FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Phone-based Heart and Lung Functions Monitor

João Filipe Trindade da Silva

Master in Informatics and Computing Engineering

Supervisor: António Miguel Pontes Pimenta Monteiro (Ph.D.)

July 2011

Phone-based Heart and Lung Functions Monitor

João Filipe Trindade da Silva

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Pedro Alexandre Guimarães Lobo Ferreira do Souto (PhD)

External Examiner: Álvaro Manuel Reis da Rocha (PhD)

Supervisor: António Miguel Pontes Pimenta Monteiro (PhD)

July 2011

Abstract

The growing aging people coupled with the rise in the incidence of people living in chronic health conditions is creating an unsustainable burden on health and social care services. On the other hand, mobile technologies have been impressively growing in most countries for the last years, as greater efforts are devoted into research work in this area. And mobile phones are not an important tool only in urban regions, more and more people from rural areas are making use of them.

On the other hand, wireless systems provide "anytime and anywhere" services, allowing data entered somewhere else to be accessed by the healthcare professionals at the point of care. It also provides a way to remotely monitor a patient. This is extremely important as it allows patient mobility and efficient response in emergency situations. New applications and industries will continue to address healthcare issues in the coming years and wireless monitors will be commonly used in hospitals and for home monitoring.

This project aims to make some improvement in this still gapped-area, by studying the available technologies and combine them to develop a solution that will provide people a way to monitor their heart and lung functions everyday, taking greater care on their health status. The solution will allow people to perform daily tests and build a history of results which will be shared with health care professionals. By doing this, we will give an important contribution in the prevention of serious diseases, through early diagnosis.

In this dissertation, a review of the state of the art of m-Health systems is presented, including the theoretical background, the available technologies and existing related work in this area. The problems that this research tried to solve in the context of a project undertaken in Fraunhofer Portugal research center are also presented as well as a solution approach and some results.

Resumo

O crescente envelhecimento da população e o aumento na incidência de pessoas a viver em condições crónicas de saúde tem vindo a criar um custo insustentável para a saúde e assistência social. Por outro lado, as tecnologias móveis vêm evoluindo de forma impressionante na maioria dos países nos últimos anos, resultante do crescente esforço dedicado em trabalho de investigação nesta área. Os aparelhos móveis deixaram de ser um instrumento importante somente em regiões urbanas, cada vez mais pessoas das zonas rurais estão a fazer uso deles.

Por outro lado, os sistemas sem fios são projectados para fornecer serviços "a qualquer hora e em qualquer lugar", possibilitando a introdução de dados seja onde for e o seu acesso por parte dos profissionais de saúde no local de atendimento. Estes serviços permitem assim a monitorização de um paciente à distância, o que é de verdadeiramente assinalável visto não apenas oferecerem mobilidade ao paciente como também permitem uma resposta eficiente em situações de emergência. Novas aplicações e indústrias continuarão a abordar e a tentar solucionar questões relacionadas com a saúde nos próximos anos e os aparelhos de monitorização à distância vão ser usados com frequência em hospitais e para acompanhamento do paciente em sua casa.

Este projecto pretende dar um contributo nesse sentido, estudando as tecnologias disponíveis e combinando-as por forma a desenvolver uma solução que irá proporcionar às pessoas uma maneira de monitorar diariamente as funções do coração e dos pulmões, assumindo assim uma maior responsabilidade e um maior cuidado no seu estado de saúde. A solução permitirá às pessoas realizarem testes diários e construir um histórico de resultados que serão partilhados com profissionais de saúde. Ao fazer isso, estaremos a dar um passo importante na prevenção de doenças graves, através do diagnóstico precoce.

Nesta dissertação, é apresentada uma revisão do estado da arte dos sistemas móveis de saúde, incluindo a fundamentação teórica, as tecnologias disponíveis e trabalhos existentes relacionados. São ainda enunciados os problemas que este trabalho de pesquisa tentou solucionar no contexto de um projecto desenvolvido no centro da investigação Fraunhofer Portugal, bem como a abordagem feita e alguns resultados obtidos.

Acknowledgements

My special thanks to my supervisor Miguel Pimenta Monteiro for his guidance and advices that were very important for my work. I want also to thank Fraunhofer Portugal and particularly my co-supervisor Dirk Elias for giving me, on one hand, the opportunity to work on this project and, on the other hand, the means to accomplish it. I would like to give a special thank to Luis Carvalho. His help and criticism were fundamental. Also, I would like to acknowledge Professor Dr. José Cardoso and Dra. Carla Sousa for their time and help. Finally, I would like to thank my family and friends for their comprehension and support during all this journey.

João Filipe Trindade da Silva

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Goals and Motivation . . . 2
- 1.3 Dissertation Overview . . . 2

2 State of the Art **3**
- 2.1 Mobile Devices and m-Health . . . 3
  - 2.1.1 Mobile Devices for Healthcare Professionals . . . 3
  - 2.1.2 Emergent Technology in m-Health . . . 4
  - 2.1.3 Classification of m-Health Systems . . . 4
  - 2.1.4 m-Health Potential . . . 5
- 2.2 Cardiopulmonary System . . . 5
  - 2.2.1 Respiratory System . . . 6
  - 2.2.2 Chronic Obstructive Pulmonary Disease (COPD) and Heart Failure 9
  - 2.2.3 Cardiovascular System . . . 11
  - 2.2.4 Atrial Fibrillation . . . 14
- 2.3 Sound Signals Capture and Processing . . . 16
  - 2.3.1 Signals . . . 16
  - 2.3.2 Digital Signal Processing . . . 17
  - 2.3.3 Application Areas . . . 17
  - 2.3.4 Sampling . . . 18
  - 2.3.5 The Fourier Transform . . . 19
  - 2.3.6 Discrete Fourier Transform . . . 19
  - 2.3.7 Fast Fourier Transform . . . 21
- 2.4 Technologies . . . 22
  - 2.4.1 Android . . . 22
  - 2.4.2 Communicating with Web Services . . . 25
  - 2.4.3 Google Health . . . 29
- 2.5 Related Work . . . 29
  - 2.5.1 iStethoscope . . . 29
  - 2.5.2 Instant Heart Rate . . . 30

3 Requirements and Architecture **31**
- 3.1 Architecture . . . 31
- 3.2 Requirements . . . 32
  - 3.2.1 Functional Requirements . . . 32
  - 3.2.2 Non-Functional Requirements . . . 33

3.2.3 Why Android . . . 33
3.2.4 The stethoscope . . . 34
3.3 Functionalities . . . 35

4 System Implementation **37**
4.1 Audio Recording . . . 37
4.1.1 Audio Properties . . . 39
4.1.2 Prepared for Multi-core Smartphones Generation . . . 40
4.2 Digital Signal Processing Algorithms . . . 41
4.2.1 Heart Sounds . . . 41
4.2.2 Lungs Sounds . . . 48
4.3 Mobile application . . . 52
4.3.1 The Graphical User Interface . . . 52
4.3.2 Local Storage . . . 60
4.4 REST Web Services Implementation . . . 60
4.4.1 Data Compression - ZIP archive . . . 60
4.4.2 Web Service Specification . . . 62
4.4.3 JSON . . . 64
4.5 Integration with Google Health . . . 65

5 Tests and Results **69**
5.1 Algorithms Testing . . . 69
5.2 Google Health Integration Testing . . . 70
5.3 Graphical User Interface . . . 71
5.4 Integration and System Testing . . . 71
5.5 DSP benchmarking . . . 72

6 Conclusions **73**
6.1 State of the project . . . 74
6.2 Future work . . . 75

References **77**

List of Figures

2.1 Patient's medical history. 7
2.2 Admissions of patients with COPD in portuguese hospitals [dA08]. . . . 10
2.3 The heart anatomy. 11
2.4 Points of auscultation of the heart. 12
2.5 Maximal intensity and radiation of isolated systolic murmurs 13
2.6 Comparison between a normal sinus rhythm and atrial fibrillation 14
2.7 Analogy and Digital Signals. 16
2.8 Signal sampling. 18
2.9 The Discrete Fourier Transform. 20
2.10 From time-domain to frequency representation with FFT. 21
2.11 The FFT decomposition 22
2.12 Flow diagram of the FFT 22
2.13 Android OS architecture 23
2.14 SOAP model 26
2.15 SOAP message 26
2.16 Rest web services concept 28
2.17 Heart wave form on iStethoscope 29
2.18 Heartbeat monitoring with Instant Heart Rate 30

3.1 High-level architecture of the solution. 31
3.2 The electronic low-cost stethoscope used. 34
3.3 System's Use Cases Diagram. 35

4.1 AudioRecord and MediaRecord communication. 38
4.2 Heartbeat sound in time and frequency representation. 44
4.3 Collecting the peaks of the signal after filtering. 45
4.4 Atrial fibrillation appearance in the time-domain. 47
4.5 Crackles sample 48
4.6 Crackles' frequency components and timing 50
4.7 Crackles recording before and after filtering. 51
4.8 Look of the Start menu. 53
4.9 Toast notification display. 53
4.10 Choosing a Google account. 54
4.11 Result after choosing the account. 54
4.12 Help on a button. 55
4.13 Some guidelines before the exam. 55
4.14 Exam screen before record. 56

4.15 Exam screen while recording. 56
4.16 A short tutorial demo explaining how to auscultate. 56
4.17 History folders listing in landscape mode - grid view. 57
4.18 History folders listing in portrait mode. 57
4.19 List of exams from April 2011. 58
4.20 Results of an exam. 58
4.21 Weight evolution on the last 8 days. 59
4.22 Changing the period of time. 59
4.23 Credits of the application. 59
4.24 Commenting the application. 59
4.25 Local database tables. 60
4.26 UML Sequence Diagram of the system. 61
4.27 JSON syntax 65
4.28 Choosing an account to store data in Google Health. 65
4.29 Tree diagram with the CCR elements supported by Google Health. 66
4.30 Google Health: Table with heart rates sent from the mobile. 67
4.31 Weight progress tracking in Google Health. 67

List of Tables

2.1 Typical Chest Examination Findings in Selected Clinical Conditions . . . 9
2.2 Major causes of death in the world. 10

4.1 Heartrate variation in atrial fibrilation. 47
4.2 Distinguishing between fine and coarse crackles. 49

5.1 Average execution time (in milliseconds) for 1D complex forward FFT (powers of two). 72

LIST OF TABLES

Acronyms

3G	3^{rd} Generation Mobile Telecommunications
3GP	3GPP (Third Generation Partnership Project) File Format
4G	4^{th} Generation Mobile Telecommunications
AAC	Advanced Audio Coding
ACM	Apache Commons Math
AF	Atrial Fibrillation
AMR	Adaptive Multi-Rate
API	Application Programming Interface
CCR	Continuity of Care Record
COPD	Chronic Obstructive Pulmonary Disease
CRM	Customer relationship management
ECG	Electrocardiogram
EDGE	Enhanced Data rates for GSM Evolution
EEG	Electroencephalography
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
FhP	Fraunhofer Portugal AICOS
FFT	Fast Fourier Transform
GH	Google Health
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
IDE	Integrated Development Environment
HF	Heart Failure
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IPC	Inter-Process Communication
JAX-RS	Java API for RESTful Web Services
JNI	Java Native Interface
JPEG (JPG)	Joint Photographic Experts Group
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Experts Group
MP3	MPEG-1 or MPEG-2 Audio Layer III
m-Health	Mobile Health
NDK	Native Development Kit
OpenGL	Open Graphics Library

PCM	Pulse-Code Modulation
PNG	Portable Network Graphics
RPC	Remote Procedure Call
R&D	Research and Development
S1	1^{st} heart tone
S2	2^{nd} heart tone
SD	Secure Digital Card
SOAP	Simple Object Access Protocol
UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language
W3C	World Wide Web Consortium
WAR	Web application ARchive
WAVE	WAVEform audio format
WSDL	Web Service Description Language

ACRONYMS

Chapter 1

Introduction

Mobile Phones have become an integral part of most people's life. The majority of people today owns a mobile phone. There are many reasons behind this increasing popularity of mobile phones, although the main one remains to be to keep in touch with family, friends and work contacts. However, the continuous growth of the Web and the services it provides, as well as the reliability it offers, have been diversifying the reasons and goals that drive people to purchase mobile devices.

Nowadays, these devices are no longer used only with the purpose of communication but are also loaded with a great diversity of entertainment features. The latest mobile phones are wrapped with an ample variety of features such as high resolution camera with video recorder, Internet, MP3 player, radio, GPS and gaming. Apart from this, these devices also come up with ultimate connectivity options like Bluetooth, EDGE, GPRS, etc.

Though the generalized crisis, the market for mobile device applications didn't stop growing. According to International Data Corporation, a market research firm specializing in information technology, and if the expectations were met, the worldwide market reached $3.5 billion by the end of 2010 [oCoEoC].

The creation of applications related to health and health care is also moving quickly. This sector deserves more and more attention and efforts from the developers of applications for mobile devices. It is believed that these devices can play an important role in the future people's healthcare, taking advantage of its features and capabilities to make daily tests and, that way, monitor people's medical status with the convenient regularity.

1.1 Context

This dissertation project was developed at the Fraunhofer Portugal Research Center for Assistive Information and Communication Solutions (FhP AICOS), the first research center operated by Associação Fraunhofer Portugal Research.

The research center aims to popularize the access to Information and Communication Technologies (ICT) through the development of market-oriented R&D assistive solutions. Focused on emergent technologies, Fraunhofer Portugal explores mutual interests in science and technology, targeting the enhancement of people's living standards by offering them intuitive and useful technology solutions.

FhP has particular concern with people that have been deprived of using technology due to the frequent non-user-friendly solutions, that somehow exclude groups less able, as are the elderly. Continuous collaboration between Fraunhofer Institutes, R&D institutions in Portugal and customers will contribute to provide people with a good and sustainable quality of life.

1.2 Goals and Motivation

The main target of this work is to connect a low cost stethoscope microphone to the hands-free audio connector of the device and to develop software that will allow specifically elderly people to perform every-day tests of their heart and lung functions.

Very special attention shall be given to the fact that the users will be aging, so not only a user friendly interface is important, but also an interactive guidance that will tell users, step by step, how to use the application and perform the tests. In addition, the mobile phone shall build a history of results and match the results with a "health pattern" that characterizes a "healthy and safe" status. Functions to exchange the test results with the health care professionals shall be implemented as well.

The target device to be used will be an Android based smart phone, well suited to the task due to the set of sensors and the possibility to install additional software to be used as mobile e-Health device. A highly automated, daily monitoring of the heart and lung functions can help to prevent serious illnesses by early diagnosis. No only there will be an increased quality of life, but also a reduction of related health care costs.

1.3 Dissertation Overview

Besides introduction, this dissertation contains 5 more chapters.

Chapter 2 describes the state of the art, regarding the theoretical background, as well as the technologies and related work. Chapter 3 specifies the system's requirements, along with the designed architecture and the functionalities it provides. Then, in Chapter 4 is detailed the system's implementation, explaining how things were done and why, whenever necessary. In Chapter 5 are presented some tests carried out and the respective results obtained. Finally, Chapter 6 presents some conclusions about the research carried out suggests some future work.

Chapter 2

State of the Art

2.1 Mobile Devices and m-Health

The progresses in the technologies provided by mobile devices, its portability and ease of use along with the falling of prices, make these devices an appropriate tool for healthcare purposes. More and more, m-technology is being used in the healthcare field [PL07]. The use of this technology tends to become a cost-effective method of identifying and monitoring health issues.

Rob Chesters, manager of MedilinkWM, said, at a conference focused on existing and emerging technologies for home-based healthcare, that mobile technologies deliver convenient personalized healthcare solutions using mainstream technologies and help individuals to self-manage their conditions and take greater ownership of their health. He added that the future for the health and social care services lies in the collaboration of healthcare providers, users and industry, in order to provide a range of solutions in which vital signs monitoring are included [New10].

Besides offering patients the opportunity to monitor themselves regularly, m-Health also provides healthcare professionals with access to the patient data, extremely valuable and useful in the diagnosis and formulation of the treatment. This technology can be of vital importance for people living in remote areas or for those who are physically impaired [uni07].

2.1.1 Mobile Devices for Healthcare Professionals

> "One physician, while vacationing with his family in Arapahoe, N.M., used a smartphone to access patient updates from the ski slopes some 12,000 feet above sea level. The doctor knew he had left his patients back at the hospital in good hands. He just felt the need, as doctors often do, to check and see how they were doing." [Cro10]

Although this may be one of the most extreme examples of physician commitment, the story indeed illustrates the today's reality. A significant majority of doctors and healthcare professionals rely on the emergent technology to improve the healthcare they deliver and to solidify relationships with their patients.

In contrast to the common perception of healthcare professionals, most doctors willingly adopt technologies that meet a business need or improve the quality and safety of patient care. Since the first handheld devices were launched, the impact and adherence was tremendous among doctors, who found the devices convenient for taking notes, accessing information and managing their schedules while making patient rounds.

Though some people may not think about doctors this way, the truth is that they are information workers and, therefore, they require access to current information about diseases, medications and patient histories to determine the most appropriate treatment. Only thus it is possible to provide patients with the excellent healthcare they deserve. The only one difference between them and most information workers is that they cannot always be at their desks to reference the material they need. Instead, they require information at the patient's bedside, in the exam room or when responding to emergencies out in the field [Cro10].

2.1.2 Emergent Technology in m-Health

Home observation through a network equipped mobile is sometimes the only practical approach when a large number of potential patients exist. Continuous monitoring of the heart (m-ECG) and brain (m-EEG) is possible from the technical point of view, though difficult due to the presence of perturbing signals and noise, frequently called artifacts. Breathing abnormalities associated with bronchial asthma and other chronic respiratory system diseases as well as sleep disorders are also areas being focused in the emergent m-Health techniques [SM].

2.1.3 Classification of m-Health Systems

Based on the definitions given above, m-Health systems can be classified according to the characteristics of the source and destination of the medical information flow [SM]:

1. Patient to medical supervisor;
2. Patient to physician;
3. Physician to physician;
4. Physician to expert system;

5. Patient to medical CRM system (management of patients and medical interventions).

Depending on the target group, m-Health systems can be classified as follows:

- m-Health for hospital patients;
- m-Health for healthy people (preventive m-Health);
- m-Health for the chronically ill or vulnerable individuals.

2.1.4 m-Health Potential

The current changing in healthcare environment is increasing the need to transform the sector in order to meet new challenges and to benefit from new opportunities [RHI06]. It is expected that in the next few years both wireless technologies and m-Health systems with will be developing at great rhythm.

The rising demands from different medical applications and roaming applications, in what concerns to data traffic, might lead to some incompatible issues with 3G's data rates, under specific conditions. Therefore, the implementation of 4G systems will be of great importance in medical care. Along with the proliferation of 4G systems, home medical care and remote diagnosis will become common. From prescription of medicines at home to remote surgery, there is a long way to be built and followed targeting the realization of a virtual hospital, with no resident doctors. Preventive medical care will also be focused and personal accessories that are used every day by the individual will be able to constantly transmit data to the hospital and, after medical diagnosis, receive such results [Ist04].

The dominant trends in healthcare are shifting towards shared and integrated care. The individual's healthcare will no longer be the sole responsibility of a team of professionals across all levels of the healthcare system hierarchy but also of the individual who must cooperate in favor of a benefit that is, primarily, of his interest.

2.2 Cardiopulmonary System

The cardiovascular and respiratory systems work closely together to ensure that enough oxygen is received by organ tissues in order to be able to perform cellular functions. The oxygen contained in the air breathed in and held in the lungs is transferred to the blood, which is responsible for transporting the oxygen to all cells of the body. The oxygenated blood is pumped by the heart, from the lungs to the body. Additionally, the respiratory and circulatory systems work together to remove carbon dioxide, which is a metabolic waste product.

2.2.1 Respiratory System

2.2.1.1 The Lungs

The lungs are spongy organs, approximately with 25cm in length, where carbon dioxide and oxygen are exchanged. The process is called gas exchange. When a person inhales, the alveoli in the lungs fill with oxygen. The oxygen is then sent to blood cells in the capillaries that surround the alveoli. In reverse way, the carbon dioxide in the blood is sent to the alveoli, where it is expelled from the body through the process of expiration.

2.2.1.2 Approach to the Patient with Disease of the Respiratory System

A disease of the respiratory system may be identified based on symptoms, on an abnormality detected on a chest radiograph, or both. There is a set of different diagnostic possibilities, which will be shortened, until there is only one. To do so, additional information is required including patient's medical history, physical examination, testing of the pulmonary functions, additional images, etc.

This Section considers only the approach to the patient based on the major patterns of presentation, clinical history and physical examination, Although its importance on the diagnosis, the radiography to the chest will not be contemplated here, since it is non-relevant for the project, at least for now.

CLINICAL HISTORY

Four main symptoms can be experienced by a patient with respiratory system disease: dyspnea, cough, hemoptysis and chest pain, with higher incidence rate on the first two.

- **Dyspnea**, generally known as shortness of breath, it frequently occurs in situations of high demanding physical effort. However, it may present a pathological condition in case it occurs in unpredicted situations. It is mostly caused by asthma, pneumonia, congestive heart failure, chronic obstructive pulmonary disease or psychogenic reasons;

- **Cough** is not per se a lung disease indicator, since it may be caused by a no number of factors. Nevertheless, if accompanied by sputum it may suggest airway disease (asthma, chronic bronchitis).

- **Hemoptysis** can originate from disease of the airways, the pulmonary parenchyma, or the vasculature. Diseases of the airways can be of two different types: inflammatory or neoplastic [ASF08a].

- **Chest pain** - usually presents diseases of the respiratory system. If the pain originates from some problem in the parietal pleura, it is likely to be accentuated by respiratory motion. [ASF08b].

	Chest Pain (n = 373)			Dyspnea (n = 391)			Cough (n = 337)		
Analysis	Ppos	Pneg	Kappa (95% CI)	Ppos	Pneg	Kappa (95% CI)	Ppos	Pneg	Kappa (95% CI)
Unstratified	74	78	0.52 (0.44, 0.60)	70	76	0.46 (0.37, 0.54)	63	75	0.38 (0.28, 0.48)
Stratified by sex									
Male	76	74	0.51 (0.38, 0.64)	75	76	0.52 (0.40, 0.65)	63	74	0.38 (0.24, 0.53)
Female	72	78	0.48 (0.37, 0.60)	64	75	0.40 (0.28, 0.52)	61	75	0.37 (0.24, 0.51)
Age, y									
18-34	65	79	0.44 (0.17, 0.70)	62	85	0.48 (0.21, 0.75)	74	81	0.56 (0.27, 0.84)
35-49	78	78	0.55 (0.37, 0.73)	49	78	0.27 (0.02, 0.52)	45	77	0.23 (0.04, 0.50)
50-64	76	74	0.51 (0.35, 0.66)	69	75	0.45 (0.28, 0.61)	50	79	0.31 (0.12, 0.49)
≥65	71	76	0.47 (0.32, 0.62)	75	72	0.47 (0.34, 0.60)	69	66	0.37 (0.23, 0.52)

Figure 2.1: Patient's medical history.

ADDITIONAL HISTORIC INFORMATION

There are some risk factors for lung disease about which it is important to collect facts, ensuring complete historic data of the patient. One crucial information concerns current and past smoking, particularly cigarettes for being more harmful. In case the patient presents smoking history, things like the number of years of smoking and the intensity (numbers of packs per day) should be referred. If, on the other hand, the patient has quitted smoking, it is relevant to know the exact interval of time since cessation. The reason for this is that the risk of lung cancer falls progressively in the decade after the interruption of smoking, and loss of lung function above the expected age-related decline ceases as soon as the patient stops smoking.

Although the two most serious respiratory diseases are *neoplasia* and *chronic obstructive lung disease*, there are some other complications associated with smoking which must not be forgotten, for instance, pneumothorax and respiratory bronchiolitis. Also significant passive inhalation of smoke, i.e. by exposure, whether at home or at workplace, shall be reported since it is a factor that can lead to the aggravation of airways disease [ASF08c].

There is yet the chance that the patient has been exposed to other kind of inhalable agents related with lung disease. This agents can act either via direct (immediate toxic injury) or indirect (through immune mechanisms) and include organic antigens (e.g. from animal proteins) and inorganic dusts (e.g. from silica). The exposure to environmental allergens normally exacerbates diseases such as Asthma, which is more commonly found in woman. Sometimes, this pathology may even originate from occupational exposures, for instance people who work daily with cork during some years [ASF08c].

2.2.1.3 Physical Examination

The general principles of inspection, palpation, percussion, and auscultation apply to the examination of the respiratory system. It is important that the physical examination focus not only on detecting abnormalities on thorax and lungs but also on recognizing another findings that give important information towards the lung disease diagnosis.

On **inspection**, it is examined the pattern and rate of breathing, on one hand, and observed the symmetry of lung expansion, on the other hand. Regarding the first point, an uncommon fast breathing normally indicates either a breathing work increase or a boost in respiratory demands. In relation to the expansion of the chest, if it shows up to be asymmetric it is because there is an disproportionate process affecting the lungs (an obstruction on a large airway, pleural disease, etc). Visible alterations on the size of the thorax increase the breathing work, and result in dyspnea [ASF08c].

On **palpation**, it is verified the symmetry of lung expansion, generally confirming the findings observed by inspection. Spoken sounds produce vibration that is transmitted to the chest wall, allowing the healthcare practitioner assess the presence or absence of the tactile fremitus, as well as its symmetry. If there is liquid in the pleural space, i.e. between the lung and the diaphragm, the transmission of the vibration is diminished. Contrarily, an area of pulmonary consolidation will amplify the vibration [ASF08c]. The normal sound coming from an air-containing lung is resonant, in contrast to the consolidated lung that sounds dull. Air between the chest and the lung results in a hyper-resonant percussion note.

On **auscultation** of the lungs, the examiner focus on both the quality and intensity of the breath sounds, while at the same time tries to listen to extra sounds. Placing the stethoscope's diaphragm at the periphery of the lungs, normal breath sounds (described as vesicular breath sounds) can be heard, which are clearly divided in two moments: inspiration and expiration, the first of which is longer and louder than the second one. In some situations, there is an endobronchial obstruction or just air or liquid existing in the pleural space, compromising the sounds transmission. In the opposite case, when a consolidated lung improves the propagation of the sound, the bronchial transmission can be verified by listening to whispered sounds.
The most common pathologic sounds that can be heard include crackles, wheezes and rhonchi.

Crackles are referred to as discontinuous, intermittent and brief sounds. Though they can be heard either on inspiration or expiration, they are typically an inspiratory sound caused by fluid in the small airways called alveoli or by interstitial lung disease. Crackles appear to be caused by both the sudden open of airways and the presence of fluid inside the airways. There are two types of crackles: fine and coarse.

Wheezes, unlike the crackles, are more frequently heard during expiration and are described as continuous sounds. They are caused by air moving through airways narrowed by constriction or swelling of airway or partial airway obstruction.

Rhonchi is the term applied to the sounds created when there is a mucus in the airway; the viscous interaction between the secretion and the moving air creates a low-pitched vibratory sound [ASF08c].

Table 2.1 found below, summarizes the patterns of physical findings in common pathologic conditions regarding the respiratory system.

Table 2.1: Typical Chest Examination Findings in Selected Clinical Conditions

Condition	Percussion	Fremitus	Breath Sounds	Voice Transmission	Adventicious Sounds
Normal	Resonant	Normal	Vesicular	Normal	Absent
Consolidation of atelectasis	Dull	Increased	Bronchial	Bronchophony, whispered	Crackles
Asthma	Resonant	Dull	Decreased	Decreased	Decreased
Intersticial lung disease	Resonant	Normal	Vesicular	Normal	Crackles
Pneumothorax	Hyperresonant	Decreased	Decreased	Decreased	Absent
Pleural effusion	Dull	Decreased	Decreased	Decreased	Absent

A meticulous general physical examination is mandatory in patients with disorders of the respiratory system.

2.2.2 Chronic Obstructive Pulmonary Disease (COPD) and Heart Failure

Several studies that have been undertaken revealed a high rate of coexistence of COPD and heart failure (HF), which is more accentuated in aging people [GOL06] [PM08] [NMJP09]. Between 7,2% and 20,9% of patients with COPD also present heart failure evidences [CS06] [MP03]. On the other hand, studies report that approximately 10,0% to 39,9% of patients with heart failure show up to have COPD [BJ03] [HE02]. All these investigations conclusions were based on clinical-data or on self-reported information.

Tobacco smoke is seen as the main triggering factor for COPD, for both active smokers and passive ones (exposed to it). More than 85% of cases of COPD are caused by tobacco.

COPD is a serious disease, conditioning relevant degrees of disability and high mortality. In fact, it is responsible for 3 million deaths every year, as can be shown in Table 2.2, being the 4^{th} main cause of death worldwide (nearly 5,1% of the the total deaths). This disease is slightly more frequent in men than women, with percentages of 53,5% and 46,5% respectively.

Table 2.2: Major causes of death in the world.

Rank	Disease	Deaths per year (millions)	% of the total deaths
1	Ischaemic heart disease	7,2	12,2
2	Cerebrovascular disease	5,7	9,7
3	Lower airway inflammation	4,2	7,1
4	COPD	3,0	5,1
5	Diarrhoea	2,2	3,7
6	AIDS	2,0	3,5
7	Tuberculosis	1,5	2,5
8	Lung cancer	1,3	2,3

2.2.2.1 COPD prevalence in Portugal

More than 5000 interviews conducted between smokers over 45 years revealed that about 70% had symptoms consistent with being at risk of or already suffering from COPD [dA07] [dA08]. The currently accepted value points to the existence of at least 540,000 cases in Portugal [dA08] but it appears to be underestimated. The prevalence is particularly high over 60 years, reaching 1 in 4 men and 1 in 7 women.

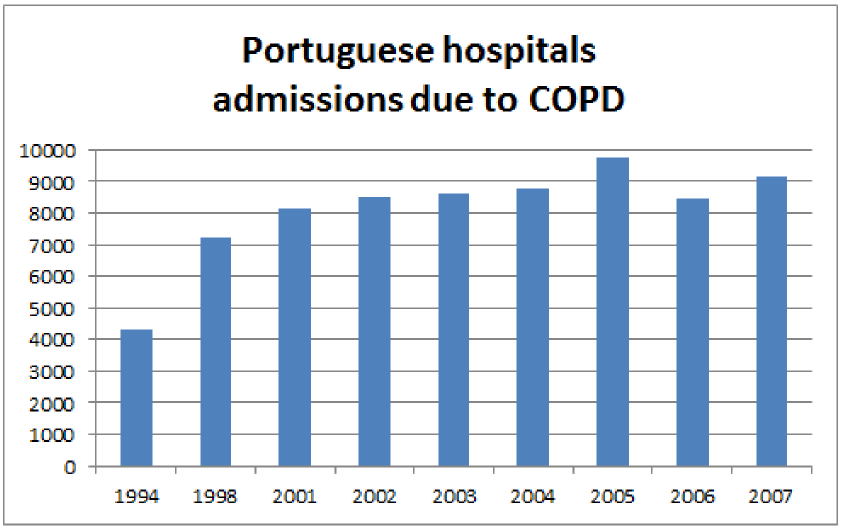

Figure 2.2: Admissions of patients with COPD in portuguese hospitals [dA08].

COPD is one of the most frequent causes of hospitalization for respiratory disease. A chart with the official values of last years is presented in Figure 2.2.

COPD is considered the cause of death of more than 2400 cases every year, which represents between 27% and 30% of deaths from respiratory disease [dA06] [dA07] [dA08]. It is the second leading cause of death in this group (excluding lung cancer).

The estimated direct costs with patients with COPD exceed 240 million euros annually [dA05] [dA07] [dA08].

2.2.3 Cardiovascular System

2.2.3.1 Heart and heart murmurs

The heart is a muscular organ found in every animal with circulatory system, that is responsible for pumping the blood throughout the blood vessels by rhythmic contractions.

The heart anatomy is shown in Figure 2.3.

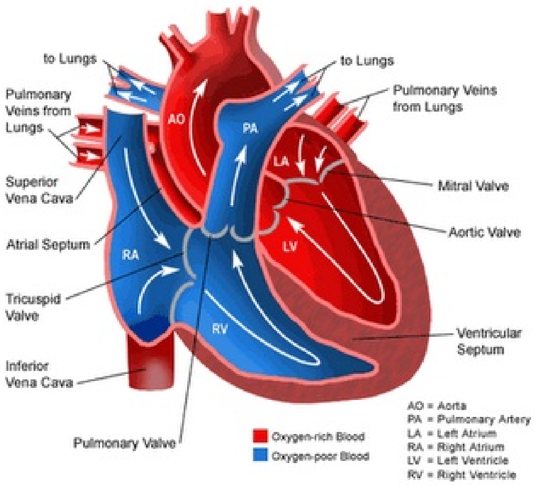

Figure 2.3: The heart anatomy.[1]

The speeding up of blood flow through orifices (normal or abnormal ones) generates vibrations which are reflected in the form of audible turbulence called heart murmurs. These murmurs are typically defined in terms of their timing within the cardiac cycle.

Thus, there are three distinct kinds of murmurs: systolic, diastolic and continuous. The first one starts with or after the first heart sound, S1, and ends at or before the the second heart sound, S2, while the diastolic mutter begins with or after the sound S2, and

[1] http://www.medicalook.com/Heart_diseases/

finishes before or at the following S1 sound. Oppositely, the continuous murmur starts in early systole (S1) and continues until part or the totality of diastole (S2), so it is not restricted to either phase.

It is easy then to understand that the very first step in their identification is to determine with accuracy the timing of the heart mutter. Differentiating the S1 and S2 components is quite simple using the auscultation method but it might be quite more complicated in a tachyarrhythmia situation. In this case, palpation of the carotid arterial pulse (in the neck) shall be done simultaneously. S1 should be closely followed by the upstroke [ASF08d].

2.2.3.2 Approach to the Patient with a Heart Murmur

A differential diagnosis requires a complete evaluation of the patient, although there are some major attributes. The relevance that is given to a heart murmur is based not only on the associated findings but also on additional information such as the patient history and clinical context.

If at the end of these steps, still any ambiguity or doubt prevails, a noninvasive testing may follow, in order to provide some extra information regarding anatomic and physiologic issues.

Accuracy on the heart murmur identification is very important for posterior referral to a cardiovascular specialist. Meanwhile, it is told to the patient some physical activity restrictions that he should follow, along with the medicines prescribed [ASF08d].

There are four distinct points where the stethoscope must be placed in order to auscultate the heart functioning, which are illustrated in Figure 2.4.

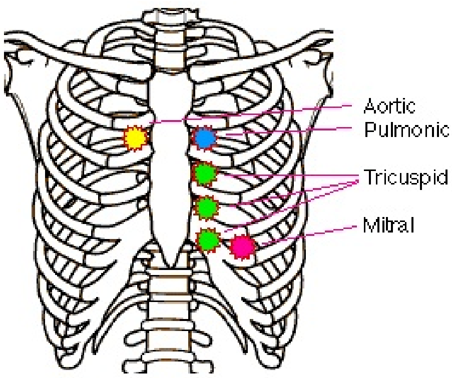

Figure 2.4: Points of auscultation of the heart.[2]

When auscultating, the examinator listens to the internal sounds of the body and characterizes them in terms of:

[2] http://medinfo.ufl.edu/year1/bcs/clist/cardio.html

- **Duration** — The duration of a heart murmur is related with the elapsed time in the cardiac cycle during which a difference of pressure exists between two cardiac chambers: the left ventricle and the aorta, the right ventricle and the pulmonary artery, or the great vessels. The magnitude of this pressure, along with its variation, is responsible for the flow speed, the turbulence level and the intensity and configuration of the mutter.

- **Intensity** — The intensity of a heart murmur is measured using a scale that ranges from 1 to 6. Depending on the grade, it characterizes different kinds of murmurs.

  1. very soft, hardly heard;
  2. easily heard but not particularly loud;
  3. loud, with no palpable thrill;
  4. very loud, accompanied by a thrill;
  5. loud enough to be heard with only the edge of the stethoscope touching the chest;
  6. loud enough to be heard with the stethoscope slightly off the chest.

  From grade 3 on (inclusive), the murmur is likely to be caused by important structural heart disease and it indicates high speed on the blood flow on the area of production. Obesity or an obstruction on the lung may decrease the intensity of the sound. However, this intensity can be misleadingly low when cardiac output is considerably reduced;

- **Location and Radiation** - For an accurate identification of the murmur, it is important to recognize its exact location and radiation. Additional clues can be provided by evidence on abnormalities of the components S1 and S2 or by some adventitious sounds (f.e. a systolic click or a diastolic snap).

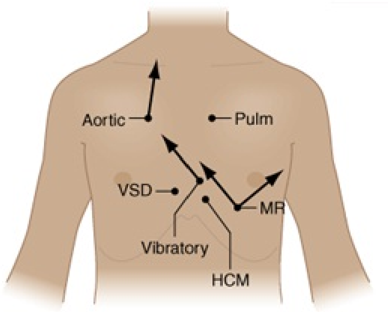

Figure 2.5: Maximal intensity and radiation of isolated systolic murmurs.[3]

[3] http://www.accessmedicine.ca/

Particular attention shall be given to the characteristics of the murmur, in order to successfully complete the physical examination.

2.2.4 Atrial Fibrillation

Atrial fibrillation is the most common heart rhythm abnormality in adults worldwide [SMW+04], affecting over 9 million people in the European Union and the United States alone [MBG+06].

AF occurs when a chaotic electrical activity develops in the upper chambers or atria, which completely inhibits the action of the sinus node. Then, the left atrium starts to contract haphazardly and the blood is pumped in a less efficient way. The atrioventricular (AV) node is overloaded with impulses, trying to move on to the ventricles. Consequently, the ventricles beat faster, but not as fast as the atriums. Fortunately, some of these impulses can be prevented by the AV node due to its limited conduction speed. If not this protection, there could be severe ventricular tachycardia resulting in a serious reduction of cardiac output.

Figure 2.6 shows the behavioral differences on a heart with AF.

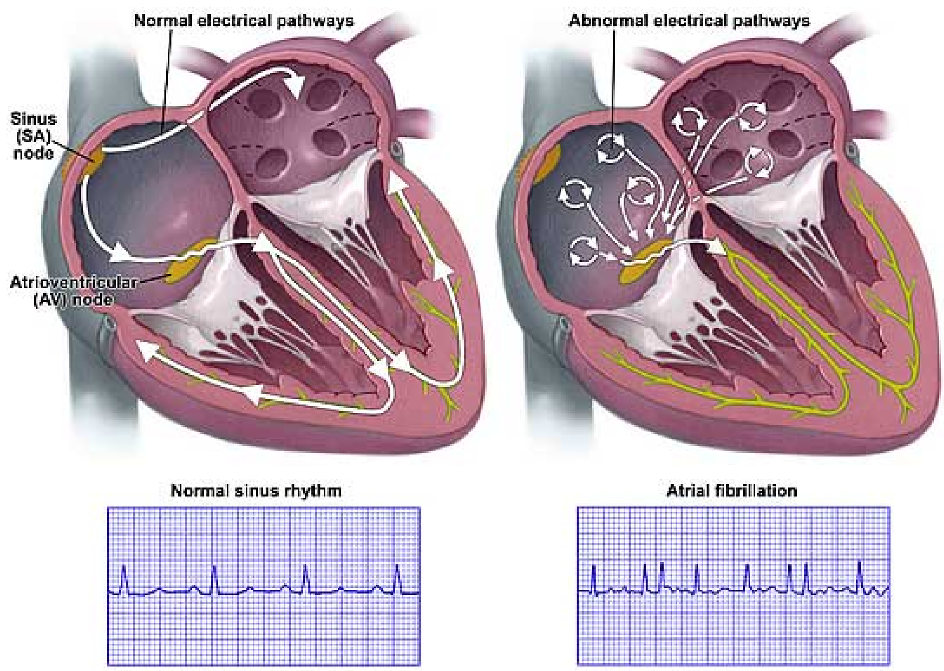

Figure 2.6: Comparison between a normal sinus rhythm and atrial fibrillation.[4]

[4] http://mykentuckyheart.com/information/AtrialFibrillation.htm

The result of this behavior is a fast and irregular heart rhythm [BDI09]. The heart rate in AF may range from 100 to 175 beats per minute, whereas a normal resting pulse ranges from 60 to 100 beats.

AF can be classified into 3 categories, depending on its persistence along the time:

- **Paroxysmal**, if episodes cease within maximum 7 days;
- **Persistent**, if episodes persist for more than 7 days;
- **Permanent**, if episodes last for more than 1 year.

AF is a risk factor for stroke, reason why it is essential to promote prevention strategies.

Causes and Symptoms

The reasons behind AF are not yet fully understood, though its frequent development in patients with abnormalities or damage in the heart's structure. Some causes for AF are:

- High blood pressure
- Congenital heart defects
- Heart attacks
- Abnormal heart valves
- Sleep apnea

Atrial Fibrillation is also sometimes associated with changes in the functioning of thyroid gland, high alcohol intake and lung infections.

The most common symptoms include, among others, palpitations, tiredness, shortness of breath, dizziness and chest paint.

2.2.4.1 AF in Portugal

According to a recent study released by Instituto Português do Ritmo Cardíaco (IPRC) 2,5% of the Portuguese over 40 years suffer from atrial fibrillation, which reveals that 121,825 Portuguese live in this condition, some of them without knowing [BMA+10]. Comparing to studies carried out in other countries, this is a relatively high incidence rate.

The research included 10,477 individuals from all regions of the country, 55% of which were women. The data collected allowed not only to access the prevalence of FA in +40 years old people but also to characterize the population in demographic, socioeconomic, clinical and therapeutic terms [BMA+10].

> "The probability that a person with atrial fibrillation has a stroke is five times higher than the general population in that age group." [BMA+10].
>
> Daniel Bonhorst, President of IPRC

João Primo, from Associação Portuguesa de Arritmologia, Pacing e Electrofisiologia (APAPE), stated that after 50 years old, the incidence rate doubles for each decade of life, and it can exceed the 10% among people over 80 years.

Among the 261 individuals that were identified with AF, 38% were not diagnosed, which reveals that a great percentage of the people living in this condition is not aware of the situation. Moreover, only 74% from the 72% of people aware of their FA condition were taking medication. In every 12 episodes only 1 is symptomatic, i.e, the symptoms may go unnoticed, specially if people reduce their physical activity [BMA+10].

The study found that about one quarter of strokes happen, directly or indirectly due to this arrhythmia. Control of atrial fibrillation can reduce the number of strokes - a leading cause of death in Portugal.

2.3 Sound Signals Capture and Processing

2.3.1 Signals

Signal processing deals with signals' analysis, in either discrete or continuous time. A signal is any variable that contains some kind of information that can be conveyed, modified, manipulated or displayed. They are analogue or digital representations of time-varying or spatial-varying physical quantities.

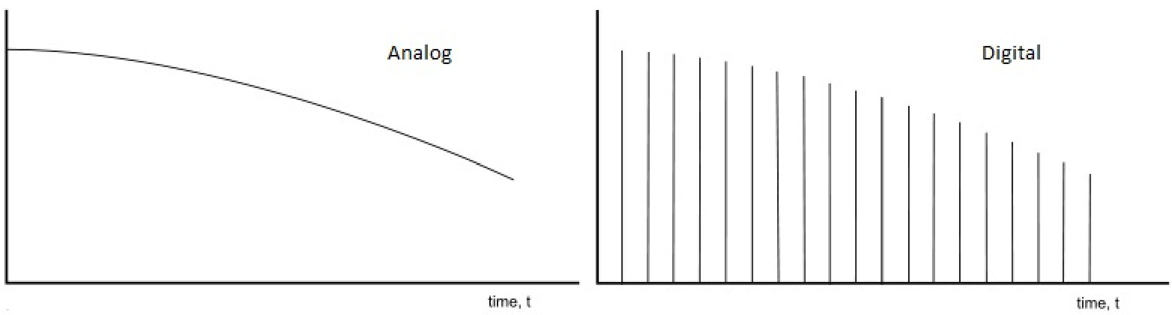

Figure 2.7: Analogy and Digital Signals.

The following types of signals are of particular interest:

- speech;
- biomedical signals;
- sound and music;
- video and image;
- radar signals.

The majority of signals in nature are analogue, which means that vary continuously with time. In digital signal processing (DSP), these signals are sampled at regular intervals and converted into digital form [PSRD10].

2.3.2 Digital Signal Processing

Digital signal processing is concerned with the representation of the signals in the digital form, i.e. a sequence of numbers, and to process, modify or extract information from signals. DSP can be used for example to remove noise or interference from a signal or to get its data spectrum. DSP is becoming more and more popular among a great diversity of areas, some of them in which analogue methods are difficult or even impossible [PSRD10].

Some advantages of this method are guaranteed accuracy, greater flexibility, superior performance and, in some cases, it presents the only viable option (when the information is already in digital form).

2.3.3 Application Areas

DSP is continuously having a major and increasing impact in many areas of technology, where information is handled in a digital form or controlled by a digital processor. Application areas include [Kes03]:

- **Image Processing**
  - Pattern recognition
  - Robotic vision
  - Image enhancement
  - Satellite weather map
  - Animation
- **Audio/Speech**
  - Digital audio
  - Speech recognition
  - Speech synthesis
  - Text to speech
  - Equalization

- **Telecommunications**
  - Echo cancellation
  - Data communication
  - Video conference
  - Adaptive equalisation
  - Satellite weather map
- **Military**
  - Radar processing
  - Sonar processing
  - Missile guidance
  - Secure communication
  - Navigation
- **Biomedical**
  - Patient monitoring
  - Scanners
  - EEG brain mappers
  - ECG analysis
  - X-ray storage/enhancement
- **Consumer applications**
  - Digital, cellular mobile phones
  - Digital televisions
  - Digital cameras
  - Voice mail system
  - Internet phones, music and video

The list, though not complete, reveals the importance of DSP.

2.3.4 Sampling

Signals that appear in nature are continuous in time, and not few times, it is of our interest to study them using powerful algorithms. But to be possible for a computer to process it, we first need to convert it into a discrete time digital signal which is accomplished with the sampling method [vdL95]. This operation consists in picking up discrete values from the original signal at certain points of time, which must be equidistant. The sample rate defines the number of samples to be taken per unit of time (often second) [Smi97a]. This method is demonstrated in Figure 2.8.

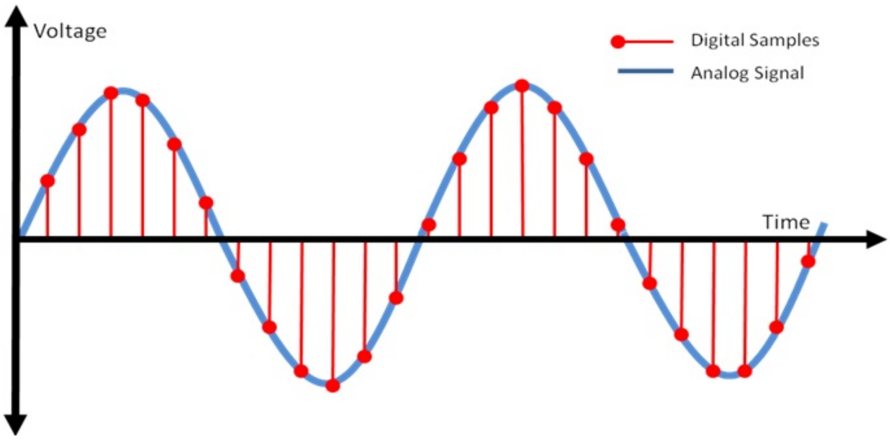

Figure 2.8: Signal sampling.

From the built discrete signal, it is always possible to go back and reconstruct the analogy signal, as long as the sampling was done properly. Integrally or not, if the critical data has been captured, the reversing process is possible. [Smi97a].

2.3.5 The Fourier Transform

The Fourier transform, given the time-domain representation of the signal, obtains the corresponding spectrum, decomposing the signal into its frequencies. For a single-variable continuous function, f(t), the Fourier transform, F(f), is defined by:

$$F(t) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi ft}\, \mathrm{d}t$$

and the its inverse by:

$$f(t) = \int_{-\infty}^{\infty} F(f) e^{-j2\pi ft}\, \mathrm{d}f$$

with j being the imaginary number

$$j = \sqrt{-1}$$

which denotes the exponent

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$

Normally, the signal's decomposition takes sine and cosine waves, although it can be done in an infinite number of ways. This operation is intended to get something easier to deal with, and the sine and cosine waves offer something that the others do not. Giving a sinusoid input to a system, produces a sinusoidal wave too, perhaps with different phase and amplitude, but its shape remains exactly the same, as well as frequency. It is frequently know as the sinusoidal fidelity. [Smi97a].

2.3.6 Discrete Fourier Transform

The discrete Fourier transform (DFT) is a particular kind of discrete transform, used in Fourier analysis. Given a signal in the time-domain representation, the sampling method described in 2.3.4 is used in order to obtain the input function [Bri88]. This function must to be discrete and to have a finite duration for non-zero values.

For complex series of discrete values, it is used the DFT. Say we have a complex series s(k), with N number of samples in the form

$$s_0, s_1, s_2, \ldots, s_k, \ldots, s_{N-1}$$

with s representing the complex number

$$s_i = s_{real} + js_{imaginario}$$

then the DFT is given by:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) e^{-jk2\pi^{n/N}} \qquad \text{with n=0, ..., N-1.}$$

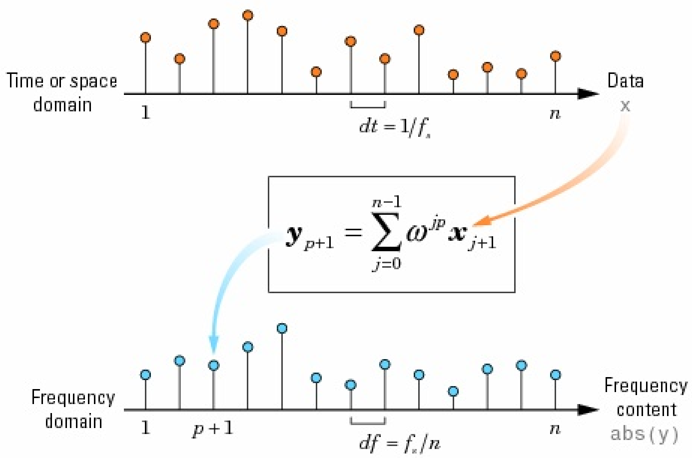

Figure 2.9: The Discrete Fourier Transform.

To calculate the DFT, three different approaches can be taken [Smi97b]:

- view the problem as **a set of simultaneous equations**, which is quite good method to understand the DFT, but too inefficient for practical use;
- use **correlation**, focused on detecting a known waveform in another signal;
- use the **Fast Fourier Transform** (FFT), an algorithm that is typically hundreds of times faster than the other methods.

2.3.7 Fast Fourier Transform

The Fast Fourier Transform computes the same as DFT, but performs it incredibly faster. The results are exactly the same on both methods, just the time they take differs.
Using the definition, it takes O(N^2) arithmetical operations to compute a DFT of N points, while an FFT does it in only O(N log N) operations (producing the same result). Particularly for long data sets, say N being in the thousands or millions, the difference in performance can be incredibly disparate [Smi97c].

This set of mathematical operations decomposes a signal into its constituent frequencies. On the right, it is shown a random signal, firstly, in the time domain and then in its frequency representation, computed by the FFT.

Firstly, the FFT decomposes an N point time domain signal into N time domain signals, each of them having a single point. This decomposition method is illustrated in Figure 2.11.
Then, it calculates the N frequency spectra corresponding to these N time domain signals.

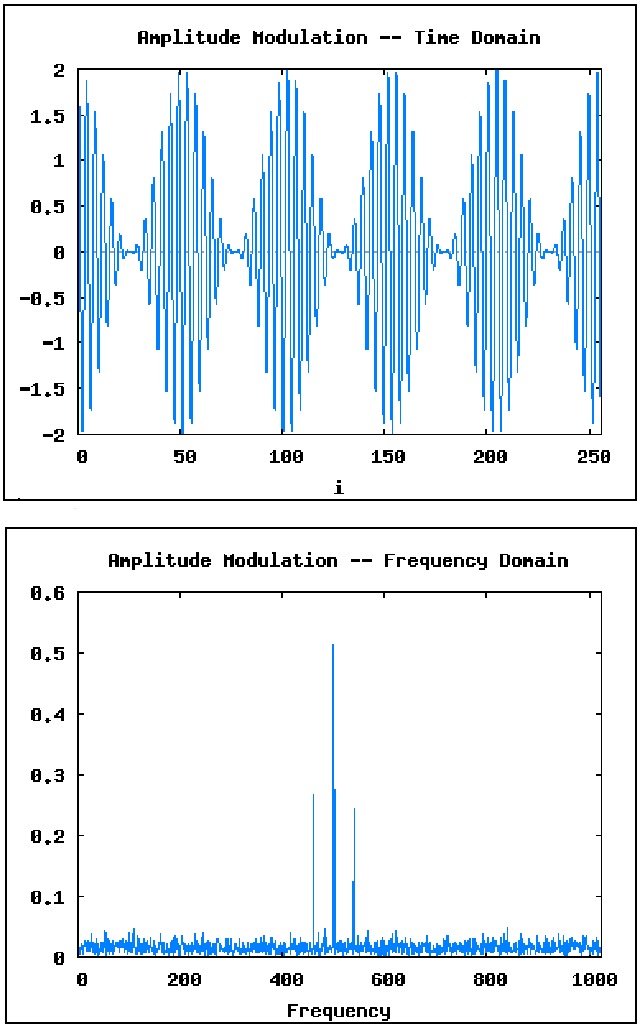

Figure 2.10: From time-domain to frequency representation with FFT.

Finally, the N spectra are combined and originate a single frequency spectrum.

Calculated the FFT, we can from this know the value of the frequency of a signal by calculating the maximum value of the spectrum, which corresponds to the signal frequency. In Figures 2.11 and 2.12 respectively, are shown the decomposition method and an overall view of the FFT algorithm.

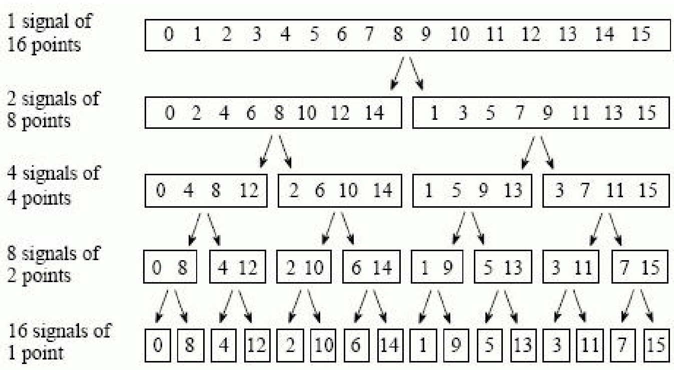

Figure 2.11: The FFT decomposition.[5]

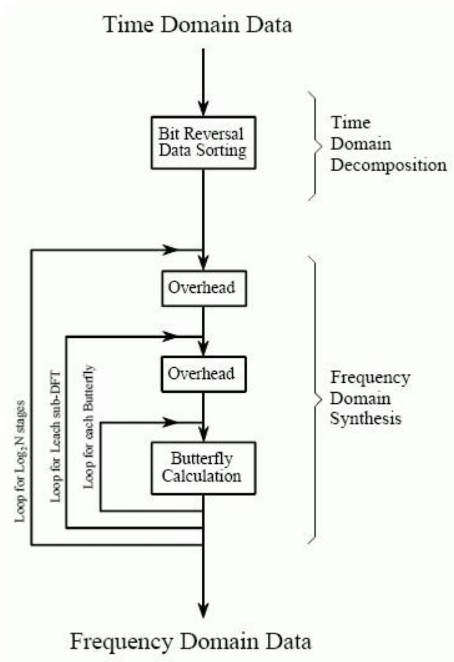

Figure 2.12: Flow diagram of the FFT.[5]

2.4 Technologies

This Section presents some of the technologies that were considered and used for the development of the project.

2.4.1 Android

Android is a linux-based mobile phone operating system, initially developed by Android Inc., a company later purchased by Google (2005). Nowadays, it is supported by Open Handset Alliance which has the collaboration of over 70 major hardware, software and telecoms companies including Google, HTC, Motorola and T-Mobile.

2.4.1.1 Android Architecture

The Android operating system's architecture is structured in layers. On the bottom, there is the (optimized) virtual machine, Dalvik VM, which runs a set of Java libraries. On the top of these libraries, a Java-based application framework is responsible for running the Java applications. The Android layered-architecture is shown in Figure 2.13 and main components are described in further detail below [Bur09].

[5] http://dspguide.com/ch12/2.htm

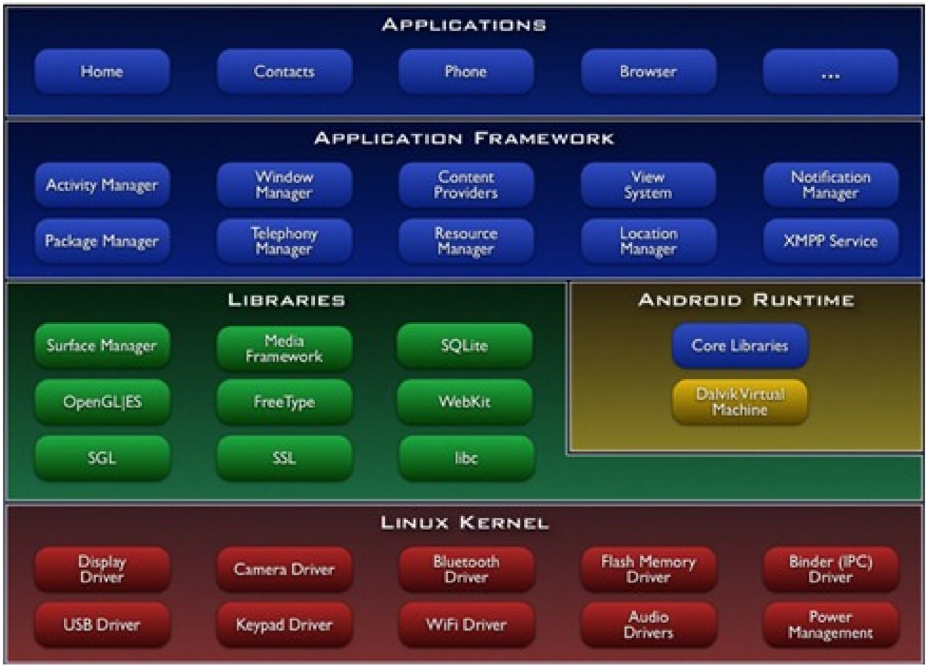

Figure 2.13: Android OS architecture.[6]

APPLICATIONS LAYER

A core set of applications that include an e-mail client, calendar, maps, browser and contact manager. All done in Java.

APPLICATION FRAMEWORK

The way the framework was designed and developed simplifies the concept of component reuse. The developers are encouraged to publish and share their contents, their applications functionalities so that they can be reused in other systems. The developer is given complete access to the framework APIs that are used in the central application, allowing to take advantage according to the interest [Bur09].

LIBRARIES

A set of C/C++ libraries that allow working with common media files such as MP3, AAC, AMR, MPEG4, JPG and PNG run on the top of kernel and different components use them. The Surface Manager is responsible for the 2D and 3D content display. An implementation of 3D graphics based on OpenGL is also included.

ANDROID RUNTIME

Every application runs in its own process, which in turn have their own instance of the Dalvik virtual machine (several VMs can be run efficiently). The executable files take

[6] http://developer.android.com/guide/basics/what-is-android.html

the format Dalvik Executable (.dex) and are optimized in order to occupy a small slice of memory. The files are created by a Java compiler, which converts the result into .dex format [Bur09].

LINUX KERNEL

The Linux kernel provides a hardware abstraction layer (HAL) between hardware devices and the remainder of the stack. A Linux 2.6 version is used to handle core services including memory management, process management, security, network and power management

2.4.1.2 Android SDK

Android SDK is a software development kit that enables developers to create applications for Android platform. It includes required libraries to build Android programs, development tools, an emulator and some sample projects with source code [Mei08].

The following list highlights some of the most important features that the APIs, in which Android development environment lies, provide.

- No licensing, distribution, or development fees or release approval processes;
- Wi-Fi hardware access;
- GSM, EDGE and 3G networks telephony or data transfer;
- Comprehensive APIs for location-based services such as GPS;
- APIs for using sensor hardware (accelerometer, compass);
- Libraries for using Bluetooth ;
- Shared data stores;
- IPC message passing;
- Multimedia hardware control, including playback and recording (both with the microphone and camera);
- Background applications and processes;
- Home-screen Widgets, Live Folders, and Live Wallpaper;
- An integrated open-source HTML5 WebKit-based browser;
- Full support for applications that integrate map controls as part of their user interface;

- Media libraries for playing and recording a variety of audio/video formats;
- Mobile-optimized hardware-accelerated graphics, including a 2D graphics library and support for 3D graphics using OpenGL ES 2.0;
- An application framework that encourages application components reuse.

2.4.2 Communicating with Web Services

Web service is a solution used in systems integration and communication between different applications. This technology makes possible the interaction between new applications and existing ones, regardless the platform in which they were developed or the architecture on which they are deployed, the compatibility is ensured.

Different programs communicate through web services, sending and receiving data in a standard format, frequently using XML. Therefore, it is not important what language they "speak" (they may have their own language) since to communicate the data is "translated" into a standard format [Erl04]. Web services allow applications to dispose their contents and features over the network, in a recognized way.

Some other technologies do pretty much the same. The browsers, for instance, use Internet standards such as HTTP and HTML to access the content of a specific web page. Unfortunately, they are not successful in applications communication [W3S].

Web services aim to create an environment where any networked client, running on any type of device, is able to find a service on the network and use it as it was a local service.

2.4.2.1 SOAP Solutions

SOAP is a communication protocol for exchanging information in a distributed environment. In other words, SOAP allows two processes (i.e. two applications), running in separated machines, to communicate and share data and functionalities, independently of the hardware and platform they run on. [FCM03].

A SOAP web service consists of 3 main components: a service registry, a service provider, a service consumer, as shown in Figure 2.14.

This protocol uses XML to format its messages, applying conventions and encoding rules. The trading and messaging process is done after connecting to other application layer protocol, specifically the Hypertext Transfer Protocol (HTTP).

Figure 2.15 illustrates the SOAP message model. It includes a SOAP envelope, a header (optional) and a body.

Summing up, this communication protocol consists of 3 basic things:

- an envelope that defines the message and the way to process it;
- a set of rules specifying data types instances defined in the application;
- a convention for representing calls and responses.

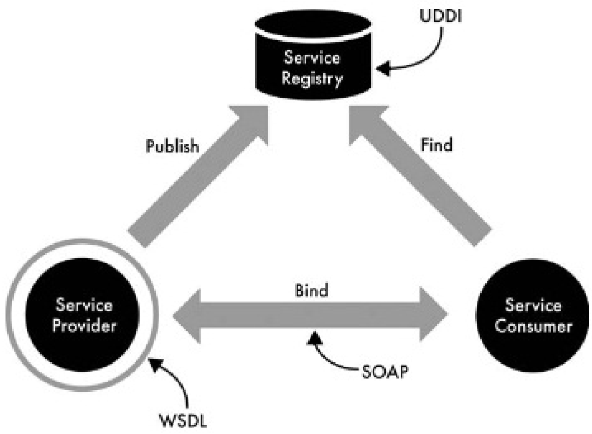

Figure 2.14: SOAP model

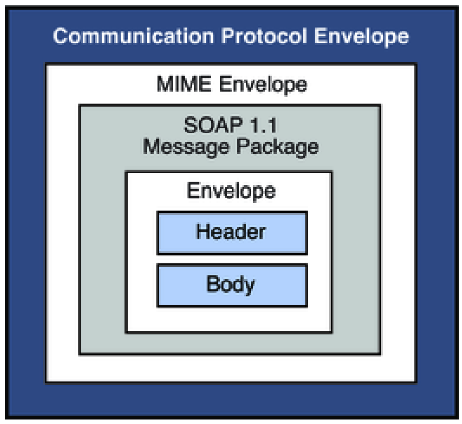

Figure 2.15: SOAP message.[7]

EXTENSIBLE MARKUP LANGUAGE (XML)

eXtensible Markup Language defines a set of rules for encoding data in machine-readable form, and it is a W3C recommendation for generating markup languages for special needs, as tags are not predefined and the programmer must define his own tags. XML technology was designed to carry data, not to display it and it works along with some other technologies that complement it. It actually represents a distinct way of doing things and the main novelty is that it enables to share data at all levels, across all platforms and applications. Thus, XML has an important role in globalization and compatibility between systems, since information sharing in a easy, safe and reliable way lies on it [FCM03].

Moreover, XML allows the developer to focus and devote their efforts to the important tasks rather than to time-consuming tasks such as structural validation which is standardized, and so the programmer does not have to worry about it. It is extensively used in Web services paradigm as it provides a transportable means of describing system and configuration information that can be shared between applications. Both SOAP and WSDL files are written using XML [Mah05].

[7] http://download.oracle.com/docs/cd/E19798-01/821-1796/aeqex/index.html

Web Service Description Language (WSDL)

WSDL is a document written in XML that describes a Web service. It specifies the operations the service provides, the way to invoke them, where it can be found on the network as well as its interface.

It simplifies the communication between applications, since the client can know which operations are available on the web service by accessing this WSDL file, when connecting. The client can then use SOAP to actually call one of the operations listed in the WSDL file [RC07].

The main conceptual objects of WSDL 2.0 are listed below:

- **Service:** container for the set of functions exposed to web-based protocols;
- **Endpoint:** defines the address to connect to the Web service, typically a HTTP URL string;
- **Binding**: specifies the interface and defines the binding style as well as the transport (SOAP);
- **Interface:** defines the web service, its operations and the messages required to perform those operations;
- **Operation:** defines a function or method;
- **Types:** describes the data.

2.4.2.2 REST

REST stands for Representational State Transfer and it is a software architectural style. It was developed in parallel with HTTP/1.1, which conforms with REST style (in fact it represents the largest implementation of a REST architecture).

The concept behind defends an environment where clients and servers co-exist and communicate. Clients send requests to servers which in turn process the requests and return appropriate responses. All the messages are built around the transfer of representations of resources, which are identified by uniform resource identifiers (URIs). Resources are manipulated through their representation and messages are self-descriptive.
Figure 2.16 illustrates the REST concept above explained.

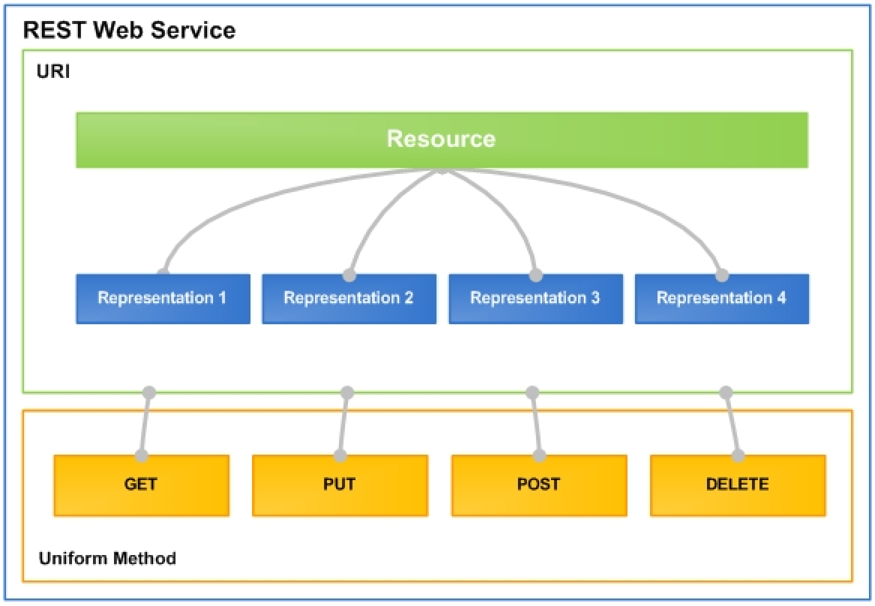

Figure 2.16: Rest web services concept.[8]

RESTful systems follow some architectural constraints:

- **Client-server:** clients and servers are separated;
- **Stateless:** there is no client session stored on the server, all the necessary information is held in the request;
- **Cacheable:** clients are able to cache responses;
- **Layered System:** the client does not know if it is connected to the end server, an intermediary may be the connector;
- **Code on demand (optional):** servers can transfer executable programs to the clients, upon request;
- **Uniform interface:** it constitutes the bridge between clients and servers, allowing both to be changed or developed independently.

This trendy way of creating web services presents some important advantages over SOAP services, specifically the fact they are lightweight, easy to build and present human readable results.

[8] http://incubator.apache.org/wink/1.0/html/1%20Introduction%20to%20Apache%20Wink.html

2.4.3 Google Health

Google Health is a personal health record (PHR) service launched in 2008 that allows users to store and manage their health information online. The use of the service is completely free, only a Google account is required.

Users are allowed to enter basic personal information, along with details about any health conditions, medications, allergies, procedures, etc. GH follows the "anywhere, anytime" systems concept, by having all the data stored in the cloud. Users do not have to worry with issues that few time ago were a reality such as the free space in disk, how to access information stored in a different device, etc. Users just concede GH the right to organize and manage all their health and wellness information. Things like monitor the progress with weight loss goals or create custom trackers for wellness issues (e.g. daily sleep) are now straight forward using this platform.

2.5 Related Work

In this Section the related work on the field is presented, specifically two different-purpose applications, one for iPhone and the other one for Android. Both case studies are fairly brief and do not go into much detail mainly because there is a lack of information to support them.

2.5.1 iStethoscope

iStethoscope [Ben] is an iPhone application that turns the device into a stethoscope, allowing listening to the heartbeat and seeing the respective heart waveform.

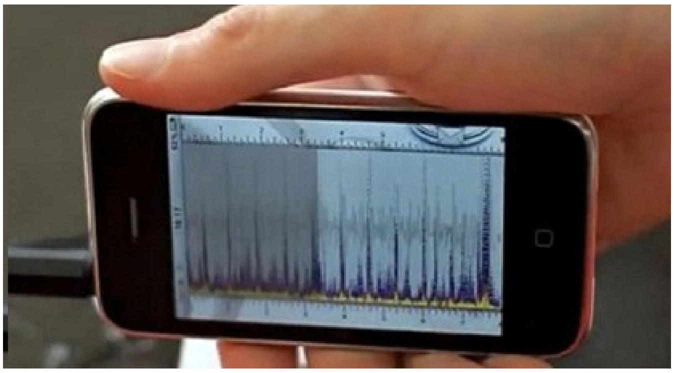

Figure 2.17: Heart wave form on iStethoscope.[9]

To pick up on the heart beat, the surface of the iPhone where the microphone is built-in in, must be pressed against the chest. Shaking the iPhone after some seconds, the

[9] http://www.dailymail.co.uk/sciencetech/article-1307646/iStethoscope-The-iPhone-app-replacing-real-thing-hospitals.html

last eight seconds of recording are played back at the same time that a spectrogram is displayed on the screen. Both the recorded sound and the diagram may be emailed to a specialist for further information.

2.5.2 Instant Heart Rate

Designed for the Android platform, Instant Heart Rate [ins] is an accurate heart rate monitor application that does not need any external hardware. Below, in Figure 2.18, is presented a screenshot of this application working.

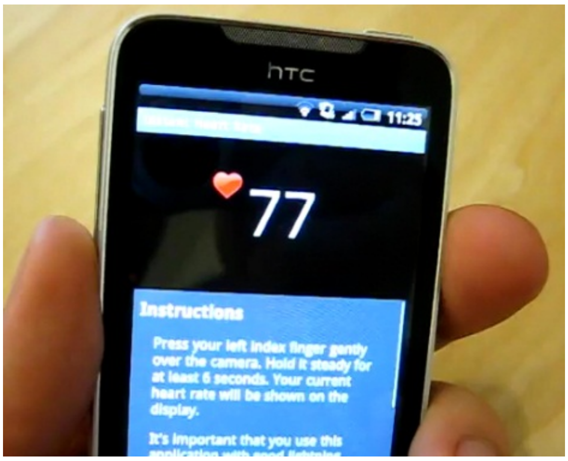

Figure 2.18: Heartbeat monitoring with Instant Heart Rate.[10]

It works by placing and holding the finger over the built-in camera while waiting for a reading, which should not take more than 10 seconds. It measures the color of the skin on the fingertip, detecting tiny shifts in color each time the heart beats.

[10] http://medgadget.com/2010/09/instant_heart_rate_turns_your_android_phone_into_a_heart_rate_monitor_1.htm

Chapter 3

Requirements and Architecture

3.1 Architecture

The system's architecture is based on the client–server model. In this distributed application structure, tasks and workloads are divided between the providers of a resource or service, called servers, and service requesters, called clients. An high-level representation of the arquitechture of this project is presented in Figure 3.1.

Figure 3.1: High-level architecture of the solution.

In this model, there is a network environment in which information exchanges are performed between a server machine and a client machine, and where server has resources that can be shared by different clients. Clients and server communicate over a computer network generally on separate hardware, although both may reside in the same system. A client, an Android smartphone, requests a server's content or service function and therefore initiates communication with the server which is waiting for incoming requests. The client then just waits for the answer from the server which, on the other hand, does whatever the client asked it to do and then returns the expected result. In this project, the server is responsible for the digital processing of the sound recordings captured using the smartphone.

3.2 Requirements

3.2.1 Functional Requirements

The system shall:

- Allow people to perform every-day tests to their heart and lung functions;
- Process the results of the exam, by cloud computing;
- Match the results with a "heath pattern" that characterizes the patient's status;
- Build a history of results;
- Exchange the test results with health care professionals;
- Advice patient to go to the doctor if some pathology is found;

The target device to be used is an Android based smartphone. The stethoscope microphone must be low-cost.

Graphical User Interface

Very special attention shall be given to the fact that the users will be aging and elderly people, so not only a user friendly interface, but also an interactive user manual that will guide the users through the tests will be needed.

Information sharing

To be effective, an application requires good data visibility and ease of integration with other applications so that the right information is always at our fingertips. The application is intended to offer information of great value and that might be reused along with other m-health systems. Functions to exchange the test results with health care professionals shall be implemented as well.

3.2.2 Non-Functional Requirements

Usability

The system should follow the usability standards so that the users can use it easily. The design of the mobile application should take in consideration the possible different aging groups of users, specially the elderly which capabilities are naturally decreased. Simplicity and intuition must be present.

Reliability

Although the system's purpose is not to substitute the Healthcare professionals (and it should not be taken as such), it is still extremely important to be careful with the messages that are displayed to the user, avoiding misinformation about their health status. The results shall then be accurate and reliable. Private user data must be confidential in order to establish a high degree of trust.

Performance

In mobile applications, the device's characteristics, features and limited resources dictate tough requirements to satisfy user's expectations. Therefore, the mobile application should be designed carefully and employ every possibility to improve its performance. It is important to be reasonably fast in every single task.

3.2.3 Why Android

One of the reasons that supported the decision to develop for Android was the fact that it is open source and so we do have access to the code as anybody else has. On the other hand, with Android we are not limited to a particularly device manufacturer or service provider, as it happens with iOS.

Also the programming language influenced the option, since Android is based on the powerful Java language, while in iOS we would have to program in Objective-C which is not widely used.

One last reason is related with the market. It is far easier and faster to get your application accepted and published in the Android Market than in the App Store. While in Android it just needs to conform with the Market rules and it is quickly checked, in App Store, apart from the time it takes to be reviewed, it might be rejected at the end just because [app].

3.2.4 The stethoscope

To collect the heart and lung sounds, it was used a very simple low-cost electronic stethoscope which is shown below in Figure 3.2.

Figure 3.2: The electronic low-cost stethoscope used.

The official description is presented below:

- Ambient Noise Reduction technology cancels out room noise;
- Sprague diaphragm - single head;
- Sensitivity: -44 dB;
- No battery required.

SPECIFICATIONS:

- Chestpiece Design and Technology: Finished Plated Alloy;
- Standard Length: 20" or 62.6 cm;
- Total Weight: 150 grams;
- Diaphragm Diameter: 1.31" or 2.9 cm;
- Bell (20-200Hz);
- Diaphragm (100-500 Hz);
- Extended Range (20-1000 Hz);

3.3 Functionalities

All the requirements specified in Section 3.2 were accomplished. The mobile application allows users to perform daily-tests to their heart and lung functions and keep a history of results somewhere on the cloud (GoogleHealth) and also locally on a small database. The users can at anytime and anyplace access their information, either from the mobile or on a browser through Google Health.

A diagram with the actors that interact with the system and the correspondent use cases are shown below in Figure 3.3.

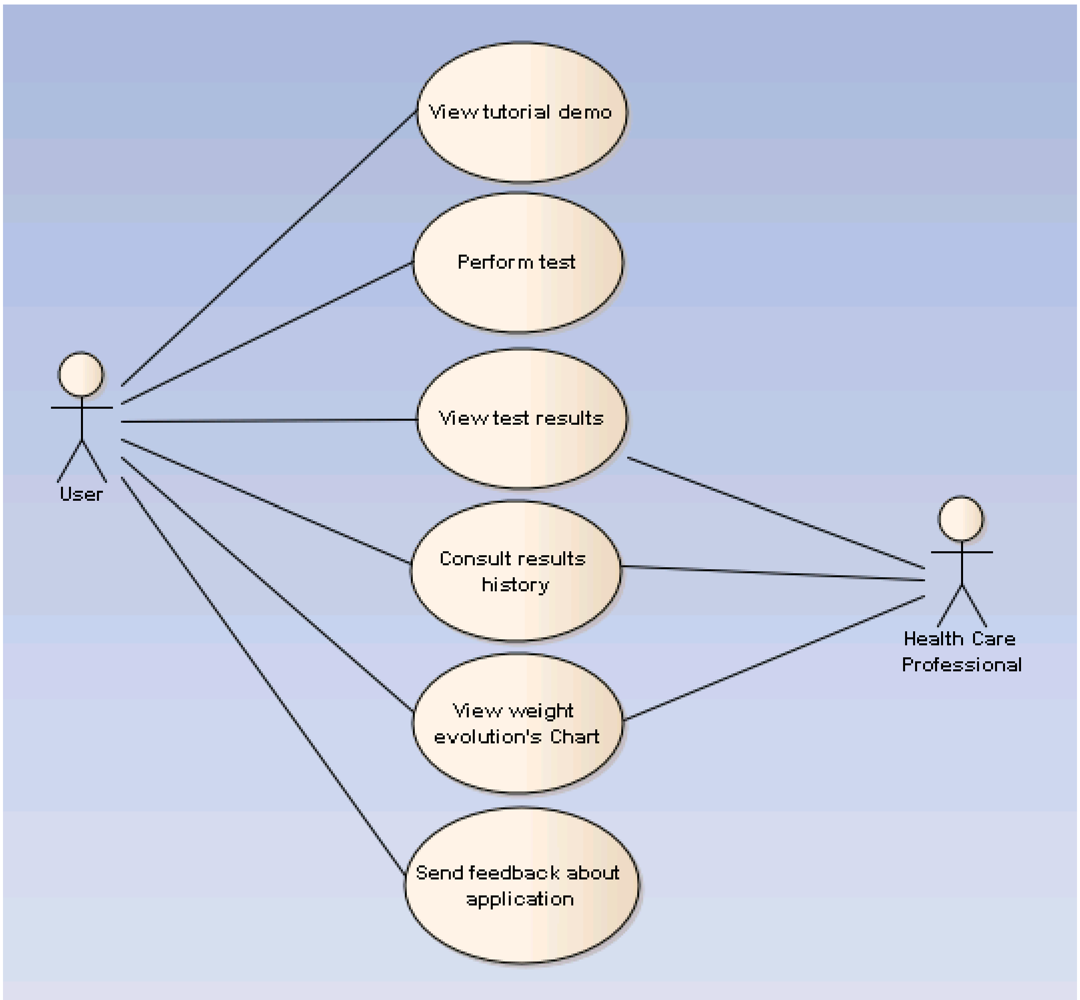

Figure 3.3: System's Use Cases Diagram.

The term "heart and lung functions" is too general and there is a wide range of pathology medical conditions that can be diagnosed through the physical examination using a stethoscope, reason why we had to choose a more specific direction to focus on. It it would be unwise to focus on all of them on this project, bearing in mind the short time allocated for it. Therefore, it was decide to concentrate on the prevention of strokes and heart failure, through the detection of:

- **Atrial fibrilitation** - disfunctioning of the heart (chaotic);
- **Crackles** in the lungs - frequently audible during inspiration;

Additionally, the system calculates the heartbeat rates of the patient during the examination, and its variation. Along with some more information, the user can visualize on the screen the maximum, minimum and average values of heartrate.
The application also matches the patient status with a "health pattern", and in case any abnormality is detected, the patient is advised to go to the doctor.

Each exam consists of two different steps:

- Auscultation of the heart;
- Auscultation of the lungs.

Chapter 4

System Implementation

In this chapter, it is detailed the system implementation. Is is described what was done, how it was done and what were the alternatives in some particular situationsl. The necessary justifications are added whenever adequate.

4.1 Audio Recording

Android offers built-in encoding and decoding for a variety of common audio and video types, allowing easy integration of media into the application.

There are three distinct ways to capture and record audio from the built-in microphone of the Android smartphone, using three different classes: *MediaRecorder*, *AudioRecord* and *AudioRecord - Native Interface*. These media capabilities, which can be accessed fairly straightforwardly, are useful in different situations, depending on the purpose and needs. Thus, the various alternatives were compared so that the most appropriate one was chosen. Below are enumerated some advantages and disadvantages of each solution.

1- MEDIARECORDER

MediaRecorder, as the name suggests, records media. Thus, it can be used to record audio from the built-in microphone of the device and write it into a file on the SD card. However, the recorded audio is encoded in one of the following formats: AMR, MPEG4, WAR or 3GP.

+ Pros +

- Android standard class to record audio and video;
- Easy to use;

- Cons -

- Records audio in compressed format;
- No access to the audio buffers;
- Difficult to process the recorded audio due to its compressed format;
- Sampling rate cannot be changed;
- No control on the recording.

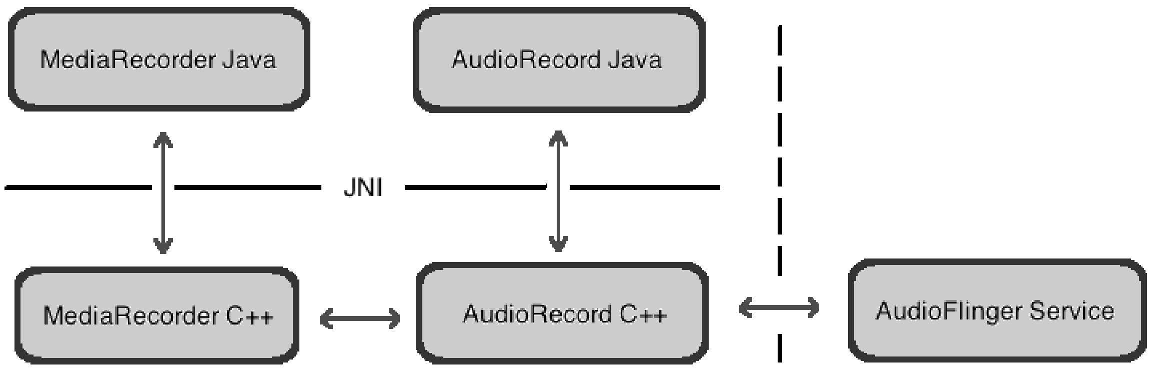

Figure 4.1: AudioRecord and MediaRecord communication.

The picture 4.1 shows how this class works. The MediaRecorder class works like a wrapper API, invoking the C++ MediaRecorder class via JNI. Most of the android APIs have been implemented as such. For efficiency reasons, the core functionalities are implemented in the C++ and then these are called from the Java API. The service that actually records the audio is called AudioFlinger.

2 - AUDIORECORD

AudioRecord solution overcomes the limitations of MediaRecorder, providing greater flexibility. Audio can be recorded into buffers for post processing. The Java AudioRecord API invokes the C++ AudioRecord class via JNI.

+ Pros +

- Records audio in uncompressed and lossless format: Pulse-Code Modulation (PCM) 8 or 16 bits;
- Control on how the recording is performed;
- Different properties can be set: sample rate, sample size, buffer size, etc.;
- Access to the audio buffers.

<u>- Cons -</u>

- Data is provided in buffers: more difficult to handle it;
- Data can be lost if there is no certain on how to proceed.

3 - NATIVE INTERFACE

There is also a native interface AudioRecord that can be used for recording audio. The native interface provides API that can be invoked from C/C++ libraries. These libraries can then be called from Java activity via JNI. Programs using the interface can be compiled with the NDK and used in android applications via JNI.

<u>+ Pros +</u>

- Provides a C/C++ interface for recording;
- Processing is more efficient when done in C/C++ using this API.

<u>- Cons -</u>

- No documentation available;
- Google does not officially support it.

OPTION TAKEN

Having considered and compared the different alternatives available, it was decided to use the AudioRecord (Java) option, since it allowed, on one hand, to record data in raw format and, on the other hand, to set some important parameters as the sample rate, the size of the data, etc. The API also allowed getting the data into a buffer rather than a file, which was exactly what was needed to perform some lightweight pre-processing.

4.1.1 Audio Properties

The audio was recorded using the uncompressed and lossless format PCM (Pulse-Code Modulation). Some of the properties regarding the way sound is captured are presented and described below.

```
final int samplingRate = 44100;
final int audioSource = MediaRecorder.AudioSource.MIC;
final int channelConfig = AudioFormat.CHANNEL_CONFIGURATION_MONO;
final int audioFormat = AudioFormat.ENCODING_PCM_16BIT;
final int bufferSize = AudioRecord.getMinBufferSize(samplingRate,
                channelConfig, audioFormat);
```

```

AudioRecord recorder = new AudioRecord(audioSource, samplingRate,
                channelConfig, audioFormat, bufferSize);
```

samplingRate defines the number of samples per unit of time (second) taken from a continuous signal to build the discrete signal, and it is expressed on Hertz. The high frequency limit of human hearing is about 20 kHz and, according to the sampling theorem 2.3.4, the sampling rate must be twice the maximum frequency we wish to reproduce, so the sampling rate had to be at least 40 kHz, reason why it was chosen 44100. Besides that, 44100 Hz is the standard sampling frequency in digital audio.

audioSource tells the source to be used in the recording. In this case the sounds are captured by the built-in microphone (MIC).

channelConfig describes the configuration of the audio channels. Using MONO, all the audio signals are mixed together and routed through a single audio channel.

audioFormat defines the format in which the audio data is recorded. The data is represented in the PCM format using 16 bit per sample, meaning that the values will range from -32768 to 32767 (2^16 = 65536).

bufferSize is the minimum buffer size required for the successful creation of an AudioRecord object.

4.1.2 Prepared for Multi-core Smartphones Generation

The first dual-core smartphone was launched early on this year but 2011 may even usher in the age of the quad-core mobile. The multi-core processor (currently two 1GHz cores) offers more power to be squeezed out of the processor, as the cores can be used in parallel in order to speed up performance.

However, it is necessary that the device software is optimized to take advantage of the parallel processing capability. For this reason, this application uses multi-threading, particularly in the I/O operations. This option was taken considering the incredible renewal speed in the area of the smartphones and, also, the great receptivity of the users to these changes.

In a multi-core system, these threads will take maximum advantage as they can run concurrently on separate processors, which should enable a noticeable speed and performance hike, particularly when it comes to multitasking. In addition, they can also improve program's performance on single processor system by permitting the overlap of input and output or other slow operations with computational operations.

Below, there is a simplified segment of code showing the use of multi-threading in I/O operations.

RecordThread - responsible for reading the incoming data.

```
do{
    data = new short[WINDOW_SIZE];
    framesRead = recorder.read(data, 0, WINDOW_SIZE);

    if(AudioRecord.ERROR_INVALID_OPERATION != framesRead){
        blockingQueue.put(data);
    }
} while (isRecording);
```

ProcessThread - responsible for writing the data into the file.

```
short[] tempArray = new short[WINDOW_SIZE];

while(isRecording || !blockingQueue.isEmpty()){
    tempArray = blockingQueue.take();
    framesWritten += outputFile.writeFrames(tempArray, WINDOW_SIZE);
}
```

4.2 Digital Signal Processing Algorithms

The algorithms and methodologies used to process the audio files are described in this section. Firstly, it is presented the approach regarding the heart sounds and then the techniques used in the lung sounds analysis.

4.2.1 Heart Sounds

4.2.1.1 Before applying the algorithms

As it was already referred there are two different domains that might be used in the signals analysis, one with respect to time and the other one to frequency. A time-domain graph shows how a signal changes over time, whereas a frequency-domain graph shows how much of the signal lies within each given frequency band over a range of frequencies. It can also give information on the phase shift that must be applied to each sinusoid in order to be able to recombine the frequency components to recover the original time signal.

A given signal can be converted between the time and frequency domains using a set of mathematical operations. Below are presented some lines of code, written in Java, along with comments that explain these operations one by one.

Say we have already read the ".wav" file into an array of doubles, *values_Re*, and the size of the data is *sampleSize*. The FFT computation is faster if the size of the input data is a power of 2. Therefore, we start by meeting this need.

```
% Use next highest power of 2 greater than or equal to length(x) to ...
    calculate FFT.
nfft = Utils.nextpow2(sampleSize);

% Create array of doubles and fill it with the data.
double x[] = new double[nfft];
int i=0;
for(; i<sampleSize; i++){
    x[i] = (double)values_Re[i];
}

% Pad with zeros so that length(fftx) is equal to nfft.
for(; i<nfft; i++){
    x[i] = 0;
}
```

Secondly, we compute the discrete Fourier transform using the FFT algorithm. Then we just drop the second half of the FFT array. When the length of the data array is even, which in this case was forced to be (nfft is a power of 2, and, therefore, even), the magnitude of the FFT will be symmetric, such that the first $(\frac{nfft}{2}+1)$ points are unique, and the rest are symmetrically redundant.

```
% Calculate fft.
FastFourierTransformer transformer = new FastFourierTransformer();
fftx = transformer.transform(x);
```

```
% Calculate the number of unique points.
numUniquePts = (int) Math.ceil((nfft+1)/2);

% FFT is symmetric, throw away second half.
fftx = Utils.subArray(fftx, 0, numUniquePts);
fftx_filtered = fftx;

```

```
% This is an evenly spaced frequency vector with NumUniquePts points.
frequencyArray = new double[numUniquePts];
for(i=0; i<numUniquePts; i++){
    frequencyArray[i] = (double) i*((double)sampleRate/nfft);
}
```

The DC component of x is fftx(0) and fftx($\frac{nfft}{2}+1$) is the Nyquist frequency component of x. In case the length (nfft) is odd, the Nyquist frequency component is not evaluated, and the number of unique points is $\frac{nfft+1}{2}$. This can be generalized for both cases to ceil($\frac{nfft+1}{2}$) .

```
int i=0;
magnitudeArray = new double[numUniquePts];

% Take the magnitude of fft of x and scale the fft so that it is not a ...
    function of the length of x.
% Then, take the square of the magnitude of fft of x.
for(i=0; i<numUniquePts; i++){
    magnitudeArray[i] = ...
        Math.pow((double)fftx_filtered[i].abs()/sampleSize, 2);
}

% Since wzedropped half the FFT, we multiply mx by 2 to keep the same ...
    energy. The DC component and Nyquist component, if it exists, are ...
    unique and should not be multiplied by 2.

% Odd nfft excludes Nyquist point.
if((nfft % 2) == 1){
    for(i=1; i<numUniquePts; i++)
    magnitudeArray[i] = magnitudeArray[i]*2;
}
else{
    for(i=1; i<numUniquePts-1; i++)
    magnitudeArray[i] = magnitudeArray[i]*2;
}
```

Finally, the power spectrum is plotted and the result looks like the following:

The horizontal axis of the display is labeled with the word frequency, denoting the domain of the measurement. It is noticeable that all the power is concentrated in a narrow interval of frequencies ranging from 50 Hz to 200 Hz. The absence of a single peak on the curve, although, shows that there is no predominance of a single frequency within the range of frequencies covered by the width of the display.

This tell us important information for processing the data, particularly useful for filtering the information that really interest us, since a recording taken from the mobile is

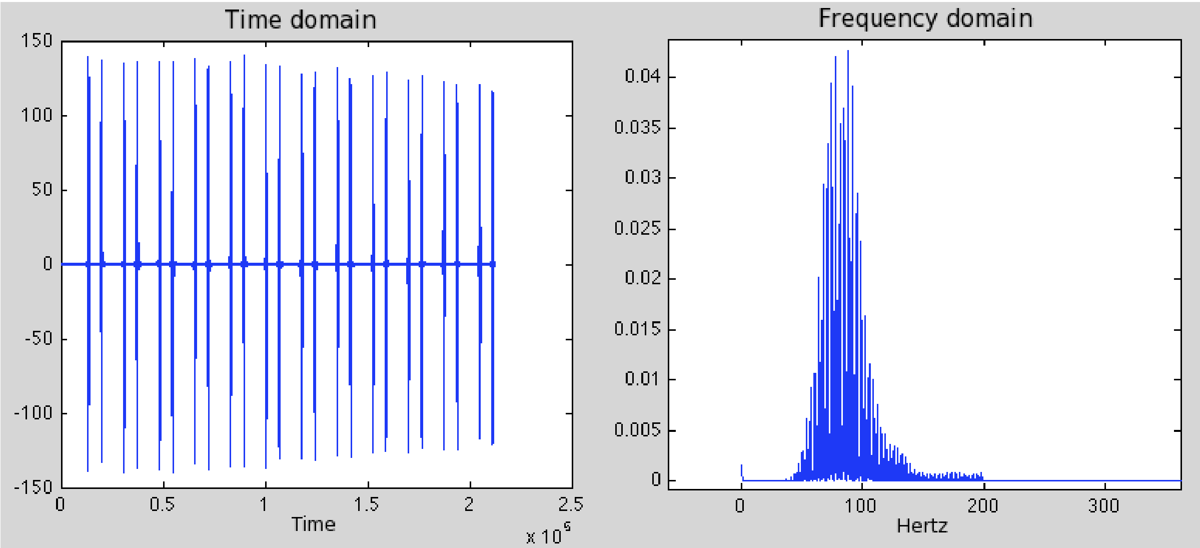

Figure 4.2: Heartbeat sound in time and frequency representation.

not so "clean", i.e., it comes with a wide variety of noises that are obviously unavoidable but necessary to put away.

4.2.1.2 Heartbeat rate detection

The high sample rate used on the recordings (44100 Hz) offers, on one hand, a good quality of the sound but has, on the other hand, an handicap since it significantly increases the amount of data to be processed, making it harder and more demanding. Each record has a duration of approximately 20 seconds which means the corresponding audio buffer will have (20×44100) of length, that is, 882.000 positions.

The sound that is shown on Figure 4.3 is a little bit smaller, it lasts 10 seconds, but is enough to explain the method used to get the heartbeat rates. It is firstly presented the original signal and then, below it, the same signal after being filtered.

The red and blue points marked on the chart correspond respectively to the heart sounds S1 and S2. These two sounds are present in a healthy patient and should be the first identifiable heart sounds. S1 precedes S2 and they can be described simply as the basic "lub-dub" sounds. In order to calculate the heartbeat frequency it is necessary to identify the peaks, colored red and blue.

The data is read and processed in windows of size 2000. This means that 1 second of sound is traversed shifting the window 22,05 times $(\frac{44100}{2000})$. For each windows (actually, for each position of the window, since it is a single window that is just being translated), the maximum value that is covered by the width of the window is calculated. This value is then saved into a list containing every window's maximum.

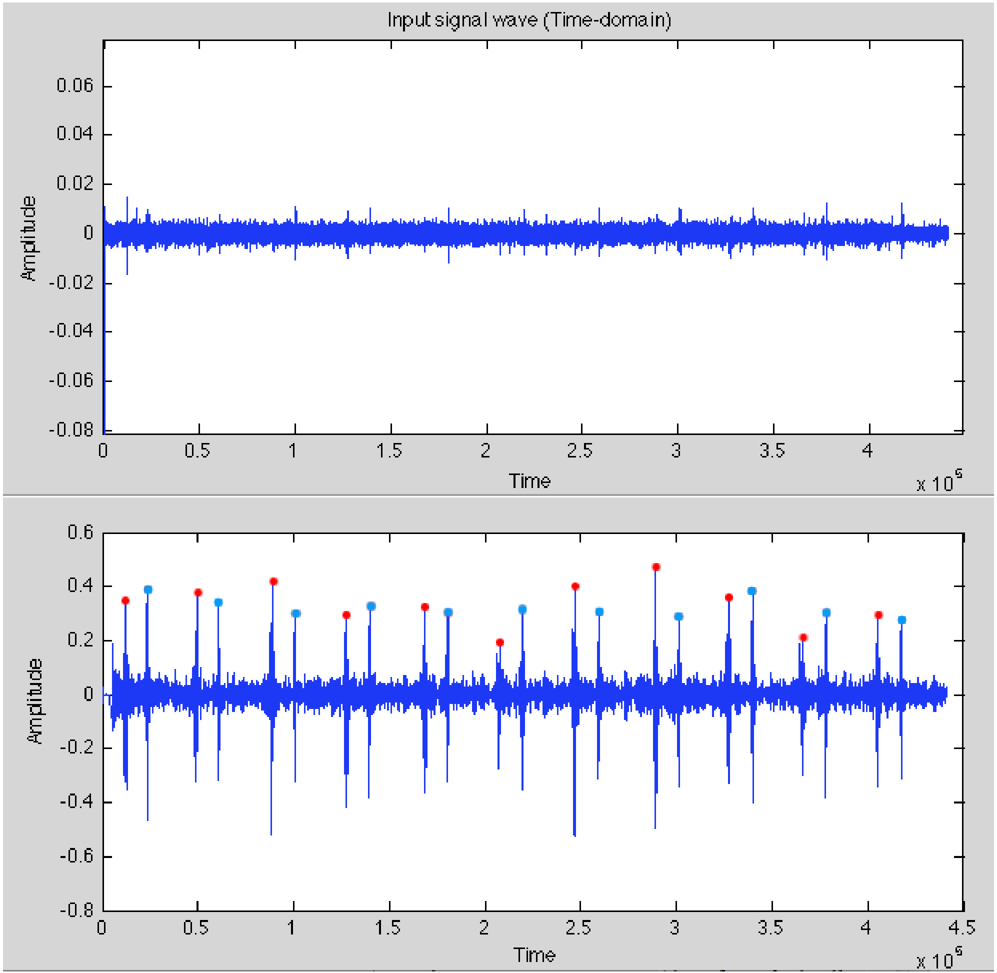

Figure 4.3: Collecting the peaks of the signal after filtering.

Each time this list is updated, it is verified if a new peak was achieved. To do so, several points are taken in consideration:

- The penultimate point on the list (last window's maximum) has to be higher (in amplitude) than the previous and following points (the following point was precisely the last to be added to the list which means it is the current window's maximum);
- The height of the point must be considerably greater than the average height of the signal so far. To discard points that might pass the first condition described above but that correspond to peaks of non-interest (for instance produced by unwanted noise), an eliminating factor is introduced, which multiplied by the average height of the signal gives a more reliable boundary.

$$pointHeight > factor * averageHeightOfTheSignal$$

This factor is based on some tests performed in different samples. It is still a very low barrier and does not guarantee that points above this limit are in fact peaks resulting of the heartbeat, but it does help discarding insignificant maximums.

If a new peak is found, the corresponding point is added to a list of peaks. In case the peaks list is not empty (there is at least on peak found), some more checks can be done in order to increase the confidence level of the algorithm:

- The distance between the last peak found and the possible new peak must be greater than $\frac{sampleRate}{6}$. This establishes a minimum time that must elapse between 2 consecutive peaks, and this minimum value corresponds to the time that goes from the sound S1 to the sound S2 of the cardiac cycle. The time that separates two consecutive heartbeats (from S2 to the next S1) is obviously longer.
- Finally, the amplitude of the point is compared to the previous peak's amplitude. The study of the available samples showed that, although there may be a considerable variation in wave amplitude in the different heartbeats, this variation is limited and it is thus possible to establish a minimum relation that must be verified between the distinct peaks. In the atrial fibrillation samples, where this variations were more noticeable, these did not go under the threshold of 0.25, which means that the peaks corresponding to the sounds S1 and S2 in the same cardiac cycle follow the relation:

$$\frac{min(amp(s1), amp(s2))}{max(amp(s1), amp(s2))} >= 0.25$$

4.2.1.3 Atrial Fibrillation

In the previous section we have shown a normal heartbeat signal in its time-representation, where the heartbeats were clearly periodic. In atrial fibrillation, things work differently. As it can be seen in Figure 4.4, the elapsed time between two consecutive heartbeats is rarely similar to the previous one.

Atrial fibrillation is distinguished from other arrhythmias for its non regular working. Generally, other arrhythmias are characterized by an accelerated but regular heartbeat. In AF, the heart might beat too fast (>120) or not so fast (< 100) but always it must be noticeable the absence of periodicity and regularity in heartbeat.

AF cannot be diagnosed based only on auscultation. In fact, what doctors can detect through auscultating is simply the existence of an arrhythmia, of any kind. Obviously that their experience and also the statistics makes them lean toward a certain type of arrhythmia. In more than 50% of the cases it is FA. In any case, only a ECG can actually confirm the pathology. So, what we seek here is to distinguish a normal and regular heartbeat from a irregular one (sometimes chaotic). We do that by analyzing the heartbeat velocity variation, i.e., comparing the different elapsed times within two consecutive points.

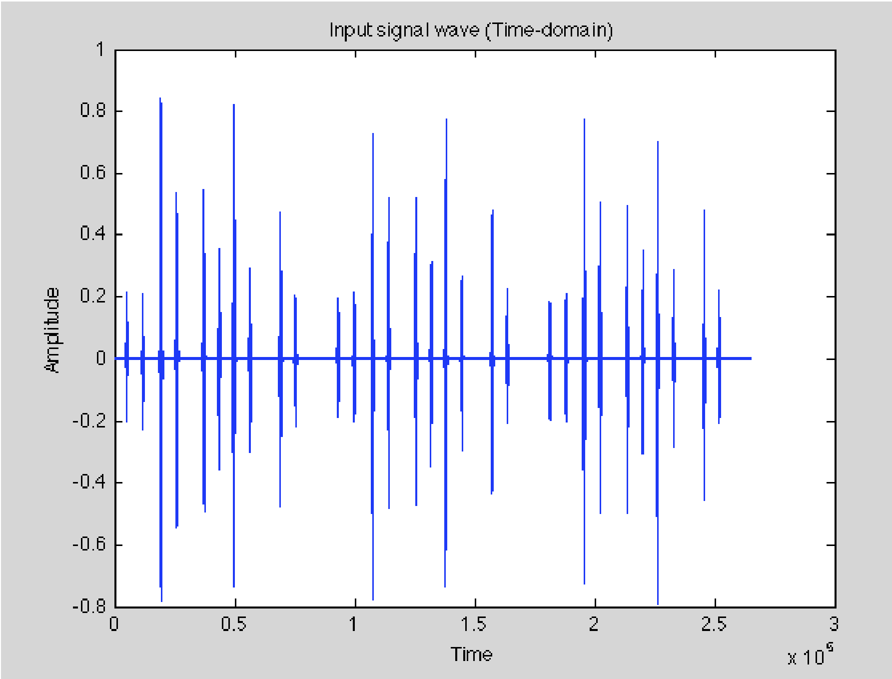

Figure 4.4: Atrial fibrillation appearance in the time-domain.

Calculating the heartbeat rates for the signal in Figure 4.4 we get the following values:

93, 73, 106, 68, 54, 92, 74, 106, 69, 55, 91, 74, 106, 68

Now calculating the variations between consecutive heartbeats we get the results presented in Table 4.1.

Table 4.1: Heartrate variation in atrial fibrilation.

Heartrate	93	73	106	68	54	92	74	106	69	55	91	74	106	68
Variation		-20	+33	-38	-14	+38	-18	+32	-37	-14	+36	-17	+32	-38

Considering the absolute values, we get an average variation of 28,4. This means that the patient at rest, comfortably sitting on a chair making no physical effort at all, would have an average variation of 28 on their heart rate, accurately measured by a blood pressure machine, which is astonishingly high.

Very small variations are usually found in a healthy heart. There is no standard threshold value to consider a variation high or low. However, a deviation of 10 points or higher would not go unnoticed by a doctor. We take not only into account the abrupt changes on heart rate but also the frequency within they occur. It would be erroneous to associate with FA if only one of the variations would be significantly high, which could have been caused by, for instance, noise that we were unable to remove. Therefore, we consider only situations where the average variation value is above 10, creating a more flexible boundary since it will not discard situations where smaller variations may occur, as long as higher differences are verified.

4.2.2 Lungs Sounds

Important information regarding the physiology and also pathology of the lungs can be found in respiratory sounds. Both the frequency and the time analysis are useful, depending on what we want. Although containing the same information, these two different representations display it in a distinct way, making facts more or less evident. This means that to get a specific information, it might be easier to go for one of the representations, while in another situations the other option might be preferable.

4.2.2.1 Crackles Detection

A lung sound with adventitious sound components may indicate possible pathologic changes in the pulmonary tissue. Crackles, as referred in 2.2.1.3, are discontinuous explosive sounds very frequent in cardiorespiratory diseases. Some people describe the sound as being similar to the crackling sound heard as wood burns, while others compare it to the sound produced when rubbing hairs.

Non-surprisingly, crackles appear in the time domain as intermittent spike-like deflections, as show in Figure 4.5. They are generally short in time and their frequency content typically is wide, ranging from 100 to 2000 Hz or even higher. Although crackles can sometimes be detected in a healthy patient, for instance during a deep inspiration, they almost always reflect a sick lung.

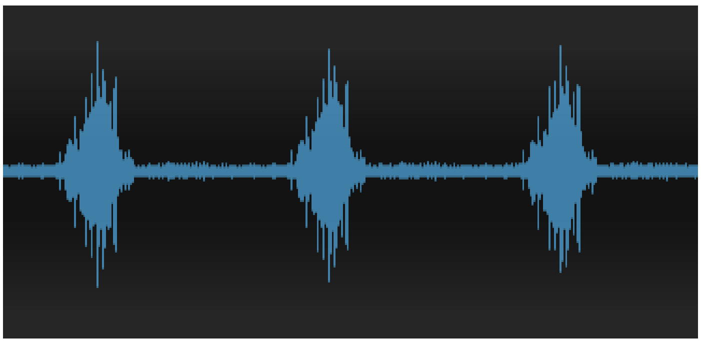

Figure 4.5: Crackles sample

The two distinct types of crackles, fine and coarse, differ on the timing they occur, on the time they last, on the sound they produce and also, of course, on the consequences they can involve. Table 4.2 distinguishes them.

Table 4.2: Distinguishing between fine and coarse crackles.

	Fine crakles	**Coarse crackles**
Sound	Soft and high pitched	Loud and low in pitch
Duration	Very brief	Long lasting
Timing	Late inspiratory	Early inspiratory
Consequences	Pulmonary fibrosis Congestive heart failure	Airway diseases (ex: bronchiolitis)

On auscultation fine crackles are in general higher pitched, less intense and of shorter duration than coarse crackles. Crackle waveform features such as initial deflection width, largest cycle deflection, number of zero crossings, etc. are important characteristics that must be measured in order to automatically detect and classify the crackles, using computer algorithms.

Before processing the signal, it is important to have separated the sounds produced by the crackles from other sounds that the stethoscope might have captured. These not only include internal sounds coming from other organs (heart) but also environmental noise that we can not avoid. The latter generally consists of very low frequency components below 40Hz. On the other hand, despite the fact the sound recording is performed separately and in a lung-auscultation specific zone of the body, still some heart murmurs might be audible.

For that reason, we eliminate frequency components below 200Hz, which does not removes relevant information concerning the crackles. The energy produced by the crackling sounds largely concentrates above 200Hz, as can be seen in Figure 4.6. In this situation they range approximately from 200Hz to 1500Hz, but they can reach higher values.

The 2d heat map represents the content of the respiratory sound of a 55 year old woman. The sound was recorded over her right posterior lung. The horizontal axis represents the time while the in the y axis are represented the frequencies. The color scale indicates the sound intensity from low (grey) to loud (red). The air flow is displayed at the top with the inspiration (I) period being represented above zero whereas the expiration (E) is reflected below zero. Fine crackles can be identified in the end of the inspiration, resulting in a great increase of high frequency components on the spectrogram.

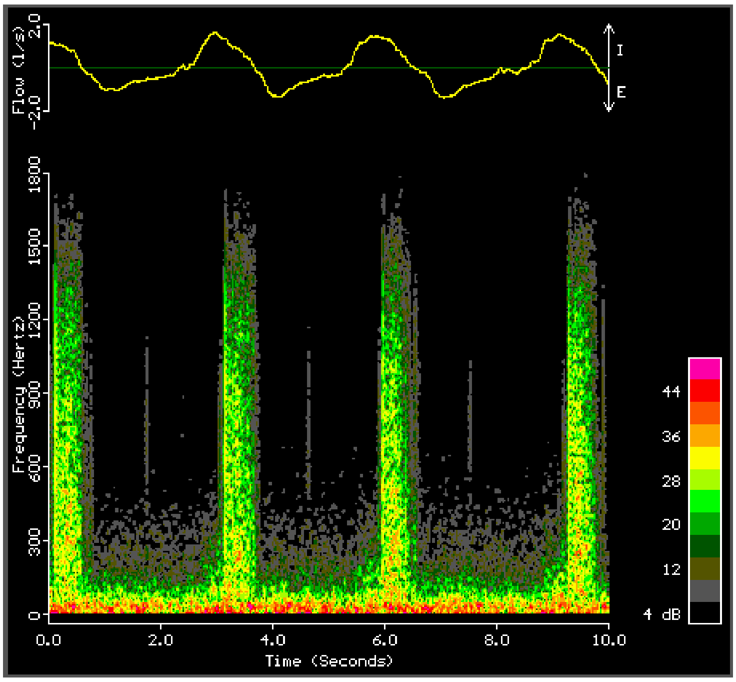

Figure 4.6: Crackles' frequency components and timing.[1]

The picture 4.7, taken after filtering the signal through a high pass filter, offers a lot more informative sample for the presence of crackles. The cut off frequency was set at 200 Hertz.

The picture speaks for itself. We get a quite good approximation to a typical crackles waveform. The noise was completely eliminated and also part of normal breath sounds disappeared. Listening to the output sound file, we can hear nothing but the crackles. A healthcare professional would probably find it weird and artificial, since the sound was modified, but for DSP it is just fine. The key information was kept and this filtered sample allows us to verify the presence or absence of crackles. Note that if no crackles would be detected, the signal positioned at the bottom of the picture 4.7 would have an horizontally flat waveform, close to zero.

What we try to do is to separate the frequency components produced by normal breath sounds from the ones produced by crackles. Although this is not achievable in perfection, since there is not a frequency interval that clearly separates them, the key information is well distinguished. After having filtered the crackling sounds, through a highpass filter with a cut off frequency of 200 Hertz, we calculate the respective frequency spectrum. Then, we compute the same frequency spectrum but now of the original file, i.e., before having been filtered, exactly as it was recorded by the smartphone.

[1] http://www.rale.ca/Crackles_b.htm

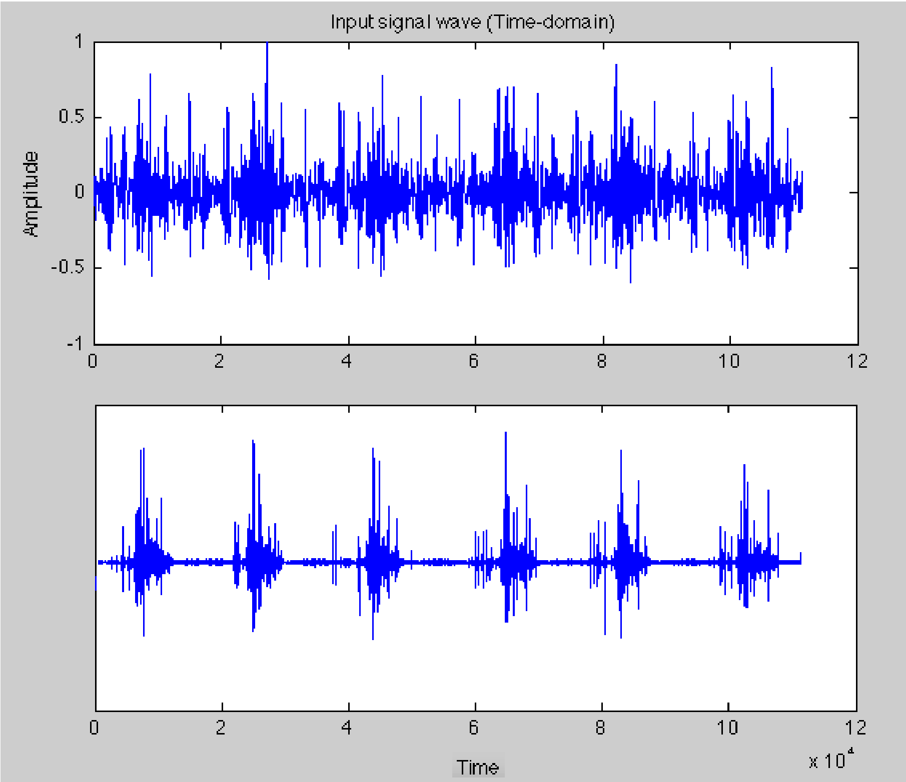

Figure 4.7: Crackles recording before and after filtering.

Having done this, we have two different arrays: one representing the frequency spectrum of the original audio recording and the other one representing the frequency spectrum that characterizes crackles. These arrays contain the energy produced in each frequency level.

For each array, we calculate its total energy, by summing up all the their values. Finally, we just compare this two energies in the form:

$$\frac{\text{Energy produced in crackles frequency range}}{\text{Energy produced in all frequencies}} \times 100$$

This gives us the percentage of the total energy whose components in frequency are covered by the possible frequency range of crackles.

In a normal vesicular breath sound, this percentage shows up to be very low, inferior to 5%. In contrast, when there is clear evidence of crackles, this percentage rises to considerable values such as 30%, 50%, 60%, depending on the type of crackles. But it seemed evident that there is a comfortable confidence interval, between 5% and 25% that separates this two happenings. Therefore, we defined a threshold of 10% under which we consider there is no evidence of crackles and above which we do consider there are crackles.

4.3 Mobile application

Since the very first lines of code were produced, it was intended to produce a GUI (graphical user interface) as much simple and pleasant as possible. But it is also important to be efficient, i.e., that does not induce the users in error, or at least that does it few times. In any application, its usability and efficiency are very important points in the decision between using it or not. Bearing that on mind, the contents were placed in the most concise and simple way possible, accompanied whenever possible with colors and images in order to be more aesthetically pleasing.

To develop the application it was used the Android SDK tool, programming in Android and XML. The code was structured and organized in order to simplify possible future changes. Everything that concerns the layout, the appearance and organization of the diverse components of the screen is defined in XML files. Each screen, an *Activity* in Android, has its own XML file whose name is similar to the activity that inflates it. All the strings used in the application are defined in a single file, named strings.xml.

Locally, data is stored in a SQLite database, which is updated every time the results of a new exam are returned from the web service.

4.3.1 The Graphical User Interface

The key for the acceptance of a mobile application is the *user experience* it provides, i.e., how easy and pleasant it is for the user to navigate through the application features. No matter how useful it might be if people cannot use (too complicated) or they just do not like its graphical appearance. Therefore, we tried to design a user-friendly interface, keeping it as much simple and intuitive as possible. We included pictures and icons whenever we found it helpful. The user has also access to help buttons, test guides and a tutorial video. Almost all the screen views are presented in this section.

Let us start with the *Start menu*, which comes right after the splash screen with the application name and logo, not presented here. This screen appearance is shown in Figure 4.8.

In the center of the screen there is a circular menu which consists of 6 clickable layouts: (1) *Exam now*, (2) *History*, (3) *Last exam*, (4) *Weight*, (5) *Help*, and (6) *About*. The reason why we have numbered the different options has to do with their spatial organization, that we will now explain. It was not by mere chance (neither randomness) that the "buttons" were placed in these positions. In fact, it was based in some tests carried out without without telling people in advance that we were about to check their interaction with the application. We just kindly asked them to try it.

We found that the majority of users firstly visualize the option in the center of the circle. It is the one that immediately captures the attention of their eyes. Consequently, it is also the one that is more easily clicked.

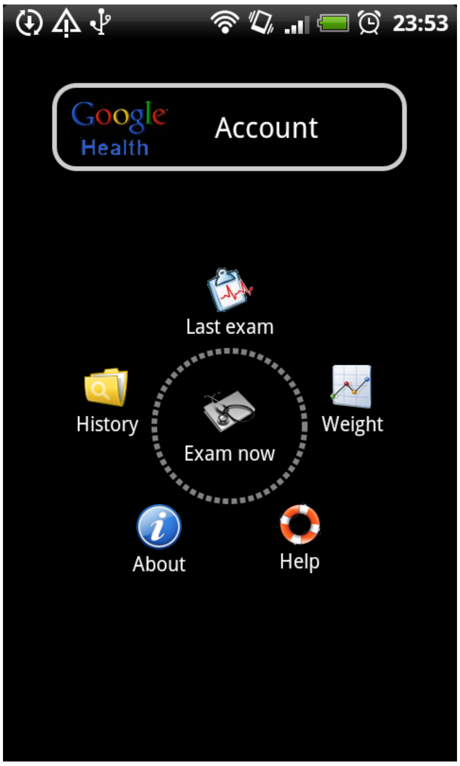

Figure 4.8: Look of the Start menu.

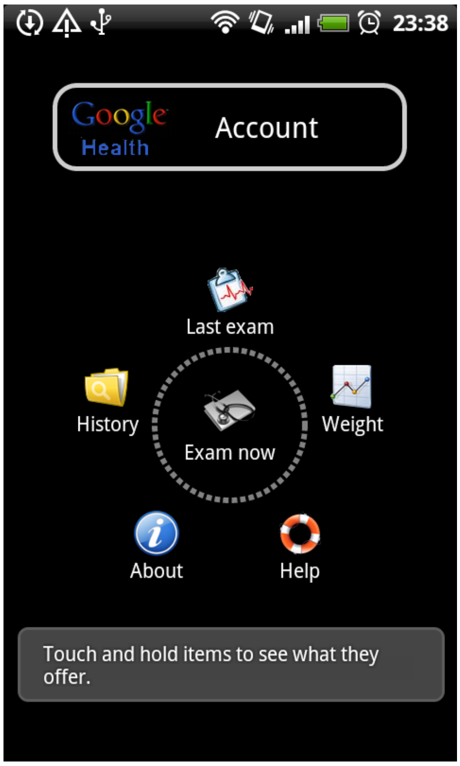

Figure 4.9: Toast notification display.

As most important functionality, the button to start the exam was placed here. After looking at this option, people tend to turn their eyes to the left, where it is placed the *History* folder, which is, we believe, the second most important taking into account the functionality it provides and the number of times that it will possibly be requested. The rest of the options are arranged as a clock and the degree of importance decreases clock-wisely.

The notification that appears on the bottom of the screen in 4.9 is displayed when pressing the help button. It tells us the users to press and hold the other buttons to get a clue about their functionality. This is presented later in this section.

Yet on this screen, at the top of it, we have the *Google Health Account* button. It was placed here, above the menu, not because it is more relevant, but because it is important that people see it, either in the first time the application is used so that an account is chosen, or in further uses where it might be selected somebody's else account. It would not be good to send the data into the wrong account. Clicking this button opens an *AlertDialog* with a list of selectable items like is shown in Figure 4.10. These items are Google accounts previously configured and saved on the *AccountManager* of the device. The list synchronization is done automatically.

Figure on right-side, 4.11, illustrates the result after picking the account *joao.c7@gmail.com*. The username is now visible on Google Health button.

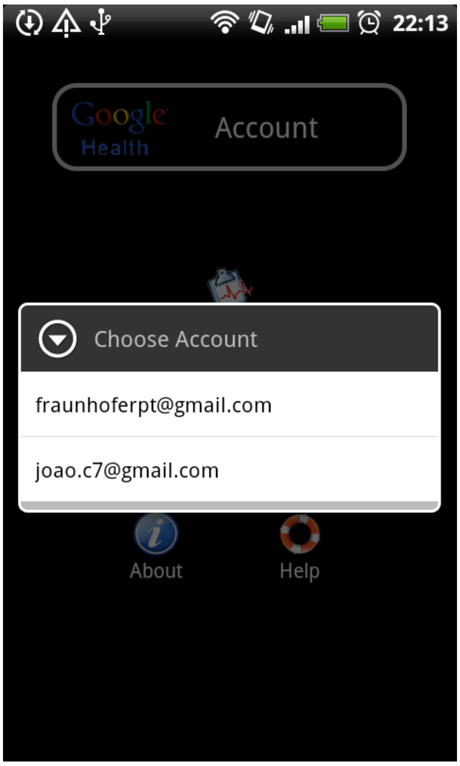

Figure 4.10: Choosing a Google account.

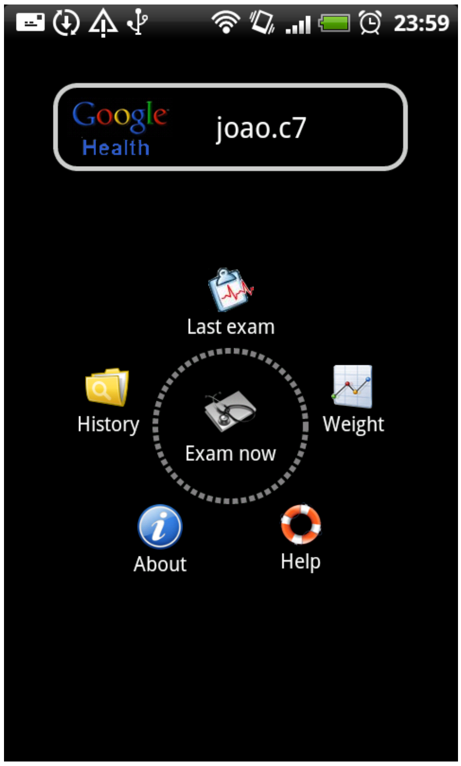

Figure 4.11: Result after choosing the account.

By pressing and holding a button from the start menu, a small dialog balloon is displayed explaining in few words the action implied on it. This is shown in Figure 4.12.

4.3.1.1 Exam

The option *Exam* from the *Start menu* allows us to examine our heart and lungs. This process is divided in different steps and textual guides and tips are presented for better results.

Firstly, the user is presented some important issues they should keep in mind, as shown in Figure 4.13. The first one concerns the environment where the exam will take place. This should be the most silent possible, since the noise can prevent from listening and

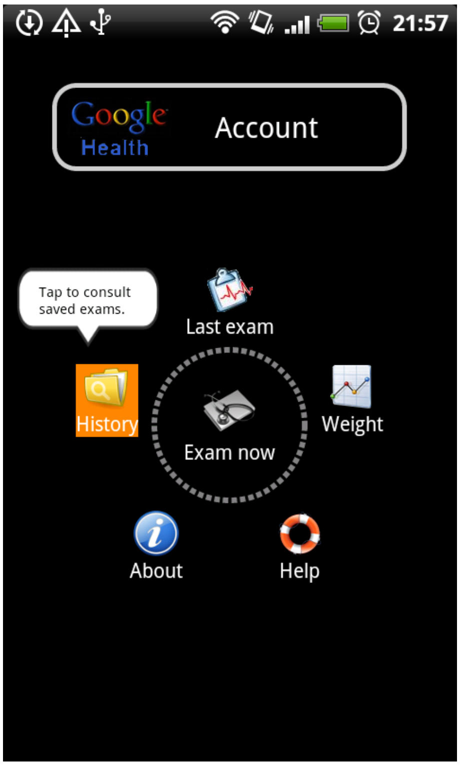

Figure 4.12: Help on a button.

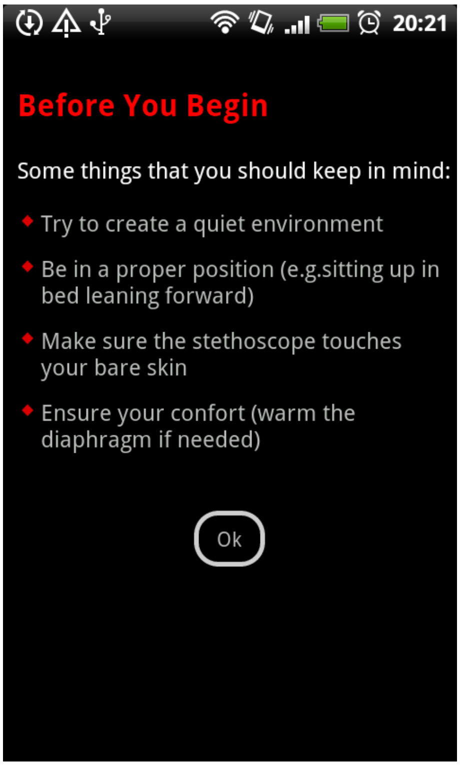

Figure 4.13: Some guidelines before the exam.

recording what really interest us. The position of the person is also important, they should avoid both lying and uncomfortable positions. The stethoscope should softly touch the skin and it may be warmed with the hands, in case it causes discomfort.

Then, after having read these tips and pressed the *Ok* button, it comes the part of recording the auscultation. For reasons concerning the quality of the sound samples, the auscultation of the two organs is spaced in time. The stethoscope must be positioned over different points on the skin depending if listening to the heart or to the lungs. Therefore, two identical screens were built for that purpose and each one includes a adequate picture showing the point where the diaphragm should touch the skin. It is important to note that this is a critical aspect for the proper functioning of the application. Murmurs and whispers coming from inside of the body, especially from the heart, are hardly listened with a good acoustic stethoscope so it turns to be even more difficult with this low-cost electronic stethoscope.

Figures 4.14 and 4.15 present the same screen in two different moments: before and after pressing the start button. On the latter, the circle, initially empty, is gradually filled as time passes. This is complete when the recording terminates.

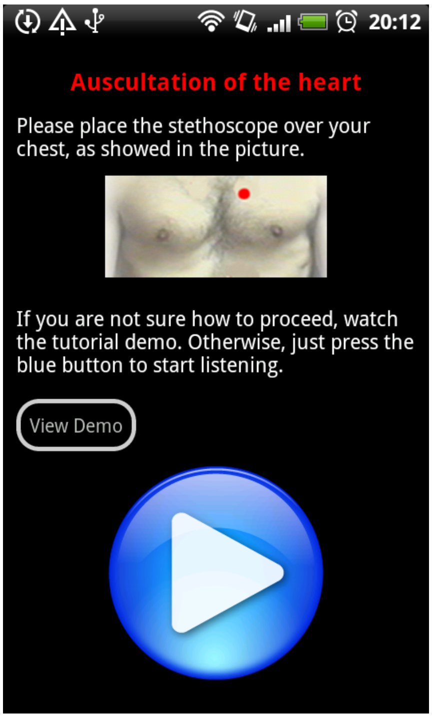

Figure 4.14: Exam screen before record.

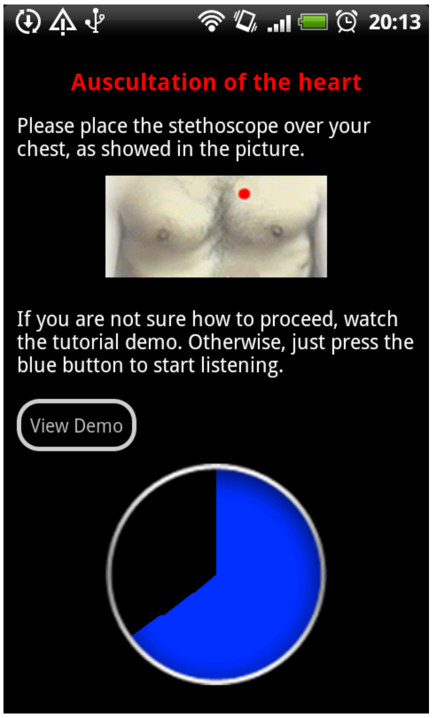

Figure 4.15: Exam screen while recording.

Tutorial video

Thinking on people that might feel unsure about how to properly use the stethoscope, this optional screen was built. Sometimes, a short explanatory video is all we need, and particularly in this case it is indeed of great value. On the last screen presented, where the user is asked to press the record button when ready, it is also given the chance to first navigate to this screen and watch a tutorial video.

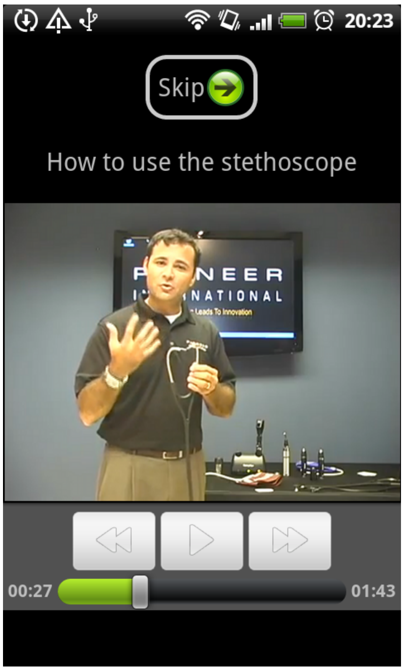

Figure 4.16: A short tutorial demo explaining how to auscultate.

History and Exam Results

By pressing the *History* option from the *Start menu* users have access to the history of tests. On this screen, a set of folders is listed. The appearance of the screen differs depending on its vertical or horizontal orientation. In landscape mode, the data is displayed in a GridView, which displays items in a two-dimensional, scrolling grid. For each grid element the icon appears at the top of it and the text at the bottom.

Figure 4.17: History folders listing in landscape mode - grid view.

All these folders contain exams inside, duly ordered by date. Thus, every time a new exam with no corresponding folder is built, a new folder is created. For instance, on a person that keeps a daily track of their conditions, in the beginning of every month (day 1), a folder for that month is created. The reason for organizing the files this way is to avoid having the data storage messed up, with hundreds of files all together, demanding more time and patience from the user to access the data of a specific exam, which would not be reasonable at all.

When in portrait mode, the items are display as a typical list, one after another vertically. The folder icon is placed on the left of the list item and the corresponding text on the right of it.

Navigating into one folder, a list of exams is displayed, similar to the arrangement of folders. The files are sorted in decreasing order, i.e., the last day of the month appears on the top of the list. The picture 4.19 evidences it. On its right, in 4.20, it can be seen how the exam results are displayed to the user.

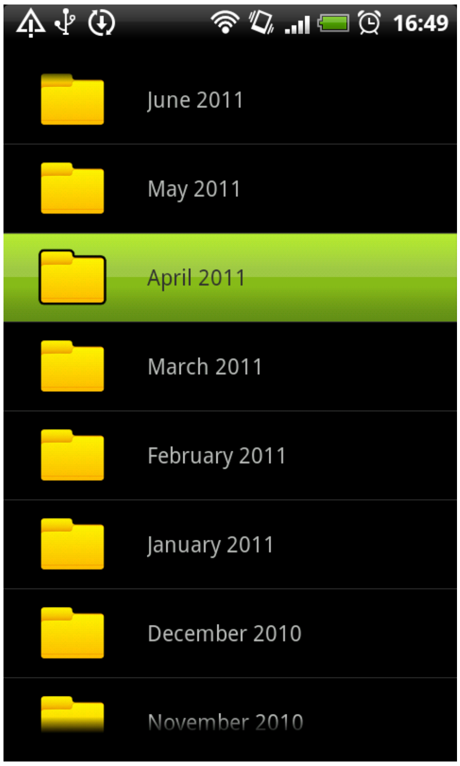

Figure 4.18: History folders listing in portrait mode.

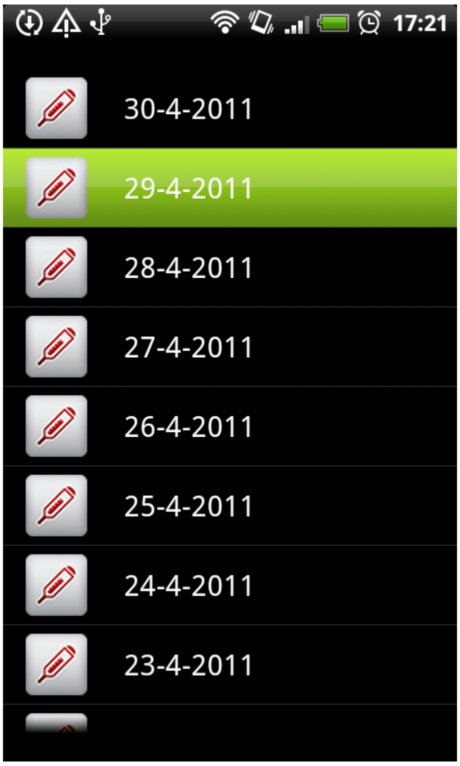

Figure 4.19: List of exams from April 2011.

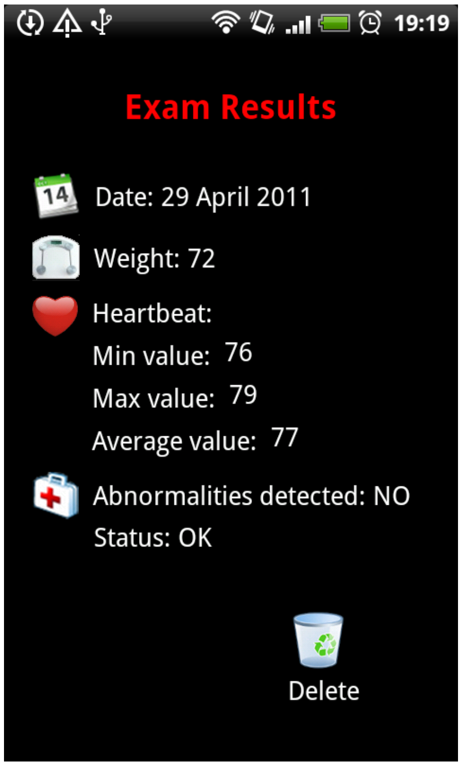

Figure 4.20: Results of an exam.

4.3.1.2 Weight Progress

The application also allows users to monitor their weight progress. It is displayed a line chart with the time being represented in the X axis and the weight on the Y line. The different weights values are marked with red points and a yellow line connects them. Near the points are displayed the absolute values of weight.

The display period can be changed on the top-right *Spinner* - a drop-down list that allows scrolling when the list exceeds the available vertical space on the screen, which is not the case.

Figures 4.21 and 4.22 exhibit these two screens.

4.3.1.3 Credits and Feedback

Information about the application can be found on *About* screen, navigable from the *Start menu*. Here, apart from some information regarding the application use, there is also a button which redirects users to another screen where they can send feedback about the application and its features. These messages will be particularly useful in improving it. Both screens are shown below.

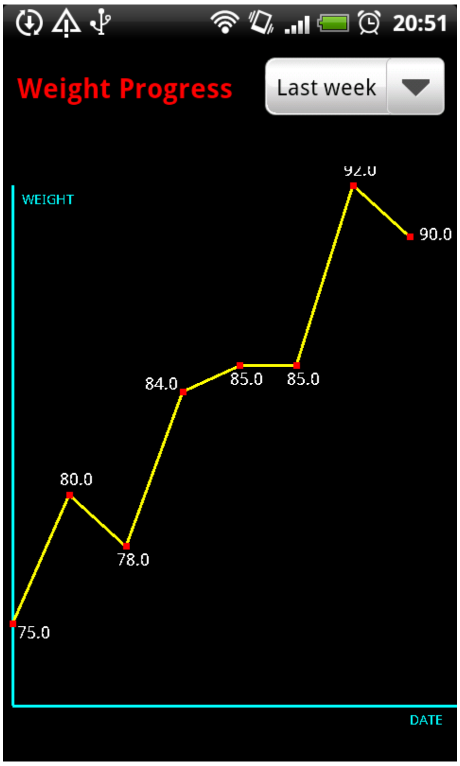

Figure 4.21: Weight evolution on the last 8 days.

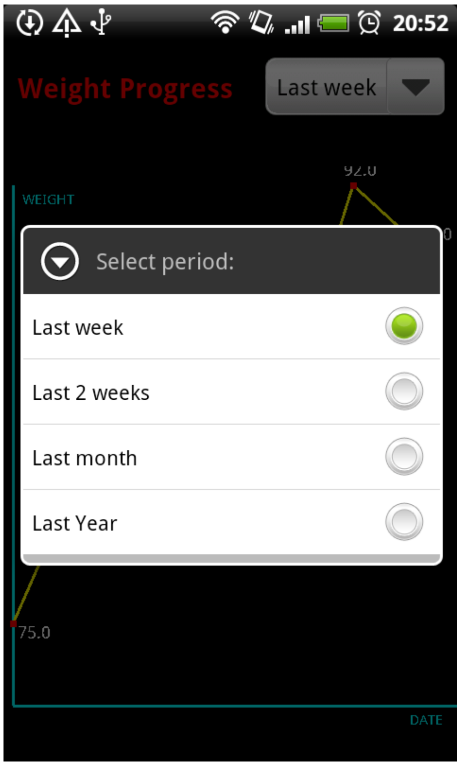

Figure 4.22: Changing the period of time.

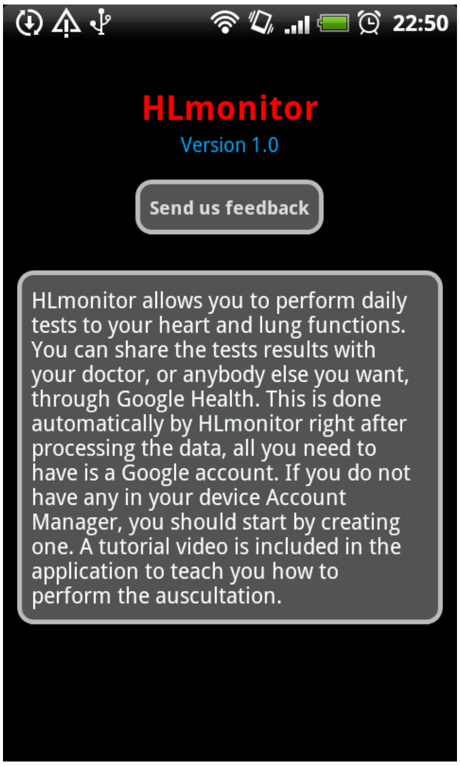

Figure 4.23: Credits of the application.

Figure 4.24: Commenting the application.

4.3.2 Local Storage

Locally, data is stored in a SQLite database, whose UML classes diagram is presented below. Each user is uniquely identified by their Google account and they can have a no number of exams. To save the different values of heart rates within an exam, it was crated the "Heartrate" table.

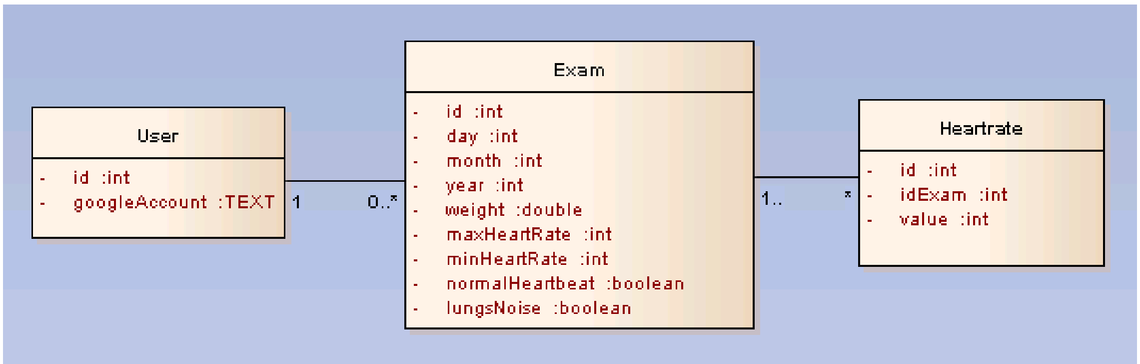

Figure 4.25: Local database tables.

4.4 REST Web Services Implementation

The web services were implemented using Netbeans 6.9.1. This IDE comes with an open source application server, called GlassFish, which was used to run our web service locally. Before starting describing the web service functioning, both from the web server and client side, we will explain the reason why the files which are sent via web service are firstly compressed into a ZIP archive. The web services follow the REST architecture and were developed using Jersey framework.

4.4.1 Data Compression - ZIP archive

The ZIP file format is a data compression and archive format. The most common extension among zip files is ".zip" and these contain one or more documents that have been compressed, using a specific algorithm, in order to minimize their size. An archive, besides taking less time to be downloaded or uploaded, also needs less space in the disk to be stored, using thus fewer system resources.

Why to compress

A 20 seconds audio record acquired using the smartphone and the sound properties specified in 4.1.1 requires 1.8 MB (megabytes) of disk space. And for each exam, we have two WAVE files which means around 3.6 MB of data to transmit to the web server, which is quite heavy.

After compression, the data size is reduced from 3.6 MB to 2.6 MB, approximately 72% of the original size and thus we have a compression rate around 28%, which is quite satisfactory.

Although the compression process means time-consuming, it will shorten the time needed to send the file in almost $\frac{1}{3}$ and thus, it is worth. By shortening this time, we reduce the probability of data lost and components' failures, and consecutively the overall failure of the system (for instance, in case the Wi-Fi connection is lost in the middle of the transmission).

4.4.1.1 The logic flow

For an overall view of the system functioning, it is presented in Figure 4.26 a sequence diagram that models the logic flow within the system.

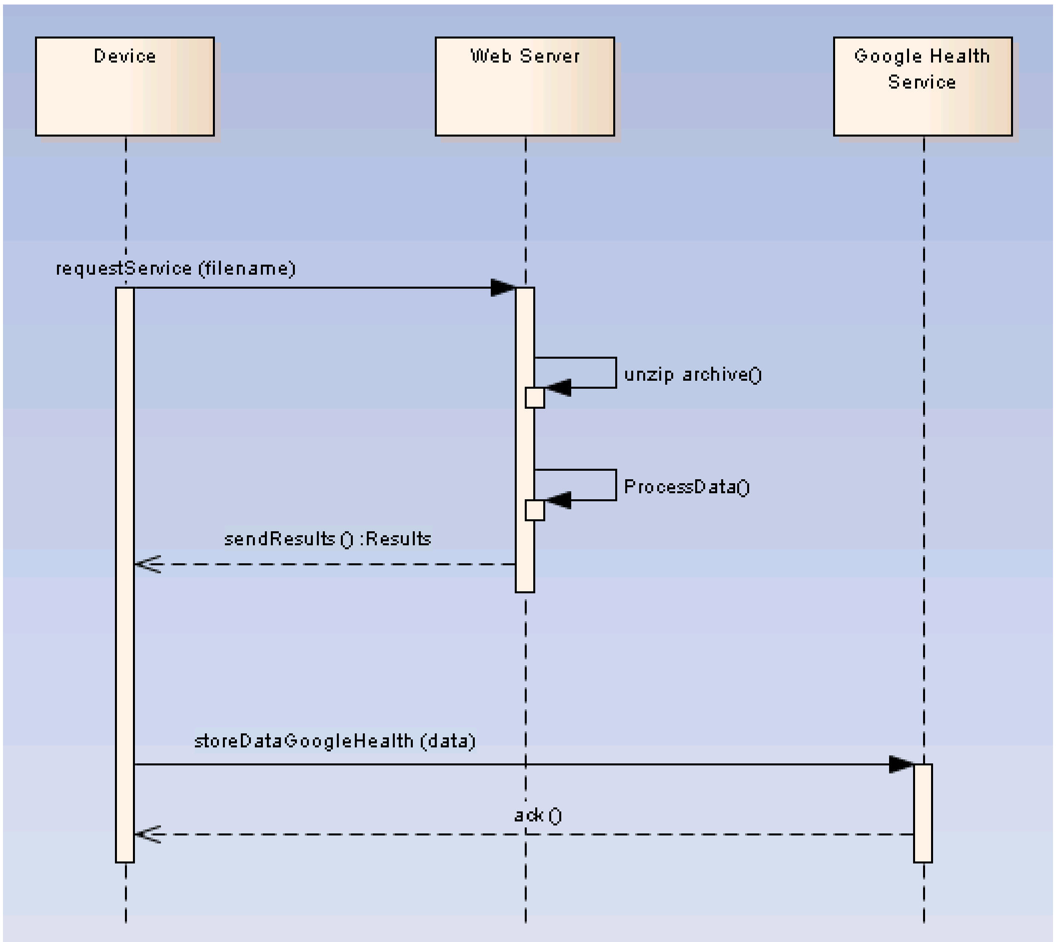

Figure 4.26: UML Sequence Diagram of the system.

4.4.2 Web Service Specification

Explained the reasons behind the compression of the files, we explain now the web service implementation as well as its operating mode. Let us firstly consider the server side, where it was implemented.

4.4.2.1 Web Server

For better understanding, few lines of code describing the web service behavior are presented and explained below.

```
@POST
@Path("test") @Consumes(MediaType.MULTIPART_FORM_DATA)
@Produces("application/json")
public String test(@FormDataParam("zipfile") File file,
                @FormDataParam("zipfile") FormDataContentDisposition info,
                @Context ServletContext context)
                throws FileNotFoundException, IOException {

                (...)
        Utils.decompressZipArchive(ofile.getAbsolutePath());
        Exam e = new Exam();
        DSP dsp = new DSP(oFile);
                (...)
        Gson gson = new Gson();
        String result = gson.toJson(e);
        return result;
}
```

The @POST annotation tells the web service that every time a client sends a request, the web server should receive some additional data sent together in the request. It contrasts with the GET request, which just fetches data based on a URL.

The @Path annotation defines a relative path for the URI. In the web service above, the Java class will be hosted at the URI path */test*. JAX-RS is particularly useful as it allows to embed variables in the URIs.

The @Consumes annotation specifies the MIME (Multipurpose Internet Mail Extensions) media types of representations the resource consumes (accepts). Therefore, the web service will be only accepting MIME of the type *MULTI_PART_FORM_DATA*.

A multipart-form-data type, as the name suggests, consists of different parts. The content of each part is specified in the header as a *name-value* pair, where the *name* defines the field name within the form and the *value* is "form-data". The parts of the multipart stream, i.e. the form fields, are sent in the order in which they occur within the form. In our web service, there is only one part: the ZIP archive.

The @Produces annotation specifies the MIME media types of representations the resource produces and sends back to the client. Our web service produces contents in JSON, as specified by the MIME type "application/json".

When a client sends a POST request, the web service starts by copying the received file to the web server. Then, the ZIP archive is uncompressed and it is started the digital signal processing of the WAV files, using the algorithms and techniques described in section 4.2. An instance of the object type *Exam* is created and assigned the resulting values from the sound files processing. The object is converted into a JSON string and then returned by the web service.

The *Exam* consists of:

- A **date** in the form dd/mm/yy hh:mm:ss
- A list with the **heartbeat rates** captured during the auscultation
- The **maximum** value of heart rate detected
- The **minimum** value of heart rate detected
- The **average** value of heart rate
- An indicator the measures a possible completely deregulated functioning of the heart, regarding **Atrial fibrilliation**
- An indicator of the presence or absence of pathologic sounds in the breathing, specifically the **crackles**

The implemented REST service is:

- platform-independent (no matter the server is Unix, the client a Mac, etc);
- language-independent (C can talk to Java, etc.);
- based on the standards.

4.4.2.2 The client

From the client-side, a POST request in accordance with the specified in 4.4.2.1 is sent to the web server. This is accomplished with few lines of code, as presented below. Some comments were included to explain the statements.

```
HttpClient httpclient = new DefaultHttpClient();
File file = new File("/data/data/com.hlmonitor/files/test.zip");

%Prepare request and response objects
HttpPost httppost = new ...
    HttpPost(http://10.211.103.127:8080/RestService/Res/files/test);
HttpResponse response;

%Create multipart entity
MultipartEntity dataEntity = new MultipartEntity();

%Add the .zip Archive to multipart entity
dataEntity.addPart("zipfile", new FileBody(file, "application/zip"));
httppost.setEntity(dataEntity);

try {
        %Execute the request
        response = httpclient.execute(httppost);
        %Get the response entity
        HttpEntity entity = response.getEntity();
        %Convert from stream to string
        result = EntityUtils.toString(entity);
    }
```

4.4.3 JSON

JSON stands for JavaScript Object Notation and it is a lightweight data-interchange format. As the noun self-suggests, it is based on JavaScript Programming Language and it is ideal for data exchange as it is completely language-independent, although it uses some conventions from object-oriented languages.

JSON is built on two structures: (1) a collection of name/value pairs (object, struct, hashtable, associative array) and (2) an ordered list of values (array, vector, list).

The JSON format is often used for serializing and transmitting structured data over a network connection. It is primarily used to transmit data between a server and web application, serving as an alternative to XML. Below is presented a JSON message returned by the web service, which conforms with the JSON syntax that is shown in Figure 4.27.

{"heartrates":[76,75,76,76,76,75,76,76,76,75,76], "af":false,"crackles":false}

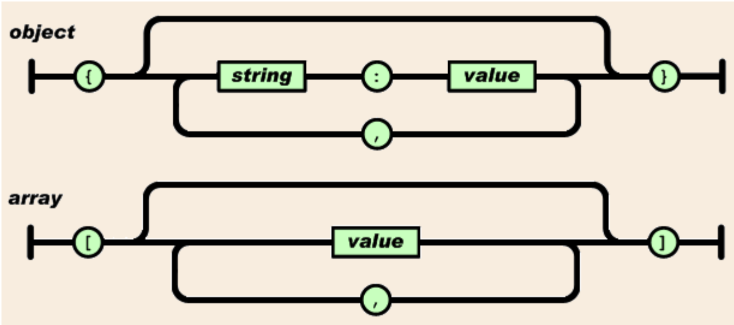

Figure 4.27: JSON syntax.[2]

4.5 Integration with Google Health

It was our intention to integrate our application with an online platform where it would be possible to store and maintain health information. The options were very limited, the only alternative capable of providing the same Google Health does, would be perhaps Microsoft Health Vault, but it would require a license to work on it, so we decided for GH.

The device, after receiving the results of an exam from the web service, automatically starts sending them to GH so that they will be available online. But for this to be possible, the user must own a Google account, which is free to get, and must have it stored on the mobile, on the *AccountManager*. The *AccountManager* provides access to a centralized registry of the user's online accounts. All the user needs is to enter their credentials, username and password, once.

By doing it, they are granting applications access to online resources with simple click approval and they will not be asked in the future to repeatedly insert the credentials, since they will be available on the device. The first time the application will try to store the results in Google Health, it will search for saved Google Accounts on the device. If Google accounts are found, the user will be asked to choose one from the list, like is illustrated in Figure 4.28. Below it is presented a small example of the syntax used to submit a result of a test (in this case, weight).

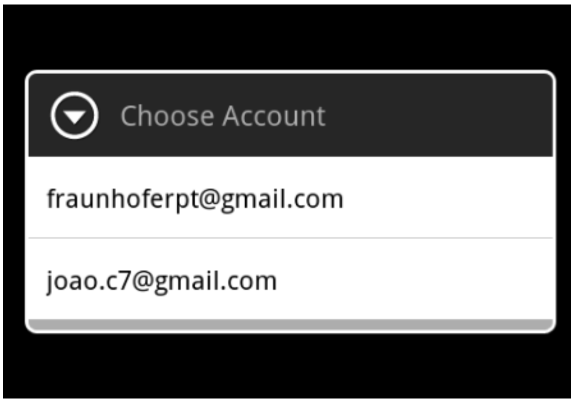

Figure 4.28: Choosing an account to store data in Google Health.

It is a fragment of code, not the complete XML file, and conforms to GH API CCR Reference. Part of the supported CCR elements are shown in Figure 4.29.

[2] http://mql.freebaseapps.com/ch03.html

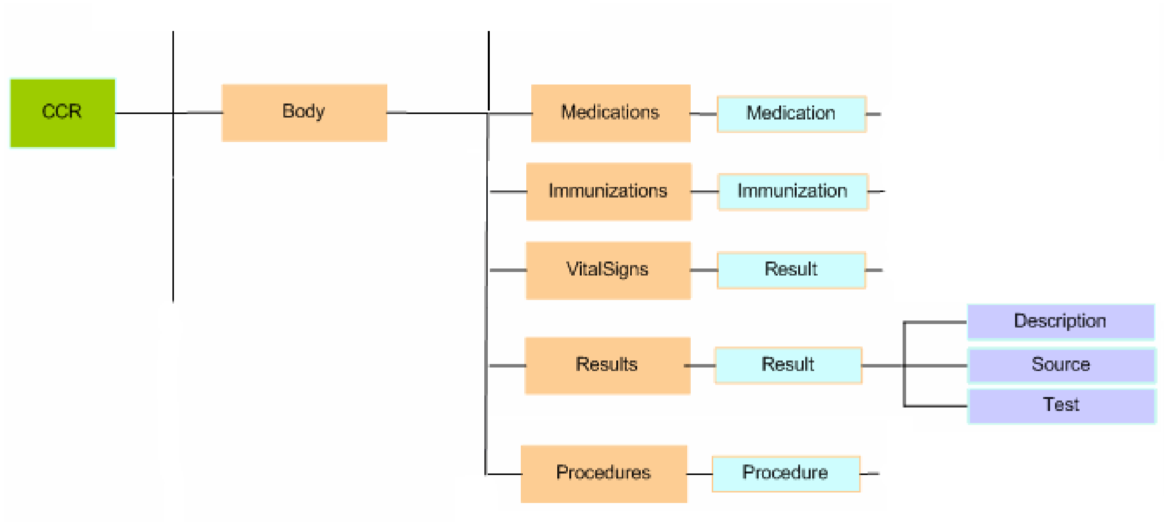

Figure 4.29: Tree diagram with the CCR elements supported by Google Health.

```
 <Result>
 <Test>
     <DateTime>
         <Type>
             <Text>Collection start date</Text>
         </Type>
         <ExactDateTime>2011-04-21T07:00:00Z</ExactDateTime>
     </DateTime>
     <Description>
         <Text>Weight</Text>
         <Code>
             <Value>363808001</Value>
             <CodingSystem>SNOMED</CodingSystem>
         </Code>
     </Description>
     <TestResult>
         <Value>72</Value>
         <Units>
             <Unit>kg</Unit>
         </Units>
     </TestResult>
 </Test>
</Result>
```

Two different screenshots were taken to Google Health platform, which are now presented here. They evidence a table with heartbeats and a chart of weight progress respectively in 4.30 and 4.31. The data was sent from the smartphone and successfully shared into Google Health platform.

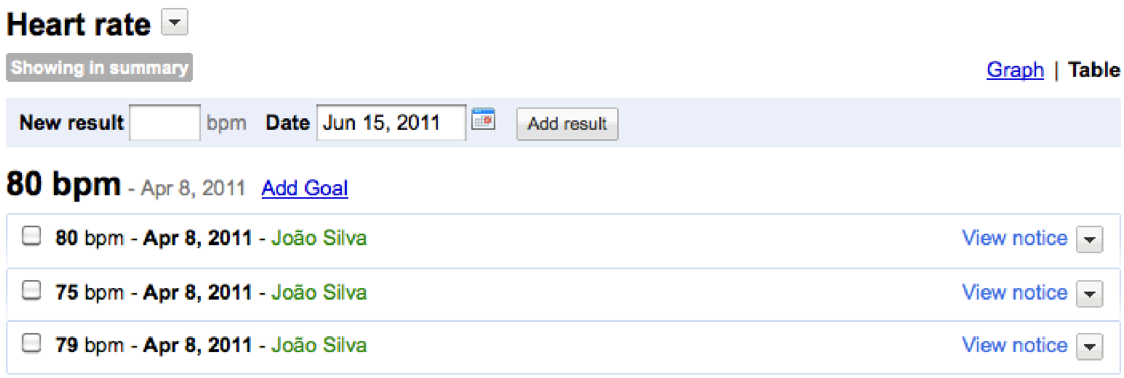

Figure 4.30: Google Health: Table with heart rates sent from the mobile.

Users can opt to share this information with others, who must also have Google accounts, and can stop doing it at any time. This is particular interesting for this project as it allows the users to share their medical information with their family doctor or whoever they want. This way, they will be continuously monitored by the healthcare professional, with regard to the major risk factors.

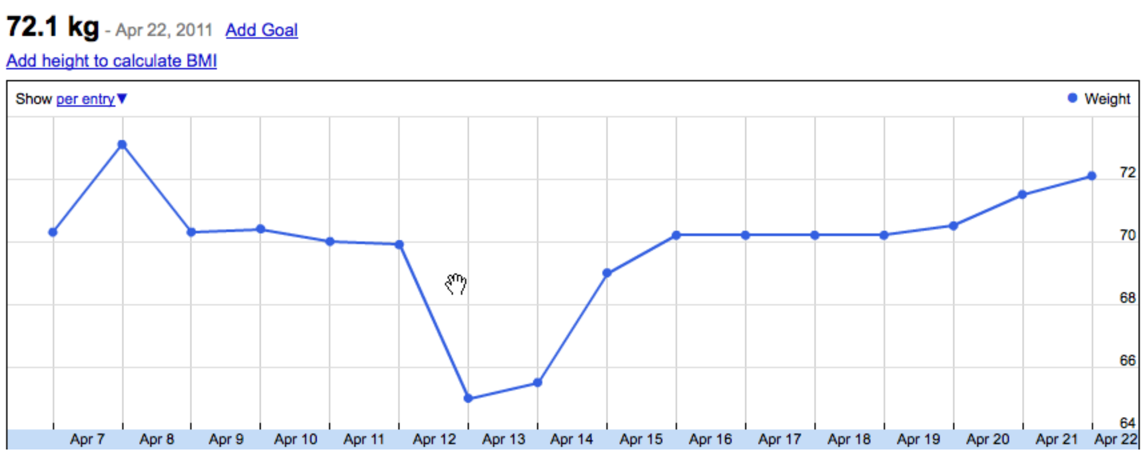

Figure 4.31: Weight progress tracking in Google Health.

Files

Google Health allows the import of health information and records from many health providers across the Web. A complete profile of a person's health may include more than just what is currently possible to import or enter from medical records. Thinking on that, Google Health allows to upload files from a computer into the Google Health profile. The details for the uploading are itemized below:

- Uploaded files are visible to the account owner when signed in, and to anyone else this person has shared the profile with;
- The maximum file size for upload is set to 4MB;
- Each person (considering one account) can upload a maximum of 100MB.

However, the health API does not support this feature yet . It was intended to create something like a report with more complete information about the results of the exam (including charts) and share it with to the healthcare professional, which was not possible for now to include it in the project due to this API limitation.

Chapter 5

Tests and Results

This Chapter presents some tests carried out and the results obtained. Software testing is very important and its impact on software cannot be underestimated. It is fundamental to ensure quality assurance and represents a review of specification, design and coding.

5.1 Algorithms Testing

The DSP algorithms were initially tested using a relatively small data set of samples collected over the Internet, which was composed by 25 WAVE files. This set included samples of normal heartbeat, atrial fibrillation, normal breath sounds and crackles in the lungs.

The quality of these sound files is good (few noise), and they offered us a way to have the algorithms reasonably tested before trying them in real life. The data set was continuously used to improve the algorithms robustness. They were tested one by one and the results were very satisfactory. Regarding the heart sounds, normal and healthy heart sounds were well differentiated from atrial fibrillation characteristic sounds. And in all them, the heartbeat rate was apparently well measured. About the lung sounds, the implemented algorithm was able to detect when there were present crackling murmurs and when there were not. Therefore, for this data set, we got 100% of success rate.

Later on, the same algorithms were tested using real samples captured by an Android device, using the stethoscope and our application for recording. The tests took place at Fraunhofer AICOS facilities, in a very silent place. It was asked the collaboration of 20 people. The testing only allowed to truly validate the heartbeat rate measurement, since it was not possible to confirm whereas the remaining results were correct or not - we did not have information regarding the examinees' medical status. For the heartbeat, it was asked another collaborator to place his index and ring fingers on the wrist of the examinee, at the base of the thumb, just over the radial artery which runs on the anterior part of the forearm. Each time the heart beats it is created a wave of pressure, as blood flows along the arteries. In areas where these arteries lie closest to the surface, this pressure can be felt and it is called pulse. This is a simple method that allows to measure the heart rate, without any special equipment and it is frequently used by doctors. During the recording the collaborator was responsible for counting the number of times he could feel the pulse. That number was then compared with the actual peaks number present on the time-domain representation of the signal, allowing us to know if the sounds were well recorded and enough clear after the filtering process. Finally the heart rate calculated by the algorithm was compared with these two values we have just talked, in order to measure the degree of reliability.

From the 20 tests, 90% (18/20) revealed positive results. The average beat rate was covered by the typical heartbeat range found in a normal resting pulse of an adult, which is from 60 to 100 beats per minute. Some sporadic errors were detected in the calculation of instantaneous beat rates, i.e, considering the elapsed time between two consecutive beats, and which are not caused by any unexpected behavior of in the algorithm. Rather, they are due to the poor quality of the stethoscope, that adds a lot of noise to the recording (sometimes even non-existent noise on the room, originated on the appliance itself). Occasionally this noise is graphically reflected in a peak very similar to the ones caused by a hear beat, both in frequency and amplitude, making it hard to be ignored.

Unfortunately it was not possible to get the algorithms medically validated. It is far more complicated than it might seem, there are a lot of ethical issues that must be combined and also some bureaucracy that needs to be overcame. Anyway, it is expected that in a near future this will become a reality.

5.2 Google Health Integration Testing

Google Health offers a developer's sandbox called H9, which was used to perform tests. H9 exists precisely for testing, so we do not need to worry about the quality of the data we send there, since we will be only "spamming" our, lets call, testing account instead of our real Google Health account. This one will be still "clean" and safe.

All that is needed to use the sandbox is to have a normal Google account, so we started by adding two existing ones to the Account Manager of the smartphone. Then, a simple stand-alone application was created with the intuit of learning how to send and get existing data on Google Health from an Android device. After having managed to do it, the required functionalities were implemented and then transferred to the main mobile application. For sending and retrieving information, the use of asynchronous threads was preferred, avoiding the block of the screen while the task is running. Four different tests were performed to verify the correct synchronization between the device and H9:

- Add a new test on the device;

- Delete a test on the device;
- Add a new test on H9;
- Delete a test on H9.

5.3 Graphical User Interface

Concerning the GUI design, no tests were conducted with elderly. However, Human-Computer Interaction Department from FhP has given some recommendations and guidelines on the topic which were followed targeting the continuous enhancement of the user interface, both in terms of its usability and visual appearance. Some of these tips are enumerated below.

- Provide good contrast between text and background;
- Use a consistent color coding;
- Place the main information in the middle of the screen;
- Sufficient lighting;
- Choose colors that prevent glare;
- Make certain that text has the size and dimensions in relation to reading distance and light;
- Use simple, plain words in an understandable language;
- Relegate to second plan tasks that the user does not need to see;
- Avoid popups abuse;
- Use of icons associated with the underlying text;
- Use of textual guidelines;
- Use of tutorial media.

5.4 Integration and System Testing

At the end, all components were integrated into a complete final system, and some tests were conducted on it to evaluate the system's compliance with the requirements specified in Section 3.2. The purpose of these integration tests were to detect any inconsistencies between the software components that were joined together or between any of the units and the hardware.

To conclude the testing, the application was used and tested several times, by different users, and everything showed up to be working properly. All functionalities of the system were tested and validated.

5.5 DSP benchmarking

Some benchmarking tests were performed in order to evaluate the Fourier Transform computation, which is highly computationally demanding and, thus, the most significant slice of the DSP.

Initially, it was used the *Apache Commons Math* (ACM) library, which is a complete and well maintained math library. With respect to the Fast Fourier Transform, this library was not satisfactory enough since the calculation speed is limited by implementation reasons, such as the use of objects to represent complex numbers.

Later on, when the project was well underway, some effort was spent trying to improve the computation speed. This was achieved with the multi-threaded JTransforms library, which is supposed to be nearly as fast as the C library FFTW. The necessary changes were made and the results were incredibly good.

Below, in Table 5.1, are shown the results obtained of the comparison of the two FFT libraries JTransforms and ACM. For each one, there were performed 500 runs. This was conducted on a Quad-Core laptop machine running Mac OS X 10.6.7. The arrays were filled with random data.

Table 5.1: Average execution time (in milliseconds) for 1D complex forward FFT (powers of two).

Input size	JTransforms (ms/run)	Apache CM (ms/run)	Factor by which JTransforms is faster
2^{16}	2,737	78,870	28,816
2^{17}	3,683	205,274	55,736
2^{18}	7,355	741,955	100,878
2^{19}	13,883	2871,046	206,803
2^{20}	29,932	Out of memory	—

ACM is well over an order of magnitude slower than JTransforms. The only advantage to using ACM for FFTs is that the rest of the library comes with it, reason why it was deprecated.

Chapter 6

Conclusions

This chapter presents some conclusions about this project and the environment in which it operates and assesses the potential areas of future development. The overall importance and contribution given with this project is handled first. Then, the continuous move to smaller, portable and low-power devices is discussed. Finally, different areas for future studies are suggested.

The continuous progresses in technology are delivering new portable systems every day. The devices tend to be smaller and lighter but they offer nearly the same computational capabilities as the non-portable ones. And even when they do not provide enough computational resources, they do provide wireless technology allowing to perform a specific task anywhere else rather than in the phone itself. The use of these devices in telemedicine is not something new, it has been used for a couple of years, although their computing power and size were very limiting. But nowadays, with the introduction of portable devices like smartphones and tablets, systems designers are able to deliver systems that would be unimaginable a short time ago. And the great thing is that these devices are moving towards to be wearable, to have the shape and weight of a necklace, for instance.

Mobile health is playing an important role in the future of health supervision and will continue to do it. A lot of effort concerning the individuals' health is currently being devoted to ensure that a wide range of issues are all done virtually over the phone. Specifically on this project, we sought to give a small contribution on that direction, making a mobile device able to auscultate the human's heart and lung functions and detect some important pathologies like atrial fibrillation and lungs crackling.

Now that the dissertation is finished, the overall balance of the work done is positive. The feeling is of accomplishment, we combined science with low-cost technology to develop a system that can be extremely important in prevention of heart failure and strokes, which are major causes of death in Portugal and in the world. This prevention will be possible by early diagnosis and will have greater relevance among the elderly, who are obviously more exposed to the risks.

With an Android smartphone and a low-cost stethoscope, people are able to track their heart and lung conditions without needing to go to the doctor. This device makes uses Wi-Fi connectivity in order to send the signal and process it remotely. The idea is that people will perform daily tests and that their doctor will be access the exam results. For now, this is done using Google Health, whose API is still a bit limited but we expect it to satisfy our needs in a near future. The idea of having every individual's medical data on a secured central data system is extremely interesting. By keeping a history of medical records and making them available from anywhere, we provide a major contribution to all healthcare professionals, including doctors, since we simplify their work, specially in countries or places where the doctor to patient ratio is astoundingly low. Being able to get the entire medical history of a specific patient on their phone by just hitting a simple tab on the screen of the device would be a huge step forward in health care, and it would surely be welcome by the professionals on this area. Besides saving a lot of time, it also gets rid of all the paper work that it still involves nowadays.

We have tested the system with small data sets and the results were quite positive. However, with respect to crackles and atrial fibrillation detection, it was not possible to test the system in real, since we need "sick" people apart from the healthy ones that are at our disposal. We were unable to have access to patients diagnosed with these pathologies, so it lacks of medical validation, that we hope it will be possible in the future.

6.1 State of the project

In the beginning, there was the hope that the work accomplished in this thesis could be a first step towards a general purpose m-health system, and we will now describe the progress made towards this goal.

Concluded the thesis, it is now possible to capture heart and lung sounds with an Android device and filter important information about the individual's health status, specifically regarding causes of Heart Failure, which is responsible for thousands of deaths every year. The developed system differentiates a normal heart functioning from a completely deregulated one, which characterizes Atrial Fibrillation. Also crackling murmurs can be detected by the device, when listening to the lungs. In addition, the heartbeat is measured and included in the exam information, along with the weight of the individual that must be introduced by himself. It also allows to graphically visualize the evolution of the weight for a period of time.

This information is saved in Google Health's cloud, allowing its share with other applications. It is possible for the healthcare professionals to access this information, after the patient grants him the necessary permission.

6.2 Future work

In this section, there are presented some suggestions for future research directions that could provide the next steps along the path to a practical and widely applicable system.

Future work related to this project inevitably involves investigating other pathologies that can be detected through the auscultation method, and which can have serious consequences for the individual's health, if not diagnosed in time. A number of open problems (regarding other pathologies) must be solved to allow the development of a truly general purpose prevention system. And such a diversity of problems that can affect the cardiopulmonary functioning implies a variety of research directions that need to be pursued to make such a system feasible. Depending on the motivation and effort engaged on it, we might be talking about a great piece of research that may last for several years.

Apart from the scientific component of the project, also the mobile application can be enriched and provided with other features. Particularly, it would be important to make it possible for the user to listen to the sounds/murmurs in real-time with reasonably good quality, so that it will be easier for him to know the best place to make the stethoscope touch the skin and thus get the best record for examination. Another good idea would be to change the way the individual's weight is introduced in the system. It would be far more interesting and practical to gather the information directly from the weighing scale, rather than asking the person to type it. To do so, we could use Bluetooth technology.

From the perspective of the development of a real-time collaborative system, it might be also challenging to add audio and video streaming capabilities, allowing the doctor to examine in real-time, from his home or office, his patient, who would have not to stand from his comfortable couch. And not few times, the doctors need more information that what they listen to can offer them. What the patient feels, what he has been doing, if is has been taking the medicines as prescribed or not, etc., is also very important. And so, this condition would be satisfied with such feature.

References

[app] Apple application submission feedback. http://appreview.tumblr.com/. Accessed on June 15, 2011.

[ASF08a] Dennis L. Kasper Stephen L. Hauser Dan L. Longo J. Larry Jameson Joseph Loscalzo Anthony S. Fauci, Eugene Braunwald, editor. *Harrison's Principles of Internal Medicine*, chapter 34. The McGraw-Hill Companies, 17 edition, 2008.

[ASF08b] Dennis L. Kasper Stephen L. Hauser Dan L. Longo J. Larry Jameson Joseph Loscalzo Anthony S. Fauci, Eugene Braunwald, editor. *Harrison's Principles of Internal Medicine*, chapter 13. The McGraw-Hill Companies, 17 edition, 2008.

[ASF08c] Dennis L. Kasper Stephen L. Hauser Dan L. Longo J. Larry Jameson Joseph Loscalzo Anthony S. Fauci, Eugene Braunwald, editor. *Harrison's Principles of Internal Medicine*, chapter 245. The McGraw-Hill Companies, 17 edition, 2008.

[ASF08d] Dennis L. Kasper Stephen L. Hauser Dan L. Longo J. Larry Jameson Joseph Loscalzo Anthony S. Fauci, Eugene Braunwald, editor. *Harrison's Principles of Internal Medicine*, chapter e8. The McGraw-Hill Companies, 17 edition, 2008.

[BDI09] National Heart Blood, Lung Institute Diseases, and Conditions Index. What is atrial fibrillation? http://www.nhlbi.nih.gov/health/dci/Diseases/af/af_what.html, 2009. Accessed on June 10, 2011.

[Ben] Peter J Bentley. istethoscope. http://www.peterjbentley.com/istethoscope.html. Accessed on June 10, 2011.

[BJ03] Gerstenblith G et al Braunstein JB, Anderson GF. Non-cardiac comorbidity increases preventable hospitalizations and mortality among Medicare beneficiaries with chronic heart failure. *Journal of the American College of Cardiology*, 42:1226–1233, 2003.

[BMA+10] Daniel Bonhorst, Miguel Mendes, Pedro Adragao, João de Sousa, João Primo Eva Leiria, and Pedro Rocha. Prevalência de fibrilhação auricular na população portuguesa com 40 ou mais anos. -estudo fama. *Revista portuguesa de cardiologia*, 29(3):331–350, 2010.

[Bri88] E. Brigham. *The Fast Fourier Transform and Its Applications*. Prentice Hall, Englewood Cliffs, 1988.

[Bur09] Ed Burnette. *Hello, Android.* Pragmatic Bookshelf, 2 edition, 2009.

[Cro10] Dr. Bill Crounse. Mobile devices usher in new era in healthcare delivery, 2010.

[CS06] Jones JK et al Curkendall SM, DeLuise J. Cardiovascular disease in patients with chronic obstructive pulmonary disease, Saskatchewan Canada cardiovascular disease in COPD patients. *Annals of Epidemiology*, 16:63–70, 2006.

[dA05] A. Teles de Araújo. Relatório do observatório nacional das doenças respiratórias. *Observatório Nacional das doenças respiratórias*, 2005.

[dA06] A. Teles de Araújo. Relatório do observatório nacional das doenças respiratórias. *Observatório Nacional das doenças respiratórias*, 2006.

[dA07] A. Teles de Araújo. Relatório do observatório nacional das doenças respiratórias. *Observatório Nacional das doenças respiratórias*, 2007.

[dA08] A. Teles de Araújo. Relatório do observatório nacional das doenças respiratórias. *Observatório Nacional das doenças respiratórias*, 2008.

[Erl04] Thomas Erl. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[FCM03] Gustavo Alonso Fabio Casati, Harumi Kuno and Vijay Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer Verlag, 2003.

[GOL06] GOLD. Global Initiative for Chronic Obstructive Lung Disease. Global strategy for the diagnosis, management and prevention of chronic obstructive pulmonary disease: NHLBI/WHO workshop report. 2006.

[HE02] Westfall KA et al Havranek EP, Masondi FA. Spectrum of heart failure in older patients: results from National Heart Failure project. *American Heart Journal*, 143:412–417, 2002.

[ins] Instant heart rate official website. `http://www.instantheartrate.com/`. Accessed on June 10, 2011.

[Ist04] R. Istepanian. Non-telephone healthcare: The role of 4g and emerging mobile systems for future m-health systems. *IOS PRESS*, 2004. Medical and Care Compunentics.

[Kes03] Walt Kester. *Mixed-Signal and Dsp Design Techniques*. Newnes, Oxford England, 2003.

[Mah05] Qusay H. Mahmoud. Service-oriented architecture (soa) and web services: The road to enterprise application integration (eai). `http://www.oracle.com/technetwork/articles/javase/soa-142870.html`, 2005. Accessed on January 21, 2011.

[MBG+06] Y. Miyasaka, M.E. Barnes, B.J. Gersh, S.S. Cha, K.R. Bailey, W.P. Abhayaratna, J.B. Seward, and T.S.M. Tsang. Secular trends in incidence of atrial fibrillation in olmsted county, minnesota, 1980 to 2000, and implications on the projections for future prevalence. *Circulation*, 114(2):119–25, 2006.

[Mei08] Reto Meier. *Professional Android Application Development*. Wrox Press Ltd. Birmingham, UK, UK, 2008.

[MP03] Nowak RM et al McCullough PA, Hollander JE. Uncovering heart failure in patients with a history of pulmonary disease: rationale for the early use of Btype natriuretic peptide in the emergency department. *Academic Emergency Medicine*, 10:198–204, 2003.

[New10] Midlands Business News. Healthcare conference spotlights importance of mobile technologies, 2010.

[NMJP09] Hawkins NM, Petrie MC, and et al Jhund PS. Heart failure and chronic obstructive pulmonary disease: diagnostic pitfalls and epidemiology. *European Journal of Heart Failure*, 11:130–139, 2009.

[oCoEoC] Networks of Centres of Excellence of Canada. Meeting global demand for mobile device applications. http://www.nce-rce.gc.ca/ReportsPublications-RapportsPublications/ExcelleNCENewsletter-BulletinExcellence/v1_i3/MobileDevice-AppareilsMobiles_eng.asp. Accessed on June 10, 2011.

[PL07] Valerie Gay Peter Leijdekkers. Personal heart monitoring and rehabilitation system using smart phones. Technical report, Faculty of IT, University of Technology Sydney, 2007.

[PM08] LeJemtel TH Padeletti M, Jelic S. Coexistent chronic obstructive pulmonary disease and heart failure in the elderly. *International Journal of Cardiology*, 125:298–215, 2008.

[PSRD10] Sergio L. Netto Paulo S. R. Diniz, Eduardo A. B. da Silva. *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2 edition, 2010.

[RC07] Arthur Ryman Sanjiva Weerawarana Roberto Chinnici, Jean-Jacques Moreau. Web services description language (wsdl) version 2.0 part 1: Core language. http://www.w3.org/TR/wsdl20/, 2007. Accessed on January 25, 2011.

[RHI06] Constantinos S. Pattichis Robert H. Istepanian, Swamy Laxminarayan. Remote monitoring for healthcare and for safety in extreme environments. Technical report, Centre for Telematics and Information Technology/Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente,, 2006.

[SM] Indra Pratap Singh Saroj Mishra. mhealth: A developing country perspective. Publication Date: August 9, 2008.

[Smi97a] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing 2nd Edition*, chapter 3. California Technical Publishing, 1997.

[Smi97b] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing 2nd Edition*, chapter 12. California Technical Publishing, 1997.

[Smi97c] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing 2nd Edition*, chapter 12. California Technical Publishing, 1997.

[SMW$^+$04] S Stewart, N Murphy, A Walker, A McGuire, and JJV McMurray. Cost of an emerging epidemic: an economic analysis of atrial fibrillation in the uk. *Heart*, 90:286–292, 2004.

[uni07] *Compendium of ICT Applications on Electronic Government*, volume 1: Mobile Applications on Health and Learning. United Nations, 2007.

[vdL95] Marten Derk van der Laan. Signal sampling techniques for data acquisition in process control. Master's thesis, Universiteitsdrukkerij Groningen, 1995.

[W3S] W3Schools. Web services tutorial. `http://www.w3schools.com/webservices/default.asp`. Accessed on January 20, 2011.