

POLITECNICO DI MILANO
Scuola di Ingegneria dell'Informazione



POLITECNICO
MILANO 1863

POLO TERRITORIALE DI COMO
Master of Science in Computer Engineering

ECG-ira: An efficient mobile app for ECG analysis

Supervisor: Prof. Giuseppe Pozzi

Co-Supervisor: Ulisse Pizzagalli

Assistant Supervisor: Eng. Name Surname

Master Graduation Thesis by:

Antonello Fodde - 817371

Chai Botta - 817333

Academic Year 2015-2016

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur malesuada suscipit nisl, vitae gravida odio imperdiet a. Etiam sed auctor tellus. Donec sed mauris eget nibh luctus accumsan. Sed imperdiet purus in elit iaculis, non commodo nisl varius. Integer sit amet diam laoreet, viverra lacus id, placerat nisl. Aenean ultricies sollicitudin elit in sodales. Vivamus euismod eleifend justo, ac pellentesque risus. Donec eleifend, justo a pharetra laoreet, est dolor condimentum nulla, nec auctor ligula justo eget diam. Mauris augue eros, elementum quis vulputate eget, ullamcorper eget mauris. Maecenas quis hendrerit velit. Donec vehicula dictum tellus, et aliquam sem viverra maximus. Maecenas gravida purus quis dui vulputate ornare. Morbi ac orci ut nunc tristique ultricies nec mattis massa. In imperdiet nisl ut risus faucibus, ut semper libero egestas. Nam imperdiet ullamcorper nunc, eu dictum felis dapibus a. Aliquam nec ante posuere, tristique dui semper, hendrerit erat. Nunc purus massa, lobortis a laoreet vel, posuere sit amet nisl. Duis ut viverra nisl. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Nulla elit risus, efficitur condimentum erat id, laoreet ullamcorper tellus. Nam non rutrum massa. Suspendisse vitae mauris vitae arcu elementum molestie et at diam. Proin efficitur vehicula ligula, id rhoncus dui pulvinar et. Nunc nec ultrices nunc, ut cursus libero. Donec mattis vehicula ex eget efficitur. Ut tellus arcu, vehicula nec ullamcorper ut, vulputate eu arcu. In cursus ut justo non sodales. Sed massa urna, eleifend eu nisi eget, tincidunt efficitur nibh. Donec at molestie arcu. Duis elementum lectus at tristique scelerisque. Donec sodales purus viverra urna interdum, sit amet convallis velit lacinia. Aliquam mollis tempus rhoncus. Quisque ultricies nisi quis metus rhoncus mattis. In fermentum facilisis tristique. Fusce ultricies quam id suscipit venenatis. Vestibulum hendrerit nibh eget ligula tristique, a tincidunt metus viverra. Aliquam finibus dui velit, a accumsan lacus tempus sit amet. Fusce in justo lorem. Mauris a porttitor justo, eget tincidunt erat. Ut quis semper risus. Aliquam

eu malesuada metus. Aliquam erat volutpat. Sed ac diam finibus, pellentesque enim ut, commodo mi. Nam ligula odio, semper eget diam id, tempus cursus risus. Donec viverra, elit quis molestie ultrices, odio risus ornare orci, at eleifend nisl risus elementum tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Suspendisse eu aliquet est, scelerisque vulputate justo.

Suspendisse egestas posuere lacinia. Integer non mi maximus, rhoncus massa eu, aliquam risus. Nam nisl nisi, semper nec efficitur eu, tincidunt vel justo. Sed vestibulum tristique consequat. Phasellus sodales nunc quis pharetra mattis. Pellentesque eu fermentum sem, vel efficitur odio. Suspendisse consectetur turpis et nisi viverra commodo. Quisque ornare porta nisi, eget aliquam nisl interdum sed. Nulla elementum quam sit amet enim mollis, in rhoncus felis lacinia. Nam vel justo purus. Nulla pellentesque ex in eros rutrum tincidunt. Morbi consequat felis a libero dictum, vel malesuada est dignissim. Maecenas diam lacus, laoreet ut quam vel, pharetra aliquet nisi. Suspendisse auctor aliquam odio eu tincidunt. Aenean lacinia semper diam. Suspendisse potenti.

Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur malesuada suscipit nisl, vitae gravida odio imperdiet a. Etiam sed auctor tellus. Donec sed mauris eget nibh luctus accumsan. Sed imperdiet purus in elit iaculis, non commodo nisl varius. Integer sit amet diam laoreet, viverra lacus id, placerat nisl. Aenean ultricies sollicitudin elit in sodales. Vivamus euismod eleifend justo, ac pellentesque risus. Donec eleifend, justo a pharetra laoreet, est dolor condimentum nulla, nec auctor ligula justo eget diam. Mauris augue eros, elementum quis vulputate eget, ullamcorper eget mauris. Maecenas quis hendrerit velit. Donec vehicula dictum tellus, et aliquam sem viverra maximus. Maecenas gravida purus quis dui vulputate ornare. Morbi ac orci ut nunc tristique ultricies nec mattis massa. In imperdiet nisl ut risus faucibus, ut semper libero egestas. Nam imperdiet ullamcorper nunc, eu dictum felis dapibus a. Aliquam nec ante posuere, tristique dui semper, hendrerit erat. Nunc purus massa, lobortis a laoreet vel, posuere sit amet nisl. Duis ut viverra nisl. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Nulla elit risus, efficitur condimentum erat id, laoreet ullamcorper tellus. Nam non rutrum massa. Suspendisse vitae mauris vitae arcu elementum molestie et at diam. Proin efficitur vehicula ligula, id rhoncus dui pulvinar et. Nunc nec ultrices nunc, ut cursus libero. Donec mattis vehicula ex eget efficitur. Ut tellus arcu, vehicula nec ullamcorper ut, vulputate eu arcu. In cursus ut justo non sodales. Sed massa urna, eleifend eu nisi eget, tincidunt efficitur nibh. Donec at molestie arcu. Duis elementum lectus at tristique scelerisque. Donec sodales purus viverra urna interdum, sit amet convallis velit lacinia. Aliquam mollis tempus rhoncus. Quisque ultricies nisi quis metus rhoncus mattis. In fermentum facilisis tristique. Fusce ultricies quam id suscipit venenatis. Vestibulum hendrerit nibh eget ligula tristique, a tincidunt metus viverra. Aliquam finibus dui velit, a accumsan lacus tempus sit amet. Fusce in justo lorem. Mauris a porttitor justo, eget tincidunt erat. Ut quis semper risus. Aliquam

eu malesuada metus. Aliquam erat volutpat. Sed ac diam finibus, pellentesque enim ut, commodo mi. Nam ligula odio, semper eget diam id, tempus cursus risus. Donec viverra, elit quis molestie ultrices, odio risus ornare orci, at eleifend nisl risus elementum tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Suspendisse eu aliquet est, scelerisque vulputate justo.

Suspendisse egestas posuere lacinia. Integer non mi maximus, rhoncus massa eu, aliquam risus. Nam nisl nisi, semper nec efficitur eu, tincidunt vel justo. Sed vestibulum tristique consequat. Phasellus sodales nunc quis pharetra mattis. Pellentesque eu fermentum sem, vel efficitur odio. Suspendisse consectetur turpis et nisi viverra commodo. Quisque ornare porta nisi, eget aliquam nisl interdum sed. Nulla elementum quam sit amet enim mollis, in rhoncus felis lacinia. Nam vel justo purus. Nulla pellentesque ex in eros rutrum tincidunt. Morbi consequat felis a libero dictum, vel malesuada est dignissim. Maecenas diam lacus, laoreet ut quam vel, pharetra aliquet nisi. Suspendisse auctor aliquam odio eu tincidunt. Aenean lacinia semper diam. Suspendisse potenti.

Contents

Abstract	iii
Sommario	v
1 Introduction	1
2 Electrocardiography overview	5
2.1 The heart	5
2.1.1 Human heart structure	5
2.1.2 Human heart function	7
2.2 Heart electrical activity	9
2.3 Electrocardiogram	10
2.3.1 Lead I	12
2.3.2 Lead II	12
2.3.3 Lead III	13
2.3.4 Augmented leads	13
2.3.5 Precordials leads	13
2.3.6 How to read a ECG record	14
2.4 Noises and interferences	15
2.4.1 Artifact	15
2.4.2 Interference	16
2.4.3 Wandering baseline	16
2.4.4 Faulty equipment	17
3 State of Art	19
3.1 Device	19
3.1.1 Mortara ELI 10 Mobile	19

3.1.2	Philips DigiTrak XT Holter Recorder	20
3.1.3	AliveCor ECG	21
3.1.4	M-Trace (PC)Mobile	21
3.1.5	ECG Expert	22
3.2	Mobile application	22
3.2.1	Visualization only application	23
3.2.2	Visualization and Analysis	23
4	Objective	25
4.1	Preface	25
4.2	Fully functional medical mobile app as replacement to desktop app .	25
5	Requirement	27
5.1	Functional	27
5.1.1	Connection management with the acquisition device	27
5.1.2	Acquisition, storing and management of ECG records	27
5.1.3	Different ECG formats support	28
5.1.4	Dynamic display scaling	28
5.1.5	ECG record analysis integration on mobile platform	28
5.1.6	Analysis results displaying	29
5.1.7	Highly parameterizable	30
5.2	Nonfunctional	30
5.2.1	Reduced memory usage	30
5.2.2	Minimum performance rate and scalability on performance .	31
5.2.3	Wide platform compatibility and accessibility	31
5.2.4	Documentation	31
6	Problems	33
6.1	Mobile platform fragmentation	33
6.2	Native vs Cross-Platform	34
6.2.1	Native	35
6.2.2	Cross-Platform	36
6.3	Mobile hardware limitations	37
6.3.1	Memory limitations	37
6.3.2	Performance limitations	37

6.4 ECG baseline wander	38
6.4.1 State of Art	38
6.5 Signal visualization	40
7 System architecture	43
7.1 Acquisition device	43
7.1.1 Mobile app	44
8 Final result	51
8.1 App Screens	51
Acknowledgement	55

List of Figures

2.1	The human heart structure.[1]	6
2.2	The circulatory system with blood flow.	8
2.3	The SA node fires and electrical impulses travels through the right and left atrium.	9
2.4	The impulse then move to the ventricular area.	10
2.5	An example of a normal ECG waveform.	11
2.6	The Einthoven's triangle show the right position to place leads over the chest.	12
2.7	Precordial leads and their position related to the heart and the chest horizontal plane.	14
2.8	A typical ECG paper.	14
2.9	ECG waveform interference due to artifact may cause monitoring to fail due to unreadable signals.	15
2.10	Electrical interference causes the baseline to be unstable and the signal is corrupted.	16
2.11	An example of baseline wandering due to artifacts.	16
3.1	Mortara ELI 10 Mobile, ECG acquisition device box.	20
3.2	DigiTrack, the ECG visualization.	20
3.3	AliveCor device real time acquisition on a tablet.	21
3.4	M-Trace PC device for ECG acquisition.	22
3.5	M-Trace PC device for ECG acquisition.	22
5.1	Istogram from the desktop application resulted from an analysis on a MIT/BIH record.	29
5.2	Tacogram from the desktop application resulted from an analysis on a MIT/BIH record.	29

5.3 ST+/ST- graph from the desktop application resulted from an analysis on a MIT/BIH record.	30
6.1 Mobile platform share evolution (smartphone sales), 2009–13.	34
6.2 Native app development process.	35
6.3 Native vs Cross-Platform cost and time factors.[2]	36
6.4 Memory reserved to an app in Android.	38
6.5 The baseline wander artifact in the ECG signal.	39
6.6 Relation between distortion and moving window length for wavelet package translation (WPT), traditionally used MA, and our proposed high-pass filter based on a statistical weighted moving average (SMA). (a) Maximum error (ME) vs. moving window length; (b) Normalized root mean square error (NRMSE) vs. moving window length. M is the number of sub-bounds. SMA is the same as MA when M=1.	41
7.1 ZEcg device block diagram including all the modules components.	44
7.2 ADS1198 functional diagram showing the 8 channels representing the 8 leads used during the ECG record acquisition.	45
7.3 Android Build system process. How and which component are involved during an android application build and compilation.	46
7.4 ECG-ira project structure and packages inside Android Studio IDE.	47
8.1 The home screen of the application ECG-ira.	52
8.2 The form to be compiled before a record can start. It is necessary to bind any record to a patient data in order to avoid confusion between ecg records. This data is also useful for a fully understanding of the record itself.	53

Chapter 1

Introduction

ECG-ira stands for ECG (electrocardiogram) instant rapid analyzer and it is an android application developed for the acquisition, visualization and analysis of the electrocardiogram signals. The application acquires and stores the record from the zecg acquisition device (using the zecg device format) property of the Politecnico of Milan but it can also open and visualize other ecg format from other devices. Ecg-ira is part of a greater and long term project (zecg itself was part of a first step). ECG-ira aims to exploit the reliability and performance of a complex software for electrocardiogram signal acquisition, visualization and processing within a smartphone device through a mobile application. Apart of the analysis algorithm already implemented during a past thesis (by Diego Ulisse Pizzagalli), the application was designed and implemented from scratch. During the process we dealt with the devices limitation in term of performance power and limited memory. We overcome many issues, related also to small and different screen size and density of pixel, due to the great number of different devices currently on the market. We had to exploit the multithreading and efficient memory usage strategies in order to achieve responsiveness and fulfill the medical requirement for an ECG compliant application. The result is an application which is easy to use because it follows all the best design principles according to the official Google guidelines for responsive UI and UX. By taking advantage of the multithreading capabilities and the usage of all the available cores into a device, we achieved an application which performs fast and well.

This thesis is structured as follow:

- After this introduction we give a brief but detailed overview about the heart, its functionalities and how an electrocardiogram is related to it by explaining

from what its an ecg to how heart electrical signals are detected and read.

- In the State of Art chapter we show the panorama of the actual devices for an ecg acquisition and some availables application on the market which allow to open and read an ecg record.
- Then it follows the Objective chapter in which we describe and explain the goal of this thesis work.
- The Requirements chapter is where we list a set of functional and nonfunctional set of features came up during the planning phase. The final result should be compliant with all of them.
- The Problem chapter deal with all the issues related to the project development. Here we discussed about the problem of choosing a development platform with respect to another, the hardware limitation on a smartphone device and the problems strictly related to the ecg signals.
- The Solution choice chapter is where we explain and provide our reasonings to the implementation choices, from the platform choice to the programming language choice to why we decide to avoid using drawings libraries preferring a custom and proprietary implementation.
- In the System architecture chapter we provide a general overview of the project structure for both the acquisition device implementation and architecture both the mobile application structure and main functionalities.
- In the Implementation details chapter we described all the main components and Java classes. Here we provide also hint for future implementations and adaptations.
- In the Final result chapter we propose some screenshots, with descriptive captions and description, of the main screens of the application; then we go through a performance analysis of application by observing how it affect the memory, the cpu and the response time of some portion of the code. The test were conducted on many different devices but only data from the older devices (2011) and the newest (November 2015) are compared.

- In the Conclusion chapter we sum up the overall results giving an overview of what has been done and providing some post consideration about the work done.
- In the Future works chapter we provide hints for further implementation in order to make ECG-ira really an all in one solution as an ECG mobile application. We also put our consideration about the future of the mobile software with respect to the desktop one, according to the new trend and the fact that the mobile environments is penetrating even more in our daily lives and can really positively change the way patients are connected with the health care providers.

Chapter 2

Electrocardiography overview

This chapter will introduce some basic but fundamental concepts about electrocardiography starting from the heart to the ECG and all issues related to the topic. We will start introducing the heart, its functionality and the entire circulatory system. After that we will describe in details the electrical activity inside the heart and how heart beats are generated. Following there will be a description of the electrocardiogram and the ECG signals. In the last section of this chapter we will discuss about all the noises and interference related to the ECG signal during its acquisition.

2.1 The heart

This chapter's focus is to describe in details the human heart. We will start from the structure to end up describing the heart functionality.

2.1.1 Human heart structure

The human heart is an organ that pumps blood throughout the body via the circulatory system, supplying oxygen and nutrients to the tissues and removing carbon dioxide and other wastes.

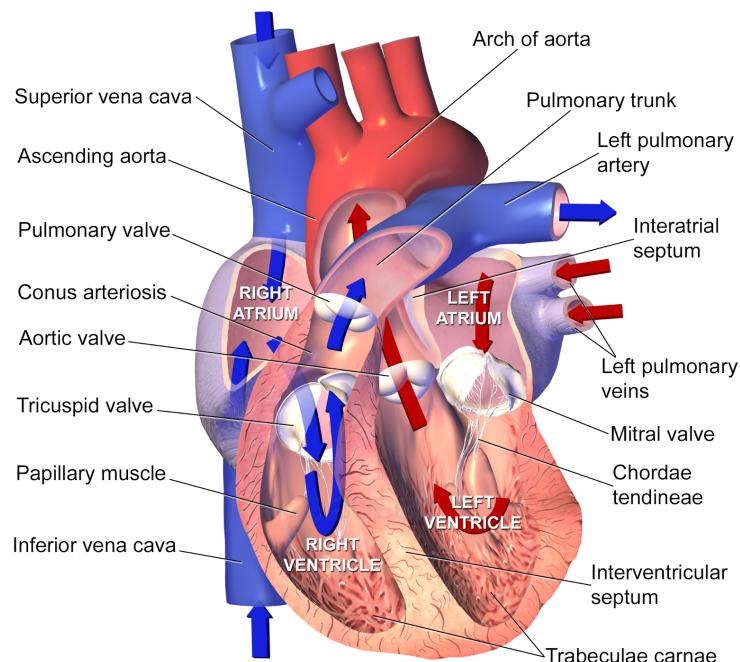
This fundamental organ has four chambers: two upper chambers(the atrial) and two lower ones(the ventricles). The right atrium and the right ventricle together make up the “right heart”, and the left atrium and left ventricle make up the”left heart”. The two sides of the heart are separated by a muscle called the septum.

A double-walled sac called the pericardium, encases the heart, which serves to protect

the heart and anchors it inside the chest. Between the outer layer, the parietal pericardium, and the inner layer, the serous pericardium, runs pericardial fluid, which lubricates the heart during contractions and movements of the lungs and diaphragm. The heart outer wall consists of three layers. The outermost wall layer, or epicardium, is the inner wall of the pericardium. The middle layer, or myocardium, contains the muscle that contracts. The inner layer, or endocardium, is the lining that contacts the blood.

The tricuspid valve and the mitral valve make up the atrioventricular (AV) valves, which connect the atria and the ventricles. The pulmonary semilunar valve separates the right ventricle from the pulmonary artery, and the aortic valve separates the left ventricle from the aorta. The heartstrings, or chordae tendineae, anchor the valves to heart muscles.

The sinoatrial node produces the electrical pulses that drive heart contractions.



Sectional Anatomy of the Heart

Figure 2.1: The human heart structure.[1]

2.1.2 Human heart function

The heart circulates blood through two pathways: the pulmonary circuit and the systemic circuit.

In the pulmonary circuit, deoxygenated blood leaves the right ventricle of the heart via the pulmonary artery and travels to the lungs, then returns as oxygenated blood to the left atrium of the heart via the pulmonary vein.

In the systemic circuit, oxygenated blood leaves the body via the left ventricle to the aorta, and from there enters the arteries and capillaries where it supplies the body's tissues with oxygen. Deoxygenated blood returns via veins to the venae cavae, re-entering the heart's right atrium.

Of course, the heart is also a muscle, so it needs a fresh supply of oxygen and nutrients too. After the blood leaves the heart through the aortic valve, two sets of arteries bring oxygenated blood to feed the heart muscle. The left main coronary artery, on one side of the aorta, branches into the left anterior descending artery and the left circumflex artery. The right coronary artery branches out on the right side of the aorta.

Blockage of any of these arteries can cause a heart attack, or damage to the muscle of the heart. A heart attack is distinct from cardiac arrest, which is a sudden loss of heart function that usually occurs as a result of electrical disturbances of the heart rhythm. A heart attack can lead to cardiac arrest, but the latter can also be caused by other problems.

The heart contains electrical "pacemaker" cells, which cause it to contract — producing a heartbeat.

Each cell has the ability to be the 'band leader' and have everyone follow. In people with an irregular heartbeat, or atrial fibrillation, every cell tries to be the band leader, which causes them to beat out of sync with one another.

A healthy heart contraction happens in five stages:

1. In the first stage (early diastole), the heart is relaxed.
2. Then the atrium contracts (atrial systole) to push blood into the ventricle.
3. Next, the ventricles start contracting without changing volume.
4. Then the ventricles continue contracting while empty.
5. Finally, the ventricles stop contracting and relax.

Then the cycle repeats.

Valves prevent backflow, keeping the blood flowing in one direction through the heart.

Some interesting data about the human heart are:

- A human heart is roughly the size of a large fist.
- The heart weighs between about 280 to 340 grams in men and 230 to 280 grams in women.
- The heart beats about 100,000 times per day (about 3 billion beats in a lifetime).
- An adult heart beats about 60 to 80 times per minute.
- Newborns' hearts beat faster than adult hearts, about 70 to 190 beats per minute.
- The heart pumps about 6 quarts (5.7 liters) of blood throughout the body.
- The heart is located in the center of the chest, usually pointing slightly left

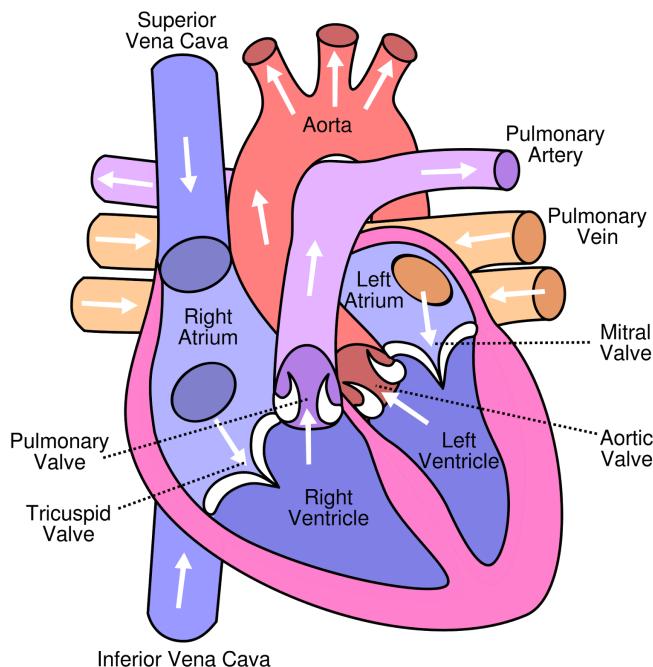


Figure 2.2: The circulatory system with blood flow.

2.2 Heart electrical activity

The heart has a natural pacemaker that regulates the pace or rate of the heart. It sits in the upper portion of the right atrium (RA) and is a collection of specialized electrical cells known as the SINUS or SINOATRIAL (SA) node.

Like the spark-plug of an automobile it generates a number of "sparks" per minute. Each "spark" travels across a specialized electrical pathway and stimulates the muscle wall of the four chambers of the heart to contract (and thus empty) in a certain sequence or pattern. The upper chambers or atria are first stimulated. This is followed by a slight delay to allow the two atria to empty. Finally, the two ventricles are electrically stimulated. In an car, the number of sparks per minute generated by a spark plug is increased when you press the gas pedal or accelerator. This revs up the motor. In case of the heart, adrenaline acts as a gas pedal and causes the sinus node to increase the number of sparks per minute, which in turn increases the heart rate. The release of adrenaline is controlled by the nervous system. The heart normally beats at around 72 times per minute and the sinus node speeds up during exertion, emotional stress, fever, etc., or whenever our body needs an extra boost of blood supply. In contrast, it slows down during rest or under the influence of certain medications. Well trained athletes also tend to have a slower heart beat.

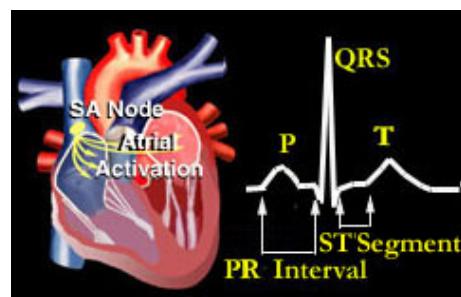


Figure 2.3: The SA node fires and electrical impulses travel through the right and left atrium.

The sequence of electrical activity within the heart is displayed in the diagrams above and occurs as follows:

1. As the SA node fires, each electrical impulse travels through the right and left atrium. This electrical activity causes the two upper chambers of the heart to

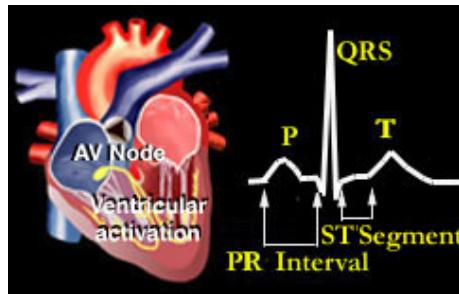


Figure 2.4: The impulse then move to the ventricular area.

contract. This electrical activity can be recorded from the surface of the body as a "P" wave" on the patient's EKG or ECG (electrocardiogram).

2. The electrical impulse then moves to an area known as the AV (atrio-ventricular) node. This node sits just above the ventricles. Here, the electrical impulse is held up for a brief period. This delay allows the right and left atrium to continue emptying its blood contents into the two ventricles. This delay is recorded as a "PR interval." The AV node thus acts as a "relay station" delaying stimulation of the ventricles long enough to allow the two atria to finish emptying.
3. Following the delay, the electrical impulse travels through both ventricles (via special electrical pathways known as the right and left bundle branches). The electrically stimulated ventricles contract and blood is pumped into the pulmonary artery and aorta. This electrical activity is recorded from the surface of the body as a "QRS complex". The ventricles then recover from this electrical stimulation and generates an "ST segment" and T wave on the ECG.

2.3 Electrocardiogram

An electrocardiogram(abbreviated as ECG or EKG) is a test that measures the electrical activity of the heartbeat. With each beat, an electrical impulse (or wave) travels through the heart. This wave causes the muscle to squeeze and pump blood from the heart. A normal heartbeat on ECG will show the timing of the top and lower chambers.

The right and left atria or upper chambers make the first wave called a "P wave" following a flat line when the electrical impulse goes to the bottom chambers. The right and left bottom chambers or ventricles make the next wave called a "QRS

complex.” The final wave or “T wave” represents electrical recovery or return to a resting state for the ventricles.

Each waves in the figure 2.5 is no other than the result of different views or

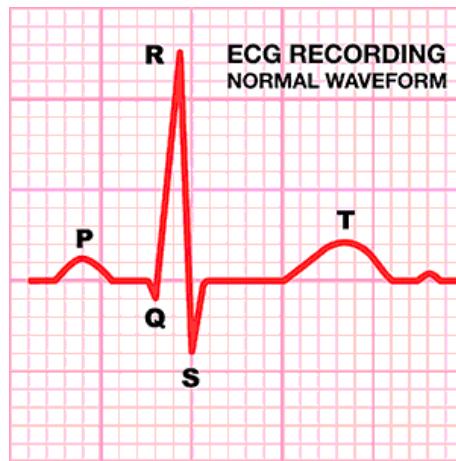


Figure 2.5: An example of a normal ECG waveform.

perspectives of the waveforms generated from the current in the heart.

There are two type of ECGs recordings: the 12-lead ECG and the rhythm strip. Both give valuable information about heart function.

We will focus our attention on the 12-lead ECG. It records information from 12 different views of the heart and provides a complete picture of electrical activity. The limb leads and the chest, or precordial, leads reflect information from the different planes of the heart. Different leads provide different information. The six limb leads I, II, III, augmented vector right (aVR), augmented vector left (aVL), and augmented vector foot (aVF) provide information about the heart’s frontal (vertical) plane. Leads I, II, and III require a negative and positive electrode for monitoring, which makes those leads bipolar. The augmented leads record information from one lead and are called unipolar.

The six precordials or V leads V1, V2, V3, V4, V5, and V6 provide information about the heart’s horizontal plane. Like the augmented leads, the precordial leads are also unipolar, requiring only a single electrode. The opposing pole of those leads is the center of the heart as calculated by the ECG.

The position of the leads are crucial for a right ECG recordings. It is common to use the so called Einthoven’s triangle, a set of positions to set up standard limb leads. The electrodes for leads I, II, III are about equidistant from the heart and form an

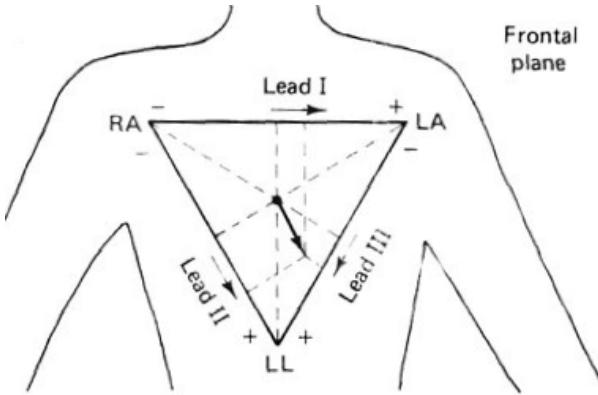


Figure 2.6: The Einthoven's triangle show the right position to place leads over the chest.

equilateral triangle.

2.3.1 Lead I

It provides a view of the heart that shows current moving from right to left. Because the current flows from negative to positive, the positive electrode for this lead is placed on the left arm or on the left side of the chest; the negative electrode is placed on the right arm. Lead I produces a positive deflection on ECG tracings and is helpful in monitoring atrial and hemiblock.

2.3.2 Lead II

Lead II produces a positive deflection. Place the positive electrode on the patient's left leg and the negative electrode on the right arm. For continuous monitoring, place the electrodes on the torso for convenience, with the positive electrode below the lowest palpable rib at the left midclavicular line and the negative electrode below the right clavicle. The current travels down and to the left in this lead. Lead II tends to produce a positive, high voltage deflection, resulting in tall P, R, and T waves. This lead is commonly used for routine monitoring and is useful for detecting sinus node and atrial arrhythmias.

2.3.3 Lead III

Lead III produces a positive deflection. The positive electrode is placed on the left leg; the negative electrode, on the left arm. Along with lead II, this lead is useful for detecting changes associated with an inferior wall myocardial infarction. The axes of the three bipolar limb leads I, II, and III form a triangle around the heart and provide a frontal plane view of the heart.

2.3.4 Augmented leads

Leads aVR, aVL, and aVF are called augmented leads. They measure electrical activity between one limb and a single electrode. Lead aVR provides no specific view of the heart. Lead aVL shows electrical activity coming from the heart's lateral wall. Lead aVF shows electrical activity coming from the heart's inferior wall.

2.3.5 Precordials leads

The six unipolar precordial leads (V1, V2, V3, V4, V5 and V6) are placed in sequence across the chest and provide a view of the heart's horizontal plane.

- Lead V1—The precordial lead V1 electrode is placed on the right side of the sternum at the fourth intercostal rib space. This lead corresponds to the modified chest lead MCL1 and shows the P wave, QRS complex, and ST segment particularly well. It helps to distinguish between right and left ventricular ectopic beats that result from myocardial irritation or other cardiac stimulation outside the normal conduction system. Lead V1 is also useful in monitoring ventricular arrhythmias, ST-segment changes, and bundle-branch blocks.
- Lead V2—Lead V2 is placed at the left of the sternum at the fourth intercostal rib space.
- Lead V3—Lead V3 goes between V2 and V4. Leads V1, V2, and V3 are biphasic, with both positive and negative deflections. Leads V2 and V3 can be used to detect ST-segment elevation.
- Lead V4—Lead V4 is placed at the fifth intercostal space at the midclavicular line and produces a biphasic waveform.

- Lead V5—Lead V5 is placed at the fifth intercostal space at the anterior axillary line. It produces a positive deflection on the ECG and, along with V4, can show changes in the ST segment or T wave.
- Lead V6—Lead V6, the last of the precordial leads, is placed level with V4 at the midaxillary line. This lead produces a positive deflection on the ECG.

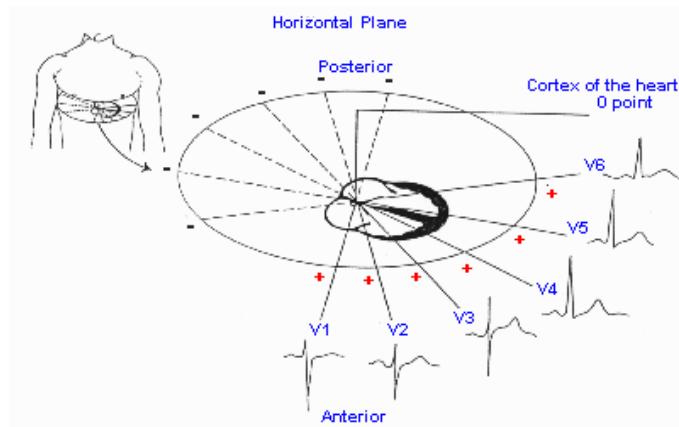


Figure 2.7: Precordial leads and their position related to the heart and the chest horizontal plane.

2.3.6 How to read a ECG record

Waveforms produced by the heart's electrical current are recorded on graphed ECG paper by a stylus. An ECG paper consists of horizontal and vertical lines forming a grid. A piece of ECG paper is called an ECG strip or tracing. The horizontal axis of

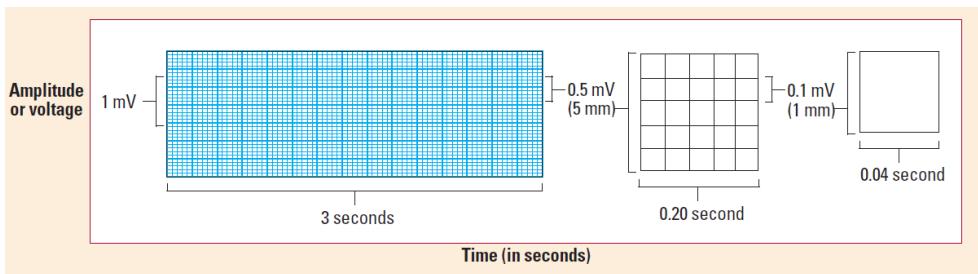


Figure 2.8: A typical ECG paper.

the ECG strip represents time. Each small block equals 0.04 second, and five small blocks form a large block, which equals 0.2 second. This time increment is determined

by multiplying 0.04 second (for one small block) by 5, the number of small blocks that compose a large block. Five large blocks equal 1 second (5×0.2). When measuring or calculating a patient's heart rate, a 6-second strip consisting of 30 large blocks is usually used. The ECG strip's vertical axis measures amplitude in millimeters (mm) or electrical voltage in millivolts (mV). Each small block represents 1 mm or 0.1 mV; each large block, 5 mm or 0.5 mV. To determine the amplitude of a wave, segment, or interval, count the number of small blocks from the baseline to the highest or lowest point of the wave, segment, or interval.

2.4 Noises and interferences

Obtaining a reliable ECG recording is still an issue. In fact there may occur many problems interfering with the signals. Some of these problems include artifacts from patient movement and poorly placed or poorly functioning equipment.

2.4.1 Artifact

Artifact , also called waveform interference, may be seen with excessive movement (somatic tremor). The baseline of the ECG appears wavy, bumpy, or tremulous. Dry electrodes may also cause this problem to occur due to poor contact.

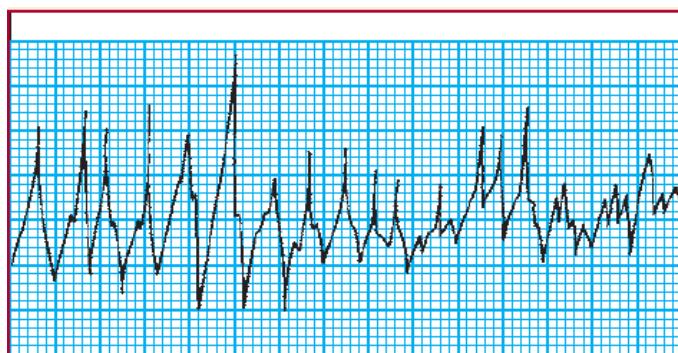


Figure 2.9: ECG waveform interference due to artifact may cause monitoring to fail due to unreadable signals.

2.4.2 Interference

Electrical interference, also called 60-cycle interference, is caused by electrical power leakage. It may occur due to interference from other room equipment or improperly grounded equipment. As a result, the lost current pulses at a rate of 60 cycles per second. This interference appears on the ECGs as a baseline that is thick and unreadable.

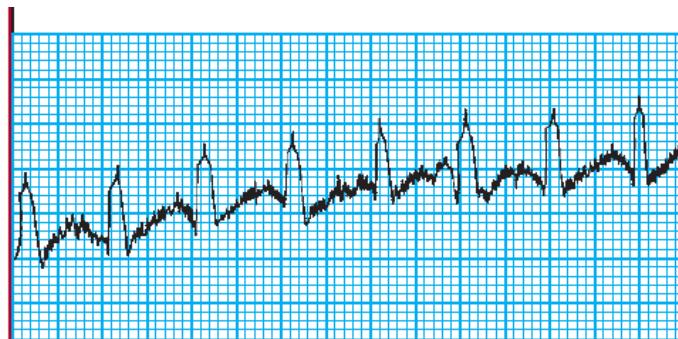


Figure 2.10: Electrical interference causes the baseline to be unstable and the signal is corrupted.

2.4.3 Wandering baseline

A wandering baseline undulates, meaning that all waveforms are present but the baseline is not stationary. It can be caused by movement if the chest wall during respiration, poor electrode placement, or poor electrode contact.

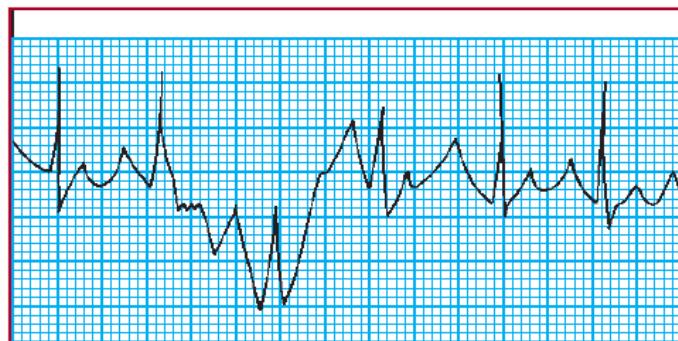


Figure 2.11: An example of baseline wandering due to artifacts.

2.4.4 Faulty equipment

Faulty equipment, such as broken lead wires and cables, can also cause monitoring problems.

Chapter 3

State of Art

3.1 Device

The personal health care market has changed a lots and recently new products and devices are showing up on the market. We will describe briefly the most relevant and similar products as mobile ECG acquisition devices. We evaluate the following solutions:

- Mortara ELI 10 Mobile
- Philips DigiTrak XT Holter Recorder
- M-Trace (PC) Mobile
- ECG Expert

3.1.1 Mortara ELI 10 Mobile

This device offers an all in one solution for 12 leads ECG acquisition. It is compact and complete as it provides an alphanumeric keyboard and a screen for real time visualization and the possibility to send the record via GPRS/3G channels. For each devices a SIM card is required . The device can also read and interpret the ECG supporting the doctor. Interesting feature is its great interoperability with the main ECG data management systems.



Figure 3.1: Mortara ELI 10 Mobile, ECG acquisition device box.

3.1.2 Philips DigiTrak XT Holter Recorder

This is the smaller acquisition device on the market. Thanks to a proprietary algorithm from Philips it can derive all the 12 ECG leads using only 5 leads. It weighs 62g and the internal battery lasts till 7 days. It also has a small screen showing 1 real time signal at a time.



Figure 3.2: DigiTrack, the ECG visualization.

3.1.3 AliveCor ECG

An innovative solution even though it doesn't offer a complete solution for ECG acquisition and analysis. This small sensor can be attached on the back of your smartphone making it an ECG acquisition device. It can record only one ECG signal (D1), so also the analysis is limited to a few types of arrhythmias . The record length is also limited to 5 minutes.



Figure 3.3: AliveCor device real time acquisition on a tablet.

3.1.4 M-Trace (PC)Mobile

M-Trace PC is an completed 12 leads ECG acquisition device. With the device it comes a mobile application and a desktop pc application used to visualize and analyze the ecg signals. The device is really portable with dimensions 95x64x28mm. The company offers also a more portable device (M-Trace Mobile) to be used by privates at their home. The mobile version cannot acquire a full record but only test records with 6 leads. Its main purpose it to send the test records via GSM/GPRS to the doctor for a faster review.



Figure 3.4: M-Trace PC device for ECG acquisition.

3.1.5 ECG Expert

ECG Expert produced by CSE Medical is a completed solution for ECG acquisition. The device comes with fully supported software for both PC desktop (Windows and Mac) both smartphones (Android, iOS). The device is rechargeable and makes use of a wireless connection via Bluetooth as exchanges data communication with the handheld smartphone or PC software.

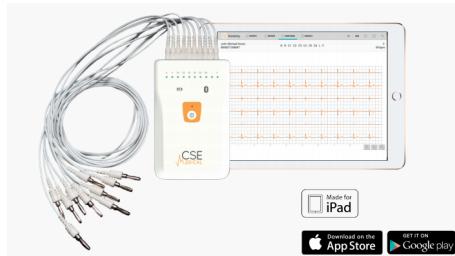


Figure 3.5: M-Trace PC device for ECG acquisition.

3.2 Mobile application

There are already mobile applications on the market store for ECG visualization and analysis supporting different formats. We can distinguish applications that only visualize the signal and the ones that also apply some analysis on the ECG signals. We listed only applications on the Google Play Store, so only Android applications because they are the only comparable with the solution we propose.

3.2.1 Visualization only application

The application on the market able to visualize the ECG signal are:

- *StribogECG*: an Android application based on an open source project under GLP v3 licence. It uses Biosig library to read ECG formats such as scp, xml (hl7), ecg and dgf. The software is only provided as it is and it requires to the user to already have the ecg files stored in those supported formats.
- *AndroidECG*: application on beta release, it was developed by Paco Gonzàles as thesis project during the Master course in Computer Science at the University of Murcia. The application is able to show ECG signals of the following formats: binary, scp, 212. As additional feature it is a basic analysis over the signal to detect QRS complex, P waves, ST segments and T waves. It is also possible to send the ECG record via email.

3.2.2 Visualization and Analysis

The applications on the market that also provide a more detailed analysis over the ECG signal are all bind to a specific proprietary acquisition device. By this way they lack the compatibility and interoperability requirement with other software and ECG formats.

- *M-Trace PC*: the application was developed by *M4Medical*, a Poland company providing medical devices for professionals and private customers. The application only works with the company 12-channel ECG M-Trace PC register device. The main features are the real-time monitoring interface, a patients' database management system and the possibility to share the record.
- *ECG Expert*: developed by *CSE Medical* the application works only connected to an ECG-Expert acquisition device. The main features are the real-time view of the acquisition, the analysis of the record providing information about QRS complex and heart rate, the possibility to manage patient information bind to the record and a heartbeat Normal/Abnormal classifier.

One last mobile application, which is not strictly related to ECG signal visualization and analysis but it worth to be mentioned, is the *ECG Interpretation*. This application instead provides enough detailed information about how to read and interpret the

ECG signals through 32 small lectures. All the lectures provide a picture and a short description and explanation.

Chapter 4

Objective

4.1 Preface

For a clear understanding of the next chapters we will make use of some terms listed below with the proper meanings:

1. Mobile application: it is a software running on smartphones and tablets
2. Desktop application: it is a software running on desktop pcs or notebooks
3. Acquisition device: named ZEcg, it is the device (hardware) used to acquire the ECG signal from the electrodes connected to a patient body.

4.2 Fully functional medical mobile app as replacement to desktop app

The main purpose for this thesis is to develop a medical mobile application as replacement to an original desktop application. The application needs to be standalone and independent from other softwares, still it can share its content and integrate other software content.

As a starting point we planned to reproduce all the desktop features such as the connection between the application and the remote device ZEcg for the ECG signal acquisition. It should also save the ECG records inside the mobile device, plot the signals and run arrhythmia recognition algorithms on them. We are aware that the user experience is different from a desktop one due to the differences in capabilities

and functionalities. Having in mind these differences, we didn't try to reproduce the desktop experience. We developed instead the application having a mobile experience at first position, following the standards of mobile application designs and principles. We took advantage of the new and latest technologies mounted on the new smartphones, trying to provide to the end user the best in term of user experience, performance and application design. The main difficulty is probably to redesign and re-imagine the desktop feature from a mobile point of view. For example, if a desktop application usually makes use of keyboard and mouse, inside a modern mobile application there is only the touch input as user interaction. The differences in term of screen size, memory and also cpu performance matters and should always be kept in mind during the initial planning phase. We will deal with these and others limitations, trying to achieve the best results and performances.

We believe this application can be really a replacement to a desktop application as the technology trends point, to future devices with better performances in term of lower power consumption and higher operational capabilities.[3]

Chapter 5

Requirement

In the project there was the need for a deep analysis of all the tied requirements. The result of this analysis was essential to identify the subsequent problems.

We will describe all of them, distinguishing between functional and nonfunctional requirements.

5.1 Functional

5.1.1 Connection management with the acquisition device

Fundamental feature to be included inside the mobile application is the capability to directly connect the smartphone device to the acquisition device ZEcg. Since this last one was designed to transmit the signals through a bluetooth channel, we have to implement and manage a bluetooth socket connection inside the application in order to receive the data.

5.1.2 Acquisition, storing and management of ECG records

For a matter of medical feature as it is a fact that there are many “standards” on saving an ECG signal, the application has to be able to manage different formats. Even though this application is designed to be used mostly for acquisition from the ZEcg device, it is also able to open and read other standard format such as the MIT-BIH, one of the most common standard in the literature of ECG. The code software behind is designed in a such way that the integration of other format is made extremely easy to add just by implementing few interfaces and classes.

5.1.3 Different ECG formats support

For a matter of medical feature as it is a fact that there are many “standards” on saving an ECG signal, the application has to be able to manage different formats. Even though this application is designed to be used mostly for acquisition from the ZEcg device, it is also able to open and read other standard format such as the MIT-BIH, one of the most common standard in the literature of ECG. The code software behind is designed in a such way that the integration of other format is made extremely easy to add just by implementing few interfaces and classes.

5.1.4 Dynamic display scaling

The mobile device market is huge and there are a very large number of devices with completely different hardware and screens. As first classification we can distinguish mobile devices into smartphones and tablets. The most obvious difference is based on the screen size and the pixel density. Building a mobile application means also to deal with these number of different devices. To achieve the same experience and look and feel the application should be able to scale its view according to the device screen and the pixel density. A typical ECG signal is plot on a paper with squares of well defined size in millimeters. The mobile application has to respect such a standard independently on the screens capabilities and pixel density, so it should be able to properly scale the view and the plotting based on the hardware provided by the device.

5.1.5 ECG record analysis integration on mobile platform

To complete the set of features for the application we plan to integrate the algorithms of ECG signal processing. To have a mobile device able not only to acquire and visualize in real time the ECG but also to analyze it at runtime, can be of vital importance, especially if the user has little knowledge about reading and interpreting an ECG graph. The integrated algorithms for arrhythmia analysis are based on a Neural Network trained to recognized the nature of the signals for the given record with high accuracy. The algorithms come from a previous thesis work[4] which belongs to Ulisse Pizzagalli, student at Politecnico of Milan.

5.1.6 Analysis results displaying

After analysis there are results that need to be shown to the user in the most friendly and understandable way. The most important results from an ECG analysis are called Istogram, Tacogram and the ST+/ST-. They are respectively graphs showing the number of heart rates of a certain value, the average heart rate at each heart beat and the difference between the area ST+ and ST-, the area above the segment ST and the one below. This last graph is useful for ischemia detection.[5]

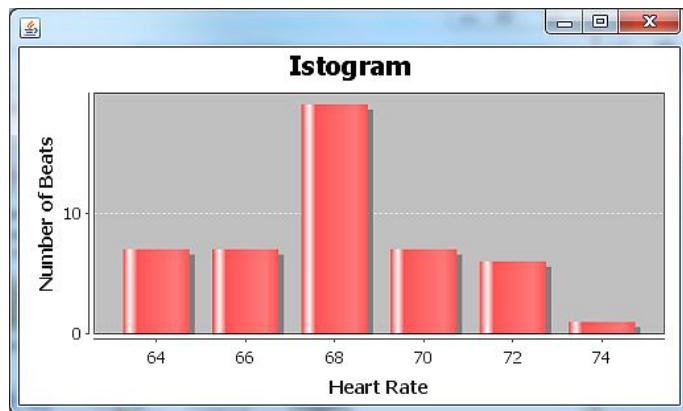


Figure 5.1: Istogram from the desktop application resulted from an analysis on a MIT/BIH record.



Figure 5.2: Tacogram from the desktop application resulted from an analysis on a MIT/BIH record.

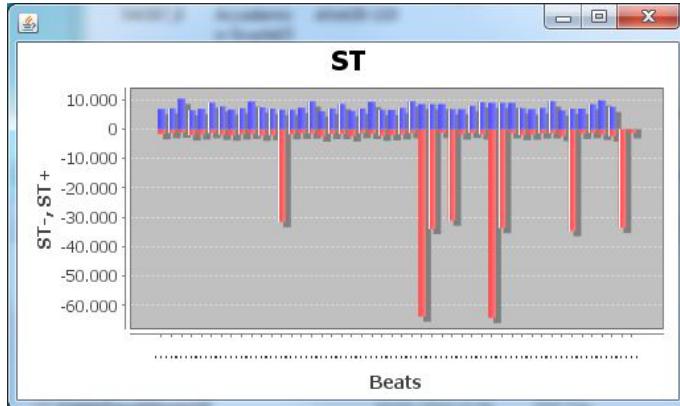


Figure 5.3: ST+/ST- graph from the desktop application resulted from an analysis on a MIT/BIH record.

5.1.7 Highly parameterizable

We believe in dynamic software, that is why we planned from the beginning on making this application dynamic. Even if the application is build on top of the ZEcg standard, we plan to make the software responsive also to other standards. To achieve this, we planned to abstract all the acquisition device independent features and functionalities. In order to support as much as possible any variants of the original acquisition device, we plan to setup customizable parameters, the only related to the hardware implementation. With a little of changes the application will be able to interface with other devices as well for acquisition.

5.2 Nonfunctional

5.2.1 Reduced memory usage

This requirement is fundamental for any project related to mobile application development. In fact, if a desktop pc in general doesn't have any problem related to memory usage (even if it is a good practice not to waste memory), on mobile devices this over-usage can bring the application to crash and get killed by the OS. The memory available is higher on new devices with respect to older ones, but it is still small so it is always a good practice to use it carefully.

5.2.2 Minimum performance rate and scalability on performance

Nowadays the new high level mobile devices has quad-core or even octa-core cpu processors. Any application should take advantage of a such configuration, but on the other hand mobile application developers should always consider the fact that the market is still full of older and low-end devices. In order to cover at least most of the market devices their application have to run fine (with a minimum acceptable performance rate) starting from the low-end devices and, at the same time, taking advantage of last devices capabilities.

We believe modern applications should seriously take this aspect in consideration, because it will make their application scalable also from a performance point of view.

5.2.3 Wide platform compatibility and accessibility

Developing a mobile application implies building a software that has to be executed on many different platforms. The smartphone and tablet market is huge with many different devices mounting different hardwares and running of the three major mobile OS (iOS, Android and Windows Phone). In the next chapters we will deal with this issue.

5.2.4 Documentation

This thesis includes also a more technical documentation about the development phase and the choices we starting from the planning phase to the development phase. The software is fully documented and with annotations and comments to increase code readability and future development on top of it. The technical documentation is included in the next chapters where we are going to discuss and motivate the implementation and the results.

Chapter 6

Problems

By identifying the requirements, we could then be able to highlight the related problems that we had to face in order to fulfill all of them. We will describe the related requirement as source of each faced problem.

6.1 Mobile platform fragmentation

Considering the requirement of a wide platform compatibility, we obviously need to face a really big problem in the mobile application world: the platform fragmentation. Starting from the first smartphone release on 2007, the iPhone from Apple, the sale of such devices keeps increasing each year. Between 2007 and 2008 sales proceeded upwards reaching the same sale rate of their computing parent, the PC. On 2009 the market signed an important inflection point, representing the beginning of an inexorable vertical rise. Although PCs were still the only ones to offer some types of functionality due to their longer replacement cycle, they were sold with a ratio of 1 : 2, compared to smartphones, over a 5 year period. This new market' growth leads to an obvious seeking of various participants, some by choice and some by necessity, in order to extract value. Android has been the prime beneficiary, having been announced on 2007 and having gone on to account for well over half of smartphone sales worldwide. Apple, meanwhile, has maintained a steady ship, leveraging its skill in product design and user experience with a finely honed marketing machine, attracting a customer that rarely defects.[6] Fast-forwarding to 2016, if we consider the global market share held by the leading smartphone operating systems, at first position we have Android with a share of 80.7%, followed by iOS with a 17.7%. Minor percentages are represented

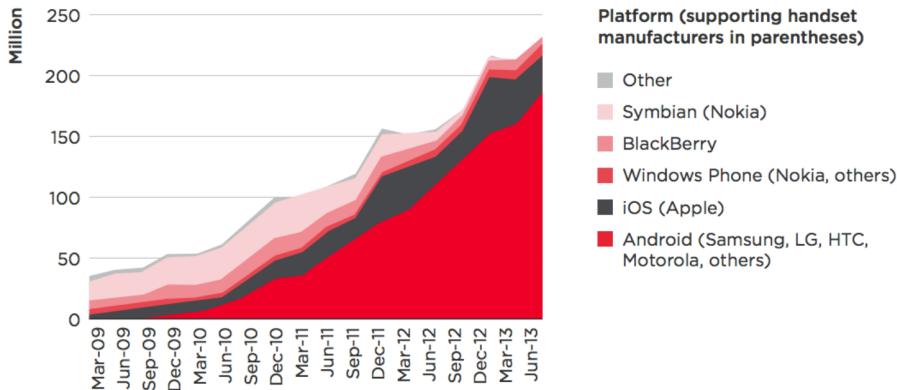


Figure 6.1: Mobile platform share evolution (smartphone sales), 2009–13.

by Windows Phone (1.1%), RIM (0.2%) and others (0.2%).[7]

6.2 Native vs Cross-Platform

As mentioned earlier, one of the main challenges when moving to a mobile solution is the software development within a technology landscape that is highly fragmented and rapidly evolving. Mobile apps require a fair amount of customization to run on diverse platforms and a constant update due to the steady stream of new hardware, OS versions and browsers. Even a single platform (Android, Windows, Blackberry, and to a lesser extent Apple) has numerous flavors that require some degree of customization. There are also other factors such as the overlay software from different manufacturers that can affect behavior of an app on a particular device.

In response, the mobile industry has spawned a rapidly growing ecosystem of cross-platform and cross-device frameworks, source code analyzers, libraries of reusable components, and other tools designed to accelerate and simplify multi-platform development. New tools are constantly emerging, with new functionality, different capabilities, strengths and weaknesses.

Developer's preferences change and evolve, particularly as new tools and capabilities become available. However, the basic goals are the same: to code less and accomplish more, to reuse and recycle across multiple platforms as much as possible, and consider developing from scratch as the last resort. In addition, any tool or framework should be able to work with current and future evolving offers, and not to be locked because of a particular platform or technology.[8]

6.2.1 Native

Native app development involves developing software specifically designed for a specific platform, hardware, processor and its set of instructions. Typical programming language are Java, Object C, Swift, C# and many other.

Native apps' major advantage over cross-platform apps is the ability to leverage device-specific hardware and software. This means that native apps can take advantage of the latest technology and API available on the mobile devices and it can well interface with other platform apps. Other advantages are a predictable increase on performance, streamlined support, native UI, native API and coherent library updates. However, the mobile platform's fragmentation makes the task of keeping up with the pace of emerging technology onerous and costly, having to develop different software for each of the different platforms (Android, iOS, Windows Phone, Symbian).

Going further on a more technical analysis, native applications are represented by executable binary files that will be installed into devices without the need for other abstraction layering to the operating system. They are able to call built in functionalities such as calendar, notifications, the dialer, email provider and others services and functionalities provided natively by the OS. Despite the fact that native applications development requires platform specific skill and expertise, this strategy delivers a higher quality user experience than other mobile application development methods (cross-platform or hybrid approach). Native apps are also best distributed through an app store.[9]

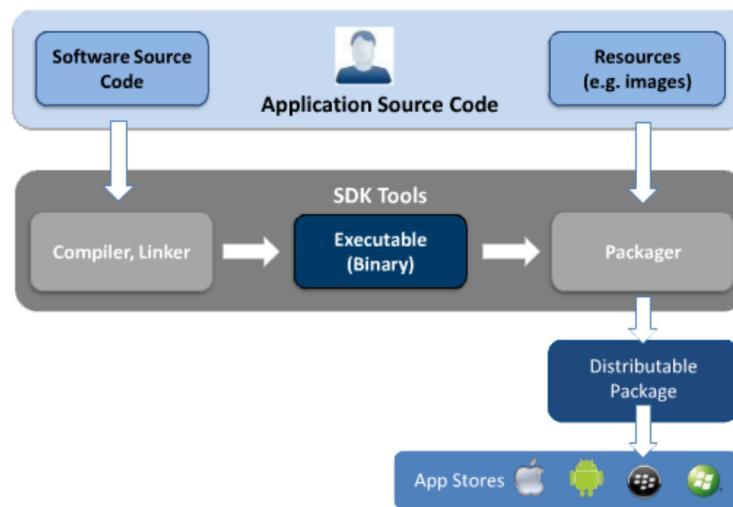


Figure 6.2: Native app development process.

6.2.2 Cross-Platform

Cross platform development produces one code base to maintain and write targeting multiple devices and platforms. It promises lower time of development and costs. The main categories of this group are web, hybrid, interpreted and generated apps. None of the previous is neither prevalent nor the best solution to the problem of developing cross-platform mobile applications. The main benefits of cross-platform development

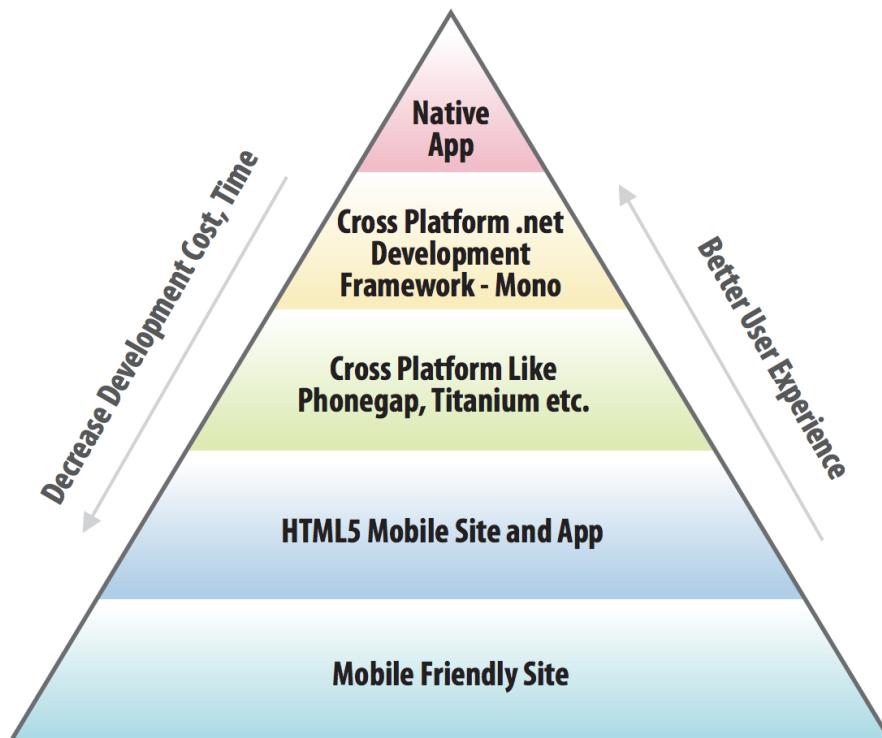


Figure 6.3: Native vs Cross-Platform cost and time factors.[2]

are:

- Reduction of required skills to develop applications due to the use of common programming languages;
- Reduction of coding work, because the source code is written once and it is compiled for each supported OS;
- Reduction of development time and long term maintenance costs;
- Decrement of API knowledge, because with these tools is not needed to know the API's of each OS, but only the API's provided by the selected tool.[10]

6.3 Mobile hardware limitations

As this project started from a previously developed desktop application, we needed to take into account all the limitation of mobile devices hardware. Despite the trend of mobile hardware is a constant increasing in computational power and memory availability, we cannot compare it with the one available nowadays in PC architectures, as the architectures on this two platforms are totally different: most of all pc are based on x86 architectures, whereas mobiles rely on ARM architectures. Furthermore, we could not commit the error of targeting only recent devices as they represent only a small percentage of all devices in the market, and it would have been in contrast to our main goal, which is a wide device compatibility. We will discuss briefly about memory and computational differences in these type of architecture in the next paragraphs.

6.3.1 Memory limitations

The first point to consider is the device memory limitation. In the previous desktop application not so much effort was spent to reduce memory allocation: all the ECG record samples were allocated in memory both during acquisition and record opening. This approach could result convenient considering its low implementation effort, but it could be source of lots of problem when moving to a mobile device, where not only the physical memory are often small, but also the memory reserved to an application is limited (figure 6.4), both due to the coexistent apps running in the system and the power management of the mobile OSs. So we needed to find a good approach in order to manage these problems and not to face with memory leakages.

6.3.2 Performance limitations

Talking about cpu computational power there is a huge gap between the two architectures.. Not considering the evident difference for what regarding the space availability for the CPU package, the main difference is that ARM architectures are focused on reducing the power consumption, a crucial characteristic in mobile devices. A technical report analyzed the execution time of a benchmark using a cross-platform version of WebKit showed that the tested ARM A9 architecture had a ratio of 5.8 (normalized time) compared to the tested x86 architecture.[11]

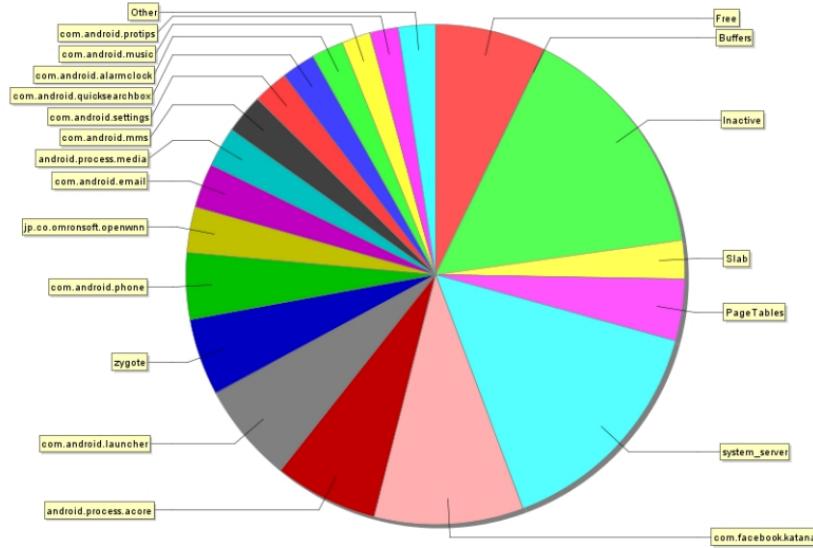


Figure 6.4: Memory reserved to an app in Android.

6.4 ECG baseline wander

For a right interpretation and identification of physiological and pathological phenomena, a low error is required in the ECG signal. But often ECG recordings are distorted by two main artifacts:

- high-frequency noise caused by electromyogram induced noise, power line interferences, or mechanical forces acting on the electrodes;
- baseline wander (BW) that may be due to respiration or the motion of the patients or an instrument fault (Figure 6.5).

These artifacts strongly limit the utility, readability and interpretation of a recorded ECGs. In ECG enhancement, the goal is to separate the valid ECG from the undesired artifacts so that we can extract a signal that allows an easy visual interpretation.[12] The first class of artifact is mostly solved within the acquisition device at hardware low-level though filters and a proper settings. In this project we will deal with the second class of artifact which can be managed at software level.

6.4.1 State of Art

Two methods, in recent years have often been applied to remove BW:

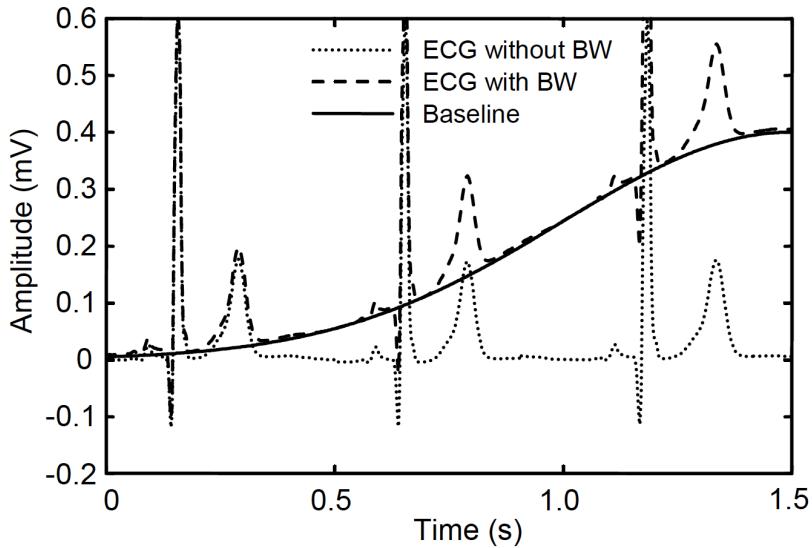


Figure 6.5: The baseline wander artifact in the ECG signal.

- Polynomial fitting: in order to assess the baseline, it uses a polynomial interpolation. The baseline is fit from some fiducial points that are determined from P-R intervals, whereas these fiducial points are difficult to accurately locate before noise is removed from the ECG signal. As result this approach is ineffective if the ECG signal is contaminated by noise.[13]
- High-pass filtering: to implement this type of filtering, usually a moving average filter[14] and wavelet translation[15]. This approach however would unavoidably introduce distortions in various parts of the ECG signal, especially in the ST segment due to the spectra of the ST segment that overlaps the spectra of BW.

Simple Moving Average Filter

This type of filtering is defined as

$$y(n) = \frac{1}{2N+1} \sum_{i=-N}^N x(n+i) \quad (6.1)$$

where $x(n)$ and $y(n)$ are input signal and output signal of the moving average respectively, and N specifies the observation window length equal to $2N + 1$. So the

baseline can be estimated as

$$z(n) = x(n) - y(n) \quad (6.2)$$

where $z(n)$ is the output signal of the high-pass filter.

Distortion using Simple Moving Average Filtering

Obviously the baseline values estimated from the P-R segment (between 0.3 s and 0.6 s) are very close to real baseline values, while the baseline values estimated from segments including QRS complex and T wave are far away from the real baseline. Therefore, if an observation window covers some sample points with extreme amplitudes, an ECG signal would be distorted.

CPU intensive operation

Many methods are known in the literature that perform better and reduce the error in the ECG signal due to the filtering, like wavelet package translation filter[16], or the statistical weighted moving average filter[17]. Defined the normalized root mean square error ($NRMSE$) and maximum error (ME) as

$$NRMSE = \sqrt{\frac{\sum_{i=1}^L (ecg_{in}(i) - ecg_{out}(i))^2}{\sum_{i=1}^L ecg_{in}(i)}} \quad (6.3)$$

$$ME = \max_{i=1,2,\dots,L} |ecg_{in}(i) - ecg_{out}(i)| \quad (6.4)$$

The figure 6.6 shows a comparison between the methods. Of course the computational complexity these advanced methods is really high. Considering that this type of filtering is supposed to be used during a real-time acquisition in our project, they are obviously prohibitive.

6.5 Signal visualization

The problem of an efficient ECG signal visualization shows up when the application has to plot samples of an acquisition device with a rate of 500Hz. Because of this the app has to be able to draw $500 - 1$ lines (connecting two samples) for each mapped second in the ECG paper grid. Taking into account the standard size of a second

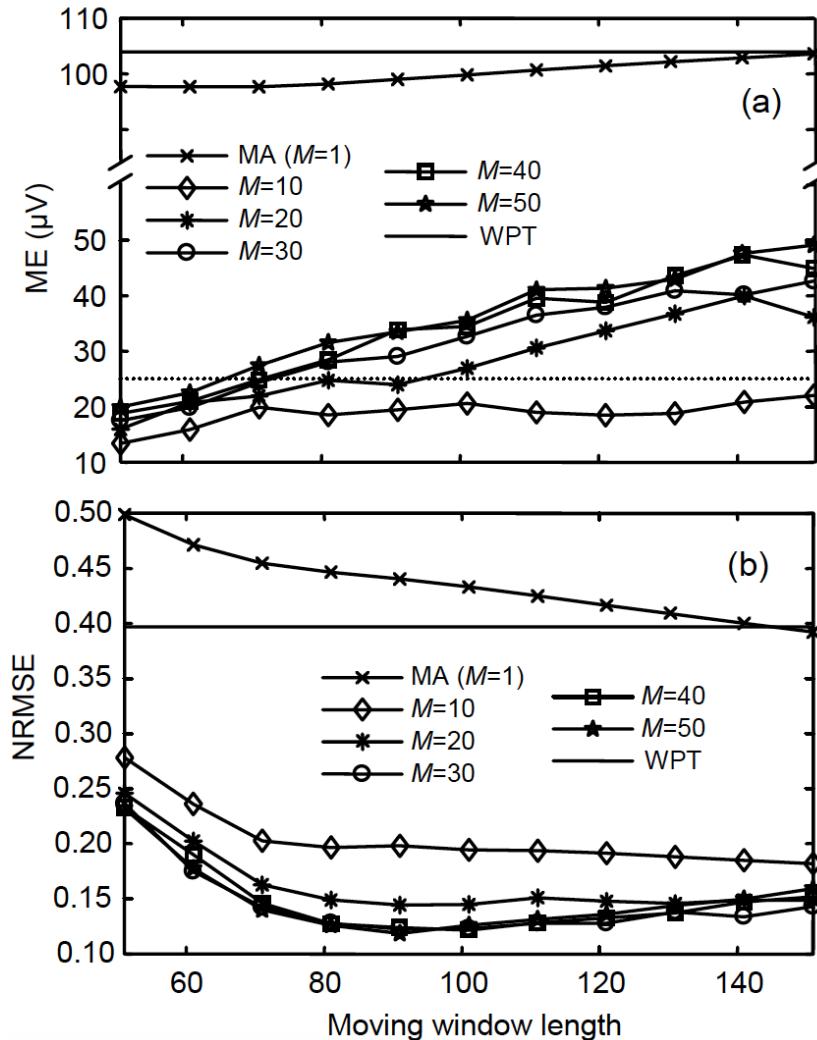


Figure 6.6: Relation between distortion and moving window length for wavelet package translation (WPT), traditionally used MA, and our proposed high-pass filter based on a statistical weighted moving average (SMA).

(a) Maximum error (ME) vs. moving window length;

(b) Normalized root mean square error (NRMSE) vs. moving window length. M is the number of sub-bounds. SMA is the same as MA when M=1.

in the standard ECG paper of 5mm, and the fact that the visualization size has to respect this measure, if we consider a device screen of 4.6 inch in portrait, we have around 11 seconds of samples to visualize. Hence we have $(500 - 1) * 11 = 5489$ lines to plot. If we are supposed to plot ideally all the ECG derivation (12), the number of total lines become $5489 * 12 = 65868$. Now, this is the number of lines we need to plot in order to cover the total visible window screen space. If we want to

have a good display smoothness we must guarantee at least 30FPS (frame rate per seconds), ideally 60 frame/s (FPS). Assuming a minimum of 30 FPS we have a total of $65868 * 30 = 1976040$.

These numbers gives us an idea of the computational effort that is spent only to plot the samples.

Chapter 7

System architecture

The entire system is based on an acquisition device named Zecg, and a native mobile application on the Android platform. Our focus and main effort were on the developing of the mobile application fulfilling all the requirements. The acquisition device was instead developed and designed by Crespi Alessandro and Ulisse Pizzagalli during their thesis work at the Politecnico of Milan.

7.1 Acquisition device

Zecg is composed by the following different modules:

1. OVP: used to protect the patient from high voltage or voltage leakage.
2. LFP EMI Filter: anti-aliasing RC filter used to remove noises due to high frequencies.
3. RLD: Active electrode driver for the right leg
4. WCT: Derivator used to compute the precordials (Wilson Central Terminator)
5. PGA: 8 gain amplifier with programmable inputs
6. ADC: 8 analog-digital 16 bit 8KSa/s converter.
7. MCU: microcontroller
8. Bluetooth: bluetooth module for data transmission

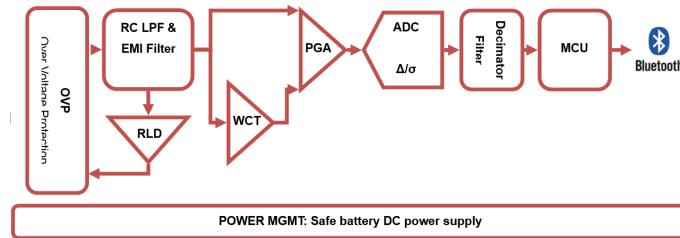


Figure 7.1: ZEcg device block diagram including all the modules components.

9. Power MGMT: to manage battery recharge and stabilizer

The core component is the Texas Instrument system on chip ADS1198. This chip has 8 input bipolar channels, representing the 8 clinical leads.

The channels 1 and 2 produce the lead I and lead II. Channel 1 measures the potential difference between the electrode RA(-) and the electrode LA(+), the channel 2 the difference between RA(-) and LL(+). Lead III and the augmented leads are obtained from a combination of lead I and lead II at software level.

V1, V2,...,V6 are computed as difference between the respective electrode and the signal related to the negative value of the WCT (Wilson Central Terminator).

The WCT signal comes from the average between RA, LA and LL and it's connected to the channels 3,4,5,6,7,8. Each channel is amplified using a programmable gain (PGA) and a CMRR, before being converted into digital. For more details about the entire device architecture we invite you to read the thesis of our colleagues Crespi Alessandro and Ulisse Pizzagalli[18] who designed and developed zecg.

7.1.1 Mobile app

The mobile application we developed for this thesis work was implemented having in mind all the user best interface design principles and the best practice starting from the planning and design phase to the final coding phase. As this application was designed for Android OS, we strictly followed Google specific standards and procedures.

We made use of Android Studio as IDE (strongly suggested by Google as main IDE to develop Android native applications). Starting from 2013 the development of native Android application moved from Eclipse + Android Plugin to Android Studio. The entire project building system has changed and moved to use the Gradle building

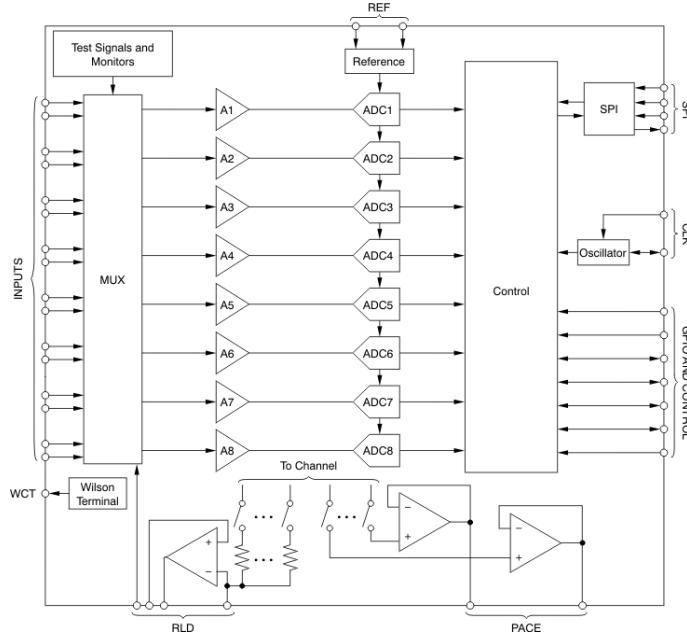


Figure 7.2: ADS1198 functional diagram showing the 8 channels representing the 8 leads used during the ECG record acquisition.

system[19]. The entire process during project building to compilation can be resumed in the following image: The most important steps along the building process are:

1. The Android Asset Packaging Tool (aapt) takes the application resource files, such as `AndroidManifest.xml` file and the XML files for the Activities, and compiles them. A `R.java` file is produced so that all the resources can be easily accessed within your application.
2. The aidl tool converts any `.aidl` interfaces into Java interfaces.
3. The Java compiler will compile the `R.java` and `.aidl` files generating the `.class` files.
4. The Dex tool will convert the `.class` files to Dalvik bytecode. Any 3rd libraries and `.class` files included in the project build will be also converted into `.dex` files so that they can be later packed into the final `.apk`.
5. All non-compiled resources(such as images), compiled resources, and the `.dex` files are sent to the apkbuilder tool that will output the `.apk` file.

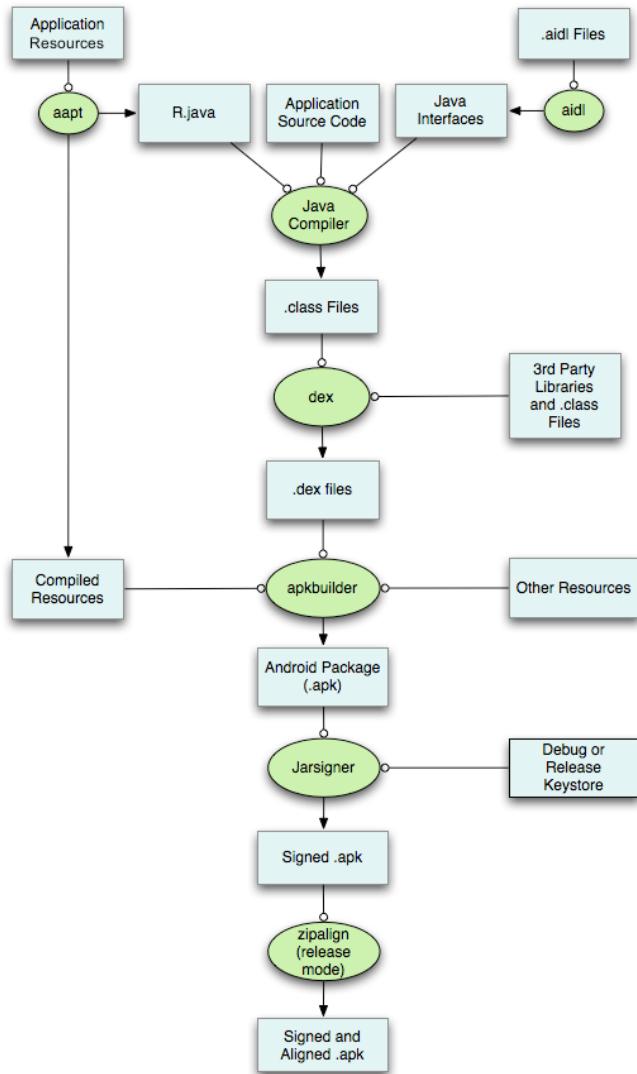


Figure 7.3: Android Build system process. How and which component are involved during an android application build and compilation.

6. Once the .apk is built, it must be signed with either a debug or release key before it can be installed to a device.
7. To reduce the size of the .apk and to decrease the memory usage for releasing mode the zipalign tool is launched.

We split the application functionalities into different packages. The packages content are divided as follow:

- Activity: it contains all the Activity used inside the Application.

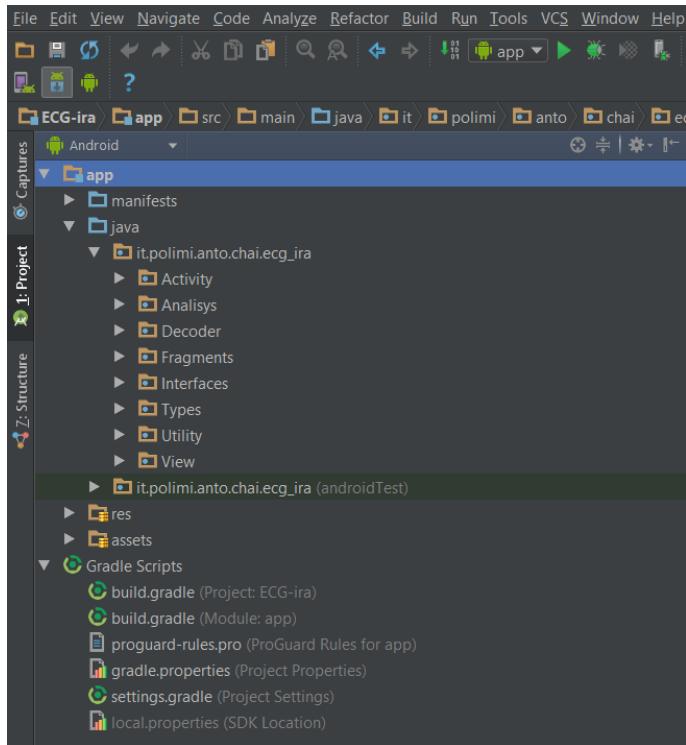


Figure 7.4: ECG-ira project structure and packages inside Android Studio IDE.

- Analysis: All the java classes used for the analysis of the ECG records; they include the Neural Network, the QRSDetector, the PWaveDetector and some signal Filters implementations.
- Decoder: this package contains the java classes in charge to decode and parse the .hea and .dat files.
- Fragments: contains some fragments used inside the application, for example the ones used to show the chronograms (Istogram, Tacogram, ST graphs).
- Interfaces: all interfaces declarations to abstract the specific application behaviour with more general one.
- Utility: some utility class such as the in application FileManager.java class, and the Adapters used within the application (for example to show ListViews or RecycleViews).
- View: it contains all the classes working directly on the View and the custom view themselves such as the View used to show the ECG grid.

The key points of this application are based on three main concepts:

- Code reuse and maintainability
- Maximize performance depending on the device constraints (cpu, memory, storage capacity)
- Usability in term of interface and interaction

Code reuse and maintainability

The most important classes representing the core of the application functionalities are all abstracted through interfaces or in case of the classes related to the View through a set of parameters. This makes the application highly customizable and easily to extend. For example if in a next future there will be a new ECG data format, it can be possible to give the app the capability to read and parse such a format just by implementing the interfaces and writing the specific code to parse such a new data format.

A concrete example is given in our code by observing that both the MITDataReader and the ZEcgDataReader extends the abstract class DatReader, so the next format reader has just to extend it as well. We know that MIT-BIH format is completely different from the ZEcg format, that why specific code to parse the file content to extract the signal is mandatory.

Any new format reader has just to override the method:

```
public void readAndAddSample(int num0fSample, int addPosition) {  
}
```

More concrete details about the effective implementation will be exploited later on in the next chapters.

Maximize performance depending on the device constraints

We do well know about the great number of constraints due to the huge number of different devices with different hardware on. We decide to make our application available starting from Android API 16 (the minimum sdk API recommended by Google to support). To overcome the problem of the difference devices and Android OS version starting from Jelly Bean (API 16) we took maximum advantages from

the device hardware by splitting the thread jobs in between the maximum number of cores available. In the class CpuInfoExtractor we discover the hardware capabilities (number of cores) and according to it we split the calculus between a certain number of threads. More the cores, more the threads we can take advantage of.

From an interface point of view instead, we are forced to depend on the density of pixel of the device screen and its size, but at the same time to accomplish our requirements related to achieve a perfect grid of squares of centimeters. We found a way to always achieve the same dimensions of square independently by the screen size of the devices by retrieving and taking in account the display exact pixels per inch size in the X dimension. Then, we computed the number of pixels needed to achieve the right sizing. In this way, we solved the problem of having same dimensions and so metrics on different devices. On the other hand, if this method is quite functional and device independent, on devices with different screen size (width for example) the number of grid cells may vary from just a few on small screen, to many of them on tablets. This is a direct consequence due to the difference in number of pixels and the pixels size itself (some are square, most of them are rectangle).

Usability in term of interface and interaction

One mobile application is usable if it does what the user expects it to do when interacting with it.

We followed all Google guidelines in term of user experience using native view and patterns. We took advantage of the last API features, such as RecycleViews instead of the old classic ListViews. We made use of the button ripple effects available starting from Lollipop (Android API 21), but at the same time we provided to the older Android OS version the selector effect which still gives a nice response to the user interaction with command and buttons. We made use of the typical android Preference Settings so familiar to Android users and most important we always inform the user about the operations going on so he never feels lost inside the application.

Chapter 8

Final result

ECG-ira (ECG instant rapid analyzer) is a fully working mobile software application able to replace and fulfill most of the functionalities of an analog desktop application. The solution was designed according to the best mobile application development principles and it resulted working fine on all the smartphones[20] we tested it on. In this chapter we will show the final results providing significant screenshots of the application and metrics related to the performances evaluated and calculated with the application running on different devices.

8.1 App Screens

By launching the application the first screen is the home screen. It is the starting point, here the user can create a new record (starting a new acquisition with the ZEcg device), he can open the list of records (previously acquired or already present in the device system folder), he can discover and establish a connection with an acquisition device for future acquisitions and he can access to the application settings section where he can customize the application behaviour and tune some settings like the default folder to store in and open the records from.

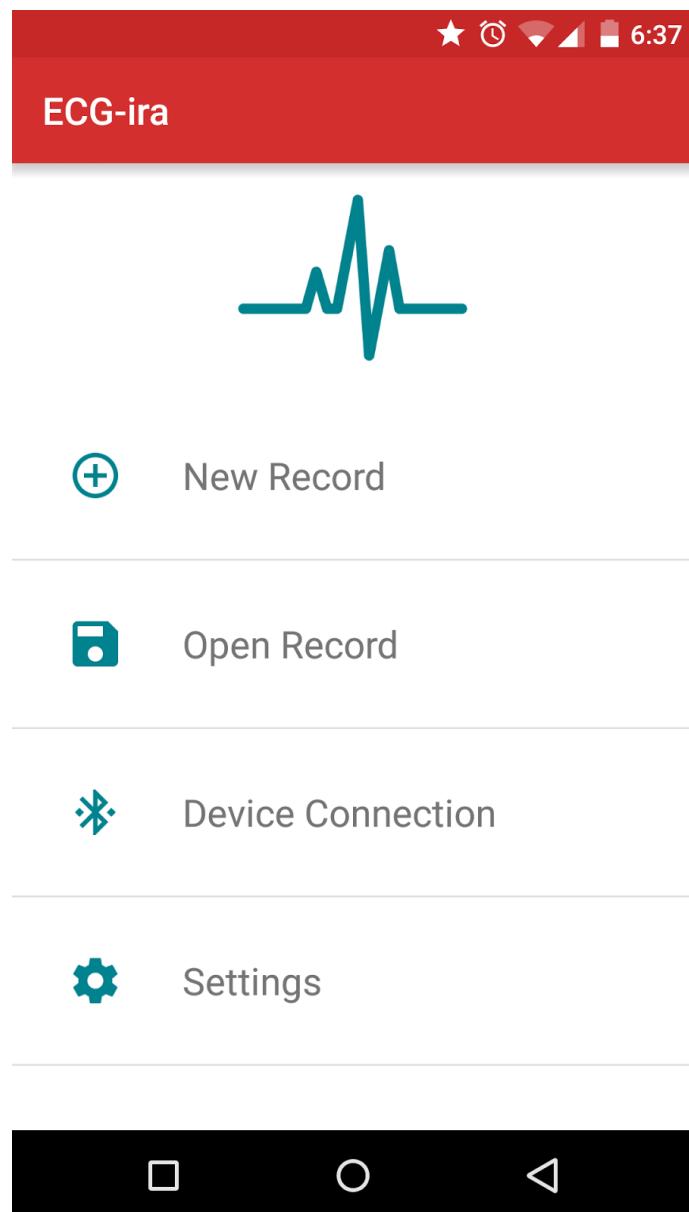


Figure 8.1: The home screen of the application ECG-ira.

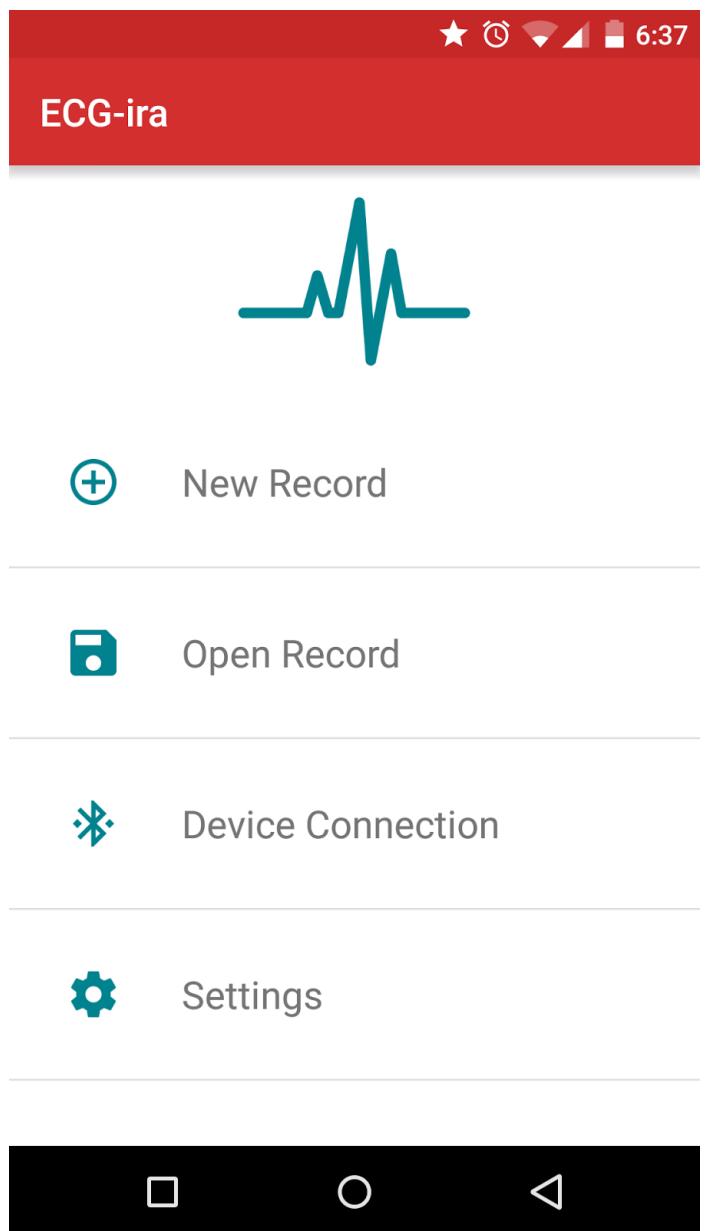


Figure 8.2: The form to be compiled before a record can start. It is necessary to bind any record to a patient data in order to avoid confusion between ecg records. This data is also useful for a fully understanding of the record itself.

Acknowledgement

Desidero innanzitutto ringraziare...

Bibliography

- [1] Blausen Medical Communications. Heart illustrations.
- [2] AllianceTek. Comparison chart for mobile app development methods. Norwegian University of Science and Technology, 2011.
- [3] Mediatek helio x20 - the world's first mobile processor with tri-cluster cpu architecture and ten processing cores (deca-core) and 30consumption: <http://mediatek-helio.com/x20/>.
- [4] Simone Battaglia Diego Ulisse Pizzagalli. Algoritmi per il riconoscimento automatico di eventi aritmici ed ischemici in segnali ecg. Master's thesis, Politecnico di Milano.
- [5] Nicos et al Maglaveras. An adaptive backpropagation neural network for real-time ischemia episodes detection: development and performance analysis using the european st-t database. *Biomedical Engineering, IEEE Transactions*, 45(7):805–813, 1998.
- [6] GSMA Intelligence. Mobile platform wars, Feb. 2014.
- [7] Gartner. Global market share held by smartphone operating systems, Dec. 2015.
- [8] Lionbridge. Mobile web apps vs. mobile native apps, 2012.
- [9] Redda Yonathan Aklilu. Cross platform mobile applications development. Norwegian University of Science and Technology, 2012.
- [10] Palmieri Manuel, Inderjeet Singh, and Antonio Cicchetti. Comparison of cross-platform mobile development tools. In Intelligence in Next Generation Networks (ICIN), editor, *16th International Conference*, pages 179–186, Oct. 2012.

- [11] Emily Blem, Jaikrishnan Menon, and Karthikeyan Sankaralingam. A detailed analysis of contemporary arm and x86 architectures. *UW-Madison Technical Report*, 2013.
- [12] Blanco-Velasco, Binwei Weng Manuel, and Kenneth E Barner. Ecg signal denoising and baseline wander correction based on the empirical mode decomposition. *Computers in biology and medicine*, 38(1):1–13, Jan. 2008.
- [13] Burattini L, W Zareba, and R Burattini. The effect of baseline wandering in automatic t-wave alternans detection from holter recordings. *Computers in Cardiology*, pages 257–260, Sep. 2006.
- [14] Leski, Jacek M, and Norbert Henzel. Ecg baseline wander and powerline interference reduction using nonlinear filter bank. *Signal processing*, 85(4):781–793, 2005.
- [15] Xu and Lisheng et al. Baseline wander correction in pulse waveforms using wavelet-based cascaded adaptive filter. *Computers in Biology and Medicine*, 37(5):716–731, 2007.
- [16] Hu, Xiao, Ying Gao, and Wai-Xi Liu. Pattern recognition of surface electromyography signal based on wavelet coefficient entropy. *Health*, 1(02):121, 2009.
- [17] Hu, Xiao, Zhong Xiao, and Ni Zhang. Removal of baseline wander from ecg signal based on a statistical weighted moving average filter. *Journal of Zhejiang University SCIENCE C*, 12(5):397–403, 2011.
- [18] Crespi Alessandro and Ulisse Pizzagalli. Zecg: Acquisizione, trasmissione e memorizzazione del segnale ecg. Master’s thesis, Politecnico di Milano.
- [19] Benjamin Muschko. *Gradle in Action*. Manning, 2014.
- [20] Samsung galaxy siii mini, samsung galaxy note, oneplus x, sony xperia z3 compact, sony tablet z, huawei p8 lite.