setfield, getfield, mountdefaultencrypted, getpwtype, getpw, and clearpw.

MODULE 7: EMAIL SETTINGS

Using this module the user can create his/her own E-mail id and password. He/she can enable/disable the E-mail settings. the user should create a new G-mail account and then should configure the E-mail address to which the photo should be send. With the existing gmail account the user cannot use the applications.

8. INPUT/OUTPUT DESIGN

When a thief or an unknown user tries to access the system without the prior permission while he accessing the system with a wrong password or a wrong pin the application is enabled and the picture of the thief is taken and with the help of the GPS technology the picture of the thief with the latitude and longitude is mailed to the configured E-mail address which is configured by the user while he enters the system. To provide more security the password quality contains many features like minimum lowercase, minimum uppercase, minimum numeric and so. Initially the OS provides the minimum requirement of 4 while it can be modified according to the user will and wish.

9. SYSTEM IMPLEMENTATION

9.1 system requirements

It is an android application. It will run on any android device. The device should contain a front camera. The device should also support USB debugging. Other than the android devices an android emulator can also be used to run the application. Android virtual device(AVD) is used for this purpose. The hardware requirements for the system is an android mobile phone with a front camera and the minimum processor requirement is 4.1.6 GHz or faster processor and memory size of 1GB RAM and an storage size of 10 GB with a virtual keyboard.

9.2 Implementation scenario

The application is developed in API 19 JELLYBEAN 4.2.2. So the versions above API’s 19 can run the application. The current version of gotya is 1.0.

9.3 Developing environment

The application is developed in eclipse platform. Language used is java. The operating system used is android. The client side should be an android device. The server side should be java server page (JSP).

9.4 Familiarization with tools

9.4.1.1. Android

**Android** is a [mobile operating system](http://en.wikipedia.org/wiki/Mobile_operating_system) (OS) based on the [Linux kernel](http://en.wikipedia.org/wiki/Linux_kernel) and currently developed by [Google](http://en.wikipedia.org/wiki/Google). With a [user interface](http://en.wikipedia.org/wiki/User_interface) based on [direct manipulation](http://en.wikipedia.org/wiki/Direct_manipulation_interface), Android is designed primarily for [touchscreen](http://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](http://en.wikipedia.org/wiki/Smartphone) and [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer), with specialized user interfaces for televisions ([Android TV](http://en.wikipedia.org/wiki/Android_TV)), cars ([Android Auto](http://en.wikipedia.org/wiki/Android_Auto)), and wrist watches ([Android Wear](http://en.wikipedia.org/wiki/Android_Wear)). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a [virtual keyboard](http://en.wikipedia.org/wiki/Virtual_keyboard). Despite being primarily designed for touchscreen input, it also has been used in [game consoles](http://en.wikipedia.org/wiki/Video_game_console), [digital cameras](http://en.wikipedia.org/wiki/Digital_camera), and other electronics.

Android is the most popular mobile OS. As of 2013, Android devices sell more than [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [iOS](http://en.wikipedia.org/wiki/IOS), and [Mac OS](http://en.wikipedia.org/wiki/Mac_OS) devices combined with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013 the [Google Play](http://en.wikipedia.org/wiki/Google_Play) store has had over 1 million Android apps published, and over 50 billion apps downloaded A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for Android. At [Google I/O](http://en.wikipedia.org/wiki/Google_I/O) 2014, the company revealed that there were over 1 billion active monthly Android users (that have been active for 30 days), up from 538 million in June 2013.

9.4.1.2 Eclipse

In [computer programming](http://en.wikipedia.org/wiki/Computer_programming), **Eclipse** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE). It contains a base [workspace](http://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Written mostly in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other [programming languages](http://en.wikipedia.org/wiki/Programming_language): [Ada](http://en.wikipedia.org/wiki/Ada_(programming_language)), [ABAP](http://en.wikipedia.org/wiki/ABAP),[C](http://en.wikipedia.org/wiki/C_(programming_language)), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [COBOL](http://en.wikipedia.org/wiki/COBOL), [Fortran](http://en.wikipedia.org/wiki/Fortran), [Haskell](http://en.wikipedia.org/wiki/Haskell_(programming_language)), [JavaScript](http://en.wikipedia.org/wiki/JavaScript), [Lasso](http://en.wikipedia.org/wiki/Lasso_(programming_language)), [Natural](http://en.wikipedia.org/wiki/NATURAL), [Perl](http://en.wikipedia.org/wiki/Perl), [PHP](http://en.wikipedia.org/wiki/PHP), [Prolog](http://en.wikipedia.org/wiki/Prolog), [Python](http://en.wikipedia.org/wiki/Python_(programming_language)), [R](http://en.wikipedia.org/wiki/R_(programming_language)), [Ruby](http://en.wikipedia.org/wiki/Ruby_(programming_language)) (including Rails framework), [Scala](http://en.wikipedia.org/wiki/Scala_(programming_language)), [Clojure](http://en.wikipedia.org/wiki/Clojure" \o "Clojure), [Groovy](http://en.wikipedia.org/wiki/Groovy_(programming_language)), [Scheme](http://en.wikipedia.org/wiki/Scheme_(programming_language)), and [Erlang](http://en.wikipedia.org/wiki/Erlang_(programming_language)" \o "Erlang (programming language)). It can also be used to develop packages for the software[Mathematica](http://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial [codebase](http://en.wikipedia.org/wiki/Codebase) originated from [IBM Visual Age](http://en.wikipedia.org/wiki/IBM_VisualAge). The Eclipse [software development kit](http://en.wikipedia.org/wiki/Software_development_kit) (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

9.4.1.3. GPS

The **Global Positioning System** (**GPS**) is a space-based [satellite navigation](http://en.wikipedia.org/wiki/Satellite_navigation) system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The system provides critical capabilities to military, civil and commercial users around the world. It is maintained by the United States government and is freely accessible to anyone with a [GPS receiver](http://en.wikipedia.org/wiki/GPS_receiver).

The GPS project was developed in 1973 to overcome the limitations of previous navigation systems,integrating ideas from several predecessors, including a number of classified engineering design studies from the 1960s. GPS was created and realized by the [U.S. Department of Defense](http://en.wikipedia.org/wiki/U.S._Department_of_Defense) (DOD) and was originally run with 24 satellites. It became fully operational in 1995. [Bradford Parkinson](http://en.wikipedia.org/wiki/Bradford_Parkinson), [Roger L. Easton](http://en.wikipedia.org/wiki/Roger_L._Easton), and [Ivan A. Getting](http://en.wikipedia.org/wiki/Ivan_A._Getting) are credited with inventing it.

Advances in technology and new demands on the existing system have now led to efforts to modernize the GPS system and implement the next generation of [GPS III](http://en.wikipedia.org/wiki/GPS_III) satellites and Next Generation Operational Control System (OCX) Announcements from Vice President [Al Gore](http://en.wikipedia.org/wiki/Al_Gore) and the [White House](http://en.wikipedia.org/wiki/Clinton_Administration) in 1998 initiated these changes. In 2000, the [U.S. Congress](http://en.wikipedia.org/wiki/U.S._Congress) authorized the modernization effort, GPS III.

10.SOURCE CODE

**10.1 photo capture**

package com.camera.antitheft.services;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import javax.mail.AuthenticationFailedException;

import android.app.ListActivity;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.app.Service;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.SharedPreferences.Editor;

import android.graphics.PixelFormat;

import android.hardware.Camera;

import android.hardware.Camera.CameraInfo;

import android.hardware.Camera.PictureCallback;

import android.hardware.Camera.ShutterCallback;

import android.net.Uri;

import android.os.Bundle;

import android.os.Environment;

import android.os.IBinder;

import android.preference.PreferenceManager;

import android.util.Log;

import android.view.Gravity;

import android.view.SurfaceHolder;

import android.view.SurfaceView;

import android.view.WindowManager;

import android.view.WindowManager.LayoutParams;

import android.widget.Toast;

import com.dieutek.antitheft.AntiTheftAdmin.EmailSettingsFragment;

import com.dieutek.antitheft.R;

import com.dieutek.antitheft.ResultActivity;

import com.dieutek.antitheft.utils.GMailSender;

import com.dieutek.antitheft.utils.Utility;

public class BackgroundImageCapture extends Service implements

SurfaceHolder.Callback {

public static final String TAG = BackgroundImageCapture.class

.getSimpleName();

// public static String IMG\_URI = null;

private WindowManager mWindowManager;

private SurfaceView mSurfaceView;

private Camera mCamera = null;

private int cameraId = 0;

Context context;

static Bundle bundle;

public static Uri fileUri;

SharedPreferences sp;

Editor e;

@Override

public void onCreate() {

sp=PreferenceManager.getDefaultSharedPreferences(this);

e=sp.edit();

Intent resultIntent = new Intent(this, ResultActivity.class);

// Intent resultIntent = new Intent(this, AntiTheftAdmin.class);

// Log.d(TAG, "Image URI " + IMG\_URI);

// putParcelable("file\_uri", IMG\_URI);

// resultIntent.putExtra("imageUri", bundle);

// Because clicking the notification opens a new ("special") activity,

// there's no need to create an artificial back stack.

PendingIntent resultPendingIntent = PendingIntent.getActivity(this, 0,

resultIntent, PendingIntent.FLAG\_UPDATE\_CURRENT);

// Start foreground service to avoid unexpected kill

Notification notification = new Notification.Builder(this)

.setContentIntent(resultPendingIntent)

.setContentTitle("GotYa! Anti-Theft")

.setContentText("Your Photo has been taken by the System")

.setSmallIcon(R.drawable.ic\_launcher).build();

NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

// hide the notification after its selected

notification.flags |= Notification.FLAG\_AUTO\_CANCEL;

notificationManager.notify(1234, notification);

// startForeground(1234, notification);

// Create new SurfaceView, set its size to 1x1, move it to the top left

// corner and set this service as a callback

mWindowManager = (WindowManager) this

.getSystemService(Context.WINDOW\_SERVICE);

mSurfaceView = new SurfaceView(this);

LayoutParams layoutParams = new WindowManager.LayoutParams(1, 1,

WindowManager.LayoutParams.TYPE\_SYSTEM\_OVERLAY,

WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH,

PixelFormat.TRANSLUCENT);

layoutParams.gravity = Gravity.LEFT | Gravity.TOP;

mWindowManager.addView(mSurfaceView, layoutParams);

mSurfaceView.getHolder().addCallback(this);

}

@Override

public void surfaceCreated(SurfaceHolder holder) {

cameraId = findFrontFacingCamera();

Log.d("CAMERA", "Camera ID " + cameraId);

// mCamera.enableShutterSound(false);

try {

mCamera = Camera.open(cameraId);

if (mCamera != null) {

// mCamera.setPreviewDisplay(holder);

mCamera.setPreviewDisplay(mSurfaceView.getHolder());

mCamera.startPreview();

mCamera.takePicture(shutterCallback, rawPictureCallback,

jpegPictureCallback);

// mCamera.setOneShotPreviewCallback(null);

}

} catch (RuntimeException e) {

Log.e("CAMERA", "Camera currently unavailable");

e.printStackTrace();

} catch (IOException e) {

Log.e("surfaceCallback",

"Exception in setPreviewDisplay()", e);

e.printStackTrace();

} finally {

//mCamera.release();

}

}

ShutterCallback shutterCallback = new ShutterCallback() {

@Override

public void onShutter() {

Log.d(TAG, "onShutter'd");

// if(Camera.CameraInfo.CAMERA\_FACING\_FRONT);

}

};

/\*\* Handles data for raw picture \*/

PictureCallback rawPictureCallback = new PictureCallback() {

@Override

public void onPictureTaken(byte[] data, Camera camera) {

Log.d(TAG, "onPictureTaken - raw");

}

};

/\*\* Handles data for jpeg picture \*/

PictureCallback jpegPictureCallback = new PictureCallback() {

@Override

public void onPictureTaken(byte[] data, Camera camera) {

FileOutputStream outStream = null;

try {

File mediaFile = getOutputMediaFile();

// outStream = new FileOutputStream(getOutputMediaFile());

outStream = new FileOutputStream(mediaFile);

Log.d(TAG,

"PATH :"

+ Environment

.getExternalStoragePublicDirectory(

Environment.DIRECTORY\_PICTURES)

.toString());

outStream.write(data);

outStream.close();

Log.d(TAG, "onPictureTaken - wrote bytes: " + data.length);

fileUri = getOutputMediaFileUri();

Log.d(TAG, fileUri.toString());

// modified May 3 2015

sendEmailViaGMail(mediaFile);

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} finally {

// camera.release();

}

Log.d(TAG, "onPictureTaken - jpeg");

/\*

\* File mediaFile = new File(fileUri.getPath());

\* sendEmailViaGMail(mediaFile);

\*/

}

};

@Override

public void surfaceChanged(SurfaceHolder holder, int format, int width,

int height) {

}

@Override

public void surfaceDestroyed(SurfaceHolder holder) {

}

**10.2 GPS tracker**

package com.dieutek.antitheft.services;

import android.app.AlertDialog;

import android.app.Service;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.os.IBinder;

import android.provider.Settings;

import android.util.Log;

public class GPSTracker extends Service implements LocationListener {

private final Context mContext;

// flag for GPS status

boolean isGPSEnabled = false;

// flag for network status

boolean isNetworkEnabled = false;

// flag for GPS status

boolean canGetLocation = false;

Location location; // location

double latitude; // latitude

double longitude; // longitude

// The minimum distance to change Updates in meters

private static final long MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES = 10; // 10 meters

// The minimum time between updates in milliseconds

private static final long MIN\_TIME\_BW\_UPDATES = 1000 \* 60 \* 1; // 1 minute

// Declaring a Location Manager

protected LocationManager locationManager;

public GPSTracker(Context context) {

this.mContext = context;

getLocation();

}

public Location getLocation() {

try {

locationManager = (LocationManager) mContext

.getSystemService(LOCATION\_SERVICE);

// getting GPS status

isGPSEnabled = locationManager

.isProviderEnabled(LocationManager.GPS\_PROVIDER);

// getting network status

isNetworkEnabled = locationManager

.isProviderEnabled(LocationManager.NETWORK\_PROVIDER);

if (!isGPSEnabled && !isNetworkEnabled) {

// no network provider is enabled

} else {

this.canGetLocation = true;

if (isNetworkEnabled) {

locationManager.requestLocationUpdates(

LocationManager.NETWORK\_PROVIDER,

MIN\_TIME\_BW\_UPDATES,

MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

Log.d("Network", "Network");

if (locationManager != null) {

location = locationManager

.getLastKnownLocation(LocationManager.NETWORK\_PROVIDER);

if (location != null) {

latitude = location.getLatitude();

longitude = location.getLongitude();

}

}

}

// if GPS Enabled get lat/long using GPS Services

if (isGPSEnabled) {

if (location == null) {

locationManager.requestLocationUpdates(

LocationManager.GPS\_PROVIDER,

MIN\_TIME\_BW\_UPDATES,

MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

Log.d("GPS Enabled", "GPS Enabled");

if (locationManager != null) {

location = locationManager

.getLastKnownLocation(LocationManager.GPS\_PROVIDER);

if (location != null) {

latitude = location.getLatitude();

longitude = location.getLongitude();

}

}

}

}

}

} catch (Exception e) {

e.printStackTrace();

}

return location;

}

/\*\*

\* Stop using GPS listener

\* Calling this function will stop using GPS in your app

\* \*/

public void stopUsingGPS(){

if(locationManager != null){

locationManager.removeUpdates(GPSTracker.this);

}

}

/\*\*

\* Function to get latitude

\* \*/

public double getLatitude(){

if(location != null){

latitude = location.getLatitude();

}

// return latitude

return latitude;

}

/\*\*

\* Function to get longitude

\* \*/

public double getLongitude(){

if(location != null){

longitude = location.getLongitude();

}

// return longitude

return longitude;

}

/\*\*

\* Function to check GPS/wifi enabled

\* @return boolean

\* \*/

public boolean canGetLocation() {

return this.canGetLocation;

}

/\*\*

\* Function to show settings alert dialog

\* On pressing Settings button will lauch Settings Options

\* \*/

public void showSettingsAlert(){

AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

// Setting Dialog Title

alertDialog.setTitle("GPS is settings");

// Setting Dialog Message

alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");

// On pressing Settings button

alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog,int which) {

Intent intent = new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS);

mContext.startActivity(intent);

}

});

// on pressing cancel button

alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

// Showing Alert Message

alertDialog.show();

}

@Override

public void onLocationChanged(Location location) {

}

@Override

public void onProviderDisabled(String provider) {

}

@Override

public void onProviderEnabled(String provider) {

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override

public IBinder onBind(Intent arg0) {

return null;

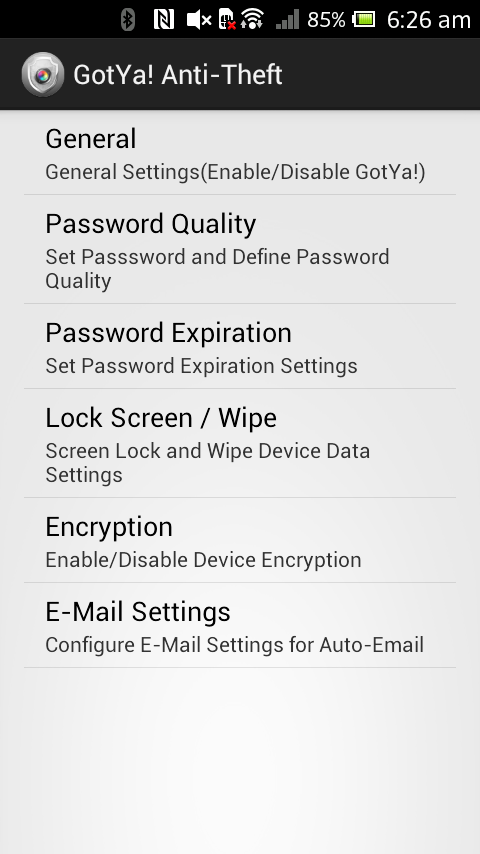
}

}

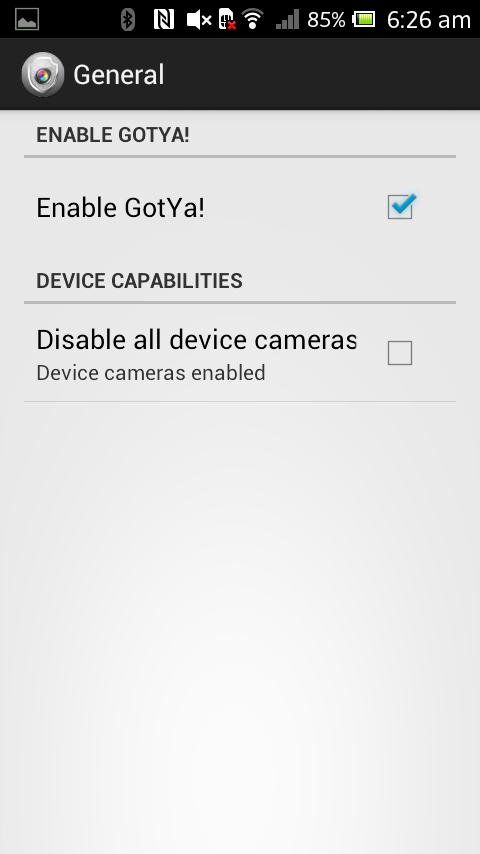
11. RESULTS AND FINDINGS

**11.1 Output Screenshots**

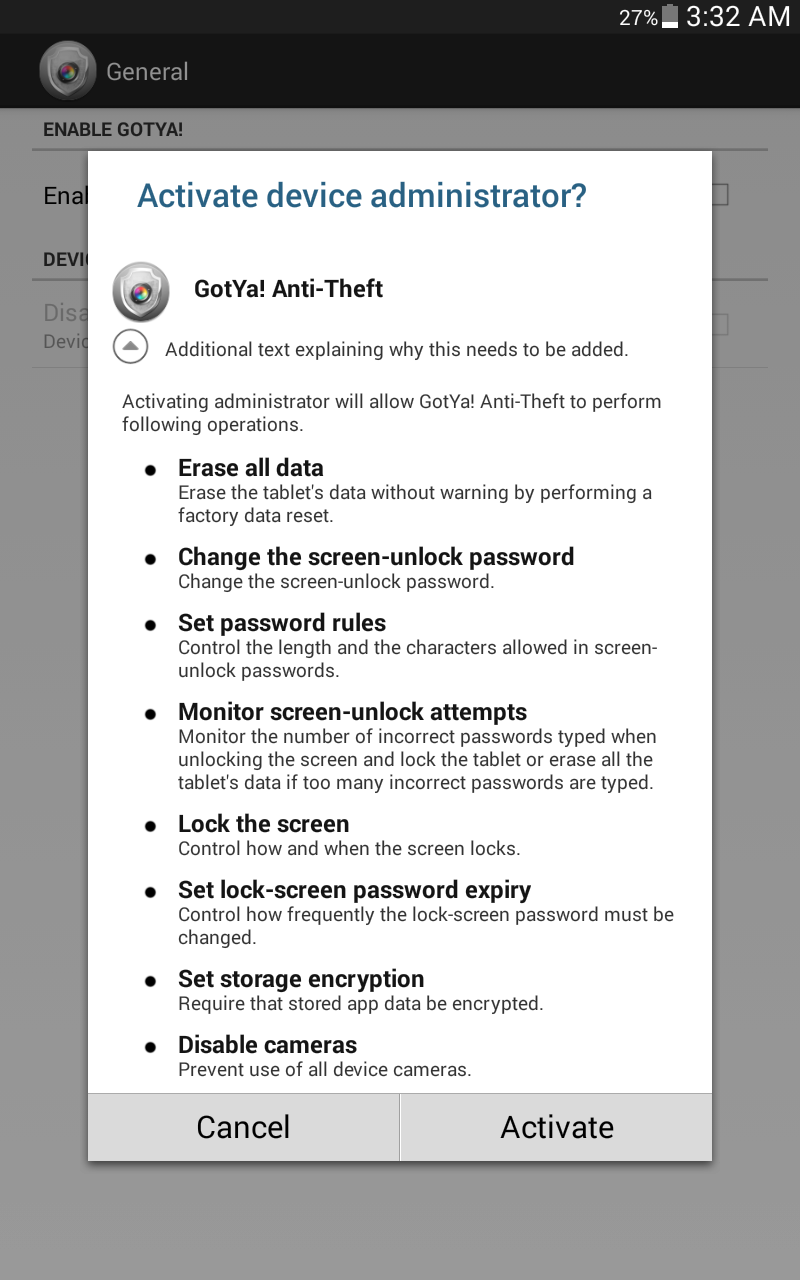
Home Page



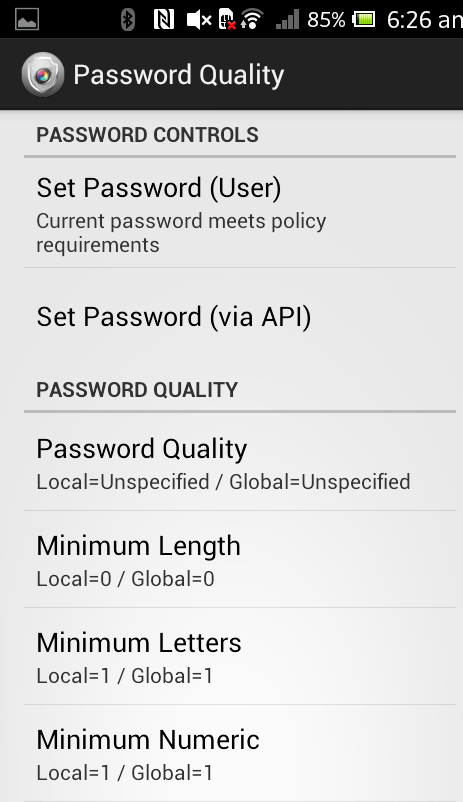
Enabling Gotya



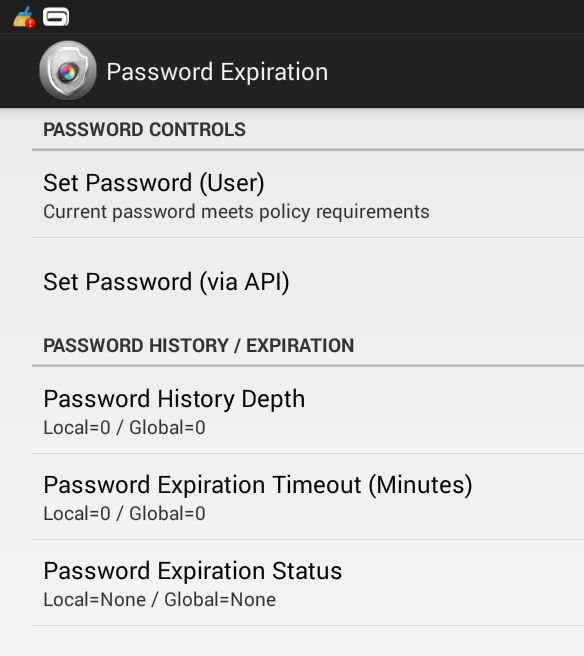
Activating Gotya



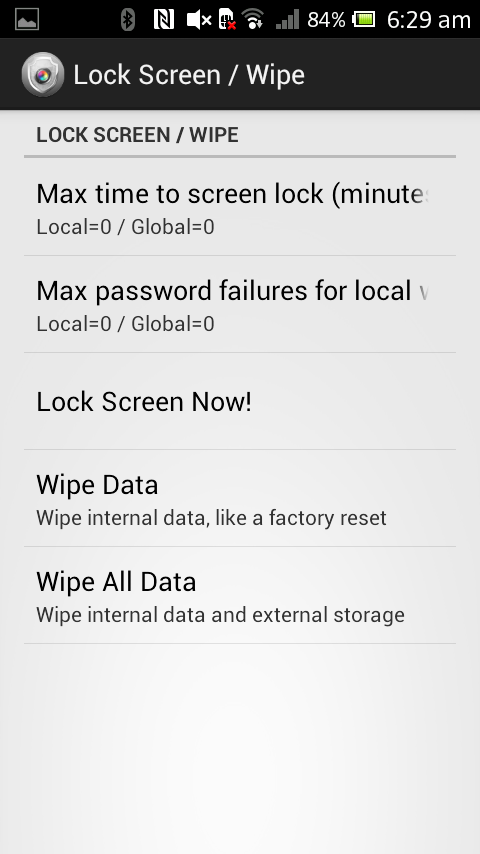
Password Quality



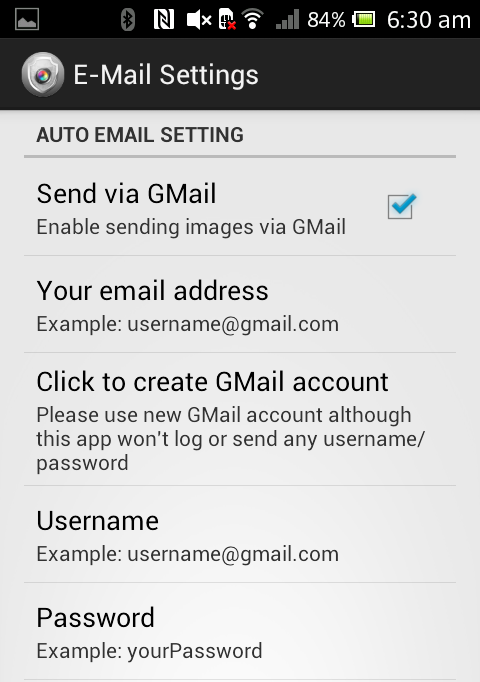
Password Expiration



Lock screen/wipe



E-mail Settings



**11.4 Limitations**

The project gotya have certain limitations.

1. To take the image of the thief the system should have an active data connection.

2. The exact place and the name of the place will not be displayed while the latitude and longitude details are provided.

3. The image captured by the front camera will be clear only if the thief gives a clear view of his face.

11.5 future scope

The main factors like face detection can be employed and the exact place can also be found out. To get more accurate alerts we can have some missing alerts like we can configure a group of numbers to which a missing alert is given when the thief tries to access the system. Features like video recording and sound recording is also possible in the future system.

9. CONCLUSION

Gotya will be a great security application which help as to find which unknown person took our smart phone. This application is mainly based on security and systems. Gotya provides better protection for the smartphones. The system will work only if there is a properly configured network connection. User get much protection than using other applications. Data encryption/decryption is included in this software, so that it guarantees the protection of the user.

Gotya also help as to erase all our personal data from the phone when some thief try to access our smart phone. . It also help as to encrypt our personal data in our phone. Gotya help as to set a strong password or a strong swipe pattern for our smartphone. It will also provide password expire technology.

9. REFERENCE

* Stefano LevialdiGhìron and Serena Sposato, ―NFC Ticketing: a Proto-type and Usability test of an NFC-based Virtual Ticketing application‖, First International Workshop on Near Field Communication, 2009.
* Jalotte Pankaj,” *Software Engineering”*, third edition, spring publishers, Narosa and Propagation (EUCAP)
* Elias M. Awad, *“System Analysis and Design”*, second edition, McGraw-Hill Professional publication.
* Rainer Steffen, JörgPreißinger, and Tobias Schöllermann, Near Field Communication (NFC) in an Automotive Environment‖, Second International Workshop on Near Field Communication