FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Mobile contextual information gathering concerning a phone number

João Carlos Teixeira de Sá

DISSERTATION

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: António Miguel Pontes Pimenta Monteiro (PhD)

July 27, 2016

Mobile contextual information gathering concerning a phone number

João Carlos Teixeira de Sá

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. José Magalhães Cruz, Universidade do Porto (PhD)
External Examiner: Prof. Helena Rodrigues, Universidade do Minho (PhD)
Supervisor: Prof. António Miguel Pimenta Monteiro, Universidade do Porto (PhD)

July 27, 2016

Abstract

Nowadays there are many communication channels that people can use to communicate with each other and mobile phones continue to be one of the most used. Most of the mobile phones used today are smartphones having features such as internet access or allowing the installation of third party applications.

The advances in mobile technolgies alongside the increasing use of social networks and related applications for smartphones contributed to the increasing amount of information shared by people and companies today. On one side, this increasingly available information can be helpful to a person when trying to contextualize an incoming phone call; on the other side, it can also be helpful to companies employees' that talk to customers (for example, sales people or technical assistance people) having access to information about the person they are talking to helping them understand the context of a call and enabling them to offer a better service.

Although there is increasingly available information about people and companies, the information can be spread across different places. Besides, there is the risk that the information stored is not accessed before or in the moment a person talks to another making it useless. Therefore, it would be interesting to aggregate the information available from different sources about a phone number and present it when needed (for example, when receiving a phone call).

To address the problem described before, a prototype of an Android application for smartphones was developed that tries to identify the person or company associated with a phone number, collecting, in real time, information about that person or company. The collection and processing of the information resorts to parallel searches in different sources simultaneously using the information obtained in a source to search on another (chain collection). The information can come from public and/or private sources of information (e.g. companies' customers databases) being displayed in a pop-up while in a phone call or when searching for a phone number.

Resumo

Hoje em dia existem diversos meios de comunicação que as pessoas podem usar para comunicar umas com as outras e os telefones móveis continuam a ser um dos mais usados. Muitos dos telefones móveis usados hoje em dia são *smartphones* possuindo características como o acesso à *internet* ou a possibilidade de instalar aplicações de terceiros.

Os avanços nas tecnologias juntamente com o crescente uso das redes sociais e aplicações relacionadas contribuíram para o aumento de informações partilhadas pelas pessoas e empresas. Por um lado, estas informações cada vez mais disponíveis podem ajudar uma pessoa no momento de contextualizar uma chamada telefónica que está a ser recebida; por outro lado, pode também ser benéfico para os empregados das empresas que falam com clientes (por exemplo, pessoal de vendas ou assistência técnica) o acesso a informação sobre a pessoa com quem estão a falar ajudando-os a perceber o contexto da chamada e permitindo a oferta de um serviço melhor.

Apesar de existir cada vez mais informação disponível acerca de pessoas e empresas, esta pode encontrar-se dispersa por diferentes locais. Além disso, existe o risco da informação não ser acedida antes ou durante o momento em que uma pessoa fala com outra tornando-a inútil. Assim sendo, seria interessante agregar a informação disponível em diversas fontes acerca de um número de telefone e apresentá-la quando necessário (por exemplo, ao receber uma chamada telefónica).

Para colmatar o problema anteriormente descrito foi desenvolvido um protótipo de uma aplicação Android para *smartphones* que tenta identificar a pessoa ou empresa associada a um número de telefone recolhendo, em tempo real, informação acerca dessa pessoa ou empresa. A recolha e processamento da informação recorre a pesquisas em paralelo em várias fontes usando a informação obtida numa fonte para pesquisas noutras fontes (pesquisa em cadeia). A informação pode ser proveniente de fontes de informação públicas e/ou privadas (e.g. base de dados de clientes) sendo mostrada sob a forma de *pop-up* durante uma chamada telefónica ou ao efetuar um pesquisa por um número de telefone.

Acknowledgements

First of all, I would like to thank my parents, my brother, my sister and the rest of my family as well as my friends for all the support until this moment.

I would also like to thank FEUP for providing the conditions to a great education including a special thank you to the MIEIC secretariat's staff that are always there to help solve students' problems.

And last (but not least), I would like to thank my Supervisor, Professor Miguel Pimenta Monteiro, and WIT Software for all the support provided namely WIT's Development Manager Paulo Sousa, WIT's Senior Sofware Engineer Alexandre Brito and WIT Software's Human Resources members Marta Coelho and Simone Guedes.

João Carlos Teixeira de Sá

"Computers are incredibly fast, accurate and stupid; humans are incredibly slow, inaccurate and brilliant; together they are powerful beyond imagination."

Albert Einstein

Contents

List of Figures

LIST OF FIGURES

Abbreviations

App	Application
API	Application Programming Interface
CRM	Customer Relationship Management
DB	Database
IDE	Integrated Development Environment
OS	Operating System
SDK	Software Development Kit

Chapter 1

Introduction

Mobile phones changed the way we communicate enabling us to be reachable almost anywhere and anytime. Despite being around for some time now, it is very common to ask about one's information namely the name or location if the phone number is not on the device's phonebook. It is also quite common to repeat the same information when talking to a company's customer support or sales representative. In this last case, it can be obtrusive leading to a bad experience for the customer.

On the other hand, with the advances in technologies there is increasingly available information about people and companies from both public (e.g. social networks) and private data providers (e.g. customer relantionship management softwares). However, there is still the possibility of not accessing this information before or when talking to someone through a mobile phone. Therefore, it would be interesting to collect and aggregate information from different sources about a person or company and display it on a mobile phone when talking to someone.

To address the problem described before a prototype of a mobile application for smartphones running the Android operating system was developed that collects and aggregates information from different sources associated with a phone number, whether the number is associated to a person or a company. The app can be triggered by receiving or making a phone call or by searching for a phone number inside it. The sources of information can be the device itself, public data providers or private data providers available through the internet. After aggregating the information, the application displays the result on the smartphone's screen.

1.1 Domain

This dissertation presents a solution in the enriched calls domain which is a part of the Rich Communication Services program by the GSM Association composed by a wide range of subdomains like multimedia content sharing during a voice call, video call or screen sharing. It is also inserted

in the mobile development domain as it presents an application prototype for Android mobile phones.

The mobile application prototype presented in this dissertation was developed in collaboration with a portuguese software company with many years of experience in the mobile industry, WIT Sofware, S.A..

Figure 1.1: WIT Software S.A. logo

1.2 Motivation

As mentioned before, there is increasingly available data about people and companies; however, this data can be spread over multiple sources including one's mobile device, public data providers available through the internet (including social networks) or a company's customer relationship software. Therefore, it would be interesting to collect and aggregate the data available from all these different sources, process it and obtain information which could then be displayed on a mobile phone's screen when in a phone call or when searching for a phone number.

Moreover, the use of smartphones (which have internet access) has also been increasing [Stab] providing an opportunity for developing a mobile application that collects, aggregates and displays contextual information on a mobile phone's screen. This can benefit both companies by providing their support or sales staff with information when talking to a customer which could enhance the customers' experience and the general public by, for instance, helping a person decide to take or not a phone call when presented with information about the caller.

1.3 Goals

The main goal of this dissertation is to present a viable solution for displaying contextual information about a phone number on a mobile phone, more precisely, on a smartphone. This goal includes developing a mobile application prototype for Android smartphones as a proof of concept.

The development of an algorithm responsible for searching, collecting and aggregating information about a person or a company given a phone number as input is also a goal of this dissertation being implemented in the developed proof of concept.

1.4 dissertation's structure

Besides this introduction, this dissertation contains 5 more chapters.

In chapter 2, the state of the art is addressed including related apps currently available.

In chapter 3, the goals, requirements and *modus operandis* of the proposed solution are presented.

In chapter 4, details of the developed solution are presented including its architecture.

In chapter 5, the results of implementing the proof of concept of the proposed solution are presented.

In chapter 6, the conclusions, discussion and possible further developments are presented.

Chapter 2

State of art

2.1 Evolution

Having some information about the person on the other side of the line can help a person contextualize a phone call; however, despite the advances in technologies, almost the same information is displayed on a mobile phone now than it was in the past namely the phone number or, if the number is on the device's phonebook, the information stored on the device's associated with that number (e.g. a name).

We are creating more information than ever [Teca] specially because of the advent of the internet and more recently, social networks. Also, companies have now access to customer relantionship management softwares that enables them to store customers' information and retrieve it later helping their employees contextualize a phone call when talking to a customer over the phone.

In addition to internet-related advances, mobile technologies have also advanced and current mobile phones, namely smartphones, are now capable of retrieving information from the internet and allow the installation of third party applications easily.

2.2 Related articles

There are some articles published on the topic of contextual information sharing in mobile communications namely "Utilizing Contextual Information for Mobile Communication" [KSHS13] published in 2013 and "Context Awareness and Mobile Phones" [Lju01] published in 2001. Both these articles address the issue of presenting contextual information on a mobile device in order to understand if a moment is appropriate to initiate a conversation and therefore preventing inappropriate interruptions.

Also, both articles referred in the previous paragraph discuss the currently available systems related to contextual information sharing in mobile communications and address the difficulties

of deploying these kind of systems. [KSHS13] also presents a prototype app for Android phones based on a survey that was conducted to assess the usefulness and the sharing preferences of contextual information between callers and callees; the app shares and shows contextual information of users in the contact list such as location, phone's usage, appointments or manual clues.

2.3 Related patents

There are also several patents published on the topic of contextual information display on mobile phones, namely [Tul07], which describes the retrieval of data from a mobile device's call history and the display of an alert with contextual information based on the collected data or, [NPA09], which describes the retrieval of data related to a call's party from a database by a microprocessor which is also responsible for arranging the presentation of the data found on the display of the communication device.

Information retrieval about a caller has also been addressed in several patens such as [NAA03] which presents a system and method for identifying a caller by looking up for information in an communication node's address book or [MHG14] which presents a method that includes storing contacts' information on a network-based server's database and then comparing the information of an incoming call with the information stored on the database leading to a CNAM query if no entry is found on the database.

Finally, [EA13] describes methods and systems for displaying notes about a contact when a phone call is initiated or received on a mobile device by accessing contact records to determine if there are notes to be displayed during the call being also related to contextual information display on mobile devices, more precisely, notes.

2.4 Related mobile applications

Several mobile applications for smartphones have become available on the market that try to identify the person or company associated with a phone number displaying the name of the person or company on the device's screen if successful. These apps are commonly referred to as caller ID apps and some of them also allow searching for information about a phone number outside a phone call. There are some of them that are very popular reaching millions of downloads on the most popular smartphones apps' stores like Play Store on Android OS or App Store on iOS.

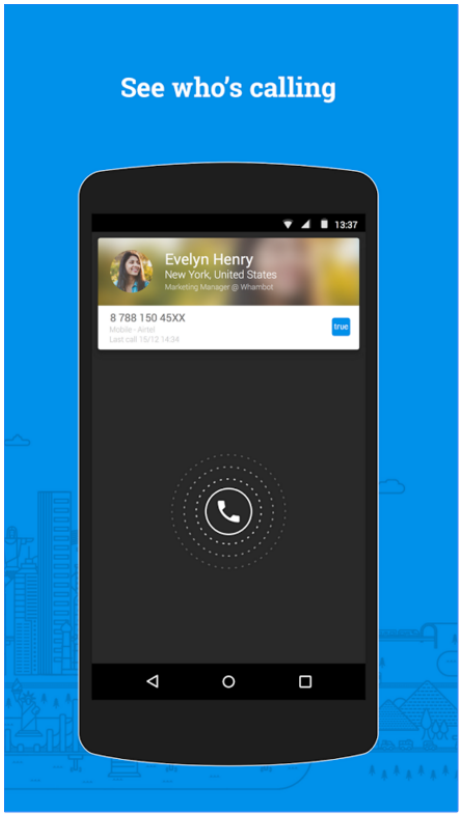
As most of these apps are intended to personal use they do not show historical data resulting from past interactions with a phone number's owner. Also, it makes sense that most of them have a community based spam list where users can report a phone number as spam since this is one of the main concerns of personal users. As stated earlier, many of these apps are publicly available on the most popular smartphones apps' stores and in order to gain competitive advantage over their competitors there is little information regarding how they search, collect and aggregate information about a phone number.

2.4.1 Examples

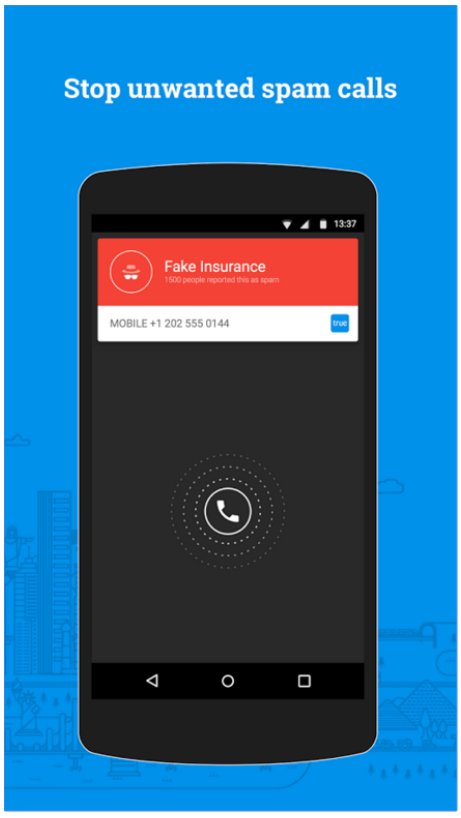
Truecaller is a mobile application for smartphones developed by True Software Scandinavia AB that searches for information about a phone number namely the name associated to it showing the collected information on the device's screen. It also has a community based spam list in order to support a call-blocking functionality and social media integration to keep the phonebook up-to-date with pictures and birthdays. The app is available for the main mobile operating systems including Android, iOS and Windows Phone.

Recently Truecaller also added a new feature related to contextual information sharing in mobile communications by displaying if an user is available to talk or busy (e.g. in a phone call); this is possible because it is a two-sided app where both the caller and the callee have the app installed.

There are several others caller ID apps available on the market offering similar features to Truecaller's including Hiya or Sync.ME.



(a) Truecaller app: Caller ID functionality



(b) Truecaller app: Spam alert functionality

Figure 2.1: Truecaller app's functionalities

2.5 Technologies

Mobile phones have seen huge advances in terms of hardware and software over the last years leading to a smartphones' widespread in the 21st century. Smartphones are mobile phones with an

advanced mobile operating system that combine mobile use features with features of a personal computer'soperating system. These type of mobile devices present a touchscreen user interface as well as features such as internet access or allowing installation of third-party apps.

There are two main operating systems available right now in terms of worlwide usage. One being iOS from Apple and the other one being Android from Google. Currently, Android is the most used mobile operating system and forecasts predict that it will continue to be [Staa] over the next years. The Java programming language, which is currently appointed as the most used programming language in the world, is the main programming language for the Android platform along with the XML programming language for design elements.

2.5.1 Android APIs

Mobile operating systems offer a wide range of APIs for developing mobile applications. In the specific case of the Android OS, the API that handles the telephone related aspects of a device such as monitoring changes in telephony states on the device (e.g. ringing) or the access to information about the telephony services on the device is the android.telephony API which is available since Android 1.0. The use of this API in the development of the mobile application prototype was essential since one of its core features is to collect and display the collected information about a phone number on a device's screen when an incoming or outgoing phone call is detected.

2.5.2 Web services

The developed mobile application makes use of web services in order to communicate with different information providers available through the internet, more precisely, web APIs. Most social networks and other public data providers as well as most customer relantionship software vendors provide a web API for accessing data through an internet connection; in fact, all the public data providers implemented on the mobile appplication prototype (Facebook, Pipl and FullContact) provide a web API as well as all customer relantionship software vendors also implemented in the prototype (Salesforce and Zoho).

A more detailed description of each of the web services provided by each of the data providers implemented on the mobile app can be found in chapter 4.

2.5.2.1 Facebook

Facebook provides a web API, commonly referred to as Graph API, that enables developers to make data requests; however, developers need to create an account and register an application in order to obtain a token that is required to make requests using its API. Regarding the topic of this dissertation, Facebook's API allows searches for information concerning a phone number or e-mail associated with a page (not a person's personal profile in order to protect users' privacy).

To search for a page using a phone number as a parameter through the Facebook's API the following URL should be used:

https://graph.facebook.com/v2.5/search?q=<phone_number>&type=page&access_token=<access_token>

where <phone_number> should be replaced by the phone number to search for and <access_token> by the token previously obtained.

The previous request returns a page's id if successful which can be used to retrieve information from that page such as the location, website or about using the following URL:

https://graph.facebook.com/v2.5/<page_id>?fields=<fields>&access_token=<access_token>

where <page_id> should be replaced by the page's id, fields by the a list of fields to retrieve from that page (separated by comma) and <access_token> by the token previously obtained.

The results of a search are sent in JSON format.

Example:

Request: https://graph.facebook.com/v2.5/search?q=351239801030&type=page&access_token=...

Response: { "data": [{ "name": "WIT Software", "id": "421088278010471" }, { "name": "WIT Software", "id": "155266524534608" }], "paging": { "cursors": { "before": "MAZDZD", "after": "MQZDZD" } } }

Request: https://graph.facebook.com/v2.5/421088278010471?fields=location,emails,website,about,likes&access_token=...

Response: { "location": { "city": "Coimbra", "country": "Portugal", "latitude": 40.194263277253, "longitude": -8.5105078485168, "zip": "3045-508" }, "emails": ["marketing@wit-software.com"], "website": "http://www.wit-software.com", "about": "WIT is a software company that creates advanced solutions and white-label products for the mobile telecommunications industry.", "likes": 2785, "id": "421088278010471" }

2.5.2.2 Pipl

Pipl is a people search engine that provides an web API to make data requests; as usual it requires developers to create an account in order to obtain a key that is required to make searches using its API. There are three different types of accounts available on Pipl:

1. Contact: Searches for name, age, gender, address and phone number;
2. Social: Searches for all Contact account fields plus images, associates and social handles;
3. Business: Searches for all Social account fields plus job, education and email address.

To search for a person using a phone number as a parameter through the Pipl's API the following URL should be used:

https://api.pipl.com/search/?phone=<phone_number>&key=<key>

where <phone_number> should be replaced by the phone number to search for and <key> by the key previously obtained.

The results of a search are sent in JSON format.

Example:

Request: https://api.pipl.com/search/?phone=351223393070&key...

Response: { "http_status_code": 200, "visible_sources": 159, "available_sources": 159, "persons_count": 1, "search_id": "160614160251168721482377532667616199 1", "query": { "phones": [{ "country_code": 351, "number": 223393070, "display": "22 339 3070", "display_international": "+351 22 339 3070" }] }, "available_data": { "premium": { "addresses": 1, "phones": 1, "languages": 1, "names": 1, "emails": 1 } }, "person": { "id": "b9190029-d57c-46a0-86d2-338f2ccc7317", "match": 1.0, "search_pointer": "...", "names": [{ "first": "Confeitaria", "last": "Cunha", "display": "Confeitaria Cunha" }], "emails": [{ "email_provider": false, "address": "full.email.available@business.subscription", "address_md5": "a5a3112a70cb5291edd8ecb4f7bd88bf" }], "phones": [{ "country_code": 351, "number": 223393070, "display": "22 339 3070", "display_international": "+351 22 339 3070" }], "languages": [{ "inferred": true, "language": "pt", "display": "pt" }], "addresses": [{ "country": "PT", "display": "Portugal" }] } }

2.5.2.3 FullContact

FullContact is a people and companies search engine that also provides an web API to make data requests; it also requires developers to create an account in order to obtain a key that is required to make searches using its API.

To search for a person using a phone number as a parameter through the FullContact's API the following URL should be used:

https://api.fullcontact.com/v2/person.json?phone=<phone_number>&apiKey=<key>

where <phone_number> should be replaced by the phone number to search for and <key> by the key previously obtained.

To search for a person using an e-mail as a parameter through the FullContact's API the following URL should be used:

https://api.fullcontact.com/v2/person.json?email=<email>&apiKey=<key>

where <email> should be replaced by the e-mail to search for and <key> by the key previously obtained.

It is also possible to search for information about a company using a domain as input using the following URL:

https://api.fullcontact.com/v2/company/lookup.json?domain=<domain>

where <domain> should be replaced with the company's domain.

The results of all the previous mentioned searches are sent in JSON format.

Example:

Request: https://api.fullcontact.com/v2/company/lookup.json?domain=wit-software.com&apiKey=...

Response: { "status" : 200, "requestId" : "d0e7ee8e-d744-4878-a364-9aebe7b25c31", "category" : [{ "name" : "Other", "code" : "OTHER" }], "logo" : "...", "website" : "http://www.wit-software.com", "languageLocale" : "en", "onlineSince" : "2000-10-23", "organization" : { "name" : "WIT Software", "approxEmployees" : 500, "founded" : "2001", "contactInfo" : { "emailAddresses" : [{ "value" : "marketing@wit-software.com", "label" : "publicity" }], "phoneNumbers" : [{ "number" : "+351 239 801 030", "label" : "other" }], "addresses" : [{ "locality" : "Coimbra", "region" : { "name" : "Coimbra" }, "country" : { "name" : "Portugal", "code" : "PT" }, "label" : "other" }, { "addressLine1" : "Centro de Empresas de Taveiro Estrada de Condeixa", "locality" : "3045-508 Taveiro", "region" : { "name" : "Coimbra" }, "country" : { "name" : "Portugal", "code" : "PT" }, "label" : "other" }] }, "images" : [{ "url" : "...", "label" : "twitter" }, { "url" : "...", "label" : "google" }, { "url" : "...", "label" : "logo" }, { "url" : "...", "label" : "logo" }, { "url" : "...", "label" : "other" }], "keywords" : ["Algarve", "Android", "Apps", "Communicator", "Development", "Ios", "Lisbon", "Magazines", "Ott-tv", "Portugal", "Portuguese", "Rcs Application Server", "Rcs-e", "SAAS", "Software", "Telco", "Telecommunications", "Vodafone", "Web", "Webrtc"] }, "socialProfiles" : [{ "bio" : "WIT is a software company that creates advanced solutions and white-label products for the mobile telecommunications industry.", "followers" : 757, "following" : 158, "typeId" : "twitter", "typeName" : "Twitter", "url" : "https://twitter.com/WIT_Software", "username" : "WIT_Software", "id" : "32636507" }, { "followers" : 5, "typeId" : "angellist", "typeName" : "AngelList", "url" : "https://angel.co/wit-software", "username" : "wit-software", "id" : "69044" }, { "typeId" : "crunchbasecompany", "typeName" : "CrunchBase", "url" : "...", "username" : "wit-software" }, { "bio" : "WIT is a software company specialized in rich and unified communications for Mobile Operators and Mobile internet companies. WIT was established in 2001 and has customers in over 15 countries. Our range of products in the Telecom arena includes:- a complete set of RCS5.1 client applications (iOS, Android, tablets, PC and Web browsers) - a RCS Application Server capable of delivering full RCS functionality in IMS and pre-IMS networks. - a WebRTC Gateway, that makes WebTC work and provides interoperability with the Mobile RCS Apps. Our excellence in this area allowed us to become the selected GSMA vendor for Android and iOS Clients for RCSe. The company is also specialized in the development of Mobile VoIP solutions, IMS applications, and is preparing the evolution to VoLTE. The TV Unit develops a software product for IPTV Operators. The product includes a package of TV Widgets, a VoD Storefront, a platform for Converged Communications, Contextual TV Apps and a solution for multiscreen viewing of TV (smartphones, tablets, Web). The Mobile Unit is specialized in the development of software for Mobile Banking, Mobile Payments, Electronic Recharges, and developed a middleware framework that speeds up the development of mobile sites and mobile apps.", "followers" : 63, "typeId" : "google", "typeName" : "GooglePlus", "url" :

"https://plus.google.com/u/0/108859362327826810638", "id" : "108859362327826810638" }, { "typeId" : "klout", "typeName" : "Klout", "url" : "http://klout.com/WIT_Software", "username" : "WIT_Software", "id" : "44473051086205870" }, { "typeId" : "instagram", "typeName" : "Instagram", "url" : "https://instagram.com/witsoftware" }, { "typeId" : "facebookpage", "typeName" : "Facebook", "url" : "https://www.facebook.com/pages/WIT-Software/421088278010471", "username" : "WIT-Software", "id" : "421088278010471" }, { "bio" : "WIT is a software development company specialized in advanced solutions for mobile telecommunications companies. WIT provides converged solutions based on the IP Multimedia Subsystem (IMS) for voice (VoIP, Video over IP, Mobile VoIP and Voice over LTE), for messaging (SMS and MMS over IP), Rich Communication Suite (RCS2.0, RCS-e and RCS 5) and Multimedia Telephony Services (MMTel). Our offer includes a complete set of client applications (for PC, Web, iPhone, iPad and Android) to deliver full RCS functionality in IMS and non-IMS networks. Established in 2001, WIT has customers is more than 15 countries and in major operators like Vodafone, TeliaSonera, Deutsch Telekom, Telefonica and Orange.", "typeId" : "linkedincompany", "typeName" : "LinkedIn", "url" : "https://www.linkedin.com/company/wit-software", "username" : "wit-software", "id" : "66025" }, { "typeId" : "youtube", "typeName" : "YouTube", "url" : "https://youtube.com/user/WITsoftware", "username" : "WITsoftware" }], "traffic" : { "ranking" : [{ "rank" : 1476763, "locale" : "global" }] } }

2.6 Categorized information

The information supplied by most information providers available through the internet about a person or company is based on a category system, i.e., each piece of information provided is related to a category (e.g. name, location etc.); in fact, many categories such as the name, location, e-mail or website are transversal to different information providers.

Therefore it is possible to differentitate the information and, in some cases, it is required by the information providers that categories are specified when making a request (e.g. Facebook); other providers just respond with all the information they have available associating each piece of information with a category (Pipl and FullContact).

2.7 Conclusions

There have been huge advances in mobile technologies over the last years and nowadays functionalities such as internet access or installation of third-party applications are common in mobile devices usually referred to as smartphones which became widespread in the 21st century.

Despite the advances in mobile phones, the information displayed about the person on the other side of a phone call does not differ much from the past. In fact, almost the same information is shown nowadays than it was in the past namely a name asscіated to a phone number if the number is on the device's phonebook.

In spite of what was said in the previous paragraph, information retrieval associated with a phone number is an issue already addressed by several companies; however, most of them take it on a caller ID perspective not focusing on retrieving data from past interactions (e.g when was the last conversation with a phone number's owner or if the last call from a phone number was missed). This is particularly important for companies because they have access to customer relantionship software where they store customers' information, including details about past conversations, that could be presented when in a phone call with a customer.

Chapter 3

Requirements

3.1 Goals

As referred in chapter 1, the main goal of this project is to present a mobile application prototype for Android smartphones for searching, collecting and aggregating information from different information providers about a phone number when starting a phone call or when searching for a phone number inside the app displaying it on the mobile device's screen.

On one hand, the presented solution should help a person's decision of taking a phone call by letting him/her know some information about the caller (e.g. the name if the phone number is not on the device's phonebook); on the other hand, it should help companies' employees provide a better customer support by presenting them with information about a customer including historical information resulting from past interactions (e.g. previous calls' notes).

3.2 Technical Requirements

The mobile application prototype requires a smartphone with the Android operating system as well as an internet connection in order to search and collect information about a phone number available from public and/or private information providers on the internet (if there's no connection available the application can only show information from the device's phonebook and the app's local database where notes associated with a phone number are stored).

3.3 Functional Requirements

Regarding functional requirements, the mobile application prototype should be able to:

1. Detect a call

Pre-Conditions: The mobile application must be installed and active (on the aplication's settings)

When the phone receives or makes a call the application should detect it.

2. Identify a caller

Pre-Conditions: An incoming call has been detected by the mobile application

When a call is detected by the application it should start searching for the identity of the caller, i.e. a name. This search should start by the phone's phonebook and then continue on public data providers and/or a private data provider previously selected on the app's settings.

3. Collect information

Pre-Conditions: The caller has been identified

When the caller has been identified the application should more information about the caller. This search should collect available information (e.g. photo, location, e-mail address) from public data providers and/or a private data provider previously selected on the app's settings in parallel.

4. Aggregate (and process) information

Pre-Conditions: Information has been collected

While the application collects information about the caller an algorithm responsible for comparing the information collected from the different sources should be running alongside another one responsible for ranking the collected information. These algorithms should run until the search for information ends, updating the results in real-time, if needed.

5. Display information

Pre-Conditions: Information has been processed

As soon as the application has collected and processed information it should be displayed on the screen in a movable pop-up window and the user should be able to see information from a specific source. Also, as the algorithm responsible for processing the collected information continues running on the background andthe information presented on the screen can be updated.

6. Select data providers

Pre-Conditions: The mobile application must be installed

When accessing the app's settings, a list of public and private data providers should be presented where the providers that should be used when searching for information about a phone number can be selected (selecting no public data providers or no private data provider should be allowed).

7. Activate/Deactivate app

Pre-Conditions: The mobile application must be installed

When accessing the app's settings, there should be an option to activate or deactivate the application; if the app is active the search, collection and display of information should be automatically triggered when a phone call is detected, otherwise that doesn't happen.

3.4 Visual Requirements

The mobile application prototype should display the collected and processed information associated with a phone number in a pop-up window that should appear on top of every other screen (including the call screen). This window should be movable by long presssing it and moving it to the desired position; it should also be possible to minimize it and maximime it (when minimized) when in a phone call in order to provide more flexibility for interacting with the screen's elements behind it.

3.5 Main Use Cases

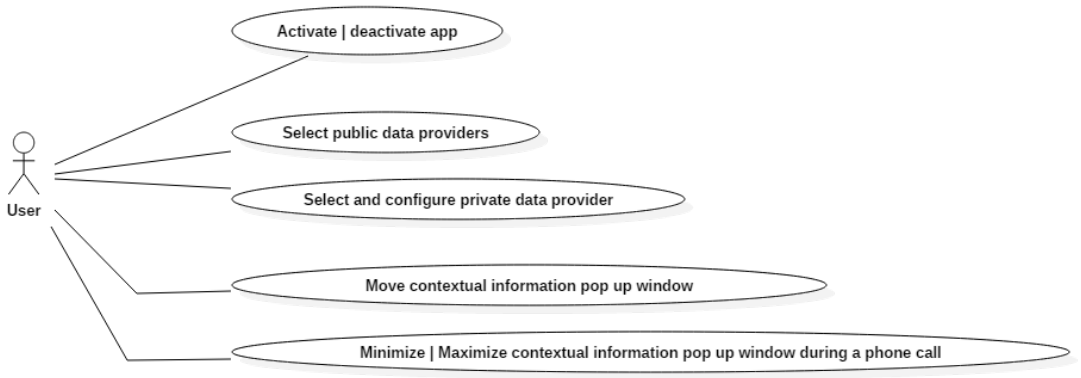


Figure 3.1: App's main use cases

3.6 Modus operandis

The prototype tries to identify the person or company associated with a phone number by searching for information on previous selected information providers (on the app's settings). If successful, it collects, in real time, information about the identified person or company from the different providers, in parallel. The collected information is aggregated and then displayed on the smartphone's screen in a pop-up.

In general, there are three main phases associated with the application *modus operandis*:
1st phase: Search for information associated witha phone number on previously selected information providers;
2nd phase: Collect and aggregate the collected information;
3rd phase: Display the processed information.

These three phases are explained in more detail in chapter 4.

3.7 Survey on a mobile application for displaying contextual information concerning a phone number

An online survey was conducted to assess which contextual information concerning a phone number people would consider relevant to display in a mobile application. Also, questions about when they would like to see the information displayed on the screen, what is the level of interest in it and what utility would they see in it.

3.7.1 Method

The online survey consisted of three parts and was set up on the internet in order to recruit a large number of participants. The first part of the questionnaire included only one question about having a smartphone. If the answer was yes the participant would go to the second part of it; otherwise he/she would go directly to the thid part. In the second part, questions about having a network data plan and being usually connected to the internet were made. Finally, in the third part, several questions about an application for displaying contextual information were presented to the participants: which information they would consider relevant showing from a list that included name, location, job, other contacts (e.g. e-mail), profile networks and a field that could be filled in with other suggestions; when they would like to see the information displayed on their mobile phone's screen from a list that included when receiving or making a phone call, when receiving or sending a text message, when searching for a contact, when searching for a phone number and a field that could be filled in with other suggestions; if the participant would be interested in an application of this genre and what would be the level of interest in it from a scale of 1 to 5 where 1 being not interested and 5 being very interested; what utilities would they see in it from a list that included helping decide to take or not a phone call, avoid providing repeated information to companies they were customers of and a field that could be filled in with other suggestions.

The survey was distributed through the students mailing lists of the Faculties of Engineering, Economics and Medicine of the University of Porto and was available for approximately 7 days registering 432 answers. It took approximately 1 to 2 minutes to answer all questions.

Aplicação para visualizar informação associada a um número de telefone

No âmbito de uma Dissertação do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia do Porto pretende obter-se a opinião da comunidade académica em relação a uma aplicação para dispositivos móveis que mostre informação obtida a partir do próprio dispositivo, da Internet (várias fontes) e/ou de bases de dados de clientes (no caso de empresas) acerca de um número de telefone.

*Required

1. **Possui smartphone?** *
   Mark only one oval.

   ◯ Sim *Skip to question 2.*
   ◯ Não *Skip to question 4.*

Aplicação para visualizar informação associada a um número de telefone

No âmbito de uma Dissertação do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia do Porto pretende obter-se a opinião da comunidade académica em relação a uma aplicação para dispositivos móveis que mostre informação obtida a partir do próprio dispositivo, da Internet (várias fontes) e/ou de bases de dados de clientes (no caso de empresas) acerca de um número de telefone.

2. **Possui plano de dados móveis?**
   Mark only one oval.

   ◯ Sim
   ◯ Não

3. **É usual ter ligação à Internet (via WiFi ou dados móveis) disponível no seu smartphone?** *
   Mark only one oval.

   ◯ Sim
   ◯ Não

Aplicação para visualizar informação associada a um número de telefone

No âmbito de uma Dissertação do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia do Porto pretende obter-se a opinião da comunidade académica em relação a uma aplicação para dispositivos móveis que mostre informação obtida a partir do próprio dispositivo, da Internet (várias fontes) e/ou de bases de dados de clientes (no caso de empresas) acerca de um número de telefone.

4. **Quais as informações que considera relevante serem mostradas acerca de um número de telefone?** *
Tick all that apply.

- [ ] Nome
- [ ] Localização
- [ ] Local e posto de trabalho
- [ ] Outros contactos (e.g. e-mail)
- [ ] Perfis sociais
- [ ] Other: .....

5. **Quando é que gostaria de ver as informações mostradas no smartphone?** *
Tick all that apply.

- [ ] Ao receber/efetuar uma chamada de voz
- [ ] Ao receber/enviar uma mensagem de texto
- [ ] Ao procurar por um contacto
- [ ] Ao procurar por um número de telefone à escolha
- [ ] Other: .....

6. **Estaria interessado/a numa aplicação deste género?**
Mark only one oval.

- ◯ Sim
- ◯ Não

7. **Qual o seu nível de interesse numa aplicação deste género?**
Mark only one oval.

	1	2	3	4	5	
Nada interessado(a)	◯	◯	◯	◯	◯	Muito interessado(a)

8. **Quais as utilidades que vê numa aplicação deste género?**
Tick all that apply.

- [ ] Ajuda na decisão de atender uma chamada (a informação acerca de quem está a ligar é mostrada no momento em que se recebe uma chamada)
- [ ] Evitar repetir o fornecimento de informações a empresas das quais se é cliente (a empresa vê a informação do cliente no momento em que efectua uma chamada para um cliente)
- [ ] Other: .....

Powered by Google Forms

2/2

Figure 3.3: Survey's questions (in portuguese)

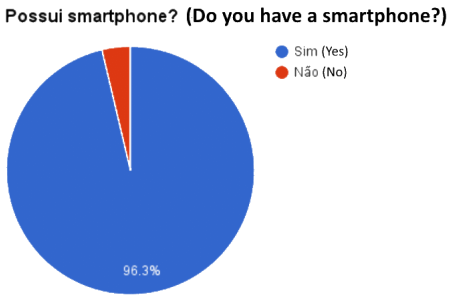
3.7.2 Results

A large majority of the participants on the survey, approximately 96%, have a smartphone; from these participants, approximately 87% also have a network data plan and approximately 95% usually have a connection to the internet.

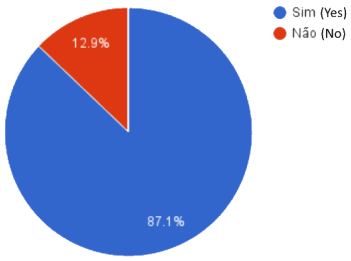
Regarding which information they would consider relevant showing in the app, 40.7% of the participants selected the name, 19.1% selected other contacts (e.g. e-mail), 14.4% selected location, 14% selected profile networks and 10.2% selected job; also, 17 people filled in the suggestions' field with the photo associated with a phone number; regarding when they would like to see the information displayed on their mobile phone's screen 32.1% selected when receiving or making a phone call, 29% selected when searching for a contact, 19.5% selected when receiving a text message and 18.6% selected when searching for a phone number.

One of the purposes of the survey was to understand if people have interest in having a mobile application capable of displaying contextual information associated with a phone number on their mobile devices; regarding this question, 69.3% of the participants said they would be interested in a mobile application for this purpose whereas 30.7% said not being interested. As for the level of interest in a scale of 1 to 5 where 1 being not interested and 5 being very interested, 173 people selected a value above the middle value (with 146 selecting a 4 and 27 selecting a 5) while 121 people selected a value below the middle value (with 52 selecting a 1 and 69 selecting a 2); as for the middle value of 3, 137 people selected it.

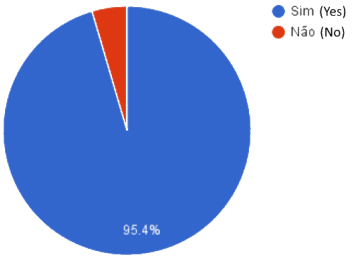
Finally, regarding what utilities participants would see in a mobile application capable of displaying contextual information associated with a phone number, most people, more precisely 60.9%, selected helping a person decide to take or not a phone call whereas 34.2% selected avoid providing repeated information to companies they were customers of; other suggestions placed in the suggestions' field included acquiring other channels to contact someone or leave a message, acquiring information about someone or better organize their mobile phone's phonebook.



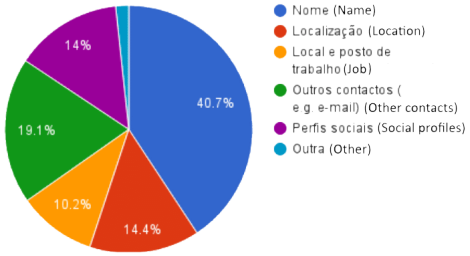
Possui plano de dados móveis? (Do you have a data plan?)



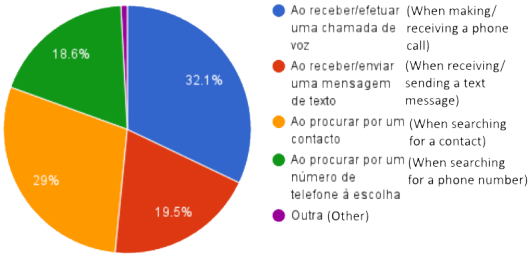
É usual ter ligação à Internet (via WiFi ou dados móveis) disponível no seu smartphone?
(Do you usually have a connection to the Internet (via Wifi ou data plan) available on your smartphone?)



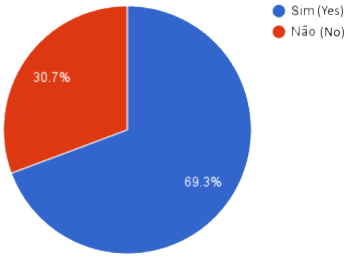
Quais as informações que considera relevante serem mostradas acerca de um número de telefone?
(What information would you consider relevant being displayed about a phone number?)



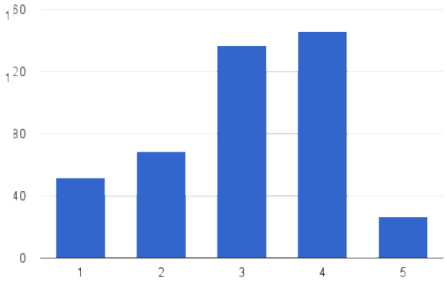
Quando é que gostaria de ver as informações mostradas no smartphone?
(When would you like to see the information displayed on the smartphone?)



Estaria interessado/a numa aplicação deste género?
(Would you be interested in an app of this kind?)



Qual o seu nível de interesse numa aplicação deste género?
(What is your level of interest in an app of this kind?)



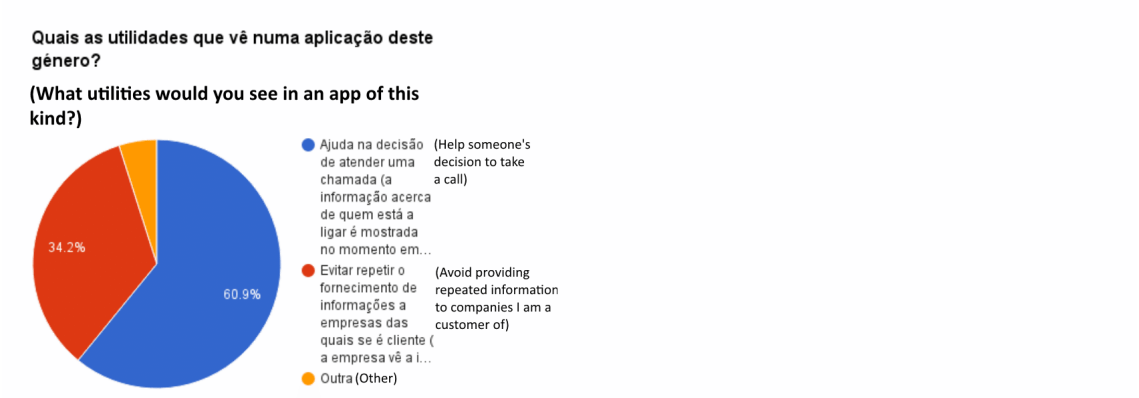


Figure 3.5: Survey's results (with questions and answers translated to english)

3.7.3 Discussion

The results reveal that most participants in the survey have a smartphone and that they usually have an internet connection available on it. On the other hand, most people also revelead being interested in a mobile application capable of displaying contextual information associated with a phone number with the level of interest in it being mostly average or slighlty higher.

Regarding a mobile application per se, the results about what information to display and when to display it were somehow fractured; however it is possible to observe that the name associated with a phone number was the most selected option from the list of what information participants would like to see displayed as well as when receiving or making a phone call being the most selected option when it comes to when they would like to see the information displayed on their mobile devices.

3.7.4 Related surveys

Article [KSHS13] presents a related online survey that was conducted to assess the usefulness and the sharing preferences of contextual information between callers and callees. A list with 18 contextual informations was presented to the participants asking them to rate how helpful each information was to them as well with whom (nobody, only few selected, friends, everyone) they would share each of the presented informations. The authors also asked participants to provide themany additional contextual cues they believed could be helpful for initiating a phone call.

The results of the survey reveal that having an appointment, abstract location and current activity were selected as the most useful information whereas information such as number of surrounding people, exact position, or being inside a building were considered less helpful; the results also showed that participants were willing to share contextual information as long as the disclosed information was abstract enough.

3.8 Information Requirements

Based on the results of the conducted survey previously described, the prototype app should include all the proposed information categories such as the name, location, job, other contacts and social profiles since they were selected by, at least, 10% of the participants; the most suggested category should also be included, i.e.,photo. Other categories of information that should be implemented include information based on the device's call history (because having information about past calls can enhance a phone call), notes and tasks (both are core information in a customer relantionship software and can also be important to understand the context of a call).

3.9 Conclusions

The presented requirements for a mobile application for Android smartphones that searches, collects and aggregates information associated with a phone number from different sources previously selected (when there is information available to be collected) displaying it on the screen satisfies important needs at the personal or corporate level depending on the information providers selected; for instance, it can both benefit companies when it comes to support or sales staff talking to a customer or simply help a person decide to take a phone call when presented with some information about the caller.

Chapter 4

Solution

4.1 Methodology

Prior to the mobile application prototype development there was a requirements analysis which included technical, functional and visual requirements as described in chapter 3. The visual requirements analysis was accompanied by WIT's User Experience team. A first prototype was then developed to evaluate multi threaded parallel searches for information using an internet connection and visual display of the collected information in a pop-up window on top of any other screen.

A research about which information providers to use in the prototype was conducted in order to understand which information providers were available to use, the input parameters accepted by each one of them and ultimately which of them had the characteristics to be considered relevant (e.g. amount of data). Based on this research, some information providers were integrated into the prototype being described more in detail further in this chapter.

An information processor was then developed to aggregate and process the collected information from the different sources as well as an information ranker responsible for determing the order of display of each piece of information; the prototype's user interface was upgraded as the visual requirements were defined.

Throughout all the development process, a collection of functional tests were made in order to understand the app's development progress.

4.2 Architecture

As mentioned before, the presented solution included the development of a mobile application prototype for Android smarphones as a proof of concept which includes a local database for storing notes associated with a phone number. It also included the development of a custom middle layer in order to homogenize the interaction between the mobile application and customer relantionship management softwares from different vendors. This middle layer includes a server that acts as an

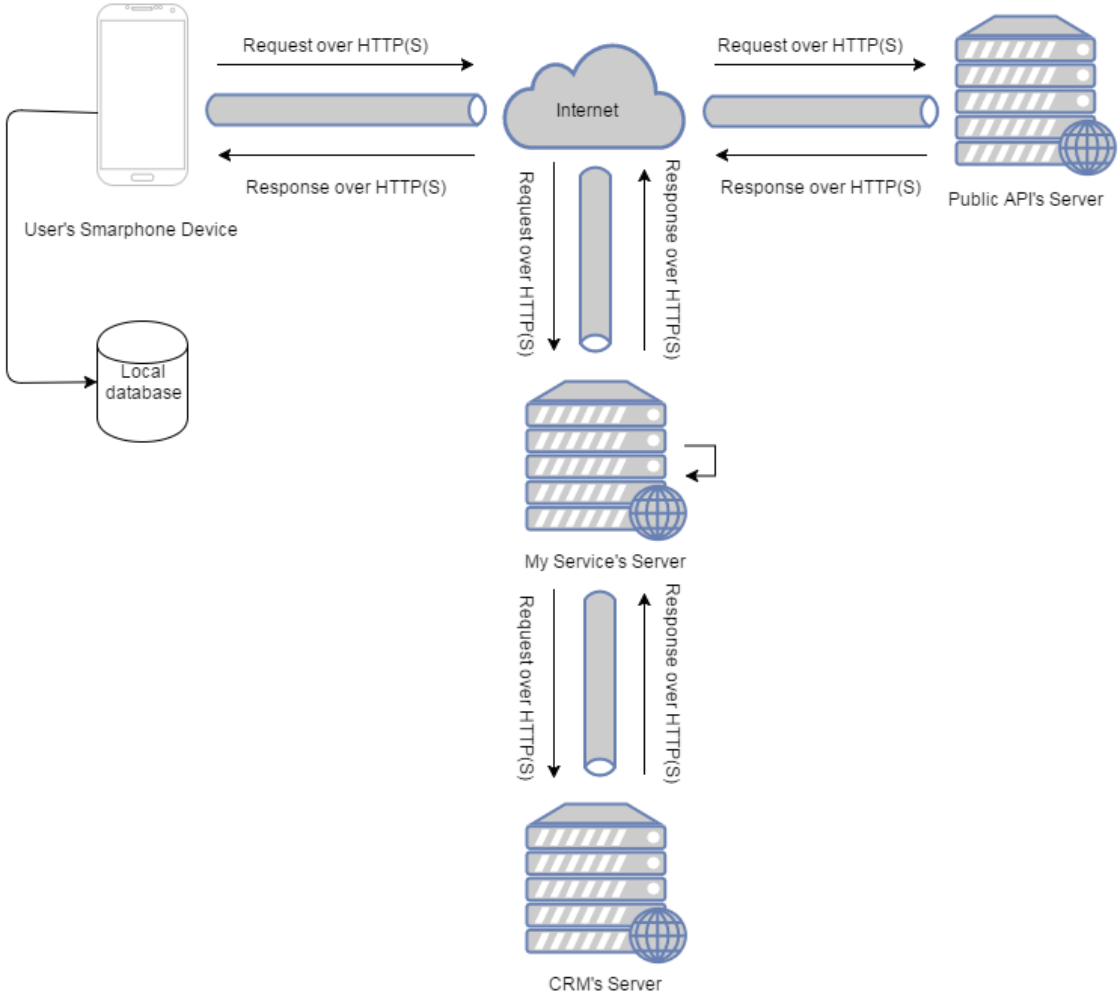


Figure 4.1: Solution's architecture

intermediary between the app's information requests and the customer relantionship management softwares' servers.

As mentioned in the previous paragraph, the mobile application runs on smartphones with the Android operating system; the custom middle layer's server runs on a remote machine which should always be available through the internet in order to allow information requests to a CRM's server. The information requests to public data providers' APIs are sent directly from the app to the respective server which also responds directly to the app using an internet connection.

In short, the app can obtain information from two different ways:

1. Internally,:

1.1. From the device's phonebook;

1.2. From the app's database (which stores notes associated to a phone number).

2. Externally, by:

2.1 Requesting information from servers of public social networks' APIs or other public data

providers' APIs which send their responses directly to the app.
2.2. Requesting information from a CRM's server using a custom service's server as an intermediary; the CRM's server responds to the custom service's server which responds to the app.

4.3 Technologies

The mobile application runs natively on devices with the Android operating system and was developed using the official Android Studio IDE with Java being the main programming language alongside XML to design the app's user interface. The custom middle layer's API runs on a machine under the Node.js platform and was developed using the Javascript programming language.

4.4 Data collection

Data about people or companies can be collected from public data providers like social networks or people/companies search engines or from private data providers like companies' customer relationship management softwares or a device's notes local database. Regarding the public data providers it is important to refer that although considered public they can have some restrictions in order to protect users' privacy. For instance, Facebook allows to search for a phone number when searching for a company page but does not allow it when searching for a person. Also, generally, public data providers request some sort of access key in order to provide data.

Private data providers play an important role when collecting data about a person or company especially for companies which have a customer relationship management software enabling them to improve their customers' experience by correctly using the data stored on these type of systems. However, in order to achieve this, it is important that companies store customers' data whenever possible.

Allying public and private data providers becomes important in order to obtain better results since there's some information like the name or location associated with a phone number's owner that can be available publically but there's also some information like previous calls' notes that are only available from private data sources.

4.4.1 Included public data providers

Many social networks provide a web API to access users' data, but this access is only granted by most of them by using some sort of access key previously provided. There are also some other people/companies data providers that provide a web API which also request some sort of access key previously provided.

In the case of the developed mobile application, the API of Facebook was used along with the APIs of Pipl and FullContact. Facebook is currently the market leader in social networking having surpassed 1 billion registered accounts [Tecb]. Pipl and FullContact are services that allow searching for information about a person or company.

Nowadays Facebook's API only allows to search for a phone number or e-mail associated with a company (not a person) in order to protect users' privacy. Therefore the developed app only shows information from Facebook when it comes to a phone number associated with a company. On the other hand, Pipl's API allows searching for a phone number or e-mail associated with a person or company as well as FullContact's API.

In order to improve data collection results the app uses information collected from a source to search on another source, when possible. For example, if a search for a phone number on Pipl collects information about an e-mail address another search will be made on FullContact using the e-mail address collected from Pipl as results can differ when using different input parameters on FullContact.

4.4.1.1 Facebook

As previously referred, Facebook only allows to search for a phone number associated with a page (not a person's personal profile) using its web API. Thus, the prototype firstly makes a request for a page's name associated with a phone number. If found, the prototype then requests the page's default photo followed by information related to location, emails, website, about, number of likes and last post.

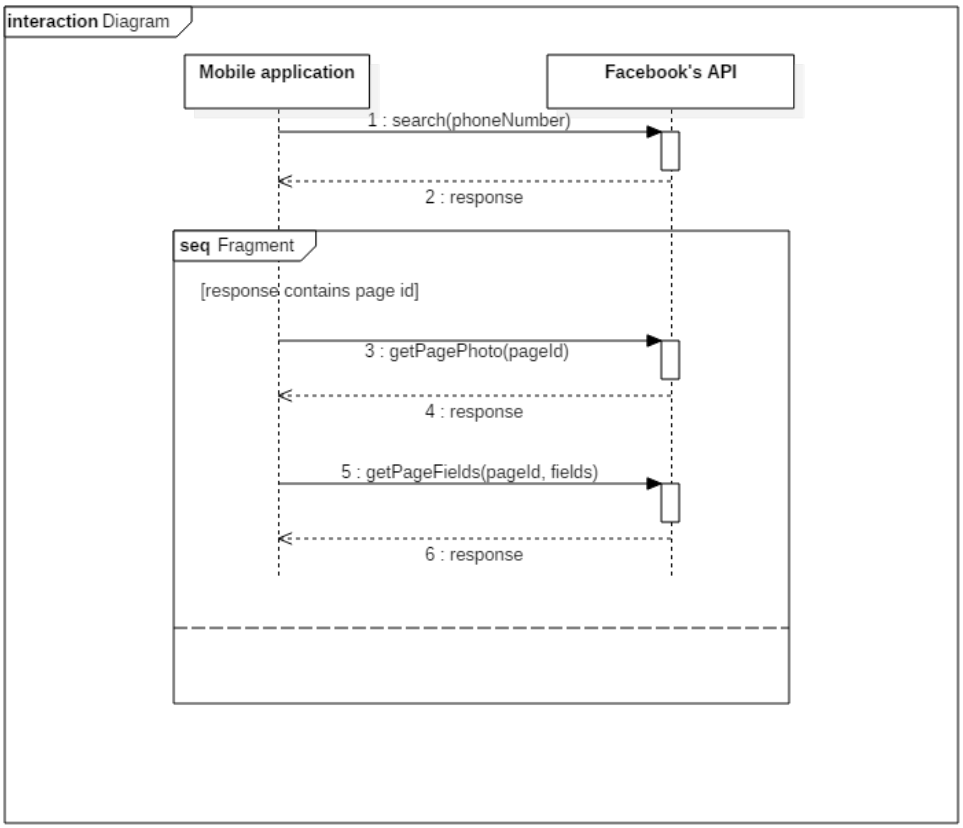


Figure 4.2: Facebook's API sequence diagram

4.4.1.2 Pipl

Regarding Pipl, the prototype makes a general request for information about a phone number. If successful, the prototype checks in the response for information related to the following categories: name, photo, location, job, e-mails and social profiles.

4.4.1.3 FullContact

FullContact allows to search for information about a person using a phone number or e-mail address (it also allows to search for information about a company using a domain address). Therefore, firsly the prototype makes a general request for information about a person or company using a phone number. If successful, the prototype checks for information concerning the following categories in the reponse: name, photo, location, job, e-mails and social profiles.

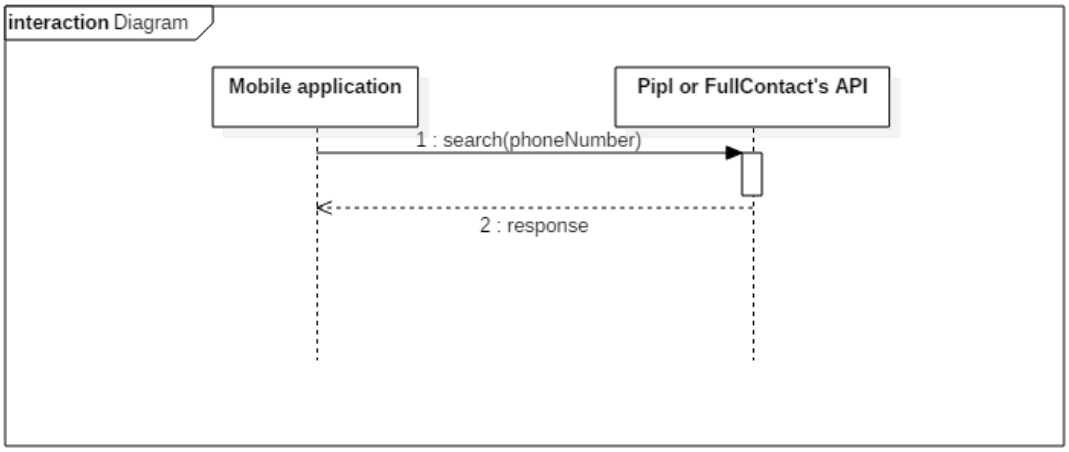


Figure 4.3: Pipl / FullContact's API sequence diagram

4.4.2 Included private data providers

Many companies store information about their customers including personal information like their birthday or interests and historical information like their previous calls notes or purchases. To help them record this information many use customer relationship management software. There are several CRM softwares available on the market but it is also possible for a company to use a custom built software (this option having the disadvantage of making it harder to integrate with other systems).

Some CRM softwares offer a ready-to-use web API to access the information stored through the internet. These information requests require some sort of token that is provided when successfully logged in with a valid username and password pair in order to protect users' privacy.

The URI and request parameters can differ a lot from CRM software vendor to CRM software vendor; therefore a middle layer of communication was built in order to make the app requests

more homogenous. The app communicates with this middle communication layer which communicates with the CRMs' APIs.

The mobile phone's phonebook is also an important private data provider providing data about a contact and call history information with a phone number as well as the app's developed database which allows to store notes associated with a phone number even if the phone number is not a contact. In fact the device itself is the most reliable source of information being the first place to look for information about a phone number.

4.4.2.1 Mobile Phone

There is some information that can be retrieved from the data stored on a mobile phone namely contacts' related data and call history. Regarding contacts' related data, it is possible to retrieve the data associated to a device's contact such as the name, photo or e-mail address by querying the app's content resolver using ContactsContract's class ContactsContract.CommonDataKinds class that contains definitions of common data types. As an example, to retrieve a contact's e-mail address the following piece of code is used:

```
1 Cursor contactEmailCursor = context.getContentResolver().query(ContactsContract.
      CommonDataKinds.Email.CONTENT_URI, null, ContactsContract.CommonDataKinds.Email
      .CONTACT_ID + " = ?", new String[]{contactId}, null);
2
3 while (contactEmailCursor.moveToNext()) {
4 String email = contactEmailCursor.getString(contactEmailCursor.getColumnIndex(
      ContactsContract.CommonDataKinds.Email.DATA));
5
6 // Do something with data
7 }
```

By accessing call history's data and analyzing it, it is possible to generate contextual information about a phone number like if the last call was missed and, in that case, how many successive calls were missed. It is also possible to know when was the last phone conversation with that number by searching for the last call received or made that was answered (if no entry is found then the contextual information is that there are no records of previous phone conversations with that number) and this data can be retrieved using the CallLog provider and the CallLog.Calls class also supplied by Android as in the following piece of code:

```
1 Cursor cursor = context.getContentResolver().query(CallLog.Calls.CONTENT_URI, null,
       null, null, null);
2
3 if (cursor.moveToLast()) {
4 do {
5 String callNumber = cursor.getString(cursor.getColumnIndex(CallLog.Calls.NUMBER));
```

```
6  Date callDate = new Date(Long.valueOf(cursor.getString(cursor.getColumnIndex(
       CallLog.Calls.DATE))));
7  String callType = cursor.getString(cursor.getColumnIndex(CallLog.Calls.TYPE));
8  String callDuration = cursor.getString(cursor.getColumnIndex(CallLog.Calls.DURATION
       ));
9
10 // Do something with data
11 }
12 while (cursor.moveToPrevious());
13 }
```

Also, the developed prototype allows the user to write notes associated with a phone number (even if the phone number is not a contact on the device's phonebook) storing them on a local database being possible to edit or delete a note. This local database is stored as a SQLite database that contains one table with the following columns: the phone number the note is associated to, the date when it was stored and its content. In order to retrieve the notes associated to a phone number, a query that selects the rows where the input phone number matches the phone number's column is made.

4.4.2.2 CRMs

As previously referred, a middle layer of communication was built in order to make the app requests to different CRMs' APIs more homogenous. In order to achieve this homogeneity four endpoints were defined:

/<crm>/login: for logging in into a CRM account using an username and a password supplied as a request's parameters. The result message can either be an ok message if successful or an error message otherwise.

/<crm>/search: searches for the existence of a phone number supplied as a request's parameter on the logged in account's contacts. If a contact is found, its data such as name, phone or e-mail are sent in the response; if not, a message indicating that no contact was found is sent.

/<crm>/taks: returns a list of taks associated to a phone number supplied as a request's parameter on the logged in account.

/<crm>/notes: returns a list of notes associated to a phone number supplied as a request's parameter on the logged in account.

These endpoints were defined based on CRMs usually having a username-password pair system to access an account and that customers are stored as contacts which can have taks and notes associated to them.

Salesforce and Zoho, two of the most used CRMs in the world, were both integrated in the middle layer as a proof of concept. Both provide a simple (and powerful) REST API to access an organization's information. In the case of Salesforce, its API uses standard objects that represent

database tables; for example, the contact object represents a contact. The term “record” describes a particular occurrence of an object being analogous to a row in a database table. Regarding Zoho, its API has a well-defined set of methods that can be used with a well-defined set of modules such as Contacts with data being extracted in XML or JSON format.

Although both CRMs mentioned in the previous paragraph have an official API, there are several libraries available on the internet in a variety of programming languages that capsulate the access to the official APIs. In the developed middle layer of communications, a Javascript library called JSforce was used to communicate with Salesforce’s official API since it offers access to most of the official API’s features and the fact of it being Javascript code makes the integration with Node.js cleaner. As an example, to search for a phone number on a Salesforce account’s list of contacts and retrieve its data the following piece of code is used:

```
app.post("/salesforce/search", function(request, response) {

var phone = request.body.phone;

jsforceConnection.login(username, password, function(err, userInfo) {
if (err) {
return console.error(err);
}

jsforceConnection.query("SELECT Id, Name, Phone, MobilePhone, Email, MailingCity,
    MailingCountry, Title FROM Contact WHERE Phone='" + phone + "' OR MobilePhone='
    " + phone + "'", function(error, result) {
if (error) {
response.send(error);
}
else {
if(result.records.length > 0) {
response.json(result.records[0]);
}
else {
response.json({ Error: "Contact not found" });
}
}
});
});
});
```

On the other hand, in the case of Zoho, its official API was used directly in the middle layer of communications since no Node.js library with all the Zoho’s official API’s features was not found. Also, as an example of how to search for a phone number on a Zoho account’s list of contacts and retrieve its data, the following piece of code is used:

```
app.post("/zoho/search", function(request, response) {

var phone = request.body.phone;

requestLib( {
url: "https://crm.zoho.com/crm/private/json/Contacts/searchRecords?authtoken=" +
    authToken + "&scope=crmapi&criteria=((Phone:" + phone + ")OR(Mobile:" + phone +
     "))",
json: true
}, function(err, res, json) {
if (err) {
throw err;
}

if(json["response"]["result"] != undefined) {

var jsonData = {};

for(var i = 0; i < json["response"]["result"]["Contacts"]["row"]["FL"].length; i++)
     {
var key = json["response"]["result"]["Contacts"]["row"]["FL"][i]["val"];
var value = json["response"]["result"]["Contacts"]["row"]["FL"][i]["content"];
jsonData[key] = value;
}

response.json(jsonData);
}
else {
response.json({Error: "Contact not found"});
}
});
});
```

The developed middle layer of communications allows the integration of other CRM vendors in the future. In order to integrate a new CRM vendor, the four endpoints previously mentioned should be implemented being that, the implementation of each one of them depends on the API supplied by each vendor; also, if a library is available in the Javascript programming language offering the access to most of the official API's features it is preferable to making requests directly to the official API.

4.4.2.3 Chain information collection

The developed prototype uses information found on a source to search on another; for istance, if an e-mail or website is collected from any source other than FullContact, it is used to search for more information on FullContact since it allows searching for information using an e-mail or domain address as a parameter and the responses to a search can differ when using different input parameters.

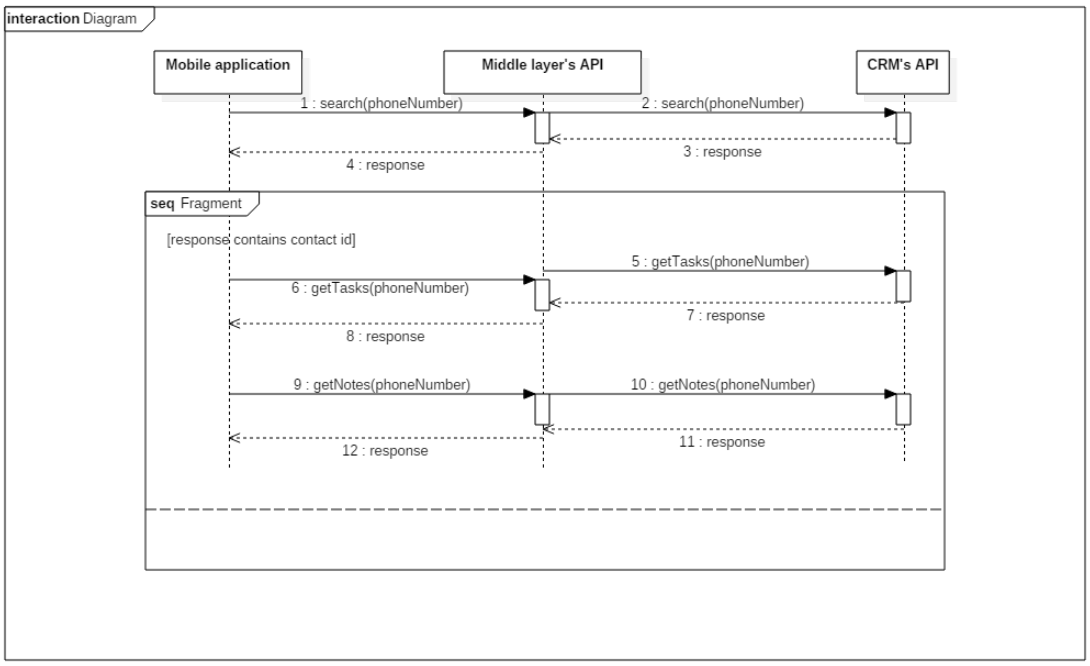


Figure 4.4: CRMs' APIs sequence diagram

4.5 Information processor

Requesting information from multiple sources at the same time can cause some conflicts so it becomes necessary to have an entity responsible for managing the information in order to avoid these conflicts. Therefore, an information processor was developed and all the information obtained from the different sources should go through this entity.

The information processor receives the information from the sources and saves the data itself along with the source it came from; if, for some reason, the same information is received from the same source it is ignored. An algorithm then runs to compare the received information with the previously saved information evaluating what to display, which is explained more in subsection 4.5.2.

4.5.1 Information categories

There are different categories of information that can be collected concerning a phone number. The developed prototype implements a system of categories that includes: name, photo, location, email, website, job, company information, notes, tasks, social profiles' urls, number of likes on Facebook, last post on Facebook and call history. Each of this categories has a correspoding array where the data itself and the source it came from are stored as a pair of strings as shown in the following piece of code:

```
1 ArrayList<MyDataSourcePair> callerNames;
```

```
2 ArrayList<MyPhotoSourcePair> callerPhotos;
3
4 // ...
```

4.5.2 Information comparator algorithm

The developed information processor has a set of methods defined for receiving information related to each of the categories previously referred. For example, the following methods are used to add information related to the name and photo categories, respectively:

```
1  public static void addName(String name, String source) {
2  if(!sameDataFromSameSource(callerNames, name, source)) {
3  callerNames.add(new MyDataSourcePair(name, source));
4
5  // ...
6  }
7  }
8
9  public static void addPhoto(Bitmap photo, String source) {
10 // ...
11
12 callerPhotos.add(new MyPhotoSourcePair(photo, source));
13 }
```

As mentioned before, when the information processor receives an information from a source it checks if it was not previously received from the same source and, if not, saves the data itself and the source as a pair of strings in the array correspoding to the information's category. For example, when the processor receives an information about the name associated with a phone number, it checks if that information was not previously sent by that source and, if not, adds it to the names category array.

When an information is received by the information processor it also checks for similarity with previously stored data of the same category. It uses the Levenshtein distance algorithm that calculates the distance between two strings where the result is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other to check for similarity between informations in the same array; there is a maximum distance accepted for the information to be considered the same being normalized if that is the case.

After adding a pair of information to the respective array, the algorithm starts searching for the data that appears the greatest number of times in the array (remember that the same data from the same source is ignored by the processor not biasing the results and that the data is already normalized) displaying the data that appears the greatest number of times.

4.5.3 Information ranker algorithm

As the information is being selected by the information comparator algorithm it is also being ranked by another algorithm. This ranker algorithm determines the order of appearance of the information in the app refreshing the order of information displayed on the screen as the information is being received and processed.

The ranking of the information is based on the relevancy of the received data which is defined by a list of relevancy included in the app as follows:

1. Tasks; 2. Notes; 3. Call history; 4. Location; 5. E-mail; 6. Website; 7. Job description; 8. Company information; 9. Social profiles; 10. Number of likes on Facebook; 11. Last post on Facebook.

In the top three of the list are categories related to information supplied by private data providers; these are followed by the most common information categories relative to both people and companies that can be supplied by both private or public data providers and in the bottom two are categories related to information that is specific to a source (in this case, Facebook).

As each piece of received information is stored in the corresponding array according to its category (e.g. name, location, note, ...), the list of relevancy is consulted in order to obtain the array's category ranking, attributing it to the received piece of information.

4.6 Construction

4.6.1 Android prototype structure

There are three main classes responsible for the interaction with the user: SetupActivity.java for handling the initial setup screens, MainActivity.java for handling the main screens and SettingsActivity.java for handling the settings screens. Also regarding the user interface, the class MyFloatingWindow.java handles all aspects related to the floating window where the contextual information is shown such as its creation and destruction.

Figure 4.5: App's user interface's related classes

An abstract superclass for handling the tasks related to data requests from a data provider was constructed in order to to facilitate the integration of different data providers; every data provider has its own subclass that extends the superclass implementing its abstract methods.

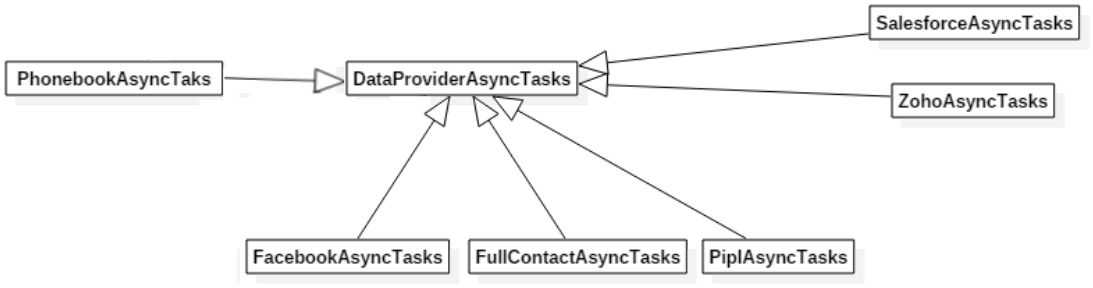


Figure 4.6: App's data providers's related classes

The data collection that happens inside each of the data providers classes resorts to a class responsible for processing the collected information; this class includes a set of methods for processing different types of information (e.g. name, location, etc.) that should be called when a piece of information is found called MyInformationProcessor. As refered in 4.5.2, every piece of data is saved as a pair of strings of the data itself and the source it came from so a class was created to represent such pair as well as a class for representing a pair of an image and the source it came from for the collected photos called MyDataSourcePair and MyPhotoSource respectively..

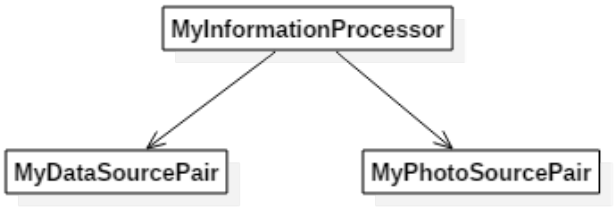


Figure 4.7: App's information processor related classes

Finally, there is also an abstract class responsible for receiving telephony state changes notifications and determining which call event happened (e.g. receiving a call, making a call, received call ended, etc.) called PhoneCallReceiver which extends the Android's BroadcastReceiver class and uses the Android's TelephonyManager class. A class called MyCallReceiver extends the abstract class and handles the call events accodingly; for instance, when receiving a call the method responsible for creating a floating window on the MyFloatingWindow class is called and when a call ends the method responsible for destroying it is called.

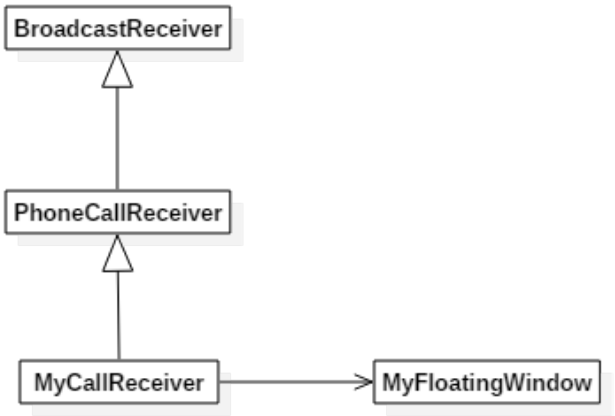


Figure 4.8: App's call receiver's related classes

4.6.2 Floating window details

The floating window where the contextual information is displayed resorts to the Androis's WindowManager.LayoutParams class to set the type of the window as a phone window since these type of windows are normally placed above all applications providing user interaction with the phone which is appropriate for the developed prototype. Also, the flag FLAG_NOT_FOCUSABLE is set on the window so it doesn't get key input focus as it is aimed to display information.

4.7 Privacy concerns

All the data collected by the developed prototype was shared at some point by people or companies with some sort of service (either an online service such as a social network or an offline service such as a company's customers database). These services usually have a terms and conditions agreement that people should agree in order to use it which usually describes what can happen to the information provided.

Also, all the collected data is the responsibility of the corresponding source and the developed prototype does not save any of it besides the notes inserted by the user inside the app (which are saved on a local database).

4.8 Conclusions

Collecting and aggregating information from different sources about a company or person using a phone number as an input presents some challlenges. The first one being where to collect information from, i.e., what sources of information to use. There are some public data providers currently available that provide an API which allows requesting for information using a phone number as input. However, some of these public data providers have some restrictions; for instance, Facebook only allows to search for information about a phone number when it comes to companies' pages, not people. There are also private data providers that companies' should use like customer relantionship managements software which allows them to retrieve information from previous interactions with a customer or a mobile device itself which can have information associated with a phone number.

In addition to the challenges related to data providers there are challenges related to information collection and aggregation as information can be collected from different sources and may even come from both public and private data providers. Therefore, it makes sense to have an entity responsible for processing all the collected information meaning that all of it should go through this entity. As information is received by this entity, it is compared with previous information in order to understand if the information is new and if there is any inconsistency with previous collected information. This entity is also responsible for ranking the collected information which determines the order of display on a mobile device's screen.

Solution

Chapter 5

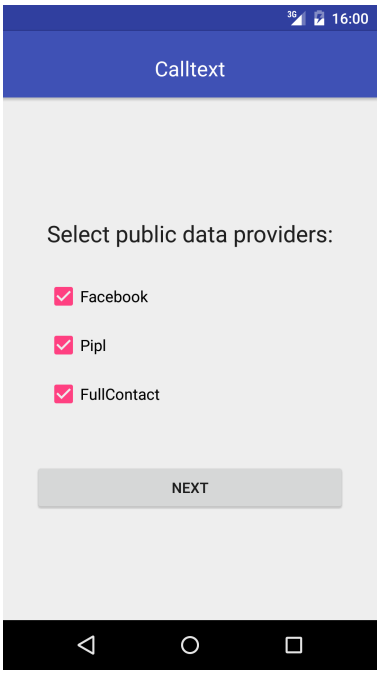
Results

A mobile application prototype for Android smartphones was successfully developed that searches, collects and aggregates information (when found) from different sources about a phone number, either assocatiated to a person or a company, displaying it on the device's screen; the app can be triggered by searching for a phone number inside it or when receiving or making a phone call.

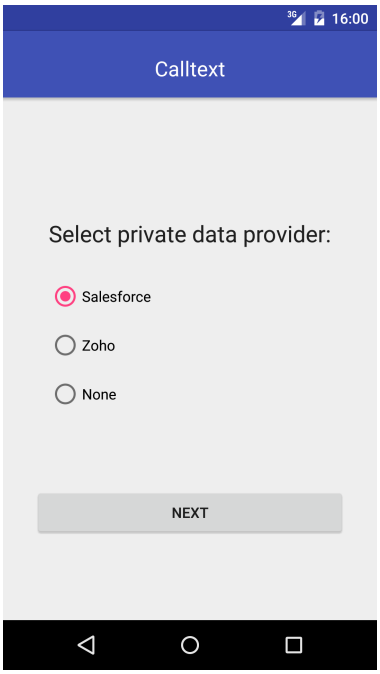
5.1 App's walkthough

5.1.1 Setup screens

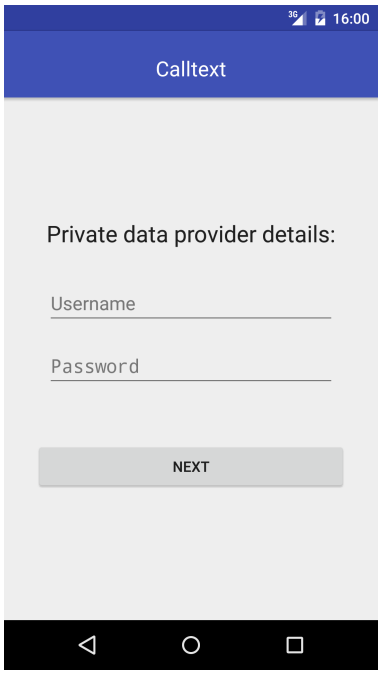
When opening the app for the first time, a set of setup screens are presented. The first setup screen lets the user choose what public data providers to use by selecting the corresponding checkboxes. After pressing next, a second setup screen is displayed regarding the use of a private data provider namely a customer relantionship software. The user can choose from not setting up a connection or select a CRM vendor from the list. If a CRM vendor was selected a third screen is displayed asking the user for the username and password in order to login and establish a connection to the selected private data provider; if no CRM was selected or after successfully logging in to the selected private data provider, a final setup screen is displayed informing the user that the setup process is complete with a button to finish the process.



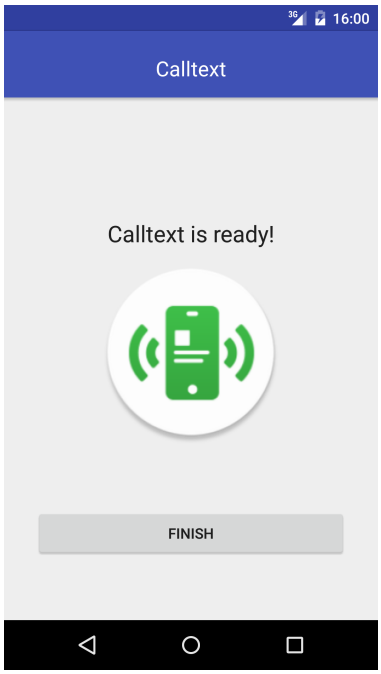
(a) Setup screen: public data providers selection



(b) Setup screen: private data provider selection



(c) Setup screen: private data provider login



(d) Setup screen: setup complete

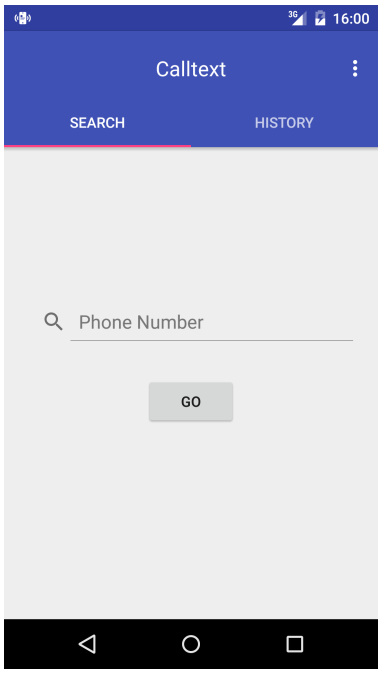
Figure 5.1: App's setup screens

5.1.2 Main screen

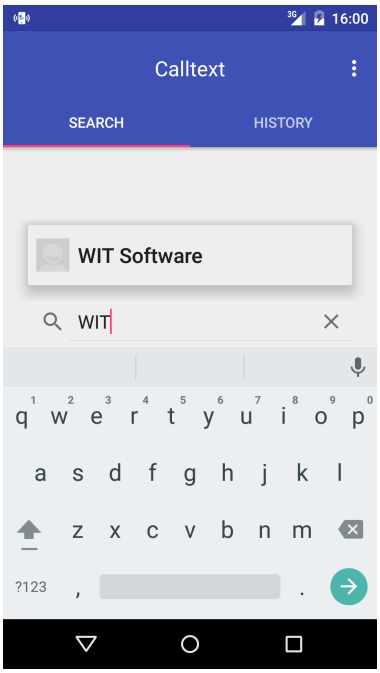
After concluding the initial setup, the main screen of the app is presented. In this screen there are two tabs the user can choose from: search and history (being the search tab selected by default). There is also an icon next to the app name on the action bar where the user can click to access a menu that presents only one option: settings.

When the search tab is selected, the user is presented with a search box where a phone number or the name of a contact on the device's phonebook can be inserted to search for contextual information associated with it; when inserting a phone number or the name of a contact, if the input matches any entries on the device's phonebook, these entries are shown as suggestions. After inserting a phone number and then clicking on the search button located underneath the search box, a floating window appears with information associated to the inserted phone number (this window is described more in detail in subsection 5.1.4).

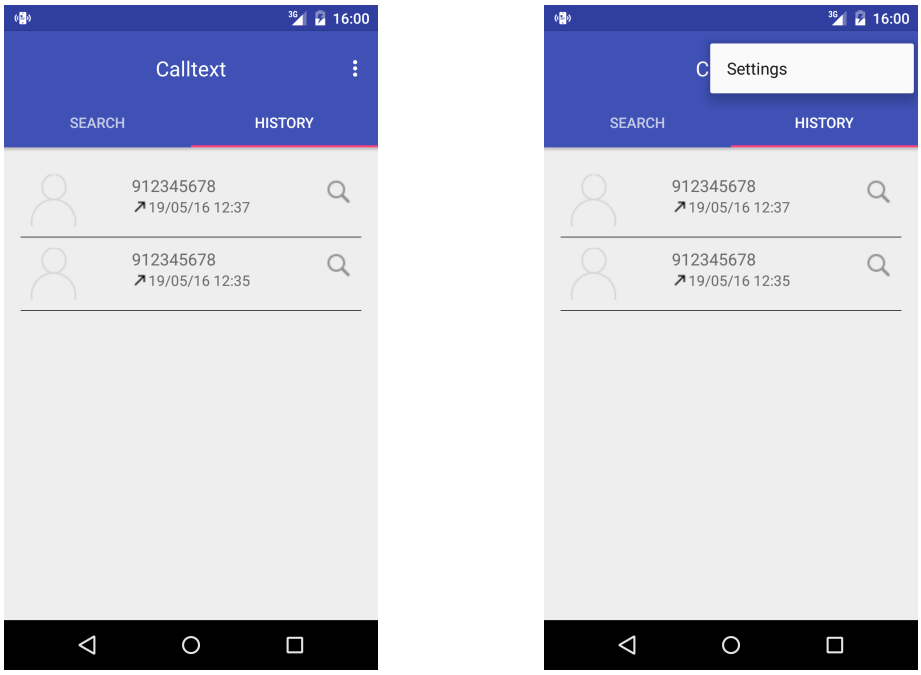
When the history tab is selected on the main screen the user is presented with the device's call history entries; each of these entries present the phone number associated with the call or a name if the phone number is on the device's phonebook, the call's date, an icon representing the call's type (received, made or missed), a photo associated with the phone number (if no other photo is available, the device's default contacts photo is displayed) and a magnifying glass icon that when clicked opens a window with information associated to that entry's phone number (this window is described more in detail in subsection 5.1.4).



(a) Main screen: search



(b) Main screen: search suggestions



(c) Main screen: call history

(d) Main screen: menu

Figure 5.2: App's main screen

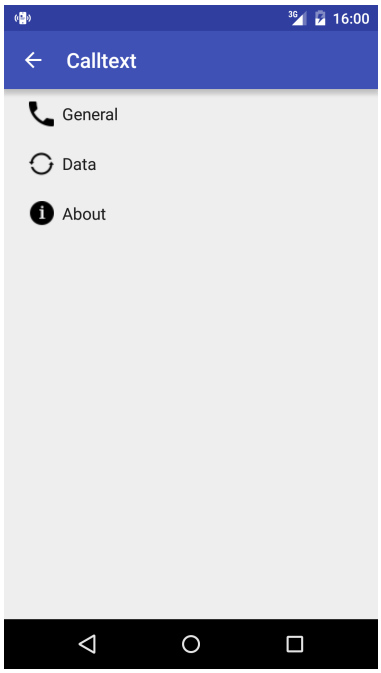
5.1.3 Settings screens

When the settings option is selected from the main screen's menu the user is presented with a main settings screen. From this screen the user can choose to go to the general settings screen, data providers settings screen or the about screen. Also, from the main settings screen, if the user clicks on the icon next to the app's name on the action bar the app returns to the main screen.

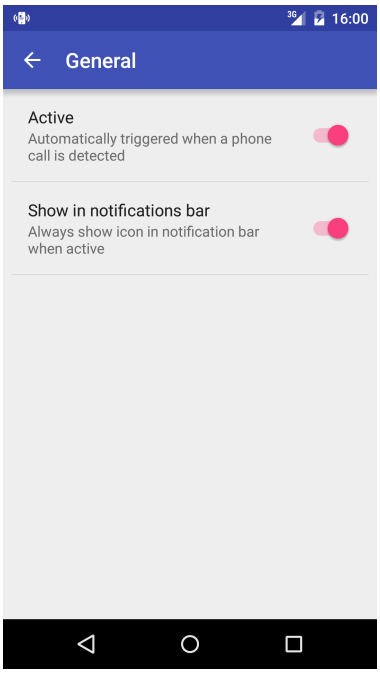
In the general settings screen it is possible to activate or deactivate the app which defines if the search for information associated to a phone number is automatically triggered when a phone call is detected. From this screen, it is also possible to enable or disable the app's icon on the notifications bar.

In the data providers settings screen it is possible to enable or disable the use of public data providers and/or a private data provider and if enabled, choose what data providers to use; if a private data provider is selected, an option to login (after filling in the username and password fields) or logout from a CRM's account is also presented.

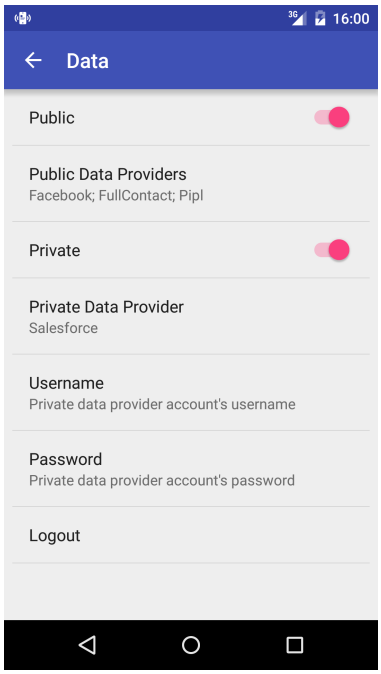
Finally, in the about settings screen it is possible to check the app's version as well as other app's related information.



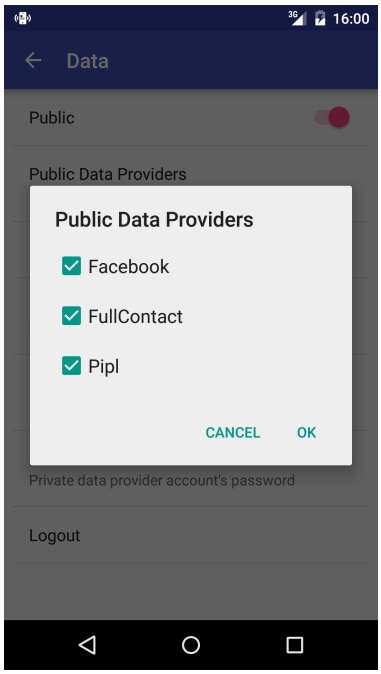
(a) Settings screen: main scren



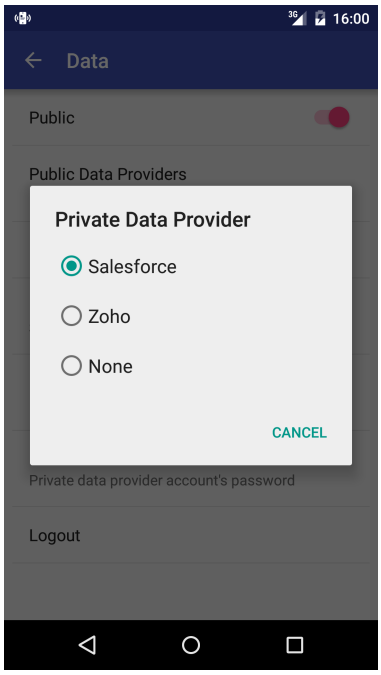
(b) Settings screen: general settings creen



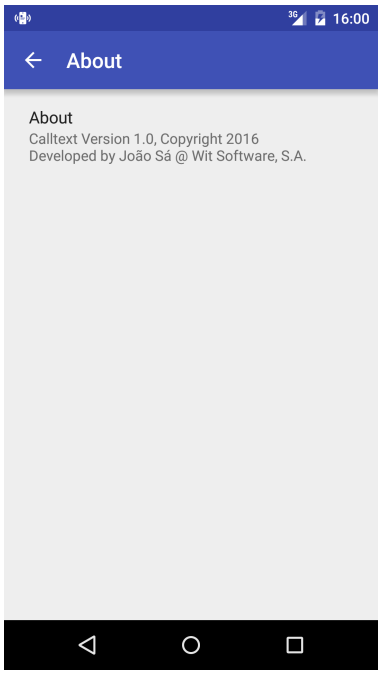
(c) Settings screen: data providers screen



(d) Settings screen: public data providers selection



(e) Settings screen: private data provider selection



(f) Settings screen: about screen

Figure 5.3: App's settings screen

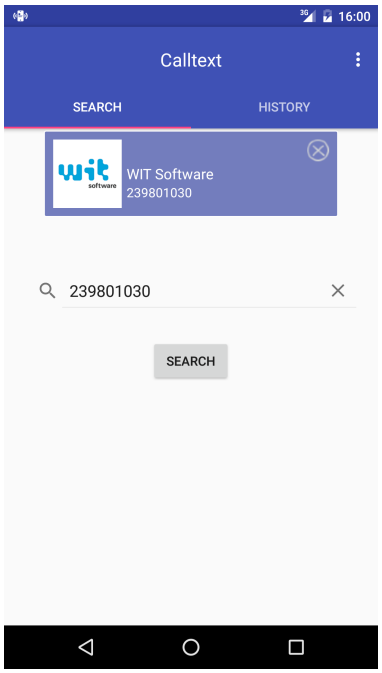
5.1.4 Contextual information floating window

When the app is active and a phone call is detected or when the user searches for a phone number from the main screen, a floating window appears on the screen on top of all other screens including the call screen. This floating window can be moved by long pressing it and then dragging it to the desired position.

The phone number, name and photo associated with it are displayed on the floating window and more information can be shown (or hidden if is already being shown) by clicking on the photo being displayed on an extension of the floating window. Each piece of data (location, email address, website, etc.) is displayed in a box where the background color represents the source of the information (for example, dark blue for Facebook and light blue for Pipl).

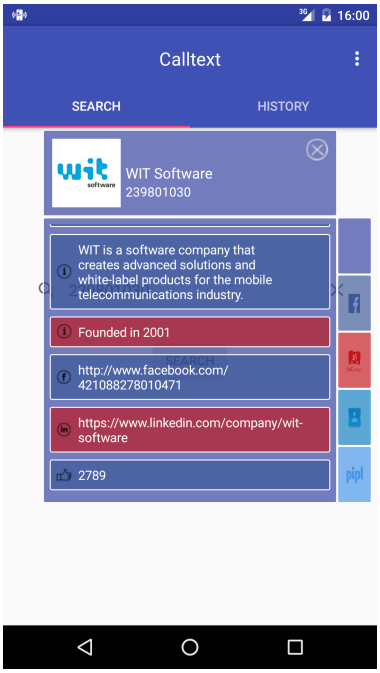
The user should be able to scroll up/down in the extension of the floating window to see other information boxes and swipe left/right to choose to display data from a specific source of information. When displaying data from a specific source of information, the background color of the extension of the floating window should change to the color associated with that source. The amount of data from a specific source of information is variable (including none).

If the floating window results from searching for a phone number inside the app's main screen, an icon to close it is displayed on the top right corner of it; instead, if the floating window results from detecting a phone call an icon to minimize it is displayed on the top right corner of it that when clicked makes the floating window disappear and a floating app's logo appear which, in turn, when clicked disappears and makes the floating window appear again.

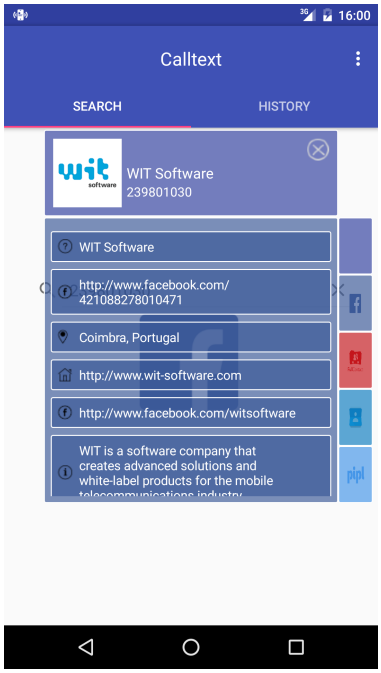


(a) Contextual information floating window: name and photo found associated with a phone number

(b) Contextual information floating window: information found associated with a phone number from different sources displayed on the window's extension (after clicking on the photo)

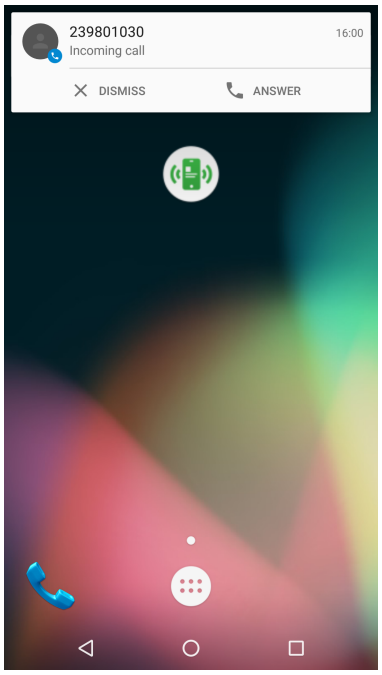


(c) Contextual information floating window: information found associated with a phone number from different sources displayed on the window's extension (after scrolling down)



(d) Contextual information floating window: information found associated with a phone number from a specific source displayed on the window's extension (after swipping left)

(e) Contextual information floating window: automatically displayed after detecting a phone call

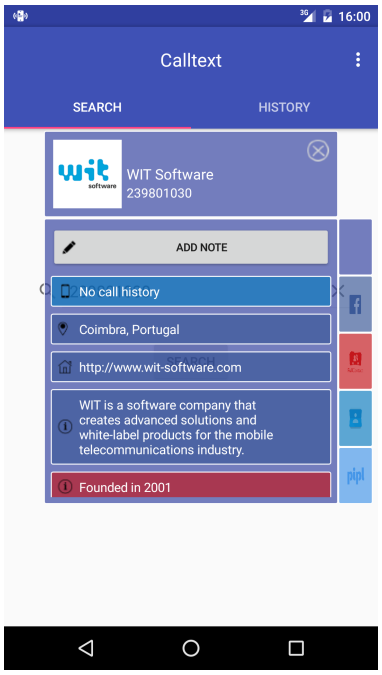


(f) Contextual information floating window: minimized

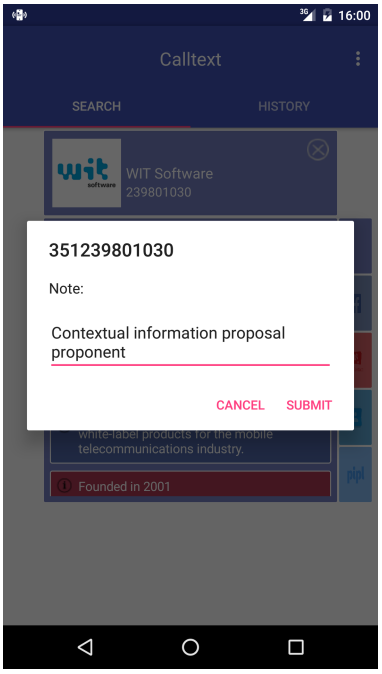
Figure 5.4: App's main screen

5.1.4.1 Add notes associated with a phone number

When the contextual information floating window is extended there is a button always available on the top of the window's extension that allows the user to add notes associated with the phone number the window reffers to. Each time this button is clicked a pop-up dialog appears in order for the user to insert a note. After saving a note, it is displayed, along with the date and time it was created, next to other information collected on the floating window's extension; by long pressing a note, the user can edit or delete it.

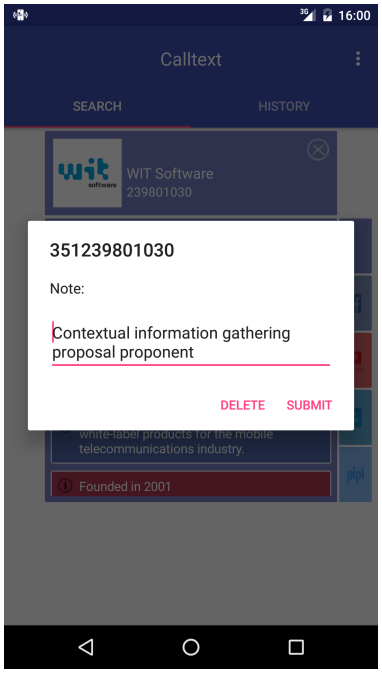


(a) Add note's button on the extension of the contextual information floating window



(b) Note's submission pop-up displayed after clicking the add note's button on the extension of the contextual information floating window

(a) Note being displayed on the extension of the contextual information floating window after being submitted



(b) Note's edition/deletion pop-up displayed after clicking the note's box on the extension of the contextual information floating window

Figure 5.6: App's notes feature

5.2 App's usage example scenarios

In order to measure the results of the presented solution, a collection of scenarios were put in practice, three of them being detailed next.

5.2.1 Displaying information about a company

In this scenario all public data providers available on the app were turned on (Facebook, Pipl and FullContact) and the phone number of WIT Software was used as input. No private data provider besides the device's phonebook (which didn't have the number as a contact) and the app's local notes database (which didn't have any notes previously saved) was used, i.e. no CRM connection was set up.

The app was able to identify the phone number as being associated to WIT Software displaying several other informations including the company's logo, location, web address, description, year of foundation, Facebook's and LinkedIn's addresses, number of likes on Facebook or the last post on Facebook (see figure 5.4). From Facebook the app collected the company's location, web address, description, Facebook's address, number of likes on Facebook and last post on Facebook while from FullContact it also collected the description and Facebook's address plus the year of foundation and LinkedIn's address; no information was collected from Pipl. Therefore, there was some information (e.g. company's description) that was available from more than one source and information that was only available from each of them which was aggregated by the app.

Regarding the device's phonebook, as the phone number was not associated to any contact on the device and no notes previously saved on the app's local database associated with that phone number, no information was displayed other than not existing previous calls which resulted from searching for the phone number on the call history.

5.2.2 Displaying information about a person

In this scenario all public data providers available on the app were turned on (Facebook, Pipl and FullContact) and the phone number of a WIT Software's collaborator was used as input. No private data provider besides the device's phonebook (having the phone number as a contact) and the app's local notes database (which didn't have any notes previously saved) was used, i.e. no CRM connection was set up.

The app was able to identify the phone number as being associated to the expected person displaying several other informations including the location, e-mail address, job and Facebook's and LinkedIn's addresses. From Pipl the app collected the person's name, location and e-mail address while nothing was found using the phone number as input on FullContact. On the other hand, using the collected e-mail address from Pipl the app was able to find more information from FullContact including the person's name, a more complete location, job and Facebook's and LinkedIn's addresses; no information was collected from Facebook although the address was known because of the Facebook's API restrictions concerning data gathering from people's pages. Therefore, there

was some information that was available from more than one source and information that was only available from each of them which was aggregated by the app.

Regarding the device's phonebook, as there were no details filled in associated to the contact on the device and no notes previously saved on the app's local database associated with that phone number, no information was displayed other than the date of the last call with that number (including the duration) which resulted from searching for the phone number on the call history.

5.2.3 Displaying information from a CRM

In this scenario all public data providers available on the app were turned on (Facebook, Pipl and FullContact) and a connection to a Zoho account having the WIT Software's phone number as a contact was set up being that phone number used as input.

The app displayed the same information as in the first scenario plus information provided by the connection with the customer relantionship management software account namely notes associated with that phone number.

Figure 5.7: Information from Zoho CRM being displayed on the floating window's extension

5.3 Hidden information

As previously referred in this chapter, many social networks such as Facebook have some restrictions when requesting information in order to protect their users' privacy such as not allowing to search for a phone number associated with a person (unlike pages belonging to companies). Therefore, although a Facebook profile associated with a phone number can be found by searching for a number on a data provider, it is not possible to collect the public information from that profile without the person's explicit authorization; and as the mobile application prototype is intented to be used by the party seeking for information and not the party searched for, this authorization cannot be granted.

The public information available from an online social profile that is not shown by the prototype due to users' privacy protection restrictions can however be accessed by clicking on the information box that displays a person profile's url; the click will open the selected url in the mobile device's browser and the public information available from that profile can then be seen.

5.4 Respect for related patents

The developed prototype was developed from an original idea of gathering information from different information providers, both public and private, using an internet connection and thus does not violate any of the patents referred in chapter 2. Also, despite the topic being related, the methods and system presented in this chapter differ from those presented in the previously mentioned patents.

5.5 Conclusions

There is a huge and increasing amount of information available nowadays about people and companies available from different sources, either public or private; thus, by integrating different data providers on a mobile application prototype it was able to collect and aggregate information from different data providers about most of the phone numbers tested.

Finally, it is important to refer that, if an internet connection is stable, the information should start appearing on a smartphone's screen just a few seconds after initiating a search; the displayed information is updated as it is collected and processed by the app.

Chapter 6

Conclusions

In conclusion, there is infomation available from different sources of information that could be used by both people and companies to help contextualize a phone call. Thus it makes sense to have a solution capable of collecting and aggregating the information from different sources of information and display it in a mobile device when needed since most mobile phones, namely smarthphones, have internet access.

To achieve the goal referred in the previous paragraph a prototype of an application for Android smartphones was presented and developed as a proof of concept. The prototype functions in a transparent way for the user as it is triggered automatically when a a phone call is detected altough having also the option of explicitly searching for information associated with a phone number inside it.

6.1 Preliminary work

A research in order to understand what was already published and available on this the topic of contextual information on mobile phones was conducted. This research showed that there are already some mobile applications available on the market that are able to identify and present some information about the caller however most of them are intended to personal use and do not contain any historical data resulting from previous interactions with the caller.

Also a research in order to undestand which sources of information were available and had the characteristics needed to fit in a proof of concept prototype was conducted. This research was primarily focused on online social networks (e.g Facebook) and online people/companies search engines (e.g. Pipl or FullContact).

A preliminar work about the technical, functional and visual requirements of the mobile application was also done.

6.2 Achievements

As mentioned in the introduction of this chapter, a mobile application prototype for Android smartphones was developed as a proof of concept of the presented solution. This prototype tries to collect and aggregate information associated with a phone number from different data providers displaying it on the screen a phone call is detected or when searching for information about a number inside the app.

Also, as a result of the work done it was possible to show that it is possible to link different information providers by using the information collected on an information provider as an input to search on another one; this was implemented in the prototype app which includes an information processor for aggregating the information collected from different sources.

The prototype's information providers include the device itself (by searching for information in the call history, the phonebook or notes associated to a phone number in the app's local database), online public data providers (social networks APIs or people/companies APIs) and companies' customer relantionship softwares.

6.2.1 Degree of satisfaction

Overall, the degree of satisfaction concerning the developed prototype is high. The app was able to display information about most of the phone numbers tested even if in most cases it only displayed the name, photo or location (mostly the country) besides the information retrieved from the device's call history and notes stored in the app's local database.

Regarding the app's user experience the degree of satisfaction is also high since it is self-explanatory being easy to use.

6.2.2 Possible improvements

The developed prototype could use some improvements with regards to the amount of data providers available i.e. more public data providers could be added as well as private data providers namely integration with other customer relantioship management software vendors.

6.3 Future work

An in-depth study about how can contextual information sharing become a standard in telecommunications can be done. This includes understanding why it is not a standard today when technologies are suffering major advances and what is mobile networks' role regarding this.

References

[AB] True Software Scandinavia AB. Truecaller app. http://www.trucaller.com.

[Cla03] Edward O. Clapper. Providing information to facilitate telephone conversations, 08 2003.

[EA13] Dror Zelber Elinor Axelrod. Mobile contact notes, 10 2013.

[GA05] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. *WPES '05 Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, 2005.

[KSHS13] Johannes Knittel, Alireza Sahami Shirazi, Niels Henze, and Albrecht Schmidt. Utilizing contextual information for mobile communication. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pages 1371–1376, 2013.

[Lju01] Peter Ljungstrand. Context awareness and mobile phones. *Personal and Ubiquitous Computing*, 5:58–61, 2001.

[MHG14] Jr. Mark H. Gosselm, Thomas P. Maliska. Methods and systems for improved caller name identification on a telephone network, 02 2014.

[NAA03] James E. Gauger Nandakishore A. Albal. Methods and apparatus to deliver identification information, 08 2003.

[NPA09] Raymond Paul Vander Veen Neil Patrick Adams. Enhanced phone call context information, 06 2009.

[Staa] Statista.com. Market share worldwide smartphone shipments by operating system from 2014 to 2020. http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems//.

[Stab] Statista.com. Number of smartphone users worldwide from 2014 to 2019 (in millions). http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[Syn] Sync.ME. Sync.me app. http://www.sync.me.

[Teca] TechCrunch.com. Eric schmidt: Every 2 days we create as much information as we did up to 2003. http://techcrunch.com/2010/08/04/schmidt-data/.

[Tecb] TechCrunch.com. Facebook speeds past 1.55 billion users and q3 estimates with 4.5b revenue. http://techcrunch.com/2015/11/04/facebook-earnings-q3-2015/.

REFERENCES

[Tul07] Apaar Tuli. Presenting to a user contextual information regarding a communication before the user takes action concerning the communication, 12 2007.