Handy warning:

Android Collaborative Applications to keep Children safe

Piotr Masior

B00064347

0879354688

B00064347@student.itb.ie

A dissertation submitted to the Institute of Technology Blanchardstown
in accordance with the requirements of the degree of BSc.H in Computing
in the Faculty of Science of Computing.

Course title: Bachelor of Science (Honours) in Computing (BN402)

May 2016

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of the study leading to the award of Bachelor of Science in Computing in Information Technology in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfillment of the requirements of that stated above.

Dated: 28 April 2016

Author: Piotr Masior

Abstract

This project attempted to develop an Android Collaborative Applications to keep children safe and that would help a guardians to observe their children, to establish their location, and mainly to provide an easy way to notify guardians about potentially dangerous situations.

The child's application must be capable to send alarm message with the location of the Android device (call child's device) and send it to another device (guardian's device) by way of a message with the exact location. The child's device has three options to send the alarm: by pressing notification bar, by shaking a phone, and by screaming (speaking) "help" to the phone.

The guardian application is able to receive and store an alarm messages from the child's device in the guardian's device. The guardian application after having received the alarm message will get a notification with Google Map, with child's device location on it, and name of the place (street, town).

Also on the screen the app will show two buttons. One button provides a link to emergency call 112 to get appropriate contact with an ambulance, the fire brigade, or the police (An Garda Siochana). The second button provides the link to call back a child with just one click. An application, child's app and guardian's app, it works on devices such as smartphone, with using the Android operating system.

Keywords: Android, Sensors, GPS, GSM, Speech Recognizer.

Acknowledgements

I would like to thank my supervisor Dr. Simon McLoughlin who was a tremendous support, and who gave insight which greatly helped in the direction of the development process.

Table of Contents

TABLE OF FIGURES

Images

Tables

INTRODUCTION

Introduction

The aim of this project was to create two different Android Applications, one for children (or anyone who needs to legitimately monitor the activity of another person) that allow children to have emergency contact with their guardians in situations requiring immediate attention, and secondly for their guardians, to receive the alarm message with the map showing a child location on it. The main goal of this project is to make it possible to collaborate between these two applications on Android devices, in order to support guardians and their children in their normal day life. Cooperation is based on monitoring of people whose security is essential and notification to the guardian about potentially dangerous incidents.

This thesis covers the areas of Mobile Application, Location Based Services and children in public places.

There are a large number of books and articles on the subject associated with this project, which provide necessary knowledge for programmers. These books and articles, are necessary and beneficial for those who would like to acquire knowledge on how to build an Android application, which will use all the hi-tech technologies and be significant for users like children and their guardians.

Research of literature, carried out as part of this project, shows that technologies like Java[1], SQL(sqlite)[2] , and GCM[3] have the biggest impact on the Android [4] software applications. Android Studio[5], the official integrated development environment for Android applications, based on IntelliJ IDEA[6] ,also plays a very important role.

---

[1] Dietel, Paul. *Java how to program*. PHI, 2009.
[2] Hacigümüş, Hakan et al. "Executing SQL over encrypted data in the database-service-provider model." *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* 3 Jun. 2002: 216-227.
[3] Developers, Android. "Google cloud messaging for Android." *2014-02-03]. http://developer. android. com/google/gcm/index. html* (2014).
[4] Developers, Android. "What is android." 2010 (2011).
[5] "Android Studio Overview | Android Developers." 2014. 14 Apr. 2016 <http://developer.Android.com/tools/studio/index.html>
[6] "IntelliJ IDEA the Java IDE - JetBrains." 2006. 14 Apr. 2016 <https://www.jetbrains.com/idea/>

This thesis has been compiled to investigate the manner in which all the technologies mentioned above can collaborate efficiently together. Also, the research shows (Playing Outdoors)[7]what dangers lie in wait for children in public places and how useful and important a role an Android application can have for children and their guardians. This thesis describes the relationship between the segments of the project and the importance of new technologies.

[7] Tovey, Helen. *Playing outdoors: Spaces and places, risk and challenge*. McGraw-Hill Education (UK), 2007.

Background

Monitoring and concerns about the safety of our children in a public place is an essential part of everyday life. Whenever and wherever our children are, it is vital that we know if they are safe. In the case of the appearance of danger, we must be able to know about this situation and collaborate with our children to help them and solve their problems. Being a parent poses daily challenges due to dangers which may be lurking in a public place. This project aims to create a collaborative application which allows children and their guardians to co-operate and keep children safe.

The child application, call "Guardian Angel", must be capable to send the alarm message with the geographical location of the child's device to a guardian's application, call "Guardian". Guardian application must be capable to notify a parent and display the coordinates with a location on the Google Map of the child device ("Guardian Angel").

The guardian application will be able to record and monitor the children's device as long as the guardian will need it. The guardian application will have a button which provides a link to emergency call 112 to get appropriate contact with an ambulance, the fire brigade, or the police (An Garda Siochana[8]). Both applications, will work on Android devices like smartphones and smartwatches with GPS[9] and GSM network, which are required for full operations.

Currently I have found one software application similar to that proposed in my project (Safe Kids– Kaspersky Lab[10]), but my project will stand out in that it will have the following additional features: Monitoring the location of the Android device (child's device) displayed on the map on guardians' smartphones, children can alert their guardians about potentially dangerous situations immediately. The option exist for guardians to directly contact emergency services (112).

This project will help any family to provide a sense of safety in their own domestic life.

---

[8] Siochana, An Garda. "Garda Síochána policy on the investigation of sexual crime crimes against children child welfare 2010." (2011).

[9] Kaplan, Elliott, and Christopher Hegarty. *Understanding GPS: principles and applications*. Elliott Kaplan & Christopher Hegarty. Artech house, 2005.

[10] "Kaspersky Safe Kids | Cyberbullying Protection | Kaspersky ..." 2015. 14 Apr. 2016 <http://www.kaspersky.com/safe-kids>

Main Research Questions

The general concept of this project leads to reflect a few main research questions:

- How mobile applications can help with the safety of our children?
- How will guardians keep track of their children?
- How can children collaborate with their guardians to be more secure in public place?
- How can guardians help their children in danger situation?

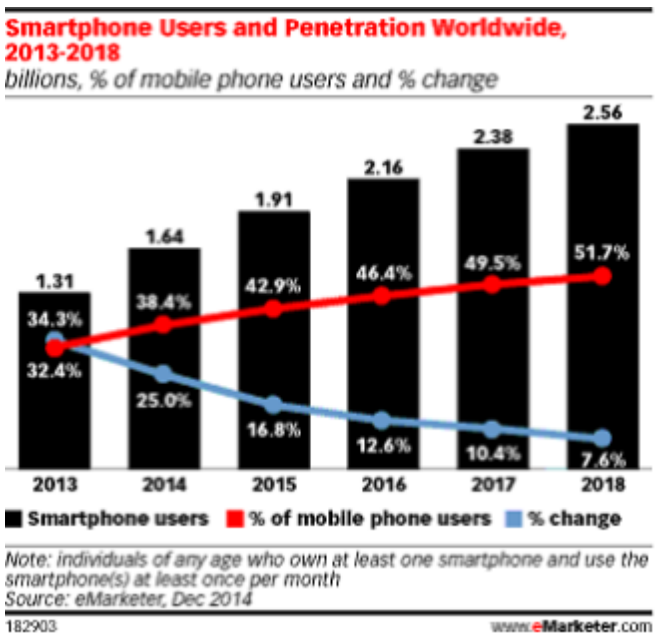


Image 1: Smartphone Market

Mobile devices have a big impact in every day of our life today, they are an inseparable part of our lives. The number of smartphone users worldwide will surpass 2 billion in 2016, this is more than one-quarter of the global population. With the growth of these devices, is also increasing their high tech technology. The mobile devices' hardware allow a software engineer to create very useful and significant applications, which greatly facilitate daily life.

Using these technologies it is possible to create Android smart-phone application which can provide an important tool to *help with the safety of our dears* in the public place.

Using the technologies like The Global Positioning System (GPS) the guardians are able to keep track of their children, when their children are outdoor with a device equipped

with GPS . The guardians can *keep track of their children* device's location by monitoring GPS signal on their phones, and displaying their geographical coordinates on the guardian smartphone.

On the other side, children are able to send a message to their guardians, that they are in danger situation. These children can send this message discreetly by pressing widget, or by shaking the device, or shouting for help. The message can be sent by Google Cloud Messaging (GCM) or Global System for Mobile Communication (GSM[11]) technologies to guardians and contain message with a location on the map. All these technologies like a text message, GCM (Google Cloud Messaging) Wi-Fi[12], an audio recorder, a GPS, an accelerometer and gyroscope can significantly improve the message and provide an accurate view of a danger. In this way children are able to *collaborate with their guardians to be more secure in public place*.

The application on the guardian device can have an emergency button which provides a link to emergency call 112, to get appropriate contact with an ambulance, or the fire brigade, or the police (An Garda Siochana). In situation that does not require to an emergency call, the guardians may call back to the child with one click (button) and *help his child in danger situation* remotely or collect him from the place of the problem.

[11] Mouly, Michel, Marie-Bernadette Pautet, and Thomas Foreword By-Haug. *The GSM system for mobile communications*. Telecom publishing, 1992.

[12] Alliance, Wi-Fi. "Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks." *White paper, University of Cape Town* (2003): 492-495.

Justification/Benefits

This project will be done to provide the implementation of all necessary feature and to gather all-important knowledge to develop the collaborated applications. Guardians and their children will derive the greatest benefits from the Android applications. The benefits are that it will provide the guardians with a handy and smart tool to keep an eye on their children at all times, but particularly when children are out of sight. It will be very beneficial to children too, given them a feeling of security when they are without guardians and allow them to be more independent.

The children device (“Guardian Angel”l app) must be capable to send an alarm message to parent device ("Guardian" app) by way of a message (GCM, or GSM) with the geographical location.

The guardian application will receive an alarm message and display the location of the children device. The guardian application will be able to record and monitor the children's device as long as the supervisor will need it. The guardian application will have a button which provides a link to emergency call 112 to get appropriate contact with an ambulance, the re brigade, or the police (An Garda Siochana). An applications, will work on smartphones with Android system. In a short steps and a quick time every user should be ready to use an Android application.

The main goal of the project is to make it possible to collaborate between two or more applications on Android devices to provide for the monitoring of people whose safety we want to safeguard and to send an alert about potential dangerous incidents.

Other benefits are: The ability to directly contact emergency services (112, on the child application), establishing and recording the voice from the children's device (camera shot optionally), keeping track of the child(s). This could be done from anywhere in the world, at any suitable time, once they have mobile connection with GPS The main goal of the project is to increase the safety of the loved ones, by make it possible to collaborate between two or more applications on an Android devices. The goal is reached by providing the monitoring of people whose safety we want to safeguard and to send an alert about potential dangerous incidents.

After the investigation of details and the resources needed to complete the project have been predefined, it is clear that existing technology and resources allow the project to be technically feasible. Most technologies and resources are free and widely accessible.

LITERATURE REVIEW

Introduction

This literature review covers the areas of Mobile Application[13], Location Based Services[14] and children in public place.

There are a large number of books and articles on the subject associated with this project, which provide necessary knowledge for programmers. These books and articles, are necessary and beneficial for those who would like to acquire knowledge on how to build an Android application, which will use all the hi-tech technologies and be significant for users like children and their guardians.

Research of literature, carried out as part of this project, shows that technologies like Java, SQL and Database have the biggest impact on Android applications. Android Studio, the official integrated development environment for Android applications, based on IntelliJ IDEA , plays a very important role.

This literature review has been compiled to investigate the manner in which all the technologies mentioned above can collaborate efficiently together. Also, the investigation shows what dangers lie in wait for children in public places and how useful and important a role an Android application can have for children and their guardians. This Literature Review describes the relationship between the segments of the project and the importance of new technologies.

[13] Holzer, Adrian, and Jan Ondrus. "Mobile application market: A developer's perspective." *Telematics and informatics* 28.1 (2011): 22-31.

[14] Schiller, Jochen, and Agnès Voisard. *Location-based services*. Jochen Schiller & Agnès Voisard. Elsevier, 2004.

Mobile Applications

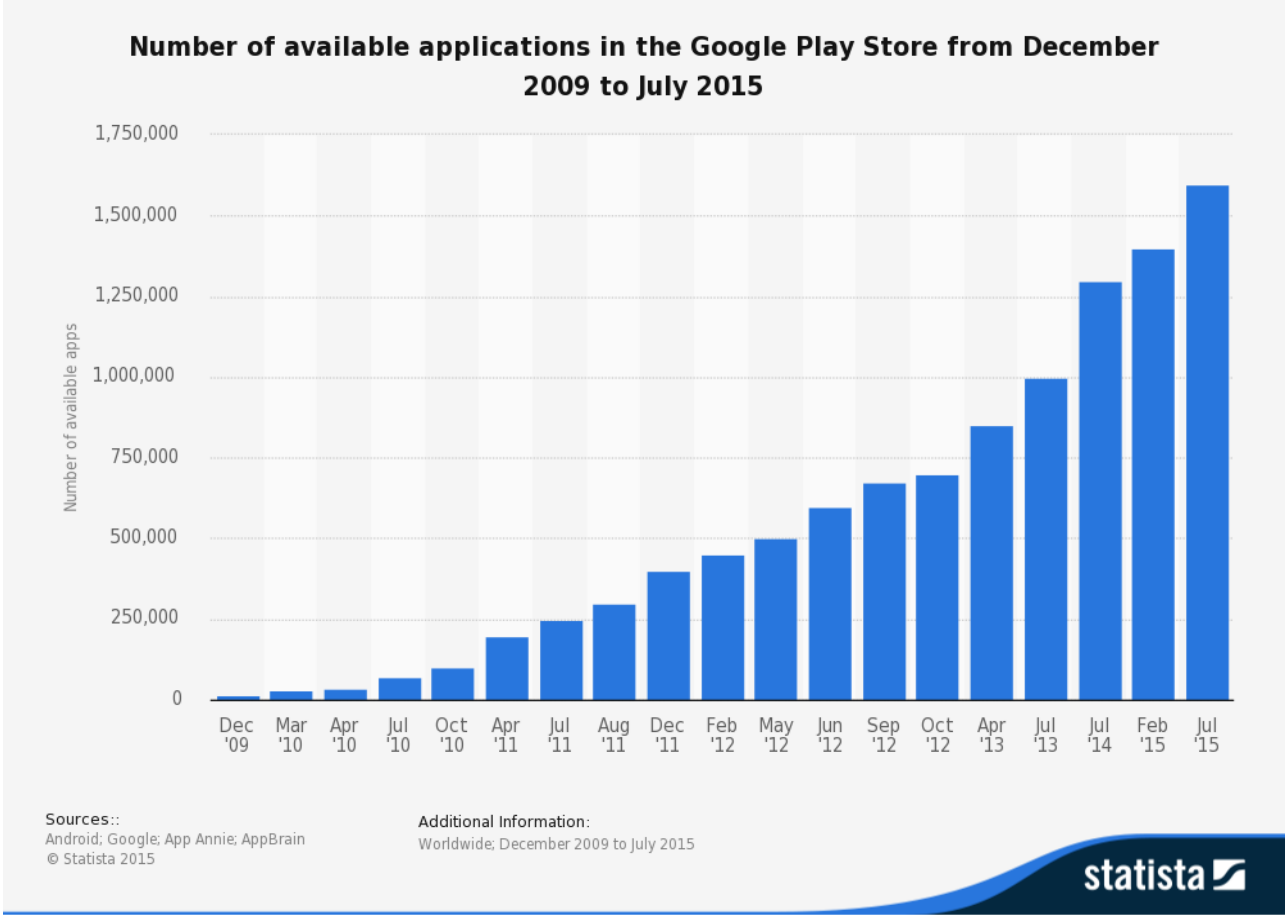


Image 2: Number of available apps in the Google Play Store from 2009 to 2015

A mobile application is a software program created to run on mobile devices such as smartphones, tablets or smartwatches. An analysis made by Danyl Bosomworth (Statista[15]) for Smart Insight website[16] shows that the number of mobile users overtook the number of desktop users in 2014. This shows how huge an impact mobile device applications have today.

Different devices have different platforms (operating system) on which an application runs. The Statistics Portal shows that the most popular platform among all mobile devices is the Android System, which was developed by Google. With the number of available apps

[15] "• Statista - The Statistics Portal for Market Data, Market ..." 2006. 27 Apr. 2016 <http://www.statista.com/>

[16] "Website design - Smart Insights Digital Marketing Advice." 2011. 14 Apr. 2016 <http://www.smartinsights.com/user-experience/website-design/>

standing at 1.6 million in Google Play Store in July 2015 (Illustration 1), the Android platform is ahead of Apple iOS[17], Windows Mobile[18], Symbian[19], Blackberry OS[20] and others. The Apple iOS is in second place with 1.5 million available applications in the Apple App Store in June 2015.

Android and Apple platforms have the largest portion of the mobile market in the world. A comparison between these is available in the following article: "Smartphone Development: iOS vs Android" The authors of the article compare the two platforms, and address the question: "which should faculty teach – iOS or Google Android?". This question is strongly relevant to the project and in the author's opinion, affirms the conviction that an Android platform is the best one to develop a mobile application. Comparisons of the two platforms are effectively summarized in

Table 1: *Comparison of iOS vs Android*

	iOS	Android
Minimum Operating System Requirement for Development	Macintosh computers running Mac OS X 10.6 (Snow Leopard)	Windows Linux Mac OS
Hardware	Requires a specific type of hardware (Apple brand)	Any modesty equipped device among many brands, like Samsung, HTC, Sony
Language	Objective-C	Java (Dalvik VM) Scripting (SL4A) LogoBlocks (AppInventor)
IDE	Xcode	Eclipse (requires Android SDK plug-in) Android Studio
Garbage Collection	Not available for performance reasons	Available

[17] Apple iOS, Android Extended Lead While. "Market Share Growth Paused." *Gartner Says Worldwide Sales of Mobile Phones Declined* 2.

[18] Grønli, Tor-Morten, Jarle Hansen, and Gheorghita Ghinea. "Android vs Windows Mobile vs Java ME: a comparative study of mobile development environments." *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments* 23 Jun. 2010: 45.

[19] Ortiz, C Enrique. "An Introduction to the Symbian OS™ Platform for Palm OS® Developers." *Metrowerks Corp* 21.

[20] "BlackBerry OS 10 – BlackBerry 10.3 OS Software Features ..." 2013. 27 Apr. 2016 <http://us.blackberry.com/software/smartphones/blackberry-10-os.html>

GUI Creation	Xcode	XML
Simulator	iOS SDK bundled GPS and Accelerometer not supported	Android Virtual Devices (AVDs) that run on the Android Emulator GPS and SMS interrupt signals that can be proceed to the emulator through telnet connection Accelerometer, coordination and compass examination can be manage using OpenIntents Sensor-simulator
Graphics	OpenGL (support for 2D and 3D)	OpenGL (support for 2D and 3D)
Database	SQLite	SQLite
Tutorial Resources	Apple's technical documentation provides vast information	From Google website is available official development guide tutorial
Reference Website	https://developer.apple.com/ios/download/	http://developer.Android.com/intl/pt-br/index.html

Table 1: Comparison of iOS vs Android

The author think that this article, "Smartphone Development: iOS vs Android[21]", describes very well the advantages and disadvantages of the platforms compared. The article states that computing on cell phones is rapidly growth, and unmindful of software platform, opinion on the subject will probably be oversubscribed. Each of two, iOS or Android will empower staff to exhibit the key ideas of mobile computing; support understudy programming abilities that will serve them in great stead paying little heed to what platform they wind up coding on;

Based on the research performed for this project, the author would likely recommend Android as the better OS for the following reasons. The source code is released by Google

[21] Goadrich, Mark H, and Michael P Rogers. "Smart smartphone development: iOS versus Android." *Proceedings of the 42nd ACM technical symposium on Computer science education* 9 Mar. 2011: 607-612.

under open source license, has more functions, better support, is easier to develop, and is the most popular mobile platform in the world.

Location Based Services

Location Based Services, in short is called “LBS”, and is all about how a user application is integrated with geographic location service. Nowadays most of the smart devices are implemented with this technology, which can prove their usefulness, and quality, in many real-life situations such as car navigation systems, web services and tour planning. There are also drawbacks, such as those associated with sharing by users of their location, with untrusted servers. This fact led developers to create techniques to limit the risks in such situations.

One of the solution is “a user-defined privacy grid system, called, dynamic grid system (DGS)” as proposed by R. Schlegel[22]. This solution covers the most important requirements, including continuous broadcasting of the user's location and contains the technologies necessary to secure the user's privacy. The system is based on four significant rules. First rule say that semi-trusted applications do not use a user geographic coordinates. The next rule refers to guaranteed privacy of user location under the DGS's 'defined adversary model'. A further rule states that communication costs for the user, rely on the points of interest near the user, and are not linked to the user's privacy level settings. DGS's final rule describes potential scaling of their services to support queries found in a greater physical area which would not interfere with the applications and database server processes of the semi-trusted system being accessed by the user. The authors of the DGS proposal conducted an experiment which they claim proves that their proposal would be better for users than the current techniques being used with LBS.

[22] Schlegel, Roman et al. "User-Defined Privacy Grid System for Continuous Location-Based Services." *Mobile Computing, IEEE Transactions on* 14.10 (2015): 2158-2172.

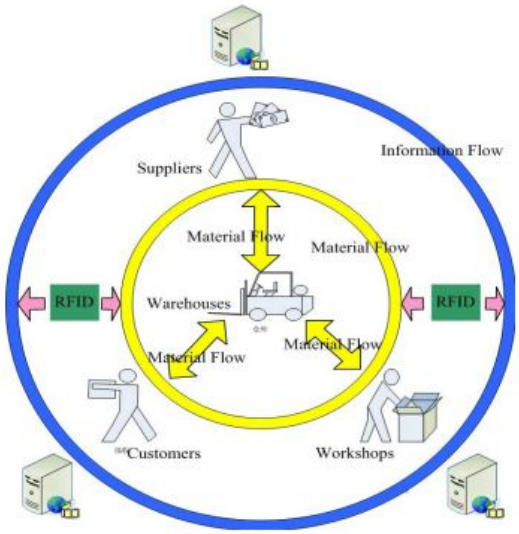


Image 3: Prototypical intelligent RFID guided vehicle used in the warehouse

Location Based Services usually uses Global Positioning System (GPS) for tracking a location on the earth, which naturally is necessary for geographic location. GPS is the most popular today, but beside it are its European equivalent GALILEO[23], and Russian equivalent GLONASS[24]. These systems all rely on satellites orbiting the earth. The inconvenience of these systems is that they are not able to report a user location inside roofed facilities. Radio Frequency Identification (RFID)[25] is a solution to this issue using wireless communication technology. This technology was clearly described at the International Conference in China, 2007 in the paper 'A RFID-based Material Tracking Information System' [26]. Authors of this publication introduced in their proposal two methods for the user's device to communicate with the RFID. Their work covers both the architectural solution (hardware) and function of the client application (software). One method is to build an RFID middleware server which will automatically record and store the motion of an object/device and track its location. The identification of these objects or devices can be done via tags attached to the object or embedded within the device, which communicate wirelessly via antenna. RFID is quite attractive from a cost point of view and is quite an efficient technology. The main concept of

[23] Swann, John, Eric Chatre, and Daniel Ludwig. "Galileo: Benefits for location based services." *Positioning* 1.04 (2009): 0.

[24] Daly, P. "Navstar GPS and GLONASS: global satellite navigation systems." *Electronics & Communication Engineering Journal* 5.6 (1993): 349-357.

[25] Finkenzeller, Klaus, and RFID Handbook. "Radio-frequency identification fundamentals and applications." *Chippenham: John Wiley & Son* (1999).

[26] Min, Zhang et al. "A RFID-based material tracking information system." *Automation and Logistics, 2007 IEEE International Conference on* 18 Aug. 2007: 2922-2926.

this paper is to encourage the use of RFID as a solution for material reporting information systems using databases and software applications.RFID has an important role to play in LBS, but is only one of a number of technical solution which could be applied to customize LBS.

Children in public place

The environment in which children are growing up today is more dangerous than was the case years ago. Even playing outdoors carries risks and challenges. “There is always a certain risk to being alive and if you are more alive there is more risk”[27] The author of this sentence is saying that there is no guarantee of security in day to day life, particularly for children when unsupervised by parents or guardians. Danger lurks, especially outside of the home, school and other public domicile. For example, from bullies while at play in the playground, strangers while on the way home from school or even going to the local store.

Defining the potential risks hidden openly in public places could help to avoid such problems and even dangerous situations. Department of Children and Youth Affairs[28] defined four recognised forms of child abuse:

- Neglect
- Physical Abuse
- Emotional Abuse
- Sexual Abuse

The HSE reported, that in 2012 the total number of children abused nationally was 40187.<9>

The proposal contained within this project will hopefully go some ways towards reducing the above figures in the future, and have a big impact on the influence of parents and guardians on our children's security and happiness. Happy children make happy parents.

[27] Tovey, Helen. *Playing outdoors: Spaces and places, risk and challenge*. McGraw-Hill Education (UK), 2007.

[28] "Home Page - Department of Children and Youth Affairs ..." 2011. 14 Apr. 2016 <http://www.dcya.gov.ie/>

Review of existing work

Currently I have found one software application similar to that proposed in my project it is Safe Kids – Kaspersky Lab[29]. Kaspersky Safe Kids help to manage and monitor device usage, set rules for internet access and applications. This application is present in two versions - free and paid. Free version contains benefits like:

- Helps to manage a children's Internet activities, e.g. access types of website,
- To manage a children's use of applications, e.g. appropriate content & apps specify for their age
- To manage a children's use of devices

The paid version, except those listed above benefits has more options like:

- ➢ Keeps an eye on a child's location, which shows a child's location on a real-time map
- ➢ Makes it easy to monitor public Facebook activity
- ➢ Monitor Android calls & SMS[30] messages
- ➢ Sends you real-time alerts & notifications if they leave the defined safe area

My project will stand out in that will have the following additional features:

- ★ Children can alert their guardians about potentially dangerous situations immediately when need it
- ★ Displaying on guardians smartphones the Google map with a children's location on it and address (free version)
- ★ The option for guardians to directly call an emergency services (112)
- ★ The option for guardians to directly call their child
- ★ Two language version (English, Polish)

[29] "Kaspersky Lab US | Antivirus & Internet Security Protection ..." 2012. 14 Apr. 2016 <https://usa.kaspersky.com/>

[30] Le Bodic, Gwenaël. "Short Message Service." *Mobile Messaging Technologies and Services: SMS, EMS and MMS*: 35-116.

Conclusion

After researching Mobile Application, Location Based Services and children in public places, it is recognized that the most important thing is to acknowledge the potential dangers which can befall children, of any age, while unsupervised, and then use the latest technical innovations to address these issues.

In this project the biggest impact will be the use of Java computer language, together with Android Studio, Database and mySQL. Android Collaborative Application to keep Children Safe is an important application that allows guardians to increase the safety of their children as well as help those children to feel more secure when they are without guardians while allowing them to become more independent. The interface of this app should be easy to setup and easy to handle.

Main features:

- monitoring the location of the Android device (child's device) displayed on the map on
- the guardians smartwatch
- children can alert their guardians about dangerous situations immediately

In conclusion the project will create an application with technology and resources that are widely accessible, making the project technically feasible and useful for families and people who are concerned about the security of their children or others who might be considered to be vulnerable.

METHODOLOGY

Development Model

The handy warning system is split into two collaborative Android applications. One app is designed for guardians, the second app is dedicated for the child.
Guardian (protector) app by function, can be split into three main sections and every of these sections can be complemented with optional extra functions.

A basic and necessary function is a possibility to add a child("ADD PERSON") to the system. This possibility could be canceled at any time of the using it(during adding a person) or submitted to the memory of the phone. The method used to save a child is Android SQLiteDatabase. SQLite is an Open Source database which like other databases support standard SQL syntax. The database requires constrained memory at runtime (approx. 250 KByte[31]) which is enough to be used in other runtimes. SQL data types in Android are different than in Java. Example: in Android instead of String[32] is TEXT, Integer is similar to "long" in Java and Real acts as double in Java. The rest of the types must be converted into proper data types, before being used in database.

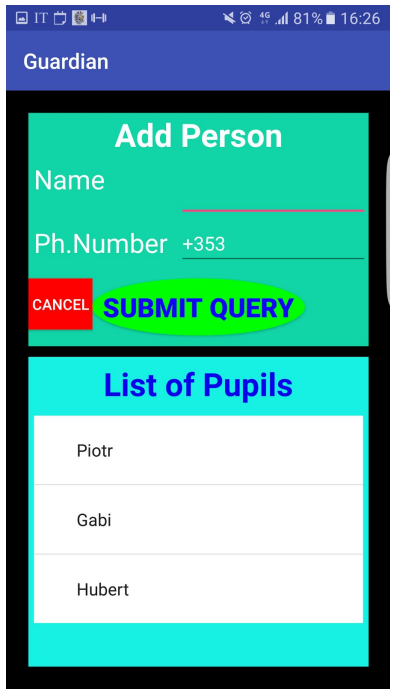


Image 4: Guardian App Main Screen

Another function is a "List of Pupils", where a user can manipulate ("UPDATE") the data inserted into the system, therefore can change a name or phone number. In this function is found a very important feature, an option to send a short message service (SMS) with GCM registration id, which is crucial for communication between both applications.

Without this registration id, the child app won't send an alarm message to a desirable receiver. To display the list of children is used Android ListView class that shows items in a vertically scrolling list. ListView must implemented the ListAdapter class, which is associated with this view. The ListAdapter have to contain ArrayAdapter with ArrayList.

[31] Dahm, Markus. "Byte code engineering." *JIT'99* (1999): 267-277.
[32] Arnold, Ken et al. *The Java programming language*. Reading: Addison-wesley, 2000.

Last function is less important, but very handy. It is a history of the alarms which have occurred since the app was set up. In this "History" could be found list of children, and list of the events. Every element of these lists can be displayed with a map and address, to remind where a dangerous situation occurred. This function contains a snackbar to display option for the list. The snackbar is an interesting Android material design component, and works just like Toast messages expect they provide action to interact with. Another interesting feature of Snackbar is that the component is displayed at the bottom of the app and is swipeable to remove them.

The child app, also called "Guardian Angel", can be split into two main functions. First function allows a child to add a Guardian and list them on the screen. Like in Guardian app, the user has ability to add a guardian, and the "List of Guardian" allows the user to update the name and phone number of a guardian. Almost the same classes are used to display a list and interact with them, like in "Guardian" app.

It is worth mentioning here about hidden action, which is part of the "List of Guardian"function. This action reads SMSs from people listed in the guardian list, and save the string with GCM registration id, if this registration id is sent. This approach activates the guardian in the list. This activation is necessary to allow the application to find and send a message to appropriate guardian. To make possible to read the SMSs this application must contain BroadcastReceiver class, which works like human ear, listening all the time.

Image 5: GuardianAngel Main Screen

This class also checks if the sender is on the "List of Guardian" and read only his text messages. If the text contains in first token "naidraug", than the registrationId is saved to database. The second and last factor is also very important. The "Select the alarm methods" activity allows a user to make decision how should he trigger the alarm message. Three options occur there. One is "Shaking", which allow a user to shake a phone to send the alarm message. Another option is "Screaming". Activated "Screaming" function continuously listen

and react for a loud speech. If that happens, the app recognize if the words spoken were “help”, “help me” or “help me please” and if this is true, send the alarm message. Last option in this function is “Pressing”, which simple lets a user press a Guardian Angel notification on the top of the smartphone, and send the alarm message. To make this option work, the application must have components like Android CheckBox and Android button to start selected services.

When “Shaking” option is selected the system involves a service class which works on the same thread as the main app and implements SensorEventListener. The method “onSensorChanged” checks the sensor events and reacts on exceeded threshold by starting another service (IntentService) to send an alarm message.

When “Screaming” option is selected the system involves a couple of classes, which work on separate thread. The classes used object like AudioManager, MediaRecorder[33] and SpeechRecognizer Google API. All classes together provide a service which first checks the level of noise, then if the noise exceeded the threshold the SpeechRecognizer check the spoken words. If is spoken word: “help”, the service calls another service to send an alarm message.

When the last option, “Pressing” is selected, it simply creates a notification using Android Notification object, Intent and PendingIntent. The PendingIntent is used to set an action, which being pressed calls the service to send an alarm message.

Usability of both applications will be addressed through Graphical User Interface designs optimised for ease-of-use and simple aesthetics. I will design my own use case diagrams using the Unified Modelling Language to determine the software's functionality.

A Push Notification will be used to communicate between “Guardian Angel” and “Guardian” app. A push notification is a message that originates from a server to a client, eg. pops up on a mobile device. Similar to SMS text messages, push notification broadcasts a message to reach users who are designated to receive the piece of information. Different platform and different devices may use different push notification. Nowadays, push notifications are the most common type of messages on mobile platforms and the biggest impact on propagation of it have Google[34], iOS[35], Fire OS[36], Windows [37]and BlackBerry[38].

---

[33] "MediaRecorder | Android Developers." 2014. 27 Apr. 2016 <http://developer.android.com/samples/MediaRecorder/index.html>

[34] Battelle, John. "The search: How Google and its rivals rewrote the rules of business and transformed our culture." (2005).

This technology allows smartphones to receive and display an alert message, eg. from social media like Facebook [39]even if the OS device's are sleeping and even when the social media app that is pushing the notification is closed.

Once the functionality have been described, a prototype will be implemented using the following technologies:

- Java
- Android APIs
- XML (EXtensible Markup Language)
- SQL(Structured Query Language)
- GCM (Push notification)

These technologies have been chosen due to the research and experience gathered during the academic year. Upon successful completion of the prototype, further iterations will be refined during the testing phase. Test-user feedback will then be examined for further modifications.

[35] Tracy, Kim W. "Mobile Application Development Experiences on Apple?? s iOS and Android OS." *Potentials, IEEE* 31.4 (2012): 30-34.

[36] "Amazon's Fire OS vs. Google's Android: What's the ..." 2015. 27 Apr. 2016 <http://www.howtogeek.com/232973/amazons-fire-os-vs.-googles-android-whats-the-difference/>

[37] "Windows - Microsoft." 2012. 27 Apr. 2016 <http://www.microsoft.com/en-us/windows>

[38] "BlackBerry Enterprise Mobility, BBM, Smartphones ..." 2010. 27 Apr. 2016 <http://us.blackberry.com/>

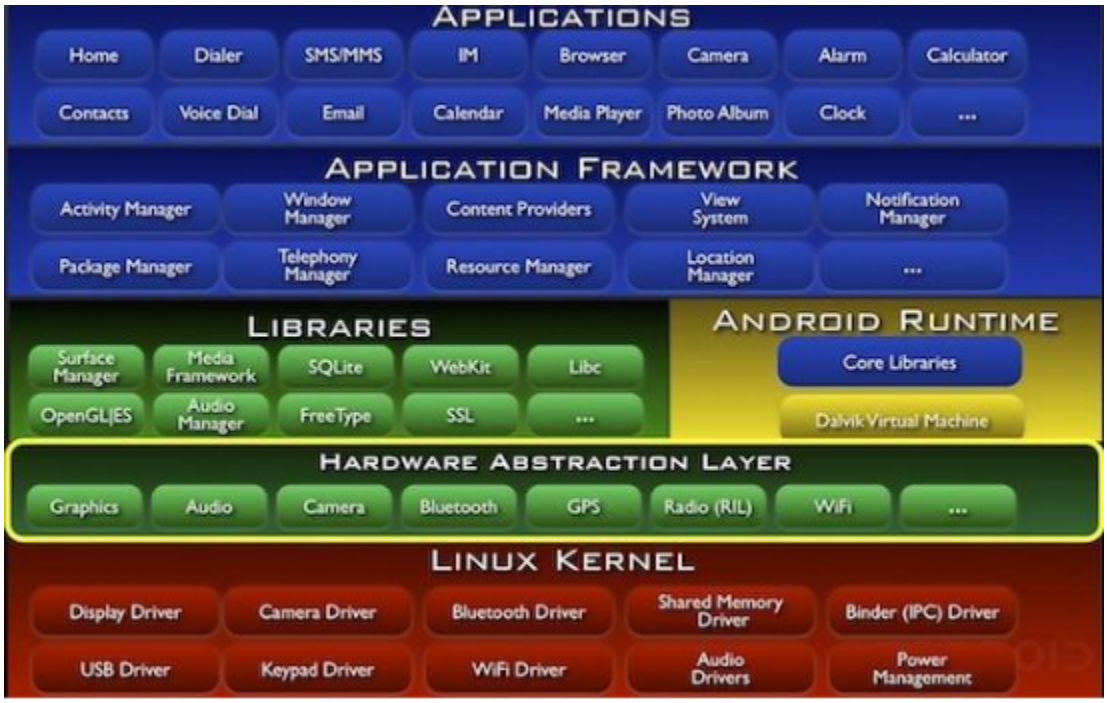
[39] Ellison, Nicole B, Charles Steinfield, and Cliff Lampe. "The benefits of Facebook "friends:" Social capital and college students' use of online social network sites." *Journal of Computer-Mediated Communication* 12.4 (2007): 1143-1168.

SYSTEM DESIGN AND ARCHITECTURE

Android App Architecture Overview

Android Application Architecture can be roughly divided into five software components. This architecture is design for mobile devices and includes an operating system, run time environment, middleware, services, libraries, and key applications. The software stack shown below in the diagram outlines four layers:

1. The Linux Kernel[40],
2. Android Runtime - Dalvik Virtual Machine[41] and Core Libraries,
3. Android Libraries,
4. Application Framework
5. Applications.



Android App Architecture

Image 6: Android Architecture

[40] Bovet, Daniel P, and Marco Cesati. *Understanding the Linux kernel*. " O'Reilly Media, Inc." 2005.
[41] Ehringer, David. "The dalvik virtual machine architecture." *Techn. report (March 2010)* 4 (2010).

Guardian App Design

Given the description previously outlined, and the experience of using Android applications on a mobile phone with Android System allow evaluation of the concept of how the system should look. To take control of the software project, particularly to analyze and design an application, object-oriented analysis and design techniques with UML[42] tools are applied. UML stands for the Unified Modeling Language and is a general-purpose, developmental, modeling language.

Use Case Diagram

The first and probably the most common functional technique among UML tools is Use Case Diagram. The purpose of this diagram is to capture the functional requirements of a system. This diagram is split into three items:

1. Actor's (customers) in this case, human user - Guardian
2. Functionalities which represent what actors can do in the system (use cases)
3. Relationships clearly show dependencies between the actor and use cases

This concept is illustrated in an image below.

[42] Warmer, Jos B, and Anneke G Kleppe. "The Object Constraint Language: Precise Modeling With Uml (Addison-Wesley Object Technology Series)." (1998).

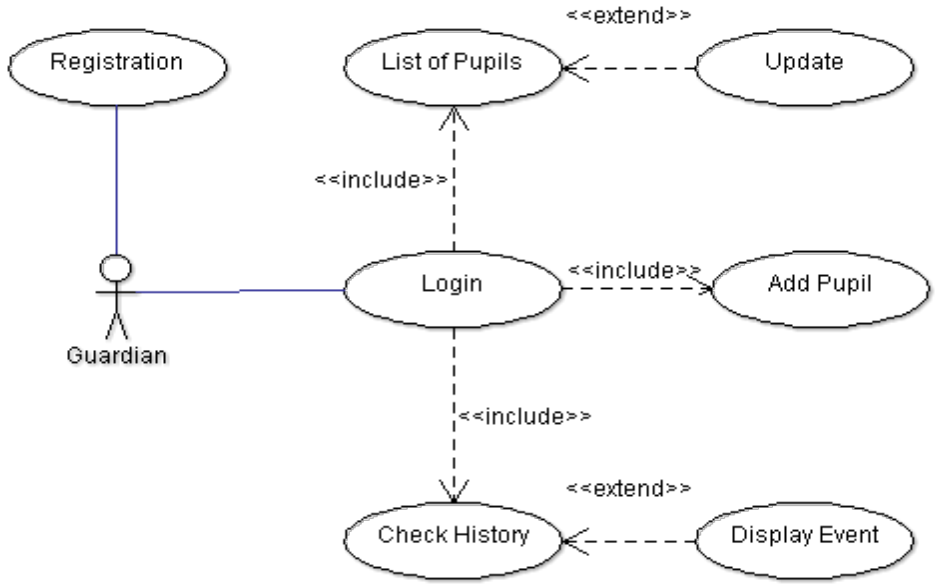


Image 7: Guardian Use Case Diagram

Use Case Diagram provides the following benefits:

- Allow to design the structure of the system (classes)
- Identify interaction among the actors and functionalities
- Show general overview of a system (used to test if all function are there)

Basic Flow

- The use case starts when a guardian wants to log into the System.
- The system requests for the first time to register guardian username and password (security level) in Registration Form.
- Each subsequent time, the system will be directed to the login screen,and ask for the correct username and password.
- The system validates the entered username and password and logs the actor into the system.
- It is necessary to add a person to the List of Pupil for collaboration with Guardian Angels app by the guardian

- The guardian must send GCM[43] registration ID to the child by clicking Update on the List of Pupil's (prior to this, child must have a guardian on his list to be able to save the ID)
- The guardian can be directed to the History list, and display events on a Google map.

Alternative Flows

If in the Basic Flow the guardian enters a wrong username and/or password, the system displays an error message.

The guardian can choose to clear the login preference, at login screen

The guardian can cancel adding a child in the Add Person screen box

The guardian can cancel updating a child in the Update screen box

Class Diagram

Use Case Diagram is usually the first attempt to build the software system. Another, interesting and also very important step in developing this application is Class Diagram. The Class Diagram purpose is to model the static view of a software. The class diagram is the main diagram which can be specifically mapped with class objects and therefore broadly utilized at the time of development. The class diagram seems to be the most popular among the coder community, because of its object oriented benefits. In fact the Class Diagram is not only a diagram, but can also be used for constructing executable script of the programming application.

Class Diagram provides the following benefits:

- Depicts obligations of a framework
- Investigation and outline of the static view of the software system
- Variable and method implementation
- Relationships and dependencies between the classes
- Depicting the functionalities performed by the framework.

[43] "Google Cloud Messaging - Google Developers." 2015. 27 Apr. 2016 <https://developers.google.com/cloud-messaging/>

The image below illustrates the Guardian Class Diagram

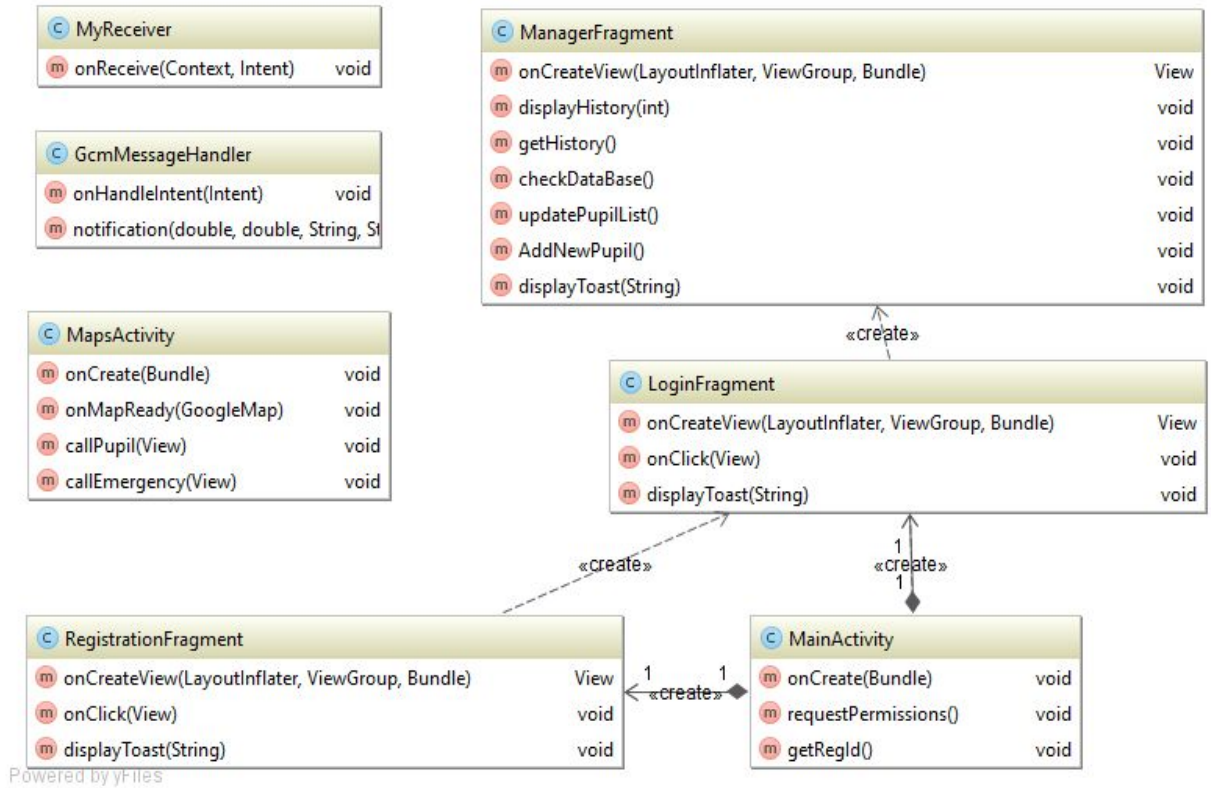


Image 8: Guardian Class Diagram

The diagram above displays 7 classes. Four of them are related by dependency. Others loosely associated with each other.

MainActivity[44] class is the main class in the app. This class create LoginFragment class and RegistrationFragment class. LoginFragment create another Android fragment - ManagerFragment class. ManagerFragment is the class where a guardian can find all the important functions of the software application.

Other classes like GcmMessageHandler (IntentService[45]) and MyReceiver (BroadcastReceiver) work separately in the background, waiting for a message from the child. They are only triggered if a message to this application is sent. Last class, MapActivity, is used to display Google Map[46] on the guardian screen. Like previous classes, is only triggered if the guardian wants to display a location of the child on the map. This can happen

[44] Vogel, Lars. "Android development tutorial." *Luettu* 16 (2012): 2012.
[45] Lee, Wei-Meng. *Beginning Android Tablet Application Development*. John Wiley & Sons, 2011.
[46] Kobayashi, Shinji et al. "A geographical information system using the Google Map API for guidance to referral hospitals." *Journal of medical systems* 34.6 (2010): 1157-1160.

after receiving a new alarm message (notification bar) or could be triggered from the History list.

It should be noted that the diagram also shows the method to store data in the phone. Most of applications need to save data, especially this app needs to keep recorded alarm messages, password and username. Due to this software the option to store data is “Saving Data in SQL Database”. Another very important method to spot is “getId” method. This method involve a separate thread, and is used to register application with project number on Google Cloud Messaging (GCM) and giving back a registration ID. More about processes occurring during the registration and communication in GCM, will be in the next chapter.

Sequence Diagram

The last diagram which is used for purpose of this software application is Sequence Diagram. Sequence Diagram together with a collaboration diagram are known also as Interactive Diagrams. The main difference is that a sequence diagram focused on time sequence of messages, but collaboration diagram focused more on structural organization of the objects.

Sequence Diagram provides the following benefits:

- Control of new message flow by time sequence
- Capture the order of flowing messages
- Understand dynamic behaviour of a system
- Forward and reverse engineering

The image below illustrate the Guardian Sequence Diagram

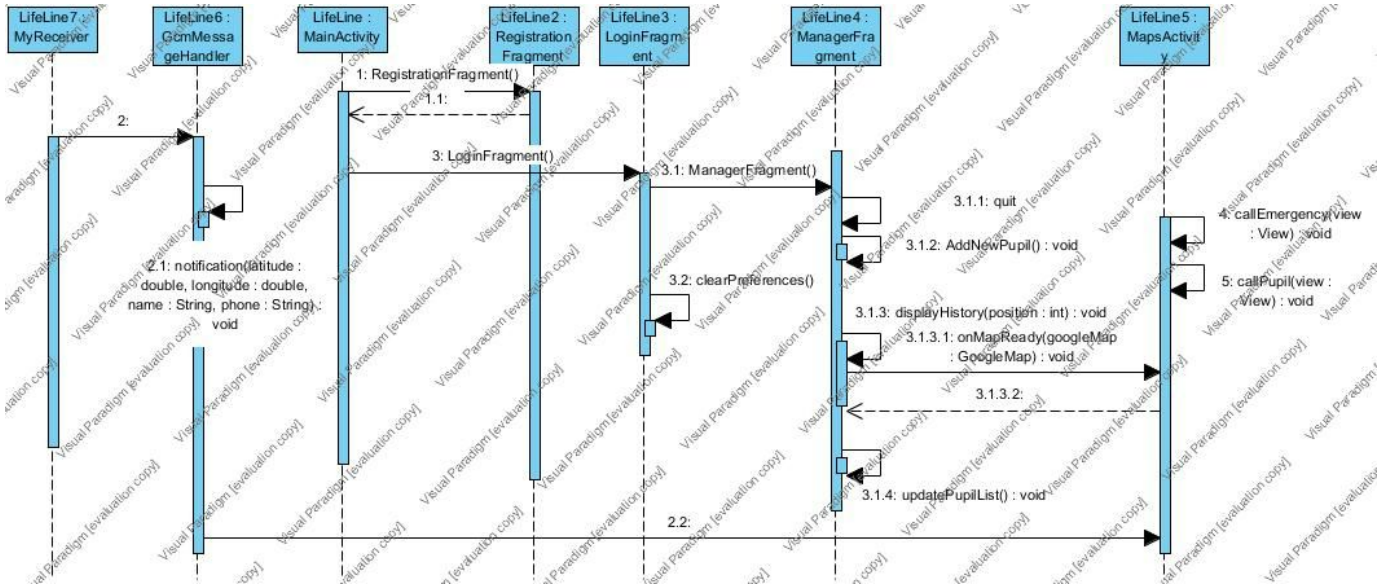


Image 9: Guardian Sequence Diagram

The sequence diagram above shows how the “Guardian” is working by the time sequence and the order of flowing messages. The first message of a sequence diagram always starts at the top. In this diagram, MainActivity sends first Registration message to RegistrationFragment. Usually the first message located is on the left side of the diagram, but in the image above, on the left side are placed the services: MyReceiver and GcmMessageHandler, which has not a relationship to a main class, but works all the time in the background, and triggers notification for a guardian if a gcm message arrived. Next subsequent message “login” is then added to the diagram slightly lower than the previous message. This message forwards to LoginFragment object, where there are other two messages. The flow shows that in the LoginFragment exists a loop message to clear preference and slightly higher, message to ManagerFragment. ManagerFragment is a frame with four loops and a message to MapsActivity. The four loop stages are - quit from program, add new child action, update child list, and display history of child with a link to MapsActivity. The MapsActivity has two functions, used to ring a child, or emergency 112 number. At every stage of the flow, the instance of the app can be killed by clicking Android back key.

Children (GuardianAngel) App Design

Use Case Diagram

The image below illustrate the GuardianAngel Use Case Diagram

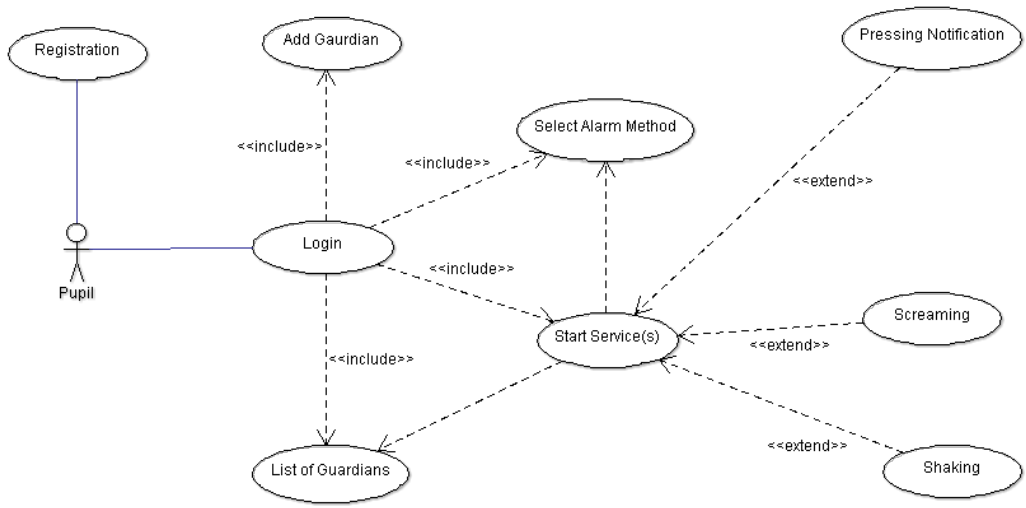


Image 10: GuardianAngel Use Case Diagram

Basic Flow

- The use case start when a child wants to log into the application.
- The system requests from the user, for the first login, to register child with username and password (security level) in Registration Form.
- Each subsequent time, the system will be directed to log in screen, to ask for the correct username and password.
- The system validates the entered username and password and logs the actor into the system.
- It is necessary to add a guardian to the guardian list for collaboration with Guardian application
- The child must received GCM registration ID from a guardian by a text message (SMS) and save it into “GuardianAngel” database (prior to this, child must have a guardian on his “List of Guardians” to be able to save the ID)

- A child can select an alarm method which would be used to send an alarm message
- A child must start the chosen service (prior to this, must have a minimum one guardian with gcm id)
- Started services are working in the background independently, and they can be turned off

Alternative Flows

Invalid Name / Password

- If in the Basic Flow the actor enters an invalid username and/or password, the system displays an error message.

The child can choose to clear the login preference, at login screen

The child can cancel adding a guardian in the Add Guardian screen box

The child can cancel updating a guardian in the Update screen box

Class Diagram

The image below illustrate the GuardianAngel Class Diagram

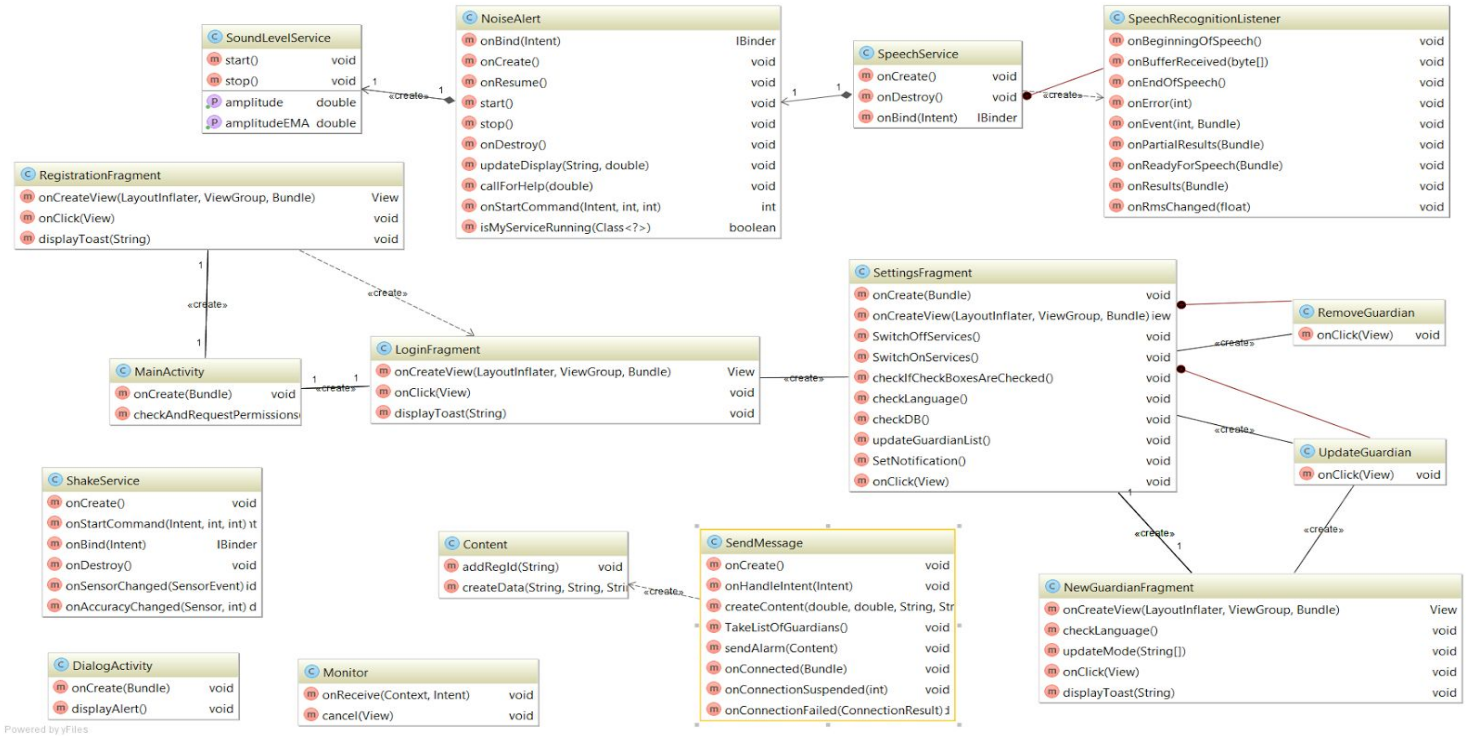


Image 11: GuardianAngel Class Diagram

The diagram above displays 13 classes plus 3 private classes. The main group with MainActivity class has seven related dependency classes. Others groups (packages) are GCM package, NoiseAndSpeechRecognizer, ReadSMS and ShakePackage associated with each other.

MainActivity class (main group) is responsible to create LoginFragment class and RegistrationFragment class. RegistrationFragment allows a user to put new username and password and save it into GuardianAngel database. LoginFragment is the security option with username and password field. The correct username and password create another Android fragment - SettingFragment class. SettingFragment class is an important class where a child can add guardian to the list (NewGuardianFragment class), select an alarm method and start services. Except that, SettingFragment has implemented two private classes, one to update a guardian, second to remove a guardian. Other packages and classes work separately in the background.

GCM package has Content and SendMessage classes. They are responsible to send an alarm message through Google Cloud Messages. SendMessage class has implemented GoogleApiClient to get last known location. Before the message is sent, the Content class creates data with gcm registration id (address) and uses java Map interface to put together all details (latitude, longitude, sender, phone number) into a message.

NoiseAndSpeechRecognizer package contains three classes, NoiseAlert, SoundLevelService, SpeechService plus one private class SpeechRecognizerListener. NoiseAlert class is a root class. This class creates a separate thread with SoundLevelService class, which constantly check the level of noise around the device with “GuardianAngel” application. When NoiseAlert gets callback from SoundLevelService that it has exceeded a certain level of noise, NoiseAlert call SpeechService class. SpeechService check then if “help” word is said(during next 20 seconds). If SpeechService recognized the “help” word, it calls SendMessage class and the SpeechService is defunct. NoiseAlert starts again after the time is gone or after the call is made.

ReadSMS package has two classes, Monitor (BroadcastReceiver) and DialogActivity. Monitor is responsible for listening to incoming “sms”. If the sender is a guardian, reads a message. If the message contain a gcm registration id, DialogActivity is called to display inquiry on the screen to ask user if the gcm registration id should be saved to the database.

ShakePackage contains only one class, ShakeService. This service class uses accelerometer sensor to monitor the location of the device. If the measurement exceeds a set threshold, the SendMessage class is called.

Sequence Diagram

The image below illustrate the GuardianAngel Sequence Diagram

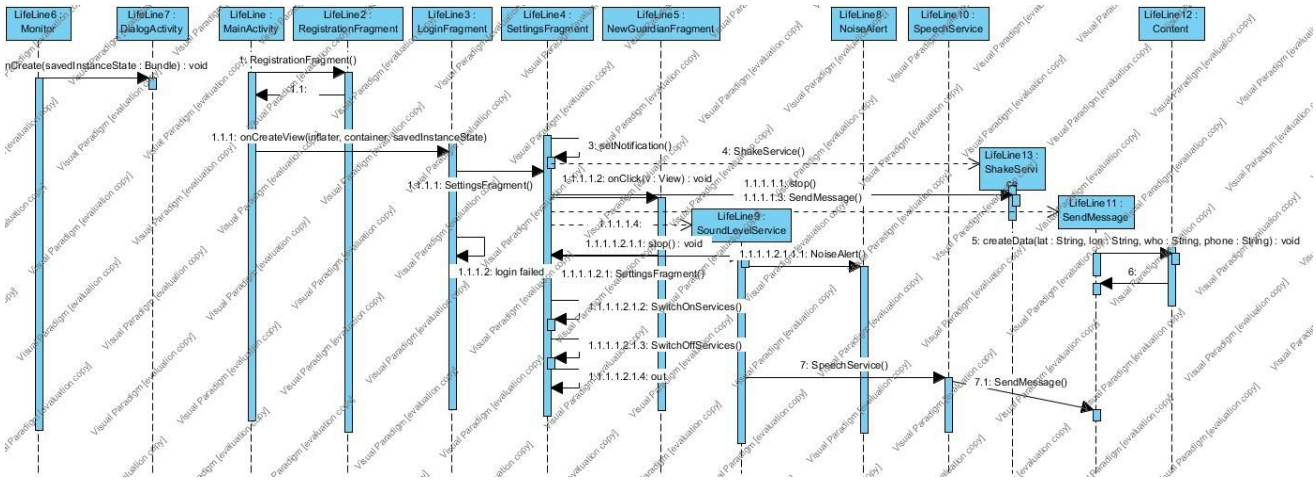


Image 12: GuardianAngel Sequence Diagram

The following sequence diagram above has 13 objects. The first call is getRegister() which is a method of RegistrationFragment with a return and always starts at the top. The next call is login() which is a method of LoginFragment object and owns a self call method if login failed. Slightly lower is message to call SettingsFragment object, where then is other few calls. addNewGuardian() is a first call on this lifeline to NewGuardianFragment. The self calls in this lifeline are getNotification(), switchOnService(), switchOffService(). There are also calls to services like ShakeService and SoundLevelService. SoundLevelService has two calls, first to NoiseAlert object and second to SpeechService. A getNotification() method and services, SpeechService and ShakeService, have calls to SendMessage. The flow shows that in the SendMessage is one more call to Content object with return.

Separate to the listed flow, exist two more objects, Monitor with call message to DialogActivity. Monitor object is monitoring all text messages (sms) and DialogActivity displaying alert on the screen if needs be.

SYSTEM IMPLEMENTATION

System Requirements

Hardware Compatibility

Android is an operating system based on the Linux kernel, and designed mostly for mobile devices, like smartphone, smartwatch, tablet and even Android TV.
Guardian and GuardianAngel applications are design for smartphones devices. It could also be run on the tablet, and on smartwatch after a little fix. But the main point of this project is to use the software applications on devices which can be easy and comfortably keep with every person, both adults and children. Best compromise is a smartphone, because of its advantages like mobile network, mobile internet, built in GPS antenna, sensors (accelerometer), long battery, small size. In fact smartwatch could suit better for children, primarily because its size (much smaller than smartphone), and not every parent (guardian) wishes their child to have a smartphone. But in today's market is not easy access to smartwatches with build in GPS antenna and mobile network. You can find few, but usually very expensive. So this is why smartphone with Android system is the best option today.

Unfortunately not every smartphone can run Guardian app and GuardianAngel app. It requires that the smartphone includes Android operating system. A typical example of a mobile device with Android platform is that the hardware components have a corresponding API for third-party developers. Also requirements for Android 4.4+ requires an ARMv7 processor[47], and 512MB RAM[48]. For a display and graphics Android device MUST properly implement the units referenced by the Android APIs requirements defined as follows:

- Physical diagonal size measured in inches between two opposing corners of the display.
- Dots per inch[49] (dpi). The number of pixels contained in a linear horizontal or vertical span. Both horizontal and vertical dpi must fall within the range.
- Aspect ratio. The proportion of the pixels of the one dimension to the second

[47] Abdurachmanov, David et al. "Explorations of the viability of ARM and Xeon Phi for physics processing." *Journal of Physics: Conference Series* 2014: 052008.
[48] "What is RAM (random access memory)? - Definition from ..." 2015. 27 Apr. 2016 <http://searchstorage.techtarget.com/definition/RAM-random-access-memory>
[49] "DPI (Dots Per Inch) Definition." 2015. 27 Apr. 2016 <http://techterms.com/definition/dpi>

dimension of the screen. For example “16:9”.

- Density-independent pixel[50][51] (dp) is calculated as: pixels = dps * (density/160); normalized to a 160 dpi screen.

All modern smartphones with Android OS are equipped with components which allow to instal Guardian and GuardianAngel application.

[50] "Supporting Multiple Screens | Android Developers." 2009. 27 Apr. 2016 <http://developer.android.com/guide/practices/screens_support.html>

[51] Meier, Reto. *Professional Android 4 application development*. John Wiley & Sons, 2012.

Software Requirements

Android is the most powerful mobile operating system, because is the most common system among mobile devices, and its source code is released by Google under open source licenses. This make Android very flexible and is eagerly developed by enthusiastic developers. Because Android is encountered on many mobile devices, like Samsung, HTC, Sony, Motorola and many, many more, it usually ships with a combination of open source and proprietary software made by the phone corporations.

So the software can vary considerably depending to the brand of mobile devices. For example Samsung has own app market "Galaxy Apps", with software applications designed particularly for Samsung smartphones.

Guardian and GuardianAngel are intended for mobile devices with Android 4.4 (KitKat)[52] and above. All devices, from all brands, with these versions of Android (last version today is Android 6.0 Marshmallow), are using components and software which allow the full exploitation and trouble-free operation of Guardian and GuardianAngel apps. In this place should be mention that Android 6.0 gives even now more control by implementing new app permissions (dangerous permission), and the battery life can last significantly longer. The image below show a comparison between two same mobile devices running two different operating system.

Comparison standby consumption Nexus 5 after X hours in percent			
	8th*	24 *	48
Android 5.1.1	4.0%	12.0%	24.0%
Android M DP 1	1.5%	4.5%	9.0%
** projected value*			
Extrapolated maximum runtime in standby			
Android 5.1.1	200 hours		
Android M DP 1	533 hours		

Image 13: Comparison standby consumption (battery life)

As you can see, the Marshmallow is 2.7 times longer! on standby mode.

[52] "Android – 4.4 KitKat." 2014. 27 Apr. 2016 <https://www.android.com/versions/kit-kat-4-4/>

Guardian and GuardianAngel are written in Java environment, but with great use of Android API[53] library. Android SDK[54] provides all the tools and API library which are used to build an Android app. To develop an Android application it is required to understand how is built the Android UI[55] using XML[56], and how to use the different Android subsystems, because developing is more than just Java.

The most often used Android API library in Guardian and GuardianAngel are:

- Android.Manifest
- Android.content.Context
- Android.content.SharedPreferences
- Android.content.pm.PackageManager
- Android.content.Intent
- Android.os.Build
- Android.os.Bundle
- Android.support.v4.app.Fragment
- Android.support.v4.app.ActivityCompat
- Android.support.v4.content.ContextCompat
- Android.support.v7.app.AppCompatActivity
- Android.support.design.widget.Snackbar
- Android.database.sqlite.SQLiteDatabase
- Android.database.sqlite.SQLiteException
- Android.database.Cursor
- Android.graphics.BitmapFactory
- Android.graphics.Color
- Android.view.LayoutInflater
- Android.view.View
- Android.view.ViewGroup
- Android.widget.Button

[53] "Package Index | Android Developers." 2009. 27 Apr. 2016 <http://developer.android.com/reference/packages.html>

[54] Android, SDK. "Developer Guide." 2015.

[55] Fulcher, Richard et al. "Android ui design patterns." *Google IO. EU* (2010).

[56] Bray, Tim et al. "Extensible markup language (XML)." *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210* 16 (1998).

- Android.widget.CompoundButton
- Android.widget.EditText
- Android.widget.Switch
- Android.widget.TextView
- Android.widget.Toast
- Android.widget.AdapterView
- Android.widget.ArrayAdapter
- Android.widget.CheckBox
- Android.widget.ListView
- Android.widget.ToggleButton
- Android.app.Notification
- Android.app.NotificationManager
- Android.app.PendingIntent
- com.Android.support:appcompat-v7:23.1.1
- com.Android.support:support-v4:23.1.1
- com.Android.support:design:23.1.1
- com.fasterxml.jackson.core:jackson-databind:2.7.0
- com.google.Android.gms:play-services-location:8.4.0
- com.google.Android.gms:play-services-gcm:8.4.0

Google Maps APIs

Guardian app to its advantages must include Google Maps. Google Maps are used to display a map with localization associated with a point on the earth. Parent (guardian) after received notification about an alarm message, has the option to click the notification message and open a Google Map showing the location of the child's device. This function is very useful and important at the same time, provide a guardian with a chance to find a child and to help him.

Google Maps are the most popular map service today. Millions of apps and websites use Google Maps APIs to find location on the earth. Google Maps APIs are available for every platform, what makes it the most powerful map application in the app market.

For purpose of this project, Guardian app uses Google Maps API v2 for Android. There are a number of steps required to implement this API into this project. These steps are:

- Download Google Play Services[57]
- Generate SHA1[58] fingerprint (API key)
- Configure maps

Depending to the IDE[59], you can download Google Play Service from Android sdk folder, or like in "Android Studio" using SDK Manager on the toolbar.

Android Studio provides a great component "Google Map Activity" which is very useful tool to create an Android FragmentActivity which implements OnMapReadyCallback.

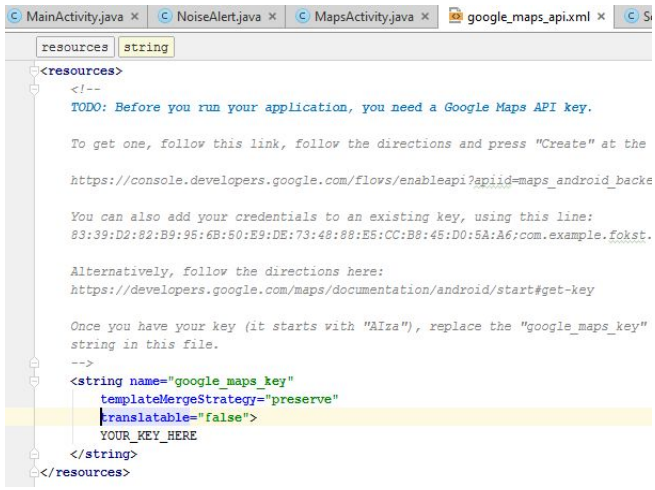


Image 14: Google maps api xml file

[57] "Overview of Google Play Services | Google APIs for Android ..." 2015. 27 Apr. 2016 <https://developers.google.com/android/guides/overview>

[58] Wang, Xiaoyun, Yiqun Lisa Yin, and Hongbo Yu. "Collision search attacks on SHA1." 13 (2005).

[59] "Integrated development environment - Wikipedia" 2011. 24 Apr. 2016 An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

The first thing to do after creating this class is to generate Google Maps API key. This could be done by few ways. The easiest way is to copy the https link from xml file created for us by Android Studio, and follow the direction and press “Create” at the end.

Another way is to add the credentials to an existing key, using the text from the same xml file. Alternatively the API key could be created from the Google Developers Console (https://console.developers.google.com).Generated key needs to be pasted into the same xml file. This will guarantee that the Google Maps will work.

Everything what is need to do more is a configuration. This configuration is made in MapsActivity Guardian app. OnMapReady() method is the place where can be add markers or lines, add listeners or move the camera. This callback is triggered when the map is ready to be used and display map with the marker in localization, which is linked to the alarm message.

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    // Add a marker and move the camera
    LatLng target = new LatLng(lat, lon);
    if(name.equalsIgnoreCase("Hubert")){
        mMap.addMarker(new MarkerOptions().position(target).title(name).
                icon(BitmapDescriptorFactory.fromResource(R.drawable.hubert)));
    }else {
        mMap.addMarker(new MarkerOptions().position(target).title(name));
    }
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(target, 15));
}
```

Image 15: Google maps method onMapReady(GoogleMap googleMap)

In addition to this configuration, it is worth mentioning also about Geocoder Android library, which together with latitude and longitude (child localization) provide address, which is displayed on the bottom of the screen (map screen).

Google Maps can be implemented with many more option on it. For example the developer can add option to change Map Type, from Normal to Satellite or Terrain or Hybrid. Another good example is showing current location, which is used successfully in second app, GuardianAngel.

Google Maps was created by two Danish brothers[60] in the company, which was acquired by Google Inc in 2004, and from this time is always developed to improve his service and to add new features.

[60] "The genius brothers (Lars and Jens Rasmussen) behind Google Wave - CNN.com." 2009. 27 Apr. 2016 <http://www.cnn.com/2009/TECH/10/27/rasmussen.brothers.google.wave/>

Google Cloud Messaging

Google Cloud Messaging (GCM) is a service that allow to send push notification messages from a GCM server to users devices (GCM clients) and also is able to receive messages from the GCM clients apps. Google Cloud Messaging (GCM) come from Android Cloud to Device Messaging (C2DM) service. C2DM was first used in Android Froyo (2.2) in 2010 and was rapidly developed by developers. In 2013 GCM service was first announced and replaced C2DM with better authentication and delivery, and with new API and messaging parameters. GCM today can distribute messages to a clients on Android, iOS[61] and Chrome[62].

GCM can be divided into two parts:

- Server implementation
- Client implementation

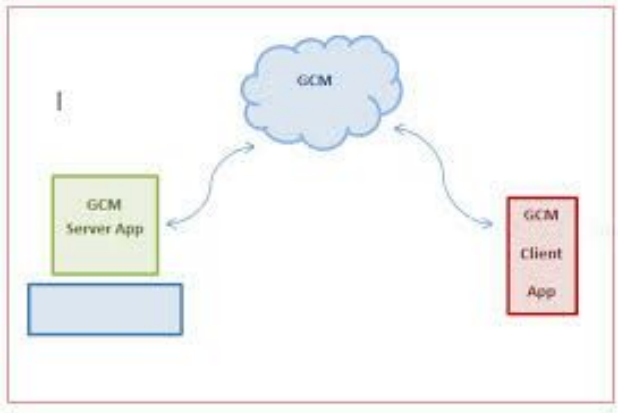


Image 16: Google Cloud Messaging (GCM) diagram

GCM server implementation must consists two components. One component is on Google side: GCM connection server, and the second is on the user side (user environment): an application server. The first step to send push notification makes the app server (user device). The app server sends a data to gcm connection server (Google), and then gcm connection server send a message to a client app. The app server before the act, must choose appropriate protocol: XMPP or HTTP.

XMPP protocol design is for upstream and downstream messages, it means that the client can send a message to the server, and the server can send a message to the client. XMPP also provide asynchronous messaging. It means that sending and receiving messages is at full speed and that the XMPP sends acknowledgment or failure message, in the form of

[61] "Ios - Mashable." 2010. 24 Apr. 2016 iOS (previously iPhone OS) is a mobile operating system developed and distributed by Apple Inc.

[62] "Chrome Browser - Google." 2014. 24 Apr. 2016 Google Chrome is a fast, simple, and secure web browser, built for the modern web.

special messages(ACK and NACK JSON) asynchronously.

HTTP has less job to do. HTTP does not provide upstream and is synchronous messaging. It means that the app server broadcast push notification as HTTP POST requests and hold for a response. This approach blocks the sender from broadcasting new notifications until the response is received.

For aim of the project, GuardianAngel has implemented the app server on his device with HTTP protocol. To support the gcm in this app, is used class Content to tie data together(latitude, longitude, name, and phone number) and Jackson (jackson-databind library) ObjectMapper to convert object into JSON and send the request.

The implementation of GCM client is dependent to the platform on which the client want to work. Android has different implementation, than is for iOS and Chrome. Same for iOS, the implementation is different than is for Chrome (and Android indeed).

For Android implementation is recommend to use Google Cloud Messaging API and Android Studio with Gradle. To meet a requirements applicable for Android, the gcm client device must operate minimum on Android 2.2 (FROYO with limited GCM features), and have the Google Play Store app installed. To use new and full GCM features, is required Android 2.3 (GINGERBREAD) or higher with the Google Play Services. For user with devices running Android 4.0.3 or less is required to set up their Google Account.
In this project Guardian app is a GCM client. Because Guardian device must running Android 4.3 (JELLY BEAN) or higher, all the GCM features are used, and is not required Google Account for these devices.
To use the GCM service to send push notification messages is required:

1. Project Number (Google Developer Console)
2. Server API key (Google Developer Console)
3. Registration ID (client)

The first thing to do is to create a Project in Google Developers Console (https://console.cloud.google.com/home/dashboard). In the step should be checked if Google

Cloud Messaging is listed as enabled API, if not need to be enabled.

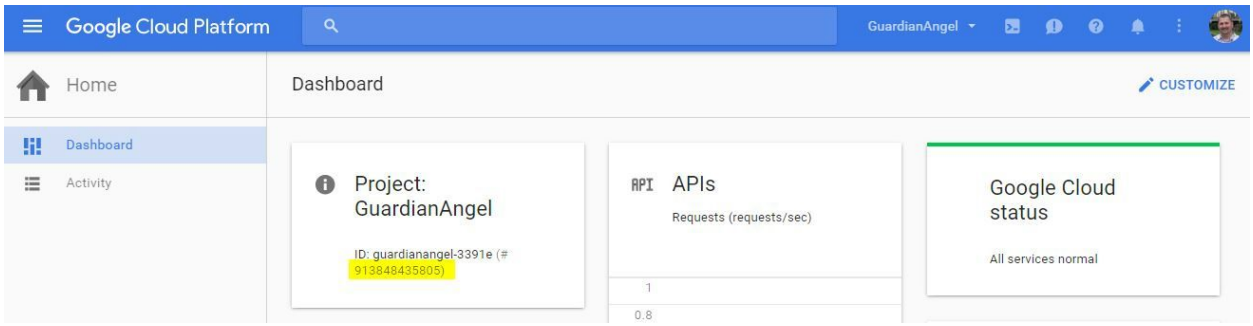


Image 17: Google Developers Console Dashboard

Beside the project id, we can find Project Number which is very important to create a client registration id(used in Guardian).

Next step is to get an API key, which is necessary to send push notification from the app server (GuardianAngel). To do this we have to find the Credential link, and click "Create credentials" button. After successfully added credential, we should have API key which start with letters "AIz".

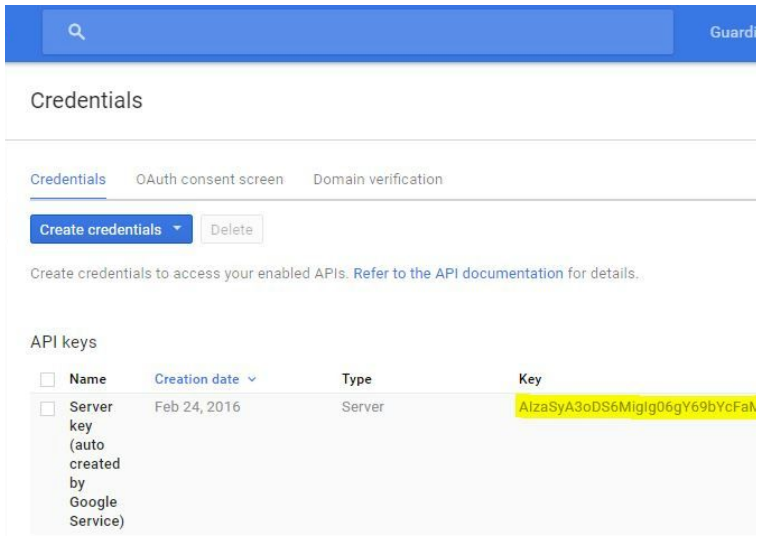


Image 18: Google Developers Console Credentials

Final step belongs to configuring the applications, client (Guardian) and app server (GuardianAngel). To allow app server to send a message to a client we need a GCM registration id of this client. To get the id, we could either import configuration file from Google Developers Console into the application, or like in case of Guardian, establish the id by the app client. To register the Guardian device with GCM service, is simply added method to get the registration id. In this method is important to use the correct Project Number, passed to instance of Google Cloud Messaging.

```
private void getRegId() {
    new AsyncTask<Void, Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            String msg = "";
            try{
                if(gcm == null){
                    gcm = GoogleCloudMessaging.getInstance(getApplicationContext());
                }
                regId = gcm.register(PROJECT_NUMBER);
                msg = "Device registered, registration ID = "+regId;
                Log.d("tag", msg);
                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putString("regId", regId);
                editor.commit();
            } catch (IOException e) {
                e.printStackTrace();
                msg = "Error: "+e.getMessage();
            }
            return msg;
        }
        @Override
        protected void onPostExecute(String msg) { Log.d("log", "Registration id: "+msg); }
    }.execute(null, null, null);
}
```

Image 19: Google Cloud Messaging register client

Once the method is called, client application is registered with GCM, and server can begin sending push notification to this client. To use this method, Guardian app has implemented Google GCM library (com.google.Android.gms:play-services:8.4.0) into build.gradle module, and has updated Android Manifest with certain permission:

- Android.permission.INTERNET
- Android.permission.WAKE_LOCK
- com.google.Android.c2dm.permission.RECEIVE
- applicationPackage + .permission.C2D_MESSAGE

To make possible by client (Guardian) to receive a gcm messages, Guardian include two classes, one to receive GCM message (MyReceiver extends WakefulBroadcastReceiver) and second to handle gcm message (GcmMessageHandler). The receiver simple listen for upcoming messages and pass component with received gcm package to gcm handler. The handler class makes Android Notification, which is displayed on the top of the client screen.

The app service (GuardianAngel) has involved more coding than the client

(Guardian). Similar to the client, it needs to import Android GCM API, and add appropriate permissions to Android Manifest (INTERNET). The GuardianAngel GCM service is supported with Content class and Jackson library (optional). Because it is one way notification, we (it?) use(s?) HTTP protocol with a POST request. A message request is built from two parts:

- HTTP header
- HTTP body

HTTP header contains Authorization (API key) and Content Type. Content Type informs the type of data that it is sending. It could be JSON (application/json) or plain text(charset=UTF-8).

The app server code should implement the response format to check the outcomes when trying to send a message. The message can be processed successfully or rejected. If is successful, the HTTP status shows 200, if is rejected, any other number.

The example below shows a nice snapshot of the GuardianAngel code implementation:

```
try {
    URL url = new URL("https://gcm-http.googleapis.com/gcm/send");
    //2. Open connection
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    //3. Set the headers
    //Log.d("tag", "apiKey: "+apiKey);
    conn.setRequestProperty("Authorization", "key=" + apiKey);
    //conn.setRequestProperty("Authorization", "key=AIzaSyA3oDS6MigIg06gY69bYcFaMcQdIiv9tpo");
    conn.setRequestProperty("Content-Type", "application/json");
    //4. Specify POST method
    conn.setRequestMethod("POST");
    //5.
    conn.setDoOutput(true);
    //6. Add JSON data into POST request body
    //a) Use Jackson object mapper to convert Contnet object into JSON//offline mode
    ObjectMapper mapper = new ObjectMapper();
    mapper.setVisibility(PropertyAccessor.FIELD, JsonAutoDetect.Visibility.ANY);
    //b)Get connection output stream
    DataOutputStream dataOutputStream = new DataOutputStream(conn.getOutputStream());
    //c)Copy Content "JSON" into
    mapper.writeValue(dataOutputStream, content);
    //dataOutputStream.write(json.getBytes("UTF-8"));
    //d)Send the request
    dataOutputStream.flush();
    //e)Close
    dataOutputStream.close();

    int responseCode = conn.getResponseCode();
    Log.d("tag", "Content Type : " + conn.getRequestProperty("Content-Type") + " Response Code: "
            + responseCode + " Authorization " + conn.getRequestProperty("Authorization"));
    Log.d("tag", "\nSending 'POST' request to URL : " + url);
    BufferedReader in = new BufferedReader(
            new InputStreamReader(conn.getInputStream())
    );
    String inputLine;
    StringBuffer response = new StringBuffer();
```

Image 20: Google Cloud Messaging application server implementation

TESTING AND EVALUATION

Testing is the last step before release the application to use, and is crucial for software development work. Testing enhances the nature of the applications, guarantee better client fulfillment, and diminish general development time spent on fixing faults.

To make all the tests I used five Samsung phones:

- Galaxy S4, Android 5.0.1 (rooted)
- Galaxy S5, Android 5.0.1 (rooted)
- Galaxy S7, Android 6
- Note 3,
- Note 4.

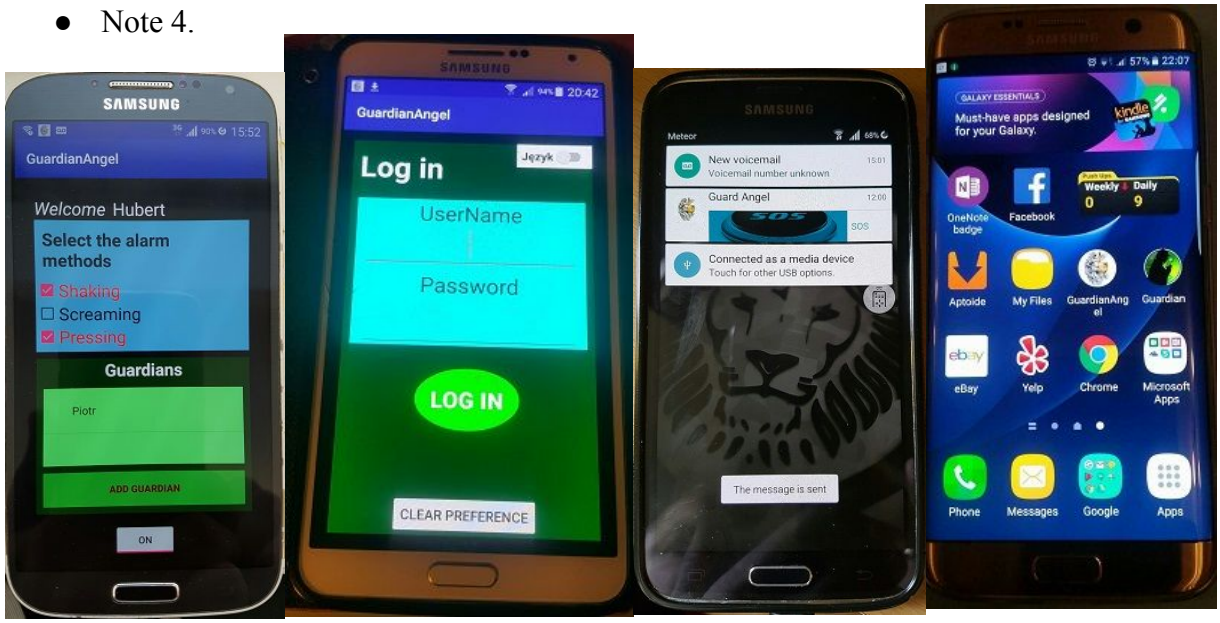


Image 21: Samsung smartphones from left: Galaxy S5, Galaxy Note 4, Galaxy S4, Galaxy S7

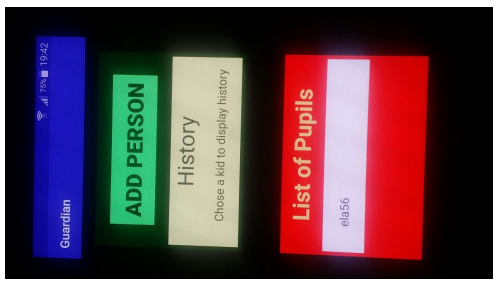


Image 22: Samsung Galaxy Note 3

GPS Coordinates

GuardianAngel used Google API play-service-location to find a location for the device. The test showed that the location service works correctly for all devices tested. Both parameters, latitude and longitude were provided every time when Google Connection Callback was used.

```
public class SendMessage extends IntentService
        implements GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener {

    private GoogleApiClient googleApiClient;
    private Location location;
```

```
location = LocationServices.FusedLocationApi.getLastLocation(googleApiClient);
if(location != null){
    Content content = createContent(location.getLatitude(), location.getLongitude(), user, userPhoneNum);
    for(int i=0; i< guardiansId.size(); i++){
        Log.d("tag", "Sending message from "+user+" to "+ guardiansName.get(i).toString() +
                " position " +
                "lat: " + location.getLatitude() +
                " lon: " + location.getLongitude() );
        sendAlarm(content);
    }
}
else{
    Log.d("tag", "Location is null");
}
```

Image 23: Google getLastLocation method

```
04-25 23:00:10.419 6019-6019/com.example.piotr.guardianangel D/tag: Get
 last Location Google Api 53.4056686 -6.2936414
04-25 23:00:10.439 6019-6019/com.example.piotr.guardianangel D/tag:
 GuardianPiotr
 APA91bE_DtWC4d5jQCdibi6tb9EI_dnaYHsTVqdOaWk6dZZTUtnKS6NcCCe0HqnpRR8j48ZvCgWCV
 -3Y_a_6R8lslywv064nJQVeQ8e8QdKlzi5etpZJJSoYZjTuChy1374M0NIVkwD
04-25 23:00:10.439 6019-6019/com.example.piotr.guardianangel D/tag:
 Guardians id from db:
 APA91bE_DtWC4d5jQCdibi6tb9EI_dnaYHsTVqdOaWk6dZZTUtnKS6NcCCe0HqnpRR8j48ZvCgWCV
 -3Y_a_6R8lslywv064nJQVeQ8e8QdKlzi5etpZJJSoYZjTuChy1374M0NIVkwD
04-25 23:00:10.439 6019-6019/com.example.piotr.guardianangel D/tag: Sending
 message from Hubert to Piotr position lat: 53.4056686 lon: -6.2936414
```

Image 24: Android Studio logcat Google latitude and longitude snapshot

GCM Connection

For GCM test I used *Android Studio logcat and GCM Notifications Test Tool*[63] By implementing Android util log into the class with GCM code, I was able to display the response code from HttpURLConnection (int responseCode = conn.getResponseCode();) in logcat (Android Studio).

On the GCM Notifications Test Tool webpage I entered GCM API Key (Server key) and GCM ID (Client Registration ID), then I typed simple message, and its value.
The test was carried out four times (each tool twice), and was successful every times .
The only small issue appeared with the time that message was travelling between the app service and app client. The message almost always arrived to the client one-, or two seconds after the message was sent from the app server. But a few times the message was traveling for up to one minute. It is presumed that this is Google server fault.

```
6019-6019/com.example.piotr.guardianangel D/tag: Sending message from Hubert to Piotr position lat: 53.4056686 lon: -6.2936414
6019-6019/com.example.piotr.guardianangel D/tag: Content Type : application/json Response Code: 200 Authorization key=AIzaSyA3oDS6MigIg06
6019-6019/com.example.piotr.guardianangel D/tag: Sending 'POST' request to URL : https://gcm-http.googleapis.com/gcm/send
6019-6019/com.example.piotr.guardianangel D/tag: {"multicast_id":6033231499154541762,"success":1,"failure":0,"canonical_ids":0,
ıge_id":"0:1461621611408088%a9954bbcf9fd7ecd"}]}
6019-6019/com.example.piotr.guardianangel D/tag: response {"multicast_id":6033231499154541762,"success":1,"failure":0,"canonical_ids":0,
ıge_id":"0:1461621611408088%a9954bbcf9fd7ecd"}]}
```

Image 25: Android Studio logcat Google Cloud Messaging test snapshot

URL: http://android.googleapis.com/gcm/send

Headers: Array ([0] => Authorization: key=AIzaSyA3oDS6MigIg06gY69bYcFaMcQdIv9tpo [1] => Content-Type: application/json) 1

Fields: Array ([registration_ids] => Array ([0] => APA91bE_DtWC4d5jQCdibi6tb9EI_dnaYHsTVqdOaWk6dZZTUtnKS6NcCCe0HqnpRR8j48ZvCgWCVW-3Y_a_6R8lslywv064nJQVeQ8e8QdKlzi5etpZJJSoYZjTuChy1374M0NIVkwD) [data] => Arr
[message] => Hello World! [itb] => Hello ITB [] =>)) 1

Result: {"multicast_id":5753951119197146987,"success":1,"failure":0,"canonical_ids":0,"results":[{"message_id":"0:1461624972966327%a9954bbcf9fd7ecd"}]}

You will be redirected to the original page after 10 seconds

Image 26: GCM GCM Notifications Test Tool - TechZog test

[63] "GCM Notifications Test Tool - TechZog - Technology and ..." 2015. 25 Apr. 2016 <http://techzog.com/development/gcm-notification-test-tool-Android/>

GuardianAngel Selected Alarm Option

Testing of GuardianAngel Alarm Method is a multi app testing with composition of UI testing, a software class (method) testing and human interaction testing.
The testing should verify functionality of applications, and the result should be used to correct unexpected behavior or to proof their correct ability to work.

Shaking

The “Shaking” test had to be conducted manually; by shaking physically all devices with enabled function “Shaking”. A device was connected to the PC with running Android Studio (did the PC have running studio installed or was the device connected to the PC via “running Android Studio?. The first would mean the sentence should be “The device was connected to the PC that had Android Studio installed” the second would have the sentence “The device was connected, via Android Studio to the PC) and the output from the logcat was recorded. The simulation of shaking was made with low accuracy and high accuracy. For low accuracy, I shook a phone approximately 10 shakes per 3 seconds. For high accuracy, I shook a phone faster, more than 12 shakes per 2 seconds. All the tests were completed successfully.

```
04-26 11:46:36.789 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...8
04-26 11:46:59.879 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...9
04-26 11:46:59.879 28633-28633/com.example.piotr.guardianangel D/sensor: Two second gone, shake=0
04-26 11:46:59.879 28633-28633/com.example.piotr.guardianangel D/sensor: Start Time: 1461667619
04-26 11:47:00.369 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...1
04-26 11:47:00.709 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...2
04-26 11:47:00.729 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...3
04-26 11:47:01.049 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...4
04-26 11:47:01.209 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...5
04-26 11:47:01.289 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...6
04-26 11:47:01.309 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...7
04-26 11:47:01.469 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...8
04-26 11:47:01.569 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...9
04-26 11:47:01.709 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...10
04-26 11:47:01.789 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...11
04-26 11:47:01.809 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...12
04-26 11:47:01.849 28633-28633/com.example.piotr.guardianangel D/sensor: 12 a shakes. Alarm message sent
04-26 11:47:01.889 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...0
04-26 11:47:01.909 28633-28633/com.example.piotr.guardianangel D/sensor: Start Time: 1461667621
04-26 11:47:01.989 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...1
04-26 11:47:02.099 28633-28633/com.example.piotr.guardianangel D/Sensor: Shaking...2
```

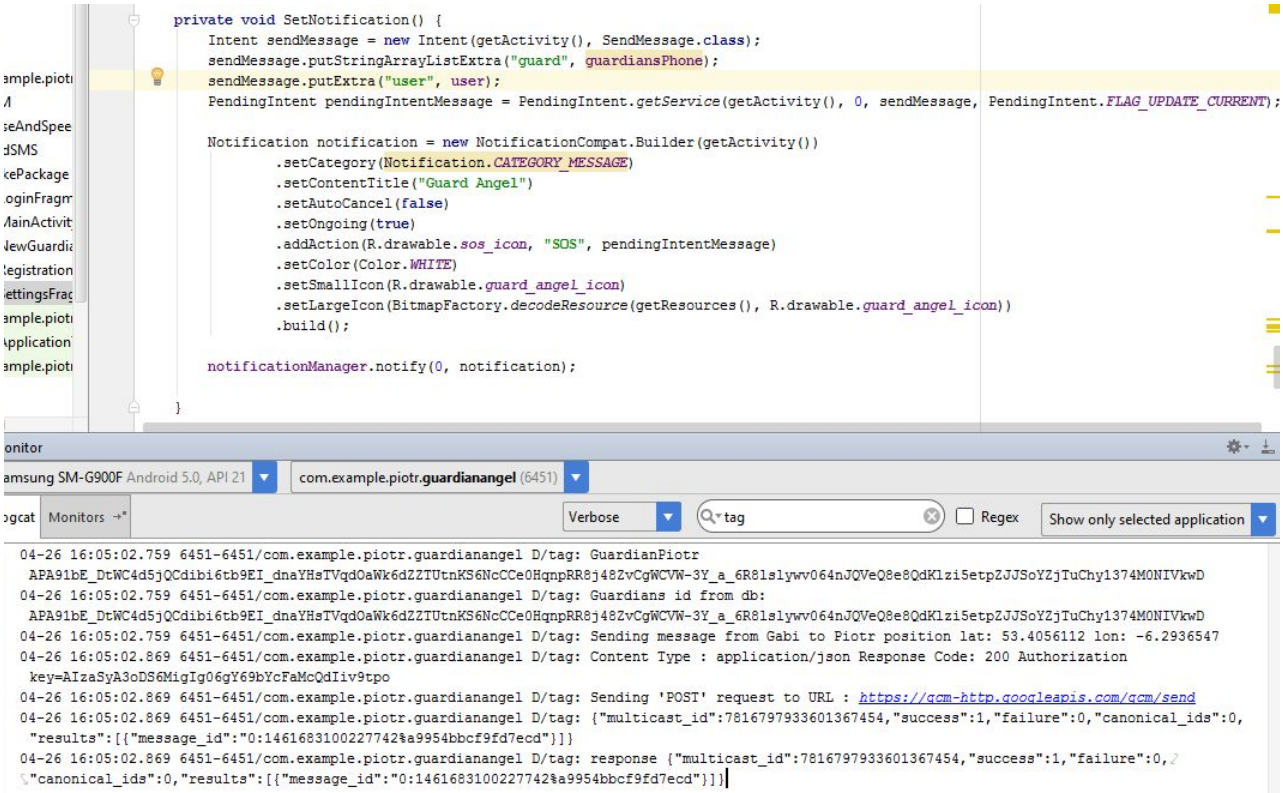
Image 27: Android Studio logcat shaking proof snapshot

Pressing

The “Pressing” test was performed by touching the screen on the “GuardianAngel notification bar”. A user has to touch the “SOS” image or “SOS” text beside the image. The notification then triggers the background work; check the last location and send gcm push notification. In fact this is UI test which involves more background work than UI, which is triggered by simple PendingIntent class.

All the phones have done this job without a problem.

Image 28: Android Studio logcat and code notification proof snapshot



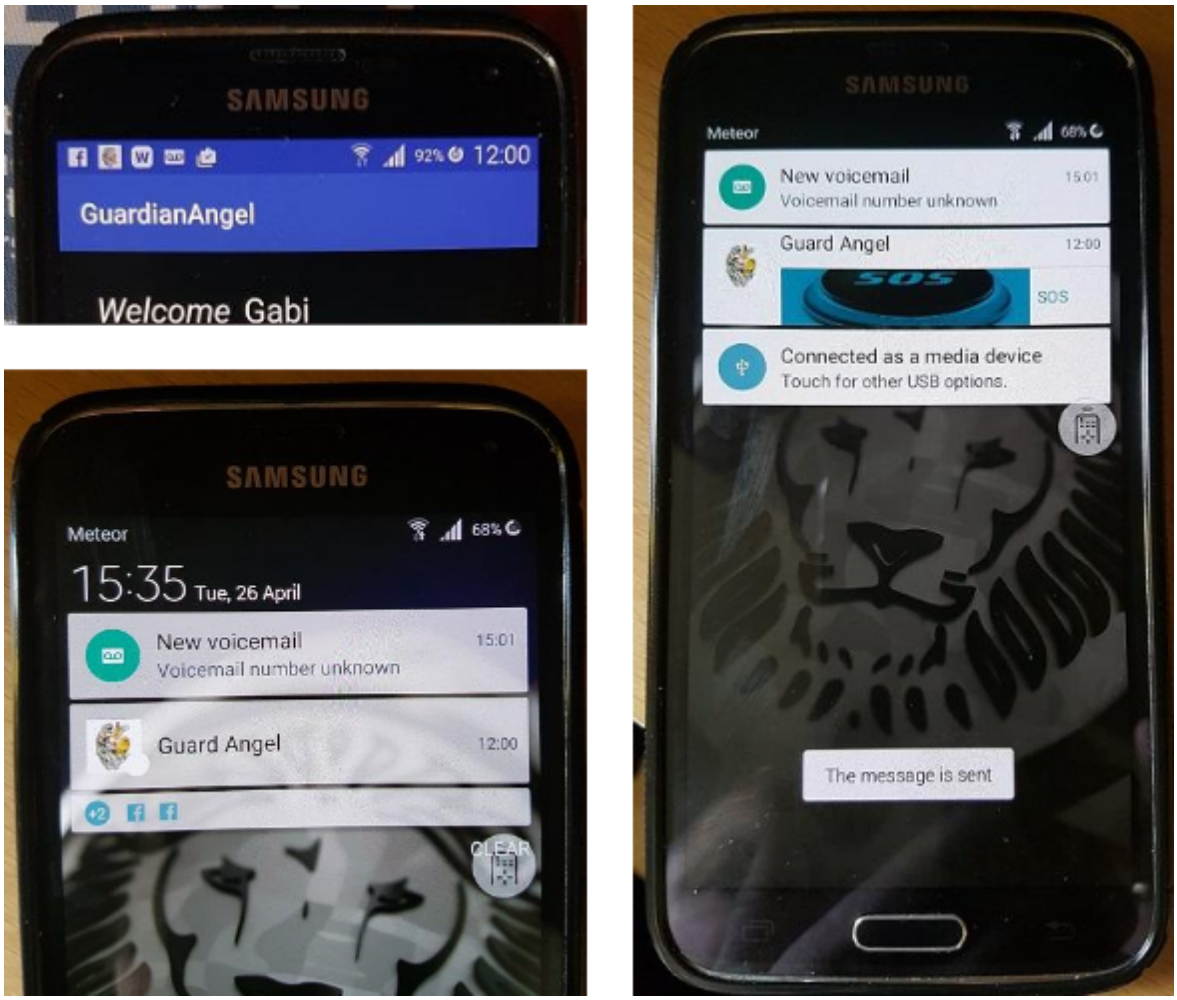


Image 29: Samsung Galaxy S5 GuardianAngel Notification “Send Alarm”

Screaming

The “Screaming” test is a human voice interaction test. The test is testing the ability of mobile device to interact with a human voice and a level of noise surrounding the device. The test is performed with “screaming method” turned on. Unfortunately the test was not successful for all devices. Samsung Galaxy S4 crashed few times with MediaRecorder Android class, where other devices worked well. The problem concerns multithreading and illegal state exception. The application try to stop MediaRecorder in a wrong state.

```
/AndroidRuntime: FATAL EXCEPTION: main
                 Process: com.example.piotr.guardianangel, PID: 24525
                 java.lang.IllegalStateException
                     at android.media.MediaRecorder.stop(Native Method)
                     at com.example.piotr.guardianangel.NoiseAndSpeechRecognizer.SoundLevelService.stop(SoundLevelService.java:49)
                     at com.example.piotr.guardianangel.NoiseAndSpeechRecognizer.SoundLevelService.start(SoundLevelService.java:32)
                     at com.example.piotr.guardianangel.NoiseAndSpeechRecognizer.NoiseAlert.start(NoiseAlert.java:96)
                     at com.example.piotr.guardianangel.NoiseAndSpeechRecognizer.NoiseAlert.onResume(NoiseAlert.java:84)
                     at com.example.piotr.guardianangel.NoiseAndSpeechRecognizer.NoiseAlert$3.onFinish(NoiseAlert.java:142)
                     at android.os.CountDownTimer$1.handleMessage(CountDownTimer.java:127)
                     at android.os.Handler.dispatchMessage(Handler.java:102)
                     at android.os.Looper.loop(Looper.java:145)
                     at android.app.ActivityThread.main(ActivityThread.java:5942) <2 internal calls>
                     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:1400)
                     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1195)
```

Image 30: Android Studio logcat MediaRecorder error snapshot

Other device work well, and Android Studio logcat shows correct result.

```
04-26 17:59:45.897 24525-24525/com.example.piotr.guardianangel D/tag: SoundLevel class is NOT running
04-26 17:59:45.897 24525-24525/com.example.piotr.guardianangel I/tag: ==== start ===
04-26 17:59:46.458 24525-24525/com.example.piotr.guardianangel D/tag: mSpeechRecognizer destroy
04-26 18:00:11.782 24525-24525/com.example.piotr.guardianangel I/tag: ==== Stop Noise Monitoring===
04-26 18:00:11.993 24525-24525/com.example.piotr.guardianangel D/tag: Call for help 21.119735109919dB
04-26 18:00:12.143 24525-24525/com.example.piotr.guardianangel D/tag: created mSpeechRecognizer
04-26 18:00:12.313 24525-24525/com.example.piotr.guardianangel D/tag: A half way of time
04-26 18:00:12.503 24525-24525/com.example.piotr.guardianangel D/tag: error = 7
04-26 18:00:12.933 24525-24525/com.example.piotr.guardianangel D/tag: onReadyForSpeech
04-26 18:00:14.055 24525-24525/com.example.piotr.guardianangel D/tag: onBeginingOfSpeech
04-26 18:00:15.816 24525-24525/com.example.piotr.guardianangel D/tag: onEndOfSpeech
04-26 18:00:15.976 24525-24525/com.example.piotr.guardianangel D/tag: Result [help me, Hackney, hug me, help meet, hook me]
04-26 18:00:15.976 24525-24525/com.example.piotr.guardianangel D/tag: Found Token. Send Alarm Message!!!!!!!!!!!!!!!!
04-26 18:00:15.986 24525-24525/com.example.piotr.guardianangel D/tag: mSpeechRecognizer destroy
04-26 18:00:15.986 24525-24525/com.example.piotr.guardianangel D/tag: Found Token. Send Alarm Message!!!!!!!!!!!!!!!!
04-26 18:00:15.996 24525-24525/com.example.piotr.guardianangel D/tag: mSpeechRecognizer destroy
04-26 18:00:15.996 24525-24525/com.example.piotr.guardianangel D/tag: Send Alarm Message!!!!!!!!!!!!!!!!
04-26 18:00:15.996 24525-24525/com.example.piotr.guardianangel D/tag: mSpeechRecognizer destroy
04-26 18:00:16.006 24525-24525/com.example.piotr.guardianangel D/tag: start service again
04-26 18:00:16.006 24525-24525/com.example.piotr.guardianangel D/tag: SoundLevel class is NOT running
04-26 18:00:16.006 24525-24525/com.example.piotr.guardianangel I/tag: ==== start ===
```

Image 31: Android Studio logcat Google SpeechRecognizer proof example

Another issue came up during the testing. Because the microphone is occupied by MediaRecorder process all the time when the “screaming” function is on, other Android applications like “phone call”, are working without the microphone. This make the “screaming” method not convenient and useless for long periods, because it absorbs and bothers all the other running applications which require the use of a microphone. It is recommended to use this function only for a short time.

CONCLUSIONS AND FURTHER WORK

Conclusion

I am satisfied with the prototype that I developed, despite the minor issue that appeared in the screaming option. My current restrictions in the development of applications for Android, particularly the threading and the speech recognizer, was the main restriction on progress which, when overcome, will definitely bring profit in the future. The main restriction was due to not enough research in the area of threading for Android, and the speech recognizer will definitely be studied in greater detail.

The next stage is to look for other options than Google API, like CMU Sphinx[64], which is a great option, but requires more time than Google Speech Recognizer API. The time restraints on testing meant there was not enough time to cover every method and class in the system. However, the main functions, which are necessary to fulfil the objectives, are done. Children are able, with the GuardianAngel application, to quickly (and secretly if they want) alarm their guardians (parents) about potential dangers. The three options to choose from the alarm method can be on and off, to meet the requirements, but after securely logging to the system. There is also Guardian application for guardians which meets all assumed goals. Guardian app in handy way notify the guardian about the alarm message, then allow the user to display Google Map with a location and address of the child. Also the guardian can very quickly choose to call the child, or the EU emergency phone number - 112[65]. In the result, I designed and created two Android application prototypes which cooperate with each other. The applications use a free software technology widely available today in the market. The only problem could be the price of the mobile device (hardware).
While this thesis has exposed the potential of the handy warning between two Android applications, to keep children safe and to efficiently collaborate between them, many opportunities for extending the scope of this thesis remain.

[64] Lamere, Paul et al. "Design of the CMU sphinx-4 decoder." *INTERSPEECH* Sep. 2003.
[65] "112.ie." 2006. 28 Apr. 2016 <http://www.112.ie/>

Future Work

In spite of its advantages and disadvantages the prototype delivers a basic core model which is ready to use, but needs future changes and further development. While working on this project, even more ideas came up, which should be very beneficial for this system.

One of them, is better control by parents, and activities for their children. The system currently only allows a child to send a message to his/her guardian. It is one way communication. However two way gcm communication could allow guardians to send message to the child's device, to make a request about their current location. This could allow the guardian to quickly localize the child, every time when the guardian wants to know where their child is located at the time of the request. This method should not bother a child, who will not even know that his/her guardian is worrying about his/her activity.

Another beneficial feature could be GPS safe zone. This option should allow a guardian to create zones on which a child is allowed to stay, like the way from home to school, to shop or the area around a place of residence. In the case of when the child has left the safety zone, the guardian must get notification on his Android device. This will increase the level of security and makes the child's life even safer.

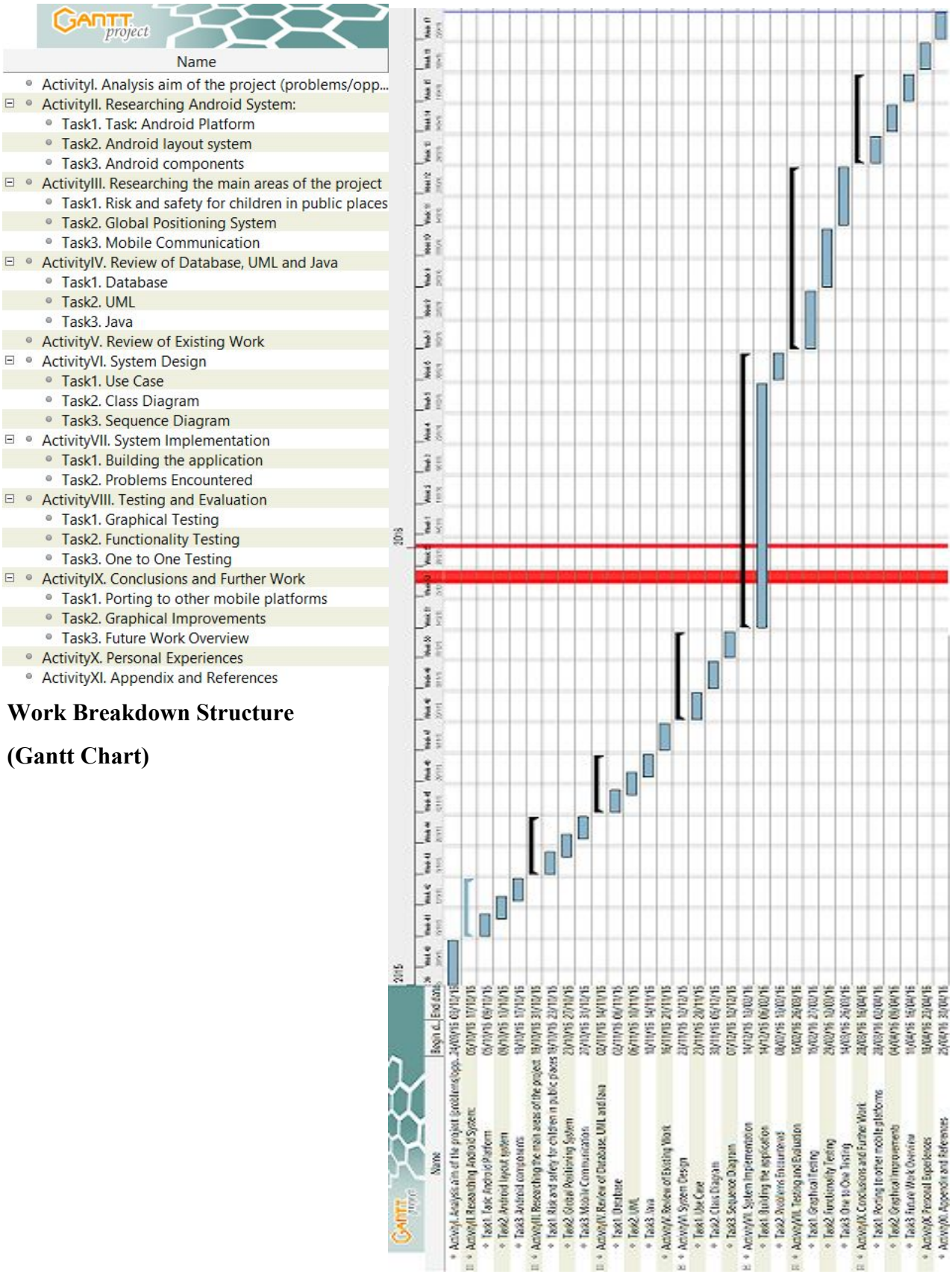
The last idea, which occurred to me, is a battery life. Because it can happen that the child's device battery can be low and soon become flat , the guardian could get a notification that the child's device will soon need charging. The first notification should be sent where battery is on 15% level with a location of the child. Last notification should be sent when battery is lower than 5%, and again should has a location of the child. Another important notification should be sent when the child device is going to be powered off. This can happen in a case of stolen a mobile phone.

During the testing I found that the GuardianAngel need to have the further work on "screaming "option; to make it hassle-free. Other optional amendments include:

- Instruction for users (both application)
- Time stamp on Google Map screen (Guardian app)
- Possibility to remove a child's history from the History List (Guardian app)
- Smoothly and slicker layout to fit different screens (both application)

APPENDIX

Appendix A



Work Breakdown Structure

(Gantt Chart)

BIBLIOGRAPHY AND REFERENCES

Dietel, Paul. *Java how to program*. PHI, 2009.

Hacigümüş, Hakan et al. "Executing SQL over encrypted data in the database-service-provider model." *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* 3 Jun. 2002: 216-227.

Developers, Android. "Google cloud messaging for Android." *2014-02-03]. http://developer. android. com/google/gcm/index. html* (2014).

Developers, Android. "What is android." 2010 (2011).

Developers, Android. "Android studio." *developer. android. com* (2015).

IntelliJ, IDEA. "the most intelligent Java IDE around."

Tovey, Helen. *Playing outdoors: Spaces and places, risk and challenge*. McGraw-Hill Education (UK), 2007.

Siochana, An Garda. "Garda Síochána policy on the investigation of sexual crime crimes against children child welfare 2010." (2011).

Kaplan, Elliott, and Christopher Hegarty. *Understanding GPS: principles and applications*. Elliott Kaplan & Christopher Hegarty. Artech house, 2005.

Kaspersky, Kris. *Hacker disassembling uncovered*. БХВ-Петербург, 2007.

Mouly, Michel, Marie-Bernadette Pautet, and Thomas Foreword By-Haug. *The GSM system for mobile communications*. Telecom publishing, 1992.

Alliance, Wi-Fi. "Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks." *White paper, University of Cape Town* (2003): 492-495.

Holzer, Adrian, and Jan Ondrus. "Mobile application market: A developer's perspective." *Telematics and informatics* 28.1 (2011): 22-31.

Schiller, Jochen, and Agnès Voisard. *Location-based services*. Jochen Schiller & Agnès Voisard. Elsevier, 2004.

Apple iOS, Android Extended Lead While. "Market Share Growth Paused." *Gartner Says Worldwide Sales of Mobile Phones Declined* 2.

Grønli, Tor-Morten, Jarle Hansen, and Gheorghita Ghinea. "Android vs Windows Mobile vs Java ME: a comparative study of mobile development environments." *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive*

Environments 23 Jun. 2010: 45.

Ortiz, C Enrique. "An Introduction to the Symbian OS™ Platform for Palm OS® Developers." *Metrowerks Corp* 21.

Goadrich, Mark H, and Michael P Rogers. "Smart smartphone development: iOS versus Android." *Proceedings of the 42nd ACM technical symposium on Computer science education* 9 Mar. 2011: 607-612.

Schlegel, Roman et al. "User-Defined Privacy Grid System for Continuous Location-Based Services." *Mobile Computing, IEEE Transactions on* 14.10 (2015): 2158-2172.

Swann, John, Eric Chatre, and Daniel Ludwig. "Galileo: Benefits for location based services." *Positioning* 1.04 (2009): 0.

Daly, P. "Navstar GPS and GLONASS: global satellite navigation systems." *Electronics & Communication Engineering Journal* 5.6 (1993): 349-357.

Finkenzeller, Klaus, and RFID Handbook. "Radio-frequency identification fundamentals and applications." *Chippenham: John Wiley & Son* (1999).

Min, Zhang et al. "A RFID-based material tracking information system." *Automation and Logistics, 2007 IEEE International Conference on* 18 Aug. 2007: 2922-2926.

Le Bodic, Gwenaël. "Short Message Service." *Mobile Messaging Technologies and Services: SMS, EMS and MMS*: 35-116.

Dahm, Markus. "Byte code engineering." *JIT'99* (1999): 267-277.

Arnold, Ken et al. *The Java programming language*. Reading: Addison-wesley, 2000.

Battelle, John. "The search: How Google and its rivals rewrote the rules of business and transformed our culture." (2005).

Mc Laughlin, Patrick et al. "Emergency CT brain: preliminary interpretation with a tablet device: image quality and diagnostic performance of the Apple iPad." *Emergency radiology* 19.2 (2012): 127-133.

Ellison, Nicole B, Charles Steinfield, and Cliff Lampe. "The benefits of Facebook "friends:" Social capital and college students' use of online social network sites." *Journal of Computer‐Mediated Communication* 12.4 (2007): 1143-1168.

Bovet, Daniel P, and Marco Cesati. *Understanding the Linux kernel*. " O'Reilly Media, Inc." 2005.

Ehringer, David. "The dalvik virtual machine architecture." *Techn. report (March 2010)* 4 (2010).

Warmer, Jos B, and Anneke G Kleppe. "The Object Constraint Language: Precise Modeling With Uml (Addison-Wesley Object Technology Series)." (1998).

Vogel, Lars. "Android development tutorial." *Luettu* 16 (2012): 2012.

Kobayashi, Shinji et al. "A geographical information system using the Google Map API for guidance to referral hospitals." *Journal of medical systems* 34.6 (2010): 1157-1160.

Abdurachmanov, David et al. "Explorations of the viability of ARM and Xeon Phi for physics processing." *Journal of Physics: Conference Series* 2014: 052008

Meier, Reto. *Professional Android 4 application development*. John Wiley & Sons, 2012..

Android, SDK. "Developer Guide." 2015.

Fulcher, Richard et al. "Android ui design patterns." *Google IO. EU* (2010).

Bray, Tim et al. "Extensible markup language (XML)." *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210* 16 (1998).

Wang, Xiaoyun, Yiqun Lisa Yin, and Hongbo Yu. "Collision search attacks on SHA1." 13 (2005).

Fielding, Roy et al. "Hypertext transfer protocol--HTTP/1.1." 1 Jun. 1999: 12.

Crockford, Douglas. "The application/json media type for javascript object notation (json)." (2006).

Ratabouil, Sylvain. *Android NDK: Beginner's Guide*. Packt Publishing Ltd, 2015.

Lamere, Paul et al. "The CMU SPHINX-4 speech recognition system." *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong* Apr. 2003: 2-5.

Works Cited

Satori, Hassan, and Fatima Elhaoussi. "Investigation Amazigh Speech Recognition Using CMU Tools." *Int J Speech Technol International Journal of Speech Technology* 17.3 (2014):