International Journal of Engineering & Technology, 7 (4.29) (2018) 182-191

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Android based Parental Monitoring Apps

Dalilah Abdullah[1]*, Muzaffar Mohamed[2], Herny Ramadhani Mohd Husny[3]

[1,2,3]UniKL MIIT
**Corresponding author E-mail:dalilah@unikl.edu.my*

Abstract

Nowadays the usage of handheld devices in a young generation were constantly increasing. Handheld devices had been involved in high profiles cases that involved children, therefore strict monitoring of children's devices usage are required for safety risks. This project is to develop Stealth-based Android Parental Monitoring App to help parent monitor their child's mobile activity silently. The mobile usage behavior can be analyze and then parents can plan for proactive measures to deal with the problem. This application was developed based on client server application, where Android based application is developed for client side and web application is developed on the server-side. Authentication process is required to activate the application, as only authenticated user can manage the application and at the same time, the application will run stealthily, invisible from the users. Pattern Lock and password method were used for authentication process. For the purpose of stealth process, "secret door" technique, where developer will disable an application component; this technique will have the effect of removing an application shortcut from any Launcher. The application at android based will be able to log the location, message log, call log, network log, and device information. The collected data will be uploaded to a web server to be reviewed through the internet-enabled devices. The functionality testing were done on both Android Based client side application and web application server side web application. For client side application eight test cases were developed: Android Authentication Module, Android device registration module, Android logging module, Android uploading module, Android system cleaner module, Android stealth hide/reveal icon module, Android stealth launch application via dialer module and Android protection module. For each test case, the result is indicated as pass. For Server Side Web application, four test cases were developed; Web Application Registration Module, Web Application Authentication Module, Web Application Monitoring Module and Web Application Disconnect Device Module. For each test case, the result is indicated as pass. Android Based Parental Monitoring Apps has been successfully developed and meet the development objective to collect Android critical data including network log, call log, device information log, incoming or outgoing SMS and Call logs in stealth and synchronized it to the server where it can be used as digital evidence if necessary. The result of this project is hoped to benefits the society by providing a better monitoring technique for the concerned parent via implementing an alternate way on how to continuously collect and monitor their children/teen mobile activity.

Keywords: *Stealth-based; Parental Monitoring Tool; Android*

1. Introduction

The world nowadays was shifting rapidly, and the usage of technology especially handheld devices was continuously increasing. A report by [14] revealed that smartphones ownership in Malaysia has constantly seen a growth almost every year. Based on their report which targets around 43 million of the mobile user from various telco company in Malaysia also discover that 1.8% (778 464) of the mobile user in 2014 was under 15 years old. This is appalling and dangerous as the children or teenager, especially the pre-teenager itself cannot sense danger as they still in a learning process in their life. The children/teenager can easily be manipulated and without proper guidance and monitoring from the parent, this situation could be a hazard and dangerous for them. The usage of the smartphone at this early stage of their age may expose them to online predators, adult or harmful content etc. Due to this situation, parental control application was developed in order to contain this problem. However, the drawback of an existing parental control tool today is too restrictive or difficult to configure; thus, does not always work for the kids nowadays. Kids from Generation Z nowadays tends to be more technical savvy compared to Gen Y generation as they were born with a highly-sophisticated media and computer environment [13]. Upon realizing the existing of parental control tool, they might feel insecure, try to bypass the restriction or even become more creative in finding an alternative to the problem. This situation worsens when there's some parent abdicate their responsibility to keep monitoring their children/teenager after installing the parental control software as thinking that step is enough [17]. Through analyzing the children mobile usage behavior, the parent may plan for proactive measures ahead to deal the problem instead of blocking all the application or internet access.

1.1. Proposed System

The main purpose of this project is to deliver an alternative way for a parent to keep track and monitor all mobile activities of their children/teenagers continuously by stealthy collecting the data and upload it to the server to be observed and analyzed anytime they want later. The Android open source platform has been chosen for this project due to its popularity which dominated the smartphone market in the world now [11].This reputation is contributed most by their policy towards "openness" of their source code. According to [4], Android has a good design application integration which enables the developer to write an application which can

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

leverage the core function secure and seamlessly. An Android development kit, for example, Android SDK also available free without any upfront cost.

2. Literature Review

Ownership of smartphone in the world has increased at a rapid rate and has been adapted as one of the most important parts of our life. Other than we can gain an information at the fingertips, the smartphone also acts as a gateway to connect and socialize us with others. [5] claimed that more than half of children have exposed to the internet and online social network by the age of 10. They also agreed that without parental guidance and monitoring, the children might expose to the various threat which may affect their morale and behavior as being displayed in table 1. [2] also, claimed that most of the parents do not know their children/teenager online activity. Therefore, a suitable mechanism was necessary to help parent or authorities to monitor and track their location as early as possible in order to save them before it's too late.

Despite the existing of parental control tools, they often bundled with several problem and weaknesses. Too restrictive is the common problem for this tools. The restriction might work on children, but not for a teenager. They might start to argue, rebel and feel insecure. Current parental control also tends to be too obvious while the children/teenager from this generation was extremely tech savvy. Upon realizing the presence of parental control, they might try to uninstall or bypass the restriction thus will defy the main parental control objective. Based on the situation raised, it is understood that the stealth android parental monitoring control is necessary to help the parent to monitor their children/teenager continuously.

Table 1: Internet Risk to Children/Teenager

	Commercial	**Aggressive**	**Sexual**	**Values**
Content (Children as recipient)	Advert Spam Sponsorship Personal Info	Violent / Hateful content	Pornographic or unwelcomed sexual content	Bias Misleading Info Racist
Content (Children as participant)	Tracking Harvesting Personal Info	Being Bullied Harassed Stalked	Meeting Strangers Being groomed	Self-harm Unwelcome Persuasive
Content (Children as actor)	Illegal downloading Hacking Gambling Financial scam Terrorism	Bullying or harassing others	Creating or uploading inappropriate materials	Provide misleading info / advice

2.1 Parental Monitoring Tools

[9] had defined the monitoring system as a software which is used to quietly monitor and log the device activities of family members, employees etc. However, in order to distinguish this mechanism from the malicious application, the following criteria must be met in order to be considered as legal including:

1. The software owner must also the owner of the device which going to be installed on.
2. The software owner must be a parent of a minor child
3. The software owner must consent all the user which affected by the software

In Android, there are various data that can be collected in order to help the parent or forensic investigator to analyze the mobile user's activity.

2.1.1 Types of Information Data

In order to effectively analyze the user behavior and their online activity, a proper data was necessary to provide the parents or forensic investigator a better insight of children/teenager mobile activity. Those important criteria and features listed as follows:

1. Message tracking
   This feature used to collect and log the client conversation data including SMS, MMS, and social networking. This data is important to have an idea of their children/teenager behavior and at the same time able to detect an inappropriate behavior, for example sexting, cyberbully etc.
2. Location tracking
   This is another important feature usually used by existing parental monitoring tools to monitor and tracks the children/teenager location. This feature relies heavily on the usage of GPS (Global Positioning System) which mostly embedded within the devices itself. However, the GPS was rarely enabled as this service consumes a lot of battery.
3. Network activity tracking
   This tracking activity involves the process of retrieving the connected network data from the service provider. The data that can be collected including Internet Protocol (IP) address and Network Name (SSID). This critical data may give lots of clue on target activity especially when they are on location.
4. Call tracking
   This tracking method logs the incoming and outgoing call. This information is valuable, as it will help to know who and whom their child was contacting with, especially when it involves missing or run away from home cases.
5. Device Information Tracking
   Device information occasionally imperative in providing parent a better insight of target device, which in this case their children. The usually collected data including current battery level, total RAM and OS version.

2.2 Current Method of Android Based Parental Tools

Based on the research that has been conducted on the vast amount of application which available in the market, the developer had identified the common method which has been used by the application which is local based application.

2.2.1 Local based application

The oldest and the most basic method of parental monitoring tools. A specific script/command was used in order to collect and log the required data, for example, SMS history and store it in a form of files, usually in XML (Extensible Markup Language) or HTML (Hypertext Markup Language) format. In order to access and collect this kind of data, a proper permission which is needed to access the resources must be declared within the AndroidManifest.XML file as this kind of request usually falls under the dangerous permission group category [6].All of the data or logs which has been collected was typically stored securely within the "data/data/<application name>" folder but however, a physical access toward the devices is necessary when analysis or viewing of the captured logs is required.

2.2.2. Client-server based application

This technique contains the essence of locally based application meanwhile provide an enhancement for the previous method of parental monitoring tools. By exploiting the internet technology, the captured logs were uploaded into the web server. The collected logs then were first stored inside local SQLite database before synchronized to the server. The uploaded data then will be displayed on the website thus enable an easy analysis of the data by

utilizing an internet browser from any internet-enabled devices. According to [16], The Android API has a various set of functions which enable the developer to use the HTTP requests. [8] defines the Hypertext Transfer Protocol (HTTP) as services which work as the request-response protocol between the client and servers.

The most popular method of HTTP request methods is POST and GET. The GET HTTP request used to request the data from a specified resource while the POST HTTP request was used to submit data be processed to a specified resource. The POST function, however, is the most related functions for this project as this services allow the developers to upload and update the contents of a file in a web server.

In addition to the permission requested in the previous technique, this method requires another permission, which is the INTERNET permission enabled within the AndroidManifest.XML file. This permission is necessary in order to enable the application to use the internet connection for synchronizing and uploading captured data.

2.2.3. Advantages of Android-based parental monitoring tools

The benefit of using this mechanism has been identified and listed as follows:

1. This method offers a simple yet powerful technique of information gathering.
2. These tools help parents to monitor their child/teenager activity.

2.3 Security Mechanism

2.3.1. Uninstallation Protection

The android Device Administration is an API, which has been introduced since version 2.2 (Froyo). According to [11], this services focused on offering a support for the enterprise applications by enabling the application admin to enforce security policy, for example, prompt user to set a new password, lock the device immediately and even perform a factory reset or data wiping remotely. If any breach of the security policy by the user, it is up to the application itself to decide on how to handle it. However, if the device admin application was not enabled by the user, he was not subjected to its policies. In order to force stop or uninstall the device admin applications, the user itself must disable or deactivate the application as a device administration as shown in figure 1.

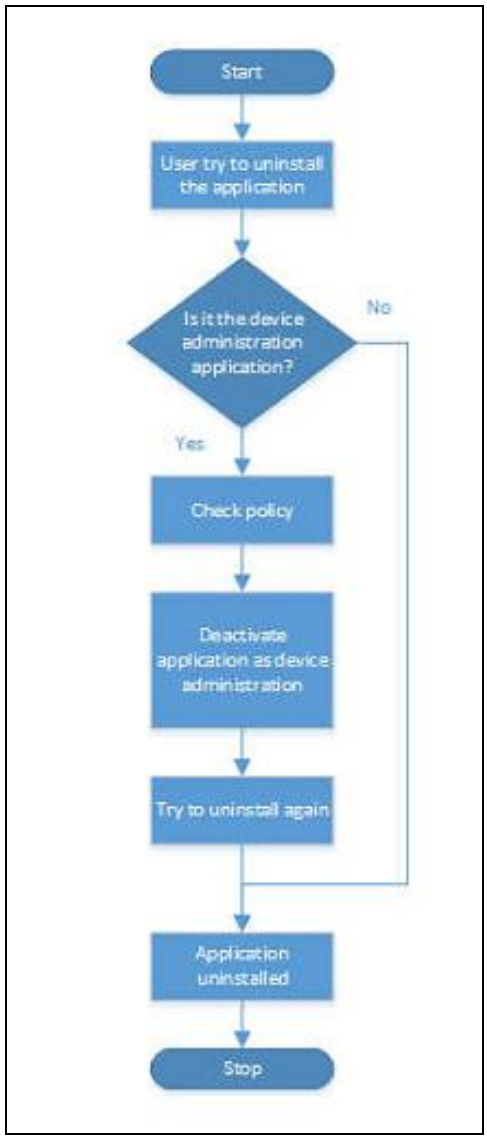


Fig. 1: Device Administration Process

In order to prevent the application from being forced to stop or uninstalled, most of the applications, which have been reviewed, were using this mechanism. In addition, they also enforce a security policy where when the user wants to disable the device admin application, the application will lock the device automatically by using the DevicePolicyManager.lockNow() and prompt a password in order to continue. The password can beprompt via local operating system or remote authentication server.

The device administration API involves the following classes, including:

1. DeviceAdminReceiver
   This base class must be included within the Device Administration application and used to interpret the raw intent action sent by the system.



Fig. 2: Android Administration Receiver

2. DevicePolicyManager: Responsible for managing the policy enforced on a device.
3. DeviceAdminInfo: Used to specify metadata for a device administration component.

2.3.2. Stealth

In area of software and application, this term was related most with the viruses, worm etc. In Android, this term refers to the ability for an application to run virtually undetectable or, at least, hard to be detected by the user [15]. There is a various technique which currently available in the market for example:

1. Application "secret door"
   One of the most popular technique or method used to hide the installed application from others. By using this method, the developers disguise their application as another application for example antivirus scanner, clock, notepad, calculator etc. This technique aimed to trick the others from knowing the existence of this application. Thus, only the valid user will know the secret way to reveal the PIN pad and open the real functionality of the application [7].
2. Hidden application shortcut
   This is another popular method used by the various application within the marketplace. These programmatically methods require the developer to manually disable an application component via PackageManager#setComponentEnabledSetting() as shown in figure 3. This method will have the effect of removing an application shortcut from any Launcher [12].

```
public void setDisabled()
{

  ComponentName componentName = new ComponentName(getApplicationContext(),PhoneCoordinateActivity.class);
  getPackageManager().setComponentEnabledSetting(
          componentName,
          PackageManager.COMPONENT_ENABLED_STATE_DISABLED,
          PackageManager.DONT_KILL_APP);

  }
```

Fig. 3: Example of setComponentEnabledSetting

As the application was hidden from others, a proper way was necessary to launch the application for the valid user so that others cannot launch it same way. To solve this problem, the developer had enables the application to be launch just via the dial pad. In order to launch Application from Dialer, the developer itself need to register a receiver for that in AndroidManifest.XMLfile as shown in Fig. 4. These features are made possible by exploiting the broadcast intent and filter for an outgoing call. By using this technique, any application that wants to enable this features can define their own code and subscribe to those broadcast. One the secret dial number has been set, the application can be executed by simply dialing the secret code number, for example, ##1234 from the dial pad.

```
<receiver
    android:exported="true"
    android:name=".OpenPassword" >
    <intent-filter >
        <action android:name="android.intent.action.NEW_OUTGOING_CALL" />
    </intent-filter>
</receiver>
```

Fig. 4: Example to register receiver within AndroidManifest.xml

3. Methodology

The development of this project were divided into two major categories which are client side Android application and server side web application.

3.1 Client Side System Design

For client-side application, the design will cover 1) user authentication; 2) device registration; 3) log data to local SQLite; 4) upload recorded data to server; 5) delete unneeded record from SQLite; 6) launch application via dialer module and; 7) protect itself from being uninstalled or force stopped. Figure 5 show Swimlane activity diagram of the Client side application.

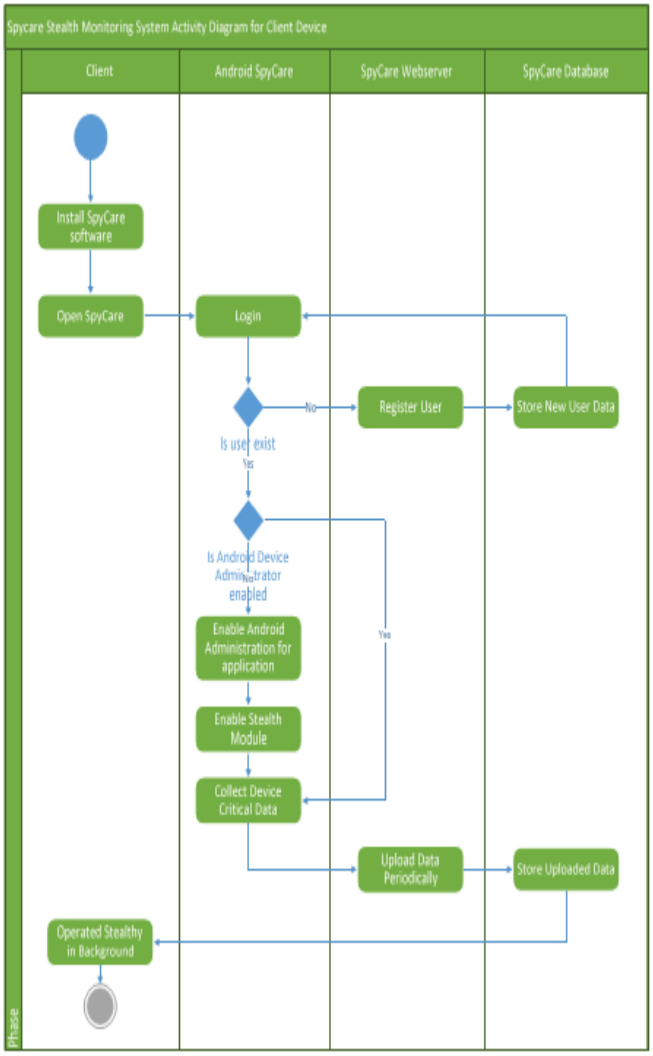


Fig. 5: Swimlane activity diagram for the client side application

3.1.1. Authentication Module

Authentication is the very first process, which will be prompt to the user before they were able to use the functionalities. This process is vital to ensure that only a right person was able to view and monitor the devices.

3.1.2. Device Registration Module

International Mobile Equipment Identity (IMEI) is a unique 14 to 16-digit decimal string, which can be used as simple identification for each mobile device [2]. In order to distinguish between each device which has been connected, the collects this information and validate it towards the record saved in the server.

3.1.3. Logging Module

Information logging is the core activity for client-side Android application. This logging module involves the application to be

executed in the background and able to function without user intercession. The app will silently collect numerous critical information which can be used as digital forensic evidence for example SMS, call, network, device information and network log and save it locally inside SQLite database. Then later, periodically those record will be synchronized with the server.

3.1.4 Uploading Module

Without this module, the previous logging module will be wasted. This module main purposes are to upload and synchronize the collected data with the MYSQL server. The process starts when the application periodically requests for IPV4 in the background. Failure to retrieve the IPV4 indicates that the unavailability of internet connectivity thus the application will go idle for a moment before it starts to request the IPV4 again. Once the application managed to grasp the IPV4, it will start to search and upload all the native record to the MYSQL server. The data was sent in a form of encoded JSON object before being decoded back by the PHP function on the server-side receiver to be processed.

3.1.5 System Cleaner Module

Data that had been uploaded into MYSQL server, no longer beneficial in SQLite. Therefore, as being suggested by its name, this module does a cleaning task where it will delete those records within SQLite database, in order to optimize overall application space and performance.

3.1.6 Stealth Module

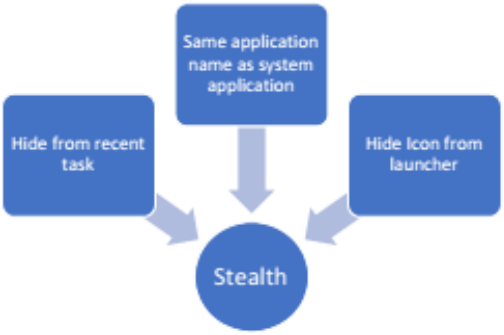


Fig. 6: Stealth Module

This module is another core component for the app. In order to accomplish "stealth" capability, three minor components have been used. The first component for this module is an ability to be hidden from the recent task. A task as shown in figure 6 is a collection of activity which arranged in the stack when performing a certain job. A capability for an application to be hidden from this task list will help to keep the application concealed from being exposed in case if the user forgot to clear the recent task



Fig 7: Android Display the Recent Task List

The following stealth component is by naming the application similar to an existing inbuilt system application for example "Google Play Service". By default, Google Play Service is an application developed by Google, which functions to provide core functionality like authentication to user Google services, synchronized contacts, access to all the latest user privacy settings, and higher quality, lower-powered location based services [7]. By naming our application as the same name of this application, the user will be less aware of its presence. Moreover, with the intention of making it more convincing; the developer used the original Google Play Service icon as its icon.



Fig 8: Google Play Service's Icon

Finally, the developer correspondingly implements a function to hide an application icon from the launcher. This component is made possible by programmatically modifying the manifest as shown in the code snippet below.

```
public void HideIcon(){

    ComponentName componentName = new ComponentName(getApplicationContext(),Login_Activity.class);

    getPackageManager().setComponentEnabledSetting(
            componentName,
            PackageManager.COMPONENT_ENABLED_STATE_DISABLED,
            PackageManager.DONT_KILL_APP);
    btn_hide_icon.setText(R.string.app_reveal_icon);
    myPref.saveToPref("Hide_Icon","true");
    Toast.makeText(Setting_Activity.this,R.string.app_hide_reveal, Toast.LENGTH_LONG).show();

}
```

Fig 9: Function to Hide an Application Icon

To launch the application where its icon was hidden, the developer had to implement another function, which enables the user to launch it by merely dialing a secret PIN from the dialer.

3.1.7 Protection Module

This module is made possible by implementing the Android Device Administration API. This API provides an additional feature at the system level which the developer to generate security-aware applications that are convenient in enterprise settings, in which IT professionals require rich control over employee devices. For this system, the app harnessing this API by protecting the application from being forced stopped and uninstalled. Furthermore, the user correspondingly cannot uninstall the application without a valid PIN. However, the user itself must enable the device administrator first manually to ensure this that this function is running.

3.2. Server Side web system Design

The server-side web application also has its own system design, where it will cover 1) New user registration; 2) User authentication; 3) display uploaded log and 4) disconnect device. Figure 10 show Swimlane activity diagram of the server side web based application.

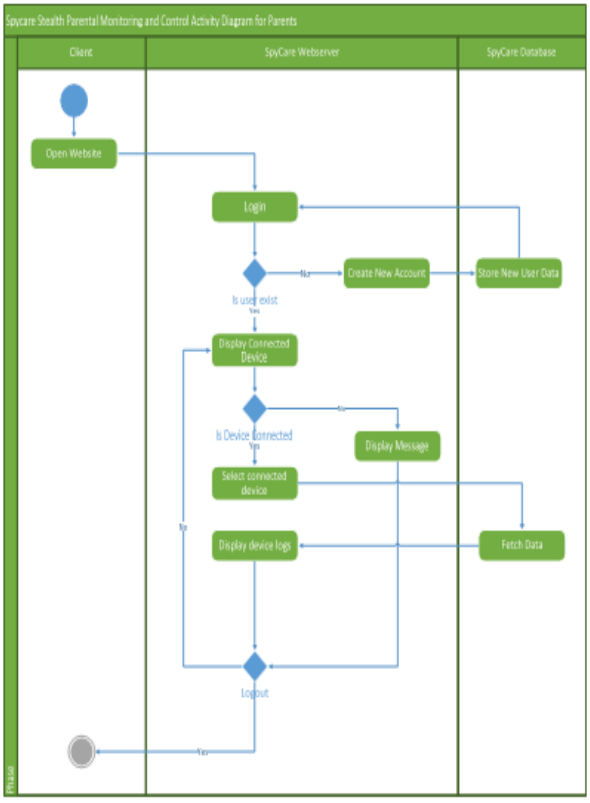


Fig. 10: Swimlane Activity Diagram for the Web Server Application

The focus of server-side web application is to display the logged data for the authorized user which in this case, the parent.

3.2.1. Registration Module

This module used to authenticate and register a new user for the system. The new user will first be authenticated against the record from the server. If only the record still not existed, then the user will be registered. For security purposes, the password will be hashed with bcrypt algorithm.

3.2.2. Authentication Module

This module authenticates the user based on the given login credential, to associate the password against the stored bcrypt hashes in MySQL record.

3.2.3. Monitoring Module

This web based PHP module is used by the legitimate user to view the logged data. All associated device which linked to user account will be shown in this module. With dashboard concept, record collected from client side will be displayed to user.

3.2.4. Disconnect Device Module

This module is used to disconnect the device from the monitoring list. This function which can be accessed from a web application homepage was useful to help the user to delete all the data and record from MYSQL server which involves the device that they want to disconnect.

4. Result and Findings

This chapter focused on the testing process of the system, which been developed to investigate whether the application works as expected and can be implemented with the same characteristic. All the main features was tested. This is to ensure all functions provide the expected output. Functional testing will complete when all features have passed all acceptable associated test cases

4.1. Functional Testing

Functional testing is a software testing methodology that is used to test the functionalities of the system or software and usually covered overall scenario including failure and boundary cases. In this project, the developer uses a Unit Testing as the main functional testing technique to test each of individual modules for issues or problems.

4.1.1. This system was tested on eight (8) modules:

1. Android Authentication Module
2. Android device registration module
3. Android logging module
4. Android uploading module
5. Android system cleaner module
6. Android stealth hide/reveal icon module
7. Android stealth launch application via dialer module
8. Android protection module

The following Table 2 to 9 are the test cases for the client-side Android-based stealth parental monitoring apps.

Table 2: Test Case for Android Authentication Module

Test Case ID	T01_C_M
Test Case Name	Testing Android authentication module
Purpose	To test the module if able to authenticate the user credential and toast message based on the response it receive from the server.
Initiation Criteria	1. The android device need to be connected to the internet 2. The user must already register
Execution Steps	1. Launch the app 2. Insert email and password in the given field and press login
Expected Results	The app will be able to authenticate the user credential and toast message based on the response it receives from the server.
Results	Pass

Table 3: Test Case for Android Device Registration Module

Test Case ID	T02_C_M
Test Case Name	Testing Android device registration module
Purpose	To test the module if able to register a new device or proceed if device already exists.
Initiation Criteria	1. The android device need to be connected to the internet 2. The user login credential must be valid
Execution Steps	1. Launch the app 2. Login to registered using a valid credential 3. If login activity is success, the user will automatically redirected to the device registration page.
Expected Results	The app will suggest user register the device if it is still not registered, while will suggest proceeding if the device is already registered.
Results	Pass

Table 4: Test Case for Android Lodging Module

Test Case ID	T03_C_M
Test Case Name	Testing Android lodging module
Purpose	To test if the module able to log network information, location data, device information, incoming or outgoing SMS and call into local SQLite database.
Initiation Criteria	1. The user must already login 2. The user manually needs to grant the related permission if requested. 3. Some Android device requires GPS enable to log the location data.
Execution Steps	Launch the app (The logging service will start in background)
Expected Results	The app will be able to log network information, location data, device information, incoming or outgoing SMS and call into local SQLite database.
Results	Pass

Table 5: Test Case for Android Uploading Module

Test Case ID	T04_C_M
Test Case Name	Testing Android uploading module
Purpose	To test the module if able to upload all the logged data to remote web server
Initiation Criteria	1. The android device needs to be connected to the Internet. 2. In some device, this module only works when the screen is on.
Execution Steps	Launch the app (The uploading service will be running periodically in background)
Expected Results	The app will be able to upload all the logged data to remote web server
Results	Pass

Table 6: Test Case for Android System Cleaner Module

Test Case ID	T05_C_M
Test Case Name	Testing Android system cleaner module
Purpose	To test the module if able to able to delete all the local SQLITE record with UPLOADED flag is TRUE
Initiation Criteria	1. The android device needs to be connected to the internet. 2. In some device, this module only works when the screen is on.
Execution Steps	Launch the app (The system cleaner service will be running periodically in background)
Expected Results	The app will be able to delete all the local SQLITE record with UPLOADED flag is TRUE
Results	Pass

Table 7: Test Case for Android Stealth, Hide/Reveal Icon Module

Test Case ID	T06_1_C_M
Test Case Name	Testing Android stealth module: Hide/Reveal icon from launcher
Purpose	To test the module if able to be hide/reveal the icon in launcher
Initiation Criteria	1. The user need to activate this function manually from setting page 2. This function usually takes 30 seconds to active 3. Some device requires restart to hide/reveal the icon
Execution Steps	1. Launch the app 2. Go to *Setting* activity 3. Click *Hide Application Icon* or *Reveal Application Icon*
Expected Results	The app will be able to hide or reveal its icon from launcher
Results	Pass

Table 8: Test Case for Android Stealth, Launch Application via Dialer Module

Test Case ID	T06_2_C_M
Test Case Name	Testing Android stealth module: Launch Application from Dialer
Purpose	To test the module if able to launch the application from the dialer by using a secret PIN
Initiation Criteria	The user need to insert 4-digit secret PIN first from main activity
Execution Steps	1. Launch the app 2. Change secret PIN 3. Minimize application 4. Launch application by dialing secret 4-digit PIN using following preset ##PIN##
Expected Results	The app will be able to launch the application from the dialer by using a secret PIN
Results	Pass

Table 9: Test Case for Android Protection Module

Test Case ID	T06_C_M
Test Case Name	Testing Android protection module
Purpose	To test the module if able to protect application process from being force stopped or being uninstalled from Android *Setting*
Initiation Criteria	-
Execution Steps	1. Launch the app 2. Activate the application into device administration list first by clicking *Setup Device Manager*
Expected Results	The app will be able to protect application process from being force stopped or being uninstalled from Android *Setting*
Results	Pass

The Table 10 to 13 are the test cases for the web-application based stealth parental monitoring app. The system was tested on four (4) modules:

1. Web-application registration module
2. Web application authentication module
3. Web application monitoring module
4. Web-application disconnect device module

Table 10: Test Case for Web Application Registration Module

Test Case ID	T01_S_M
Test Case Name	Testing Web Application Registration Module
Purpose	To test the module if able to register a new user inside its database
Initiation Criteria	The browser/device must have connected to the internet
Execution Steps	1. Open http://spycare.oligococo.tk/ 2. Click *Register* button at the top right corner of website 3. Insert the credential (username, email, and password) 4. Click *Create Account*
Expected Results	The web application will be able to register a new user inside its database
Results	Pass

Table 11: Test Case for Web Application Authentication Module

Test Case ID	T06_C_M
Test Case Name	Testing Web Application Authentication Module
Purpose	To test the module if able to authenticate the user against registered credential inside the database and redirect the user to home page.
Initiation Criteria	The user must already register
Execution Steps	1. Open http://spycare.oligococo.tk/

	2. Click *Login* button at the top right corner of website 3. Insert the credential (email and password) 4. Click *Create Sign In*
Expected Results	The web application able to authenticate the user against registered credential inside the database and redirect the user to home page.
Results	Pass

Table 12: Test Case for Web Application Monitoring Module

Test Case ID	T03_S_M
Test Case Name	Testing Web Application Monitoring Module
Purpose	To test the module if able to display selected device log including device information, network log, location log, incoming or outgoing SMS and call log in the proper format.
Initiation Criteria	- The user must login first to access this page - The user need to select which phone they want to monitor first - For a new device, user need to wait for 10 minutes before the device information will appear
Execution Steps	1. Login to web-application at http://spycare.oligococo.tk/login.php 2. If login success, the user will be redirected to the homepage. Choose the device to monitor and click *View Device* button.
Expected Results	The web application able to display selected device log including device information, network log, location log, incoming or outgoing SMS and call log in the proper format.
Results	Pass

Table 13: Test Case for Web Application Disconnect Device Module

Test Case ID	T04_S_M
Test Case Name	Testing Web Application Disconnect Device Module
Purpose	To test the module if able to delete all the record which involving the selected device from its MYSQL database.
Initiation Criteria	- The user must login first to access this page - The device must already connect
Execution Steps	1. Login to web-application at http://spycare.oligococo.tk/login.php 2. If login success, the user will be redirected to the homepage. Choose the device to be disconnect and click *Disconnect Device* button.
Expected Results	The web application able to delete all the record which involving the selected device from its MYSQL database.
Results	Pass

4.1.2. Test Case Results

As mentioned, the main purpose of functionality testing is to access each module for bugs while at the same time to test whether it is working according to the proposed design specification. The developer does the unit testing and the outcomes of the test cases is recorded. Based on the outcomes which have been collected from the test cases *T01_C_M* to *T08_C_M*, the result can be concluded that all elements are functioning as expected and the result is indicated as *"Pass".*

The unit testing for a server-side web application is correspondingly being conducted by the developer and the results of the test cases are logged. Based on the outcomes which have been collected as of the test cases T01_S_M to T04_S_M, the test for the server-side web application also can be determined that all elements are working as expected and the result is specified as *"Pass".*

4.2. Non-Functional Testing

Compatibility testing is a type of test used to test the compatibility of the developed system against various other objects such as web browser, hardware platform, operating system and so on. There are three (3) component of Android app that has been highlighted by the developer for compatibility testing including:

1. To test, if application can auto start after device reboot
2. To test if application can auto restart its services if killed by task manager
3. To test if application able to log location without GPS toggle enabled
4. To test if application able to log data
5. To test if application able to synchronize data to server

Each of this component has been rated with priority flag which either *mandatory* or *preferable*. If the tested component with its *mandatory* flag is failed, the application can be considered incompatible with the stated device. *Preferable* flag in the other hand means it is required by the application to run correctly but not mandatory to be fulfilled as it may not critically affect the application performance. Several devices also have been detected unable to log location without GPS toggle enabled. The result details for this test is shown in Table 14.

Table 14: Compatibility Testing Result

Test Condition	**Device**			**Priority**		
	Xiaomi Redmi Note 3	**Asus Zenfone Go**	**Samsung Galaxy Note 8.0**	**Samsung Galaxy SIII**	**OPPO R1011**	
Android Version	Lollipop 5.1.1	Lollipop 5.1.1	KitKat 4.4.2	Jelly Bean 4.3	KitKat 4.4.0	-
Able to auto restart after device reboot?	Pass	Pass	Pass	Pass	Fail	M
Able to auto restart background service is killed?	Pass	Pass	Pass	Pass	Fail	M
Able to log GPS without GPS toggle is enabled?	Pass	Pass	Fail	Fail	Fail	P
Able to log data?	Pass	Pass	Pass	Pass	Fail	M
Able to synchronize logged data to the server?	Pass	Pass	Pass	Pass	Fail	M

*Mandatory: M, Preferably: P

This chapter had discussed the test result of the apps for both client and server side application. Based on the functional testing that has been conducted, the actual test result for each module is recorded and compared against the functional requirement which previously stated in section three (3) which is in chapter methodology and project purpose which have been discussed in section one (1). This project is considered to be a success based on its encouraging achievement in the real test result where it was successfully meet the project objective.

5. Conclusion

A SWOT analysis was performed to present the conclusion of this project. Figure 11 below shows the strengths, weakness, opportunities and threats. Simple yet effective is one of the strength points of this project.

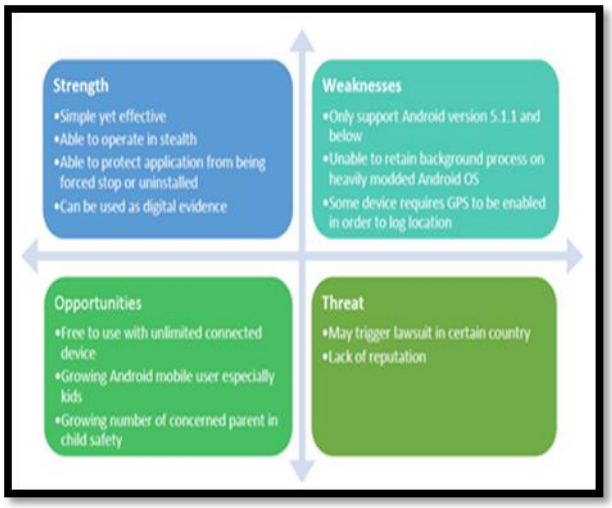


Fig. 11: SWOT Analysis

This app has been built with self-explanatory straight to the point user interface elements thus enable the parent to use the application easily. An ability to be operated in stealth and prevent its own process from being force stopped or uninstalled is also a golden element of this app. Furthermore, the data collected by this application also can be used as an important digital evidence if necessary.

Despite its strength, this application however only able to support Android running version 5.1.1 Lollipop and below. In some cases, even the Android version is supported; the application still unable to run correctly due to heavily modded Android operating system by device vendor. Some device in other cases requires a GPS to be enabled in order to log location data.

Several opportunities of growth for this project were discovered. Many existing products, which available in the market nowadays was in poor quality while the good one is rather too expensive. The app, however has been offered free with unlimited connected device usage. The growing numbers of smartphone mobile user in Malaysia, especially by the younger user, was also the growth opportunity for this project. Due to this trends, the number of concern parent also growing thus at the same time open the same opportunity for the app. Lastly the threat for this project. In some country, the user privacy is taken seriously and the misuse of this application may trigger a lawsuit series despite its benefit. The lack of reputation was also a threat as it still new in the market.

5.1. Recommendation

For this project, some recommendation for future enhancement has been highlighted:

1. Custom lock screen:
   Currently, the Android app is still using the default lock screen to lock the device. This is challenging as when the user activates the device manager for this application, the default device lock screen PINs will also be changed. The development of custom lock screen will eliminate this problem as it will separate the usage of both of this security measures
2. Battery optimization
   Android based app by default had preset the synchronizing interval in every five (5) minutes. This setting may consume lots of battery on some devices. A proper way for data synchronized interval can be researched to optimize the battery usage.
3. Enhanced logging capability
   As per requirement, the app was able to collect lots of critical information without requires superuser permission. These functions can be enhanced by adding more logging capability including logging calendar, browser history, and others.
4. Implementing Secure Socket Layer (SSL)
   SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client. By using these services, the client-server transaction is protected against Man-In-The Middle (MITM) attack, as this service will encrypt the transmitted data.

Summary

The Android Based Stealth Parental Monitoring App is developed to collect Android critical data including network log, call log, device information log, incoming or outgoing SMS and Call logs in stealth and synchronized it to the server where it can be used as digital evidence if necessary. The result of this project is hoped to benefits the society by providing a better monitoring technique for the concerned parent. This application will be providing an alternate way on how to continuously collect and monitor their children and/or teen mobile activity thus at the same time helps them to identify problems concerning the children before it is too late by planning ahead for any unavoidable incident from happening. In addition, with a proper adjustment; this application also suitable to be used by any organization with the aim to protect and monitor their employees

References

[1] Alavi, K., Nen, S., Mohamad, M., Sarnon, N., Ibrahim, F., & Mohd Hoesni, S. (2013). Social Science. Understanding the Factors of Children Missing/Running Away from Home in Malaysia, 2-5.

[2] Austin, K. (2015, February 4). 5 Dangerous Behaviors Teens do With Their Smartphones. Retrieved from Phone Sheriff: http://www.phonesheriff.com/blog/5-dangerous-behaviors-teens-do-with-their-smartphones/Brookes,

[3] T. (2012, November 16). What Is My Phone's IMEI and What Is It For?
Retrieved from Make Use Of: http://www.makeuseof.com/tag/phones-imeimakeuseof-explains/

[4] Conder, S., & Darcey, L. (2009). Android Wireless Application Development. Addison-Wesley Professional.

[5] Daily Mail Reporter. (2014, November 19). More than half of children use social media by the age of 10: Facebook is most popular site that youngsters join. Retrieve from Dailymail: http://www.dailymail.co.uk/news/article-2552658/More-halfchildren-use-social-media-age-10-Facebook-popular-site-youngsters-join.html

[6] Developer Android. (2015, 12 28). System Permissions. Retrieved from Developer Android: http://developer.android.com/guide/topics/security/permissions.html

[7] Gardener, R. (2014, February 14). Keep your Private Pictures hidden on Android with Keepsafe. Retrieved from Blogtech tips: http://www.blogtechtips.com/2014/02/14/keep-private-pictures-hidden-androidkeepsafe/

Google. (2016, December 31). Google Play services. Retrieved from Google Play:https://play.google.com/store/apps/details?id=com.google.android.gms&hl=en

[8] W3Schools.com. (2015, 12 29). HTTP Methods: GET vs. POST. Retrieved from W3Schools.com: http://www.w3schools.com/tags/ref_httpmethods.asp

[9] Webopedia. (2015, 12 28). Monitoring Software. Retrieved from Webopedia: http://www.webopedia.com/TERM/M/monitoring_software.html

[10] DC. (2015). Smartphone OS Market Share. Retrieved from International Data Corporation: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[11] Jones, K. (2012, October 11). Android Device Policy Administration Tutorial. Retrieved from New Circle: https://newcircle.com/s/post/1291/android_device_policy_administration_tutorial

[12] Patel, N. (2013, August 26). How To Hide App From Launcher And Launch It Through Dialer. Retrieved from Techno Nutty: http://www.technonutty.com/2013/08/howto-hide-app-from-launcher-and.html
[13] Schroer, W. J. (2015, 12 28). Generations X,Y, Z and the Others. Retrieved from The Social Librarian:http://www.socialmarketing.org/newsletter/features/generation3.htm
[14] SKMM. (2015). HandPhone Users Survey 2014. Cyberjaya: Malaysian Communications and Multimedia Commission (MCMC). Retrieved from http://www.skmm.gov.my/skmmgovmy/media/General/pdf/Hand-Phone-User2014.pdf
[15] Spy Agent. (2015, 12 29). SPY AGENT - stealth and undetectable PC monitoring software. Retrieved from Spy Agent: http://www.spyagent.net
[16] WebTutsDepot. (2015, 12 29). Android Tutorial: How To Post Data From An Android App To a Website. Retrieved from WebTutsDepot:http://webtutsdepot.com/2011/11/15/android-tutorial-how-to-post-data-from-anandroid- app-to-a-website/
[17] Winder, D. (2009, 12 24). Does parental control software work? Retrieved from Alphr:http://www.alphr.com/features/354349/does-parental-control-software-work