

Systems Engineering Faculty

**PONTIFICIA UNIVERSIDAD
JAVERIANA**

S-3R4

Néstor Diazgranados

15/01/2011



Engineering Faculty (Systems Engineering)

Degree work

Student

Néstor Enrique Diazgranados (ndiazgranados@javeriana.edu.co)

Supervisor

Oscar Xavier Chavarro (ojchavarro@javeriana.edu.co)

Juan Pablo Garzón (jpgarzon@javeriana.edu.co)

Dedication

Engineering is about science, creativeness and technology. Engineering is for those with good spirit, who understand innovation, the entrepreneurship and dedicate his effort to people, because of people is everything. Engineering is for our Latin Americans the balance generator. (Néstor Diazgranados)

Contents

- Abstract** **3**

- 1 Executive Summary** **5**
 - 1.1 General review 5
 - 1.1.1 Business Description 5
 - 1.1.2 What is the problem? 7
 - 1.2 Effort 8
 - 1.2.1 Degree Work 8
 - 1.3 Advices (Technical Aspects) 9
 - 1.4 About Document. 10

- 2 Theoretical Framework** **11**
 - 2.1 Direct Sell 11
 - 2.1.1 What is the key to build your dream? 12
 - 2.2 Supply Chain 12
 - 2.3 Web services 13
 - 2.3.1 SOAP Implementation 13
 - 2.3.2 REST Implementation 14
 - 2.3.3 Android 15
 - 2.3.4 Android Development 16

- 3 The Methodology Steps** **17**
 - 3.1 Overview 17
 - 3.2 The General Steps 17
 - 3.2.1 Elevator Pitch and Work Plan 17
 - 3.2.2 Need Assessment and Use Case 18
 - 3.2.3 System Design and Architecture 19
 - 3.2.4 Wireframes of the mobile phone and Web Services 20
 - 3.2.5 Demo of Web services and mobile phone 21
 - 3.2.6 Demo in the mobile phone. 22

- 4 S-3R4 Application** **25**
 - 4.1 Small historical review. 25
 - 4.1.1 What is S-3R4 application? 26
 - 4.1.2 What is the scope of S-3R4 application for degree work? . . . 27

4.2	The Development Steps of S-3R4 application	28
4.2.1	Elevator Pitch and Work Plan	28
4.2.2	Need Assessment and Use Case	29
4.2.3	System Design and Architecture	33
4.2.4	Architecture Perspective	38
4.2.5	Wireframes of Mobile application.	47
4.2.6	Demo of Web services and mobile phone.	52
4.3	Future work (Technical Aspects)	56
4.3.1	Database synchronization.	56
4.3.2	Personalization and Adaptation Models.	57
4.3.3	Arithmetic Model (distribution of money).	57
4.3.4	Distributed Model.	57
	Acknowledgments	63
	Bibliography	65

List of Figures

1.1	Work distribution	8
1.2	Work activities	9
2.1	Supply Chain Framework [1]	13
3.1	Elevator Pitch and Work Plan	18
3.2	Need Assessment and Use Case	19
3.3	System Design and Architecture	20
4.1	S-3R4	25
4.2	The Big Picture	26
4.3	S-3R4 Use Case	31
4.4	Requirements Template	33
4.5	Domain Model	34
4.6	Business Objects	35
4.7	People and State Objects	36
4.8	Preference Objects	37
4.9	Multiple inheritance and generic functionality.	39
4.10	Multiple inheritance and generic functionality.	40
4.11	Multiple inheritance and generic functionality.	41
4.12	Server View	42
4.13	Server Model	43
4.14	Component or Development Diagram.	44
4.15	Physical or Deployment Diagram	45
4.16	Organization of Android Code	46
4.17	Organization of Server Code	47
4.18	Navigation Controller	48
4.19	Dashboard (Left image) and Action Bar (Top Screen)	50
4.20	Business Rules and Consult	51
4.21	Technology	52
4.22	REST Web Service	54
4.23	The EJB method	55
4.24	The root class	56

Abstract

Welcome to Smart-phone ERA application and Methodology for Android Development; A methodology that covers Android application development cycle with Java Technology. When I was started to do my degree work, I was enthusiastic about the opportunity to spread the wealth of knowledge that I'd picked up over the past year of Android development. I hope you enjoy finding out about how to do a complete software cycle with Android platform and Java technology from this document as much as I enjoyed writing it! When Android was acquired by Google in 2005 (yes, Android was a start-up company at one point), I'll be honest, I didn't have much idea of it. I heard that Google might be entering the mobile space, I didn't see the great chance to work in this area until I had my opportunity in Colombia with [2]NextLab program (MIT-Mobile Development Research). Fast-forward to a few months later, I was giving seminars to companies like (Spring Wireless-Multinational about Android mobile Development) and learning about JEE6 (Java enterprise Edition). The platform is barely six years old, and I see no sign of the platform development slowing down. Without doubt, this is an exciting time in Android development, the current version is 4.0 (Ice Cream Sandwich). Methodology is working with Android 2.3 but even if you are excited to work with Ice Cream Sandwich, this document is going to be really useful.

1 Executive Summary

1.1 General review

1.1.1 Business Description

"Life is public so life is business". This phrase of Jeff Jarvis particularly describes how the idea of S-3R4 application was inspired, reminding me the idea of an open world for open minds because of the fact life rules are changing. Internet is the magnificent creator of a brand new possibilities where networks are becoming the new markets. S-3R4 application is intended to be an extended solution of a great idea created with NextLab program (MIT-Mobile Development Research) and Pontificia Universidad Javeriana PUJ) in 2011, but trying to explore part of that new rules that technology imposes nowadays in order to create successful and profitable ideas[3].

All started as a business idea, NextLab an PUJ worked together with biggest local industry company's. I worked with Mobile CLYK team and our goal was to developed a mobile application capable to manage orders over a multilevel structure taking advantage of online information (products inventory). Market is really interested in this kind of business model instead of the common manufacturing and independent business models, because of it reduce costs, doesn't require physic infrastructure and provides a surprising growth mechanism[2].

Android is a great opportunity to explore the business segment, typical Android applications are covering topics as tools, games and entertainment, however, business sector is so poor yet. Due to some android shortcomings in security and aversion incompatibility people find an important piece of evidence that explain why doesn't exist a considerable number applications in entrepreneurial sector. In the other hand is possible to enrich common processes of industry with technology provided by mobile devices, moreover inside Android world, developers can exploit most of the mobile market. There are more than 200 million activated Android devices, and around the world Android has the bigger percentage, resulting as number one at least until now[4].

Mobile CLYK was the responsible team to generate a first prototype, a private prototype that was able to make orders and to manage online inventory, but multilevel model could not be developed by the time limitation . S-3R4 is a free application, is a platform, a complete social network and is available for each individual person. S-3R4 apply new rules[3]:

1. New relationship: This is not for a private market, this is a tool for each person who needs a easy, cool and fun way to offer knowledge, experience and concretely products or services inside of a social circle. Mobile CLYK was a good idea, but Why people was going to use application?. [3]“Give control to the people and then you will use it”, so before charging, S-3R4 needs to create a great tool, that’s why S-3R4 is a free mobile application focusing in provide the best and easy way to offer products and services. S-3R4 is not only for big company’s, indeed, is the chance for everyone to offer valuable product or services to others (social circle), because of life is about work and S-3R4 thoughts the best environment where there is a good potential market is the social circle around you, exactly the people know you or is closer to you.
2. New architecture: Be a platform Jeff says!. This is not about a mobile application, this is about a complete new platform that could be capable to grow, functioning with distributed models, taking benefit from new technology and exploring a deeply and challenge vision, so consequently offering a entirely new product[3].
3. New Methodology: To achieve good results experience becomes the better master to transform theory in good results. The proposal methodology used here is based in this philosophy. Methodology is not establishing rules, is writing good practices and parameters to guide begginers in Android development with JAVA technology.
4. New attitude: “Trust in people, learn about them is imperative”. Growing together the success path is going to be opened. Innovation is the essential element for new business models, unless you create something beautiful to people, they wouldn’t like to work with you. Remember that your worst enemy is your coworker, the most important new rule. That’s why S-3R4 is a free application, people criticisms and improvements are welcome, is about to work together, and to build in several software mobile cycles a wonderful mobile application[3] .

This document is not intended to describe the business plan of S-3R4 application that is now in progress as a E-3R4 application. In contrast the technical details are the core, the pure hearth of this document. Detail doesn’t mean to think about granularity perspective, instead, is more about high design, and to understand the big picture between smart devices (Android) and servers (JAVA). The north of S-3R4 application was introduced to push others to do the same before start a project, because of everything needs a reason to exist, everything must attack a real problem and to push innovation. People will always be the core of your business, they are your coworkers and clients. S-3R4 methodology and application will be described in next sections, with a complete software cycle for mobile computing using Android and Java technology.

1.1.2 What is the problem?

Mobile supply chain management is increasingly recognized as an area offering significant potential for generating improvements and creating competitive advantage to corporations[5]. Obviously, high efficiency mobile chain network needs simultaneous and accurate information about demand, supply, sale, inventory, shipment and operational activities. S-3R4 is obeying this postulate, and precisely improving the different flows of supply chain. Nowadays supply chain management, organizations need to address complex interdependencies and create an extended enterprise that looks beyond its manufacturing plants, and at the same time customers increasingly expect companies, suppliers and distributors to provide real time information about their orders[6]. These escalating customer demands are forcing corporations to adopt mobile technology to optimize and streamline the information, product and financial flows in the supply chain. But supply chain processes in companies today are desktop-centric.

Today, the effectiveness of an organization's response to rapidly changing market conditions is vital. Members within the supply chains should "seamlessly" work together to serve the end consumer. With mobile web service applications in supply chains, short lead-time and optimization in supply chains can be fulfilled.

1.1.2.1 Information flow

Online inventory, transactions of money, catalog of products, prices, promotions, campaigns and much more just in a few clicks, touch gestures or even saying to your mobile device what you need. All is immediately in your hands, everything is open, is connected. Web services enable a flexible environment to reproduce this kind of activities, benefiting application and data integration, versatility, code re-use and cost savings [6, 5].

1.1.2.2 Financial Flow.

Could you imagine imagine avoiding long lines in supermarket. No more bother transaction stuff, your device and NFC technology are changing the way you and your wallet coexist. This is not a company benefit, this is S-3R4 revolution for each person to do business, to pay orders, to control your finance world and to ask or sell a service to someone in your social circle. Mobile devices are gaining wide spread usage in business and financial applications, and there is incredible information out there as frameworks to control dynamic changes, mobile social networks and green computing environments for managing the needs of their complex business processes and systems, authentication schemes improving convenience and usability, mobile banking as a cornerstone of mobile commerce and even vehicular mobile commerce is being explored to work with location-based services[7, 8, 9, 10, 11, 12].

1.1.2.3 Material Flows.

Technology-based solution could help optimize distribution routes and reduce inefficiency. By knowing the location and details of transactions, the company can better direct marketing strategies and manage human and material resources. Complete control about your resources, access to cool and useful statistics, observing behavior, managing assets and much more. This is creating a whole new innovation possibilities in material flows[13, 14].

1.2 Effort

1.2.1 Degree Work

The minimum working hours that credits system (PUJ) establish for degree work are 192 hours. The daily work, (attached document) shows that actually work takes 410 hours, equivalent to 213.5% of effort. It has been a really incredible time, obviously with difficulties but at the end a great satisfaction for excellent application performance in use cases and using web services . Here will be described some interesting statistics about distribution of these 410 hours.

Android work takes 185 hours, documents take around 56 working hours without taking January (2012) work. The server side takes 130 hours and 39 hours were about additional work as installation, configuration and use additional tools as regular expressions.

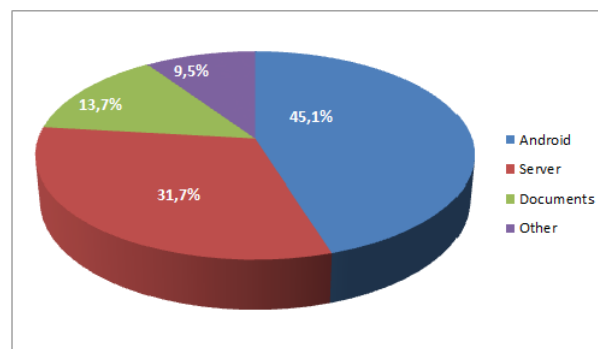


Figure 1.1: Work distribution

S-3R4 Methodology is using approximately 77% for development and 23% for documents support. I think that is important to increase that percentage at least to 30%. The January work was traduced in documents, being important to dedicate time to each diagram for guarantee future work, quality and continuous progress. In the other hand the numbers are correct, if knowing that S-3R4 is based in a first

1.3 Advices (Technical Aspects)

experience (Mobile CLYK) that clarified several aspects. S-3R4 highlight the main activities of this process, showing developing and designing tasks as the core of its development. Also the previous work (Mobile CLYK) help to save a lot of hours about support documents.

The work was classified in four colors:

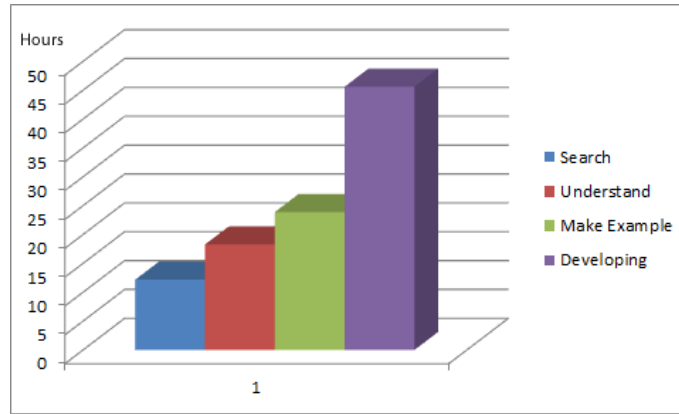


Figure 1.2: Work activities

The 410 hours were classified in different activities. The image shows the corresponding percentage for every color, indicating the hours percentage over the total hours. Search activity consumed 50 hours, understanding concepts consumed 75 hours, examples consumed 98 hours and development activities consumed 187 hours. Observe that less than 45% was to write code, so the key is more about to comprehend and to understand and not about to create multiple code lines.

The examples consume a lot of time, an evidence of why the schedule was delayed. However examples are good for understand and generate quality code, facilitating code refactoring, delegating coherent responsibilities and improving task design. In other words every step is more efficient if you take a time to search and understand. Search is important when concepts and examples are done, they convert to itself into a powerful tool. Is a fundamental key: “First comprehend concepts, then guess the kind of solutions are needed and for last create the correct solution”. So understanding different approaches, identifying good and bad details, give you a base reference or a judgment to modify or create new solutions.

1.3 Advices (Technical Aspects)

1. The first advice as a conclusion from S-3R4 application effort is to design and establish time to update and document every requirement, diagram and support document periodically. Understanding behavior of real prototypes is

possible to clarify designs, to correct and elaborate the right solution and to keep updated communication between group.

2. S-3R4 application delays were related with server issues, it means about 20 hours or approximately 5% of total work, suggesting that possible risks need to be treated carefully. I think that the software cycle during a semester is perfect time to deal with this kind of possible emerging problems. In fact issues were related about new work but not about defined elements. Requirements changes were minimum, design was beautiful because of experience with Android and complex tasks as voice recognition were widely explored. To define a reasonable scope project, experience plays the main role, experience remember that to work in familiar environments control excessive risks, moreover power other elements as intuition and investigation, result in more productive effort.
3. Use cases and requirements are the core of your application. In mobile development is important to differentiate your model domain, your business methods, and to make easier maintainability and scalability. The package organization, tiers and architecture work proposed in next sections use an approach to achieve it.
4. Examples are not about just pure functionality. The patience and creativity are good tools to try different approaches. Explore alternatives, trust in your intuition, take risks and then examples will be a great tool in development process. As an excellent example, android functionality tier was designed through different kind of tests and brain storm ideas. Keep your head working, you will enjoy next steps if you use your time wisely, so analyze before to put your hands in the matter.
5. Database model need to be defined carefully. Create your query's, explore alternatives, because web services take most of the time. Database model defines complexity when working in business model, so to define the more complex query's, try to balance relations and use smart foreign keys (strong and weak relations) to reach simplicity in design.

1.4 About Document.

Document describes all the technical aspects about S-3R4 development. Next section expose the methodology steps that were designed, later readers will find how each step was developed by S-3R4 application. All is about Android and Java technology, and the purpose with all this is to establish a good way to initiate android applications that consume web services over a scalable and good performance platform. The server side is part of the development process, including web services and Architecture. enjoy it!

2 Theoretical Framework

2.1 Direct Sell

Today, more than ever before, people are considering the possibility of starting their own business[15][16]. The promise of time freedom, prestige, independence, and fulfilling work pulls seductively at our heartstrings as an alternative to being unfulfilled and underpaid. But the truth is, starting a traditional small business is backbreaking work, requiring long days that stretch late into the night, never-ending amounts of capital, and strong mental stamina to juggle the many roles and responsibilities of today's entrepreneur. Is this any way to live?.

Enter the direct-selling business—small, lean, and incredibly efficient. As a direct-selling professional, you can choose from hundreds of fine companies that sell every product or service imaginable—from women's apparel to state-of-the-art nutritional products. In exchange for a minimal investment (most under \$100), the company agrees to research, design, test, manufacture, store, ship, and, in many cases, service its product—all at its expense. You, in turn, agree to become the company's "marketing department," acting independently to share the products and find new customers, who then become your clients. For every sale that you make, you are paid a commission ranging from 20 to 50 percent.

As a direct-selling professional, you have the opportunity to work part-time or full-time and can expand your business enterprise by engaging and training other people, who also love the products, share their enthusiasm for the business, and gain new customers. For this you are paid additional bonuses. But it doesn't stop there. Your company has a stake in your success and goes to great lengths to inspire you with incentives that range from free products to all-expense-paid trips to exotic destinations. Companies understand that their success depends on your success and provide ongoing support in the form of training material and live events to inspire you to continually grow and learn. Unlike traditional entrepreneurs who usually fly solo, you are never alone in your direct-selling business. You always have someone to support you along the way. And last but not least, you'll form lifelong friendships with some of the most incredible people you've ever met. So if you are one of the hundreds of thousands of individuals who is thinking about starting a small business, be sure to look before you leap into the life of a traditional entrepreneur. You might be surprised to learn that the trappings of a conventional business limit your freedom instead of enhancing it. So why not consider starting a direct-selling

business of your own? Perhaps like thousands of others before you'll discover more rewards than you ever thought possible.

2.1.1 What is the key to build your dream?

There are some variables in the field, but technology is essential. You can connect with customers and team members at the speed of light, improve your efficiency, and expand your market to reach beyond the borders of your state and even your country. Technology gives you a chance to:

1. Build Strong Teams by Phone and Web (Platform)
2. Build Relationships on the Internet (Open world for open minds)
3. Train with Technology (Why not mobile?)
4. Boost Your Online Image (GUI design over a Platform)
5. Become a Master Networker on the Internet (How to do the common activities through technology? for example in a smartphone.)
6. Make Friends with Technology (For Business)

2.2 Supply Chain

The linkage between communication technology, the supply chain environment and end-consumer needs assists in better communication between all activities across supply chain tiers. Successful supply chain management depends on tight coordination between supply chain partners. The Internet provides the perfect communications channel for the information, decisions, transactions, and processes that form supply chain management. It is a journey of change that is already under way. To keep up with the evolution of supply chain management, and remain competitive in their markets, future supply chain leaders will need to simultaneously exhibit tight coordination in all areas: integration, collaboration, and synchronization with a better understanding of communications systems function. Integration is the coordination of supply chain activities such as purchasing, manufacturing, distribution, spares management needed for maintenance and operational availability of military systems, and customer service within a single enterprise of technological readiness.

Supply chain manage tree main flows (INFORMATION, FINACIAL, MATERIAL)[1]. Integrated to a communication system Ralph L. Harper propose this framework:

Supply Chain Framework

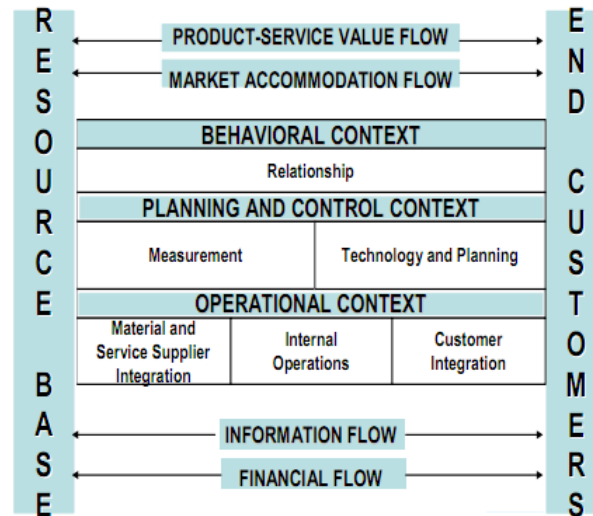


Figure 2.1: Supply Chain Framework [1]

2.3 Web services

Web services technologies are undergoing some significant changes at several levels. The first fundamental change is the diversification of architecture styles from SOAP to REST based services. This change is witnessed by the recently rectified WSDL 2.0 at W3C that supports both SOAP and REST bindings. The second significant change is the diversification of message formats. There are message formats besides SOAP and even XML, such as JavaScript Object Notation (JSON), RDF N-Triples and Google Protocol Buffer, to name a few.

2.3.1 SOAP Implementation

At a fundamental level, a SOAP-based architecture revolves around the transmission of XML-encoded messages over HTTP. Specific SOAP service sets are defined in web service definition language (WSDL) files which are essentially XML files adhering to a W3C-specified grammar. The WSDL file for the Portal service set references (or defines internally) a series of XML schema types (or XSD types) that mirror the server-side data model. These types maps out the structure of parameters that may be included in service requests / responses and, furthermore, may even be used to generate language-specific bindings for various client and server platforms. Specifically in Portal's case, these schema types define how application profiles,

device profiles, user profiles, and user accounts are constructed; so that they may be used as parameters in requests for the creation, retrieval, modification, or deletion of such objects in the back-end data model. Additionally, types representing the formats of service requests and responses are defined here. All of these XSD types are referenced within the WSDL file to define the SOAP interface for a particular service set. First, this file defines a collection of multi-part messages and maps them to individual XSD types. For the Portal SOAP implementation, each message is composed of an instance of one type, which acts as the container for a specific service request or response. Then, the WSDL file defines a so-called port which is later mapped to the overarching Portal SOAP service set. The port itself is composed of multiple operations, each one representing a single service to be implemented in the set. In turn, each operation maps previously-defined SOAP messages as its input and output types. Finally, the WSDL file specifies a URL where servers that implement this particular service may be located by interested parties. The content of this WSDL file represents a language- and platform-neutral method of remotely communicating the SOAP service interface. Once the WSDL file for the SOAP implementation of the service-oriented architecture is written, it needs to be made publicly available to all Portal clients capable of reaching a particular Portal server. This is done through the use of a SOAP-enabled HTTP web server; and may be enabled in a variety of different ways. Regardless of how the HTTP server to be used is configured, server-side code needs to be written that is responsible for handling incoming service requests and formulating appropriate responses. However, the actual execution of these requests are accomplished by these server-side hooks invoking appropriate methods within the data model's DAO. By separating these responsibilities we permit the data transmission component to focus solely on the expedient transportation of request / response messages while allowing the data model component to sift through their contents and handle them accordingly [17].

2.3.2 REST Implementation

The REST implementation differs from its SOAP counterpart in very fundamental ways. While SOAP adheres very closely to the RPC model, REST revolves around the concept of resources and focuses on using the inherent power of HTTP to retrieve representations of these resources in varying states. In the REST style, every resource is signified by a unique URL which may be operated on by a subset of the core set of HTTP commands: Get, Post, Put, and Delete. However, instead of a specialized SOAP client, any HTTP client library may be used to interact with the Portal REST server. This is important to note, as there are only a handful of well-supported SOAP client libraries out there, but practically every contemporary language comes equipped with a built-in HTTP library as it is a universal protocol for communicating over the Internet. Furthermore, there is no need for service discovery and interpretation on the client's part, merely the IP address or network name for the Portal server it wishes to contact. Once the REST client is pointed at

a particular server, it sends an HTTP command to a specific URL in that server's domain which relates the exact resource which should be operated on[17].

2.3.3 Android

Android, the world's most popular mobile platform powers millions of phones, tablets, and other devices and brings the power of Google and the web into your hands. With an amazingly fast browser, cloud sync, multi-tasking, easy ways to connect & share and the latest Google apps (and thousands of other apps available on Android Market) your Android powered device is beyond smart.

The Android operating system is released under the open source Apache License and is built on Linux kernel version 2.6. Android is a project of the Open Handset Alliance (OHA). Founded by Google, OHA is an association that includes 65 hardware/software companies and operators, such as KDDI, NTT DoCoMo, Sprint Nextel, Telefónica, Dell, HTC, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, and Nvidia[4].

The first Android phone, T-Mobile G1 (also marketed as HTC Dream), was released in October 2008, followed by the release of 12 additional android phones in 2009. There are now dozens of Android mobile devices, including both phones and tablets. In addition to the natural fragmentation of screen size, capabilities, and OS version, developers saw incompatibilities between devices that require specific workarounds for both native applications and browser-based applications.

The Android mobile operating system has a rich set of features. 2D and 3D graphics are supported, based on OpenGL ES 2.0 specifications, and there is good media support for common audio, video, and image formats. Animated transitions and high-resolution, colorful graphics are integrated in the operating system and commonly seen in applications. The Android operating system supports multi-touch input (although it is not supported in every Android device). The web browser is based on the powerful WebKit engine and includes Chrome's V8 JavaScript runtime[18].

Multitasking of applications is supported. In Android, multitasking is managed by structuring applications as "activities." Activities have a distinct visual presentation and should be single-purpose, such as taking a photo, searching and presenting results, or editing a contact. Activities may be accessed by other applications as well. A simple application may implement a single activity, but more complex applications may be implemented as a number of activities cohesively presented as a single application[18].

Android lacks authoritative human interface guidelines, except for fairly narrow icon, widget, and menu design guidelines and broad advice about structuring activities. This lack of standards can make it more challenging to design and develop for Android; however, Android does include a set of common user interface components that are comparable to those available on the iPhone.

2.3.4 Android Development

To develop for the Android, you can use Windows, Linux, or Mac. Android applications are typically written in Java, but there is no Java Virtual Machine on the platform; instead, Java classes are recompiled in to Dalvik bytecode and run on a Dalvik virtual machine. Dalvik was specially designed for Android to reduce battery consumption and work well with the limited memory and CPU-power of a mobile phone. (Note that Android does not support J2ME.) Since the release of the Android NDK (Native Development Kit) in June 2009, developers may also create native libraries in C and C++ to reuse existing code or gain performance.

The most commonly used and recommended editor is Eclipse with the Android Development Tools plug-in. The plug-in provides a full-featured development environment that is integrated with the emulator. It provides debugging capabilities and lets you easily install multiple versions of the Android platform. If you don't want to use Eclipse, there are command-line tools to create a skeleton app, emulator, debugger, and bridge to an actual device.

3 The Methodology Steps

3.1 Overview

S-3R4 offers a complete methodology, from requirements elicitation to final prototypes in Android. The S-3R4 structure and Java technology power are great tools for easy development, to fully exploit its potential, S-3R4 offers guidelines for readers and pushes innovation in cycle life for mobile technology.

3.2 The General Steps

3.2.1 Elevator Pitch and Work Plan

3.2.1.1 Goal:

Showcase the project and team to supervisors with a one-minute elevator pitch and draft work plan. If you are working without supervision work with your team.

3.2.1.2 Methodology:

Define your Solution: Knowing what you are proposing is easier to explain others. So the first step is to define the correct solution (service/app/device/platform). Brain Storm is an excellent activity to define possible alternatives, to find real innovation and to be creative for offering your services.

Define Roles: Once you have defined a route and some alternatives, it's time to define roles for the team members and to give to each one specific responsibility. It's important to define communication channels, to confirm at least one week face meeting, to talk about strengths and weaknesses and to consider potential risks according to that information. In my experience I will say that work alone is exhausting, and probably you will need to find a good team but it's going to increment your performance and your goals are going to go deeper of what you were expecting.

Define a scope: Remember that you need to define a scope; it could be easier after knowing your team skills and taking into account the limitations as the time. Keep your challenge attitude, and push your team to take some risk, but first define reasonable goals and enrich those with challenging ideas.

Schedule: The last step is to create a schedule, explaining the activities and tasks to achieve the proposal across the milestones established by the program. So what you should define is:

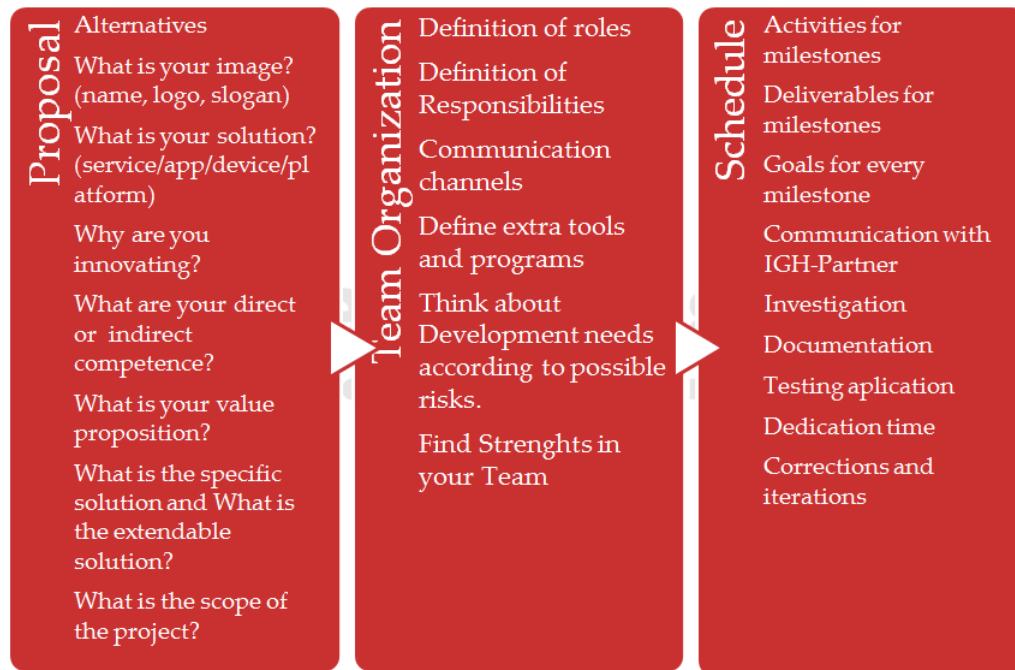


Figure 3.1: Elevator Pitch and Work Plan

3.2.2 Need Assessment and Use Case

3.2.2.1 Goal:

Define the scope of your project team's problem and possible solutions by interviewing stakeholders, conducting surveys and brainstorming.

3.2.2.2 Methodology:

Find Stakeholders; The milestone two is realized to analyze the problem that you are facing, to understand the stakeholders that are inside of it, and to go beyond the specific needs to determinate the correct solution. To achieve it, define the relation

between stakeholders, analyze interaction between them and translate it into use cases.

Determine the expandable solution: Remember that the purpose of your application is about expandable solutions, mobile world requires it. Beyond the specific solution, think about your proposal value and how you could expand your solution to satisfy a market niche.

How to do it? Once the solution is selected, is time to use creativity to push innovation, and define what kind of technology and services are suitable and proper. To make available those services is necessary to define the interfaces between stakeholders and your services. So what you should define is:

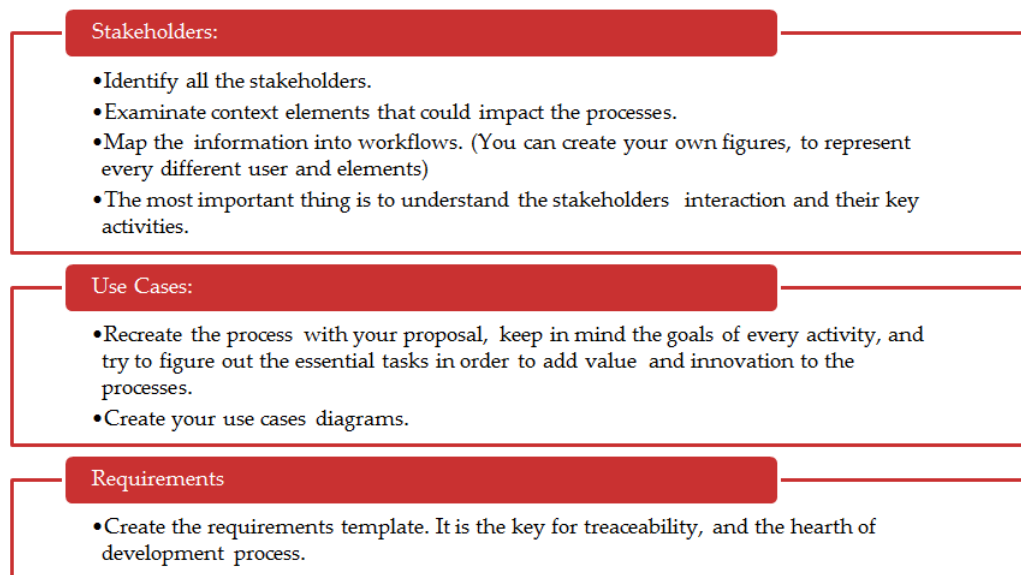


Figure 3.2: Need Assessment and Use Case

3.2.3 System Design and Architecture

3.2.3.1 Goal:

Design the system architecture that clearly shows the stakeholders, the services and the interfaces among them.

3.2.3.2 Methodology:

Architecture elements: The software architecture defines the components and the interaction between them. Also the architecture reflects the non-functional requirements where patterns impact on it. Every component is composed by several modules; each module defines certain functionality, so the components should join a group of functionalities that obey some important topic into the system.

Perspectives: The other architecture element is the horizontal and vertical perspectives. In the horizontal perspective the architecture defines the tiers that separate logically the system levels. The vertical perspective takes into account all the systems interconnected.

Service Flows: When structure is defined, the next step is to define services flow; it is possible through sequence diagrams or activity diagrams. Here should be visible the functions, its description, inputs and outputs. So what you should define is:



Figure 3.3: System Design and Architecture

3.2.4 Wireframes of the mobile phone and Web Services

3.2.4.1 Goal:

Show the wireframes of the graphic user interface of the mobile phone and test web services.

3.2.4.2 Methodology:

GUI: Every UI deals with big challenges, because of the existence of different users having several flavors, preferences and expectations. But the art of give most

users a way to supply their needs and make them comfortable are possible through the analysis of certain patterns.

Building and application Theme: There are two important aspects here. The first one is about multiple devices, and the second one is about providing a suitable environment to attract the user attention. So firstly is good to remember that the application should be functional for different devices. Nowadays there are multiple smart-phones with different dimensions and features. But also is possible to think about other devices as tablets, or computers. The second thing is inspired by culture and philosophy of organization. Is a recreation of a suitable environment to motivate and facilitate users to realize the actions you dispose into the application. This environment precisely should seem similar and compact into the congregate devices for which is going to work.

Organizing content: The idea here is to focus on coherent workflows into the application; because of necessity of users, for understand what they can do into the application. So the innovation here is to define some kind of structure and philosophy for realize common tasks in the application. The users learn faster and they appreciate simple and instinctive ways to reach their goals and find specific actions into the applications. This is a key step that impacts the common phrase “to be fast and fun”. Content means something big nowadays. The sequence of screens and the number of clicks or widget actions should be defined carefully. Users need more than simplicity, they are fast, and they need full access to main functionality when navigating by the application.

Making it look good: Themes are the root of this decision. Here readers deal with design of icons, backgrounds, fonts, colors, images, multimedia and other application resources. But all this stuff should be designed before implementation, thinking possible screen templates for similar actions into the application, supporting simplicity and intuitive actions, and focusing in the achievement of produce the correct sensations, motivations and expectations on users.

Services: Working efficiently, the proposal is to build the functions that enable services of application while developing wireframes. Web services are the perfect way to achieve this goal, because of when you define inputs, processes and outputs, is easier to define templates and good patterns for the application.

3.2.5 Demo of Web services and mobile phone

3.2.5.1 Goal:

Test developed functions with Web services.

3.2.5.2 Methodology:

Services: The readers should work in final prototype working with all corresponding services. All services need to be defined at this step, although could be possible some functions are not implemented in Android phone.

Testing: In development cycle is vital to prove your functions. There are several methods and tools provide for that matter. The main goal for this milestone is to prove that your functions are being effective and are doing what you proposed. Simulate real situations with considerable amount of data knowing values of the correct result is a good practice; also this activity is good for feedback about effectiveness and efficiency of your services.

Accomplishing Use cases: The essence of this step is to take definition of use cases and prototype to explain how you are achieving the proposal realized on before milestones. Take every interaction you defined before and use suitable information and situations to explain it.

3.2.6 Demo in the mobile phone.

3.2.6.1 Goal:

Final demo in the mobile phone and project documentation.

3.2.6.2 Methodology:

Functionality: Here you introduce your final solution, what you developed is just what you are going to explain and enrich with the correct speech. Remember is important to translate all the good work realized into satisfaction words for your niche market (clients).

How you are achieving it? Accomplish your solution demands two perspectives, the technical and business perspective. This is the technical perspective, and this is done because of effective solution is good, but much better is the way you used to achieve your solution. This step should be really easy, which means to talk about the work realized in this document, what is done, what is not implemented, the good news, the bad ones and how to deal with the whole picture.

Support for your product: Every project needs the correct documentation and manual. So here you are going to deliver the code documentation, the necessary manuals to install and use the application and all the important information for the final users. Remember you are contributing to people and making useful your work for others, it is not an extra task, is part of a complete definition of build mature projects.

4 S-3R4 Application



Figure 4.1: S-3R4

4.1 Small historical review.

I've been working as mobile developer one year ago. In 2011 I had the honor to be in the NextLab program in Pontificia Universidad Javeriana Bogotá (Colombia). NextLab is a project that reduce the gap between academia and biggest industries of a local sector. Our team (Mobile CLYK) was responsible from development of a Multilevel Model also known as Direct Sell to build an application capable to make orders in real time with real time and accurate information through Android mobile device.

Our incredible adventure gave us a totally new experience, because of it was our first time in real field, or at least the most closer experience with the industry. I've to say that is the most beautiful experience lived during my career, and the beginning of S-3R4 Application. S-3R4 Application was intended to be the second stage of Mobile CLYK application as it was called then, but really S-3R4 today is an expandable solution, totally redesigned and configured to cover more processes than mobile orders, but anyway, S-3R4's hearth was created thanks to the experience and good work done by Mobile CLICK.

4.1.1 What is S-3R4 application?



Figure 4.2: The Big Picture

S-3R4 is thought to be a free application that manages purchase and sale of products and services. What kind of services or Products? The idea is to be the least restrictive, for that reason at the beginning there is no strict rules. The idea is to provide a useful tool for a single person with an Android device to offer a product or service to someone in his social circle.

Why is this useful? Well there is a lot about social circles around, and some of them are worried about your life, others about your current activities and some other about work possibilities according to experience and detailed characteristics. So maybe one of those circles should give you the possibility to offer something as a service or as a product. I believe that everyone, or at least the vast majority would like to offer their knowledge, work and talent to others, and doesn't exist an organized an easy way to do it.

Why today? In a a closer future every person will have a mobile device. Social, sharing, collaborating are one of those words we are hearing every day and a big part of our current activities are stocked with online sources. There is beautiful information over the network, but just like everything, if you want to go deeper in specific details or to have the big picture about something, you have to be patience, and it consumes a big amount of time. People love to share information, sometimes for free, sometimes not. S-3R4 wants to facilitate this work, to enable collaboration with people in their social circles in new ways.

What is the innovation in social circles? If you have something to offer as a complete product or service, create your account, specify details of what are you offering,

use the technology of the moment (maps, voice recognition, NFC, cloud computing), multimedia and online resources with S-3R4 application to take a chance in your social circle.

4.1.2 What is the scope of S-3R4 application for degree work?

4.1.2.1 Important Concepts

The social circle is something that need to be defined carefully. For that reason, this prototype assume social circle as a limited concept for degree work. This stage of S-3R4 will assume the social circle as the members of the same enterprise. This context is going to define all the possible actions that a specific user will perform inside of the S-3R4 application.

4.1.2.2 What is going to be taken from Mobile CLYK?

Practically all the concept of Mobile CLYK is the hearth of S-3R4 application. Direct sell model, the stakeholders and main workflows are keep it in S-3R4 structure, but are exposed, designed and developed in a completely new way. The idea is to be generic in the functional development process, because of Mobile CLYK had an Industry partner which specified certain functionality according to their needs. The code of two stages (Mobile CLYK and S-3R4) is going to be uploaded, so you can verify the big difference between them.

4.1.2.3 What is going to offer S-3R4 application?

Is an incredible free application for business, but at the same time is a completely methodology for android using Java technology. Based on power of Android paradigm and Java EE6, S-3R4 developed a complete mobile prototype software with good functional and non functional requirements (specifically performance and scalability). There is a lot of ways to do this process, but doesn't exist a standard guide to get started, so here readers will find a good one.

S-3R4 is an expanded solution because has a different horizon, and that horizon is about why, how and which customers are being looked for S-3R4 objective. Is thought to be totally free tool that anyone can use to offer a well defined product or service into a social circle. An unique space to manage your own business, to take decisions, to find clients and make profitable the use of emerging technologies.

What it required to be realized? A good experience with Android development, and some Architectural concepts to achieve better results in complete communication between mobile devices and a server powered by Java. S-3R4 is a methodology

that explore typically android problems, how to decrease it's impact and how to have a great experience taking advantage of personalization and good patterns of Android O.S.

4.1.2.4 Technical Concepts

Is important to make known readers that NextLab-MIT Servers erased all the information, and because of its own architecture isn't easy to replicate the good work done by Mobile CLYK Team. Specifically NextLab controled and configurated the servers, they provided two of them (Front-Server and Back-server). All that we knew about it, was they enabled JAVA or PHP technology. To upload services (SOAP services) or create a database (MySQL) was specified a detailed process. Taking into account that the application should be rebuild from database, is going to be totally redesigned and developed; the main objective focuses in create an excellent functional design (the performance wasn't really cool in Mobile CLYK prototype), giving space to non-functional requirements, and enabling scalability for next stages in the application.

S-3R4 is working with Glassfish Server 3.1, JAVA EE6, EJB 3.0, JPA (2.0 with Eclipse Link), Simple XML (2.6.2), JAX-RS (Jersey). The database was redesigned on MySQL but during the development problem there were several problems so Oracle 10g Express Edition was selected.

The mobile device is using Android 2.3.3 (Gingerbread), Simple XML (2.6.2), is based in Mobile CLYK GUI and Android Market applications (Barcode Scanner by ZXing Team and Google voice recognition).The application could be run in Android Gingerbread version (2.3.4) and Honeycomb (3.1) for tablets, besides the methodology propose an easy way to migrate to Ice Cream Sandwich (Android 4.0).

The model client-server requires a totally new work on server side. That's why the objective was to work with the latest technology available (Java) to benefit the development process and introduce a investigation phase (What is the best way to achieve it?) . In fact the goal here is to get a better result from the server side with Web services, and introduce the advantage of business tools as JAVA EE6 (Java Enterprise Edition 6) .

4.2 The Development Steps of S-3R4 application

4.2.1 Elevator Pitch and Work Plan

This is going to be presented in degree stall work.

4.2.2 Need Assessment and Use Case

4.2.2.1 Stakeholders

The stakeholders of S-3R4 Application are:

Administrator: Is a member capable to configure a root account (Enterprise), enterprise rules and act as a member inside the application.

Member: The member can access to full S-3R4 functionality, but cannot modify the root account or Enterprise rules.

Client: The client is someone capable to do orders, but has limited actions inside the application.

4.2.2.2 Interfaces

The interfaces in interaction are provided by the same mobile devices, the main workflows occur depending on mobile device, but was convenient to examine the impact of context elements and the typical users behavior to enrich the possible services that S-3R4 provides.

Could be annoying to an user to create elements in a mobile device (forms with a lot information), probably is more useful to retrieve information, or maybe modify (delete or update) small fields about an element. Looking around existing applications (Amazon, Google, facebook, grooveshark, wunderlist to name a few), its important to recognize the concept of platforms, enabling user to do the hard work via common devices as (desktop or laptops), and at the same time enabling access to the information over mobile devices. That's why more than thinking in simple communication with Web services, is necessary to think about a platform to offer users all benefits of normal and new era applications. This simple thought reminds us the vital work of architecture that will be discussed later[19].

The mobile devices are the main interface, dealing with the context elements, so let's take a look to what have been explored by S-3R4 application, and how was defined some alternatives to resolve this exposed scenario.

4.2.2.3 Important context elements

Smart-phones and mobile devices are evolving quickly, but they still keeping certain characteristics as [4]:

1. Small screens in some cases.

- a) Is good to remember that there are thousands of devices, even is more important to remember that each mobile device has different screen sizes. Android resolve this working with three pixel densities. When you open a project in Eclipse, the application folders contains three “res/drawable-<density>” folders, corresponding to each of the density pixels. So the concept is simple, worry about understanding the three pixel densities and your application will be ready for most devices in mobile market.
2. Limited resources, as memory.
 - a) Work in android application requires to use resources intelligently, as the size of images (better to work with bitmap images), video format, databases, and XML resources. Android specify the size of icons and provides some templates. Remember to use appropriate formats and to take advantage of XML resources when those not require to be built dynamically. Following the same point of view keep persistent methods in the application as a good practice, and choose the better one depending on situation.
 3. Availability of Connection and fluctuations in the connection.
 - a) Connection comes and goes all the time, so handling exceptions and giving some chances to make requests is good practice. Android provide the use of services, enabling asynchronous tasks, and binding information to GUI thread (main thread) to avoid application crashes, and not bother users with excessive waiting.
 4. Requires alternatives to achieve specific needs in devices, because of age, different hardware characteristics, diseases and environment restrictions.
 - a) The scope of project involves a big people segment, then thinking and evaluating customers will bring some creative and good alternatives to satisfy the same needs in different ways. Technology in devices is increasing, so use Maps, voice recognition, biometric technology, NFC, GPS, virtual reality to make easier user experience.
 5. Usability and “fast an fun” characteristics.
 - a) A good application could be exceptional, but to do the magic requires to be intuitive, easy, flexible and eliminate subjectivity. Managing personalization and adaptation models is possible to handle great user experience. Google has some good videos over the web, and those are increasingly focusing in enchant, simplicity and emotional aspects.
 6. Different locations weathers and sound levels.
 - a) Services could vary according to location, in fact location and context variables are new variables that impact in real smart actions (Artificial

Intelligence) as increase bright of screen, show the information in different ways, manage automatic filters, pushing notifications, generating advices just to name a few examples.

4.2.2.4 Use Cases

Definitely in my short experience with documentation for large applications, I learned that sometimes documentation was more expensive than create the applications, however there are some diagrams and processes that inevitably help and support a good development. The purpose of use cases is to analyze the main workflows in system. Here are defined three actors, but remember that all functionality provided to client actor will be available for the other two actors (Member, Administrator). In the same way all the member actions are available for administrator actor.

Let's take a look to Use Cases:

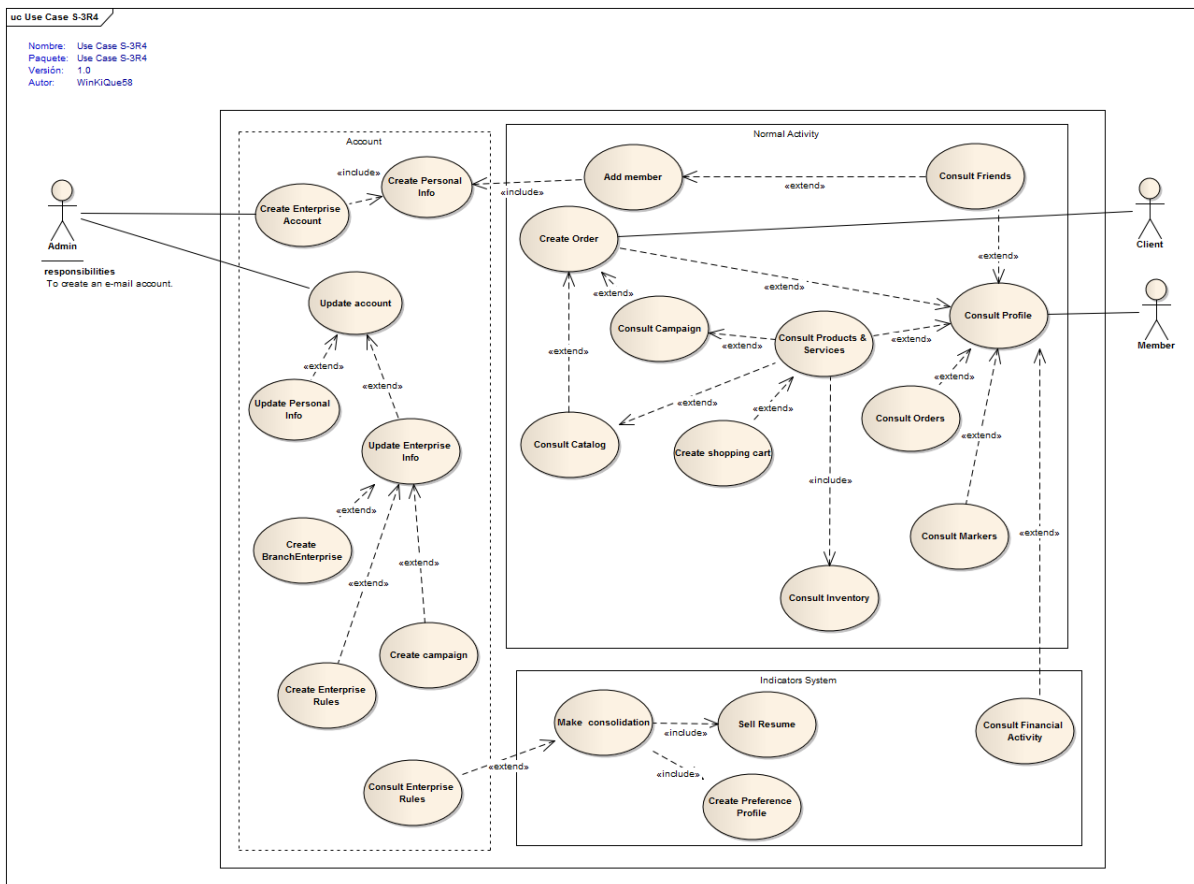


Figure 4.3: S-3R4 Use Case

4.2.2.5 Requirements

Requirements Template Work without requirements is like to shoot into the air and hope that something falls. The development core are requirements, those requirements guide all the process, determine time, people, resources and should be define according to mission and vision from who undertake the project.

The first important thing is to define a requirements template. If you want to see the list of requirements please see S-3R4 Requirements file.

ID:	Unique name.
Creation Date:	When was added to S-3R4 design.
Alias:	An alternative name to ID. Easy to remember, used to highlight some kind of issue or to pay attention to special details.
Origin:	Why was important to included it?
Description:	The requirement
State:	Is a proposal?, is being tested?, is being developed?
Complexity:	How difficult could be to implement it in a 1-4 scale.
Language:	JAVA
Use Case:	A Traceability tool for design.
Priority:	A restriction or necessary step for others requirements.
Tested:	Functions, Unity or Integrated with system.
Restrictions:	The inputs for realize it work.
Notes:	Something to keep in mind.
Metric:	How is expected to be qualified?
Associated Requirements:	In the same Use case or inputs for the correct functionality or development.

Table 4.1: Requirements Template

FUNCTIONAL REQUIREMENTS				Non Functional Requirements			
ID:		Version:		ID:		Version:	
Creation Date:				Creation Date:			
Alias:				Alias:			
Origin:				Origin:			
Description				Description			
State:		Use Case:		State:		Use Case:	
Comple		Priority:		Complexity:		Priority:	
Language:		Tested:		Language:		Tested:	
Restrictions:				Restrictions:			
Notes:				Notes:			
Metric:				Metric:			
Associated Requirements:				Associated Requirements:			

Figure 4.4: Requirements Template

4.2.3 System Design and Architecture

The model is divided into color zones. The relational data model was created with Data modeler (Oracle SQL Developer Tool). To obtain the relational model is necessary to do the logical diagram model. The reason is that each logical diagram could have several relational data models, and that's why is valuable this process.

Why colors? The application contains a considerable number of tables, in order to understand, navigate and modify the diagram, the colors are useful. Every color makes reference to a group of tables related. The blue color specifies the structure of multilevel model, green is related to the people and the important objects that they have access in S-3R4 application. Red color is for element states, and yellow is about user preferences.

Why those tables? The analysis was designed with the most common multilevel models, their processes, usual operations, and organizational hierarchy. Because of the experience with Mobile CLYK team, all this elements were redesigned to support localization, and to provide a flexible way to configure new models as distributed, making profitable the mobile characteristic[16].

Why no more or less tables? As main restriction is the time, is important to evaluate which is going to be this second approach to S-3R4 application, using Mobile CLYK experience. So no less tables to really show what should be capable to do the system, and no more tables because of the scope of S-3R4 project. Either way the design is made to grow in each stage of development.

What is new and why is better than Mobile CLYK design? The design is different because of specific elements as Branches, Warehouses, distinction of Products and items and how are mapped users. Location is a characteristic of several elements, and maybe the most important new characteristic is to provide an easier way to make consolidations, applying enterprise rules is more understandable and simple, and exist a first approximation to an adaptation model.

Is better for query's, for performance, for model scalability and definitely for Location. Query's are simple than before, entity responsibilities are assigned in better way, allowing just necessary join tables, reducing complexity and impacting in performance which is really amazing for S-3R4 application. The model is thought, designed and developed to be magnified assigning the correct responsibilities, and making a good use of relations (strong and week relations) with foreign keys.

ADVICE: Take time to design a good database, practically is the hearth of application and cause the 60% of problems in development with android and JAVA technology.

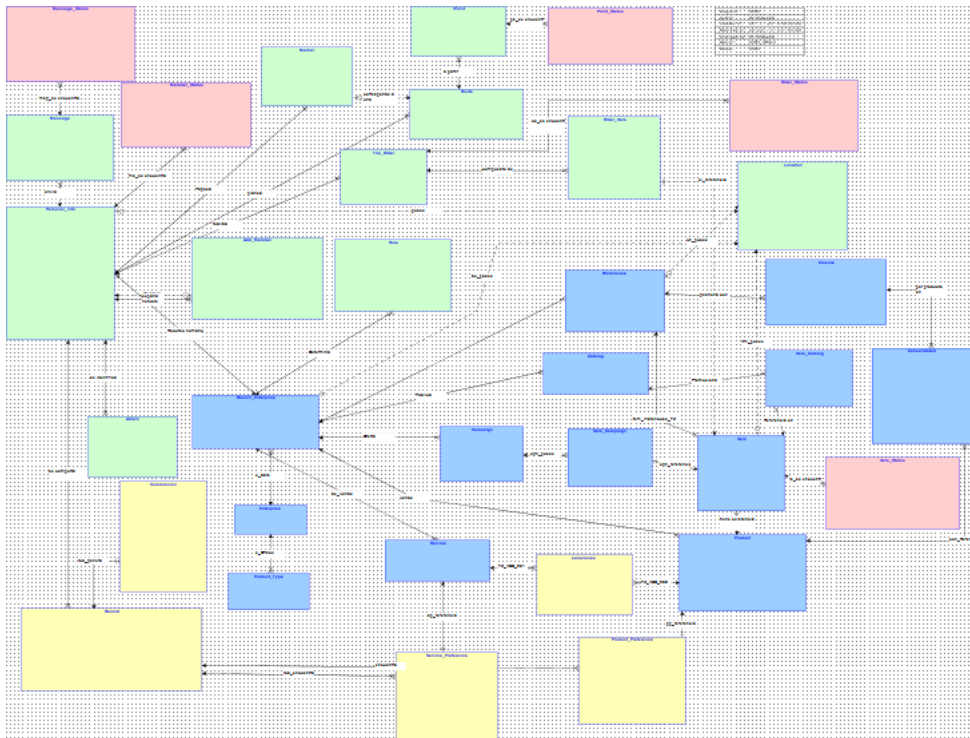


Figure 4.5: Domain Model

4.2 The Development Steps of S-3R4 application

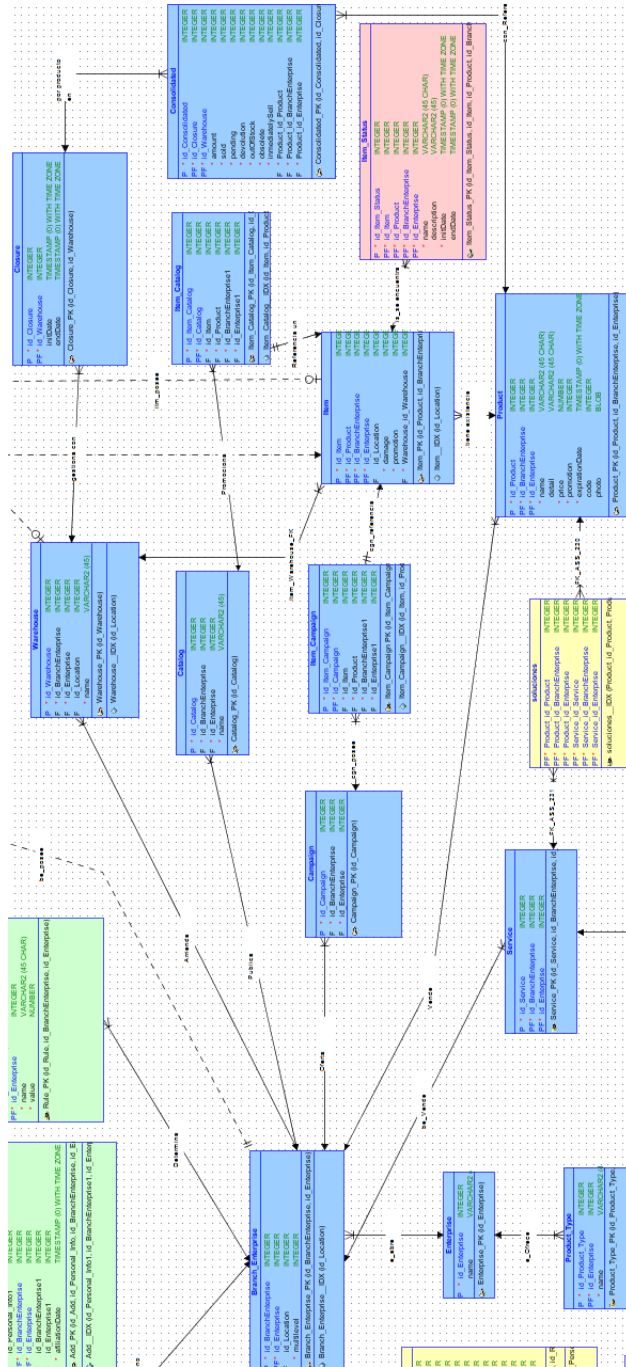


Figure 4.6: Business Objects

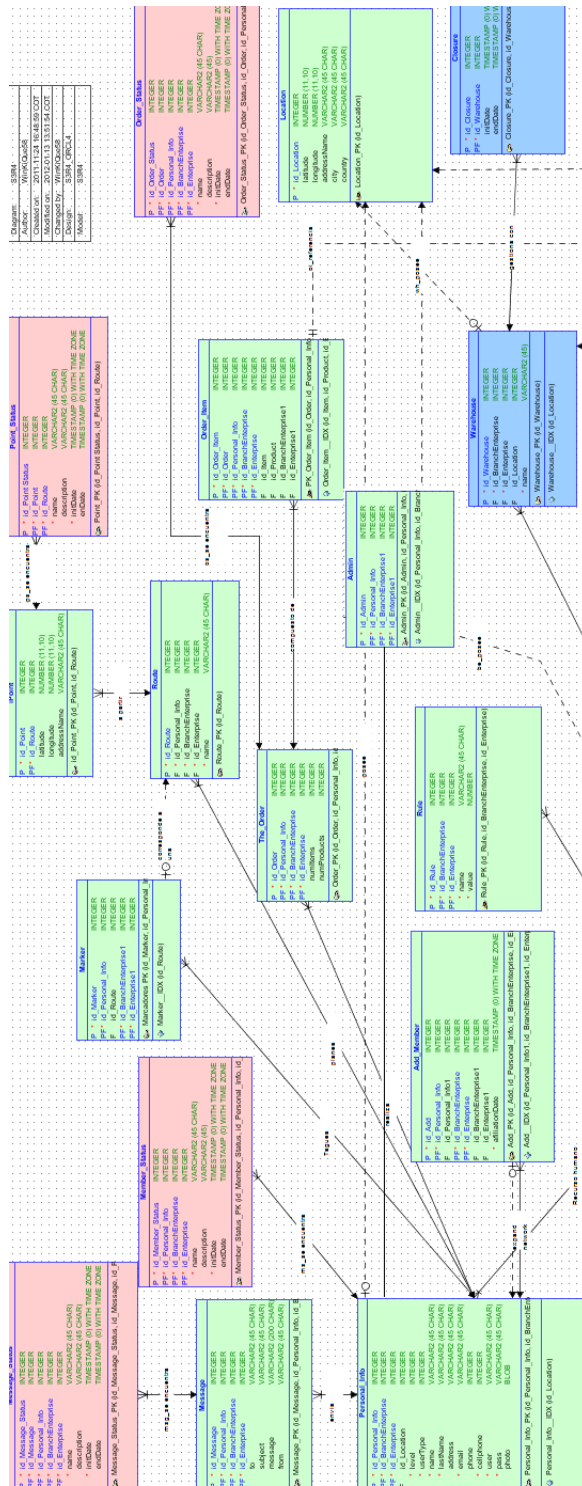


Figure 4.7: People and State Objects

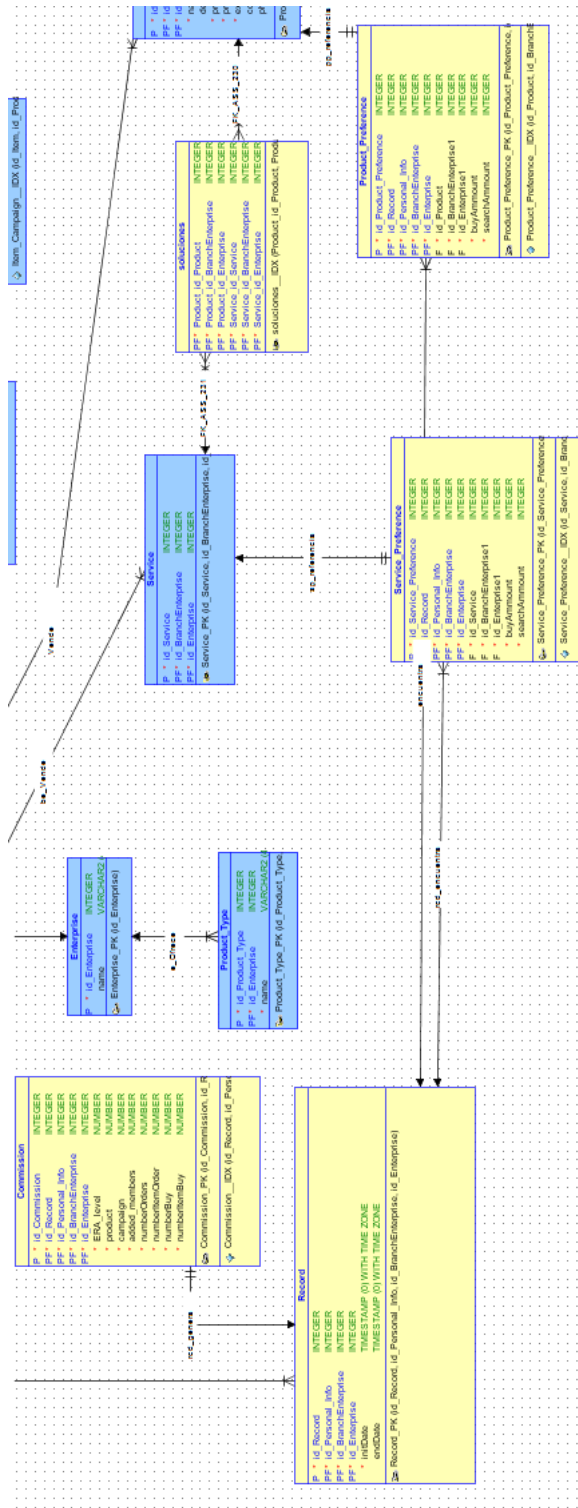


Figure 4.8: Preference Objects

4.2.4 Architecture Perspective

4.2.4.1 The Big Picture

Deeply inside of basic client-server architecture, there are two important perspectives of S-3R4 application: the horizontal and vertical perspective. According to Krutchten logical diagram could be done with a E-R diagram, and S-3R4 E-R diagram was exposed in the last section, that's why logical diagram is not necessary. The process diagram is not useful for this phase of S-3R4 application, because of the unique parallel work is doing S-3R4 application is related with web services, so the details given in other diagrams are enough to comprehend the structure of application. The Vertical perspective for S-3R4 application is captured in Components Diagram (Layers and Packages of S-3R4 application), and the Horizontal perspective is captured in deployment diagram (nodes and components).

Imagine the most simple flow of what is happening in S-3R4 application interaction, architecture shows a high level model, revealing how E-R diagram is wrapped and enable to work through subsystems, enabling good performance and scalability.

4.2.4.2 Development Diagram

This diagram is the development view, here is possible to see the layers, modules, packages and their corresponding relations.

There are four layers in Android application:

1. Android Functionality: This layer was built to mitigate the most criticized issue in android, and make reference to the version compatibility when Android upgrade the SDK version. The platform is evolving but the applications are suffering because of a big change produced in Android API by new versions.
 - a) In order to deal with this problem is proposed a multiple inheritance that works with specific functionality provided by android framework. So all the core Android API's are defined in a separate module from GUI application. The idea is to create generic uses of Android technology which are linked directly with android operative system. The GUI application just extends the last class from multiple inheritance and make use of that generic functionality.
 - b) As a good example the NFC API that brings Android 4.0 requires to do application changes, and this technique is looking to minimize rewriting a lot of code, so is just necessary to reorganize the NFC generic class, and maybe some supported classes without interfere with the rest of application. This is a good practice for reuse of code, maintainability, scalability and version migration.

- c) The figure is showing how SRC folder is organized. First package is for activity templates, where exist two activities, one for dialog activities and the other for common activities. The second package is for supporting main functionality and the last one is functionality package that contains the technology provided by Android. Please take a look to the screen class that is showing the multiple inheritance, and the generic functions as open video and open audio

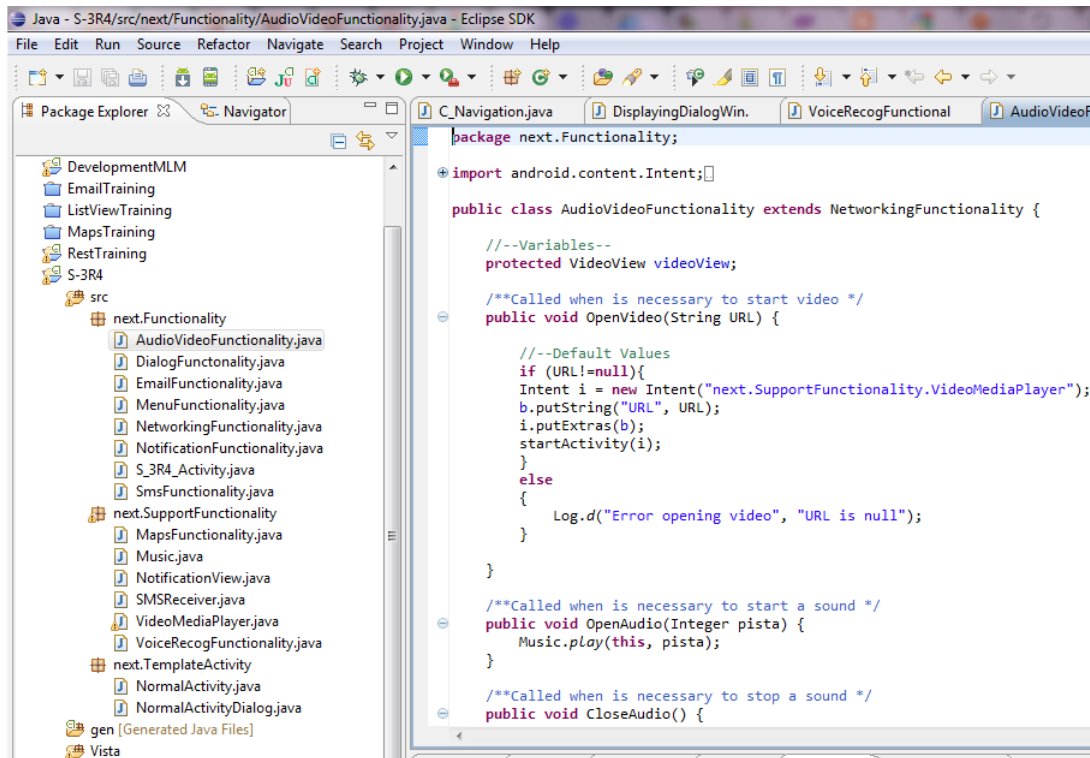


Figure 4.9: Multiple inheritance and generic functionality.

2. View: Is part of a typical Three-tier-pattern, and this layer deal with application views (layouts and GUI elements). The listeners are not part of this layer, because of despite making easier code creation, affect maintainability, scalability, traceability and reuse of code.
 - a) Is important to declare some structured names for resources and layout, because of Android doesn't enable creation of new folders or other packages, resulting an annoying disorder when exist a big number activities, therefore multiple layouts.
 - b) Use resources to save application Strings, so GUI is decoupled with language of the application, This enable creation of GUI templates, and easy change for string values in all the application.

- c) Customize your objects, android is totally about personalization. S-3R4 application use only custom elements, showing the easy and incredible power of android personalization.

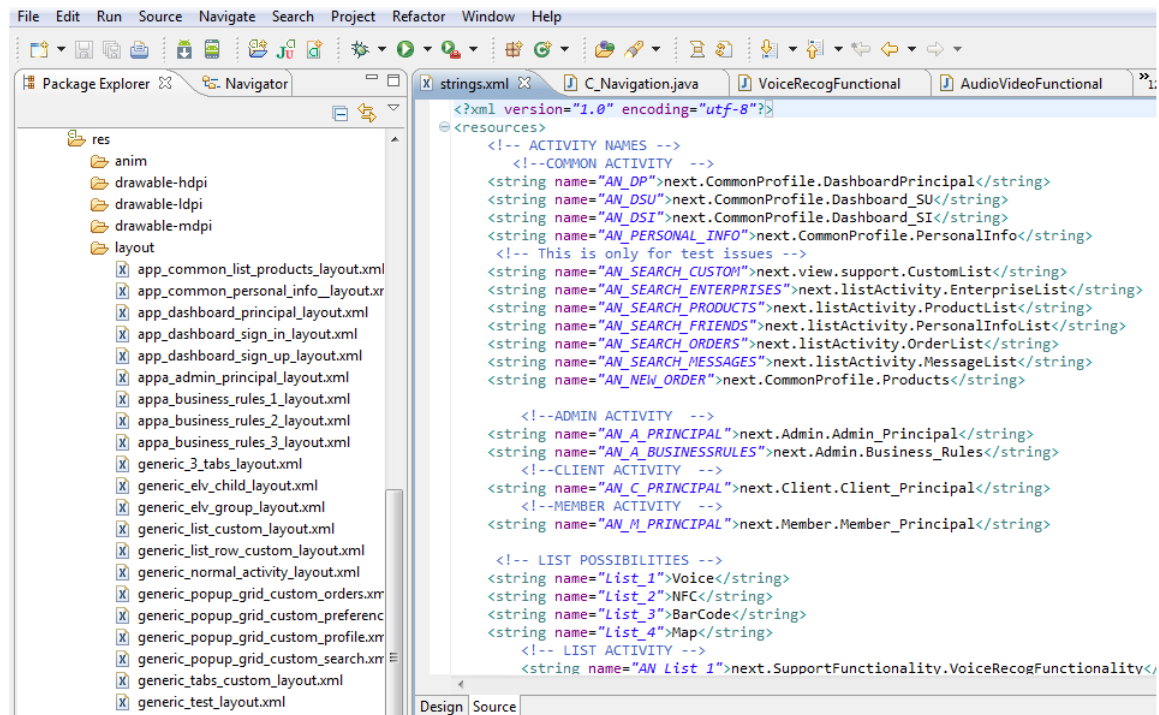


Figure 4.10: Multiple inheritance and generic functionality.

- d)
3. Controller: Is also part of a typical Three-tier-pattern, and ensures S-3R4 workflows, it means that to proceed becomes necessary to ask the controller if is possible. Controller provides the correct contact, the relevant information or application status and verify the quality of the processes. Controller is in charge of possible navigation sequences, listeners, global variable, exceptions, and enable communication with the server side.
- Responsible for navigation providing control in workflows and managing activity stack. Also benefit database synchronization and smart web services, depending on what is using user .
 - Tip: Remember to use Manifest file to specify what activities should have an only instance. Managing not sequence processes could result in a big disorder, and a completely failure for your application, because the application could be calling the same activity several times.
 - Listeners, global variables and exceptions should be manage by controller. On this way is easier to find mistakes, modify actions and make clearer the developed code.

- d) Controller enable activities, services, broadcast and content providers, is responsible to communication, and control threads work.

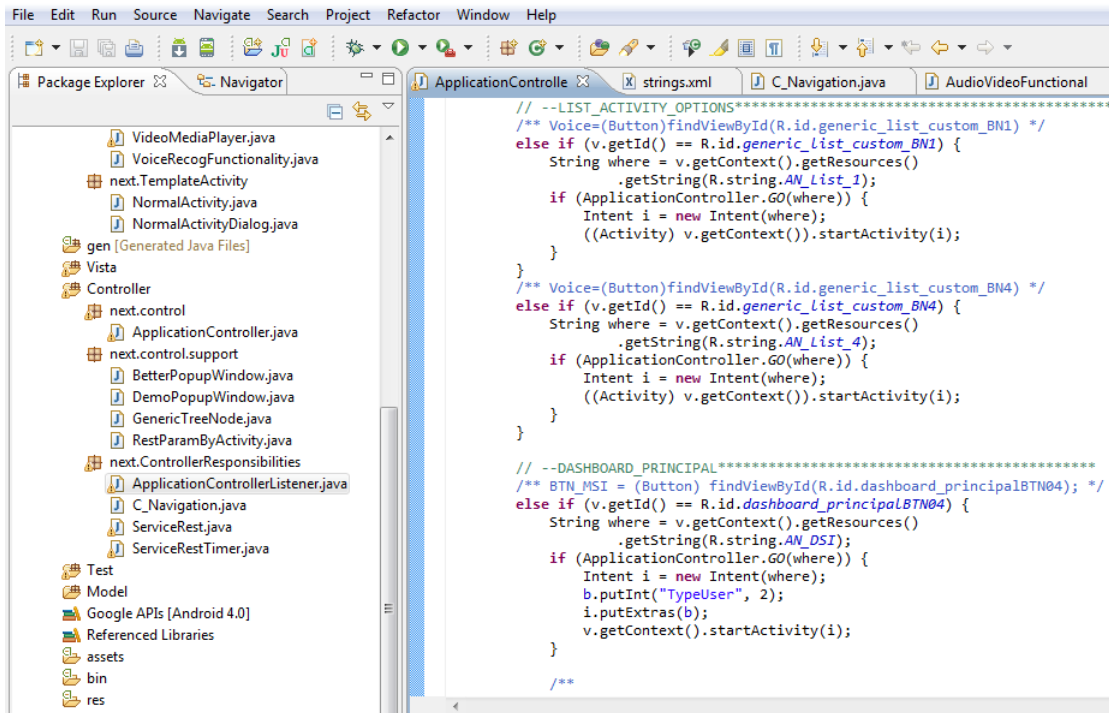


Figure 4.11: Multiple inheritance and generic functionality.

- e)
4. Database: Persistence is an important element when working in mobile devices. Every step procedure requires secure transactions and to use resources in the better possible way. The communication with server should be reduced, and user experience is going to be fantastic only if the mobile device define good persistence methods.
 - a) MySQL lite is a great tool to manage current information, and the idea for Android devices is to synchronize external databases with MySQL lite in order to get the best performance for your application. (Content providers and asynchronous tasks are the elements needed to do this job.)

The server side is working with Java EE 6 architecture and respecting its architecture the same MVC pattern is used.

1. View: Is responsible for Web server needs, specifically about JSP pages and Servlets. The same MVC pattern is inside of View, where servlet act as a controller of HTTP requests, and the returned answers that will be rebuild in the client side as the View.

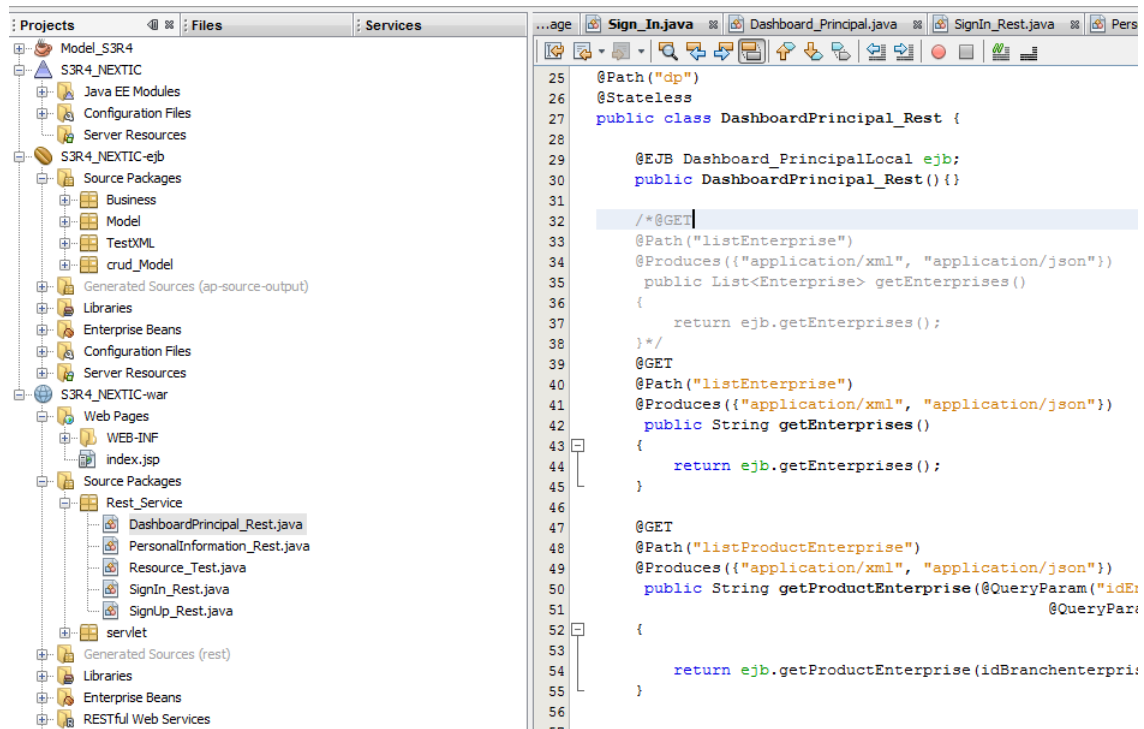


Figure 4.12: Server View

a)

2. Controller: The EJB container is responsible for transactional operations and the analyzer of coherent requests, ensuring atomicity and consistency in database.
3. Model: The server model is the same of mobile device, the unique difference is that is not expressed as a Entity in mobile device. All EclipseLink annotations are deleted and act as a simple POJO in mobile device. The model contain three important modules, the business module, the CRUD operations and the model itself.
 - a) Business model: Use the CRUD operations to satisfy user needs, and decouple S-3R4 functionality with base model.
 - b) CRUD operations: Create, retrieve, update and delete operations over database.
 - c) Model: The entity classes with custom query's, typical constructors, getters, setters and XML annotations.

4.2 The Development Steps of S-3R4 application

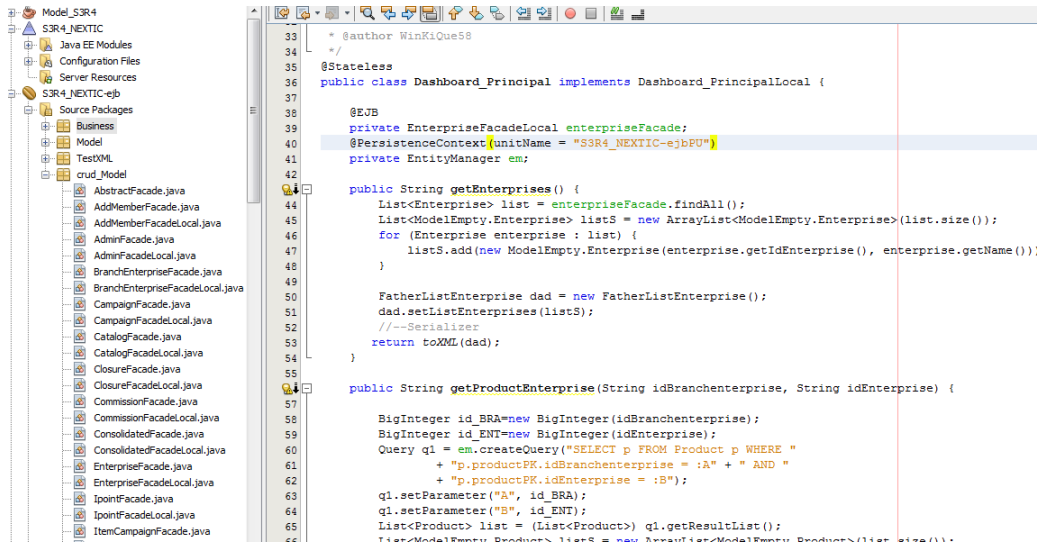


Figure 4.13: Server Model

d)

Practically mobile devices just need to know how to read objects, in the same hand the server is using Restful services, light and flexible services that are being used by facebook, Google and twitter company's. Soap was good with Mobile CLYK experience, but REST are enriching speed and does not need specific tools. Also during S-3R4 development and research was demonstrated that EJB and Restful Web services could work together with excellent results. Better yet, Google provides since 2010 an API for Restful services, so this is maybe the main key to gain excellent performance in user experience.

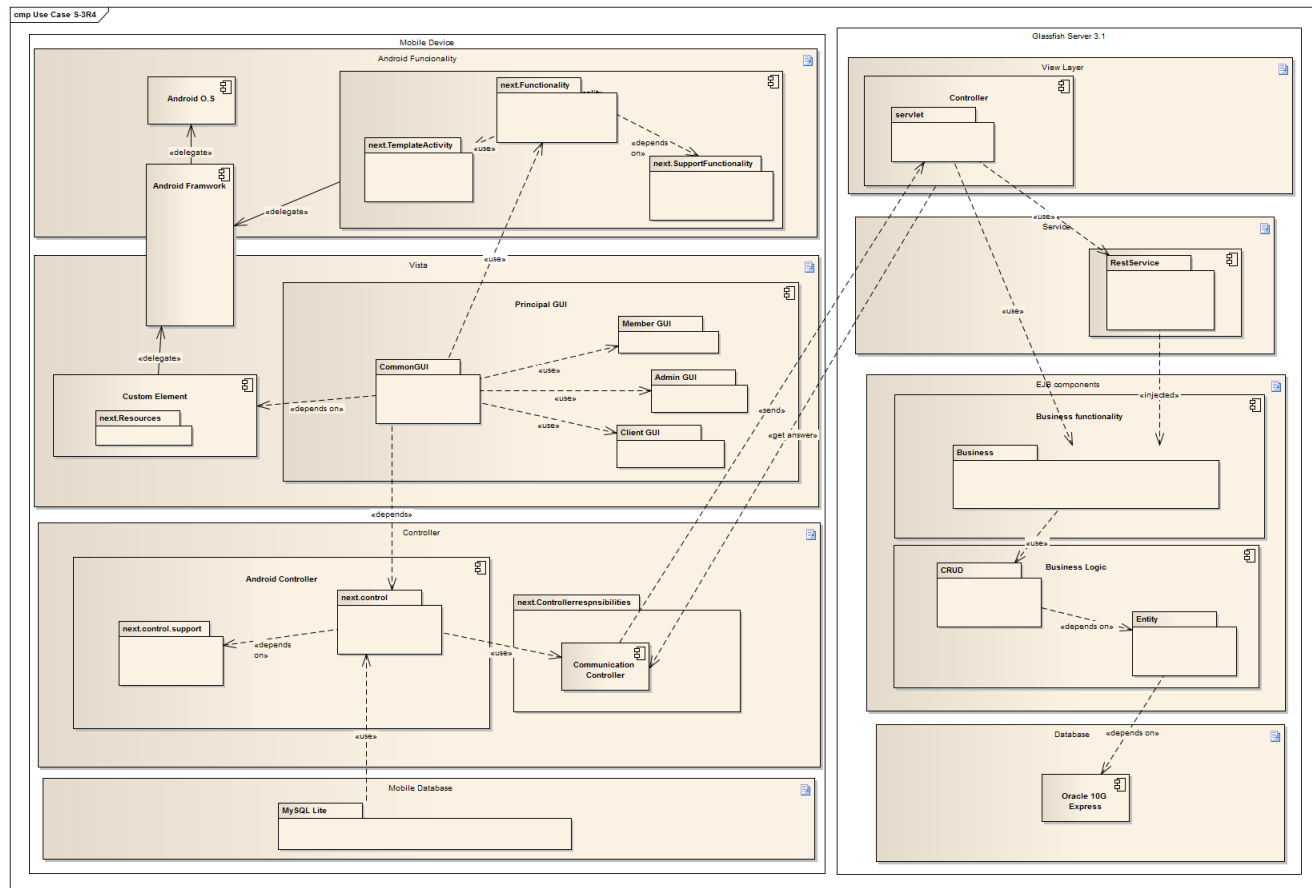


Figure 4.14: Component or Development Diagram.

4.2.4.3 Deployment Diagram

This is simple diagram for S-3R4 application (phase one). Three nodes, where Glassfish server and Oracle database are in the server side.

The communication via server is the most common protocol HTTP, by the other side database connection is using OJDBC to do the magic. The main archives that is using application are:

APK: An executable file that contains all the necessary libraries to be functional for an Android device with 2.3 or higher operative system.

S-3R4-war: Is the file that is packaging all the web server information, that is interpreted by Glassfish server.

S-3R4-ejb: Is the file that contains all the model and business information.

S-3R4-ear: Is a file that contains S-3R4-war and S-3R4-ejb, is a standard file and makes easier deployment of modules of S-3R4 application.

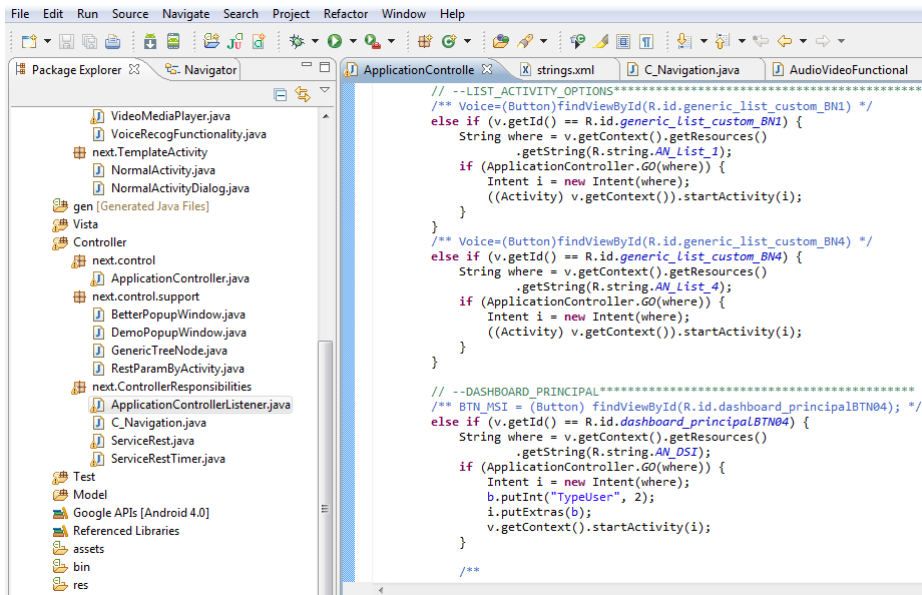


Figure 4.15: Physical or Deployment Diagram

4.2.4.4 Package Organization

In the android device:

Android has both restrictions about resources as the inability to create new folders or packages or the condition of lowercase names. For that reason S-3R4 application specified a name parametrization for resources making possible scalability and at the same time acting as a design tool for maintainability in Android applications.

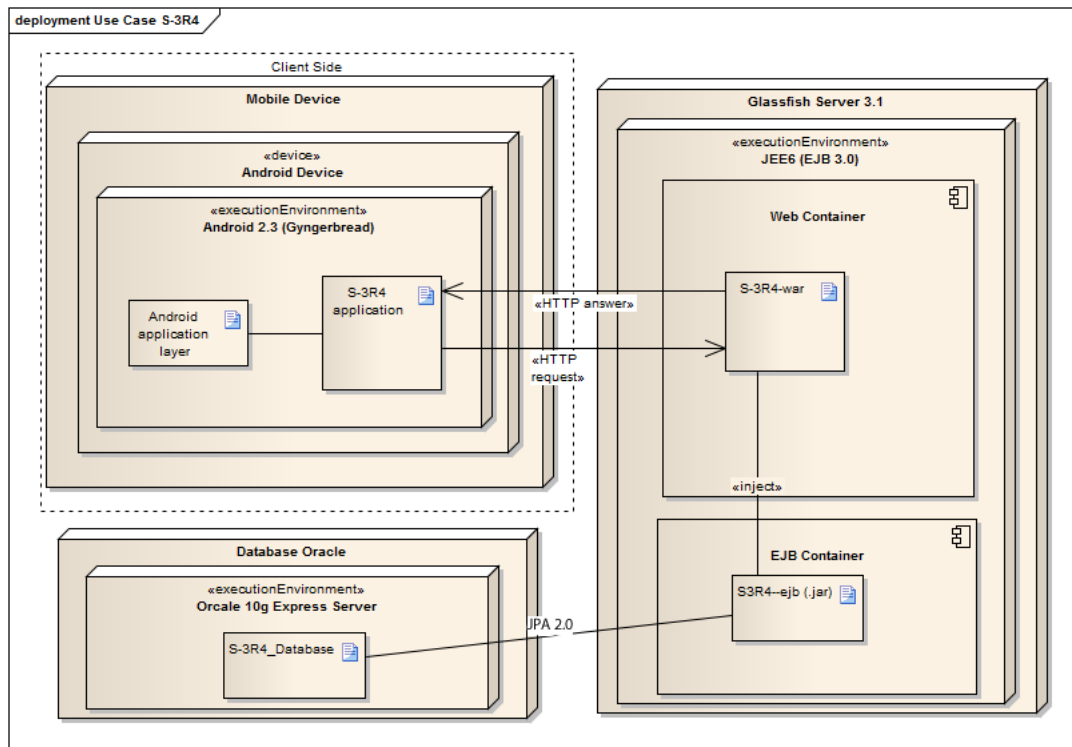


Figure 4.16: Organization of Android Code

In the Server side:

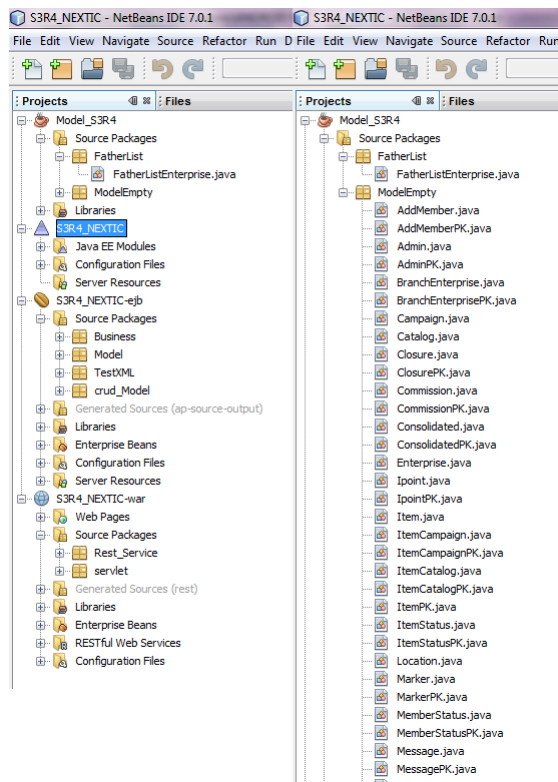


Figure 4.17: Organization of Server Code

4.2.5 Wireframes of Mobile application.

The visual design as was described in S-3R4 introduction, preserve the base visual design of Mobile CLYK. So here will be described the recommended GUI Android patterns and the main screens of S-3R4 application.

4.2.5.1 The Navigation Three.

This document is emphasizing the work realized by S-3R4 application, for that reason if you want to check the navigation three, you could explore Mobile CLYK files. Anyway, the reason for this section is to make known readers that this step is an incredible good practice before to implement GUI application. Navigation three evade circle activity calls for complex applications, and enable to realize smart actions depending of what is doing the user.

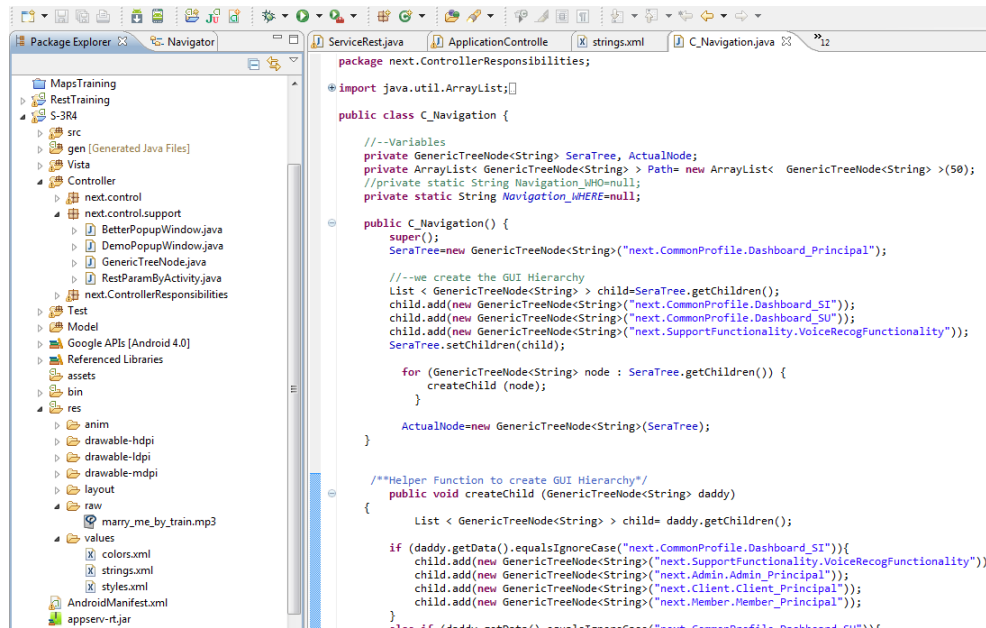


Figure 4.18: Navigation Controller

4.2.5.2 The GUI Android patterns.

[4, 19]Philosophy on android devices is about:

1. Clear vs Simple
 - a) The idea is to show a clear way to understand the common functionalities of application. Simple is a word has sometimes a bad connotation because of the lack of power and poor scalability. S-3R4 is thought to be a really complex platform in the future, so is necessary to use patterns as Action bar and Quick action bar (are described later in this document) to make easier user experience.
2. Content vs Chrome.
 - a) Use all screen of android devices. You need to deal with complex screen all the time, but you need to make profitable the space. The lists are so cool in android, there is a total world for list view in Android. Avoid overload of widgets, specially buttons, and try to generate a location sense inside the application, because of navigation options are always those actions a user will search.
3. Consistent yet engaging (elegant variation)
 - a) Visual alternatives, different designs are good tools to enhance user experience. Keeping application consistence but varying the visual expression and treatment you will find a live product, to exploit the psychological area and make happier users.

4. Enhanced by cloud

- a) User are expecting to do things once, to make profitable the information they provide and the information they usually use. That's why all smart actions are about adaptation, personalization, and synchronizing information across different devices (platform) to understand in a better way your user.

4.2.5.3 Dashboard

This is the first experience between user and the application. Love at first sight is not all, but is the starting of everything. The user should experiment some kind of good sensation, and it is produced not just by a good design, but is produced by the facility to understand the elements that are being presented and how these elements are organized to give user a good idea of what is possible to do with them.

Dashboard is not expressing complex things, is more about a set of main elements, and the possibility to explore the incredible power they provide. S-3R4 dashboard focus in the user hierarchy organization, showing the possibility to create an account or to go to your profile. Why are those accounts there? That's why the text box is placed in the bottom, showing that users experience is around products and services from an Enterprise. The idea is to produce this kind of thoughts: Free account, and product Enterprises? come on lets see it!

4.2.5.4 Action Bar

Maybe the most important pattern. It makes visible the main actions inside the application and control navigation and searches. This pattern contains the title to know the current position of user in the application, the icon for showing the kind of user and the main buttons with expressive draws to guide actions.

1. People: Here is the personal information, specifically the profile.
2. Message: The communication inside of the application with other users.
3. Favorite: Custom and powerful actions in S-3R4 application.
4. Search: Enable different kind of searches inside S-3R4 application.



Figure 4.19: Dashboard (Left image) and Action Bar (Top Screen)

4.2.5.5 Quick Action Bar

The quick action bar is a pop-up triggered from distinct visual target. It is minimally disruptive to screen context and contain straightforward actions. So all S-3R4 main possibilities are inside of this pattern, and is proper to determine the specif actions of an item, so users know what they can do with a GUI element. Quick action bar is applied to the Action Bar, so if a user want to search something, probably quick action bar will determine what kind of elements can be searched.

4.2.5.6 The world of List View.

The list are magnificent to show information, is the most common, simple and clear way to do it. Avoiding several widgets or complex forms is possible to filter information, to manage POJO'S and to customize the visual design in Android GUI. S-3R4 application use lists for show any kind of objects and uniquely use other type of structure to show an important or specific details of S-3R4 elements.



Figure 4.20: Business Rules and Consult

4.2.5.7 The use of technology.

As an important context element was founded that a mobile device should have several ways to realize a task. Technology is an innovative manner to resolve it. Voice recognition and Google Maps are fun and amazing tools as a good example of how S-3R4 application make profitable technology to get, show and save information in different ways.

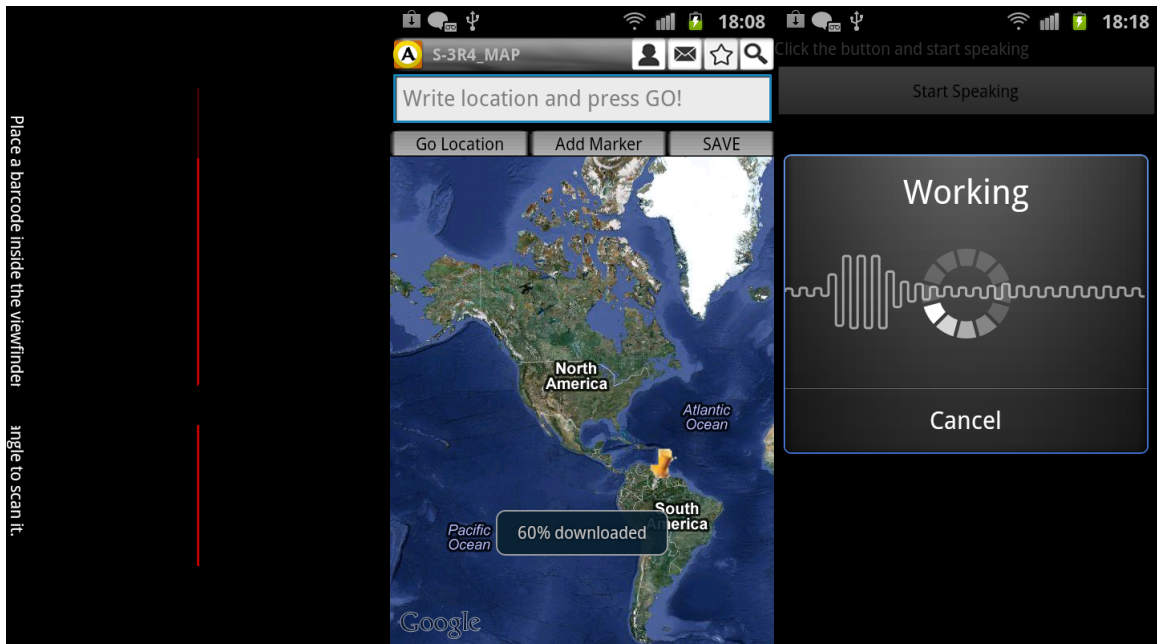


Figure 4.21: Technology

4.2.6 Demo of Web services and mobile phone.

4.2.6.1 HTTP Protocol.

HTTP is an application-level protocol that defines operations for transferring representations between clients and servers. In this protocol, methods such as GET, POST, PUT, and DELETE are operations on resources. This protocol eliminates the need for you to invent application-specific operations such as create Order, get Status, update Status, etc. How much you can benefit from the HTTP infrastructure largely depends on how well you can use HTTP as an application-level protocol. However, a number of techniques including SOAP and some Ajax web frameworks use HTTP as a protocol to transport messages. Such usage makes poor use of HTTP-level infrastructure.

4.2.6.2 Web Services.

There are two philosophies when writing web services:

1. REST: using XML, JSON and HTTP. Each URL represents an object that can do POST, GET, PUT and DELETE (the typical operations of HTTP).
2. SOAP is all XML-based infrastructure. Where each object can have methods defined by the developer with the necessary parameters. It is Rigid, strong typing, is a contract.

Advantages of REST

1. Light: Object transfer JSON or XML.
2. Legible results.
3. Easy to implement: There are not specific tools.

REST (Representational State Transfer), in essence, was created to use the HTTP specification correctly. Gaining wide adoption across the web as a simpler alternative, REST replace SOAP and Web services based on WSDL (Web Services Description Language). Large companies of Web 2.0 as Yahoo, Google and Facebook left SOAP and WSDL services and began to use a more user-friendly, resource-oriented. That's why S-3R4 application is using REST web services. In the other hand KSOAP2 was experimented in by Mobile CLYK team, and results was more delayed and more complex to implement.

4.2.6.3 What is needed to start with REST Web Services?

URI's are identifiers of resources that work across the Web. A URI consists of a scheme (such as HTTP and HTTPS), a host (such as www.example.org), a port number followed by a path with one or more segments (such as /users/1234). REST services use URI's where is necessary to:

1. Use domains and sub-domains to logically group or partition resources for localization, distribution, or to enforce various monitoring or security policies.
2. Use the forward-slash separator ('/') in the path portion of the URI to indicate a hierarchical relationship between resources.
3. Use the comma (',') and semicolon (';') to indicate nonhierarchical elements in the path portion of the URI.
4. Use the hyphen ('-') and underscore ('_') characters to improve the readability of names in long path segments.
5. Use the ampersand ('&') to separate parameters in the query portion of the URI.
6. Avoid including file extensions (such as .PHP, .ASPX, and .JSP) in URI's.

REST web services enable JSON and XML, S-3R4 application decided to use XML, but exist several libraries to work with JSON format. The library to create XML format is free license JAVA library called Simple XML. XML is a general-purpose format, and it is the server's responsibility to design a way to include links in XML-formatted representations and document that design to clients. Clients can refer to that design to learn how to find and use links included in representations.

4.2.6.4 How to implement REST Web Services?

Sometimes is better to watch the code and later to do questions. The first step to define a web service is to create a class under Web project (Server side), and over the source folder to create a simple JAVA class. To convert JAVA class to a REST file is just necessary to use Jersey library that is the Reference Implementation for JSR 311 (Java API for RESTful Web Services). Annotations that comes with Glassfish Server 3.1 enable to do the job (please take a look to API for RESTful Web Services Java Go to java community process):

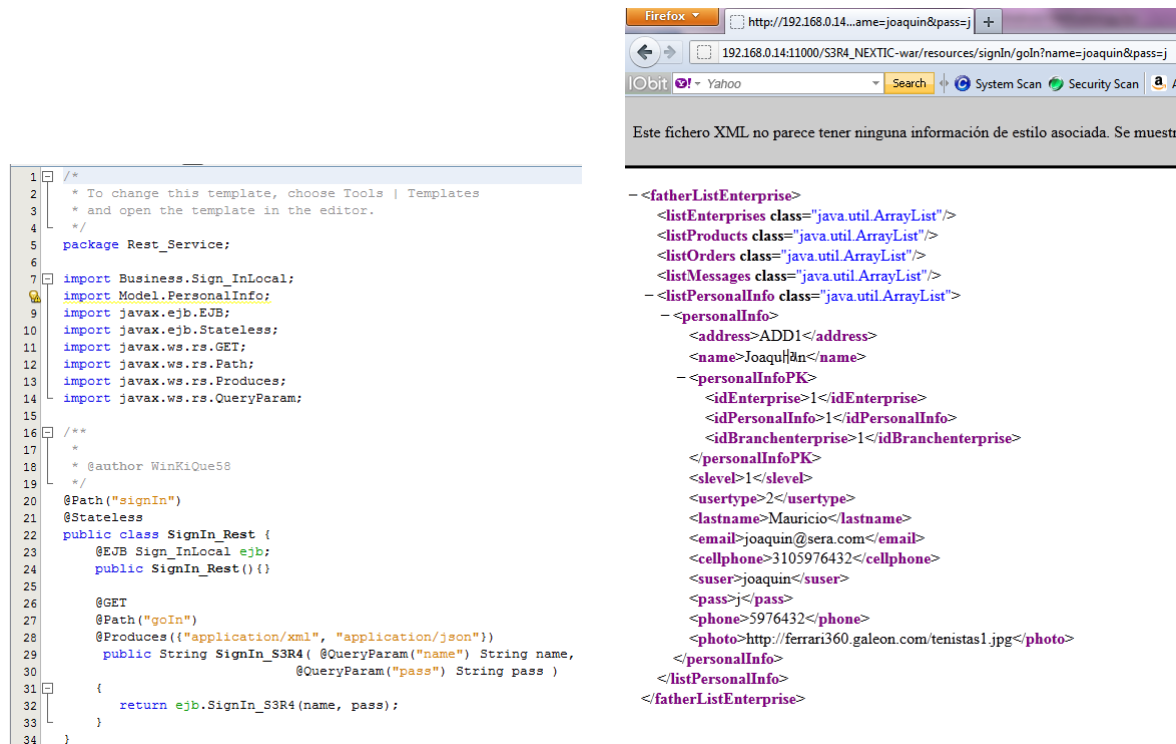


Figure 4.22: REST Web Service

4.2.6.5 What is Happening?

The resource is exposed in URI expression:

- “HTTP://IP:PORT/WEBProject-war/resources/CLASS_PATH/MethodPath?parameter1=

It means that class is the second step or hierarchy, and all the methods of the class are a set of related services. Observe that REST class is an stateless EJB in Glassfish Server 3.1, and the most beautiful about all this is the simplicity to inject a remote EJB from EJB project (.jar from EJB container).

1. <IP>: The host where is server.

2. <PORT>: the port of server being used. It could vary depending of server but for Glassfish server commonly is 8080.
3. <WebProject-war>: The REST service require to be deployed in war project for Glassfish 3.1.
4. <Resources>: You can set the name of resources path, by default is “/resources”.
5. <CLASS_PATH>: Is mandatory and should be placed before class declaration.
6. <MethodPath>: Is the same path used in class, but is using hierarchy in this case.
7. <?>: Enable to set parameters for method if exist.
8. <parameter=value of parameter>: Parameters are separated by “&” and always travel over the web in URL as a strings.

```

@Stateless
public class Sign_In implements Sign_InLocal {

    @PersistenceContext(unitName = "S3R4_NEXTIC-ejbPU")
    private EntityManager em;

    public String SignIn_S3R4(String name, String pass) {
        PersonalInfo pi = null;

        //--Name is Unique restriction!
        Query q1 = em.createQuery("SELECT pi FROM PersonalInfo pi WHERE pi.suser = :NAME AND pi.pass = :PASS ");
        q1.setParameter("NAME", name);
        q1.setParameter("PASS", pass);

        try {
            pi = (PersonalInfo) q1.getSingleResult();
        } catch (NoResultException e) {
            return null;
        }

        List<ModelEmpty.PersonalInfo> listS = new ArrayList<ModelEmpty.PersonalInfo>(1);
        ModelEmpty.PersonalInfoFK PK = new ModelEmpty.PersonalInfoFK(pi.getPersonalInfoPK().getIdPersonalInfo(),
            pi.getPersonalInfoPK().getIdBranchenterprise(), pi.getPersonalInfoPK().getIdEnterprise());

        ModelEmpty.Location l = null;
        if (pi.getIdLocation() != null) {
            l = new ModelEmpty.Location(
                pi.getIdLocation().getIdLocation(), pi.getIdLocation().getLatitude(),
                pi.getIdLocation().getLongitude(), pi.getIdLocation().getAddressname(),
                pi.getIdLocation().getCity(), pi.getIdLocation().getCountry());
        }

        listS.add(new ModelEmpty.PersonalInfo(PK, pi.getSlevel(),
            pi.getUsertype(), pi.getName(), pi.getLastname(),
            pi.getEmail(), pi.getCellphone(), pi.getSuser(), pi.getPass(),
            l, pi.getPhone(), pi.getAddress(), pi.getPhoto()));

        FatherListEnterprise dad = new FatherListEnterprise();
        dad.setListPersonalInfo(listS);

        return Dashboard_Principal.toXML(dad); }
}

```

Figure 4.23: The EJB method

The translation to XML : The XML translation use model with corresponding annotations and defines a root class denominated “FatherLisEnterprise”. The reason

is that Simple XML requires a root class. Taking advantage of it, this root class is composed with array lists of every object in database enabling to return several objects just calling a web service. Observe that XML translation is a responsibility of EJB module, and that apparently the return object is a String, however annotations in Rest service determine the real return type.

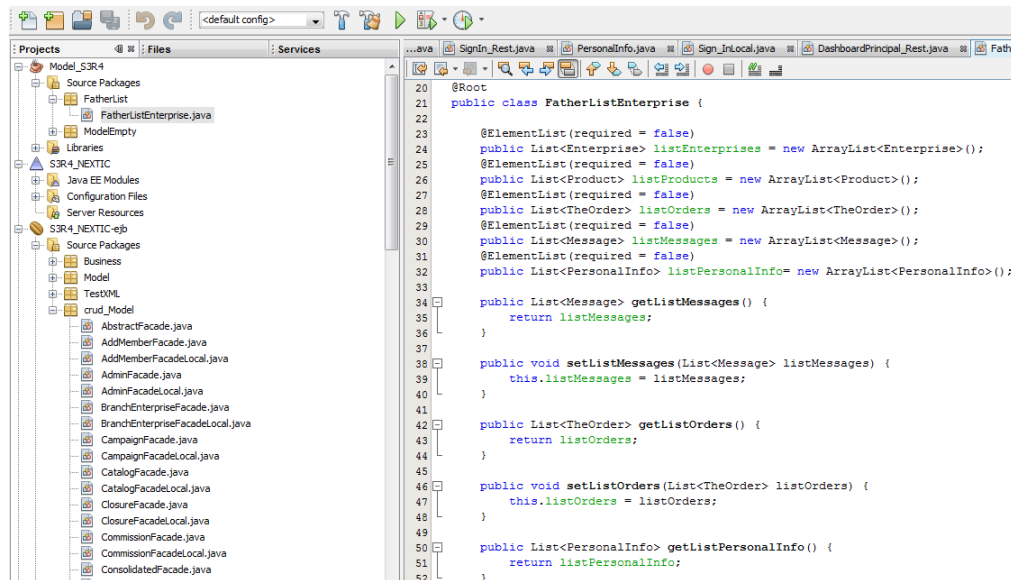


Figure 4.24: The root class

4.3 Future work (Technical Aspects)

4.3.1 Database synchronization.

MySQL Lite 2.0 is a free database from Android that weights 150 KB. There is no server, no configuration, it doesn't need administrator, MySQL Lite 2.0 is simple file. Android should work with external databases, and MySQL Lite should be used as a resource to synchronize user data, producing faster interaction with user (talking about business mobile applications)[2].

The problem resides in synchronize MySQL database with external database. The first problem is manipulation of two logic models where consistency, and treatment of current information as recent changes could affect the good performance or desired functionality of an application. This Google document is a first approximation on how to achieve a technical solution but is imperative create transactional rules to keep consistence and to remember that local database must contain uniquely data about current userREST in Android .

4.3.2 Personalization and Adaptation Models.

Every day are more common and more wanted by users the adaptation models. People want to use software in simple and intuitive ways, an to do things once. One challenge in mobile computing is precisely to know your clients, and treat each one as a unique client. Using different theory's and designs developers are understanding and adding features to distinguish each person. This consideration impact the key lines to build new era applications, and specifically some variables could play interesting roles as location web services. The artificial Intelligence is another interesting variable, dealing with agents theory, the context and smart communication between elements in order to provide a great chance to explore the inmanent ubiquity of mobile devices.

4.3.3 Arithmetic Model (distribution of money).

There are multiples articles, books and proposals about distribution of money in a multilevel structure. A complete degree work could be needed to deal with an arithmetic model capable to manage financial movements inside the network. Beyond there, to provide flexible multilevel rules complexity could increase , but in the other hand could be the key to differentiate core business from possible competitors[20, 21, 22].

4.3.4 Distributed Model.

Based on location oriented design S-3R4 database (Oracle 10g Express) was re-designed. Oracle has advanced tools to work with distributed models, so thinking about scalability, multiple users, realistic scenarios and testing consolidations (calculating results from multilevel model) is easier. [23]Tools as Jdeveloper and Oracle database provide flexible configuration and administration (JAVA technology) of several node servers, where is possible to emulate distributed environments.

Glossary

1. API: API, an abbreviation of application program interface, is a set of routines, protocols, and tools structured and developed for building software applications.
2. APK File: Android application package file. Each Android application is compiled and packaged in a single file that includes all of the application's code (.dex files), resources, assets, and manifest file. The application package file can have any name but must use the .apk extension [4].
3. EJB: Enterprise beans are Java EE components that implement Enterprise JavaBeans (EJB) technology. Enterprise beans run in the EJB container, a runtime environment within the GlassFish Server. Although transparent to the application developer, the EJB container provides system-level services, such as transactions and security, to its enterprise beans. These services enable you to quickly build and deploy enterprise beans, which form the core of transactional Java EE applications [24].
4. EclipseLink: The EclipseLink project delivers a comprehensive open-source Java persistence solution addressing relational, XML, and database web services[25].
5. Glassfish Server 3.1.1: A final release implementing the full Java EE 6 standard. Includes an extensible core based on OSGi, with Full Clustering, Centralized Admin, SSH provisioning, High-Availability, Admin Console, Update Center, embedded API, and more. This is fix release for GlassFish 3.1 with multiple component updates (Weld, Mojarra, Jersey, EclipseLink), JDK 7 support, AIX support and more[26].
6. GPS: The Global Positioning System (GPS) is a U.S.-owned utility that provides users with positioning, navigation, and timing (PNT) services. This system consists of three segments: the space segment, the control segment, and the user segment. The U.S. Air Force develops, maintains, and operates the space and control segments.
7. JAVA SE: Java Platform, Standard Edition (Java SE) lets you develop and deploy Java applications on desktops and servers, as well as in today's demanding embedded environments. Java offers the rich user interface, performance, versatility, portability, and security that today's applications require[27].
8. JAVA EE: Java Platform, Enterprise Edition (Java EE) 6 is the industry standard for enterprise Java computing. Utilize the new, lightweight Java

EE 6 Web Profile to create next-generation web applications, and the full power of the Java EE 6 platform for enterprise applications. Developers will benefit from productivity improvements with more annotations, more POJOs, simplified packaging, and less XML configuration[27].

9. JPA: The Java Persistence API provides Java developers with an object/relational mapping facility for managing relational data in Java applications. Java Persistence consists of four areas: The Java Persistence API, the query language, the Java Persistence Criteria API and object/relational mapping metadata [24].
10. PHP: PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML[28].
11. POJO: Plain Old Java Objects. Objects instantiated from a java class characterized by attributes. Objects are read and modified via "getters" and "setters" and may have other business methods.
12. UI-GUI: In computing is an acronym used to describe the human-computer interface (i.e., a way for humans to interact with electronic devices) that uses windows, icons and menus and which can be manipulated with haptics, non haptics devices and other ways as gestures and touch.
13. JAX-RS: RESTful web services are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol[24].
14. JSP: Java Server Pages (JSP) is a technology that lets you mix regular, static HTML with dynamically-generated HTML. Many Web pages that are built by CGI programs are mostly static, with the dynamic part limited to a few small locations. But most CGI variations, including servlets, make you generate the entire page via your program, even though most of it is always the same. JSP lets you create the two parts separately [24].
15. JSON: JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that

are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language[29].

16. **JAVA Servlet:** A servlet is a Java programming language class used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes[24].
17. **JDBC:** Call-level interfaces such as JDBC are programming interfaces allowing external access to SQL database manipulation and update commands. They allow the integration of SQL calls into a general programming environment by providing library routines which interface with the database. In particular, Java based JDBC has a rich collection of routines which make such an interface extremely simple and intuitive[30].
18. **MVC :** By applying the Model-View-Controller (MVC) architecture is possible to separate core business model functionality from the presentation and control logic that uses this functionality. Such separation allows multiple views to share the same enterprise data model, which makes supporting multiple clients easier to implement, test, and maintain [27].
19. **NFC:** Near Field Communication (NFC) is a set of short-range wireless technologies, typically requiring a distance of 4cm or less to initiate a connection. NFC allows you to share small payloads of data between an NFC tag and an Android-powered device, or between two Android-powered devices [4].
20. **URI's:** Uniform Resource Identifiers (URIs, aka URLs) are short strings that identify resources in the web: documents, images, downloadable files, services, electronic mailboxes, and other resources. They make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail addressable in the same simple way. They reduce the tedium of "log in to this server, then issue this magic command ..." down to a single click. It is an extensible technology: there are a number of existing addressing schemes, and more may be incorporated over time[31].

Acknowledgments

Thanks to NextLab for their support, to work with talent people disposed to help you and thanks for all opportunities and links with enterprise incubators as Wayra (Telefonica Project). Thanks to Pontificia Universidad Javeriana for these years of learning and the chance to explore the the world of informatics and engineering. Thanks to professor Oscar Chavarro, incredible human being, someone with great spirit, with amazing knowledge. Thanks to Efrain Ortiz Pabon for his business orientation and to push innovation in my daily job. Efrain help students to believe they can create their own company's. Thanks to Juan Pablo Garzon, a restless character in mobile area, without him this degree work would not be possible. Thanks to professor Maria Consuelo Frankie, for all her contributions with Java Enterprise Edition, and her patience.

Bibliography

- [1] R. Harper, “Warehouse technology in the supply chain management systems,” in *Reliability and Maintainability Symposium (RAMS), 2010 Proceedings - Annual*, jan. 2010, pp. 1 –5.
- [2] (2011, December) MIT Lab for Mobile research. [Online]. Available: <http://nextlab.mit.edu/>
- [3] J. Jarvis, *What Would Google Do?*, H. E-books, Ed. HarperCollins E-books, 2009.
- [4] (2011, December) Google. [Online]. Available: <http://developer.android.com/index.html>
- [5] Q. Yuan, Z. Xiaokang, and Z. Qiong, “Key technology and system design in mobile supply chain management,” in *Electronic Commerce and Security, 2008 International Symposium on*, aug. 2008, pp. 258 –262.
- [6] J. Luo, “Utilizing mobile web services for supply chains,” in *Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference on*, june 2006, pp. 887 –892.
- [7] R. Dass and R. Muttukrishnan, “Security framework for addressing the issues of trust on mobile financial services,” in *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on*, oct. 2011, pp. 99 –104.
- [8] R. Di Pietro, G. Me, and M. Strangio, “A two-factor mobile authentication scheme for secure financial transactions,” in *Mobile Business, 2005. ICMB 2005. International Conference on*, july 2005, pp. 28 – 34.
- [9] S. Min and C. Fei, “Research of mobile banking service in china: A cross rode of communication technology and financial innovation,” in *E-Business and Information System Security, 2009. EBISS '09. International Conference on*, may 2009, pp. 1 –5.
- [10] Y. Zhiqiang, A. Sen, and X. Han, “Dynamic financial application framework for the mobile phones,” in *Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on*, sept. 2010, pp. 45 –48.
- [11] A. Q. Gill, D. Bunker, and P. Seltsikas, “An empirical analysis of cloud, mobile, social and green computing: Financial services it strategy and enterprise architecture,” in *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, dec. 2011, pp. 697 –704.

-
- [12] U. Varshney, "Vehicular mobile commerce," *Computer*, vol. 37, no. 12, pp. 116 – 118, dec. 2004.
- [13] P. S. Javid and T. S. Parikh, "Augmenting rural supply chains with a location-enhanced mobile information system," in *Information and Communication Technologies and Development, 2006. ICTD '06. International Conference on*, may 2006, pp. 110 –119.
- [14] W. Huang and Z. Liu, "Research on supply chain model based on multi-agent," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, oct. 2008, pp. 1 –4.
- [15] D. S. W. Alliance, *Build it Big*, D. S. W. Alliance, Ed. Dearborn Trade Publishing A Kaplan Professional Company, 2005.
- [16] (2011, December) Direct Sell Association. [Online]. Available: <http://www.avd.es/asociados/index.html>
- [17] G. Mulligan and D. Gracanin, "A comparison of soap and rest implementations of a service based interaction independence middleware framework," in *Simulation Conference (WSC), Proceedings of the 2009 Winter*, dec. 2009, pp. 1423 –1432.
- [18] W.-M. Lee, *Beginning Android Application Development*, I. Wiley Publishing, Ed. Wiley Publishing, Inc, 2011.
- [19] S. Glissmann, S. Smolnik, R. Schierholz, L. Kolbe, and W. Brenner, "Proposition of an m-business procedure model for the development of mobile user interfaces," in *Mobile Business, 2005. ICMB 2005. International Conference on*, july 2005, pp. 308 – 314.
- [20] M. Oudin and J. Delmas, "Asymptotic generalized eigenvalue distribution of block multilevel toeplitz matrices," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 382 –387, jan. 2009.
- [21] J. Jingxi, B. Veeravalli, and D. Ghose, "Adaptive load distribution strategies for divisible load processing on resource unaware multilevel tree networks," *Computers, IEEE Transactions on*, vol. 56, no. 7, pp. 999 –1005, july 2007.
- [22] M. Aguirre, L. Calvino, V. Corasaniti, and M. Valla, "Multilevel current source inverter to improve power quality in a distribution network," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, july 2010, pp. 3292 –3297.
- [23] (2011, December) ORACLE. [Online]. Available: <http://www.oracle.com/index.html>
- [24] (2011, December) JAVA EE 6 developers. [Online]. Available: <http://docs.oracle.com/javaee/6/tutorial/doc/>
- [25] (2011, December) Eclipse. [Online]. Available: <http://www.eclipse.org/eclipselink/>

- [26] (2011, December) JAVA. [Online]. Available: <http://glassfish.java.net/>
- [27] (2011, December) JAVA developers. [Online]. Available: <http://www.oracle.com/technetwork/java/index.html>
- [28] (2011, December) PHP developers. [Online]. Available: <http://www.php.net/>
- [29] (2011, December) JSON. [Online]. Available: <http://www.json.org/>
- [30] (2011, December) Stanford University. [Online]. Available: <http://infolab.stanford.edu/~ullman/fcdb/oracle/or-jdbc.html>
- [31] (2011, December) World Wide Web Consortium. [Online]. Available: <http://www.w3.org/>

