Internship Report

Masters in Computer Engineering Mobile Computing

Business Oriented Applications for Android Platform

André Ferreira de Sousa

Leiria, September 25, 2013

Internship Report

Masters in Computer Engineering Mobile Computing

Business Oriented Applications for Android Platform

André Ferreira de Sousa

Internship report for Master degree performed under the guidance of Doctor Marco Monteiro, Professor at School of Technology and Management of the Polytechnic Institute of Leiria and co-orientation of Engineer Rui Macedo, Paulo Sousa and Rui Gil from WIT Software, SA.

Leiria, September 25, 2013

To My Family

This page was intentionally left blank

Acknowledgment

My thanks go firstly to my advisor, Professor Marco Monteiro by extraordinary sense of project management advice and guidance along the development of the report, and my supervisors inside WIT, the engineers Paulo Sousa, Rui Macedo and Rui Gil, for the support on the development of the solutions.
I also want to thank my family, in particular my parents, friends and siblings for their support and patience.

In all, my sincere thanks.
André Sousa

This page was intentionally left blank

Abstract

Given the current growth of mobile devices and new concepts of portability, the need for mobile applications focused on the business environment is increasing. These applications take advantage of the mobile devices capabilities in order to help everyone involved in a project to easily chat and discuss ideas. This report presents the work taken developing business solutions for Android operating system, with the purpose of letting the users chat, send files, search content and make audio/videos calls. A powerful component, which we have called "tags", that lets the user contextualize its data, will also be presented.

Keywords: business collaboration, android, tablet, chat, audio, video, tags

This page was intentionally left blank

List of Figures

List of Tables

This page was intentionally left blank

List of Abbreviations

API	Application Programming Interface
B2BUA	Back-to-Back User Agent
GPS	Global Positioning System
GSMA	Global System for Mobile Communications Association
HD	High Definition
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IETF	Engineering Task Force
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
JAIN-SIP	Java APIs for Integrated Networks SIP
JSON	JavaScript Object Notation
MMS	Multimedia Messaging Service ()
OMA	Open Mobile Alliance
OS	Operating System
OSI	Open Systems Interconnection
OTT	Over-The-Top
PSTN	Public Switched Telephone Network
RCS	Rich Communication Services
REST	Representational State Transfer
RFC	Request For Comments
SDK	Software Development Kit
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMS	Short Message Service
TCP	Transmission Control Protocol
UA	User Agents
UAC	User Agent Client
UAS	User Agent Server
UC	Use Case
UDP	User Datagram Protocol
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VoIP	Voice over IP

XML	Extensible Markup Language

Index

This page was intentionally left blank

Chapter 1 - Introduction

Nowadays, network infrastructures of mobile operators that provide 3G or 4G are based on IP Multimedia Subsystem (IMS) architecture. IMS divides the network into logical components, and allows new communication services to be developed using the standard protocol Session Initiation Protocol (SIP). WIT Company aims to develop a communication application using this type of architecture in such a way that it provides services to end-users, advanced telephony and instant messaging. This document will present the developed work and the acquired knowledge during the internship, conducted within computer engineering mobile computing master course of the school of technology and management of the polytechnic institute of Leiria, in the academic year 2012/2013. The internship took place at the company WIT Software, SA, in Leiria, from September 2012 to May 2013.

1.1 Internship Goals

The objective of the internship was to develop a prototype of a business collaboration application (from now on called "business prototype") for an Android Tablet and to implement share content capabilities based on the user localization (from now on called "share interests"). Business collaboration applications are focused on the enterprise market, helping a business to accomplish specific goals, such as improving the communication and the exchange of ideas between different people involved in a project [1] [2].

The business prototype connects to an IMS network and uses SIP communication protocol to give end-users telephony and instant messaging. The key features to be implemented include chat, audio and video calls on single and conference mode, call recording, history log and a form of information contextualization. The User Interface (UI) is also an important part, namely the contact list, chat list, chat messages and video/audio calls. Communication between clients is made through a server that will manage all the actions and the exchange of data.

The second internship goal, share interest, was to implement into an Android application developed by WIT Company, called Joyn [3]. This involves the study and usage of a server

produced by Sprylogics enterprise [4] that retrieves interest suggestions to the user based on his location, such as weather, movies, points of interest and YouTube videos.

1.2 Motivation

"The Future of Mobile Is the Future of Everything" - Matt Galligan (co-founder of SimpleGeo) [5]

It is practically impossible not to be aware of the exponential growth of mobile devices. Every week new devices shipped with complex operating systems and the latest mobile hardware are launched. These so-called smartphones are considered as handheld computers, which have multiple features that let them run applications. They present user-friendly input method, the touch screen, enabling the user to perform actions with just one or more fingers, opening up a whole new way of navigating and interacting. Now we can tap, scroll, swipe, scale, rotate and move elements, by simply touching the device [6].

As these new interactions applications are becoming more complex, handling computing needs traditionally reserved for PCs, a more immersive User Experience (UX) to the mobile devices is brought. This led to an exponential growth of the number of applications available and, consequently, the need of making them accessible to every user has increased. With this problem in mind, online application stores were created, such as Google Play from Google and App Store from Apple. These stores let every developer reach a worldwide market in a very short period of time, opening an endless number of business opportunities and new ways of making a great revenue. All in all, producing a mobile application is a great opportunity that we don't want to miss.

1.3 Report Structure

This document is structured in the following way: in chapter 2 (background) we will describe the structure of WIT Company, my role in that structure and the list of tasks assigned to me during the internship. Next, in chapter 3 (planning, risks and methodology) we will explain the development plan and the methodology that was used. In chapter 4 (state of the art), the most relevant competitors in business collaboration environment, Android usability for this type of applications and required technologies are presented. In chapter 5 (product backlog) we will address the requirements functionalities and user stories for the business prototype. Chapter 6 (implementation) will cover the work concerning the business prototype implementation, its

characteristics and the integration of share interests based on the user localization into an existing Android application. Finally in section 7 (conclusions), we will present the conclusions and future work analysis.

(Starts on page odd)

Chapter 2 - Background

This chapter describes the internal structure of WIT Company, the business areas and my role in that structure as an intern. It also describes the tasks carried out during the development of each internship goal, the time to perform each task and, finally, a critical analysis of the whole internship.

2.1 WIT Software, SA

WIT Software is a company that was established in March 2001 as a spin-off of the University of Coimbra, specialized in software development for mobile operators and digital TV. It also develops software solutions for companies in other business areas that require the convergence of mobile communications and internet. WIT headquarters are in Coimbra, supplemented by three development centers in Porto, Leiria and Lisboa and a branch office in San Jose (California, U.S.) [7].

WIT organization includes three business units: telco, mobile and digital TV. The Telco unit develops software for mobile operators, serving customers in Europe, USA and Africa. This unit is also responsible for the WIT Communications Suite product, a software solution for convergence of mobile communications and Internet.

The mobile unit is dedicated to developing applications for mobile devices using the following platforms: iPhone / iPad, Android, Blackberry, J2ME and Symbian. The customer base ranges go from the banking area (BCP, CGD and BANIF), the media area (Impresa and Controlinveste), up to international clients, such as TomTom and Real Networks.

Finally, the digital TV is responsible for the development of a platform for TV widgets whose targets are cable operators and IPTV. This unit has customers in Europe and United States and has expertise in software development for Microsoft Mediaroom and OpenTV.

2.2 Tasks

The objective of the internship was to develop a business prototype for Android Tablet and to implement share interests on an existing Android application. For the development process we have used an adaptation of Scrum agile software development methodology, an internal policy of the company. This method is flexible, iterative and has an incremental approach to optimize predictability and control risk [8].

The business prototype was implemented for Android Tablet, which connects to an IMS network and uses its telephony and instant messaging through SIP communication protocol. The first task was to study the IMS architecture that is used by most telecom operators, in order to understand its structure. After consolidating the knowledge about IMS, the study was conducted into the state of the art around functionalities and business collaboration applications for IMS and Voice over IP (VoIP) systems in general. For this study we have collected, analyzed and organized the information about existing applications in the market in order to apply it in the development of the business prototype. The next task was to study the standard protocols with the objective of collecting the necessary technical information for the implementation phase and the analysis of project requirements. The final task included the business prototype implementation and its integration within WIT test environment. This task also involved the creation of a demo for further demonstrations and internal benchmarking studies. The whole project was assigned to four team members, and the aforementioned tasks were executed by each member individually. Three of them developed the chat, call and management servers and I was assigned to develop the Android client. Table 1 represents the tasks distribution over time.

<table>
<tr><th>SEPTEMBER</th><th>OCTOBER</th><th>NOVEMBER</th><th>DECEMBER</th><th>JANUARY</th></tr>
<tr><td>State of the art</td><td>Requirements</td><td colspan="3" rowspan="2">Development</td></tr>
<tr><td>Technologies research</td><td>Development</td></tr>
</table>

Table 1 - Business prototype tasks distribution

The work routine during the business prototype development was the following: each day a version of the project was created and sent to an online storage service in order to create a backup and to increment project functionalities. During a period of three weeks, project

meetings were held with two engineers responsible for the project, in order to discuss its development and establish a set of new objectives for the next meeting.

The next internship goal was the implementation of share interests based on the user localization into an Android application developed by WIT Company, called Joyn [3]. The first task involved the study of Sprylogics library [4], which enables the user to search for content based on its location, e.g.: weather, movies, points of interests and YouTube videos. Next, Joyn application implementation was analyzed with the purpose of checking how the new features must be developed. Finally, the development process was divided into the following Use Cases (UC): ability of search for content, the possibility of sharing content with the user contacts and suggesting content based on user's location. Table 2 presents the share interests tasks distribution over time.

<table>
<tr><th>FEBRUARY</th><th>MARCH</th><th>APRIL</th><th>MAY</th></tr>
<tr><td>Sprylogics SDK Study</td><td rowspan="2">Integration Sprylogics SDK</td><td rowspan="2">Development first UC</td><td>Development second UC</td></tr>
<tr><td rowspan="2">Host application Study</td><td rowspan="2">Development third UC</td></tr>
<tr><td>Development first UC</td><td>Development second UC</td></tr>
</table>

Table 2 - Share interests tasks distribution

At this stage, project meetings were held during a period of two weeks with the responsible of the project and the work was done individually.

2.3 Critical Analysis

The business prototype was produced according to the internal requirements of the company, being proposed as its highlight an information contextualization system, called tags. Unfortunately, the business prototype did not advance to the production phase. This was due to the fact that from the WIT Company customers' point of view it is more important to implement a new functionality into an existing solution than developing a prototype. This led to a restructuration of goals and priorities. So, the already developed code was reviewed, in order to check if any part was useful for the existing Android application developed by WIT Company. Although the project was not continued it served to increase the knowledge about

the Android platform and to improve the development skills, useful for the development of the share interests. This second project was developed and presented to Engineer Rui Gil, and by the time this report was being written the solution was in commercialization process with WIT Company clients.

Chapter 3 - Planning, Risks and Methodology

This chapter is related to the planning phase and the methodology used to develop the assigned projects. In terms of planning a Gantt chart [9] was used, which is a type of bar chart used to illustrate a project schedule, including start and finish dates of activities and a summary of activities of a project. As for the methodology, an adaptation of Scrum for each internship goal was used. Scrum is an agile methodology that uses a group of techniques based on iterative and incremental development [11].

3.1 Planning

This section will describe the schedule for each project, namely the business prototype and the share interests. It will also describe each phase associated with the two projects, the corresponding tasks, and the risk analysis for the business collaboration project.

3.1.1 Business Prototype Planning

The development process for the business prototype was divided into four separated phases: Begin, Analysis, Development and Documentation. Each phase has associated tasks and an estimated time to conclude them.

- **Begin (2 days):** This phase refers to the initial phase of the project where the initial product was described and objectives were established;
- **Analysis (18 days):** This phase was intended for research, requirements definition and user interface mockups creation. The research was used to find and analyze similar business collaboration applications, with the purpose of using their strengths as a starting point, and to find technologies that can be used to help the business prototype implementation. The requirements definition was used to specify the functionalities, the

application core and to create the mockups that were used to discuss the business prototype UI;

- **Development (94 days):** During this phase we have used the state of the art as a starting point to implement the initial group of features and all the application design. It was also in this phase that the UI was developed and the performance mechanism was implemented;
- **Documentation (117 days):** The purpose of this phase was to produce documents that will report all the work related to the creation of the business prototype.

In order to better understand each phase individually, in terms of planned time, Figure 1 presents a Gantt chart with a general overview of the four phases.

Figure 1 - General Gantt chart for the business prototype planning

From Figure 1 we can see that the longest phase (critical path) is the one involving the report writing. This means that any delay on that activity will directly affect the planned project completion date [15]. The complete Gantt chart with all the tasks in the business prototype planning can be seen in Appendix A - Business Prototype Planning.

Table 3 shows the tasks associated with the four phases of the business prototype development. The estimated time, in days, to complete each task is also shown.

PHASE	TASK	ESTIMATION
Begin (2 days)	Initial phase	2 days
Analysis (18 days)	State of the art	5 days
	Requirements analysis	4 days
	Preparation of development plan	1 day
	Study of web technologies	5 days
	Mockups creation	3 days
Development (94 days)	Implement cache system (database)	3 day
	Implement base project	35 days

PHASE	TASK	ESTIMATION
	Implement authentication	1 days
	Implement sign up	2 days
	Implement user profile	3 days
	Implement contacts	5 days
	Implement discussions	6 days
	Implement send files	3 days
	Implement share location	2 days
	Implement view files content	2 day
	Implement tags	8 days
	Implement drag-and-drop	4 day
	Implement search mechanism	2 days
	Implement application settings	2 days
	Implement to-do lists	3 days
	Implement UI for video and voice calls	4 days
	Implement background service	5 days
	Functional tests	4 days
Documentation (117 days)	Initial template	10 days
	Introduction and internship description	16 days
	State of the arte	25 days
	Planning and methodology	17 days
	Development chapter	30 days
	Conclusions and appendixes	19 days

Table 3 - Business prototype tasks lists

3.1.2 Share Interests Planning

In order to implement share interests based on the user location the planning was divided into three separated phases: Study, Implementation and Documentation. Each phase has associated tasks and an estimated time to conclude them.

- **Study (20 days):** This phase was intended for study the Sprylogics library and the technologies associated with it (JSON and server communication). This phase also involves the understanding of the structure of WIT application, Joyn;
- **Implementation (50 days):** Using Sprylogics library, this phase refers to the implementation of share interests based on user's location;
- **Documentation (75 days):** The purpose of this phase was to produce documents that will report all the work concerning the implementation of the new functionalities.

Figure 2 presents a Gantt chart that gives a general overview of each phase over time.

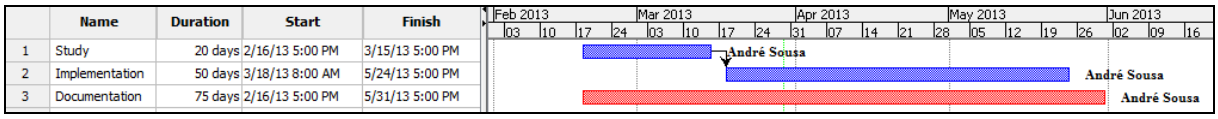


Figure 2 - General Gantt Chart for the feature implementation planning

From Figure 2 we can see that the critical path is the one involving the report writing, just like what happened in the business prototype plan. The complete Gantt chart is in Appendix B - Share Interests Planning.

Table 4 shows the tasks associated with the four phases of the business prototype development. Each task as an estimated complete time, measured in days.

PHASE	TASK	ESTIMATION
Study (20 days)	Sprylogics library analysis	4 days
	JSON analysis	2 days
Implementation (50 days)	Integration of Sprylogics library	2 day
	Show interests based on user is location	18 days
	Implement share interests functionalities	10 days
	Implement search content functionalities	12 days
	Functional tests	8 days
Documentation (75 days)	Technologies	13 days
	Planning	15 days
	Development chapter	29 days
	Conclusions and appendixes	18 days

Table 4 - Messages storage and persistence tasks lists

3.2 Risks Management

"Project risk management is the art and science of identifying, analyzing, and responding to risk throughout the life of a project and in the best interests of meeting project objectives. A frequently overlooked aspect of project management, risk management can often result in significant improvement in the ultimate success of projects." [17]. The absence of identification and management of potential problems could undermine an entire project leading to its possible failure. Our analysis was performed by request from WIT Company. Only the business prototype was analyzed. We will begin with the identification of the moments where the evaluation of risks was taken and, then, the identification and ranking of the project risks.

3.2.1 Risks Assessments

In order to assess the risks associated with the project, we have performed several risk assessments and checked their impact on the project schedule. Risks assessments were performed in the following stages:

1. Delivery of the analysis;
2. Development;
3. Report document.

3.2.2 Risks Identification

Table 5 provides a list of the risks that were found during the risk identification phase. Its identification was made only by the Android tablet application team as a proposal for WIT Company. Each line represents a risk and is associated with an identification number, a risk event, a rank and the respective impact in the project. The risk event refers to a specific event that may occur to the detriment or enhancement of the project. The rank is a numeric value that is established and is related with the impact in the project [16]. The numeric values are as follow: 1 is considered high impact risk, 2 and 3 is a medium impact and 4 is a low impact.

NUMBER	RISK EVENT	RANK	IMPACT
1	Failure to identify and correct errors	1	High
2	Fail to set project priorities	1	High
3	Constant requirements changes	1	High
4	Low productivity	1	High
5	Unexpected changes in the project scope	1	High
6	Failure to identify complex functionalities	1	High
7	Inadequate or non-existent project planning	1	High
8	Insufficient / inadequate resources	2	Medium
9	Absence of leadership	3	Medium
10	Complex product implementation	3	Medium
11	Failed communication between the team member	3	Medium
12	Failure in the requirements definition	3	Medium
13	Failure in the specification definition	3	Medium
14	Developing the wrong user interface	3	Medium
15	Lack of training	4	Low
16	Disease of a member of the team	4	Low
17	Absence of an effective development methodology	4	Low

Table 5 - Business prototype risks identification

3.2.3 Preventive and Reactive Measures

In order to reduce the effects of some of the risks that we have found, we made a list of preventive and reactive measures. These measures were defined only for the risks that have the biggest impact in the project (medium and high) and they are applied only to this particular project. Although the measures were identified by the Android team, their implementation can only be decided by WIT Company. Table 6 presents the preventive and reactive measures for the top ten risks. The risks that were chosen from the team's view point, had the highest probability of happening.

NUMBER	RISK EVENT	PREVENTATIVE MEASURES	REACTIVE MEASURES
1	Failure to identify and correct errors	- List of companies specialized in this subject (outsourcing); - Contact experience testers.	- Hire a specialized company.
2	Fail to setting project priorities	- Establish priorities and review the same along the project; - Adjust tasks; - Design an alternative project planning.	- Meetings to reinforce the project objectives.
3	Constant requirements changes	- Work more close to the client; - Create a document to prevent continuous changings.	- Adapt task in the project planning.
4	Low productivity	- Proper training; - Aside funds for bonus; - Improve the work space.	- Motivation seminars; - Bonus rewards.
5	Unexpected changes in the project scope	- Try to work closer with the client of the project.	- Restructure the entire project planning; - Rethinking the project objectives.
6	Failure to identify complex functionalities	- Work with a skilled resource.	- Restructure project priorities.
7	Inadequate or non-existent project planning	- Work closer with the project leaders.	- Rethinking the project priorities.
8	Insufficient / inadequate resources	- One or two back up teams; - Aside funds for emergencies.	- Restructure the project planning; - Buy new equipment.
12	Failure in the requirements definition	- Work with the project leaders; - Let the client of the project sign the requirements definition.	- Restructure project planning; - Establish new priorities for the project.

NUMBER	RISK EVENT	PREVENTATIVE MEASURES	REACTIVE MEASURES
14	Developing the wrong user interface	- Work with mockups for visual understanding; - Define with client of the project the interface.	- Restructure the entire project design; - Adapt the project planning.

Table 6 - Preventive and reactive measures for business prototype risks

3.3 Methodology

During the development phase, we have used an adaptation of Scrum, an agile methodology to create the business prototype and share interests. Jim Highsmith says that being Agile means being able to *"Deliver quickly. Change quickly. Change often."* [10] It consists in a group of techniques based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change [11]. There are some alternative agile methods, such as: Extreme Programming (XP) [12], Scrum [8] and Dynamic Systems Development Method (DSDM) [13].

For the development process, we have adopted an agile methodology based on some concepts of Scrum, as this is the methodology adopted by WIT Company. The used methodology does not follow a rigid specification of Scrum, but adopts the basic concepts that define it. This allows a better adaptation to the company projects and a more flexible internship. In the next section we will give a description of what Scrum consists, the terms used in it and its usage within WIT Company projects.

3.3.1 Scrum

Scrum is an agile software development methodology. This method is flexible, iterative and has an incremental approach to optimize predictability and control risk. In Scrum there are some specific terms: product backlog, sprint, sprint backlog and daily Scrum [8].

The **Product Backlog** is owned by the product owner, responsible for its availability, content and ordering. It is an ordered list of everything that could be needed in the product and is the single source of requirements [8].

Scrum is based on **Sprints** which is a basic unit of development, with a fixed-length periods of time of one to four weeks (usually two) during which a list of things is performed or certain

functionalities are delivered. The set of features that go into a sprint come from the product backlog. Which backlog items go into the sprint (Sprint Backlog) is determined during the sprint planning meeting. At the end of each Sprint, a sprint review meeting is held. During this meeting the Scrum team shows what has been accomplished during the sprint and adapts the product backlog if needed. Based on that and any changes to the product backlog, attendees collaborate to decide what to do next [8].

The **Sprint Backlog** is the group of product backlog items selected for the sprint and a group of measures for delivering new increments to the product and perform the sprint goal (what the product owner and the team define as the top target to that sprint). The sprint backlog is a forecast by the development team about what features will be in the next increment and the work needed to deliver them [8].

The **Daily Scrum** is an event that takes around 15 minutes in which the development team synchronizes the activities and creates a plan for the next day. This is done by analyzing the work since the last Daily Scrum and planning the work that could be done before the next one. The Daily Scrum is made at the same time and place each day in order to reduce complexity [8]. During the meeting, each development team member explains:

- What has been accomplished since the last meeting?
- What will be done before the next meeting?
- What obstacles are in the way?

The Development Team uses the Daily Scrum to assess progress toward the sprint goal and to assess how progress is trending toward completing the work in the sprint backlog. The daily Scrum optimizes the probability that the development team will meet the sprint goal. The development team often meets immediately after the daily Scrum to re-plan the rest of the sprint's work [8].

Previous concepts are represented by the diagram on Figure 3 [18].

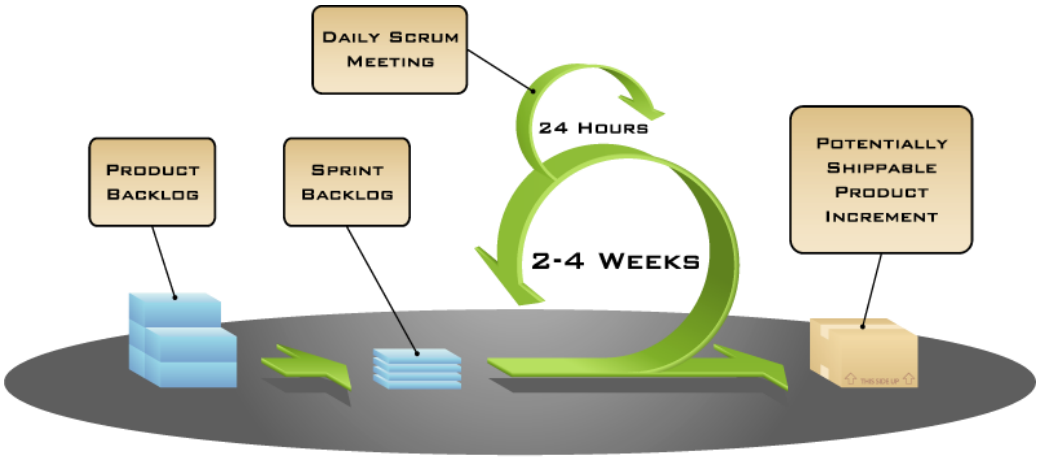


Figure 3 - The Scrum diagram

On Scrum there are some core roles, which are committed to the project in the process and that are responsible for developing the product and representing the scrum team. The roles are:

- **The Project Owner**: Represents the stakeholders and is the voice of the customer or someone who has his vision of him. The product owner writes the user stories, prioritizes them, and adds them to the product backlog. He is accountable for ensuring that the team delivers value to the business [14];
- **The Scrum Master**: He is responsible for making sure that the team lives by the values and practices of Scrum and for removing impediments to the ability to achieve the sprint goal. The scrum master is not the team leader, but acts as a buffer between the team and any distracting influences, protecting the development team and keeping it focused on the tasks at hand [14];
- **Development Team**: It is responsible for delivering product increments at the end of each sprint. A development team is self-organizing, and it is composed of three to nine people who do the actual work (analyses, development, testing, documentation, etc) [14].

3.3.2 Scrum on WIT

WIT Company adopted the Scrum methodology adjusting it to each project specific needs. In our case, the role of project owner was taken by the Engineer Paulo Sousa, the role of scrum master by the Engineer Rui Macedo and the development team consisted in one element. For the share interests project the role of project owner was taken by the Engineer Rui Gil. The sprints were established during a period of three weeks, where the two engineers to talked

about the project status and established a new group of goals to the next sprint. During all the development time, the project was supervised by the scrum master, who has helped in some tasks and gave expert guidance. Although we didn't follow a rigid specification of Scrum, a version of the project was created every day and sent to an online storage service in order to create a backup and to increment project functionalities.

3.3.3 Quality Policies

WIT Company has a high quality policy in order to create the best possible solutions. One of the policies that are used is the ISO 9001, which is a group of technical standards that establish a model of quality management for organizations, whatever their size or type [19]. Also there is the internal policy Z2M2 - where Z stands for zero bugs and zero delays, and M stands for maximum profitability and maximum satisfaction for the client and the enterprise. In our projects, only the Z2M2 policy was applied, in order to achieve the best possible quality.

Chapter 4 - State Of The Art

In this chapter we will describe the research involving the business prototype and the share interests functionalities. We will start by presenting the existing applications in the business collaboration area, their highlights and a comparative analysis. Next, we will focus on Android usability guidelines, by examining some successful Android applications and Google's guidelines. Finally, a list of technologies that will be used during the implementation of each project will be presented, as well as a description of WIT's application, Joyn.

4.1 Existing Business Collaboration Applications

In the business environment there is a great need for a tool that helps everyone involved in a project to easily talk and discuss ideas from a mobile device. With this in mind, several companies have developed some tools to solve this necessity. We will analyze seven of these solutions: Adobe Connect, Cisco WebEx Meetings, Fuze, iMeet, Blue Jeans, GoToMeeting and HipChat. The next topics will describe each of these solutions, their strengths and weaknesses.

4.1.1 Adobe Connect

The Adobe Connect application is used as a web conferencing solution for online meetings. It allows users to share files or device screens, chat, broadcast live audio and video, and participate in interactive online activities through a conferencing interface with several users. The solution has support for web browser, desktop (Windows and MacOS) and mobile - iOS, Android, and Blackberry Playbook [20].

Users talk to each other through meeting rooms, which are a live online conference for multiple users. The meeting room is an online application that the user uses to conduct a meeting, which consists of various display panels (pods) and components, with each pod

performing a specific role (i.e. chat, whiteboard, notes, etc.). Adobe Connect have a high level of customization, allowing the rearrangement of the look and feel of meeting rooms.
Adobe Connect features are [21]:

- Unlimited and customizable meeting rooms;
- Multiple meeting rooms per user;
- Audio and video conferencing;
- Chat;
- Meeting recording;
- Screen sharing;
- Notes, chat, and whiteboards;
- File sharing;
- User management, administration, and reporting;
- Collaboration builder Software Development Kit (SDK).

4.1.2 Cisco WebEx Meetings

WebEx is the oldest and best-known web-conferencing solution in the market. It helps companies to have meetings in a more easy and efficient way. It combines file and presentation sharing with voice and High Definition (HD) video, collaborative documents and videoconferencing. It supports mobile platforms - iOS, Android and Blackberry - and has a free version with some limitations, and paid plans to use all solution capabilities [22].
Different from other applications, the user does not need to install any kind of add-on to use it. One only needs to have java installed and cookies enabled. WebEx was created with the simplest UI, where the administrator can start a meeting by simply clicking on the option "One-Click Meeting". Once the meeting is initiated the user can either forward the e-mail invitation to participants or invite participants directly from the room.
WebEx features are [23]:

- Remote control from a user present in the meeting;
- Record meeting;
- Collaboration on shared documents;
- Sending files;
- Chat;
- Voice and video conference.

4.1.3 Fuze

Fuze is also a conferencing business service, designed for those who use social networking. It can easily import contacts and Instant Messaging (IM) from a large range of applications such as Facebook, Twitter, Linked-In, Google Talk, Windows Live, AOL, Yahoo and Skype. It is the youngest service on the market. It is entirely browser based, so the interface is always up-to-date and the users do not need to download anything. It also has native applications for the most popular platforms: iOS, Mac, Android and PC. Fuze supports multiple monitors, up to 12 simultaneous camera feeds and allows users to host meetings from their iPad [24].

To start using Fuze on a Mac or PC, the users will have to install a simple application. The process of setting up and joining meetings is very simple. To set it up, the user only needs to click an icon and invite his guests. To join a meeting the users just need to enter a unique Uniform Resource Locator (URL) that the guests receive from their invitations. The user can also call "Fetch" which is a tool that will automatically dial people and invite them to the audio portion of a meeting [25].

Fuze features are:

- Integration with third-party applications;
- Audio and video conference;
- Record meeting;
- Group chat;
- File sharing;
- Real time video stream;
- Share desktop.

4.1.4 iMeet

iMeet has a different concept of meeting: small businesses can rent a custom URL to host audio and videoconference calls. Using a cubed based interface, iMeet is available for iOS platforms and it can be integrated with other applications like Outlook and Evernote. It is very simple to configure and only needs a few information to create a new account (name, email address, and primary phone number). Then, the user receives a custom URL (e.g. https://imeet.com/example) and call-in numbers [26].

iMeet features are:

- Group chat;

- HD Video;
- Auto-connect (feature that lets the user instantly join a conversation. iMeet can be configured to call the user as soon as he enters the meeting room);
- File and screen sharing;
- Integration with third-party applications.

4.1.5 Blue Jeans

Blue Jeans is an online web conferencing tool created by Blue Jeans Network. The solution is compatible with mobile and a wide range of other devices and software, like Skype, Google Voice, Outlook, Google Calendar, Linked-in and others. Blue Jeans is a business grade conferencing solution with multi-point video that lets the user manage who is heard and seen, supporting up to 25 endpoints per call.

The innovation key of Blue Jeans is the ability for browser-based access to meetings, regardless of where they were initiated. Users can join conference calls via Skype, their web browser, classic videoconference rooms, Google Video Chat, or Microsoft Lync. Blue Jeans also has a license that allows it to join Skype users to non-Skype users [27].

4.1.6 GoToMeeting

This solution is a business collaboration software to arrange meetings using the web infrastructure. Starting a meeting is a one-click prospect, invitations can be distributed on the fly, and users can audio conference via VoIP or a toll-free call-in number. It supports almost every platform (iOS, Mac, PC and Android) and is focused in the UX, especially for small businesses seeking a straightforward web conferencing solution from a trusted name [28].

If a user wants to use GoToMeeting on a PC or MAC they need to install an application. Once the application is installed, it is easy to create or join a meeting. The user can either initiate from the dedicated application or begin from the website.

GoToMeeting features are [29]:

- Work face to face with high-definition video conferencing;
- Conference calls and videos;
- Share screen or just a specific application;
- Record meeting sessions;

- Share keyboard and mouse control to cooperatively edit files on screen.
- Instantly change presenters to see each other's work;
- Draw and highlight on screen;
- Group chat.

4.1.7 HipChat

In comparison with other solutions, HipChat presents a simpler UI. It has a web version, a desktop version supporting Mac, Windows and Linux, and a mobile version supporting the most widely used mobile operating systems: iOS and Android. It gives the possibility of free usage, but only for a total of four users. The great advantage of HipChat is that it allows users to chat over Short Message Service (SMS), for one-on-one conversation [30].
HipChat features are [31]:

- Group and one-to-one chat;
- Conference calls and videos;
- File sharing;
- Images, YouTube videos and twitter preview;
- Spell checking;
- Chat and file history.

4.1.8 Comparison between Solutions

Table 7 aims to give a better understanding of the features present in each of the presented solutions, highlighting only the features that are relevant for this project. Since the scope of the project is business collaboration, the features were chosen having this in mind. The most important is an easy way to set up meetings and to express ideas, and this involves features like chat, video and audio conferencing, file sharing and a history log. Other features like using the device's camera to stream video to the meeting and third-party application (e.g. Skype, Facebook, and twitter) integration are also desirable.

Functionality	Adobe Connect	Cisco WebEx	Fuze	iMeet	Blue Jeans	GoToMeeting	HipChat
Video Conference	☑	☑	☑	☑	☑	☑	☑
Conference Call	☑	☑	☑	☑	☑	☑	☑
Group Chat	☑	☑	☑	☑		☑	☑
Record Meeting	☑	☑	☑			☑	
Chat and File History	☑	☑		☑		☑	☑
Stream Video			☑				
Third-party applications			☑	☑	☑		☑
Files Sharing	☑	☑	☑	☑			☑
Collaborative Documents	☑	☑	☑			☑	
Screen Sharing	☑	☑	☑	☑	☑	☑	
Real Time Stream	☑	☑	☑				
Remote Control		☑				☑	
Notes Tool	☑						
Join The Meeting From Other Applications					☑		
Chat Over SMS							☑

Table 7 - Comparison between business collaboration applications

Among the features highlighted in Table 7, there are a few that are not present in all the applications. These differences are what make each application unique. For example the ability to chat over SMS, join a meeting from other applications, remote control and video stream. Furthermore there is a group of functionalities transversal to all the solutions (e.g. video and audio conference and chat), suggesting that these are mandatory features for a business collaboration application. This comparative study was focused primarily on popular tools in this segment, for the business world. Although some tools have a different scope, being focused on micro-blogging or real-time communication, all have features that are to be included in the tool to develop. This analysis allowed us to create an initial group of requirements and to better understand some of the key features (contact lists, chat and audio/video calls) that are to be included in the business prototype, in order to create a distinctive and desirable application.

4.2 Android Usability

Usability is a very important part of any application, because the UX will determine a huge part of its success. Therefore, in order to captivate the customer, it must be well designed and implemented. For business collaboration applications, it is easy to create a UI with a lot of functionalities, but very confusing from the user's perspective. Because of this common problem, Google released a group of guidelines to help developers to create Android applications with a clean, consistent and optimized interface. These guidelines define the size, location and design that the visual elements should have [32]. In this section we will present some successful Android applications that follow these guidelines as well as their UI for contact lists, chat and audio/video calls. This information will be used to help the design of our business prototype.

4.2.1 Instant Messaging Solutions

In this topic we will present some well know instant messaging Android applications, along with their user interface.

- **Hangouts** is a meeting software initially developed for Google+ social network, having a web and a mobile version. This application includes some unique user interface features that can be replicated in our solution, such as the highlight of the window of the user who is talking, change the window size if the user is calling from a smartphone or a device with a larger screen, and others [33].
- **Viber, WhatsApp** and **textPlus** are very similar applications that give to the end-user free Voice Calls, Video Calls, and Text Messaging for multiple mobile platforms. These are relevant applications because their UI use Google's guidelines [34].
- **MightyText** was created from ex-employees of Google, Maneesh Arora and Amit Sangani. They decided to develop an iMessage application for Android devices. It allows users to send SMS and Multimedia Messaging Service (MMS) messages from the desktop or tablet device, using client/server architecture. The relevance of this project is the feature of sending text messages and the disposition of the chat list [35].
- **Skype** is one of the most popular voice communication services in the world and is used not only for private but also for business-related communication. Skype is one of the pioneers in voice communication over the Internet, and many other applications and services have followed suit [36].

4.2.2 User Interface

Taking into account the scope of the project, this topic is divided into four categories: contacts list, chat list, chat messages and audio and video conference. Each category will describe patterns that are used when developing a UI for a business collaboration application. These patterns will help the creation of the business prototype's appearance, helping making it simpler and easier to use.

4.2.2.1 Contact List

The contact list is usually presented with a list of users and some information associated with them - picture, name, preview of the last message sent or the status (whether they are available or not). The position of the contact picture (if present) is commonly added on the left size of the screen, followed by its name on the right. The name is usually composed of the forename and surname of the contact to make it easier to distinguish. In this category we will give the example of three applications. Figure 4 presents the design of the contact lists from the following applications: HipChat, Viber and WhatsAppp. This will allow the observation of the pattern of their design.

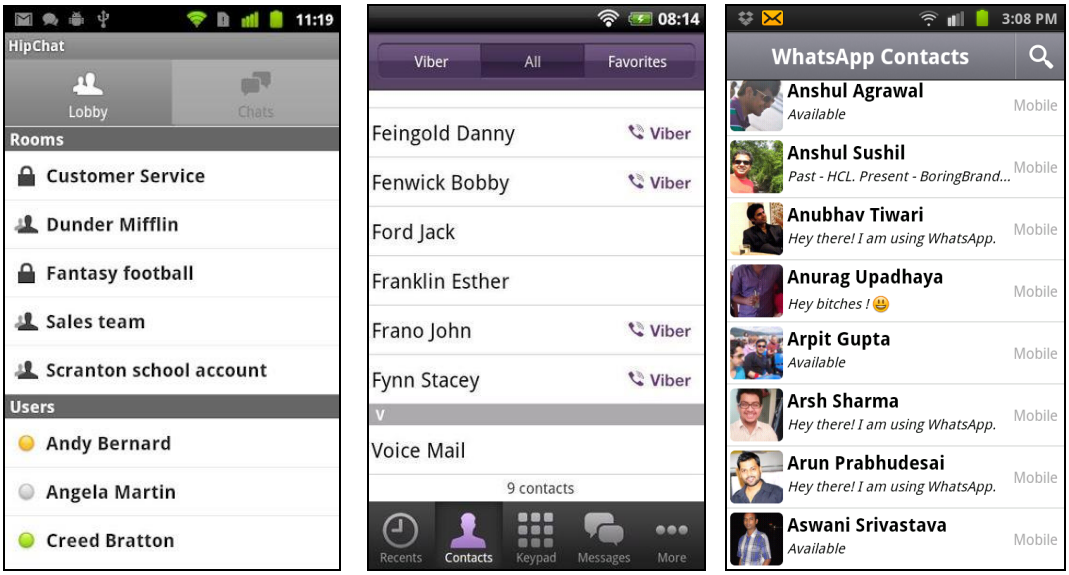


Figure 4 - Contact list example for Android (HipChat, Viber and WhatsApp)

As we can see, the big difference in this category is the amount of information that is shown. Some applications (HipChat and Viber) preferred to keep it very simple, presenting only the name of the contact. Other applications (WhatsApp) try to show more useful information, like

the user's picture and its availability. We have adapted the combination between name and contact picture, as in our opinion it seems to be a better approach, because with these two elements the user can easily distinguish contacts.

4.2.2.2 Chat List

When we talk about instant messaging applications, some features are very common, like the ability to chat only to one contact or to a group of contacts. The design of a chat list follows up a very common pattern as well. Each chat has a group of information associated to it - the contact picture, name, preview of the last message sent and the status (whether they are available or not). In this category we will give the example of two applications: Viber and textPlus. The Figure 5 shows their design.

Figure 5 - Chat list example for Android (Viber and textPlus)

In Figure 5 we can see some interesting visual elements. Each solution uses visual elements to distinguish between one-to-one or a group chat. One of the options is by presenting all the pictures of the involved contacts. Another way is to show the number of contacts that are on the chat. In addition, a group chat could have a name or only present the names of the participants separated by commas.

4.2.2.3 Chat Messages

The chat messages can be presented to the user either in bubbles or simply in a continuum text separated by a divisor. Figure 6 and Figure 7 will show some applications that have an interesting design to show chat messages and to make it easy for the user to know who has sent each message: Viber, HipChat and textPlus.

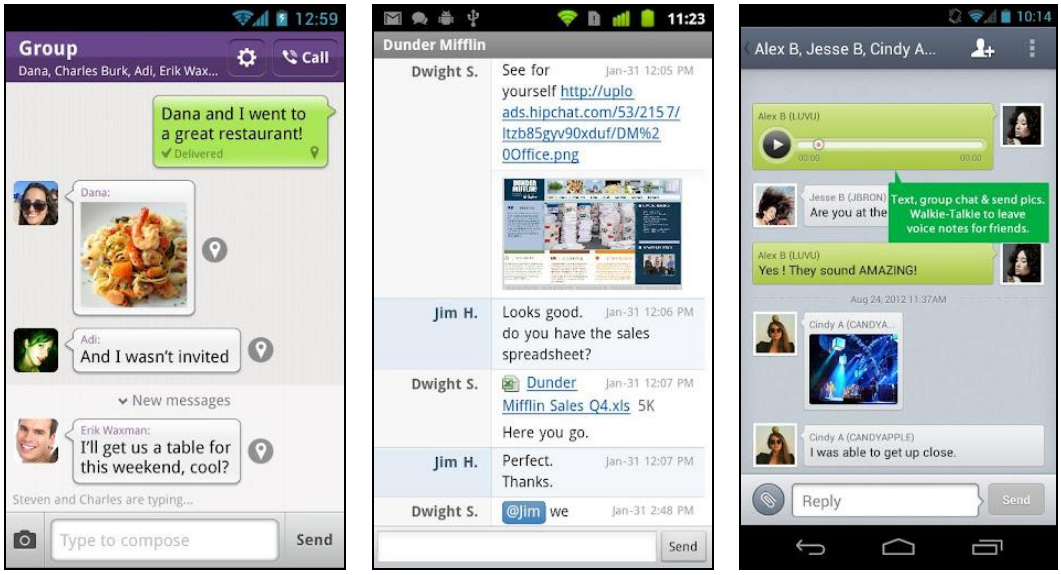


Figure 6 - Chat examples for Android (Viber, HipChat and textPlus)

In our opinion, by observing Figure 6 we can understand that bubbles chat solution (like Viber and textPlus) makes the reading of messages easier and more pleasant. It also helps the user to visually distinguish if the message was sent from him or was received from one of his contacts. Since we are working for an Android tablet, besides showing only the contact list, we might also have enough space to show the chat list associate with a contact or a group of contacts. This design can be seen in textPlus, as shown in Figure 7:

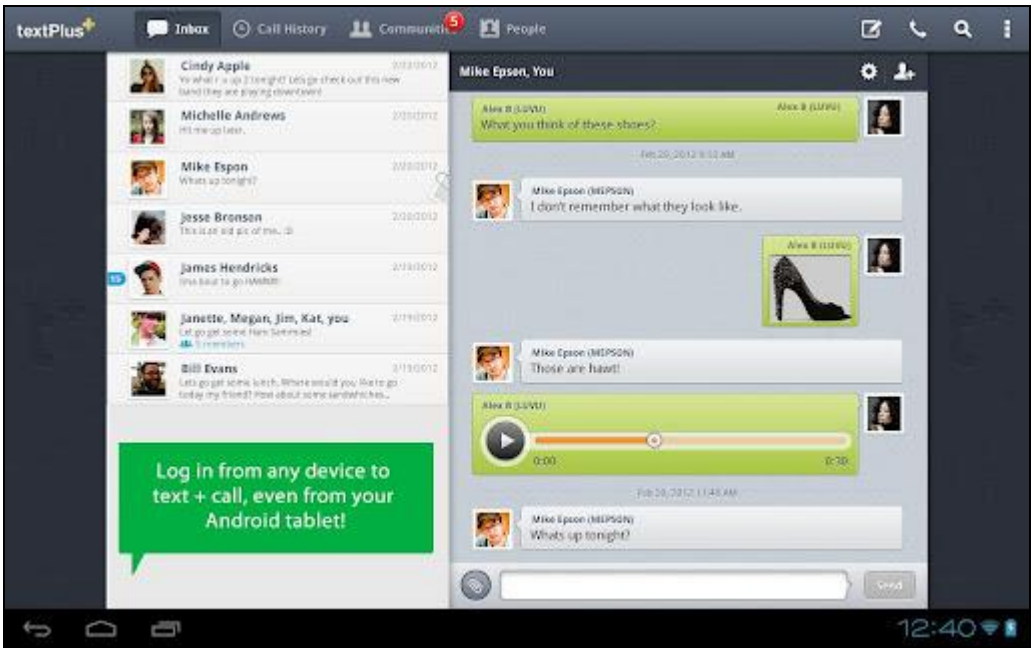


Figure 7 - Textplus for Android tablet

As we can see in Figure 7, we can take benefit of the large screen that an Android tablet provides and use this characteristic to show more information to the user.

4.2.2.4 Audio and Video Conference

Having a meeting through audio or videoconference makes it easier to communicate and express ideas. Therefore it is important to create a good user interface, in order to show the important information and the controls associated to the call.

In Figure 8, Figure 9 and Figure 10 we show some examples of what was made by other applications (Viber, Skype, textPlus, Adobe Connect and Cisco WebEx) in terms of audio and video calls.

Figure 8 - Voice calls in Android from left to the right: Viber, skype and textPlus

We can take some ideas from Figure 8. It is important that all the controls are intuitive and placed in the right place. Normally the "end call" button is placed in the bottom right corner or taking all the space on the bottom. It is also highlighted by a different color that distinguishes it from the other controls. It is also important to show some of the contact's information, name and picture, so that the user may easily know who is talking. Also, the call time is an important part that must be present in every call. The last two solutions' interface are focused on video call. They have some interface facilities that can inspire us for the conception of our prototype's user interface.

Figure 9 – Example of videos calls for Android (Fuze and Cisco WebEx)

Figure 9 shows that when we are on a video call it is useful to present it in full screen and show only the important information, controls and video streams of the contacts involved in the call.

It is imperative that the person who is talking, be clearly distinguishable from the others. This could be achieved by adding the user stream to the background of the interface or by adding to his stream a different border. Figure 10 shows these two points of view.

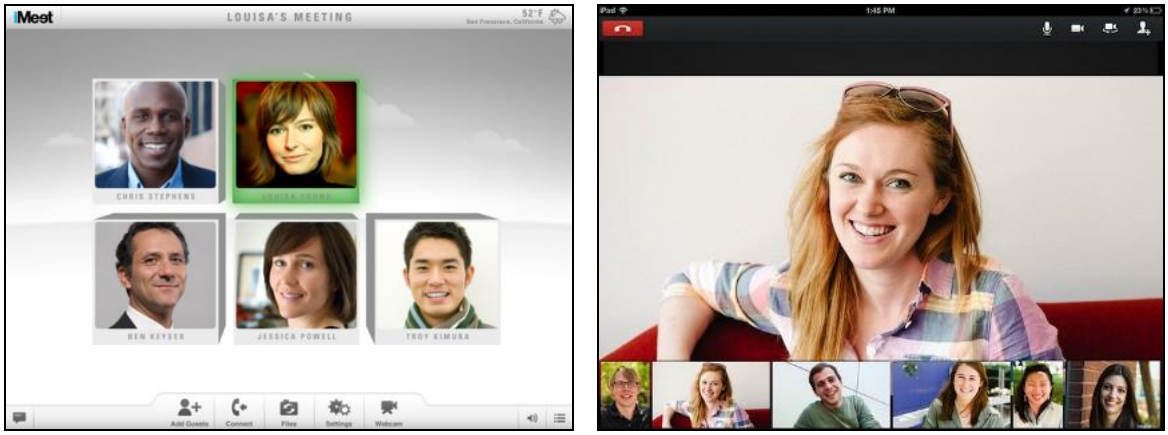


Figure 10 - iMeet and Hangouts for Android tablet

Besides adding the talking person stream to the background, Hangouts also have two other interesting features: it changes the stream screen size to give the user visual feedback from what type of device the contacts are using and highlight the screen of the person who is talking.

4.3 Technologies

In this section we will explain some network architecture concepts and protocols required for the implementation of the business prototype most important features, chat and audio/video calls. It is also presented the technologies used to implement the share interests and some libraries that are used for the implementation of SIP protocol and the advantages of each one. As described in the introduction chapter the business prototype was not continued in the second half of development, so the choice of a SIP stack library for the project was not carried out. However this study was requested by WIT in order to know the available options and this is the reason why it is presented in this report.

4.3.1 Session Initiation Protocol

SIP is an application-level signaling protocol for the creation and management of sessions over an IP network. It has been designated by the Internet Engineering Task Force (IETF) as an internet protocol and it is currently defined in Request for Comments (RFC) 3261. It is widely

used for controlling communication sessions such as voice and video calls over IP, and it is the base of IMS architecture [37].

Like other protocols, SIP works in the Application layer of the Open Systems Interconnection (OSI) communications model, responsible for ensuring that communication is possible. SIP can establish, modify, and shutdown communication sessions. The protocol can also invite participants to unicast or multicast sessions that do not necessarily involve the initiator. Because the SIP supports name mapping and redirection services, it makes it possible for users to initiate and receive communications and services from any location, and for networks to identify the users wherever they are [37].

SIP is a request-response protocol, dealing with requests from clients and responses from servers. Requests can be sent through any transport protocol, such as User Datagram Protocol (UDP) or Transmission Control Protocol (TCP). SIP determines the end system to be used for the session, the communication media and media parameters, and the called party's desire to engage in the communication. Once these are assured, SIP establishes call parameters at either end of the communication, and handles call transfer and termination. The parameters that describe the multimedia sessions are negotiated by a protocol named session description protocol (SDP) [37].

4.3.1.1 SIP Agents

SIP defines a group of participants that play a specific role in the communication session. The first one are the User Agents (UA), which are applications installed on SIP endpoints, such as IP phones, mobile phones, wireless devices or a laptop or desktop PC, that interface between the user and the SIP network. A UA can act either as client or a server. When sending SIP requests, the UA acts as a User Agent Client (UAC), and when servicing a request, it acts as a User Agent Server (UAS). A Back-to-Back User Agent (B2BUA) is an application that acts as an intermediary between two parties, but appears as an endpoint to both parties — like a middleman. It serves as both UAS and UAC simultaneously to process session requests [39].

4.3.1.2 SIP Communication

A SIP communication is characterized by a group of messages, requests and responses that are exchanged between the UA. Figure 11 represents a SIP communication, where the vertical bars represents UA and the horizontal arrows the direction of the messages [38].

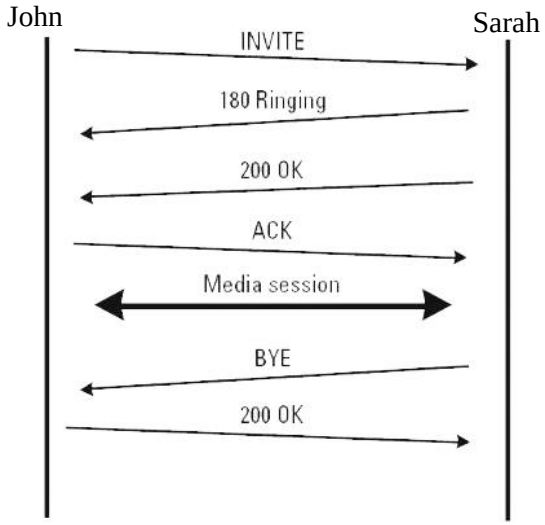


Figure 11 - SIP communication diagram

The next topics explain the diagram from Figure 11 [39]:

1. John, UAC, initiates a new session by generating an INVITE request, which is sent to Sarah. The INVITE message contains SDP parameters that define the types of media he is capable of accepting and where he wants the media to be sent.
2. Sarah sends a signaling, which is used to respond to INVITE. This message indicates that Sarah has accepted the INVITE. Sarah may hear a ring, see a text message, or see a blinking LED.
3. Sarah's acceptance is sent back to John. The body of Sarah acceptance includes SDP parameters defining the selected media chosen from what John had originally offered and where Sarah wants the media to be sent.
4. John's UA responds to the acceptance with an ACK (acknowledgement) directly to Sarah's UA, which tells Sarah's UA that John is ready to start the call.
5. At the end of the conversation, Sarah hangs up her phone. Her UAC sends a BYE message directly to John's UA.
6. John's UAC responds with a 200-OK message directly to Sarah's UA, which ends the session.

4.3.2 IP Multimedia Subsystem

The exponential proliferation and acceptance of the Internet, and at its core, Internet Protocol (IP) technology, has had a profound impact on the way people communicate. Its openness has led to a new era of innovative new applications. With this constant innovation the old architectures do not satisfy the needs of mobile operators' new customers. So, an architecture and framework that enables the convergence of voice, video, data and mobile network technology over an IP-based infrastructure, the IMS, was created [40].

The IMS is a standardized IP-based architecture that aims to fill the gap between two of the most successful communication paradigms: cellular and internet technology. It combines voice, text, pictures, and video in seamless call sessions, offering significant ease-of-use to subscribers and allowing service providers to drive branding through a common interface, while substantially reducing operating costs. It was originally designed for third-generation mobile phones, but it has already been extended to Wi-Fi networks. It was created around the SIP in order to connect traditional telephony services and non-telephony services, such as instant messaging, push-to-talk, video streaming, and multimedia messaging [41].

The IMS architecture is commonly presented has a three layer model structure. The lower layer referred to as the **Transport Layer**, is made up of the physical resources necessary for a connection to be established and for the payload (either voice or data) to be carried from the origination point to the termination point. The second layer, **Session and Control Layer**, provides elementary media functions to the higher-level applications and it is referred to as the control, which comprises "intelligent" elements that determine whether a user is allowed to use the network and how to route a call or data path (also referred to as setting up a session). Finally the top layer is referred to as the **Applications and Services Layer** and it uses a higher level of application services and Application Programming Interface (API) which contains all the application logic and data necessary to offer services to an end user. Figure 12 describes the common design of the IMS architecture [41]:

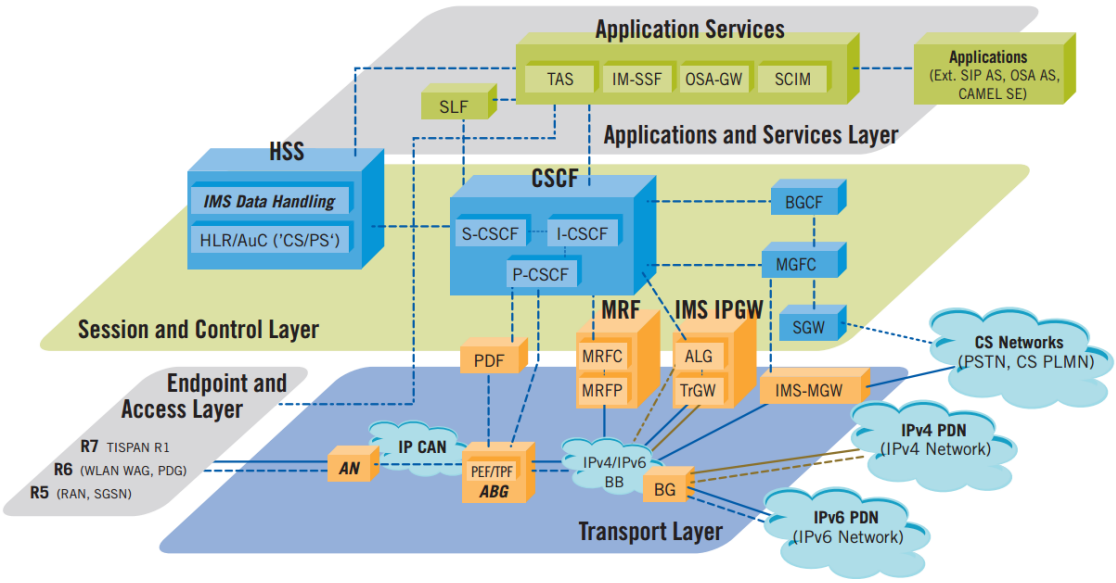


Figure 12 - The IP Multimedia Subsystem architecture

Although IMS cannot be realistically deployed by a network operator who expects to make a return on their investment without all three layers working together as a unit, the Applications and Services Layer is responsible for generating revenue and end-user services that will determine whether the deployment of an IMS network is successful or not. The aim of this project is to work on this layer and to consume a group of services provided by a server on the network and use them to give business collaboration features to the customers.

4.3.3 SIP Stacks

One of the most important problems in Android OS is its fragmentation. There are a lot of versions of the operating system and, as a result, a lot of devices do not have the latest release. Figure 13 shows the distribution of Android versions across all devices (8 August 2013) [42].

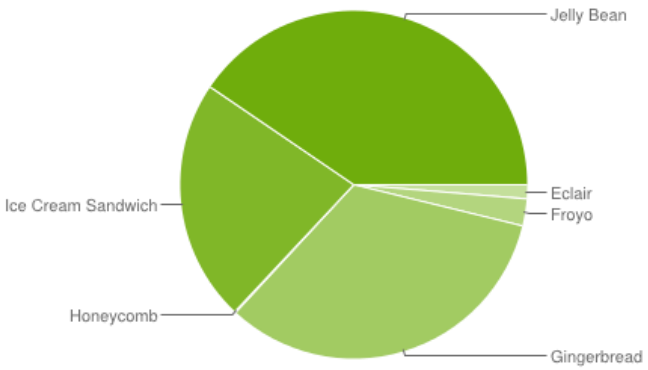


Figure 13 - Distribution of Android platform versions

Since we want to implement the application on an Android tablet we need to limit the installation to Android version 3.0 (honeycomb) or later. This covers about 63% of the Android devices. In Android 2.3, Google introduced a native SIP stack to be used by the applications, but with several limitations, such as: it only works over Wi-Fi; the author user name is not configurable, which is required for registration; proxy authentication is not supported, which is required for calls to the Public Switched Telephone Network (PSTN); SIP Replaces header is not supported, which is required for attended transfers and invites without SDP are not supported (which is required for music on hold) [43].
To get around these limitations, some companies have developed external libraries and others have developed their own open-source SIP stacks and SIP clients. Some of these open-source projects are:

- De-tiny-sip, which use the Jain-SIP for SIP Stack [44];
- Csipsimple, which uses the PJSIP for SIP stack [45];
- Sipdroid, which use the MjSIP for SIP Stack [46];
- Imsdroid, which uses the doubango for SIP stack [47].

Each of these projects uses a different SIP stack library, allowing us to test different solutions and, thus, analyze their behavior in order to find one that fits the business prototype requirements. The next topics will present a critical analysis about each one of these libraries: Jain-SIP, PJSIP, MjSIP and doubango.

4.3.3.1 JAIN-SIP

Java APIs for Integrated Networks SIP (JAIN-SIP) is a java standard interface to a SIP signaling stack, providing a business prototype implementation of some of the SIP extensions, most notably for IM. The intended users of this project are SIP developers who require powerful access to the SIP protocol to help them implement server-side and Android applications. JAIN-SIP can be utilized in a user agent, proxy or imbedded into a service container. However this API is typically used for client-side application development [48].

4.3.3.2 PJSIP

PJSIP is a free and open source multimedia communication library written in C language implementing standard based protocols. It combines signaling protocol (SIP) with rich

multimedia framework and traversal functionality into high level API that is portable and suitable for almost any type of systems ranging from desktops, embedded systems, to mobile handsets.

PJSIP supports audio, video, presence, and instant messaging. It is written in C, which means that to implement PJSIP features on an Android application we need an adapted library for Eclipse, such as pjsip-jni wrapper, in order to integrate the SIP into the Android application [49].

4.3.3.3 MjSIP

MjSip is a complete java-based implementation of a SIP stack. It includes all classes and methods for creating SIP-based applications. It implements the complete layered stack architecture and includes higher level interfaces for Call Control and User Agent implementations. MjSip comes with a core package implementation that includes [50]:

- All standard SIP layers and components;
- Various SIP extensions;
- Some useful Call Control APIs (e.g. Call-Control, UserAgent, etc).

4.3.3.4 Dobango

Doubango is an open source framework for both embedded and desktop systems. The framework is written in ANSI-C and has been carefully designed to efficiently work on embedded systems with limited memory and low computing power and to be extremely portable. Because of its portability imsdroid project brought this framework to Android applications [51].

4.3.4 Representational State Transfer

Representational State Transfer (REST) is a variation of software architecture for distributed systems that takes advantage of the technologies and protocols of the World Wide Web. REST has emerged as a predominant web API design model, describing how data objects, or resources, can be defined and addressed, stressing the easy exchange of information and scalability. It attempts to minimize latency and network communication while at the same time maximizing the independence and scalability of components implementation [56].

REST uses a standard method of access: the HyperText Transfer Protocol (HTTP). The HTTP specification provides a number of generic methods, but only four are relevant in terms of REST: GET, used to get access to some resource; POST, used to perform actions of insert, update or extend of a resource; PUT, used to create, update or replace an existing resource; DELETE, as the name suggests, is used to delete a resource [57].

4.3.5 JavaScript Object Notation

JSON (JavaScript Object Notation) is a text-based open standard designed for human-readable data interchange. It was originally specified by Douglas Crockford, and is described in RFC 4627. The official internet media type for JSON is "application/json" and filename extension is ".json". It is derived from JavaScript scripting language for representing simple data structures and associative arrays, called objects. Each object can have one of the following values: number, string, boolean, array, object and null. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages [58]. The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and a web application, serving as an alternative to eXtensible Markup Language (XML) [59]. An example of JSON notation is presented in Figure 14.

```
{
        "employees": [
                { "firstName":"John" , "lastName":"Doe" },
                { "firstName":"Anna" , "lastName":"Smith" },
                { "firstName":"Peter" , "lastName":"Jones" }
        ]
}
```

Figure 14 - JavaScript Object Notation example

4.4 Joyn

Joyn is a specification of GSMA (Global System for Mobile Communications Association, originally Groupe Spécial Mobile), also known as Rich Communication Services (RCS), designed to bring a more engaging communication service supported by a variety of architecture implementation options [52] [53]. RCS is an upgrade that enhances the capabilities of SMS and MMS technologies, from circuit switched technology to an IP based system. It

integrates with the operators IMS architecture, connecting via the appropriate access network, be it Wi-Fi, LTE, 3G or another packet data network. For consumers, it presents the following main features [54]:

- Contact List: service that enables new contacts information such as presence and service discovery;
- Messaging: service that enables a large variety of messaging options, such as chat, group chat, location share, emoticons and file sharing;
- Calls: service that enables the sharing of multimedia content during a voice call, video call and video sharing.

This allows operators to combat increasingly popular Over-The-Top (OTT) services like Skype, WhatsApp and Viber. At present, Joyn is implemented for Android and iOS, with the same name and it uses the user address book to enable the services described above. An example of a Joyn Android implementation is represented in Figure 15 [55].

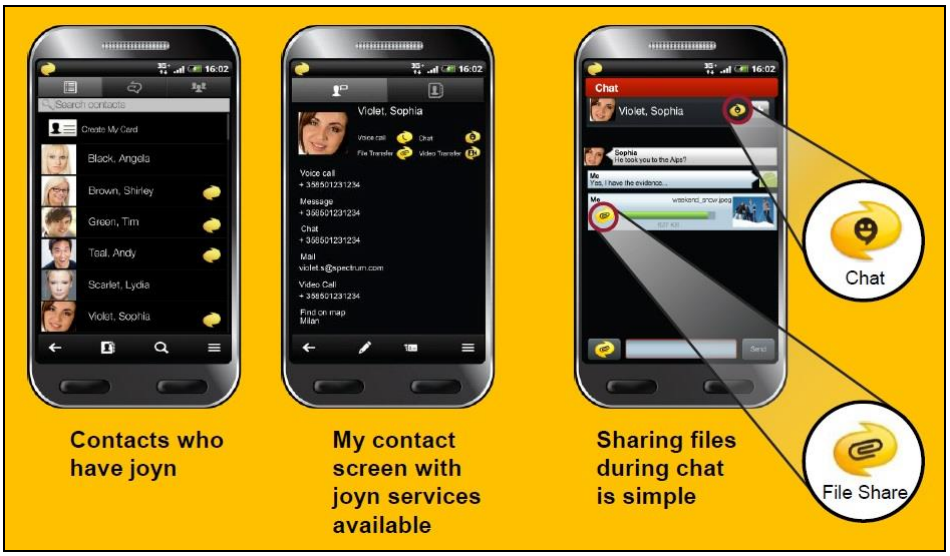


Figure 15 - Joyn features example

4.5 Sprylogics Server

The Sprylogics server uses REST architecture and aims to give interests suggestions based on a certain location (using Global Positioning System (GPS) or network triangulation). It allows the search of a certain interest and is responsible for retrieving all the information requested from a client terminal in JSON format. All the logic and server maintenance is made from

Sprylogics Company. The interests categories that are relevant to the share interests project are the following:

- Weather: check the current weather and forecast for the user current location;
- Movies: check the current box office to search for a certain movie by cast, director, producer, writer, genre, distributor, theater name and movie title;
- Locals to eat: find restaurants, cafes and bars near the user's location and search for a specific place;
- Places: find places near the user's location such as, hospitals, banks, gas station and parking;
- YouTube Videos: search and find videos from YouTube, from the following categories: top rated, top favorites, most popular, most discussed and most viewed. Since YouTube is a global channel there is no need for the user's location.

The request of interests is made through an HTTP GET method using an URL. This URL has a specific structure: application ID, type of data, latitude and longitude of the user, search words and other options. An example of such URL can be seen in Figure 16.

```
http://d.2ya.it/dre/api/rest/search/local?appId=eaee24d8-2e48-4815-a5cb-
fc76718a18b6&query=pizza&lat=43.653226&lng=-79.3831843&user=frankie
```

Figure 16 - Server URL request example

Chapter 5 - Product Backlog

This chapter is related to the first phase of the scrum methodology adapted to the business prototype. Here, all the information involving the process before advancing to the implementation phase, the definition of requirements and functionalities will be presented.

5.1 Functionalities

This section will describe the group of functionalities to be implemented for each internship goal, the business prototype and the implementation of share interests. Each feature has a section, a description, and a priority. The priority is mapped from 1 to 5, where 1 stands for the highest priority and 5 stands for the lowest priority. Functionalities marked with priority levels 1 and 2 are mandatory, and the remaining levels are optional.

5.1.1 Business Prototype Functionalities Definition

Table 8 shows a group of functionalities that are relevant to the implementation of the business prototype.

SECTION	NAME	DESCRIPTION	PRIORITY
Instant Messaging	Individual chat	Communicate with contacts in an one to one chat.	1
	Group chat	Create a group chat by clients.	1
	Private and public chats	Create private and public conversation.	1
	User state	Check if a contact is online, away or occupied.	1
	SMS messages	Use SMS to continue an individual chat.	3
	History	Save all the data exchanged for future review, messages, files, etc.	2

SECTION	NAME	DESCRIPTION	PRIORITY
	Offline	The solution should notify the user by email or SMS if he do not authenticate in the system for a long period of time.	4
Chat Options	File sharing	Add attachments to the chat messages.	1
	Hashtags	Allow users to use special characters to set the topic of a post, such as '#' for the topics and '@' for users.	4
	Smart pastes	Show data with a different design based on the written information (ex: a piece of java code or a link).	5
	Images preview	If some participant in a chat shares an image, a preview is shown.	2
	Videos preview	If the user shares a link from YouTube, a preview under the link is shown in the chat window.	2
	Spell checking	When a user write a message the system auto-correct the input words based on the defined language.	5
	Emotions	The chat detects a group of characters and converts it to small images.	5
	Real-Time Translation	Translate the message sent when the users have the application defined with different languages.	5
	Streaming video	Allow users to use their camera device to stream a video for the chosen contacts.	3
Third party applications	Documents	Documents uploaded can go to the tag repository (e.g. Google Docs).	3
	Social networks	User can login using other services accounts.	2
	Upload	Upload all files to an online storage service with automatic synchronization (e.g. Dropbox).	3
	Tag Management	Manage tags using some services like github, assembla, jira, confluence, redmine, scrumdesk and scrumdo.	4
	Calendar	Integrate Google Calendar into the tags schedule.	
Calls	Conference and individual video calls	User can communicate with his contacts using video calls in an individual or group way.	1
	Conference and individual audio calls	User can communicate with his contacts using audio calls in an individual or group way.	1
	Record calls	Option that allows the recording of every call.	1
Tags	Discussions	Start a group discussion inside a tag.	2
	Calendar	Create a tag calendar.	3

SECTION	NAME	DESCRIPTION	PRIORITY
	To-do lists	Create list of things to do inside a tag.	2
	Tasks	Create tasks inside the lists.	2
	Collaborative Gantt chart	Allow a group Gantt charts for the person inside a tag.	3
	Notes	Create notes to save drafts, concepts and ideas.	3
Other	Search	Search mechanism to find data inside the application.	1
	Sound and notifications	Sound and notifications.	1
	Editing files	Allow user to edit files in real-time files.	3
	Security	Encryption to protect the user data.	2
	IDE	Online development environment (IDE) for common languages.	5

Table 8 - Business prototype functionalities list

5.1.2 Share Interests Functionalities Definition

The group of functionalities for the implementation of share interests is presented in Table 9.

SECTION	NAME	DESCRIPTION	PRIORITY
UI	Interests Feed	Present a variety of interests based on the user's location.	1
	Share interest	Share an interest with a contact or a group of contacts.	1
	Search	Search for interests based on keywords.	1
	Content Presentations	Present in the chat window the shared interest.	1
Content	Weather	Present weather results based on the user's location.	1
	Movies	Present movies results based on the user's location.	1
	Locals	Present locals to eat results based on the user's location.	1
	Places	Present places of interest results based on the user's location.	1
	YouTube Videos	Present YouTube video results based on the user's location.	1

Table 9 - Share interests functionalities list

5.2 Requirements

In this section, we will present all requirements and constraints of the product to be developed. The final solution will adhere to these requirements. Each of the requirements has a unique identifier (code) so it can be traced throughout the entire project, where "SR_" refers to "Software Requirement", the first number is associated with the version of the software and the other two is the number of the requirement. Every requirement is accompanied by a description and a priority level. This priority level is mapped on integers from 1 to 5, where 1 stands for the highest priority and 5 stands for the lowest priority. Software requirements marked with priority levels 1 and 2 will be implemented regardless of resources. Software requirements marked with priority levels 3, 4 and 5 are optional.

5.2.1 Business Prototype Functional Requirements

The functional requirements describe the solution functionalities visible to the final users. These features make the UX richer and the final solution more mature. The list of the functional requirements is the following:

CODE	DESCRIPTION	PRIORITY
[SR_101]	The solution should have an authentication mechanism based on username and password.	1
[SR_102]	The solution should have an authentication mechanism based on third party solution.	3
[SR_103]	The solution should have a way to create new accounts.	1
[SR_104]	The solution should allow users to edit their profile.	2
[SR_105]	The solution should display a list of contacts.	1
[SR_106]	The solution displays the information for each contact.	2
[SR_107]	The solution should allow adding a new contact.	2
[SR_108]	The solution should allow removing a contact.	2
[SR_109]	The solution should display a list of messages exchanged with a contact or a group of contacts.	1
[SR_110]	The solution should display a list of files exchanged with a contact or a group of contacts.	1
[SR_111]	The solution should allow users to see a contact state: online, away or offline.	2
[SR_112]	The solution must allow users to search for a contact, tag, to-do lists, tasks and files.	2
[SR_113]	The solution must allow the users to communicate with their contacts in an individual discussion mode.	1
[SR_114]	The solution must allow the users to communicate with more than one contact in the same discussion conversation.	1

CODE	DESCRIPTION	PRIORITY
[SR_115]	The solution should allow users to delete discussions.	1
[SR_116]	The solution must allow audio calls with one contact.	1
[SR_117]	The solution must allow users to communicate through video calls with one contact.	1
[SR_118]	The solution must allow users to communicate through audio call with more than one contact a time.	1
[SR_119]	The solution must allow users to communicate through video call with more than one contact a time.	1
[SR_120]	The solution must allow the users to share files.	2
[SR_121]	The solution must allow the download of shared files.	2
[SR_122]	The solution must allow users to see files content.	2
[SR_123]	The solution must allow the users to record audio calls.	2
[SR_124]	The solution must allow the recording of video calls.	2
[SR_125]	The solution must allow the user to send audio and video records.	2
[SR_126]	The solution should notify the user by email or SMS if he does not authenticate in the system for a long period of time	4
[SR_127]	If a user sends a message when the solution is offline, it should save it to send it when it is back online.	4
[SR_128]	The solution should save a history of text messages and files exchanged in a discussion.	2
[SR_129]	The solution should notify the user of new evens, when the solution is in background.	3
[SR_130]	The solution should show the user a preview of data attached to the exchange messages.	3
[SR_131]	The solution should allow creating tags.	2
[SR_132]	The solution should display a list of tags.	2
[SR_133]	The solution should allow the creation of to-do lists.	2
[SR_134]	The solution should allow creating tasks for a certain to-do list.	2
[SR_135]	The solution should display to-do lists and their tasks in a tag.	2
[SR_136]	The solution should display a list of files associated with a tag.	2
[SR_137]	The solution should display a list of discussions associated with a tag.	2
[SR_138]	The solution should display a list of contacts associated with a tag.	2
[SR_139]	The solution should allow to edit and eliminate a tag.	2
[SR_140]	The solution should allow a user to assign a task.	2
[SR_141]	The solution should allow to edit and eliminate a to-do list.	2
[SR_142]	The solution should allow to edit and eliminate a task.	2
[SR_143]	The solution should allow video streaming.	3

Table 10 - Business prototype functional requirements

5.2.2 Business Prototype Non-functional Requirements

The non-functional requirements define the overall qualities or attributes of the resulting system rather than its functionalities.

CODE	THEME	DESCIPTION	PRIORITY
[SR_144]	Performance	The system should be responsive to user interactions, having a fluid response to actions, with the minimum response time. This requirement has to be taken as a quality requirement and it is very important to the whole solution.	2
[SR_145]	Usability	The solution should have a friendly interface, simple and intuitive, but with a professional look and feel adapted to a tablet screen size. Interaction with the system will be provided by an Android application user interface.	2
[SR_146]	Security	Security is a very important part in the business scope, which includes the software used. So the system should encrypt the exchange data between users.	3
[SR_147]	Recoverability	The solution should make constant backups of the user data into a cloud service provider in a specific period of time. If the user for some reason needs to re-install the application, all the information is restored.	3
[SR_148]	Availability	The solution should allow the users to check their data even when the device is offline.	2
[SR_149]	Compatibility	The solution should work on any Android tablet, with a camera and a microphone.	1
[SR_150]	Language	The language used in the solution will be English.	1

Table 11 - Business prototype non-functional requirements

5.2.3 Share Interests Functional Requirements

The list of the functional requirements for the share interests is the following:

CODE	DESCIPTION	PRIORITY
[SR_151]	The solution should have a location mechanism.	1
[SR_152]	The solution should divide the interests into categories.	1
[SR_153]	The solution should present a list of interests to the user.	1
[SR_154]	The solution should allow the user to search for an interest.	1
[SR_155]	The solution should allow the user to share an interest.	1

Table 12 - Share interests functional requirements

5.2.4 Share Interests Non-functional Requirements

The non-functional requirements define the overall qualities or attributes of the resulting system rather than its functionalities. For the implementation of share interests they are presented in Table 13.

CODE	THEME	DESCIPTION	PRIORITY
[SR_156]	Performance	The system should be responsive to user interactions, having a fluid response to actions, with the minimum response time. This requirement has to be taken as a quality requirement and it is very important to the whole solution.	2
[SR_157]	Usability	The solution should have a friendly interface, simple and intuitive. Users shouldn't need to learn how to use it and it must present a professional... screen size, using the host application design.	2
[SR_158]	Recoverability	The solution should have a cache system to store data from the user.	3
[SR_159]	Availability	The solution should allow the users to check their data even when the device is offline.	2

Table 13 - Share interests non-functional requirements

5.3 Business Prototype User Stories

In this section we present the user stories related to the most relevant features of the business prototype. Each story has a unique identifier and is accompanied by a description, what is needed to access the feature, the related requirement identifier, its priority and the effort needed to implement it. The estimation is mapped using a planning poker sequence {0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?}. Where 0 stands for a user story that is already done or it just needs a few minutes of work. The ½ to 13 represent the effort inside a single sprint, the 20 represent the effort of one sprint, the 40 two sprints and the 100 three sprints. Finally the "?" is for when there is no idea about the effort the user story needs.

CODE	NAME		
DESCIRPTION			
TAKS	RELATED REQUIREMENTS	PRIORITY	ESTIMATION

[US_101]	**Contacts**		
As a user, I want to have a list of contacts so that I can easily stay in touch with them.			
To do that the user only needs to access the contacts section.	[SR_105][SR_106] [SR_107]	**1**	**2**

[US_102]	**User availability**		
As a user, I want to check my contacts availability so that I know if they are available to communicate.			
To do that the user only needs to access the messages section and a view will signalize the availability option.	[SR_111]	**1**	**2**

[US_103]	**Discussions**		
As a user, I want to be able to send messages so that I can communicate with my contacts.			
To do that the user only needs to access the discussions section and choose the "write new message" option.	[SR_109][SR_110] [SR_113][SR_114]	**1**	**13**

[US_104]	**Voice calls**		
As a user, I want to be able to make voice calls so that I can stay in touch with my contacts.			
To do that the user only needs to access the messages or the contacts section and choose the "voice call" option.	[SR_116][SR_118]	**1**	**20**

[US_105]	**Video calls**		
As a user, I want to be able to make video calls so that I can stay in touch with my contacts.			
To do that the user only needs to access the messages or the contacts section and choose the "video call" option.	[SR_117][SR_119]	**1**	**20**

CODE	NAME		
DESCIRPTION			
TAKS	RELATED REQUIREMENTS	PRIORITY	ESTIMATION

[US_106]	**Record calls**		
As a user, I want to be able to record all my calls so that I can review them when I want.			
To do that the user only needs to make a voice or a video call first and then choose the option "record".	[SR_123][SR_124]	**1**	**13**

[US_107]	**Search**		
As a user, I want to be able to search so that I can easily find information.			
To do that the user needs to access the search section and enter the keywords for the needed object.	[SR_112]	**1**	**8**

[US_108]	**Tags**		
As a user, I want to be able to create tags so that I can give a context to the information.			
To do that the user only needs to access the tag section and choose "create new tag" option.	[SR_131][SR_132] [SR_139]	**2**	**5**

[US_109]	**To-do lists**		
As a user, I want to be able to create lists so that everything needed to be done in the project is organized.			
To do that the user only needs to access the tag section and choose "create a new list", or access the to-do lists section and do the same process.	[SR_133][SR_135] [SR_141]	**2**	**5**

[US_110]	**Tasks**		
As a user, I want to be able to create tasks so that I can organize my work.			
To do that the user only needs to access the tag section, select a list and choose to "create a new task", or access the to-do lists section and make the same steps.	[SR_134][SR_140] [SR_142]	**2**	**3**

CODE	NAME		
DESCIRPTION			
TAKS	RELATED REQUIREMENTS	PRIORITY	ESTIMATION

[US_111]	**Send files**		
As a user, I want to be able to send files so that I can share important documents with my contacts.			
To do that the user only needs to access the projects section or access the messages section.	[SR_120][SR_121] [SR_122][SR_125] [SR_130]	**2**	**8**

[US_112]	**History**		
As a user, I want to check a discussion history so that I can remember my and other contacts activities.			
To do that the user only needs to access the messages section and choose the contact.	[SR_128]	**1**	**8**

Chapter 6 - Implementation

This chapter aims to report all the work involving the development of the business prototype for Android tablet and the implementation of share interests in Joyn application. We will describe how each project was designed and the different elements that compose them.

6.1 Business Prototype

This section will address all the work involved in the business prototype implementation. We will start by describing the Android application's high level and software architectures integrated with the various modules, the local database, the design of the user interface and the integration between all the necessary components.

6.1.1 High Level Architecture

In order to create a mature and useful business collaboration tool, it is essential to link some key components, such as an appropriate front end, access to the internet, a local database, and finally, a mobile device where the application will be running. Figure 17 illustrates the high level network architecture for the proposed business prototype, showing the relationship between the different elements.

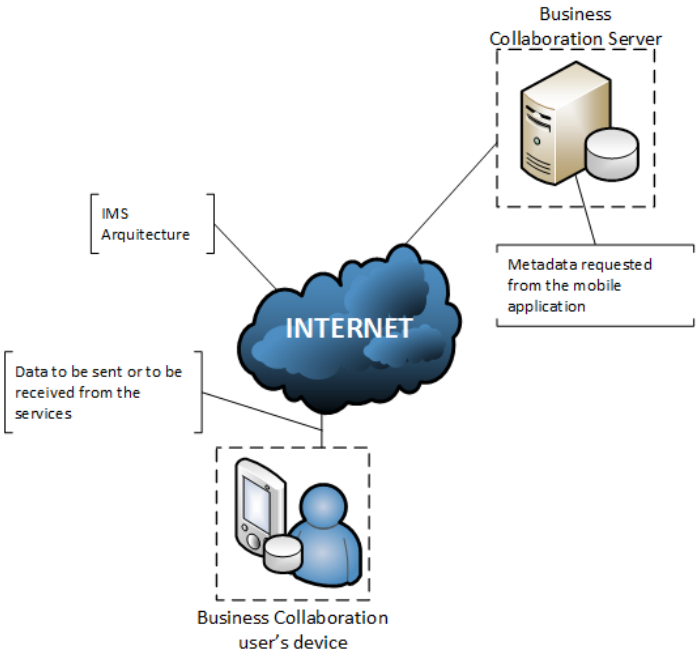


Figure 17 - Business prototype high level architecture

The architecture in Figure 17 is composed by several elements that communicate with each other through the internet. This allows the communication between users and the control of what is stored on the server and on the devices. The Android application is responsible for presenting the information to the user. The server is responsible for storing the user information and for running and maintaining a group of services that allow business collaboration features, such as chat, audio and video calls and file share.

6.1.2 Software Architecture

The software architecture is represented in Figure 18 by a block diagram. This architecture is divided into several layers, each one with a specific purpose. Analyzing from the bottom to the top, we start by describing the **local database**, which is used by the application to store several types of information and it works as the first level of the implemented cache system. This database will be explained with more details in the next section. In the next layer we find different components that do all the communication with the local database, the **managers**. Although each of these components has a distinct functionality, they have similar purposes, namely, to prepare the information, send it to the server or storing it on the local database. At the top of the architecture there are layers that communicate with the cloud service. The first layer, called **SIP Stack Library**, provides the SIP features needed to establish multimedia

communications. The next layer, **Cloud Service API**, has the objective of implementing all the methods for the communication with the server. Finally we have the **Background Service** and the **User Interface**. The former represents the service that runs transparently to the user and performs all the needed operations in the background. This service has the current data used by the user and it works as second level of the cache system. The latter is responsible for presenting the current state of application to the user. In this layer, there are the controllers for each view of the application.

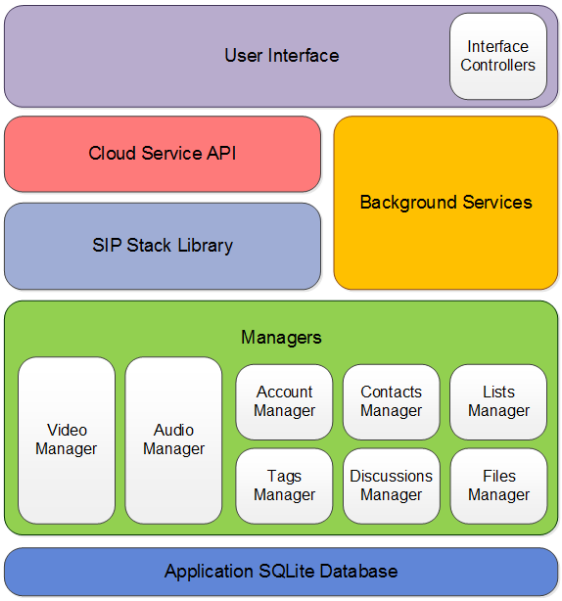


Figure 18 - Application software architecture

6.1.3 Local Database

The local database was created with the objective of forming a system that keeps all the information from the server in the application, working as a cache system. Built using SQLite database [61], it is composed of the following tables: users, relationships, participants, discussions, messages, attachments, tags, to-do lists, tasks and notifications. There are different relationships between these tables, as represented in Figure 19:

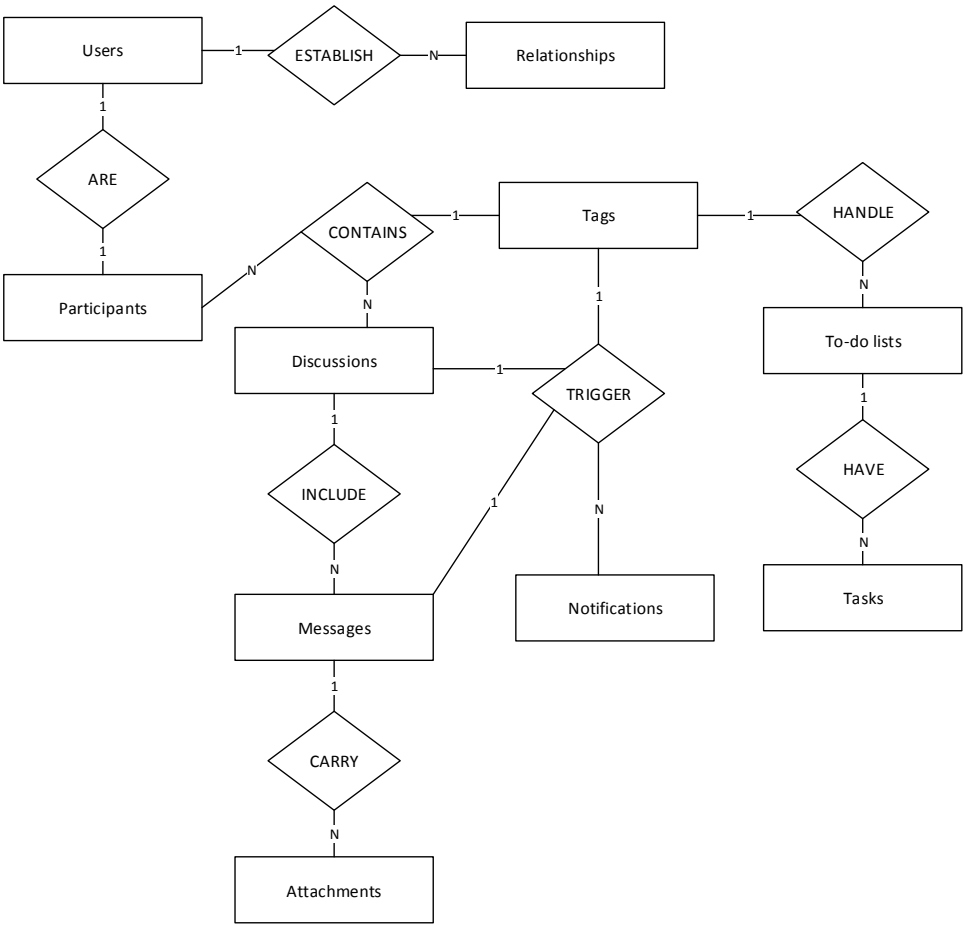


Figure 19 - Local database schema

The local database structure is identical to the server database, having several tables related to different types of content and having support for multi-user. These tables contain information involving every user that logs into the application, such as his name, email, phone number, picture, messages exchanged, contact list and tags created. We made the decision of storing digital data, like pictures, in the database using BLOB format instead of using files because we want the data to be persistent and always available to the user even when an external storage is not available. This local data allows the application to be quicker, reduces the requests to the server, and improves the availability, allowing the user to access the information in the context of the application, even if the device is offline.

6.1.4 User Interface

The graphical user interface was intended to be as simple as possible. The different screens must be consistent, so that the user does not have to learn different menus in order to use them intuitively. These screens were all built using Eclipse Integrated Development Environment (IDE) with the Android SDK, which produced XML for graphical design.

With the objective of having an appealing interface, Google's guidelines and all the information presented on the section "4.2 Android Usability" were used in the process of designing the application. The first step was to create a group of mockups to be used as guidelines for the designers. Figure 20 represents a mockup of the user profile, created using Balsamiq tool [60] (all the developed mockups involving all the application interfaces and the popups dialogs are on the Appendix C - Mockups).

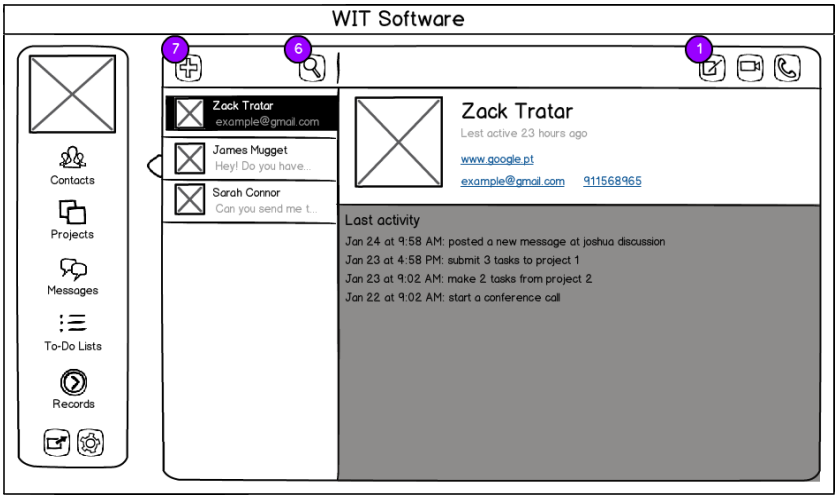


Figure 20 - User profile mockup

The development of the UI was an iterative process divided into two phases, for the purpose of creating a solution that fits Google's guidelines. During the development of the first version of the UI that came out of the mockups we realized that some elements did not fit the patterns established by Google. Those elements included the vertical tabs on the left, and the second bar for each tab. So we had to make some changes and a second version of the UI was created. Some new elements included the action bar [62], contextual action bar [63] and navigation tabs. Figure 21 and Figure 22 show the changes between the UI versions.

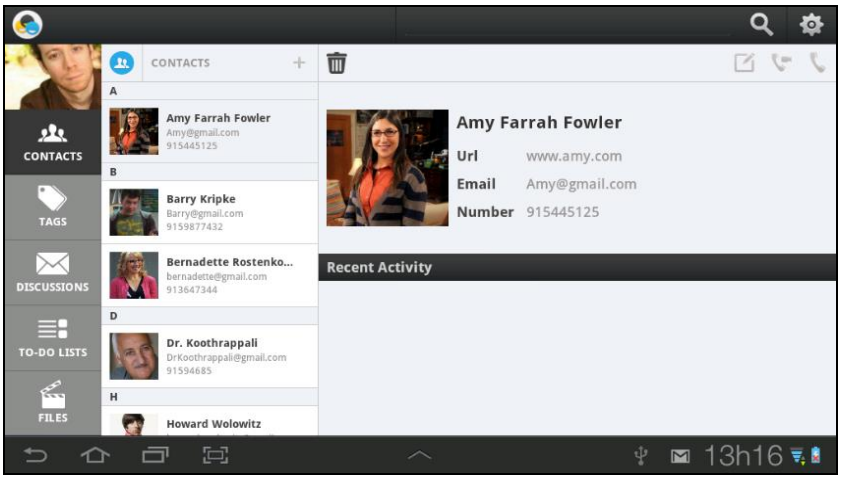


Figure 21 - Contacts interface first version

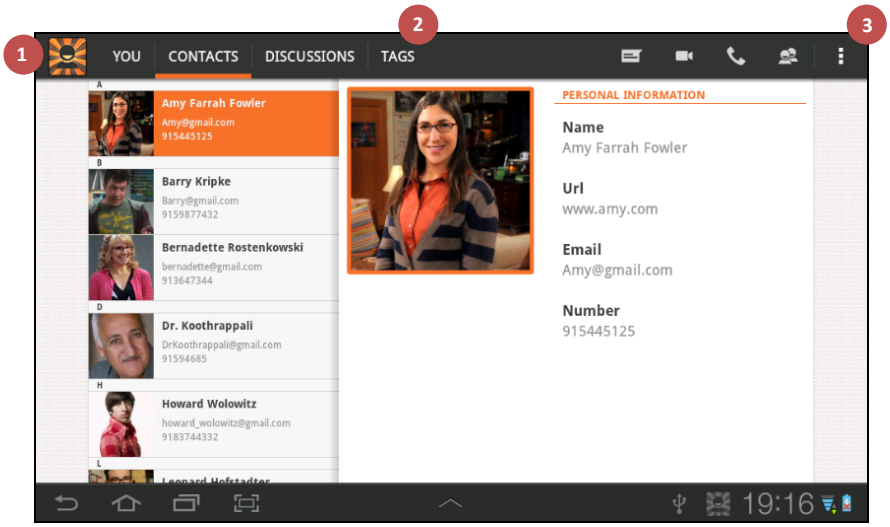


Figure 22 - Contacts interface second version

Legend of Figure 22:

1. Action bar “is a window feature that identifies the application and the user location, and provides user actions and navigation modes” [62];
2. Navigation tabs;
3. Contextual action bar “appears at the top of the screen to present actions the user can perform on the currently selected item(s)” [63].

6.1.5 Contextualization Mechanism

Tags constitute the highlight of the business prototype's mechanism, and they are based on labels used for the contextualization of objects inside the Android application (contacts, discussions, files and to-do lists). A label in this context is composed of a word (e.g. work, friends and meetings) and a color, making it easy to differentiate them. Inside the Android application the user creates a label and objects associated to it through a drag and drop system. After the association is done, an indicative object is created so that the user can visually distinguish which label belongs to each object. This allows the user to know in what context he had a certain conversation or shared a file. Figure 23 presents the user interface implemented into business prototype for tags.

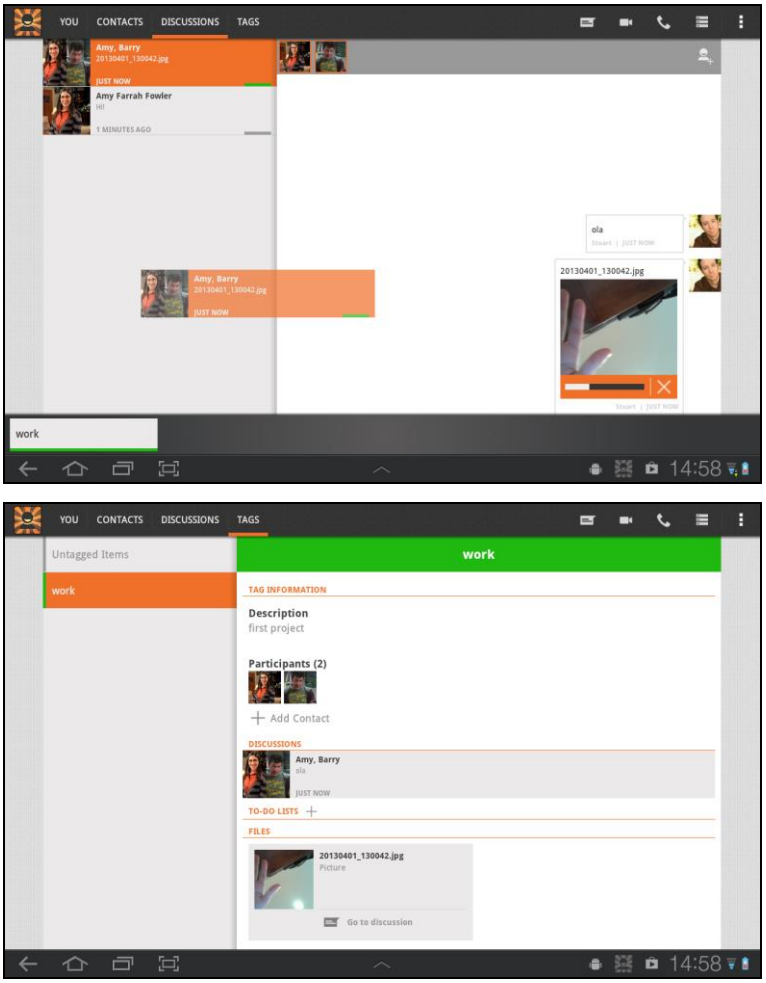


Figure 23 - Business prototype tags user interface

6.1.6 Performance Concerns

In order to create a responsive application, several measures were taken, such as the creation of a background service for a second level of cache, implementation of threads and the selection of variables type with the best performance. The background service is responsible for storing the data that is being used by the user, in order to avoid multiple requests to the database. When a group of information that is not present on the second level of cache is requested, a request is sent to the database through a thread, the new elements are added to the background service and presented to the user.

The use of threads in the entire application is very important, as far as performance is concerned. When a user starts an Android application, a thread called "main", or “UI”, is automatically created. The main thread is in charge of dispatching the events to the appropriate elements and this includes the drawing events. It is also this thread that the user interacts with, when touching the device. A single thread model can yield poor performance in an Android application that does not consider these implications. Since everything happens on a single thread performing long operations, like database queries or network access, this will block the whole UI. No event can be dispatched, including drawing events, while the long operation is underway [64]. This will make the application look hung, from the user's perspective. Even worse, if the UI thread is blocked for more than 5 seconds currently, the user is presented with the "application not responding" dialog [65]. Having this in mind, threads where implemented when making database queries and communicating with the server through the network.

The last measure was the variables type applied to the second level of cache. We want the information to be retrieved as fast as possible. Instead of using HashMaps, we have used SparseArrays. We have made this decision because the HashMaps has a big overhead in terms of memory. This is due to its structure: a map is represented internally as an array of lists, where each element in these lists has a key. Both the key and value are object instances. Differently from the structure of SparseArray, which only has arrays of primitives as keys and an array of objects as values, in HashMaps both the key and value are object instances. SparseArray saves the data according to the key value (e.g. a key with the value 5 will save the data in the position 5 of the array, having the position 0,1,2,3 and 4 empty) [66]. It is intended to be more efficient than using a HashMap to map integers to objects and this improves the performance of the cache system. The second decision was the change of various ArrayList to HashSet. This modification was made because, in some occasions, we only want to store a certain ID and check if an ID is present on the list (e.g. before adding a new contact to a

discussion, we need to check if it is already there). For this particular problem HashSet proved to have a better performance [67].

6.1.7 Communication Logic

The communication logic is the core of the Android client application, establishing its structure and relationship between the different components. Figure 24 represents an example of a connection between the different elements present on the business prototype. The example given is when we want to get all the contacts of a user. The diagram is divided into three types of connections represented by three colors: the blue color represents the request of information, the green the connection with the second and the first level of cache and finally the yellow color the update of the views with the new data. First, the view request the controllers for the data to be presented (1) and the controllers check if the element required exists on the cache (2). If not, the controllers ask the manager to communicate with the database to get the data (3). The manager launches a thread that gets the data from the database (4, 5) and updates the elements on the cache (6). Finally, an update request is sent to the controllers (6) and the view is refreshed and the data is presented (7).

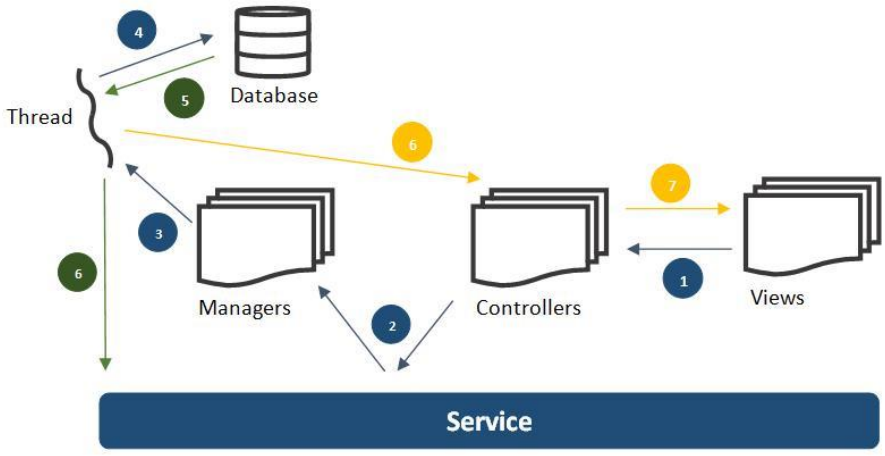


Figure 24 - Get all the user contacts diagram

All other connections stand on the same diagram. First, the system tries to get the data from the second level of cache and if the elements do not exist a request to the database (second level of cache) is made. If the information is not present on the database neither, a request to the server is made.

6.2 Share Interests

This section is focused on the work related to the share interests' implementation, describing how Joyn and Sprylogics server were integrated. The server has the main purposes of keeping the application updated with interests suggestions based on the user's location. The Android application, Joyn, uses the server information to present interests to the user and to let him share them with his/her contacts.

6.2.1 Server Communication

As described in section "4.5 Sprylogics Server", Sprylogics server aims to give interests suggestions (weather, movies, locals, places and YouTube videos) based on the user's location. Although the server supplies other types of interest categories, such as products and web search, it was decided by WIT Company not to integrate these types of interests. The communication between the Joyn application and the Sprylogics server is described in Figure 25.

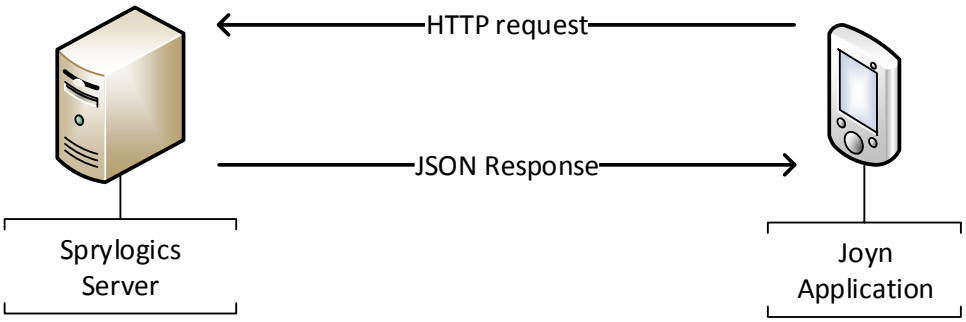


Figure 25 - Joyn integration with Sprylogics architecture

From Figure 25 we can see that the communication is made from an HTTP request and a JSON response from the server. From the user selected option on Joyn application, an HTTP request with a specific URL is made. This URL is generated by Joyn with specific structure. After the request, the server responds with a JSON structure containing the requested information. The response is processed in the Joyn application, using a parser, and finally presented to the user.

6.2.2 Software Architecture

The software architecture related to the integration between Joyn and Sprylogics is presented in Figure 26. It is divided into several layers, each one with its purpose. The last layer is composed of the **Interests Manager** and has several components, each representing a unique type of interest. This layer components are responsible for communicating with Sprylogics server, parsing the data retrieved by it and storing it into the cache element. In the layer above we find the **Location Service** and the **User Interface**. The former represents the service that is connected to the user interface and provides the latitude and longitude of the user, based on GPS or network triangulation. The latter is responsible for presenting the interest suggestions to the user. In this layer, there are controllers for each view of the application.

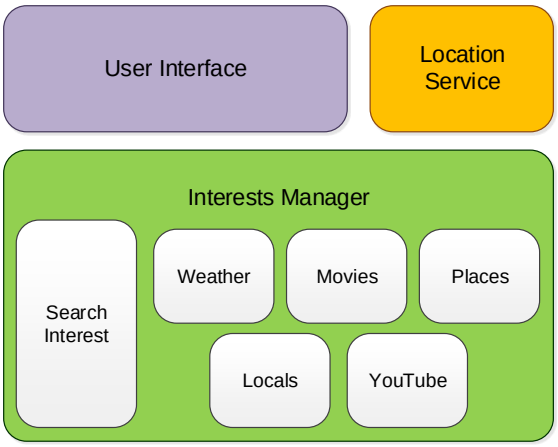


Figure 26 - Sprylogics integration architecture

With the purpose of minimizing the number of request to the server, the retrieved information and the user location are stored into the manager. If the user's location changes 200 meters from the last one, the manager will ask the server for new data, the cache is rewritten and the UI is refreshed.

6.2.3 Interests Feed

The interests feed is the local where the interest suggestion are shown to the user. The interests feed UI was developed using Joyn black theme and guidelines in order to fit the design of the application. Besides showing the interests list, a group of other functionalities were developed. One of these features is the ability to search for a certain interest. By simply clicking the search

option (1) a search bar will be shown to the user (2). The user enters some search words and the interests manager will ask the Sprylogics server for the information of every interest category that matches the entered information. The UI receives the response, treats the information and refreshes itself. If no information exists, a message is shown to the user (4). A performance concern in the search functionality is that the searched information does not override the cache stored into the interests manager. This allows an instant load of the information when the user closes the search box (3). The final two functionalities are related to the sorting of the interests list. The user can sort the list by name by selecting the option to sort the list from A to Z. If the user selects this option again the list is sorted from Z to A (6). There is also a default button to set the list item to their default positions (5) and a refresh option (7) to reload the items in the interests list. Figure 27 includes several screenshots of Joyn application interests feed with all of these functionalities.

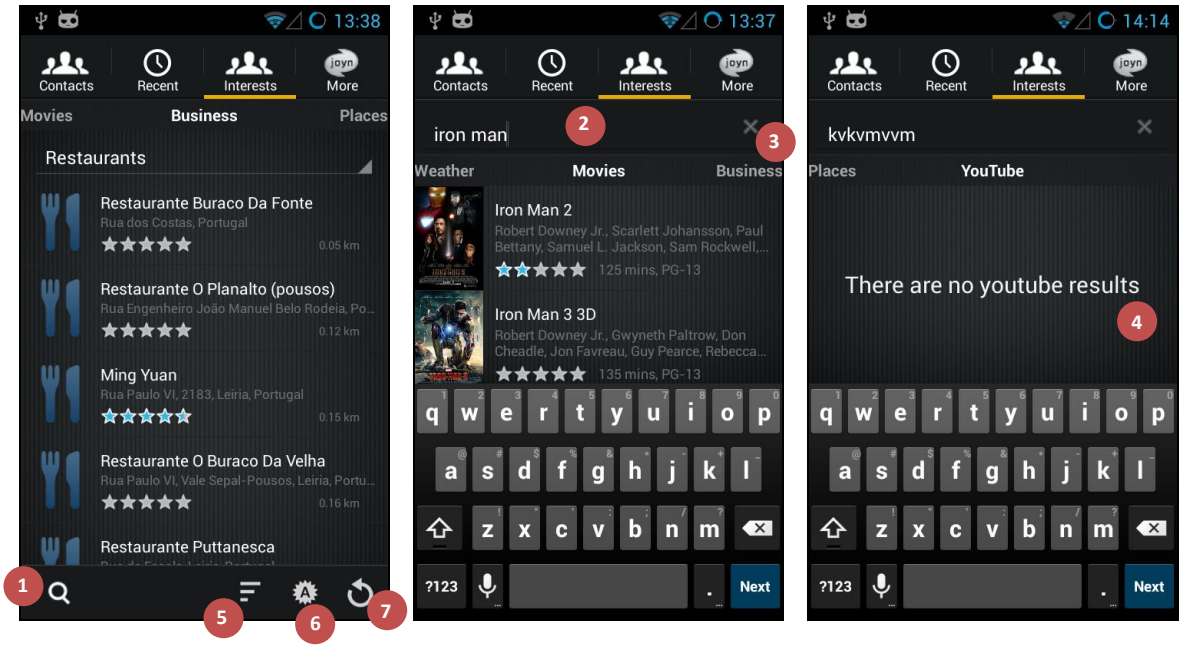


Figure 27 - Joyn interests feed screenshot

The interests feed is responsible for managing the presentation of all the interests retrieved by Sprylogics server based on the user's location. It is integrated into Joyn by a new tab, using a group of pages (called view pager) for presenting, at the same time, the following interests categories: weather, movies, locals, places and YouTube videos. When the user selects the interests tab, the location service is activated and begins to check if the user's location is stored in the interests manager. If the cache does not hold the user's location, the location service searches for it. The service will start by using the GPS and if it is offline, it uses the network

triangulation. When the location is found, it is stored into the interests manager in order to be used later, if needed. An example of a communication between the interests feed and the manager is shown in Figure 28.

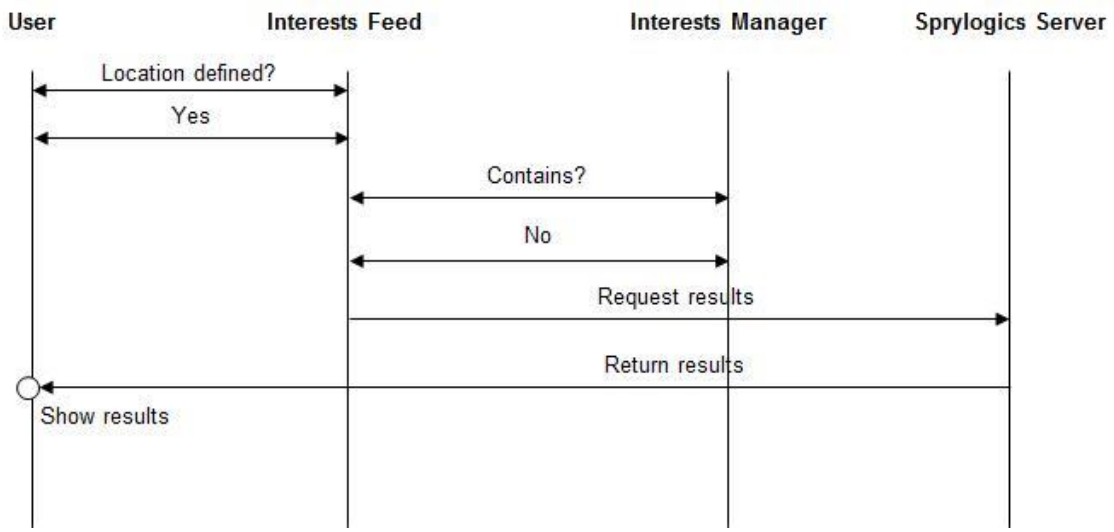


Figure 28 - Interests feed communication UC

Figure 28 shows a UC where the interests manager produces a cache miss, and that will result in a request to Sprylogics server. The result from the server will be loaded asynchronously according to the user's location, and the new object is stored into the cache inside the interests manager.

6.2.4 Share Interest Object

The share interest is a functionality that lets the user share a certain interest with one of his contacts. This feature is available using the share option (8 and 9 of Figure 29) in two different places: the interest details or the chat window. From the interests details the user chooses the share option (8) and then the contact list window is shown. The user simply selects the contact and the interest is shared with it. For the second option, the chat window, the user follows the same process by selecting the share option (9) and the interest feed is shown and interests categories (weather, movies, locals, places and YouTube videos) are presented. Differently from the original interests feed, if the user chooses an interest, it is shared with the contacts present in the previous chat window. Figure 29 exhibits a group of screenshots taken from Joyn that show the described functionalities.

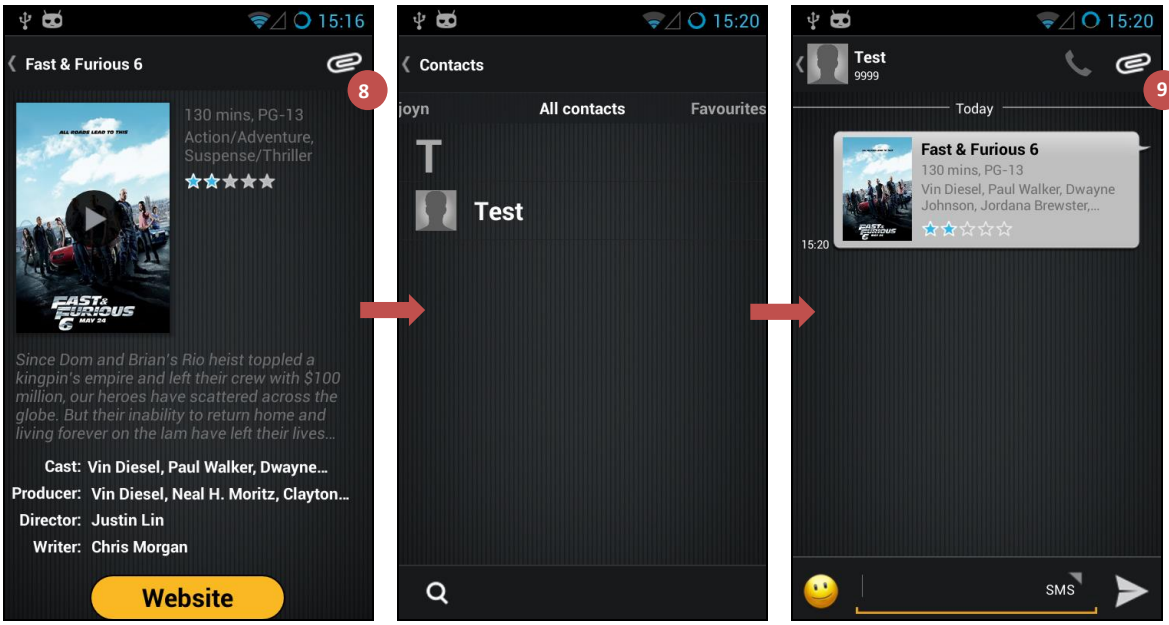


Figure 29 - Joyn share interest screenshot

6.2.5 Performance Concerns

With the purpose of creating a fast and responsive UI, some performance considerations were taken into account when developing the interests feed. One of the problems with this element is the number of items that are retrieved from the server from the initial request. Requesting a lot of items, for example 45, will mean a total of 180 items to be loaded because we have 4 views, movies, places, locals and movies, each with a list of item (4 x 45 = 180).This leads to a slow UI and, consequently, a bad UX. So we limit the number of items initially retrieved from the server to 15 items. If the user wants to see more suggestions, he simply needs to scrolls down and another 15 items will be loaded into the current list. This reduces significantly the overhead of the network and improves the UX.

Another concern is related to the views hierarchy. A lot of level in a layout (views inside views) may cause a significant increase of memory consumption, especially if that layout is repeated through a list of elements, which is the case. Sprylogics SDK as a sample for Android platform that exemplifies how to communicate with their server and how to present the retrieved information to the user. The problem is that they design every item with a lot of hierarchical levels (5 to 6) which really increases the memory needed to draw every element in a list. A memory test was made using “little eye” tool [68] and Figure 30 presents the result.

Figure 30 - Memory consumption test (Sprylogics Android and Joyn)

From Figure 30 we can verify that Joyn consumes less memory than Sprylogics SDK (approximately 23% less) which really improved the available memory. However, there are some considerations about previous result, as Joyn uses a view pager (this decision was made because we want to improve the user UX) [69] to show all the interests categories at the same time, in contrast to Sprylogics SDK, which uses a list to show one category at a time. In order to view locals and movies interests, the user has to choose the option locals, check the results, go to the previous window and select movies. With this in mind we opted to use layouts with the maximum of two hierarchical levels and thereby reducing the memory consumption. The full test result is presented in the Appendix D - Performance Tests.

(Starts on page odd)

Chapter 7 - Conclusions

In this chapter I will describe the knowledge acquired during the internship and during the use of the mobile operating system, Android. It is also made a reflection about the developed projects and future work.

7.1 Acquired Knowledge

From the beginning until the end of the internship it was a great experience in terms of mobile development. Developing for Android using Scrum gave me the possibility of learning and use a new development methodology. Also, I have increased my developer skills, learning some performance and stability techniques.

7.2 Reflection

The major contribution of the business prototype is to bring the concept of tags into a business collaboration solution. The tests using the business prototype of the application revealed that the solution is able to store all the data on the cache system, accessible on the background service. This proved that the user can access to its information even when offline. Unfortunately the business prototype did not advance to the production phase, which led to a restructuration of internship objectives and priorities. All the work made in the business prototype was used for code review with the objective of checking if any part was useful for other Android applications. Since WIT Company clients prefer new features to existing products, as opposed to a new application the share interests project was created. This new project was developed following WIT Company requirements which let me explore new technologies, such as JSON and REST. The developed solution was integrated into a used product, Joyn, and by the time this report was being written the solution was in commercialization process with WIT Company clients.

7.3 Future Work

Future work includes integration of the business prototype project with third party applications, giving the user the possibility to login into the application using his credentials from other services. Another improvement would be refining the user interface that is being used in the tags system. Yet another future consideration is related to the problem that schedules and collaboration on files are not implemented. It would be interesting to integrate the solution with Gantt charts and solutions like Google docs. In relation to the share interests a future concern is to implement more types of suggestions, like products and web search, and the ability to let the user choose which types he wants to view.

References

[1] Management Information Systems Glossary of Terms. [Accessed on 30 May 2013]. Available: http://www.gregvogl.net/courses/mis1/glossary.html

[2] Business Software. [Accessed on 30 May 2013]. Available: www.techopedia.com/definition/24067/business-software

[3] WIT-Software. “Joyn”. [Accessed on 1 May 2013]. Available: https://play.google.com/store/apps/details?id=com.witsoftware.wmc

[4] Sprylogics. “Solution”. [Accessed on 20 April 2013]. Available: www.sprylogics.com/solutions/

[5] Frommer, Dan. "The Future Of Mobile Is The Future Of Everything" (2011). [Accessed on 20 April 2013]. Available: www.businessinsider.com/future-of-mobile-experts-2011-6?op=1

[6] Arandilla, Rachel. " How The Smartphone Invasion Changed The Way We Live ". [Accessed on 20 April 2013]. Available: www.1stwebdesigner.com/design/smartphone-invasion-changed-our-lives/

[7] UPTEC. “Empresas WIT-Software”. [Accessed on 16 March 2013]. Available: www.uptec.up.pt/empresa/wit-software

[8] Schwaber, Ken, and Jeff Sutherland. "The scrum guide." Scrum Alliance (2011).

[9] Wilson, James M. "Gantt charts: A centenary appreciation." European Journal of Operational Research 149.2 (2003): 430-437.

[10] Highsmith J.,Orr K.,Cockburn A., “Extreme programming”, in:E-Business Application Delivery, Feb. 2000, pp. 4-17. Available: http://www.cutter.com/freestuff/ead0002.pdf

[11] Fowler, Martin, and Jim Highsmith. "The agile manifesto." Software Development 9.8 (2001): 28-35.

[12] Wells, Don. "Extreme Programming: A gentle introduction". Available: http://www.extremeprogramming.org

[13] Stapleton, Jennifer. DSDM Dynamic Systems Development Method: the method in practice. Cambridge University Press, 1997.

[14] Schwaber, Ken. Agile project management with Scrum. Microsoft Press, 2004.

[15] Duncan, William R. "A guide to the project management body of knowledge." (1996).

[16] Boehm, Barry W. "Software risk management: principles and practices." Software, IEEE 8.1 (1991): 32-41.

[17] Schwalbe, Kathy. Information Technology: Project Management. Course Technology Ptr, 2010.

[18] Mountain Goat Software. "Daily Scrum", 2005. Available: www.mountaingoatsoftware.com/scrum/figures/

[19] Mello, Carlos Henrique Pereira, et al. ISO 9001: 2000: sistema de gestão da qualidade para operações de produção e serviços. Atlas, 2002.

[20] Fenton, William, PCMAG. "Adobe Connect". [Accessed on 7 September 2012]. Available: www.pcmag.com/article2/0,2817,2387818,00.asp

[21] Adobe Systems Incorporated. "Adobe Connect Overview" (2012).

[22] Fenton, William, PCMAG. "Cisco WebEx Meeting Center". [Accessed on 7 September 2012]. Available: www.pcmag.com/article2/0,2817,2387928,00.asp

[23] Cisco Systems. "Cisco WebEx, Features". [Accessed on 12 September 2012]. Available: www.webex.com/products/web-conferencing/features.html

[24] Palmer, Jeremy. "Fuze Meeting Review, Online Meeting Reviews". [Accessed on 14 September 2012]. Available: www.onlinemeetingreviews.com/reviews/fuzemeeting

[25] TopTenReviews. "Fuze Meeting". [Accessed on 14 September 2012]. Available: web-conferencing-services.toptenreviews.com/fuze-meeting-review.html

[26] Ken Hess, "Pleased to iMeet you: a Cloud-based Audio and Video Conferencing Service", ZD Net, August 13, 2012.

[27] VB News. "Blue Jeans: Flipping 100 billion minutes of audio conference calls into video". [Accessed on 14 September 2012]. Available: www.venturebeat.com/2012/06/27/blue-jeans-video-conference-browser

[28] Lynn, S. "GoToMeeting 4.1." Retrieved July 8 (2010): 2011.

[29] GoToMeeting. "Features". [Accessed on 14 September 2012]. Available: www.gotomeeting.com/fec/online_collaboration

[30] Fink, Cody. Macstories. "HipChat, Business Communication's Must Have". [Accessed on 1 April 2013]. Available: www.macstories.net/reviews/hipchat/

[31] Alcala, Sydney. "Slick Team Communication With HipChat". [Accessed on 14 September 2012]. Available: www.windows.appstorm.net/reviews/slick-team-communication-with-hipchat/

[32] Google. “Android Guidelines”. [Accessed on 15 September 2012]. Available: developer.Android.com/design/index.html

[33] Warren, Gabriela. “Google Hangouts Review - Google+'s Video Chatting Application”. [Accessed on 21 September 2012]. Available: www.mobileoffice.about.com/od/conferencing-tools-reviews/fr/google-plus-hangouts.htm

[34] Mezo, Hub Pages. “Best Android Chat Apps”. [Accessed on 21 September 2012]. Available: www.mezo.hubpages.com/hub/Best-Android-Chat-Apps

[35] Shontell, Alyson. “Early Google Employees Create MightyText To Free Text Messages From Your Phone”. [Accessed on 21 September 2012]. Available: www.businessinsider.com/early-googler-employees-create-mightytext-to-free-text-messages-from-your-phone-2012-6

[36] Weinmann, Tobias, et al. "Testing Skype as an interview method in epidemiologic research: response and feasibility." International journal of public health 57.6 (2012): 959-961.

[37] Rosenberg, Jonathan, et al. SIP: session initiation protocol. Vol. 23. RFC 3261, Internet Engineering Task Force, 2002.

[38] Colaghost. “A simple example of SIP call”. Available: www.colaghost.net/translate/93

[39] Gregory, Peter, et al. SIP communications for Dummies. Wiley, 2006.

[40] Wuthnow, Mark, Jerry Shih, and Matthew Stafford. IMS: A New Model for Blending Applications. Vol. 9. Auerbach Publications, 2009.

[41] Dialogic. “The Architecture and Benefits of IMS” White paper (2009).

[42] Google, Dashboards. [Accessed on 8 August 2013]. Available: developer.Android.com/about/dashboards/index.html

[43] Zheng, Leo. “Our evaluation of Android Gingerbread is native SIP calling with the Nexus S”. [Accessed on 21 September 2012]. Available: www.onsip.com/blog/2011/01/05/our-evaluation-of-Android-gingerbreads-native-sip-calling-with-the-nexus-s

[44] Google Code. “de-tiny-sip”. [Accessed on 21 September 2012]. Available: code.google.com/ /p/de-tiny-sip/

[45] Google Code. “csipsimple. [Accessed on 21 September 2012]. Available: code.google.com/p/csipsimple/

[46] Google Code. “sipdroid”. [Accessed on 21 September 2012]. Available: code.google.com/p/sipdroid/

[47] Google Code.”imsdroid”. [Accessed on 21 September 2012]. Available: code.google.com/p/imsdroid/

[48] Phelim O'Doherty, Mudumbai Ranganathan, "JAIN SIP Tutorial Serving the Developer Community", 2003, PDF.

[49] Prijono, B. "pjsip.org Open source SIP stack and media stack for presence, im/instant messaging, and multimedia communication."

[50] Veltri, Luca. "MjSip-Mini-Tutorial." (2005).

[51] Jons, Johan. EIGC Integration with a Web-browser: Voice Communication through a Web-browser. Diss. Uppsala University, 2011.

[52] GSMA. "joyn Blackbird Product Definition Document" (03 May 2013) , PDF.

[53] GSMA. "Rich Communications". [Accessed on 18 May 2013]. Available: http://www.gsma.com/rcs/

[54] Joyn. "About". [Accessed on 18 May 2013]. Available: www.joynus.com/about/

[55] Mantica Solutions. "Joyn Service". [Accessed on 3 June 2013]. http://www.mantica-solutions.com/rcs_joyn_service.php

[56] Jakl, Michael. "Representational State Transfer."

[57] Allamaraju, Subbu. RESTful Web Services Cookbook. Yahoo Press, 2010.

[58] Crockford, D. RFC 4627 - The application/json Media Type for JavaScript Object Notation. Technical report, Internet Engineering Task Force, 2006.

[59] Bray, Tim, et al. "Extensible markup language (XML)." World Wide Web Journal 2.4 (1997): 27-66.

[60] Guilizzoni, G. "Balsamiq Mockups, Balsamiq." (2010).

[61] Owens, Mike. The definitive guide to SQLite. Apress, 2006.

[62] Android Developers. Action bar. [Accessed on 3 June 2013]. Available: http://developer.android.com/guide/topics/ui/actionbar.html

[63] Android Developers. Menus. Available: http://developer.android.com/guide/topics/ui/menus.html

[64] Android Developers, Blog. "Painless threading". [Accessed on 3 June 2013]. Available: http://android-developers.blogspot.pt/2009/05/painless-threading.html

[65] Android Developers. "Keeping Your App Responsive". [Accessed on 3 June 2013]. Available: http://developer.android.com/training/articles/perf-anr.html

[66] Guihot, Hervé. Pro Android Apps Performance Optimization. Apress, 2012.

[67] Kiourtzoglou, Byron. "Java Best Practices – Vector vs ArrayList vs HashSet". [Accessed on 3 June 2013]. Available: http://www.javacodegeeks.com/2010/08/java-best-practices-vector-arraylist.html

[68] Little Eye Labs, Performance, Analysis and Productivity Tools. [Accessed on 18 May

2013]. Available: http://www.littleeye.co/

[69] Android Developers, “Using ViewPager for Screen Slides”. [Accessed on 12 July 2013]. Available: http://developer.android.com/training/animation/screen-slide.html

(Starts on page odd)

Appendixes

Appendix A - Business Prototype Planning

This appendix is related to the detailed Gantt chart of the business prototype planning, Figure 31. Each task related to the development of this internship is represented as well as the estimated time to complete them.

	Name	Duration	Start	Finish
1	Initial phase	2 days	9/3/12 8:00 AM	9/4/12 5:00 PM
2	State of the art	5 days	9/5/12 8:00 AM	9/11/12 5:00 PM
3	Requirements analysis	1.5 days	9/12/12 8:00 AM	9/13/12 1:00 PM
4	Preparation of development plan	1 day	9/13/12 1:00 PM	9/14/12 1:00 PM
5	Study of web technologies	5 days	9/14/12 1:00 PM	9/21/12 1:00 PM
6	Mockups creation	5 days	9/21/12 1:00 PM	9/28/12 1:00 PM
7	Implement cache system (database)	3 days	9/28/12 1:00 PM	10/3/12 1:00 PM
8	Implement base project	35 days	10/3/12 1:00 PM	11/21/12 1:00 PM
9	Implement authentication	1 day	11/21/12 1:00 PM	11/22/12 1:00 PM
10	Implement sign up	2 days	11/22/12 1:00 PM	11/26/12 1:00 PM
11	Implement user profile	3 days	11/26/12 1:00 PM	11/29/12 1:00 PM
12	Implement contacts	5 days	11/29/12 1:00 PM	12/6/12 1:00 PM
13	Implement discussions	6 days	12/6/12 1:00 PM	12/14/12 1:00 PM
14	Implement send files	3 days	12/14/12 1:00 PM	12/19/12 1:00 PM
15	Implement share location	2 days	12/19/12 1:00 PM	12/21/12 1:00 PM
16	Implement view files content	2 days	12/21/12 1:00 PM	12/25/12 1:00 PM
17	Implement tags	8 days	12/25/12 1:00 PM	1/4/13 1:00 PM
18	Implement drag-and-drop	4 days	1/4/13 1:00 PM	1/10/13 1:00 PM
19	Implement search mechanism	2 days	1/10/13 1:00 PM	1/14/13 1:00 PM
20	Implement application settings	2 days	1/14/13 1:00 PM	1/16/13 1:00 PM
21	Implement to-do lists	3 days	1/16/13 1:00 PM	1/21/13 1:00 PM
22	Implement UI for video and voice calls	4 days	1/21/13 1:00 PM	1/25/13 1:00 PM
23	Implement background service	5 days	1/25/13 1:00 PM	2/1/13 1:00 PM
24	Functional tests	4 days	2/1/13 1:00 PM	2/7/13 1:00 PM
25	Initial structure	10 days	9/3/12 8:00 AM	9/14/12 5:00 PM
26	Introduction and internship description	16 days	9/17/12 8:00 AM	10/8/12 5:00 PM
27	State of the arte	25 days	10/9/12 8:00 AM	11/12/12 5:00 PM
28	Planning and methodology	17 days	11/13/12 8:00 AM	12/5/12 5:00 PM
29	Development chapter	30 days	12/6/12 8:00 AM	1/16/13 5:00 PM
30	Conclusions and appendixes	19 days	1/17/13 8:00 AM	2/12/13 5:00 PM

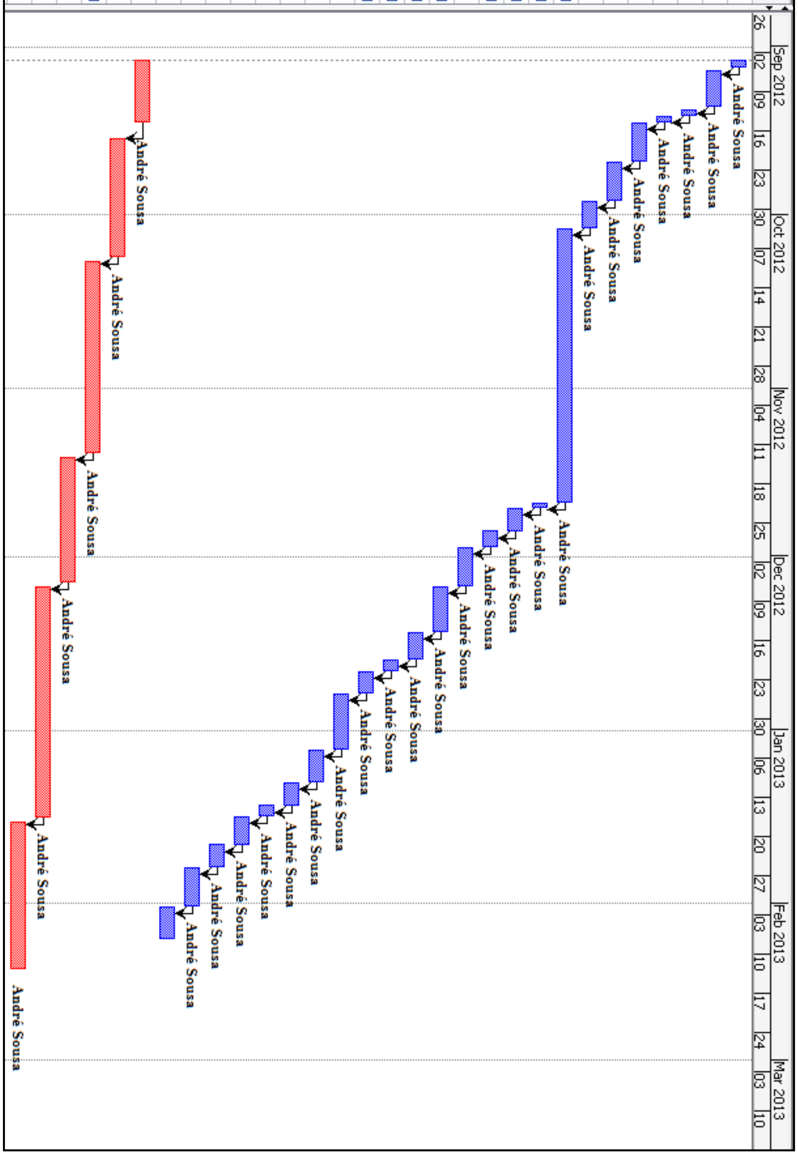
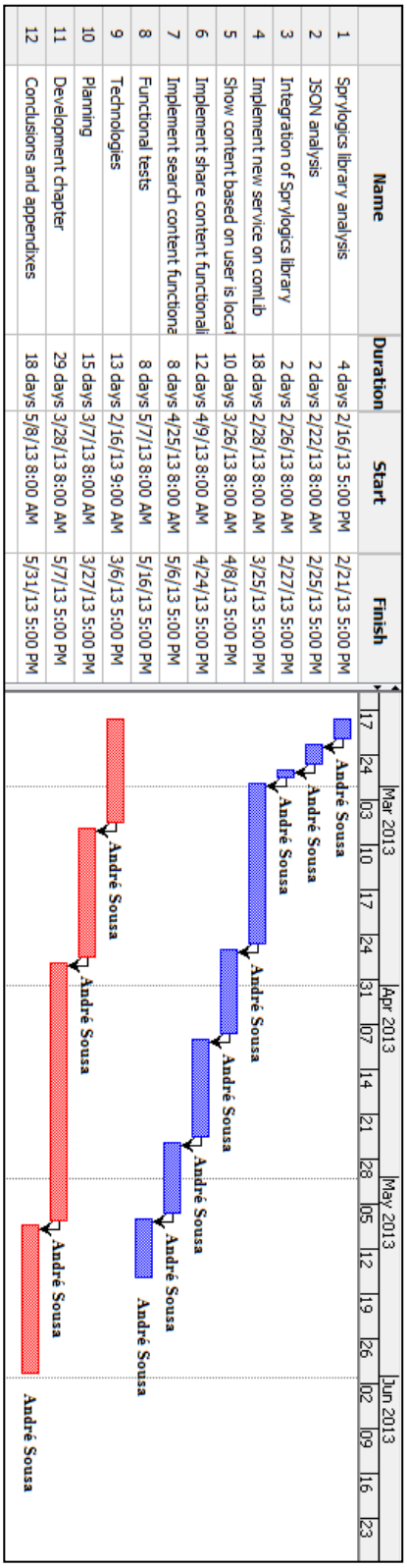


Figure 31 - Specific Gantt chart of the business prototype planning

Appendix B - Share Interests Planning

In this appendix is presented the planning of the share interests through a detailed Gantt chart.



	Name	Duration	Start	Finish
1	Sprylogics library analysis	4 days	2/16/13 5:00 PM	2/21/13 5:00 PM
2	JSON analysis	2 days	2/22/13 8:00 AM	2/25/13 5:00 PM
3	Integration of Sprylogics library	2 days	2/26/13 8:00 AM	2/27/13 5:00 PM
4	Implement new service on comLib	18 days	2/28/13 8:00 AM	3/25/13 5:00 PM
5	Show content based on user is local	10 days	3/26/13 8:00 AM	4/8/13 5:00 PM
6	Implement share content functionali	12 days	4/9/13 8:00 AM	4/24/13 5:00 PM
7	Implement search content functiona	8 days	4/25/13 8:00 AM	5/6/13 5:00 PM
8	Functional tests	8 days	5/7/13 8:00 AM	5/16/13 5:00 PM
9	Technologies	13 days	2/16/13 9:00 AM	3/6/13 5:00 PM
10	Planning	15 days	3/7/13 8:00 AM	3/27/13 5:00 PM
11	Development chapter	29 days	3/28/13 8:00 AM	5/7/13 5:00 PM
12	Conclusions and appendixes	18 days	5/8/13 8:00 AM	5/31/13 5:00 PM

Figure 32 - Specific Gantt chart of the share interests planning

Appendix C - Mockups

This section presents the mockups produced during the process of designing the user interface, giving an overview of how the UI was originally thought.

C.1 User Interface

The next pictures are related to the initial phase of developing an application interface. Each picture is related to a specific screen on the application.

Figure 33 - Business prototype login user interface mockup

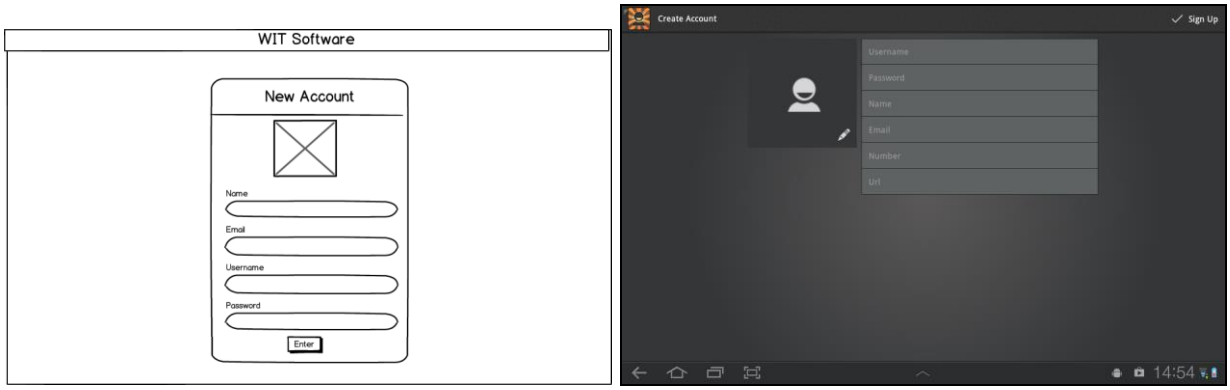


Figure 34 - Business prototype register user interface mockup

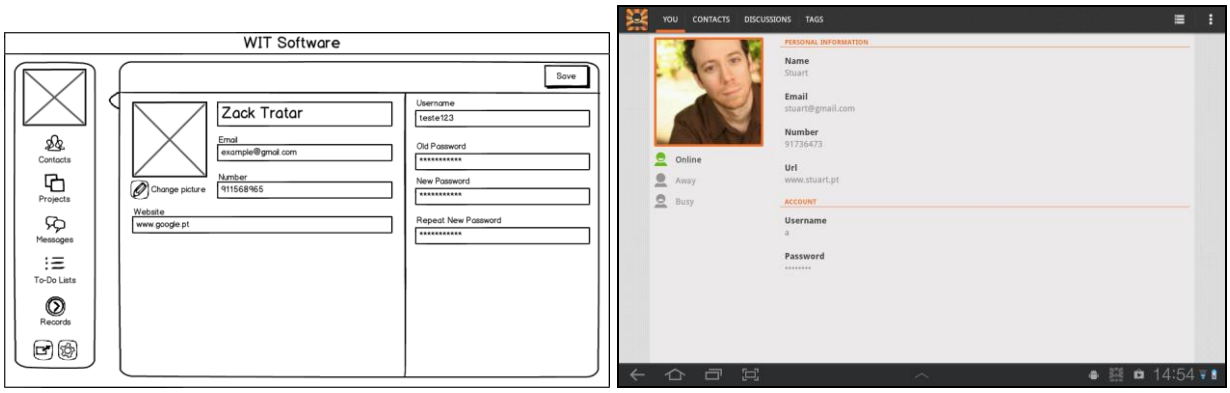


Figure 35 - Business prototype profile user interface mockup

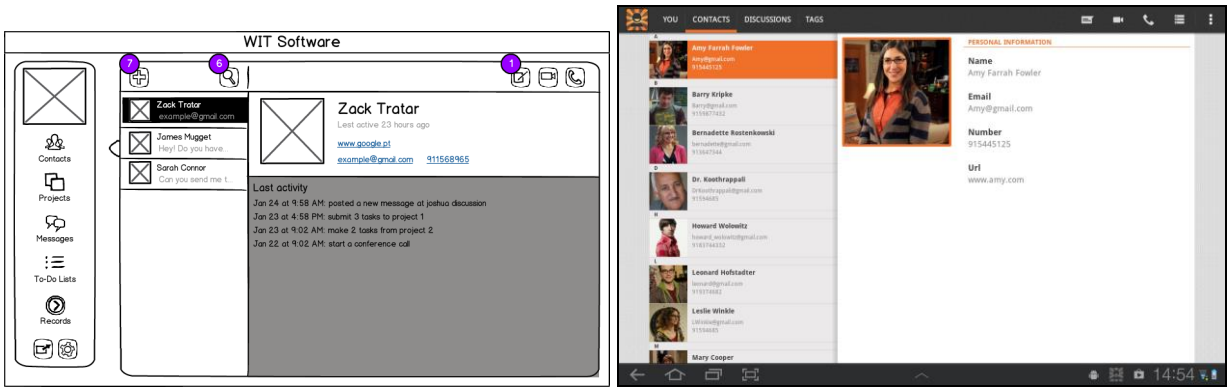
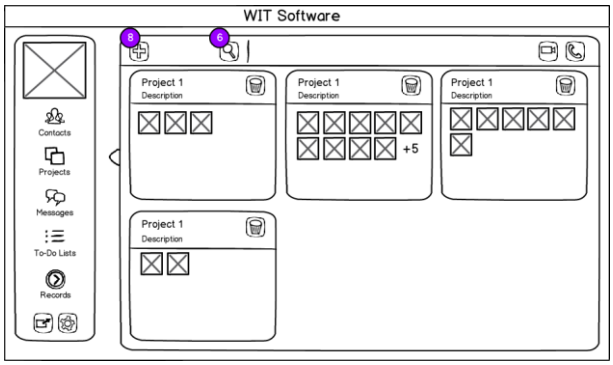
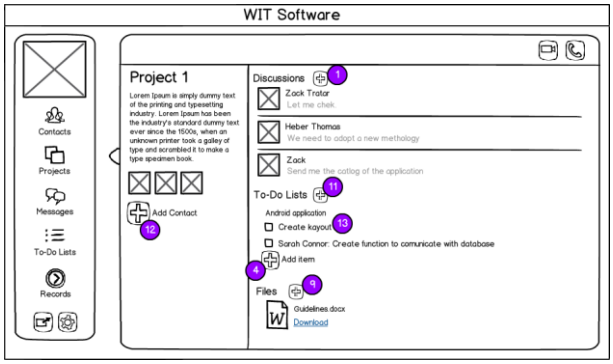


Figure 36 - Business prototype contact list user interface mockup





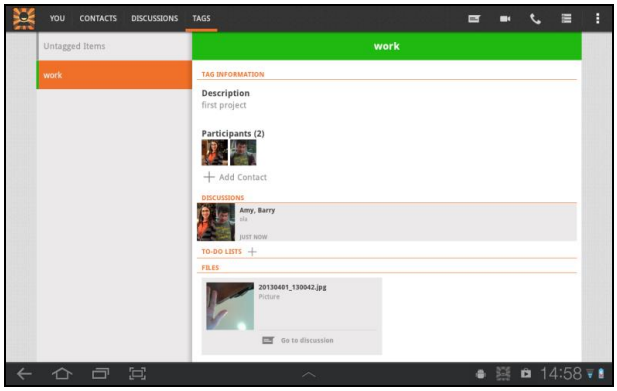


Figure 37 - Business prototype tags user interface mockup

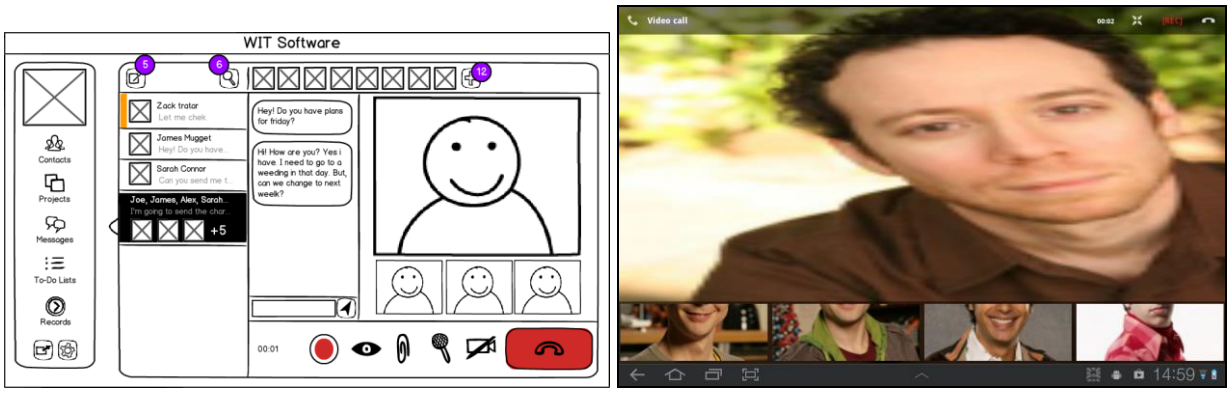


Figure 38 - Business prototype video call user interface mockup

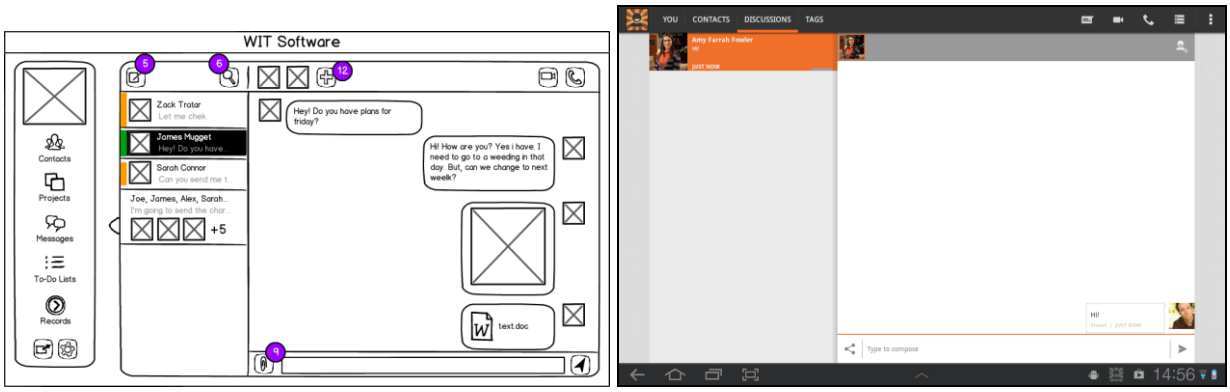


Figure 39 - Business prototype chat user interface mockup

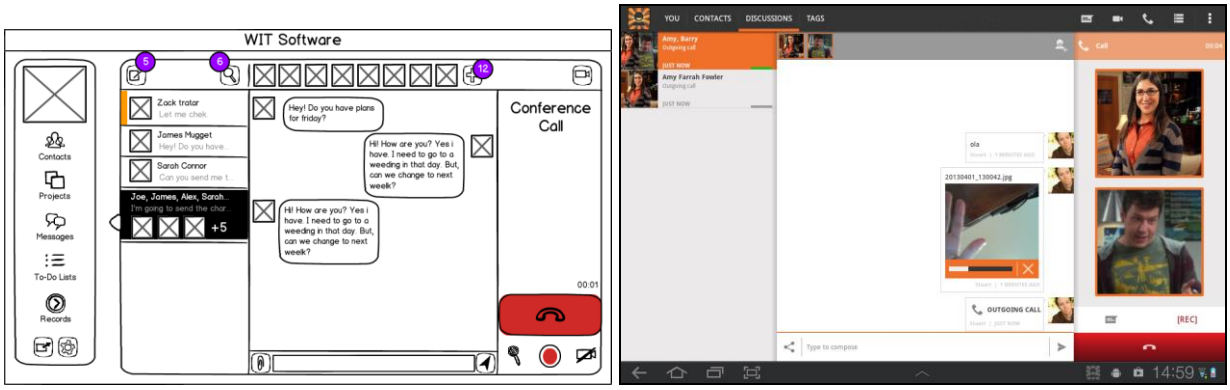


Figure 40 - Business prototype voice call user interface mockup

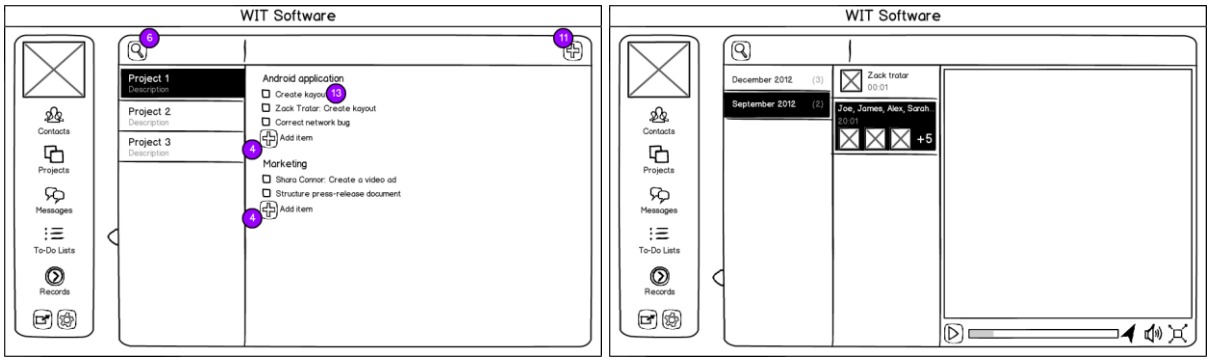
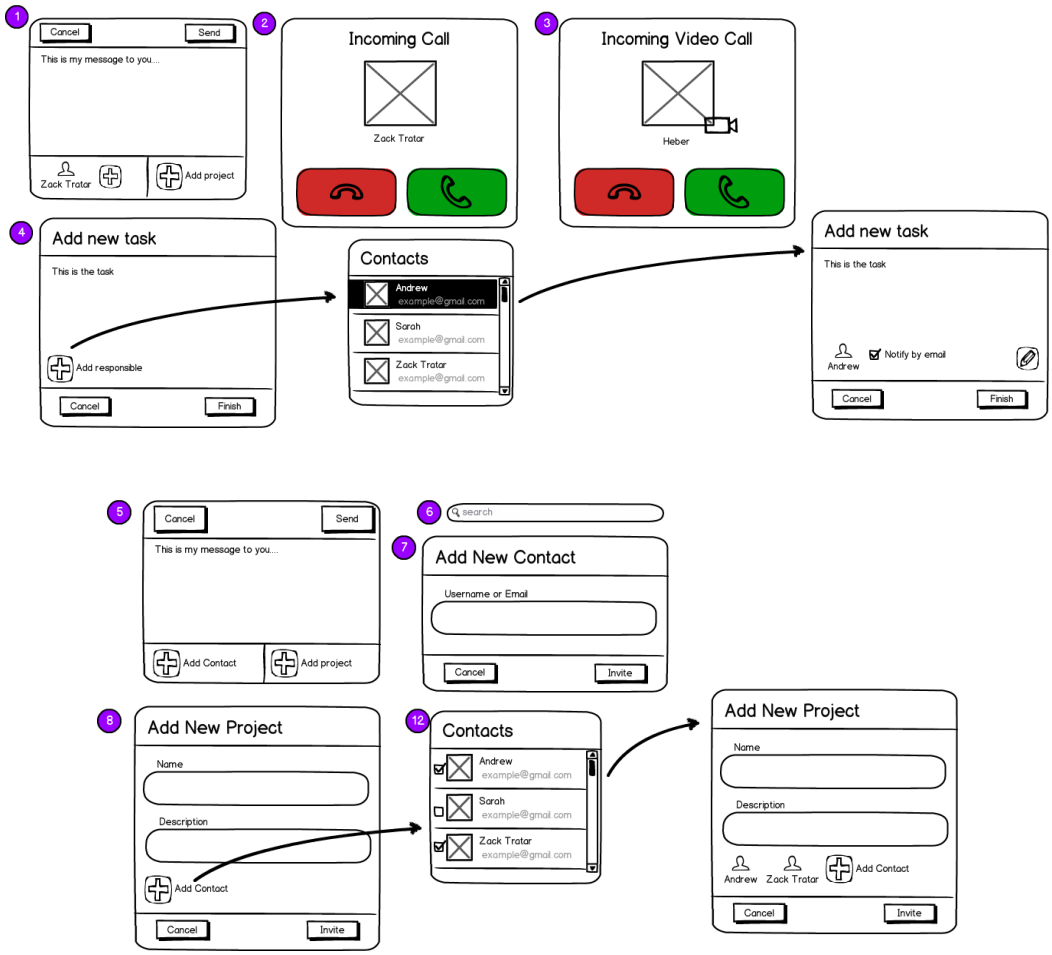


Figure 41 - Business prototype records and to-do lists user interface mockups

C.2 Dialogs

The next few pictures are related to the initial phase of developing an application interface. Each picture is related to a specific screen on the application.



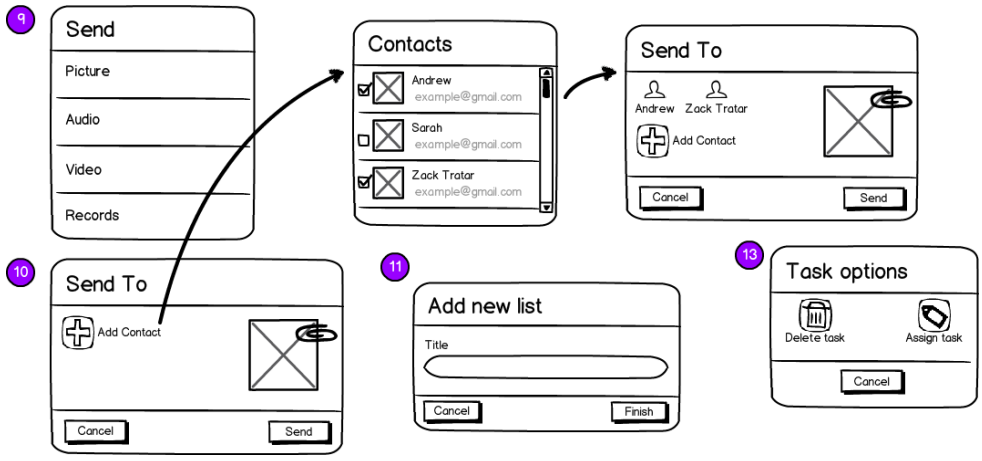


Figure 42 - Dialogs mockups of the business prototype user interface

Appendix D - Performance Tests

This section will address the memory test taken with little eye tool in order to analyze the memory consumption between Sprylogics SDK and Joyn application. The results are showed in Figure 43 and Figure 44.

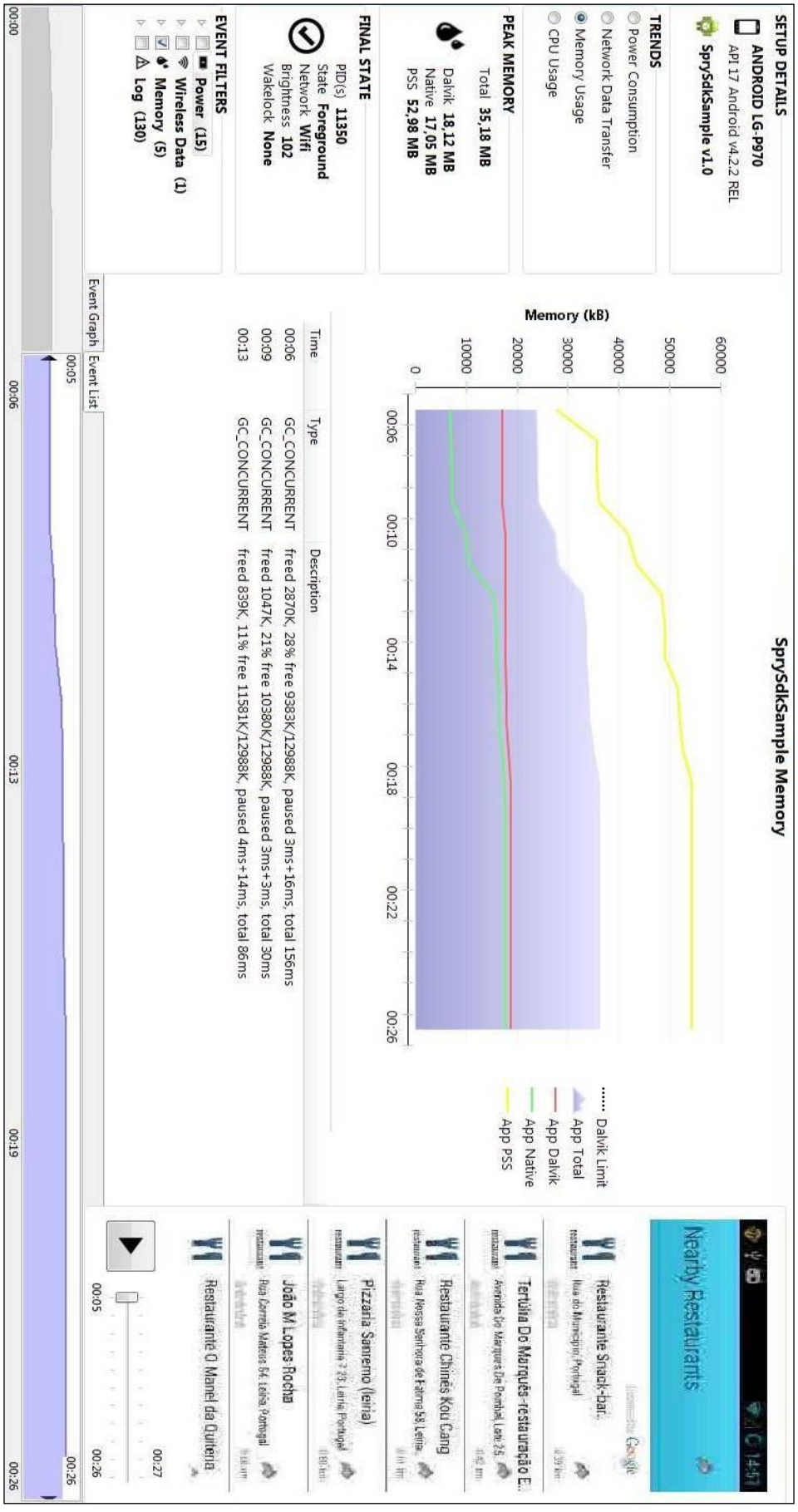


Figure 43 - Sprylogics SDK memory test

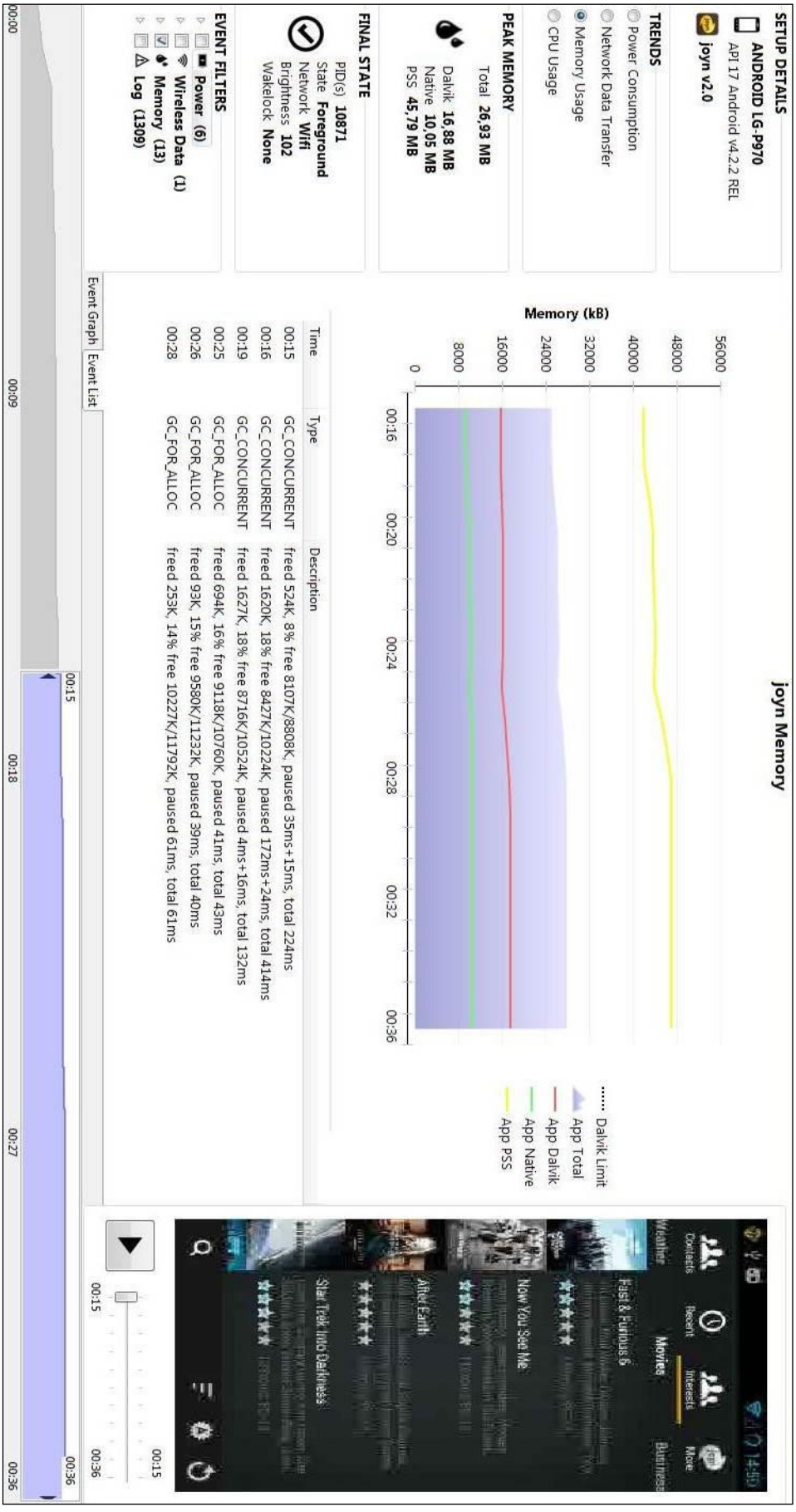


Figure 44 - Joyn memory test