Using Vital Sensors in Mobile Healthcare Business Applications
Challenges, Examples, Lessons Learned

Johannes Schobel, Marc Schickler, Rüdiger Pryss, Hans Nienhaus, and Manfred Reichert
Institute of Databases and Information Systems, University of Ulm, James-Franck-Ring, Ulm, Germany
{*johannes.schobel, marc.schickler, ruediger.pryss, hans.nienhaus, manfred.reichert*}*@uni-ulm.de*

Keywords: Sensor Framework, Android, Smart Mobile Device, Healthcare, Vital Signs, Mobile Business Application.

Abstract: Today, sensors are increasingly used for data collection. In the medical domain, for example, vital signs (e.g., pulse or oxygen saturation) of patients can be measured with sensors and used for further processing. In this paper, different types of applications will be discussed whether sensors might be used in the context of these applications and their suitability for applying external sensors to them. Furthermore, a system architecture for adding sensor technology to respective applications is presented. For this purpose, a real-world business application scenario in the field of well-being and fitness is presented. In particular, we integrated two different sensors in our fitness application. We report on the lessons learned from the implementation and use of this application, e.g., in respect to connection and data structure. They mainly deal with problems relating to the connection and communication between the smart mobile device and the external sensors, as well as the selection of the appropriate type of application. Finally, a robust sensor framework, arising from this fitness application is presented. This framework provides basic features for connecting sensors. Particularly, in the medical domain, it is crucial to provide an easy to use toolset to relieve medical staff.

1 INTRODUCTION

Today, mobile business applications, running on both smartphones and tablet computers, become increasingly important for the medical domain. This ranges from schedule management applications to complex process-oriented, mobile patient management systems (Pryss et al., 2010; Pryss et al., 2012). Since these applications have evolved from administrative (e.g., bed planning) to operational (e.g., measuring blood pressure) tasks, it is necessary to support medical staff in their daily routine. This can be supported through the integration of external sensors, which are not built in the smart mobile device, for monitoring vital signs.

Due to heterogeneity of available external sensors, it becomes necessary to classify these devices. This can be done, for example, based on the type of vital signs a sensor can monitor (e.g., pulse or oxygen saturation), or the type of connection (e.g., Bluetooth or USB) the sensor supports. In addition to the diversity of mobile operating systems (e.g., Android or iOS), various development aspects emerge that must be addressed. This ranges from different types of provided sensor data to the quality of gathered data (e.g., real-time or bulk-load data). Furthermore, coping with connection problems (e.g., disconnections between the participating devices) during mobile data collection is challenging.

To evaluate these specific aspects, we present our implemented mobile fitness application "XFitXtreme" that communicates with external sensors. Based on insights of this application, we want to create a sensor framework, which allows for the provision of a variety of different vital signs to athletes or patients using mobile applications with sensors.

The remainder of this paper is structured as follows: Section 2 discusses our application scenario. In Section 3, basic concepts for developing a mobile fitness application are introduced, and the designed system architecture is presented. Section 4 discusses the current status of our fitness application "XFitXtreme" and presents particular issues emerging during the implementation. Section 5 discusses related work, while Section 6 gives a summary and a brief outlook for further research questions.

2 APPLICATION SCENARIOS

To evaluate our approach of integrating sensors in mobile business applications, we present a proper real-

world scenario (cf. Section 2.1). For this purpose we consider the fitness domain, in which we want to use external sensors to support athletes in daily workout sessions. In particular, we want to show different ways of developing such mobile business applications integrating external sensors. These considerations are the basis for the system architecture and sensor framework in the further course of this paper.

2.1 Running Example

To evaluate respective issues presented in Section 1, we implemented a realistic mobile business application. It provides the basis how to use sensors for monitoring vital signs within a mobile application.

We decided to develop a fitness application. Besides standard requirements, such as robustness and applicability, we focus on flexible real-time data collection from external sensors. To give athletes more options for collecting information about their vital signs, we wanted to support a variety of vital sensors. Specifically, in this application, we used two different sensors measuring two vital signs (heart rate, and oxygen saturation). Furthermore, both the processing and the visualization of gathered data shall be adequately supported by the application.

The "XFitXtreme" application is used to support athletes on doing CrossFit. The goal of this type of fitness workout is to practice on many different methods, such as strength, speed, agility, or stamina. Thus, a maximum fitness of the entire body shall be achieved. A standard training session of CrossFit usually lasts 60 minutes. Consequently, recording data of such a training results in a considerable amount of data. In addition, athletes should be in constant control of their vital signs, to prevent any risk of injury. These requirements of the fitness domain serve as a proper application scenario for the first usage of external sensors within a mobile application.

2.2 Other Application Scenarios

Regarding sensor integration, we basically focus on mobile healthcare applications. We present three scenarios from the medical domain, in which we revealed by interviewing physicians that they could be dramatically relieved in their daily routines. The healthcare domain therefore has revealed different scenarios, in which patients could benefit from the use of these devices.

- **Medical Ward Rounds in Hospitals:** In their daily medical round, physicians have to deal with unscheduled patient examinations (Pryss et al., 2012), e.g., measure the blood pressure, or blood sugar level. To collect this data more accurately, mobile healthcare applications integrating external sensors could be used. With the help of such mobile applications, vital signs can be collected and automatically stored in the electronic patient record. In addition, certain procedures could be automatically started, depending on the current patient's vital signs. Consider for example, a doctor determines an increased blood sugar level using his smart mobile device providing a blood sugar sensor. The information will then be documented and archived in the patient's electronic record. The clinical decision support system (Trowbridge and Weingarten, 2001) then could suggest the procedure "schedule medication for lowering blood sugar level".
- **Rescue Service:** Particularly in rescue services it is important to ensure fast, reliable, and real-time measurement of a patient's vital signs. In this case, patients could benefit by using sensors that provide information about their health condition. For example, rescue staff checks the oxygen level of a patient at accident site. With the help of a smart mobile device and external sensors, it can be easily checked, whether the patient is hyperventilating. If oxygen level is above 98%, countermeasures must be performed quickly to stabilize the patient.

Aside from the mentioned scenarios, the domain of clinical psychology provides other interesting possibilities using sensor data in mobile applications.

- **Psychological Questionnaires:** In other research projects, we have already gathered experience regarding the development of mobile business applications for clinical psychology. This includes for example the development of digital questionnaires for collecting patient data. In this context demands emerged to integrate external sensors. For example, the patient's vital signs could be monitored during data collection. Thus, an exceptional patient behaviour will be indicated by his vital signs. Consider the following example: A question "Do you take drugs?" is provided to a patient. If the patient answers with "No", and the interviewer observes that the patient's pulse rises while giving the answer to this question, then she has more indications to evaluate the quality of the respective answer. Therefore, pulse data associated with this question is stored electronically and can be used when evaluating the questionnaire.

The usage of external sensors, coupled with a mobile business application could improve the quality of medical healthcare procedures as presented above.

3 BASIC CONCEPTS & ARCHITECTURE

This Section will cover basic concepts, which have to be carefully considered in the course of the development of mobile business applications aiming at the integration of external sensors. In Section 3.1, we basically focus on the appropriate development type of mobile applications (Web Application, Hybrid Web, or Mixed Application, or Native Application). We consider this discussion as a fundamental aspect for mobile application development. Section 3.2 describes our implemented system architecture, whereas Section 4 discusses important implementation details.

3.1 Different Ways of Developing Mobile Business Applications

Before beginning the development of a mobile business application, one has to consider the basic development type for the application. It determines the scope of provided features available in the application. In addition to this, based on the chosen type of application development, aspects like programming language (e.g., Java or HTML5 and JavaScript) or architectural design patterns (e.g., Model-View-Controller) will be basically determined (Heitkötter et al., 2013). In the following, we define four different development types for mobile applications which have to be distinguished.

- **Web Applications:** This type of mobile business application, e.g., jQuery Mobile Applications, is implemented using HTML5, CSS, and JavaScript. Usually, the application code is rendered within a mobile web browser. According to this, these applications are independent of a platform and device. Compared to native applications, this is a lightweight approach. In contrast to their minimal implementation effort, such applications provide the smallest set of functionality, because they rely on the feature set of a web browser.
- **Hybrid Applications - Web:** The application code of this type of mobile business applications, e.g., Adobe PhoneGap Applications (Adobe Systems Inc., 2013), consists of the same web technologies as for real Web Applications. However, they run in a native application container (e.g., iOS WebView), which, in turn, can provide additional functions.
- **Hybrid Applications - Mixed:** These type of Hybrid Applications, e.g., Appcelerator Titanium Applications (Appcelerator Inc., 2013), provide a core of mobile development APIs which will be normalized across platforms. These APIs can be used like known web technologies (cf. Hybrid Applications - Web). However, during the deployment of the application a mapping between elements of HTML5 and elements of the corresponding native platform must be automatically performed.
- **Native Applications:** Native Applications, e.g., Android Applications, represent the way of implementing mobile business applications using the standard API. They are implemented using the native programming language (e.g., Java for Android), and therefore require more special knowledge of device-specific development issues. They, in turn, offer the most possible functionality through direct use of respective platform provided programming libraries and interfaces. In addition, native applications show the best performance and applicability for users.

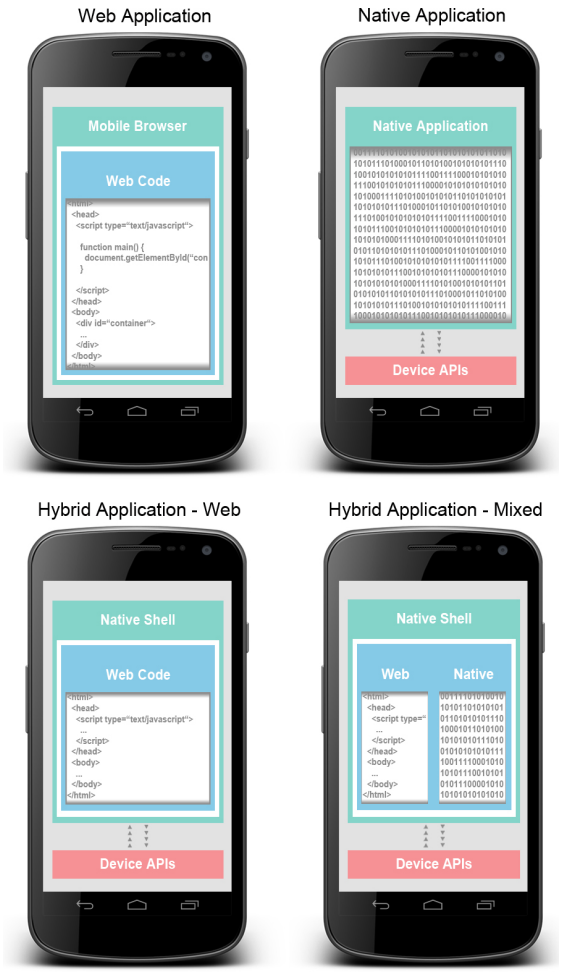

Figure 1: Different types of mobile business applications according to (IBM Corporation, 2013).

Figure 1 illustrates the previously mentioned development types of mobile applications pointing out the differences related to application code.

Table 1 presents to advantages and disadvantages of the different development types of applications. Considering the fact that we need access to device features like the Bluetooth communication stack, we only can address the type of application that supports this specific feature. In addition, we want to achieve a high motivation using our fitness application. Therefore, a robust retrieval of sensor data and finally visualize the vital signs adequately will be crucial for the overall user acceptance. Moreover, it should be possible to integrate vendor-specific software, like custom drivers for sensors. All requirements have indicated to us implementing the fitness application "XFitXtreme" as native application. We decided to develop on the Android platform, because of its full access to the Bluetooth communication stack.

3.2 Architecture

We have discussed different development types of applications, and considered their advantages and disadvantages. Based on this, we present our architecture applied to "XFitXtreme". Figure 2 therefore shows our architecture. In the further course of this Section, we will discuss the different conceptualized layers of the architecture in detail.

3.2.1 Android Application Framework

The Android Application Framework ① encompasses a set of APIs enabling the development of native Android applications. Therefore it provides native libraries that are accessible through those APIs. With these libraries, it is possible to access a database, call webservices, or communicate via Bluetooth. Android Applications (e.g., our "XFitXtreme" application) directly interact with this application framework, which manages basic functions of the smart mobile device, such as automatical resource management. Being a mobile business application developer, this basic toolset enables for building your own application.

3.2.2 XFitXtreme Application

The architecture of "XFitXtreme" ② includes the following major components of a mobile application:

- To ensure maximum flexibility and portability of various components, the latter are strictly separated. This means that user-interface components ③, such as widgets or activities, can be easily replaced or adjusted without altering the specific application logic.
- The SQLite database ④ is a default feature of Android and is used to store user data (e.g., such as training information), application settings (e.g., such as language settings), and, for example, recorded vital signs from external sensors. Thus, it is basically possible to evaluate and visualize them afterwards. Accessing a database has been provided through an encapsulated management component. The same applies to the local file system ⑤, which contains images or videos.
- The "XFitXtreme Specific Components" ⑥ include classes and interfaces for the various features necessary in the fitness domain (e.g., administration of workout). Furthermore, visualizing sensor data is implemented by the usage of the Google Charts API (Google Inc, 2013). This allows for a comfortable creation of charts based on gathered data from external sensors.
- The most important component of our "XFitXtreme" Application is the sensor framework ⑦ (cf. Section 4.2.5). This framework enables us to integrate new sensors easily, without reimplementing basic functions like the establishment of a Bluetooth connection. Our mobile business application "XFitXtreme" currently uses the Bluetooth component feature of the framework, since external sensors in the fitness domain are usually connected wireless.

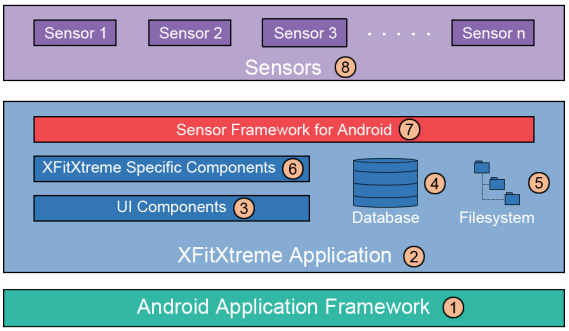

Figure 2: General architecture of XFitXtreme.

3.2.3 Sensors

This Section introduces sensors ⑧ used in this project for collecting vital signs of the CrossFit athlete. In addition to general information, we will present technical challenges of the device sensors. Currently, we integrated two different sensors, namely the "Polar Wearlink+ Heart Rate Monitor" (Polar Electro GmbH

Table 1: Advantages and disadvantages for the different development types of applications.

Feature	Web App	Hybrid App - Web	Hybrid App - Mixed	Native App
Execution within	Web browser	WebView within Native Shell	Native Environment within Native Shell	Native Runtime Environment
Programming Language	Web only	Web only	Native and Web	Native only
Code Portability	High	High	Medium	None
Access Device Features	None	Low	Medium	High
Graphics and Animations	Medium	Medium	High	High
3rd Party Libraries	JavaScript	JavaScript	JavaScript and Native	Native
User Experience	Low	Low	Medium	High
Performance	Low	Low	Medium	High

MedChoice Oximeter MD300X318T

Polar WearLink+

Figure 3: Sensors used in our mobile business application.

Deutschland, 2013), and the "MedChoice Oximeter MD300C318T" (MedChoice, 2013), which are shown in Figure 3.

The "Polar WearLink+ Heart Rate Monitor" is one of only two Bluetooth athletic heart rate monitors available for Android smart mobile devices. The sensor itself is clipped to a soft textile belt and is worn around the chest (i.e., directly below the chest muscles). As soon as the belt is put on correctly and paired with a device, it starts sending data packages via Bluetooth (2,4GHz frequency band). For pairing, no handshake is required, making it easy and straightforward to use this external sensor.

The "MedChoice Oximeter MD300C318T" not only provides the heart rate, but also measures the oxygen saturation of the blood. It was designed for medical use and therefore offers more information than the Polar belt. The sensor itself has an OLED display which visualizes the retrieved vital signs showing both plain values and corresponding diagrams (e.g., a bar chart as amplitude of the pulse). Compared to the Polar belt, this sensor needs a complex pairing and synchronization protocol to send vital signs, but offers two different modes (e.g., real-time mode and non-real-time mode). Due to the fact that we want to support the athlete during his training session, we have focused on the real-time mode.

4 PROOF-OF-CONCEPT

In this Section, the developed mobile business application "XFitXtreme"[1] is presented as a proof-of-concept implementation. Section 4.1 describes the implementation of the business application, while Section 4.2 presents the lessons learned of this project.

4.1 The XFitXtreme Application

"XFitXtreme" is an application designed to display the functionality offered by the developed sensor framework. The framework establishes and manages a Bluetooth connection to a sensor the training athlete is wearing. The sensor used here is a chest belt with an integrated heart rate monitor from the "Polar WearLink+". Other sensors have also been paired (e.g., "MedChoice MD300C318T", see Figure 4). "XFitXtreme" has the task of recording and graphically evaluating the vital parameter supplied by the framework. Simultaneously, it is a tool satisfying all needs of the athlete training CrossFit. The UI is held clean and simple to ensure an adequate display of the necessary data and guarantee a good usability. "XFitXtreme" offers useful features described in the following.

The most important feature is supplying the individual training plan of the day. This is done in two different ways. If internet access is available, the workout is parsed from CrossFit website (CrossFit, Inc., 2013) using the "Workout of the Day" (WOD) feature. This website posts and updates the "WOD" on a daily basis, which is a large advantage making training plan logic unnecessary. If no connection is available, the "Classix" feature offers a database, listing many workouts held locally. These workouts are displayed in a list, offering the possibility to view the exercises included in the workout before selection. After choosing or parsing a workout, the exercises build-

[1] video available under `http://vimeo.com/60436751`

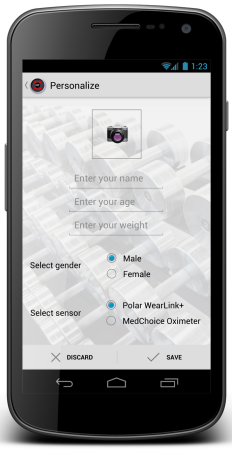

Figure 4: Dialog to connect to various external sensors.

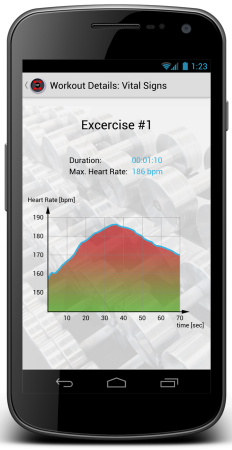

Figure 5: Visualization of vital signs during the workout.

ing up the workout are displayed in a list. This list can optionally be posted on Facebook. The exercises, if not familiar, can be searched on YouTube by simply tapping on the element in the list. If all exercises are familiar, the play button leads to the final activity. This activity displays the entire workout and shows all relevant data, including the heart rate data offered by the framework. The best time for the daily workout is automatically loaded by a web service accessing an online database. The sensor is connected automatically via Bluetooth and, after that, starts transmitting data. The beginning of the workout is signalized by pressing the go button. The application takes the time the athlete needs to complete the workout and starts recording the heart rate data. Finishing the workout uploads the taken time, and the daily ranking is displayed so the athlete can classify his performance. The monitored data is visualized in a graph to give the athlete a quick overview of his vital activities during the workout (cf. Figure 5).

In addition to these main features, "XFitXtreme" offers further features to support the athlete in his training. The first is a timer, which combines a simple stopclock with a lap option and an interval timer. The stopclock can take time and a random number of laps, which are displayed in a list. The interval timer offers the possibility to set up individual timers and save these persistently. These timers are displayed by name in a list after selecting this feature. Each timer can be named, an arbitrary number of rounds can be set and finally a random number of intervals can be added. Each interval is characterized by a name, a time, and a color. These attributes are displayed later on while playing the timer, so the athlete knows which interval is currently running. Selecting a timer from the list plays the timer. Each interval added to the timer is displayed in the order set before, counting down the set time, displaying the name and the color set as background color. That way, the change of interval is not only signalized acoustically, but also visually by a change in the background color. The end of the workout is signalized by an applause. The timers can be deleted if no longer used.

CrossFit is unfortunately not too popular. Depending on the area, in which the athlete is training, it can be difficult to find other people sharing their passion for this sport. Therefore, "XFitXtreme" offers an IRC chat. The latter is an open chat, in which all users share a forum. This increases the chance of athletes being online simultaneously, establishing an online community. It is possible to send messages privately to other users as well. These messages are not visible for the other users in the forum, and are displayed in a different color. That way both users know these messages are private.

4.2 Lessons Learned

In this Section, we present aspects elicitated during the development of this mobile business application using the two mentioned external sensors.

4.2.1 Basic Development Types for Mobile Applications

The market research company Gartner predicted in early February 2013, that by 2016, 50% of all mobile business applications are developed using hybrid technologies (Gartner, Inc., 2013). We realized while designing and implementing "XFitXtreme", that none of the current available hybrid approaches (e.g., PhoneGap, Appcelerator) met the essential requirements for our application. The same applied to the implementation of pure web applica-

Table 2: Bluetooth support for different development types of business applications.

	Bluetooth Support
Web App	No Support
Hybrid App - Web	Limited Support
Hybrid App - Mixed	Limited Support
Native App	Full Access

tions. We revealed unrestricted access to the Bluetooth API and full support of the specified Bluetooth profiles (e.g., SPP profile) as the most important arguments for developing our native mobile business application "XFitXtreme". These features are not available within a hybrid development framework, such as PhoneGap or Appcelerator. Without this capability, the integration of external and wireless sensors is not possible.

Table 2 shows the Bluetooth support for the different development types of applications defined in Section 3.1.

Furthermore, these hybrid development frameworks did not fulfill our requirements regarding performance of data processing, and a suitable as well as interactive visualization (e.g., using animated diagrams).

4.2.2 Establishing a Connection

One challenging issue we encountered while using external sensors was the handling of different types of connections, such as Bluetooth, USB, Infrared, or Wi-Fi. Depending on their characteristics, some vendor-specific communication mechanisms have to be used. For example, the "Polar WearLink+ Heart Rate Monitor" belt (Bluetooth connection) requires, after the successful pairing with the smartphone, no further synchronization or handshake techniques to obtain data. In contrast, the "MedChoice Oximeter MD300C318T" (Bluetooth connection) sensor needs extended synchronization and a test sequence before data is available. Both sensors have in common that data is actively and periodically transmitted to the smart mobile device (cf. Figure 6).

4.2.3 Structure of the Sensor Data Packages

In this Section, insights regarding to the structure of the data packages received by an external sensor are shown. The most challenging aspect has been the vendor-specific and proprietary exchange format. No official documentation on the structure of data packages was found for the "Polar WearLink+ Heart Rate Monitor". This information must be gathered through

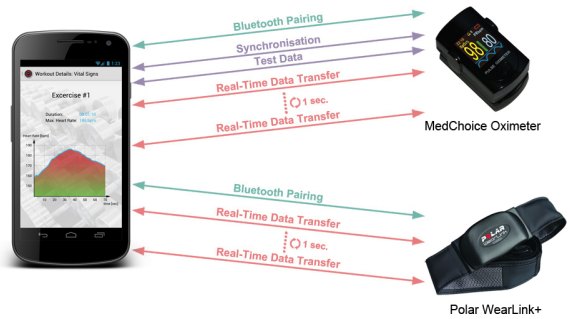

Figure 6: Different communication mechanisms for the external sensors.

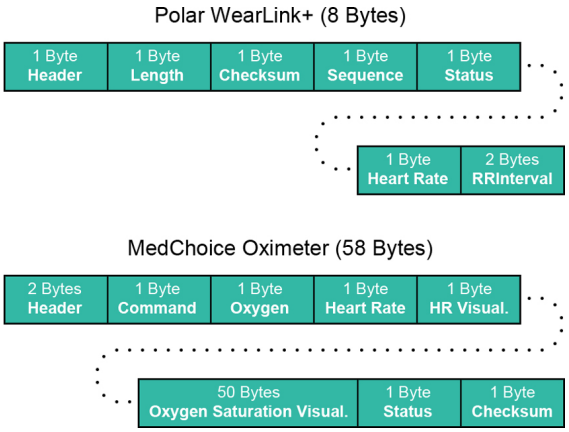

Figure 7: Different structure of data of the implemented sensors.

internet research, or even worse, through reverse engineering of the data packages. In the case of "MedChoice Oximeter MD300C318T" we received an extensive documentation about the data packages after sending a request to the producing company.

Figure 7 shows differences in the structure of the data packages of our used sensors. They greatly differ not only in length of transmitted data packages, but also in contained data. The "MedChoice Oximeter MD300C318T" provides more information about the vital signs than the "Polar WearLink+ Heart Rate Monitor", which results in a more complex data package structure (cf. Figure 7).

4.2.4 Connection Problems

We encountered a problem concerning connection issues. Both Bluetooth and Wi-Fi (B and G standard) transmit in the unlicensed ISM (Industrial, Scientific and Medical) frequency band of 2.4 GHz (Lansford et al., 2001). This means, it may cause interference with a Wi-Fi network. This noise from other networks possibly causes a connection loss between the smart mobile device and the external sensor, or a complete

Table 3: Bluetooth classes and approximate range

Bluetooth Class	approx. Range
Class 1	approx. 100m
Class 2	approx. 20m
Class 3	approx. 10m

disconnection. In addition to this, Bluetooth has limited range, depending on the Bluetooth class. Table 3 shows the approximate range of different Bluetooth classes (Bluetooth Developer Portal, 2013). Short ranges as a characteristic property of Bluetooth can be critical. Especially in the medical domain when the physician moves too far from the patient and the connection gets interrupted.

If the connection between the smart mobile device and the external sensor gets disconnected, the question arises, how to suitably respond. One option would be that the smart mobile device should try to initiate a reconnect periodically. Depending on the scenario, it is important to interact with the user and inform him about the disconnection. For our presented fitness application "XFitXtreme", it is not important, because no critical information must be exchanged. Nevertheless, in the medical domain, it could be very important to tell a physician that currently there is no coupling between the devices (e.g., smartphone and heart rate monitor), and hence no real-time information about the vital signs is available.

On the one hand, some external sensors have a small internal memory, where data can be cached in case of a disconnection. Thus, it is possible to cope with a short time of disconnection and provide cached data in addition to the real-time values which have been gathered in the meantime. On the other hand, the smart mobile device has to deal with missing data. In some cases, already recorded data needs to be discarded, since the measurement is no longer valid.

In the context of a wireless connection between the smart mobile device and external sensors, all these questions need to be considered.

4.2.5 Sensor Framework

Reflecting on the previously described lessons, we developed a basic mobile sensor framework which can be easily integrated into existing Android applications. It shall encapsulate already mentioned heterogeneity of data structure of different sensors (cf. Section 4.2.3) as well as the communication between the external sensor and the smart mobile device (cf. Section 4.2.2). In addition, connection problems (cf. Section 4.2.4) should be handled. Furthermore, the

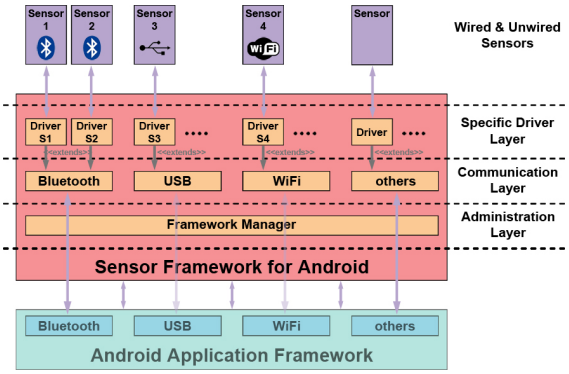

Figure 8: Sensor framework architecture.

sensor framework should provide basic functionality, such as establishing a connection to a sensor, or receiving data from a sensor. Finally, requirements, such as the extensibility, maintainability, or scalability have to be considered. Figure 8 shows the architecture of our sensor framework. The individual application layers are described in the following sections.

The "Android Application Framework" serves as the basis for our mobile business application. This framework provides the fundamental functions and interfaces (e.g., for the Bluetooth or USB connection).

Our described sensor framework consists mainly of three layer, namely the Administration Layer, Communication Layer, and Specific Driver Layer.

The "Administration Layer" contains the "Framework Manager", which is responsible for the basic communication with the framework. It provides basic methods which an application developer needs to use external sensors in his mobile business applications. This includes methods concerning the connection (e.g., open or close a connection), receiving data from a sensor, or configuring a sensor using special messages. An application that implements the sensor framework communicates only with this framework manager.

The second layer, namely the "Communication Layer", provides all classes to manage the different types of connections, like Bluetooth, WiFi, or USB. These classes rely on the Android Application Framework through the provided interfaces.

The third layer named as the "Specific Driver Layer", includes classes for the individual sensors. These classes are derived directly from the associated communication layer class, and contain information for setting up the connection. Consequently, only methods for configuring the sensor, or receiving the data, will have to be implemented.

The "Wired & Unwired Sensors" layer represents the external sensors connected to the mobile de-

vice. In our application scenario, these are the aforementioned discussed sensors "Polar WearLink+ Heart Rate Monitor" and "MedChoice Oximeter", which provide us with the gathering of vital signs.

5 RELATED WORK

In the following we will discuss related projects. For this purpose, we distinguish these projects in two categories, namely "Consumer" and "Research" projects.

The most known example from the consumer domain is the mobile application "runtastic" (runtastic GmbH, 2013), which is available on all major mobile device platforms. This application offers the ability to integrate different pulse sensors to the mobile device. The vital signs as well as the GPS position are recorded in real-time. Finally, the gathered data can be analyzed within the runtastic platform and shared among friends. Runtastic offers an easy to use application, with almost no installation or configuration effort. Additionally, runtastic now offers its own hardware shop, in which 100% compatible GPS and pulse sensors can be bought.

Another well-known example of such an application is "Nike+" (Nike Inc., 2013). However, there is a small sensor inserted into the shoe, which will then track all movements (GPS position, distance, or stamina). This sensor is then connected to the smart mobile device to visualize the collected information. In addition, all "Polar WearLink+" sensors can be used and coupled with the smart mobile device in order to gather vital signs of the athlete.

Both consumer applications assume that the connection between the external sensors and mobile device in the context of the fitness application is reliable. Unfortunately, we were not able to adopt their communication structure because of the proprietary philosophy. In addition, they offer hitherto only the use of pulse sensors.

Research projects like "Open Data Kit" (Open Data Kit, 2013) address a suitable mobile data collection. Open Data Kit aims at providing a self-containing framework for data collection. The integration of external sensors in Android smart mobile devices are discussed in (Brunette et al., 2012) or (Chaudhri et al., 2012). Even though, we can not adopt results directly for our goal. The reason is that we can use "Open Data Kit Sensors" framework only inside the "Open Data Kit" project, which would be to complex. In addition to this, the applicability for our future usage in the medical domain would be not suitable. Instead, we are trying to create a sensor framework, which can be used as a standalone application. Thus, we want to enable application developer to use this framework in their own projects and applications.

6 SUMMARY & OUTLOOK

In this paper we presented a real-world mobile business application from the fitness domain. This application was implemented to be used with external sensors for gathering vital signs of athletes in order to support them in their frequent training sessions. To implement such a mobile application, we first had to determine the appropriate development type of application (e.g., Web Application or Native Application). In case of the "XFitXtreme" application we implemented a native application because of the lack of Bluetooth functionality for the other basic development types of mobile applications. Moreover, we aim at real-time processing and visualization of gathered vital signs and hence a challenging and realistic application was a must. Additionally, we presented a system architecture for business applications that shall be used with external sensors and explained our layered design. To strengthen our described architecture, the fitness application "XFitXtreme" using it was shown. However, we set the main focus on the lessons we have learned from this project. We therefore discussed different aspects related to the connection between the mobile device and the external sensor as well as different structures of data packages retrieved by the external sensors. Our experiences resulted in developing a sensor framework, which is currently used in a real-world fitness application. The framework was discussed, and needed functions for the CrossFit training were presented.

In the future, we plan to extend the sensor framework described in this paper. We plan a feature to add more sensors, and support different types of connection standards as well, such as ANT, ANT+ (Dynastream Innovations Inc, 2013), or ZigBee (ZigBee Alliance, 2013). Furthermore, we want to implement a mobile process engine to interact with the coupled external sensors using our described sensor framework. This would empower us to carry out the data collection with sensors as a process. However, problems, like flexibility, should be carefully considered (Reichert and Weber, 2012). Besides technical improvements, we want to add more features to visualize gathered data from sensors. We therefore want to implement the Google Charts Tools (Google Inc, 2013) in our sensor framework to allow athletes to interactively browse through their different vital signs. Another goal we pursue is to provide this sensor frame-

work on other mobile operating system, like Apples iOS. This enables us for example to reveal insights to the interplay between the iOS platform and external sensors. Finally, we focus more on scenarios of the medical domain using our framework to ease daily work of medical staff.

REFERENCES

Adobe Systems Inc. (2013). PhoneGap. http://www.phonegap.com/. last visited: 22. February 2013.

Appcelerator Inc. (2013). Titanium Mobile App Development Platform. http://www.appcelerator.com/platform/titanium-platform/. last visited: 22. February 2013.

Bluetooth Developer Portal (2013). Compare with other Technologies. http://developer.bluetooth.org/Technology Overview/Pages/Compare.aspx. last visited: 22. February 2013.

Brunette, W., Sodt, R., Chaudhri, R., Goel, M., Falcone, M., Van Orden, J., and Borriello, G. (2012). Open data kit sensors: a sensor integration framework for android at the application-level. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 351–364. ACM.

Chaudhri, R., Brunette, W., Goel, M., Sodt, R., VanOrden, J., Falcone, M., and Borriello, G. (2012). Open data kit sensors: mobile data collection with wired and wireless sensors. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, page 9. ACM.

CrossFit, Inc. (2013). CrossFit - Forging Elite Fitness. http://www.crossfit.com/. last visited: 22. February 2013.

Dynastream Innovations Inc (2013). THIS IS ANT - the Wireless Sensor Network Solution. http://www.thisisant.com/. last visited: 22. February 2013.

Gartner, Inc. (2013). Gartner Says by 2016, More Than 50 Percent of Mobile Apps Deployed Will be Hybrid. http://www.gartner.com/newsroom/id/2324917. last visited: 22. February 2013.

Google Inc (2013). Google Chart Tools. https://developers.google.com/chart/. last visited: 22. February 2013.

Heitkötter, H., Hanschke, S., and Majchrzak, T. A. (2013). Evaluating cross-platform development approaches for mobile applications. In *Web Information Systems and Technologies*, pages 120–138. Springer.

IBM Corporation (2013). IBM Worklight - Mobile application platform. http://www-01.ibm.com/software/mobile-solutions/worklight/. last visited: 22. February 2013.

Lansford, J., Stephens, A., and Nevo, R. (2001). Wi-fi (802.11 b) and bluetooth: enabling coexistence. *Network, IEEE*, 15(5):20–27.

MedChoice (2013). The MedChoice MD300C318T. http://www.pulsoximetergerate.com/products.asp. last visited: 22. February 2013.

Nike Inc. (2013). Nike+. http://nikeplus.nike.com/plus/. last visited: 22. February 2013.

Open Data Kit (2013). Open Data Kit - magnifying human resources through technology. http://opendatakit.org/. last visited: 22. February 2013.

Polar Electro GmbH Deutschland (2013). Polar WearLink+ Sensor Bluetooth. http://www.polar-deutschland.de/de/produkte/. last visited: 22. February 2013.

Pryss, R., Langer, D., Reichert, M., and Hallerbach, A. (2012). Mobile task management for medical ward rounds - the medo approach. In *1st Int'l Workshop on Adaptive Case Management (ACM'12), BPM'12 Workshops*, LNBIP. Springer.

Pryss, R., Tiedeken, J., Kreher, U., and Reichert, M. (2010). Towards flexible process support on mobile devices. In *Proc. CAiSE'10 Forum - Information Systems Evolution*, number 72 in LNBIP, pages 150–165. Springer.

Reichert, M. and Weber, B. (2012). *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer, Berlin-Heidelberg.

runtastic GmbH (2013). runtastic - makes sports funtastic. http://www.runtastic.com. last visited: 22. February 2013.

Trowbridge, R. and Weingarten, S. (2001). Clinical decision support systems. *Making health care safer: a critical analysis of patient safety practices. Evidence Report/Technology Assessment*, (43).

ZigBee Alliance (2013). ZigBee Alliance - Control your world. http://www.zigbee.org/. last visited: 22. February 2013.