FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Doktorarbeit in Informatik

A multimodal biosensor-based system with compatibility for telemonitoring and epidemiological services

Chen Chen

FAKULTÄT FÜR INFORMATIK

Lehrstuhl für Echtzeitsysteme und Robotik

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

A multimodal biosensor-based system with compatibility for telemonitoring and epidemiological services

Chen Chen

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr.rer.nat.)
genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Alin Albu-Schäffer

Prüfer der Dissertation:
1. Univ.-Prof. Dr. Alois Knoll
2. Prof. Dr. Alexander Horsch, University of Tromsø, Norwegen
3. Univ.-Prof. Dr. Dr. Heinz-Erich Wichmann (em.) Ludwig-Maximilians-Universität München

Die Dissertation wurde am 13.08.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 19.01.2016 angenommen.

Ich versichere, dass ich diese Dissertation selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 10. August 2015 Chen Chen

Acknowledgments

I would like to thank many people for their support during the time I finished this thesis. Without them, the thesis might be postponed for some more years, or even never be finished.

First I want to thank my supervisor, Prof. Alexander Horsch, for his guidance during the last four years. The PhD was initially inspired by a talk between him and Andre Dias, who became my lovely colleague later on. Whenever I felt confused about the way my project should go forward, Prof. Alexander Horsch always provided me with his insights and remarkable suggestions. I appreciated each elaborate discussion with him for significant decisions made for my research work. And I also thank him for his understanding and patience due to my special family status.

Next, I want to thank Prof. Alois Knoll and his chair for their professional and kind support. For the specialization of my PhD topic, I got not only feedback in my field from those people working at Information Institute VI, but also workplace in the office there. I give special thanks to Prof. Alois Knoll, who always reminds me to think over the innovation of my work. More thanks are given to his secretary Mrs. Gertrud Eberl, who helps with the paperwork most of the time.

I was lucky to choose IMSE to start my PhD four years ago. Here there are lovely colleagues with their willingness to share their experience. Thank Prof. Klaus A. Kuhn for hosting me and sharing the workplace resources in his institution. It is a great pleasure to stay with Dr. Petra Striebeck, who gave me a lot of help on system testing, as well as Mr. Andreas Enterrottacher, who helped setup and maintain the system server running at IMSE with technical support. And most of all, I am grateful that I am a member of an international group. Except for Prof. Alexander Horsch, it is the ever best thing in my life to meet Lukas Gorzelniak, who inspired me on the project and I admired to learn from. He brought laughs to the office which made me good mood almost every day. I do hope he will find a good girl to go through together the rest of his life with happiness. Andre Dias, a typical Portuguese with humor, helped me from the first day of my PhD on programming, and the operation of Ubuntu. Despite colleague relationship, I also treated him more as a good friend and elder brother, with whom we can enjoy music shows and football games with beer in the pubs around Schwabing. I wish him and his baby the best. Thank Kathrin Thaler and Sandra Ortlieb, especially the latter, for their help to deal with the routine issues when I was absent from the office.

Additional appreciation should give to Dr. Stefan Karrasch from LMU, and Prof. Holger Schulz from the Helmholtz center for their valuable feedback on the system requirement analysis and system design. The PhD project also benefited from the great contribution of students in terms of interdisciplinary projects. I would particularly like to thank Julian Horsch, Felix Schindler, Michael Dorner, Ivaylo Dimitrov, and Volker Jacht. I am also very much thankful for the financial funding from GSISH, and the GSISH management team for their support.

Last but not least, I must express my sincere gratitude to my beloved family. Though exhausted, my parents and parents-in-law took care of my baby so that I could concentrate on thesis writing. I dedicate this thesis to my dear husband Xiaochen, whom I met during my PhD, and my dear daughter Xiao, who was born during my thesis writing. Without you two, this thesis would have been done one year earlier.

Abstract

Wireless body sensor networks are popular in various application fields. Often, such a system serves for a specific purpose and is not extendable to integrate other sensors for other usages. When multiple purposes with different application needs have to be achieved, usually several systems have to work together, whose redundant structures bring extreme inconvenience to both subjects and healthcare personnel or other researchers. How to integrate these systems with similar architectures into one flexible platform which can be used for both telemonitoring and epidemiological studies is the question that this thesis attempts to answer.

Thus, according to the typical software development procedure, a dedicated requirement analysis has been conducted, resulting in 44 requirements extracted from 8 sessions with 13 people (including 2 doctors, 1 physician, 2 PhD students, 3 patients and 5 technicians). These requirements are grouped into three categories of subsystems. Based on the system requirement analysis, a model called SensCoAP is proposed. It is a generic design by utilizing state-of-the-art which reserves extensibility and modularity for future use. To be innovative, a scheme that allows the system to be flexibly customized for telemonitoring and epidemiological services is specially designed, which is supposed to avoid any redundant or duplicated structure in the design as much as possible in the two different scenarios. In general, it has also the capacity of including more types of sensors for other application purposes. It is supposed to have the capabilities of acquiring, storing, forwarding, analyzing the datasets from the biosensors as well as interacting with the users by an application user interface on the smartphone when necessary. The design of the SensCoAP and its subsystems is modeled respectively by UML 2.0. A vertical slice of the complete design has been implemented as prototype involving Shimmer sensors, Android smartphones running a specific application, and a server in Django framework. Except the interactions with the subjects through the app in the smartphones, the server also provides user interfaces for the users with an access control scheme depending on their roles. The prototype has been tested by 31 volunteers for two rounds, including a technical update of the system in between. A proof of concept indicates that the SensCoAP system guarantees the mobility of the subjects and provides them with reliable services. From the analysis of the questionnaires in the two test rounds, it is shown that the SensCoAP won a relatively high acceptance among the testers, especially among those elderly who are not technology-friendly. Moreover, it matches different needs from the healthcare and research fields respectively by its wide applicability in different scenarios.

Though the SensCoAP is only partially implemented and there is still some future work to do, it demonstrates that it may be one solution to solve the dilemma caused by the lack of customizable systems serving for both fields of telemedicine and epidemiology.

Contents

List of Figures

List of Tables

List of Abbreviations

BAAS	Biosignal Archiving and Analysis Server
BAN	Body Area Network
BSN	Body Sensor Network
CA	Certification Authority
COPD	Chronic Obstructive Pulmonary Disease
CTT	Concur Task Tree
CTTE	Concur Task Tree Environment
DBMS	Database Management System
ECG	Electrocardiography
EEG	Electroencephalography
ERD	Entity-Relationship Diagram
GUI	Graphical User Interface
HCP	Healthcare Professional
ICT	Information and Communication Technology
MDA	Model-Driven Architecture
MDD	Model-Driven Development
MERMOTH	the Medical Remote Monitoring of clothes
MVC	Model-View-Controller
ORM	Object-Relational Mapping
RBAC	Role Based Access Control
RES	Researcher
REST	Representational State Transfer
SensCoAP	Sensor Communication and Analysis Platform
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI	User Interface
UML	Unique Modeling Language
WBSN	Wireless Body Sensor Network
WSGI	Web Server Gateway Interface
WSN	Wireless Sensor Network

Glossary

base station is a device with basic processing capabilities to receive vital information, and either displays processed information on a user interface or transmits the information to a main server, where the collected data can be accessible through an infrastructure network, such as Internet. It might be a PDA, a smartphone, a pocket PC or a custom designed microcontroller-based device *(Pantelopoulos et al. A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis. IEEE J. SMCC. 2010 vol. 40 no. 11-12)*.. 10

biosignal archiving and analysis server is the subsystem of SensCoAP containing the server, responsible for storing data, processing data and managing devices and user data. It interacts with an external database via the defined interfaces and is secured by a user access control scheme.. 37

body area network is a network topology around human body formed by one or more sensor nodes and a body-worn or closely located device as base station, e.g. smartphone, PDA, laptop. They are usually connected as a mesh net, sometimes also as a star or tree structure. It is a small-scale network and each node communicates with wires or wireless technologies. Those sensors are in charge of collecting data and the base station serves as data sink to receive data packets sent from sensors for further processing. In this thesis, it particularly refers to the subsystem of SensCoAP running on the smartphone, which together with the biosensors constitutes the BAN. It is in charge of receiving data, caching data and forwarding data to the BAAS.. 13

concur task tree is a method commonly used for user interface design. CTT is represented by a tree, where each node represents one of four task categories: Abstract, Application, User and Interaction.. 43, 65

entity-relationship diagram is the graphical representation of the E-R model. It provides a conceptual method to describe the entities, attributes and their relationships and is usually used for high level data model. It explicitly explains which data are stored in the system (entity) and how the system shares the data (relationship).. 61

model-driven development is a newly proposed method for fast and automatic software development. In MDD, applications are transformed to a higher level of abstract model and to specific models or code later in the development. MDD uses models systematically as main components in the entire software development life circle, thus to reduce the complex of the software, and make it more flexible to extend.. 43

Model-View-Controller is a well-known design pattern for medium-scaled projects. It consists of Model, View and Controller which represents data, presentation and logic correspondingly. Model is used to encapsulate all related data and the data processing methods. View level is to intended display data, while controller is to organize and control the flow between different levels, and respond to events. MVC pattern is especially useful for web application development.. 49

Representational State Transfer uses the existing HTTP protocol features like the request methods GET, POST,PUT,DELETE and status codes to build up a resource oriented webservice. When a resource marked by a URL needs to be changed, the action is able to be performed on the resource directly with the corresponding HTTP request methods, instead of designing a new vocabulary for its interface in the XML encoded requests.. 57

SensCoAP refers to the entire system that consists of biosensors, BAN and BAAS. It is based on wireless communication protocol and used particularly for sensor networks attached to human bodies. It intends to be used in healthcare and medical centers for monitoring the behavior of the patients and for epidemiological studies as well.. 43

TinyOS is an open source operating system developed by Jason Hill as part of his PhD, which was especially designed for small wireless-embedded devices. It is now in a significant proportion of academic research in this field and even as the basis for some commercially produced operating systems. It couples with NesC programming language *(Hao Y, Foster R. Wireless body sensor networks for health-monitoring applications. Physiol. Meas.2008 Nov;29(11):P27 - P56)*.. 12

Part I.

Introduction

Chapter I

Introduction

Contents

Remote monitoring systems for elderly applying information and communication technology (ICT) are widely used nowadays. Meanwhile the systems for multimodal biosensor-based measurements with similar structures are increasingly used in epidemiological studies as an innovative tool under various circumstances. This chapter will describe the current situation of the ageing society, where the doctorate project was motivated. It derives the urgent need of a system with compatibility for both medical telemonitoring and epidemiological services, and then the objectives of this thesis are stated.

1. Ageing Society and the Rising Demands from Epidemiology

Ageing society is a global phenomenon. The worldwide elderly population with the age above 65 will reach 761 million by 2025 and even more than 1 billion by 2050 [1]. The ageing problem severely burdens the costs of public healthcare service, for long-term healthcare is usually quite expensive in western countries, which directly causes the 3-5 times higher healthcare costs for elderly above 65 years old compared to people under 65 [2]. Moreover, it is a heavy pressure on public healthcare expenditure to provide the growing number of elderly with health- and long-term–care services [3–5]. On the other hand, the growing number of elderly brings new challenges to technology development. In order to reduce hospitalization and realize continuous health monitoring, telemonitoring provides novel technological solutions to enhance elderly independence in their daily living environment. ICT-based home monitoring systems has been proved to be reliable, accurate, efficient and be able to highly reduce the healthcare cost [3, 6–10].

Meanwhile, there are newly rising demands from epidemiology. Traditionally, epidemiological studies can be conducted through various assessment methods including surveys, self-report, behavioral observation, dietary measure, etc [11]. However, the reliability and validation of those methods are not promising [12]. Along with the development of the sensor technologies, the advantages of sensor-based measurement in the application domain of epidemiological studies turn to be evident after comparative studies [12–15], which fulfills the criteria for use in epidemiological studies: valid, reliable, practical and nonreactive [11]. Epidemiological methods using sensor-based measurement are proven to be more objective than the other traditional assessment approaches and extremely useful for large-scale cohort studies [10, 11, 13].

2. Sensor Technology and Sensor Network

Development of sensor technologies and applications of sensors as means for data acquisition started six decades ago, far early in 1950s [16]. As time goes by, the trends of sensor development went through from wired to wireless, from monofunctional to multifunctional, and from low-frequency to high-frequency [17]. Programmable sensors with streaming possibility were also announced several years ago, e.g. Shimmer (Realtime Technologies Ltd., Dublin, Ireland) and wireless Actigraph wGT3X+ (ActiGraph, Pensacola, US). The appearance of wireless sensors released the subjects from wires and ensured their mobility out of homes, which triggered the utilization of wireless sensors in sensor networks. In the scope of wireless sensor network (WSN), sensors embedded with transceivers form an ad-hoc wireless network, in order to achieve certain common monitoring aims under specific environment [10, 17]. More details regarding WSN will be introduced in Chapter II.

Many sensor-based systems applied for telemonitoring and epidemiological studies nowadays are based on WSN [18]. On one hand, in telemonitoring field instead of sensors being wired to a personal computer, wireless devices and communication protocols greatly increase mobility of the subjects because they are set free from being bounded with their homes [10, 18]. On the other hand, traditional epidemiological studies usually have to log the sensor data on sensor units, and upload the data to the study center later on via wires e.g. USB cables. Wireless technologies reduce risk of data loss on their way to the study center, and also solve the problem of limited storage capacity on the sensor units [10, 19].

3. Statement of the Problem

Sensor-based monitoring systems for telemonitoring and for epidemiological research share common technical issues. Both scenarios need sensor units to capture physiological data from human body, and a receiver device to receive and store data. Most of the time the receiver device also has the capability to analyze data [20, 21]. To connect sensors and the receiver, certain communication protocols and transmission safety should be taken into account.

Nevertheless, after a dedicated literature review (which will be concluded in details in Chapter II), it is found that in general there are still hundreds of solutions simultaneously available for monitoring systems due to the following aspects:

a) Devices. Devices involved in the WSN monitoring systems are various: different types of sensor units, ordinary mobile phones, smarthpones, PDA, PC, tablets, etc. In recent years, wireless and mobile devices became more popular among system designers [10, 22]. The choices of number and placement of devices also depend on the movements and activity patterns [10, 19].

b) Communication protocols. In early days, data might be transferred via fix telephone line when most of the devices were wired. As the wireless age goes by, new communication protocols are evoked for more convenient and faster data transmission. For example, the IEEE 802.11 and 802.15 standards are the basis of WSN systems [17, 18].

c) Architectures. The composition of system architectures distinguishes the WSN monitoring systems. Depends on how collected sensor data are forwarded to the receiver, WSN architectures can be classified as two-layer (direct data forward) or multiple-layer (indirect data forward). Usually two-layer or three-layer architecture is common for WSN system design.

d) Purposes of use. A system with specific structure always serves particularly for a certain purpose of use. In telemonitoring scenarios, monitoring system should be interactive, sometimes with medical interventions from doctors to patients. Data forwarding is supposed to be real-time, and the monitoring is continuous without interruption. However, things go to another way in epidemiology. Epidemiological studies require silent observations, without disturbance to the life of the subjects. Real-time data are not necessary so that most of the time the data can be stored temporally on sensor units or other devices for several hours. Thus, locally storage requires a special design in system mechanism which is totally different from that in telemonitoring services [10, 19].

Despite the waste of research resources caused by duplicated development work, validation and usefulness of these monitoring systems still remain with questions. Investigating the published solutions, there is a lack of research on how to enhance flexibility and compatibility for both telemonitoring and epidemiological services, which is represented by a monitoring system that is also able to be used for epidemiology. The joint application has not been explored yet. A system which can be customized in order to shift between the two application scenarios is lacking.

4. Research Question and Objectives

This dissertation is derived from the incompatibility of telemonitoring and epidemiological studies in current WSN monitoring systems. The purpose is to look for the possibility of enhancing the compatibility for the two types of services in one system. Therefore, the following main research question was formulated:

> **How to design a multimodal biosensor data acquisition and feedback system for both medical telemonitoring and epidemiological research, and make it customizable for both types of services?**

The objective of the thesis is to create a generic design and system implementation to fulfill the requirements both in telemedicine and epidemiology, based on open healthcare and ICT standards, by applying state of the art in software design, which involves:

1. *Analyzing system requirements derived from use cases of epidemiological and telemedicine scenarios, and utilizing business processes;*
2. *Utilization of appropriate tools for modeling and state of the art technologies for software development;*
3. *A prototype with modular, adaptable and extendable architecture, at the same time as generic as possible for further extension;*
4. *A proof-of-concept study following the design, demonstrated by the prototype, and corresponding user tests;*
5. *A software design that supports the system to be configurable between the two extreme data communication modes, "near real-time" (e.g. telemonitoring service) and "store & forward" (e.g. epidemiological study);*
6. *Secured data transmission supporting safety and privacy.*

According to the objectives mentioned above, the design procedure will follow the classic "waterfall" life cycle model defined by Royce in 1970 [23] in the scope of software development (see Figure I-1). This style is chosen because the system requirements are not supposed to be changed from time to time due to the limited time frame of doctorate work. However, after proof-of-concept the system design and implementation shall be refined to increase its chance to be ready for practical use. Another round of evaluation was then conducted in order to prove its maturity to be routinely applied for medical and epidemiological research. The project was finally summarized in terms of some intermediate documents and this doctorate thesis.

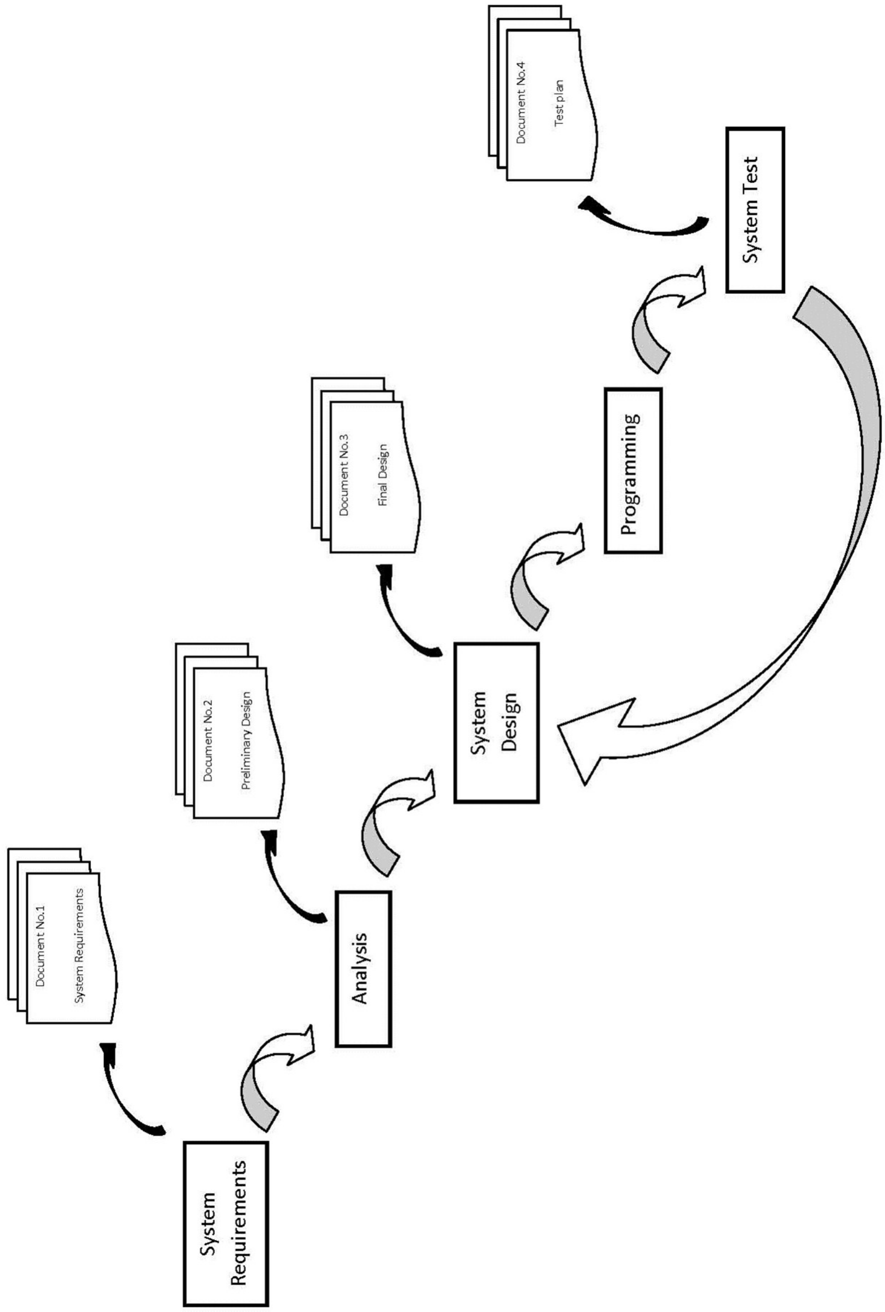


Figure I-1. Life cycle model of the doctorate project

Chapter II

Theoretical and Technical Background

Contents

This chapter consists of two sections of which the first outlines the development of WSN in the scope of healthcare. A classification of the published research projects is summarized in order to compare with the proposed system. State of the art will be explained. In the second section relative theories and technical terms which will be addressed in the following chapters are roughly introduced.

1. State of the Art

In this section, a related literature review on WSN applications in healthcare services is provided, and current achievements in this field are categorized according to their significant features of the system functionalities. The characteristics of each application type are summarized. A comparison of the proposed system with those published solutions is followed by a description of state of the art to emphasize the innovation of this thesis.

1.1. Evolution of WBSN

WSN, which has experienced a long time of development, is an emerging technology area of utilizing wireless sensors to build complete ICT-based health monitoring systems. Compared with wired sensors, wireless sensors obtain the advantages of being flexible and portable to measure in various physical locations. They are more rigid and consume less power [24]. A wireless sensor might be equipped with one or more transceivers together with wireless communication modules, to measure environmental parameters such as temperature, humidity, and light intensity. Tens up to thousands of wireless sensors distributed spatially cooperate with each other or base stations to form a wireless ad-hoc network, namely a WSN. Usually a WSN applies a multi-hop routing algorithm, which treats each node inside the WSN as a forwarder to relay data packages to the gateways. WSN was first motivated by military applications, then later promoted to industry and civil living areas like home automation, process monitoring and healthcare applications [3]. Nowadays, purposes of using WSN were also extended to detecting physical factors, monitoring, collecting data, accessing and evaluating information, decision-making and alerting [24].

However, WSN is not specially designed for monitoring human individuals as the human body is a smaller scale environment which requires appreciation of a slightly different set of challenges. The unsuitable WSN design for monitoring the human body and its internal environment has directly triggered the development of wireless body sensor networks (WBSNs) with locally computational intelligence, which provide pervasive monitoring environments while at the same time guaranteeing the mobility of monitored patients [25]. WBSN comprises a series of miniaturized wearable or implantable sensors which accurately capture physiological data from patients and transmit all the data, either raw or low-level processed, wirelessly to a local mini-processor unit (or known as a base station) or directly to a central server for further processing [26]. The typical scenarios for WBSN applications are vital signals monitoring in hospitals [27], large scale of patient monitoring in disaster, healthcare delivery for elderly, mobile healthcare, etc [28].

1.2. Where are we now?

Due to the fact that a WBSN system is usually formed by miniaturized devices which have several restrictions including short battery life and low processing capabilities [21], many challenges hamper the design of a WBSN system in healthcare monitoring: robustness and reliability, energy conservation, data privacy and security, comfort and unobstrusive operation, mobility, cost and size of the system, quality of service, etc [24]. There is an increasing number of research prototypes and commercial products of WBSN systems for healthcare purposes. However, after examining them, it was found that there was no ideal design. Many systems adopt varying architectural approaches based on their specific application areas. From this perspective, a standard design of WBSN is rather a trade-off between those contradictory parameters of the system constraints [29]. Figure II-1 depicts a general architecture of WBSN systems in the healthcare domain [29]. When designing a concrete system, system designers may vary the details for the application purposes of the system. For example, some systems only adopt certain parts of the architecture. And sometimes even though the components are complete, the transmission protocols and the communication approaches between biosensors and medical center may differ from system to system.

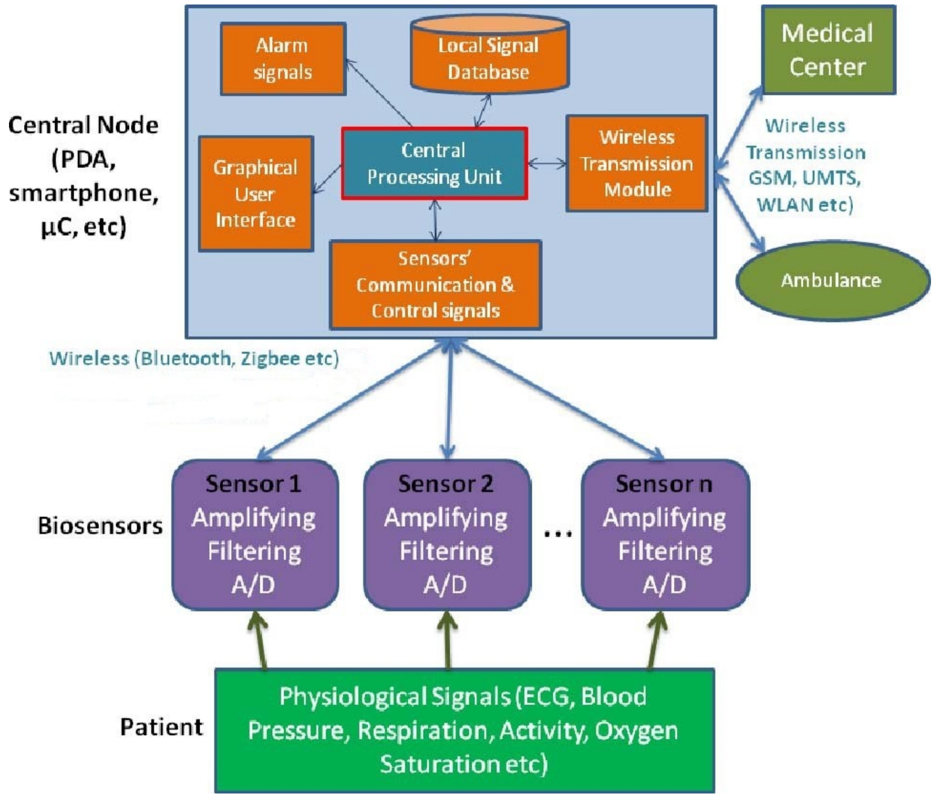


Figure II-1. A general architecture of WBSN used in healthcare monitoring[1] (Copyright ©2010 IEEE)

Early in 2005, the Media Laboratory of MIT announced their wearable feedback systems for ambulatory long-term health monitoring called "LiveNet" [30]. LiveNet used a Linux-based PDA, a SAK2 sensor hub for data acquisition, and a customizable physiological sensing board (BioSense) integrating several self-developed sensors, which also had the possibility of interfacing a wide range of commercial available sensors. It is intended for real-time data streaming and context classification of medical conditions with closed-loop medical feedback. Therefore it developed a three-layer software architecture to support wireless communication, real-time context classifier, data distribution and processing of higher bandwidth biosignals [29]. LiveNet system had been evaluated by many pilot studies incorporating with several healthcare givers for e.g. soldiers' health monitoring, Parkinson symptom detection and long-term behavioral modeling.

Other system proposals based on custom designed platforms revealed many problems in practice. For instance, some prototypes were lack of validation studies [31, 32], and some highly depended on end-user interaction which were not ideal for long-term unobstrusive monitoring [33]. Another typical example is AMON, a project supported by the EU FP5 IST program [34]. It integrated multiple sensor units for various vital signs as a wrist-worn device and utilized GSM communication cycle to forward collected data. After the raw data were analyzed in the telemedicine center, WHO standards were applied for classifying the health condition of the patient, depending on which appropriate inter-

[1] Figure from Pantelopoulos et al. [29]

actions was made from the device. It was the first time that a WBSN research prototype included an evaluation software for real-time processing and analysis of the transmitted data [29]. Nevertheless, the author also pointed in the paper that AMON had certain problems on data reliability and device usability through validation study.

With the development of the sensor technologies, sensor units are gradually miniaturized so that they can be integrated together on the garment to form a sensor network which is also called as smart textiles. MyHeart [35] was a project supported by European Commission and involved 33 partners including Nokia, Vodafone, Philips and Medtronic. The primary aim of the project was activity pattern recognition of the patients for prevention and early diagnosis. Here a kind of smart clothes was used with sensors knitted like normal textiles into the garment, in order to increase the comfortableness of the wearability of the smart system. The advantage of the textile was to save the power supply on the garment and all the sensors were connected by one device for control and synchronization. Thus the overall size of the system was significantly reduced. With the activity sensor embedded in the clothes and an ECG heart belt around the chest or the underwear of the patients, the activity modes could be accurately classified by the system for further medical references such as walking, running, lying, going up/down. Another similar European project MERMOTH [36], which indicated the Medical Remote Monitoring of clothes project, incorporated electrostrictive fabrics and dry electrodes in addition. It measured more physiological parameters such as skin temperature and respiratory inductance plethysmography. Compared with MyHeart, MERMOTH replaced the main device with a PDA, linking a local PC via RF for displaying and interpretation.

Along with the establishment of TinyOS, a significant operating system especially designed for small wireless devices, another trend of WBSN development is to build WBSN applications on motes. Mote-based WBSN employs sensor motes, which is capable of collecting one or more types of physiological signals, to form a body area network (BAN) for transmitting vital data towards a central node or a base station. In the early stage of BAN development there was usually one or two sensors directly connecting with a controller, forming a simple star network topology. It was then followed by a growth in the number of sensors in the network, along with utilization of ad-hoc or mesh networking technologies [37]. A typical representative in the early time of BSN development was the CodeBlue project developed by Harvard University [38] in 2005. It was a medical sensor network platform based on ZigBee compliant MicaZ and Telos motes, for monitoring multiple patients and their environments. To address the reliability communication between medical sensors and multiple end users, a software framework of CodeBlue provided services for multi-hop routing selection, special query of certain data from a specified network node, and localization of patients and doctors/nurses. An evaluation study involving a 30-node testbed of this prototype revealed weakness on reliability of transmission and security issues [29]. Later, Chung et al. [29] enhanced the customization of a mote-based WBSN application “u-healthcare”, consisting of custom 802.15.4 capable nodes and a mobile phone for data visualization and feature extraction. The novelty of the system lay in processing data locally and extracting simple features. A simple decision making algorithm following if-then-else mechanism was applied to those features. Hence, only identified suspicious data were transmitted to the hospital, in case abnormal pattern occurred in patient health status. The system reduced the cost of wireless communication and the time for the healthcare givers to process the incoming data as well. The strength of Human++ project [39] in the Netherlands was a promising long battery life by using the ultra low power radio. Generally, these attempts of constructing WBSN based on sensor motes were limited by the lack of standardization in platforms, system software support and wireless communication for WBSN.

Besides ZigBee, other wireless technologies are also utilized in WBSN applications. WBSN systems based on commercial sensors and mobile devices mostly take the use of Bluetooth due to its large proportion of market occupation. HealthGear from Microsoft [40] launched in 2006 contained a multifunctional sensor board with a non-invasive blood oximeter for measuring oxygen saturation and heart rate, a Bluetooth module for wireless transmission and a cell phone for user interaction via interface. It aimed at detecting sleep apnea of the users by applying detection methods in both time and frequency domains for monitoring the thresholds and peaks of the oximetry signals. The current release has been proven by 20 individual testers and until now there was no technical problem reported during the tests. According to the test results, the system successfully detect mild and sever obstructive sleep apnea events, and the user experience on its functionality and wearability was very good. However, some researchers doubted the validation of its result as the measurements were never been compared with a detailed polysomnograph [29]. Fensli et al. [41] empowered the mobile device in their system to enhance its local processing capability. In their publication they declaimed the true detection of the abnormal signals by the algorithm running on the PDA reached up to 99.2%. Their prototype realized a three-layer architecture and once emergency detected the alarm signals together with the recorded ECG data would be transmitted to the remote clinical stations via GPRS, where the doctors were responsible of setting the threshold parameters of alarms used in the detection algorithm.

All the systems mentioned above stayed only in the research domain. However, some other systems have already come into market for years serving various application purposes from manufactures like Philips [42], Polar [43], Omeron [44] and Zephyr [45]. The common characteristics of those commercial WBSN systems are that their components are small, wearable, low cost, lightweight and with ease of use. The target application areas are normally wellness and fitness. The maturity of those products is still to be improved and the reliability is not high enough for medical purposes.

Overall, due to the weaknesses of current WBSN systems, WBSN has a long way to go before it can replace the traditional monitoring in hospitals.

1.3. The lack in current WBSN development

WBSN systems have experienced decades of development. Regardless of their communication technologies, hardware and design purposes, their structures are all variants based on the general frame showed in Figure II-1 with small adaption. For example, HealthGear [40] realized only two layers from patients to biosensors and to the central node, leaving out further communication to medical centers or healthcare givers. 2-layer architecture loses the possibility of expanding the system functionality to more scenarios, which leads to a limited range of usage. A complete communication chain with 3-layer gains its popularity recently. There is a detailed literature summary of WBSN systems with 3-layer architecture published in recent 10 years [3] for further reference. Considering the usefulness and the research focus of the 3-layer architecture, the proposed prototype in this thesis is also based on the general architecture with three layers.

Meanwhile, more and more epidemiologists have chosen WBSN systems as their tools to conduct epidemiological studies, due to the characteristics of WBSN systems: low cost, accuracy, reliability and convenience. Compared with other epidemiological measurement instruments, WBSN systems are capable of guaranteeing the truthfulness and accuracy of the collected data to the greater extent. However, epidemiology requires services different from telemonitoring.

- Firstly, most epidemiological studies have no need for real-time streaming, as the physiological data are not in urgency. The collected information can wait for several hours up to several days before being forwarded to the study centers. It is also the very reason that the epidemiological study centers generally have a strict budget cost plan, instead of the inestimable cost for real-time communication in the telemonitoring scenario.

- Secondly, considering time and money, epidemiological studies always have predefined duration, whereas telemonitoring usually request long-term continuous services.

- Thirdly, in epidemiology, researchers are supposed to be silent, transparent to the subjects, and the used systems should keep non-invasive, unobstrusive to their daily life. In contrast, telemonitoring sometimes needs feedback from healthcare givers or interactions between patients and doctors/systems, in order to provide medical suggestions and treatments in time.

- Fourthly, epidemiological studies should be conducted pseudonymously, i.e. the relevant personal information regarding the participator subjects which can retrieve a specified person reversely, must not be recorded. In the telemonitoring scenario, the transmitted data contain the private information of the patients for medical treatment purposes. Hence, data privacy and security are more significant than in epidemiology. All communication channels and data in the air should be encrypted to protect the patients from malicious attacks.

Unfortunately, when examining through the history of WBSN development, it was found that so far there is no such systems serving for epidemiological purposes. Epidemiologists have to either adapt current WBSN monitoring systems or build new systems fitting their needs, which typically implies an enormous waste of efforts as well as money. Therefore, a key question is: since they share the same basic requirements for WBSN applications, is there a solution to enhance the flexibility, to ensure a customizable WBSN system to serve both scenarios, and easily shift between epidemiological and telemonitoring modes, as illustrated in Figure II-2?

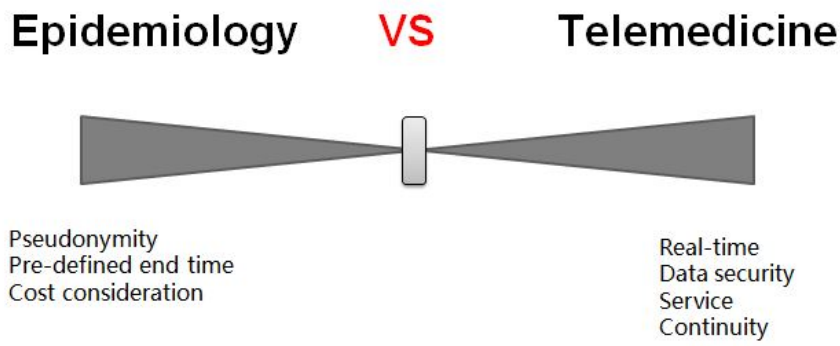


Figure II-2. Flexible WBSN customization between epidemiology and telemedicine

The goal of the PhD project is to design a generic WBSN data acquisition system based on open source and ICT standards, applying state of the art in software development. The system is supposed to fulfill both requirements in telemonitoring and epidemiological scenarios, and is customizable between two types of services. The novelty lies in its generic frame for further expansion suitable for certain application purposes, and its flexible switch between two different services. As to my knowledge, there is no such attempts in the WBSN domain, yet. The thesis aims to create a new degree of generity in WBSN design, for the benefits of both system developers and system users.

2. WBSN Theories and Relevant Techniques

In this section, the theories regarding WBSN are roughly introduced here, as well as relevant technologies that are necessary for building WBSN. Necessary and important technical terms mentioned here are referred to glossary list [2].

2.1. General architecture of WBSN with three layers

A typical architecture of WBSN is based on three layers which includes the interactions among sensors, the base station and the remote server. Figure II-1 depicts the typical architecture of three-layer WBSN systems in healthcare domain. However, also other types of WBSN architectures exist. Based on their application purposes, system designers usually adapt the three layers into other alternatives. Sometimes, when it is not necessary to conduct telemonitoring in real-time, the data streams are stored in the base station instead of being forwarded, in order to reduce costs [46]. However, when the patient is trapped in a small living area, like in the hospital where the base station does not really help, the base station may gets excluded from the architecture. For instance, holter ECGs used in the hospitals have no base stations for relaying data. Furthermore, there was multi-layer architecture proposed in France which took use of both home PC and PDA as base station to fit various scenarios [47]. Trinugroho [32] has recently proposed a four-layer architecture which integrates smart homes with wearable monitoring sensors and serves for various purposes via an independent information integration platform based on the normal three-layer scheme.

For the architecture with three layers, the main constitutions of each layer are characterized as follows.

The sensor nodes

In a WBSN system, on the base level there stay the biosensors, mono- or multifunctional, to measure various physiological vital signals. Sensor nodes sense, collect and process life signals. Pedometer, accelerometer and ECG are widely used sensors on the first layer. All the sensors should be equipped with a wireless transceiver unit, a mini-processor, and an energy supply device like battery. Miniaturization technology nowadays is a key in sensor design which allows the device to be small enough to contain several transistors on the same integrated circuit [25]. Usually wireless transceiver components using leading communication protocols such as Bluetooth and ZigBee by which monitoring data are sent out, are placed together on the circuit board. Since sometimes those sensors are programmable, there is often a small operating system particularly for sensor networks running on the processor of the sensor. TinyOS is an excellent representative of such operating systems which provides open source code for fast implementation on the biosensors due to their memory constraints. The biosensors then collect data periodically (for instance, at a sampling frequency of 50 Hz) or continuously and transmit either raw data or preliminarily processed data wirelessly in real-time to a base station where those data will be further

[2] All results, tables, figures, descriptions and discussions, are copied from the publication by Chen, et al., "A Review of Three-Layer Wireless Body Sensor Network Systems in Healthcare for Continuous Monitoring" [3]

processed. Most of the sensors nowadays have storage capacity although their embedded memory vary from 16 MB to 2 GB or even larger. Once their memory fills [6], they may transmit data to the base station at the same time of storing the vital data locally for future use.

The base station

A base station is a local processing unit in charge of data forwarding, data processing and analysis and giving feedback to patients. PDAs, smartphones or laptops are the most common equipments used as base stations in WBSN. Nevertheless some sensor research organizations and companies on the market prefer to develop their own base stations for their innovative sensors, e.g. the e-AR sensor developed by London Imperial College [1]. Principally, the base stations are portable devices, for releasing the patients from bounding to their houses as well as providing them real-time feedback. There is no need for patients to worry about going out of wireless transmission range and breaking the connection between sensors and the base station. Different platforms can be chosen to run on the base station, for instance for smartphones, Android [48, 49] is a good option because of its compatibility and extendibility to many monitoring applications. Other alternatives like Microsoft Windows Mobile Pocket PC platform [50] and the newly released Window Phone 8, are sometimes also preferred under certain development circumstances, e.g. when considering security or compatibility. Mobile network is a kind of convenient resource for data transmission tasks, so GPRS/3G are all preferences for WBSN developers. When the base station receives the incoming data, it reacts in three steps:

1. collecting and storing the received data;
2. streaming data to a remote central server via wireless LAN or GPRS/3G networks [26];
3. analyzing the data in order to provide feedback services to the patients or any predefined destination, by e.g. SMS or initializing a telephone call [51, 52].

In some cases the base station is more intelligent to be used as a personal medical guidance [53]. As a relaying unit, the base station distinguishes monitoring locations if the patients staying at home or outside, and then executes different tasks correspondingly. Its operation modes allow the application to switch between indoor and outdoor environments and make decisions on taking out either a periodical transmission of routine physiological data, or immediate transmission of emergency data [6]. The most important concern in task-shifting is to select the best routing path. Figure II-3 shows a proposal of typical routing selection [6]. When the patient stays at home, the base station forwards data via home gateway through WLAN to remote server; however in the scenario of outdoor activities, the base station becomes less active and transmits data at a lower frequency via GPRS. To realize long-distance wireless transmission, some nearby relay stations in the covering range of available wireless networks may be involved.

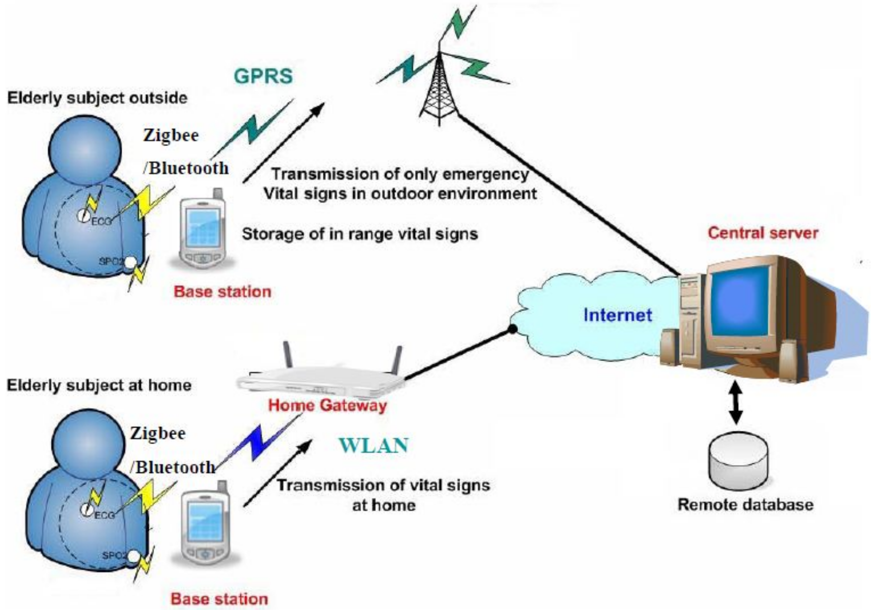


Figure II-3. Routing of the base station in an elderly healthcare monitoring system architecture [3]

To reduce the cost of wireless streaming, in most cases the base station will store the in-range vital signs locally, and initiate uploads when the patient returns to the home network coverage [51]. In case an abnormal condition is triggered, it uploads the emergency data immediately to the remote central server over GPRS/3G networks. The thresholds for emergency data can be predefined according to the advices from the healthcare providers.

The intelligence of base stations depends on the monitoring requirements and individual implementation. In summary, portable hardware devices, processor software platforms and transmission channels constitute the second layer of WBSN infrastructure.

The central server

To complete the WBSN model, a top layer comes up with a remote server, where the physiological information of the monitored patients is stored and deeply examined by health care professionals. The main task of the server is to store data into its database for further references. This ensures the doctors a possibility of long-term analysis, and keeps records of analysis results and patients personal data for diagnosis and healthcare instructions. Once any indication of life threatening to the patients has been found, corresponding departments and resources can get prepared in advance. The server mainly utilizes web applications and services, while taking advantage of Internet when communicating among service centers on the same layer. The three-layer WBSN is easy to scale and configure according to the concrete system requirements [51], so that this infrastructure can be applied to whatever healthcare settings: in-home, global or in-hospital monitoring systems.

[3] Figure reproduced from Saadaoui [6].

Basically, we can divide all monitoring systems using 3-layer architecture into three groups [54]: the first group contains the systems that do not really provide "real-time" services. This includes the systems doing some offline or postponed processing. The second group comes up with the demand of operating real-time monitoring but the remote server will do the processing [55, 56]. The third group performs the data analysis tasks on the base station and only if the emergency happens, the smartphone or PDA will contact the remote server for help [53].

2.2. Relevant ICT techniques

WBSN systems need not only the components in its three layers, but also various techniques to run smoothly. In this subsection, different technical components that constitute a WBSN system are introduced in more details.

2.2.a. Sensor types

Many WBSN studies nowadays focus on innovation of sensor hardware design [3, 57, 58], in order to improve the system performance. Types of the commonly used sensors which are capable of being integrated into sensor nodes are: pulse oximeter [6, 59], accelerometer [53, 60], electrocardiography (ECG) sensor [6, 60, 61], electroencephalography (EEG) sensor [62], body temperature sensor [61], blood pressure sensor [53], etc. For the convenience of the patients, wearable devices are preferable, as they are usually easier to initialize and maintain after being issued to the patients. Moreover, following the success of pacemaker, today implantable sensors also offer the most exciting component to WBSN [25]. But sometimes people keep their distance from implantable devices as those devices may scare people somehow. Ambient sensors contribute another important part to accelerate the wide spreading of pervasive sensing.

2.2.b. Software platforms

From the perspective of sensor nodes, TinyOS [63] is a popular platform to program on the sensor microprocessors [25, 64–66]. As it is an open source, the sensor developers can easily access to its repository for some existing components with minimized code size that share the same purposes for rapid deployment, such as Blink, Test Serial, etc., for different types of sensor motes. It is compatible with the programming language nesC, a variant of C, which ensures a straightforward understanding of the source code to any developer with the basic knowledge of C.

Software running on the central server is much simpler. Since the central server has various web applications, the most practical technology is XML. Besides, ASP.net and JSP also appear frequently with the presence of XML and HTML [50, 51, 53, 67]. For the integrated database that is used to permanently store the incoming data, usually some relational database servers like SQL server 2008 [58] are used.

What makes WBSN complicated is the design of the base station. For several concerns, developers have to choose a suitable communication protocol, an appropriate operating system as well as cost-effective telenetworks. Leading operating systems for mobile devices include Microsoft Windows Mobile [40, 50], Android [66, 68] and iOS [69]. Android is adopted in most of the studies for its openness, flexibility and compatibility during the life cycle of deployment. As for mobile communication, WBSN platforms normally take advantage of GSM (2G), GPRS (2.5G), UMTS (3G) or other wireless mobile networks [46, 51, 66, 70, 71] to realize data streaming from the base station towards the remote server. 4G mobile communication network is expected to provide worldwide seamless access to Internet and thus to guarantee the gathering of the real-time healthcare monitoring measurements from a remote location [29]. However the cost for these networks is considerably high, which postpones the wide acceptance of healthcare monitoring products by families.

2.2.c. Communication protocols

As found in a literature review [3], dominating wireless communication standards in WBSN domain currently focus on IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee) and IEEE 802.11b (WLAN). Some healthcare equipment use Bluetooth [72] which has utmost 100mW output power and ideal for indoor environment; while the typical output power of ZigBee [73] is around 1-2mW and suitable for equipment that requires long battery life, such as terminal equipment and communication nodes [74]. In general, ZigBee looks rather similar to Bluetooth but it is much simpler [37]. Several points have to be taken into account when applying them into medical care fields [75]. With the concerns of data traffic, coverage range, network architecture and power management, many studies claimed that IEEE 802.15.4 radio was the ever best choice for WBSN applications [6, 51, 66]. In principle, the standards IEEE 802.15.4, IEEE 802.15.1 and IEEE 802.11b offer respectively 0.25 Mb/s, 1 Mb/s and 11 Mb/s data rate. In the WBSN design, IEEE 802.15.4 is capable of covering personal area while the IEEE 802.11b covers the local area. Moreover, since ad-hoc network scheme is appreciated by most of the WBSN researchers and designers, IEEE 802.15.4 works appropriately for those applications with low data rate and low power requirements.

Table II.1 illustrates the specification comparison of the three aforementioned wireless communication standards. The parameters in the table indicate that although IEEE 802.15.1 and IEEE 802.11b offer high bandwidth, they are more expensive and power-consuming, and neither of them is suitable for wireless multi-hop networking [52]. Moreover, IEEE 802.15.1 needs 3s to wake up from sleep mode (IEEE 802.15.4 wakes up nodes from sleep mode within 15 ms). On contrary, IEEE 802.15.4 obtains the advantages of low cost, low power consumption, fast response, portability, unobtrusiveness, ease of deployment, scalability, almost real-time re-configurability and self-organization [51].

Nevertheless, from above it is clear that IEEE 802.15.1 and IEEE 802.11b which offer much higher data rate are good options for the applications requiring frequent burst data transmission. IEEE 802.15.1 is suitable for short-distance communication between lightweight electronic devices with relatively high data rate (compared with IEEE 802.15.4), as well as for building ad-hoc networks [37]. However, it sometimes has a limitation of communicating up to 8 nodes simultaneously in a piconet, but the fact is that most of the WBSN research studies are small-scaled projects including less than 5 sensor nodes together into the systems [53, 58]. The worst drawback of IEEE 802.15.1 concerning energy consumption compared to IEEE 802.15.4 has been solved by the release of new Bluetooth low energy V4.0 standard, which is now exploring its usage in sports and wellness, and healthcare

	IEEE 802.15.4 ZigBee	**IEEE 802.15.1 Bluetooth**	**IEEE 802.11b WLAN**
Frequency	2.4Ghz/915Mhz/868Mhz	2.4Ghz	2.4Ghz
Data rate	250K/40K/20Kb/s	1Mb/s	11Mb/s
Range	10-30m	10m,30m,100m	100m
Network Size	65535	8	32
Protocol complexity	Simple	High	Medium
Security	Authentic, encryption	Authentic, encryption	Authentic, encryption
Power consumption	60-70mW (Chipcon CC2420)	200mW (National LMX9820A)	400-700mW (Philips BGW200)

Table II.1. Comparison of wireless communication standards for WBSN

markets. With the appearance of the smart ready mobile phones using the latest Bluetooth technologies like iPhone 4S, the existence of ZigBee will be threatened to the biggest extent. In reality, Bluetooth overwhelms ZigBee by its high maturity level of which the compatible common interfaces are available on most of the mobile devices. Mobile phones, PDAs, laptops are almost equipped with Bluetooth components, which ensures quick and convenient deployment if using them as the base stations in WBSN. Unfortunately seldom of these devices have the integrated ZigBee components. From this perspective it makes sense to exploit further potentials of Bluetooth for healthcare monitoring services.

Other standards like UWB and XBee [46, 76, 77] have gradually come into WBSN developers view. Due to their unsophisticated technologies, most WBSN studies still tended to accept either Bluetooth [50] or ZigBee [70, 78, 79] according to the finding from the literature review of the author [3]. Other proposals of planning the development of communication standards optimized for BAN, e.g. the 802.15.6 IEEE Task Group [80], are still under development.

2.3. Challenges in WBSN design

WBSN technologies do not only bring new opportunities to healthcare monitoring. There are also though challenges rising with the ongoing development of WBSN systems. The first challenge comes along with sensor design. Many researchers pursue the systems which integrate more sensors on one node and consume less power [57]. Multi-parameter measurement is more efficient than mono-parameter measurement. It also requires higher reliability, robustness and accuracy than single sensor node.

Battery is another concern for sensor design. Principally the battery life of a sensor node depends on the communication protocol, the memory for data logging, and processor intelligence. The more tasks processors are able to handle simultaneously, the more energy they require. So power supply and power scavenging is a key topic within WBSN research fields [1]. The concept of body sensor network (BSN) itself has already contained the concern for self-powering or drawing energy from outside resources, such as fluorescent lights or sunlight via solar panels and even WiFi signals around. Recently Taylor et al. [81] proposed a new networking structure specifically designed to accept various energy sources, including wireless energy transmission, which appealed to any researcher devoting into energy aspect of WBSN. Recharging approaches are also something worth to consider for convenience.

PDAs and smartphones [66] are the most preferred handheld devices as base stations for WBSN developers. Principally mobile devices ensure the patients more mobility compared with laptops or home PCs which nevertheless have limited processing capacity as well. From the perspective of the elderly, it is still a big challenge for them to operate those modern electronic devices which are typically designed for young users without functional limitations.

Wang et al. [82] raised a new concern for WBSN in their paper. They claimed that resources should be reallocation through the cross-layer framework once crucial data were identified for protection purpose. For instance, once any emergency data are identified by the system, the system should switch off some unnecessary functions temporarily, in order to enhance the success of data transmission with the saved resource such as power and bandwidth. However, the feasibility and necessity remain to be further investigated.

As for the base station design, roaming is an issue when choosing appropriate communication protocols, in order to guarantee robust outdoor monitoring quality without interruption. Furthermore, security [83] and privacy [84], which are the main factors influencing user acceptance of the WBSN systems, are problems that have been explored for decades during WBSN development. Ethical aspect will be another important consideration in case one day this technology becomes the mainstream of telemonitoring [25].

Part II.

Sensor Communication and Analysis Platform (SensCoAP)

Chapter III

Analysis of SensCoAP

Contents

There are increasing needs for telemonitoring from remote locations and WBSN systems used for epidemiological studies. Unfortunately no effort has been made so far to merge them based on their similar architectures. The goal of the PhD project is to design a generic WBSN system for data acquisition and feedback that can be used in both telemonitoring and epidemiology scenarios, and customizable for the two types of services. The prototype is built on open healthcare and ICT standards applying state of the art in software development. It is called SensCoAP, short for "Sensor Communication and Analysis Platform". SensCoAP is constructed of three layer infrastructure considering the flexible extension for further use and the complete communication chain which provides possibilities for various application environments. The biosensors, the mobile devices (e.g. PDA, smartphones) and the server communicate with each other by different types of wireless communication standards. Besides the duplex communication between all the entities, necessary applications and friendly GUIs on smartphones and servers should also be provided to end users for device configuration and feedback. The data transmission and data storage must be as secure as possible. The server is supposed to run with high performance to ensure continuous monitoring. The entire development procedure obeys the rule of the optimized "waterfall" life cycle model in software design shown in Figure I-1.

The development of SensCoAP started with system requirement elicitation and analysis. This chapter first depicts the system functionalities by business processes. Both telemonitoring and epidemiological scenarios are involved. The methods and procedure of requirement analysis are introduced and the results of the analysis are presented in terms of UML diagramming and use cases.

1. Scenarios Illustrating the Functionalities of SensCoAP

In order to abstract the business processes of SensCoAP, some scenarios of daily monitoring routine using SensCoAP from the perspectives of patients and healthcare providers may help.

Scenario I: Home monitoring

Tobias Maurer is a 65 year-old COPD (chronic obstructive pulmonary disease) patient who lives alone. He has a relative, Kata Müller, living nearby. Tobias usually gets up at 6:30 am, puts on all the sensors he took off last night and immediately switches on his mobile phone. The mobile phone starts to detect all the sensors inside its range and displays a list of detected sensors. Tobias selects the sensors belonging to him and the mobile phone starts streaming the data from the sensors. A visualization shown on the display clearly indicates the system works properly.

The same early morning, Peter Schlet, a healthcare professional in the remote telemonitoring center responsible for Tobias, opens the browser on his laptop and connects to the website of the telemonitoring center. He logs in with his ID and password, and the system turns to the management homepage for medical staff. There is a name list of patients he is responsible for, including Tobias Maurer. He clicks on the name of Tobias, selects to view the accumulative update of data. The system provides him a monitor window in which several physiological curves depict the health status of Tobias today. The curves keep refreshing every 10 minutes. Peter is not sure of the normality of the signs and he wants more real-time data, so he configures the refreshing interval for the data of Tobias in the menu to 2 minutes. Now the curves move much faster. However he still has doubts, so he decides to do some further analysis to these data. He selected from the analysis menu the appropriate processing function, and after some seconds a detailed analysis report pops up showing all vital parameters of Tobias are uncritical. Peter saves the report and returns to the management webpage of Tobias.

Scenario II: Outdoor monitoring

Tobias Maurer plans to go for a short walk in the park near his home after lunch. When he leaves his home, the mobile phone searches for a new mobile connection automatically because the home network is not reachable now. Along with his further and further distance from his home, the mobile phone changes the mobile networks in use from time to time. Finally Tobias reaches the forest where no mobile signal is available in the range. The mobile phone stops sending real-time data and switches to the buffering mode, showing on the display how many data are buffered until now.

At the telemonitoring center, Peter notices that there are no more updates from Tobias. He clicks the name of Tobias Maurer in the list and there pops a message in the system that the subject is now out of range and in buffering mode. Peter then sets a reminder at 4pm to watch out the status of Tobias.

Tobias slowly walks towards his house. Once the mobile detects usable mobile networks (at 4:15pm) it starts transmitting buffered data immediately and turns back to streaming mode.

At 4:10 pm, Peter finds there is still no update from Tobias, so he decides to call Kata for help. He is about to pick up the phone suddenly he sees the streaming data from Tobias again. He looks at the buffered data carefully until he makes sure everything is in control for Tobias. Then he cancels his plan to contact Kata.

Scenario III: Emergency situation

Tobias Maurer gets up at 6am as usual but he does not feel very well. Peter Schlet in the remote monitoring center also notices abnormalities in the vital signals. So he calls Kata, asking if she was going to stay at home today, in case an emergency happens. At 10am the system alerts Peter that the health status of Tobias declined below the threshold during the last 10 minutes. He confirms the request of contacting the ambulance from the system, and calls Kata to go over to Tobias at once.

Scenario IV: Sensor measurements in a study cohort

Christian Nobel is a researcher working in a cohort center, where an epidemiological study is carrying on. He has to prepare the sensors and other devices before issuing them to the subjects in the cohort, by connecting the mobile phones with corresponding sensors. After pairing the sensors, he is also responsible for configuring all the parameters on the mobile phones to serve the study purposes. Every subject will get one set of devices and wear them according to the instructions of Christian. No more interaction is expected from the subjects except recharging the devices.

The four scenarios describe the routine process that SensCoAP is expected to fulfill. The business process for telemonitoring environment extracted from the first three scenarios is interpreted by a flow chart. From Figure III-1 it is clear that SensCoAP initializes itself with detecting wireless sensors nearby. Once any paired sensors are detected, it starts to build up the connection with them. The entire initialization finishes within a few minutes and involves little interaction with the patients. Afterwards, system distinguishes the monitoring environment as indoor or outdoor, in order to decide if real-time streaming should be conducted. When the patient stays at home where a stable Internet connection is available, the base station is supposed to maintain a continuous data relay, with the option of providing some feedback to the patient. The server where the physiological data are stored and processed alerts the corresponding healthcare givers in the medical center if any abnormal situation happens to the patient. In case the patient leaves his home, SensCoAP must searches for available mobile networks around. If there is some network connections, SensCoAP switches to outdoor mode and only stream data periodically according to the preset time interval. Otherwise, it has to buffer the data in the base station until the network connection is again in the range. Exception is some abnormal data are detected compared with the predefined thresholds on the base station. The alarms are triggered for the patient and SMS or phone call is made to contact the healthcare givers under life-risk situation. Those buffered data are uploaded immediately for diagnostic reference when networks are resumed.

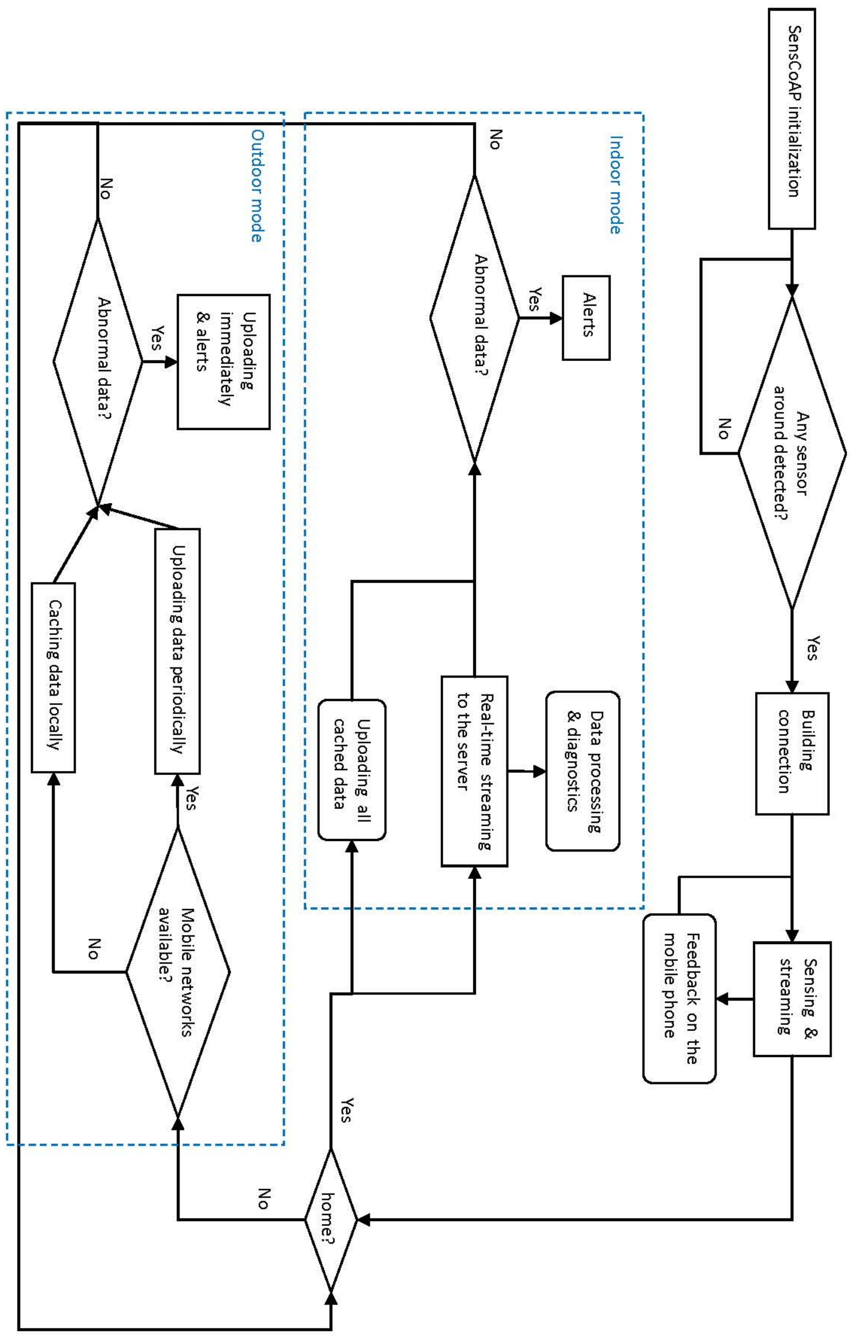


Figure III-1. Business process of SensCoAP for indoor and outdoor monitoring

The business process of SensCoAP for cohort measurement service (epidemiological or clinical observation studies) is much simpler, as shown in Figure III-2. As in an epidemiological study, normally no interaction with the subject is allowed. Hence, there are neither feedbacks nor alarms available to guarantee its invasive and unobstructive properties. As it is not supposed to recognize critical conditions, real-time streaming is not necessary in SensCoAP, either. Data are buffered for some time and uploaded periodically according to a preset interval, e.g. once every several hours. When there is no Internet access, data are buffered and wait for uploading once connections are available.

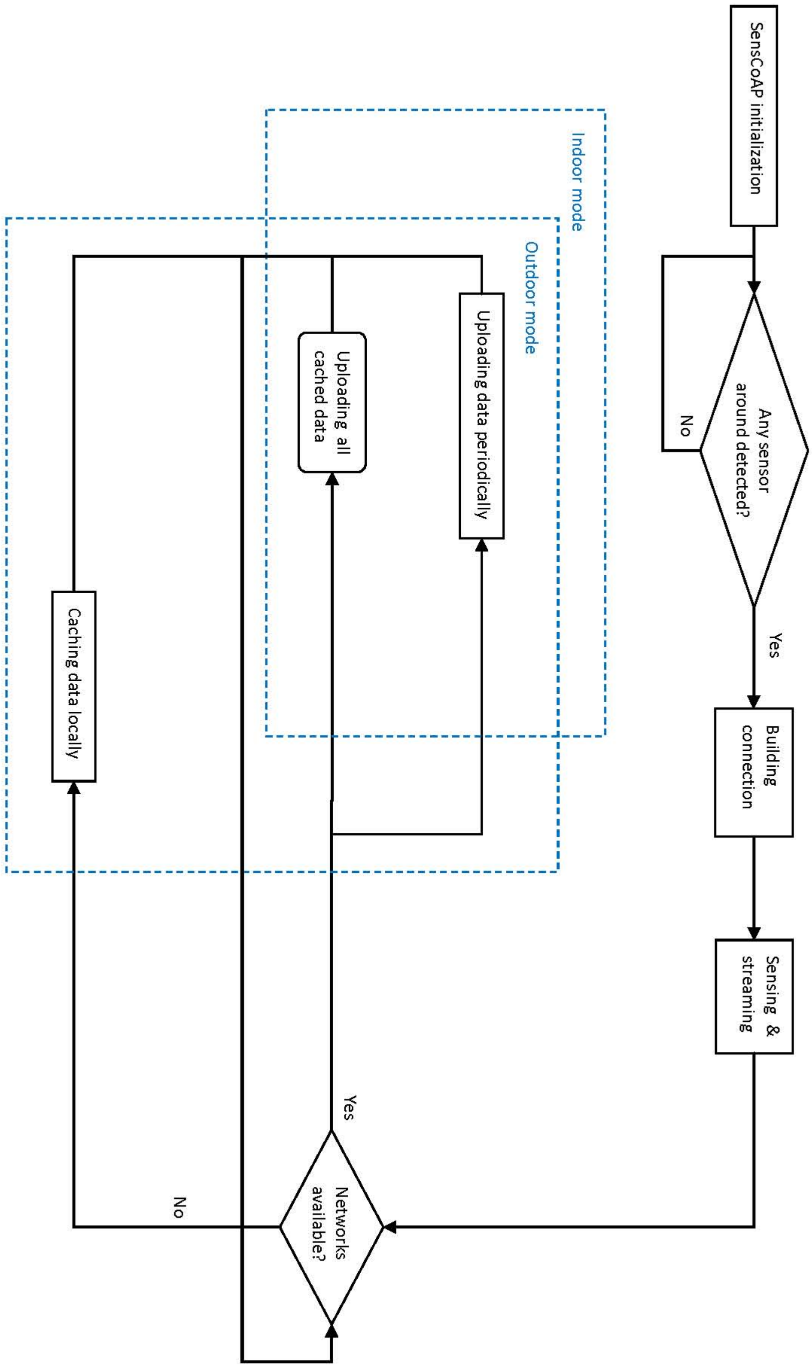


Figure III-2. Business process of SensCoAP for epidemiological and clinical observation study

2. Methods and Process of System Requirement Analysis

The entire system requirement analysis includes the phases collection, elicitation, classification and analysis of requirements. This section gives an elaborate overview of how the requirement analysis process was conducted. In order to finalize the list of system requirements, classic methods [3, 85] (see Subsection 2.2) were applied to gather all types of requirements, i.e. application-, system-, technology- or performance-specific requirements, in an exhaustive manner. All relevant details, e.g. methods that were used according to the 5 phases in the project plan, are summarized. Appendix A provides more details.

2.1. Preparation phase

The goal of the preparation phase is to set up the team and environment. Due to the limited funding and project time, the target team members have to be limited into three types of people: the end users, the technicians and the project manager (the author herself). The end users (physicians, doctors, subjects and patients) were found from the hospitals, and the informatics department of TU München can probably provide some experts on the sensor network. Because it is not an industrial product design project, there is no facility provider, manufactures or stakeholders involved into the team. The members mainly consist of:

- people majoring in medicine and working in the hospitals
- people working in the medicine institutes
- the collaborators of the biosensor group in Munich
- patients from the hospitals
- friends around as subjects
- researchers and informatics students (master and PhD) from the informatics department of TU München as technicians

Invitations were sent out in terms of poster in the institutes, individual phone calls, emails, face-to-face talks, etc.

2.2. Requirement collection phase

Participants and their roles

As a result of the previous phase, all together 13 people were involved into the requirement collection phase. Among them there were 2 doctors, 1 physician and 2 PhD students all of whom have experience with biosensors, epidemiological studies and monitoring in clinical trials and cohort. They were from hospitals and medical institutes. 3 patients with chronic diseases and their relatives who were familiar with the life style of the chronically ill patients were found in the hospital. Besides, 5 technicians with the background of sensor networks and computer science offered help from the informatics department of TU München.

Techniques

The collecting methods include:

- Brainstorming
- Interview open-ended [86]
- Storyboarding
- Scenarios [85]
- Walk-throughs
- Diagramming
- Potential problem analysis

Considering the limited time the team members could offer, the collection phase was condensed into several face-to-face individual interviews. Basically each member had an individual open-ended interview with the project manager. Other methods like brainstorming, scenarios (see Appendix A), walk-throughs and diagramming were also used as supporting tools. Questionnaires were dropped out because the long subjective paragraph composition definitely demotivated the participants and significant information got lost during this procedure. Nevertheless, all other techniques described above were used.

Acquisition sessions

In total 8 sessions were finally conducted including 7 individual interviews and 1 brainstorming meeting. Those sessions took place mostly at the working places of the participants e.g. IMSE, university clinic of LMU München, institute of epidemiology at the Helmholtz Zentrum München, and the informatics department of TU München in Garching campus. Each interview took approximately 40-90 minutes depending on the time schedule of each participant and the brainstorming meeting took even longer, nearly 2 hours. The project manager made records for each session. The entire procedure took 3 months, from end May 2011 until mid August 2011.

2.3. Requirement consolidation phase

The collected requirements (or their descriptions) from different aspects should be first summarized and classified within this phase. Decomposition of the proposed requirements is possible when necessary, to guarantee that each requirement statement only contains one single concern of the system. Similar requirements regard the same concern should be grouped according to the system requirement categories. A suggested concern list of the potential system requirements is (but not limited to) as follows:

- Compliance with wireless communication standards
- Security and safety
- Necessary functions

- System and application related requirements, e.g.
  a. Performance: latency, jitter, etc.
  b. Architecture: modular, extendable
- Economic and financial cost
- Other

According to the criteria, at first similar requirements were combined. Among those, requirements regarding e.g. necessary functions, technical key points, data security, and system performance were taken and analyzed. Other finance-oriented requirements were considered as out of the project scope because it is only a research project. However, these requirements were documented in the requirement specification as Section 4 “Out-of-Scope Requirements” (see Appendix A), as a supplementation to future development of this project when necessary.

2.4. Phase of requirement extraction from research

Besides the input from participants, the system analysis also benefited from the literature which aimed at the same scope as SensCoAP. From the literature review conducted by the author [3] on body sensor networks for healthcare monitoring involving the latest research results in 5 years, some performance-focused requirements were extracted, e.g. REQ-13, REQ-29 (see Appendix A). Technique constraint is also another concern when extracting compliance requirements from some international standards e.g. REQ-42. In addition, there were requirements e.g. REQ-6, based on the inputs from the internal group work and discussions.

2.5. Requirement elicitation phase

This phase was then decomposed into three steps:

Assessment of requirements

The requirements inherited from last phase were then examined by their importance, feasibility and other constraints. Unrealistic requirements were removed from the systems requirement list while documented for the reasons in requirement specification (see Appendix A Section 4 “Out-of-Scope Requirements”). Determined requirements went through next step.

Classification of requirements

Selected requirements were then classified into the following categories according to their feature definitions.

- Functional requirements
  Requirements that define the functionalities of the system, without which the system can not run. These requirements are usually the ones that the end users and stakeholders/sponsors care most about.

- Non-functional requirements

  a. *Operational requirements*
     Requirements that define the back-end functions of the system, which keep the system operational over time.
  b. *Technical requirements*
     Requirements that define the technical constraints or the conditions the system must fulfill for a good performance.
  c. *Administrative requirements*
     Requirements that define the features important for administration, including the budget, management, and support, etc.

As SensCoAP was decomposed into several subsystems, the requirements were documented according to the subsystem they belong to. Requirements for overall system were split into subsystem-specific requirements. The requirement descriptions follow the format of use case descriptions by defining the user, the flow and the interaction with the user. Thus principally the functional requirements were represented as the normal flow and the non-functional ones were represented as the exceptional cases in use case descriptions.

Prioritization of requirements

After documenting all necessary information, values of frequency (of this flow that will happen), risk (of the failure of this flow) and priority (of this requirement to be implemented) have been assigned to each requirement. The evaluation of frequency, risk and priority were agreed by the team based on the concerns that had been mentioned from the user perspective during the 8 acquisition sessions for each requirement.

3. Analysis Results

This section is mainly presenting the results of system requirement analysis, in terms of UML diagramming, use cases and relative requirements.

3.1. UML modeling

UML is helpful for the system diagramming [87]. The purpose of UML modeling is to build the model-driven architecture of the system functions [88] as much as possible by using state-of-art UML tools for platform-independent modeling tasks [89]. Results came up with the use case diagrams which were revised based on the diagramming used in the collection phase. The necessary functionalities were modeled and added into the diagrams. Some basic functions were also split into two detailed subsystems–the BAN and the BAAS (biosignal archiving and analysis server). Besides the general use cases providing a thorough overview of SensCoAP functionalities, the subsystems on the BAN as well as on the BAAS were also modeled. The modeling tool was Visual Paradigm Community Edition, version 5.2 [90], a powerful UML-based modeling tool which is available in a free edition for researchers.

Figure III-3 defines three types of roles as users of SensCoAP: subject (in telemonitoring environment it presents patients), healthcare professional (HCP) and system administrator. In general, the subject can use this system to monitor his own health status and also request feedback report from the server, for instance comparative activity level improvement. Once the physiological data collected from the subject are forwarded to the BAAS by the BAN hub (e.g. smartphone, PDA, etc.), the healthcare professional is able to conduct some analysis based on the datasets. At the same time when the data come in a "near real-time" way, the healthcare professional can also take the responsibility to monitor the subject, in case any emergency happens. The system administrator is in charge of maintaining the entire system and updating the plug-ins from time to time. The upgraded system is supposed to be reverse-compatible with all its old versions and never crash down due to those updates.

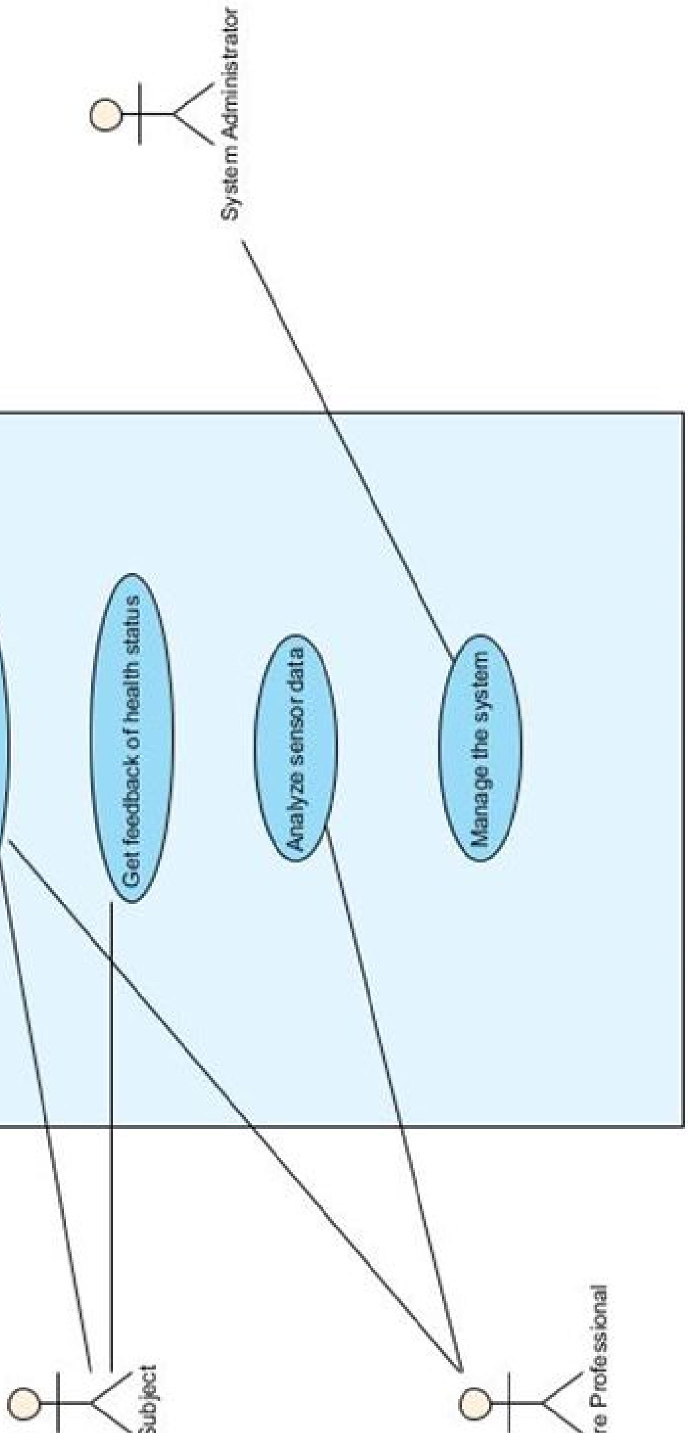


Figure III-3. General use cases of SensCoAP

On the BAN side as shown in Figure III-4, the healthcare professional prepares the devices which will be issued to his subject soon by means of starting the sensor detection through the BAN hub. The sensors should be paired with a specific BAN hub for each subject. The BAN hub is able to start the biosensors for measurement automatically, namely with few interactions. The collected data are visualized on the display of the BAN hub as an instant feedback to the subject. Besides, the subject can also require other kinds of feedback services through the BAN hub, which is internally authenticated before. The BAN hub connects automatically to the remote BAAS and synchronizes with it by a certain frequency. Only the healthcare professional is supposed to configure the BAN hub directly on it and contact it via the BAAS interface.

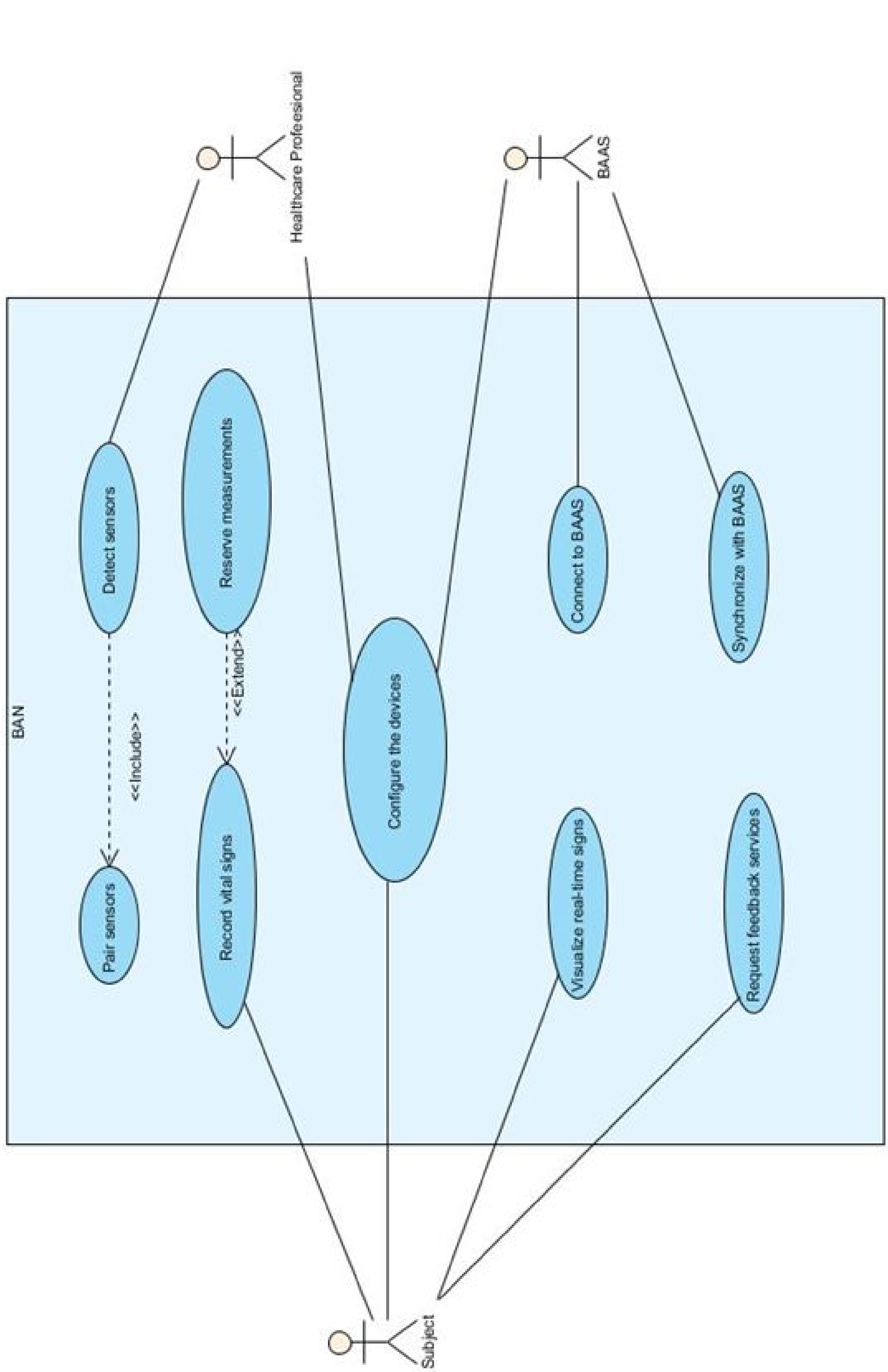


Figure III-4. Use cases on the BAN subsystem

As depicted in Figure III-5, anyone who wants to access to the BAAS must log in with his user name and password due to the security concerns. The BAAS distinguishes the role of the user and leads to the respective webpage. Principally the subject can only use the BAAS to request the feedback services. The healthcare professional uses this subsystem to manage his responsible subject groups, as well as analyze vital data from the subject by the integrated statistical plug-ins. The system administrator is in charge of managing system users, maintaining the entire system and updating the plug-ins periodically.

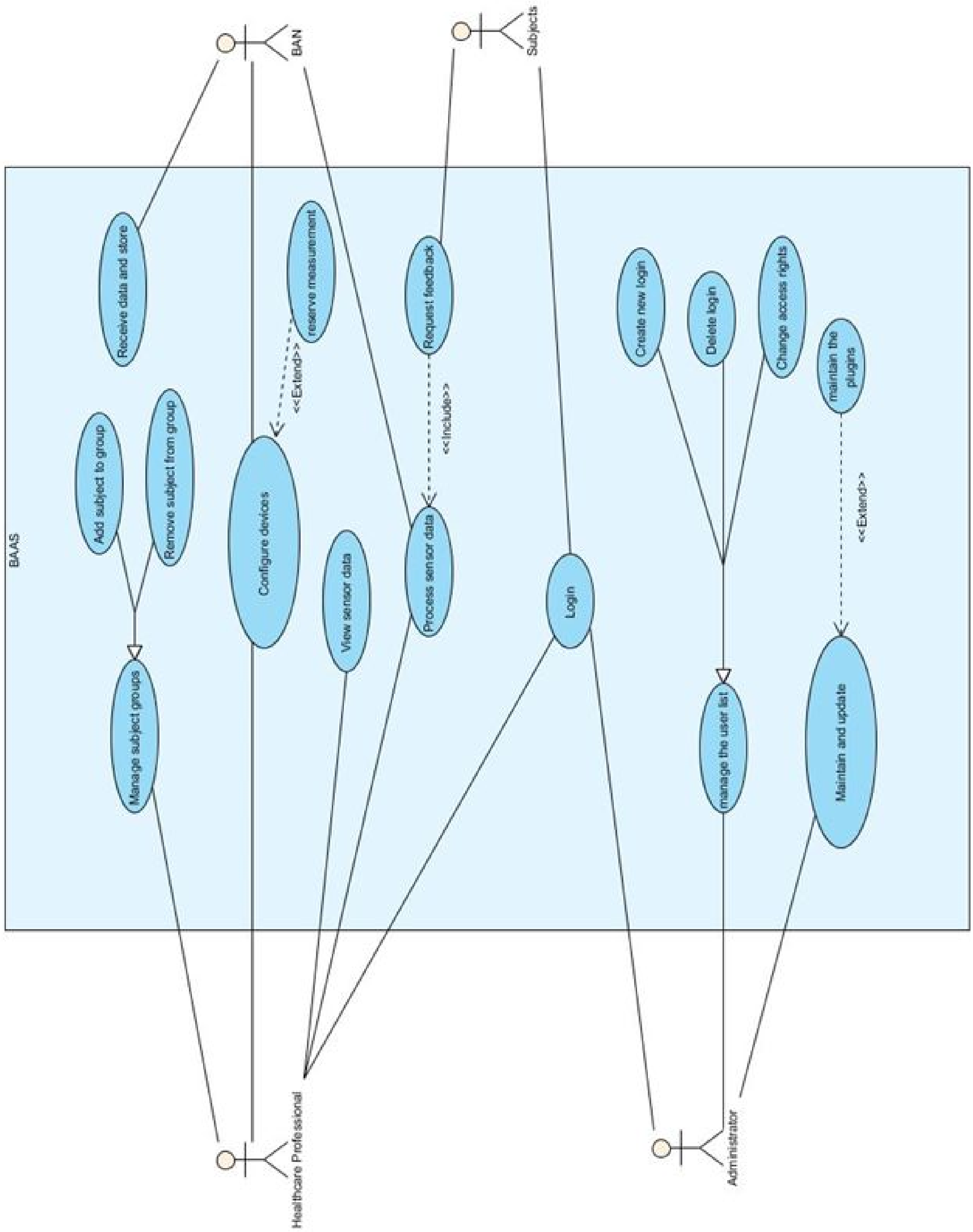


Figure III-5. Use cases on the BAAS subsystem

3.2. System requirements

In this section, an overview of the finalized requirements for SensCoAP prototype are described (all the necessarily functional and non-functional requirements can be found in Appendix A). As quality control, there are some guidelines accepted in common for requirement validation before officially documenting according to Loucopoulos et al [85]:

- Internal consistency
- Non-ambiguity
- External consistency
- Minimality
- Completeness
- Redundancy

Validation of the elicited requirements was conducted based on the rules before documenting them officially in the system requirement specification (see Appendix A). Requirements against those rules have been examined and evaluated again by the team, to decide either to adapt them, or simply abandon as out-of-scope objects. Considering the special multi-layer structure of SensCoAP, the professional consulting style [91] was taken to specify all the requirements. Instead of separating the requirements into functional and non-functional parts as usual, the requirements were organized as groups according to the physical partition of the system, e.g. to which systems the requirement applies. The specification was based on the use case diagrams mentioned above and merged with use case descriptions [92]. The exceptional path of a specific function (usually indicating a non-functional requirement) was pointed out especially for certain functions (e.g. REQ-21 in Appendix A). Following the description a value of priority was issued to the requirement considering its frequency and risk to the entire system.

As a result, a total of 44 requirements were documented, 31 of which were system-related. 13 requirements were marked as out of scope according to the evaluation criteria. The 31 system requirements were divided into three groups: biosensor-, BAN- or BAAS-related. Among them, 4 requirements were concerning biosensor design, 12 requirements were concerning the BAN and 15 requirements were applied on the BAAS. For a detailed list of requirements and corresponding use case description, please refer the Section 3 "Specific Requirements" of the system requirement specification in Appendix A.

Chapter IV

Design of SensCoAP

Contents

Based on a high-level and detailed requirement specification in Chapter III developed with collaboration of those potential users, this chapter main deals with the architecture design of SensCoAP on the system- and subsystem-level and states the design concepts behind. Not only all the mandatory and optional components of the system are modeled with state-of-the-art modeling methodologies, but also the associations, interactions and dependencies between all the system components are described. Besides the system requirement specification, this system design also has dependence on system design document standards [93] and part of student work [94, 95].

According to the major requirements REQs 5-16, 17-31 (the main functional requirements of the BAN and BAAS) and the use cases, different models of the BAN and the BAAS subsystems are partially developed by the model-driven development (MDD) methods. The design presents all mandatory and optional elements as generic as possible, in terms of UML diagrams such as component diagrams and sequence diagrams, to demonstrate the functions of the subsystems and the interactions between the system components, even though the following implementation will only cover part of them due to the limited time. Moreover, the design of the database and the external user interface (UI) is presented by entity-relationship diagram (ERD) and concur task tree (CTT). Interfaces between the SensCoAP components are defined by existing communication protocols, e.g. HTTP. There are explicit descriptions attached with all the components of the models in the corresponding sections.

The result following the system design will be an implemented demonstrator which represents a live scenario of how to deliver telemonitoring/study service via wireless technologies and communication protocols. The proposed SensCoAP prototype will be constructed into three layer infrastructure. Biosensors, mobile devices (e.g. PDA, smartphones) and the server communicate with each other by different types of wireless communication standards. Besides the duplex communication between all the entities, necessary applications and friendly graphical user interfaces (GUIs) on mobile devices and the servers should also be provided to end users for device configuration and management. The data transmission and data storage must be as secure as possible. The server runs with high performance to ensure continuous monitoring. The fully developed system offers a broad space for further extensions.

1. Modeling Methods

Before going into the design, this section briefly summarizes the state-of-the-art modeling methods that have been used to design the SensCoAP system: MVC, MDD, ERD, and CTT.

Model-View-Controller (MVC)

MVC is a well-known design pattern for medium-scaled project. It consists of model, view and controller representing data, presentation and logic respectively. Model is used to encapsulate all related data and the data processing methods. View level is to intended display data, while controller is to organize and control the flow between different levels, and respond to events [96]. MVC pattern makes the software more robust and the code reusable. It is especially useful for web application development. Now many platforms such as Java and .NET support this design pattern. MVC is used in the design of both two subsystems in Sections 3 and 4.

Model-Driven Development (MDD)

In MDD, applications are transformed to a higher level of abstract model. MDD uses models systematically as main components in the entire software development life cycle, thus to reduce the complexity of the software, and make it more flexible to extend [97].

As MDD abstracts everything into a higher level, it should convert the meta-model into specific model or to code later on in the development. The advantages of MDD are: it not only accelerates the speed of the development, but also is not error-prone and easy to maintain the architecture. From this point, it suits fast changes of business requirements, which happens quite often in industrial projects. As MDD is able to generate code from models, to some extent it keeps consistency among components by automatic software development.

MDD can be combined with UML 2.0. It is realized by the definition in UML profile. Many released open source tools support MDA (Model-Driven Architecture) modeling. The platform used to present models and MDA in this thesis is StarUML 5.0.2, a platform for both UML and MDA [98].

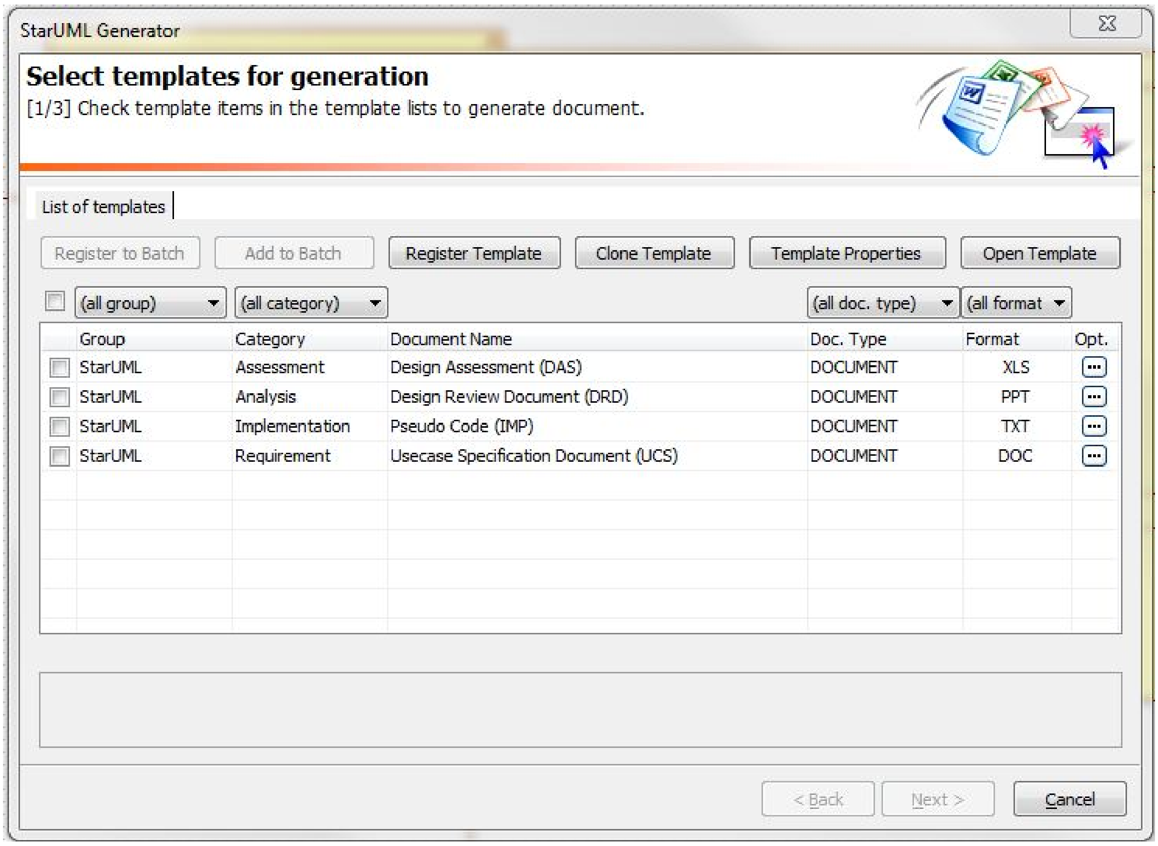


Figure IV-1. StarUML generator

Figure IV-1 shows the generator of the StarUML to support other procedures except modeling in the software development life cycle as a MDA platform. Depending on the UML models this software is able to automatically generate supporting documents to support e.g. system requirements, system analysis and pseudo code for implementation. The user can define his preferred programming language in the UML profile and thus there are tools that will convert the UML models to the code in the desired language. For example, each component in the conceptual model will be transformed into a Java class if Java profile is applied to the conversion. From this perspective, MDD is quite helpful to shorten the life cycle of the software development and keep consistency by mutual support between each stage.

Due to its complexity, there are few successful cases applying MDD into projects until now. MDD is best suited for large scale software development projects. Since SensCoAP is a small project, MDD has only partially been used. Precisely speaking, in the following sections:

1) UML 2.0 will be utilized for modeling.

2) the models will only be used further to generate pseudo code, in order to provide reference for the implementation work.

Moreover, because the prototype is implemented in parallel with the system design due to the limited time, the biggest advantage of using MDD, the automatic generation of the real code, turns out to be not really necessary in SensCoAP project.

Entity-Relationship Model (E-R Model)

E-R model provides a conceptual method to describe the entities, attributes and their relationships. It is usually used for high level data model. An entity-relationship diagram (ERD) is the graphical representation of the model which mainly applies to the organization of the data within database and information systems. It explicitly explains which data are stored in the system (entity) and how the system shares the data (relationship). ERD is used to describe the data model of the SensCoAP system in Section 5.

Concur Task Tree (CTT)

CTT is commonly used for user interface design [99]. It is represented by a tree, where each node represents one of four task categories: abstract, application, user and interaction. As CTT is usually task-based [100], it is also used as a background for usability testing [101]. CTT uses XML schema for an XML serialization of CTT models and is able to achieve visualization of CTT notation as well [102]. CTT is used to model the UI parts of the SensCoAP system in Section 6.

2. System Overview

This chapter provides a rough overview of the entire system architecture. Generally speaking, the design goal is to represent a generic system with highly flexibility and compatibility for further development. Therefore a modularized approach is chosen. Besides, considering the involved personal data and privacy, a secure communication channel should be guaranteed for its users.

2.1. Design decisions

The SensCoAP system is designed according to the requirements and use cases in Chapter III. It is split into two subsystems: the subsystems running on the BAN Hub (e.g. smartphone) and the BAAS respectively. The design of the BAN Hub subsystem exactly meets the REQs 5-16 in the system requirement specification (see Appendix A), while REQs 17-31 are fulfilled by the BAAS subsystem. BAN REQs 43-44 are not reflected in the system design as they are out of the scope of the SensCoAP system, as described in Appendix A. Design decisions are generally made based on the corresponding requirements. For example, a user-access control scheme is involved in the design to bind the smartphone and sensor devices with specific credentials, in order to reduce interactions with the subjects as much as possible (REQ-3, 13). According to REQ-16 the HTTPS is used and the emergency alarms are designed based on REQ-14. Some requirements have to be discarded due to the technique or condition limitations, such as the reliability and battery life of the sensor (REQs 32-33) as they are mainly affecting the hardware design rather than the software, and the hardware design is out of the project scope.

Sections 3 and 4 will provide more details on the consistence of the requirements and the design models.

2.2. General architecture

In Figure IV-2, the general conceptual model of the entire SensCoAP system is depicted. The system consists of two parts: the BAN and the BAAS. An Android smartphone is used as the BAN Hub which is in charge of receiving data, executing preliminary processing, interacting with the subject, and forwarding data to the BAAS server.

In the BAN Hub, the most important part is the *DataManager* component. Once data are recorded by the *RecordingController*, the *DataManager* immediately caches the datasets temporarily. Meanwhile, according to the selection of the user, the BAN Hub is able to process the data preliminarily. Processed data are presented in terms of visualization, graphs or diagrams as to motivate the subjects. Also healthcare professionals can utilize the results to trigger alarms under certain circumstances, by setting thresholds of abnormal data in advance.

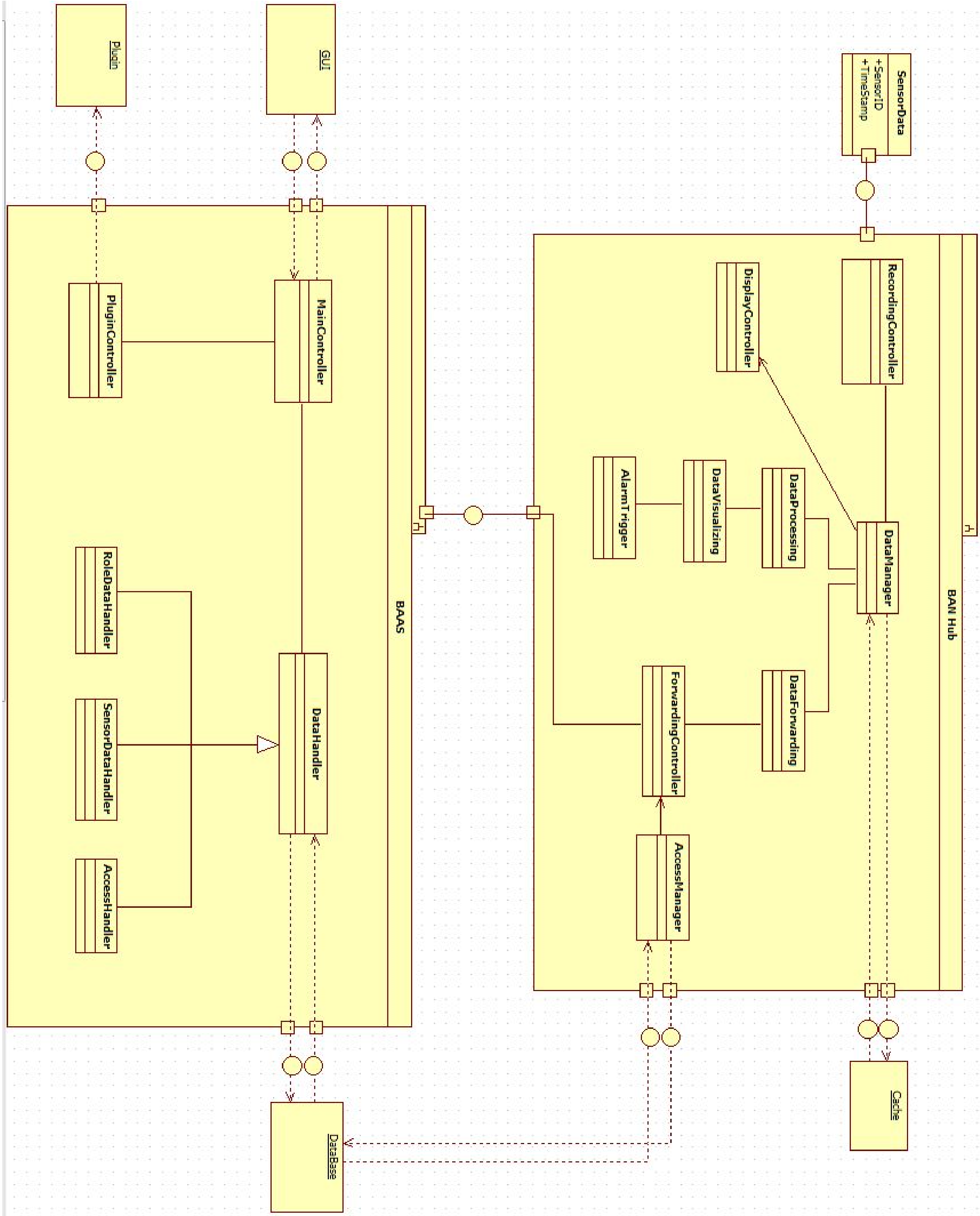


Figure IV-2. General conceptual model of the SensCoAP system

The *ForwardingController* is a component for communication between the BAN Hub and the BAAS. It initializes a request to the server. After getting response it will build up a secure channel to transmit the data.

The BAAS locates in distance from the subjects and provides services to healthcare professionals. Data sent by the BAN Hub will be permanently stored here. Meanwhile, healthcare professionals have the rights to do some data processing and keep track of the health status of certain subjects.

In BAAS, the *MainController* is a black box where all the functionalities of the BAAS are realized, thus it is also the most complicated component in the system. Different types of data, such as personal information, sensor information and raw sensor datasets, are handled by the abstract class *DataHandler*, which has an interface communicating with an external database. A *UI* component interacts with the BAAS for input and output.

The entire model is based on the Model-View-Controller (MVC) pattern, which clearly separates the presentation, logic and data. This makes the system easy-to-maintain and reusable [95]. Besides, both the BAN Hub and the BAAS are guaranteed by an access control scheme, to ensure the safe transmission of the physiological data.

3. The BAN Subsystem

In this chapter, the details of the BAN subsystem are presented by UML diagrams. Each component in the subsystem is described exclusively.

3.1. Design overview

The BAN Hub app running on an Android smartphone is mainly responsible for receiving data, forwarding data and interacting with the subjects, according to the use cases on the BAN subsystem and REQs 9 and 10 (see Appendix III). Figure IV-3 depicts the functional components of the BAN Hub with UML 2.0 component diagram [94].

Each component in the diagram can be treated as a black box and it offers corresponding functionalities and interfaces to the other components. The design of the BAN subsystem follows the rules of MVC pattern. A detailed description of each component is listed below.

SensorController This is the component in charge of building up wireless communication with the biosensors. Depending on the sensor type different communication protocols might be enabled such as Bluetooth [72] and ANT [103]. It distinguishes sensors by something similar to MAC address and the next component *ProcessingController* can access the received sensor data via its interface. To handle multiple types of sensors, two solutions are available from the design: either one *SensorController* handles all sensors which makes the structure of the *SensorController* more complicated, or multiple instances for the abstract *SensorController* must be created and each *SensorController* instance only has to deal with one specific type of sensors. However, the following *ProcessingController* has to recognize the sensor-specific components and in this case the complexity of *ProcessingController* will increase inevitably. This component has capsulated all sensor specific details. Once new types of sensors are amended this is the only part to be changed, which highly ensures the modularity and flexibility of this system.

ProcessingController This is the main component of the BAN subsystem. It contains the logic of starting/stopping sensor recording, caching data, uploading data and processing data for interactions. It coordinates almost all the other components in the subsystem by interfaces. After reading commands from the *UIController*, it starts/stops the connection with the biosensors correspondingly via the abstract interface to the *SensorController*. It collaborates with a caching system to store data temporarily before uploading them to the BAAS. The mechanism is able to deal with the situation when connection to the BAAS is lost. *ProcessingController* synchronizes with the BAAS according to the presetting time interval and uploads cached data actively through the *SyncController*. Meanwhile, it executes some preliminary data processing e.g. visualization to motivate the subjects. Interactions like alarm scheme (alarm is triggered when abnormal data exceeds the thresholds and interaction follows) should also be available to the users. The events of interactions coming from the users are caught by the *UIController* and it refreshes the visible display to inform the users at the same time as it goes through the exposed interface.

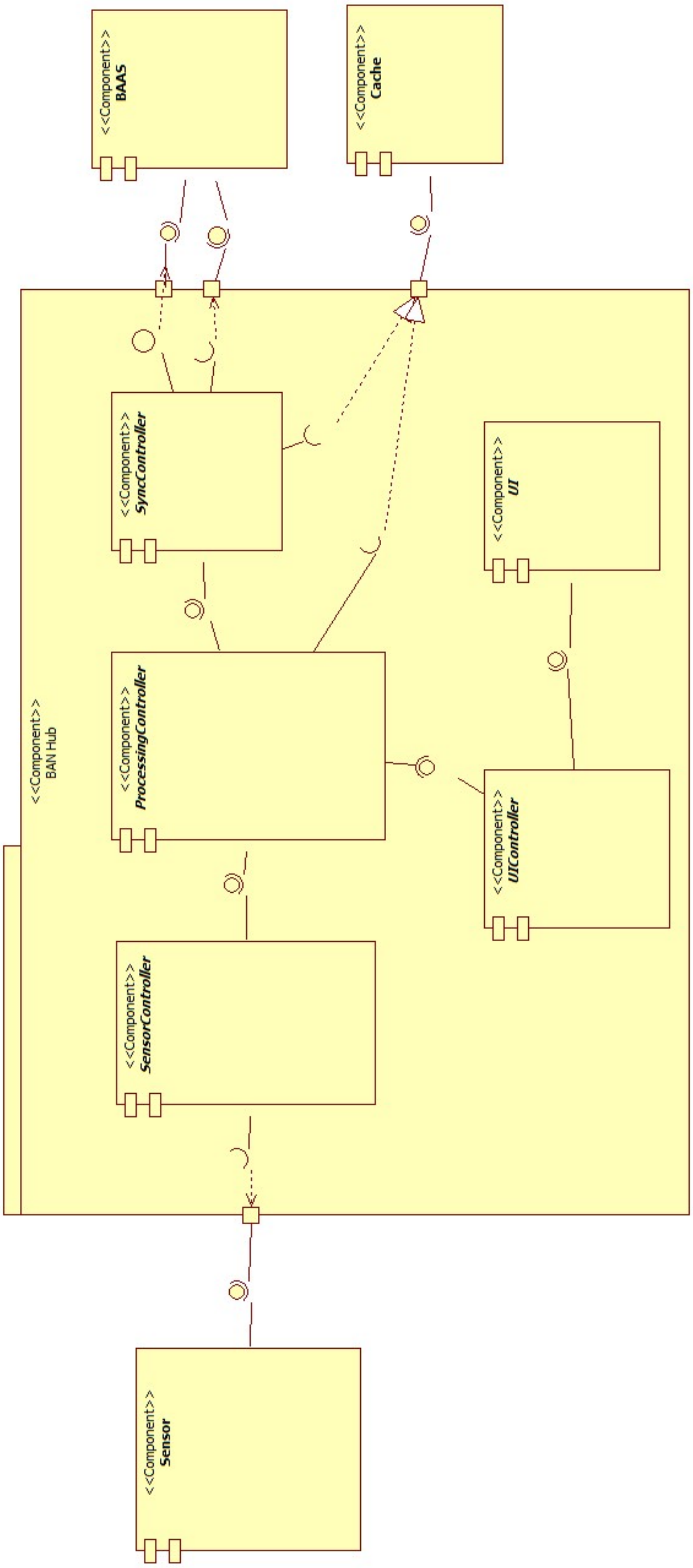


Figure IV-3. UML 2.0 component diagram of the BAN Hub

SyncController This is the part that is responsible for uploading the datasets to the BAAS. The function on the interface is quite simple, only sending a chunk of data stream. It grabs the cached data from the caching system only under the command from the *ProcessingController*. After uploading it provides feedback message to *ProcessingController* regarding the success transmission of the data, on which the *ProcessingController* adjusts the interval of synchronization accordingly. In case the BAAS interface is replaced by other component, the *SyncController* is the only part to be modified, which increases the maintainability and reduces the complexity of the system.

UIController It handles the input from the users and the output of the *UI*. Once it detects a tangible input from the *UI* it processes the input and issues commands to the *ProcessingController* which offers an appropriate interface to it. It is also responsible for building up the *UI* and updating the output.

UI This is the View component in terms of MVC pattern. It offers some basic functionality and deals with the external interface of the BAN Hub. Its internal interface is used by the *UIController* for exchanging the input/output information. In this design, this component is assumed to be a primitive View with a hierarchy of abstract View, at the same time to be a container for other abstract Views.

3.2. Interactions between the BAN components

An interaction example is presented here to demonstrate how those system components cooperate with each other. It describes the activities that happens from a user starts sensor recording until the data is successful uploaded to the BAAS and the corresponding data is deleted from the caching system. This scenario is depicted with the help of UML 2.0 sequence diagram as Figure IV-4 [94].

This event starts with a click on the smartphone by the user who want to start a measurement of a specific sensor. The request is caught by the *UIController* and is forwarded to the *ProcessingController*. The *ProcessingController* then issues a command to the *SensorController* to start data recording on that sensor. The sensor starts to continuously generate data and stream them to the *SensorController*. The received data are passed to the central controller *ProcessingController* by the *SensorController*. To store the data the *ProcessingController* is communicating with the Cache waiting for synchronization. Once a sync interval is timeout, the *ProcessingController* asks the *SyncController* to start uploading data to the BAAS. The data are then taken out from the Cache and transmitted to the BAAS with an acknowledge indicating the success. The acknowledgement is passed towards the *ProcessingController* and it will delete the corresponding data chunk from the Cache.

3.3. Design constraints and considerations

The entire BAN subsystem will be built into an Android app so the required hardware is a smartphone with touch screen and Android system (version 2.3 or above) installed. This smartphone should integrate the wireless communication modules e.g. Bluetooth. The app should be running all the time so it is necessary to avoid the Android system to kill the app after a period of inactive time.

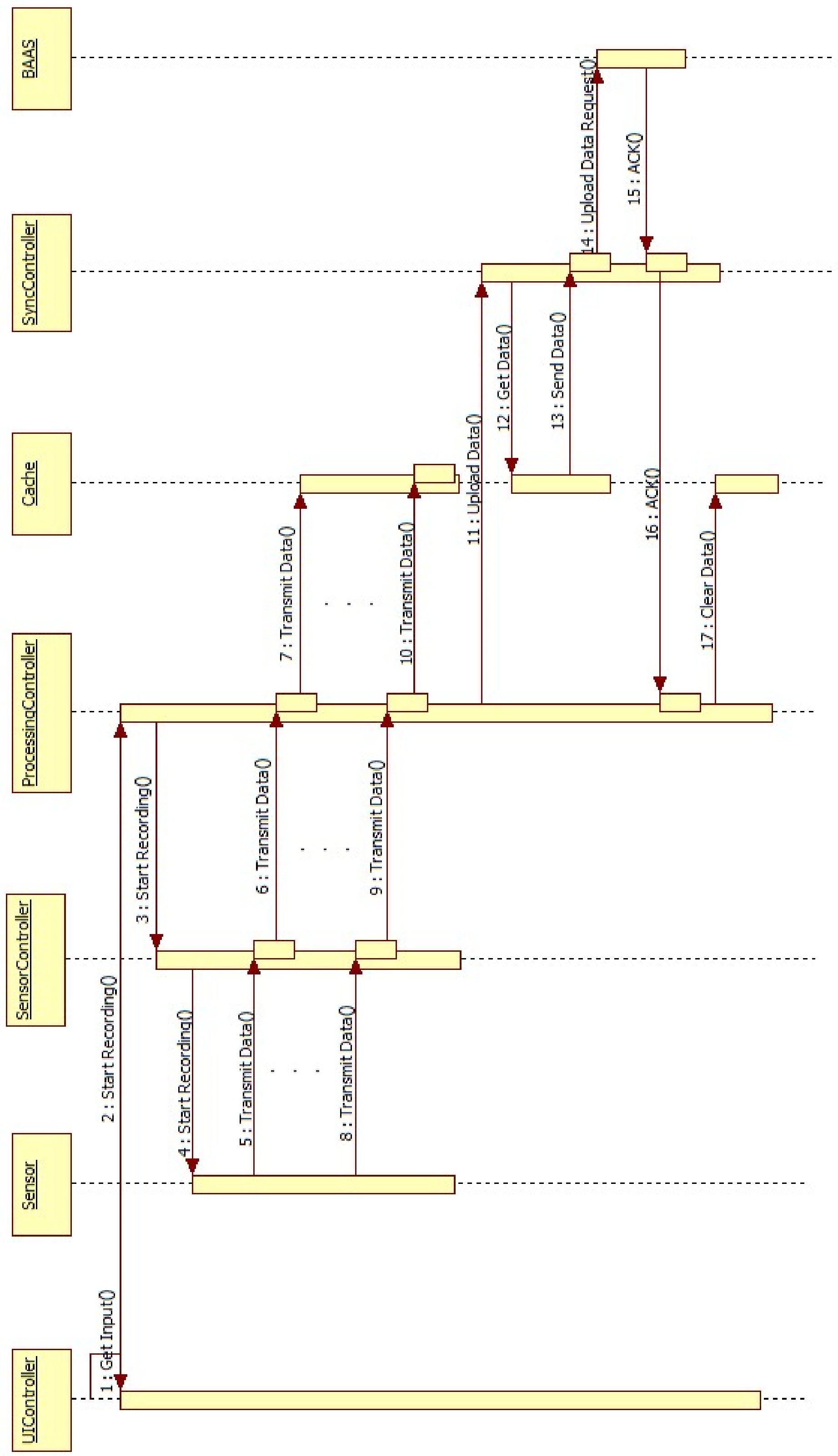


Figure IV-4. UML 2.0 sequence diagram of uploading cycle on the BAN Hub

As the capacity of connection in short distance might be limited, there should also be a maximal number of sensors that can connect with the BAN Hub simultaneously.

To switch between epidemiological study and telemedicine service, not only the synchronization interval can be adjusted according to the application aims, but also an interaction part is reserved for those patients when doing interventions. Interaction function can provide the patients some real-time feedback and is also supposed to detect life-threatening emergency. While it is used for epidemiology, the interaction function can be deactivated and the users can ignore it, so that the subjects will never get disturbed by the system.

4. The BAAS Subsystem

This chapter mainly describes the details of the design for the BAAS subsystem. The architecture will be presented on a component basis. As in Section 3, the BAAS design is also be divided into functional components and their interfaces in between.

4.1. Design overview

The main functions of the BAAS are to store data permanently, provide services to both healthcare professionals and subjects, maintain devices and accounts, and upgrade itself from time to time, corresponding to REQs 17, 19-23, 25 and 29 in Appendix III. It distinguishes the access of administrators from healthcare professionals and subjects. The healthcare professionals can use this system to monitor the subjects and analyze the data, while the subjects will be redirected to a read-only webpage for information (REQ-24). A UML 2.0 component diagram of the BAAS is shown in Figure IV-5.

The BAAS design also follows the MVC pattern as the BAN Hub does. Each component in Figure IV-5 will be interpreted below with its functions and interfaces.

DataModel It is the basic component of the entire BAAS. It encapsulates all specific details related to the *Database* and abstracts the access to other internal components of the system. It communicated with the external *Database* through a SQL-based interface while it offers interfaces to the *MainController* in a specific application programming language. It might contain some business logic e.g. manage the user profiles and create/delete users. Besides, it provides data access policy (REQ-24) acting as a gatekeeper to assure secure access.

MainController This is the place where user access control is actually realized. It is the most complex component in the whole system. It coordinates and mediates between the other components and incorporates most of the application logic [94]. It offers interfaces for requests from external components as the *BAN Hub* and the *UI*. According to the requests the *MainController* responses and acts to e.g. check the permission or read data from the *Datamodel*. The results are sent back to the clients and may be rendered on the corresponding *View* component. To process data, it has to cooperate with the *PluginController* as well to achieve data analysis results.

View It is the View of the system in terms of MVC. It has interface to input data which is then rendered or prepared in a view specific way to be sent to the client by the *MainController* [94]. The *View* utilizes the exposed specific interfaces of the client to display webpages or information formatted in a specific way (e.g. in JSON). The ability of the *View* component depends on the approach of the server implementation, leading to a more specific interface of the *View*, which will be discussed later in Subsection 4.3. It is possible to easily add new *Views* to the system due to the MVC pattern. Meanwhile, it is also flexible to expend the client types to the BAAS in future.

PluginController Plugins can be developed for the system specifically according to the aim of use. This part is then handling the bridge work between plugins and the *MainController*. Its interface is defined in the plugins; while it gets information and uses

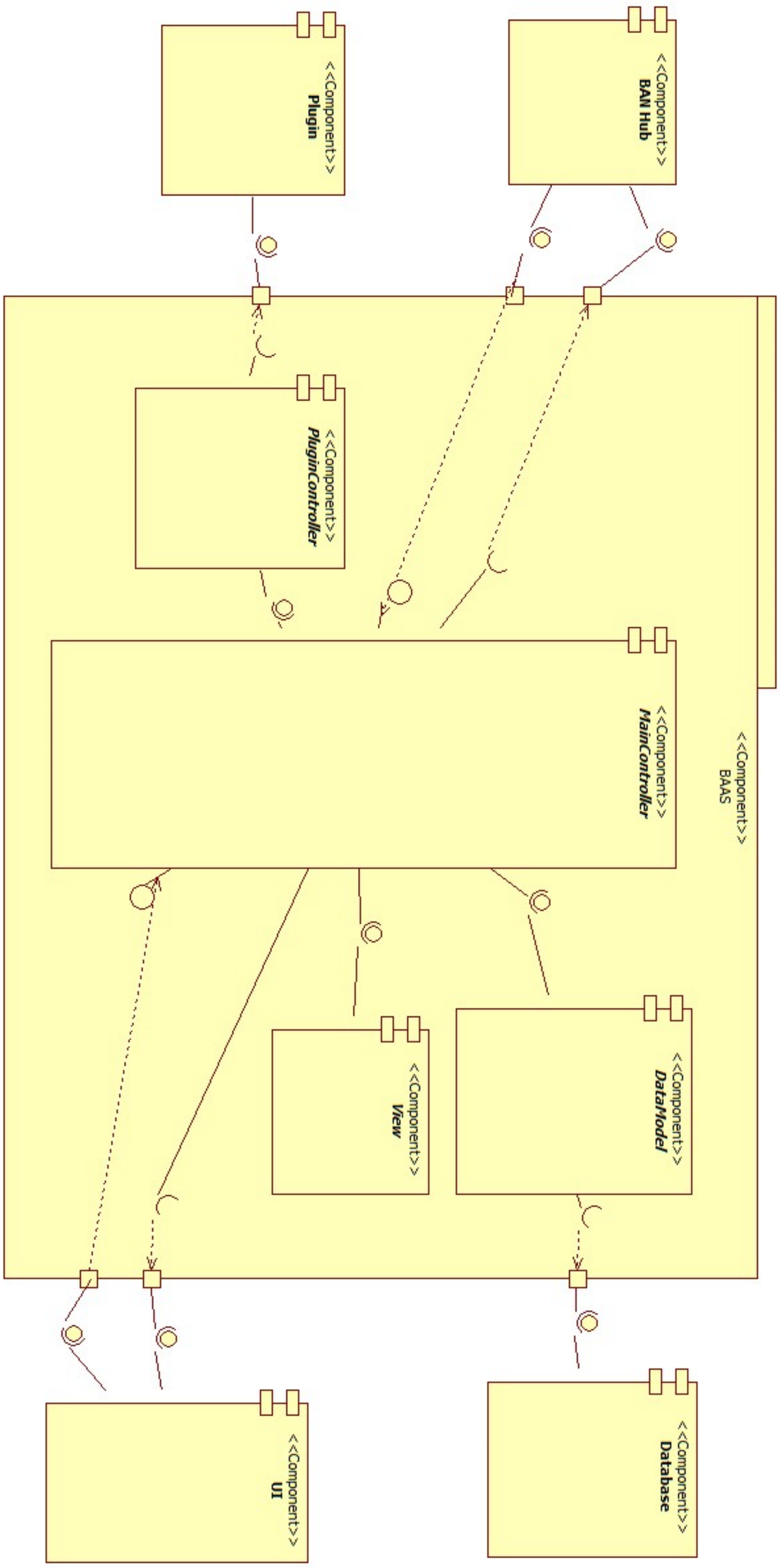


Figure IV-5. UML 2.0 component diagram of the BAAS the functionalities of the plugins through those interfaces. It is possible to incorporate the *PluginController* into the *MainController*. However, they were designed as independent components in order to keep the subsystem as elaborate as possible.

4.2. Selection decision of server implementation approaches

An important decision to be taken in the design process is how to build up the server. It is a necessary step before moving to the next step of interface design. An appropriate approach should be chosen for building up the physical server.

A self-defined server is the first solution. As the entire program has to be written from scratch, a low-level, self-contained program fits very well. It includes the tasks of designing, developing the data exchange protocol and implementation of the protocol based only on the lower network layers, i.e. TCP/IP. It achieves the best fit and is most applicable to the SensCoAP project goal, thus it completely fulfills the need of the application. Moreover, as the SensCoAP application is highly dependent on the wireless transmission, the full use of the bandwidth is meant to be efficient if a proprietary protocol is defined. That leads to a fast reacting system which is critical to a mobile system. However, this approach is time consuming and error prone, which will risk the security and the stability of the system. And this type of approach is very difficult to maintain and extend in future as the proprietary protocol is involved.

The second solution is to rely on an existing server framework and utilize the standard communication protocols. The part that fulfills the system requirements can be adopted and the remaining proprietary part should be developed for the application. The strengths of this solution is the promising protocols that have been tested over many years and are well known to people who will take over the maintenance work. A HTTP [104]-based server is suitable in this case. The HTTP protocol combined with open web server services is able to not only transfer the HTML pages from a server to a client, but also provide webservices on HTTP basis with the help of paradigms like REST (Representational State Transfer) [105]. Therefore, it easily supports a RESTful interface for communication and a platform independent HTML webpage for user access. To accelerate the project progress, a secure transmission protocol is ready to use with HTTPS which requires less development effort.

The interfaces mentioned in Subsection 4.1 can now be specified in more details, especially for the *View* component. For example, the interface used by the BAN Hub is meant to understand REST responses based on a certain formatting, and the interface to the *UI* is capable of displaying HTML webpages. Both two interfaces (HTML and RESTful) have an interface exported by the BAAS and accepting the incoming requests, while another interface exported by the client and used by the respective *View* component is used to present the results of the requests. It should be emphasized that the former one does not have to be technically different. Both sets of URLs exist in a HTTP based server and the interfaces are differentiated only logically by using a common root path for each of them [94].

4.3. Interactions between the BAAS components

To provide a better understanding of the interaction between the BAAS components, an example of the data flow indicating that the BAN Hub initiates a request to the BAAS is presented. This is a rather generic procedure that can be applied for the scenarios of uploading data or other types of requests. The BAAS always processes this type of requests in a similar way. The process is depicted in Figure IV-6 graphically by a UML 2.0 sequence diagram.

At first, the *BAN Hub* generates a request to the server using the REST interface which is accepted by the *MainController*. Thus, the *MainController* requests the user data from the *DataModel* to verify the credentials which are provided together with the request. The *DataModel* obtains the corresponding information from the *Database* and passes it to the *MainController*. After successful verification, the request is processed and responded by the *MainController* accordingly. This may lead to additional read/write events to the *Database* via the *DataModel* component. The query of the *Database* information is then processed by the *MainController* again. When an appropriate answer is rendered by the REST-*View* for the *BAN Hub*, the results is translated into a format that can be understood by the *BAN Hub*. At last, the *BAN Hub* receives the expected answer from the *MainController*.

In general, the process can be summarized as: requests are received and processed by the *MainController* which uses the *DataModel* to get access into information on the server. All answers sent back to clients are always translated by the client specific *View*.

4.4. Design constraints and considerations

The server will be deployed on a machine physically locating at IMSE. It may runs a Linux-kernel system e.g. Ubuntu Server or similar. To ensure the capacity of dealing with large amount of data, enough storage space should be guaranteed on the server machine. The server should be directed accessed only by the developers and is open to the World Wide Web with a fixed IP address for users. The user access is realized by webservices so that the system should be platform independent. Considering the convenience for most healthcare personnel and subjects with little computer knowledge, there is no installation needed to use the application.

Security is a big problem involved into the BAAS. The data security is highly ensured by the user access control scheme which applies both on the BAAS and the BAN Hub. The user authentication is examined by the *MainController* component in the BAAS. The connection to the clients is authenticated by HTTPS, which uses the secure sockets layer (SSL) or the newer transport layer security (TLS) standard to encrypt the exchanged messages and supports authentication of the server towards the clients using asymmetric cryptography [94, 106]. That means a server certificate will be used establishing a SSL/TLS connection to authenticate the server towards the client and encrypt the messages afterwards. Meanwhile, the client sends its credentials via the secure channel to authenticate the client towards the server, with the aim of mutual authentication and confidentiality.

The server certificate is either self-created or self-signed which is compulsory to manually install on the BAN Hub. If the certificate is signed by a root certification authority (CA) already known to the BAN Hub, then there is no need to install certification in the client.

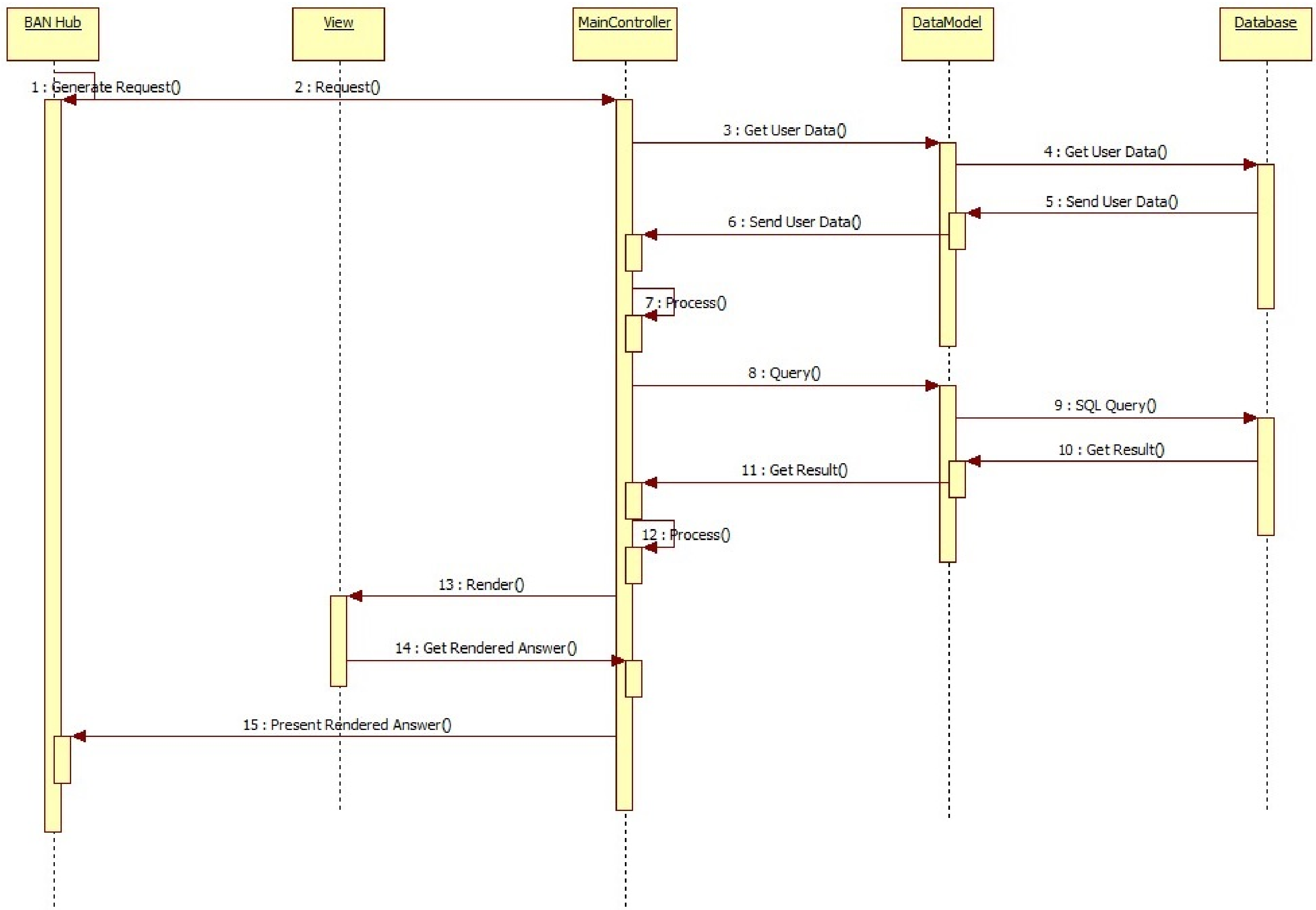


Figure IV-6. UML 2.0 sequence diagram of responding to a request on the BAAS

5. Database Design

In this chapter, the design of the database is elaborately introduced, which connects with the BAAS in SensCoAP for permanent storage of the biosensor data and other relevant information.

5.1. Design overview

The database is the basic system component for implementing the user management, the storage of the biosensor data and the maintenance of plugins, according to the REQs 17-21, 24-26 and 29. The structure of the internal database tables and their interactions can be represented by an entity-relationship diagram as Figure IV-7. The entities are presented by rectangles and ovals indicate properties. Associations are represented with rhombuses. The generic diagram depicts a role based access control (RBAC) scheme of the data model which enhances the security of the system.

The relational database constitutes of the entities, properties and their relationships in between. Each element of the diagram will be explained in details as follows.

Entities

User This central entity is treated as an abstract entity based on which different types of concrete users will be derived. All the users created together with the *Role* will inherit the properties from the *User*. The *user_ ID* is the unique identifier to distinguish users in the system regardless of their roles.

Role The users distinguishes from each other by the *Role* entity and this is where the identification of a user is really defined. Each *User* entity will be assigned to exactly one *Role*, e.g. HealthCare Professional (HCP)/ Researcher (RES) (under telemedicine / epidemiology circumstance respectively), subject or administrator. A *role_ ID* is used to identify the *Role* types and associates with the *Role*-based specific permissions. One role may have many types of permissions simultaneously when accessing to the system.

Permission The *Permission* restricts the behaviors a *User* can perform in the system depending on his/her *Role*. For example, if the *Role* of a *User* is "subject", he/she will only be assigned the "read" *Permission* during his/her login period, which means he/she can not freely edit the personal data or the biosensor data. The *superadmin* thus can create new roles by issuing different *Permission* combination. The system always works in the following way: once a *User* initiates a request for data, the system has to check if the user has been authenticated and has the permission to do this operation. If not, the operation fails and the request will be denied. By this way the data security is ensured.

Group This entity is necessary for both subjects and HCPs/RESs because the subject lists in the system are organized in form of groups. Therefore, there should be at least two relationships associating with the *Group* which will be introduced later. A *group_ ID* and a human readable *group_ name* are helpful for a HCP/RES to manage his

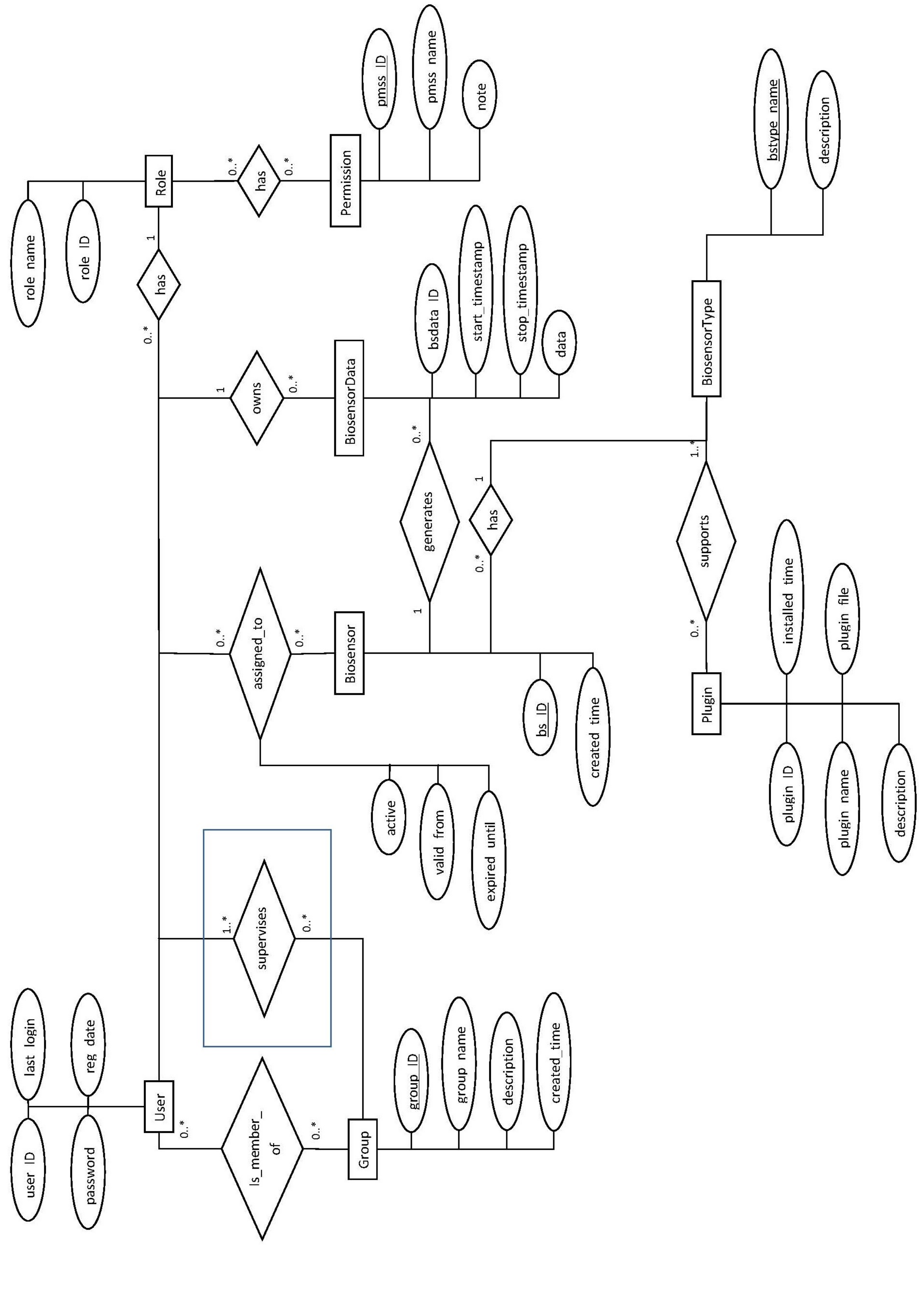


Figure IV-7. Entity-relationship diagram of the data model groups. A *Description* property is used to record the group-related information. The *Group* entity is the only link through which the HCPs/RESs can connect with their subjects.

Biosensor The *Biosensor* entity represents the biosensor devices that have been registered in the system. It is identified by the *bs_ ID* property which can be any unique string e.g. the Bluetooth MAC address.

BiosensorData The data generated by the *Biosensor* and forwarded by the BAN Hub are then stored in the *BiosensorData* entity. The received raw data are stored in the part of *data* property by terms of individual files and attached with overheads such as *bsdata_ ID*, *start_ timestamp*, *stop_ timestamp*, for the information of further processing. The *data* property is used to describe the name of those files. The raw format data are recognized in processing with the relationship to the recording *Biosensor* and *BiosensorType*.

BiosensorType As the BAN Hub is supposed to be compatible with multiple types of sensors, the *BiosensorType* entity is necessary to mark the types of *Biosensor*. Here the *bstype_ name* is used as unique identifier to distinguish the *BiosensorType* and the *description* property is used to notify the character of a specific type. With the *BiosensorType* entity it is easy to involve new types of *Biosensor* for system extension.

Plugin To fulfill the dynamical needs of data processing and analysis from the HCPs/RESs, the *Plugin* entity is involved. It marks each plugin with *plugin_ ID* and includes a *plugin_ name* for management. *Description* property is used to describe the functionalities of the *Plugin* and it indicates the installation history in *installed_ time* property. The location of the executable plugin file is stored in *plugin_ file*. With a relationship to the *BiosensorType* one *Plugin* can support multiple biosensor data formats.

Relationships

has It is a common relationship indicating one entity is a nature character of the other entity. It exists between *User* and *Role*, *Role* and *Permission*, *Biosensor* and *BiosensorType*. The former owns the latter as a unique character, thus usually their relationship is always many-to-one: the former should have exactly one latter entity. However, the latter entity, as a kind of character, can be assigned to multiple former entities.

supervises This relationship is particularly highlighted by a blue frame in Figure IV-7, as it should only exist when the *Role* of a *User* is "HCP/RES" or similar *Role*s who are responsible for taking care of the subjects. Other *Role*s should not have this type of relationship in the data model. It connects the *User* and *Group* and provides a bridge for the HCP/RES to manage his groups of subjects. Normally one HCP/RES can handle more than one group simultaneously and one group can have several HCPs/RESs to be its manager as well.

is_ member_ of This relationship is offered to those *User*s whose *Role* is "subject". That indicates to which group the *User* belongs. A group has one or more members and one member/subject can be supervised in several groups by different HCPs/RESs.

assigned_to Before starting to measure the subject, the *Biosensor* devices should be bounded to a specific *User*. Thus, it records the issuing time, when the *Biosensor*s should be returned and if the assignment is active at the moment. By using the *assigned_to* relationship the system can keep track of the *Biosensor*s. One subject can wear several *Biosensor*s at a certain time, whereas one *Biosensor* can be only worn by one single subject during the assignment period. Therefore the assignment of a specific *Biosensor* must only be active for one subject at a time. The many-to-many relationship is able to keep a complete history of the assignments from the past until now.

owns In order to know to which subject a chunk of data belongs, the *owns* relationship is introduced. A subject can *own* many chunks of *BiosensorData*. Meanwhile, one piece of *BiosensorData* can only have one source subject.

generates To be able to determine from which *Biosensor* a specific chunk of *BiosensorData* comes, the *generates* relationship is used. A *Biosensor* can *generate* many chunks of *BiosensorData*, whereas each chunk of *BiosensorData* should only have one generating *Biosensor* device.

supports Different types of *Biosensors* have different data formats. When installing a new *Plugin*, the developers should indicate which *BiosensorType* it supports. A *Plugin* should be able to deal with at least one *BiosensorType*, and one *BiosensorType* is compatible with several *Plugins*.

5.2. Design constraints and considerations

A generic data model was drawn in Figure IV-7 especially for the *User* entity to meet the requirement of a role-based access model. Actually a more realistic model is to specify the *User* entity as three roles: HCP/RES, subject and administrator (as described in REQs 24-25). The fixed roles make the model simple and based on that several specific relationships can be developed. However, as the SensCoAP system is supposed to be modeled as generic as possible, the design does not work with these fixed roles, and therefore the system will be more complex and its performance will possibly be reduced.

The rationale of accepting the generic model is: it makes the system flexible to add new roles and define new permissions dynamically. That will be preferred in the further development for the sake of maintenance.

As for the design of the database interface, a relational database management system (DBMS) should be used. In this case the data model will store, maintain and query data through SQL interfaces. As different databases differentiate slightly on statements and representation, the data model should be adjusted if replacing the current database with another type. A solution is to use an object-relational mapping (ORM) inter-component, which maps the object-oriented programming paradigms to different databases so that it allows the independent data model regardless of any specific database type. Although it reduces query performance, the ease of development and maintenance is increased.

The candidate database should support *datetime* fields with fractional seconds as the BAAS needs it to store time stamps (REQ-18). Databases like MySQL are not supposed to be used. PostgreSQL might be a good choice supporting this feature.

6. External UI Design

External interfaces include the interfaces to external components (e.g. biosensors) and the input/output user interfaces. The interfaces to the biosensors and between the subsystems were already explained in Sections 3, 4 and 5 exclusively. This chapter only discusses the design of the UIs for the BAN Hub and the BAAS.

6.1. BAN Hub UI design

To meet the requirements (REQs 5,7-9,11,15) and the use cases in the system requirement specification, the UI to the BAN Hub should at least be capable of accomplishing the following tasks: start/stop recording, search for sensors, upload data, and configure the system. An additional function for visualization of the cached data is also preferred, in order to interact with the users. It can be handled as an additional button aside each listed sensor. Once the button is clicked, it redirects to another dedicated view where the data are visualized and the alerting system may be used based on those data curves. This will solve the presentation problem for recording multiple sensors at the same time.

Concur task tree (CTT) model will be used to demonstrate the interaction activities triggered on the BAN Hub according to its functionalities. CTT elicits the system functional tasks into four categories: abstract task, user task, application task and interaction task. The strengths of CTT lie on that not only its graphically hierarchical structure is intuitive to the designers, but also it is activities focused that helps make the system aim clear, without considering the complicated design stages and implementation details [102]. Under certain circumstances the CTT can also be combined with UML 2.0 and notation for XML for more delicate modeling if necessary [107].

Concur task tree environment (CTTE) version 2.6.0 is used to model the system in this section. Figure IV-8 shows an example of the BAN Hub modeled by CTT. This CTT diagram depicts the tasks into four levels of subtasks with their sequences marked.

The aim of the example is to upload data to the BAAS. This task is an abstract one (marked with a cloud symbol) and can be divided into four subtasks, with sequence relations connecting them. To be able to connect with the biosensors, the application (application task is marked with computer symbol) should firstly search for the available biosensors around and the user has to pair them in order to establish the connection (interaction task is marked with small color picture). After the user starts the recording, the data generated by the biosensors are received and then stored in the cache, which are included into the abstract task "cache data". When synchronizing with the BAAS, the BAN Hub first initializes an uploading request, following by the response of checking the identification of the requester. The account information is input by the user with his ID and password. After successful verification, the BAN Hub starts uploading data to the BAAS.

A more detailed function-based flow chart of the UI is depicted as below (Figure IV-9). This diagram models the functions and operations of the BAN Hub UI. It also performs as a basis of the following mockups in Figure IV-10.

In Figure IV-9, firstly all important activities should retry if the first attempt fails. For example, if the search for the surrounding devices fails, it will have a second try later after several seconds. Secondly, there are at least two graphical items on the main UI so the "search for devices" and "configure" are available to the user from the very beginning.

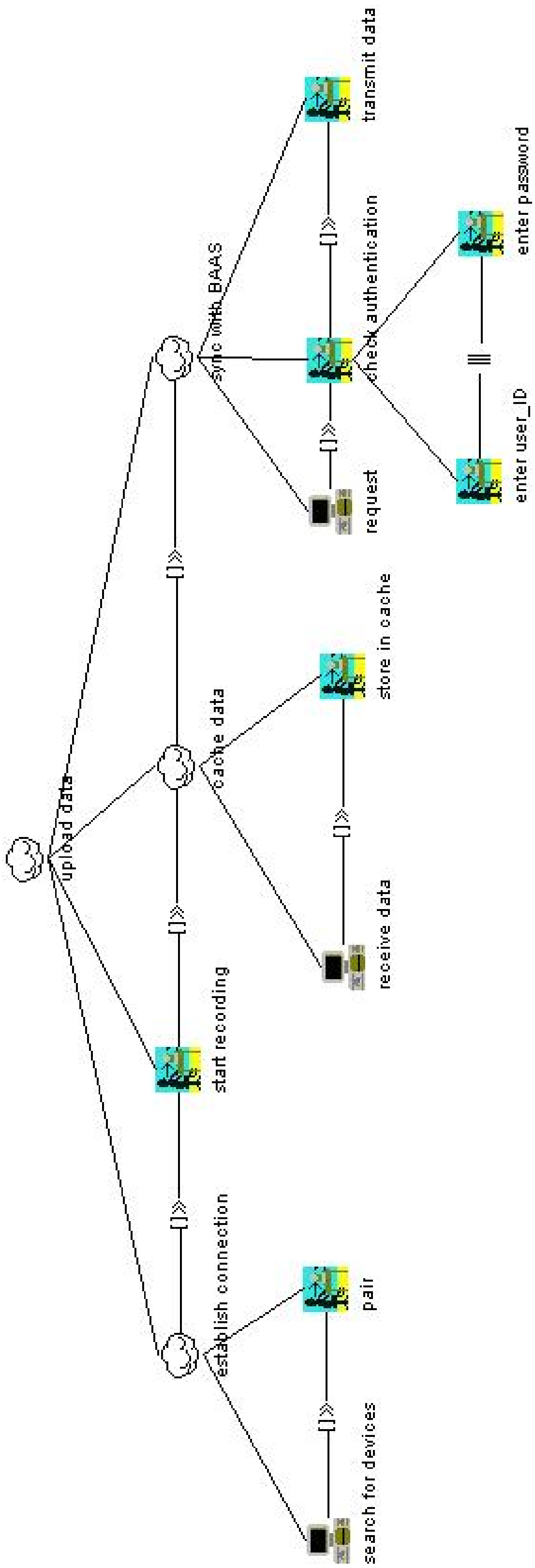


Figure IV-8. Concur task tree model of uploading data from the BAN Hub to the BAAS

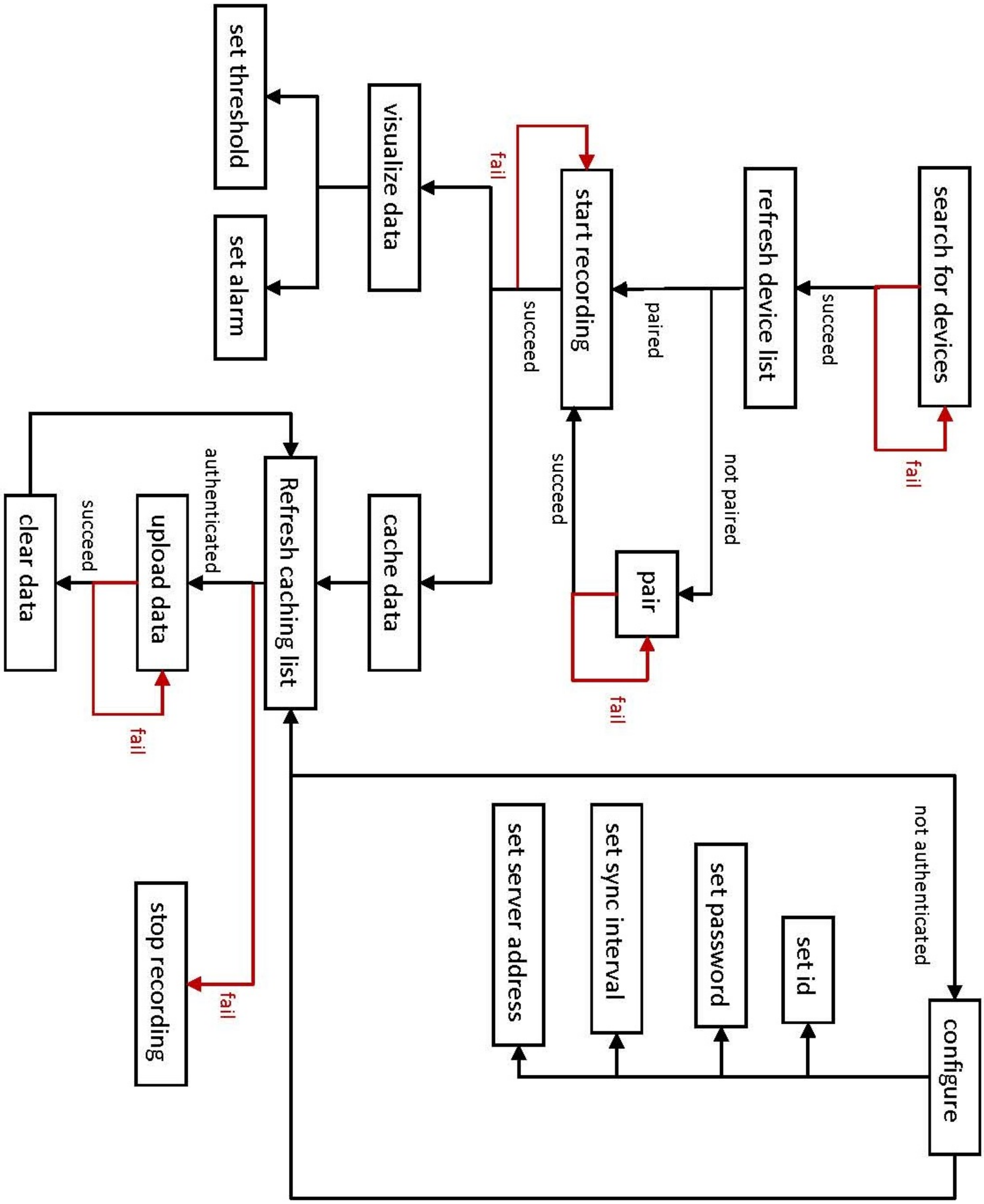
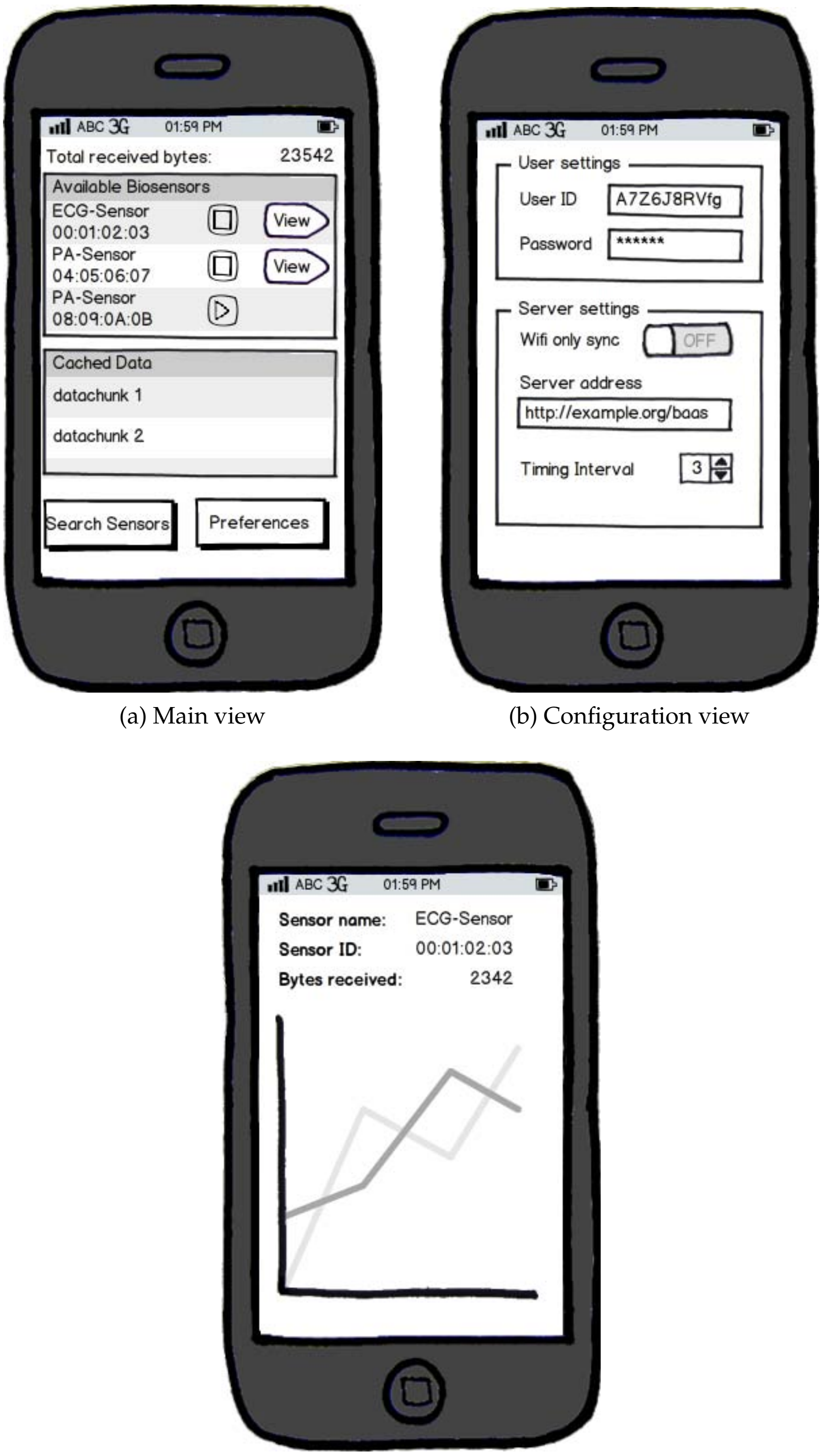


Figure IV-9. Flow chart of the BAN Hub UI functions

Three mockups are shown in Figure IV-10 for the general appearance of the BAN Hub UI design [94]. They are created using the open source Balsamiq Mockups software (version 2.2.1) [108]. Figure IV-10a represents the main view of the application with three divisions. The upper one represents the total sum of the received bytes from last start. In the middle there is a list of available sensors with "start/stop recording" buttons aside. Additionally, it may also contain a link to the visualization view of the data in Figure IV-10c. The last part is the division for dynamically displaying the cached data chunks to the user. Two buttons performs as the starters of the app as mentioned above. By clicking the "preference" button it can redirect the user to the configuration view of the app as in Figure IV-10b. It provides the user the possibility to set his credential as well as relative information regarding the system. Once the setting is done, the user can redirect back to the main view via the back button on the Android smartphone.



(a) Main view

(b) Configuration view

(c) Visualization view

Figure IV-10. Mockups of the BAN Hub UI [1]

[1]Figure from Horsch et al. [94]

6.2. BAAS UI design

The interfaces directly exposed by the BAAS are: database interface, REST interface, plugin interface and web interface. They are designed to fulfill the REQs 19-26 and 29.

The database interface is already discussed in Section 5. The following paragraphs start from a short introduction of the principles of Representational State Transfer (REST). A RESTful interface builds a resource-oriented webservice through the existing HTTP features such as different request methods GET, POST, DELETE, PUT combined with the status codes (e.g. 201 Resource Created). The resource identified by a URL can be requested by using those request methods. Compared with other types of webservices, the designers do not have to define new and functional vocabulary for the interface, so that it promises the simplicity and concision for the BAAS interface. There are more details regarding the application of the REST by terms of HTTP methods and status codes combination in [94].

The plugin framework provides a way to process sensor data differently, and moreover it has the ability to deal with different data formats. The BAAS has to offer a standardized interface to add or remove the plugins in a generic way. Thus the plugin interface should have at least two logical parts. The first part containing all relevant information regarding the plugin name, the description of its function and the data type it is able to handle. The second part is regarding processing, which offers the functions for processing data and all other metadata-associated information e.g. the timestamps. The returned processing results should include a format description to inform the BAAS covering from single values to time series.

The web interface is the real object of this subsection. Its design directly decides how it looks like to the users and how it interacts with the users fulfilling its functions. It can be accessed by a browser thus should be platform independent.

Its main features are role dependent, which means not all its users are allowed to perform the same operations on the BAAS (REQs 24-25). The main features are summarized as follows.

Authentication The server should have a session layer to handle the login and logout of the users. All users who have authentication can get access to the system; otherwise every attempt to access any pages of the system will be redirected to the login screen. The users may use a cookie based system so that their authentication can be stored temporarily on the browser. An accessible logout option should stay at the same location on each page. There is a timeout scheme to invalidate the user session after a while of inactivity to protect the data.

User management It is possible to create/remove users and maintain their information in the system. Basically the superadmin has the right to add new administrators and the administrators have the right to add new subjects. Once a new user is created, the user ID is generated automatically by the system and thus the user has to type a password for his credential. For the HCPs/RESs and administrators they also have to additionally provide their personal information such as names and email addresses for proof. A user profile containing all his information and details exists in the system. In case a user wants to edit his data, the user ID must be provided.

Group management According to a specific aim, the HCPs/RESs manage their subjects as groups. The HCPs/RESs have the right to add/remove groups and add/remove their subjects to/from the groups. The responsible HCP/RES is also a part of his group. In addition the system should also offer details of the group information such as the group name and the description of the group, which is also the required information to create a new group.

Biosensor management The interface has the ability to register new biosensors and manage them as well. The registered biosensors with details can be assigned to a specific subject for a fixed period of time. An important part of the biosensor information is the biosensor type which is also managed by the system. A name and a description are necessary for adding a new biosensor type. A new sensor is marked with a sensor ID and corresponding biosensor type in the system.

Biosensor data management The interface provide a list of all uploaded biosensor data which can be sorted by their uploaded time, the source sensors or other attributes. The view of the data details can also provide the HCPs/RESs with the access to the processing plugins. The processing results will be returned on the screen depending on the functions of the selected plugins. Graphical result representations are possible by using web application technologies.

Plugin management Plugin scheme is an essential part of the BAAS system. The plugins are also added/removed for management and details of the installed plugins can be requested.

CTT model is used to demonstrate a simple example for helping understand the working features of the BAAS UI design. Figure IV-11 shows only one part of the functionalities of assigning a new type of biosensor to a new subject by the HCP/RES. The task begins with either creating a new biosensor or a new subject. To register the new biosensor, a new type of the biosensor should be created before the HCP/RES enters the biosensor information. When both the subject and the biosensor are registered in the system, the HCP/RES can select from the biosensor list and assign it to the target subject.

There is a complete function modeling in Figure IV-12 by terms of a flow chart. It describes the UI that a HCP/RES will face after successful login. The main functions are managements of groups, subjects, biosensors and data. The HCP/RES has the right to change corresponding information under those four sections with communication to the data model introduced in Subsection 5.1.

As in the BAN UI design, the free software Balsamiq Mockups (version 2.2.1) [108] is also used to create mockups for the BAAS UIs. A mockup showing the group management view after the HCP/RES login is depicted in Figure IV-13. It offers a function to create a new group and a list below containing all details of the group information. The HCP/RES will be added into the newly created group automatically by the BAAS. There are also links on the group names which will lead the HCP/RES to an individual page with the group details and member list.

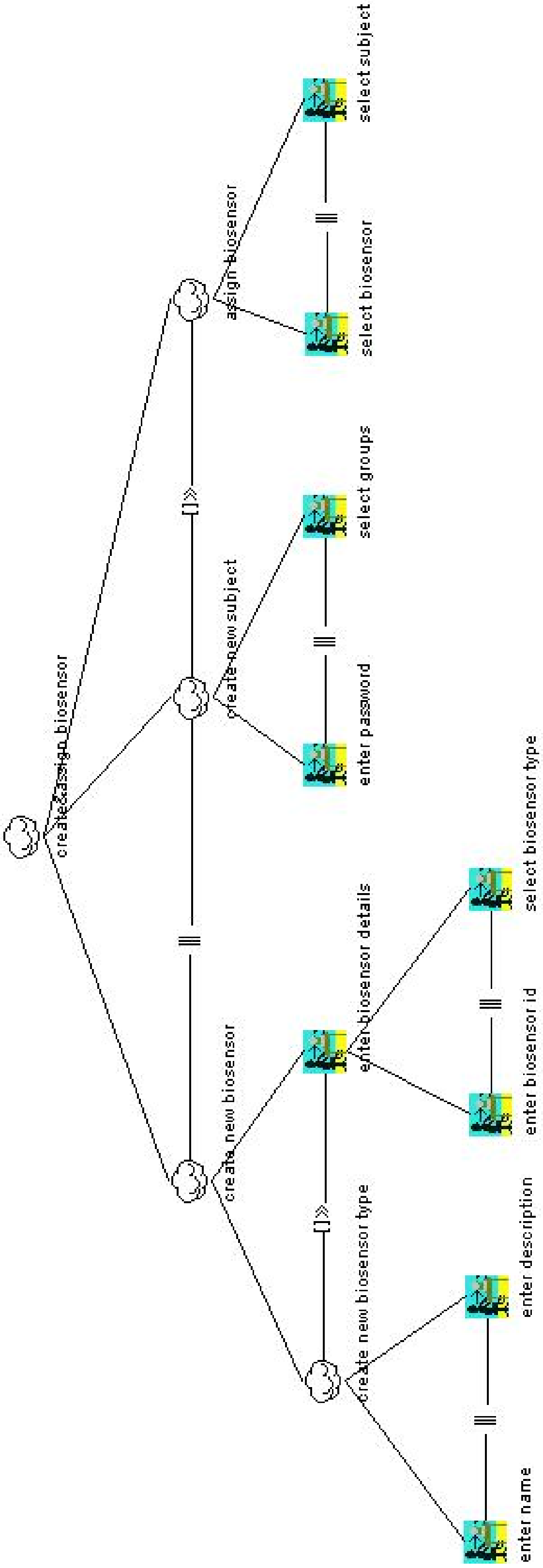


Figure IV-11. Concur task tree model of assigning a new type of biosensor to a new subject

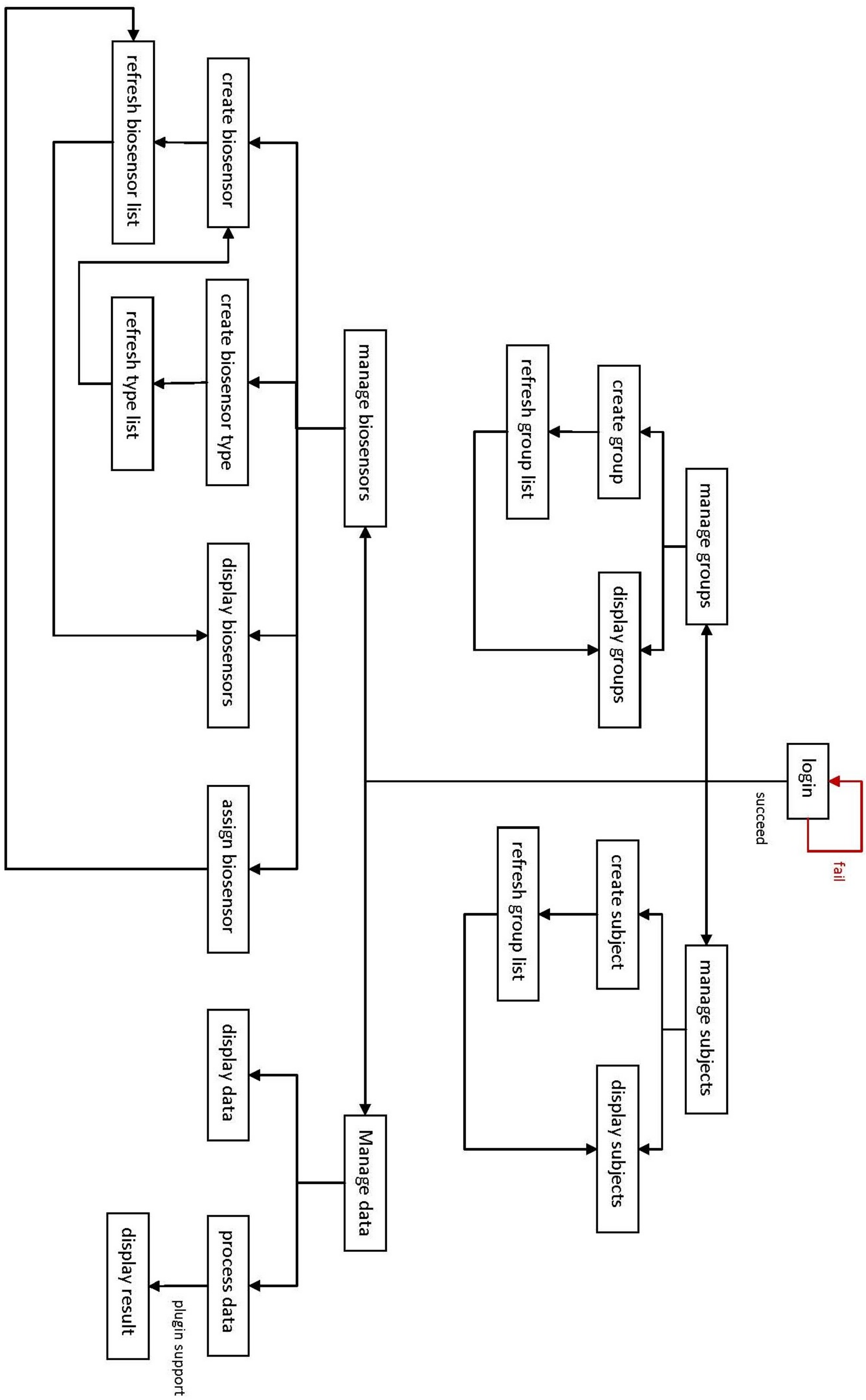


Figure IV-12. Flow chart of the BAAS UI functions

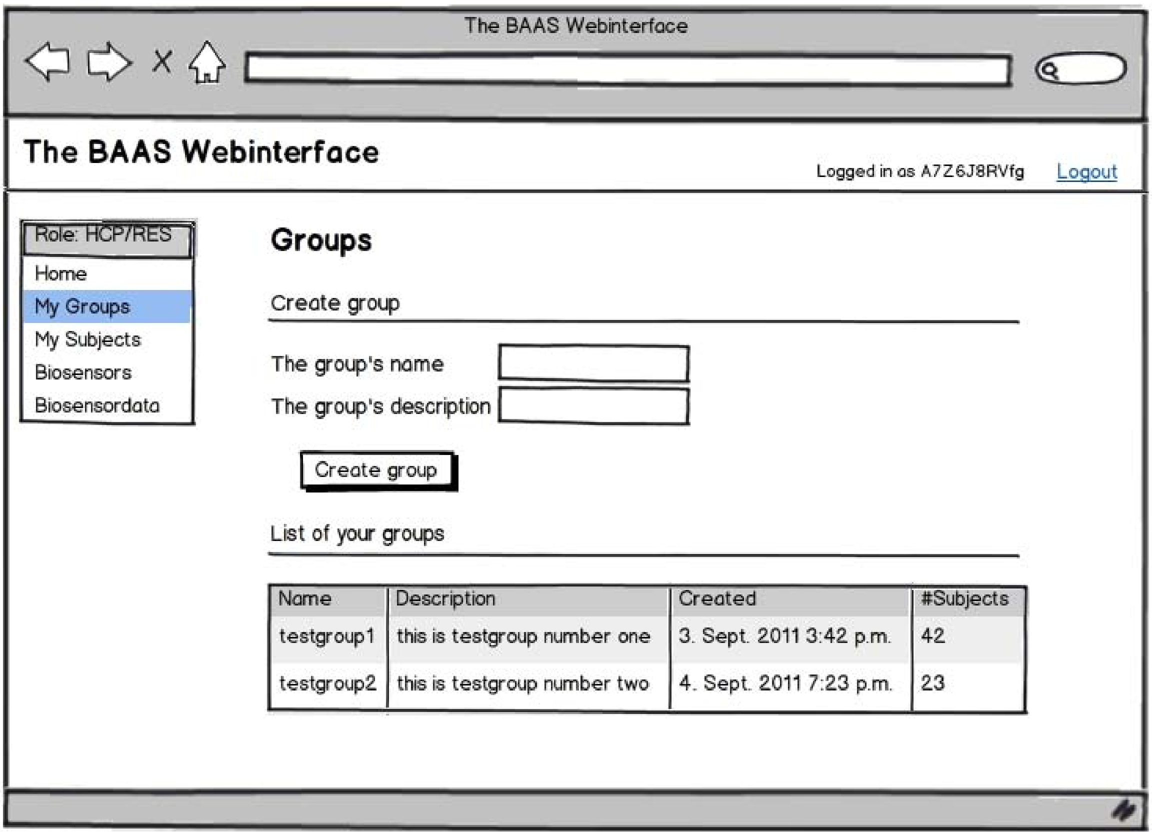


Figure IV-13. A mockup of the BAAS Web interface showing the groups view in the role of a HCP/RES [2]

The general layout of each view in the BAAS interface is like this: on the left side there is always the navigation bar for the user to choose from all available functions. The head showing the login information and logout button together with the navigation bar will stay at stationary locations when the login user is redirected from page to page. The main part of the page will be changed according to the features the user selects. The upper part of the main area is always the elements to add new entities and the lower part is a list of the available entities that are already registered in the BAAS. There will be links in the list of e.g. names or IDs that lead to new pages with details of the specific entities.

[2]Figure from Horsch et al. [94]

Chapter V

Implementation of SensCoAP

Contents

This chapter emphasizes the practical part of the SensCoAP project. It introduces the procedure and all details related to prototype implementation such as the build-up environments and the realized functions. The open source and ICT that were used to build up the system are described. More technical aspects are mentioned here. The chapter starts with an overview of the SensCoAP prototype, followed by details of its subsystems. It should be pointed out that not all the requirements and use cases presented in Chapter III are supported by the prototype. This chapter ends with a discussion of the missing features that could not be realized so far, along with the reasons [1].

1. Prototype Overview

The prototype of SensCoAP logically consists of two parts, as shown in Figure V-1. On one side, there is a server running a BAAS application within the Django framework. On the other side, there is a client that communicates with the server using HTTP as underlying protocol, in the form of either a BAN Hub based on a RESTful interface, or a browser based on a HTML web interface.

[1] All results, tables, figures, descriptions and discussions, are based on the student work carried out by Preg A, Horsch J, Schindler F, Sievers J, Schmidt J, Dorner M, Dimitrov I, Eberspächer T and Jacht V.

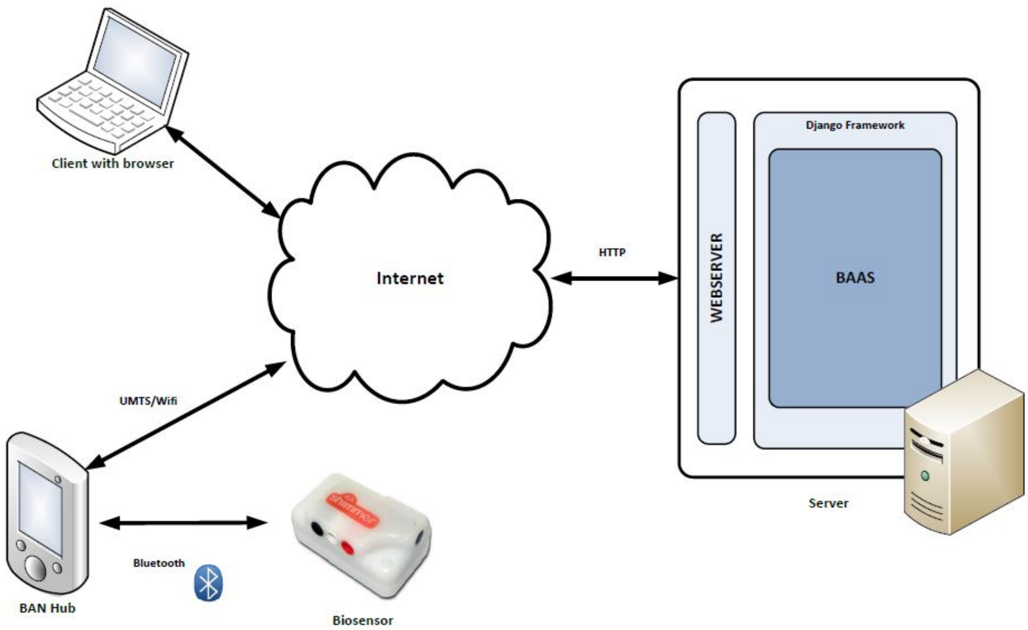


Figure V-1. The prototype of SensCoAP [2]

The BAN Hub is running an Android operating system, connecting with the biosensors via Bluetooth and with the server via WiFi or mobile networks such as UMTS, GPRS and 3G. The server is dealing with HTTP connections, the attached BAAS application is used within the Django Python web framework which integrates with the web server by the web server gateway interface (WSGI) [109]. Once a request to the BAAS is initiated, the web server identifies the requester by its URL and forwards the request to the BAAS. In the same way, the web server sends a response to the corresponding requester after the BAAS has processed it. For example, the BAN Hub is now trying to upload data to the BAAS. This is initiated by one of the following trigger events:

1) The subject stops the recording of the biosensor via the UI on the BAN Hub.

2) The preconfigured synchronization interval has expired.

3) The subject manually uploads the data chunks by clicking on the screen of the BAN Hub.

4) An internet connection which got lost is resumed.

The BAN Hub initializes a HTTP POST request to the server which contains the data chunks as well as the user credentials. After the BAAS gets the POST request, it first

[2]Figure from Horsch et al. [94]

checks the user name and the password, if they stay consistency with the system record. Once it passes the authentication check, the BAAS stores the data and responds with a HTTP status code 201. The BAN Hub will remove the cached data from its memory after getting this message.

File orientation

The SensCoAP prototype is file oriented, which treats the data chunks from the biosensors as individual files. The BAN Hub receives the biosensor data without decoding them, and transmits the data files in their original formats ("as-is") to the BAAS. The BAAS stored them in the formats of the files as well. Instead of storing the entire data chunks, the database of the BAAS stores only the metadata (information relevant to the data file) of the data and a reference to the corresponding data file.

The biosensor data are stored as raw data from their source sensors on the BAN Hub without any decoding. Exception only exists on demand when the user requires the visualization function or activates the alarm function. Non-decoding brings many advantages to the BAN Hub. Without decoding, all the data are forwarded as they were generated on the biosensors, so it greatly reduces the power consumption on the BAN Hub, which is a big problem for current smartphones. Moreover, as there is no effort for decoding, there is no need to formulate any specific data format on the BAN Hub. This will be helpful when involving more types of biosensors into the system.

The file oriented prototype guarantees a special file naming scheme that includes all the metadata of the data file. The source biosensor ID indicating where the data chunk is generated and the start/stop timestamps indicating the recording period are all involved into the file name of the data chunk. With this scheme, the BAN Hub is able to easily interpret its temporarily cached data even if it loses all its relevant records of the data, e.g. after a restart. The naming format is:

```
<biosensor-id>_<start-timestamp>_<stop-timestamp>
```

The biosensor ID is normally the Bluetooth address of the biosensor. The timestamp is represented in milliseconds since the first of January 1970, which is how Android produces them. A file name example for a specific recording by a sensor with the Bluetooth address 00:06:66:42:23:FB may look like this:

```
0006664223FB_1314160797959_1314160813545
```

With this scheme, it is able to be compatible with all types of data formats in the system. The part dealing with data format is moved to the BAAS plugins when processing the data. Under this concept, the storage of the data is completely independent from the data formats in the SensCoAP prototype.

Session orientation

The data files cached on the BAN Hub may have different sizes. The file size depends on the synchronization interval which is highly influenced by various factors like network connection status. Thus a session in this system is defined as the data recorded from the manual start until the manual stop of a recording. It is possible that several uploaded data files with various file sizes belong to one session. The BAAS involves a mechanism to distinguish those data files and merge them into one session file in the database. It works with checking a newly uploaded data file if there is any files in the same recording session found on the server before or immediately after the uploading. When files belonging to the same session are found, the BAAS merges them into one file, as the BAAS can only handle a single file in later processing and analysis.

2. BAN Hub Prototype

This section discusses the details of the implementation on the BAN Hub. The decisions made during implementation as well as the technical considerations are reported here.

2.1. Implementation decisions

Before going deep into the techniques, decisions made on selecting hardware and software together with necessary technologies are documented. One principle of the SensCoAP project is to utilize open source and off-the-shelf products as much as possible. Therefore, there are some choices made by the group before implementation.

Biosensors

For prototyping, two sensors have been used to demonstrate the SensCoAP functionality: Shimmer and ActiGraph wGT3X+.

The Shimmer [110] sensor is a good candidate to be used with SensCoAP because of its compatibility, openness and flexibility. It is a wireless motion sensor from Realtime Technologies Ltd. (Dublin, Ireland) and offers the possibility to expand its sensor board with more sensors. Its baseboard integrates 10 DoF motion sensing including 3-axial accelerometer, gyro, magnetometer and altimeter (see Figure V-2a). It is allowed to add more sensor daughter boards for modular expansion such as ECG/EMG and GSR (see Figure V-2b).

(a) Shimmer 2 base unit (b) Shimmer 2 with ECG daughter board

Figure V-2. Shimmer sensors

Shimmer is small enough to be used as a wearable sensor and it uses Bluetooth and 802.15.4 radio modules as potential means for realizing real-time data streaming. It is able to be tailored to any private application due to its TinyOS-based platform, which make Shimmer easy to program for different application purposes. One or more Shimmer nodes may form a star network topology to collect data simultaneously at a certain sampling frequency (normally 50 Hz) [3]. The hardware version used in the SensCoAP implementation is Shimmer 2 base unit with accelerometer and gyro.

In our group dealing with measurement of physical activity in epidemiological cohorts, there was a strong wish to support the ActiGraph [111] wGT3X+ accelerometer in the

SensCoAP. ActiGraph sensors are widely accepted by researchers worldwide for objective physical activity measurement. wGT3X+ is the wireless version of GT3X+ sensor which equips a micro-electro-mechanical system based accelerometer and an ambient light sensor (see Figure V-3a). The wireless option of the wGT3X+ extends the capabilities of the devices and enables the inspection of the measured data in real-time via ANT+ [112].

ANT+ [103] is an open source wireless sensor network technology. Compared with Bluetooth, it has the advantages for low computational overhead, high efficiency and low power requirement. However, it is not reliable due to lack of validations. ANT+ communication module is enabled on some Android smartphones from certain manufacturers, such as the Xperia series from Sony (See Figure V-3b).

(a) ActiGraph wGT3X+ (b) Sony Ericsson Xperia

Figure V-3. ANT+ enabled devices

ActiGraph keeps the data source and the ANT+ communication parameters of wGT3X+ proprietary and after several attempts it was found not possible to involve wGT3X+ into SensCoAP at that moment. But it is found that ActiGraph released Bluetooth enabled wGT3X+ recently, which is definitely worth to try again in future development.

Smartphone

As stated in the design decisions, an Android smartphone has been chosen because of openness and flexibility. The device used in the SensCoAP implementation is HTC One with Android Version 2.1. In later evaluation phase other smartphones like Samsung S3 with Android 4.1 were also involved. The communication with the biosensors is carried out by Bluetooth, even though Shimmer has also Zigbee component. The reason is that currently smartphones on the market seldom contain Zigbee modules, even if Zigbee consumes much less power than Bluetooth. At the moment, the smartphones used in the project are equipped with the normal Bluetooth instead of the new Bluetooth low energy version.

2.2. Implementation overview

For development on Android, the Android SDK is needed together with the IDE and ADT plugin. The necessary development tools and their versions are listed below:

- Eclipse IDE, Version 3.7.1
- Android SDK Tools, Revision 15
- Android SDK Platform-tools, Revision 9
- Android ADT plugin, Version 15.0.0

The implementation of the BAN Hub prototype is based on the understanding of the component design Section in 3 and Figure IV-3 of Chapter IV. The MVC pattern used in the structure design reduces its complexity and increases its maintainability [94].

Figure V-4 depicts the structure of the BAN Hub prototype by a UML class diagram. All the classes in the diagram and their mapping to the component design in Figure IV-3 will now be introduced in details.

MainActivity, *Visualization* and *Preferences* These three classes represent three screen views of the BAN Hub UI correspondingly. They are part of the *UIController* component in the system design. The *MainActivity* represents the main screen where the users interact with the system for starting/stopping recording, whereas the *Preferences* screen view is used for configuration. The *Visualization* involves the decoding and interpreting of the incoming data for visual presentation. The *MainActivity* is derived directly from the superclass *Activity* in Android. The *Preferences* is derived from a special superclass called *PreferenceActivity*, which offers a predefined method to implement a preference screen without dealing with the details. The stored *Preferences* is accessible by all the classes in the application through the *PreferenceManager* and a *SharedPreferences* object.

ProcessingService This is an important class in the core of the prototype which maps to the *ProcessingController* component in the design. It is responsible for all components and functions relevant for recording, such as start/stop recording through *BluetoothManager*, store the data in caching files and transmit the data files through *TransmitService*. The caching files are by default stored on the internal memory of the smartphone. However, it has the option to store them on an external SD card, as well. The *ProcessingService* is a continuous service which is supposed to be also running at the back-end when the application is shut off. Therefore, it is a so-called bound service which allows other components to bind to it for calling public functions of this service instead of sending messages. There is only one component, namely the *MainActivity* in the application, binding to it. This results in an automatic service termination once the main screen is shut off. A solution is to start the service by calling `startService()` before the bound, which ensures that the service keeps running

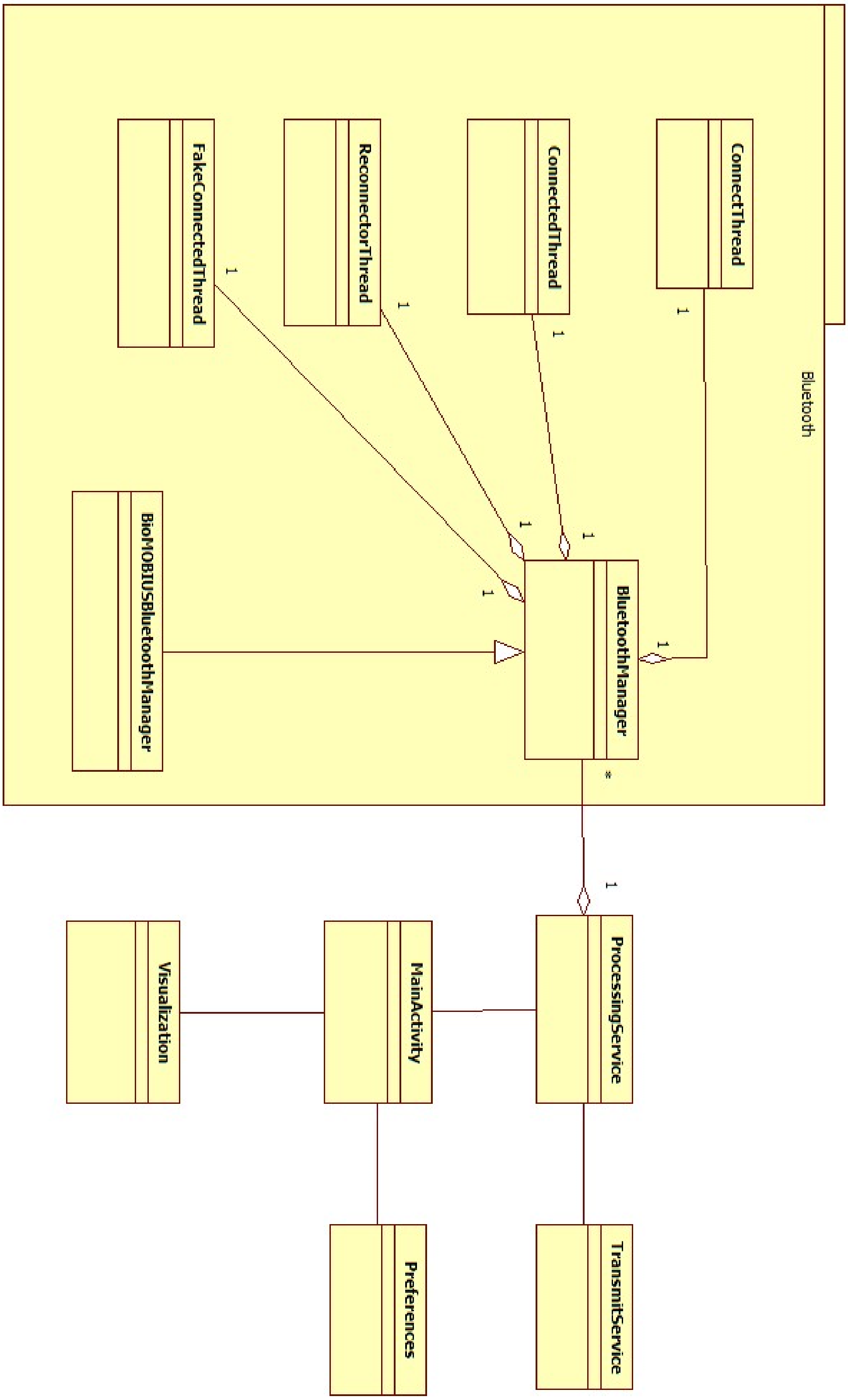


Figure V-4. UML2.0 class diagram of the BAN Hub prototype unless it is killed explicitly. In order to avoid the service to be killed by Android when too little memory is left, the *ProcessingService* uses a user notification scheme by which the `startForeground()` makes the service visible to the user. It shows an icon in the notification bar so that it is unlikely to be killed as it is actively noticed by the user. Depending on the network connectivity, the *ProcessingService* also incorporates the function to adjust the time between two send requests (the synchronization time interval) dynamically. It uses a *broadcast receiver* to catch the connectivity changes and then trigger the update of the synchronization interval.

BluetoothManager The sensor communication related package *Bluetooth* is corresponding to the *SensorController* component in the system design. The *BluetoothManager* coordinates the other classes, including: the *ConnectThread* which is used to establish a connection to a sensor; the *ConnectedThread* which reads the sensor data and forwards them to other classes; the *ReconnectThread* in charge of resuming lost connections; and the *FakeConnectedThread* which simulates the behaviors of a fake Shimmer sensor for testing and debugging purposes. What is worth to be mentioned is that every *ConnectedThread* or *FakeConnectedThread* owns its *AlarmMonitor* instance associated with specific sensor types for alarm setup under emergency situation. The *BluetoothManager* deals with the general data handling in order to achieve the file oriented scheme mentioned before. As one *BluetoothManager* is bounded to exactly one sensor during its lifetime, one *ProcessingService* should be able to handle more than one *BluetoothManager* so that it is possible to start multiple sensors for recording simultaneously. Once a new *BluetoothManager* is initialized, its own set of associated thread classes are issued. A specific *BluetoothManager* is removed once the recording has been stopped. In order to keep the threads and respective data safe, Hashtables are used by the *ProcessingService* to manage multiple instances of the *BluetoothManager* and other necessary variables. The advantage of this method is that it achieves the decoupling of sensor specifics from general sensor connection and recording functionality [94]. A general base class *BluetoothManager* integrates all common functionalities of all sensors. For a specific sensor type and specific features, a subclass should be instantiated by its name. The *BioMOBIUSBluetoothManager* is the subclass particularly for Shimmer sensor incorporating start/stop commands.

TransmitService It corresponds to the *SyncController* component and is responsible for transmitting the biosensor data to the BAAS with encapsulating all server related and specific code. It is an *IntentService* started by using a matching intent together with the `startService()` function. The intent includes the name of the file to be uploaded. It automatically spawns a worker thread to handle its intents in turn and stops itself when it runs out of work. Therefore, the *ProcessingService* only needs to upload the intents of the files without caring about the lifecycle of the *TransmitService*. In order to be able to send HTTP POST requests with the content's MIME-type "`multipart/form-data`", the *TransmitService* uses the external Apache HttpMime library (version 4.1.2) [94].

2.3. Interactions between the BAN classes

In order to help understand the interactions between the BAN Hub classes, Figure V-5 dipicts a UML sequence diagram of the actions taken from starting until stopping a recording as example.

The actions start from the moment the user starts the app on the smartphone and clicks on the screen for initializing a recording. The interactions are caught by the `start / stoprecording()` method of the *ProcessingService* with the sensor address as parameter. After the type of the biosensor has been recognized, an instance of the *BioMOBIUSBluetoothManager* is initiated. The *ProcessingService* then calls the `connect()` and the `startForeground()` to make itself run in the foreground. The *BioMOBIUSBluetoothManager* tries to connect with the biosensor via the *ConnectThread*. When the thread is successfully established, the method `connected()` is called with the parameters of the open Bluetooth socket as well as the device representation. The *BioMOBIUSBluetoothManager* then instantiates a *ConnectedThread* with the Bluetooth socket and starts data transmission by calling `startTransmission()`, which results in a specific start command to be sent out to the biosensor by `write()`. The *ConnectedThread* keeps calling `read()` to fill a buffer with the sensor data from the socket, packing them up with the corresponding timestamps (from the Android system) together with the sensor address, and sending it back to the *ProcessingService*. The *ProcessingService* identifies the data according to their address and caches them into a data file following the name rules described before by using an extra handler. Once the recording is stopped by the user, the *ProcessingService* calls `stop()` to the *BioMOBIUSBluetoothManager*. This, in turn, stops the transmission in the way it was started, but with the actions in reverse order, with a specific start command and `cancel()` kills the *ConnectedThread*. Meanwhile, the data file is closed and uploaded by sending an intent to the *TransmitService*. After everything is done, both the *BioMOBIUSBluetoothManager* and the *ConnectedThread* are removed and the *ProcessingService* cleans the user notification icon on the screen by using `stopForeground()` to indicate no sensor is recording at the moment.

2.4. UIs of the BAN Hub prototype

The UIs of the BAN Hub is basically consistency to their design in Section 6 of Chapter IV. The Figure V-6 depicts the activities of the BAN Hub prototype. Figure V-6a is the *MainActivity* UI as well as the main screen of the Android app. On the upper part of the screen there is a list with all reachable sensors around, where the connected one is noted. The lower part shows the cached data files on the smartphone, which is ready to be uploaded when the network connectivity allows. There is a status bar on the top of the app indicating the connection status and the cached file sizes. As a reminder, a green Android icon appears on the notification bar of the smartphone indicating there is at least one sensor recording at the moment. The sensor list entries can be started simultaneously when multiple sensors are needed for the recording.

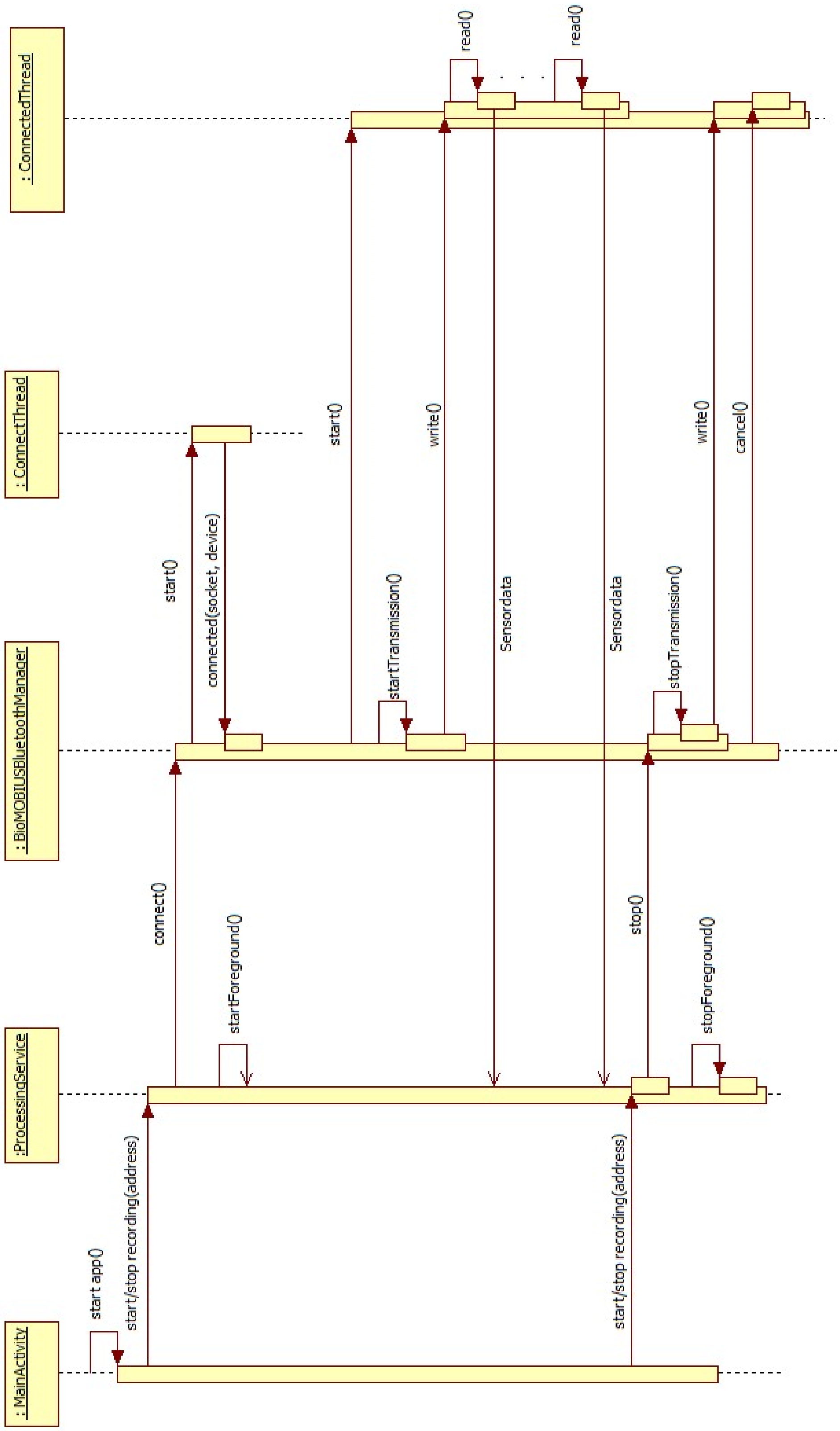
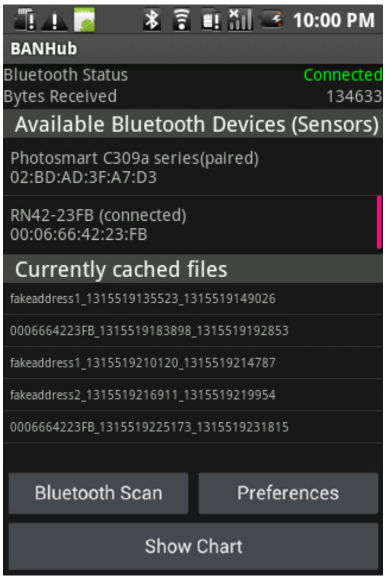
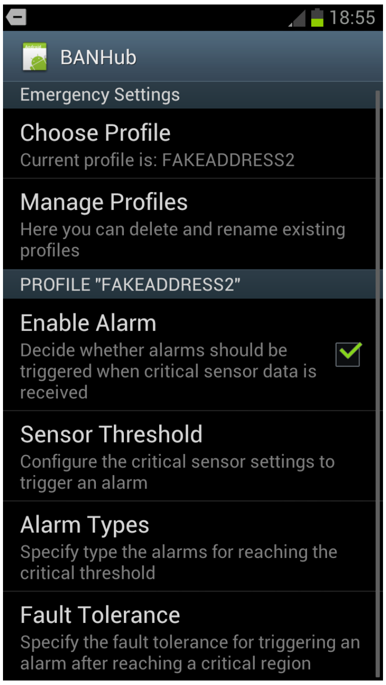


Figure V-5. UML2.0 sequence diagram of the BAN Hub prototype

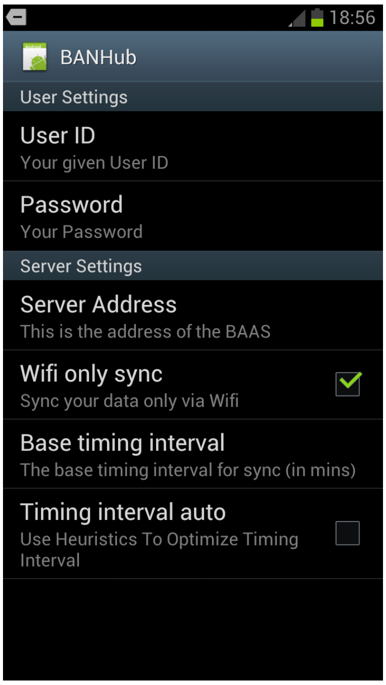
Besides the clicks on the sensor entry for starting/stopping recording, long press on a sensor entry will redirect the user to an alarm function screen, as shown in Figure V-6b. In the alarm setting page, predefined alarm functions can be used by importing an existing profile, or the user is allowed to customize his/her own parameter settings. This alarm function is supposed to be used only by the HCPs in telemonitoring environments before the devices are issued to the patients, with the aim of detecting potential life threatening events. For epidemiological studies this function will never be used.



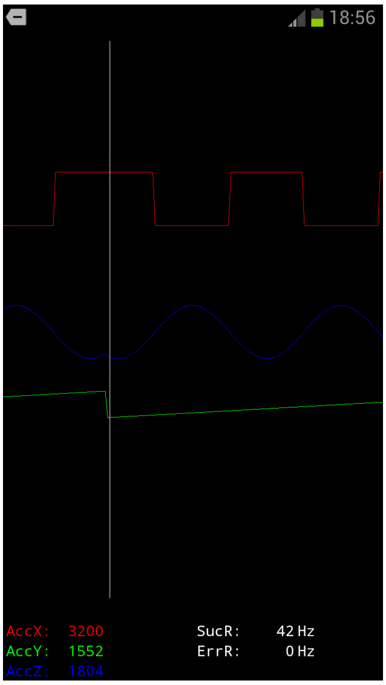
(a) Main UI with sensor connected



(b) Alarms function



(c) Preference UI overview



(d) Visualization UI

Figure V-6. UIs of the BAN Hub

The preference activity is interpreted as the UI in Figure V-6c when a user clicks on the "Preferences" button. All system configuration related parameters can be set here including the user access control information. An entry called "Base timing interval" should be emphasized as it is the answer how to realize the shift between telemonitoring and epidemiological services according to the usage purposes of the application. When "real-time" streaming is needed for telemonitoring, the interval can be set for short period of time such as synchronization is due every minute. Otherwise, it can be set as 24 hours or even longer for epidemiological studies. Similar to the alarms, this preference setting should be done by the HCPs/RESs before assigning the devices to the patients/subjects.

The visualization function is used to motivate the patient as means of feedback by clicking the button "Show Chart". Figure V-6d depicts the real-time visualization of the incoming data in three axis when connecting to a fake sensor which generates random data. This function is only available for the patients under telemonitoring when necessary, and is never supposed to be used by any subject in an epidemiological study. As decoding occupies much computational capacity and memory of the smartphone, the running speed of the entire app would slow down if this function is activated.

In summary, the BAN Hub prototype provides a solid basis for further development. The file oriented approach ensures the system the possibility of integrating more types of sensors. Another advantage is the session oriented method which supports more sensors recording at the same time. Decoding is only on demand when visualization is necessary, which means a faster running app and less power consumption.

3. BAAS Prototype

The BAAS prototype is realized by the Django web application framework (version 1.3), in which the web applications are written in Python programming language.

3.1. Implementation overview

Django separates the interpretation of the data model, the business logic and the user interface as clearly as the MVC does. In contrast to the MVC, it has a different convention for the three concepts. The part implementing the business logic is called *View* in Django instead of *Controller* in the MVC, whereas the part implementing the user interface is called *Template* instead of *View*. Normally more than one HTML templates can be generated by one *View*. Thus, it corresponds to "Model-Template-View" in Django comparable to "Model-View-Controller" pattern.

The BAAS prototype is implemented as a Django app, which dispatches the request based on the URL and generates responses correspondingly. Every Django app incorporates some basic Python modules (source files) as shown in Figure V-7.

For example, Django handles an incoming request as follows: when a HTTP request is received, the Django framework uses the `url.py` to distinguish which function of the `view.py` should be called based on the requested URL. Then the function passes the HTTP request as a Python object and generates a response based on the `models.py` and `forms.py`. The response can be rendered by one HTML template and passed to the web server and other controlling logic by the Django framework. Figure V-7 also depicts the plugin modules used by the plugin scheme in the BAAS, which is used to generate a response to a request by a BAAS view function [94].

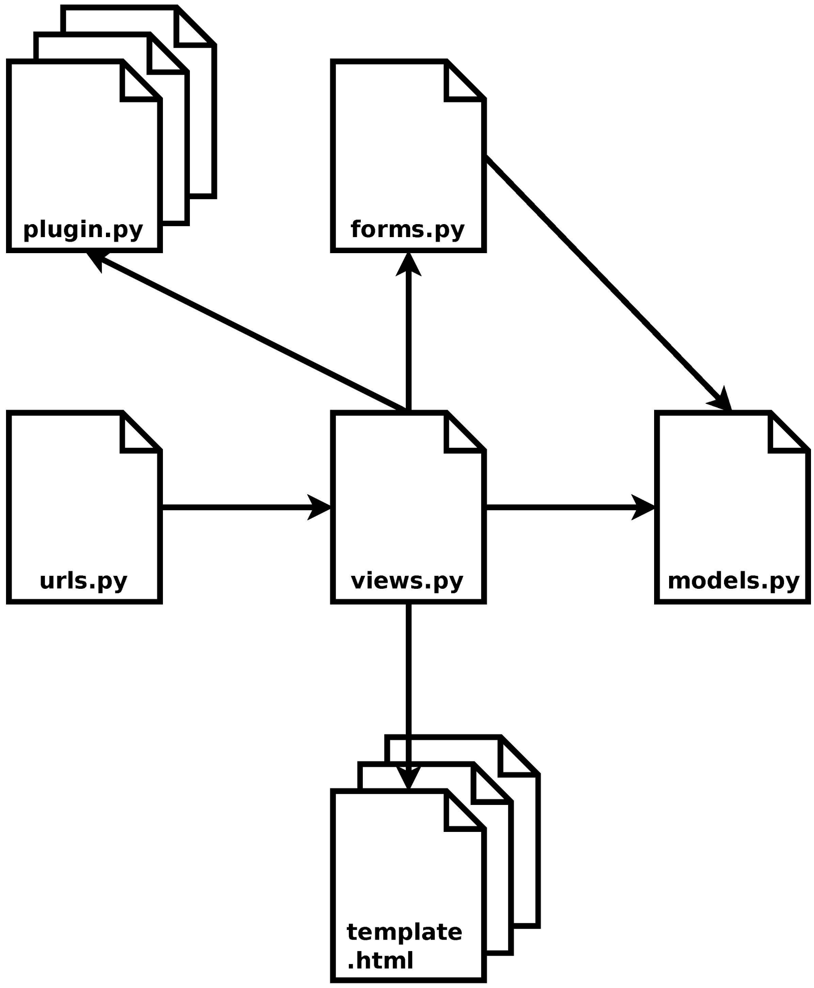


Figure V-7. File dependencies in the Django framework [3]

Models This corresponds to the *DataModel* component in the BAAS design in Section 4 of Chapter IV. The data model of the BAAS is implemented in the `models.py` module with the object-relational mapping (ORM) offered by the Django framework instead of SQL. A subclass of the `models.Model` defines an entity type in the Django ORM and the class member variables that are assigned with Django field types determine the entity type attributes. The class representing the entity directly maps to a database table with respective specifics which may creates relations to the other tables by using specific field types for many-to-many or foreign-key relationships. The implemented data model can be used by any other modules of the BAAS written in Python language without SQL. Creating, saving or querying can be achieved via *Manager* which is by default obtained by the class variable `objects`. For instance, in

[3]Figure from Horsch et al. [94]

order to save or change a user ID, a query for the user information with the primary key (pk) A7Z6J8RVfg is made and the returned value is saved to the `user` variable like this:

```
user = User.objects.get(pk="A7Z6J8RVfg")
```

Views They are the kernel of the Django framework. According to the naming convention, *Views* and the entire Django framework map to the *MainController* component in the BAAS design. They are simply Python functions defined in the `views.py` module of the BAAS responsible for generating a response to a request depending on the parameters extracted from the URL or sometimes depending on the payload of the request (for POST request). A *View* relies heavily on the underlying data model, namely the classes defined in the `models.py` to generate the response. Each *View* in the BAAS has to be able to handle both HTML and REST requests for its URL patters which are normally passed with the `requesttype` parameter. Besides, a *View* should also distinguish the HTTP request methods (GET, POST or DELETE) that its URL pattern supports. A generic URL defined in the `urls.py` looks like this:

```
http://example.org/baas/(rest|html)/<resource-type>/<id>
```

In order to guarantee the user access control mechanism, all *Views* have to deal with the user authentication issues. This is achieved differently depending on the request types. For HTML requests it is realized through the session -middleware offered by Django and the `login` *View* saves the verified `user` entity into the `request.session` object, which can be used compatible with cookies persistently afterwards until the user logs out either by himself or due to time expiration. For REST requests the user credentials have to be verified every time along with any individual request which is sent in terms of GET or POST parameters. The decorator that handles this functionality for all *Views* except the `login` *View* is `requires_https` which ensures the request is made via HTTPS for security reasons.

Templates In the Django framework, templates are written in a mixed style of normal HTML together with a special syntax to present the responses to the HTML requests as displayable HTML pages in the client browser. It builds up the web pages hierarchy by using a basic *Template* at the root for common parts and a renderer for generating request specific contents for each *View*. The results of *Templates* collection are passes as output to the *UI* component in the BAAS design. The `user` object and the `expiry_date` are two necessary parameters that must attach to each rendering of any individual *Template*.

Forms Creating a new entity of certain resource by a POST request can be easily solved by using *Forms* in the Django framework. *Forms* are defined in the `forms.py` module either manually or by referencing directly a model class from `models.py` [94]. The latter brings great convenience to the system implementation. Once the model changes, the corresponding form will automatically change as well.

3.2. Plugin scheme

The plugin scheme of the BAAS is a special part to make the capability of data processing in the BAAS easily extensible. The plugins containing a class `Plugin` are standalone modules which provide two functions for the BAAS to access. Generally, a plugin should be dynamically imported in the BAAS Python modules by the Python `imp` module and instantiate the `Plugin` class of it before being used. Then the two functions `get_plugin_description` and `process()` can be called in a normal way [94].

`get_plugin_description` is used to register the plugins to the system by the BAAS `plugins-View`. Its return value is a Python dictionary containing three key value pairs: `plugin_name`, `plugin_description` and `supports`. These three key value pairs together describe a specific plugin in details by its name, the function it provides as well as the biosensor types it supports. Every time before rendering the plugins web interface page, the `plugins-View` scans the plugin folder for any not installed plugins files. If any of them is found, the administrator of the system may choose from the list for installation using the `get_plugin_description` by the `plugins-View`.

`process()` conducts the processing task of the plugin through parameters. Normally, at least one parameter is needed, namely an object of the type `BiosensorData` defined in `models.py`. The plugin gets all the metadata associated with the biosensor data such as the timestamps and biosensor ID. Its return value is also a Python dictionary containing four key value pairs: `status`, `result_type`, `result_format` and `result`. The four define the processing success/failure and the processing result by its type and format. In the current BAAS implementation, there are two `result_type` that have been defined, which is possible to be extended with more types:

`feature` declares that the result should be some aggregated values, e.g. mean values of the biosensor data.

`timeseries` declares that the result should be a series of values with associated timestamps.

The plugin functionality can be accessed through the web page showing biosensor data details. Once a specific plugin is selected and conducted, The ID of the chosen plugin is parsed and issued to an asynchronous Javascript. The result is received as JSON string and to be plotted by the `flot.js` library in an appropriate but generic way as the JSON response includes the metadata of the data type and format. In order to reduce the plotting complexity, a maximal number of points to be plotted has been defined in the script.

3.3. UIs of the BAAS prototype

The UIs of the BAAS are consistent with their design in Section 6 of Chapter IV. Figure V-8 describes the web page for managing the biosensors in the BAAS. It is rendered by the `biosensors-View` based on the `biosensors.html` template. The upper part of the web page provides the possibility of registering new biosensors to the system as well as creating new sensor types with corresponding descriptions. The lower part shows a list containing the already registered biosensors and their details. A HCP/RES is able to assign one or more biosensor devices to his patients/subjects through the list.

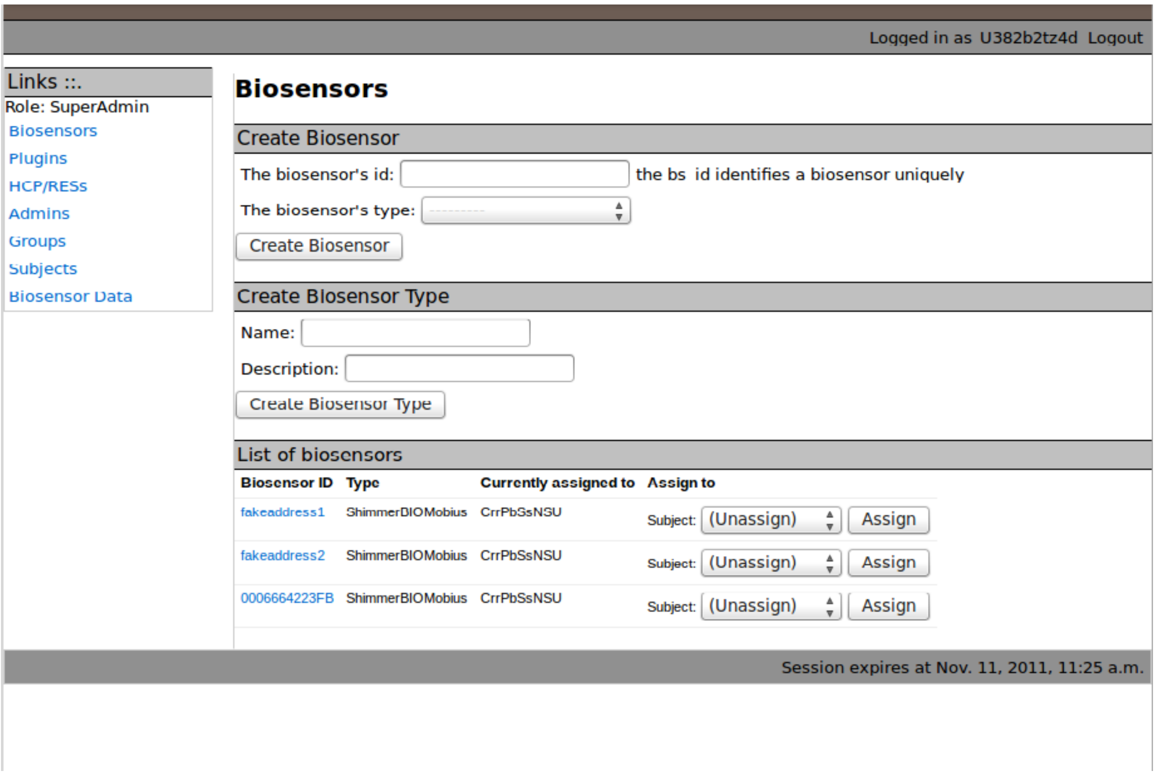


Figure V-8. The UI of the BAAS biosensor page

Another example is the biosensor details web page with data processing options as shown in Figure V-9. The upper part of the page shows all metadata of a specific data file including timestamps, file path and file size, etc. There are data processing options listed on the lower half page realized by the plugins. The processing is conducted by selecting an appropriate plugin which supports the type of the data file. The example plugin in the figure achieves the visualization of the biosensor data from the file with the `return_type timeseries`, which represents the sensor data in 6 values for each time spot.

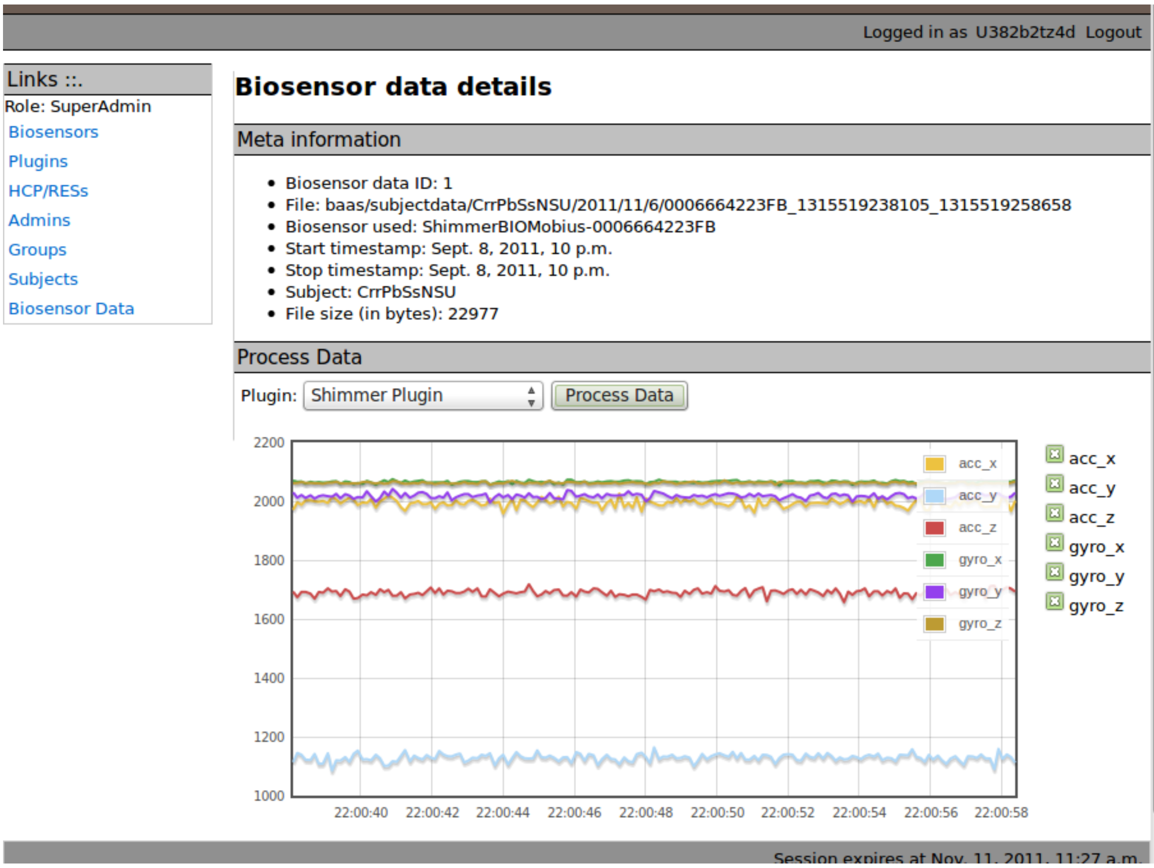


Figure V-9. The UI of the BAAS biosensor details page with plugins

4. Future work

The previous sections presented a prototype of the SensCoAP system, but not all functions identified in the system requirement analysis and the use cases were implemented. This section summarizes the non-implemented features as a reference for future improvement of the SensCoAP prototype. It follows the partition of the subsystems and their communication.

On the BAN Hub

The first missing feature is the decoding of sensor data in real-time. It has already been identified in the BAN design. Although the prototype provides a preliminary visualization function on the smartphone, it is only compatible with Shimmer sensors. Decoding data of more sensor types is necessary in future development.

Another missing feature is to require feedback regarding his own health data of the user from the BAAS directly on the BAN Hub. This would currently be done by using a browser on the smartphone. However, this function should definitely bind to the BAN app in future by the REST interface the system supports, in order to ensure the convenience of feedback requests.

On the BAAS

One current limitation of the BAAS is that there is no automatic processing of the incoming data yet, and therefore no alerts can be triggered for abnormal data under telemonitoring environment. As describe in the Scenario III in Chapter III, the HCP is able to set thresholds on the BAAS UI defining what are considered emergency data, i.e. data which might indicate a patient is now facing a possibly life threatening condition. This function can be realized by the visualization plugin implanted into the framework which interprets the raw data as graphical curves. In this case, the existing visualization plugin should be improved to present the real-time data instead of the individual data files. That means the presentation would be dynamically refreshing curves and the output can be defined in a generic data format.

A limitation regarding the maintenance is the inability to remove the plugins as well as to deactivate the users. Adding a corresponding flag into the user data model as an extra attribute can solve this problem.

It is not allowed to add new roles to the system, which can be extended further for wider usage. This issue has been discussed in the system design (see Section 5 in Chapter IV).

Communication between the subsystems

An interesting option to improve the communication performance between the BAN Hub and the BAAS is the on-the-fly compression of the data files during transmission which will significantly save bandwidth. In case it uses an expensive mobile network connection, this is an important aspect. However, this modification will require serious changes on the BAAS in order to decompress the data files on the root level, which needs to be considered carefully before implementation.

Chapter VI

Proof-of-Concept

Contents

This chapter presents the proof-of-concept for the SensCoAP prototype. Two rounds of system tests with volunteers are designed for usability and satisfaction, and the collected feedback are documented and analyzed, as evidence of making suggestion for further improvement of the system. An implementation revision is done between the two test rounds, using the output of the first test round, and the updated system is used as input of the second round of system test [1].

Before going into the validation studies, several iterations of unit tests have been conducted to evaluate the system performance. The visualization and alarm functions were tested for over 8 hours (with a Motorola Defy+ smartphone, 1 GHz processor and 512 MB RAM, running Android version 2.3.6.) continuously until the Shimmer sensor ran out of battery. The performance of the BAN Hub application was found to be stable and reliable, with small CPU consumption (on average, 12% for normal functions, 35% for real-time visualization). The alarm function worked properly with rare "false alarms".

The entire system has been tested for some long measurement sessions up to 80 minutes, connecting with 5 sensors (2 Shimmer Accel+Gyro, 1 Shimmer Accel+ECG, 2 fake sensors) simultaneously. The testing environment of the BAN Hub was a Huawei Pulse smartphone running Android version 2.1. It turned out to be no data loss and no data corruption on the fly. The time latency between the BAN Hub and BAAS were less than 2 seconds and could be even much smaller under a higher frequency. The BAAS with plugin scheme responded to a processing request (the dataset sample size was 21.3MByte and contained 1017014 timestamps with six values for each) within 10 seconds on average.

[1] Part of the summaries, numbers and figures listed in this chapter are based on the student work from Jacht V, Horsch J, Schindler F, and Velden L.

This response time can be shortened by using a skipping plotting method in the plugins (already described in Chapter V).

1. Study Design

The study design aims to provide a feasibility study protocol for evaluation of the prototype of the SensCoAP system, in order to acquire user experiences with the system, especially on usability and user satisfaction. More details regarding the study protocol can be found in Appendix B.

The experiments were divided into two phases:

1. system performance (usability, stability, reliability and robustness) test involving young healthy people who are familiar with technology;
2. system acceptance test involving elderly or healthcare personnel.

Phase 1 was designed as a fast test with shorter period of time. The system has accepted a small technical modification after Phase 1 according to the feedback from the users. Then Phase 2 was executed in a more realistic environment with elderly. They were supposed to test the system for longer time, which was close to the reality of telemonitoring and cohort measurements.

In both phases Shimmer sensors with 3-axis accelerometers together with Android smartphones were used. Methods applied included observation, survey and think aloud. The experience on the system collected from the users was analyzed afterwards.

1.1. Study questions

Phase 1

1. Usability and stability
   Is the system working smoothly during the test? Which kind of bugs do appear, if any?
2. Reliability
   How is the reliability of the system from collecting data to successfully uploading data?
3. User friendliness
   Are the user interfaces (smartphone & server) comfortable and easy to use for the participants?

Phase 2

4. Usability
   How is the usability of the system? e.g. the operations on the smartphone and the server UI?
5. Acceptance
   How the participants perceive the system and the sensor positioning?
6. Satisfaction
   How are the satisfaction and acceptance of the prototype among the participants?

1.2. Experiments

The experiments of Phase 1 was conducted from October till December in 2012, and from October till April in 2013/2014 for Phase 2. The participants were mainly recruited from the students and staff at IMSE and Campus Garching of TU München. In addition, staff from collaborating clinics and organizations as HCPs and elderly around as subjects, were also involved in Phase 2. Each participant was equipped with one Shimmer 2R sensor (Triaxial accelerometer, 50Hz sampling rate) and one Android smartphone (HTC one, T-mobile, Samsung S3, Sony Xperia or Huawei Hornor 6, Android version 2.1, 2.2, 4.1 or 4.4) running the SensCoAP app. Laptops/desktops with browsers were used when necessary. Both experimental phases mainly took place in the relevant institutes as well as the participants' home where internet access (WiFi or 3G) was available.

Phase 1

The sample size was expected between 15 to 30, which reached 20 at last. The average age of the participant group should be around 30 (young people). They were supposed to wear the sensor on body (e.g. in the pocket) and take the smartphone with them till the end of the experiment. The experiment lasted approximately 2-3 hours for each participant, and during this period time they were asked to fulfill some operational tasks (see Appendix B) with the SensCoAP system. All the tasks should be finished by the participants alone without any instructions from the study investigators. Participants were encouraged to think aloud when they performed the operations on the user interfaces. Later when the experiments finished, they were required to fill a questionnaire.

Phase 2

The sample size was 11, among which 5 participants were elderly with an average age of 55, and 6 participants were young people with an average age of 30. They were expected to wear the fully charged sensors on body until the battery died. Meanwhile, they had to perform some tasks (see Appendix B) with the SensCoAP system, either as HCPs or subjects/patients. The tasks can be finished by the participants under the instructions from the study investigators if necessary. Both roles were required to fill a questionnaire regarding acceptance and satisfaction at last.

1.3. Variables and attributes

The tests aimed to gather the information of user experience of the system. The following points were measured by questionnaires and observations during the tests. Thus the answers to the questionnaires were critical in the study.

- Age/gender/background (the personal information of the participants)
- The duration of the actual running time of the system (measured by the study investigators)
- The encountered bugs/errors during the test (recorded by the study investigators)

- The ease of use of the system (measured by questionnaires)
- The robustness and usability of the system (measured by questionnaires)
- The acceptance and satisfaction of the system (measured by questionnaires)
- The suggestions of the user interface improvement (measured by questionnaires)

2. Results and Discussion

As a result, 20 participants has been involved in the tests of Phase 1 with an average test duration of 1.5 hours. 11 participants are included in the tests of Phase 2 who has an average test duration of 7 hours approximately.

2.1. Recruitment bias

Due to the limitations of the project, there are some slight recruitment bias on gender, professional background and age.

In Phase 1, among 20 participants 5 are female and the rest of them are male, with an average age of 23.4. Students majoring in computer science constitute almost half of the test group, and the rest of the participants are also own technical background, except one confectioner. Hence, summarily speaking, almost all the participants are technology-friendly people.

In Phase 2, of the 11 participants, 5 are elderly around 55 years old. The other 6 participants are Master and PhD students with an average age of around 30 years old. The technical and educational background of the elderly (2 senior engineers, 1 merchant, 2 workers) varies from "unable to read English (3), unable to use computers and smartphones (1)" to "basic English knowledge (2), familiar with computers and smartphones (4)". The young group members are all students with technical background.

2.2. Technical problems

Figure VI-1 depicts the technical problems during the tests of Phase 1. Most of the technical problems occured during the tests of Phase 1, before the modification between the two phases was implemented. Some problems that occur quite often are proved to be device dependent. For instance, the Bluetooth connection was not stable during the tests, or sometimes the Bluetooth radio was hard to detect when searching for the sensors around. It turns out to be the problem of the old smartphones (with Android 2.1) used in the tests. These devices brought some other problems to the tests: poor response of the touch screen on the smartphones, unable to boot properly and WLAN function not available on the smartphone. All of these problems may be conquered by changing the smartphones.

Another two problems exposed during the tests of Phase 1 are the BAAS server related. As the server is physically located at IMSE, the reachability of the server sometimes depends on the maintenance plan of the entire IMSE servers. This may cause the unavailability of the server and the upload failure of the datasets.

In Phase 2, the problems caused by the old smartphones still existed. But no other technical problems occured in the tests of Phase 2.

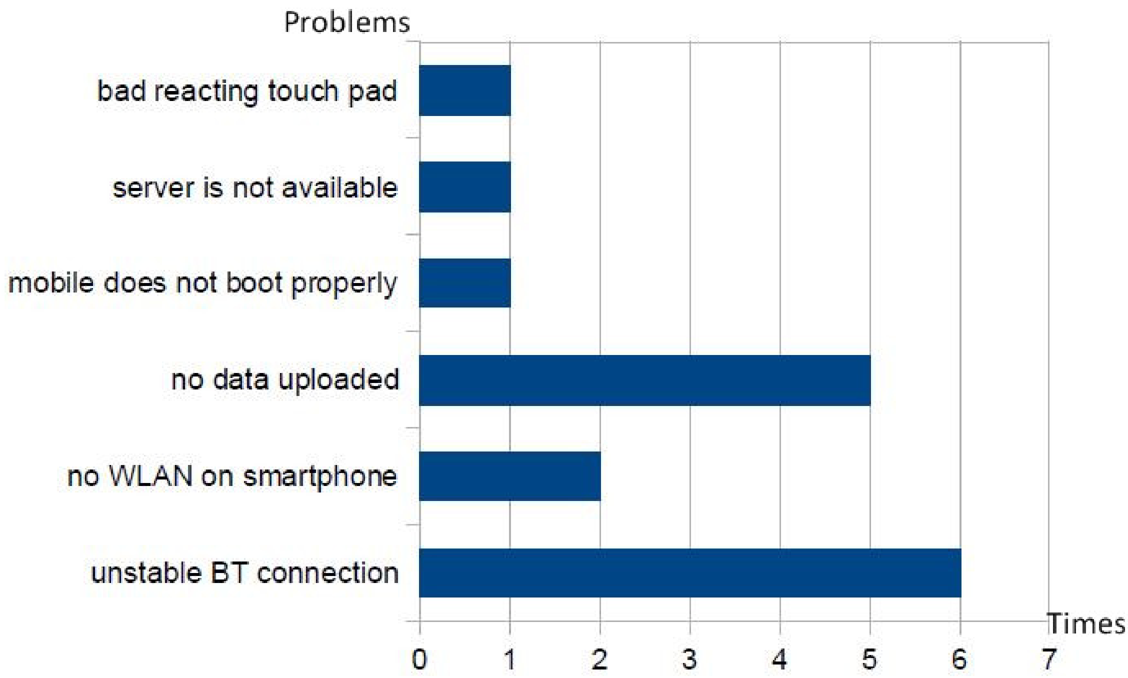


Figure VI-1. The technical problems that occurred during the tests of Phase 1 [2]

2.3. Feedback from the questionnaires

It should be noticed that different age groups have their own preferences, which will influence their acceptance of the SensCoAP system afterwards. The young group gets used to take the smartphones and other miniature electronic products like the sensors with them all day long, while the elderly (especially those old people who never use the smartphones) are inclined to take nothing. This reason directly results in the high acceptance of the SensCoAP system among the young group and much lower acceptance among the elderly group.

Except the preference survey, the questionnaires also provide the participants' opinions on the system usability, especially on the appearance and interaction design (user interfaces and operations). There is a summary of these suggestion listed in the next section. The main problem relies on the text-based UIs of the BAN Hub and the BAAS. It is said that both UIs are more like debug mode, where the users have to read long sentences without any intuitive symbols. The comprehensibility of the UIs influences the difficulty of a task operation. This is why most participants considered the task operations on the smartphones and the web browsers as difficult. Other suggestions e.g. on the layout of the UI are documented into the improvement lists in the next section, which provides a development reference for the next version in future.

As a result, it can be concluded that the SensCoAP system basically fulfills its functions, and most of the participants are able to perform the specific tasks with the system without

[2]Figure reproduced from Velden. [113]

any help. However, an improvement to the system is definitely needed before starting the acceptance tests of Phase 2.

In Phase 1, it is mentioned by most young participants that they are worried such a system is too complex for the elderly to use. Figure VI-2 shows a rating diagram of some questions answered by the young and elderly groups in the questionnaires of Phase 2. The participants rate the system according to their experience of interactions (tasks performed which can be found in Appendix B) with the scale 1-5 (1 for disagree, 5 for agree) and their average rates are calculated respectively for each group. It is clear that in the acceptance tests of Phase 2, the elderly group has a good acceptance towards the SensCoAP system, especially for those who are familiar with smartphones. Even for those elderly without any technical knowledge, they are not strongly against the SensCoAP system under the clinical observation mode, when interactions between the users and the system are not required.

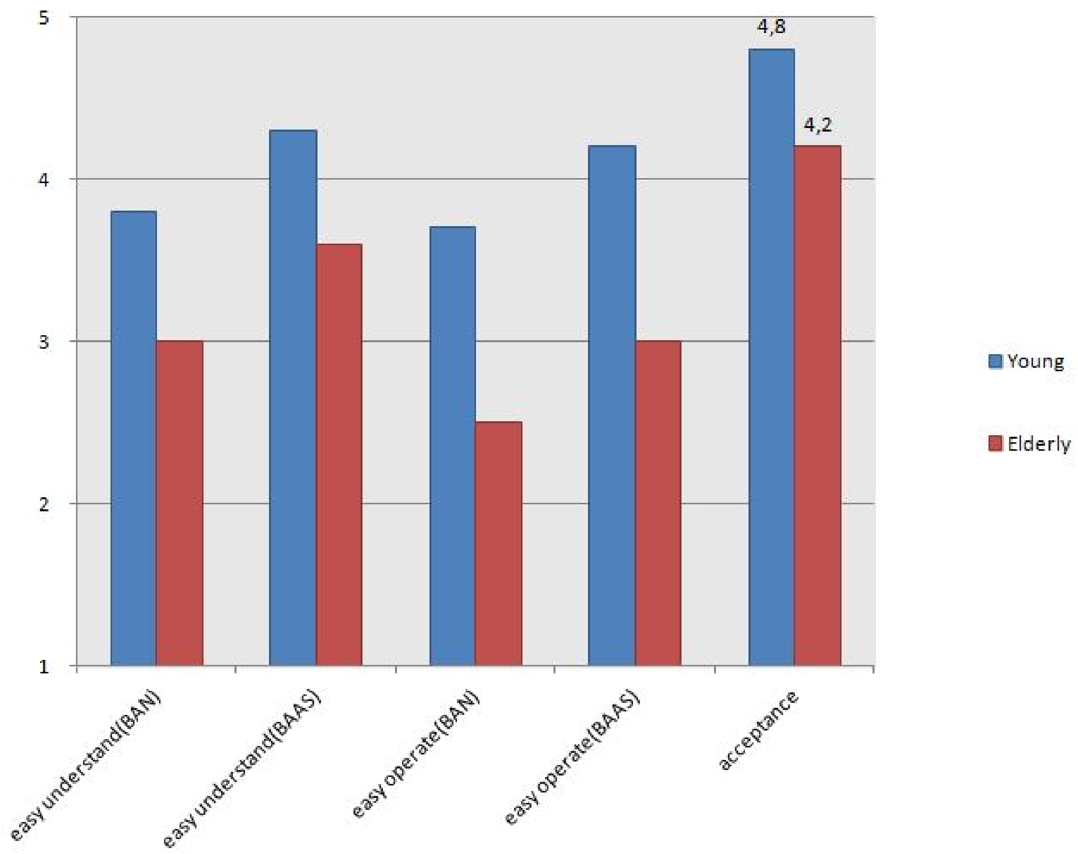


Figure VI-2. The rates for 5 critical questions by young and elderly groups in Phase 2

2.4. SWOT analysis

A SWOT analysis is performed, focusing on the features of the SensCoAP system on the basis of the 2-round system tests. The strengths, weaknesses, opportunities and threats of the prototype are summarized, in order to provide an intuitive impression of the SensCoAP system towards the users and its peer projects.

Figure VI-3 describes the internal abilities (strengths and weaknesses) of the SensCoAP prototype (current version), and lists the external factors (opportunities and threats). It clearly locates where the prototype is and how far it should go in future. The strength of this prototype mainly relies on its novel architecture, and the weaknesses are all revealed in the tests. The opportunities of the SensCoAP development are very optimistic, which has been stated before in the project motivation. And because it is not a commercial project, it faces rare threats from outside environment. For the same reason, no more market strategies (SO, WO, ST or WT) are derived from the analysis.

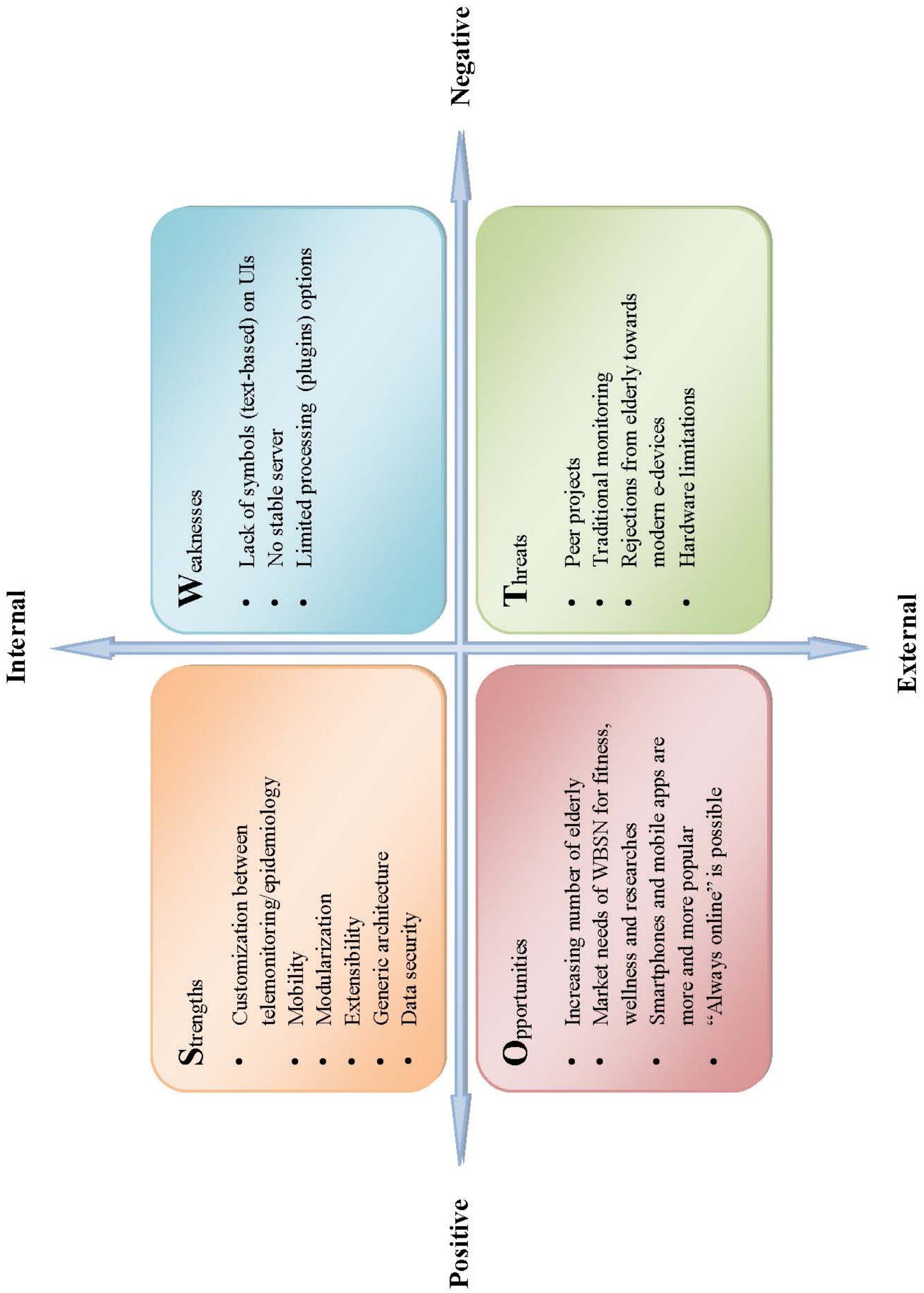


Figure VI-3. SWOT analysis of the SensCoAP prototype

3. Improvement Suggestions Derived from Study Results

This section documents the feedback collected from the system tests. Some improvement suggestions after Phase 1 has already been accepted as the rationale of the implementation modification between the two rounds, and the list after Phase 2 remains to be the basis of future development e.g. to meet the special needs of the elderly or reduce the complexity of the application.

List 1

For Android application

- There should be default setting for some necessary inputs of user preferences.
- To enhance the user experience, the system should be more tolerant to input errors: e.g. allow IP addresses without "http".
- Menu items of the "Preferences" should be more intuitive, or add some explanations under each item for better understanding.
- An automate check for the inputs should be done before saving the settings.
- Renaming "Preferences" to "Settings" might be more intuitive.
- The current UIs looks like for debugging and technical purposes only. It should be designed more appealing for normal users.
- More language versions should be provided to those elderly who are not familiar with English.
- Font sizes and interface elements should be larger and the possibility to enlarge the screen should be enabled. By focusing only on important and actionable interface elements the complexity of the application can also be reduced.
- The Bluetooth connection is better to establish once the app is running. The non-relevant sensors should not appear in the device lists to make it clear to the users.
- The biosensors Bluetooth IDs should be replaced by human-friendly names for easy operations.
- The scroll bar should be visible beside the biosensor list.
- The number of the transmitted bytes should be hidden to keep the main screen simple.
- Acknowledgments of critical events e.g. Bluetooth connection fails, upload success should be provided on the screen in order to trace the status of the Android app.
- Identifiers and headlines like "Currently cached files" could be more informative.
- Help information should be provided in the app.

- Selecting and starting the sensors on the biosensor list should be highlighted and done separately by different operations.
- The pop-up "Service created" message should be omitted.
- An error log is necessary to collect bugs information from the users and sent back to the developers.

For web services

- A DNS name should be used instead of a numeric IP address, as it is easier to memorize and handle for humans and makes the application more serious.
- It is more common to use the label "User name" instead of "User_id" on the login screen.
- The random generated user IDs is not suitable to be used as the user names. The IDs should be encapsulated by human-friendly names for the users.
- Instead of a empty start screen, important functions or some explanation text can be placed there.
- The Web front-end should follow modern design principles, which, for example, can be easily achieved by using Bootstrap.
- The role of the user should be displayed along with the user account, as it is a part of the account itself.
- The log out button on the right is hard to find.
- A clearer structure and hierarchy of functions and elements on the UI should be pointed out, so that it is clear for the users what they can do by using the webpages.
- After assigning a subject to a group the assign button should keep the name or respectively the ID of the assigned subject and not jump back to "Unassigned".
- "This field is required" alert is shown twice although needed only once while creating a new biosensor. The alert also occurs when an admin wants to create an HCP and forgets to enter a password, but the alert does not indicate that it refers to the password field.
- Detailed views should include all information that is already available on the overview, e.g. "Biosensor Details" does not list the type of the biosensor, although that is provided on the "Biosensors" overview.
- Short names for data files instead of the whole URL make it more readable and linking that name and not the ID to the file content is also more intuitive.
- The new items in the lists should be highlighted.
- Lists should be sorted by e.g. alphabetical or chronological orders and the sort filters can be changed manually.

- The default list sorting should be consistent on different pages, e.g. always ascending.
- Long lists, especially the data file lists, make it hard to find the right items. It might be useful to place new or recently used files separately. Pagination is also a possibility to handle long lists if only a few items are usually interesting.
- Axes and curves on plots should be better explained and better distinguishable colors for the curves would increase the readability.
- An "Add" button under the "Groups details" view would make it easier to add a subject to that group.
- Descriptions like a "Plugin Description" should be more expressive.
- It would be great if assigning the sensors to subjects is also available on the subject view under user details.
- It might be better to have more distinguishable and human-friendly biosensor IDs.
- The order of first "Password" then "name" when creating a new HCP is unusual. It should be reversed to make it conventional.
- Commas at the end of lists should be removed and UI elements should be aligned.
- Passwords should be changeable. And the initial passwords should be generated randomly during account creation instead of setting by the administrators or the HCPs.

List 2

For Android application

- For multifunctional biosensors, the users have the option to choose desired functions.
- Font size is quite critical for elderly, but the smartphone usually has limited screen size. Tablet is not realistic because it is too big to carry everywhere.
- Some smartphones are too complicated for the very old elderly. Maybe a simpler Android version should be developed for those.
- Bluetooth is not always reliable. Other communication channels should be involved into the system as alternatives.
- Biosensors and smartphones might get lost. So the system should have a functionality to trace the lost sensors and locked them for data privacy.
- Biosensors with longer battery life should replace the equivalent biosensors.
- Mobile networks are not always reliable. Other communication method between the smartphones and the server should be discovered for outdoor mode.
- The current app UI should be replace with more intuitive UI e.g. with icons and graphs so that it is easier for the elderly to understand.
- More healthcare services from the healthcare centers should be included via the system.
- Depending on the sensor size, if the sensor is too big, it will be annoying to wear it all day long, esp. for women.
- The system can be merged with other necessary apps in life, e.g. taxi, food delivery, etc.
- If necessary, the patient data should also be shown on the smartphone when emergency happens.
- The appearance of the biosensors should be modern and attractive, from the fashion perspective. It may integrates with other accessories e.g. watch, necklace, jewelry.
- If possible, the biosensors may be integrated with the smartphones so that the patients do not take both with them.
- The Bluetooth connection should be fixed by the HCP once and never bother the patients to recover it.
- A tutorial (use guide) of the system should be available.
- The sensor should be waterproof so the patients do not have to take it off when taking a shower.

For web services

- Server should be more stable, without any crash.
- The current web UI should be replace with more intuitive UI e.g. with icons and graphs so that it is easier for the elderly to understand.
- More healthcare services from the healthcare centers should be included into the web service via the system.
- A tutorial (use guide) of the web service should be available.
- Its dependence on Internet and electricity might be a potential problem.

Chapter VII

Conclusion

Contents

1. Summary

The entire thesis aims to give an answer to the research question:

> **How to design a multimodal biosensor data acquisition and feedback system for both medical telemonitoring and epidemiological research, and make it customizable for both types of services?**

Due to technological progress, WBSN systems in various application fields start to increase tremendously. Nevertheless, a large proportion of redundant and incompatible framework exists. To make things worse, a lack of standardization on validations of the WBSN systems leads to the reliability and robustness of these systems still remain as questions.

This thesis, with the steps which follows a typical "waterfall" software development life cycle, has proposed a generic design with high flexibility to provide telemonitoring/epidemiological services. After collecting requirements from potential users, the SensCoAP is designed to be generic, compatible, modular and flexible. It contains a three-layer architecture connecting the biosensors and the BAAS via a BAN Hub, in order to collect, forward, store and analyze the physiological data from multiple biosensors in one mutual platform. Moreover, it is able to be customized for telemonitoring ("real-time") and epidemiological ("store & forward") services.

A two-round system validation involving 31 volunteers has proven that the SensCoAP prototype functions smoothly and its usability fulfills all expectations towards its functionalities. A small modification has been carried out between the two test phases. The

SensCoAP prototype was well accepted by users of different ages. The high acceptance indicates that the SensCoAP prototype can be considered one possible answer to the research question.

2. Future work

Though the WBSN systems are now facing a period of rapid development, some factors in the design still limit its capacity and usages, such as battery life, power consumption, security, etc. These challenges that are met by almost every developer has been discussed in Chapter II.

The SensCoAP system also has the limitations of these technical concerns. Currently, the battery of the SensCoAP prototype does not live long enough (8 hours for Shimmer 2R sensor, and less than one day for all smartphones). Breaking the battery life barrier will definitely bring a new revolution to the technology and motivate more inspiration and ideas to the design of the SensCoAP in further development. Also due to the hardware limitation, the communication protocol is the normal power-consuming Bluetooth version instead of Zigbee or low-power Bluetooth, which will be changed with the update of the smartphones. As mentioned in Chapter V, the vendor openness is another factor to hinder the compatibility of the SensCoAP prototype. It is hoped that one day SensCoAP is able to involve as many sensors as possible without any worry of the proprietary of the sensor techniques. HTTPS should be activated on the BAN Hub as well (currently only on the BAAS side) for more secured data transmission.

Besides, the SensCoAP prototype is only a vertical slice of the entire design. As documented in Chapter V, there are still missing features which are supposed to be implemented in the next version. And the feedback got from the system validation tests (the two suggestion lists) can also be referred in the further development of the SensCoAP system.

Bibliography

[1] Lo B, Yang GZ. Body sensor networks-research challenges and opportunities. 2007;Available from: http://digital-library.theiet.org/content/conferences/10.1049/ic_20070541.

[2] Jacobzone S, Oxley H. Ageing and health care costs. Internationale Politik und Gesellschaft. 2002;(1):137–156.

[3] Chen C, Knoll A, Wichmann HE, Horsch A. A Review of Three-Layer Wireless Body Sensor Network Systems in Healthcare for Continuous Monitoring. Modern Internet of Things. 2013;Vol. 2 Issue 3.

[4] Goulding MR, Rogers M, Smith S. Public health and aging: trends in aging - United States and worldwide. MMWR Morb Mortal Wkly Rep. 2003;52(6):101–106.

[5] Brockmann H, Gampe J. The cost of population aging: forecasting future hospital expenses in Germany. Max Planck Institute for Demographic Research, Rostock, Germany; 2005.

[6] Architecture Concept of a Wireless Body Area Sensor Network for Health Monitoring of Elderly People; 2007. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4199235.

[7] Seto E. Cost comparison between telemonitoring and usual care of heart failure: a systematic review. Telemedicine and e-Health. 2008;14(7):679–686. Available from: http://online.liebertpub.com/doi/abs/10.1089/tmj.2007.0114.

[8] Meystre S. The current state of telemonitoring: a comment on the literature. Telemedicine Journal & e-Health. 2005;11(1):63–69. Available from: http://online.liebertpub.com/doi/abs/10.1089/tmj.2005.11.63.

[9] COMPETITIVENESS CO. Highway to health: transforming US health care in the information age. Washington, DC Mar. 1996;.

[10] Dobkin BH. Wearable motion sensors to continuously measure real-world physical activities. Curr Opin Neurol. 2013 Dec;26(6):602–608. Available from: http://dx.doi.org/10.1097/WCO.0000000000000026.

[11] Laporte RE, Montoye HJ, Caspersen CJ. Assessment of physical activity in epidemiologic research: problems and prospects. Public health reports. 1985;100(2):131. Available from: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1424723/`.

[12] Steele BG, Belza B, Cain K, Warms C, Coppersmith J, Howard J. Bodies in motion: monitoring daily activity and exercise with motion sensors in people with chronic pulmonary disease. J Rehabil Res Dev. 2003;40(5 Suppl 2):45–58.

[13] Warren JM, Ekelund U, Besson H, Mezzani A, Geladas N, Vanhees L. Assessment of physical activity–a review of methodologies with reference to epidemiological research: a report of the exercise physiology section of the European Association of Cardiovascular Prevention and Rehabilitation. European Journal of Cardiovascular Prevention & Rehabilitation. 2010;17(2):127–139. Available from: `http://cpr.sagepub.com/content/17/2/127.short`.

[14] Brage S, Brage N, Franks P, Ekelund U, Wareham N. Reliability and validity of the combined heart rate and movement sensor Actiheart. European Journal of Clinical Nutrition. 2005;59(4):561–570. Available from: `http://www.nature.com/ejcn/journal/v59/n4/abs/1602118a.html`.

[15] Buskirk E, Harris D, Mendez J, Skinner J. Comparison of two assessments of physical activity and a survey method for calorie intake. The American Journal of Clinical Nutrition. 1971;24(9):1119–1125. Available from: `http://ajcn.nutrition.org/content/24/9/1119.short`.

[16] MORRIS JN, HEADY JA, RAFFLE PA, ROBERTS CG, PARKS JW. Coronary heart-disease and physical activity of work. Lancet. 1953 Nov;265(6796):1111–20; concl.

[17] Chong S C Y ; Kumar. Sensor networks: evolution, opportunities, and challenges. 2003;91(8):1247–1256. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1219475`.

[18] Rodenas-Herraiz D, Garcia-Sanchez AJ, Garcia-Sanchez F, Garcia-Haro J. Current trends in wireless mesh sensor networks: a review of competing approaches. Sensors (Basel). 2013;13(5):5958–5995. Available from: `http://dx.doi.org/10.3390/s130505958`.

[19] Dobkin BH, Dorsch A. The promise of mHealth: daily activity monitoring and outcome assessments by wearable sensors. Neurorehabil Neural Repair. 2011;25(9):788–798. Available from: `http://dx.doi.org/10.1177/1545968311425908`.

[20] Machado NWGTS R ; Ansari. Adaptive density control in heterogeneous wireless sensor networks with and without power management. IET Communications. 2010;4(7):758–767. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5454245`.

[21] Larios DF, Barbancho J, Sevillano JL, RodrÃguez G, Molina FJ, Gasull VG, et al. Five years of designing wireless sensor networks in the Donana Biological Reserve (Spain): an applications approach. Sensors (Basel). 2013;13(9):12044–12069. Available from: `http://dx.doi.org/10.3390/s130912044`.

[22] Free C, Phillips G, Galli L, Watson L, Felix L, Edwards P, et al. The effectiveness of mobile-health technology-based health behaviour change or disease management interventions for health care consumers: a systematic review. PLoS Med. 2013;10(1):e1001362. Available from: http://dx.doi.org/10.1371/journal.pmed.1001362.

[23] Royce WW. Managing the development of large software systems. In: proceedings of IEEE WESCON. vol. 26. Los Angeles; 1970. .

[24] Wireless Sensors Networks based monitoring: Review, challenges and implementation issues; 2008. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4757163.

[25] Aziz O, Lo B, Pansiot J, Atallah L, Yang GZ, Darzi A. From computers to ubiquitous computing by 2010: health care. Philos Trans A Math Phys Eng Sci. 2008 Oct;366(1881):3805–3811. Available from: http://dx.doi.org/10.1098/rsta.2008.0126.

[26] Lo B, Yang GZ. Key technical challenges and current implementations of body sensor networks. In: Proc. 2nd International Workshop on Body Sensor Networks (BSN 2005); 2005. .

[27] Physiological information acquisition through wireless biomedical sensor networks; 2005. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1635137.

[28] Understanding the Mobility Model of Wireless Body Sensor Networks; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4097948.

[29] Pantelopoulos A, Bourbakis NG. A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis. 2010;40(1):1–12. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5306098.

[30] Sung M, Marci C, Pentland A. Wearable feedback systems for rehabilitation. J Neuroeng Rehabil. 2005;2:17. Available from: http://dx.doi.org/10.1186/1743-0003-2-17.

[31] Tura A, Badanai M, Longo D, Quareni L. A medical wearable device with wireless bluetooth-based data transmission. Meas Sci Rev. 2003;3:1–4.

[32] Trinugroho YBD. Service-Oriented Architecture for Patient-Centric eHealth Solutions. University of Agder; 2014.

[33] A New Wireless-Type Physiological Signal Measuring System Using a PDA and the Bluetooth Technology; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4238012.

[34] Anliker J, Ward JA, Lukowicz P, Troester, Dolveck, Baer, et al. AMON: a wearable multiparameter medical monitoring and alert system. 2004;8(4):415–427. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1362650.

[35] The myheart project - fighting cardiovascular diseases by prevention and early diagnosis; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4030648.

[36] Weber JL, Porotte F. Medical remote monitoring with clothes. In: Int. workshop on PHealth, Luzern, Switzerland; 2006. .

[37] Hao Y, Foster R. Wireless body sensor networks for health-monitoring applications. Physiol Meas. 2008 Nov;29(11):R27–R56. Available from: http://dx.doi.org/10.1088/0967-3334/29/11/R01.

[38] Shnayder V, Chen BR, Lorincz K, Fulford-Jones TRF, Welsh M. Sensor networks for medical care. Division Eng. Appl. Sci., Harvard Univ.; 2005.

[39] Gyselinckx B, Penders J, Vullers R. Potential and challenges of body area networks for cardiac monitoring. J Electrocardiol. 2007;40(6 Suppl):S165–S168. Available from: http://dx.doi.org/10.1016/j.jelectrocard.2007.06.016.

[40] HealthGear: a real-time wearable system for monitoring and analyzing physiological signals; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1612896.

[41] A wearable ECG-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation; 2005. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1467724.

[42] Philips Healthcare;. Available from: http://www.healthcare.philips.com/.

[43] Polar;. Available from: http://www.polar.com/.

[44] Omeron;. Available from: http://www.omron.com/.

[45] Zephyr, Inc.;. Available from: http://bioharness.com/.

[46] Design and Evaluation of a Wireless Body Sensor System for Smart Home Health Monitoring; 2009. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5425471.

[47] Medical and Home Automation Sensor Networks for Senior Citizens Telehomecare; 2009. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5208093.

[48] WAVE and CalFit — Towards social interaction in mobile body sensor networks; 2010. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5452684.

[49] A lightweight middleware for an e-health WSN based system using Android technology; 2012. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6320312.

[50] Personal Heart Monitoring and Rehabilitation System using Smart Phones; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4124124.

[51] Towards a Novel In-community Healthcare Monitoring System over Wireless Sensor Networks; 2008. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4548253.

[52] Yang GZ. Body Sensor Networks. Springer; 2006. Available from: http://www.amazon.com/Body-Sensor-Networks-Guang-Zhong-Yang-ebook/dp/B004PEIG08%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3DB004PEIG08.

[53] iCare: A Mobile Health Monitoring System for the Elderly; 2010. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5724906.

[54] Rodriguez J, Goni A, Illarramendi A. Real-time classification of ECGs on a PDA. 2005;9(1):23–34. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1402444.

[55] Framework for pervasive health monitoring; 2008. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4570659.

[56] Docksteader L, Benlamri R. MORF: A Mobile Health-Monitoring Platform. IT Professional. 2010;12(3):18–25. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5401151.

[57] Huang YM, Hsieh MY, et al HCC. Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks. 2009;27(4):400–411. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4909279.

[58] An integrated system for wireless monitoring of chronic patients and elderly people; 2011. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6085743.

[59] Medical MoteCare: A Distributed Personal Healthcare Monitoring System; 2009. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4782627.

[60] A WBAN-based System for Health Monitoring at Home; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4201256.

[61] Vital Sign Monitoring System with Life Emergency Event Detection using Wireless Sensor Network; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4178671.

[62] Design and Development of a Wireless Remote Point-of-Care Patient Monitoring System; 2007. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4380373.

[63] TinyOS;. Available from: http://www.tinyos.net/.

[64] Jovanov E, Milenkovic A, Otto C, de Groen PC. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. J Neuroeng Rehabil. 2005 Mar;2(1):6. Available from: http://dx.doi.org/10.1186/1743-0003-2-6.

[65] Hill J, Culler D. Mica: a wireless platform for deeply embedded networks. 2002;22(6):12–24. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1134340.

[66] A distributed wireless body area network for medical supervision; 2012. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6229260.

[67] An open and reconfigurable Wireless Sensor Network for pervasive health monitoring; 2008. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4571044.

[68] Martin ESE, Wang C, Roy M, Shia V, Bajcsy R. Opportunistic Strategies for Lightweight Signal Processing for Body Sensor Networks;.

[69] Yazar D, Tsiftes N, Osterlind F, Finne N, Eriksson J, Dunkels A. Demo Abstract: Augmenting Reality with IP-based Sensor Networks;.

[70] Communication Platform for Biosensor-based Sleep Management Applications; 2006. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4462643.

[71] Research of portable community-oriented health monitoring terminal; 2010. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5554122.

[72] Bluetooth;. Available from: https://www.bluetooth.org/.

[73] ZigBee Alliance;. Available from: http://www.zigbee.org/.

[74] Short-range wireless network and wearable bio-sensors for healthcare applications; 2009. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5373623.

[75] Cypher D, Chevrollier N, N Montavont ea. Prevailing over wires in healthcare environments: benefits and challenges. 2006;44(4):56–63. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1632650.

[76] Interference characterization and UWB channel measurements for wireless intensive care patient monitoring; 2009. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5297264.

[77] Research for Multi-user Wearable Monitor System of Physiological Signals; 2007. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4303988`.

[78] Malhi K, Mukhopadhyay S, J Schnepper ea. A Zigbee-Based Wearable Physiological Parameters Monitoring System. 2012;12(3):423–430. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5629425`.

[79] Intelligent Sensing Systems for Measuring Wellness Indices of the Daily Activities for the Elderly; 2012. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6258549`.

[80] IEEE 802.15 WPAN Task Group 6 (TG6) Body Area Networks;. Available from: `http://www.ieee802.org/15/pub/TG6.html`.

[81] Taylor S, Farinholt K, E Flynn ea. A mobile-agent-based wireless sensing network for structural monitoring applications. Journal of Measurement Science and Technology. 2009;20(4).

[82] Wang H, Peng D, W Wang ea. Resource-aware secure ECG healthcare monitoring through body sensor networks. 2010;17(1):12–19. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5416345`.

[83] Verification of key establishment protocols for a home health care system; 2008. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4762015`.

[84] Privacy and security in biomedical applications of wireless sensor networks; 2008. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4712575`.

[85] Loucopoulos P, Karakostas V. System Requirements Engineering. McGraw-Hill International Series in Software Engineering.. McGraw-Hill; 1995.

[86] Graham I, Jones PL. Expert Systems: Knowledge Uncertainty and Decision. Bury St Edmunds, UK: St Edmundsbury Press; 1988.

[87] SysML Open Source Specification Project;. Available from: `http://www.sysml.org/`.

[88] Kleppe A, Warmer J, Bast W. MDA Explained: The model-driven architecture: practice and promise. Addison-Wesley; 2003.

[89] Group OM. MDA;. Available from: `http://www.omg.org/mda/`.

[90] Visual Paradigm;. Available from: `http://www.visual-paradigm.com/`.

[91] Firesmith D. Global Personal Marketplace SRS; 2003. Available from: `http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf`.

[92] Maciaszek LA. Requirements Analysis and System Design. 3rd ed. Pearson Education Limited; 2007.

[93] DoIT. System Design Document;. Available from: doit.maryland.gov/SDLC/Documents/sys_design_doc.doc.

[94] Horsch J, Schindler F. Personal physical activity and heart physiology analysis and feedback service based on multifunctional biosensors and Android smartphones; 2011. IDP project documentation.

[95] Sievers J, Schmidt J. Server and database deployment for WBSN prototype; 2012. IDP project documentation.

[96] An overview of the MVC pattern in Java from the Sun website.;. Available from: http://www.oracle.com/technetwork/java/mvc-140477.html.

[97] Haan JD. MDE–Model Driven Engineering reference guide; 2009. Available from: http://www.theenterprisearchitect.eu/archive/2009/01/15/mde---model-driven-engineering---reference-guide.

[98] StarUML;. Available from: http://staruml.io/.

[99] Paterno F. Model-based design and evaluation of interactive applications. Springer; 2000.

[100] Pribeanu C, Vanderdonckt J. A methodological approach to task-based design of user interfaces. Studies in Informatics and Control. 2002;11:145–158.

[101] Towards visual analysis of usability test logs using task models; 2006.

[102] Submission CWWG. CTT; 2012. Available from: http://www.w3.org/2012/02/ctt/.

[103] ANT+ in Mobile;. Available from: http://www.thisisant.com/business/opportunities/mobile//?android-api/.

[104] Berners-Lee T, Fielding R, Gettys J. Hypertext Transfer Protocol-HTTP/1.1-RFC 2616; 1999. Available from: http://tools.ietf.org/html/rfc2616.

[105] Fielding RT. Architectural styles and the design of network-based soft-ware architectures.; 2000.

[106] Open SSL/TLS;. Available from: http://www.openssl.org/.

[107] qun Wang Z. Approach for transformation of ConcurTaskTree to UML2.0. Computer Engineering. 2009 June;35(11).

[108] Balsamiq;. Available from: http://balsamiq.com/.

[109] WSGI - Web Server Gateway Interface;. Available from: http://wsgi.readthedocs.org/en/latest/.

[110] Shimmer Research;. Available from: http://www.shimmersensing.com/.

[111] ActiGraph;. Available from: http://www.actigraphcorp.com/.

[112] Preg A. Android based ANT+ connection; 2013. IDP project document.

[113] Velden L. System Test of SensCoAP prototype; 2013. IDP project documentation.

[114] Walters DL, Sarela A, Fairfull A. A mobile phone-based care model for outpatient cardiac rehabilitation: the care assessment platform (CAP). BMC Cardiovascular Disorders. 2010;.

[115] Fletcher RR, Tam S, Omojola O. Wearable Sensor Platform and mobile Application for Use in Cognitive Behavioral Therapy for Drug Addiction and PTSD. 33rd Annual Intl Conf IEEE EMBS. 2011;.

[116] Doherty ST, Oh P. A Multi-Sensor Monitoring System of Human Physiology and Daily Activities. Telemedicine and e-Health. 2012 April;18(3).

[117] Fei D, Zhao X, Boanca C. A biomedical sensor system for real-time monitoring of astronauts' physiological parameters during extra-vehicular activities. Computers in Biology and Medicine. 2010;40:635–642.

[118] Riveline J, Schaepelynck P, Chaillous L. Assessment of Patient-Led or Physician-Driven Continuous Glucose Monitoring in Patients With Poorly Controlled Type 1 Diabetes Using Basal-Bolus Insulin Regimens. Diabetes Care. 2012;35:965–971.

[119] Wagenaar RC, Sapir I, Zhang Y. Continuous Monitoring of Functional Activities using Wearable, Wireless Gyroscope and Accelerometer Technology. 33rd Annual Intl Conf IEEE EMBS. 2011;.

[120] Alexander GL, Wakefield BJ, Rantz M. Passive Sensor Technology Interface to Assess Elder Activity in Independent Living. Nurs Res. 2011;60(5):318–325.

[121] Golomb MR, McDonald BC, Warden SJ. In-Home Virtual Reality Videogame Telerehabilitation in Adolescents With Hemiplegic Cerebral Palsy. Arch Phys Med Rehabil. 2010 Jan;91(1).

Appendix A

System Requirement Specification

This appendix provides detailed information on system requirement analysis. Where information on system requirement specification was already given in Chapter III, the appendix will reference this information.

1. Methods of Requirement Analysis

See Section 2 in Chapter III.

2. General Specification

See Subsection 3.1 in Chapter III.

3. Specific Requirements

On the Biosensors

REQ-1

The biosensors are able to trigger audible or visible alerts as reminder in case the biosensors are about to run out of battery.

- *Frequency: high*
- *Risk: low*
- *Priority: high*

REQ-2

The sampling rate of the biosensors must be configurable.

- *Frequency: high*

- *Risk: medium*
- *Priority: high*

REQ-3
Before each individual measurement starts, the mandatory parameters of the biosensors must be configured, in order to avoid further interactions with the subjects.

- *Frequency: high*
- *Risk: low*
- *Priority: high*

REQ-4
The biosensors must be paired with only one specific smartphone before use. The authentic pairing is protected by password.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

On the BAN

REQ-5
The BAN hub is able to detect the surrounding wireless biosensors.

Normal case:
The healthcare professional is able to search for devices to be connected.
Pre-condition:
The BAN hub and its wireless function are on.
Interactions:

1. The healthcare professional refreshes the list in order to find out his desirable devices.
2. The BAN hub initializes a new detection and refresh the list

Post-condition:
All detected devices will show up in a list.

Exceptional case:
If no devices are found, the BAN hub should pop up a message.

Exceptional case:
If there is any measurement running on the BAN hub, new detection should not interrupt the ongoing process.

- *Frequency: medium*
- *Risk: medium*
- *Priority: medium*

REQ-6
The BAN hub is able to pair with the desirable biosensors.

Normal case:
The BAN hub succeeds to pair with target biosensors.
Pre-condition:
The BAN hub has detected the nearby biosensors and the target biosensors are present in the list.
Interactions:

1. The healthcare professional selects the target by device ID from the list, initializing a connection request by pressing "connect" button.
2. The BAN hub requires him to type the correct pairing password.
3. The healthcare professional types password and presses OK.
4. The BAN hub confirms the input and connects to the biosensor.

Post-condition:
The connection between the BAN hub and the target biosensor has been established.

Exceptional case:
If the paring password is wrong, the BAN hub must deny the connection request.

Exceptional case:
If the system takes too long time to respond, the BAN hub will terminate the connection request.

- *Frequency: medium*
- *Risk: low*
- *Priority: medium*

REQ-7
The BAN hub should allow the healthcare professional to configure necessary parameters for the measurements in advance.

Normal case:

The corresponding healthcare professional configures the parameters for a measurement in advance.

Pre-condition:

The healthcare professional is authenticated.

Interactions:

1. The healthcare professional selects the right measurement session.
2. The BAN hub corresponds with a configuration page for that session.
3. The healthcare professional sets all the necessary parameters in that page.
4. The healthcare professional saves the change on the BAN hub.

Post-condition:

The new parameter set will be activated when the measurement session starts.

Exceptional case:

When the healthcare professional types a parameter value which is out of normal range, the BAN hub will pop up an error message and reset the parameter as its default value.

- *Frequency: low*
- *Risk: medium*
- *Priority: low*

REQ-8

The BAN hub is able to start and stop a measurement session, either automatically or manually.

Exceptional case:

When the connection between the BAN and the biosensor gets lost, the measurement must stop automatically.

- *Frequency: high*
- *Risk: medium*
- *Priority: high*

REQ-9

The BAN hub is able to receive the data sent from the biosensors and visualize them on the display in real-time.

Exceptional case:
If the connection between the BAN hub and the biosensors is lost during data transmission, the visualization process stops and will start again once the connection resumes.

- *Frequency: high*
- *Risk: medium*
- *Priority: high*

REQ-10
The BAN hub is able to synchronize with the BAAS, according to the pre-configured synchronization interval.

Normal case:
The BAN hub is able to forward the received sensor data to the BAAS during the synchronization.

Pre-condition:
The connection between the BAN hub and the BAAS is already established.
Interactions:

1. The BAN hub sends out request of uploading data.
2. The BAAS requires its authentication.
3. The BAN hub replies with the corresponding information.
4. The BAAS agrees to receive data from the BAN hub.
5. The BAN hub starts transmitting data.
6. The BAAS confirms it got all the data.
7. The BAAS terminates the synchronization session.

Post-condition:
All data are stored in the BAAS database.

Normal case:
The BAN hub is able to download commands from the BAAS during the synchronization.

Pre-condition:
The connection between the BAN hub and the BAAS is already established.
Interactions:

1. The BAAS sends out request of synchronizing commands.
2. The BAN hub provides its authentication information for identification check.

3. The BAAS confirms its identification and starting sending the relative commands to the BAN hub.
4. The BAN hub confirms the receiving.
5. The BAN hub terminates the synchronization session.

Post-condition:
The BAN hub gets all its commands from the BAAS.

Exceptional case:
The synchronization should not interrupt the current running measurement process.

Exceptional case:
If the connection from the BAN hub to the BAAS is lost, the BAN hub should try to resume the connection and send the data immediately once it resumes.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

REQ-11
The synchronization interval is configurable, either manually via the BAN hub by the healthcare professional, or automatically adjusted regarding current connection availability.

Normal case:
The corresponding healthcare professional configures the synchronization interval on the BAN hub.
Pre-condition:
The healthcare professional is authenticated.
Interactions:

1. The healthcare professional goes to the configuration option of synchronization interval.
2. The BAN hub corresponds with a slide bar within a certain range, on which the current value is set by default according the connection status.
3. The healthcare professional sets it to the value he wants.
4. The healthcare professional saves the change on the BAN hub.

Post-condition:
The new synchronization interval will be activated immediately on the BAN hub.

- *Frequency: high*

- *Risk: low*
- *Priority: high*

REQ-12

The BAN hub should have a local encrypted database and buffer for data caching in case the connection is gone.

- *Frequency: medium*
- *Risk: medium*
- *Priority: medium*

REQ-13

If the connection gets lost, the BAN hub should have the capability to auto-reestablish the connection without interrupting the subjects.

- *Frequency: medium*
- *Risk: low*
- *Priority: medium*

REQ-14

When the battery of the BAN hub is under certain level, or the connection status is not stable, the BAN should switch to power-saving mode, under which only critical data would be forwarded and alarm would be triggered if necessary.

- *Frequency: medium*
- *Risk: medium*
- *Priority: low*

REQ-15

The subject is able to access to his own data and require feedback services regarding his health status through the BAN hub.

Normal case:

The subject gets feedback about his health status via the BAN hub.

Pre-condition:

The subject has his BAN hub and the connection between the BAN hub and the BAAS is already established.

Interactions:

1. The subject presses the feedback services option on the BAN hub.
2. The BAAS asks for the password of the subject.
3. The subject types his password and presses OK.
4. The BAAS confirms his authentication and the BAN hub leads to the feedback services menu.
5. The subject selects the services and the time slot of the data to be processed as he wants.
6. The BAAS corresponds with the results to the subjects.

Post-condition:
The subject is able to view his data and other feedback on the BAN hub display.

Exceptional case:
If the authorization information provided by the subject is wrong, the BAAS will reject the request of feedback services.

- *Frequency: medium*
- *Risk: medium*
- *Priority: high*

REQ-16
All data transmission should be reliable, encrypted and as secure as possible.

- *Frequency: high*
- *Risk: high*
- *Priority: medium*

On the BAAS

REQ-17
The BAAS is able to receive data from the BAN and store them permanently in its database.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

REQ-18

Timestamps must be included in the incoming data using relative time (master clock + time differentials) instead of the absolute time.

- *Frequency: high*
- *Risk: low*
- *Priority: low*

REQ-19

The healthcare professional is able to use a common interface to export data from the BAAS database to other destinations via a common interface.

Normal case:

The healthcare professional is able to export data via the BAAS.

Pre-condition:

The healthcare professional is authenticated by the BAAS to have the right to export data.

Interactions:

1. The healthcare professional selects the data set he wants to export.
2. He defines the exported data format, destination and other mandatory fields on the exportation interface.
3. The BAAS executes the exportation operation and comes back to him with confirmation message.

Post-condition:

All the data sets that are supposed to be exported are in their destination.

- *Frequency: low*
- *Risk: low*
- *Priority: low*

REQ-20

The healthcare professional is able to view the data of the subjects he is in charge of on the BAAS.

Normal case:

The healthcare professional visualizes the physiological data of one specific subject in his group.

Pre-condition:

The healthcare professional is already authenticated by the BAAS.

Interactions:

1. The healthcare professional selects the corresponding group which he is in charge of.
2. The BAAS responds with a member list of this group.
3. The healthcare professional selects the specific subjects.
4. The BAAS responds with all data options for this subject.
5. The healthcare professional specifies the data source, time duration, type of data, etc.
6. The BAAS visualizes the selected data set in graphics on the interface.

Post-condition:
The healthcare profession gets the data curves of his subject.

- *Frequency: high*
- *Risk: low*
- *Priority: high*

REQ-21
The BAAS has the ability to process data according to the function of the plug-ins the healthcare professional selects.

Normal case:
The healthcare professional processes the data of his subject by means of plug-ins on the BAAS.

Pre-condition:
The healthcare professional is already authenticated by the BAAS.
Interactions:

1. The healthcare professional selects the corresponding processing plug-ins from the menu.
2. The BAAS redirects him to the processing interface.
3. The healthcare professional uploads the dataset of one specific subject in his group, and configures all the parameters that are necessary for the processing.
4. The healthcare professional clicks "OK" to execute.
5. The BAAS provides him the processing results in the way he prefers.

Post-condition:
The healthcare professional gets the processing results on which his further diagnosis will be based.

Exceptional case:
When the healthcare professional inputs a parameter value which is out of normal range, the BAAS will pop up an error message and reset the parameter as its default value.

- *Frequency: high*
- *Risk: medium*
- *Priority: high*

REQ-22

The healthcare professional is able to export and share the processing results under permission to other authenticated specialists in the same system for advices.

Normal case:

The healthcare professional shares the processing results of his subject with others for advices.

Pre-condition:

The healthcare professional already gets the processing results and all involved staff are authenticated by the BAAS.

Interactions:

1. The healthcare professional goes for the sharing functionality on the processing interface.
2. The healthcare professional selects the specialists to whom he wants to get advices from the lists and confirms his choice.
3. The BAAS will forward the processing results to the target specialists and remind them, e.g. via email or personal message.

Post-condition:

The other specialists also get copies of the processing results regarding that specific subject.

- *Frequency: low*
- *Risk: medium*
- *Priority: low*

REQ-23

The BAAS is able to provide feedback services to the subjects.

Normal case:

The subject acquires feedback information about his health via the BAAS.

Pre-condition:

The subject is authorized by the system through password protection.

Interactions:

1. The subject selects from the menu any available types of feedback services.

2. The BAAS redirects him to the feedback service interface.
3. The subject is asked to limit his preference e.g. by terms of time interval.
4. The subject confirms his inputs.
5. The BAAS provides him the processing results in the way he prefers.

Post-condition:
The subject gets the feedback results regarding his health status.

Exceptional case:
When the subject inputs by mistake a value which is out of normal range, the BAAS will pop up an error message and reset the parameter as its default value.

- *Frequency: medium*
- *Risk: low*
- *Priority: high*

REQ-24
The BAAS is able to distinguish the user type (subject/healthcare professional/system administrator) by his login and redirects him to the corresponding home page.

Normal case:
The BAAS recognizes the user identity and presents him the respective user interface.
Pre-condition:
The user has the logging information to the BAAS.
Interactions:

1. The user types his login and password.
2. The BAAS validates the inputs.
3. The BAAS detects the identity category this user belongs to.
4. The BAAS redirects him to the corresponding access point of the system.

Post-condition:
The user is logged in and can operate in his management webpage.

- *Frequency: high*
- *Risk: low*
- *Priority: medium*

REQ-25

The authorized user (healthcare professional/system administrator) is able to add/remove/modify other users (subjects/healthcare professionals) to/ from his group lists and their attributes in the BAAS.

Normal case:

The healthcare professional/system administrator uses the BAAS to manage his group lists.

Pre-condition:

The healthcare professional/system administrator login is authenticated by the system.

Interactions:

1. The healthcare professional/system administrator goes to the management page, selects the operations (remove/add/modify) he wants.
2. After finishing changing, he confirms his operations.
3. The system saves and updates the database about the change, and refreshes the page.

Post-condition:

The information in the database and on the webpage has been changed.

Exceptional case:

If the system does not respond in a certain period of time, the session would be closed automatically.

- *Frequency: high*
- *Risk: medium*
- *Priority: high*

REQ-26

It is possible for the healthcare professional to pseudonymize the subjects when necessary.

Normal case:

The healthcare professional decides to pseudonymize his subjects for a certain study.

Pre-condition:

The healthcare professional has enough rights in the BAAS system to pseudonymize the studies/subjects.

Interactions:

1. The healthcare professional creates/manages a study/subjet on his management webpage.

2. The healthcare professional chooses the psedonymization option for the study/subject.
3. The BAAS double-checks with the healthcare professional of his choice.
4. The healthcare professional confirms he wants to pseudonymize this study/subject.
5. The BAAS pops up message stating that the operation is successful.

Post-condition:
The personal information of the subject will not be recorded in the BAAS database.

- *Frequency: medium*
- *Risk: high*
- *Priority: low*

REQ-27
The healthcare professional is able to configure all the parameters on the BAN hub from the BAAS side remotely.

Normal case:
The healthcare professional configures the BAN hub of his subject in distance.
Pre-condition:
The healthcare professional is authorized and the connection between the BAAS and the BAN hub is established.
Interactions:

1. The healthcare professional selects the BAN hub he wants to configure.
2. The healthcare professional sets all the parameters he wants to change for that BAN hub and saves his change.
3. When synchronized with the BAAS the BAN hub downloads all the new settings for it.

Post-condition:
The new settings are activated on the BAN hub.

- *Frequency: medium*
- *Risk: low*
- *Priority: low*

REQ-28
The BAAS system is maintainable and backups regularly by the system administrator.

- *Frequency: medium*

- *Risk: high*
- *Priority: high*

REQ-29

The BAAS is able to update/install new plug-ins by the system administrator. The new version of the system should be compatible with the old ones.

- *Frequency: medium*
- *Risk: low*
- *Priority: high*

REQ-30

All data storage and transmission within the database should be secure.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

REQ-31

When the mean value of the incoming physiologic data in certain period of time from a specific subject falls below the threshold (e.g. the subject is inactive for a long time), the BAAS will alert the responsible healthcare professional.

- *Frequency: low*
- *Risk: medium*
- *Priority: low*

4. Out-of-Scope Requirements

This section mainly documents those requirements that were mentioned and considered critical in the requirement collection interviews by the potential users and technicians. However due to the technical limitations and project scopes, they were finally classified as out-of-scope and will never be implemented later on. The reasons include e.g. hardware design which is not relevant to the project goals of SensCoAP and currently it is infeasible to be improved. As reference for future, in case techniques allow any modification very soon, all the out-of-scope requirements are still documented here in this appendix. All requirements are grouped by subsystems, the same style as in Section 3.

On the biosensors

REQ-32

Performance of the selected biosensors must be reliable.

Exceptional case:

The biosensors must be robust enough that no data should be lost even though the battery goes dead.

Exceptional case:

Measurement must be as accurate as possible, regardless of low energy supply or high sampling rate.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

REQ-33

The wireless biosensors should have a long battery life.

Exceptional case:

The batteries of the biosensors are able to afford a long time performance of sampling and wireless transmission even in a high frequency.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

REQ-34

The biosensors should have the ability to sample at high frequencies.

- *Frequency: medium*
- *Risk: low*
- *Priority: high*

REQ-35

The biosensors must have ability to buffer data for a while, when wireless connection is lost or there is a need to save power.

Exceptional case:

When the biosensor loses its connection, it buffers the data until connection recovers, then it sends out all buffered data immediately.

- *Frequency: medium*
- *Risk: high*
- *Priority: high*

REQ-36

The biosensors should give out visible indication about their working status, e.g. low battery level, sampling, idle, sleep mode, etc.

- *Frequency: high*
- *Risk: low*
- *Priority: medium*

REQ-37

The biosensor is supposed to have an acceptable design for its size and appearance, i.e. the size of the biosensor should be as small as possible and the appearance of the biosensor should be acceptable especially for the subject group (elderly) for long time wear.

- *Frequency: high*
- *Risk: low*
- *Priority: low*

REQ-38

The positioning of the biosensors should be comfortable, neither irritating nor annoying.

- *Frequency: high*

- *Risk: low*
- *Priority: low*

REQ-39
The biosensor is available in the market with reasonable price.

- *Frequency: no*
- *Risk: no*
- *Priority: no*

REQ-40
The biosensor is better to be waterproof.

- *Frequency: low*
- *Risk: high*
- *Priority: medium*

REQ-41
The biosensor should be easy to wear. The positioning of the sensor should be specified before applying to the subjects.

- *Frequency: high*
- *Risk: low*
- *Priority: low*

REQ-42
The biosensors which use the same transmission radio should not interfere with each other.

- *Frequency: high*
- *Risk: high*
- *Priority: high*

On the BAN

REQ-43

The BAN should be straightforward to operate, especially for elderly. When getting pressed, the BAN gives audible or perceivable feedback to the elder subject.

- *Frequency: high*
- *Risk: low*
- *Priority: low*

REQ-44

The interactions between the subject and the BAN should be reduced as few as possible.

- *Frequency: no*
- *Risk: low*
- *Priority: high*

5. Scenarios

This section documents the scenarios which was used in the requirements collection phase. As some scenarios were already mentioned in Chapter III, they will not be repeated here once again.

Scenario V: self-monitoring

> *Tobias feels recently that his health status has been improved. He is more active now and walks longer every day. To proof this, he logs into website of the telemonitoring center with his ID and password which were issued to him together with his devices. The system verifies his login information as a subject account and authenticates him read-only access right. Now Tobias is able to view his health records stored in the monitoring system. From the menu of the webpage he has the options to choose from several methods of feature abstraction, to examine different index of his health status according to the data collected from the previous sensor measurement sessions. He chooses "physical activity level" and limits the time slot into today. The system provides him the graphical view of his activity data together with a percentage of his performance today compared with an average value. Tobias is glad to see that he reaches 83% level on average, indicating he is active enough today.*

Scenario VI: management

> *Peter Schlet is dealing with his monitoring group issues. He has two subjects who terminate their monitoring courses and one more subject who would like to join in. After login he goes to the group page he is in charge of. He removes those two subjects from the respective groups and creates a new subject account with the bounded device IDs. Then he adds this new subject ID to one of his groups in charge.*
>
> *Stephan Friedrich is the administrator of the monitoring system and he is also responsible for maintaining the system in the telemonitoring center. According to the request of Peter Schlet, he is now assigning a superuser right to Peter. When this is done Stephan tries to activate a new feature extraction program in the system which can provide the doctors a new way of monitoring heart rate. After the installation and activation, the new program appears in the "tools" menu for Peter.*

Scenario VII: battery low

> *Tobias Maurer decides to take a walk into the forest after lunch. His mobile phone keeps searching for all available networks once he leaves home. On his way the mobile phone detects that one of his attached sensors falls low voltage of the battery. It alerts Tobias with a pop-up message on the screen. However Tobias still feels like continuing his walk. 2 hours later, his mobile phone also falls in low battery. It warns Tobias by ringing and stops streaming sensor data. Instead it begins to buffer the collected data in order to save power. Tobias returns home and charges both sensors and mobile phone. When connecting with the power supply the mobile phone uploades all the data in its local buffer to the remote center.*

Appendix B

Study Protocol of the SensCoAP Validation

This appendix provides the complete study protocol for system evaluation. Where information on system test experiments was already given in Chapter VI, the appendix will reference this information.

Feasibility Assessment of the Prototype of the SensCoAP System

Study coordinator: Chen Chen
Investigators: Chen Chen, Lisa Velden
Head of the biosensor group: Prof. Alexander Horsch

Technische Universität München (TUM)
Institut für Medizinische Statistik und Epidemiologie (IMSE)

Klinikum rechts der Isar
Ismaninger Str. 22
81675 München

Abstract

This document presents the design of a feasibility study for evaluation of the prototype of SensCoAP system. The study aims to acquire user experiences with the system, especially on usability and user satisfaction. The experiments are divided into two phases: system performance test by young healthy people and system acceptance test by some elderly or healthcare personnel. In both phases Shimmer sensors with 3-axis accelerometers together with Android smartphones are used.

Background

Many remote health monitoring systems using sensor networks have come to reality in recent years and studies on those systems can be found in literature. Those systems are usually based on the Information and Communication Technology (ICT) and involve popular mobile devices e.g. smartphone and web services [114–116]. In some systems real-time data streaming and interaction with an external database are also tested [117, 118]. In wireless sensor networks, the Bluetooth radio is proved to be reliable in the sensor-based monitoring systems [115, 119]. System usability, organizational workflow and user satisfaction are the factors usually measured by experiential studies [120].

However, there is no standard on the study design. Different systems have been evaluated by experiments with different study designs. The participants varied from 3 [121] to 257 [118], and the test time varied from several hours until 1 year [118]. Large-scaled study even involved several study centers and hospitals [118]. The methods commonly used are: observation, interview, survey, think aloud [120] and self-report [116].

SensCoAP, the system that is evaluated in this study, is the system developed as part of the coordinator's PhD project. It provides a prototype which needs validation and assessment for its usability. The goal of the study is to provide a proof-of-concept and to get feedback on the user experience of the SensCoAP system, including both system usability and user satisfaction. SensCoAP uses wearable sensors, smartphone and central server and transmits data via Bluetooth. The study will measure the performance of the system. Methods applied include observation, survey and think aloud.

Study Questions

See Chapter VI.

Experiments

The experiments are divided into 2 phases. Phase 1 tests the usability and reliability of the system by young healthy participants, and Phase 2 tests the acceptance and satisfaction of the system by both young and elderly participants or epidemiological researchers/healthcare personnel.

Phase 1

The test of Phase 1 will take place mainly at IMSE and Campus Garching. However, any other places where the smartphone is able to connect to the internet are also possible, e.g. the participant's home.

The participants will be assigned with one Shimmer sensor and one Android smartphone. The participants will wear the sensor (e.g. in their pockets) for approx. 2-3 hours; at the same time perform some tasks on the smartphone. Later after taking the sensor off, they are also supposed to finish some tasks on the server. At last they are required to fill a questionnaire.

The tasks include:

On the smartphone

- Run the app.
- Discover and connect the target sensor.
- Set the user credential and other properties.
- Start/Stop the sensor recording.
- Upload the data manually.

On the server

- Login as Healthcare Professional (HCP)/subject/admin respectively.
- When login as HCP, add a new subject/biosensor/group to the system.
- When login as HCP, assign the new device to the new subject who belongs to a new group.
- When login as HCP, view the group/ subject/biosensor information.
- When login as HCP, process a specific data chunk by selecting a plugin.
- When login as subject, view his personal data and uploaded dataset.
- When login as admin, add a new HCP to the system.
- When login as admin, install a new plugin to the system.

Details of the tasks can be referred to the information sheet for tests. All the tasks should be finished by the participants alone without any instructions from the study investigators. Participants are encouraged to think aloud when they perform the operations on the user interfaces.

The participants will be recruited mainly from the students and staff at IMSE and Institute for Informatics in Garching. Other young people around can also be involved. The number of the group can be 15 up to 30. The average age of the group should be around 30.

Phase 2

The test of phase 2 will take place mainly in the relevant clinics or the organizations; however it will not exclude the possibility of conducting the tests at the participant's home, where internet access must be ensured for the smartphone connecting to the SensCoAP server.

The participants will be assigned with one Shimmer sensor and one Android smartphone. The participants will either wear the sensor in their pockets for approx. 1 day (until the battery dies) and perform some tasks on the smartphone as a subject, or finish some tasks on the server as a HCP. Both roles are required to fill a questionnaire regarding acceptance and satisfaction at last.

The tasks include:

For subjects

- Discover and connect the target sensor.
- Recharge the sensor.
- View the real-time data curves.
- When alarm is triggered, respond with the corresponding operations.
- Login the server with user credential, and view his own personal/sensor data.

For HCPs

- Start the Android app.
- Discover and connect the target sensor.
- Set the user credential and other properties.
- Set the alarm thresholds and relevant parameters.
- Set the HCP mobile phone number for emergency.
- Login the server.
- Add a new subject/biosensor/group to the system
- Assign the new device to the new subject who belongs to the new group.
- View the group/subject/biosensor information.

- Process a specific data chunk by selecting a plugin.

Details of the tasks can be referred to the information sheet for tests. The tasks can be finished by the participants under the instructions from the study investigators if necessary.

The participants will be recruited mainly from the collaborating clinics and organizations. The group number is estimated as no less than 13. The average age of the participants as subjects should be around 60. There is no age restriction for the participants as HCPs.

Devices to be Used

1. Shimmer 2R (Real-time Technology, Dublin, Ireland) Triaxial accelerometer, SD card depended memory, 10-256Hz sampling rate
2. Android Smartphones (HTC, T-Mobile, Samsung, Sony) with the "BAN Hub" app (version 3.0)
3. Laptops or desktops with internet browsers

Study participants

Inclusion criteria:

Phase 1

- Healthy young people
- Able to use a smartphone
- Age: $20 < x < 40$

Phase 2

- Elderly and people with healthcare experience or epidemiological researchers
- Able to use a smartphone/computer
- Age (only for elderly): $x \geq 60$

Exclusion criteria for both phases:

- No Wifi/3G is available when doing the test

Type of the Study

Feasibility study; experimental study

Which Variables and Attributes are Requested, Measured and Calculated?

See Chapter VI.

Considerations Regarding the Sample Size

The more participants attend the tests, the better the results will be. But the final number of the participants also depends on the time available for the test and the real situation considering the condition of Phase 2. The expected minimal number of participants to be included for Phase 1 is 15 healthy young people. For Phase 2 at least 3 elderly and 10 healthy young people have to be included.

Which Influences can Perturb the Test Results?

- Bias of the participants: only technology-friendly people are involved.
- The connections between sensor and smartphone, or between smartphone and server are easy to lose.
- The battery life of the sensor and the battery life of the smartphone.
- No Wifi/3G available
- The UIs are difficult to operate by the participants.

Quality Management

- The function of the sensors has to be verified before the experiment.
- Only fully charged sensors and smartphone will be used.
- After the experiment the answers to the questionnaires will be analyzed to get a first impression of the user experience of the system.

Organization of the Experiment

The study is a part of PhD work of the study coordinator. For Phase 1, the experiment is mainly conducted at IMSE and Informatic Institute in Campus Garching, but other places are also possible. The recruitment is carried out by asking students, staff and friends around. For Phase 2, the experiment is mainly conducted at the collaborating clinics and organizations, or the subjects' home. The recruitment is carried out by asking patients, subjects and staff there.

Data Privacy

All personal data will be pseudonymized to avoid identification of the participants.

The following data will be stored for every participant:

- Age
- Gender
- Background

An information sheet with the conditions of the participation is provided to each participant. A consent form must be signed by each participant. If a participant resigns, all his data will be deleted.

The data measured by the sensors will also be pseudonymized. All data will be stored on the server at the IMSE. Access to the data is limited to the members of the biosensor group at IMSE.

Publications

The pseudonymous participant list with the personal data and the measured accelerations will be stored at IMSE. The analysis and a final report are done by the study investigators.

As authors of eventual publications will be listed the supervisor at IMSE and the head of the biosensor group at IMSE.

Is a Publication of the Study Protocol Planned?

No

Ethics

All subjects will provide a written informed consent and get an information sheet.

If Phase 2 includes patients in the clinics, an ethics proof might be needed.