ESTUDIOS DE INGENIERÍA DE TELECOMUNICACIÓN

mDurance, a innovative method to accurately assess the trunk endurance tests

REALIZADO POR:
Ignacio Díaz Reyes

DIRIGIDO POR:
Miguel Damas Hermoso
Oresti Baños Legrán
Jose Antonio Moral Muñoz

DEPARTAMENTO:
Arquitectura y Tecnología de Computadores

Granada, Julio de 2015

MDURANCE, A INNOVATIVE METHOD TO ACCURATELY ASSESS THE TRUNK ENDURANCE TESTS

Ignacio Díaz Reyes

PALABRAS CLAVE:
Salud móvil; salud digital; condicionamiento físico; fisioterapia; rehabilitación; resistencia del tronco; sensores inerciales; sensores de electromiografía; dispositivos móviles; tests resistencia

RESUMEN:
El dolor lumbar es la enfermedad reumática más frecuente. Este trastorno constituye una de las causas más comunes de discapacidad a nivel mundial, y tiene como resultado un grave impacto socioeconómico. Los tests de resistencia se consideran normalmente en la práctica de la rehabilitación de los dolores lumbares para entender la causa muscular que provoca el dolor. Sin embargo, los procedimientos tradicionales para evaluar estos tests sufren limitaciones prácticas, que potencialmente conducen a inexactitud. El uso de tecnologías digitales se conciben aquí para facilitar la tarea de los expertos e incrementar la interpretabilidad de los test de resistencia. Este trabajo presenta mDurance, un nuevo sistema de salud móvil que pretende ser un apoyo para los especialistas en la evaluación funcional de la resistencia del tronco mediante el uso de dispositivos móviles y vestibles. Concretamente, se usa un sensor inercial vestible para seguir la postura del tronco del paciente, mientras que se emplea un sensor de electromiografía portátil para medir de forma continua la actividad eléctrica producida por los músculos del tronco. Esta información registrada por los sensores es procesada y dirigida por una aplicación móvil que facilita la rutina del experto, mientras reduce los errores humanos y acelera el análisis de los tests.

KEYWORDS:
Mobile health; digital health; physical conditioning; physical therapy; rehabilitation; trunk endurance; wearable inertial sensors; wearable electromyography sensors; mobile devices; trunk endurance tests

ABSTRACT:
Low back pain is the most prevalent musculoskeletal condition. This disorder constitutes one of the most common cause of disability worldwide, and as result, it has a severe socioeconomic impact. Endurance tests are normally considered in low back pain rehabilitation practice to understand the muscular cause provoking the pain. However, traditional procedures to asses theses test suffer from practical limitations, which potentially lead to misestimation and inaccurateness. The use of digital technologies is devised here to facilitate experts task and increase the reliability and interpretability of the endurance tests. This work presents mDurance, a novel mobile health system aimed at supporting specialists in the functional assessment of trunk endurance by using wearable and mobile devices. Concretely, a wearable inertial sensor is used to track the patient trunk posture, while portable electromiograph sensor is employed to seamlessly measure the electrical activity produced by the trunk muscles. This information registered by the sensors is processed and managed by a mobile application that facilitates the expert normal routine, while reducing human errors and accelerating the analysis of the tests.

D. Miguel Damas Hermoso, Profesor del departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada, como director del Proyecto Fin de Carrera de D. Ignacio Díaz Reyes.

Informa:

que el presente trabajo, titulado:

mDurance, a innovative method to accurately assess the trunk endurance tests

Ha sido realizado y redactado por el mencionado alumno bajo nuestra dirección, y con esta fecha autorizo a su presentación.

Granada, a 6 de julio de 2015

Fdo. Miguel Damas Hermoso

Los abajo firmantes autorizan a que la presente copia de Proyecto Fin de Carrera se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, a 6 de julio de 2015

Firmado: Miguel Damas Hermoso
Ignacio Díaz Reyes

ANEXO IV: INFORME DE VALORACIÓN DE UN PROYECTO

El tribunal constituido para la evaluación del proyecto PFC titulado:

mDurance, a innovative method to accurately assess the trunk endurance tests

Realizado por el alumno: Ignacio Díaz Reyes

Y dirigido por el tutor: Miguel Damas, Oresti Baños y Juan Antonio Moral

Ha resuelto asignarle la calificación de:

- ☐ SOBRESALIENTE (9 - 10 puntos)
- ☐ NOTABLE (7 – 8.9 puntos)
- ☐ APROBADO (5 – 6.9 puntos)
- ☐ SUSPENSO

Con la nota[1]: ☐ puntos.

El Presidente: _____

El Secretario: _____

El Vocal: _____

Granada, a 6 de julio de 2015

[1] Solamente con un decimal.

To my father, wherever you are and my mother that they have taught me everything.

Acknowledgements

I would like to acknowledge people that supported me during throughout the development of this project: Miguel Damas for encouraging me to take the project; Oresti Baños, who trusted me from the beginning of this project and its advices and work have made this possible; José Antonio Moral, for your great ideas and your support during all this time; Alejandro Sáez and Rafael García for your support and the patience to explain me their project; Cristian Rivera with whom i have enjoyed good moments of test assessments; my family because they always trusted and supported me in all moment; Marina Iza for their graphic designs and support; Antonio Porras for putting up with each other during long working hours; Soledad Ortíz for all the inspiration and everything that has helped me to learn and improve; all my friends for their inspiration; And finally my father, wherever you are, I know you would be proud. Thanks.

Contents

List of Figures **ix**

List of Tables **xiii**

1 Introduction **1**

1.1 Motivation and Context 1

1.2 Objectives 8

1.3 Structure 9

2 State of the Art **11**

2.1 Biomedical applications 11

2.2 Portable biomedical devices 14

2.3 Biomedical frameworks 16

2.4 Angular algorithms 19

3 Methodology **25**

3.1 Trunk endurance assessment 25

3.2 Electromyography (EMG) 28

3.2.1 EMG: definition and technical description 28

3.2.2 Muscle fatigue: analysis and description 30

4 Project Description **33**

4.1 Overview 33

4.2 Android 34

4.2.1 Operating System 34

4.2.2 Advantages/Drawbacks 36

4.2.2.1 Advantages 36

4.2.2.2 Drawbacks . . . 37
4.2.3 Why Android? . . . 38
4.3 Automatic Measurement of Trunk Posture . . . 38
4.4 Application description . . . 41
4.4.1 Specialist login . . . 43
4.4.2 Main Menu . . . 44
4.4.3 Endurance tests assessment with application . . . 45
4.4.3.1 Wearable sensors connection . . . 46
4.4.3.2 Tests execution . . . 49
4.4.3.3 Tests results . . . 54
4.4.4 Historical data consult . . . 56
4.5 App implementation . . . 57
4.5.1 mHealthDroid description . . . 58
4.5.2 Communication implementation . . . 59
4.5.2.1 How we use it in mDurance . . . 63
4.5.3 Computation algorithm of Euler angles . . . 64
4.5.3.1 How we use it in mDurance . . . 68
4.5.4 Data structure . . . 70
4.5.4.1 How we use it in mDurance . . . 70
4.5.5 Storage implementation . . . 72
4.5.5.1 How we use it in mDurance . . . 76
4.5.6 Visualization implementation . . . 77
4.5.6.1 How we use it in mDurance . . . 82

5 Evaluation **85**
5.1 Tests evaluation . . . 85
5.2 Expert suggestions . . . 87
5.3 Usability . . . 90

6 Conclusions **91**
6.1 Future work . . . 92

References **95**

List of Figures

1.1 Top 10 conditions globally with respect to YLDs in 2010 in comparison with 1990. Figure obtained from (1) 2
1.2 (**a**) Ranking top 10 causes of disability in 2010 by world region; (**b**) Global DALYs of LBP. Figures obtained from (1). 3
1.3 Evolution of LBP in 2010 function: (**a**) all ages; (**b**) range of 15-49 years. Figures obtained from (1) 4
1.4 Strategic diagram based on documents for the period 2000-2011: (**a**) Sub-period 2000-2005; (**b**) Sub-period 2006-2011. Figures obtained from (2). 5
1.5 Strategic diagram based on citations for the period 2000-2011: (**a**) Sub-period 2000-2005; (**b**) Sub-period 2006-2011. Figures obtained from (2). 6
1.6 Global market trends or wearable technology in function of US millions and units millions. Figure obtained from (3) 8

2.1 Interest over time in "Low Back Pain" (blue chart) and "Medical App" (orange chart) terms. Results obtained through *Google Trends*. The values, expressed in percentage, reflect the amount of searches that have been done for each term, relative to the total number of searches done on Google over time. 12
2.2 MIO FUSE HEART RATE ACTIVITY TRACKER (4) 16
2.3 mHealthDroid structure. Figure obtained from (5). 17
2.4 Google fit platform overview. Figure obtained from (6). 19
2.5 Inertial sensor of Shimmers (7) 20
2.6 (**a**) Diagram of the Kalman filter sensor fusion approach (8), (**b**) Diagram of the LMS/RLS filter sensor fusion approach. (8) 23

LIST OF FIGURES

3.1 Functional trunk endurance tests: (**a**) STEET; (**b**) TCSET; and (**c**) SBET. 28
3.2 Intramuscular fine wire electrode (9). 29

4.1 mDurance system is based in wearable sensors plus an application mobile. 34
4.2 Android Architecture Diagram 35
4.3 Mobile/Tablet Top Operating System Share Trend (10). 38
4.4 Madgwick's algorithm perfomance (11). 40
4.5 Sensor deployment for (**a**) STEET, (**b**) TCSET and (**c**) SBET procedures. 42
4.6 Screens: (**a**) Specialist Sign Up; (**b**) Specialist Login with a registered expert in the system. 43
4.7 Specialist main menu screen with a selected patient 44
4.8 Screens: (**a**) Registered patient list in the system) Specialist Login; (**b** New patient register. 45
4.9 Colors of the status circles: Red (*Disconnected*), Green (*Connected*) and Orange (*Streaming*) 46
4.10 Bluetooth interface to pair the devices with the app. 47
4.11 (**a**) Paired EMG device; (**b**) Paired IMU device. 48
4.12 Colors of the power button: Red (*Disconnected*), Green (*Connected*) and Blue (*Streaming*) 49
4.13 Final connection status to start a test 49
4.14 Tests selection screen. 50
4.15 Guide about how to place the sensors in STEET test: (**a**) Main screen; (**b**) IMU place; (**c**) EMG place; and (**d**) Electrodes place. 51
4.16 Test activity screens: (**a**) Initial scenario before to start a test; (**b**) Final scenario when the test has finished 52
4.17 Graph configuration menu 53
4.18 Tests Results Screen 55
4.19 Historical data menu 56
4.20 Example of a historical query 57
4.21 mDurance architecture building on mHealthDroid. 59
4.22 Data flow diagram from the devices to the framework 61
4.23 Class diagram of the framework data structure 71
4.24 Table of data specialists from the internal database: *UserDB*. 73

4.25 Table of patient's data from the internal database: *UserDB*: (**a**) Personal information; (**b**) STEET results; (**c**) TCSET results; and (**d**) SBET Right results 74
4.26 Table of data sensors from the internal database: *UserDB*. 75
4.27 Passing message from the Communication Manager to the different graphs 81

5.1 Agreement analysis between traditional trunk endurance assessment and mDurance through Bland-Altman plots: (**a**) STEET; (**b**) TCSET; (**c**) SBET Right; and (**d**) SBET Left. The mean of differences ($\bar{x}$) is represented by a blue line, while the limits of agreement ($\bar{x} \pm 1.96\sigma_X$) are depicted in red. 89
5.2 System Usability Scale (SUS) scores obtained from seven experts after using mDurance. 90

LIST OF FIGURES

List of Tables

3.1 Trunk endurance tests description. 27

4.1 Statistical overall duration of the patient and execution ratios. 55

5.1 Case study results. BMI values are expressed in kg/m^2 and test duration in s. 87

5.2 Inter-rater reliability between traditional trunk endurance assessment and mDurance. 88

LIST OF TABLES

1

Introduction

1.1 Motivation and Context

Low back pain (LBP), is considered an extremely common health problem, and the major cause of activity limitation and work absence in the world. LBP is ahead of 290 other conditions and causing an enormous economic burden on individuals, families, communities, industry and governments. This condition increased from 58.2 million (95% uncertainly intervals: 39.9-78.1 million) in 1990 to 83.0 million (95% UI: 56.6-111.9 million) in 2010. Furthermore, LBP is ranked sixth in terms of overall burden of disability-adjusted life years (DALYs). DALYs is defined as the sum of two components: years of life lost due to premature mortality (YLLs) and years lived with disability (YLDs). DALYs is an absolute measure of health loss; they count how many years of healthy life are lost due to death and non-fatal illness or impairment. They reflect the number of individuals who are ill or die in each age-sex group and location. Population size and composition influences the number of DALYs in a population.

To help in the rapid dissemination of results, we are going to use a visualization tool developed by the Institute for Health Metrics and Evaluation (IHME) (1), where it is possible to consult all kind of statistics about LBP and other conditions. For example, Figure 1.1 displays the rankings for the top 23 conditions globally with respect to YLDs in 2010 in comparison with 1990.

In addition, in the Figure 1.2(a) we also can compares the top 10 causes of disability in 2010 per world region and per developed or developing countries. Low back pain is the leading cause of disability in both developed and developing countries and ranks

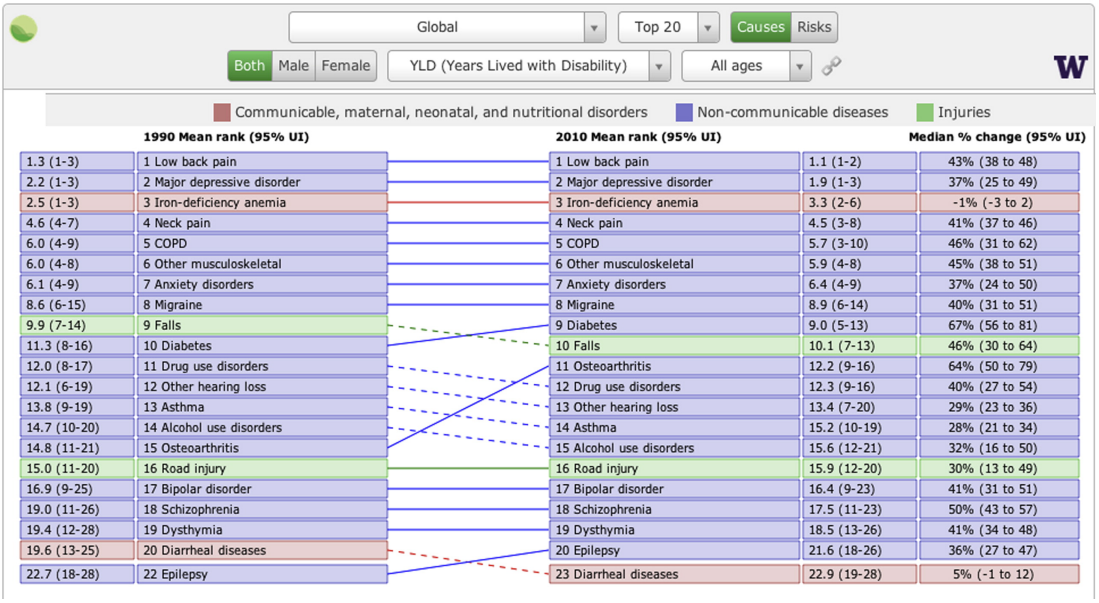


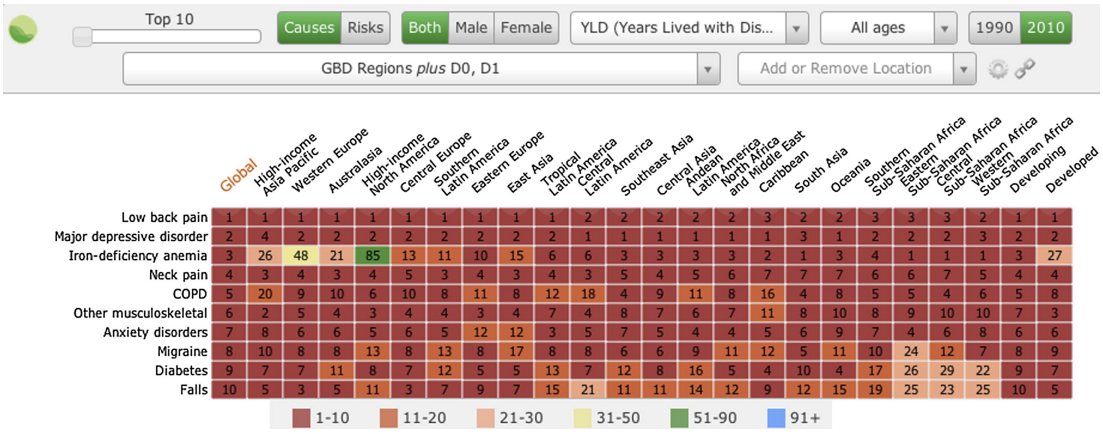
Figure 1.1: Top 10 conditions globally with respect to YLDs in 2010 in comparison with 1990. Figure obtained from (1)

among the top three causes in all 21 regions of the world. With respect to regional statistics, it is also important to display the number of DALYs that LBP causes by world region, such as it is shown in Figure 1.2(b). In this case, North and Central Europe and North American remain the leaders in the same way with respect to the YLDs.

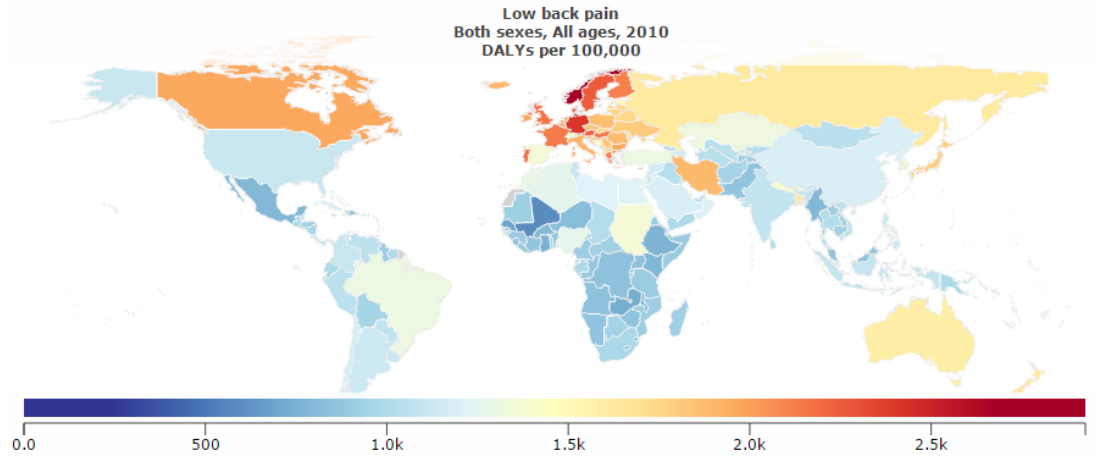
Figure 1.3(a) displays the evolution of LBP in 2010 function of people aged in both sexes. It can be demonstrated as the peak of DALYs is found in the range from 40 to 44 years old. In spite of this statistics, the most worried may be the increase of young that suffer LBP since 1990 to 2010 such as Figure 1.3(b) shows us.

A recent study (12) shows that in 2012, LBP reached a point prevalence of 11.9%, one-month prevalence of 23.2% , one-year prevalence was of 38% and the lifetime prevalence was 39.9%. In chronic LBP has been estimated that the average age-related prevalence is 15% in adults and 27% in the elderly. Furthermore, LBP had the highest prevalence among women aged between 40 and 80 years. LBP has an enormous social and economic impact (13), and is a leading cause of absenteeism in all professions (14).

It is noteworthy that the growing interest of the scientific community in the study of LBP (2). If we search in the ISI Web of Science(ISIWoS), one of most important



(a)

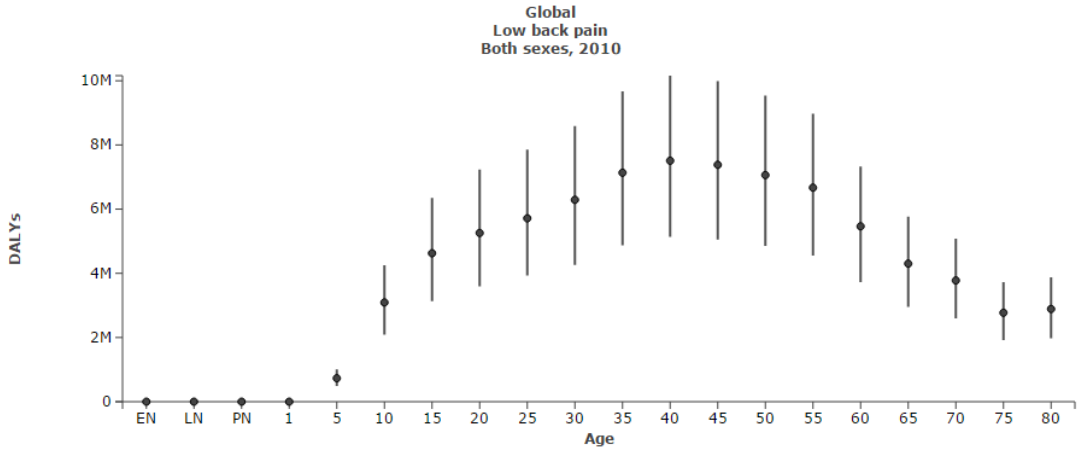


(b)

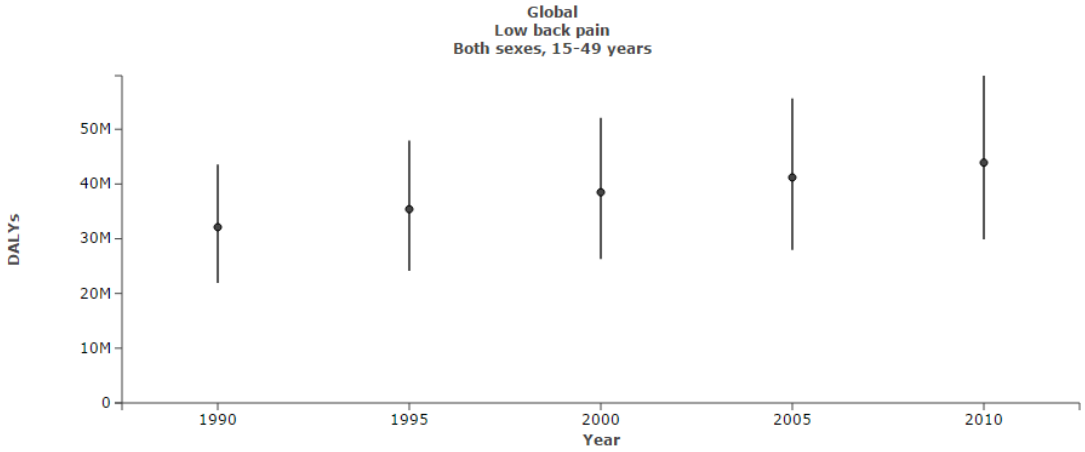
Figure 1.2: (**a**) Ranking top 10 causes of disability in 2010 by world region; (**b**) Global DALYs of LBP. Figures obtained from (1).

bibliographic databases, about the main research works of LBP, we find that from 2000 to 2011, the theme LBP is focused on the characteristics of low back pain and its treatment with therapies. This is showed in Figure 1.4. It also obtains an important citation rate, such as is referenced in Figure 1.5.

Although general exercises for the whole body have been beneficial for the patient with chronic LBP, increasing emphasis has been placed on giving more specifically directed exercises for the spinal muscles in addition to the general exercise programs (15). Due to pathopsychology, LBP is associated to a wrong lumbar-pelvic stability. In order



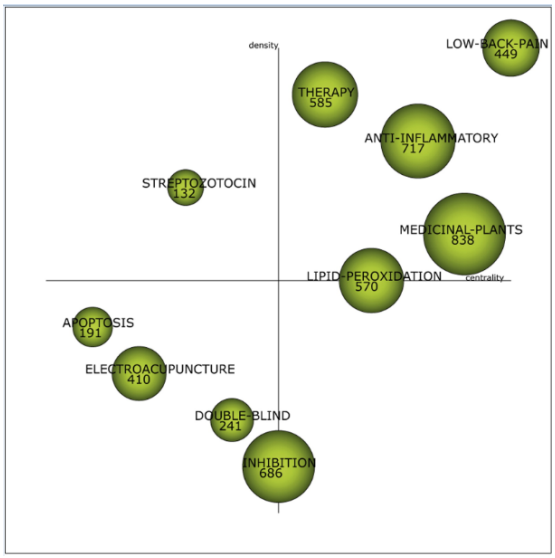
(a)



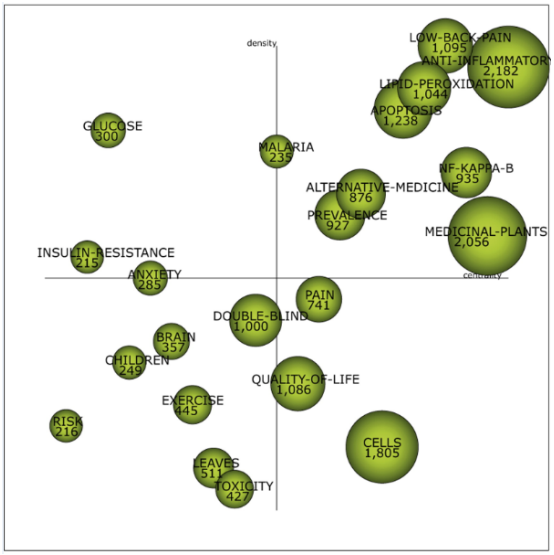
(b)

Figure 1.3: Evolution of LBP in 2010 function: (**a**) all ages; (**b**) range of 15-49 years. Figures obtained from (1)

of to be more effective and efficient, these specific exercises were developed to target the muscles that are associated with lumbar-pelvic stability. General exercises for the whole body and encouragement of the individual to stay active have been shown to be beneficial for preventing and dealing with chronic LBP (16). Thus, a functional assessment of the trunk stabilization or endurance turns to be utterly necessary to establish goals, monitor progress toward those goals and guide the prescription of specific exercises in order to prevent future injuries and improve the people conditions (17). Trunk muscle endurance assessment, normally referred as to trunk endurance assessment, con-



(a)



(b)

Figure 1.4: Strategic diagram based on documents for the period 2000-2011: (**a**) Sub-period 2000-2005; (**b**) Sub-period 2006-2011. Figures obtained from (2).

sists in the evaluation of the muscular capacity of an individuals trunk. To determine the resistance of the trunk muscles, experts traditionally measure and annotate the observed time a patient can hold a given posture part of a test. Nevertheless, this form of evaluation is subject to potential errors, mainly posed by the subjectivity associated

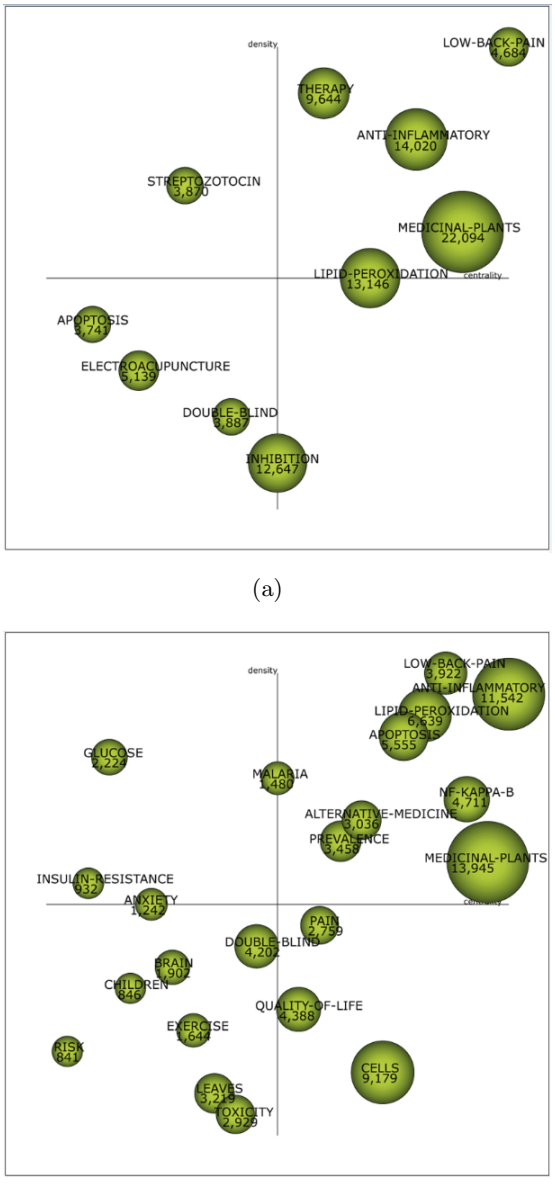


Figure 1.5: Strategic diagram based on citations for the period 2000-2011: (**a**) Sub-period 2000-2005; (**b**) Sub-period 2006-2011. Figures obtained from (2).

to the estimation of the test finalization and the effective measurement of the time elapsed during its execution(18).

Within this context of clinical needs is emerging the digitization of medical procedures. During the last years, the use of devices and software in healthcare disciplines

has become more common due to the constant technological improvement (14, 19, 20). Digital technologies can serve to cope with some of the limitations introduced by human errors during the practice of medical procedures. There are different factors attributable to the development of this type of systems:

- The globalization of health systems (21). A number of major actors involved in global health currently are engaged in health systems strengthening, including the World Health Organization, the World Bank, the G8, and several global health initiatives and private foundations. Many have come to pay attention to the issue only in the past few years.

- The demand by healthcare users for novel forms of treatment and prevention. (22). Prehabilitation is a process that will help prepare to people by improving their functional capacity before encountering the event. This concept has been employed by athletes in an effort to reduce the risk of injuries and help accelerate their rebound from any injuries.

- The need of reduction of healthcare cost. Consistently, as many as 20% of patients who participate in shared decision making choose less invasive surgical options and more conservative treatment than do patients who are not consulted (23).

- The major advances in information and communication technologies (24).

Telehealth, eHealth, Social Health, and Health IT are some of the most prominent areas in which telecommunications and computer technologies are combined to expedite and enhance healthcare procedures. Currently, at the forefront of the digital health revolution is the so-called mobile health (mHealth) (25),which refers to the practice of medicine and public health supported by mobile devices and applications. mHealth technologies offer real-time monitoring and detection of changes in health status, support the adoption and maintenance of a healthy lifestyle, provide rapid diagnosis of health conditions, and facilitate the implementation of interventions ranging from promoting patient self-care to providing remote healthcare services. The interest in this domain has been boomed by the growth of wearables and mobile technologies(26). In regard the global wearable technology market, in Figure 1.6 is displayed the continuous growth in each one the application fields of wearable devices since 2011. Starting at less than 120 millions units in 2013, the total wearables market worlwide is expected to top

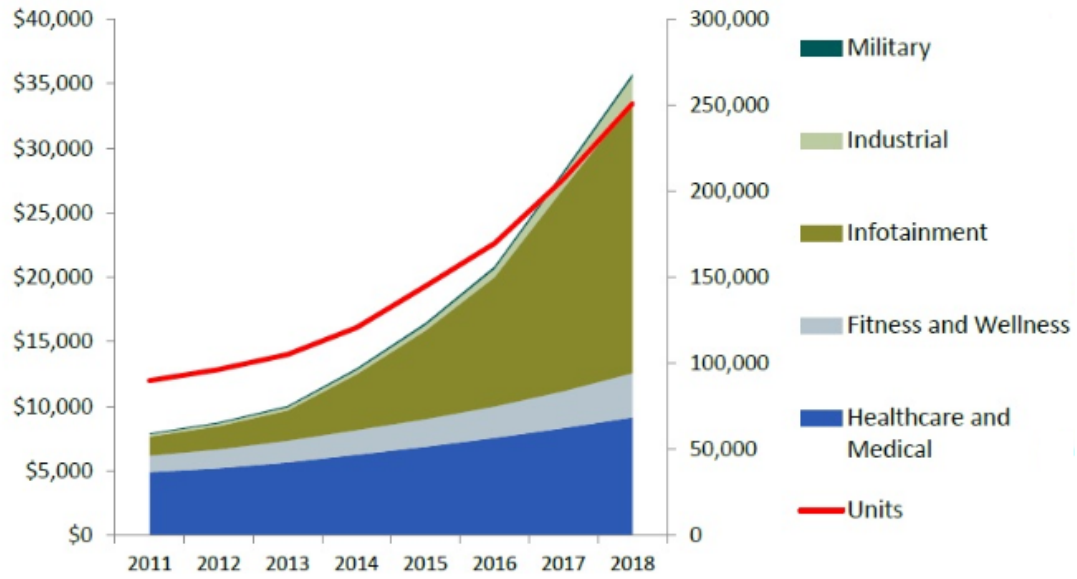


Figure 1.6: Global market trends or wearable technology in function of US millions and units millions. Figure obtained from (3)

230 millions unit by 2019, nearly doubling in volume withun the six-year period. And unlike, some other markets, revenues in this segment are forecast to grow even faster than unit shipments, more than tripling in value to over 32$ billion by 2019, up from 10$ billion in 2013. It should be pointed that North America and Europe contribute to 70% of global revenue (3). Smartphone adoption is also other important point to treat in this context. In 2016, over 1.2 billion smartphones are forecast to be shipped worlwide as the smartphone market experiences rapid growth. Therefore, as the number of smartphone users grows and the smartphone becomes the hub of information for its users, there will be an increasing number of devices that will connect to and exchange data with smartphones (3).

1.2 Objectives

In the light of present challenges of physical rehabilitation and conditioning routines as well as the potential of mHealth technologies, this work presents mDurance, a mobile health system intended to support experts in the functional assessment of trunk endurance by using wearable and mobile devices. The system has been defined to over-

come some of the most relevant limitations faced by specialists during the course of endurance tests, such as:

- The determination of the patients initial posture.
- The estimation of the duration of the tests.
- The measurement of the muscle fatigue (both in relation to time and in relation to electromiography values).
- The final analysis of the results through online results when each test is finished and through offline results save in an internal database.

The mDurance system leverages the use of wearable inertial sensors to track the patient trunk posture, and portable electromiography sensors to seamlessly measure the electrical activity produced by the trunk muscles. All the information registered through these sensors is intelligently managed by a mobile application that facilitates the expert normal routine, helps mitigate human errors and accelerates the analysis of the tests.

1.3 Structure

This document is divided in six chapters:

- **Chapter 1. Introduction**. It is the current chapter which is composed by four sections. The first section titled *Motivation and context* 1.1 describes the importance and need that urges in modern world to develop biomedical applications, that is the subject of this thesis. The second section, *Objectives* 1.2, contains an overall summary of the proposal objectives at the beginning of the thesis. Finally, the last section is *Thesis and Structure* 1.3 and presents the structure and composition of the thesis.
- **Chapter 2. State of the Art**. This chapter presents a review of four fields related to biomedical engineering: Biomedical applications 2.1, portable biomedical devices 2.2, biomedical frameworks 2.3 and angular algorithms 2.4.

- **Chapter 3. Methodology**. This chapter, composed by three sections: Trunk endurance assessment 3.1, where we describe the different tests that we use in our system; Electromyography (EMG) 3.2, which it consists in a explication and a description about the different EMG procedures 3.2.1 and how we assess the muscle fatigue 3.2.2.

- **Chapter 4. Project description**. In this chapter, we present a general overview about mDurance 4.1; The OS that supports to mDurance in 4.2; How mDurance means the trunk posture 4.3; A deep description of the functionality of the application to help the specialist to assess the different tests proposed in mDurance is explained 4.4; Finally, it is described how mDurance has been implemented 4.5.

- **Chapter 5. Evaluation**. This chapter contains the results of the evaluation realized to check the reliability of mDurance, the expert suggestions 5.2 and the usability of the system from the viewpoint of these experts 5.3.

- **Chapter 6. Conclusion and future work**. This chapter contains the conclusions reached in this thesis and proposal of future work 6.1.

2

State of the Art

In this chapter we describe the present and future investigation lines about biomedical applications Section 2.1, portable biomedical devices Section 2.2, biomedical frameworks Section 2.3 and angular algorithms Section 2.4.

2.1 Biomedical applications

The applications of the biomedical engineering have a range so wide that it could be the topic of a thesis. Therefore this is just a description of the most relevant aspects. When we talk about biomedical engineering, we refer to that field of study which mixes medicine with computer science, electronics and other similar fields. The main goal of this subject is to develop the necessary technology for improving and making easier everything related to medicine. In this moment, we are going to start to talk about mHealth apps. The number of apps has increased importantly, and more than 40000 fitness, health and medical apps currently are available on the market (26). According to a survey performed in UK (27), the use of medical apps is of 72.4% among doctors, and as high as 83.3% among medical students. The majority of both students and doctors owned from 1 to 5 apps, which they used on a regular basis. Moreover, this study highlights that the most frequently used apps are devoted to detail medication references as well as disease diagnosis and management. In relation to these findings, it is fairly justified the continuous development of medical apps focused on clinical aspects. In fact, the high level of smartphone ownership and the more intuitive and user-friendly applications are compelling reasons suggesting that medical apps will offer

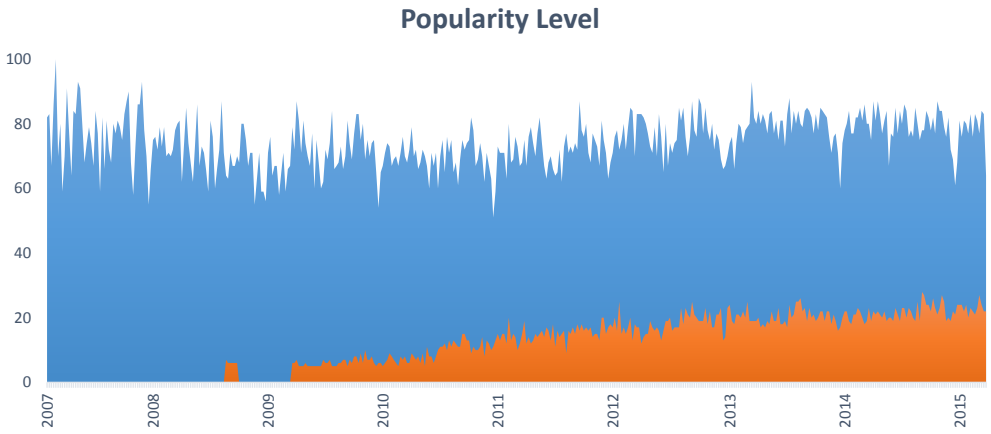


Figure 2.1: Interest over time in "Low Back Pain" (blue chart) and "Medical App" (orange chart) terms. Results obtained through *Google Trends*. The values, expressed in percentage, reflect the amount of searches that have been done for each term, relative to the total number of searches done on Google over time.

a real opportunity to impact on the efficiency of working practices and patient care. The market of medical applications is primarily led by Apple's iOS platform (28); however, its use is tailored to a reduced and expensive catalog of devices. Alternatively, Android provides its users with a wider variety of systems of different prices and vendors at the reach of a broader audience, which is increasing its competitiveness in this domain (29, 30). In our society, the utilization of the Internet to seek medical information has unarguably increased during the recent years. The analysis of the searches done over the Internet helps better understand the interest of people in medical tools and illnesses. Concretely, Figure 2.1 depicts the worldwide trends with respect to the search of "Low Back Pain" and "Medical App" concepts. LBP shows a sustained popularity in people searches over the last seven years, which might be related to the high prevalence of this disease and the necessity of information regarding symptoms and potential treatments for this condition. With respect to medical applications, it is clear from the trends that people are growingly interested in this sort of technology. It should be taken into account that these trends only refer to searches in English; thus, if other languages are considered, the popularity level could likely increase.

Having a look at the main application catalogs, i.e., Google Play and Apple Store, several apps can be found in relation to LBP. The vast majority of apps are planned to promote exercises to prevent or relief LBP. Also, apps with informative or academic

purposes and others focused on diagnosis are available. The number of apps to help alleviate LBP symptoms is especially elevated. Some examples are:

- *Stretch Away* (31).
- *Back Doctor* (32).
- *iREHAB* (33).
- *Prevent Back Pain* (34).
- *Yoga for Back Pain Relief* (35).
- *WebMD Pain Coach* (36).
- *Upper & Lower Back Pain Relief* (37).

The operation of these applications is mainly oriented to provide trunk exercise recommendations. They fundamentally consist of a database of image or video exercises, which are used to guide the patient or person suffering from LBP on how to execute them. This category of apps is available for any sort of users, and normally, they do not take into account the potential diseases that may lead to LBP.

The group of apps focused on providing patient or professional-oriented LBP information is also considerable. Some examples within this domain are:

- *Back Pain Guide* (38).
- *Back Pain Complete Guide* (39).
- *Back Pain: An Algorithmic Approach to Low Back Pain* (40).
- *Back Pain Causes And Cures* (41).
- *Back Pain Nerve Chart* (42).

This group of apps only offer information regarding the essentials of LBP, including causes, treatments or even descriptions of the back anatomy.

Lastly, the group of apps dedicated to support diagnosis of habits or postures that can lead to LBP constitutes the less relevant at the moment in the marketplace. Some examples related to this group are:

- *PostureScreen Mobile* (43).
- *Clinical Pattern Recognition: Low Back Pain* (44).
- *Virtual Diagnosis Spine* (45).

The main purpose of these apps is to recognize LBP through requesting the users to provide information related to different LBP symptoms. Some of them also help customers identify different posture alterations.

A comprehensive search has been performed to find specific applications and systems to evaluate trunk endurance using traditional tests. However, no relevant results have been obtained. In view of the search result, it seems that there is a clear opportunity for the development of applications and systems that may help specialists perform trunk endurance assessment.

2.2 Portable biomedical devices

The design and implementation of wearable biomedical systems for health monitoring has gotten lots of attention throughout the world, specially in the industry and scientific community during the last years. To address this demand, a variety of systems and commercial products have been produced in the course of recent years, they propelled by all the new technological advances like smart textiles, microelectronics, miniature biosensing devices and wireless communications. The continuous advance of wearable sensor-based systems will potentially transform the future of healthcare by enabling proactive personal health management and ubiquitous monitoring of a patients health condition.These systems can comprise various types of small physiological sensors, transmission modules and processing capabilities, and can thus facilitate low-cost wearable unobtrusive solutions for continuous all-day and any-place health, mental and activity status monitoring (46). These biosensors are capable of measuring significant physiological parameters like heart rate, muscular fatigue, posture analysis, blood pressure, body and skin temperature, oxygen saturation, respiration rate, electrocardiogram, etc. Some examples of devices than can display these parameters are:

- *Shimmer.* This device, although released in 2008, was originally developed by Intel in 2006, but nowadays its distribution and development is in charge of Shimmer Researcher which is a division of the company Realtime Technologies. The device is composed by a mother board in which it is attached to the different available sensors (47).

- *Mio Fuse.* This wrist's monitor provides the heart rate with the same accuracy that an ECG sensor with chest's bands (4).

- *Lumo Lift.* It is a small device which has two different features to detect and improve your posture: Posture Alert and Coaching Sessions. The main difference between the two features is the time it takes to receive vibratory feedback on your posture. Posture Alert will prompt you after a set period of time, and Coaching Sessions will give you immediate feedback on your posture. Coaching session is a popular feature of the Lumo Lift that alerts you when youve been slouching for a set period of time (48).

- *DataLOG MWX8 and DataLINK.* They are overall, a system for data acquisition technology developed to meet the needs of researchers for portable data collection and monitoring in human performance, sports science, medical research, industrial ergonomics, gait laboratories, and educational settings. DataLINK is an on-line general purpose subject worn programmable Data Acquisition System allowing the user to collect both analog and digital data from a wide range of sensors including Biometrics' Goniometers, Torsiometers, active EMG sensors, accelerometers, pinchmeters, hand dynamometers and Contact Switches (49).

- *Genesis.* This device is connected to a smartphone and sends the data via Bluetooth. The device manages and regulates the level of the glucose like it is an artificial pancreas (50).

- *Asthmapolis sensor.* It is treat of a device which is placed above the asthmatics inhaler and every time it is used, the device sends the hour and the location in order to know the environment where the asthmatic uses it (51).

- *IHealth blood pressure monitor.* It is a wrist blood monitor. Overall with the user-friendly mobile app, simply they follow the on-screen guide to take a measurement

Figure 2.2: MIO FUSE HEART RATE ACTIVITY TRACKER (4)

or compare the previous systolic, diastolic and pulse rate numbers from your smartphone or tablet (52).

2.3 Biomedical frameworks

According to (5, 53), powerful frameworks and tools that support the design, development and validation of multidisciplinary mHealth applications are required. The main goal of mHealth frameworks is to foster the research and development in health and medical domains as well as to accelerate the market of mobile health technologies and applications. Furthermore, mHealth makes it feasible for patients to collect and share relevant data at any time, not just when they happen to visit a clinic, allowing more rapid convergence to optimal treatment. An important feature for a framework is to have an open architecture, components, well defined, published interfaces that allow interconnection and use in ways other than as originally implemented or intended. A certain level of abstraction from heterogeneous resources should be ensured to make hardware and its communication transparent to the developer. For the sake of interoperability, a framework should define a unified model for multimodal health data. Secure

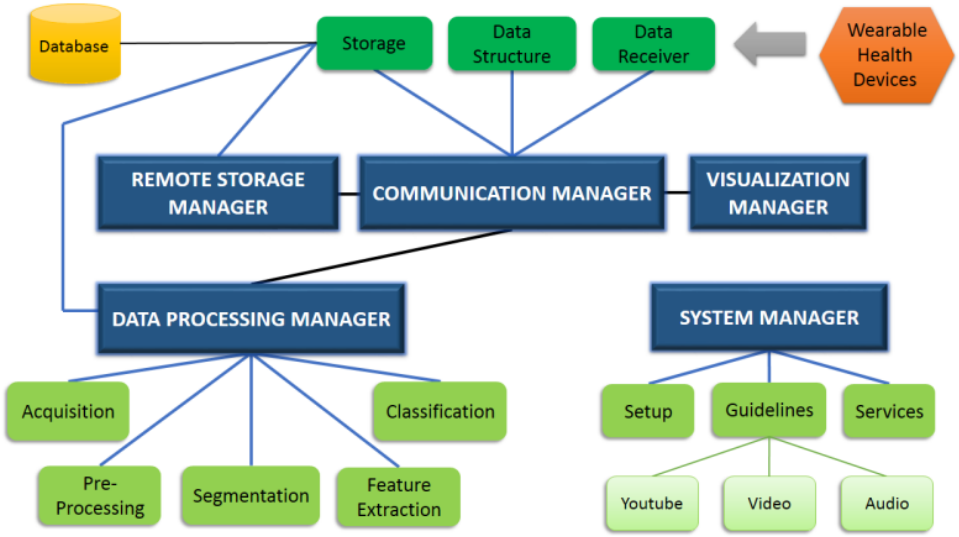


Figure 2.3: mHealthDroid structure. Figure obtained from (5).

local and remote storage of health data is required to ensure persistence. A framework should provide techniques to extract health knowledge from raw medical and physiological data. Mechanisms to visualize medical and health information in a user-friendly fashion must be also provided for both average users and specialists. Another major requirement refers to the provision of healthcare services such as health delivery, personalized guidelines and intelligent recommendations. Other aspect to be considered is that a framework should be modular and extensible to future sensor technologies and application needs. These are some examples of frameworks:

- *mHealthDroid.* It is a novel mHealth Framework for the development of safe, scalable and effective applications. It provides resource and communication abstraction, biomedical data acquisition, health knowledge extraction, persistent data storage, adaptive visualization, system management and value-added services (5). mDurance has been implemented using this framework, although we talk after of this subject. We can observe an overview about its working in Figure 2.3.

- *ECG frameworks.* There are a big variety of software which is based in ECG systems. For example, (54) is an application aimed at screening and recording the ECG waveforms by connecting the phone with an ECG recording device via

Bluetooth. The signal processing is then performed on the phone and the user is provided with essential feedback. Another system could be (55), it is a framework which using the Android system and physiological measurement modules, an interface-independent uplink and downlink packet communication protocol is established to monitor vital signs, such as ECG, blood pressure, oxygen saturation, respiration, pulse, and body temperature, while providing applications that integrate graphic user interface (GUI) monitoring and display collected data. In the end, (56) is a wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing.

- *PhysioDroid.* It combines both commercial wearable health devices and mobile devices to provide continuous and personalized patient physiological monitoring. An application that runs on the Android platform is implemented to enable the collection, sharing, and exchange of physiological data registered through the ubiquitous wireless health monitor. Other features such as health data visualization, storage, and alerts on conditions are further available from this (57).

- *Mining Minds.* This is an innovative framework that builds on the core concepts of the digital health and wellness paradigm to enable the provision of personalized healthcare and wellness support. Mining Minds is further devised to intelligently exploit digital health and wellness data to generate new businesses and services, which are unquestionably called upon to change the actual healthcare and wellness panorama (58).

Fitness platforms collect fitness data from various sources, and the end users can share the data with apps of their choosing. Polar (59) is integrating into two main fitness platforms: *Google Fit* and *Apple HealthKit*. With the users permission, developers can easily get access to the users basic training data from Flow compatible products. *Google Fit* and *Apple HealthKit* fitness platforms are optimized solutions for application developers wanting to integrate training data into their application. We are going to explain a little more about these 2 plattforms:

- *Google Fit.* According to (6), Google Fit is an open ecosystem that allows developers to upload fitness data to a central repository where users can access their data from different devices and apps in one location:

- Fitness apps can store data from any wearable or sensor.
- Fitness apps can access data created by any app.
- User's fitness data is persisted when they upgrade their fitness devices.

Its working can be observed in the Figure 2.4.

- *Apple HealthKit.* According to (60), HealthKit allows apps that provide health and fitness services to share their data with the new Health app and with each other. A users health information is stored in a centralized and secure location and the user decides which data should be shared with your app. In this case, HealthKit allows to build hardware accessories that work with health and fitness apps through HealthKit.

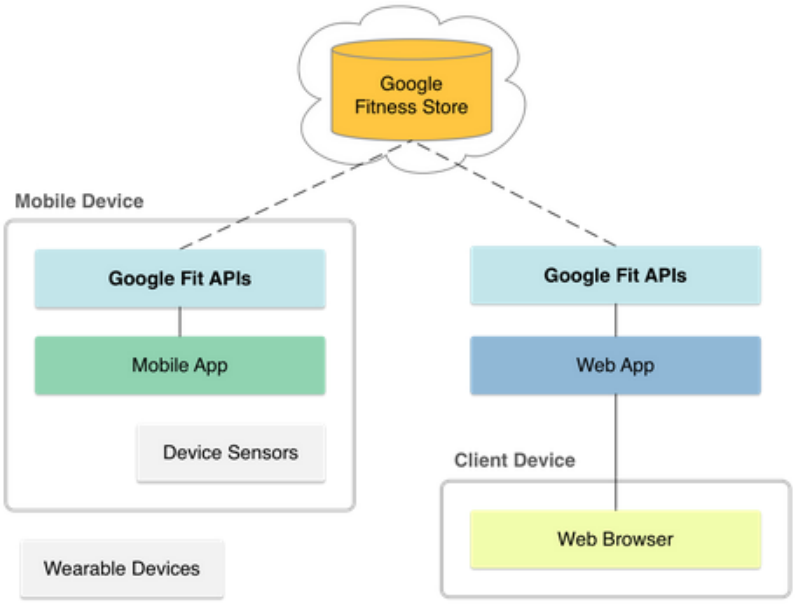


Figure 2.4: Google fit platform overview. Figure obtained from (6).

2.4 Angular algorithms

We have already studied how the miniaturization of the inertial sensors has permitted the design of inertial measurement units (IMUs) that are perfect to be put on any part

Figure 2.5: Inertial sensor of Shimmers (7)

of the human body. Now, inertial sensors, can be used in activity measurement applications and human body movement and due to their low weight and low size. IMUs are mainly composed of accelerometers, gyroscopes and magnetometers. To characterize and get the movement on the human body and the posture, the acceleration, the angular velocity and the position with respect to the Earth's magnetic field are the parameters that the three IMUs respectively use which is being monitored (8). Promoted by the study of the signals generated by the sensors that was carried out in the last few decades in order to be used in positioning applications, new algorithms have been developed for IMUs due to the existing algorithms prior to the development of the new IMUs are not totally applicable to new sensor technology, since sensors present a set of negative factors that need to be treated to avoid a loss of precision in the measurements. These factors could be noise, axis misalignment due to assembly errors and cross axis sensitivity among other factors that can cause the output of the sensor to be erroneous if they are not dealt with properly. One of the main negative effects is the angle wander produced when the angular velocity gyroscope signal is integrated. What are the solutions?

- *Sensor fusion.* It is defined as the use of various sensors to improve the accuracy of the measurements under situations where one or more sensors of the network are not working properly. This strategy can compensate the negative effects of

the angle wander, for example, to help to determinate the tilt angle with a major precision it can be used an accelerometer (61).

- *Kalman filter.* At first, we need to identify the process we want to estimate and remove. Later, we need an observation process that will act as a reference to help determine the hidden process. Depending on the application scenario, the filter can be turned in such a way as to decide which signal to trust more by modifying the values of the filter parameters. This behavior allows us to tune the filter attending the activity that is being monitored. When monitoring intense activities, the accelerometer signal will be given a lower degree of trust so that dynamic accelerations can be filtered. Alternatively, when non-intense movements are monitored, we give the accelerometer angle signal a higher degree of trust. The working is showned in Figure 2.6(a). All tested filters outperform the Kalman filtering approach as it does not behave properly when the error between the estimated and the actual value is rather large (61, 62).

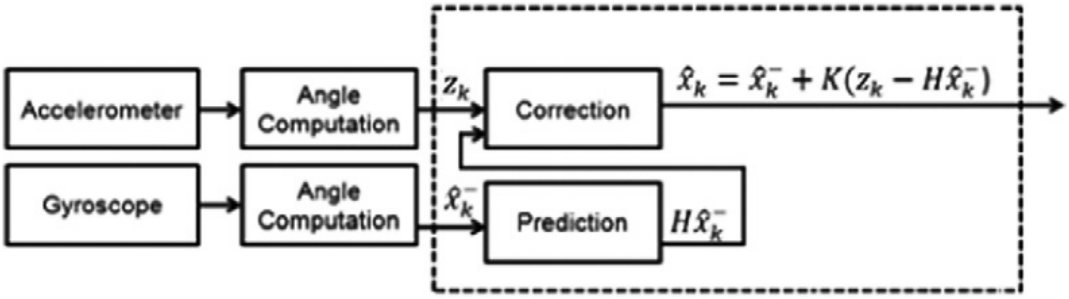
- *Least mean squares (LMS) filter.* These are linear adaptive filtering algorithms, consist of two basic processes: the first is a filtering process, which involves, compute the output of a linear filter in response to an input signal and generate an estimation error by compare this output with a desired response; the second is an adaptative process which involves the automatic adjustment of the parameters of the filter in accordance with the estimation error. The combination of these two processes working together constitutes a feedback loop. A significant feature of the LMS algorithm family is its simplicity since it does not require measurements of the pertinent correlation functions, nor does it require matrix inversions (61, 63).

- *Recursive least squares (RLS) filter.* This is an algorithm which recursively finds the filter coefficients that minimize a weighted linear least-squares cost function relating to the input signals. This is in contrast to the LMS filter that aims to reduce the mean square error. Its rate of convergence is typically an order of magnitude faster than that of the simple LMS filter due to the fact that the RLS filter whitens the input data by using the inverse correlation matrix of the data, assumed to be of zero mean. For high-intensity input signals, the standard RLS

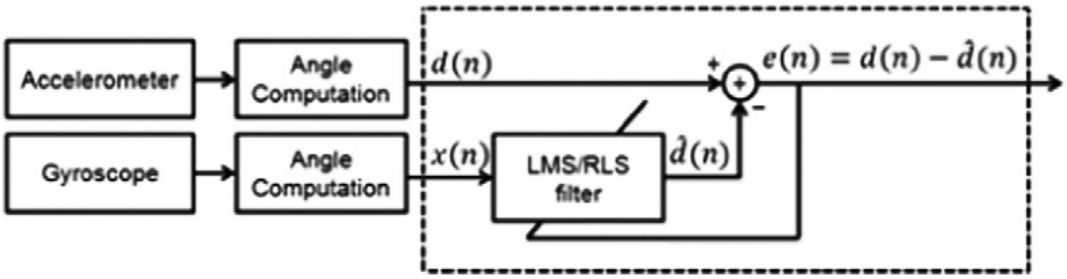
filter obtained the best results as the dynamic acceleration peaks filtering was more balanced than in the rest of the tested filters. However, it is achieved at the expense of an increase in computational complexity (61). Overall, standard RLS has revealed itself to be the most robust algorithm for obtaining accurate limb angle determination under both low and high intensity movement conditions. The working of LMS/RMS is displayed in Figure 2.6(b).

- *Sensor based three dimensional (3D) joint angle measurement technique.* Joint angles are determined from the orientation of one segment relative to another and are not concerned with absolute segment orientation. A complimentary inertial/magnetic sensor based technique is presented which can be used to measure 3D joint angles between two body segments. It is dependent only on the periodic existence of a common reference coordinate system which need not be fixed and may be suitable for use in a dynamic system. Each sensor unit is attached securely to the respective segment such that the orientation of the sensor unit with respect to the body segment does not change. An orientation matrix is used to describe the orientation of the body segment coordinate system with respect to the coordinate system of the sensor unit attached to it (64).

- *Gradient descent algorithm.* This is novel orientation algorithm designed to support a computationally efficient, wearable inertial human motion tracking system for rehabilitation applications. The algorithm uses a quaternion representation, allowing accelerometer and magnetometer data to be used in an analytically derived and optimised gradient descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative (65). This algorithm will be explain with more detail in .



(a)



(b)

Figure 2.6: (**a**) Diagram of the Kalman filter sensor fusion approach (8), (**b**) Diagram of the LMS/RLS filter sensor fusion approach. (8)

3

Methodology

In this chapter, we describe the methodology followed in the realization of this study. The chapter is divided in three parts: Trunk endurance assessment Section 3.1, where we describe the different tests that we use in our system; Electromyography (EMG) Section 3.2, which it consists in a explication and a description about the different EMG procedures Section 3.2.1 and how we assess the muscle fatigue Section 3.2.2.

3.1 Trunk endurance assessment

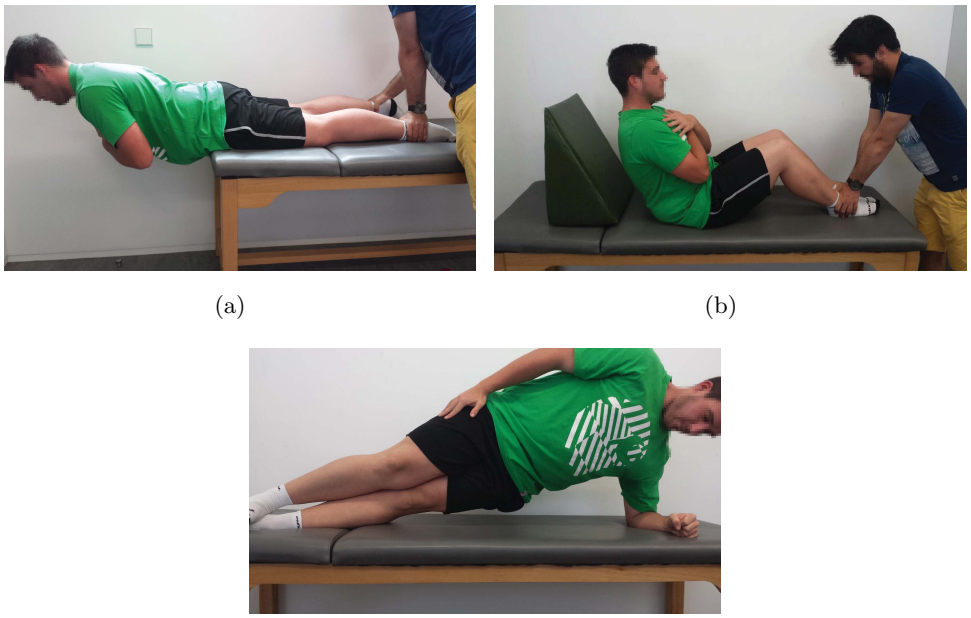
Different tests are available to assess the trunk endurance in people with or without LBP. These kind of tests are performed by a specialist, and they normally consist in the measurement of the time a person can hold a specific posture involving the trunk muscles. During the execution of the test, the health professional has to control the patient position and decide when the test ends, according to some established termination criteria. The results obtained for a given patient help experts determine their status and muscular capacity, as well as their ability to hold a posture normally related to daily living activities. To assess the low back stabilization several functional trunk endurance tests can be found in the literature (17, 66). The most widely used ones are the *static trunk extensor endurance test* (STEET), also known as Sorensen test (67), the *trunk curl static endurance test* (TCSET), also known as trunk flexor endurance test (68), and the *side bridge endurance test* (SBET) (18) (see Figure 3.1). In the STEET, the subject has to maintain a horizontal unsupported posture with the upper body extending beyond the edge of the bench. In the TCSET, a curled position

must be hold with only the scapulae clearing the table. Finally, the SBET requires the individual to lie on their side while lifting the torso and thigh off the bench, such that the body weight is on the elbow and feet. Special remarks are that two chances are given to the individual to execute the STEET, while evaluation of both left and right sides are considered as part of the SBET. A detailed description of each test, including posture, procedure and finalization criteria, is shown in Table 3.1.

According to the aforementioned scientific literature, there are some established normal endurance times for young and healthy men and women. The average endurance time for STEET is established from 62 to 131s. In TCSET the mean duration is 134s, while for the SBET it is approximately 84s, with a standard deviation of 24.5s. Using these endurance time references, experts can estimate whether a person has a good trunk endurance or not. But not only the independent duration of each test is of relevance for the trunk endurance assessment, but also the relation among these values. Hence, ratios between flexor/extensor muscles and right/left sides are normally considered. These ratios show the equilibrium or disequilibrium between muscle groups. The ratio of trunk flexor to extensor endurance is 0.77 normally (0.84 in young males and 0.72 in young females). The ratio of right side bridge to left side bridge endurance is normally 0.96. A reduced ratio of trunk flexor to extensor help discriminate between LBP patients and healthy individuals, while a side to side difference greater than 0.05 suggests unbalanced endurance. The estimation of these reference values is explained in (17). Based on the regular experience of specialists, practical limitations can be observed during the course of the realization and evaluation of endurance tests. First of all, it is generally accepted that the tester has an important responsibility while determining the different phases of the test. The estimation of the beginning and end of the tests is completely subject to the expert visual interpretation. In fact, specialists often report on the difficulties faced during the observation of the trunk angle variation and termination criteria (69). Moreover, according to the nature of the tests (70, 71), the specialist normally needs to control several aspects simultaneously, such as, time, position, and possible abnormalities during the test, which in traditional procedures could be despised. Finally, the results are mainly elaborated on the time recorded during the performance of the test, and that is the unique information to compare with in future tests. These drawbacks make complex the comparison of values measured by different testers and among sessions.

	Static trunk extensor endurance test (STEET)	**Trunk curl static endurance test (TCSET)**	**Side bridge endurance test (SBET)**
Patient position	- Prone with the inguinal region/anterior superior iliac spine at the edge of the bench.	- Arms are folded across chest and back laid on a piece of wood to support the patient at a fixed angle of 60°.	- The subject lies on one side supported by their pelvis, lower extremity and forearm.
	- Arms at sides, ankles fixed (by strap or hands), holding horizontal position.	- Toes are anchored either with a strap or by the tester.	- The top leg is placed in front of the lower leg with both feet on the floor.
		- Both knees and hips are flexed 90°.	- The upper arm is placed against the chest with the hand touching the anterior lower shoulder.
Procedure	- The patient maintains the horizontal position as long as possible.	- The wood is pulled back 10 cm (4 in.).	- The pelvis is raised off the table as high as possible and held in a line with the long axis of the body, supporting the weight between the feet and elbow.
	- Timing begins when posture is horizontal and unsupported.	- Timing starts when the initial posture is achieved.	- Timing starts when the initial posture is achieved.
	- Subjects are verbally encouraged to hold this position as long as possible.	- The subject holds the isometric posture as long as possible.	- Subject statically maintains this elevated position.
Termination Criteria	- The position is held up to a maximum of 240s.	- No specific time limitation although generally considered a maximum of 240s.	- No specific time limitation although generally considered a maximum of 240s.
	- If patient drops below the horizontal position more than 10° (an additional chance to regain it is given after first attempt).	- When any part of the subject's back touches the wood. This generally equals to a drop of more than 30° with respect to the reference.	- Subject is unable to lift their body up from the floor or drops their pelvis or thigh part way more than 10° and cannot raise it up to the start position again.
	- If patient reports LBP or cramping in their legs the test may be stopped.	- Significant LBP causes the test to be stopped.	- Significant LBP causes the test to be stopped.

Table 3.1: Trunk endurance tests description.



(a) (b)

(c)

Figure 3.1: Functional trunk endurance tests: (**a**) STEET; (**b**) TCSET; and (**c**) SBET.

3.2 Electromyography (EMG)

In this section, we explain importantly, what is the EMG and the meaning of muscle fatigue and we describe the different techniques to measure the EMG signal and how we can use that signal to extract the most important features to characterize the muscle fatigue.

3.2.1 EMG: definition and technical description

According to (72), electromyography (EMG) is based on the study of muscle activity through observation and analysis of the electrical signals monitored during voluntary or involuntary muscle contractions. EMG is useful to study of muscular function during activities of sports, biofeedback training, daily living or detect pathological states of the musculoskeletal systems. In order to assess all those functions, EMG offers useful information about timing of muscular activity and its relative intensity. There are several ways to collect EMG data:

- *Intramuscular fine wire electrode.* In this technical is used a needle electrode

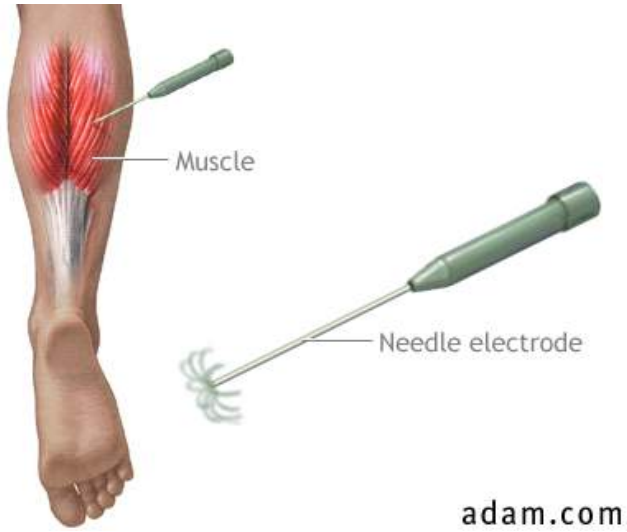


Figure 3.2: Intramuscular fine wire electrode (9).

which is inserted through the skin, into the muscle tissue Figure 3.2. It is used to study the physiology and pathology of the unit motors, such as denervation, reinnervation and myopathies, among other units. Therefore, a big disadvantage is that it is an invasive procedure. Moreover, intramuscular fine wire electrodes are less repeatable as it is very difficult to place the needle in the same area of the muscle each time. Based on the following study (73), where it is investigated the ability of pattern recognition to resolve movements that are simultaneous and dynamically changing and where it is compared the use of surface and untargeted intramuscular EMG signals for this purpose, it is concluded that the perfomance of intramuscular EMG on modulated data was poorer than nonmodulated signals, so the perfomance of perfomance of intramuscular EMG decreased when training with more complex data that surface EMG.

- *Surface electrodes.* The most important feature, from the application point of view, of the surface EMG is that a noninvasive technique. The surface EMG is able to provide a general picture of the electrical activity of a superficial muscle. As a consequence it is useful in monitoring the degree of muscle involvement during specific motor tasks, as a tool to monitor the development of localized muscle fatigue and temporal activity patterns (74).

3.2.2 Muscle fatigue: analysis and description

According to (74), muscle fatigue is manifested as a reduction in the ability to keep a certain level of strength in a sustained contraction or as the inability to achieve a level of initial strength in intermittent contractions and it is accompanied by changes in muscle electrical activity. Muscle fatigue has central and peripheral components. The first are manifested as an inability to realize an induced activity for the development of this. The second stand out as a muscle inability to produce a certain level of strength. In sustained contractions are produced central and peripheral fatigue during maximal and submaximal efforts, while in intermittent contractions are produced central fatigue when the effort is maximum, but periphery fatigue just is happened if the effort is submaximum or there is enough rest among contractions. In applications of electrical stimulation it is important to assess the development of muscle fatigue to prevent deterioration of the mechanical behavior of the muscle being stimulated.

Muscle fatigue can be evaluated by EMG signal. The effect of fatigue on muscle mechanics is well represented by various metrics, such as the mean frequency (MNF), median frequency (MDF), the root mean square (RMS), the rectified root mean square (RRMS), the average rectified value (ARV) and the maximum voluntary muscle contraction (MVC) (72, 75). This information is of much interest to compare the evolution of the muscle strength among sessions, as well as to measure the effectiveness of potential treatments. In this study, we monitor the EMG signal, and we give a time window or epoch of N samples then, the RMS, ARV and MVC values can be calculated as follows:

$$RMS = \sqrt{\frac{\sum_{k=1}^{N} EMG^2(k)}{N}} \tag{3.1}$$

$$ARV = \frac{\sum_{k=1}^{N} |EMG(k)|}{N} \tag{3.2}$$

$$MVC = max\left(EMG(k)\right) \tag{3.3}$$

In the time domain the RMS of the EMG signal is considered the most reliable parameter. An increase of the EMG RMS with advancing fatigue has been reported in many studies. RMS is not affected by the cancellation due to motor unit action

potential train superposition, which may affect other processing techniques involving rectification (75). This study (72) showed that muscle fatigue was accompanied with an increase in the RMS and ARV and a decrease in the MNF.

4

Project Description

In this chapter, we present a general overview about mDurance in Section 4.1; The OS that supports to mDurance in Section 4.2; How mDurance means the trunk posture in Section 4.3; A deep description of the functionality of the application to help the specialist to assess the different tests proposed in mDurance is explained in Section 4.4; Finally, it is described how mDurance has been implemented in Section 4.5.

4.1 Overview

Taking into account the limitations of traditional trunk endurance assessment approaches, in this section is presented, *mDurance* 4.1, an innovative system to support practitioners during regular trunk endurance assessment procedures. The *mDurance* system combines wearable sensors, capable of measuring physiological and biomechanical data, and mobile devices, dealing with the gathering, processing and persistence of the sensory data as well as the visualization of health outcomes. Concretely, the system consists of a wearable inertial sensor to estimate the trunk position and an attachable electromiography sensor to measure the activity of the skeletal muscles of the trunk. All the information generated by the sensors during the execution of the endurance tests is seamlessly and securely transmitted to a mobile application. This app develops on some of the functionalities provided by a recent mobile health framework, here considered for the level of abstraction and agility offered to the developers. In the following, the key features of the *mDurance* system are thoroughly described.

Figure 4.1: mDurance system is based in wearable sensors plus an application mobile.

4.2 Android

4.2.1 Operating System

Android is an operating system based on the Linux Kernel, specially designed for touchscreen mobile devices such as smart-phones and tablets computers. It was initially developed by Android Inc., which was backed financially by Google and later bought in 2005. Google announced Android in 2007 along with the founding of the Open Handset Alliance (76). The first Android-powered phone was sold in October 2008.

The user interface of Android consist in using touch input corresponding to real-world actions such as swiping, tapping or pinching to manipulate screen objects. Android allows users to customize their home screens with shortcuts to their applications or widget, which offers users to display live content. Applications can send notifications to the users to inform them of something relevant, such new messages, alarms or appointments. Most Android-powered devices own built-in sensors which are used by applications to provide different functionality to the user, such as measuring motion, orientation or environmental conditions. These sensors are capable of providing raw data with high precision and accuracy.

One of the strongest features of Android is the fact that it is an open source and its code is released under the Apache License (77). This allows the software to be freely modified and distributed by device manufactures or enthusiast developers. Android has a large community of developers writing applications (apps), that provides Android devices with almost boundless possibilities. In November 2013, there were almost 900.000 apps available for Android (878632 exactly, with a 22% of low quality apps) (78), and the estimated number of apps downloaded from Google Play (Android's app store) was 25 billion (79). A survey conducted in April-May 2013 revealed that Android is the most popular platform for developers, used by 71% of the mobile developer population (80).

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It is the largest installed base of any mobile platform and growing fast every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content (81).

The Android architecture is formed by the following layers:

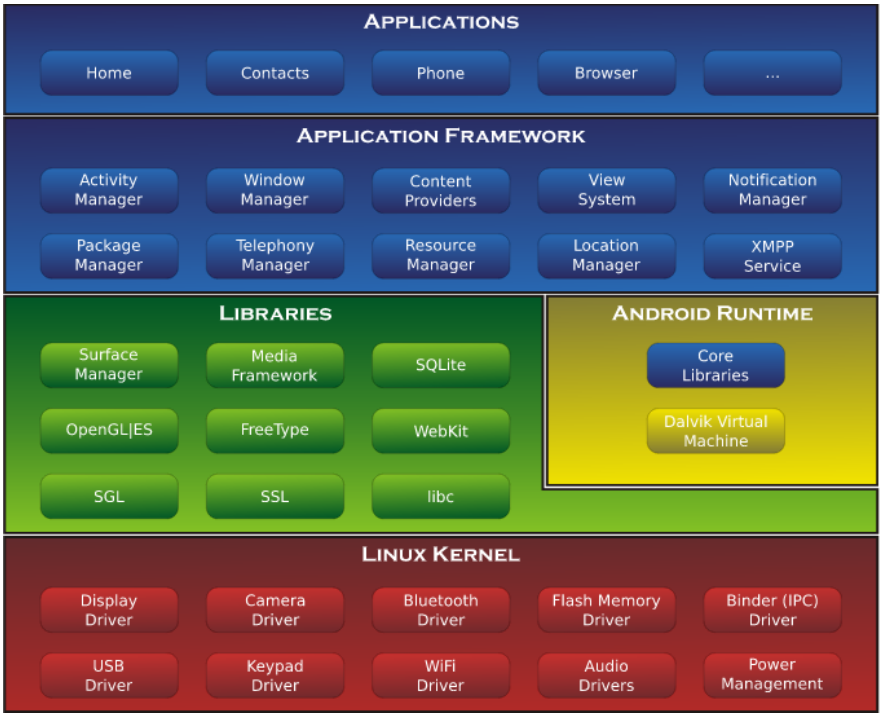


Figure 4.2: Android Architecture Diagram

1. **Applications**. Android applications are not exactly a part of the Android operating system and lies on the top of the three layers that form the Android Software Architecture. Android devices are delivered with a set of applications capable to provide basic services, such as web browsing, sending SMS or making phone calls. This is accomplished and managed by the underlying layers like the Applications framework one, just under this application layer. Android applications are written in Java and executed in Dalvik (DVM), a specific Java runtime engine.

2. **Application Framework**. The upper layer of the diagram. Provides framework services and libraries to interact with the application layer. These are written mostly in Java. Some examples are the Activity Manager, Location Manager, Notification Manager or View System.

3. **Libraries**. Special libraries written in C or C++. These libraries offer services to applications like SQLite for Databases, OpenGL or WebKit for accessing website.

4. **Android Runtime**. Consisting of two components, the DVM and core libraries responsible for process management. The DVM is a Java runtime engine which processes are isolated from the environment. It executes program written in Java by using a DX tool to convert it into bytecode that is readable by the DVM.

5. **Linux Kernel**. Is the lowest layer of Android Architecture and is the abstraction layer between software stack and hardware. It includes drivers for hardware, networking, security, energy consumption file, system access, processes and memory management.

4.2.2 Advantages/Drawbacks

Some advantages and drawbacks of Android Operating System are presented to show what Android may provide to an user and Android weaknesses:

4.2.2.1 Advantages

- Providing the code to everyone by making it open and free source makes it easy for developers to implement their own libraries.

- Populist Operating System. Android Phones, opposite to iOS which are limited to the IPhone from Apple, have many and potential manufacturers such as Sony or Samsung. That is the reason why there exists a wide variety in phones (style and capacities), suiting user budget.
- Easy access to the Android App Market (Google Play), where Android users can download thousands of free apps.
- Multitasking, which means that many applications may be running at the same time.
- Frequent OS updates, for improving performance.
- Installing ROM modification. There are many custom ROM installable on Android phones, guarantying they will not harm the Android device.
- USB full facilities, which means user may replace battery, mass storage, disk driver or USB tethering.
- Support all Google Services.
- And many other like easy in terms of notifications, widgets, router functionality...

4.2.2.2 Drawbacks

- In order to provide full functionalities it needs an active Internet connection.
- Some phones become slow after installing many apps. However, new Android versions and mobiles devices which offer better capacities deal better with this problem.
- Many applications contain malware, even if they are available in the Google Play Market users need to be careful. Besides, advertising is usually presents in applications.
- Wasteful battery. Android is said to be more wasteful than any other mobile operating system, due to constant processing in the background that lead to quickly drains batteries.

- Sometimes applications, external devices or just protocols as Bluetooth presents strange behaviors depending on the mobile device where are being used, due to the wide variety of mobile devices existing (manufacturers are free to insert modifications in Android).

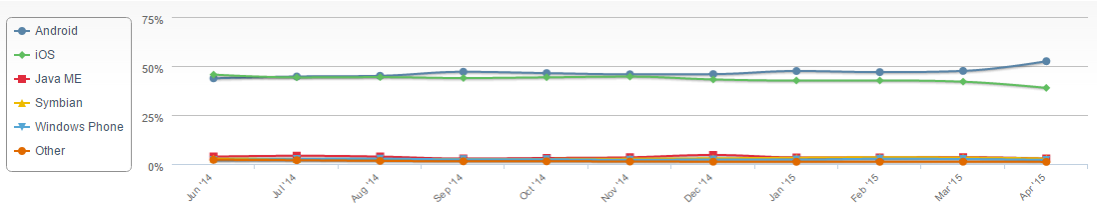


Figure 4.3: Mobile/Tablet Top Operating System Share Trend (10).

4.2.3 Why Android?

It is aim of this work to provide an app to permit to the specialist to realize an easier trunk endurance assessment with respect traditional assessments. Having this in mind, Android seems to be the mobile operating system which is the most appropriate, owing to the amount of users and mobiles devices Android powered. The fact that is the most used such as Figure 4.3 displays, converts it in the perfect system for development it.

4.3 Automatic Measurement of Trunk Posture

Determining the human trunk posture is of crucial importance to set the start of the endurance test as well as to automate the identification of its completion. To do so, mDurance benefits from the use of an inertial measurement unit (IMU), which combines triaxial accelerometers, gyroscopes, and magnetometers, enabling the measurement of the absolute attitudes or inclinations of the body part the sensor is fastened to. This technology, extensively used in the navigation domain (82), has been exploited during the recent years for body movement analysis (83, 84, 85, 86, 87). Apart from their precision, these sensors are particularly interesting since they are completely self-contained, thus introducing constrains neither in motion nor any specific environment.

IMUs provide raw acceleration, angular rate and magnetic field data that need to be fused together to obtain a sole, optimal estimate of orientation. Such as we saw

in Section 2.4, diverse algorithms have been proposed in the literature to that end, including Kalman filters , Least Squares filters (63) or Gaussian Particle filters (88), among many others (89, 90). The mDurance system particularly implements a recent technique, the Madgwick's algorithm (65), which outperforms most existing approaches in terms of implementation complexity, sampling rate requirements and computational needs. This technique does not suffer from well-known limitations of other solutions, like the singularity problem associated with the Euler angle representation (gimbal lock). Besides, this method also omits the use of computational expensive trigonometric functions, making it more efficient and easier to implement for real-time purposes. Madgwick's algorithm employs acceleration, angular rate and magnetic field measurements to analytically derive, through an optimized gradient-descent method, a quaternion representation of motion (91). Since the process is the same for the 2 vectors, we called $S_{\vec{v}}$ to the measure represented vector in the system of the sensor and $E_{\vec{V}}$ to the known represented vector in the system of the Earth. We can define the orientation of the system as a rotation which does that the vector $S_{\vec{v}}$ and the $E_{\vec{V}}$ are aligned. Among the infinities rotations that get this, we are going to stay with the smallest. Madgwick solves this as a optimization problem with the gradient-descent method.

The Earth's magnetic field has the same direction at all points on the planet. Thus, it is required a initial calibration of the algorithm to kwow the field direction in the system of Earth ($E_{\vec{M}}$).

The coordinates of the measured magnetic vector are expressed in the system of the sensor $s_{\vec{m}}$ and to get $E_{\vec{M}}$ we need the S orientation with respect to E.

To realize this calibration we are going to calculate the quaternion ${}^{E}_{S}q$ that allows us to get the vector $E_{\vec{M}}$ by the vector $s_{\vec{m}}$ by the next Equation 4.1.

$$E_{\vec{M}} = {}^{S}_{E}q^{\wedge} \bigotimes S_{\vec{m}} \bigotimes {}^{E}_{S}q^{\wedge} \tag{4.1}$$

If we integrate the angular velocity that the gyroscope transmits to the system in each moment we can get the orientation of system S.

The ${}^{E}_{S}q(\,{}^{E}_{S}\dot{q}$ derivative represents the velocity of the change of the orientation of system S with respect a E. We can get it through the vector of angular velocity ($S_{\vec{g}}$) that the gyroscope provides us by the Equation 4.3.

$$S_{\vec{g}} = \begin{bmatrix} 0 & g_x & g_y & g_z \end{bmatrix} \tag{4.2}$$

$$\begin{matrix} S \\ E \end{matrix} \dot{q}_G = \frac{1}{2} \begin{matrix} S \\ E \end{matrix} q \bigotimes S_{\vec{g}} \tag{4.3}$$

When we have already calculate $\begin{matrix} E \\ S \end{matrix} \dot{q}_t$, we can define the numeric integration by the Equation 4.4.

$$\begin{matrix} E \\ S \end{matrix} q_t = \begin{matrix} E \\ S \end{matrix} q_{t-1} + \begin{matrix} E \\ S \end{matrix} \dot{q}_t \Delta t \tag{4.4}$$

It should be pointed that this derivative is better if Δ t is small.

Now we are going to describe the algorithm to get the orientation of the system S with respect to E by the measure of field magnetic $s_{\vec{m}}$ through the magnetometer. We are going to use the difference between the vector $E_{\vec{M}}$ and $s_{\vec{m}}$ to calculate the error in the estimation of the orientation of the system. We are going to do it as a optimization problem and such as Madgwick proposes, we use the descent-gradient algorithm. Thus, the output of the algorithm is a quaternion, a compact vector in the form (q_1, q_2, q_3, q_4), which dynamically represents the orientation of the sensor. A detailed description of the foundations of the considered algorithm can be seen in (92) and in Figure 4.4.

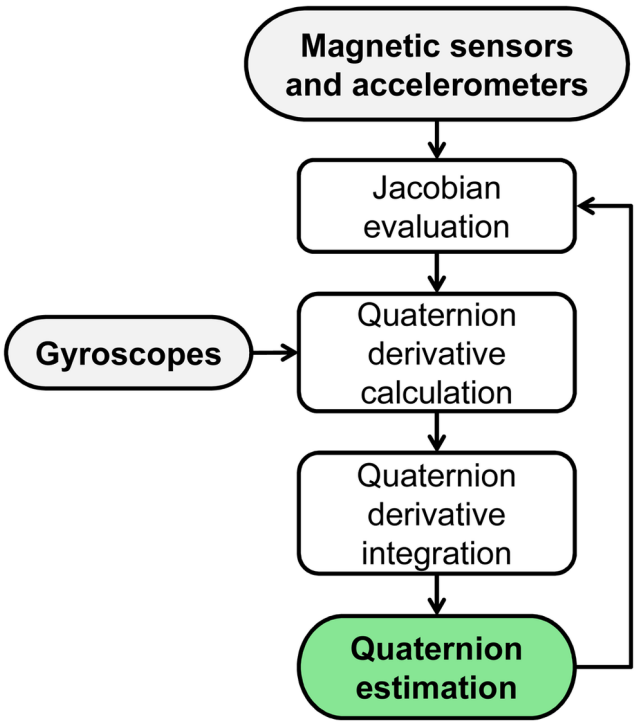


Figure 4.4: Madgwick's algorithm perfomance (11).

Quaternions are frequently used in orientation estimation algorithms because of their numerical stability and computational efficiency. However, this representation is difficult to interpret and visualize since it defines a $\mathbb{R}^4$ space that cannot be represented in a human-understandable three-dimensional view. Accordingly, a translation into Euler angles is performed here, after all the calculations to estimate the quaternion are carried out. Euler angles represent the possible rotations around the three cardinal axes, namely, yaw (φ), for the X axis, pitch (θ), for the Y axis, and roll (ϕ), for the Z axis. Given the estimated quaternion, the Euler angles can be simply obtained as follows:

$$\varphi = arctan\left(\frac{2(q_1q_4 - q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \tag{4.5}$$

$$\theta = arcsin\left(2(q_1q_3 - q_4q_1)\right) \tag{4.6}$$

$$\phi = arctan\left(\frac{2(q_1q_2 - q_3q_4)}{1 - 2(q_2^2 + q_3^2)}\right) \tag{4.7}$$

4.4 Application description

One of the main aims of the mDurance system is to help experts assess, in a precise manner, the time invested by the patients during the execution of the trunk endurance test, as well as the amount of muscle fatigue experienced in that process. To attain the first objective, an IMU sensor is considered to determine when the test termination criterion is met, based on the principle presented in Section 4.3. For the second goal, an EMG sensor is used to continuously detect the electrical potential generated by the muscle cells in the course of the test, as explained in Section 3.2.2.

Shimmer wearable sensors, concretely, version 2 for the EMG and version 3 for the IMU are employed, given the high reliability yielded by these commercial devices. The default sampling rate configuration, i.e., 51.2Hz, is used for both sensors since it proves to be enough for an accurate estimation of the trunk angle and EMG metrics.

Figure 4.5 shows the sensor deployment for each of the three trunk endurance tests supported by mDurance and described in Section 3.1. The sensors are located in convenient positions to ensure stability and comfortability, as well as an accurate measurement of both trunk angles and EMG values for each test. In the STEET and TCSET, the

(a)

(b)

(c)

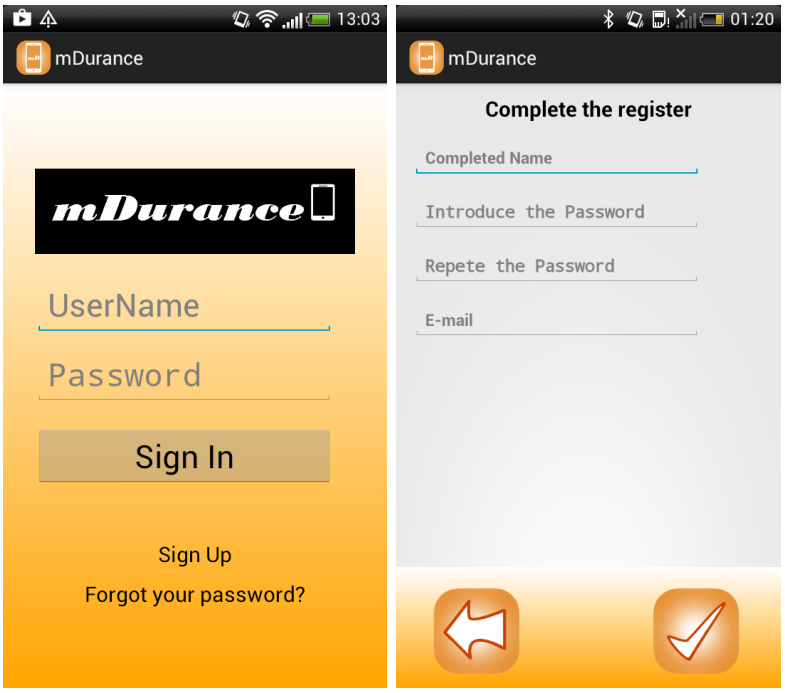
Figure 4.5: Sensor deployment for (**a**) STEET, (**b**) TCSET and (**c**) SBET procedures.

trunk angle is measured with respect to the coronal plane, while for the SBET the reference corresponds to the sagital plane. Accordingly, the IMU sensor is attached to the lumbar zone (D12-L1 vertebra) for the STEET and TCSET procedures, and to the dorsal for the SBET. Taking into account the placement of the IMU sensor for each case, and its local frame of reference orientation, the roll angle (ϕ) is used to represent the trunk angle in all tests. The EMG sensor is placed on the lumbar (erector spinae),

abdominal (rectus abdominis) and external oblique parts for the STEET, TCSET, and SBET, respectively. The electrodes are distributed to cover a sufficient muscle area. Both IMU and EMG sensors are safely and firmly fastened to each corresponding body part through ergonomic straps that ensure no misplacement of the sensors. Moreover, there is little room for errors during the placement of the devices since the sensors must be positioned by the experts.
In the following the mDurance application is described stet by step.

4.4.1 Specialist login

This is the first screen Figure 4.6 after initiation of the application. For the first time use, the expert is requested to sign up with their personal information to register in the system Figure 4.6(b).



(a) (b)

Figure 4.6: Screens: (**a**) Specialist Sign Up; (**b**) Specialist Login with a registered expert in the system.

This information is used by mDurance to uniquely identify the specialist, and also preserve the patient's data collected by the system in a confidential and integral man-

ner. In a way that, when the register is completed, a new table is created in *DBUser* with the specialist's username as tittle. This table will store the patient's information associated to this expert. Each new specialist created is a new table added to this internal database. Once an expert profile is created, the practitioner can log into the application contents by using their username and password, such us Figure 4.6(a) displays.

4.4.2 Main Menu

The expert is directed to a new screen Figure 4.7, in which as first possibility, they can either selected one of the existing patients associated to actual specialist in his corresponding table from the system database (which we are mentioned in the last section) or include a new one.

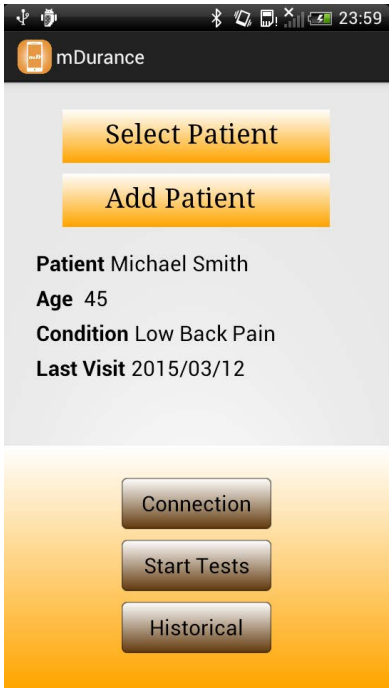
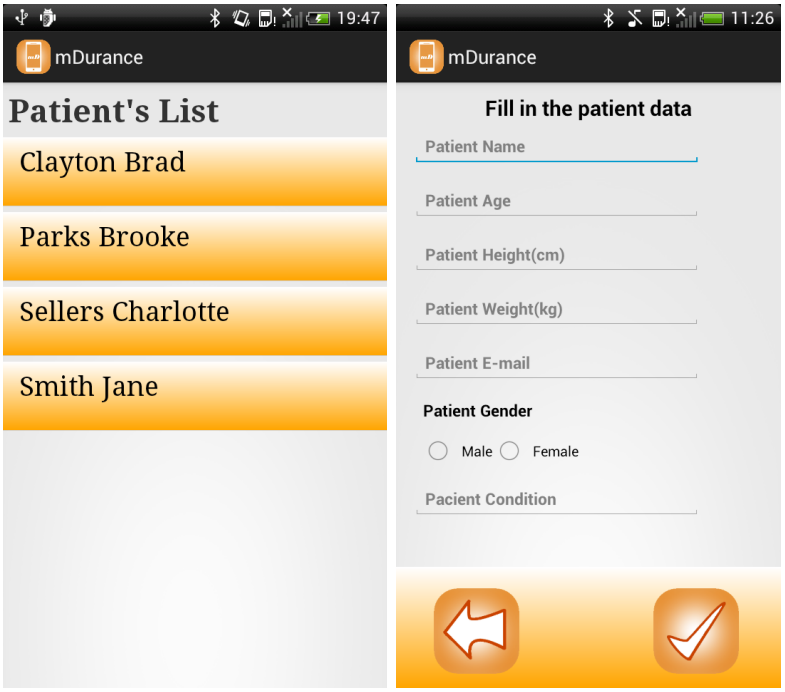


Figure 4.7: Specialist main menu screen with a selected patient

For the first option, a specialist can select one patient from a list like Figure 4.8(a). The list shows every existing patients from the specialist's table in alphabetical order. In the other hand, if a user wants to add a new patient, the specialist have to select the "Add Patient" option and fill in personal information, such as name, age, height,



(a) (b)

Figure 4.8: Screens: (**a**) Registered patient list in the system) Specialist Login; (**b** New patient register.

weight, gender and possible health conditions Figure 4.8(b). Thereupon selecting a patient, their more relevant information is presented to the expert for quick inspection, including the date of the last endurance session and particular conditions they suffer from, such as it appears in Figure 4.7. Moreover, from this main screen the expert has three different options: he can either initiate the connection with the wearable sensors, starts the endurance tests or visualizes the historical data collected during previous sessions. In the next sections, we are going to study each case.

4.4.3 Endurance tests assessment with application

The first goal of this project is make easier the specialist tasks to assess trunk endurance tests. In order to achieve this goal, application follows the next steps for the execution of the tests:

4.4.3.1 Wearable sensors connection

Before the execution of the tests, the specialist must to connect the sensors with the application by clicking on "Connection" of the main menu. If which user will try to access to the execution of the tests, a new emerging window will advise and stop them. The specialist will return to the main menu. Now, we are going to describe the different parts of the screen. At first, an imaginary rectangle appears on the top, which it shows the device's names and a circular colored flag associated each device that it indicates the status from each device in each moment by a determinate color. The set of colors are indicated in Figure 4.9. The names are: *EMG device* and *Angle Device*. The order

Figure 4.9: Colors of the status circles: Red (*Disconnected*), Green (*Connected*) and Orange (*Streaming*)

of the names is important because when the specialist wants to pairs it has to be in this order such as we are going to explain thereupon. On the bottom, it is displayed the power button, which allows connect and disconnect the devices. During the very first configuration of the system, the sensors must be paired with the mobile device. The progress of pairing follows the next steps:

1. When there are not any paired devices and the user pushes the power button, the Bluetooth interface is activated Figure 4.10. When this happens, there are several possibilities. The first is that no ID devices appears in the Bluetooth interface. In this case, the expert has to press the scan button to scan all devices that they have the Bluetooth connected. When the system finds a new device, on the bottom of the screen appears both the ID device and the MAC address of the same device. The second case is that the application has already use before and therefore on the top of the screen the specialist can get the ID device. In both cases, the specialist must to select the ID device which it corresponds to the EMG sensor. Once this is selected, EMG sensor is paired with EMG device in the application and every changes in its status(disconnected, connected and streaming) will appear reflected through its status circle. In this first case, if there are not any problem connection, the circle must to change from red to green color to signal the status connected Figure 4.11(a).

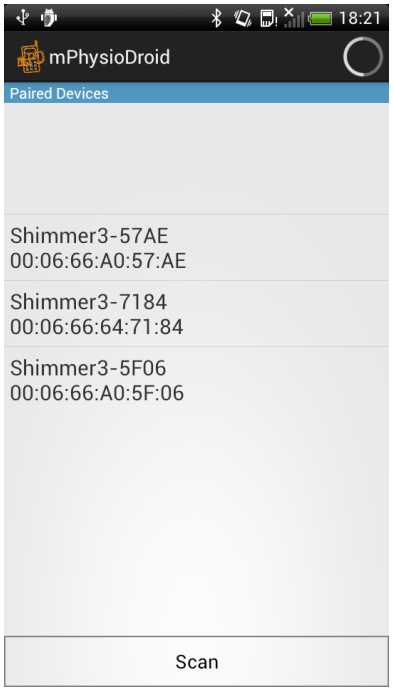


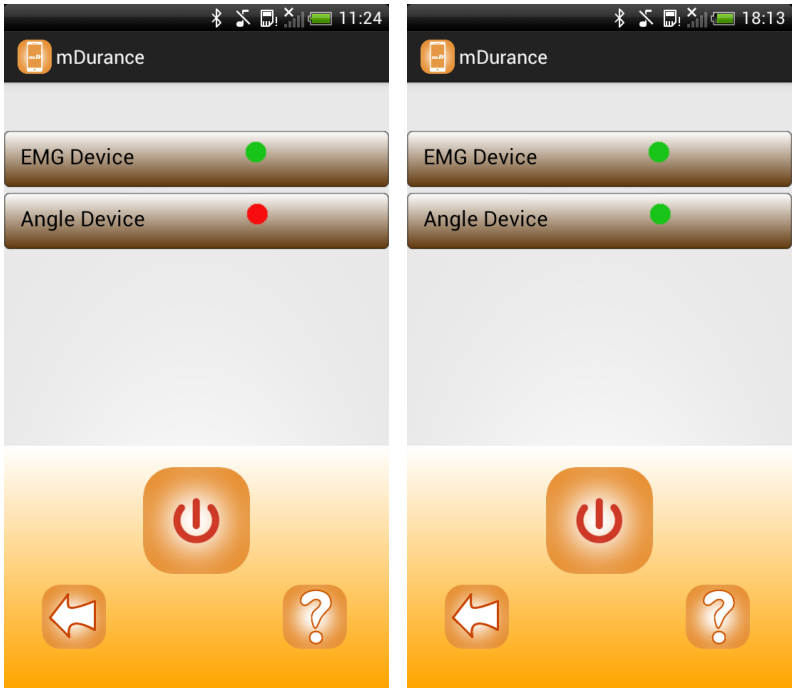
Figure 4.10: Bluetooth interface to pair the devices with the app.

2. When the first device has been paired with the application, the next step is to pair the second: IMU sensor. The process is the same that the previous device. The specialist has to select the ID device that it corresponds with the IMU sensor and the user has to check the final status circle of the Angle Device in the application. Finally, if there are not any problem connection the result has to be like Figure 4.11(b) and both the mobile device and the Shimmer sensors bound.

3. If there was some problem with the pairing in some device, the possible solutions are seem in SECTION.

After configuration, this one-time process is no longer required, unless the sensors are replaced. From the on, the expert can normally trigger the connection of the mobile and the wearable devices by pressing the power button. And now, we are going to show in Figure 4.12 and to explain, the different status colors of the power button depending of the status of the system and the own status in which the system can be find:

- Red color. In this case there 3 possibilities, the first one is that no devices are



(a) (b)

Figure 4.11: (**a**) Paired EMG device; (**b**) Paired IMU device.

paired or there is just one paired device. In that situation the specialist has to go to previous explanation. The second case is that both devices are paired with their corresponding application's names, but there is some problem connection (this occurs if the power button is red and some status circle is red too). When it happens, it is enough with press the power button until we get the next status (green color). The last case is the good. This happens when both IMU sensor and EMG sensor are connect correctly (both status circles are green). It is enough to push the power button to change to the next status.

- Green color. This status is the corresponding with the activation of the software of the sensors. It indicates that the sensors are ready to stream.
- Blue color. In the end, when the system achieves this status, it means that the sensors are streaming their corresponding information and data.

It should be pointed that the user can manage and change the status system in each moment. It wants to say that if the system is find in streaming (blue color), if

Figure 4.12: Colors of the power button: Red (*Disconnected*), Green (*Connected*) and Blue (*Streaming*)

the specialist press the power button the devices stop to stream and the status changes to connected status (green color) and the same with the step from connected status to disconnected status (red color).

In the end, the status of the system to start the assessment of the test would be as the displayed in Figure 4.13.

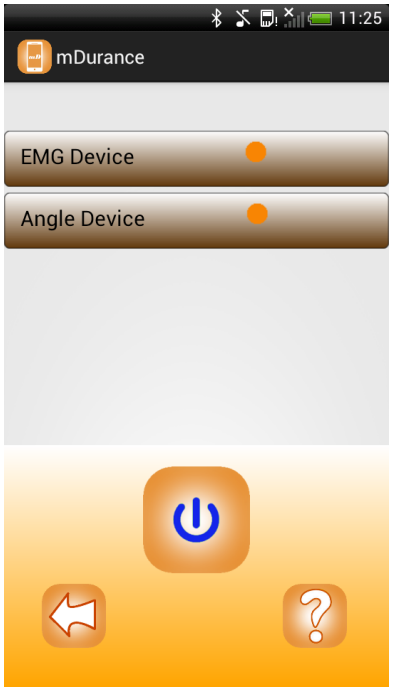


Figure 4.13: Final connection status to start a test

4.4.3.2 Tests execution

Once the sensors are streaming, and in order to proceed with the execution of the rests, the expert has to press "Start Test" in the main menu. As a result, the specialist is directed to a new window in which the particular test to be performed can be chosen

Figure 4.14. Consequently, the application offers to the specialist, the possibility to realize each test individually.

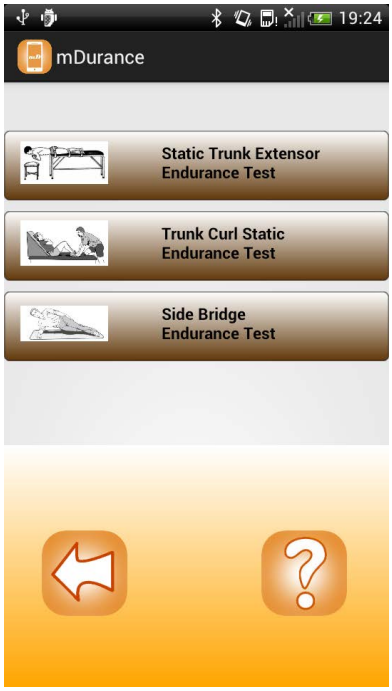
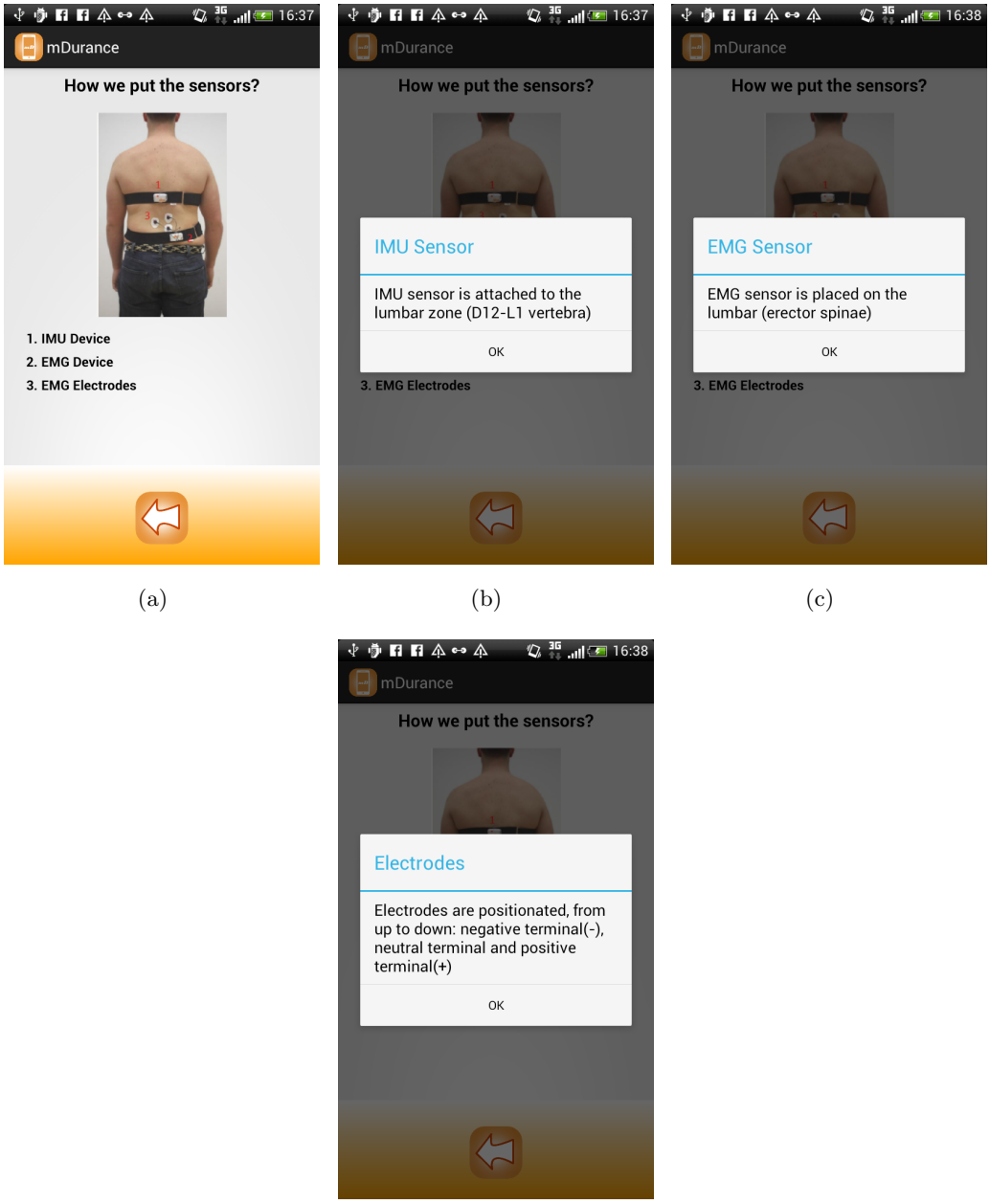


Figure 4.14: Tests selection screen.

In the bottom of the screen, if the expert presses the "?" button, it can be consulted a guideline of each test that where the specialists can consult how they can assess each test or the placement of the sensors on the patient for each test individually. We can observe an example about how the specialist has to place the sensors in the STEET test in Figure 4.15. If the expert press whatever point (1, 2 or 3) about the different sensor places, an emergent window appears to explain the different elements that appears in the image.

After selecting a test, another screen is displayed with the essential elements required by the expert to perform the test. Figure 4.16(a). This includes, from up to down:

- A graph to visualize the recorded EMG signal at run-time. The parameters to visualize this information are the EMG data expressed in mV in the axis Y and in the axis X, the total number of samples that the EMG device has streamed during the execution of the specific test where we find. The specialist can control



(a) (b) (c)

(d)

Figure 4.15: Guide about how to place the sensors in STEET test: (**a**) Main screen; (**b**) IMU place; (**c**) EMG place; and (**d**) Electrodes place.

in each moment several graphic features (pressing the configuration button), such us:

(a) (b)

Figure 4.16: Test activity screens: (**a**) Initial scenario before to start a test; (**b**) Final scenario when the test has finished

- View Port. The specialist can define which is the window size of the graph to represent the samples.
- Y coordinates. The expert can specific the minimal and maximal Y axis values.
- Legend. Finally, it can be controlled the legend position with three different places.

- A timer to control the time left according to the maximum duration allowed for the realization of the test.

- The trunk angle continuously measured by the system. The trunk angle is particularly useful for the expert to determinate when the patient is correctly positioned.

Once the specialist determines that the starting position is reached, the test can be initiated by clicking on star test button. The angle measured at that moment is saved as reference, and used by the system to check whether the user exceeds the range

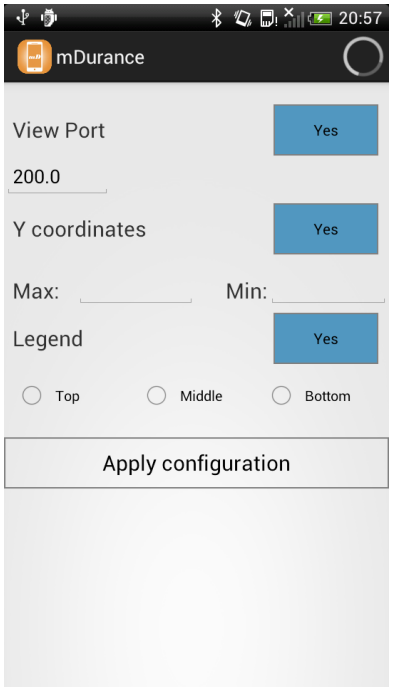


Figure 4.17: Graph configuration menu

defined for each test as part of the termination criteria. Thus, is the patient relaxes their posture more than $\pm 10°$ in the STEET and SBET, or $\pm 30°$ in the TCSET, the test is automatically finished. The end-of-test is also attained when it lasts more than 240s or when the expert explicitly considers that it should be finalize, to which the stop button can be used. We have to mentioned a few particularities of the test individually: in STEET exists 2 opportunities, therefore when the patient exceeds the range defined in the first occasion, the timer do not stop but the graph is stopped. To go to the next step in the assessment of this test the specialist has to press the second opportunity button. In SBET the expert has the option to choose the side to assess in first place. Each side must to be assess individually. It is noteworthy that in the beginning of each test, a new table is created in the internal database (*UserDB*) called *Sensors_Table*, which has as parameters to save:

- Patient's name. It is used for identify the patient whose the data come from.
- Label. This is an identification to classify the data according to the test where they come.

- Date. This is the date which the data are gotten.
- Angles. This is the set of angular values expressed in degrees that are gotten during the execution of each test individually and signalized by "label".
- Emg. This is the set of EMG values expressed in mV that are gotten during the execution of each test individually and signalized by "'label"'.

Thereupon, when the specialist starts a test and during the execution of the each test, the system is going to fill in this table with the angular and emg values get through the sensors. We have used a circular buffer of 800 samples of size which will fill in the mentioned table according to 2 possibilities. At first is that the buffer is full. Then the buffer is downloaded and is classified each data in its corresponding column (angular values in angles' column and EMG values in emg's column). In the other hand, the other possibility is the end or the stop of the test. In this case, the buffer is downloaded with the number of samples which has in that moment independently.

After the test finalization, in the bottom appears the result button which allows to the specialist to go to the results of the assessment of the test. It is observed in Figure 4.16(b).

4.4.3.3 Tests results

After the test finalization, the expert can observe a summary of the results Figure 4.18. This includes:

- The total duration of the test. It is the sum of the two attempts for the STEET case. For the rest of tests, this time is obtained of:

$$Time_{total} = 240 - time_{duration} \tag{4.8}$$

  For example, if the duration has been 100s, the final time is 140s.

- The result of the execution of the test. The session is categorized into "bad", "good" and "perfect" based on the statistical overall duration of the patient, introduced in Section 3.1. Concretely, we can consult them in the Table 4.1.

Figure 4.18: Tests Results Screen

	Bad result	**Good result**	**Perfect result**	**Execution ratio**
STEET	0 to 61s.	62 to 131s	132 to 240s	total_time/96.5
TCSET	0 to 133s	No specific	134 to 240s	total_time/134
SBET	0 to 49s	50 to 119s	120 to 240s	total_time/84.5

Table 4.1: Statistical overall duration of the patient and execution ratios.

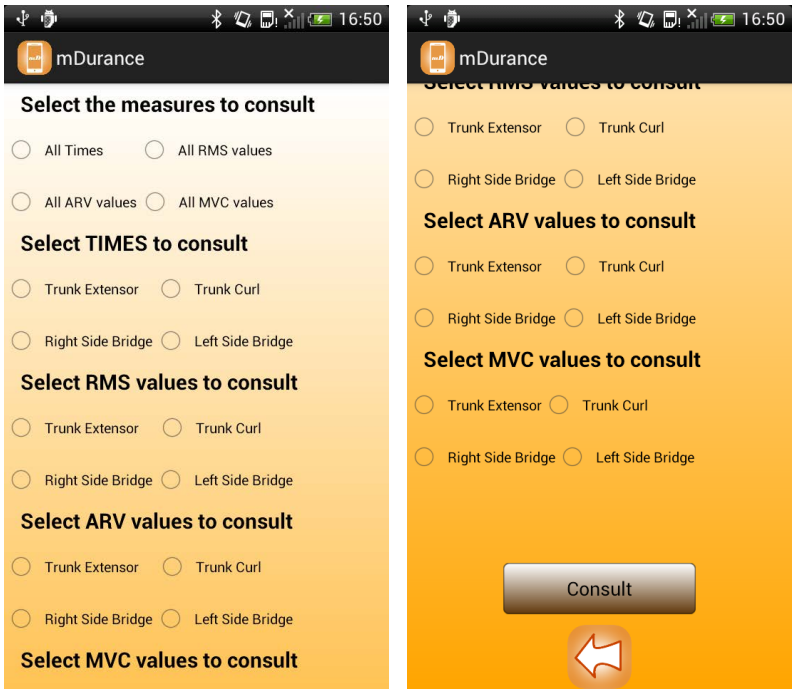
- The endurance ratio. This is a normalized value that represent the total time measured explained previously and divided by the theorist mean according to the healthy values that a healthy person must to obtain in the assessment of the corresponding test.

- The RMS, ARV and MVC values. This muscle fatigue values are explained in Section 3.2.2.

The set of results obtained in each assessment represent an important aid to achieve an improve in the result interpretation when the specialist wants to get a reliable diagnostic of each patient about the endurance and muscle fatigue overall.

4.4.4 Historical data consult

In addition, the expert can inspect the patient's historical data by clicking on the "Historical" button Figure 4.7. This opens a new screen Figure 4.19, in which diverse type of representations can be selected, such as:



(a) (b)

Figure 4.19: Historical data menu

- The time invested by the patient during the execution of the tests. This is both the set of times of every tests along the time and the individual time of each test along the time too.
- The muscle fatigue metrics. Alike, this can represent both RMS, ARV and MVC in the set of values obtained in every test along the time or the representation of RMS, ARV and MVC along the time individually.

When the specialist has selected a data and press the "consult" button, the user is conduce to a new screen. The result are depicted in a multidate basis for the different

past sessions registered in the system for the specific individual. Other consult can be the set of times or RMS. With the arrows, the specialist can move it the calendar and consults all historical patient. An example of this king of queries is the showed in Figure 4.20. All the information is obtained from the patient's table of the specialist

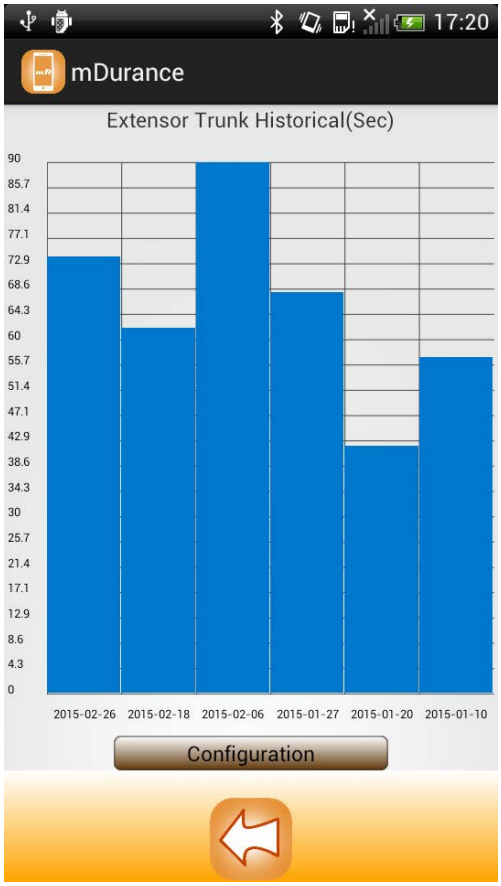


Figure 4.20: Example of a historical query

who is registered in the application in that moment. We will explain more details of this table in Section 4.5.5.

4.5 App implementation

In this section we will explain how we used mHealthDroid in the application and as we have implemented the main modules mDurance.

4.5.1 mHealthDroid description

mDurance has been implemented using mHealthDroid, an open source framework devised to support the agile and easy development of mHealth applications on Android. mHealthDroid, which is released under the GNU General Public License version 3 and available at (93), provides resource and communication abstraction, biomedical data acquisition, health knowledge extraction, persistent data storage, adaptive visualization, system management and value-added services. All that through of the several managers as: Communication, System, Data processing, Remote Storage and Visualization, all these are singleton class. mHealthDroid has considerably facilitated the implementation of the mDurance core functionalities (Figure 4.21), such as:

- The interface to the wearable sensors. Communication Manager is the responsible of this function. This is one of the most important managers. It is responsible for, among other things, the communication management with the biomedical devices. It makes the communication transparent to the application, and provides a unified and interpretable data format.

- The calculation of the test results. Communication Manager, its main function is to manage the data received by the devices, and deliver them to the rest of the framework's package/managers that need them. Thus, the manager delivers the data packages to the application to process them and calculate the different parameters of the application.

- The persistent data storage. In this case, it is the Storage module which it is responsible of the storage in the local database once the data packages are delivered and processed by the application.

- The visualization of the collected sensor information and historical test results. For these functions, it is used the Visualization Manager. It is in charge of the data visualization. It is possible to perform an online visualization of the data streaming, as well as an offline visualization of the data retrieved from the storage unit. It also allows to build graphics with different purposes, such as to show the test results in the last month of each patients or show the online visualization from EMG sensor.

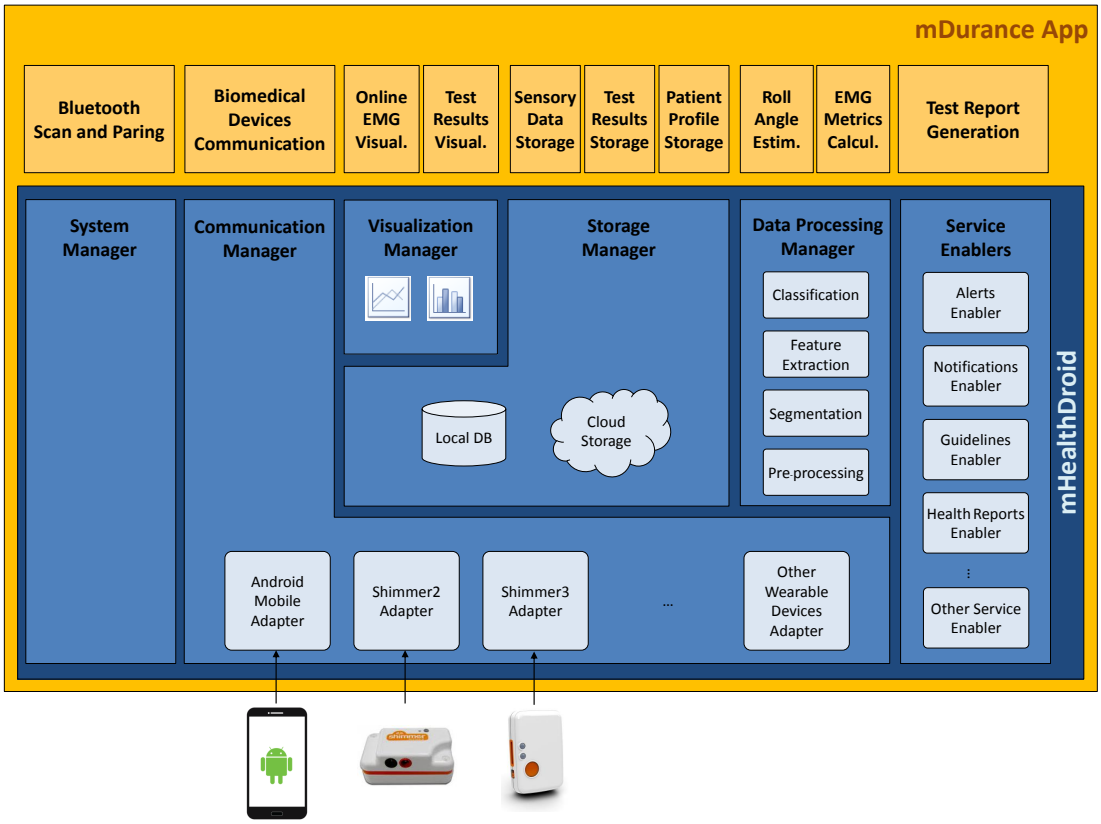


Figure 4.21: mDurance architecture building on mHealthDroid.

4.5.2 Communication implementation

The mDurance communication functionality relies on the mHealthDroid Communication Manager, which abstracts the underlying mobile and biomedical devices, makes the communication transparent to the application, and provides a unified and interpretable data format.

In order to do its work properly, the manager should run in background. Therefore it is defined as a *Service* (94). The data acquisition is handled by the drivers. They also create the framework's data structure along with them, so it is necessary to design some kind of mechanism for sending the data structure to the manager. The drivers send the data structure by *Message* (95) through a *Handler* (96) (both tools are provided by the Android API). However the manager do not do all the work. Independent threads are created instead. It is done like this to take advantage of the benefits of the approach "divide and conquer". Having said that, it is clear that a thread will be in charge of

just its own device. Thus, there will be as many threads as devices connected. In our case, there are 2 threads. The threads are defined within the manager, namely *CommunicationThread*. They only have the *Handler*, as mentioned before, and a *Looper* (97), which is a class used to run a message loop for a thread. Threads receive the message with the framework's data structure and work accordingly: store data, send data to the rest of the framework, or check whether the communication has failed. But there were other details that needed to be controlled, such as linking the thread with its device. To do this, the *ObjectCommunication* object was created. This object is an inner class within the manager and is formed by some flags,an array with the graph's name (this will be explained in Section 4.5.6), a *Communication Thread* and a *Device* (the driver previously mentioned).
The manager has, among other things, a Hashtable of String and *ObjectCommunication*. This Hashtable stores the devices. The device name is the key (so it has to be unique) and the *ObjectCommunication*, which make all the work described previously, is the object. To add a device to this hashtable the manager provides a pattern function: *AddDeviceSomething*. In particular, the manager provides *AddDeviceShimmer* and *AddDeviceMobile*. We only use the first function.

When the appropriate flags are activated, this manager also sends the data to the Storage package for storing them into the local database (which will be explained later), to the Visualization Manager for the data representation or to activate the communication with mDurance.

The two most important functions in the manager are *startStreaming* and *stopStreaming*. *startStreming* checks whether the device is connected or not. If so, the database tables are created, and the *CommunicationThread* is started. Then, and just in the first session, a database consistency check is performed in case the application stopped owing to any exception. Finally, the *startStreaming* function defining in the device driver is called (a device driver is needed in order to use the framework, but this will be explained later). The Communication thread runs in background receiving the messages sent by the device driver. When the first sample is received, the session metadata are stored in the database. In case the proper flags are activated, the thread also stores the samples into the database and sends them either to the *Visualization Manager* or to the *Data Processing Manager*. If the connection is lost, the thread stores

the rest of the session metadata, and in case the Store flag is activated the last block of samples is stored.

stopStreaming calls the device driver *stopStreaming* function.

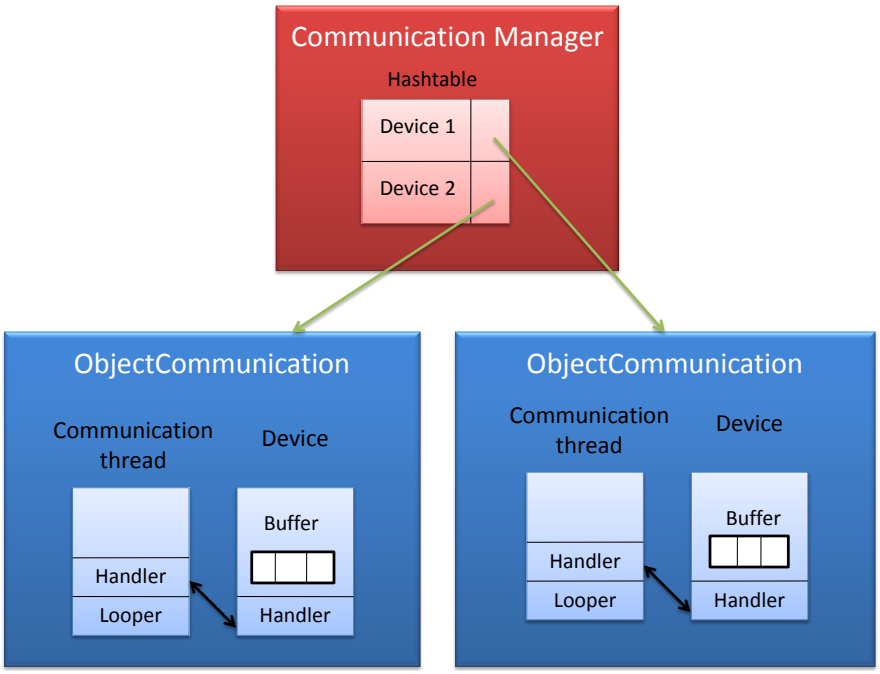


Figure 4.22: Data flow diagram from the devices to the framework

DATA RECEIVER

The Data receiver package contains the "drivers" for the devices that will be connected with the Android application. The aim of these drivers is to manage the connection with the device and fill the framework data structure with the sample received. Other kinds of functionalities offered by the device are also managed. The package is composed by 1 interface and 2 classes:

DEVICE

This interface is the pattern for creating the drivers, so obviously they must inherit from Device. Some of the most important functions in the interface are *connect/disconnect* which define the connection between the device and the Android application.

Start/stop streaming are also very important since they define the beginning and

ending of the data streaming. There are also other functions defining this pattern. It is possible to change the sample rate with the function *setRate* or consult it with *getRate*. In the same way, it is possible to change and consult the enabled sensors with *writeEnabledSensors* and *getEnabledSensors*. Sensor device drivers are required for their use within the framework.

DEVICE SHIMMER

This is the driver developed for the Shimmer device. Its aim is to manage the connection between the Shimmer device and the Android OS, to receive the data sent by the device and to create the framework data structures. The company which provides the Shimmer devices also provides a driver. As the mHealthDroid's manager, it is defined as a *Service* so that it can work in background. It is responsible for managing the connection with the Shimmer device. To do this, first create and start the *ConnectThread*. Once the connection has succeeded, the *ConnectedThread* is created and started. This thread manages all the incoming and outgoing transmissions. It receives the raw samples from the device. Then, the Shimmer driver converts them to calibrated and uncalibrated values. A hashtable is created with these values and sent in a message. It also comprises a wide range of functions for controlling all the device's features. From now on, this driver will be referenced as either *Shimmer driver* or *intermediate driver*.

The driver is composed, among other things, by a *BluetoothAdapater* object (98), a *BluetoothDevice* object (99), a circular buffer, an instance of the Shimmer driver, and a handler. The security in the communication is handled through the Security Manager Protocol (SMP) of the Bluetooth Low Energy (BLE) protocol used for the data transactions between the mobile device and the wearable sensors. mDurance performs a Bluetooth scan to detect available wearable devices and pairs then with the mobile phone. This functionality is also implemented by using the Android class BluetoohAdapter and BluetoohDevice, which builds on the standard Android API (100). The circular buffer is defined as an ArrayList. Its size is four times the number of samples to store. Thus, the buffer stores four blocks of samples. The number of samples to stored can be set in order to make the buffer more suitable for each situation. The handler is in charge of to receive and handler the messages sent by the Shimmer driver. Depending on the message received, the handler will make a different task. When the handler receives a message with a sample, first it is checked to see if it is the first one. If so, an offset timestamp and the beginning of the session are set to this moment. Then,

the framework data structure (which is stored in the buffer), is filled with the sample values. Finally, a data structure reference is sent to the handler of the Communication Manager.

The most important functions in this class are *connect/disconnect* and *start/stop-streaming*.

connect and *disconnect* are not complex functions since the hard work is done by the Android OS and the intermediate driver.

StartStreaming is a function that requires more work since it is necessary to initialize the buffer. To do this, first the enabled sensors are consulted, then the appropriate data structure is created and the buffer fills with that data structure. Finally, the intermediate driver's *startStreaming* function is called and session's metadata are set.

StopStreaming function calls to the intermediate device's *stopStreaming* function and updates sessions metadata. The driver also has to implement the other functions of the interface *Device* mentioned previously.

4.5.2.1 How we use it in mDurance

As it was said before, this class (Communication.class) is a singleton. It means that first, to can work with all the functionalities of this class it is necessary to get its instance like this:

```
CommunicationManager cm = CommunicationManager.getInstance();
```

To add devices to the manager, one just needs to call the appropriate functions:

```
cm.AddDeviceShimmer(getApplicationContext(), "Device Shimmer", true);
```

For example, in mDurance we had to add the 2 devices required to the proper functioning of the application: *Device IMU* and *Device EMG*.

Once they have been added, they need to be connected. This work is done by the function *Connect*, giving as parameter the mac address:

```
cm.connect("Device Shimmer", address);
```

Before the device starts to stream, the enabled sensors of the devices can be set. In our case, we had to enable the next sensors:

```
ArrayList<SensorType> sensors = newArrayList<SensorType>();
sensors.add(SensorType.ACCELEROMETER);
sensors.add(SensorType.MAGNETOMETER);
sensors.add(SensorType.GYROSCOPE);
```

```
cm.writeEnabledSensors("Device Shimmer", sensors);
```

After all these things are done, it is the suitable moment to start (and stop whenever is wanted) to stream:

```
cm.startStreaming("Device Shimmer");
cm.stopStreaming("Device Shimmer");
```

4.5.3 Computation algorithm of Euler angles

In this section we show how to obtain the *Euler* angles of the Shimmer device that mDurance uses to assess the test, using the Accelerometer, Gyroscope and Magnetometer. We remember that the orientation algorithm used was proposed in the (92). Originally, Shimmer only included the *GradDes3DOrientation* class into the Shimmer driver. This class is composed by a *Quaternion* class into to the main class and some configuration variables as *mSampligPeriod* and *mBeta*. It is created 4 double variables that represent the future output quaternions. Thus, the output of the algorithm is a quaternion, a compact vector in the form (q_1, q_2, q_3, q_4), which dynamically represents the orientation of the sensor.

```
public class GradDes3DOrientation {

double mBeta = 1;
double mSamplingPeriod = 1;
double q1, q2, q3, q4;
public GradDes3DOrientation(double beta, double samplingPeriod, double q1,
    double q2, double q3, double q4) {
        mBeta = beta;
        this.q1 = q1;
        this.q2 = q2;
        this.q3 = q3;
        this.q4 = q4;
        mSamplingPeriod = samplingPeriod;
    }

    public class Quaternion{

        public double q1, q2, q3, q4;
```

```
        public Quaternion(double q1, double q2, double q3, double q4)
        {
            this.q1 = q1;
            this.q2 = q2;
            this.q3 = q3;
            this.q4 = q4;
        }
    }
}
```

Finally, the main class is also composed by 2 main functions:

- *update*. This is the function which represents the Madgwick's algorithm. It is used to calculate the orientation sensors based in quaternions. This function was created by Shimmer and is included in its driver package. Such as we commented, we have to introduce the tri-accelerometer, tri-gyroscope and tri-magnetometer data as input parameters.

```
public Quaternion update(double ax,double ay,double az,double gx, double
    gy, double gz, double mx, double my, double mz) {
        double norm;
        double hx, hy, _2bx, _2bz;
        double s1, s2, s3, s4;
        double qDot1, qDot2, qDot3, qDot4;

        double _4bx, _4bz;
        double q1q1, q1q2, q1q3, q1q4, q2q2, q2q3, q2q4, q3q3, q3q4, q4q4,
            _2q1q3, _2q3q4;

        _2q1q3 = 2.0 * q1 * q3;
        _2q3q4 = 2.0 * q3 * q4;
        q1q1 = q1 * q1;
        q1q2 = q1 * q2;
        q1q3 = q1 * q3;
        q1q4 = q1 * q4;
        q2q2 = q2 * q2;
        q2q3 = q2 * q3;
        q2q4 = q2 * q4;
        q3q3 = q3 * q3;
        q3q4 = q3 * q4;
        q4q4 = q4 * q4;
```

```
// Normalise accelerometer measurement
norm = Math.sqrt(ax * ax + ay * ay + az * az);
if (norm > 0.0){
   norm = 1.0 / norm;
   ax *= norm;
   ay *= norm;
   az *= norm;
}
else{

}
// Normalise magnetometer measurement
norm = Math.sqrt(mx * mx + my * my + mz * mz);
if (norm > 0.0){
   norm = 1.0 / norm;
   mx *= norm;
   my *= norm;
   mz *= norm;
}
else{

}
hx = mx * q1q1 - (2 * q1 * my) * q4 + (2 * q1 * mz) * q3 + mx *
    q2q2 + (2.0 * q2) * my * q3 + (2.0 * q2) * mz * q4 - mx * q3q3
    - mx * q4q4;
hy = (2 * q1 * mx) * q4 + my * q1q1 - (2 * q1 * mz) * q2 + (2 * q2
    * mx) * q3 - my * q2q2 + my * q3q3 + (2.0 * q3) * mz * q4 - my
    * q4q4;
_2bx = Math.sqrt(hx * hx + hy * hy);
_2bz = -(2 * q1 * mx) * q3 + (2 * q1 * my) * q2 + mz * q1q1 + (2 *
    q2 * mx) * q4 - mz * q2q2 + (2.0 * q3) * my * q4 - mz * q3q3 +
    mz * q4q4;
_4bx = 2 * _2bx;
_4bz = 2 * _2bz;

// Corrective step
s1 = -(2.0 * q3) * (2.0 * q2q4 - _2q1q3 - ax) + (2.0 * q2) * (2.0
    * q1q2 + _2q3q4 - ay) - _2bz * q3 * (_2bx * (0.5 - q3q3 -
    q4q4) + _2bz * (q2q4 - q1q3) - mx) + (-_2bx * q4 + _2bz * q2)
    * (_2bx * (q2q3 - q1q4) + _2bz * (q1q2 + q3q4) - my) + _2bx *
```

```
        q3 * (_2bx * (q1q3 + q2q4) + _2bz * (0.5 - q2q2 - q3q3) - mz);
    s2 = (2.0 * q4) * (2.0 * q2q4 - _2q1q3 - ax) + (2.0 * q1) * (2.0 *
        q1q2 + _2q3q4 - ay) - 4.0 * q2 * (1.0 - 2.0 * q2q2 - 2.0 *
        q3q3 - az) + _2bz * q4 * (_2bx * (0.5 - q3q3 - q4q4) + _2bz *
        (q2q4 - q1q3) - mx) + (_2bx * q3 + _2bz * q1) * (_2bx * (q2q3
        - q1q4) + _2bz * (q1q2 + q3q4) - my) + (_2bx * q4 - _4bz * q2)
        * (_2bx * (q1q3 + q2q4) + _2bz * (0.5 - q2q2 - q3q3) - mz);
    s3 = -(2.0 * q1) * (2.0 * q2q4 - _2q1q3 - ax) + (2.0 * q4) * (2.0
        * q1q2 + _2q3q4 - ay) - 4.0 * q3 * (1.0 - 2.0 * q2q2 - 2.0 *
        q3q3 - az) + (-_4bx * q3 - _2bz * q1) * (_2bx * (0.5 - q3q3 -
        q4q4) + _2bz * (q2q4 - q1q3) - mx) + (_2bx * q2 + _2bz * q4) *
        (_2bx * (q2q3 - q1q4) + _2bz * (q1q2 + q3q4) - my) + (_2bx *
        q1 - _4bz * q3) * (_2bx * (q1q3 + q2q4) + _2bz * (0.5 - q2q2 -
        q3q3) - mz);
    s4 = (2.0 * q2) * (2.0 * q2q4 - _2q1q3 - ax) + (2.0 * q3) * (2.0 *
        q1q2 + _2q3q4 - ay) + (-_4bx * q4 + _2bz * q2) * (_2bx * (0.5
        - q3q3 - q4q4) + _2bz * (q2q4 - q1q3) - mx) + (-_2bx * q1 +
        _2bz * q3) * (_2bx * (q2q3 - q1q4) + _2bz * (q1q2 + q3q4) -
        my) + _2bx * q2 * (_2bx * (q1q3 + q2q4) + _2bz * (0.5 - q2q2 -
        q3q3) - mz);

    norm = 1.0 / Math.sqrt(s1 * s1 + s2 * s2 + s3 * s3 + s4 * s4);
    s1 *= norm;
    s2 *= norm;
    s3 *= norm;
    s4 *= norm;

    // Compute rate of change of quaternion
    qDot1 = 0.5 * (-q2 * gx - q3 * gy - q4 * gz) - mBeta * s1;
    qDot2 = 0.5 * (q1 * gx + q3 * gz - q4 * gy) - mBeta * s2;
    qDot3 = 0.5 * (q1 * gy - q2 * gz + q4 * gx) - mBeta * s3;
    qDot4 = 0.5 * (q1 * gz + q2 * gy - q3 * gx) - mBeta * s4;

    // Integrate to yield quaternion
      q1 += qDot1 * mSamplingPeriod;
    q2 += qDot2 * mSamplingPeriod;
    q3 += qDot3 * mSamplingPeriod;
    q4 += qDot4 * mSamplingPeriod;
      norm = 1.0 / Math.sqrt(q1 * q1 + q2 * q2 + q3 * q3 + q4 * q4);

    q1 = q1 * norm;
```

```
        q2 = q2 * norm;
        q3 = q3 * norm;
        q4 = q4 * norm;

        return new Quaternion(q1,q2,q3,q4);
    }
```

- *quattoeuler*. It is created to convert the orientation of the sensor mentioned before in another output vector with the *Euler* angles (*Pitch, Roll, Yaw*)

```
public double[] quattoeuler(double q0, double q1 , double q2, double
    q3){
   double[] euler = new double[3];

        euler[0]=Math.atan2(2*(q0*q1+q2*q3),(1-(2*(q1*q1+q2*q2))));
        euler[0]= (euler[0]/Math.PI*180);
        euler[1]=Math.asin(2*(q0*q2-q3*q1));
        euler[1]= euler[1]/Math.PI*180;
        euler[2]=Math.atan2(2*(q0*q3+q1*q2),1-(2*(q2*q2+q3*q3)));
        euler[2]= euler[2]/Math.PI*180;
        return euler;
}
```

4.5.3.1 How we use it in mDurance

Consequently, we are going to explain the steps necessary to use the previous functions in mDurance.
At first, when we would wanted to introduce the new "virtual" sensors (Pitch, Roll and Yaw) into the Shimmer driver, we had to modify the *ShimmerObject* class in the next manner: the *MAX_NUMBER_OF_SIGNALS* represents the maximal size of the vector of signals which the sensors signal are saved. Originally, this number was 40 but when we introduced the new sensors:

```
int MAX_NUMBER_OF_SIGNALS = 43;
```

The next step was define the new conditions to enable the new sensors. In this case, it was created a new flag called *mEulerEnable* which is true when the new Euler sensors are activated. Thus, when the Euler flag and accelerometer, gyroscope and magnetometer are enabled, we defined a new *GradDes3DOrientation* object where to

initialize the first default orientation defined by the next vector of quaternions: [1 0 0 0] and sampled by β. This value can be defined between 0 and 1. It represents the divergence rate. It is expressed as the magnitude of a derivative of the quaternion corresponding to the gyro measurement error. From the simulation it is obtained that the best accuracy is obtained with a value of 0.4.

```
if ((mEnabledSensors & SENSOR_ACCEL) > 0 && (mEnabledSensors & SENSOR_GYRO) >
    0 && (mEnabledSensors & SENSOR_MAG) > 0 && mEulerEnable ){
   if (mEnableCalibration){
      if (mEulerAlgo==null){
       mEulerAlgo = new GradDes3DOrientation(0.4, (double)1/mSamplingRate, 1,
           0, 0,0);
}
```

Now, we define a new *Quaternion* object to use the function *update* which will refresh the default vector of quaternions with the IMUs data which mDurance received.

```
Quaternion q = mEulerAlgo.update
    accelerometer.x,accelerometer.y,accelerometer.z,
    gyroscope.x,gyroscope.y,gyroscope.z,
    magnetometer.x,magnetometer.y,magnetometer.z);
```

Finally, we created the double variables where the Euler angles are stored. Then, we called the function *quattoeuler* to convert the quaternion in the wish angles and saved them in a vector of angles. We stored all these data in the cluster structure that Shimmer used in its driver.

```
double pitch, roll, yaw;
double [] euler = new double [3];
euler = mEulerAlgo.quattoeuler(q.q1, q.q2, q.q3, q.q4);
pitch = euler[0];
roll = euler[1];
yaw = euler[2];

objectCluster.mPropertyCluster.put("Pitch",new
    FormatCluster("CAL","deg",pitch));
objectCluster.mPropertyCluster.put("Roll",new
    FormatCluster("CAL","deg",roll));
objectCluster.mPropertyCluster.put("Yaw",new FormatCluster("CAL","deg",yaw));
   }
}
```

4.5.4 Data structure

Here is defined the framework's data structure. This is design taking into account that it has to be very flexible because of the wide range of devices that exitsexists and the variety of sensors. It also has to be scalable because in the future other devices with new features or sensors can be released. The data structure is composed of four objects (classes) and one enumeration Figure 4.23:

- Data. It is composed by a double. It represents the value return by the sensor.
- ObjectData. This class stores the device name and a Hashtable where the type of sensor is the key and Data is the object. It represents a complete sample sent by the device. It means that if the sample rate is 50Hz, 50 ObjectData will be generated in 1 second.
- Metadata. This class is composed by a String. It represents the units of measure used by the devices.
- ObjectMetadata. This class is composed by a Hashtable where SensorType (this is not a class, but an enumeration. It represents the type of sensor) is the key, and Metadata is the object. It also contains two Time variables, two Int variables, and one Double variable. The Hashtable is filled with the enabled sensors of the device and their units of measure. The two Time variables represent when the device starts to stream and when it finishes. The two Int variables represent the first index of the database table which belongs to this device during a session, and the last index of this session. The period of time that goes from when the device starts to stream until the streaming is stopped is called session. Finally, the Double variable represents the devices sample rate (measured in Hz) during a determinate session.

4.5.4.1 How we use it in mDurance

The mHealthDroid Adapters for Shimmer2 (EMG) and Shimmer3 (IMU) wearable devices are used to communicate these devices with the mobile phone and to map their data to the proprietary format. In this manner, the registered EMG potential and the triaxial acceleration, rate of turn and magnetic field samples are made available to the diverse components of the application. In order to explain step by step as we use the data structure of mHealthDroid in mDurance, we are going to begin describing the function to establish the communication between mDurance and mHealthDroid. A

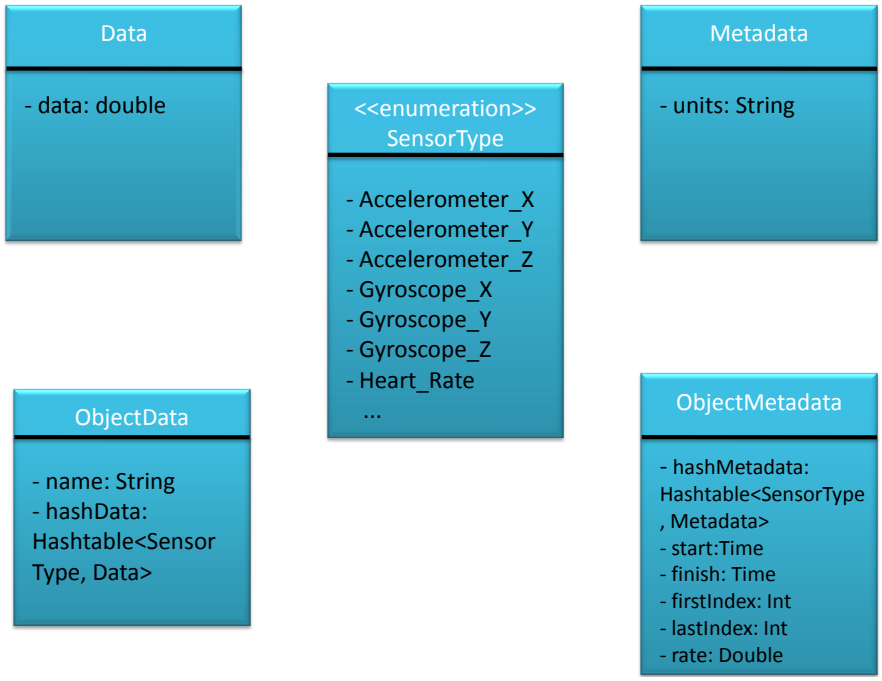


Figure 4.23: Class diagram of the framework data structure

data structure reference is sent to the handler of the Communication Manager. This is with the function:

```
cm.setHandlerTest(Handler);
cm.sendDataToTest("Device IMU", true);
cm.sendDataToTest("Device EMG", true);
```

The data collection is the most important function for mDurance. For it, we use the data structure proposes by mDurance to obtain the information and use it to calculate and display all the parameters and results of the assessment of the test. For this, the first step is identify from which sensor is the package receiver. It is possible with the class *ObjectData* and the field *name* which allows to know the package's source:

```
objectData.name.equals("Device IMU");
objectData.name.equals("Device EMG");
```

When mDurance identifies the package's origin, it saves the data value in separate variables to can assess the different parameters of the test. Then, we obtain the data value thought field *data* and *SensorType*:

```
Data angle = ObjectData.hashData.get(SensorType.EULER_ROLL);
```

```
Double angleDouble = (Double) rollD5.data;

Data emg = ObjectData.hashData.get(SensorType.EMG);
Double emgDouble = (Double) emg.data;
```

Finally, we only are left calculate and show all the different parameters which mDurance works.

4.5.5 Storage implementation

The mDurance storage functionality builds on top of the mHealthDroid Storage Manager, which provides high level of abstraction from the underlying storage technology and enables data persistence both locally and remotely. In the current implementation, the mDurance application stores data locally on a SQLite database (**?**) deployed on the mobile phone SD card. However, the mHealthDroid Storage Manager also provides remote storage capabilities which could enable the easy extension of the current mDurance application to store data on the cloud. Therefore, this module offers all the functionalities to manage everything related with the internal database. At first, it is designed a database composed by 4 tables: Table_MacAddress, Metadata_MacAddress, TypeDevice_Units and User_Profile. When Table_MacAddress and Metadata_MacAddress tables are created, the part "MacAddress" is replaced by the device's MAC address to make the table name unique, easy to identify and relate this with its corresponding device. When the Units tables are created, the part "TypeDevice" is replaced by the type of device, in this case, Shimmer. For mDurance, we had to design a new database to manage everything related with the specialist's data and the results of the tests from the patients associated with each and the register with all data transmitted by the sensors. This new database is called "'UserDB"'. It contains 3 tables to store all the information mentioned before:

- *Table_Specialist*. This table stores the main data about the specialists: in order to identify each specialist, a unique ascendent key is associated to each one when they are added to the database, called *KeyId*. The userName and the password to access to the main menu of the application and the specialist's email to can remember their both the userName and the password. In the Figure 4.24 is displayed the expert table with some specialist's personal data and their information to register their self in the system.

- *Specialist_Name*. This table is created to store the personal information of each

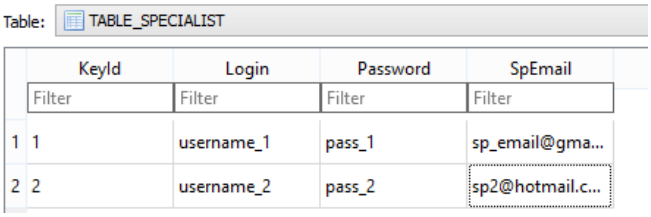


Figure 4.24: Table of data specialists from the internal database: *UserDB*.

patient associated with each specialist who is registered in the previous table. In addition, in this table is stored the results of each test that the patients have done previously: times and EMG values. Finally, it should be pointed that in order to identify each patient, a unique ascendent key is associated to each patient when they are added to the database, called *Key_Id*. It is important to mention that this "key" is used to get any personal information or kind of results of the each patient individually. We explain it in the next "'how we use it in mDurance"'. In the Figure 4.25 is displayed the specialist table with some patient's personal data and their test's results.

- *Table_Sensors*. Finally, in this table is stored all data which have been transmitted by the sensors, both angles and electrical values of each sensors respectively. In this table is also introduced 2 fields additionally, "Label" to signal the test where the data come and "Patient's name" to indicate the patient from whom the data come. In the Figure 4.26 is displayed the sensors table with some information about the received data from a specific patient, test and date.

The module is composed by 3 classes initially, but for the mDurance design we had to include a new class additionally. Finally the module is composed by 4 classes:

DataBaseHelper

This is a class specifically created for helping carry out the management role such as create, open, close or update the database.

DBAdapter

	KeyId	Date	Name	Age	Sex	Email
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2015-03-20 10:0...	Juan Antonio ...	28	male	jamoren@decs...
2	2	2015-03-20 10:3...	Manuel Parra	34	male	manuparra@g...
3	3	2015-03-20 11:2...	Rafael Cabañas	27	male	rafacabanas@g...
4	4	2015-03-20 11:5...	Sara del Río	31	female	srio@decsai.ugr...
5	5	2015-03-20 12:3...	Francisco Díaz	37	male	fjrodriguezdiaz...
6	6	2015-03-20 15:2...	pablo gutierrez	28	male	thror.el.enano...
7	7	2015-03-20 15:4...	Alejandro rosales	28	male	alektronicks@g...
8	8	2015-03-20 16:0...	Sarah vluymans	23	female	sarah.vluymans...
9	9	2015-03-20 16:1...	cristian rivera	21	male	riveraperegrina...
10	10	2015-03-20 17:0...	José moral	26	male	moralmunoz@...

(a)

orationAngleSoren	otalTimeSorensen	T1Sorensen	T2Sorensen	RMS_S	ARV_S	MVCSorensen
Filter	Filter	Filter	Filter	Filter	Filter	Filter
24.675704956054 7	32	26	6	1.086518758457 23	0.042136059632...	0.213919416069...
19.339391708374	108	74	34	5.549362158909 98	0.135166139015 88	0.824420034885...
34.648300170898 4	59	51	8	2.166670133494 92	0.072946125938...	0.458119660615...
10.873064041137 7	123	38	85	5.632431924001 27	0.132019769805...	0.961172163486...
-17.301359176 6...	60	36	24	2.777097902618 72	0.091797395685...	0.585103809833...
-29.765701293 9...	99	95	4	4.218755513603 39	0.110535185416...	0.780463993549...
16.669509887695 3	105	75	30	5.173708755257 79	0.136688786947...	0.744322359561 92
23.063898086547 9	117	91	26	6.391006173089 78	0.160464934682...	1.404639840126 04
-16.065797805 7...	85	29	56	4.792603949890 1	0.134094211168...	1.001221060752 87
22.607089996337 9	52	38	14	2.321808143720 51	0.083657755664...	0.602686226367 95

(b)

librationAngleFle	TotalTimeFlexor	RMSFlexor	ArvFlexor	MVCFlexor
Filter	Filter	Filter	Filter	Filter
59.4996376037598	66	0.889427867137...	0.021883067663...	0.532356560230...
60.2792053222656	148	0.589232411151...	0.007486122630...	0.037118438631...
59.7345237731934	74	1.42989525059382	0.039156946291...	0.354578763246...
60.1562805175781	99	2.357060813955 14	0.058599051240...	0.667155086994...
-60.8786849975...	59	1.314983136377 07	0.036371747402...	0.348717957735...
-60.5080223083...	89	0.484764435059...	0.007855374232...	0.032234434038...
63.4901313781738	94	2.199032370920 36	0.056517244696...	0.667155086994...
60.9394416809082	79	1.263455420239 85	0.033459652792...	0.396581202745...
-60.9533195495...	144	1.963362003169 76	0.038675251350...	0.635897457599 64
62.0631332397461	71	0.983224007500...	0.022010716773...	0.336019545793...

(c)

brationAngleBLRi	TotalTimeBLRight	RmsBLRight	ArvBLRight	MVCLateralRight
Filter	Filter	Filter	Filter	Filter
33.5270233154297	25	0.8248409412773	0.033456221594...	0.238339439034 46
34.2716293334961	69	2.766990917207 49	0.078362811768...	1.2288156747818
23.9881225378264	20	0.924310295276 32	0.029712220594...	0.373138739284 45
28.8566322326668	44	3.681338203275 64	0.075801764625...	1.987789988517 76
23.1440092132262	36	2.481109062388 24	0.061844127559...	1.326917900088 16
23.5551089201472	38	2.737401683455 44	0.045729902420...	1.242628956604 68
22.4346188420962	52	2.764601385642 82	0.057584215679...	0.615506726503 37
20.0274028778076	46	3.793116629570 71	0.131507112693...	1.987789988517 76
23.5266748824904	62	3.486409296720 94	0.078804213268...	1.607853545908 79
26.1098136901855	17	1.737401013457 44	0.077377721115...	0.608547031879 42

(d)

Figure 4.25: Table of patient's data from the internal database: *UserDB*: (**a**) Personal information; (**b**) STEET results; (**c**) TCSET results; and (**d**) SBET Right results

All the queries needed are made in this class with respect to DB. All of the functions used to create the databases tables include the following: *createShimmerTable, createShimmerMetadataTable, createShimmerTableUnits, createMobileTable, createDeviceMetadataTable, createMobileTableUnits, createUserProfileTable.* In order to insert all the data storage in these tables the following are used: insertShimmerSignal, insertShimmer Metadata, insertMobileSignal, insertMobileMetadata, fillMetadataRow, fillTableShimmerUnits, fillTableMobileUnits. In addition, there are a wide range of functions to retrieve any kind of data from database such as *retrieveInformationById, getMaxIndex, getColumnsName, getShimmerUnitsTable, getSessionsIds.*

DBUsersAdapter

This is the new class which has been added to the module. All the queries needed are made in this class with respect to *UserDB*. All of the functions used to create the databases tables include the following: *createTableSpecialist, createUserPro-*

	id	Name	Label	Date	Angles	Emg
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Juan Antonio ...	Sorensen1	2015-03-20 10:0...	31.6157283782959	-0.03125763311...
2	2		Sorensen1	2015-03-20 10:0...	31.81227684021	0.118192918598...
3	3		Sorensen1	2015-03-20 10:0...	31.7547168731689	0.040048841387...
4	4		Sorensen1	2015-03-20 10:0...	31.5934658050537	0.069352872669...
5	5		Sorensen1	2015-03-20 10:0...	31.3079032897949	0.025396825745...
6	6		Sorensen1	2015-03-20 10:0...	31.2656860351562	-0.05470085516...
7	7		Sorensen1	2015-03-20 10:0...	31.6044483184814	-0.01953602023...
8	8		Sorensen1	2015-03-20 10:0...	31.8155212402344	-0.03321123495...
9	9		Sorensen1	2015-03-20 10:0...	31.8109188079834	-0.00976801011...
10	10		Sorensen1	2015-03-20 10:0...	31.7661361694336	-0.06251526623...
11	11		Sorensen1	2015-03-20 10:0...	31.71657371521	0.022466422989...
12	12		Sorensen1	2015-03-20 10:0...	31.7638034820557	0.037118438631...
13	13		Sorensen1	2015-03-20 10:0...	31.7057132720947	0.019536020234...
14	14		Sorensen1	2015-03-20 10:0...	31.7752056121826	-0.05079365149...
15	15		Sorensen1	2015-03-20 10:0...	31.9058723449707	0.053724054247...

Figure 4.26: Table of data sensors from the internal database: *UserDB*.

fileTableApp, createTableSensor. In order to insert all the data storage in these tables the following are used: *insertSpecialistProfile, insertUserProfile, insertSensorSignal*. In addition, there are a wide range of functions to retrieve, update, insert or check any kind of data from database such as *checkPassword, getTimes, getEMG, getPasswordByEmail, insertParametersSorensen, etc*.

Storage

This class could be defined as the upper layer of this module. Thus, it is an intermediate class between low level databases definitions and the manager. For some tasks, functions just get the input values and call to the DBAdapter and DBUsersAdapter functions: *getMobileUnitsTable, getShimmerSignalsTable, consistencyTestMetadata, etc* for the first class and *getTestParameters, getSpecialistData, getPatientData* for the new class. However for other tasks, functions get the input values, adapt them, and then call to the DBAdapter and DBUsersAdapter functions: *insertShimmer, insertShimmerMetada, insertMobile, insertMobileMetadata* for DBAdapter and *insertSpecialistProfile, insertUserProfile, insertTestParameters* for DBUsersAdapter.

4.5.5.1 How we use it in mDurance

If one wants to storage data into the databases (myDB and UserDB respectively), the variable storage must be initialized this way:

```
cm.CreateStorage(getApplicationContext());
cm.CreateUserStorage(getApplicationContext(), true);
```

To insert a new specialists profile there are a couple of things to consider. The database must be open and the table created. All of this is possible by doing:

```
cm.openUserDatabase();
cm.createSpecialistTableApp();
cm.addSpecialistProfile(username, password, email);
cm.closeUserDatabase();
```

In this moment, a new table with the specialist's userName has been created. To insert a new users profile which is related with the previous specialist, we have to consider that, the database is open and the table created. All of this is possible by doing:

```
cm.openUserDatabase();
cm.createUsersTableApp(table_specialist_userName);
cm.addUserProfile(table, name, age, height, weight, email, gender, condition);
cm.closeUserDatabase();
```

Depending of the functionality which is required, mDurance through Communication Manager will use the defined functions to store or to get the different results obtained in the assessment of each test. For example, some typical functions are:

```
cm.insertUserParametersTest(table, key_id, totalTime, rmsEMG, arvEMG, MVC,
    angleCalibration);
cm.getTimesTest(table, "patient email");
cm.getRmsTest(table, "patient email");
cm.getArvTest(table, "patient email");
cm.getMvcTest(table, "patient email";
fillMatrix(angles, emg);
```

It is noteworthy this last function *fillMatrix()*. We use it to store all the data which the sensors have transmitted during the assessment of each test. This function stores this information in the *Table_Sensor* mentioned previously. We are going to detail it a little more:

```
public void fillMatrix(ArrayList<Double> angles, ArrayList<Double> emg){

    double[][] data;
    int size;
     if(angles.size() <= emg.size())
        size = emg.size();
     else
        size = angles.size();
     dates = new double[size][2];
     for(int m=0; m < angles.size(); m++)
        dates[m][0] = angles.get(m);
     for(int n = 0; n < emg.size(); n++)
        dates[n][1] = emg.get(n);
     cm.openUserDatabase();
     cm.insertSensors(data, "Patient name", "Label", "Table_Sensor");
     cm.closeUserDatabase();
     bdAngles.clear();
     bdEMG.clear();
     index1 = 0; index2 = 0;
        }
```

At first, the matrix *data* is created to store both angles and emg values of multi-dimensional form. The matrix's size depends of the maximal size of each individual ArrayList. Then, when the size is selected, it proceed to download each Array in the corresponding matrix. Finally, we call to the function *cm.insertSensors()* which fill the table from the database with the data, a label and the patient's name. The indexes are the corresponding index of each Array (angle and emg). It should be pointed that this function is called each 800 samples during the assessment of the test or when the test is finished.

4.5.6 Visualization implementation

mHealthDroid Visualization Manager supports diverse modes and ways to display data and builds on the open source library Graphview (101). Since Android does not provide a good API for making charts, it was decided to look for a library which could help us with this task. It has some advantages like the fact that it is a free, open source (anything can be changed in case it is demanded), and easy to use. It is also possible to implement online visualization.

Graphview library

GraphView library has five important classes:

- GraphView: This is the main class. It is abstract and extends from LinearLayout. This object represents the frame of the graph and almost everything is defined and implemented here.
- GraphViewData: It is an inner class defined inside of the *Graphview* class. It has only two variables: valueX and valueY. This object represents a point in the graph and their variables are the coordinates.
- GraphViewSeries: All the 2D graphs are composed by a set of coordinates (Y axis and X axis). This class represents the sets of data to be drawn. The set of data is defined as an array of *GraphViewData*. It is possible to choose the color and thickness of the series as well as to store a little description of it. This class also implements functions to either append data or reset them.
- LineGraphView: This class extends from the *GraphView* class, and it is only in charge of painting specifically a line graph.
- BarGraphView: This class extends from the *GraphView* class and it is only in charge of painting specifically a bar graph.

For the deal between the manager and the library, mHealthDroid uses the class *Plot*. The class *Plot* can be defined as a middle layer, and it represents a graph with series. It has three variables: a *GraphView* object and two Hashtables. The *GraphView* object is the object necessary to use the library. One of the Hashtable is made to store the list of the series belonging to the graph, a list of *GraphViewSeries* objects. The other Hastable is made to store the counter of each series. These counters are the values of the highest X coordinates of the graphs. The class defines an enumeration: *GraphType*. This enumeration is made to chose what king of graph is desired, either line graph or bar graph.

There are some methods that modify the graph and the class variables in some way, and call the library functions:

- *addSeries*. It receives the series name and the data to be represented and converts them into the coordinates of the graph by creating an array of *GraphViewData*. Then, a counter with the number of coordinates is stored, the series is created,

and added to the graph. The function returns either True in a case of success or False in a case of not success. There is another version of this function, which receives as parameters the color and the thickness of the series.

- *appendData*. This receives the series name, the value to be inserted at the end of the series, and a Boolean indicating whether or not the graph must scroll to the last value. The function increments its counter and adds the value in the end of the series.

- *appendData2*. It is a function that was necessary to implement for the online visualization. It is actually defined in the *GraphViewSeries* class and introduces a value at the end of the series. But unlike *appenData*, this function does not increase the size of the array with the values. It moves the values one position to the left in the array and introduces the new value in the last position. It could be expressed as follows: X[0] = X[1], X[1] = X[2], ... X[n-1] = X[n], X[n] = NewValue.

- *resetData*. It receives the series name and an ArrayList with the new values. The function creates a new array with the values given and replaces the current array for the new one. Finally it replaces the counters.

- *removeSeries*. This function removes a single serie which is selected.

- *removeAllSeries*. It removes all the series.

The rest of the methods of this class (*setHorizontalLabels, setViewPort, setScalable, isScrollable, getLegendAlign, etc*) are made to mask the most relevant functions of the library. That is, the methods only call to the library functions.

The Visualization Manager is used to make a representation of any kind of data, but overall, to make possible the online visualization of the data received mentioned in previous sections. It extends from *Activity* (the why will be explained later) and has four variables: an instance of itself since it is a singleton class, an instance of the Communication Manager in order to get the data sending by the sensors, a Hashtable which contains all the graphs, and finally, an array with eight different colors.

For the implementation of the online visualization, and in a similar way of how it was done in the communication manager, a new object is observed: *ObjectVisualization*. This object is an inner class which has three variables: a *Plot* object, an ArrayList and an Int. The *Plot* object is the graph, the ArrayList is a list of the device sensors to be

visualized, and the Int is a counter. The last two variables are only used for the online visualization.

As it happens with all the managers, the function *getInstance* must be used to get the unique instance of the manager in order to use it. Creating the graph is the first thing to do in order to visualize the data. To do that, this manager offers the function *addGraph*, which needs as parameters the name of the graph (it must be unique), the type of the graph, and the application context. Another important function in this topic is *paint*, which needs as parameter the graph name and a LinearLayout. What this function does is add the customized graph to the LinearLayout in order to be painted. Taking that into account, this function should be called at the end, when everything in the graph is already set. Once the graph is created, to make the offline visualization or create simple graphs, the manager offers the *addSerie* function. This function needs as parameters the name of the graph, the name of the series, and the data to be represented. The function checks whether the graph exits and call the function *addSeries* of the class *Plot*. The *appendData* function is also interesting for the statics graphs since it allows values to be added at the end of the series. It receives as parameters the graph name, the series name, the value, and a *Boolean* indicating if the graph must scroll to the end. The function first checks whether the graph exists and then calls the *appendData* function of the *Plot* class.

To make the online visualization, it is used the same approach for the Communication Manager. That is, the data received by the communication manager is sent to the visualization manager by messages. The *ObjectCommunication* of the Communication Manager defines an ArrayList with the names of the graphs which will represent the data. A flag is also defined, which is activated whenever the data must be sent to this manager. In the Visualization Manager, it is defined a *Handler*, which receives all the messages. Since the manager is going to change the user interface, the code of the Handler is defined inside of the *runOnUiThread* function. This function makes the code inside of it run in the thread of the user interface. But the function is defined in the *Activity* class. This is why the manager inherits from this class. The messages sent by the Communication Manager contain an *ObjectData*. This *ObjectData* is the sample sent by the device to the Communication Manager. The device name is retrieved from the *ObjectData* and with it, the *ObjectCommunication* is retrieved from the Communication Manager. The list with the names of the graphs, which will represent any data of the sample, is retrieved from the *ObjectCommunication*. Then, for each graph in the list, the sample is checked to see if there are any values of the sensors to be represented. In case there are, the new value is added to its corresponding series. There are two

cases when the data are added and it depends on the viewport. When the number of samples received is below of the viewport, the function *appendData* is used. Otherwise, the function *appendData2* is used. Finally, the counter of the samples received is incremented, and the graph is redrawn.

To start the online visualization, the manager offers the function *visualizationOnline*. It receives the graph name, the device name, and a list of the sensors which will be represented in the graph. The function creates a series for each sensor given and activates the flag of the *ObjectCommunication* for sending the data to the VisualizationManager. It is important to note that the data of a device can be handled by different graphs, but a graph can NOT handle the online visualization of different devices.

To stop the online visualization, the manager offers the function *stopVisualizationOnline*. It receives the graph name and the device name. The function removes the graph name from the list of the graph defined in its corresponding *ObjectCommunication*. Then, the flag for sending the data to the visualization manager is deactivated in case the list of the graph is empty.

The functions *removeSerie* and *removeAllSerie* remove a particular series or all of the series respectively from statics graphs as well as from a graph with online data visualization. It does this by calling to their homonyms functions in the class *Plot*. The rest of the functions of this manager, *setScalable, setVerticalLabels, getLegendAlign, isScrollable, isShowLegend, etc*, are made to set the graph features, and basically, just call to the functions of the library through the class *Plot*.

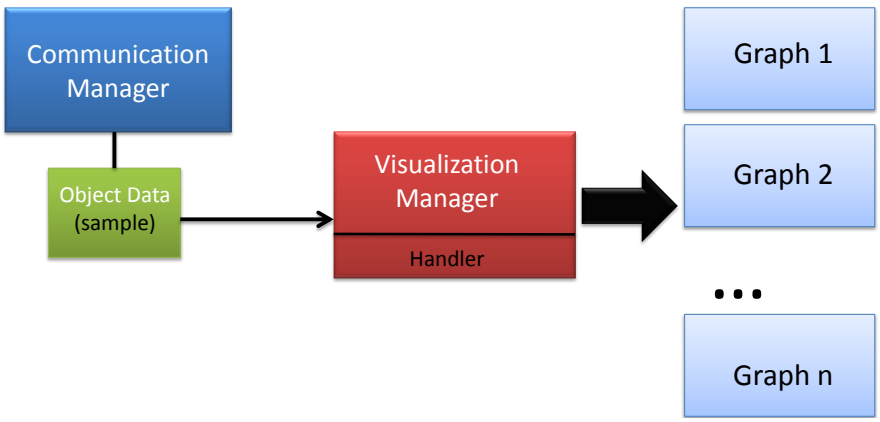


Figure 4.27: Passing message from the Communication Manager to the different graphs

4.5.6.1 How we use it in mDurance

Since the Visualization Manager is a singleton class, first is necessary to get the instance to can use it in all mDurance's class:

```
VisualizationManager vm = VisualizationManager.getInstance();
```

To continue with the process a graph must be created. A unique name to the graph and its type (Line or Bars) must be selected. The UI context must also be indicated. In mDurance, the name of the graph depends of the value or values which the specialist wants to visualize. We remember that these can be from whatever kind of time's values to the three possible muscle fatigue values. mDurance just uses the BAR type of graph to visualize all the values and the unique possible contexts are both the *Test* class and the *Historical Visualization* class. Therefore, to create a graph we use the next function, changing the name and the context uniquely:

```
vm.add("nameGraph", GraphType.BAR, getApplicationContext());
```

mDurance provides graphical representation of online EMG values collected from the wearable device, as well as of the historical endurance test results, for example, the test times and the calculated muscular fatigue values. Now, it depends on the approach desired, either online or offine. Here, both of them are explained.

Offline

The processed endurance test results, which are stored on the permanent storage and provided by the mHealthDroid StorageManager, are represented on a bar diagram in an offline operation manner. The specialist obtained all test results in the "Historical" option of the main menu. To create the graphs of individual values of a single test, for example, to consult all STEET's times of a patient, first we have to create an array with the values of the Y coordinates. In our case, mDurance use the database's functions mentioned previously to get the data to visualize. To introduce the data into the graph, the following procedure must be used:

```
arrayData = getDataConsulted();
vm.addSeries("nameGraph", "serie name", arrayData);
```

If for example, we want to add different kind of series, we can to add a color per series with this function and a description:

```
vm.addSeries("graph", "serie name", array, Color.BLUE, 1, "Description");
```

The graph must be included in a layout in order to be painted. The layout is the space that we reserve to put the graph. The procedure is:

```
LinearLayout graphLayout = (LinearLayout) findViewById(R.id.layout_graph);
vm.paint("graph", graphLayout);
```

Other important aspect are both the horizontal and vertical labels. They are created automatically, but they can be set. mDurance stores, at first, the date of the assessment of the test from each patient in the horizontal labels and the parameter's units (second or mV) in the vertical labels for the single parameter visualization (for exmaple for SCEET's time). On the other hand, mDurance represent the name of each test with the horizontal labels and the same units in the vertical labels when the specialist wants to visualize for example all MVC's values of all the test in a specific day. Therefore, we can define all that like:

```
String[] date = new String[size];
date = getDatesFromInternalDataBase();
vm.setHorizontalLabels("nameGraph", date);
```

Finally, we have implemented a myriad of functions to configure several graph's parameters through a configuration menu. These functions are:

```
vm.setShowLegend("nameGraph", Boolean var);
vm.setLegendAlign("nameGraph", LegendAlign.TOP);
vm.getGraphStyle(nameGraph.setNumHorizontalLabels(int number);
vm.setManualYAxis("nameGraph", Boolean var);
vm.setManualYAxisBounds("EMG Online", 10, -10);
vm.setScrollable("nameGraph", Boolean var);
```

Considering the wishes of the specialist, it can be change all *Boolean* vars according to if the user wants to show the legend (var change true) and indicates its position. The specialist can change the number of horizontal labels. It can be modify if we want to set manually the Y Axis and then, we can set the values of the Y axis. Finally, we want to control if we want to scroll in the graph.

Online

Like in the offline mode, it is necessary to add the graph into a layout:

```
LinearLayout graphLayout = (LinearLayout) findViewById(R.id.layout_graph);
vm.paint("EMG Online", graphLayout);
```

To visualize online data, it is necessary indicate the set of sensors desired. Concretely, in mDurance we only had to add:

```
ArrayList<SensorType> sensorsList = new ArrayList<SensorType>();
sensorsList.add(SensorType.EMG);
```

The *scrollable* feature of the graph MUST be activated. Otherwise online visualization will NOT work:

```
vm.setScrollable("EMG Online", true);
```

Now it is the time to use the function, which does all the work, to visualize the data:

```
vm.visualizationOnline("EMG Online", "EMG Device", sensorsList, (int)
    (2*viewPort));
```

To stop the visualization process, the following method may be used:

```
vm.stopVisualizationOnline("EMG Online", "EMG Device");
```

There are some features that are advisable to set in order to get a better visualization in both modes:
- Make the graph scalable (advisable for offline mode):

```
vm.setScalable("EMG Online", true);
```

- Set the init and the viewport (advisable for online mode):

```
vm.setViewPort("EMG Online", 1, 200);
```

- Set the values of the Y axis (advisable for both modes):

```
vm.setManualYAxisBounds("EMG Online", 10, -10);
```

5

Evaluation

This chapter is divided in 3 main sections. Initially, it is presented in Section 5.1 a description about the tests assessment in a group of males and females. In Section 5.2, is commented the experts suggestions and opinions. Finally, it is also important to know the usability of the system. This is related in Section 5.3.

5.1 Tests evaluation

The proposed mDurance system has been designed taking into account some of the most important limitations faced by practitioners during the course of traditional trunk endurance assessment tests. Thus, in order to show the potential of this system, a preliminary analysis of its reliability and usability has been performed. To that end, ten volunteers, eight males and two females ranging from 21 to 37 years old, were recruited to be evaluated by three external physical therapists using both mDurance and traditional procedures. Before performing the evaluation, the volunteers were informed about the research aims, risks and benefits of participation, and they read and signed an informed consent form, previously approved by the University of Granada Ethical Committee.

The procedures were executed sequentially since a simultaneous evaluation cannot be performed. The reason is that the instructions given by the tester based on visual inspection, for example, finalize the first attempt and start the second chance in the STEET, can influence the normal flow of the decisions made through mDurance and vice versa. The order of procedure was defined in a random manner to minimize the fatigue effects on the performance of the volunteers.

To procure the reproducibility of the tests, a rest time of more than one hour was

considered to ensure the full recovery of the subjects in between the execution of both procedures. It is well-accepted in the physical therapy domain that endurance test results tend to replicate, provided that the subject rests sufficiently in between tests and when these are performed in similar conditions (102). The tests were explained to the subjects before performing the sessions, assuring the full understanding of their phases. Traditional sessions were performed as detailed in Section 3.1, while those involving the use of mDurance were carried out as described in Section 4.4. Accordingly, the execution was similar from the subject perspective, but the expert had to visually determine the start and end of each test and time it using a stopwatch in the traditional approach, while in the use of mDurance these processes were automated.

The first part of this evaluation aims at estimating the inter-rater reliability between the traditional trunk endurance assessment and mDurance. For that purpose, the results of the experiment, for example, times measured for each individual, test and procedure are contrasted (Table 5.1). As it can be observed, the results obtained through both traditional and mDurance methods are generally in line, which reflects the utility of the developed system. However, to support this observation a formal statistical analysis is required. To that end, the Intraclass Correlation Coefficient (ICC)(ρ) (103), the Cronbach's α estimator (104), and the Bland-Altman "limits of agreements" statistic for continuous variables (105) are considered here. These statistics are widely utilized in the clinical domain to evaluate the agreement among two different instruments or two measurements techniques.
One-way random effects ICC(ρ) and its confidence intervals (CI) are calculated for the inter-rater reliability trials. In accordance with previous studies (19, 20), a ICC(ρ) value less than 0.4 reflects poor inter-rater reliability, 0.4 to 0.75 represents fair to good reliability, and more than 0.75 is considered an excellent reliability.

Similarly, a Cronbach's α statistic less than 0.5 is considered unacceptable, 0.5 to 0.6 is poor, 0.6 to 0.7 is questionable, 0.7 to 0.8 is acceptable, 0.8 to 0.9 is good, and a value greater than 0.9 represents an excellent reliability (106).

Finally, the Bland-Altman graphical technique is used to plot the differences between the measurements of the two procedures against their averages, which helps better understand the agreement between both methods (107). The results of the analysis are shown in Table 5.2 and Figure 5.1. SPSS version 21.0 (IBM Corporation, Armonk, NY) is used for all the statistical analysis.
The results of the statistical analysis prove a very high inter-rater reliability for all tests. More specifically, according to the ICC(ρ), the level of agreement between traditional assessment and mDurance can be categorized as excellent for all tests. Based on the

Patient ID	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Age	28	27	34	31	28	37	28	23	26	21
BMI	27.03	23.24	23.91	21.23	21.91	29.94	23.87	22.79	28.63	30.20
STEET (T)	43	56	59	121	104	48	98	123	59	75
STEET (mD)	32	59	108	123	99	60	105	117	52	85
TCSET (T)	42	79	107	112	101	79	118	78	77	154
TCSET (mD)	66	74	148	99	89	59	94	79	71	144
SBET right (T)	30	31	51	38	33	34	52	55	21	39
SBET right (mD)	25	20	69	44	38	36	52	46	17	62
SBET left (T)	26	28	54	46	35	30	35	46	28	25
SBET left (mD)	29	30	72	52	32	34	39	45	18	55
(T) Traditional method. *(mD) mDurance method.*										

Table 5.1: Case study results. BMI values are expressed in kg/m^2 and test duration in s.

Cronbach's α values, the reliability is excellent for the STEET, good for the TCSET and the SBET right, and acceptable for the SBET left. The lower reliability values obtained for the SBET can be originated by multiple factors, such as the subject awareness or lack of concentration (68), and especially the level of fatigue experienced by the users, since this was the last test performed by them. The Bland-Altman graphical analysis shows a scattered distribution of the differences between both procedures that mainly fall within the 95% CI, thus not suggesting the presence of relevant disagreement. Though these results show very promising, a study including a higher number of subjects and applied to patients with LBP would be required to further confirm these findings.

5.2 Expert suggestions

The goal of the second part of this evaluation is to assess the usability and interest of mDurance according to the experts opinion. Thus, after the realization of the tests, the three experts were asked to provide their impressions regarding the use of mDurance. First, they noted the practicality of the automatic angle measurement for initiating and

Variable	ICC (ρ)*	CI 95% of ICC+	Cronbach's α
STEET	0.92	0.68-0.98	0.92
TCSET	0.89	0.59-0.97	0.88
SBET right	0.84	0.39-0.96	0.83
SBET left	0.75	0.06-0.94	0.78
* ICC (ρ) was calculated using a one-way random model. + ICC indicates the intra-class correlation coefficient. CI, confidence interval.			

Table 5.2: Inter-rater reliability between traditional trunk endurance assessment and mDurance.

finalizing the tests. In fact, they commented that the position adopted by the subjects through following the app guidance seemed to be more adequate than the one based on instructions from visual inspection. For example, a wedge is used in the TCSET to fix the initial position to an inclination of 60°, and then this wood is pulled back ten centimeters before starting the test (Figure 3.1(b)). During the process of pulling back the wedge, individuals tend to relax the posture and bend the trunk more than required. This occurs while the expert is operating the wood, thus the initial reference is usually not conserved. Conversely, specialists experienced more reliability when using mDurance, since they could just initiate the test whenever the appropriate angle was reached by the subject as shown in the app. Likewise, the experts were truly impressed with the precision of the estimated angle and agreed that the finalization time was fairly determined. Furthermore, the real-time EMG representation was greatly appreciated, especially to observe the muscle contraction during the realization of the test. This feature, together with the calculation of RMS, ARV and MVC values, were considered important assets of the system. The experts commented on the interest of having an automated log of time and muscle fatigue values to evaluate the patient improvement during their treatments or preventive interventions. In fact, they positively valued that all the information is automatically persisted into the system, and it can be retrieved and displayed at any time, even the data from prior sessions. They also considered this of much relevance for potentially constructing an evidence training program. Finally, the simplicity in the app usage and friendliness of its interface were highlighted as well. Indeed, this was considered during the development of the application, which seeks to attain ease of use and intuitiveness without sacrificing functionality.

Although experts did not report special negative comments, they mentioned that

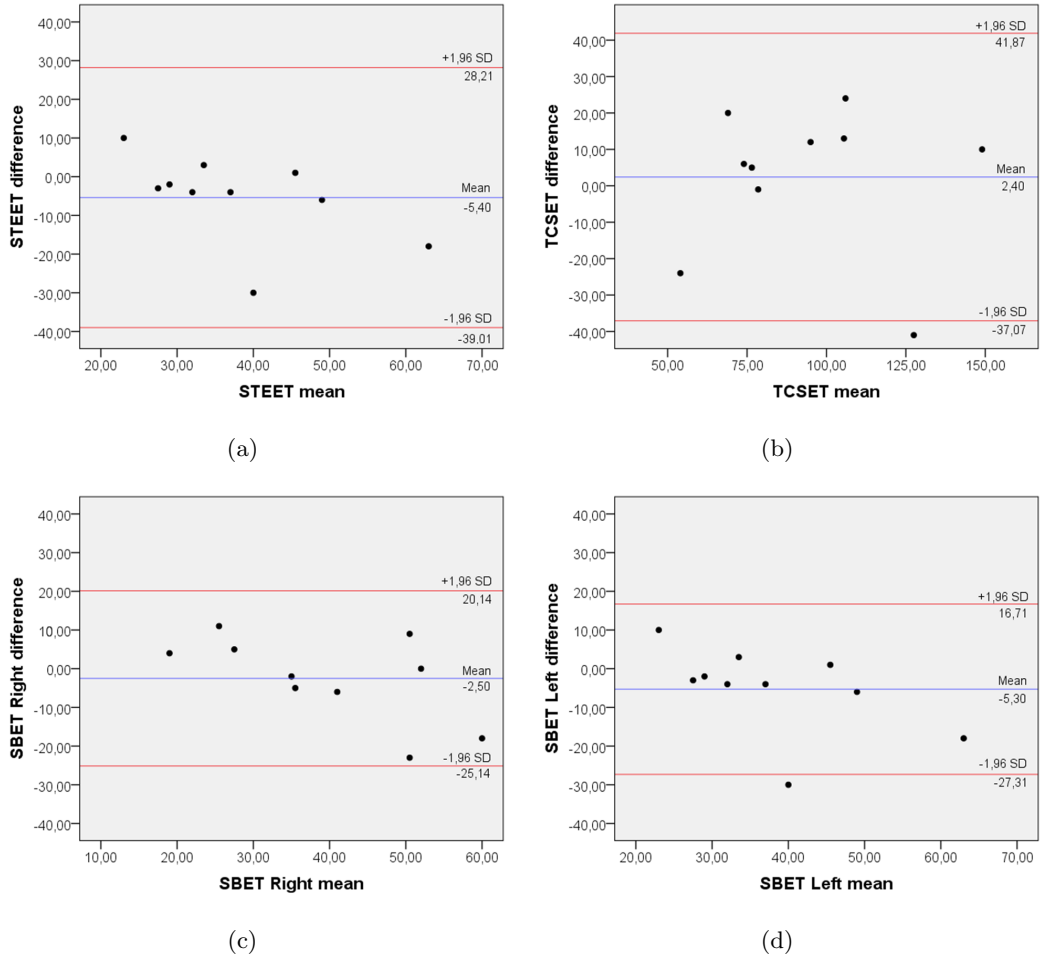


Figure 5.1: Agreement analysis between traditional trunk endurance assessment and mDurance through Bland-Altman plots: (**a**) STEET; (**b**) TCSET; (**c**) SBET Right; and (**d**) SBET Left. The mean of differences ($\bar{x}$) is represented by a blue line, while the limits of agreement ($\bar{x} \pm 1.96\sigma_X$) are depicted in red.

simpler guidelines should be provided along with the mDurance application to accelerate the understanding and usability of the whole system. During the first interaction with mDurance they faced some troubles when connecting the sensors, which were nevertheless overcome after following the instructions given by the designers. Furthermore, they considered desirable to share the data among diverse platforms, since the current version of the system limits its use to a single device. All these valuable comments have especially been taken into account for future extension of this work.

5.3 Usability

Apart from these qualitative outcomes, the usability of mDurance has also been evaluated in a more formal fashion by employing the System Usability Scale (SUS) (108, 109). This scale has nearly become an industry standard utilized to quantify the user experience with respect to diverse sort of technologies. The SUS consists of ten statements that are scored by the user through a five-point scale anchored with "strongly disagree" and "strongly agree". The SUS provides a point estimate of percentage usability. Scores of above 70 are acceptable, while highly usable products score above 90. Scores below 50 indicate unacceptable low levels of usability (110). For the evaluation, a total of seven independent experts were asked to use mDurance during a small trial. First, the experts were instructed on how to use mDurance, and after the finalization of the tests, they were requested to complete the questionnaire. The mean SUS score for the sample was 84.29±5.15 as shown in Figure 5.2. SUS scores were generally high and indicated high levels of acceptability, ease of use and confidence when using mDurance. To further confirm the usability of mDurance, a study with a higher number of experts is needed, but preliminary findings seem to be favorable.

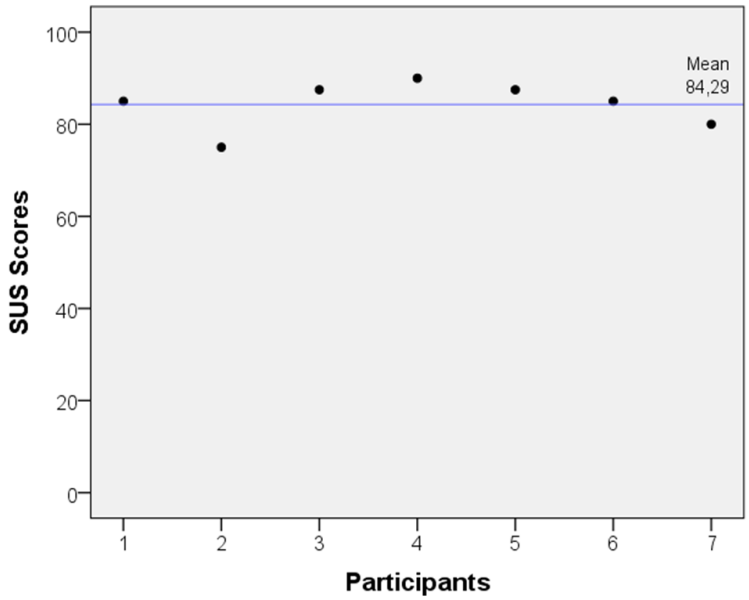


Figure 5.2: System Usability Scale (SUS) scores obtained from seven experts after using mDurance.

6

Conclusions

A spectacular proliferation of medical applications and systems has been observed during the recent years; however, more significant contributions are still necessary to simplify, expedite and improve traditional health practices. In pathophysiology, trunk endurance assessment is a clear application area lacking of appropriate tools. In fact, experts normally suffer from diverse kind of limitations during the use of traditional procedures, such as difficulties in the precise estimation of the duration of the test, challenges in the evaluation of the muscle strength, and other sort of problems related to the subjective nature of each specialist assessment. Moreover, practitioners need to concentrate on measurement and annotation tasks, instead of focusing on most relevant duties during the course of the test, like the analysis of the individuals behavior. To overcome these limitations this work has presented mDurance, an innovative system that combines wearable inertial and electromyography sensors together with mobile devices for supporting a more accurate and rapid assessment of trunk endurance. The inertial sensors are used to continuously obtain the attitude of the trunk based on quaternions theory. This absolute trunk orientation helps experts determine when the user attains the correct posture to initiate the endurance test, as well as to automatically identify its finalization based on some established termination criteria. The electromiography sensor allows practitioners to observe the trunk muscles activity during the execution of the tests, as well as the level of muscle fatigue experienced by the subject. All the information is processed by a mobile application that develops on a novel mHealth framework. The app significantly simplifies the routine of the expert and helps to manage the information collected from multiple individuals and sessions, which is considered of primal interest for tracking the evolution of the patients from visit to visit. An initial evaluation of the mDurance system has been performed to showcase the po-

tential use of this system. The remarkable reliability and usability results demonstrate that mDurance can be accepted as a tool to measure trunk endurance. Finally, taking into account the high level of satisfaction shown by experts, next steps include the use of mDurance on a large scale clinical test bed, which is currently under development.

6.1 Future work

Given the important magnitude of the engineering biomedical field and the growing of the wearable sensors, investigations and publications about health and technology. It is also important the fact that the health care is becoming essential in our lives along with the continuous release of wearable health devices, this system could be envisioned as a key tool for future investigation and development could be carried on by companies on this field. Various application functionalities could be improved and new ones may be added. Some fields are described:

- *Application functionality.* In this field, we are going to include some ideas to develop:

  - Guidelines with personalized exercises to realize according to the patient endurance trunk or its muscle fatigue. It could be associated with the mHealthDroid System Manager, which provides functionality to reproduce YouTube videos as well as local stored ones. Audios stored locally can be reproduced too.

  - Remote storage. Other important field to develop is the storage. Remote storage may provide a lot of functionalities, from to storage large amounts of data to save these data with a major level of security. mHealthDroid also provides of a Remote Storage Manager, which allows to upload information to a remote server by PHP protocol (111).

  - Mechanisms for synchronizing the wearable sensors when they are turn on. For example, the specialist could avoid some steps in the connection if mDurance is able to synchronizing by Bluetooth both devices and order them the data streaming in only one click of the "power" button.

  - To improve the data visualization including more functionalities. Represent the online visualization for different devices in the same graph. - To admit to other devices. It is possible if we create drivers for other portable health devices in mHealthDroid.

  - Add other communication protocols, such as ZigBee (112), Xbee (113), Wifi, and so on, as the market starts providing new communication interfaces.

- Implementation of mDurance in another devices. It would be interesting that this application could also run and automatically adjust to the tablets, with both the results and graphs, as texts and help videos would be displayed better. Moreover a tablet could be a more suitable device for this application thinking that it will be used by a specialist.

- *Application fields.* We propose some field where mDurance is interesting to be used:

  - Investigation fields. They exist a several possible investigation ways to develop in the future based in mDurance. For example, it may be assess the mDurance working in low back pain chronic patients. To observe the endurance and muscle fatigue that they show during the test and to compare with healthy subjects.

  - New endurance and muscle fatigue features. Other way is to include new endurance and muscle fatigue parameters in mDurance, such as the mean frequency (MNF) and median frequency (MDF) of EMG signal. In this way, the specialist would have more accurate data to diagnose the patient. Another important constant in LPB chronic patients is the phenomenon of flexion/extension during the complete flexion of the trunk. Thus, it is interesting to introduce the different relations between flexion and extension in the final assessment from each test. It can do it if we compare the times among STEET and TCSET or among both sides in SBET. The extensor/flexor ratio is a more sensitive index than peak torque in predicting future back pain in asymptomatic individuals (17).

  - New endurance tests. There are another endurance test to assess both trunk muscles and another kind of muscles of the human body. For example, the one leg standing balance test or the hip extension (17) associated to the past, present and future LBP or the muscle imbalance. The active straight leg raising test (17) associated to the postpartum sacroiliac pain.

  - Other application areas. We think that another important areas to apply mDurance may be sport or fitness area, for example for sports clubs like football clubs where these kind of tests are very important in the beginning of the seasons to prevent future injuries. Not surprisingly, to check how the researchers who have studied the relationship between core stability and performance, or the effects of muscle training the core of the alleged improvement of sports performance (114). Another interested area could be study the behavior of this system in animals. For example, if we also think in sport area, we can link them in the study of the endurance and the muscle fatigue in race horses, dogs, etc.

References

[1] IHME. *Institute for Health Metrics and Evaluation (IHME)*. June 2015. ix, 1, 2, 3, 4

[2] JA Moral-Muñoz, MJ Cobo, E Peis, M Arroyo-Morales, and E Herrera-Viedma. **Analyzing the research in Integrative & Complementary Medicine by means of science mapping**. *Complementary therapies in medicine*, **22**(2):409–418, 2014. ix, 2, 5, 6

[3] IHS Technology. *IHS Technology*. June 2015. ix, 8

[4] Mio. **Heart rate training**. Available online: `https://mioglobal.com`, 2015. Accessed: 2015-03-30. ix, 15, 16

[5] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. **mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications**. In *Ambient Assisted Living and Daily Activities*, pages 91–98. Springer, 2014. ix, 16, 17

[6] **Google fit**. Available online: `https://developers.google.com/fit/overview`, 2015. Accessed: 2015-03-30. ix, 18, 19

[7] Shimmer. **Shimmer Sensors**. Available online: `http://www.shimmersensing.com/`, 2015. Accessed: 2015-03-30. ix, 20

[8] Alberto Olivares, Gonzalo Olivares, F Mula, JM Górriz, and Javier Ramírez. **Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications**. *Journal of systems architecture*, **57**(10):905–915, 2011. ix, 20, 23

[9] Monarch Diagnostic Services. *Monarch Diagnostic Services*. June 2015. x, 29

[10] **Mobile/Tablet Top Operating System Share Trend**. https://www.netmarketshare.com/. x, 38

[11] Elena Bergamini, Gabriele Ligorio, Aurora Summa, Giuseppe Vannozzi, Aurelio Cappozzo, and Angelo Maria Sabatini. **Estimating Orientation Using Magnetic and Inertial Sensors and Different Sensor Fusion Approaches: Accuracy Assessment in Manual and Locomotion Tasks**. *Sensors*, **14**(10):18625–18649, 2014. x, 40

[12] Laxmaiah Manchikanti, Vijay Singh, Frank JE Falco, Ramsin M Benyamin, and Joshua A Hirsch. **Epidemiology of low back pain in adults**. *Neuromodulation: Technology at the Neural Interface*, **17**(S2):3–10, 2014. 2

[13] Mugdha Gore, Alesia Sadosky, Brett R Stacey, Kei-Sing Tai, and Douglas Leslie. **The burden of chronic low back pain: clinical comorbidities, treatment patterns, and health care costs in usual care settings**. *Spine*, **37**(11):E668–E677, 2012. 2

[14] Bernabé Esteban, Álvaro Tejeda-Lorente, Carlos Porcel, José Antonio Moral-Munoz, and Enrique Herrera-Viedma. **Aiding in the Treatment of Low Back Pain by a Fuzzy Linguistic Web System**. In *Rough Sets and Current Trends in Computing*, pages 250–261. Springer, 2014. 2, 7

[15] Carolyn A Richardson, Chris J Snijders, Julie A Hides, Léonie Damen, Martijn S Pas, and Joop Storm. **The relation between the transversus abdominis muscles, sacroiliac joint mechanics, and low back pain**. *Spine*, **27**(4):399–405, 2002. 3

[16] Roger Chou and Laurie Hoyt Huffman. **Nonpharmacologic therapies for acute and chronic low back pain: a review of the evidence for an American Pain Society/American College of Physicians clinical practice guideline**. *Annals of internal medicine*, **147**(7):492–504, 2007. 4

[17] Craig Liebenson. **Spinal stabilization–an update. Part 2functional assessment**. *Journal of Bodywork and Movement Therapies*, **8**(3):199–210, 2004. 4, 25, 26, 93

[18] Kerrie Evans, Kathryn M Refshauge, and Roger Adams. **Trunk muscle endurance tests: reliability, and gender differences in athletes**. *Journal of Science and Medicine in Sport*, **10**(6):447–455, 2007. 6, 25

[19] José A Moral-Muñoz, Bernabé Esteban-Moreno, Manuel Arroyo-Morales, Manuel J Cobo, and Enrique Herrera-Viedma. **Agreement between face-to-face and free software video analysis for assessing hamstring flexibility in adolescents.** *The Journal of Strength & Conditioning Research*, 2015. 7, 86

[20] Fuensanta Palacín-Marín, Bernabé Esteban-Moreno, Nicolas Olea, Enrique Herrera-Viedma, and Manuel Arroyo-Morales. **Agreement between telerehabilitation and face-to-face clinical outcome assessments for low back pain in primary care**. *Spine*, **38**(11):947–952, 2013. 7, 86

[21] Tamara Hafner and Jeremy Shiffman. **The emergence of global attention to health systems strengthening**. *Health policy and planning*, **28**(1):41–50, 2013. 7

[22] Marcia M Ditmyer, Robert Topp, and Matthew Pifer. **Prehabilitation in preparation for orthopaedic surgery**. *Orthopaedic Nursing*, **21**(5):43–54, 2002. 7

[23] Emily Oshima Lee and Ezekiel J Emanuel. **Shared decision making to improve care and reduce costs**. *New England Journal of Medicine*, **368**(1):6–8, 2013. 7

[24] Trevor G Russell, Peter Buttrum, Richard Wootton, and Gwendolen A Jull. **Rehabilitation after total knee replacement via low-bandwidth telemedicine: the patient and therapist experience**. *Journal of telemedicine and Telecare*, **10**(suppl 1):85–87, 2004. 7

REFERENCES

[25] Santosh Kumar, Wendy Nilsen, Misha Pavel, and Mani Srivastava. **Mobile health: Revolutionizing healthcare through transdisciplinary research**. *Computer*, (1):28–35, 2013. 7

[26] Adam C Powell, Adam B Landman, and David W Bates. **In search of a few good apps**. *JAMA*, **311**(18):1851–1852, 2014. 7, 11

[27] Karl Frederick Braekkan Payne, Heather Wharrad, and Kim Watts. **Smartphone and medical related App use among medical students and junior doctors in the United Kingdom (UK): a regional survey**. *BMC medical informatics and decision making*, **12**(1):121, 2012. 11

[28] Heather J Seabrook, Julie N Stromer, Cole Shevkenek, Aleem Bharwani, Jill de Grood, and William A Ghali. **Medical applications: a database and characterization of apps in Apple iOS and Android platforms**. *BMC research notes*, **7**(1):573, 2014. 12

[29] Harry McCracken. **Whos Winning, iOS or Android? All the Numbers, All in One Place**. *Time (April 16, 2013)*, 2013. 12

[30] Liguo Yu. **The coevolution of mobile os user market and mobile application developer community**. *International Journal of Advanced Computer Technology*, **2**(2):44–48, 2013. 12

[31] M.S. Huto. **Stretch Away**. Available online: http://www.stretchawaymusclepain.com/apps/, 2014. Accessed: 2015-03-30. 13

[32] K. Muller and H. Welsch. **BackDoctor**. Available online: http://www.ihanwel.com/, 2010. Accessed: 2015-03-30. 13

[33] K. Krugh. **iRehab**. Available online: https://www.irehab.com/, 2012. Accessed: 2015-03-30. 13

[34] iGlimpse. **Prevent Back Pain: Exercises For A Correct Posture and A Strong Lower Back**. Available online: http://www.iglimpse.co.uk/iglimpseapps.html, 2013. Accessed: 2015-03-30. 13

[35] Proven Digital Web Solutions. **Yoga for Back Pain Relief**. Available online: http://www.provenwebsoft.com/home, 2014. Accessed: 2015-03-30. 13

[36] WebMD. **WebMD Pain Coach**. Available online: http://www.webmd.com/mobile, 2014. Accessed: 2015-03-30. 13

[37] N. Gabriel. **Upper & Lower Back Pain Relief**. Available online: https://play.google.com/store/apps/details?id=backpain.magm, 2014. Accessed: 2015-03-30. 13

[38] WonderWorldApps. **Back Pain Guide**. Available online: https://play.google.com/store/apps/details?id=com.a83181645503bbf22429e04a.a72763738a, 2014. Accessed: 2015-03-30. 13

[39] KoolAppz. **Back Pain Complete Guide**. Available online: https://play.google.com/store/apps/details?id=com.koolappz.EP77900830001, 2014. Accessed: 2015-03-30. 13

[40] Ortho Tech Inc. **Back Pain: An Algorithmic Approach to Low Back Pain**. Available online: http://www.orthotecheducation.com/online_training.html, 2015. Accessed: 2015-03-30. 13

[41] Dnvgoods. **Back Pain Causes And Cures**. Available online: https://play.google.com/store/apps/details?id=com.appmk.book.AOTVZFAHREPPBPTEI, 2014. Accessed: 2015-03-30. 13

[42] D. Tessitore. **Back Pain Nerve Chart**. Available online: https://itunes.apple.com/us/app/back-pain-nerve-chart/id430573672?mt=8, 2012. Accessed: 2015-03-30. 13

[43] PostureCo. **PostureScreen: Posture, Body Fat & Movement Analysis**. Available online: http://postureanalysis.com/, 2015. Accessed: 2015-03-30. 14

[44] M. Wong, M. LeMoine, LeMoine A., and E. Yung. **Clinical Pattern Recognition: Low Back Pain**. Available online: http://www.clinicalpatternrecognition.com/, 2014. Acessed: 2015-03-30. 14

[45] Alpha United kft. **Virtual Diagnosis Spine**. Available online: https://itunes.apple.com/us/artist/alpha-united-kft/id574288842, 2014. Accessed: 2015-03-30. 14

[46] Alexandros Pantelopoulos and Nikolaos G Bourbakis. **A survey on wearable sensor-based systems for health monitoring and prognosis**. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, **40**(1):1–12, 2010. 14

[47] Adrian Burns, Barry R Greene, Michael J McGrath, Terrance J O'Shea, Benjamin Kuris, Steven M Ayer, Florin Stroiescu, and Victor Cionca. **SHIMMER–A wireless sensor platform for noninvasive biomedical research**. *Sensors Journal, IEEE*, **10**(9):1527–1534, 2010. 15

[48] Lumo Bodytech. **Posture corrector and activity tracker**. Available online: http://www.lumobodytech.com, 2015. Accessed: 2015-03-30. 15

[49] Deltason. **Biometrics Data Acquisition Systems**. Available online: http://www.deltason.com/products, 2015. Accessed: 2015-03-30. 15

[50] Pancreum. **Genesis by Pancreum**. Available online: http://www.pancreum.com/, 2015. Accessed: 2015-03-30. 15

[51] Asthmapolis. **Asthmapolis sensor**. Available online: http://propellerhealth.com/, 2015. Accessed: 2015-03-30. 15

[52] iHealth. **iHealth blood pressure monitor**. Available online: http://www.ihealthlabs.com/blood-pressure-monitors/wireless-blood-pressure-wrist-monitor/, 2015. Accessed: 2015-03-30. 16

[53] Deborah Estrin and Ida Sim. **Open mHealth architecture: an engine for health care innovation**. *Science(Washington)*, **330**(6005):759–760, 2010. 16

[54] Julien Oster, Joachim Behar, Roberta Colloca, Qichen Li, Qiao Li, and Gari D Clifford. **Open source Java-based ECG analysis software and Android app for atrial fibrillation screening**. In *Computing in Cardiology Conference (CinC), 2013*, pages 731–734. IEEE, 2013. 17

[55] Hung-Ming Chen and Po-Hung Chen. **Framework design-integrating an android open platform with multi-interface biomedical modules for physiological measurement**. *Journal of Conv. Infor. Tech*, **7**(12):310–319, 2012. 18

[56] Joseph J Oresko, Zhanpeng Jin, Jun Cheng, Shimeng Huang, Yuwen Sun, Heather Duschl, and Allen C Cheng. **A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing**. *Information Technology in Biomedicine, IEEE Transactions on*, **14**(3):734–740, 2010. 18

[57] Oresti Banos, Claudia Villalonga, Miguel Damas, Peter Gloesekoetter, Hector Pomares, and Ignacio Rojas. **PhysioDroid: Combining Wearable Health Sensors and Mobile Devices for a Ubiquitous, Continuous, and Personal Monitoring**. *The Scientific World Journal*, **2014**, 2014. 18

[58] Oresti Banos, Muhammad Bilal Amin, Wajahat Ali Khan, Muhammad Afzel, Mahmood Ahmad, Maqbool Ali, Taqdir Ali, Rahman Ali, Muhammad Bilal, Manhyung Han, et al. **An Innovative Platform for Person-Centric Health and Wellness Support**. In *Bioinformatics and Biomedical Engineering*, pages 131–140. Springer, 2015. 18

[59] **Polar**. Available online: `http://www.polar.com/en/connect_with_polar/google_fit_and_apple_healthkit`, 2015. Accessed: 2015-03-30. 18

[60] **HealthKit**. Available online: `https://developer.apple.com/healthkit/`, 2015. Accessed: 2015-03-30. 19

[61] A Olivares, G Olivares, et al. **Accurate human limb angle measurement: sensor fusion through Kalman, least mean squares and recursive least-squares adaptive filtering**. *Measurement Science and Technology*, **22**(2):025801, 2011. 21, 22

[62] Daniel Roetenberg, Henk J Luinge, Chris TM Baten, and Peter H Veltink. **Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation**. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **13**(3):395–405, 2005. 21

[63] Henk J Luinge, Peter H Veltink, and Chris TM Baten. **Ambulatory measurement of arm orientation**. *Journal of biomechanics*, **40**(1):78–85, 2007. 21, 39

[64] Karol J ODonovan, Roman Kamnik, Derek T OKeeffe, and Gerard M Lyons. **An inertial and magnetic sensor based technique for joint angle measurement**. *Journal of biomechanics*, **40**(12):2604–2611, 2007. 22

[65] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. **Estimation of IMU and MARG orientation using a gradient descent algorithm**. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011. 22, 39

[66] Craig Liebenson. **Spinal stabilization-an update. Part 1-biomechanics**. *Journal of Bodywork and Movement Therapies*, **8**(2):80–84, 2004. 25

[67] F. Biering-Sorensen. **Physical measurements as risk indicators for low-back trouble over a one-year period.** *Spine*, **9**(2):106–119, 1984. 25

[68] Mary T Moffroid. **Endurance of trunk muscles in persons with chronic low back pain: assessment, performance, training**. *Journal of Rehabilitation Research and Development*, **34**:440–447, 1997. 25, 87

[69] Michael P Reiman, Amber D Krier, Julie A Nelson, Michael A Rogers, Zachariah O Stuke, and Barbara S Smith. **Comparison of different trunk endurance testing methods in college-aged individuals**. *International Journal of Sports Physical Therapy*, **7**(5):533, 2012. 26

[70] Christophe Demoulin, Marc Vanderthommen, Christophe Duysens, and Jean-Michel Crielaard. **Spinal muscle evaluation using the Sorensen test: a critical appraisal of the literature**. *Joint Bone Spine*, **73**(1):43–50, 2006. 26

[71] Wolfgang Gruther, Franziska Wick, Birgit Paul, Christoph Leitner, Martin Posch, Michael Matzner, Richard Crevenna, and Gerold Ebenbichler. **Diagnostic accuracy and reliability of muscle strength and endurance measurements in patients with chronic low back pain**. *Journal of Rehabilitation Medicine*, **41**(8):613–619, 2009. 26

[72] GyuTae Kim, Mohammad A Ahad, Mohammed Ferdjallah, and Gerald F Harris. **Correlation of muscle fatigue indices between intramuscular and surface EMG signals**. In *SoutheastCon, 2007. Proceedings. IEEE*, pages 378–382. IEEE, 2007. 28, 30, 31

[73] Ernest Nlandu Kamavuako, Jakob Celander Rosenvang, Ronnie Horup, Winnie Jensen, Dario Farina, and Kevin B Englehart. **Surface versus untargeted intramuscular EMG based classification of simultaneous and dynamically changing movements**. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **21**(6):992–998, 2013. 29

[74] Juliana M Fernández, Rubén C Acevedo, and Carolina B Tabernig. **Influencia de la fatiga muscular en la señal electromiográfica de músculos estimulados eléctricamente**. *Revista EIA*, (7):111–119, 2007. 29, 30

[75] Abdullah Al Zaman, T Sharmin, Mohammad Ashraf Ali Khan, and M Ferdjallah. **Muscle fatigue analysis in young adults at different MVC levels using EMG metrics**. In *SoutheastCon, 2007. Proceedings. IEEE*, pages 390–394. IEEE, 2007. 30, 31

[76] **Open Handset Alliance**. http://www.openhandsetalliance.com/. 34

[77] **Apache License**. http://www.apache.org/licenses/. 35

[78] **Number of available Android Applications**. http://www.appbrain.com/stats/number-of-android-apps/. 35

REFERENCES

[79] **Google Play hits 25 billion downloads**. http://officialandroid.blogspot.ca/2012/09/google-play-hits-25-billion-downloads.html. 35

[80] **Developer Economics Q3 2013**. http://www.developereconomics.com/reports/q3-2013/. 35

[81] **Android, the world's most popular mobile platform**. http://developer.android.com/about/index.html/. 35

[82] JL Weston and DH Titterton. **Modern inertial navigation technology and its application**. *Electronics & Communication Engineering Journal*, **12**(2):49–64, 2000. 38

[83] Oresti Banos, Miguel Damas, Hector Pomares, Alberto Prieto, and Ignacio Rojas. **Daily living activity recognition based on statistical feature quality group selection**. *Expert Systems with Applications*, **39**(9):8013–8021, 2012. 38

[84] Oresti Banos, Miguel Damas, Hector Pomares, Fernando Rojas, Blanca Delgado-Marquez, and Olga Valenzuela. **Human activity recognition based on a sensor weighting hierarchical classifier**. *Soft Computing*, **17**(2):333–343, 2013. 38

[85] Andrea Mannini, Stephen S Intille, Mary Rosenberger, Angelo M Sabatini, and William Haskell. **Activity recognition using a single accelerometer placed at the wrist or ankle.** *Medicine and science in sports and exercise*, **45**(11):2193–2203, 2013. 38

[86] Oresti Banos, Mate Attila Toth, Miguel Damas, Hector Pomares, and Ignacio Rojas. **Dealing with the effects of sensor displacement in wearable activity recognition**. *Sensors*, **14**(6):9995–10023, 2014. 38

[87] Hsueh-Chun Lin, Shu-Yin Chiang, Kai Lee, and Yao-Chiang Kan. **An Activity Recognition Model Using Inertial Sensor Nodes in a Wireless Sensor Network for Frozen Shoulder Rehabilitation Exercises**. *Sensors*, **15**(1):2181–2204, 2015. 38

[88] Zhiqiang Zhang, Zhipei Huang, and Jiankang Wu. **Ambulatory hip angle estimation using Gaussian particle filter**. *Journal of Signal Processing Systems*, **58**(3):341–357, 2010. 39

[89] AD Young. **Comparison of orientation filter algorithms for realtime wireless inertial posture tracking**. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 59–64. IEEE, 2009. 39

[90] Pasquale Daponte, Luca De Vito, Maria Riccio, and Carmine Sementa. **Experimental comparison of orientation estimation algorithms in motion tracking for rehabilitation**. In *Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on*, pages 1–6. IEEE, 2014. 39

[91] J Kuipers. **Quaternions And Rotation Sequences: A Primer With Applications To Orbits, Aerospace And Virtual Reality Author: JB Kui**. 2002. 39

[92] Sebastian OH Madgwick. **An efficient orientation filter for inertial and inertial/magnetic sensor arrays**. *Report x-io and University of Bristol (UK)*, 2010. 40, 64

[93] mHealthDroid. Available online: `https://github.com/mHealthDroid/mHealthDroid` (accessed on 27 March 2015). 58

[94] Android API component. **Service**. Available online: `http://developer.android.com/reference/android/app/Service.html/` (accessed on 27 March 2015). 59

[95] Android API component. **Message**. Available online: `http://developer.android.com/reference/android/os/Message.html/` (accessed on 27 March 2015). 59

[96] Android API component. **Handler**. Available online: `http://developer.android.com/reference/android/os/Handler.html/` (accessed on 27 March 2015). 59

[97] Android API component. **Looper**. Available online: `http://developer.android.com/reference/android/os/Looper.html/` (accessed on 27 March 2015). 60

[98] Android API component. **BluetoothAdapter**. Available online: `http://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html` (accessed on 27 March 2015). 62

[99] Android API component. **Bluetooth Device**. Available online: `http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html` (accessed on 27 March 2015). 62

[100] Android API. Available online: `http://developer.android.com/reference/packages.html` (accessed on 27 March 2015). 62

[101] J. Gehring. **Graphview**. Available online: `http://android-graphview.org/` (accessed on 27 March 2015). 77

[102] Jane Latimer, Christopher G Maher, Kathryn Refshauge, and Ian Colaco. **The reliability and validity of the Biering–Sorensen test in asymptomatic subjects and subjects reporting current or previous nonspecific low back pain**. *Spine*, **24**(20):2085, 1999. 86

[103] J. M. Bland and D. G. Altman. **A note on the use of the intraclass correlation coefficient in the evaluation of agreement between two methods of measurement**. *Computers in Biology and Medicine*, **20**(5):337–340, 1990. 86

[104] Lee J Cronbach. **Coefficient alpha and the internal structure of tests**. *Psychometrika*, **16**(3):297–334, 1951. 86

[105] J. M. Bland and D. G. Altman. **Statistical methods for assessing agreement between two methods of clinical measurement**. *The Lancet*, **327**(8476):307–310, 1986. 86

[106] Joseph A Gliem and Rosemary R Gliem. **Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales**. In *Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education*, 2003. 86

[107] PS Myles and James Cui. **Using the Bland–Altman method to measure agreement with repeated measures**. *British Journal of Anaesthesia*, **99**(3):309–311, 2007. 86

[108] John Brooke. **SUS-A quick and dirty usability scale**. *Usability Evaluation in Industry*, **189**(194):4–7, 1996. 90

[109] James R Lewis and Jeff Sauro. **The factor structure of the system usability scale**. In *Human Centered Design*, pages 94–103. Springer, 2009. 90

[110] Aaron Bangor, Philip T Kortum, and James T Miller. **An empirical evaluation of the system usability scale**. *Intl. Journal of Human–Computer Interaction*, **24**(6):574–594, 2008. 90

[111] Betz Mehdi A, Lopes N Dovgal, Richter G Magnusson H, and Vrana J Seguy D. **PHP Manual**. Available online: `https://php.net/manual/es/index.php/` (accessed on 27 March 2015). 92

[112] **ZigBee Alliance**. Available online: `http://www.zigbee.org/`, 2015. Accessed: 2015-05-30. 92

[113] **XBee**. Available online: `http://xbee.cl/`, 2015. Accessed: 2015-05-30. 92

[114] Christopher J Durall, Brian E Udermann, Dana R Johansen, Barbara Gibson, David M Reineke, and Paul Reuteman. **The effects of preseason trunk muscle training on low-back pain occurrence in women collegiate gymnasts**. *The Journal of Strength & Conditioning Research*, **23**(1):86–92, 2009. 93