

Chapter 3

Software for an mHealth System

Abstract A description of software requirements for an mHealth system is provided. Several layers of software are required for proper system functioning: ECG body sensor firmware, applications on smartphones, Cloud applications, and stand-alone applications for personal computers. The main functionality requirements of the mHealth software are described. Additionally, approaches for fulfilling those requirements are proposed.

3.1 Hardware Infrastructure

An mHealth system is powered by a complex ICT system and comprised of heavily inter-dependent functional modules. First of all, an mHealth system requires a device with computing, storage and interconnection capabilities, which the user can carry around at all times. So-called smart devices, such as tablets, smart watches and the ubiquitous smartphones, can readily fulfill this role [1]. From here on, we shall refer to smartphones only, because currently they are leaders among personal assistant devices. In the future, smart devices of some alternative form could take over this role.

Second, a sensing device that can measure one or more parameters related to the user's well-being is required. Smartphones could be used also for this purpose, but the system becomes far more efficient if it relies on dedicated sensing devices. A dedicated device can be tailored in design for accurate measurements of specific parameters. For medical-grade ECG measurements, for example, constant electrical contact of the measuring device with the chest is required.

A diagram of an mHealth system for ECG monitoring is given in Fig. 3.1. The figure depicts the user's view of the relevant ICT parts of the mHealth platform and the communication between them. The central component of the system is the user, interacting with the remaining components through a smartphone, a personal computer (optional) and an ECG body sensor. The role of the remaining components

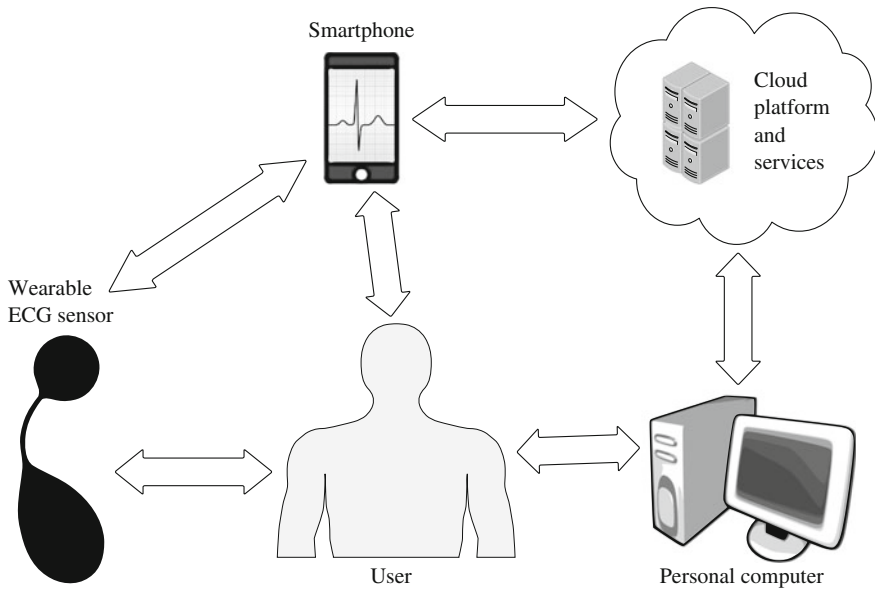


Fig. 3.1 The ICT components in an mHealth system from the user's point of view. The user interacts with an ECG body sensor, a smartphone and a personal computer, while Cloud services take care of data storage and processing. The smartphone acts as a hub for transferring data from the ECG body sensor to the Cloud. The smartphone and the personal computer both serve as an interface towards data stored on the Cloud

is twofold: to support the user in proactive health management and to collect data for analysis by medical practitioners. The ECG body sensor is used for sensing, that is, for measuring ECG, and for providing ambient and contextual information about the measurement. The ambient and contextual information can additionally be used to help in the analysis of the acquired ECG measurements. The user's interaction with the ECG body sensor is very limited—the user only wears the device. The smartphone is used for additional sensing, processing, interfacing with the ECG body sensor, visualization and alerting. It is the primary interface of the user towards the rest of the mHealth system. A personal computer would be used if the user wishes to analyze the acquired measurements beyond the scope of the real-time analysis provided by the smartphone. The Cloud platform represents the data link between the smartphone and the personal computer, and adds additional processing power, storage options and means of communication between the user and medical practitioners. The Cloud platform also enables access to the measured to the medical practitioners for diagnostic purposes.

3.2 Software

Several layers of software are required for proper system functioning. A coarse-grain division of the software into parts, according to where these parts are executed, is given below.

- *Custom software, also termed firmware, on the micro-controller-based sensing device.* The firmware controls data sampling and the storage, the analysis and the wireless transmission of sensor data to other modules.
- *An application on the smartphone.* The smartphone application acts as an interface between the mHealth system and its users, providing a convenient way for interaction with the sensing device. From the user's point of view, the application represents the core of the mHealth system. Thus, a user-friendly application design should be seen as one of the most important goals in the implementation of an mHealth system.
- *Cloud services implemented on the top of a Cloud platform.* Cloud services are already an established technical solution for data storage and analysis. They can provide custom views on the online measured health data for medical practitioners, for the users of the system, and potentially also for their caregivers. These services can also be seamlessly connected to the patient's electronic health records, which is crucial for making personalized diagnoses.
- *A set of computer programs that run on personal computers or mobile devices, able to process and visualize measurements off-line.* These applications represent an access interface to the Cloud services for medical practitioners and additional means of interaction with the mHealth system for its users. They could be built on the existing software used by the medical practitioners, by extending it with a module for accessing the Cloud services of the mHealth service. New software, which would be adapted to the type of the measurements gathered by the mHealth system, could also be developed. An example of an existing software suit that was modified to support measurements obtained from an mHealth system is described in Sect. 6.2.1.

3.3 mHealth Requirements

The main difference between mHealth and classical state-of-the-art ambulatory ECG recorders is in the user interface. A novel interface towards the user can transform an ambulatory recorder from a burden to an indispensable equipment for every-day life. On the other hand, the smartphone makes for a great platform for an mHealth system—it has become almost ubiquitous, it offers ample computing capabilities, and it provides means for communicating data along healthcare pathways.

3.3.1 Healthcare Requirements

The implementation of an mHealth system is driven by the needs to cut healthcare costs and promote “personal health systems” in the sense of higher levels of healthcare personalization and a more active role of patients in their health management [2]. For mHealth ECG monitoring, this can be achieved by ECG body sensors. With such sensors, a part of the monitoring outcomes can be made available to the patients themselves. Furthermore, patients can be given some control over the devices used in the measurement. In other words, an appropriate user interface can make the difference between a product used only by enthusiasts or a niche of professionals, and a product intended for the general population.

Next, making the user interface user-friendly should be one of the primary goals. For example, seniors are more likely to develop critical health conditions and would benefit the most from the availability of healthcare services at home. Seniors are in general less tech-savvy than the average population and therefore feel less comfortable when using modern technology, which they usually consider unintuitive. As a result, as a recent study says, “Digital health is not reaching most seniors” [3]. Therefore, an mHealth system should be designed with seniors in mind—to be simple and intuitive also for them, and to be able to provide only relevant information or alerts to specific users.

User friendliness should be the key goal in software development, since it could be crucial for the acceptance of the mHealth solutions. Another software development goal should be easy integration of the system into users’ everyday lives, which can further increase user acceptance. The system should thus be smart, i.e., interconnected into the network of existing devices and into the everyday routines of its users. Various social networks and the existing ecosystems of healthcare devices, applications and behaviors should also be considered. Finally, the option of future expansions should be left open.

The smartphone application has already been identified as an important part of the mHealth system, because it represents a user interface to the entire system. Nowadays, it can be considered also a very convenient one, since smartphones are becoming an increasingly large part of users’ lives, and because they can provide interconnection with other services and applications, and means for automated upgrades and expansions. Since the mobile application is such an important part of the mHealth system and offers so many opportunities, it is described in more detail in the following subsection.

3.3.2 Smartphone Application

We begin this section by describing the basic functionalities of the smartphone in an mHealth system: data storage and visualization, data transmission to other parts of the mHealth system, and user alerting. The description will be general and should

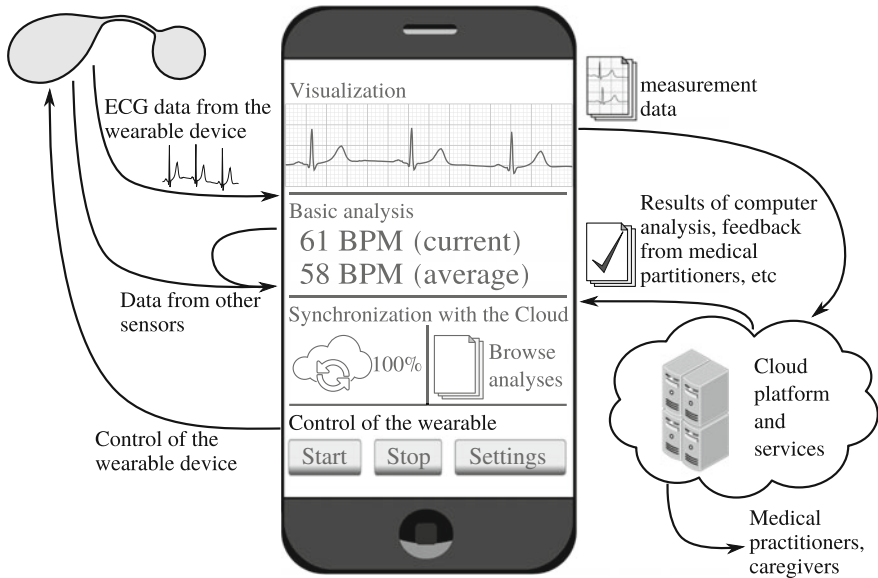


Fig. 3.2 Schematic presentation of the basic smartphone functionality in an mHealth system for ECG monitoring

hold for all mHealth application. Next, we shall focus on ECG monitoring. Figure 3.2 gives a schematic presentation of the basic smartphone functionality in an mHealth system for ECG monitoring.

According to the mHealth requirements, the measured data on the smartphone should be available to the patients themselves as well as to the medical practitioners. A good way of sharing the data among involved parties is having the smartphone process and display the data to the user in real time and in a user-friendly fashion, while also forwarding the unprocessed data to the Cloud via an Internet connection. How the data should be transferred to the Cloud is not well defined and there are not many requirements regarding that functionality. The data could be sent either in near real-time (with minimal delay) or in larger packets, for example once per hour, once per day, etc. Data transmission could be fully automatic or under patient’s control. The transmitted data could contain only manually approved segments of measurement or all the gathered data. The data could also be processed prior to transmission to the Cloud, to lower the data transmission and to limit the transfer only to relevant statistics. It all largely depends on the purpose of the monitoring.

Some reasonable requirements for the mHealth mobile application for ECG monitoring, also given in Fig. 3.2, are:

A familiar user interface. The software on the smartphone can use an interface design that is familiar to the user, and will thus require very little effort from the user to master it.

Optional visualization of the measurements. Although patients are usually not experienced in reading ECGs, it would be wrong to speculate that none of them will be capable or willing to learn the basics. Even through simple observation, the more tech-savvy users are able to quickly distinguish patterns in a single lead ECG and a portion of users would definitely enjoy the option of seeing their ECG in real time. On the other hand, the users who would not find it easy to discern any useful information from the ECG on the screen would find it more reassuring if they did not have to look at it and could replace it with some more meaningful information, for starters, with the heart rate value.

Standard statistics should be shown. Heart rate (HR) is the most basic ECG statistic and is already the norm in readily available heart monitors for recreational use. Users would expect that a powerful measuring device that can record a medical-grade ECG can also have the functionalities that cheaper and simpler heart monitors have. Users would be more likely to find the use of an mHealth device in their everyday lives and would use it more frequently if the HR and some statistics based on it are provided. Consequently, the mHealth system would proliferate better among the general public.

The mHealth system should use the Cloud. Data should be archived as much as possible, in order to assist the medical practitioners in their analysis, to aid research and to enable the extraction of new knowledge from the long-term measurements, Cloud services seem perfect for data storage, since they also facilitate the sharing of data. The archived data will become also an indispensable source of information, once the fully automatic analyses reach maturity. Having the data readily available on the Cloud will shorten the research time and enable faster integration of newly developed algorithms and procedures into operational use.

The ambient data should be analyzed. The ECG measurements have limited value if no additional information on the patient and its activity is available; e.g., the absolute value of the HR is only slightly informative, unless the patient's age and the current physical activity are also known. Furthermore, when alerting is provided, the ambient data should be included in the real-time analysis before any kind of alerting takes place. For example, an alert "Your beat rate has been 150 BPM for 30 min, this could be a dangerous situation!" could be made much smarter if an activity information is also provided: "You have been running for half an hour, your beat rate is steady at 150 BPM, keep up the good work!"

The ECG body sensor should be fully controlled by the smartphone. The general population is already familiar with the smartphone user interface and using it would make the mHealth users more comfortable with the ECG body sensor. At the same time, it would greatly simplify the ECG body sensor itself, as it would relax its requirements for human inputs and outputs, such as buttons, switches, LEDs and displays.

A configurable alerting system should be provided. If life-threatening conditions are detected, the application should alert either the user, a nearby healthcare provider, or lastly the emergency department. The software could also provide an assistance in a situation of medical emergency, to any of the pre-mentioned actors in the mHealth system. In the simplest case, the assistance could be in the

form of instructions on how to handle the patient in the detected situation, while a more sophisticated application could measure the patient's vital signals and display them on the smartphone screen to aid the actors on the scene. Nevertheless, each case would have to be tailored to the specific area of use and the specifics of the mHealth device in question.

There are many other requirements and implementation details, related to the interface design, the ability of tailoring for different technical skills, the local estimation and graphic representation of the measured parameters, etc., that would differentiate the future mHealth applications.

3.4 An Example of an ECG Monitoring System

In this section, a tentative design of an mHealth system for long-term ECG monitoring of patients or healthy and health-proactive individuals is presented. Several options for the implementation of such a system using state-of-the-art technology are considered. We pay attention to the fulfillment of the mHealth requirements and making the system appealing both to its users and to medical practitioners. Finally, we envision that such a system will have broader benefits and will ultimately help to advance the science.

The design of the system takes into account the existing technical standards, allowing easy connection of various ECG body sensors and their immediate replacement if an improved version becomes available. Besides the ECG sensor, the system architecture allows inclusion of additional sensors, on the same device or on additional devices, if such additions prove useful for monitoring the patient's condition in the future. For example, sensors for remote monitoring of respiratory acoustic phenomena (cough, obstruction) or sensors for activity detection would complement the ECG sensor nicely.

The immediate use of the mHealth system can be based on the visual observation of selected critical vital parameters and their recent changes. Through such observation, it is possible to evaluate the effectiveness of a treatment and to foresee a possible deterioration in advance. In the future, after a thorough research of the area, automatic analysis will be able to complement or even replace the visual inspection of the vital parameters. We foresee that alarms will be implemented to alert the medical practitioners on the high possibility of deterioration before the monitored parameter will reach a critical value. Based on the simultaneous evaluation of multiple variables, the automatic analysis will provide the threat level and its trend in the form of the modified early warning score (MEWS) [4]. The analysis of the vital functions in a longer time period will allow for the implementation of cognitive methods; for example, fully personalized analysis of a cardiogram over a longer time period will be used as an input for determining the patient's threat level [5].

As described previously, the device that would act as a crucial link between the mHealth system components should be carried around by the user, should have

computing and communication capabilities, and should be unobtrusive to the user. Today, such a device already exists, namely the smartphone. Smartphones are already owned by more than 2 billion people and their number is still growing. The smartphone connects to the ECG body sensor via the low-power wireless Bluetooth Smart protocol [6] and records everything that the ECG body sensor measures. Besides being extremely low-power, this protocol offers sufficient bandwidth for ECG data streams, concurrent communication with multiple sensors and data encryption.

For security, the measured data stored on the personal smartphone does not contain any information that could facilitate identification of the user. The transfer of the data to the Cloud is encrypted and largely depersonalized—no personal data other than the measurement is ever transmitted to the Cloud. No other personal data is actually required to be stored on the Cloud; aside from the users themselves, only authorized medical practitioners possess additional personal information of the user, such as the health record. The access to the data stored in the Cloud is handled through accounts of varying permissions, which are managed with a safe and reliable account management system. Customized interfaces can be provided for various medical practitioner profiles, to aid them in using, viewing and analyzing the data. These interfaces are managed by the same account management system as the data permissions. Private users also get access to a custom interface for accessing their own data, although they may prefer to use only their personal terminal, without ever connecting to the Cloud.

Another important part of mHealth systems will be open interfaces for custom-made applications and add-ons that communicate either with the Cloud or the smartphone. Such add-ons might end up with discovering a more suitable representation of the measured data for the laymen, or increasing the options for the patients to monitor their vital functions. The options for using the measured data are endless, but interfaces for accessing it will have to be approached with due caution, since sensitive personal data are at stake and should be protected with great care. For example, it may be feasible to identify individuals based on an advanced analysis of the data from the ECG sensor in the near future. When designing an mHealth system, such options can be seen as both drawback and opportunity.

3.5 Smartphone Application Challenges

The ubiquitous smartphones (also tablets, soon-to-follow smart watches, and other smart wearables) have well defined interfaces to which a large portion of the general population is already familiar with. To be acceptable to users, the mHealth mobile application should follow the guidelines set by existing and widely accepted smartphone applications; i.e., the guidelines set by the behavior of various popular software applications, their integration with the smartphone hardware and other software, and their ways of interfacing with the user. Therefore, to integrate better with the user's everyday activities, the following guidelines are recognized for the smartphone appli-

cation used in an mHealth system, based on the feedback from volunteering users of an early mHealth prototype system:

Ease of use. The users' expectations for the applications on their smartphones are increasingly strict. Applications must be installed in a standard and simple manner, set-up in a few clicks (if at all), and then executed as the user would expect. That is, if the user expects the application to start ECG measurement, then the application should do just that, without requiring the user to go through a lot of menus, settings or input boxes.

The ability to run in the background. That is, the ability to perform measurements, analyze them in real time for the requirements of the user interface, and transfer them to the Cloud to be stored, all while the smartphone screen is off.

Low-power requirements. The mobile application must respect the limitations of the battery-powered smartphone, and must not significantly increase the need for the smartphone to be charged.

Integration of the alerting system. The smartphone's operating system implements a built-in messaging, notification and alerting sub-system, which is familiar to the users and allows for the alerts to be presented in a user-defined manner.

Adjustable level of user engagement. The mobile application should target a wide range of people and their willingness to engage their time into the ECG measurement. Besides using the adjustable notifications system familiar to the user, the application should be able to adjust to the user's wishes in other areas too. Examples of adjustable features are: the level of the displayed details and the frequency of the relevant information updates, or the way the synchronization with the Cloud works (always, or only over WiFi, or only when charging, etc.).

A keen observer might notice that the last guideline seems to contradict the first one. It does not need to, though, since applications can be made very adaptable to their users. Therefore, to adhere to both guidelines, the default application behavior is first tuned to the needs of a basic user with a limited desire for engagement with the system. Then, through the provided settings, the application can be configured to show more data and allow more details to be fine-tuned. Thus, the application is made simple for the basic users, while allowing the more advanced users, who are willing to fiddle with the settings, to enable more control for themselves.

Although a proper user interface design would take care of most of the mentioned guidelines, some aspects remain to be tackled with the back-end logic. When designing their own ECG monitoring system, the authors of this book have been presented with four main challenges:

- transmission of the measured data from the ECG body sensor to a smartphone,
- synchronization of time clocks between the ECG body sensor and the smartphone,
- storage of measurements, and
- detection of heartbeats and heart rate calculation.

All of them are discussed in the following subsections.

3.5.1 Transmission of Measured Data to the Smartphone

The most basic functionality of the mHealth-enabled ECG body sensor is the ability to record ECG measurements. The design goals of the ECG body sensor dictate low weight and extremely low power consumption, and thus constrain the ECG body sensor capabilities. One of the several trade-offs that can be made in the ECG body sensor design is the approach for storing the measurements. Measurements can be either stored on-board, as done in [7], or they can be transferred in real time to another device, such as the user's smartphone, as done in [8].

While the use of on-board storage should be (if properly implemented) simpler, more energy efficient, more error tolerant, and completely self-reliant, it is hard to consolidate it with the requirements of an mHealth system. To empower the patient, the measurements should be accessible to the patient in near real-time. In contrast, transferring the measurement to the smartphone in real time, e.g., over a radio connection, enables real-time access to the measurements and thus easier integration into the mHealth system. The price for it, however, are higher power consumption, a more complex overall system, the need to synchronize the ECG body sensor and the smartphone, and the reliance of the ECG body sensor on the external hardware, e.g., the smartphone or some other data gateway.

The radio transfer can also be arbitrarily complex. In the simplest form, it would consist only of data sampling followed by broadcasting the samples through a radio transmitter. A more elaborated form could comprise complex data sampling, data caching on-board, transmission of compressed data packets with error correction mechanisms, acknowledgment of all the received packets by the receiver, and retransmission of lost data packets. A hybrid approach that would reap benefits while avoiding the drawbacks of the on-board storage could also be used. This would increase the system complexity, but any additional complexity in the system design that eases the system use should be viewed as perfectly acceptable.

3.5.2 Synchronization of Clocks

The moment the data leaves the measuring device and enters another, the problem of clock synchronization arises. Here we explain why multiple clocks in the system make an issue, and discuss possible ways of alleviating it.

Time keeping is an old problem [9] that has been largely forgotten by the general population in the last decades. The age of Internet has brought the endless manual clock synchronization to a close end, by providing several very accurate clocks which are accessible via the Internet (or via GPS) and enabling automatic software synchronization of our computers with these precise clocks. The above sentence already discloses the gist of the problem—clock synchronization did not go away, it merely became automated on a large portion of machines that we interface with. The smartphone falls under the category of Internet-connected devices. For the time being, we

shall, therefore, assume that it does not require any additional clock synchronization. For devices that are not connected to the Internet or other large interconnection networks with centralized time keeping, the clock synchronization remains an issue that needs addressing. The ECG body sensor certainly represents an example of such a device.

The electronic circuits most often use inexpensive quartz crystal oscillators to keep time to a limited precision. The modern quartz oscillators are accurate to under 100 ppm [10], which translates into time keeping devices (clocks) with similar accuracy. A clock that is stable to 100 ppm is also said to be accurate to $24 * 60 * 60 / 10^{-4} = 8.64$ s per day. Under relatively steady temperature conditions, these oscillators can perform at least an order of magnitude better, that is, they drift for less than 1 s per day. One second is in the same order of magnitude as the length of a single heartbeat, so quartz-based clocks could be considered precise enough for performing short measurements without external clock synchronization. Even for measurements of several days, such error could be considered acceptable if no synchronization with other sensors and devices is required. However, as we describe in the Chap. 2, several concurrent ECG measurements can be merged into even more descriptive forms of ECG, which requires synchronization precise to a hundredth of a second.

ECG body sensors are also miniaturized for a lower weight, and as such usually do not include a real-time clock circuitry with a dedicated battery for time keeping when the device is turned off. They should be reminded of the current time on every start-up or reset. Therefore, two arguments are in favor of synchronizing the ECG body sensor's clock with the smartphone: the need to perform long measurements (on the timescale of several days) and the need to adjust the clock after start-up or reset. Since the ECG body sensor and the smartphone constitute a simple distributed system, the solution is to use one of the many clock synchronization techniques for distributed systems [11]. In such a distributed system, the smartphone has a more accurate time keeping. Therefore, one solution would be to consider its time as a reference, and have the ECG body sensor synchronized to it at all times. This solution, however, may still cause measurements performed on different smartphones to drift in time, and is only appropriate if such drift is acceptable.

3.5.3 Storage of Measurements

One of the essential mHealth functionalities is the storage of measurements, which can be regarded as a two-level process. The first level is implemented on the smartphone, where the measurements are stored until they are uploaded to the Cloud. The second level is on the Cloud, where the measurements are stored as a part of the patients' medical record, knowledge databases, analytical Cloud services and similar. We shall not discuss much about the Cloud part of storage here, since it largely depends on the goals of the particular mHealth system. We shall rather focus on the storage of the measurements while they are stored on the user's smartphone, which makes them easily accessible to the user.

There is a plethora of file formats for storing ECG to choose from, and one should choose very wisely. The file storage should be implemented in a way that adheres to the existing ISO standards of the 22077 family [12–15], related to the storage of ECG measurements. In addition to standards, there are also some mHealth-specific requirements for the storage of measurements, which are in line with the general idea of mHealth—to help make patients more proactive and improve the integration between medical services:

- *The use of files for storing individual measurements and the use of standard file formats whenever possible.*
- *Empowerment of patients*, which can be accomplished by giving users the ability to browse, copy, export and view the measurements with standard tools available. Open file formats present a more rational choice compared to proprietary formats, since the latter would severely limit the users in their access to the measurements, and could cause vendor lock-ins and additional expenses for the developers of mHealth system extensions. A recent review of open file formats has been done in [16], and while a lot of them have been designed with very different ECG measurements in mind, there are several that fulfill quite a lot of the listed requirements.
- *The ability to store various meta data in the same file.* Since the mHealth is in its infancy, it is unreasonable to expect that the meta data structure adopted at the design stage will pass the test of time and the increasing usage. Therefore, the meta data should be at most weakly structured, to allow for future additions or modifications.
- *The ability to store measurement data other than ECG in the same file.* As explained before, the additional measurements will in time be able to provide indispensable information complementary to the ECG, allowing for a far more in-depth analysis and a more precise diagnosis. Like in the case of meta data, trying to identify all data types in advance would not be very prudent. Therefore, a flexible file format supporting additional measurements of varying sample sizes and sampling frequencies would be most welcome.
- *The use of compact file formats* to allow the storage of large data quantities on a standard-size smartphone storage.
- *Robustness of the storage procedures against abrupt and unanticipated interruptions.* The inclusion of personal smartphones into the mHealth hardware scheme is followed by a specific set of problems, which are not inherent to medical devices. The smartphones and the software running on them are not designed for the same degree of reliability as are certified medical devices. Furthermore, they are under full control of their users, who may want to abruptly pause, stop or interfere with the ongoing measurement. While performing measurements, users may also use other applications with a wide range of system resource requirements, which may inadvertently affect these measurements. Therefore, the interruptions in measurements, including those that occur while the mHealth application is in the middle of a file-storing procedure, should be considered normal operation. Either the chosen file formats should be robust enough to allow not-fully written files to remain readable up to the point of corruption, or some other form of file repair

should be available. The latter is less optimal, since it requires a more complex file management system and adds new points of possible system failure.

However, an existing file format that would fully satisfy the listed requirements is impossible to find. Since the novel mHealth systems are coming with their own special requirements, modifying the existing file formats or even designing new ones is a viable option. Here one should not forget the lesson learned from the existence of a plethora of file formats for storing ECG: the design should not be too specific and should allow for future changes and improvements.

3.5.4 Detection of Heartbeats

After a measurement is recorded and the time of individual samples well determined, the processing of the ECG can start. The most basic processing of an ECG signal, intended to derive some information of user interest, is the detection of heartbeats. More precisely, it is the detection of the times when heartbeats occur. The application can then use those times to calculate some basic heartbeat statistics, such as the heart rate [17], or perform more complex processing, such as automatic classification of beats [18]. The heartbeat information is even more useful if it can be accessed in real time. The mobile application can use such data to advise or alert the user when certain conditions are detected. In the cases when the detected conditions appear to be alarming, the mobile application could be used to automatically alert the patient's caregivers or even the nearby emergency department. From an empowered patient's perspective, the heartbeat detection can be used by the patient directly for an easier observation of the heart rhythm and the shape of characteristic ECG waves. For example, proactive patients might be curious about how their heart reacts to changes in their activity, their nutrition or emotional state. They might even learn to better recognize their current condition after they inspect the current heartbeat rhythm and shape.

Although the beat detection is a mature field in science, with plenty of known algorithms to choose from [19], one should note that these algorithms were designed for a different kind of ECG measurements. They were mostly designed to be used on 12-lead ECG measurements made in a controlled environment. As discussed in Chap. 2, the ECG body sensor delivers differential ECGs that differ from the standard conventional ECGs in several details. Furthermore, the ECG body sensor records far more noise than the standard 12-lead ECG apparatus. The additional noise is produced because of the limited number of electrodes, the limited distance between the electrodes, and most importantly, the difficult measurement conditions under which the mHealth system is used. To reach full integration into patients' lives, mHealth devices are being used during everyday activities, in totally uncontrolled environments. When patients are engaged in physical activities and exercise, the amount of noise caused by the muscle activity and the physical strain on the electrode contacts is very high. Moreover, the conditions vary during the measurement. The

users are very likely to be partly at rest and partly active during the same measurement. They are also moving in environments with variable electromagnetic interference, ranging from practically zero (e.g. in nature, far from any human-made infrastructure) up to the limits posed by the health standards. All in all, the difficult conditions should be countered by making the algorithms more robust, lowering expectations on the result accuracy, and additional statistics returned by the algorithms, e.g., noise levels.

Several modifications to the known beat detection algorithms should be implemented to make them appropriate for body sensor ECGs:

- Aggressive input filtering aimed at removing all the noise in the measurement. The input filtering can be performed as digital processing for the task of beat detection only.
- Build-in adaptivity to the input signal orientation and to gradual changes in the ECG shape because of baseline wandering, i.e., on a timescale of more than one beat. For example, user's posture will influence the ECG shape, while the varying water content in the skin will influence its electrical conductance and consequently change the amplitude of the ECG. Users might also modify the position of the ECG body sensor during the measurement, and thus change the measured ECG shape completely.
- A measure of signal noise, which can be first used as a help in beat detection, and second, as an additional output information from the beat detector.
- If the ECG body sensor provides also other measurements, e.g. activity or EMG, they can be used as additional information for beat detection.

An important question to think about is also where the beat detection takes place. It should take place close to the user to make the results available for further processing in near real time. It is worth noting though that several seconds of measurement may need to pass before beats can be detected with sufficient probability.

Beat detection could theoretically be done on the Cloud if the measurement is being transferred to the Cloud in real-time. However, this option comes with a severe drawback. When the Cloud is not available, the beat detection will not occur. Therefore, the reliance on the Cloud, which is equivalent to the reliance on a sufficiently fast Internet connection, is unacceptable for the general mHealth application. It might be acceptable for some special cases though, e.g., when monitoring is implemented only within a single health institution.

A better alternative is to perform beat detection on the smartphone or on the ECG body sensor. The smartphone is a device with increasingly large amount of computational power at its disposal, with near real-time access to all the measurement data required by the beat detection algorithm, and thus seems very fitting. The disadvantage of implementing the beat detection on the ECG body sensor, as opposed to the smartphone, are the limited computational capabilities of the ECG body sensor. Furthermore, running a computationally demanding algorithm on the ECG body sensor could seriously increase the power consumption and thus decrease the device autonomy and limit its use. The benefit of detecting the beats on the body sensor, on the other hand, is in the possibility to lower the amount of the data to be sent to the smartphone. Adaptive data compression algorithms may be used before the data is

transmitted if beat locations are known; the ECG can be compressed with varying compression ratios for different parts of the ECG signal. The noise level may also be assessed better if beat locations are given, which again enables more control over adaptive compression algorithms. Compressing the data to be sent is very welcome and can be viewed from two different angles. First, it can be seen as a means of lowering the power consumption of the ECG body sensor, since transmitting compressed data usually requires less energy. Second, it can be viewed as an opportunity for the ECG body sensor to gather and transmit more information to the smartphone, since more of the available bandwidth remains accessible. Therefore, selecting the device where the beat detection will be implemented requires deciding on a trade-off between benefits and disadvantages, and may also include hybrid solutions. Nevertheless, getting the most out of the system will require at least some rudimentary beat detection to be implemented on the ECG body sensor itself.

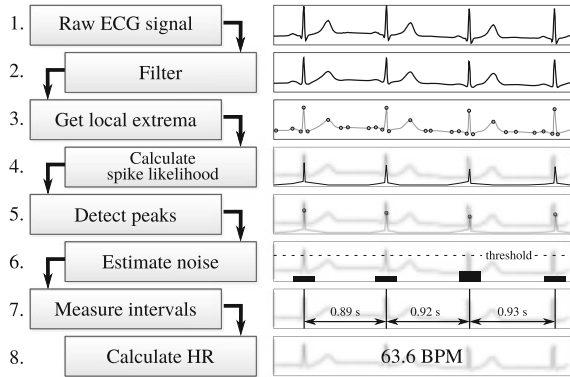
Furthermore, the beat detection can be partly done on the smartphone or body sensor and partly extended to the Cloud, where all the patient's measurement history and healthcare information is stored. Such a hybrid solution would make the most precise beat detection available to medical practitioners through the Cloud services, while also making the preliminary beat detection available to the user of the mHealth system. Extending the hybrid solution from beat detection to complete ECG processing would be a natural next step. At this point, one could even make a case for distributed computing, where the processing workload of the Cloud service for ECG processing is lowered on the account of the users' smartphones performing the bulk of the processing in advance. This, however, is a topic for further research and experimentation.

3.6 Example Algorithms

In this section, the complex task of heart rate calculation is examined and an algorithm for heart rate calculation is proposed. A schematic representation of the proposed HR calculation algorithm is shown in Fig. 3.3. Starting with a raw ECG input (step 1), the first part is the beat detection composed of filtering, extraction of extrema, estimation of spike likelihood and peak detection (steps 2 to 5). Then follows the noise estimation—the noise in the vicinity of detected beats is estimated and beats from noisy parts of the signal are discarded from further processing (step 6). Lastly, beat-to-beat intervals are measured (step 7), outliers removed, and the remaining intervals averaged to derive HR (step 8).

The first task—beat detection—is based on an algorithm [17] designed for use in an mHealth system. The proposed beat detection algorithm is in many aspects similar to previously known algorithms [19], but designed to work on differential ECGs with a low sample rate and a low signal-to-noise ratio. Furthermore, it is not affected by the orientation of the ECG features, since it searches only for rapid changes in the input signal, ignoring their exact nature. It was designed to have a small memory footprint

Fig. 3.3 A schematic representation of the HR calculation algorithm



and to use as little processing power as possible, so that it could be implemented on the ECG body sensor.

3.6.1 Filtering

The beat detection algorithm is tuned to detect only the QRS complex, because it is the most prominent feature of almost any ECG and is far less sensitive to morphological changes than the other ECG waves [20]. The frequency spectrum of the QRS is quite limited. A narrow frequency band filter that would pass only the QRS frequencies, e.g. 10–40 Hz [21], could be used to remove all noise in the frequency bands which carry no ECG-related information. Filtering is performed using a digital filter only in the beat detection phase. The filtered signal is later discarded. Therefore, the filter used in this step does not deform the measurement and is used only to help the beat detection task. For simpler and more efficient implementation, a low-pass filter is used with the threshold set at 45 Hz. The filter is implemented as Brown's exponential smoothing [22], also called exponential moving average (EMA).

Brown's exponential smoothing calculates the smoothed value $s(t_n)$ of a sampled signal $x(t_n)$ for each time step t_n as:

$$\begin{aligned}
 s(t_0) &= x(t_0) \\
 \Delta t &= t_{n-1} - t_n \\
 \alpha &= 1 - e^{-\frac{\Delta t}{\tau}} \\
 s(t_n) &= \alpha x(t_n) + (1 - \alpha)s(t_{n-1}), \tag{3.1}
 \end{aligned}$$

where α is a smoothing factor, derived from the time constant τ that specifies the time in which the smoothed response to a unit step input would reach the value $1 - 1/e$. Δt is the time step, which does not need to be constant throughout the measurement.

3.6.2 *Extrema Extraction*

The local extrema of the filtered ECG signal are extracted to lower the amount of the data that has to be processed in the following steps and thus simplify further processing. Local extrema, i.e. positive and negative peaks, are the positions in which the derivative changes its sign. They appear in an alternate fashion, with each minimum followed by a maximum, and vice-versa.

From the interval of a monotonic input signal between each pair of consecutive extrema, a single extremum in the derivative is also extracted. Based on the derivative sign, the extracted extremum is either a minimum (negative sign) or a maximum (positive sign). For example, from the interval between a minimum peak and a maximum peak in the input signal, the input signal is non-decreasing, the derivative is non-negative and, therefore, a maximum in the derivative is extracted.

3.6.3 *Spike Likelihood Estimation*

The term spike will be used here to refer to a rapid change in the signal amplitude—either up or down, followed by a rapid change in the other direction. For more distinct changes in amplitude, the spike will be termed as strong. The QRS complex contains at least one strong spike, regardless of where the ECG is measured and thus independent of the ECG body sensor placement and orientation. Therefore the beat detection can be implemented by spike detection.

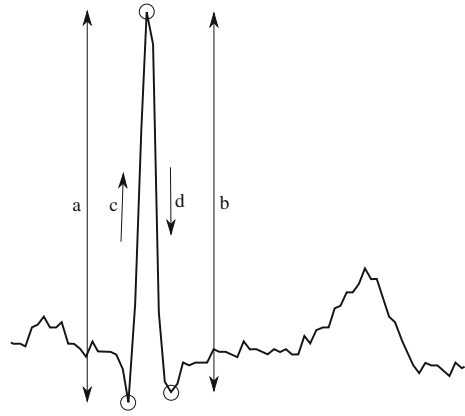
A spike contains a local extremum as its peak. In this algorithm step, each local extremum is tested to determine if it represents the peak of a strong spike. Spike likelihood is introduced as a measure of spike strength in each local extremum. The estimation takes three signal extrema and two signal derivative extrema as input, and converts them into a unit-less number. For each signal extremum, the likelihood is calculated as:

$$a \cdot b \cdot c \cdot d,$$

where a is the difference in amplitudes of the given and the previous extremum, b is the difference in amplitudes of the given and the next extremum, and c and d are the derivative extrema to the left and to the right of the given extremum. A graphical explanation of the likelihood estimation step is given in Fig. 3.4.

A part of an ECG input signal with a strong spike is shown in Fig. 3.4 with a solid line. For every triplet of extrema in the input signal, extracted in the previous algorithm step and marked with small circles, two pairs of parameters are calculated. The first two parameters, a and b , marked with double-ended arrows, are obtained as amplitude differences between two consecutive input signal extrema as follows: a is the difference in amplitudes between the second and the first extremum, b is the difference in amplitudes between the third and the second extremum. The other two parameters, c and d , marked with single-ended arrows, are obtained as extrema of

Fig. 3.4 Likelihood estimation scheme on an example of a strong spike



corresponding positive and negative signal derivatives: c is the maximum derivative of the input signal between the first and the second extremum of the input signal, d is the minimum derivative of the input signal between the second and the third extremum of the input signal.

3.6.4 Peak Detection

High likelihood values are assumed to be caused by strong spikes in the input signal, which in turn are a sign of significant probability for a QRS complex in the signal. Therefore, the spike likelihood could be taken by its absolute value; an absolute threshold, with values high enough, could be applied as a filter. Nevertheless, to achieve some adaptability, spike likelihood is rather observed relative to its previous peak values.

First, modified exponential smoothing is applied on the likelihood and the result is multiplied with a constant factor f . The result is called dynamic threshold $d(t_n)$. The best values for f are in the interval from 1 to 10, which was confirmed by a preliminary analysis on different ECG input signals. Second, as the input signal is processed in real time, the local peaks of likelihood are compared with the dynamic threshold value. When a likelihood value higher than the dynamic threshold is encountered, a peak is detected and the following two procedures take place: the threshold value is set to the value of the likelihood in the new peak and the detected heartbeat is the output from the algorithm. Formally, the peak detection algorithm can be written as follows:

$$\begin{aligned}
 d(t_n) &= f \cdot s(t_{n-1}) \\
 s(t_n) &= \begin{cases} \text{as calculated in Eq. 3.1,} & \text{if } (x(t_n) \leq d(t_n)) \\ x(t_n), & \text{otherwise.} \end{cases} \quad (3.2)
 \end{aligned}$$

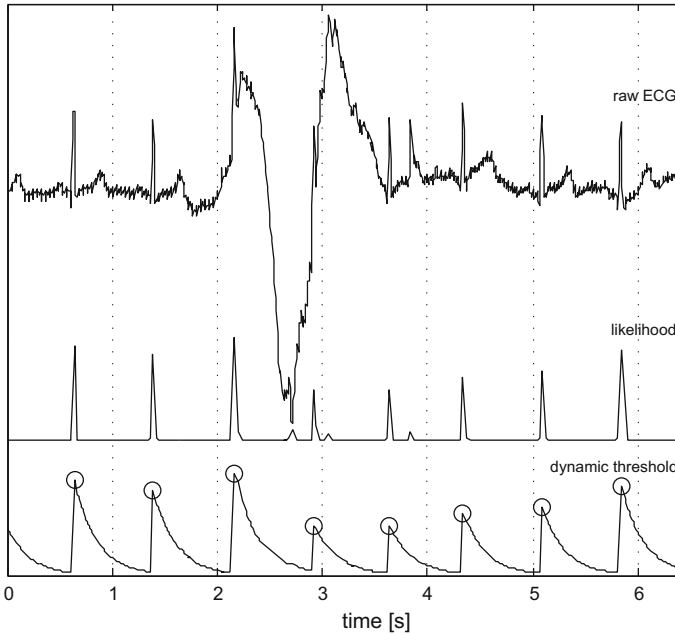


Fig. 3.5 ECG signal from a significantly distorted body surface measurement (*top*), spike likelihood estimated for the ECG signal (*middle*), dynamic threshold (*bottom*). The circles on the bottom plot mark the detected beats. The Y axis is not marked, since its absolute values are not relevant

The described peak detection algorithm step is fully adaptable to the input signal and detects only the highest peaks on a time scale that is controlled by the parameter τ . An output of this step is also the result of the complete beat detection algorithm, which is represented by the timing of QRS complexes in the input ECG signal.

Figure 3.5 shows an example of the likelihood estimation and dynamic thresholding on a segment of a significantly distorted ECG signal. In the figure, it can be seen that the peaks in spike likelihood correspond to the locations of QRS complexes in the ECG signal. The QRS complex of the fourth beat is properly detected, even though it is significantly distorted by noise. The performance of the described beat detection algorithm is further confirmed by correct exclusion of the spike-shaped noise between the fifth and the sixth beat of the ECG measurement.

3.6.5 Noise Estimator

In this section, we describe a crude noise estimator. The estimation is done by counting the number of large first derivative extrema in the area close to a detected beat. These are likely to be caused by noise and could disturb the beat detector. This esti-

mator is thus able to estimate noise in the area around detected beats, in order to discard the beats detected with low reliability on the noisy signal. The estimator has two parameters: the time interval where derivatives are observed and the threshold amplitude for derivative extrema.

The first parameter, i.e., the time interval in which derivative extrema are counted, could be set dynamically for each beat—based on the heart rate, or statically—using a fixed size time window. For example, if 200 BPM is considered the maximum heart rate that the system should detect, then the static time window can be set to the time of the detected QRS \pm 300 ms, since this time window is guaranteed not to overlap with another QRS complex. For the dynamic setting, the time window not overlapping with another QRS complex is set to the time of the detected QRS \pm 500 ms * (60/current BPM).

The second parameter, i.e., the threshold amplitude for derivative extrema (β), is used to classify extrema as either large or small, where only large are likely to cause problems in beat detection. To make this threshold adaptable, β is normalized by the largest first derivative amplitude within the time interval of the located QRS. Thus the β should be selected as a relative value from the interval [0...1]. Based on the analysis of several experimental ECG measurements, we have determined that 0.5 is usually a good value for β . Nevertheless, the choice of β should depend on the amount of the expected noise in the signal.

Some fine tuning of both parameters should be performed before implementing the algorithm on a specific ECG body sensor and for a specific range of expected ECG input signals.

3.6.6 Heart Rate Calculation

The frequency of heartbeats—the heart rate (HR)—is calculated on a short time interval and reported as the number of beats per minute (BPM). Although it sounds straightforward, robust calculation of heart rate is difficult when the measurement is noisy. The presented real-time algorithm for HR calculation works by averaging the intervals between successive beats. It is thus able to return a new heart rate value for each newly detected beat.

A list of beats is maintained while the measurement is in progress, with the following procedures applied to each detected beat:

1. The output of the noise estimator is taken to discard beats that were detected while noise levels were above the threshold. If the estimated noise is above 5, i.e., five spikes of comparative amplitude were identified near the QRS, then this beat time is discarded due to high noise and the procedure ends here. Otherwise, the beat time is added to the list of beats.
2. The list of beats is trimmed to contain only the beats from the last S seconds of the measurement, by checking the age of each beat in the list and removing those of an age more than S . The time window S is a parameter that should be adjusted

to suit the system requirements. Since the unit of HR is in beats per minute, an appropriate value for S is 60 (1 min). Lower values will make the calculation more sensitive to changes in the HR, but also more susceptible to noise.

To enable reliable results when noisy data is used, only the most reliable beat-to-beat intervals are used for averaging. For example, if a beat is not detected due to noise (false negative), this creates a longer-than-average beat-to-beat interval. Similarly, if the noise in the signal is falsely recognized as a beat (false positive), this creates two shorter-than-average beat-to-beat intervals. Such intervals should be detected and handled properly to eliminate their influence on the mean HR. The full procedure for processing the list of beats is as follows:

1. The differences between all consecutive beats in the list are calculated, to obtain a list of beat-to-beat intervals.
2. The mean beat-to-beat interval is calculated from the beat-to-beat interval list.
3. Those beat-to-beat interval outliers which deviate from the mean by more than D are removed from the list. The value of D should be 50% or higher, otherwise false positives might not be detected.
4. If less than N beat-to-beat intervals remain, the procedure is exited with an error indication. This check enforces that at least N intervals are averaged for a more accurate and more stable estimate. Higher values of N will result in a smoother evolution of HR through time and should be preferred. Formally, N is down-limited by 1 (at least one beat-to-beat interval is required to calculate the heart rate) and up-limited by the minimal number of beats that can be detected in S seconds. The upper limit is thus a function of the minimal heart rate and largely depends on the intended use of the system.
5. The value of HR is calculated as $(60/\text{mean beat-to-beat interval})$.
6. The resulting HR value is then checked against the physical bounds for the human HR. Very broad limits for HR, e.g. [20–250], can be selected, so that this last check filters only the extreme periodic noise, such as the electric mains noise that can appear on disconnected electrodes of the ECG body sensor.

If no error occurs, the calculated HR can be shown to the user (preferably rounded to the nearest integer) or used for further analysis. Otherwise, an indication that the HR cannot be reliably detected should be shown to the user.

The HR display should furthermore reflect the cases when no heartbeats have been detected in some time (e.g., in 10 s). This should not be a cause for alarm, since these cases are not necessarily caused by a cardiac arrest, but far more likely by disconnected electrodes. At this point, the cause for absent beats can be further analyzed, if measurements complementary to the ECG are also available from the sensor. An indicator often used to discern whether the electrodes are attached or not is the measurement of the electrical resistance between the electrodes; infinite resistance implies that the electrodes are not attached, while near-zero resistance implies a short circuit between the electrodes caused, for example, by submerging the sensor in water. Both cases indicate an improper ECG body sensor usage.

Finally, the calculated HR could be written in a file, to complement the ECG measurement. It could, however, be used only for providing an immediate information

to the user; after all, a more advanced beat detection and heart rate calculation can be performed off-line, when necessary. The choice whether to store HR information or not is in the designer's hands, because it represents a trade-off between additional storage requirements and the ability to quickly visualize and search through the HR values of past measurements.

References

1. Rashkovska, A., Tomašić, I., Trobec, R.: A telemedicine application: ECG data from wireless body sensors on a smartphone. In: Proceedings of MEET & GVS on the 34th International Convention MIPRO 2011, vol. 1, 2011, pp. 293–296
2. European Commission, Green Paper on mobile health (“mHealth”) (2014). <https://ec.europa.eu/digital-single-market/en/news/green-paper-mobile-health-mhealth>
3. Levine, D.M., Lipsitz, S.R., Linder, J.A.: Trends in seniors’ use of digital health technology in the United States, 2011–2014. *JAMA* **316**(5), 538–540 (2016)
4. Subbe, C., Kruger, M., Rutherford, P., Gemmel, L.: Validation of a modified early warning score in medical admissions. *Qjm* **94**(10), 521–526 (2001)
5. R. Miller, Rise of the machines: Computers construct better biomarkers (2011). <http://www.medscape.com/viewarticle/751079>
6. Decuir, J.: Introducing bluetooth smart: Part II: applications and updates. *IEEE Consumer Electron. Mag.* **3**(2), 25–29 (2014)
7. Saadi, D., Sørensen, H., Hansen, I., Egstrup, K., Jennum, P., Hoppe, K.: ePatch@—A Clinical Overview, Technical report, Technical University of Denmark, Denmark, annual report (2014)
8. Trobec, R., Avbelj, V., Rashkovska, A.: Multi-functionality of wireless body sensors. *The IPSI BgD Trans. Internet Res.* **10**, 23–27 (2014)
9. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. *Commun. ACM* **21**(7), 558–565 (1978)
10. Lam, C.S.: A review of the recent development of MEMS and crystal oscillators and their impacts on the frequency control products industry. In: Proceedings of the IEEE International Ultrasonics Symposium, 2008, pp. 694–704
11. Horauer, M.: Clock synchronization in distributed systems, Südwestdeutscher Verlag für Hochschulschriften SVH (2009)
12. Health informatics—medical waveform format—part 1: Encoding rules, Standard ISO/TS 22077-1:2015. International Organization for Standardization, Geneva, Switzerland (2015)
13. Health informatics—medical waveform format—part 3: Electrocardiography, Standard ISO/TS 22077-2:2015, International Organization for Standardization, Geneva, Switzerland (2015)
14. Health informatics—medical waveform format—part 3: Long term electrocardiography, Standard ISO/TS 22077-3:2015, International Organization for Standardization, Geneva, Switzerland (2015)
15. Health informatics—medical waveform format—part 4: Stress test electrocardiography, Standard ISO/TS 22077-4, International Organization for Standardization, Geneva, Switzerland (2015)
16. Trigo, J.D., Alesanco, Á., Martínez, I., García, J.: A review on digital ECG formats and the relationships between them. *IEEE Trans. Inf. Technol. Biomed.* **16**(3), 432–444 (2012)
17. Lavrič, P., Depolli, M.: Robust beat detection on noisy differential ECG. In: 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016, Opatija, Croatia, May 30–June 3, 2016, 2016, pp. 381–386

18. Rashkovska, A., Kocev, D., Trobec, R.: Clustering of heartbeats from ECG recordings obtained with wireless body sensors. In: 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016, Opatija, Croatia, May 30–June 3, 2016, 2016, pp. 461–466
19. Kohler, B.-U., Hennig, C., Orglmeister, R.: The principles of software QRS detection. *IEEE Eng. Med. Biol. Mag.* **21**(1), 42–57 (2002)
20. Avbelj, V., Trobec, R., Gersak, B.: Beat-to-beat repolarisation variability in body surface electrocardiograms. *Med. Biol. Eng. Comput.* **41**(5), 556–560 (2003)
21. Buendía-Fuentes, F., Arnau-Vives, M., Arnau-Vives, A., Jiménez-Jiménez, Y., Rueda-Soriano, J., Zorio-Grima, E., Osa-Sáez, A., Martínez-Dolz, L., Almenar-Bonet, L., Palencia-Pérez, M.: High-bandpass filters in electrocardiography: source of error in the interpretation of the ST segment. *ISRN Cardiol.* (2012)
22. Brown, R.G.: Exponential smoothing for predicting demand. In: *Operations Research*, vol. 5, pp. 145–145