



UNIVERSITY OF BEIRA INTERIOR
Faculty of Engineering
Department of Informatics

Ubiquitous Computing Approach for Mobile Health Applications

Ivo Miguel Charro de Matos Lopes

Submitted to the University of Beira Interior in candidature for the
Degree of Master of Science in Informatics Engineering

Supervised by Prof. Doutor Joel José Puga Coelho Rodrigues

Department of Informatics
University of Beira Interior
Covilhã, Portugal
<http://www.di.ubi.pt>

Acknowledgements

First of all, I would like to thank Professor Joel José Puga Coelho Rodrigues for giving me the chance to join the Next Generation Networks and Applications Research Group (NetGNA), for all the constant words of encouragement and for supervising my Master's Thesis. These two years were undoubtedly very important in my evolution as an informatics Engineer and person as well.

I am most grateful to the University of Beira Interior, the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, Portugal, and SAPO - Portugal Telecom for allowing me to work on this project with them.

Many thanks to all members of NetGNA, João Dias, João Isento, Bruno Silva and Pedro Rosa.

And of course, a enormous thanks to my family and my girlfriend, Vanessa Filipa dos Santos Serra, for their support and for the great help that was extremely important.

Abstract

The high penetration of mobile devices and networks globally implies that mobile technology could be used very effectively in the health field in order to compensate for the lack of resources problem, particularly in developing countries. With the proliferation of mobile technologies, mobile health (m-health) will play a vital role in the rapidly growing electronic health (e-health) area. The form of transparency between health and patients is a need to compensate for the lack of medical resources.

E-health solutions, provides patients with mobile services to support and optimize their treatment based on the monitoring of certain physiological parameters. This category of e-health is called mobile health monitoring.

Obesity is a big public health challenge and its prevention requires patients with healthy eating and physical activity awareness in their daily lives. It is estimated that over 50% of world population will be obese by 2025 if no action is taken otherwise. Therefore, obesity is rightly regarded by the World Health Organization as the global epidemic of the XXI century. In the European Union, including Portugal, the incidence and prevalence of either pre-obesity or obesity have been rising as an important public health problem.

In Portugal, the current situation presents high prevalence of obesity and its incidence, morbidity and mortality is also increasing. Furthermore, people usually use mobile phone in their daily life. Then, it creates an opportunity to create a mobile solution for personal support to reduce

obesity. This work proposes the creation of a mobile health (m-health) system for Android operating system using ubiquitous computing approaches. A solution, called *SapoFitness*, is proposed for a dietetic monitoring and assessment, and will be carried out with SAPO - Portugal Telecom. *SapoFitness* is customized per user to keep a daily personal health record (PHR) of his/her food intake and daily exercise. This PHR contains vital health information and will evaluate the nutritional state of the user. The system monitors the user and sends alerts/messages concerning his/her diet program taking into account also his/her physical activity. *SapoFitness* is a challenged mobile application that delivers the action to the user, anytime and anywhere, motivating him for a healthier life style.

The system will use the Web Services architecture, which appears as a new model for distributed computing, represents the latest proposed solution for development and application integration. Web services come with a promise to definitely bring down the barrier of communication between heterogeneous platforms, revolutionizing the way people develop applications targeted for the Internet, transforming it into a software architecture, service-oriented.

SapoFitness was evaluated in different mobile devices (Sony Ericsson x10 mini, HTC Magic, TMN a1 and Samsung Galaxy Tab), taking into different screens sizes. The system also was been evaluated and validated by SAPO team and it is ready for use.

Keywords

Mobile Computing, Mobile Health, Mobile E-Health, Mobile U-Health, PHR, EHR, Healthcare Application, Mobile Monitoring Application, Mobile and Pervasive Computing, Android, Weight Loss, Obesity, Services, Web Services.

Contents

Acknowledgements	i
Abstract	iii
Keywords	v
Contents	vii
List of Figures.....	ix
List of Tables.....	xi
Acronyms	xiii
1. Introduction	1
1.1 Focus	1
1.2 Objectives	9
1.3 Main Contributions	10
1.4 Dissertation Structure	10
2. Related Work	11
2.1 Electronic Health (E-Health).....	11
2.2 Mobile Health (M-Health).....	13
2.3 Electronic Health Record (EHR) and Personal Health Record (PHR)	16
2.4 Ubiquitous Health (U-Health).....	18
2.5 Obesity	20
3. Requirements Analysis	23
3.1 Design and application of SapoFitness	24
3.1.1 SapoFitness System Architecture	24
3.2 Essential requirements	26
3.3 Used technologies	27
3.3.1 Android GUI Architecture	31
3.4 Behavioral Diagrams	32
3.4.1 Use Case Diagrams	32
3.4.2 Activity Diagrams.....	33

3.5	Interaction Diagrams	37
3.5.1	Sequence Diagrams.....	37
3.6	Structural Diagrams	39
3.6.1	Class Diagram	39
4.	Web Services	43
4.1	Styles of use	44
4.1.1	RPC - Remote procedure calls.....	44
4.1.2	SOA - Service-oriented architecture.....	46
4.1.3	REST - Representational state transfer	48
4.2	RESTful <i>versus</i> SOAP	49
4.3	Markup Languages	55
4.3.1	Extensible Markup language (XML).....	55
4.3.2	JavaScript Object Notation (JSON)	56
4.4	JSON versus XML	57
5.	System Demonstration and Validation.....	65
5.1	Application Demonstration.....	66
5.2	Application Validation	74
5.2.1	Comparison between Web services	75
5.2.2	User Survey.....	77
6.	Conclusions and Future Work.....	79
6.1	Conclusions	79
6.2	Future Works	81
	References	83

List of Figures

Figure 1. Number of devices OS Sales to End-users by OS.....	2
Figure 2. Growth of health applications by platform over a six-month period in 2010.	3
Figure 3. Illustration of the two types of obesity.	7
Figure 4. SapoFitness System Architecture.	25
Figure 5. M-Health services framework.	26
Figure 6. Android system architecture [66].	29
Figure 7. Android GUI architecture.	31
Figure 8. User Case Diagram of SapoFitness.....	32
Figure 9. Activity diagrama of SapoFitness.....	33
Figure 10. Decision algorithm for the meal.	34
Figure 11. Decision algorithm for weight.	35
Figure 12. Decision algorithm for intake and energy needs.	36
Figure 13. Sequence Diagram - Login.	37
Figure 14. Sequence Diagram - Edit Profile.	38
Figure 15. Sequence Diagram - Social Network.	38
Figure 16. Sequence Diagram - Physical Activity.	39
Figure 17. Sequence Diagram - Add Meal.	39
Figure 18. Class diagram of the Windows.....	40
Figure 19. Class diagrama of the connection with Web services.....	41
Figure 20. WCF endpoint - ABC (Address, Binding and Contract).	47
Figure 21. Message Size (Bytes) of String Concatenation and float addition service in SOAP and REST.	54
Figure 22. Service Response Time (Milliseconds) of String Concatenation and float addition service in SOAP and REST.....	54
Figure 23. Splash and Pre-Login Screens.	66
Figure 24. Login Screen.	67
Figure 25. Profile Screen - Actual Weight.	68
Figure 26. The Main Tab	68
Figure 27. Profile Screen - Ideal Weight.	69

<i>Figure 28. Edit Profile.</i>	<i>70</i>
<i>Figure 29. Plan Screen.</i>	<i>71</i>
<i>Figure 30. Evolution Screen.</i>	<i>72</i>
<i>Figure 31. All four types of graphics Screens.</i>	<i>72</i>
<i>Figure 32. Physical Activities Screen.</i>	<i>73</i>
<i>Figure 33. SapoFit in three different devices.</i>	<i>74</i>
<i>Figure 34. Webservice Request Time (Milliseconds).</i>	<i>76</i>
<i>Figure 35. JSON and XML Message Length (Bytes).</i>	<i>76</i>
<i>Figure 36. Results of SapoFitness Survey.</i>	<i>78</i>

List of Tables

<i>Table 1. Request Comparison to Web service.</i>	<i>75</i>
<i>Table 2. SapoFitness Survey Questions.</i>	<i>77</i>
<i>Table 3. Table of SapoFitness survey.</i>	<i>78</i>

Acronyms

ACL	:	Access Control List
ADT	:	Android Development Tools
API	:	Application Programming Interface
BMI	:	Body Mass Index
BMI	:	Body Mass Index
CDC	:	Center for Disease Control and prevention
CORBA	:	Common Object Request Broker Architecture
DCOM	:	Distributed Component Object Model
DLL	:	Dynamic-Link Library
DOM	:	Document Object Models
DTD	:	Document Type Definition
E4X	:	ECMAScript for XML
ECG	:	Electrocardiogram
EHR	:	Electronic Health Record
GPS	:	Global Position System
HTTP	:	Hypertext Transfer Protocol
IDE	:	Interface Development Environment
IEEE	:	Institute of Electrical and Electronics Engineers
IP	:	Internet Protocol
JSON	:	JavaScript Object Notation
LDAP	:	Lightweight Directory Access Protocol
MIME	:	Multipurpose Internet Mail Extensions

NHANES	:	National Health and Nutritional Examination Survey
OMG	:	Object Management Group
PDA	:	Personal Digital Assistant
PGS	:	Parking Guidance System
PHR	:	Personal Health Record
REST	:	Representational State Transfer
RMI	:	Remote Method Invocation
RPC	:	Remote Procedure Call
RPM	:	Remote Patient Monitoring
RUP	:	Rational Unified Process
SDK	:	Software Development Kit
SGML	:	Standard Generalized Markup Language
SOA	:	Service-Oriented Architecture
SOAP	:	Simple Object Access Protocol
TDI	:	Total Day Intake
UML	:	Unified Modelling Language
W3C	:	World Wide Web Consortium
WCF	:	Windows Communication Foundation
WHO	:	World Health Organization
WSDL	:	Web Services Description Language
XML	:	Extensible Markup Language

1. Introduction

1.1 Focus

Health telematics is offering less expensive solutions in healthcare services. Patients that live in remote rural areas, that travel constantly, in case of accident scenes, and among other scenarios, are often physically inaccessible to receive any kind of health monitoring or treatment [1]. E-health brings a new hopefulness for such patients with more accessible and affordable healthcare solutions [2],[3]. In the last decade, with the advent of mobile communications supported on smart mobile devices, mobile computing has been increasing lots of attention from the research and business communities [4]. Thus, it offers many opportunities to create efficient mobile telemedicine applications. Mobile health (m-health) is the new edge in healthcare innovation. It proposes to deliver health-care anywhere and anytime, surpassing geographical, temporal, and even organizational barriers [5], [6]. Physicians could download medical records, lab results, images, and drug information to handheld devices like personal digital assistants (PDAs) and smartphones. Patients could be aware of their diagnostic, disease control, and monitoring with comfortable mobile devices that accompany them everywhere. M-health systems, and its inherent mobility functionalities, have a strong impact on typical healthcare monitoring and alerting systems, clinical and administrative data collection, record maintenance, healthcare delivery programs,

medical information awareness, detection and prevention systems, drug-counterfeiting, and theft [7].

In January 9 of 2007 Steve Jobs, CEO of Apple Inc., present to the world the iPhone 2G. This event triggers a rapidly evolution of smartphones and also the appearing of new mobile platforms, especially one that proof to be market dominant the Google Android. Basically, the following five players dominate mobile OS market, Google Android, Nokia Symbian, Apple iOS, BlackBerry RIM and Windows Phone 7. Figure 1 presents the number of device OS sales to end-users over the last three years. Clearly, Google Android and Apple iOS dominate the OS market. The quality of both operating system, are unquestionable, however both companies success in the mobile market is sustained by their application market, Apple app store and Google market. According to Gartner press release, worldwide mobile application store downloads are forecast to reach 17.7 billion in 2011, a 117% increase from an estimated 8.2 billion downloads in 2010 [8].

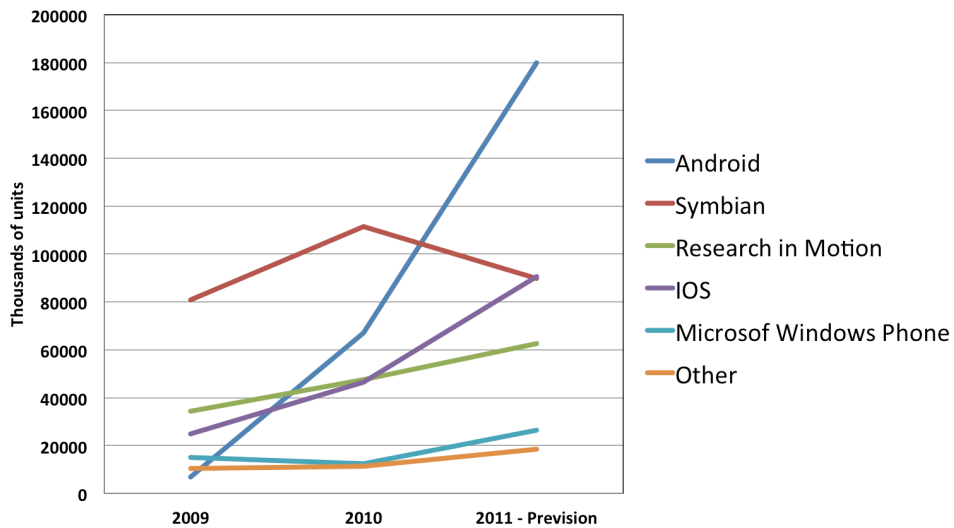


Figure 1. Number of devices OS Sales to End-users by OS.

Currently, there are dozens of app stores for different platforms such as Android Market, Blackberry App World, Apple App Store, Ovi Store, Samsung Apps and Windows Marketplace just to name a few. These market

opportunities open new and potential areas of research, such as mobile health applications. At the end of 2010, more than 200 million m-health applications were in use and about 70% of worldwide citizens were interested in having access to at least one m-health application. Overall smartphone's web browsers improved becoming easier to find free applications and information. Beside the available application markets, a user can now download free mobile health apps. Figure 2 presents the growth of health applications in the three main platforms over a six-month period in 2010 [9].

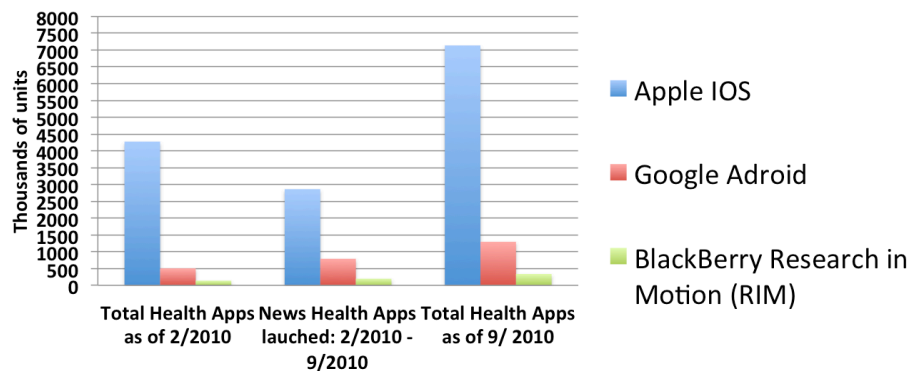


Figure 2. Growth of health applications by platform over a six-month period in 2010.

Most of the current market applications are directed toward patients, clinicians and healthcare professionals. These applications are mainly suited for diseases management, self-monitoring and drug control as well as other clinical and educational applications. Excluding the free medical applications the average price of consumer health and medical paid apps in 2010 on Apple's AppStore was \$3,63 and healthcare professional apps were \$14.23. Android Market average prices are very similar to Apple's AppStore, with the average most expensive health application costing \$13.11. Indeed the most expensive market is the Blackberry App World having the most expensive health application costing

an average of \$37.87 [9]. The low cost of these medical and health applications are an advantage and potential investment for consumers and healthcare professionals.

According to the Global Mobile Health Market Report 2010-2015 by research2guidance, in 2015 about 500 million people will be using mobile health applications [10]. This raises several important and complex questions about these medical applications, such as, security, reliability, efficiency and quality of service. These applications can really perform a complete, secure, reliable and efficient diagnostic? The fact is that there are already many mobile health apps making claims such as, “this app will lower your blood pressure” or “this app will help you to lose weight”, are these claims trustworthy? To protect mobile health application users, the U.S. Food and Drug Administration, from the Department of Health & Human Services, enforces regulations on medical device approval and clearance. Devices manufacture must first register and notify the FDA their intent to market a medical device. This is known as 510(k) clearance and allows the FDA to determine if the device is valid and if is equivalent to a device already place in the market. The next step is the premarket approval (PMA). The PMA is the most rigorous approval of a request submitted to the FDA to market. This approval is based on the valid scientific evidence that assures that the proposed device is safe and effective for its intended use. FDA regulation also allows the submission of a Humanitarian Use Exception (HDE) approval. It applies to Humanitarian Use Devices (HUDs). HUDs are intended to care patients by treating or diagnosing a disease that affects less than 4,000 people in the U.S. per year. The HDE approval is very similar to the PMA, except the effectiveness requirements [11]. FDA also defines the term “device” in section 201(h) as:

“... an instrument, apparatus, implement, machine, contrivance, implant, in vitro reagent, or other similar or related article, including any component, part, or accessory, which is ... [either] intended for use in the diagnosis of disease or other conditions, or

in the cure, mitigation, treatment, or prevention of disease, in man or other animals ... [or] intended to affect the structure or any function of the body of man..."

The FDA regulations on devices are very clear. But the regulation of mobile health application is yet to be well defined. Another important question that FDA must resolve is the definition of "medical device". A device plus a health-related application results in a medical device? If its so, it seems that smartphones, that are already used by physicians for in numerous health purposes, can be classified as medical devices.

Usually, healthcare providers keep and maintain patient health records. However, it is becoming more common that patients also request access to these data. Medical records (or health records) allow medical doctors to easily access patient information without needing to ask them in person. E-health systems are typically sustained on electronic health records (EHR) [7]. An EHR-System is basically a repository of information regarding the health records of patient/consumer in a computer form[12]. A mobile EHR-System (mEHR-System) complements a typical EHR-System enabling access to health information regardless of the patient time and place. M-health systems use the Internet and Web services to provide an authentic pervasive interaction among doctors and patients. A physician or a patient can easily access the same medical record any tome, any where through his personal computer or his smartphone.

A big problem nowadays is obesity in the population, and the EHR has been focused on this area, obesity is a disease with an important risk factor [13] for the emergence, development and worsening of other diseases. There are so many obese people worldwide than the World Health Organization (WHO) declared the disease as a global epidemic of the XXI century [14]. The Center for Disease Control and prevention (CDC) and the U.S. National Health and Nutrition Examination Survey (NHANES), indicate that an estimated 17 percent of children and adolescents between ages 2-19 years are obese [15]. The main treatment for obesity includes dieting

and frequent physical activity. Diet programs keeps and cause weight loss over short, medium or long term. However, it is necessary frequent physical exercise to maintain balanced body energy. Nevertheless, to maintain such treatments it is crucial having a strong individual discipline, motivation, and constant monitoring of its food intake [14].

According to WHO, obesity is a disease in which excess body fat build-up may reach levels that can affect health. It is a chronic disease, with marked prevalence in developed countries, affects men and women of all ethnicities and all ages, reduces the quality of life and has high rates of morbidity and mortality. Obesity causes serious health consequences[14].

There are two types of obesity [16], the android obesity, abdominal or visceral (apple) and the Obesity-type gynoid (pear) (Figure 3). The android obesity is when fat accumulates in the upper body, especially in the abdomen. It is typical of the obese man. Visceral obesity is associated with metabolic complications such as type 2 diabetes and dyslipidemia, and cardiovascular diseases such as hypertension, coronary heart disease and cerebrovascular disease as well as polycystic ovary syndrome and endothelial dysfunction (i.e. deterioration the inner lining of blood vessels). The association of obesity to these diseases is dependent on intra-abdominal fat rather than total body fat. The Obesity-type gynoid is when fat is distributed mainly in the lower half of the body, particularly in the buttocks and thighs. It is typical of the obese woman.

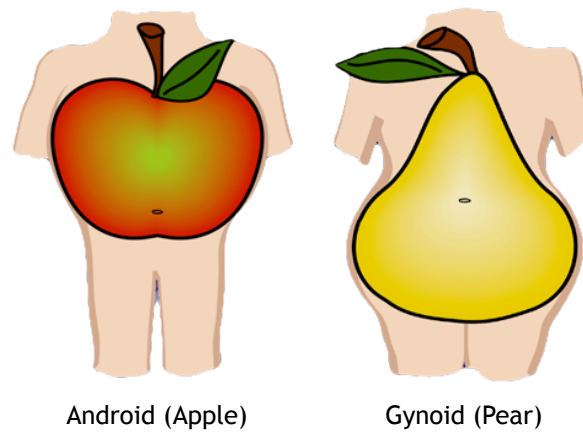


Figure 3. Illustration of the two types of obesity.

The cause of obesity is the excess fat results of successive positive energy balance, where energy intake exceeds the amount of energy expended. The factors that determine this imbalance are complex and can be genetic, metabolic, environmental and behavioural.

A diet with excess fat, carbohydrates and alcohol, combined with a sedentary lifestyle, leads to the accumulation of excess body fat. There is scientific evidence that suggest a genetic predisposition that determines, in some individuals, a greater accumulation of fat in the abdominal area, in response to excess energy intake and / or decreased physical activity.

The risk factors of obesity [17] are the following:

- Sedentary lifestyle - the more hours of television, electronic games or computer games, the higher the prevalence of obesity;
- Zone townhome - the more urbanized area of residence is the greater the prevalence of obesity;
- Grade Parents' information - the smaller the degree of parental information, the greater the prevalence of obesity;
- Genetic factors - the presence of genes involved in increased susceptibility to weight increases risk for developing obesity when the individual is exposed to environmental conditions that favor, which means that obesity is a familial tendency;

- Pregnancy and menopause can contribute to increased fat storage in women with excess weight.

Obesity brings health consequences [18] such as:

- Cardiovascular - hypertension, atherosclerosis, congestive heart failure and angina pectoris;
- Metabolic complications - hyperlipidaemia, abnormal glucose tolerance, type 2 diabetes, gout;
- Pulmonary System - dyspnea (difficulty breathing) and fatigue, respiratory distress syndrome of obesity, sleep apnea (snoring) and pulmonary embolism;
- Gastrointestinal Unit - fatty liver, gallstones (formation of sand or small stones in the gallbladder) and colon cancer;
- Genito-urinary and reproductive - infertility and amenorrhea (absence of abnormal menstruation), stress urinary incontinence, hyperplasia and endometrial carcinoma, breast cancer, prostate cancer, hypothalamic hypogonadism and hirsutism;
- Other changes - osteartroses, chronic venous insufficiency, anesthetic risk, hernias and propensity to falls.

Obesity also causes changes socio-economic and psychosocial factors, discrimination in education, difficulty of employment, social isolation, depression and loss of self-esteem.

Assessment of diet among adolescents is problematic. Early adolescents, ages 11 to 14 years, in particular, are in that period of time when the novelty and curiosity of assisting in or self-reporting of food intakes starts to wane and the assistance from parents is seen as an intrusion [19]. Dietary assessment methods need to continue to evolve to meet these challenges. There is recognition that further improvements will enhance the consistency and strength of the association of diet with disease risk, especially in light of the current obesity epidemic among this group. Preliminary studies among adolescents suggest that innovative use of technology may improve the accuracy of diet information from young people.

1.2 Objectives

The main objective of this dissertation is the design, constructing, deployment, and performance evaluation of a mobile system solution for Android to users control their weight and calories. An important aspect of this system is the usability and embedded ubiquitous approaches.

This system will have to use Web services for communicate with the database. The application to develop for mobile devices will be a diet daily diary, in which the user inserts the meals eaten through the day, allowing the system to monitoring in real time the calories eaten by the user, and at the same time the calories burned when is practice a physical activity.

To reach this main objective the following intermediate objectives were define:

- Study of the related work, about e-health, m-health, PHR/EHR and obesity;
- Detailed analysis of the Web services architecture;
- Proposal, implementation and validation of the system based on requirement analysis;
- Construction and deployment of a mobile API on Android OS;
- Performance evaluation and system validation on different Android devices with different versions of OS.

This work of research and engineering is expected to produce not only the desire API, but also to provide the opportunity to disseminate the knowledge and software through, at least, a conference paper.

1.3 Main Contributions

This section is devoted to the scientific contributions of this dissertation to the state-of-the-art on Electronic Health, Mobile Health, Ubiquitous Health, and Obesity. The main contribution is SapoFitness: A Mobile Health Application for Continuous Monitoring Dietary Evaluation. This proposal was presented at the IEEE HEALTHCOM 2011, Columbia, MO, USA, June 13-15, 2011.

1.4 Dissertation Structure

This dissertation is organized in six chapters. This chapter, the first, presents the context of the dissertation, focusing on the topic under study, the objectives, the main contributions and the dissertation structure and its main contribution.

Chapter 2 - Related Work - Presents the literature review on obesity and E-Health combined with mobile computing. Next most related and relevant works emphasizing M-health in obesity. Finally, an introduction to our application is presented.

Chapter 3 - Requirements Analysis - This chapter presents all the requirements analysis for SapoFitness.

Chapter 4 - Web Services- This chapter introduces and discusses the Web services for communication between the system and database that contains user data.

Chapter 5 - System Demonstration and validation - This chapter presents the conceptual design, used technologies and all the performance done to the SapoFitness.

Chapter 6 - Conclusions and Future Work - Concludes the dissertation and presents a few remarks for future work.

2. Related Work

This chapter addresses the current state of art on e-health and m-health. The third and fourth section of this chapter describes some related work in PHR/EHR and u-health respectively. And the fifth section talks about the state of art on obesity.

Increasingly, mobility is part of our life, every day there are new technologies that can be applied to a range of applications, but anything can serve if they do not contain the mobility that is necessary today. Mobile computing embraces a host of portable technologies that makes Internet access on the go, not only possible, but integral to every day life.

2.1 Electronic Health (E-Health)

E-health is an emergent field in the connection of medical informatics, public health and business, denoting to health services and information delivered or enhanced through the Internet and associated technologies.

In [20] the authors proposes an improved version of a music video game, called *Dance Dance Revolution*, to stimulate the older population and increase adherence to rehabilitation. They also present the system design for a mobile monitoring application to allow health professional care. Several mEHR-Systems exist on the market but they differ from witch others in functionality, this inhibited a global assessment of m-health, this

approach is presented to fill this gap, by gave a functional specification for mEHR-Systems.

A network multi-agent architecture for mobile health monitoring is presented in [21], where a team of intelligent agent collect patient data, aim collectively and recommend actions to patients and health personal in a wireless mobile environment. The architecture offers the base for the use of intelligent agents to deliver better healthcare to patients with chronic illnesses.

The paper [22], *Mobile Technologies for Enhancing eHealth Solution in Developing Countries*, describes the applicability of available mobile infrastructures and technologies in the healthcare to propose an m-Health solution to suit in developing countries and a form of transparency between patient and healthcare by using the infrastructures.

Jan *et al.* presents a study of different routing protocol's performance in a mobile ad-hoc grid environment [23]. The best routing is the requirement of a grid application, such as quick response time, routing overhead and quick and reliable responses. All the tests were made on a test scenario of e-health application in a mobile ad-hoc grid background.

In [24], the authors have suggested a new security mechanism for online communications using mobile nodes. The Internet makes important to ensure the security of the communication, safe and maintained the user's privacy. This work extends a previous research, Multilayer Communication (MLC) approach [25], where is presented a different way to secure communications in e-health, by using different types of security levels. And presented mobile nodes to MLC in e-health, using cryptographic protocols such as digital signature, encryption/decryption and hash code tables.

2.2 Mobile Health (M-Health)

In [26] the authors suggest a prototype system that uses a mobile phone with built-in camera, network connectivity, and an integrate image analysis to provide an accurate account of daily food and nutrient intake. Their objective is to use visualization tools with databases to compare and save eaten food. Acquired images are used to approximate the quantity of food and nutrients consumed.

A mobile phone application for real time monitoring caloric balance, called Patient-Centered Assessment and Counseling Mobile Energy Balance (PmEB), is presented in [13]. This approach permits users to self-monitor caloric balance in real time. The PmeB is a client mobile application and uses a Web interface that allows users to register and personalize the application. The application allows the user interaction with the PmEB system. The server application sends updated caloric reminders to the client, stores food, and activity information, keeping data updated about the users daily calorie consumed.

The approach called StepUp [27] a step counter application, presented by A. Khalil *et al.*, uses sensor-enabled mobile phones to automatically count the number of steps walked by the user. Its main goal is returning to the user the measured quantity of his/her daily activities and create a healthy competition which serves has a source of positive feedback. The StepUp application also aims to increase the users awareness and understanding the importance of physical activities and simplify the integration of regular exercise into their daily life.

A mobile phone short message service messaging for behaviour modification in a community-based weight control program in Korea is presented in [28]. Mobile phones were used to deliver weekly short message service (SMS) messaging about weight loss. It delivers information about diet, exercise, and behaviour modification once a week. A total of 927 participants that visited a public healthcare center tested this service.

Post-results showed that the majority of participants were satisfied and attested that this service may be an effective method of behaviour modification in weight control.

K. Patrick *et al.* also presents a text message-based intervention for weight loss [29]. This study describes the development and evaluation of a text message-based intervention intended to help a person to lose or maintain weight in a 4 months period. This service includes personalized SMS and multimedia message service (MMS) sent two to five times daily, printed materials, and brief monthly phone calls from a health counsellor. Post-results showed that 92% of participants would recommend this intervention for weight control to friends and family. Proving that text-based services might be a productive channel of communication to support weight loss.

The [30] presents MyDS - My Dietary Supplements, an iPhone application that allows an easy way to keep track of the vitamins, minerals, herbs, and other products intake. Another approach for iPhone is presented in [31], AIDSinfo HIV/AIDS Glossary, designed to help health care providers, caseworkers, community-based organization professionals, and people living with HIV/AIDS or even their families and friends to help them understand the complex HIV/AIDS terminology. Yet another iPhone application finds federally funded health centers near the user current location [32].

The authors of [33], presented a mobile Web interface, MedlinePlus Mobile, that browses health information to find important drug information, and other health topics.

An iPhone application, BMI calculator, is one of the most popular tools from the National Heart, Lung, and Blood Institute (NHLBI) Web site. BMI is an indicator of total body fat, which is related to the risk of disease and death. It receives 1.6 million visitors a month and ranks #1 on Google [34].

The Harvard Medical School (HMS) is encouraging students to buy a mobile device of their choice, iPhone, iPad, Android, Blackberry, etc... The

School provides to those devices software licenses and controlled host applications for medical education purposes [35]. According to the Chief Information Officer (CIO) and Dean for Technology at HMS, Dr. John Halamka, the top five mobile applications downloaded from the HMS - mobile resources web page, are the following:

- **Dynamed:** A clinical reference tool created by physicians for point-of-care situations. Designed for physicians and other healthcare professionals with clinically organized summaries for more than 3,200 topics. Dynamed is updated daily and monitors the content of over 500 medical journals [36].
- **Unbound Medicine uCentral:** Unbound Medicine delivers a wide range of medical customizable solutions for nursing schools, publishers/associations, medical schools, residency programs, departments, pharmaceutical or medical device companies and hospitals or health systems. These health-related solutions are available for almost all mobile platforms and also the Web [35]. uCentral is a completely customizable mobile and web application that delivers answers about clinical references to the point of need. Clinicians, students and researchers through their mobile devices can easily answer clinical questions [37].
- **VisualDx Mobile:** VisualDx provides physician-reviewed clinical information with thousands of medical images. It's the only medical application showing the variation of disease presentation through age, stage, and skin type. This application allows a visual validation of a diagnosis by comparing medical images, allows a quick search by disease for next steps on patient care and provides a on-the-spot patient education with real medical images [38].
- **Epocrates Essentials:** Epocrates is the #1 mobile medical application among U.S. physicians. More than 1.3 million healthcare professionals including 45 percent of physicians use Epocrates to improve patient care and efficiencies with its drug reference and clinical/educational applications. Basically the application is a mobile guide to drugs and disease with an integrated and comprehensive search tool for diseases, infectious diseases, medications, diagnosis, laboratory tests and resource centers. Epocrates is available for almost every mobile platform but also for the desktop market and the Web [39].

- **iRadiology:** This Iphone/Ipad application it's a free learning tool for medical students and residents. iRadiology provides quick reviews of classic radiology cases and images including more than 500 radiology cases to improve skills to interpret plain film, CT and MRI readings [40].

2.3 Electronic Health Record (EHR) and Personal Health Record (PHR)

An Electronic Health Record (EHR) is an evolving concept defined as a systematic collection of electronic health information about patients [41]. It is a digital record that is capable of being share across different health care systems through network connections. Such records can include a full range of data in comprehensive or summary form, including medical history, demographics, medication and allergies, laboratory test results, immunization status, radiology, vital signs, personal stats like age and weight, and other information.

It's a complete record of patient encounters that allows the automation and restructuring of the workflow in health care and increases safety through evidence-based decision support, quality management and outcomes reporting [42].

A Personal Health Record (PHR) is typically a health record maintained by its user. An ideal PHR would provide a complete and accurate summary of the health and medical history of a user by gathering data from diversification sources and making this information accessible to anyone who has the necessary electronic credentials to view the information.

Mosmondor *et al.* presents a study for integrate the remote patient monitoring (RPM) solution into an EHR system [43]. This solution grabs the CEN based EHR specification and propose numerous extensions to incorporate data acquired from the RPM system, including more details about procedures, such as which types of measurement devices were used, in what period, etc.

In [44] is describing the implementation and evaluation of the performance of an electronic health ECG monitoring system using mobile phones with Bluetooth.

Belsis *et al.* presents the wireless architecture W-EHR[45], that enables authorized medical personnel to access medical records securely, using an agent based architecture. The architecture has ubiquitous access to medical information through mobile devices, security management using hierarchical policy based, and encryption for all transactions using a hybrid approach that employs asymmetric and symmetric encryption techniques.

A mobile phone application that integrates EHR with a parking guidance system (PGS) is presented in [46]. The application provides advanced services for two healthcare groups, the patients and the healthcare providers. The zip code is one of the necessary information for the patient's basic data, the application is capable to return the three most optimal urgent care providers in the selected area, the user chose one, the patient's data is submitted, and the system returns navigation information from home to hospital, to hospital's parking facility and inside hospital, furthermore, the application would reserve a parking space and a hospital arrival registration.

The study [47], have build an mPHR system for diabetics, so that can been use mobile phones to upload their own health records and access their data online. They used questionnaires to measure the acceptance of mPHR system, to understanding the behavior between diabetics using mPHR system before and after.

In [48], is propose a new architecture, Privacy Portable Health Record (P3HR), to bridge the gap between the data in PHR and what patients say they need in PHR. A device that incorporates a smart card into a USB flash driver that provides encryption data. The strong aspects of P3HR are, the strong multifactor based authentication-using biometrics, encryption, SSL based authentication protocol and backup, to fight against theft, lost device, failures and virus.

Wang *et al.* introduce Wedjat [49], a smart phone application designed

to aim patients avoiding administration errors with medicaments. The application reminds the user to take the correct medicaments on time and record the in-take schedules for review by healthcare professional. It has two different features, alert the user about the danger of mixing drugs and can revise the in-take schedule automatically when a dose was missed.

2.4 Ubiquitous Health (U-Health)

Mark Weiser introduced and showed the world his vision of ubiquitous computing in the beginning of the 90s, describing it as follows:

“The most profound technologies are those that are invisible. They are woven in the manufacture of everyday life until they are indistinguishable from day to day.”

Weiser presented a concept of a complete abstraction of technology where the machine interacts with man [50]. However, Weiser's vision was too advanced for its time, it lacked the technological support that currently exists [50]. The Internet as we know it today, is a perfect example of a ubiquitous technology, where users focus only on their information and services rather than its intrinsic technology. Ubiquitous computing goes beyond the device, network and protocol. Technology interacts with the user transparently without the user has requested [51].

The ubiquity in healthcare (u-Health) aims to respond to several emerging problems in health systems, including, the increasing the number of chronic diseases related to lifestyle, high costs in the existing national health services, the need to empower patients and families to self-care and manage their own health care and the need to provide direct access to health services, regardless of time and place.

The u-Health uses numerous technologies such as environmental sensors and body, or actuators to monitor and improve the physical or

mental condition of patients. Therefore offers innovative services and applications in various areas of health, such as remote monitoring of elderly or sick people, advanced operating rooms, space for intelligent medical, home care or assisted intelligent environments, intelligent hospitals, etc.

The use of ubiquitous services and applications in health raises several ethical and professional issues. A key question concerns how much information about themselves to individuals is willing to reveal. Usually any individual carry's a mobile device (smartphone, tablet, iPod, etc.) capable of exchanging data with other devices, without requiring an explicit knowledge about this process. Privacy, security and reliability are issues of concern to users of technologies they use, to exchange personal data are always subject to unauthorized access by third parties. For example, in a ubiquitous environment, the elderly will be able to control information-sharing, system administrators rely on your personal preferences and maintain about who can and can not access your personal information?

Another key issue relates to errors in exchange or access information (e.g. medical records), which can cause errors in diagnosis or prevention, who will takes responsibility? These and other questions arise when technological risks inherent, but need to be asked and answered for the sake of development of own applications and services, but also of its users [52].

There are several systems and u-Health applications available for various purposes related to health care.

- A network of satellites for u-Health environment that uses four satellite networks - GALENA, DELTA, MEDASHIP EMISPHER and to improve health services together on the same network to 14 clinics in six different countries supporting rapid care to emergency cases, care in cases of ships and the constant mobility as an online service for most countries of the Euro-Mediterranean [53];
- Systems for monitoring of ECG/PPG signals [54], monitoring drug dosage and quality of life of seniors [55], monitoring of biological signals [56];

- Systems for health care in cases of chronic diseases [57];
- Systems for data collection and monitoring of blood pressure, diabetes, blood sugar and emergencies [58];
- Systems for psychological care and therapy [59].

2.5 Obesity

Obesity is measured using a parameter called Body Mass Index (BMI), a measure of body fat based on height and weight. The BMI standard has been revised to 25 kg per Metre Square, that obesity can cause serious health problems is indisputable. Study after study has suggested that obese people are more prone to heart problems, including elevated risk of heart attacks. The link between obesity and diabetes is well established and some studies suggest obese people have more than just social motivations for losing weight, obesity frequently is associated with life-threatening comorbidities like cardiovascular disease and type 2 diabetes and some forms of cancer [60].

Is a new era in health, sensors, devices, nanotechnology, imaging technologies, and other approaches have been developed through collaborations between engineers, physical scientists, and doctors with expertise in obesity and nutrition. The main goal is to inspire research to develop useful tools and technologies to facilitate the search for a better life, also, the possibility to re-engineering the environment to inspire more physical activity [61].

A mobile game/application called "*It's time to eat!*" is presented in [62]. Uses a mobile phone game to promote and motivate healthy eating habits among children. This game gives to children the control of a pet that responds with photos containing the food they will consume. The pet interacts with the children by sending healthy-eating reminders via email. The messages change depending on the day of the week. Each player must

take a photo of the meal and submit it, and then the pet will give a score based on the healthiness of the food and the amount of food eaten.

In [63] the authors propose a technological solution to assist people with obesity problems, they have conducted a field study to visualize and inform their design. It's call *Virtual Specialist (VS)*, that stays with the patient and advice him at all times on issues related to diet and physical activities.

The study [64] presents a waist circumference measurable belt with an accurate to 5mm resolution, deciding if someone has the potential for obesity. This system can assist in obesity and overweight management. The system is composed of magnets and magneto-resistive sensors to measure waist circumference, has also developed an algorithm to distinguish the belt moving direction.

3. Requirements Analysis

This chapter presents the design and implementation of the SapoFitness architecture. It starts presenting the conceptual design of SapoFitness proposal and then presenting UML diagrams of the main actions and procedures.

The Unified Modelling Language (UML) is a modelling language, non-proprietary of third-generation. The UML is not a development methodology, which means it does not tell what to do first or how to design the system, but it helps to visualize the design and communication between objects.

Basically, UML allows developers to view the products of their work on standardized charts. Along with a graphical notation, UML also specifies meanings, i.e. semantic. It is an independent rating process, although the Rational Unified Process (RUP) have been specifically developed using the UML. The goals of UML are: specification, documentation, sub-structuring for larger view and logical view of the complete development of an information system. UML is a way to standardize the forms of modelling.

The analysis of requirements is an essential process for the development of a product, because is in the analysis of requirements that is define precisely the objectives to develop. When it comes to developing software, it is of extreme importance for all the analysis requirements will constrain both the software as its final operation.

3.1 Design and application of SapoFitness

This section discusses the system architecture and the required technologies to create it. The focus will be on available and emerging architectures that foster its practical deployment.

3.1.1 SapoFitness System Architecture

SapoFitness is a mobile system that requires several daily inputs from users, mainly, food and exercise, and others like weight, age, and height. These data are updated on the user PHR through a Web service for easily and immediate access. The user profile makes use of the PHR for determining the user nutritional status. This status includes his/her Body Mass Index (BMI), user daily caloric and energetic needs.

SapoFitness must keep its user well motivated not only to use the application but also to loose weight. Thus, it allows its user to share his/her performance and achievements through well-known social networks (such as *Facebook*, *Twitter*, *Hi5*, *Myspace*, etc.). Figure 4, presents the *SapoFitness* system architecture with main actions defined for communication. All data is get and save on a remote database through HTTP on a SOAP and REST Web services that furnishes all the required information.

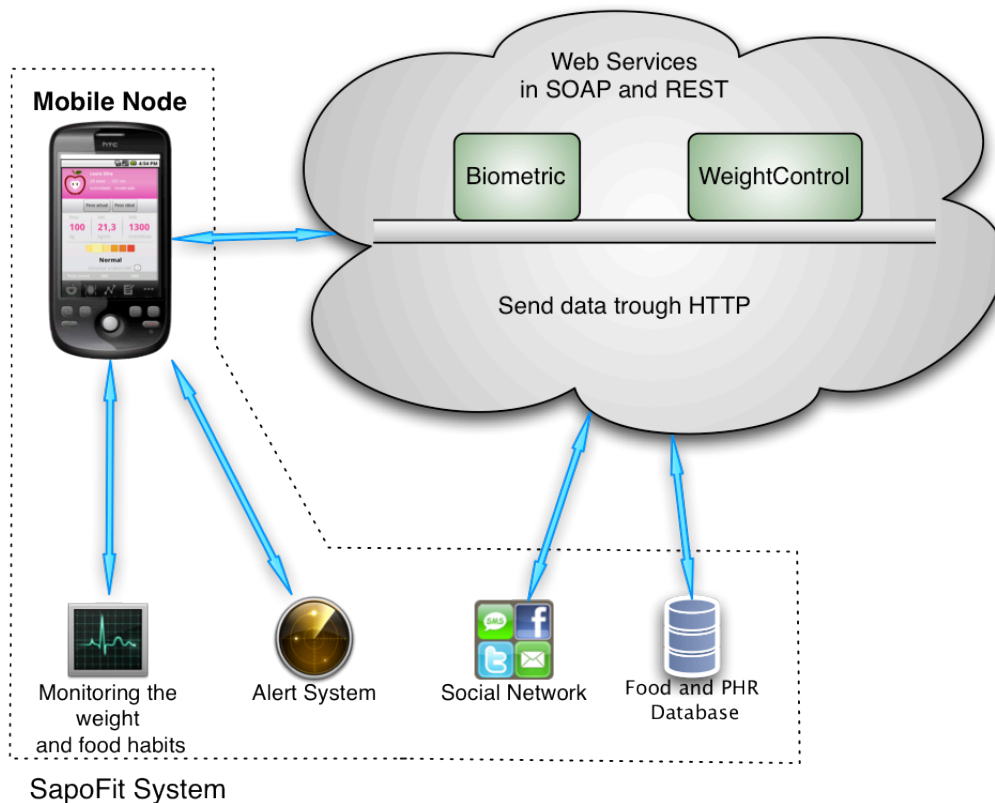


Figure 4. SapoFitness System Architecture.

The database contains the user personal data, PHRs and all the user intake food and physical activities. This information will customize the alert system and its messages to the user. This alert system maintains a frequent application-user interaction and motivates the user to follow the respective diet program and physical activities. The user food habits monitoring is essential to update the user profile.

Figure 5, presents typical m-health service architecture. The Healthcare providers offer Remote Monitoring Systems, Physician User Interface and Emergency Response in the PHR, through a global communication network for healthcare takers, in various devices such as mobile devices, sensors and specialized devices in applications that allow health data gathering and health records.

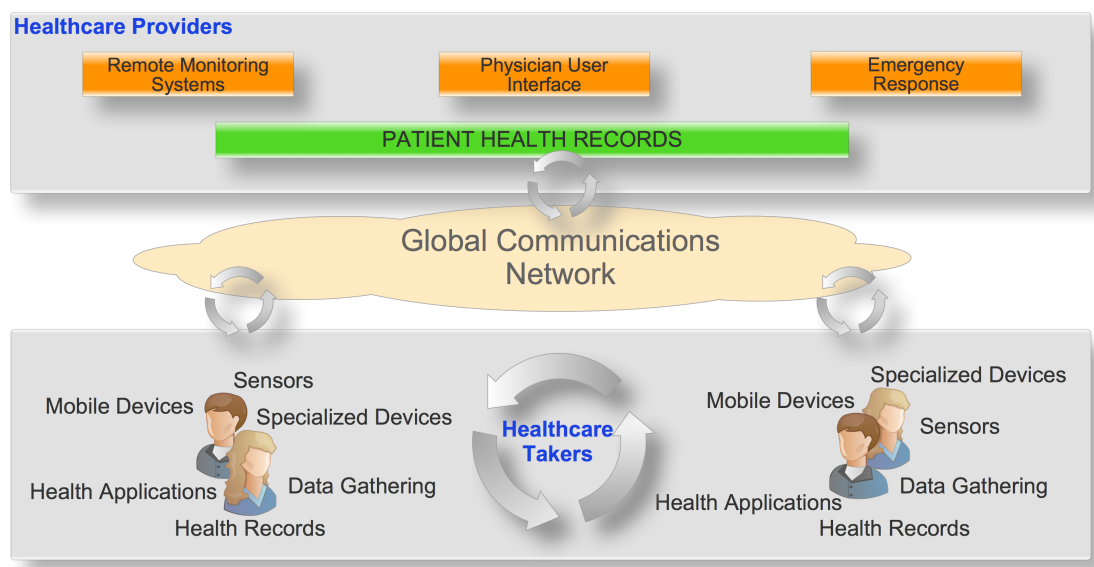


Figure 5. M-Health services framework.

3.2 Essential requirements

The aims of the application were divided into two groups, the general and partial objectives.

For the general objectives were defined as follows:

The application interface will be simple, with large buttons for the user, but with appropriate size for the use of the finger, and the user-machine interaction, offers a simple and powerful user interface to interact and handle the system. Another important aspect is the experience at the first utilization. It is very easy to use and intuitive.. The information provided by application to the user, will be concise and appropriate size for easy reading.

For partial objectives were describe as follows:

The application includes a certain dose of fun, allowing a user to interact and have fun with the application, but at the same time controlling his/her weight and promoting several physical activities.

Summarizing, *SapoFitness* offers an intuitive, stable, fun and helpful application to users. The ability to share milestones with social networks, a user-machine interaction, easy handling and a good ratio between fun/motivation and control weight, applied not only to control obesity but also to encourage on problems of malnutrition. The application constantly monitors its user. The application keeps track of all food intakes and sends several customized alerts regarding his/her diet progress.

The sharing mode is a strong aspect that allows sharing with social networking, electronic mail (e-Mail), short message system (SMS), and all the applications that are installed on the mobile device. Then, it increases the possible ways to share achievements/milestones in weight control. Using Web Services in the *SapoFitness* system, they allow the use of different platforms to communicate with the database. Web Services are well known by its general model for building applications and can be implemented for any operation system that supports communication over the Internet [65].

3.3 Used technologies

SapoFitness targets mobile devices running Android platform, which is a software stack for mobile devices that includes an operating system, middleware and key applications [66]. By the way, the system can easily be reproduced to other mobile operating systems, such as iPhone, Windows mobile, Symbian, BlackBerry, MeeGo, etc. The major solution for application development was Java programming language, using the Android SDK (Software Development Kit). This SDK provides the needed tools and APIs to create applications for Android platform using the Java programming language.

The development tool used to create the mobile application was the Eclipse IDE (Integrated Development Environment) with the ADT (Android

Development Tools) plugin. The ADT is designed to include a powerful and integrated environment on Android applications. It extends the capabilities of Eclipse to quickly set up new Android projects, debug applications including emulation debug for easy initial debugging of mobile applications. The development tool used to create the web services on Windows Communication Foundation (WCF) that is the framework for building service-oriented applications was the Visual Studio 2010, and the major solution for the web services was C#. The Visual Studio also offers a database system management, which was used to generate and manage the system database. The implemented web service communicates through Simple Object Access protocol (SOAP) messages over Hypertext Transfer Protocol (HTTP) with the PHR and the SapoFitness services. The information is returned to the mobile application in JavaScript Object Notation (JSON) or Extensible Markup Language (XML).

Android relies on a Linux 2.6 kernel for the functionality of core system and executes code written in the Java programming language in a specially designed virtual machine called Dalvik, to run Dalvik executable files compiled in Java classes that are optimized to minimize memory consumption. The overall system architecture is in Figure 6.

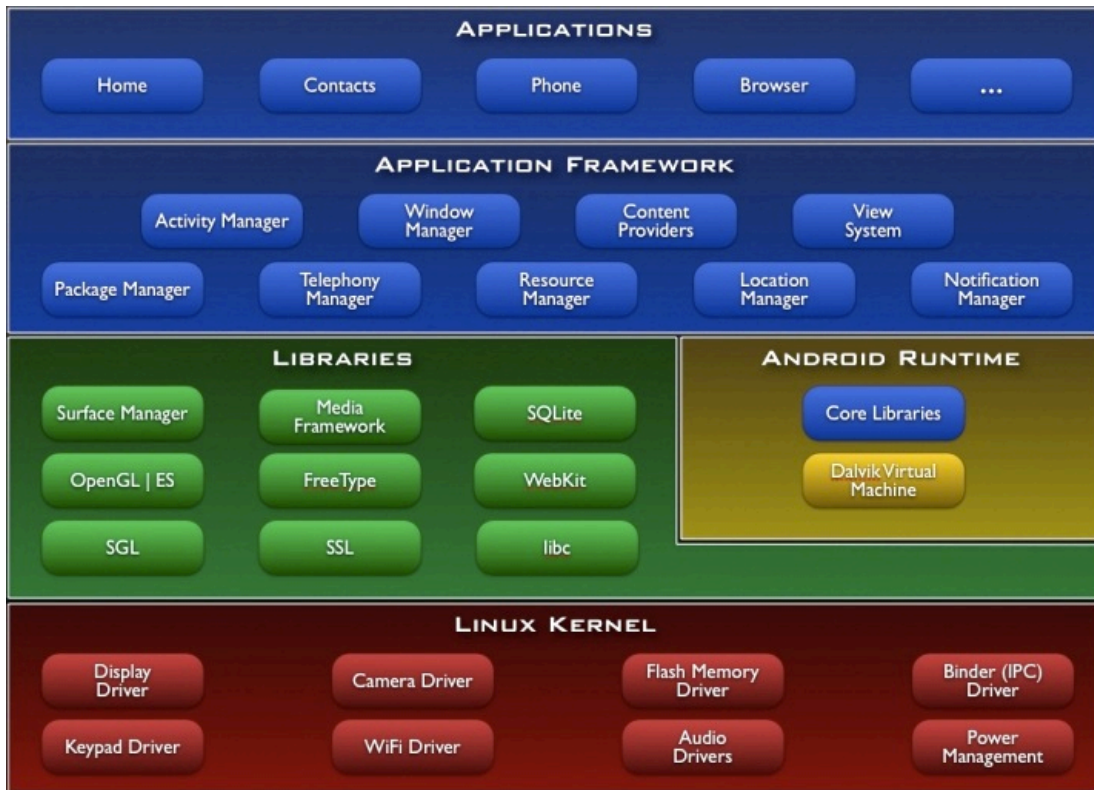


Figure 6. Android system architecture [66].

The main components of the Android architecture are the following:

1. **Applications:** These are applications written in Java. Some of basic applications include an calendar, email client, SMS program, maps, making phone calls, accessing the Web browser, accessing the contacts list and others. For the average user, this is the layer most used, rest all layers are used by Google programmers, developers and hardware manufacturers.
2. **Application Framework:** This is the skeleton or framework which all android developers has to follow. The developers can access all framework APIs an manage phone's basic functions like resource allocation, switching between processes or programs, telephone applications, and keeping track of the phone's physical location.

The architecture is well designed to simplify the reuse of components. The application framework is as a set of basic tools with which a developer can build much more complex tools.

3. **Libraries:** This layer consists of Android libraries written in C, C++, and used by various systems. These libraries tell the device how to handle with different kinds of data and are exposed to Android developers via Android Application framework. Some of these libraries includes media, graphics, 3d,SQLite, web browser library etc.
The Android runtime layer that includes set of core java libraries and DVM (Dalvik Virtual Machine) is also located in same layer.
4. **Runtime Android:** This layer includes set of base libraries that are required for java libraries. Every Android application gets its own instance of Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently and it executes files in executable (.Dex) optimized for minimum memory.
5. **Kernel - Linux:** This layer includes Android's memory management programs, security settings, power management software and several drivers for hardware, file system access, networking and inter-process-communication. The kernel also acts as an abstraction layer between hardware and the rest of the software stack.

3.3.1 Android GUI Architecture

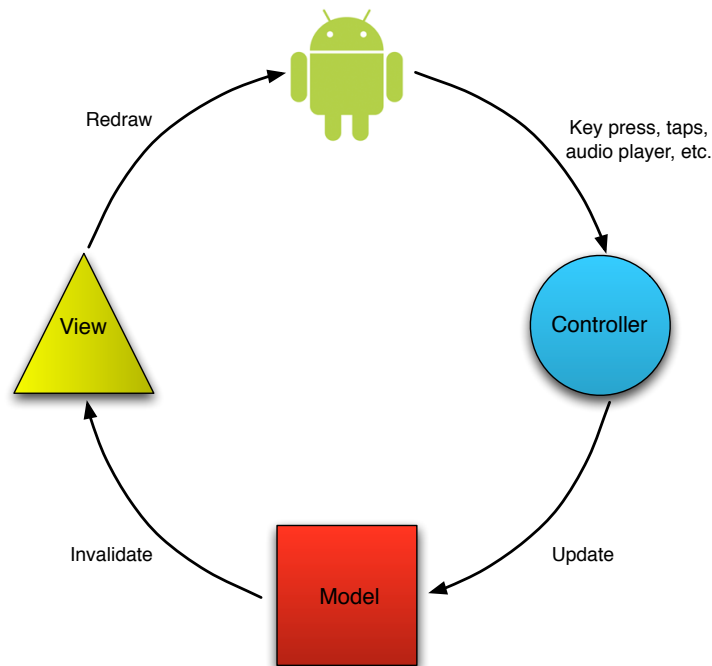


Figure 7. Android GUI architecture.

Android GUI is single-threaded, event-driven and built on a library of matted components (Figure 7). The Android UI framework is organized around the common Model-View-Controller pattern.

- The Model: The model represents data or data container. Can see as a database of pictures on the device. Any user wants to hear an audio file, he clicks play button and it triggers an event in the app, now the app will get data from data store or database and as per input and creates data to be sent back to the user. Can refer this data as Model.
- The View: The View is the portion of the application responsible for rendering the display, sending audio to speakers, generating tactile feedback, and so on. Now as per above example, the view in a hypothetical audio player might contain a component that shows the album cover for the currently playing tune. User will always interact with this layer. User actions on this layer will trigger events that will go to the application functions.

- The Controller: The Controller is the portion of an application that responds to external actions: a keystroke, a screen tap, an incoming call, etc. It is implemented as an event queue. On User's action, the control is passed over to controller and this will take care of all logic that needs to be done and prepare Model that need to be sent to view layer.

3.4 Behavioral Diagrams

3.4.1 Use Case Diagrams

Use diagrams represent how a user interacts with the features of the program in system development and a graphical overview of the functionality provided by a system in terms of actors.

The use case diagram below (Figure 8), describe the sequence of actions of one actor, the user, and how he uses the system.

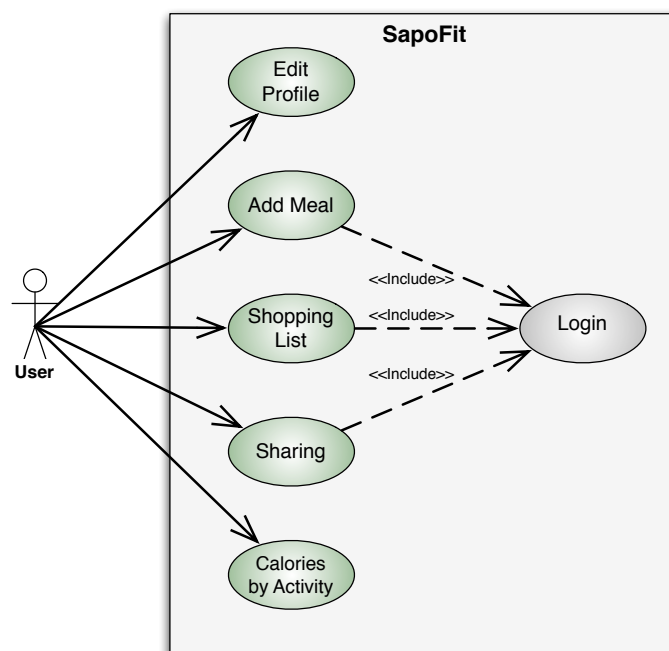


Figure 8. User Case Diagram of SapoFitness.

3.4.2 Activity Diagrams

The activity diagrams are used to describe the business operational components of a system step-by-step, and the overall flow of control. Figure 9, presents the activity diagram with main actions defined for *SapoFitness* application.

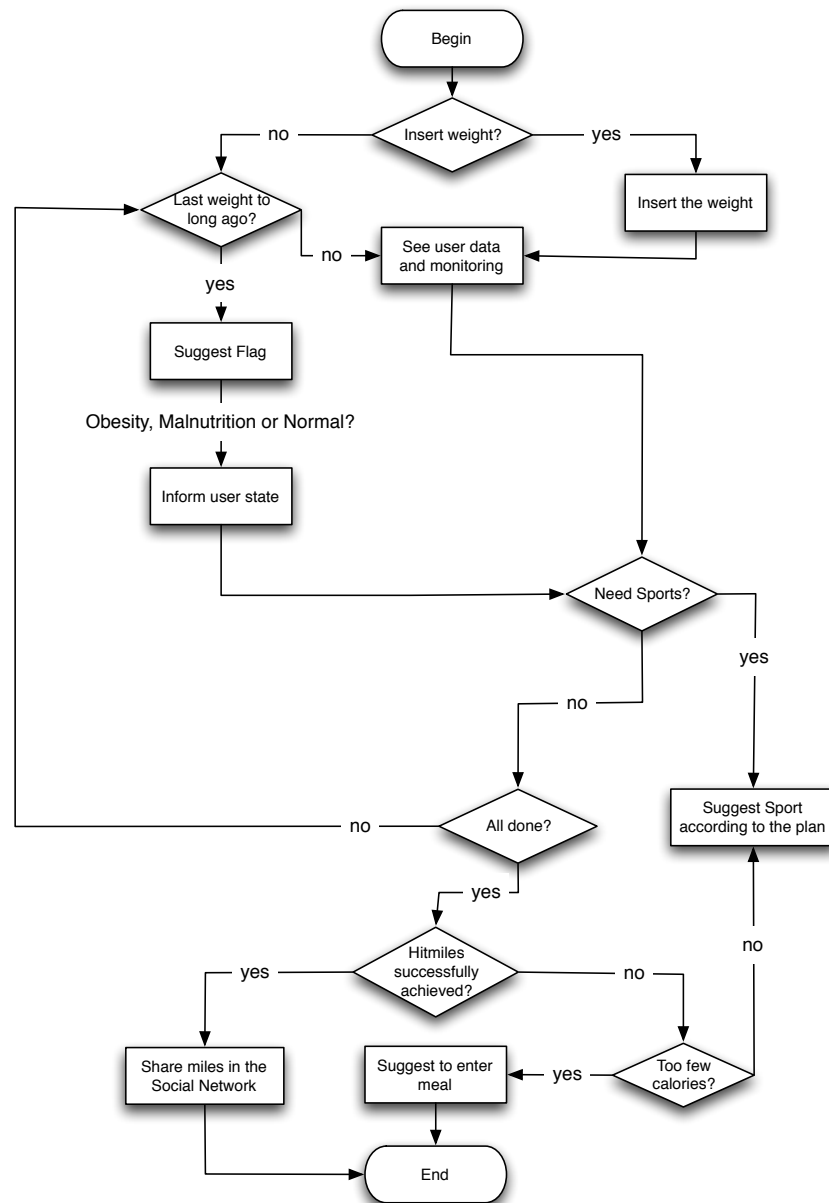


Figure 9. Activity diagrama of SapoFitness.

Figure 10 presents the decision algorithm for the meal alert, verify if a meal was taken, compare it with previous day to see the user habits and alerting the user if the meal was missed.

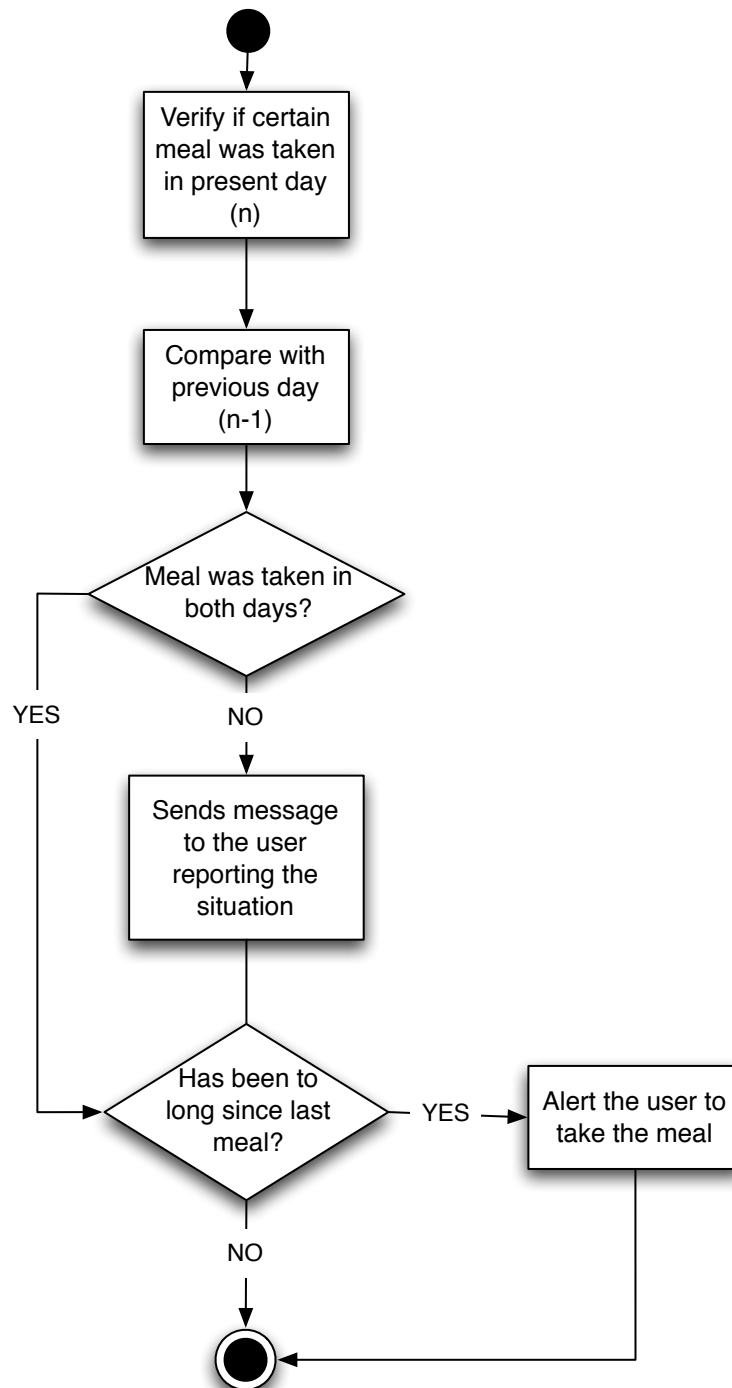


Figure 10. Decision algorithm for the meal.

Figure 11 presents the decision algorithm for the weight, verify if the weight has been updated, compare it with previous weight, if the weight was not according to the plan, the system will suggest another meals.

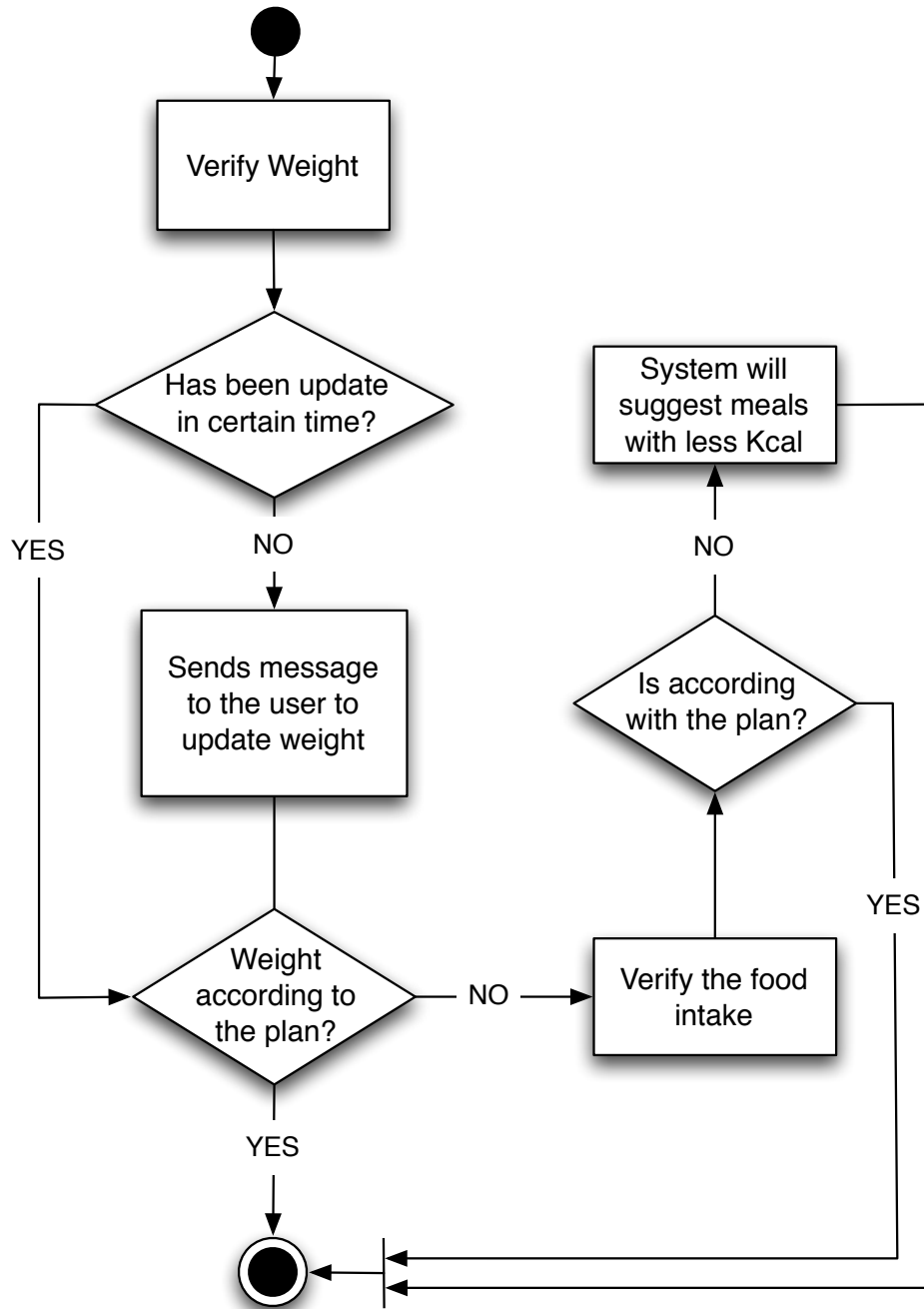


Figure 11. Decision algorithm for weight.

The intake and energy needs decision algorithm for application-user interaction is shown in Figure 12. This diagram presents the system main functions and how they interact with the user. This interaction is based on text alerts sent to the user depending on its nutritional evolution. This procedure begins with the evaluation of calories intake and energy needs.

By comparing the balance between both and also the amount of daily exercise the system will suggest and alert the user to its next step on food intake and physical activities.

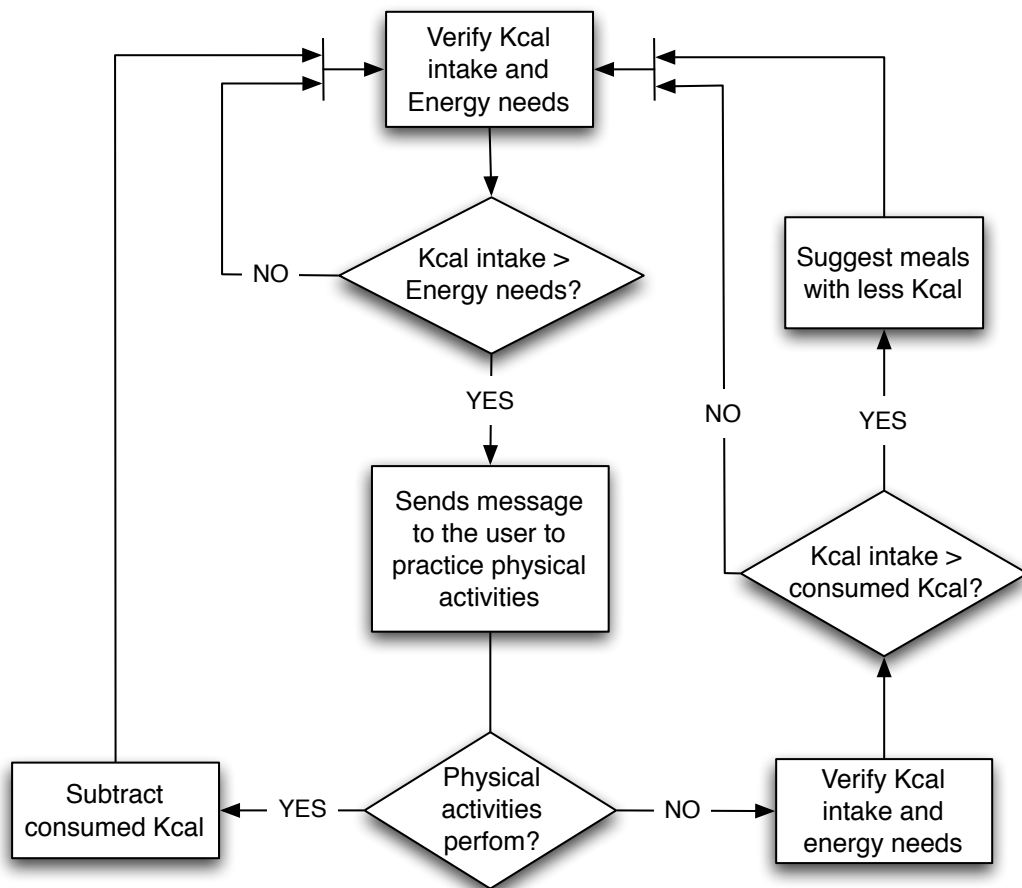


Figure 12. Decision algorithm for intake and energy needs.

3.5 Interaction Diagrams

3.5.1 Sequence Diagrams

The sequence diagram shows how processes interact, every sequence diagram is important, but some have more importance than others, the most relevant are: Login, Edit Profile, Social Network, Physical Activities and Add Meal.

Figure 13 presents the login communication. The user logs in with a valid email address and password that is verified through the web service, which is assumed in the remaining diagrams.

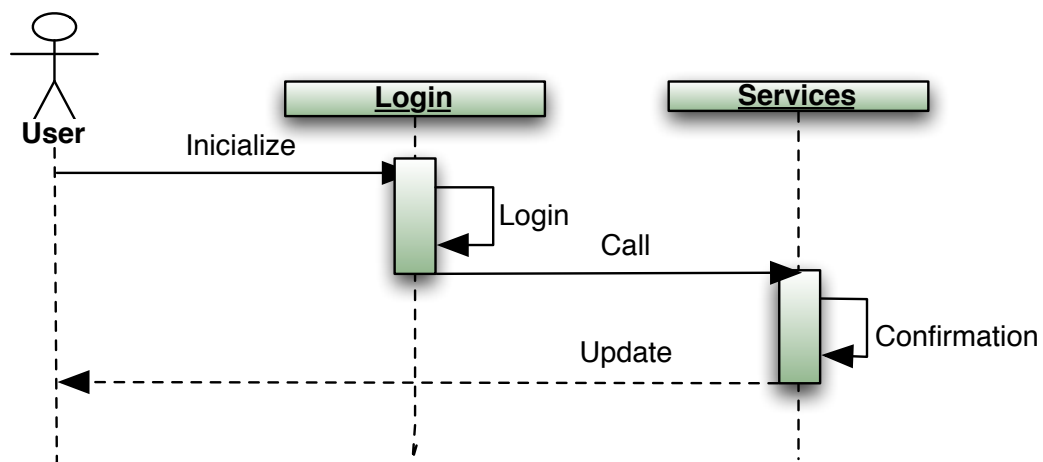


Figure 13. Sequence Diagram - Login.

Figure 14 represents the Edit Profile process where the edition of the profile is performed in the event user profile that creates a edit profile where it will communicate with web services to update the data.

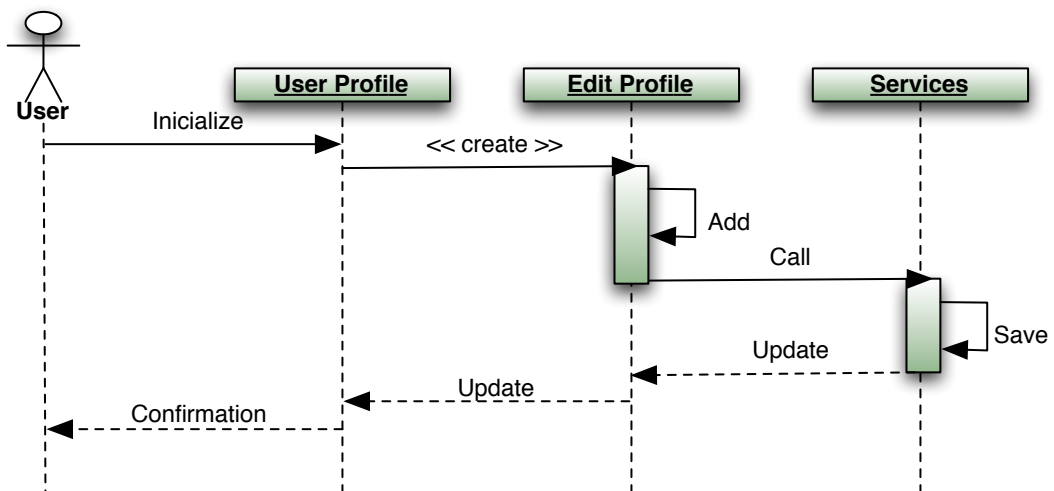


Figure 14. Sequence Diagram - Edit Profile.

In the Figure 15 is presented the communication with the social networks, all starts with the user profile once again, next is the Evolution event which will create the thread social network to post the milestones.

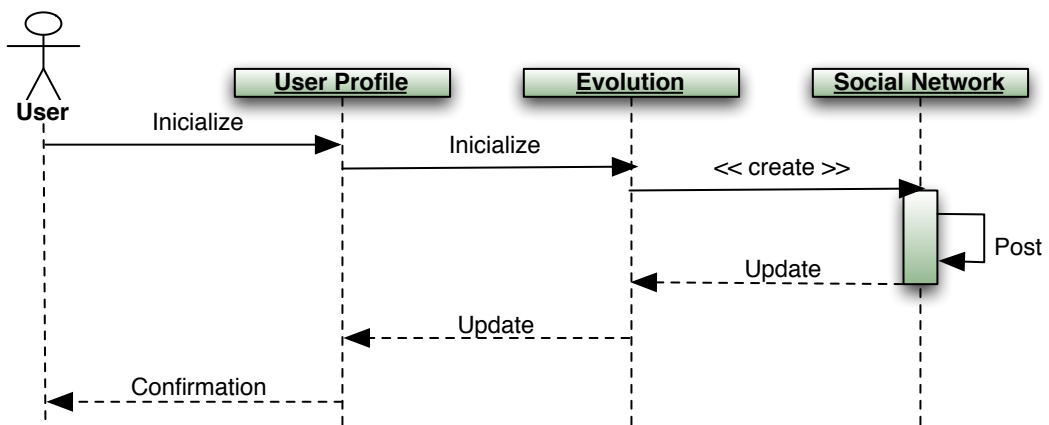


Figure 15. Sequence Diagram - Social Network.

The Figure 16 presents the process of physical activities where is possible to observe the communication with the web service to get the results by length and weight.

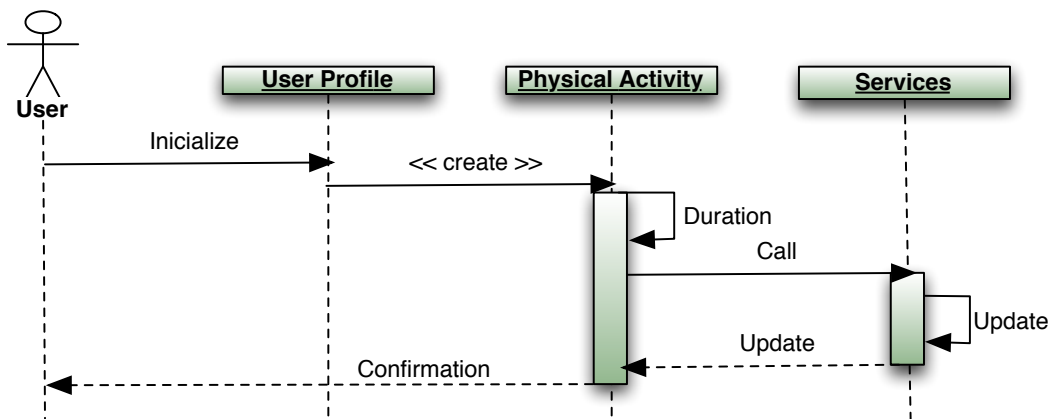


Figure 16. Sequence Diagram - Physical Activity.

Figure 17 shows the diagram which represents the add meal and the interaction with the web service to save new meals.

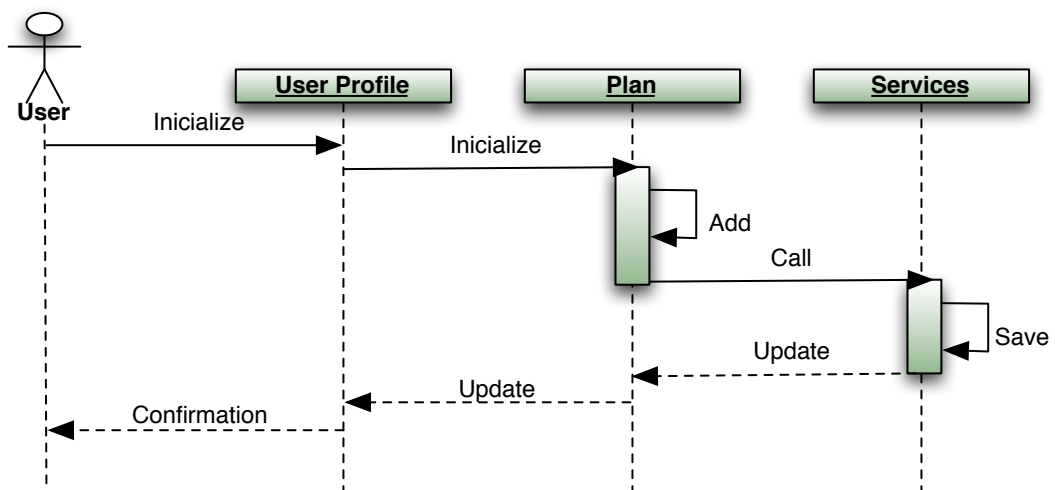


Figure 17. Sequence Diagram - Add Meal.

3.6 Structural Diagrams

3.6.1 Class Diagram

Class diagram is a representation of the structure and relationships of all classes that serve as a model for objects. Figure 18 presents the class

diagram with the attributes and methods. All classes work around the class Principal and all starts in Login_Pre.

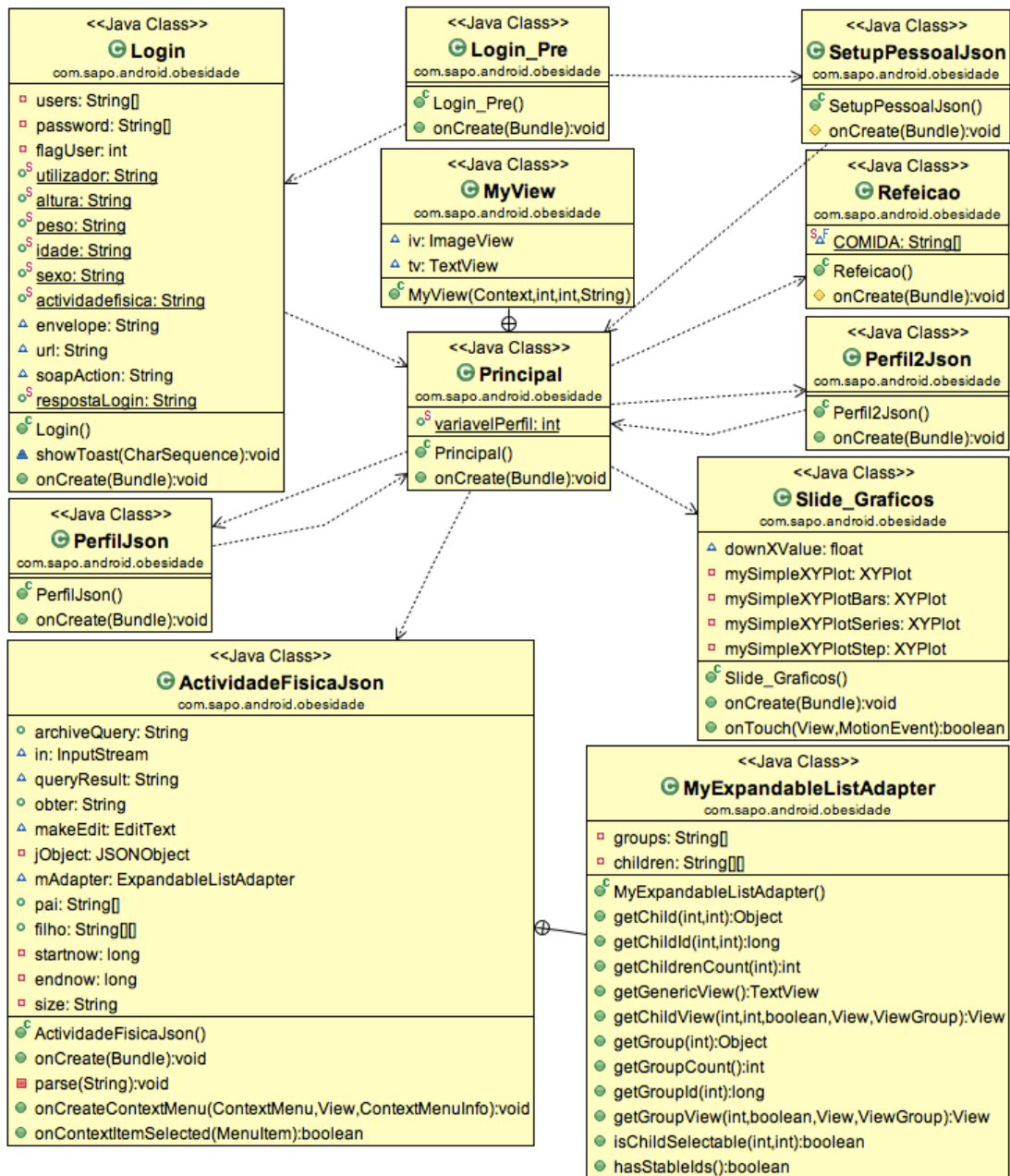


Figure 18. Class diagram of the Windows.

In figure 19 is presented the class diagram of the connection to the Web services with the rest of the application.

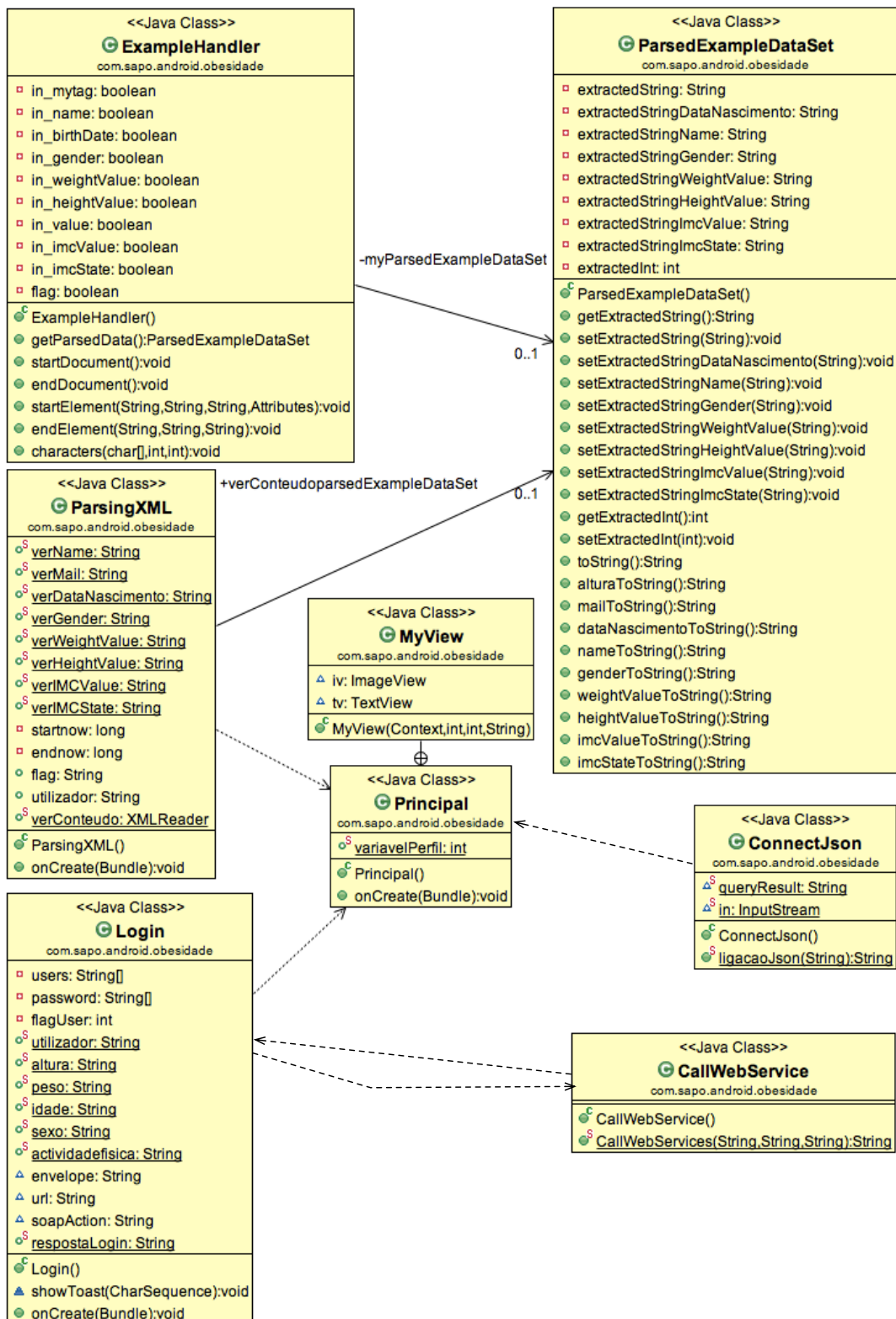


Figure 19. Class diagrama of the connection with Web services.

4. Web Services

Nowadays, more and more Web services are present in applications, consuming a service or serving information. In the scenario of distributed applications, the service has become a key piece, replacing the old paradigm of using Dynamic-Link Library's (DLLs).

The W3C [67] defines A Web service as a method of communication between two electronic devices over a network, it is a solution used in systems integration and communication between different applications. With this technology it is possible that new applications can interact with those that already exist and that the systems developed on different platforms are compatible. Essentially, the Web service causes the features of the software application available over the network in a standardized way.

Web Services achieves its goal of a technologically neutral way, which provides well-defined interfaces for distributed functionality, which are independent of hardware platform, operating system and programming language. Then distributed features or services that can run on different hardware platforms, can run on different operating systems, or can be written in different programming languages and communicate through Web service interfaces.

Models of Web service communications for mobile computing can result in unacceptable performance overhead. This potential problem comes from two factors. First, the encoding and decoding messages verbose XML-based SOAP consumes resources, therefore Web service participants, particularly mobile clients, can suffer from poor performance. Secondly, the difference in performance and quality between the wired and wireless communication will not be closed quickly, it is caused by restrictions on the mobile environment due to limiting the speed of the processor, the limited battery life and slow unreliable and intermit connection.

4.1 Styles of use

Web services are a set of tools that can be used in a number of ways. The three most common styles of use are the RPC, SOA and REST [68].

4.1.1 RPC - Remote procedure calls

RPC Web services present a distributed function (or method) call interface that is familiar to many developers. Typically, the basic unit of RPC Web services is the WSDL operation.

WSDL

The WSDL (Web Services Description Language) [69] is a specification developed by W3C that describes the Web Services according to an XML format, working as a contract of service. This is a document written in XML and describe the service, specifies how to access it and what the operations or methods available, works as a sort of "TypeLibrary" Web service, besides being used for the validation of calls methods. The WSDL is extensible to allow description of services and their messages regardless of

message formats and network protocols that are used. However, it is common to use the MIME (Multipurpose Internet Mail Extensions) and SOAP.

The WSDL describes the services available on the network through a semantic XML, this provides the necessary documentation to be called a distributed system and the procedure required for this communication is established. While SOAP specifies the communication between a client and a server, the WSDL describes the services offers.

A WSDL description contains all the details of a Web service, including the service's URL, the communication mechanisms it understands, what operations it can perform and the structure of its messages.

SOAP

Simple Object Access Protocol (SOAP) is a protocol for exchanging structured information in a decentralized and distributed platform. It is based on Extensible Markup Language (XML) to format the messages, and usually based on other application layer protocols, most notably in Remote Procedure Call (RPC) and HTTP for trading and messaging, using GET/POST over HTTP, allowing the data to be exchanged independent where the user is in the network.

Web Services Interoperability is mainly related to its XML based on open standards. The SOAP is defined in XML. Since it is text based and self-description, SOAP messages can transmit information between the services in heterogeneous computing environments without worrying about conversion issues.

Other approaches with nearly the same functionality as RPC are Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA), Microsoft's Distributed Component Object Model (DCOM) or Sun Microsystems's Java/Remote Method Invocation (RMI).

4.1.2 SOA - Service-oriented architecture

Web services can also be used to implement architecture according to service-oriented architecture (SOA) concepts, where the basic unit of communication is a message, rather than an operation. This is often referred to as "message-oriented" services.

WCF

Windows Communication Foundation (WCF) [70] is the Microsoft platform available since .NET Framework 3.0, used to handle the communication between systems. WCF is a union of a Web service, WSE, Remoting, and COM +, all in a single platform, simple to use, robust and easy integration.

WCF is designed in accordance with service oriented architecture principles to support distributed computing where consumers consume services. Clients can consume multiple services and multiple clients can consume services. Services are loosely coupled to each other. Services typically have a WSDL interface that any WCF client can use to consume the service, irrespective of which platform the service is hosted on. WCF implements many advanced Web services (WS) standards such as WS-Addressing, WS-ReliableMessaging and WS-Security. With the release of .NET Framework 4.0, WCF also provides RSS Syndication Services.

A WCF client connects to a WCF service via an Endpoint. Each service exposes its contract via one or more endpoints. An endpoint has an address, which is a URL specifying where the endpoint can be accessed, and binding properties that specify how the data will be transferred.

An endpoint is a service exposes (Figure 20). An endpoint contains three main features called ABC (Address, Binding and Contract). An analogy for these features is that the Address is where the service is hosted, Binding is how the service can be accessed, and that the Contract would be seen in service.

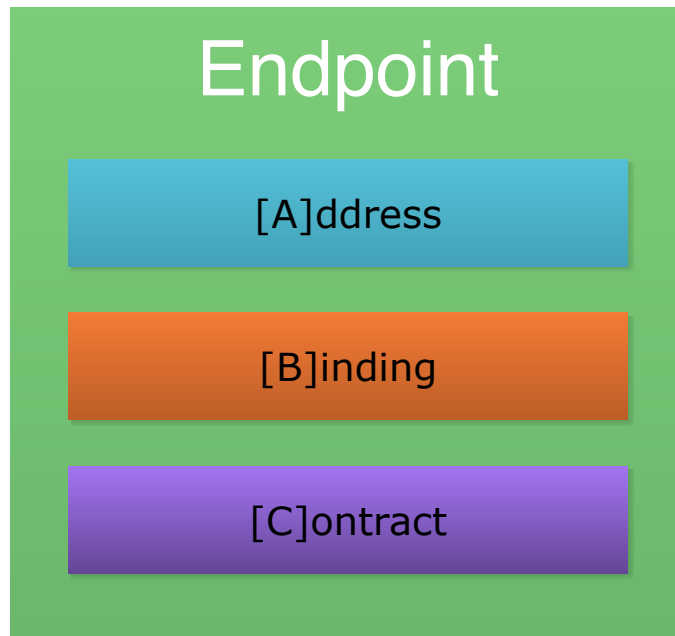


Figure 20. WCF endpoint - ABC (Address, Binding and Contract).

WCF supports interoperability with WCF applications running on the same or different Windows machine or standard Web services built on platforms such as Java running on any operating systems. WCF does not only support SOAP messages, it can also be configured to support standard XML data that is not wrapped in SOAP, or can even be used to support formats such as RSS, or JSON that makes WCF flexible for current requirements and future changes [71].

One of the keys to SOA success is that it should be easy to consume a service. The entry barrier for service consumers must be low to support SOA adoption. While a top-down SOA effort may succeed, it will certainly

take longer than bottom-up approach when developers are able to consume services as they see fit.

4.1.3 REST - Representational state transfer

Is a software engineering technique for distributed hypermedia systems like the World Wide Web the term originated in 2000 in a doctoral thesis [72] written by Roy Fielding, one of the main authors of HTTP protocol specification that is used by Web sites.

REST is an architectural style that treats the Web as a resource-centric application. Practically, this means each URL in a RESTful application represents a resource.

Dr. Roy Fielding introduced the term REST in his Ph.D. dissertation, where he referred to *“hypermedia as the engine of application state.”* This means that a resource is expected to contain hyperlinks. These hyperlinks are the method by which a transition can take place that changes the resource state or transfers to another resource. While hyperlinks are commonplace in (X)HTML applications meant to be used by humans, they have not typically been used in XML, which is meant to be consumed by machines. Like (X)HTML, REST Web services make use of hyperlinks in XML.

Traditional Web applications access resources using HTTP GET or POST operations. In contrast, RESTful applications access resources following the create, read, update, and delete (CRUD) style using the full range of HTTP verbs (POST, GET, PUT, and DELETE).

The services inspired in REST and respect the most significant aspects of its architecture and restrictive, in particular the restriction of uniform interface, are called RESTful.

There's one more key component of a REST application: RESTful applications should be stateless. This means in a REST application no session state is stored on the server. All of the information needed to satisfy the request is carried in the request message itself. A client can

therefore cache a representation of a resource, which can significantly improve the application's performance, where a service explicitly allows it.

4.2 RESTful *versus* SOAP

Applying traditional models to mobile communications may simply result in failure at the level of performance. First the encoding and decoding of verbose XML-based SOAP messages consumes important resources, putting the performance in risk.

Then comes the problem of communications, Wi-Fi, 3G/GSM are not as efficient and fast as wired, which accrue to the processing speed and limited memory and battery issues.

Together with WSDL and XML Schema, SOAP has become the standard for exchanging XML-based messages. SOAP was also designed from the ground up to be extensible, so that other standards could be integrated into it, and there have been many, often collectively referred to as WS-*

REST is an ancient philosophy more than a new technology. Whereas SOAP looks to drive the next phase of Internet development with a series of new specifications, the REST philosophy advocates that the existing principles and protocols of the Web are enough to create robust Web services. This means that developers who understand HTTP, XML and JSON can start building Web services immediately, without any toolkits beyond that normally uses for the development of Internet applications.

Interface Flexibility

The key to the REST methodology for writing web services is using an interface that is already well known and widely used: the URI. For example, exposing a stock quote service, in which a user enters a stock ticker symbol, to return to a real-time price could be as simple as a script on a Web server accessible via the URI.

Any client or server application that supports HTTP could easily call the service with an HTTP GET command. Depending on how the service provider wrote the script, the result of the HTTP response can be as simple as some standard headers and a text string containing the current price for the given symbol. Or it could be an XML or JSON.

This interface method has significant advantages over SOAP-based services. Any developer can figure out how to create and modify a URL to access the different Web resources. SOAP, on the other hand, requires specific knowledge of a new XML specification, and most developers will need a SOAP toolkit to make requests and analyze the results.

Lighter on Bandwidth

Another advantage is that the RESTful interface requests and responses can be short. SOAP requires an XML wrapper around each request and response. After typing and namespaces are declared a stock quote from four or five digits in a SOAP response may require more than 10 times as many bytes as would be the same answer in REST.

SOAP proponents claim that strong typing is a necessary feature for distributed applications. In practice, however, both the request for the service and knowing the types of data ahead of time, thus transferring that information requests and answers is gratuitous.

How does one know the data types—and their locations in the response ahead of time? Like SOAP, REST also needs a corresponding document that describes the input parameters and output data. The good part is that rest is flexible enough that developers can write WSDL files for their services, if such a formal declaration was needed. Otherwise, the statement could be as simple as a web page readable that says, “Give this service an input of some stock ticker symbol, in the format q=symbol, and it will return the current price of one share of stock as a text string.”

Security Safeguards

Probably the most interesting aspect of REST vs. SOAP debate is the safety angle. Although the field of SOAP insists that sending remote procedure calls through the standard HTTP port is a good way to ensure support of Web services across organizational boundaries, REST followers argue that the practice is a major design flaw that compromise network security. REST calls also go over HTTP or HTTPS, but with REST the administrator (or firewall) can discern the intent of each message, analyzing the HTTP command used in the request. For example, a GET request can always be considered safe because it cannot, by definition, modify the data. It can only query data.

A typical SOAP request, on the other hand, will use POST to communicate with a particular service. And without looking at the SOAP envelope, a task that is both resource-consuming and not built into most firewalls, there is no way of knowing if the application simply wants to query data, or delete entire tables from the database.

As for the authentication and authorization, SOAP places the burden on application developer's hands. The REST methodology instead ignores the fact that Web servers already have support for these tasks. By using industry-standard certificates and a common identity management, as a Lightweight Directory Access Protocol (LDAP) server, developers can make the network layer do all the heavy lifting. This is not only useful for developers, but it relieves the weight on administrators who can use something as simple as the files Access Control List (ACL) to manage its Web services just as you would any other URI.

Type Handling

SOAP provides relatively stronger typing, since it has a fixed set of data types supported. Therefore, it ensures that a return value will be available directly in the corresponding native type in a particular platform. In case of HTTP based API the return value must be deserialized into XML

and then type-cast. This may not be much effort, especially for dynamic languages. In fact, even mess of complex objects across an object is very similar to traverse an XML tree, so there is no definitive advantage in terms of ease of client-side coding.

Client-side Complexity (Thinner Clients)

Making calls to an REST/HTTP API is significantly easier than making calls to a SOAP API. The latter requires a client library, a stub and a learning curve. The first is native to all programming languages simple and involves the construction of an HTTP request with appropriate parameters appended to it. Even psychologically former seems much less effort.

Testing and Troubleshooting

It is also easy to test and troubleshoot an HTTP API, as it is possible to build a call with nothing more than a browser and check the response within the browser window. No tool of problem solving is necessary to generate a request/response cycle. Herein lies the main strength of HTTP-based APIs.

Server-side Complexity

Most programming languages make it extremely easy to expose a method using SOAP. The SOAP server library does the serialization and deserialization. To expose the methods of an object as an HTTP API can be relatively more difficult, since it may require serialization of output to XML. Doing the REST API involves additional work to map URI paths to specific handlers to import and the meaning of the HTTP request in the scheme of things. Off course there are many frameworks to make this task easier. However, starting today, it's even easier to expose a set of methods using SOAP, which is to expose them through regular HTTP.

Caching

Since HTTP based/RESTful APIs can be consumed through simple GET requests, intermediate proxy server/reverse proxy can cache the answer very easily. On the other hand, SOAP requests use POST and require a complex XML to be created, which caches the response difficult.

REST is not Perfect

REST is not the best solution for each Web service. Data that need to be safe should not be sent as parameters in URIs. And large amounts of data, such as purchase orders in detail, can quickly become cumbersome or even outside the boundaries within a URI. In these cases, the SOAP is actually a solid solution. But it is important to try REST first and resort to SOAP only when necessary. This helps keep the development of simple and affordable.

Fortunately, the REST philosophy is catching with the developers of Web services. The latest version of the SOAP specification now allows certain types of services to be exposed via URIs (although the answer is still a SOAP message). Likewise, users of Microsoft's platform .NET can publish services so that they use GET requests. All this means a shift in thinking about how best to interface Web services.

Sending and receiving a SOAP message is not always the best way for applications to communicate. Sometimes a simple REST interface and a simple text response do the trick and saves time and resources in the process.

In [73] as been demonstrated that the gap is very large between message size and response time of RESTful, with conventional SOAP web service, as it can be seen in the Figure 21 and 22.

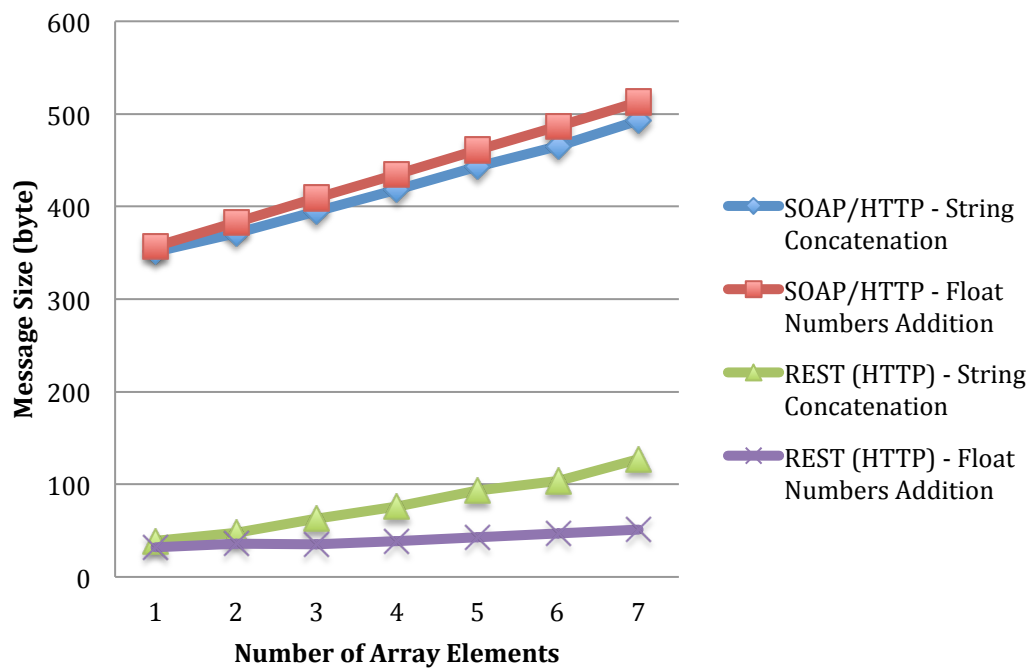


Figure 21. Message Size (Bytes) of String Concatenation and float addition service in SOAP and REST.

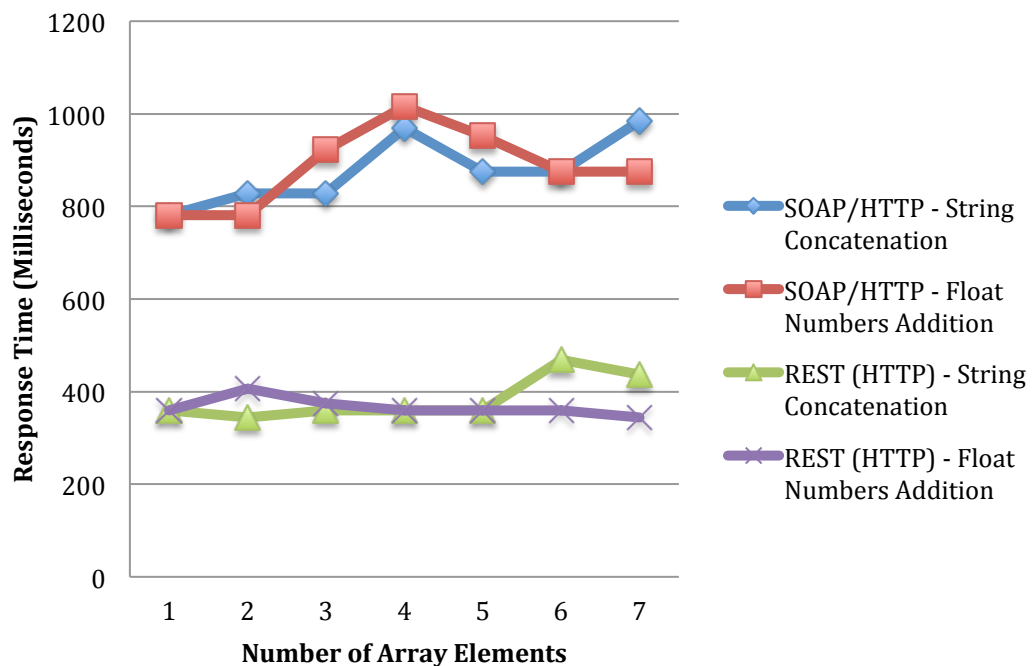


Figure 22. Service Response Time (Milliseconds) of String Concatenation and float addition service in SOAP and REST.

The evaluation results demonstrate the performance advantages of using Web services RESTful on mobile devices. Advantages include fewer sizes of messages and response time, results of performance comparison between conventional SOAP and RESTful show obvious high performance RESTful over SOAP. So RESTful offers a perfectly good solution for most deployments with greater flexibility and lower overhead.

Another comparison has been performance in [74], between SOAP framework and RESTful framework. The comparison demonstrate that RESTful Web services prove to be more suitable for mobile environment because it doesn't require large weight parses, witch supports caching and will save the limited network bandwidth, increase scalability and reliability, and doesn't consume a large amount of mobile resources.

4.3 Markup Languages

4.3.1 Extensible Markup language (XML)

Extensible Markup Language (XML) [75] is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification [76] produced by the W3C, and several other related specifications.

The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in Web services.

4.3.2 JavaScript Object Notation (JSON)

JSON (JavaScript Object Notation) [77] is a lightweight text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for most languages.

JSON is built on two structures, a collection of name/value pairs and an ordered list of values. The collection of name/value pair in various languages, this is perceived as an object, an object, hash table, record, struct, dictionary, keyed list, or associative array. An ordered list of values, in most languages, this is characterized as an array, vector, list or sequence.

These are universal data structures. Virtually all-modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

The simplicity of JSON has resulted in its widespread use, especially as an alternative to XML. One of the claimed advantages of JSON over XML as a format for data exchange in this context is the fact that it is much easier to write a JSON parser. This was important for the acceptance of JSON within the community due to the presence of this feature of JavaScript on all Web browsers today.

4.4 JSON versus XML

Many Web developers hate XML and JSON took his place as the format of choice for a lot of API work. There are some advantages to JSON, but some disadvantages, and XML has some problems, but the arguments are not as simple.

JSON is the new kid for portable data transfer and while JSON derives from JavaScript there is JSON parser support in other languages. Above go the pros and cons of using JSON or XML. It is this analysis that the large prime advantage is that JSON browsers have an easier time using it, since XML support in browsers is "irregular" and XML support in any way using the Document Object Models (DOM) model, which is a clunky bit. However XML has some distinct advantages with XPath and XSLT offering powerful ways to filter data sets to XML and transform them into other formats.

Pro XML

First, it has enough structure so it is possible to build rich data structures, and to add to that, it has some standard ways (like XHTML) with rich sets of attributes and elements that can be reused in multiple domains, and relations link pattern. The other great thing is the set of tools for extraction and processing, which are generally well designed and fairly complete. There are stream and DOM parsers widely available.

Against XML

XML's biggest disadvantage is that its parsers tend to be very large, although the large memory footprint of XML parsers may be reduced to a reasonable size by eliminating unneeded features, and the larger size of XML data records may be an issue for some applications. This can be dealt with by compressing the data before transmitting it or writing.

XML DTDs

The Document Type Definition (DTD), which XML inherited from SGML (Standard Generalized Markup Language), is an anomaly in many ways. First, it has a non-XML syntax, and then is necessary another set of parsers and tools to work with it. It has different functions that really need to be separated. The first function is as a document schema to validate against. Unfortunately it is not a very good schema language, as the constraints it can apply against documents are limited. Now exist the XML Schema and Relax-NG, which are better schema languages, but the DTD has a special position in the specification that is hard to put down.

Besides being a schema, the DTD can also define default values for attributes that the application should do just like they are in the document. This is the kind of thing that makes it difficult to preserve the textual form, as there is a syntactic but not semantic difference between certain attributes.

There are security issues due to the parsing issues with entities, which means that some parsers disable DTD analysis for security reasons. SOAP, for example, does not support DTDs. This is of course not in conformity, but clearly it's a good idea in many situations. DTDs are not namespace aware, making them unusable in many cases documents with namespaces.

XML entities

Then there are entities. The entities were designed to save typing for people, but they are not used for anything else, except for encodings of memorable characters outside the ASCII set. The thing about this use case is that it is perfectly acceptable to substitute the values for them, because they never change, whereas if is created a arbitrary entity in a DTD for the name of something that may be because it's to be used as search and replace function. This tool is an application level, and the application must use the XML tags to regulate this structure type in the user level.

XML entities can also be used as an inclusion mechanism. Again the DTD is not the place to define it. XInclude looks much better if this facility is necessary.

Entities can contain other entities, marking and so on. Recursion, marking unbalanced and are not allowed. This whole thing decisively contributes to the complexity of analysis, when the use case is entirely as character data.

XML namespaces

There are cases that make them very difficult to process in a healthy way, in particular, when reset the same namespace name to refer to multiple URIs in the same document, and the same URI will be with different names. This effectively means that all the processing necessary to refer to both the short name as the full name.

Data versus applications

Part of the conflict is due to the fact that XML is an application protocol, or a data format. Some of the pieces that have problems, such as entities, are actually part of an application data format, for a class of applications that run on the model in the minds of the XML designers, which in turn was based on real Standard Generalized Markup Language (SGML) applications. Simplicity is the winner here, complexity in a data format does not increase the wealth that can be expressed, uniform simple mechanisms can do that.

Pro JSON

JSON is simple and already well defined. It did not use to be clear in some places as a way to encode items outside the 2 byte UTF-16, encoding using the `\u` notation. Many people can generate invalid JSON (without quoted identifiers, use single quotation marks, the use of a Byte Order Mark (BOM) at the beginning), it seems like a problem. This should get better as new libraries come out. Introducing JSON parsers (in the style of

the HTML 5) seem to be unnecessary for a simple format that is usually generated by a computer. A strict JSON parser is not a lot of code.

JSON has a simple way to showing which of the permissible encodings it is in, based on the zero bytes at the beginning, as the first two characters must be in ASCII. Permitting a BOM and then allowing Unicode whitespace might be more standard, but the whitespace has no function except for use in text editors.

Despite attempts, for example, add an ECMAScript for XML (E4X) to a native XML format and a simple processing model of JavaScript, JSON, remains very easy in most languages to process, as it is built around structures that most languages have only natively while XML is not. Some languages have problems with a mismatch to JSON, but most are good. E4X, on the other side has the client-side security issues, and not seeing adoption, except in some applications on the server side.

Another advantage of using JSON instead of XML is in the inherent quality of data structuring. JSON is closer to being a HashMap as in java or an associative array in some other languages like PHP. Hence it does not require parsing libraries in an application as XML does. The time spent however, on parsing XML into the native language, is not so much significant considering the fast processing power of machines today.

The comparison of XML and JSON lies more on the network bandwidth aspect. JSON can take almost half of the bandwidth as XML for transferring the same data. It can achieve the same throughput as passing simple objects in the network in a distributed computing environment.

JSON's structure directly corresponds to basic data types of most of script languages such as array and map. It makes JSON API for script languages very simple e.g. just doing a parse, which takes JSON string and returns a map or an array. XML structure doesn't have such obvious correspondence, so it's necessary to use more complicated API such as DOM.

JSON has distinction between string, number and Boolean e.g. {"count": "1"} vs. {"count": 1}. XML document itself doesn't have the distinction. The distinction saves a bit of programmer's work to convert to appropriate type manually. You can define data type of values in XML with XML Schema, but it's complicated and not always available [78].

Against JSON

JSON is not as widespread as XML. Its tooling and language support still fall a bit short of XML, is not as powerful as XML in namespaces. XML throws everything and won't need to use every single one of its features, but when it's necessary, does it come in handy. JSON doesn't have a "Date" type, isn't efficient over the wire, binary protocols are better, JSON doesn't distinguish between floating point and decimal and JSON doesn't have a binary type, base64 is commonly used as a workaround (much like XML), but there's no standard annotation that differentiates it from a normal string [79].

There are some syntactically different representations: arbitrary white space, although this definition does not include the full Unicode whitespace characters and backslash escape, which can be represented for the most part directly in Unicode encoding of the document. Whitespace clearly needs to be preserved for readability and use of line oriented editors and tools. It is not clear how inconvenient it would be if the `\u` codes were normalized to Unicode which is the sane default.

Data model

The JSON data model is simpler than XML. XML nodes have attributes and children, JSON attributes or children, if the object to model is an attributes set and the array or list type to model an ordered list of children. This difference is not difficult circumvent.

Schemas

The validation schemes are a great thing and validation is an important activity. Essentially a validation scheme may need to be very complicated, but many are very simple. Having a choice of languages to express these constraints, it seems to be a good thing. The XML DTD (Document Type Definition) is very weak, and there should have been included in the language. Some restrictions are computationally complex and require a very expressive language.

The second function of a schema is interpretation, which may be related to validation in a field that should be read as a number say, and it read as a number. This is a different requirement, because in many cases it is object modeling and code generation, when a validated structure is then mapped to a native language object. These are conceptually separate processes, as a number can be constrained to be between 3 and 5 for reasons of domination but the representation to say that Java can be an integer, but it need not be. Of course, here the validation phase is essential for safety reasons, to prevent excesses and errors of the type, however these activities are conceptually different and may have different schemas.

JSON almost perfect

Binary data is a problem in both XML and JSON. It is necessary a lot of other formats for anything that has binary data, they are just much more efficient, even after compression. Thus, the ideas of a universal format will not happen. The fact that the XML is a lot easier to read, and JSON has a smaller footprint.

In a search for efficient serialization mechanisms for mobile devices, a test was been made between serialization with JSON and XML [80]. The XML serialization, which was compliant with XML schema used to describe the data types, uses Java features unsupported by the mobile device. In the opposite side, the JSON presents itself, as a valid alternative, following

a different paradigm, is a lightweight data-interchange text format that is completely language independent.

In this work are discussed two types of Web services, SOAP and REST. Were implemented, tested, and validated. Section 5 presents the results.

5. System Demonstration and Validation

This chapter focuses on the performance evaluation and validation of SapoFitness. Firstly it presents the application demonstration, through the windows that make up the program, starting on the splash screen, then the Login's screens, Profile's screens, Plan screen, Graphic's screens and the Physical Activity's screens. After the demonstration comes the validation, with the comparison between Web services and a users survey about the application.

5.1 Application Demonstration

As mentioned in previous chapters, the SapoFitness user's interface is simple, uses large buttons and includes appropriated size for fingers use. The application controls the user's weight, even running in the background, and alerting the user whenever is necessary, all in accordance with the customization specified for the user. Figure 23 shows the splash screen window and the pre-login window with two input options, with login or as a visitor in order to test the application, it's also possible register in “*meu sapo saúde*”.



Figure 23. Splash and Pre-Login Screens.

The Figure 24 presents the Login window where the user will enter his/her e-Mail address and corresponding password. After logging in, the application communicates with the SapoFitness Web services in order to obtain the respective credentials and load all the need data from the user, such as height, weight, age, sex and daily physical activities. These will determine the Body Mass Index (BMI) and the maximum daily calories to consume. Thus, automatically sets a user profile and the system basically informs the user about his/her obesity condition including more information surrounding, such as weight target, date for the purpose, and calories to consume.



Figure 24. Login Screen.

After load all the data, the application goes directly to the Profile window. The user Profile is the main window of SapoFitness (Figure 25). In background is portrayed an image of the user sex with its colour, pink for woman's and green for men's.

The first tab is the profile, the second is the food plan, the third concerns the evolution, the fifth to a shopping list, and the last is the physical activities (Figure 26).



Figure 25. Profile Screen - Actual Weight.

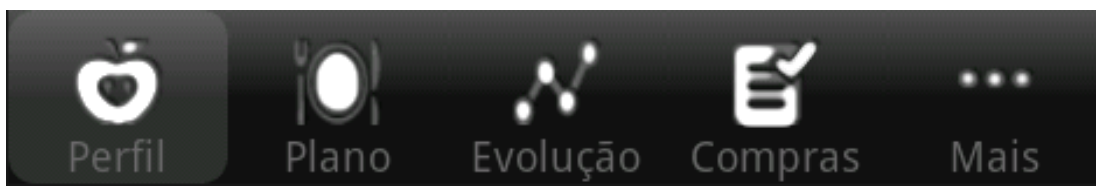


Figure 26. The Main Tab .

This window (Figure 27) is a Profile sub-window, where it's possible to watch the weight range in which the user must be maintained, also said the time information that the plan will take until it reaches the desired values for weight control.

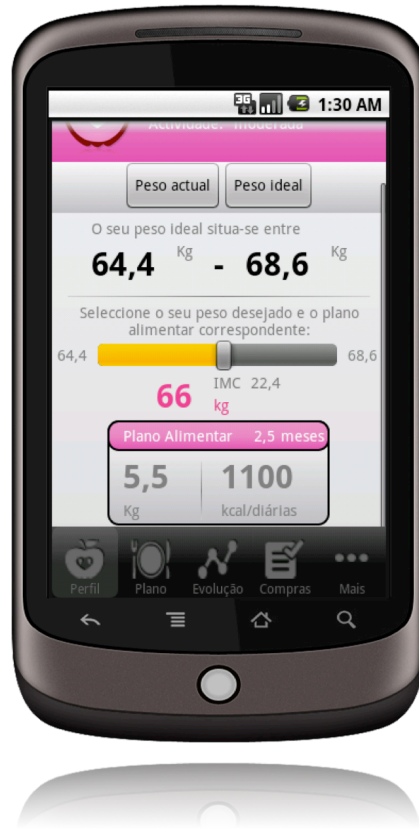


Figure 27. Profile Screen - Ideal Weight.

In the Edit Profile window (Figure 28), the user enters all necessary information such as is height, weight, age, and sex, for determining the BMI (body mass index) and the maximum daily calories must consume. Thus, automatically sets a sort of user profile and the system basically tells the user if he/she is or not on overweight and some more information surrounding, such as weight target, date for the purpose, and calories to consume.



Figure 28. Edit Profile.

In this window (Figure 29), the user records eating habits and observes its diet progress. The diary presents the current date and time, how many calories users have eaten until the current time of the day and how many calories he/she should consume on average per day. Meals and physical activity are information to be recorded daily. Based on this information, it is possible to calculate the daily calories consumed and also a weekly weight loss that can be validated or corrected by the user.



Figure 29. Plan Screen.

In the evolution window (Figure 30) is possible to see the changes in weight and BMI in the form of four types of graphics (Figure 31) and also allows the user to share this information in social networks.



Figure 30. Evolution Screen.

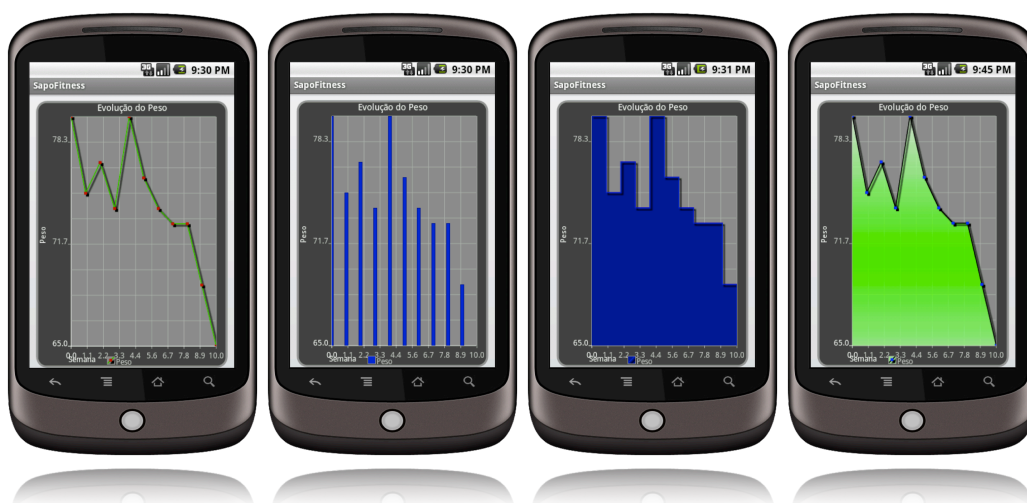


Figure 31. All four types of graphics Screens.

In the Physical Activities window (Figure 32), user can see daily activities and the corresponding calories expended. The user can choose different kind of exercises, its intensity, and see the corresponding burning calories depend on the time of exercise.



Figure 32. Physical Activities Screen.

5.2 Application Validation

The performance evaluation and real deployment of SapoFitness is presented on this section. The application validation was performed through exhaustive running experiments. Real devices were used in all the performed experiments, as may be seen in Figure 33. SapoFitness was deployed in three different devices with three different screen sizes, and the behaviour of the application user interface performed very well, as expected. General application functionality experiments focused on local application functionalities (like database testing, Web server connections, PDA screen sizes, and stability issues) were performed. These experiments enabled various debugging operations.

In order to validate the applicability of the proposal, several measurements from the SapoFitness during a regular day have been taken.

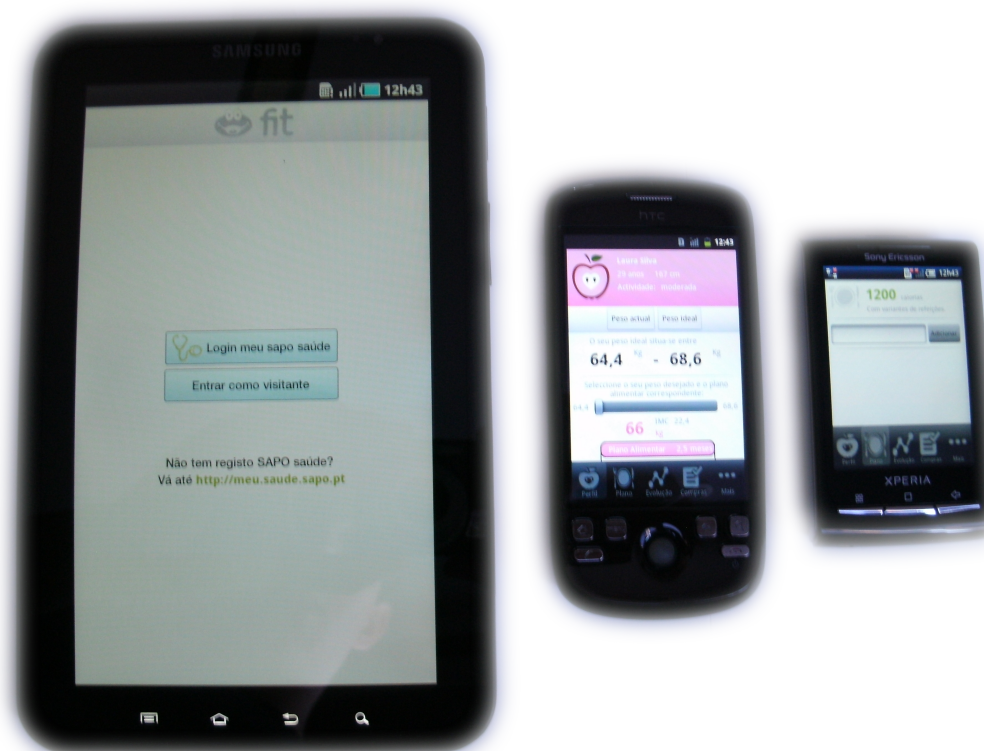


Figure 33. SapoFit in three different devices.

5.2.1 Comparison between Web services

A comparison was made between the response times of messages in SOAP and RESTful. Has been tested the times in SoapUI either in the application. SoapUI is an open source tool written in Java whose main function is testing and consume Web Services.

The times are not always the same, but on average these are the values presented in Table 1 and Figure 34. For the both services, the differences are small. But at all time the PhysicalActivity service, the application was always faster, this is due to the fact that the connection has been made in JSON in the application. Although a small difference in a world of greater number of bytes can make enough difference. The message length is shorter in JSON that XML, Figure 35, which also helps in reducing the response time.

Stay prove that RESTful web services are more suitable for mobile environment because it does not need a large weight Parsis, witch supports caching and will save the limited network bandwidth, Rise scalability and reliability, and will not consume the mobile resources.

Table 1. Request Comparison to Web service.

	GetUserProfile	PhysicalActivity
SoapUI Time (Milliseconds)	1384	348
SapoFitness Time (Milliseconds)	1797	295
SapoUi Size (Bytes)	1252	79516
SapoFitness Size (Bytes)	1252	48670

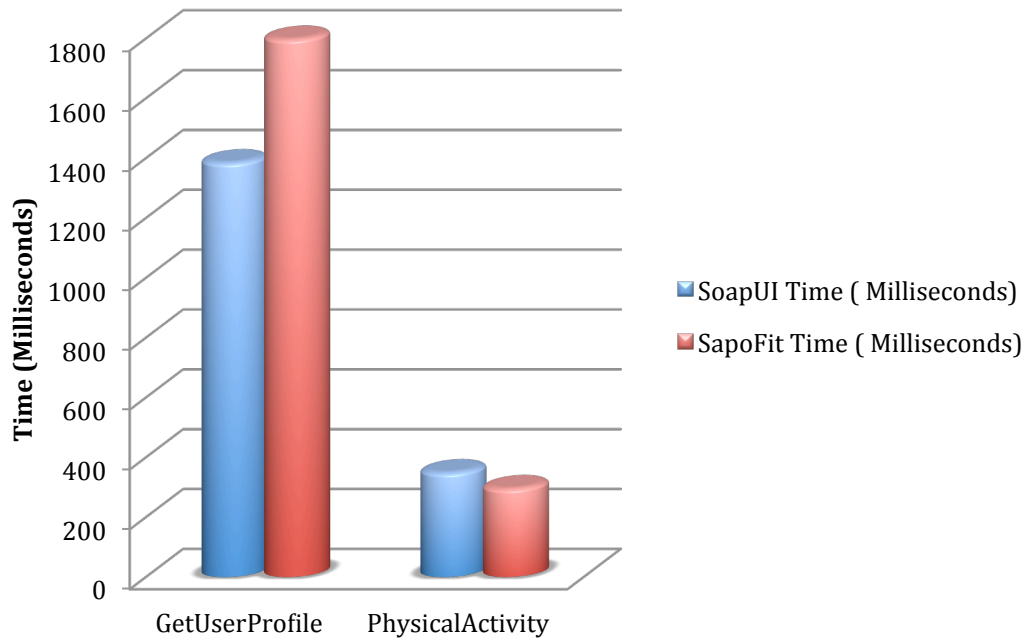


Figure 34. Webservice Request Time (Milliseconds).

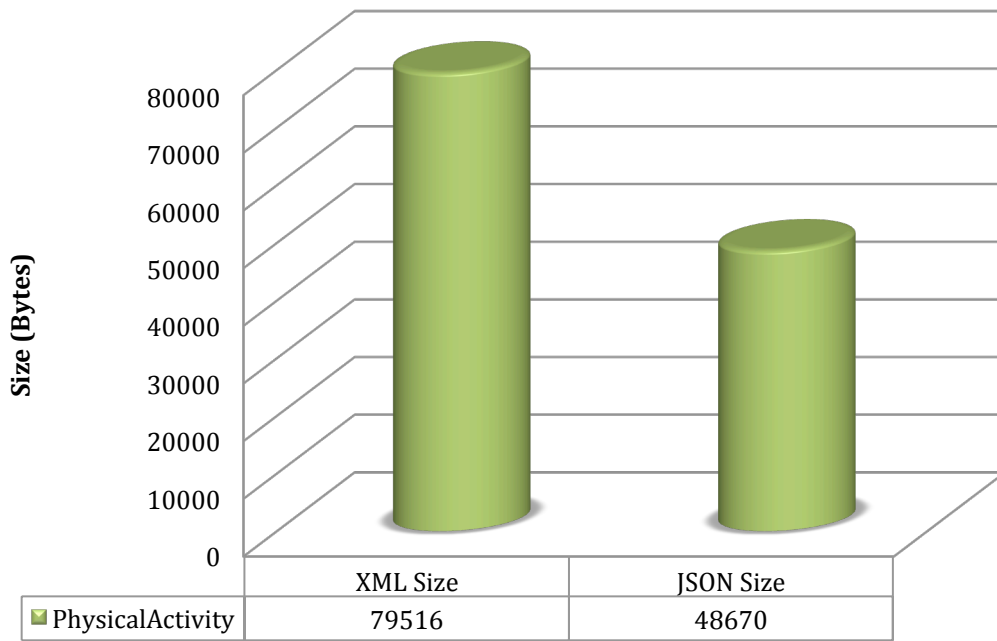


Figure 35. JSON and XML Message Length (Bytes).

5.2.2 User Survey

The surveys were made in SurveyMonkey which provides free online questionnaire and survey software. Sapofitness has been installed, configured, and accessible through the PDAs. A total of 106 users from University of Beira Interior, Portugal have answered the questionnaires. They have used the system for some time. After the experience, they completed a Sapofitness survey. The questions are shown in Table 2.

In Figure 36 and Table 3 it can be seen that the majority strongly agree that platform has an attractive design, the environment is user friendly and intuitive, navigation options are clear, consistent and text blocks are written in minimalist style, Fonts are easy to read in screen and the application help to understand the problem of obesity. Other big percentage of users thinks that the platform is very easy and the application is helpful for meal control. In almost questions the remaining percentage the users seem to agree with the questions. The only question that contains lower ratings is the 7, it is due to the connection to the server is still a test version.

Table 2. Sapofitness Survey Questions.

Question	Description
Question 1	Has the application an attractive design?
Question 2	Is the application easy to use?
Question 3	Is the application environment user friendly and intuitive?
Question 4	Are the Navigation options clear and consistent?
Question 5	Text blocks are written in minimalist style. Are they compact and useful?
Question 6	Are Fonts (style, color, saturation) easy to read in screen?
Question 7	Is the Feedback and response time of the application fast enough?
Question 8	Is the application helpful for meal control?
Question 9	Does the application help to understand the problem of obesity?

Table 3. Table of SapoFitness survey.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Strongly agree	78	49	70	68	55	76	48	45	71
Agree	20	50	14	31	22	16	33	44	34
Tend to agree	8	7	22	7	29	14	21	17	1
Undecided	0	0	0	0	0	0	2	0	0
Tend to disagree	0	0	0	0	0	0	1	0	0
Disagree	0	0	0	0	0	0	1	0	0
Strongly disagree	0	0	0	0	0	0	0	0	0

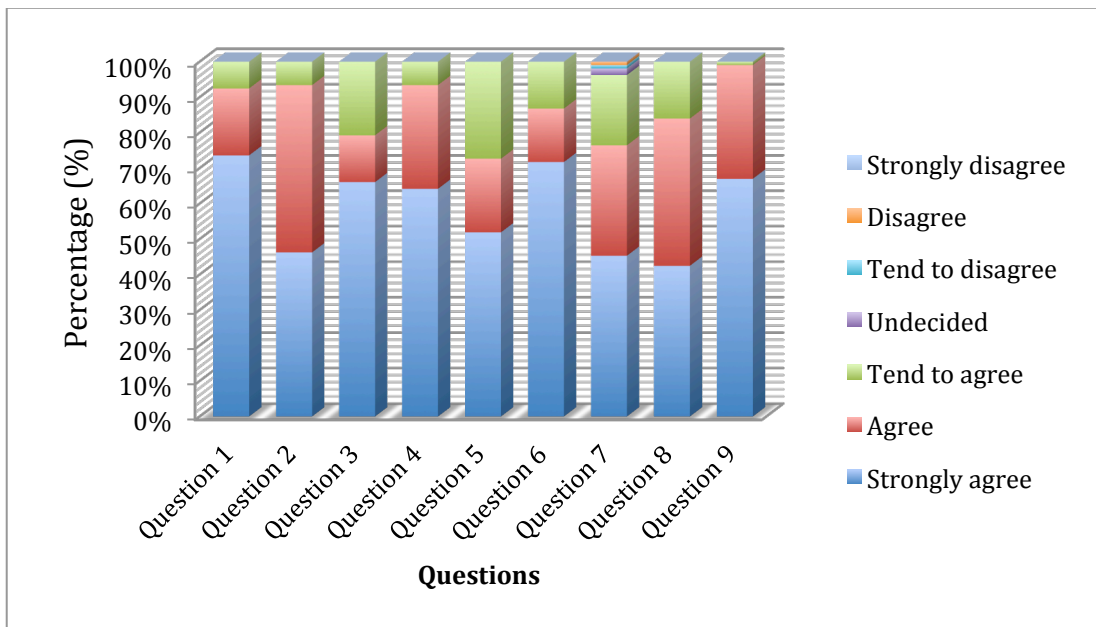


Figure 36. Results of SapoFitness Survey.

6. Conclusions and Future Work

6.1 Conclusions

This Chapter presents a synthesis of the main achievements and points to several directions for future work. The main objective of this dissertation was the development of a ubiquitous solution for a Mobile Health application. It was carried out with the construction of a mobile API, providing users to monitor and control their weight. Therefore all the dissertation objectives were successfully accomplished and the all intermediate objectives were successfully achieved.

After introducing and presenting the topic of this dissertation and define its objectives, Chapter two presented the revision on obesity review of the state of the art on EHR/PHR, M-Health, E-Health and U-Health.

Chapter three presented the requirements analysis taken before the mobile platform development process. Detailing the essentials requirements for all the development process and next the diagrams.

In chapter four presented the mobile framework, the Web services, an intensive study of Web services and their related data formats, XML and JSON. Where has addressed the differences between SOAP and REST service, and also the differences between XML and JSON. The Sending and receiving a SOAP message is not always the best way for applications to communicate. Sometimes a simple REST interface and a simple text response do the trick and saves time and resources in the process.

Regarding to the mark-up languages, the DTDs and entities are not used in many cases now. They may be in some publishing applications, especially those based on SGML, but the web document architecture does not use them significantly. Namespaces are used in a particular way, usually. HTML5 has shown what the logic of human readability and write ability implies, which is a non-XML language. The great advantage of XML is the variety of ways in which it can be processed, but issues such as security to hostile documents, parsing complexity, performance, and ease of processing really matter a lot, and despite many weaknesses JSON is showing the way of radical simplicity. But a simplified XML would be no more complex than JSON even with, the advantages of richer tool support, and widespread use. Security is limiting processing, and the traditional publishing applications that historically used more of the functionality could change too, although more slowly. Will JSON replace XML? It is difficult to predict, JSON is not perfect, the fact that the XML is a lot easier to read, and JSON has a smaller footprint. But JSON is more practical and consumes fewer resources.

The chapter five was presented the architecture of the android, the demonstration of the application and its validation. A comparison was made between Web services, SOAP and RESTful and where the big difference was the size of the messages. It has been proved that RESTful web services are more suitable for mobile environment because it does not need a large weight Parsis, which supports caching and will save the limited network bandwidth, Rise scalability and reliability, and will not consume the mobile resources.

It was also done a survey, where the feedback received was very good, and the evaluated points in the survey were answered with very positive note.

6.2 Future Works

To conclude this work, it just remains to suggest future research directions based on current work:

- Increase the number of mobile platforms where the application can be executed.
- Include the use of geo-referential technologies and algorithms to provide context and location-aware services. These services could enable collaboration and cooperation among users, socializing and helping each other to reach pre-defined thresholds.
- There are many advantages in m-health, and new possibilities to try to get to where it was never reached before, cover more population and allowing a better health.

References

- [1] B. Le Moullec and P. Ray, "Issues in E-Health Cost Impact Assessment," in *IFMBE Proceeding of the World Congress on Medical Physics and Biomedical Engineering*, pp. 223-226, Jul. 2009.
- [2] Chunming Gao, Fanyu Kong, and Jindong Tan, "HealthAware: Tackling obesity with health aware smart phone systems," *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pp. 1549-1554, 2009.
- [3] S. Akter and J. D'Ambra, *User Perceived Service Quality of MHealth Services in Developing Countries: Research Paper*. 2010.
- [4] J. Landay and A. Joseph, "Guest Editors' Introduction: Smarter Phones," *Pervasive Computing*, 2009.
- [5] S. Tachakra, X. Wang, R. Istepanian, and Y. Song, "Mobile e-health: the unwired evolution of telemedicine," *Telemedicine Journal and E-health*, vol. 9, no. 3, pp. 247-257, 2003.
- [6] Y. Zhang and N. Ansari, "Wireless telemedicine services over integrated IEEE 802.11/WLAN and IEEE 802.16/WiMAX networks," *Wireless Communications*, 2010.
- [7] J. Li, L. Land, and P. Ray, "E-Health readiness framework from Electronic Health Records perspective," *International Journal of ...*, 2010.
- [8] Gartner [Online]. Available: <http://www.gartner.com/it/page.jsp?id=1529214> [Accessed: December 2010].
- [9] Mobihealthnews, "The Fastest Growing and Most Successful Health & Medical Apps," in *Mobihealthnews 2010 Report*, November 2010.
- [10] Research2guidance, "Mobile Health Market Report The Impact of Smartphone Applications on the Mobile Health Industry," in *Mobile Health Market Report 2010-2015*, November 2010.

- [11] U.S. Mobile Apps [Online]. Available: <http://apps.usa.gov/> [Accessed: April 2011].
- [12] ISO Technical Report (ISO-TR20514), Health Informatics, Electronic Health Record, Definition, Scope and Context, Clause 4.4.
- [13] C. C. Tsai et al., "Usability and Feasibility of PmEB: A Mobile Phone Application for Monitoring Real Time Caloric Balance," *Pervasive Health Conference and Workshops, 2006*, pp. 1-10, 2006.
- [14] B. Caballero, "The Global Epidemic of Obesity: An Overview," *Epidemiologic Reviews*, vol. 29, no. 1, pp. 1-5, May. 2007.
- [15] Center for Disease Control [Online]. Available: <http://www.cdc.gov/obesity/index.html>. [Accessed: January 2011].
- [16] R. Novotny et al., "Hispanic and Asian Pubertal Girls Have Higher Android/Gynoid Fat Ratio Than Whites|[ast]|," *Obesity*, vol. 15, no. 6, pp. 1565-1570, Jun. 2007.
- [17] H. Hubert, M. Feinleib, P. McNamara, and W. Castelli, "Obesity as an independent risk factor for cardiovascular disease: a 26- year follow-up of participants in the Framingham Heart Study," *Circulation*, vol. 67, no. 5, p. 968, May. 1983.
- [18] M. Zamboni et al., "Health consequences of obesity in the elderly: a review of four unresolved questions," *International Journal of Obesity*, vol. 29, no. 9, pp. 1011-1029, May. 2005.
- [19] M. B. E. Livingstone, P. J. Robson, and J. M. W. Wallace, "Issues in dietary intake assessment of children and adolescents," *British Journal of Nutrition*, vol. 92, no. 2, pp. S213-S222, Oct. 2004.
- [20] P. Zuehlke, J. Li, A. Talaei-Khoei, and P. Ray, "A Functional Specification for mobile eHealth (mHealth) Systems," in *2009 11th International Conference on e-Health Networking, Applications and Services (Healthcom 2009)*, 2009, pp. 74-78.
- [21] V. Chan, P. Ray, and N. Parameswaran, "Mobile e-Health monitoring: an agent-based approach," *Communications, IET*, vol. 2, no. 2, pp. 223-230, 2008.
- [22] D. Vatsalan et al., "Mobile Technologies for Enhancing eHealth Solutions in Developing Countries," in *2010 Second International Conference on eHealth, Telemedicine, and Social Medicine (ETELEMED)*, 2010, pp. 84-89.
- [23] S. Jan, I. A. Shah, and H. Al-Raweshidy, "Performance Analysis of

- Proactive and Reactive Routing Protocols for Mobile Ad-hoc Grid in e- health Applications,” in *2009 International Conference on Communication Software and Networks*, 2009, pp. 484-488.
- [24] R. Sulaiman, X. Huang, and D. Sharma, “E-health Services with Secure Mobile Agent,” in *2009 Seventh Annual Communication Networks and Services Research Conference (CNSR)*, 2009, pp. 270-277.
- [25] R. Sulaiman, D. Sharma, W. Ma, and D. Tran, “10th International Conference on Advanced Communication Technology,” in *10th International Conference on Advanced Communication Technology*, 2008, pp. 999-1004.
- [26] Fengqing Zhu et al., “The Use of Mobile Devices in Aiding Dietary Assessment and Evaluation,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 4, pp. 756-766, 2010.
- [27] A. Khalil and S. Glal, “StepUp: A step counter mobile application to promote healthy lifestyle,” *Current Trends in Information Technology (CTIT), 2009 International Conference on the*, pp. 1-5, 2009.
- [28] N. Joo and B. Kim, “Mobile phone short message service messaging for behaviour modification in a community-based weight control programme in Korea,” *Journal of Telemedicine and Telecare*, vol. 13, no. 8, p. 416, Dec. 2007.
- [29] K. Patrick et al., “A Text Message-Based Intervention for Weight Loss: Randomized Controlled Trial,” *Journal of medical Internet research*, vol. 11, no. 1, 2009.
- [30] *U.S Mobile Apps - MedlinePlus Mobile [Online]*. Available: <http://apps.usa.gov/mobile-medline-plus/> [Accessed: April 2011].
- [31] *U.S Mobile Apps - BMI Calculator [Online]*. Available: <http://apps.usa.gov/bmi-app/> [Accessed: April 2011].
- [32] *U.S Mobile Apps - AIDSinfo HIV/AIDS Glossary [Online]*. Available: <http://apps.usa.gov/aidsinfo-hiv-aids-glossary/> [Accessed: April 2011].
- [33] *U.S Mobile Apps - Find a Health Center [Online]*. Available: <http://apps.usa.gov/find-a-health-center/> [Accessed: April 2011].
- [34] John D. Halamka, “Mobile Applications for Medical Education,” [Online]. Available: <http://geekdoctor.blogspot.com/2011/04/mobile-applications-for->

- medical.html* [Accessed: April 2011].
- [35] *DynaMed* [Online]. Available: <http://www.ebscohost.com/dynamed/default.php> [Accessed: April 2011].
- [36] H. Barton, "DynaMed," *Journal of the Medical Library Association*, vol. 93, no. 3, p. 412, Jul. 2005.
- [37] *Unbound Medicine - uCentral* [Online]. Available: <http://www.unboundmedicine.com/solutions/ucentral> [Accessed: April 2011].
- [38] *VisualDx - Visual diagnostic decision support - VisualDx Mobile*, [Online]. Available: <http://www.visualdx.com/mobile/> [Accessed: April 2011].
- [39] *Epocrates* [Online]. Available: <http://www.epocrates.com/company/> [Accessed: April 2011].
- [40] G. Liberman - *iRadiology* [Online]. Available: <http://itunes.apple.com/us/app/iradiology/id346440355?mt=8> [Accessed: April 2011].
- [41] T. D. Gunter and N. P. Terry, "The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions," *Journal of medical Internet research*, vol. 7, no. 1, Mar. 2005.
- [42] "HIMSS - Electronic Health Record (EHR)," *HIMSS Electronic Health Record* [Online]. Available: http://www.himss.org/ASP/topics_ehr.asp. [Accessed: March 2011].
- [43] M. Mosmondor, I. Benc, S. Desic, and A. Grguric, "A feasibility study for the integration of a remote patient monitoring solution with electronic health record system," in *MIPRO, 2010 Proceedings of the 33rd International Convention*, 2010, pp. 360-366.
- [44] H. Le, M. Kuttel, and G. Chandran, "An Electronic Health Care - Cardiac Monitoring System," in *2010 International Conference On Communications Workshops*, 2010, pp. 1-5.
- [45] P. Belsis, D. Vassis, S. Gritzalis, and C. Skourlas, "W-EHR: A Wireless Distributed Framework for secure dissemination of Electronic Healthcare Records," in *2009 16th International Conference on Systems, Signals and Image Processing*, 2009, pp. 1-4.
- [46] W. D. Yu and M. Chan, "A Modeling Approach of Web-Based Mobile

-
- Parking Guidance System and Secured EHR in e-Healthcare,” in *2009 11th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2009, pp. 116-123.
- [47] I. Chang, S. Hsiao, H. Hsu, and T. Chen, “Building mPHR to Assist Diabetics in Self-Healthcare Management,” in *2010 7th International Conference on Service Systems and Service Management (ICSSSM)*, 2010, pp. 1-5.
- [48] P. R. Marupally, V. Paruchuri, and S. Chellappan, “Privacy Preserving Portable Health Record (P3HR),” in *2009 International Conference on Network-Based Information Systems (NBIS)*, 2009, pp. 310-315.
- [49] M. Wang, P. H. Tsai, J. W. S. Liu, and J. K. Zao, “Wedjat: A Mobile Phone Based Medicine In-take Reminder and Monitor,” in *2009 Ninth IEEE International Conference on Bioinformatics and BioEngineering (BIBE)*, 2009, pp. 423-430.
- [50] M. Weiser, “The world is not a desktop,” *interactions*, vol. 1, no. 1, pp. 7-8, Jan. 1994.
- [51] J. Zhou, E. Gilman, M. Ylianttila, and J. Rieki, “Pervasive Service Computing: Visions and Challenges,” in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 2010, pp. 1335-1339.
- [52] L. Little and P. Briggs, “Pervasive healthcare: the elderly perspective,” in *Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '09*, New York, New York, USA, 2009, pp. 1-5.
- [53] G. Graschew, T. A. Roelofs, S. Rakowsky, and P. M. Schlag, “Design of Satellite-Based Networks for u-Health - GALENOS, DELTASS, MEDASHIP, EMISPHER,” in *9th International Conference on e-Health Networking, Application and Services*, 2007, pp. 168-173.
- [54] W. Shin, Y. D. Cha, and G. Yoon, “ECG/PPG Integer Signal Processing for a Ubiquitous Health Monitoring System,” *Journal of Medical Systems*, vol. 34, no. 5, pp. 891-898, May. 2009.
- [55] T. Suzuki and Y. Nakauchi, “Dosing monitoring system using iMec and ubiquitous sensors,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 6163-6166.
- [56] Kwang Suk Park, “Nonintrusive measurement of biological signals for ubiquitous healthcare,” in *Annual International Conference of the*
-

- IEEE Engineering in Medicine and Biology Society*, 2009, pp. 6573-6575.
- [57] H. J. Yoo et al., "A Ubiquitous Chronic Disease Care system using cellular phones and the internet," *Diabetic Medicine*, vol. 26, no. 6, pp. 628-635, Jun. 2009.
- [58] O. Kwon, S. Shin, S. Shin, and W. Kim, "Design of U-Health System with the Use of Smart Phone and Sensor Network," in *Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications*, 2010, pp. 1-6.
- [59] T. Taleb, D. Bottazzi, and N. Nasser, "A Novel Middleware Solution to Improve Ubiquitous Healthcare Systems Aided by Affective Information," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 335-349, 2010.
- [60] J. J. Rutherford, "Biomedical Engineering and the Obesity Epidemic: Treatments for Weight Reduction," *IEEE Engineering in Medicine and Biology Magazine*, vol. 29, no. 1, pp. 24-30, Jan. 2010.
- [61] A. Ershow, J. Hill, and J. Baldwin, "Novel engineering approaches to obesity, overweight, and energy balance: public health needs and research opportunities," *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, vol. 2, pp. 5212-5214, 2004.
- [62] J. Pollak, G. Gay, and Byrne, "It's Time to Eat! Using Mobile Games to Promote Healthy Eating," *Pervasive Computing, IEEE*, vol. 9, no. 3, pp. 21-27, 2010.
- [63] J. M. Silva, S. Zamarripa, E. B. Moran, M. Tentori, and L. Galicia, *Promoting a healthy lifestyle through a virtual specialist solution*. ACM, 2006, pp. 1867-1872.
- [64] Yongwon Jang, Seungchul Shin, and Seunghwan Kim, "A basic study for automatic waist circumference measurement using magnet with waist belt to assist obese management in daily life," *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 2333-2336, 2008.
- [65] J. B. Oladosu, F. A. Ajala, and O. O. Popoola, "On the use of Web services Technology in E-Health," *Journal of Theoretical and Applied Information Technology*, vol. 12, pp. 94-103, Mar. 2010.
- [66] "What is Android? | Android Developers," [Online]. Available: <http://developer.android.com/guide/basics/what-is-android.html>[Accessed: June 2011].

-
- [67] *Web Service Architecture [Online]. Available: <http://www.w3.org/TR/ws-arch/> [Accessed: June 2011].*
- [68] Y. Liu, Q. Wang, and M. Zhuang, "Reengineering Legacy Systems with RESTful Web Service," *Annual IEEE International Computer Software and Applications Conference*, pp. 785-790, 2008.
- [69] "Web Service Definition Language (WSDL)," *Web Service Description Language [Online]. Available: <http://www.w3.org/TR/wsdl> [Accessed: June 2011].*
- [70] "What Is Windows Communication Foundation," *Windows Communication Foundation [Online]. Available: <http://msdn.microsoft.com/en-us/library/ms731082.aspx> [Accessed: June 2011].*
- [71] "Introducing Windows Communication Foundation in .NET Framework 4," *Windows Communication Foundation Service [Online]. Available: <http://msdn.microsoft.com/library/ee958158.aspx> [Accessed: June 2011].*
- [72] R. Fielding, "Architectural styles and the design of network-based software architectures," *PhD Dissertation, University of California, Irvine, California, USA, 2000.*
- [73] H. Hamad and M. Saad, "Performance evaluation of restful web services for mobile devices," *International Arab Jourlan of e-Technology, Vol 1, No 3, Jan. 2010.*
- [74] F. AlShahwan and K. Moessner, "Providing SOAP Web Services and RESTful Web Services from Mobile Hosts," in *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on Internet and Web Applications and Services, 2010*, pp. 174-179.
- [75] "Extensible Markup Language (XML)," *Extensible Markup Language [Online]. Available: <http://www.w3.org/XML/> [Accessed: June 2011].*
- [76] *Extensible Markup Language 1.0 Recommendation [Online]. Available: <http://www.w3.org/TR/REC-xml/> [Accessed: June 2011].*
- [77] "JSON," *JavaScript Object Notation [Online]. Available: <http://www.json.org/> [Accessed: June 2011].*
- [78] "JSON: The Fat-Free Alternative to XML," *JSON: The Fat-Free Alternative to XML [Online]. Available:*
-

<http://www.json.org/fatfree.html> [Accessed: June 2011].

- [79] “The limitations of JSON (The Sun BabelFish Blog),” *The Limitations of JSON* [Online]. Available: http://blogs.oracle.com/bblfish/entry/the_limitations_of_json [Accessed: June 2011].
- [80] G. Bartolomeo and E. Casalicchio, “Design and Development Tools for Next Generation Mobile Services,” *International Conference on Software Engineering Advances (ICSEA)*, p. 16, 2007.