



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

**Marco Johns**

**Conception and evaluation of a comprehensive patient  
empowering and pervasive healthcare system**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Marco Johns

**Conception and evaluation of a comprehensive patient  
empowering and pervasive healthcare system**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stefan Sarstedt  
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 12. September 2013

**Marco Johns**

**Thema der Arbeit**

Conception and evaluation of a comprehensive patient empowering and pervasive healthcare system

**Stichworte**

Pervasive Healthcare, Patienten Empowerment, Systemkonzeption, Softwarearchitektur

**Kurzzusammenfassung**

Pervasive healthcare ist ein neues Feld, welches Computersysteme einsetzt um die Gesundheitspflege zu verbessern. Das Ziel von "Patienten Empowerment" ist es, Individuen zu befähigen sich aktiver an ihrer Gesundheitspflege zu beteiligen. In dieser Arbeit werden diese beiden Prinzipien miteinander kombiniert. Hierfür werden Technologien wie "Persönliche Gesundheitsakten" (engl.: Personal Health Records) und das Sammeln von Informationen eingesetzt um den Benutzer zu ermächtigen. Zu diesem Zweck wurde eine breit gefächerte Anforderungsanalyse durchgeführt und auf Grund der resultierenden Anforderungen ein architekturelles Systemkonzept entworfen. Schließlich wurde dieses Konzept in Kürze qualitativ und auf Grund der zuvor bestimmten Anforderungen evaluiert.

---

**Marco Johns**

**Title of the paper**

Conception and evaluation of a comprehensive patient empowering and pervasive healthcare system

**Keywords**

Pervasive healthcare, Patient empowerment, System conception, Software architecture

**Abstract**

Pervasive Healthcare is a novel field which uses computer systems for the improvement of healthcare. Patient Empowerment aims to enable and encourage an individual to take a more active role in his own healthcare. In this thesis these two concepts are combined, utilizing technologies such as Personal Health Records and Information Collection in order to empower the user. For this purpose a broadly aimed requirement analysis has been conducted and based on these requirements an architectural system concept has been drafted. Finally this concept has been briefly evaluated qualitatively and based on the previously derived architectural requirements.

# Acknowledgements

I would like to thank some people, that have helped me during the time I worked on this thesis, among these my thankfulness goes to:

My thesis tutor Prof. Dr. Stefan Sarstedt, as well as Prof. Dr.-Ing. Franz Korf (Head of Computer Science Department) and Prof. Dr.-Ing. Florian Wenck (Deputy Head of Information & Electrical Engineering Department) for helping me to visit the 7th International Conference on Pervasive Computing Technologies for Healthcare in May 2013.

My love, who got me interested in the area of healthcare and supported me with her experience.

My mother, who took a lot of work on herself to help me work on this thesis.

My father, who got me interested in computers in the first place and who passed away one month into this thesis' work.

In Memoriam

N. Johns (\* 1958 – † 2013)

---

*An ounce of prevention  
is better than a pound of Cure.*

— Benjamin Franklin, 18th Century

*I will prevent disease whenever I can,  
for prevention is preferable to cure.*

— Modern Version of the Hippocratic Oath,  
translated by Louis Lasagna, 1964

# Contents

|   |             |
|---|-------------|
| <b>List of Tables</b>                                 | <b>viii</b> |
| <b>List of Figures</b>                                | <b>ix</b>   |
| <b>1 Introduction</b>                                 | <b>1</b>    |
| 1.1 Motivation . . . . .                              | 1           |
| 1.2 Aim of the Thesis . . . . .                       | 1           |
| 1.3 Limitations . . . . .                             | 2           |
| 1.4 Structure of the Thesis . . . . .                 | 3           |
| <b>2 Pervasive Healthcare</b>                         | <b>4</b>    |
| 2.1 Health Records . . . . .                          | 5           |
| 2.2 Information Collection . . . . .                  | 6           |
| 2.3 Patient Empowerment . . . . .                     | 7           |
| 2.4 Interdependence . . . . .                         | 8           |
| <b>3 Requirement Analysis</b>                         | <b>10</b>   |
| 3.1 Stakeholder . . . . .                             | 10          |
| 3.1.1 Users & Patients . . . . .                      | 10          |
| 3.1.2 Clinicians & Healthcare Professionals . . . . . | 11          |
| 3.1.3 Health Insurances & Payers . . . . .            | 12          |
| 3.1.4 Stakeholder Requirements . . . . .              | 12          |
| 3.2 Health Record Keeping . . . . .                   | 13          |
| 3.2.1 Location & Connectivity . . . . .               | 14          |
| 3.2.2 Access & Interoperability . . . . .             | 16          |
| 3.2.3 Contents . . . . .                              | 22          |
| 3.2.4 Patient Opinions . . . . .                      | 29          |
| 3.2.5 PHR and Data Gathering . . . . .                | 31          |
| 3.2.6 PHR and Empowerment . . . . .                   | 32          |
| 3.3 Information Collection . . . . .                  | 33          |
| 3.3.1 Methods of Information Collection . . . . .     | 34          |
| 3.3.2 Lifelogging & Quantified Self . . . . .         | 36          |
| 3.3.3 Data Acquisition . . . . .                      | 37          |
| 3.3.4 Data Quality . . . . .                          | 39          |

|          |   |           |
|----------|---|-----------|
| 3.3.5    | Empowerment & Decision Support . . . . .                          | 40        |
| 3.4      | The "e-Requirements" . . . . .                                    | 40        |
| 3.4.1    | Usability & Health Literacy . . . . .                             | 41        |
| 3.4.2    | Improvement of Health Literacy and Comprehension . . . . .        | 42        |
| 3.4.3    | Online Communities . . . . .                                      | 43        |
| 3.4.4    | Guidance & Decision Support . . . . .                             | 46        |
| 3.4.5    | Motivation & Encouragement . . . . .                              | 47        |
| 3.5      | Extension Capability . . . . .                                    | 48        |
| 3.5.1    | Information Collection Methods . . . . .                          | 49        |
| 3.5.2    | Empowerment & Decision Support Methods . . . . .                  | 49        |
| 3.5.3    | Access & User Interface . . . . .                                 | 49        |
| 3.6      | Consolidation of Requirements . . . . .                           | 50        |
| 3.6.1    | List of Collected Requirements . . . . .                          | 51        |
| 3.6.2    | Requirement Groups . . . . .                                      | 56        |
| 3.6.3    | Architectural Influences . . . . .                                | 57        |
| 3.6.4    | Architectural Requirements . . . . .                              | 57        |
| <b>4</b> | <b>System Conception</b> . . . . .                                | <b>62</b> |
| 4.1      | System Context . . . . .  | 62        |
| 4.2      | Client System . . . . .   | 65        |
| 4.2.1    | Personal Health Record . . . . .                                  | 66        |
| 4.2.2    | Repository Access . . . . .                                       | 68        |
| 4.2.3    | Guest Access . . . . .  | 69        |
| 4.2.4    | Data Gathering . . . . .  | 69        |
| 4.2.5    | Empowerment . . . . .   | 71        |
| 4.2.6    | User Interface . . . . .  | 74        |
| 4.3      | Repository System . . . . .                                       | 76        |
| 4.3.1    | WebService Management . . . . .                                   | 77        |
| 4.3.2    | Persistence Management . . . . .                                  | 77        |
| 4.3.3    | Client Management . . . . .                                       | 78        |
| 4.3.4    | ACS Provider . . . . .  | 79        |
| 4.4      | Access Control System . . . . .                                   | 80        |
| 4.4.1    | Guest Access Management . . . . .                                 | 80        |
| 4.4.2    | Persistence Management . . . . .                                  | 81        |
| 4.4.3    | WebService Management . . . . .                                   | 83        |
| 4.5      | Interactions & Use Cases . . . . .                                | 84        |
| 4.5.1    | Retrieval and Local Storage of Measurement Data . . . . .         | 84        |
| 4.5.2    | Storage of Data in the Repository System . . . . .                | 87        |
| 4.5.3    | Retrieval and Storage of Data for Guest Access . . . . .          | 89        |
| 4.5.4    | Health Provider Access to Shared Data via a Web-Browser . . . . . | 91        |
| 4.5.5    | Visualization of PHR Data as a Graph . . . . .                    | 93        |
| 4.5.6    | Retrieval of Condition Information . . . . .                      | 95        |
| 4.6      | Technological Considerations . . . . .                            | 97        |

## Contents

---

|          |   |            |
|----------|---|------------|
| 4.6.1    | Health Diary Entry and iPHR interface . . . . . | 97         |
| 4.6.2    | SNOMED CT . . . . .                             | 97         |
| <b>5</b> | <b>Concept Evaluation</b>                       | <b>99</b>  |
| 5.1      | Requirement Review . . . . .                    | 99         |
| 5.2      | Proposed Changes . . . . .                      | 102        |
| <b>6</b> | <b>Conclusion &amp; Future Work</b>             | <b>104</b> |
| 6.1      | Conclusion . . . . .                            | 104        |
| 6.2      | Future Work . . . . .                           | 104        |
|          | <b>Bibliography</b>                             | <b>106</b> |
|          | <b>Glossary</b>                                 | <b>109</b> |

## List of Tables

|     |   |     |
|-----|---|-----|
| 3.1 | List of collected vague requirements. . . . .                             | 51  |
| 3.3 | Architectural decisions and their influencing requirement groups. . . . . | 57  |
| 5.1 | List of architectural requirements and their coverage. . . . .            | 100 |



# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | Usage models of pervasive healthcare outside of hospital conditions by Bardram          | 2  |
| 3.1  | Context from the Type of Monitoring by Varshney . . . . .                               | 35 |
| 4.1  | The scope of the system and possible interactions with neighbor systems. . . . .        | 63 |
| 4.2  | The deployment view of the system. . . . .  | 64 |
| 4.3  | The main components of the <i>Client System</i> . . . . .                               | 65 |
| 4.4  | The class diagram draft of the <i>Personal Health Record</i> component. . . . .         | 67 |
| 4.5  | The class diagram draft of the <i>Repository Access</i> component. . . . .              | 68 |
| 4.6  | The class diagram draft of the <i>Guest Access</i> component. . . . .                   | 69 |
| 4.7  | The class diagram draft of the <i>Data Gathering</i> component. . . . .                 | 70 |
| 4.8  | The class diagram draft of the <i>Empowerment</i> component. . . . .                    | 72 |
| 4.9  | The class diagram draft of the <i>User Interface</i> component. . . . .                 | 74 |
| 4.10 | The MVC pattern as a temporary solution for the <i>UILogic</i> sub-component. . . . .   | 75 |
| 4.11 | The main components of the <i>Repository System</i> . . . . .                           | 76 |
| 4.12 | The class diagram draft of the <i>WebService Management</i> component (RS). . . . .     | 77 |
| 4.13 | The class diagram draft of the <i>Persistence Management</i> component (RS). . . . .    | 78 |
| 4.14 | The class diagram draft of the <i>Client Management</i> component. . . . .              | 79 |
| 4.15 | The class diagram draft of the <i>ACS Provider</i> component. . . . .                   | 79 |
| 4.16 | The main components of the <i>Access Control System</i> . . . . .                       | 80 |
| 4.17 | The class diagram draft of the <i>Guest Access Management</i> component. . . . .        | 81 |
| 4.18 | The class diagram draft of the <i>Persistence Management</i> component (ACS). . . . .   | 82 |
| 4.19 | The class diagram draft of the <i>WebService Management</i> component (ACS). . . . .    | 83 |
| 4.20 | Sequence diagram for retrieval and local storage of data from the file system. . . . .  | 86 |
| 4.21 | Sequence diagram for storage data in the repository system. . . . .                     | 88 |
| 4.22 | Sequence diagram for retrieval and storage of data for guest access in the ACS. . . . . | 90 |
| 4.23 | Sequence diagram for health provider access to shared data via a web-browser. . . . .   | 92 |
| 4.24 | Sequence diagram for the visualization of PHR data as a graph. . . . .                  | 94 |
| 4.25 | Sequence diagram for the retrieval of condition information. . . . .                    | 96 |

# 1 Introduction

## 1.1 Motivation

Pervasive Healthcare is a novel field which uses computer systems for the improvement of healthcare. One of the key benefits is to improve the individuals well-being and health, for instance by empowering them to take a more active role in order to maintain it. For this purpose the term Patient Empowerment has emerged and describes principles to enable, encourage, equip, engage - and empower patients. One particular technology got increasingly more attention in the past few years, Personal Health Records.

PHR (Personal Health Record) systems are mostly web-based applications which store health related information into databases. They're functional scope ranges from basic storage of health data to basic decision support and interoperability with healthcare professionals' information systems. In contrast to information systems in use by those healthcare professionals, the PHR systems explicitly allow user submitted information. However, to date there is no consensus of what should or can be part of a PHR system and which tasks it should be able to perform. These circumstances impede further development of specific solutions in regard of Patient Health Record systems and Patient Empowerment.

## 1.2 Aim of the Thesis

The aim of this work is to conduct a requirement analysis based on relevant literature. These requirements are used to create an architectural draft of a comprehensive patient empowering and pervasive healthcare system. Additionally this thesis aims to use the design principle of *Separation of Concerns* in order to separate three main functions of PHR systems: health data storage and exchange by the *Personal Health Record*, the *Collection of Information*, and the *Patient Empowerment* (sometimes also referred to as *Decision Support*). This separation of concerns may focus future efforts on each of these three areas in order to expedite development of new methods and technologies.

### 1.3 Limitations

The topic of this thesis encompasses a wide area of novel issues. Due to time restrictions there are a number of limitations, which are listed in this section.

This work did not conduct a complete literature review, neither did it consider any resources older than 10 years (before 2003). This work did not systematically review technologies. Considered technologies have been discovered by chance during the literature research. This work is not evaluating ethical or legal aspects on the topic.

Further this work is not presenting a complete architecture, since the development of an architecture is an iterative process, only the results of the first few iterations can be presented. Consequently no implementations have been done. This also applies to the usability engineering. Additionally this work does not have any focus on hardware, despite the systems connection to devices.

Due to the critical nature of health related work, this thesis does not cover the entire aspect of disease management and prevention. Bardram et al. (2007) defined usage models of pervasive healthcare outside of hospital conditions. These usage models have different requirements in regard of criticalness, user's participation, privacy, usability, etc. Figure 1.1 shows the five usage models in a two-dimensional space whereas the vertical axis shows the need for professional care and regulations, and the horizontal axis shows whether the application is of preventive nature, or if an illness is already present and therefore it is reactive.

This thesis primarily covers the usage models of *Fitness*, and partially *Risk management*, and marginally *Chronic disease management*.

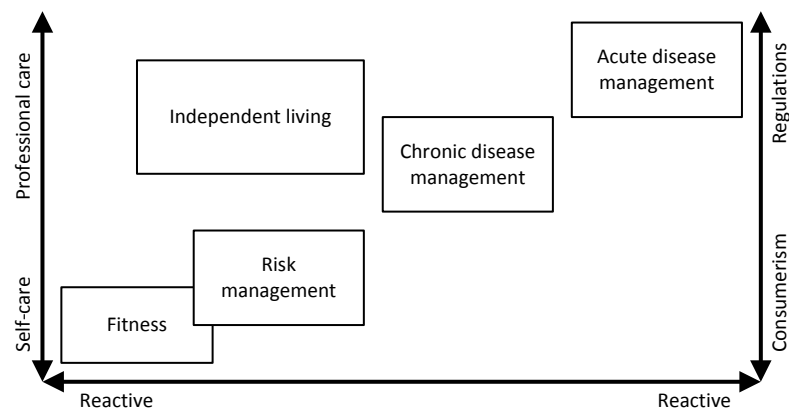


Figure 1.1: Usage models of pervasive healthcare outside of hospital conditions (see Bardram et al., 2007, p. 111)

## 1.4 Structure of the Thesis

This thesis is divided into six chapters. Chapters 1-2 belong to the introduction part. Chapters 3-5 are part of the methodology and constructive work. Lastly chapter 6 is the conclusion part. The first chapter gives an overview about motivation, aim, limitations and the structure of the thesis.

In the *second chapter* a brief introduction is given into the area of Pervasive Healthcare, particularly into technologies like Health Records, Information Collection, Patient Empowerment and their dependencies among each other

In the *third chapter* this work attempts to gather requirements from relevant literature. The researched areas include requirements of stakeholders, health record systems, information collection methods, and the ability to empower a patient. Additionally the extensibility requirements are discussed. The collected vague requirements are then consolidated in groups by taking their influences on the systems architecture into account, and lastly derive architectural requirements from these previously gathered requirements.

The *fourth chapter* is the constructive part where a system concept is developed with the aim of creating an architectural draft. Therefore the systems context and scope is shown and logical views of the subsystems and components are presented and briefly described. Consequently interactions and use cases are shown as sequence diagrams with an informal description. Lastly a few technology considerations are mentioned.

In the *fifth chapter* the previously developed concept will be evaluated by reviewing the architectural requirements and their coverage. Further a few changes are proposed in view of future iterations in the concepts development cycle.

The *sixth chapter* concludes this thesis and gives an outlook on future work that could be done from this thesis' point of view.

## 2 Pervasive Healthcare

Pervasive Healthcare is an approach to use technology to improve healthcare for "anyone, anytime and anywhere", and is highly related to e-Health and Telemedicine. During the last years Pervasive Healthcare has shown a growth in popularity and has been often defined by different authors.

We define pervasive healthcare as "healthcare to anyone, anytime, and anywhere by removing locational time and other restraints while increasing both the coverage and the quality of healthcare". This includes prevention, healthcare maintenance and checkups; short-term monitoring (home healthcare monitoring), long-term monitoring (nursing home), and personalized healthcare monitoring; and incidence detection and management, emergency intervention, and, transportation and treatment [..] The pervasive healthcare applications include pervasive health monitoring, intelligent emergency management system, pervasive healthcare data access, and ubiquitous mobile telemedicine. (Varshney, 2009, p. 50)

While the definition above includes a general description as well as applications, the next definition is separating these perspectives and additionally mentions related fields. Important similarities in these and other definitions are the removal of restraints and the patient<sup>1</sup>-centered approach.

Pervasive healthcare may be defined from two perspectives: i) as the application of pervasive computing technologies for healthcare, and ii) as making healthcare available everywhere, anytime and to anyone. Pervasive healthcare is closely related to biomedical engineering (BME), medical informatics (MI), and ubiquitous computing (UbiComp). [...] While BME and MI mostly focus on technology to improve the existing health delivery model, pervasive healthcare in contrast tries to change the healthcare delivery model: from doctor-centric to patient-centric, from acute reactive to continuous preventive, from sampling to monitoring.

---

<sup>1</sup>The term *individual* is preferred by some authors, since *patient* implies the presence of an illness. (see Tang et al., 2006, p. 122) In this thesis the term *user* is used equivalently.

Additionally, while the term "pervasive" stands for the tendency to expand or permeate, "ubiquity" is the property of being omnipresent. In this sense, the ultimate goal of pervasive healthcare is to become a mean for achieving ubiquitous health. Arnrich et al. (2010)

## 2.1 Health Records

Health Records are only one technology that is used in Pervasive Healthcare. There are several different kinds, but all of them are at least used to persist health related data about one or more individuals. Two types of health records can be easily distinguished: the Personal Health Record (PHR) and the Electronic Health Record (EHR).

What makes PHRs different is that the patient becomes the custodian of the health record instead of the more traditional model, where a provider or health organization maintains the record on behalf of the patient. Patients have the option of adding information to the record and annotating any existing information in the record, which is typically not the case with EHRs. The patient also has full control over who has access to view or add to the record, unlike EHRs, where the hosting organization controls who can add to or view the record. [9] (Robison et al., 2012, p. 184)

Therefore the main difference is the patient-centered design, as confirmed by the following description:

While EHR systems function to serve the information needs of health care professionals, PHR systems capture health data entered by individuals and provide information related to the care of those individuals. Personal health records include tools to help individuals take a more active role in their own health. (Tang et al., 2006, p. 121)

Besides the patient-centered design of a Personal Health Record, there are other features that are specific to Personal Health Records, which share goals with the Patient Empowerment idea (see section 2.3).

A PERSONAL HEALTH RECORD (PHR) is a tool to use in sharing health information, increasing health understanding, and helping transform patients into better-educated consumers of health care. (Kahn et al., 2009, p. 369)

Often a PHR is not only the record itself, but rather a whole system, which utilizes the PHR and enriches it with several additional features. These features often include interoperability with EHR systems, communications, visualizations and user interfaces.

The term PHR is mostly used, but sometimes the acronym PCHR (Patient Controlled Health Record) can be found in literature and is being used interchangeably with PHR as stated by Ogbuji et al. (2011), who further confirms the differences between PHRs and EHRs stated above.

PHRs let individuals monitor and share their healthcare information with healthcare providers as well as third-party health and wellness providers, such as applications for healthy living and weight management. The acronym PHR is often used interchangeably with PCHR (Personally Controlled Health Record), and the definition just given applies to both. The use of technology to let individuals share healthcare information makes PHRs an important aspect in how patients and providers manage care in the future. Whereas only healthcare providers can keep and maintain electronic health records (EHRs), PHR record systems are controlled by the patient. (Ogbuji et al., 2011, p. 10)

Regarding the terms of *Electronic Health Record* (EHR) and *Electronic Medical Record* (EMR) are often used alike. However, an EHR is slightly more complex and can contain one or more EMRs, or is mobile between stakeholders.

An EMR is a legal record created in hospitals and ambulatory environments, while EHR represents the ability to easily share medical information such as EMRs among various healthcare stakeholders and to have a patient's information follow him or her. (Varshney, 2009, p. 22)

## 2.2 Information Collection

Information collected by pervasive healthcare systems primarily consists of data related to health, lifestyle, and wellness. The data may come from various sources including the user himself.

The type of information that could be collected include the type of health monitoring, vital signs, prescribed medicines, sensory information, activities, environmental variables, and patient history. The information on missing doses, recent labs, known handicaps, and unusual conditions will also be very useful in health monitoring. The information may come from multiple sources such as sensors,

wearable and portable computers, some databases that may have information on the patient or from patient's input in some cases. (Varshney, 2009, p. 248)

The most common way to automatically collect health related information about an individual is *Health Monitoring*.

The health monitoring involves measuring multiple parameters simultaneously over a long-term without disturbing the daily lives of the patients. Such monitoring can be facilitated by using health monitoring devices inside patients (implanted), over patients (wearable), near to patients (portable) and around patients (environmental). (Varshney, 2009, p. 97)

### 2.3 Patient Empowerment

Patient Empowerment is an approach to get patients or individuals involved in their own healthcare with various techniques and technologies.

The term "patient empowerment" has been applied to a range of populations and settings. It is frequently used to describe a situation where patients are encouraged to be active in their own health management. Our viewpoint is that empowering patients can enable them to take more responsibility for managing their health and encourage self-management activities. Our view is also in line with Aujoulat et al., who put the patient perspective at the center of the empowerment process: "The goals and outcomes of patient empowerment should neither be predefined by the healthcare professionals, [...] but should be discussed and negotiated with every patient, according to his/her own particular situation and life priorities." Alpay et al. (2010)

An often used term in the context of Patient Empowerment is *e-Patient*, which has been brought into wider popularity in 2010 by Dave deBronkart:

Empowered, Engaged, Equipped, Enabled, Educated ... e-patients are effective partners with their clinicians, practicing what we now call "participatory medicine": "Participatory Medicine is a movement in which networked patients shift from being mere passengers to responsible drivers of their health, and in which providers encourage and value them as full partners. – Society for Participatory Medicine, April 2010" deBronkart (2010)



## 2.4 Interdependence

Health Records, Patient Empowerment, and Information Collection are highly ramified and therefore cause an interdependence. Alpay et al. (2010) grouped empowerment requirements into four areas, whereas the first three are showing this interdependence and are crucial for this thesis:

- *Insight into one's own health condition.*  
For example managing health related data personally "to better comprehend their medical history, health status, and health prognosis. For instance a personal health record (PHR), which facilitates entering, searching, personalizing and sharing personal health data, can support this data management. Further, the PHR can provide patients with access to medical data from the electronic patient record."
- *Making informed choices.*  
Deciding how to manage illness, "including choices about different self-care options (e.g., medical procedures, medication intake, and lifestyle changes) and their consequences. Online decision aids can support decision-making [...]"
- *Engaging in self-care activities and developing self-care habits.*  
"Health management options that patients choose can be translated into self-care goals and daily self-care activities, while maintaining good quality of life."
- *Living independently.*  
"Patients' personal environments also contribute to empowerment." This is especially interesting for the elderly population "to continue living independently in their safe and familiar environment."

(see Alpay et al., 2010, pp. 788–789)

As mentioned before, when referred to a PHR system there are often more functionalities included than just the Personal Health Record itself. These functionalities reflect the interdependence between PHRs and Patient Empowerment and are described by Tang et al. (2006) as *Health care management tools*.

Health care management tools are a relatively new form of PHR facilitators that may appeal to consumers by providing components of PHR systems: medication information, appointment information, care provider communication, and health

care knowledge resources. These can eventually include reminders and decision support as they develop into more complete PHR systems. (Tang et al., 2006, p. 125)

## 3 Requirement Analysis

In this section the key-requirements of the proposed system will be presented and discussed. For this purpose a literature research has been conducted in order to collect relevant information about stakeholders and their implications for requirements of the proposed system. Firstly, the stakeholders and their respective risks and benefits as well as the implications for requirements were collected. Secondly, the requirements of partial systems have been gathered and reviewed. Finally, these requirements were grouped into clusters by similarity and architectural influences identified. The resulting requirements were used for the system concept or architectural draft.

### 3.1 Stakeholder

There are three types of stakeholder for the proposed system, the users or patients, the clinicians and healthcare professionals, and the health insurances and payers.

#### 3.1.1 Users & Patients

Benefits of a PHR include patient access to health information. Since a PHR is a mandatory and an essential part of the proposed system, these benefits apply here as well.

One of the most important PHR benefits is greater patient access to a wide array of credible health information, data, and knowledge. (Tang et al., 2006, p. 123)

Along with access to health information comes the possibility to use this information to gain more control over ones own health.

People gain greater control over their health if they are informed, skilled, and motivated. (Alpay et al., 2010, p. 788)

With this control and level of information the patient is able to improve communication with healthcare professionals and participate more actively in his treatment. A more active patient is the most important part of the patient empowerment idea and is supposedly beneficial for clinicians as well.

Other important benefits [of a PHR; Author's note] are the improvement of the communication between patients and healthcare professionals., which promotes earlier intervention when problems or deviations occur and should make it easier for caregivers to care for patients. These are hypothetical benefits and to date there is limited evidence supporting them. However, many consumers have high satisfaction levels with existing early versions of PHRs. In particular, consumers place value on easy access to test results and better communication with clinicians. (see Tang et al., 2006, p. 123)

Contrary to these benefits, the user of the system is likely to be the payer, since the user should be in control of the system and thus of his own health information. It is however less likely for a user to finance the development of such a system, which will probably be done by health insurances or other investors.

#### **3.1.2 Clinicians & Healthcare Professionals**

The collected information can have a positive impact on the decisions of the clinicians in charge. Clinicians are able to base their decisions more accurately on the collected data and share information with colleagues in order to obtain a second or an experts opinion.

Having more data helps clinicians to make better decisions. The PHR may also become a conduit for improved sharing of medical records. (Tang et al., 2006, p. 124)

Additionally, even the activating effects of the patient empowerment idea have an impact on the clinicians, as briefly mentioned in the previous section.

Further benefits for clinicians include the more active and collaborative role of patients who are more engaged in their health. Tang et al. (2006)

Unfortunately there are numerous things that might have a negative influence on the acceptance of the proposed system and it's acceptance by health care providers. For example, the usefulness for health providers depends on the interoperability with EHR systems, namely those in use by the providers.

Notably, all the advantages of PHRs for providers depend on the PHR being integrated with the provider's EHR. (Tang et al., 2006, p. 124)

Additionally the idea of an empowered patient might be threatening for healthcare professionals and clinicians. There is further a risk of misleading information originating from the patient. These will require health providers and patients to develop a new relationship.

It is possible that PHRs will threaten the control, autonomy, and authority of some health care providers, based on traditional provider–patient roles. Providers and patients will need to develop different mindsets and levels of trust. Providers must learn to encourage patients to enter the information accurately and to trust that information appropriately. Consumers must trust that providers will only use the information for the individual’s benefit. (Tang et al., 2006, p. 125)

These risks through the loss of control apply to health insurances and payers as well.

#### **3.1.3 Health Insurances & Payers**

The highest costs for health insurances and health care payers are general caused by chronic diseases. Chronic disease management is not primarily a part of the proposed system. However, chronic diseases may spring from a negligence of physical activity, or mistreatment of acute illnesses. The proposed system might have the potential to prevent some chronic illnesses by motivating and helping the user to maintain a healthy lifestyle.

Potential benefits of PHRs to payers and purchasers of health care include lower chronic disease management costs, lower medication costs, and lower wellness program costs, although none of these has been well studied. The greatest area of benefit relates to the chronic disease management, where costs are typically high. (Tang et al., 2006, p. 124)

Health insurances and payers are in a unique position to actively promote these systems to a broader audience, but as mentioned in the section above, they might as well feel threatened by the idea of more involved and empowered patients and fear losing control, autonomy and authority (see section 3.1.2).

Moreover there are intentions to use employer sponsored PHR systems. This is especially interesting for countries where employers are providing a healthcare plan, or military facilities as a research conducted at an US Air Force base in Alaska shows (see Agarwal et al., 2013).

#### **3.1.4 Stakeholder Requirements**

It can be said, that there are the following requirements derived from the benefits and risks of the stakeholders. These requirements are yet quite vague and will be grouped into more

tangible requirements as well as architectural requirements in the following sections. Notably these requirements are from the perspective of the system, which will be operated by the users.

#### **Users & Patients**

**Ro1** *The user must have access to his own health information and data.*

**Ro2** *The user must be able to gain access to health knowledge through the system.*

**Ro3** *The system must allow communication between Patients and Healthcare Professionals.*

**Ro4** *Inexpensive access to the system, or the software must be given.*

Since the user has to pay for the finished system in order to use it and cover the development costs in retrospect.

#### **Clinicians & Healthcare Professionals**

**Ro5** *The system must be able to share health information with healthcare professionals.*

Give more information to healthcare professionals in order to make better decisions.

**Ro6** *The system must qualify for and encourage shared decision-making.*

Shared between healthcare providers and receivers for more cooperative and motivated patients.

#### **Health Insurances & Payers**

**Ro7** *The system must encourage a healthy lifestyle.*

In order to prevent chronic illnesses and lower the costs for health insurances and payers.

### **3.2 Health Record Keeping**

The most important requirement is the ability to keep a Personal Health Record. There are several open questions when it comes to a PHR regarding its features and limitations. The most common elements of PHR definitions have been taken as an entry point.

Common elements in the definitions of a personal health records establish it as electronic, patient-controlled, used for management of health information, and secure and private. (Fuji et al., 2012, p. 197)

Consistent with this summary the following sections will determine the requirements about the location, access and contents of the PHR component. Additionally patient opinions will be shown and connections to information collection and patient empowerment will be drawn.

#### **3.2.1 Location & Connectivity**

The location of a PHR needs to be somewhere in the control of the user and, self-evidently, on a computer system. Literature differs between three types of PHR locations and their respective connectivity. Namely these are *tethered*, *interconnected* or *networked*, and *stand-alone* PHRs. These three types are described by their properties as follows:

There are three different kinds of PHR systems. The simplest is a stand-alone application without and interconnection. The opposite is a tethered PHR system, which is basically a provider based record where the patient is allowed to access their own health information. Between these two extremes, there is the interconnected PHR which is a lot more complex and varies in its complexity depending on the level of connectivity with EHR systems. (see Tang et al., 2006, p. 122)

Stand-alone PHR systems are usually stored along with their data on home computers or portable storage devices, such as USB flash drives. Although they are normally controlled by the user, they bear several risks and disadvantages.

Although the stand-alone nature of such devices ["smart cards", USB drives, and CDs; Author's note] provides more individual control over access to the data contained in the PHR, the attendees [of the symposium about PHRs; Author's note] were concerned that, except for the most highly motivated, it is unlikely that individuals would keep records in a stand-alone PHR up to date. In addition it is unlikely that a stand-alone PHR that depends solely on patient input can act as a trusted conduit for transmission of medical record data among clinician offices or health care institutions. (Tang et al., 2006, p. 122)

Congruent concerns about stand-alone PHR systems have been summarized by Robison et al. (2012). Additionally they stated:

In addition to their limited functionality, they can pose a security risk. If the device is lost, the patient's medical record can be completely exposed. If the device were to be infected with malware it would be a threat to any provider computer it was connected to [19]. (Robison et al., 2012, p. 185)

Even if the stored data could be sufficiently encrypted as a countermeasure against exposed private data, the disadvantages are too numerous. Being up-to-date is one of the most important requirements of a PHR system and cannot be fulfilled by a stand-alone Personal Health Record system. Further problems may be caused by the limited update capability of the PHR system itself.

Tethered PHR systems and their data on the other hand are stored centrally on the computer systems of health providers and therefore solve the issues about being up-to-date, since they are stored at the source of reliable health information.

[...] PHRs integrated with EHRs, either through tethering or interconnectivity, provide much greater benefits than stand-alone PHRs. The integrated PHR-EHR approach can convey much more relevant data to the patient. (Tang et al., 2006, p. 122)

However, they are not PHRs per se, since the data is not entirely controlled by the user. Their true location and accessibility depends solely on the health provider. Additionally they may risk fragmentation of the user's health records caused by the heterogeneous nature and flawed interconnectivity of the EHR systems.

A hybrid of tethered and stand-alone systems are the networked or interconnected PHR systems. Networked PHRs are de facto controlled by the user and are able to receive health information from an EHR system.

A networked PHR allows for the possibility of providers contributing to the health record directly, or for the PHR to import data automatically from the one or more provider EHRs. (Robison et al., 2012, p. 185)

A networked PHR is currently the most popular system, since it allows up-to-date information is controlled by the user. Therefore new requirements can be defined as:

**Ro8** *The user must have complete control over his data.*

**Ro9** *The system must allow the import of EHR data.*



### 3.2.2 Access & Interoperability

The proposed system requires interoperability in order to be useful for the user. In this section different viewpoints regarding access to personal health information will be shown. These viewpoints include security, privacy, interoperability, third party access, and the choice of a frontend technology.

#### Security & Privacy

As previously explained, the ownership should lie solely in the users hands. Without doubt the data contained in a PHR is private and should be safe. Users might have trust issues when it comes to the storage of personal information in a system, which they don't entirely understand. Therefore the protection and privacy of the personal information and it's transparency is of utmost importance. But it is further required, that PHR data can be shared with other people or systems, such as healthcare professionals and their EHR systems. Kahn et al. (2009) are mentioning the importance several times:

[...] the NCHIT's [National Coordinator for Health Information Technology (USA); Author's note] definition for the ideal PHR includes the notion that a PHR can be "managed, shared and controlled by the individual." Making PHR data portable for consumers when they move among health organizations represents a key factor in wider and more rapid PHR adoption. (Kahn et al., 2009, p. 372)

Data security and identity protection are critical issues and are central for widespread consumer acceptance and adoption of PHRs. (Kahn et al., 2009, p. 373)

Consumers who adopt PHRs will require that the information be protected and private; that ownership lie solely with the consumer; that storage and use of the data be approved by the patients; and that the data be easily portable and in a format that is understandable. (Kahn et al., 2009, p. 375)

The control and ownership of the PHR are considered to be duplicates of **Ro8**. The sharing capability has been introduced in section 3.1.4 (Stakeholder Requirements) as **Ro5**.

A new requirement will be defined for the security of of personal information.

**R10** *Privacy and security of personal information must be given at any time.*

### **Authorization & Authentication**

In order to share information with anyone else but the user, these who seek access must be authorized by the user.

PHR systems are designed to share information only with those whom you authorize. (Trotter and Uhlman, 2013, p. 82)

Therefore the system requires some kind of authentication system for authorized access.

**R11** *The user must be able to authorize access to his PHR for other users.*

Besides the importance of security it should be kept in mind, that often security is a trade-off for usability and ease-of-access.

While consumers appropriately desire protection of their private health information, aggressive protection measures might hamper PHR access by patients and clinicians and impede optimal care. (Tang et al., 2006, p. 125)

Especially complicated authentication procedures might diminish the motivation to use the system and therefore decrease potential benefits. Consequently a new requirement is needed to maintain a certain level of ease-of-access.

**R12** *User authentication must be as simple as possible.*

### **Interoperability with Health Information Systems**

For the proposed system it is important to be able to receive and transmit health information to and from HIS (Health Information Systems). The most important HIS are EHR systems, which can be used to retrieve up-to-date health information from health providers.

At a minimum, PHRs must export data to and import data from other systems in a standardized way. (Tang et al., 2006, p. 124)

Modern thinking about PHRs presumes that useful data should flow bidirectionally between PHR and EHR systems. (Trotter and Uhlman, 2013, p. 71)

Whereas the two directions both promote up-to-date information of PHRs, EHRs, and consequently the healthcare professionals involved in the users wellbeing, there are different requirements depending on the direction of the information sharing. For instance, when

receiving data from an EHR system, the user should review the data and, if appropriate, accept the data import. If the user is about to share data with an EHR system, he should be able to select exactly which data he is willing to share.

A [...] concern is how to allow individuals to specify which of their own data they will allow to be shared with other health information systems. (Tang et al., 2006, p. 124)

In order to achieve this in an adequate way, the user should be able to choose the level of granularity and review possible dependencies of the data he is about to share.

The system should further warn the user if the amount of shared information exceeds an amount comprehensible by most healthcare professionals, or if the shared information is likely to be irrelevant for the targeted health provider; e.g. Information about a patients broken ankle is usually not helpful for a pulmonologist<sup>1</sup>.

[...] the volume of “clinically irrelevant” information in their patients’ PHRs might become overwhelming for a health care provider to review. (Tang et al., 2006, p. 125)

The use of common standards between health information systems (e.g. EHR and PHR systems) is required in order to achieve interoperability with those systems.

To provide interoperability, PHRs must support the same communications, messaging, and content encoding standards as other health information systems. (Tang et al., 2006, p. 124)

These standards are mainly HL7 (Health Level 7) standards. HL7 is a framework with related standards for the exchange, integration, sharing, and retrieval of electronic health information. It is maintained by the non-profit organization "Health Level Seven International". Among the primary standards is the CDA (Clinical Document Architecture) for structure and semantics of clinical documents for the purpose of exchange between healthcare providers. CDA includes another standard called CCD (Continuity of Care Document), which fosters interoperability of clinical data without loss of meaning (see Health Level Seven International, 03.09.2013).

Another standard is the CCR (Continuity of Care Record), which is basically a subset of a CCD. It contains healthcare information about a patient, covering one or more healthcare encounters, practitioners, systems, etc. The CCR standard is maintained by ASTM International

---

<sup>1</sup>A pulmonologist is a specialist for pulmonary or respiratory medicine, for example in case of lung diseases.

(formerly known as the American Society for Testing and Materials). However, since the CCD standard is a joint effort of HL7 and ASTM and an implementation of the CCR standard, the CCD standard is preferred. The CCR can be translated into a CCD, since a CCD contains more information (see ASTM, 30.08.2013; Health Level Seven International, 03.09.2013).

Another approach is described by Tang et al. (2006), who are achieving interoperability through integration.

More advanced PHRs will at some future time function as seamlessly integrated, interoperable "components" of other health systems. (Tang et al., 2006, p. 124)

A seamlessly integrated PHR system in other health systems would drastically improve and ease the automatic exchange of data. However, it might be difficult to guarantee the user's control over his health information if the PHR is a component of another health system. In this work it is one of the aims to keep the control over the proposed system and the health information completely in the user's hands. Besides, the automatic exchange of healthcare data is an extensively discussed topic.

The automatic exchange of healthcare data using computerized processes is one of the most heated policy areas in modern healthcare informatics. HIE [Health Information Exchange; Author's note] policy discussions are often heated and circular. (Trotter and Uhlman, 2013, p. 201)

Due to the fact that there are many opinions about this topic, there should be no automatic health information exchange between the proposed system and other health information systems.

Another approach has been described by Swan (2009), which depicts a health record system as the mediator of data for clinicians and other relevant parties:

Eventually, health social networks, health portals [...] or other EMR-specific websites [...] could be the application frontends and centralized repositories of quantitative patient data. These websites could orchestrate EMR access permissions to doctors and other parties. (Swan, 2009, p. 514)

Lastly, a user should be able to share his health information with health providers, that have not yet sufficient access to EHR systems. This is best achieved by offering a simple download functionality for stored health information.

Patients will need to be able to download this [electronic health information; Author's note] in human readable (e.g., a PDF) and digital (e.g., CCD/CCR) formats. (Trotter and Uhlman, 2013, p. 84)

It should further be possible to print the human readable file. For the sake of usability, the printing functionality should be included in the system in order to reduce the dependency of third party software.

As a summary the following requirements are defined based on the statements above:

- R13** *The user must be able to share data from his own PHR with EHR systems.*
- R14** *Anytime when the user is sharing data, he must be able to specifically select the data he wants to share.*
- R15** *The user must be able to download/export data from his own PHR to human readable files in various formats.*
- R16** *The user must be able to download/export data from his own PHR to machine readable files and standardized formats.*
- R17** *The user must be able to print selected health information from his PHR.*
- R18** *The user must be able to import data into his own PHR from EHR systems.*
- R19** *Health information exchange must always require the explicit approval of the user.*
- R20** *When a user attempts to share data, the data must be automatically checked for integrity, amount, and usefulness.*

Depending on the target system and it's assumed user. In case of issues regarding data integrity, amount, or usefulness, the user must be required to confirm the sharing of data again in order to start the data export. If the confirmation is denied, the export process has to be aborted.

#### **Access for Family & Caregivers**

In some cases, besides patients and health providers, even third parties may require access to the PHR. For example family members or caregivers in order to maintain the PHR for person that may currently or permanently not be able to do so.

PHRs must also support and trust "designated caregivers", such as parents in the case of young children, or spouses in the case of incapacitated adults. (Tang et al., 2006, p. 124)

In case of a temporary caregiver the access should be approved by the user himself. An emergency access might be required in case the user is incapacitated. Obviously, the access to medical data in an emergency can be abused and therefore is a trade-off between privacy/security and its benefits.

The access to EMR in emergency is really is [sic!] a trade-off between privacy and benefits [...] (Varshney, 2009, p. 31)

Two new requirements are defined:

**R21** *The system must allow access to designated caregivers.*

**R22** *The system must allow parental access in case of children.*

#### **Frontend**

For many years web-applications have been omnipresent and are still the most commonly used technology for PHR systems and their frontends. Another technology who has shown a significant increase in popularity are applications for smartphone platforms. Since a mobile phone is typically permanently carried around by the user, it can be as omnipresent as a web-application (which can usually be run in a web-browser on the smartphone as well), but further add the usability of a native application and user-interface, and further utilize its integrated sensors.

The most important disadvantage, regarding mobile device applications, is the number of different smartphone operating systems, such as Android, iOS, Windows Phone and numerous others that are in development. Most of these operating systems currently require dedicated implementations, which causes high maintenance costs.

[...] web-based applications are predominantly the common format for standalone PHRs. While smartphone-based PHR applications continue to grow, many either have a cost to download and/or are not supported by multiple operating systems [...]. (Fuji et al., 2012, p. 198)

Ideally the frontend application should not be bound to a certain technology or device. However, since user interfaces often utilize libraries of the operating system, at least differences in the appearance are inevitable.

According to the beneficial use of mobile technologies, a new requirement is defined:

**R23** *The frontend application must not be bound to a certain technology.*

Frontends must be able to be implemented for different devices and technologies, such as web-based or as an application for different operating systems (e.g. Windows, Linux, OS X, Android, iOS, Windows Phone, etc.).

#### 3.2.3 Contents

In this section the requirements regarding the contents of a PHR are discussed. It is of utmost importance to keep as much data as possible in the health record in order to avoid a delay for the data acquisition.

A fragmented system of storing and retrieving essential patient data impedes optimal care. (Tang et al., 2006, p. 121)

The different data contained in a PHR, as well as its properties are discussed in this section.

#### Multiple Information Sources

Information can come from many different sources like different healthcare institutions, devices, the user himself, and other people. Hence, to keep the PHR up-to-date and relevant, the system must allow multiple information sources.

Ideally, the PHR should include as much relevant data as possible over the individual's lifetime, from multiple sources, including health care facilities as well as the individual. (Tang et al., 2006, p. 122)

A lot of data from lots of different sources needs to be organized. In order to do so, it is advised to annotate or label the received data according to its source.

The specific data source of each item should be labeled and visible to the user. (Tang et al., 2006, pp. 122–123)

As a first distinction of relevant data, one can differ between subjective and objective PHR data. Examples for objective data are blood pressure and weight. Subjective data for instance

are symptom scores or qualitative descriptions (see Tang et al., 2006, p. 123). The source of subjective data is typically the user himself. The source of objective data are healthcare institutions, laboratories, or devices.

According to this information four new requirements are defined:

- R24** *The system must allow PHR data from Health Information Systems.*  
Therefore originally workers in healthcare facilities or laboratories.
- R25** *The system must allow PHR data from Individuals.*  
Such as the user himself, caregivers, or family and friends.
- R26** *The system must allow PHR data from Devices.*  
For example monitoring devices, measurement devices, etc.
- R27** *The system must automatically annotate received data according to its source.*  
The annotation must differ between subjective and objective data, as well as the type of source (e.g. healthcare institution, person, device), and the exact name of the source (e.g. specific blood pressure monitor model).

#### **Data from the User**

One of the most important advantages of a PHR compared to an EHR is the explicit presence of user submitted data. This data can serve many purposes, for example add qualitative information, correct previously submitted information, or comment on certain parts of the health record.

[. . .] accepting patient-generated data even for simple record correction or commentary will be a huge technical step forward. (Trotter and Uhlman, 2013, p. 85)

Unfortunately user submitted objective data is not always reliable. Some measurements can only be accurately undertaken by healthcare professionals, like the measurement of certain blood values. Reliable information from users include subjective data and objective data which can be measured easily and accurately by the user himself.

[...] patient reports are usually reliable for symptoms and easy-to-measure objective parameters, such as height, weight, and temperature by thermometer. (Tang et al., 2006, p. 122)



Tang et al. (2006) only mentions height, weight, and temperature as easy-to-measure parameters. However, new developments of easy-to-use self-testing devices enable users to perform even previously complicated measurements accurately. This allows the user to submit more accurate data to their PHR, rather than just basic physical properties of themselves.

Another very important factor for patient submitted information is the logging of over the counter (OTC) drugs. Over the counter medications do not require a prescription and can therefore be sold without the notice of a healthcare professional. These OTC drugs still may cause health issues or interact with prescribed drugs and as a result cause harm to the patient. Consequently it is important to track the usage of OTC medication in order to maintain a complete picture of the current health situation.

[...] if a patient was regularly taking an over the counter medication but lacked the ability to add that information to their record in an EHR, they might not be warned of a potential drug interaction with a prescription medication. This can particularly be a problem if the patient visits a doctor that does not use the patient's EHR. (Robison et al., 2012, p. 185)

The previously mentioned subjective data includes observations done by the patients, for instance symptoms. Subjective observations can still be an useful contribution to a PHR as Puentes et al. (2013) states:

Conventionally, nurses or healthcare assistants make observations about patients' status under particular circumstances. Such exercise implies certain training not commonly acquired by individuals making use of pervasive health systems. In addition, individuals' subjectivity plays a nonnegligible role when describing the degree of certain situations. Nevertheless, if properly formulated, these data can provide additional useful information, suggesting that personal observations should be added to sensors data in the PHR. (Puentes et al., 2013, p. 114)

Additionally individuals have the opportunity to observe themselves constantly, which is only possible for healthcare professionals in case of the individuals hospitalization.

Hence, a new requirement is defined:

**R28** *The system must allow user-submitted information and must not limit the information items, e.g. what data can be submitted.*

Since any health related information has the potential to be useful. Current important information contain: Comments on records or documents in the system, Measurement values, Symptoms, Over-the-counter medication.

#### **Data from Clinicians and Healthcare Professionals**

In section 3.2.2 the importance of interoperability between health information systems has been shown. As a consequence data from clinicians and healthcare professionals may already be part of the PHR, if it is part of an interoperable EHR system.

However, additionally the system should allow access to healthcare professionals who do not maintain an EHR system in order to submit and access and submit findings and considerations, as well as other relevant information.

[...] clinicians must also have access to their own past considerations and interpretations, as well as reliable objective data, if they are to depend on records for clinical decision making. (Tang et al., 2006, p. 122)

As a consequence a new requirement is defined:

**R29** *The user must be able to allow guest access to his PHR.*

In order to allow healthcare professionals without an interoperable or supported EHR system to access information on the users PHR.

#### **Fitness**

Fitness and prevention of illnesses by maintaining a healthy lifestyle are certainly important factors and should therefore be supported by the system. In order to do so, fitness data needs to be tracked and persisted in the PHR. Fitness and lifestyle information include activities, nutrition, and certain vital signs and measured parameters, such as weight, body fat, blood pressure, ECG or heartbeat values, etc. A pedometer, which is a device used to count steps a person walked, is a very popular example for fitness information that can be easily retrieved.

An interesting approach regarding information used for fitness by Andrew et al. (2013) is showing the meaningful combination of location data. The position paper states that available components "can be combined to create a location-based lifelogging platform for food and activity behaviors." It describes an approach to derive meaning from location data in order to support monitoring and possibly lead to behavioral changes regarding food and activities. (see Andrew et al., 2013)

Location is certainly not a health related information, but it can be utilized to add a context to collected information, just like other environmental information, which is covered in the following subsection.

#### **Environmental & Temporal Information**

Some health problems can be caused by other influences, such as contact with other people or certain environmental settings. As a consequence these influences should have a place inside the systems PHR as well.

In addition to tracking biomarkers and behavior, one's environment is the next logical area to monitor for personal health, resource utilization and other reasons. (Swan, 2009, p. 511)

Environmental influences are large in number. To name just a few, there are values about *Time, season, air pressure, temperature, humidity, altitude, biome classification, light intensity, noise exposure, geographical position, movement, acceleration etc.*

Whereas time is certainly the most important, since it is commonly used to order data chronologically.

Temporal data are the core reference that enable data studies through time [...]. This is the most critical global reference, [...]. (Puentes et al., 2013, p. 108)

As mentioned before, even other people can be an influence to ones health; Not only physical, but also psychological. These social influences include relation statuses between people and partially their own health information, e.g. in case of contagious or mental diseases as well as addictions.

Beyond an individual's personal data, PHRs may include relevant information about family members, caregivers, and home and work environments that are important to the individual's health. (Tang et al., 2006, p. 124)

The inclusion of private healthcare data is a trade-off between the usefulness of the data, due to it's integrity and completeness - and the privacy of these people, whose health data must be included or referenced in the users PHR system in order to complete the image. However, anonymous integration of peoples social influences may work to certain degree.

Four new requirements are defined regarding fitness and environmental information:

- R30** *The system must be able to persist and track fitness and lifestyle information.*  
Such as activities, nutrition, and related vital signs and measured parameters.
- R31** *The system must be able to persist and track data about environmental conditions in their respective context.*  
For example noise exposure and the time and location as context. The system must be able to handle a large variety of data.
- R32** *The system must allow the user to include social relations.*
- R33** *The system must be able to anonymously retrieve health information about social contacts of the user, if available, and persist this information as long as it is relevant.*  
After this period the data must be removed for privacy reasons.

#### **Information Items for Medical History**

In order to draw a holistic picture of a users medical history, there are some important information items, which should be present in any PHR. A well known example are immunizations. For instance, it is possible to keep track of disease outbreaks in order to advise the user to receive his immunization or refresh an old one.

Tang et al. (2006) summarized the findings of a symposium about PHRs and listed several items which should be included in any PHR:

Problem list, Procedures, Major illnesses, Provider list, Allergy data, Home-monitored data (e.g., BP, glucose, peak flow), Family history, Social history and lifestyle, Immunizations, Laboratory tests. (see Tang et al., 2006, p. 123)

Another listing of items for PHRs has been assembled by Gonzales and Tran (2012). The paper proposed 14 data components for PHR standards or guidelines, grouped into categories for critical (C), desired (D), and recommended (R) components:

Immunizations (C), Medications/Prescriptions (C), Allergies/Adverse Reactions (C), Family History (C), Lab/Test Results (C), Procedures/Surgeries (C), Health Providers (D), Insurance/Payer Info (D), Social History/Lifestyle (D), Problems/Diagnoses/Conditions (D), Clinical Encounter (D), Vital Signs (R), Care Plan (R), Home Monitoring (R). (see Gonzales and Tran, 2012, p. 2)

Based on a union of these two lists a set of mandatory information items can be deduced as listed in the following requirement.

**R34** *The PHR must be able to contain the following information items:*

- (a) Problems/Diagnoses/Conditions,*
- (b) Procedures/Surgeries,*
- (c) Major illnesses,*
- (d) Health providers,*
- (e) Home-monitored and self-measured data,*
- (f) Family history,*
- (g) Social history and lifestyle,*
- (h) Immunizations,*
- (i) Laboratory tests and results,*
- (j) Medications/Prescriptions,*
- (k) Allergies/Adverse reactions,*
- (l) Insurance/Payer info,*
- (m) Clinical encounter,*
- (n) Care plan*

#### **Information Integrity**

The collection of information from several sources may cause redundant and even controversial data. In addition users with limited health literacy or users with limited computer skills may accidentally add information wrongly. This is critical when the user, or even clinicians and healthcare professionals base their decisions on this data. As a countermeasure Varshney (2009) suggests warnings for patient submitted information, if it causes conflicts.

Some automated system may also be needed to generate warnings for conflicts, created by information entered by the patient. (Varshney, 2009, p. 24)

It is however possible, that data originating from different EHR systems may also be mutually exclusive, for instance in case the patient visited another clinician for a second opinion. As a result the system should be able to check for any kind of inconsistencies within it's PHR data.

**R35** *The system should automatically check for information inconsistencies.*

Every time new data is added to the PHR.

Moreover, some information may become invalid or outdated over time and therefore is wrong information, which is why the following requirement is added:

**R36** *The system must keep time-limits regarding the validity of it's data.*

If the time-limit has been passed, the user should be reminded to take action in order to keep the PHR up-to-date and mark the information explicitly as outdated.

#### **Changelog**

As mentioned in the previous section, what is true may change over time, and so does the information itself. However, some information items are rarely changing, for instance the users name, eye color, or gender. In normal IT-systems, changes of a users name or gender often requires the manual alteration in a database performed by an administrator. In a PHR system on the other hand, these changes should be allowed to be done by the user himself. In order to keep track of these changes, they should be noted or logged.

Normal IT systems might not allow users to update and change these "static" personal information data points [e.g. names, eye color and gender; Author's note], but EHR systems must allow these changes to occur and also track them over time. (Trotter and Uhlman, 2013, p. 57)

Moreover, a general changelog should be kept in order to make changes in the PHR more transparent to the user and caregivers.

**R37** *The system must keep a changelog about all changes of PHR data.*

For the sake of usability, these changelog entries should be categorized.

#### **3.2.4 Patient Opinions**

Since the user is the stakeholder with the highest benefits and risks, due to his health's involvement, a look has been taken at current research about patients opinions regarding PHR systems.

An online focus group by Baird et al. (2011) with an effective sample size of 463, showed several Disadvantages, Advantages, and Usage Triggers for PHRs, as well as their respective Agreement and Importance percentages. One of the significant results of this online focus group was a high agreement and importance percentage for the disadvantage of "*Privacy and security of health information*" (91% agreement, 76% importance). Further disadvantages relevant

for this work included "*Learning curve*" (36% agreement, 15% importance) and "*Technology not mature enough*" (34% agreement, 15% importance).

Relevant advantages of PHR included "*Family members can control my health information for me*" (78% agreement, 58% importance), "*Single location for all records*" (76% agreement, 46% importance), and "*Easier to track history*" (58% acceptance, 35% importance).

The top 5 usage Triggers were "*Complex medical situation*", "*If improved health outcomes are observed*", "*Chronic or severe illness*", "*Have children*", and "*Lower healthcare costs*". (see Baird et al., 2011, pp. 283–284)

Another approach was chosen by Fuji et al. (2012), who researched patient desires of PHR systems and the coverage of these desires by current PHR systems. After the identification of patients desires, a checklist has been created and 19 different standalone PHRs, which are free-of-charge for all features and web-based, have been evaluated according to this checklist.

The patients' desires from this checklist are listed below. Some of the desires have already been defined in previous requirements, but some of them are new and will be defined as new requirements subsequently.

1. *Sharing health information in the PHR with others.*  
Sharing with Healthcare Professionals defined in **R05**.  
Giving access to other users defined in **R11**.
2. *Receiving feedback based on entered health information.*
3. *Information presented in layman's terms.*
4. *Ensuring the security and privacy of health information.*  
Ensuring privacy and security of the users information defined in **R10**.
5. *Communicate directly with their health care providers using e-mail or secure messaging.*  
Allowing communication with health providers defined in **R03**.
6. *Interoperability with the provider based record.*  
Importing data from EHRs defined in **R09**.  
Exporting data to EHRs defined in **R13**.
7. *Generate a printed report of health information.*  
Printing health information defined in **R17**.
8. *Create new sections in the PHR for additional information such as donor wishes.*  
Allowing unrestricted user-submitted information items defined in **R28**.

9. *Customize the visual appearance of the PHR.*
10. *Restricting access of individuals to only view specified types of health information.*  
Restricting access to individuals defined in **R14**.
11. *Having personalized support in the PHR based on an individual's abilities and needs.*  
The personalized support is implicitly defined by the requirements that are discussed in section 3.4 (The "e-Requirements").

(see Fuji et al., 2012, p. 202, Table 5)

For the desires 2., 3. and 9. new requirements are added respectively:

- R38** *The system must present information to the user in layman's terms.*  
In order to make the information easily comprehensible.
- R39** *The user must receive feedback from the system based on information saved in the PHR.*
- R40** *The user must be able to customize the appearance of the user-interface.*

#### 3.2.5 PHR and Data Gathering

The purpose of a PHR system is to show the users health, past and present, in order to make decisions beneficial for the users health. It is therefore very important to keep the PHR as complete and up-to-date as possible. The user should maintain his PHR actively and consistently, which may seem to the user like an unwanted duty or even a burden.

While some standalone PHRs may connect with various health care providers or payors, the patient is often solely responsible for entering in all of their health information. (Fuji et al., 2012, p. 198)

The benefits on the other hand are considerably high. The user would get a self-controlled lifelong health record, which is independent from his location.

If the patients have not added/updated all information or have not imported all medical information from [sic!] one or more sources, the usefulness [...] may be limited. However, on the other side, such PHR can lead to lifelong medical records for patients and allow for continued care in case of patients moving to locations where their EMR/EHR can not be accessed. (Varshney, 2009, p. 24)



In order to help the user to enter up-to-date information into the PHR, the proposed system should have a sub-system, which automatically gathers information from different sources and persists them into the users PHR. This sub-system has been found to bring a lot of own requirements and is extensively discussed in section 3.3 (Information Collection).

#### **3.2.6 PHR and Empowerment**

PHR systems and the patient empowerment idea are complementing each other. In this section, some examples are given in order to draw the connection and show the need of a system that conjoins these concepts.

Firstly, patients with access to their own health data are able to increase their awareness or literacy. This is usually done by asking different physicians about a second opinion, or participating in communities.

When consumers have access to their data, they will be able to use the information in health applications on the Web, obtain second opinions, or discuss their concerns with others who may have similar conditions. (Kahn et al., 2009, p. 372)

Secondly, the users may use health management tools, such as PHR systems, in order to use their own health information to gain new information about treatments, communicate with healthcare professionals, or get information about their current treatments and medication.

Most studies that have evaluated Internet applications for health suggest that patients are ready to use these tools [health management tools, such as PHR systems; Author's note], and the most-anticipated Internet applications include access to information on new treatments, e-mail communication, and medication information. Some research groups also identified the importance of sharing information through social networking and the value of learning from people like themselves. (Kahn et al., 2009, p. 371)

Finally, the last statement is addressing an issue which is becoming increasingly more important regarding online-sharing of personal information. In the age of data mining where personal information can be literally mined from online sources, some users have concerns about sharing their personal information online. These concerns are significantly stronger when it comes to personal health information. However it seems there are different opinions, possibly influenced by cultural beliefs. Trotter and Uhlman (2013) state, that data retrieved by data mining is typically used to draw conclusions about populations and not the individual.

For the most part, publishing health information online is perfectly safe because no one is paying attention. Or rather, the attention that publicly posted health data gets is general use for data mining purposes. It is used to make conclusions about populations rather than conclusions about individuals. (Trotter and Uhlman, 2013, p. 81)

In any case, the shared information of a user should be only visible to certain people and never the whole public. In order to publish data publicly, for instance to create meaningful statistics about populations, the data should be anonymized.

Hence, two requirements are defined:

- R41** *The system must allow the user to share his information online.*  
But must take precautions in order to prevent publicly visible personal information.
- R42** *The system must allow the public sharing of anonymized information.*  
e.g. to help populate statistics for research purposes.

### 3.3 Information Collection

In the previous section it has been shown that a PHR contains different information from different sources. Since most PHR systems include the data acquiring methods, this work is aiming on a separation of concerns and will show in this section the requirements needed to acquire information which may be ultimately persisted in the PHR.

There are several techniques in order to gather information as mentioned in the introduction in section 2.2 (Information Collection). For instance health monitoring, which may help to prevent incidents or in long term monitoring, help to understand the disease better and improve clinician-patient communication, as stated by Bardram et al. (2007).

It is believed that the ongoing monitoring of vital signs will help healthcare professionals to identify healthcare crises before they occur. In this case, the technology would help the health professional observe trends in bodily functioning [..]. If changes in bodily function can be identified by changes in monitored vital signs, it is possible that early medical attention could be sought before a critical event, such as a stroke or heart attack, occurred. (Bardram et al., 2007, p. 207)

Long-term monitoring may also help to understand the course of the disease better. Hence, various different technologically based approaches have been developed to support the individual in taking a more active role in the disease management process and to improve patient communication with the clinician. (Bardram et al., 2007, p. 120)

A famous example showing the importance of continuous monitoring is the so called "white coat syndrome", which describes increased blood pressure values from measurement, when done by a doctor or similar health providers.

[...] 20 to 35 percent of patients may experience "white-coat hypertension," where they are hypertensive in the physicians office but normotensive at home. Therefore, continuous monitoring of a patient's blood pressure is extremely important for the physician to determine which drug therapy regime will work best with each individual. (Bardram et al., 2007, p. 146)

Generally this is an important issue also mentioned by Elizabeth Mynatt<sup>2</sup> as a keynote speaker at the PervasiveHealth Conference 2013<sup>3</sup>. In her keynote speech she pointed out, that values normal for most of the population do not necessarily mean to be normal for an individual. An example was given by a person that was having a weak case of hypotension (low blood pressure) and was starting to measure values close to the upper limit of what is a normal blood pressure for the general population. Since that persons blood pressure has increased significantly, it remained inside the boundaries of normal values and therefore didn't cause the clinicians to be alarmed. The essence of the speech was the question: "Was today a normal day?" - which can only be answered accurately when information is put into relation with previous experiences and not compared to the norm of the general population.

In order to find out if today was a normal day for an individual, this work is relying heavily on personalized and individual information collection.

#### **3.3.1 Methods of Information Collection**

As mentioned in the introduction to Information Collection (Section 2.2), health monitoring is one of the most common ways to automatically collect health related information. Varshney (2009) showed an overview of health monitoring types and their contexts in the following

---

<sup>2</sup>Elizabeth Mynatt, Executive Director, Institute for People and Technology, College of Computing, Georgia Institute of Technology

<sup>3</sup>7th International Conference on Pervasive Computing Technologies for Healthcare, May 5–8, 2013 Venice, Italy

diagram:

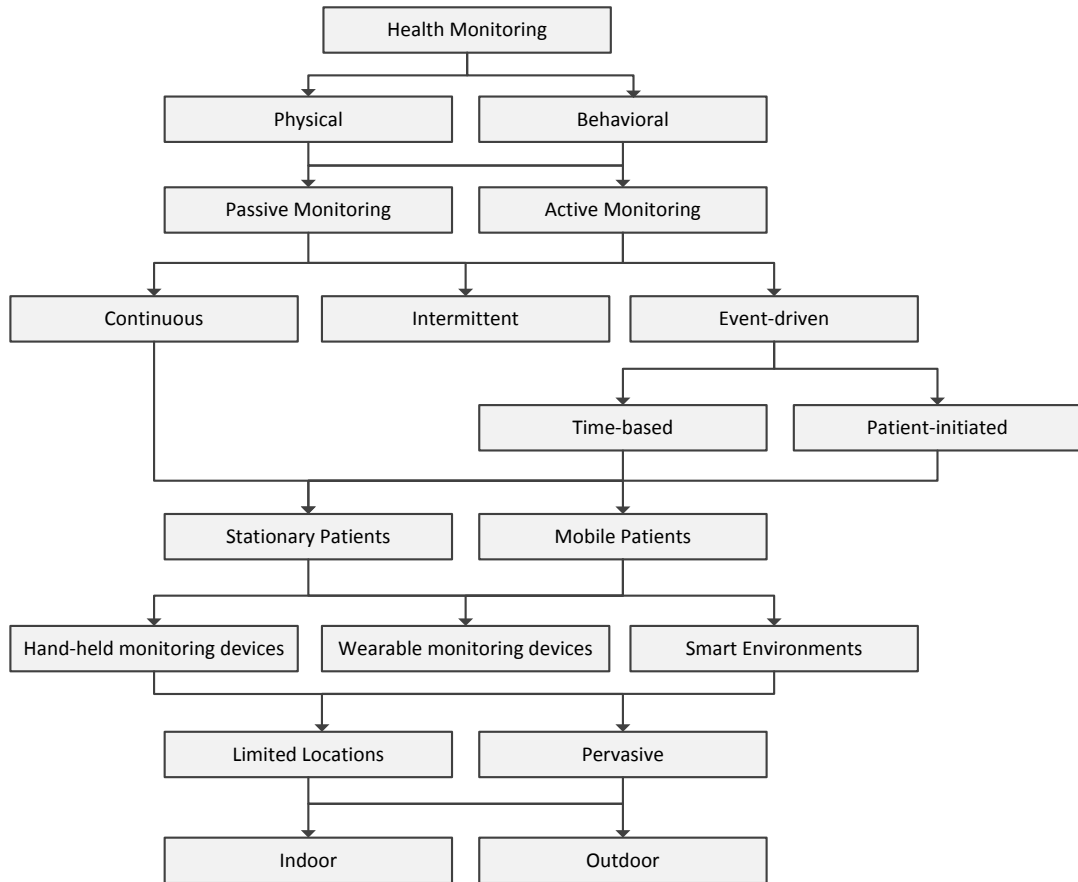


Figure 3.1: Context from the Type of Monitoring (see Varshney, 2009, fig. 11.2; p. 244)

Therefore it is important to support more than just one source of health monitoring.

**R43** *The system must support multiple health monitoring sources.*

More sophisticated approaches facilitate health monitoring to allow the user not only to collect information automatically, but further annotate the collected information. The annotation may happen manually by the user or automatically, if the system supports it.

### 3.3.2 Lifelogging & Quantified Self

Two of these more sophisticated approaches are Lifelogging and Quantified Self (QS), or Quantified Self-tracking (QST). Both approaches are not limited to health related information, but can be used as an excellent basis for health information collection including information allowing to keep the context, if needed.

Andrew et al. (2013) gives brief definitions to the two approaches and their difference:

Lifelogging is loosely defined as capturing and storing detailed information about one's life. (Andrew et al., 2013, p. 408)

The Quantified Self (QS) community presents a slightly different approach to lifelogging. QS enthusiasts typically begin with the goal of capturing and understanding one (or a few) very specific, well-defined aspects of their life. The intent is to better understand or improve something about of interest. (Andrew et al., 2013, p. 408)

For our purpose the aims of the Quantified Self movement seems to be more congruent and more aware of the empowering capabilities.

Quantified self-tracking is the regular collection of any data that can be measured about the self such as biological, physical, behavioral or environmental information. (Swan, 2009, p. 509)

An underlying assumption for many self-trackers is that data is an objective resource that can bring visibility, information and action to a situation quickly, and psychologically there may be an element of empowerment and control. Quantified self-tracking is being applied to a variety of life areas including time management, travel and social communications, as well as the health context, where the expanded definition of health is embraced as applications address both medical issues and general wellness objectives. (Swan, 2009, p. 509)

Trotter and Uhlman (2013) already concluded the usage of fitness and health data gathered by QS for health record systems, which is an essential part of this work.

Although the most extreme quantified self practices will probably never take hold in the general population, real-time health and fitness data will soon become a natural part of EHR and PHR systems. (Trotter and Uhlman, 2013, p. 79)

Quantified self-tracking further encourages the inclusion of qualitative or subjective information and the utilization for visual representations. The latter will be discussed more detailed in a later section.

Additional aspects [of quantified self-tracking; Author's note] may include the graphical display of the data and a feedback loop of introspection and self-experimentation. Health aspects that are not obviously quantitative such as mood can be recorded with qualitative words that can be stored as text or in a tag cloud, mapped to a quantitative scale, or ranked relative to other measures such as yesterday's rating. (Swan, 2009, p. 509)

As a consequence the definition of a new requirement:

**R44** *The system must be able to utilize quantified self-tracking and lifelogging systems. For the purpose of data collection.*

#### 3.3.3 Data Acquisition

Both approaches, Lifelogging and Quantified self-tacking, require the automatic collection of data. The most intuitive way to achieve this is by wearable measurement devices.

The next step in quantified self-tracking health applications is wearable devices with automated data collection [..]. (Swan, 2009, p. 510)

The most popular and solid way to get digital data is to use sensors, which are measuring a simple basic property and send the digital signal to an application for aggregation or processing.

Sensors can be implanted, on-the-body and be part of a wearable monitoring systems [sic!]. (Varshney, 2009, pp. 80–81)

On the other hand, non-wearable measurement devices could be for example scales or a laser-rule for measuring body height or other length related information. All of these devices for self-measurement should be able to send the data to a computer system by themselves, otherwise the user has to add the measured values himself into the system, which can be forgotten and therefore cause a decrease in motivation and threaten the usefulness of the system. In this work the mentioned computer system should collect the data inside of the PHR.

A future can be imagined where it would be standard for all self-measurement devices, for example heart rate monitors, to have an option for wireless network

connectivity to automatically download data to a consolidated format for storage, display and analysis. (Swan, 2009, p. 511)

But not only measurement devices help to gather information. Another often overlooked source are of daily use. For instance, some information may be acquired from calendars or mobile phone usage.

Modern technology in the form of spreadsheets, websites and mobile phone apps have made it easier to capture, analyze and share data. (Andrew et al., 2013, p. 408)

A combination of measurement devices, sensors and other information sources, can be found in the area of Ambient Assisted Living (AAL). AAL usually aims to improve quality of life for the elderly or people with mental diseases, such as Dementia or Alzheimer's. These people are often not able to perform basic Activities of Daily Living (ADL). In order to support these people, AAL systems are sometimes using a technique called ADL-Recognition, which is trying to estimate which activities are performed by the monitored individuals and assists them, or calls for help. ADL-Recognition is currently an actively researched field with several different approaches. ADL-Recognition can be another good source for the proposed system, since it can gather even behavioral information without having someone observe the individual.

Logs of ADL [Activities of Daily Living; Author's note] performance provide valuable data for healthcare professionals. Creating logs automatically would be more comprehensive and less expensive than relying on manual observations.  
[...] (Bardram et al., 2007, p. 81)

Additionally, there are two important non-functional requirements for data gathering technologies: *Unobtrusiveness* and a *low focus of attention*, which allow the system to collect data automatically and intuitively.

[..] many applications of pervasive computing, including remote monitoring devices and wearable computing, are designed to be used ubiquitously while the user carries on work activities (i.e., without switching the focus of attention to interacting with the technology). (Bardram et al., 2007, p. 195)

According to the items discussed in this section the following requirements are defined:

**R45** *The system must be able to handle raw sensor data.*

**R46** *The system must support as many different measurement devices as possible.*

- R47** *The system must be able to obtain data from data sources automatically whenever possible.*  
Or semi-automatically in case of devices that require to be plugged in a computer system.
- R48** *The system must support ubiquitous systems as data sources.*  
For instance smart homes with ADL recognition systems.
- R49** *The system must be able to use contextual information as additional data input.*  
For instance calendar data, contacts, etc.
- R50** *The system must require very little focus of attention.*  
In order to not distract the user.
- R51** *The data collection methods of the system must be as unobtrusive as possible.*  
In order to not distract the user.

### 3.3.4 Data Quality

After the approach and the technologies for data collection, it is further important to be aware that not all of the collected data is useful. Corrupt or incomplete data can be even harmful when used in a pervasive health system. Puentes et al. (2013) described the quality of data extensively in a paper, and states:

The notion of data quality is essential in order to estimate if collected data from sensors and patient observations, can be properly processed to extract the required information, given that data acquisition conditions are uncontrolled and depend mainly on how the user respects a given protocol, the type of sensors and vendor, as well as the collected data volume. (Puentes et al., 2013, p. 104)

The article further states that it is "commonly accepted that data quality relies mainly on accuracy and completeness, leading to disregard data sets on which one of these quality estimations is not appropriate". There are various reasons for incomplete or inaccurate data sets, for instance wrong handling of the sensing devices. Consequently the article suggests "before deciding if the concerned data set should be deleted [...] or can be used with certain constraints" (see Puentes et al., 2013, p. 108). Meaning that even incomplete data can be useful under certain conditions, which need to be defined.

Therefore a requirement will be defined in order to ensure data integrity:

- R52** *The system must be able to recognize and discard corrupt or incomplete data sets.*



### 3.3.5 Empowerment & Decision Support

In this work the collected information will provide the information to a separate system which is showing some similarities with decision-support (DS) systems. This system is using the provided data in order to visualize it, draw conclusions and assist, guide, or educate the user.

Data from physiological sensors and personal observations are the components of a PHR for DS [Decision Support; Author's note] that stores, exchanges, and displays the data considered to be a complement to the Electronic Health Record (EHR), with the goal of endorsing individuals' active role in their own healthcare. The use of this PHR for DS implies, nevertheless, that it is critical to understand how to handle and evaluate data imperfections before information extraction takes place. (Puentes et al., 2013, p. 104)

## 3.4 The "e-Requirements"

The role of the patient is becoming more engaged, educated, enabled, equipped, and most importantly - *empowered*. In this section the requirements for patient empowerment will be defined. Patient empowerment is enabling participatory medicine in order to practice shared decision making with clinicians and healthcare professionals.

The role of the patient is starting to shift from being a minimally-informed advice recipient to an active participant, instigating collaborator, information sharer, peer leader and self-tracker engaged in participative medicine; [...]. (Swan, 2009, p. 513)

In order to fulfill some of the requirements mentioned in previous sections, such as users access to health information and increasing their ability to play a more active role in their own healthcare, the concept of Patient Empowerment is playing an essential role. It facilitates different methods, for instance health education and decision-support.

[..] patients should (1) have access to relevant information, (2) develop skills for problem-solving and self-efficacy, and (3) be motivated to manage their health. Different patient-centered mechanisms for implementing these requirements have been adopted, such as education and health literacy, decision-making aids, and self-care support. They are solutions to drive the empowerment process. (Alpay et al., 2010, p. 788)

Besides shared decision making, the patients also supply health providers with information, which would be otherwise hard to obtain and may play an increasingly important role in the decision making process, due to an increased amount of background information.

Future interactions may be those of knowledgeable patients bringing quantitative reports from their self-testing and self-tracking activities to medical professionals for consultative co-interpretation of the results. (Swan, 2009, p. 512)

In this section requirements will be shown, which enable the empowerment of the user.

#### **3.4.1 Usability & Health Literacy**

Firstly, usability is one of the most important aspects of driving the empowerment process. Besides technical skills, users can lose motivation if the learning curve is too steep.

One of the biggest barriers to patient adoption of PHRs is usability. Patients will frequently have limited technical skills, and even more limited medical literacy, which can impair their ability to make use of a PHR unless the PHR is well designed to accommodate its user's limitations. (Robison et al., 2012, p. 188)

The second aspect is the users health literacy. It is of utmost importance to ensure the user is understanding the health related information and is able to draw correct conclusions.

In order to be useful to the patient, the PHR must present data and accompanying tools in ways that enable the individual to understand and to act on the information contained in the record. This is challenging because of patients' widely varying levels of general literacy and of health literacy. (Tang et al., 2006, p. 123)

The worst case scenario is a user, who is misinterpreting critical information.

[...] most patients lack the knowledge to interpret or analyze their medical records and may come up with incorrect and even dangerous interpretations. (Varshney, 2009, p. 24)

Demographics for health literacy are showing that the most limitations for health literacy exist within the elderly population, people which may have language barriers, and people who already suffer from health conditions.

The groups with the greatest limitations for health literacy include people older than age sixty-five; minority, immigrant, and low-income populations; and people with chronic mental or physical conditions. (Kahn et al., 2009, p. 371)

Typically these groups are congruent with those for high usability requirements. The usability requirements lie outside the boundaries of this work and are therefore not discussed in detail, however they still have to be added to the requirements.

**R53** *The system must offer a high level of usability.*

Regarding the comprehensiveness of information to the user, a second requirement is defined in this subsection:

**R54** *The system must present information to the user in a comprehensive and easily understandable way.*

#### **3.4.2 Improvement of Health Literacy and Comprehension**

The first thing that comes to mind regarding getting some knowledge about nearly any topic is using online sources from the internet. Even in lots of rural areas an internet connection is available and mostly inexpensive, which is beneficial for people with a low income.

A common technology like the Internet may be used to help connect individuals to their health information and may reduce the participatory divide by bringing relatively inexpensive online technology to those with limited financial means. (Kahn et al., 2009, p. 372)

Although the the internet can be considered to be the ultimate information source of our time, it is needless to say that it depends on the user to be able to find and understand the numerous web-sources, as well as to be able to identify relevant and reliable ones. While search engines help to find public information online, the reading comprehension is still a task which can be only done by the user himself.

Most people are more than capable of learning new information themselves. Unfortunately in regard of health information this learning process is more difficult, mostly due to missing background information. It is therefore important to not use domain specific terms when explaining circumstances or relations, because these may not be understood by laypersons.

Because the public rather than medical professionals will use PHRs, we will have to develop "lay" representations and explanations of the encoded data. (Tang et al., 2006, p. 124)

The architectural issues [...] also include [...] transitive trust and the need to present information in a manner understandable to laypersons. (Tang et al., 2006, p. 125)

Besides the terminology, also data values may be misinterpreted by people not familiar with health or medical information. For this purpose the system should offer easy to understand visualizations of the data.

Either the PHR or the provider's interacting EHR system should create useful summaries from voluminous PHR data. Ideally, the summarization tools would identify exceptions and important trends, presenting information via carefully crafted visual representations, statistical summarizations, feedback from clinical decision support systems, or, most likely, a combination of approaches. (Tang et al., 2006, p. 125)

The requirement to use layman's terms has already been defined previously in **R38**. New requirements are defined as follows:

- R55** *The system must allow online sources for health knowledge.*
- R56** *The system must be able to generate easily comprehensible visualizations of the users health information.*
- R57** *The system must be able to generate easily comprehensible summaries of the users health information.*
- R58** *The system must be able to discover correlations in the health data.*

#### 3.4.3 Online Communities

Since it is always a challenge to teach well known things to people who are new to a certain domain, e.g. health topics, it is always useful to have people in the same situation for exchange and paraphrasing. Regarding the internet, these people in the same situation can be found in online communities, such as social networks, forums, or blogging networks.

Dedicated social networks exist for health topics and are called social health networks or health social networks. These social health networks enable social exchange between users about health topics and in some cases further offer different features, all related to the field of health.

A health social network is a website where consumers may be able to find health resources at a number of different levels [...]. Services may range from a basic tier of emotional support and information sharing to Q&A with physicians to quantified self-tracking to clinical trial access. (Swan, 2009, p. 495)

Individuals in similar situations can share experiences and information with each other. The beneficial effects are similar to those of support groups.

One key value health social networks provide is the potential to find others in similar health situations and share information about conditions, symptoms and treatments. (Swan, 2009, p. 495)

In health social networks the number of individuals with similar experiences and therefore some knowledge about certain conditions is typically a lot higher than the number of people in a nearby location, e.g. city, province, state, or even country-wide. Therefore they cover a broad range of information.

[...] a key benefit of health social networks is that they can offer a more comprehensive look at a patient's health by covering a deeper and broader range of conditions than is expedient for traditional medicine. (Swan, 2009, p. 496)

Services offered by Health Social Networks include:

1. Mix of emotional support and information sharing.
2. Ability to pose questions to physicians.
3. Quantified self-tracking.
4. Provide information regarding clinical trials.

(see Swan, 2009, pp. 496–499)

These services are further beneficial to the patient empowerment idea. Especially 1. and 3. are useful extensions for the proposed system. The information offering part of the system can gather basic and extensive information from multiple online sources and allow the user to submit own data.

Websites [health social networks containing information sharing features; Author's note] may auto-populate general condition information from Internet health

resources such as Wikipedia articles and PubMed links. In addition to the general information, patients may be able to enter qualitative and quantitative data about their own conditions, symptoms, treatments and overall experiences. (Swan, 2009, p. 496)

As mentioned in the previous section, the gathering of information, for instance via quantified self-tracking (automatically or manually submitted by the user), can be combined with the empowering component and create meaningful visualizations in order to improve the users understanding of relations between health information and health literacy in general. In case of interconnected systems or networking capabilities of the proposed system, it is also possible to visualize data for whole populations, making it a useful tool for researchers as well.

The self-tracking functionality [of health social networks; Author's note] consists of easy-to-use data entry screens for condition, symptom, treatment and other biological information. The information can then be seen in a graphical display, possibly with views by individual, aggregated population or custom groups. (Swan, 2009, p. 498)

An example for a health social network is PatientsLikeMe<sup>4</sup>. In perspective of emotional support and information sharing, the website uses collaborative filtering to enable users to find other patients that are alike (based on the data retrieved by self-tracking), which are the most relevant for providing and sharing information (see Swan, 2009, p. 499).

As briefly mentioned in the previous section, the continuous collection of health related information may also help to understand conditions and diseases (see Bardram et al., 2007, p. 120). This effect is significantly increased if the number of users participating in the information collection process are interconnected and therefore the data aggregated.

[..] the depth of information generated through large online patient communities creatively interacting and monitoring their conditions with quantitative tracking tools can lead to new findings that give a better understanding of the underlying conditions. (Swan, 2009, p. 500)

The lastly mentioned service concerning clinical trials is typically only applicable for patients suffering from chronic diseases and therefore not primarily part of this work. However, the possibility exists to link and suggest clinical trials according to patients' conditions and use information tracking technologies in order to improve the quality of feedback.

---

<sup>4</sup>PatientsLikeMe: <https://www.patientslikeme.com>, [accessed August 18th, 2013]  
Social Health Network founded in 2004

[..] online health tracking in conjunction with clinical trials means that patients can make their experience feedback, including response to drugs, available as a public resource. (Swan, 2009, p. 500)

An example project for linking patients to applicable clinical trials is TrialX<sup>5</sup>.

For this section two new requirements are defined:

**R59** *The system must support information import and export with social health networks.*

**R60** *The system must allow the user to find relevant social health networks and similar online communities.*

#### 3.4.4 Guidance & Decision Support

Another critical factor of patient empowerment lies in the making of informed choices and decisions. Therefore, besides giving information to the user or improving his understanding of health information, the system needs to guide the user and help to make decisions. Support is needed especially since it can be assumed that there is always a possibility for misinterpretation or missing parts regarding the users health knowledge.

[...] PHR systems can also include decision-support capabilities that can assist patients in managing chronic conditions. (Tang et al., 2006, p. 121)

Typically these decision-support capabilities are used in the management of chronic illnesses, but also can play an important role in the prevention of health issues or incidents. A popular example is the usage of drug databases in order to prevent drug interactions or allergic reactions. Especially when the user intends to take over-the-counter medication which has not been checked in with a physician beforehand. It is important to support multiple drug databases in order to gain a sufficient amount of coverage.

**R61** *The system must be able to identify positive and negative effects of the users health conditions and lifestyle based on his health information and show suggestions or warnings.*

**R62** *The system must be able to use multiple drug databases to warn the user about drug related allergic reactions, side effects, or interactions.*

---

<sup>5</sup>TrialX: <https://trialx.com>, [accessed August 18th, 2013]

Website for connecting patients to clinical trials, by Applied Informatics Inc.

### 3.4.5 Motivation & Encouragement

Even with automatic data collection measures and excellent usability features, it can be troubling and cumbersome for users to maintain the system and keep their information up-to-date at all times. Further the user might not have the motivation to improve his health literacy. It is therefore important to keep the motivation of the user at a high level.

Usually individuals have a low interest in health related topics until they are affected by a health condition themselves.

[...] individuals may have little interest in understanding health-related terminologies or test results until they develop a chronic or life-threatening illness. Typically, such individuals then become more engaged in their health and attempt to understand a wider range of knowledge and information related to their disease process. (Tang et al., 2006, p. 123)

#### **Negative Motivation**

This could be used to motivate users through the display of negative consequences of health negligence. However, these methods of creating fears about diseases is not the best approach, especially since it could trigger hypochondriac<sup>6</sup> behaviors. A similar approach is to introduce punishments or absence of rewards in case of negligent behavior of the user. All these techniques bear the risk of finishing off the motivation of the user entirely, in case it is already impaired.

#### **Positive Motivation**

A better approach is to create positive feelings about progress that has been achieved by the individual.

The first approach is to show the user theoretical effects of his preventive actions and his active role in his own health care. Due to the theoretical nature of estimating prevented health issues, the effectiveness of this approach is questionable.

The second approach is to define rewards, which motivate the user to keep using the system properly and continuously. These rewards can range from financial benefits (possibly from health insurances in exchange for a potential decrease of costs through the preventive effects) to emotional and social components. The latter are well known from computer games and are typically simply called "Achievements", a game mechanism similar to trophies, badges, and ribbons as rewards in sports and military.

---

<sup>6</sup>Hypochondria is a condition where a patient excessively worries about health issues, even without symptoms.



Achievements are *goals* in an *achievement/reward system* [...] whose fulfilment [sic!] is defined through activities and events in other systems [...]. (Hamari and Eranti, 2011, p. 4)

As it might seem unorthodox to implement game mechanisms into a serious computer system for health information, it is essential for the proposed system to enrich repeating tasks, which can be perceived as a cumbersome duty, with motivational effects, such as rewards.

It seems that game design is starting to penetrate multiple areas of human life. [...] Therefore, the study of game mechanics should translate beyond what are traditionally perceived as games. (Hamari and Eranti, 2011, p. 17)

Besides the direct effect of the achievement mechanics, it could be further possible for users to show achieved goals to others in order to motivate them as well or to get positive reactions, which can be a secondary source for motivation caused by the achievement system. It can be additionally extended by social components, for example introducing groups of users which can collaboratively work on goals. However, the benefits are mostly limited to people with a certain level of computer literacy.

For this section, three new requirements are added about motivation effects:

- R63** *The system must avoid to create fear of diseases or health conditions.*
- R64** *The system must have an achievement system that allows the definition of goals and respective rewards upon fulfillment of these goals.*  
Further features of the achievement system are encouraged but not part of this work, since this subsystem is just exemplary for motivational mechanics.
- R65** *The system must be able to issue proof or verification for regular and meaningful use of the system in order to make a rewarding system for health insurances possible.*

### 3.5 Extension Capability

The system must have a high level of extensibility. Developers and skilled users alike must be able to implement new functions and features into the system.

The greatest value of PHRs will not just come from the data it stores, but from the various services it provides, promoting individualized coordination of care needs. (Gonzales and Tran, 2012, p. 1)

Thus, the amount of knowledge and features of the systems, as well as device support must continue to grow.

#### **3.5.1 Information Collection Methods**

Since the e-Health or Pervasive Healthcare area gained plenty of interest in the last years it is needless to say that new developments will continue to appear in the next years. Among them devices for self-measurement.

The list of manufacturers that provide consumer-oriented healthcare data devices offering direct integration with multiple online data sources will continue to grow. (Trotter and Uhlman, 2013, p. 79)

Further it has been shown in section 3.3.1 (particularly in figure 3.1), that there is a lot of context to gain from health monitoring, which can deliver important information and therefore should be supported.

**R66** *The system must offer a plugin system for future information collection methods.*

#### **3.5.2 Empowerment & Decision Support Methods**

Besides the development of technologies, research in medical-, health-, and nursing sciences will show important relations between data which are yet unknown. In order to use this new knowledge in the system, it must be easily possible to implement new modules in order to be prepared for the future.

However, to prevent harmful implementations, a certification should be required in order to introduce new empowerment or decision support modules into the system.

**R67** *The system must offer a plugin system for future decision support or empowerment methods, including for motivational purposes.*

**R68** *The system must have a protection mechanic against malicious plugins.*

#### **3.5.3 Access & User Interface**

Additionally the User Interface should not be limited to a certain device or medium. In the recent years new devices like smartphones and tablet computers have been heavily adopted by consumers. Therefore the user interface should also be able to run on these devices to achieve a high level of accessibility.

As devices become more sophisticated and application providers design mobile-ready solutions, the mobile phone may also serve as an important entry point for consumers to access their PHRs. (Kahn et al., 2009, p. 371)

For this purpose a requirement is defined as well:

**R69** *The system must be accessible for different operating systems and platforms and therefore be modifiable or extensible towards these platforms and operating systems.*

For example operating systems for: Smartphones and tablets, desktop computers, embedded systems, smart homes, home entertaining systems, etc.

### 3.6 Consolidation of Requirements

In this section all previously collected requirements are listed again. Consequently these vague requirements are consolidated and derived into more objective requirements. However, the main focus lies on architectural requirements in order to use it as a groundwork for an architectural draft. The resulting requirements are not sufficient for a full specification.

### 3.6.1 List of Collected Requirements

In this table all the vague requirements that have been collected are listed again with their respective position in this document.

Table 3.1: List of collected vague requirements and their positions in this document.

| #   | Requirement  | Section | Page |
|-----|--|---------|------|
| Ro1 | The user must have access to his own health information and data.  | 3.1.4   | 13   |
| Ro2 | The user must be able to gain access to health knowledge through the system.                                       | 3.1.4   | 13   |
| Ro3 | The system must allow communication between Patients and Healthcare Professionals.                                 | 3.1.4   | 13   |
| Ro4 | Inexpensive access to the system, or the software must be given.   | 3.1.4   | 13   |
| Ro5 | The system must be able to share health information with healthcare professionals.                                 | 3.1.4   | 13   |
| Ro6 | The system must qualify for and encourage shared decision-making.  | 3.1.4   | 13   |
| Ro7 | The system must encourage a healthy lifestyle.   | 3.1.4   | 13   |
| Ro8 | The user must have complete control over his data.   | 3.2.1   | 15   |
| Ro9 | The system must allow the import of EHR data.  | 3.2.1   | 15   |
| R10 | Privacy and security of personal information must be given at any time.  | 3.2.2   | 16   |
| R11 | The user must be able to authorize access to his PHR for other users.  | 3.2.2   | 17   |
| R12 | User authentication must be as simple as possible.   | 3.2.2   | 17   |
| R13 | The user must be able to share data from his own PHR with EHR systems.   | 3.2.2   | 20   |
| R14 | Anytime when the user is sharing data, he must be able to specifically select the data he wants to share.          | 3.2.2   | 20   |
| R15 | The user must be able to download/export data from his own PHR to human readable files in various formats.         | 3.2.2   | 20   |
| R16 | The user must be able to download/export data from his own PHR to machine readable files and standardized formats. | 3.2.2   | 20   |
| R17 | The user must be able to print selected health information from his PHR.   | 3.2.2   | 20   |
| R18 | The user must be able to import data into his own PHR from EHR systems.  | 3.2.2   | 20   |
| R19 | Health information exchange must always require the explicit approval of the user.                                 | 3.2.2   | 20   |

| #   | Requirement   | Section | Page |
|-----|---|---------|------|
| R20 | When a user attempts to share data, the data must be automatically checked for integrity, amount, and usefulness.   | 3.2.2   | 20   |
| R21 | The system must allow access to designated caregivers.  | 3.2.2   | 21   |
| R22 | The system must allow parental access in case of children.  | 3.2.2   | 21   |
| R23 | The frontend application must not be bound to a certain technology.   | 3.2.2   | 22   |
| R24 | The system must allow PHR data from Health Information Systems.   | 3.2.3   | 23   |
| R25 | The system must allow PHR data from Individuals.  | 3.2.3   | 23   |
| R26 | The system must allow PHR data from Devices.  | 3.2.3   | 23   |
| R27 | The system must automatically annotate received data according to it's source.  | 3.2.3   | 23   |
| R28 | The system must allow user-submitted information and must not limit the information items, e.g. what data can be submitted.   | 3.2.3   | 24   |
| R29 | The user must be able to allow guest access to his PHR.   | 3.2.3   | 25   |
| R30 | The system must be able to persist and track fitness and lifestyle information.   | 3.2.3   | 27   |
| R31 | The system must be able to persist and track data about environmental conditions in their respective context.   | 3.2.3   | 27   |
| R32 | The system must allow the user to include social relations.   | 3.2.3   | 27   |
| R33 | The system must be able to anonymously retrieve health information about social contacts of the user, if available, and persist this information as long as it is relevant. | 3.2.3   | 27   |

| #   | Requirement  | Section | Page |
|-----|--|---------|------|
| R34 | The PHR must be able to contain the following information items:<br>(a) Problems/Diagnoses/Conditions,<br>(b) Procedures/Surgeries,<br>(c) Major illnesses,<br>(d) Health providers,<br>(e) Home-monitored and self-measured data,<br>(f) Family history,<br>(g) Social history and lifestyle,<br>(h) Immunizations,<br>(i) Laboratory tests and results,<br>(j) Medications/Prescriptions,<br>(k) Allergies/Adverse reactions,<br>(l) Insurance/Payer info,<br>(m) Clinical encounter,<br>(n) Care plan | 3.2.3   | 28   |
| R35 | The system should automatically check for information inconsistencies.   | 3.2.3   | 28   |
| R36 | The system must keep time-limits regarding the validity of it's data.  | 3.2.3   | 29   |
| R37 | The system must keep a changelog about all changes of PHR data.  | 3.2.3   | 29   |
| R38 | The system must present information to the user in layman's terms.   | 3.2.4   | 31   |
| R39 | The user must receive feedback from the system based on information saved in the PHR.  | 3.2.4   | 31   |
| R40 | The user must be able to customize the appearance of the user-interface.   | 3.2.4   | 31   |
| R41 | The system must allow the user to share his information online.  | 3.2.6   | 33   |
| R42 | The system must allow the public sharing of anonymized information.  | 3.2.6   | 33   |
| R43 | The system must support multiple health monitoring sources.  | 3.3.1   | 35   |
| R44 | The system must be able to utilize quantified self-tracking and lifelogging systems.   | 3.3.2   | 37   |
| R45 | The system must be able to handle raw sensor data.   | 3.3.3   | 38   |
| R46 | The system must support as many different measurement devices as possible.   | 3.3.3   | 38   |
| R47 | The system must be able to obtain data from data sources automatically whenever possible.  | 3.3.3   | 39   |
| R48 | The system must support ubiquitous systems as data sources.  | 3.3.3   | 39   |

| #   | Requirement  | Section | Page |
|-----|--|---------|------|
| R49 | The system must be able to use contextual information as additional data input.  | 3.3.3   | 39   |
| R50 | The system must require very little focus of attention.  | 3.3.3   | 39   |
| R51 | The data collection methods of the system must be as unobtrusive as possible.  | 3.3.3   | 39   |
| R52 | The system must be able to recognize and discard corrupt or incomplete data sets.  | 3.3.4   | 39   |
| R53 | The system must offer a high level of usability.   | 3.4.1   | 42   |
| R54 | The system must present information to the user in a comprehensive and easily understandable way.  | 3.4.1   | 42   |
| R55 | The system must allow online sources for health knowledge.   | 3.4.2   | 43   |
| R56 | The system must be able to generate easily comprehensible visualizations of the users health information.  | 3.4.2   | 43   |
| R57 | The system must be able to generate easily comprehensible summaries of the users health information.   | 3.4.2   | 43   |
| R58 | The system must be able to discover correlations in the health data.   | 3.4.2   | 43   |
| R59 | The system must support information import and export with social health networks.   | 3.4.3   | 46   |
| R60 | The system must allow the user to find relevant social health networks and similar online communities.   | 3.4.3   | 46   |
| R61 | The system must be able to identify positive and negative effects of the users health conditions and lifestyle based on his health information and show suggestions or warnings. | 3.4.4   | 46   |
| R62 | The system must be able to use multiple drug databases to warn the user about drug related allergic reactions, side effects, or interactions.                                    | 3.4.4   | 46   |
| R63 | The system must avoid to create fear of diseases or health conditions.   | 3.4.5   | 48   |
| R64 | The system must have an achievement system that allows the definition of goals and respective rewards upon fulfillment of these goals.   | 3.4.5   | 48   |
| R65 | The system must be able to issue proof or verification for regular and meaningful use of the system in order to make a rewarding system for health insurances possible.          | 3.4.5   | 48   |
| R66 | The system must offer a plugin system for future information collection methods.   | 3.5.1   | 49   |

---

| #   | Requirement  | Section | Page |
|-----|--|---------|------|
| R67 | The system must offer a plugin system for future decision support or empowerment methods, including for motivational purposes.                                       | 3.5.2   | 49   |
| R68 | The system must have a protection mechanic against malicious plugins.  | 3.5.2   | 49   |
| R69 | The system must be accessible for different operating systems and platforms and therefore be modifiable or extensible towards these platforms and operating systems. | 3.5.3   | 50   |



---

### 3.6.2 Requirement Groups

The collected requirements have been categorized into requirement groups depending on their area of impact.

**RG1:** The first group of requirements contains requirements that affect basic system properties, such as location of the system, access to the system and extensibility features.

= { R01, R04, R08, R23, R66, R67, R68, R69 }

**RG2:** The second group of requirements contains requirements that affect only security or privacy of personal information.

= { R10, R12 }

**RG3:** The third group of requirements contains requirements that affect the collection or persistence of information.

= { R27, R28, R30, R31, R32, R34, R35, R36, R37, R43, R44, R45, R46, R47, R48, R49, R52 }

**RG4:** The fourth group of requirements contains requirements that affect the interconnectivity or data exchange with other parties or technologies, including security measures for that matter.

= { R03, R05, R09, R11, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R24, R25, R26, R29, R41, R42 }

**RG5:** The fifth group of requirements contains requirements that affect the empowerment of patients, the decision support capabilities, or the usability of the system.

= { R02, R06, R07, R38, R39, R40, R50, R51, R53, R54, R56, R57, R58, R61, R63, R64 }

**RG6:** The sixth group of requirements contains requirements that affect the empowerment of patient or decision support capabilities, but (contrary to RG5), require a significant amount of interconnectivity and networking (similar to RG4).

= { R55, R59, R60, R62, R65 }

**RG7:** The seventh group of requirements contains requirements that cannot be considered in this work. In this case one requirement due to its complexity and implied ethical, legal, and privacy requirements.

= { R33 }

### 3.6.3 Architectural Influences

There are several influences on architectural requirements that derive from the previously collected and grouped requirements and have an impact on the basic system architecture. In order to be able to make decisions about the architecture and identify its requirements, first the architectural decisions need to be defined with their respective requirement groups that are causing the influences.

| AD# | Architectural decision  | Derived from RG # |   |   |   |   |   |
|-----|---|-------------------|---|---|---|---|---|
|     |   | 1                 | 2 | 3 | 4 | 5 | 6 |
| AD1 | Type of Application<br>(e.g. web-based, distributed, stand-alone)   | X                 | X |   |   |   |   |
| AD2 | Access to the Application<br>(e.g. web frontend, mobile frontend, desktop frontend)   | X                 | X |   |   |   |   |
| AD3 | Security enforcement method<br>(e.g. encryption, data location)   | X                 | X | X | X |   |   |
| AD4 | Information persistence<br>(e.g. how to persist a PHR or the health knowledge base)   | X                 | X | X |   | X | X |
| AD5 | Connectivity methods to other devices for data collection<br>(e.g. data from wireless measurement devices)                  |                   | X | X | X |   |   |
| AD6 | Connectivity methods to other systems for data exchange<br>(e.g. EHR/EMR systems)   |                   | X |   | X |   |   |
| AD7 | Connectivity methods to online sources for knowledge retrieval<br>(e.g. online encyclopedias, journals, online communities) |                   | X |   |   | X | X |
| AD8 | Extensibility enforcement method<br>(e.g. how to ensure modularity of the system)   | X                 | X |   |   |   |   |

Table 3.3: Architectural decisions and their influencing requirement groups.

### 3.6.4 Architectural Requirements

In this section architectural requirements (AR) are shown. These requirements need to be met at an architectural level in order to make the system able to meet the other, previously defined requirements. The here shown architectural requirements have been identified according to the architectural decisions in the previous section and their respective influences.

---

**AD1: Type of Application**

- AR01** The system shall be a distributed system consisting mainly of interconnected heterogeneous clients.
- AR02** The clients shall work on the following platforms and therefore require multiple implementations: Android, iOS, Linux, Windows, and Windows Phone.
- AR03** The system shall have an online repository that stores all the users encrypted data and serves as a backup for the clients.
- AR04** The repository shall store and transfer only encrypted data.
- AR05** The repository shall be managed on an online storage with logic for security and persistence only, by an authority yet to be determined.

**AD2: Access to the Application**

- AR06** Every client shall be able to manage the clients (add, delete, or modify the trusted client pool). Any modification of the trusted client pool shall require the users master key.
- AR07** Every client within the trusted client pool shall be able to transfer encrypted data to and from the repository.
- AR08** Every client shall run exactly one client application.
- AR09** Every client application shall contain a user interface.
- AR10** Every client application shall store user data depending on the client device capacity. Devices with less memory shall store only recent and often required data, while devices with higher capacity shall store more information. Not stored information shall be received from the repository on demand.
- AR11** Every client application shall offer access to the Personal Health Record data.
- AR12** Every client application shall contain only logic which functions with the performance offered by the client device.

---

**AR13** Besides the restriction due to the performance requirements, the user shall be able to add or delete modules to or from every client separately as he wishes.

**AD3: Security Enforcement Method**

**AR14** All data shall be stored encrypted with an encryption key. A decryption key shall exist for decryption of the data.

**AR15** Only trusted clients shall be able to submit data to, or retrieve data from the repository.

**AD4: Information Persistence**

**AR16** The PHR data shall be stored in a dedicated database.

**AR17** The knowledge base shall be stored in a dedicated database.

**AR18** The persistence logic shall separate the DBMS and access methods in order to make the system independent from the DBMS in use.

**AR19** The persistence logic of the repository shall check the data to be persisted in the PHR for inconsistencies or conflicts with data already contained.

**AD5: Connectivity Methods to other Devices for Data Collection**

**AR20** Connectivity methods shall use the capabilities of the client devices.

**AR21** The system shall natively support collection of data from files in a file system.

**AR22** The system shall natively support collection of data via the Bluetooth protocol.

**AR23** The system shall natively support collection of data via WiFi (IEEE 802.11) protocol.

**AR24** The system shall natively support the collection of raw sensor data.

**AR25** The system shall natively support the collection of measurement values.

---

**AR26** The system shall natively support the collection of information from manual user entries.

**AR27** Collected data shall always be automatically annotated with the source and timestamp.

**AR28** Collected data shall be automatically annotated with location information, if possible.

#### **AD6: Connectivity Methods to other Systems for Data Exchange**

**AR29** The system shall natively support PHR data import from the file system.

**AR30** The system shall natively support PHR data import upon request of HIS (e.g. EHR, PHR) via dedicated Interfaces.

**AR31** The system shall natively support PHR data export to the file system.

**AR32** The system shall natively support PHR data export to Health Information Systems (e.g. EHR, PHR) Interfaces.

**AR33** The system shall support standardized file types for import and export (e.g. HL7 CCR/CCD)

#### **AD7: Connectivity Methods to online Sources for Knowledge Retrieval**

**AR34** The system shall natively support web protocols (e.g. HTTP/HTTPS, FTP) for knowledge retrieval.

**AR35** The system shall include a mechanism to parse retrieved web elements for information.

#### **AD8: Extensibility Enforcement Method**

**AR36** The application logic shall be modular, where one module contains logic for a certain function.

**AR37** The modules shall be independent from the clients platform.

---

**AR38** The modules shall have to be approved by an authority yet to be determined.

**AR39** The system shall allow modules for the client applications user interface in order to change the appearance and behavior of the application.

**AR40** The system shall allow modules for the empowerment / decision support component in order to be fit for future research findings, visualizations, communities, etc.

**AR41** The system shall allow modules for the data collection component in order to be fit for future measurement and data gathering devices, such as ADL recognition systems, etc.

## 4 System Conception

In this chapter the system concept will be shown as an architectural draft. Firstly the systems context and neighbor systems are shown with their possible interactions. Secondly the deployment of the system is depicted in a deployment view diagram. Next steps include the structural diagrams of the systems components. The class diagrams are considered drafts and the contained elements are described depending on their importance to the architecture. Consequently some use cases are shown as sequence diagrams. The notation used in the diagrams is UML (Unified Modeling Language) and the notation of exemplary interface methods is based on the Java programming language and it's data types.

Notably the system has been split into three main subsystems. The *Client System*, which will be deployed on client devices of the user and are able to function as stand-alone systems. Besides the Client System, there is a *Repository System*, which serves as a complete backup of the users data. Lastly there is the *Access Control System* or ACS, which is a relay storage for data exchange with third parties in order to preserve security of the Repository System.

The separation into components has been done in order to keep the system as modifiable as possible and make these components easily changeable in future iterations, if necessary.

### 4.1 System Context

In this section the context of the system will be shown by identifying the scope of the system as well as drafting a deployed setup and it's connection types.

#### Scope of the System

Firstly it is important so be aware where the system will be positioned among other systems and actors, and what are it's neighbor systems. In figure 4.1 the system is shown as a light gray box while the neighbor systems are shown as white boxes. Individuals and groups of individuals are shown as stick figures. The bold lines indicate main interactions with the proposed system, which are namely from upper left to lower right:

1. Medical data exchange with healthcare institutions.

2. Data import from external PHR systems/providers.
3. Web Platform and community search as well as information sharing.
4. Guest access for healthcare professionals.
5. Primary usage of the system by the user.
6. Data submission from healthcare and measurement devices.
7. Information and knowledge retrieval from web sources.

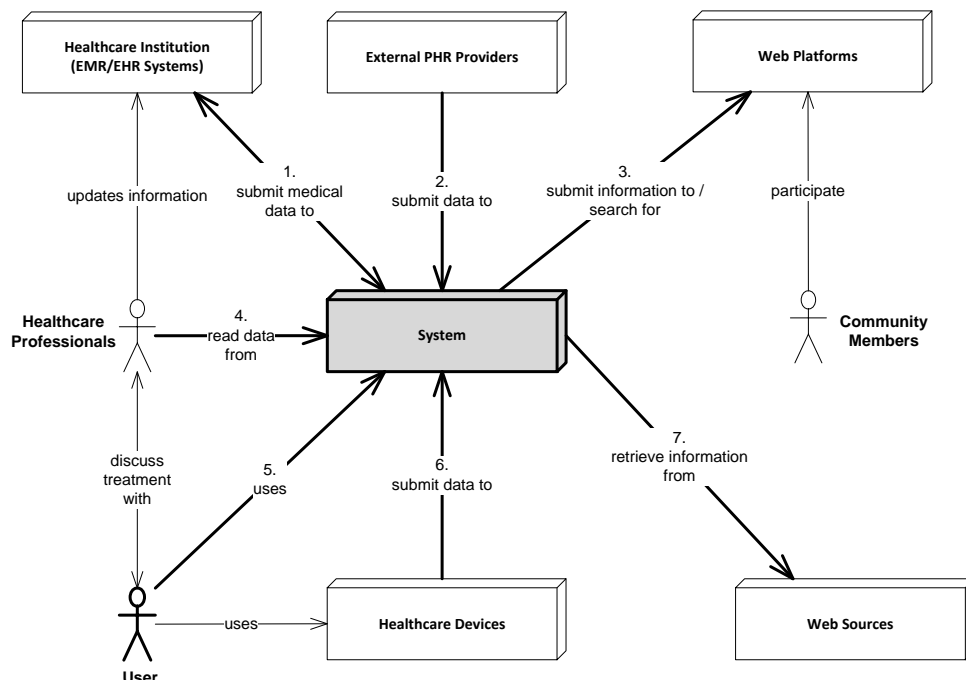


Figure 4.1: The scope of the system and possible interactions with neighbor systems.

### Deployment View

Secondly the diagram in figure 4.2 shows a sample deployment of the system. The *Client Systems* (CS) are hosted by different operating systems and devices and can vary in number. The clients application might therefore differ in their supported function sets, depending on the devices capabilities. However, the PHR-keeping functions are present on all clients, as well as the ability to connect to the *Repository System* (RS) to submit or receive PHR data, and



add access permissions for third parties in the *Access Control System (ACS)*. The connections between clients and repository systems, as well as clients and access control system are handled with SOAP (Simple Object Access Protocol) Web Services that communicate via HTTPS (Secure Hypertext Transfer Protocol). Web Services have been chosen, since they support the idea of loose coupling in order to keep the systems extensible.

Service oriented architectures, and in particular the Web Services (WS) architectures, are attractive for PHR ecosystems due [sic!] their inherent support for loose coupling of services across organizational boundaries. (Lähteenmäki et al., p. 1726)

The communication between the Repository System and the Access Control System is done via a not yet defined connection and protocol, since SOAP via HTTPS has to be already implemented, it would be an easy decision to use it here as well. However, this connection has to transfer a lot of data and the two systems (RS and ACS) may be located on the same machine and therefore an HTTPS is likely to be unnecessary.

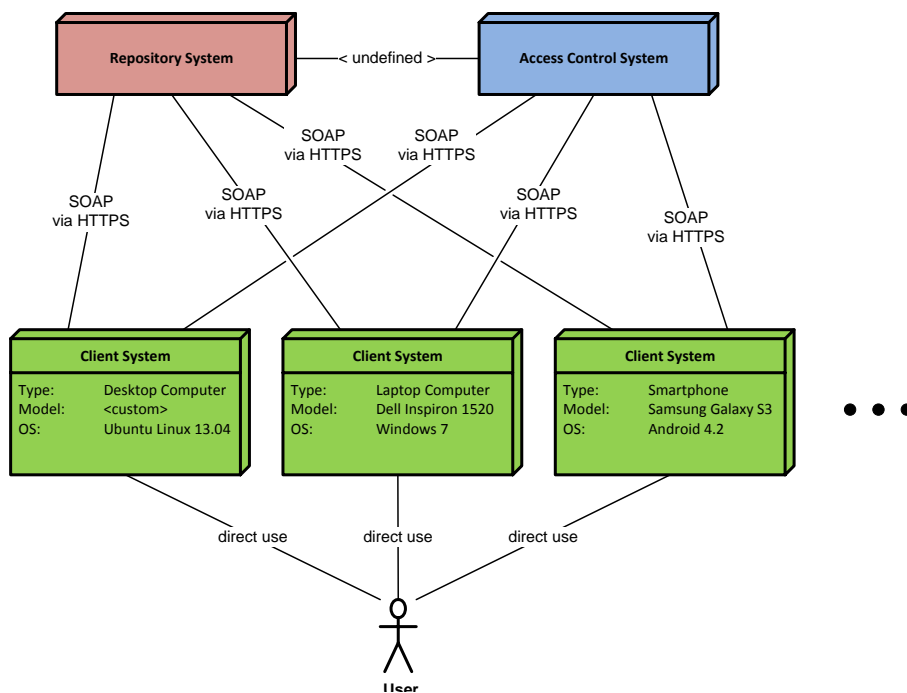


Figure 4.2: The deployment view of the system.

## 4.2 Client System

In this section the proposed architecture for the client system is shown. The logic view in figure 4.3 shows the main components and their interfaces and connectors to external systems. In this first level view the interfaces are only briefly described by their functions. The interfaces will be described more detailed in the components' subsections. Gray boxes indicate interconnections with other systems or components.

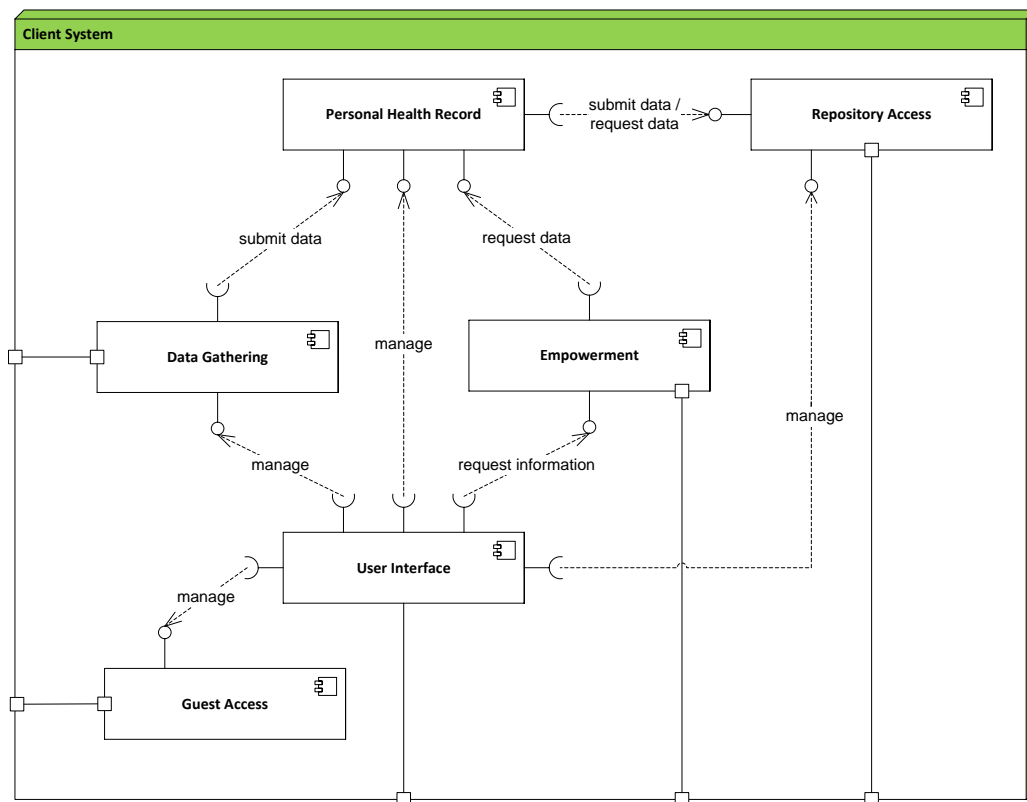


Figure 4.3: The main components of the *Client System*.

The components have been named after their respective tasks. The *User Interface* component is the window to the user. It controls the appearance of the application, displays information received from the *Empowerment* component and offers an intuitive way for the user to manage the other components.

The *Personal Health Record* (PHR) component stores data and handles requests for such data by other components. If necessary it requests data itself from the *Repository Access* component.

Notably the PHR component is responsible for en- and decryption of the users information and basic integrity checks for inconsistencies or conflicts.

The *Repository Access* component handles the data transfer between the repository system and the Client System.

The *Guest Access* component receives requests from the user to add a guest access to PHR data in the Access Control System.

The *Data Gathering* component manages data received from other devices or the user himself, depending on the activated features inside the component.

The *Empowerment* component uses data it receives from the *Personal Health Record* component and applies decision support or visualization methods, depending on the activated features inside the component.

Every component, except the PHR component, has a connector to systems outside of the client. In case of the *Repository Access* component it allows the connection to the web services provided by the Repository System. For the *Guest Access* component it allows communication with the ACS. The *Data Gathering* component uses it's connector to receive data from measurement devices, while the *Empowerment* component uses it to receive knowledge information from external sources, or search for communities. The connector of the *User Interface* component is rather a symbolic connector for the interaction of the client system with the user.

### 4.2.1 Personal Health Record

The class diagram of the *Personal Health Record* component is shown in figure 4.4.

**LocalDB** is the class which manages database access within the *Client System*. Firstly, it is the gateway to the DBMS (Database Management System) and may be replaced at a later time in order to allow the deployment of different DBMS. Secondly, upon request, it is requesting itself data from the *Repository System*, if the requested data is stored there, but not available in the local database. Since the data in the *Repository System* is only stored encrypted, the data is routed through the *CipherUtility* class and finally to the *DataSync* class, which forwards it to the *Repository Access* component.

**IntegrityHelper** is a helper class which contains basic methods to check the local database for integrity issues, incomplete data, or data conflicts. It's methods are typically called by the *LocalDB* class.

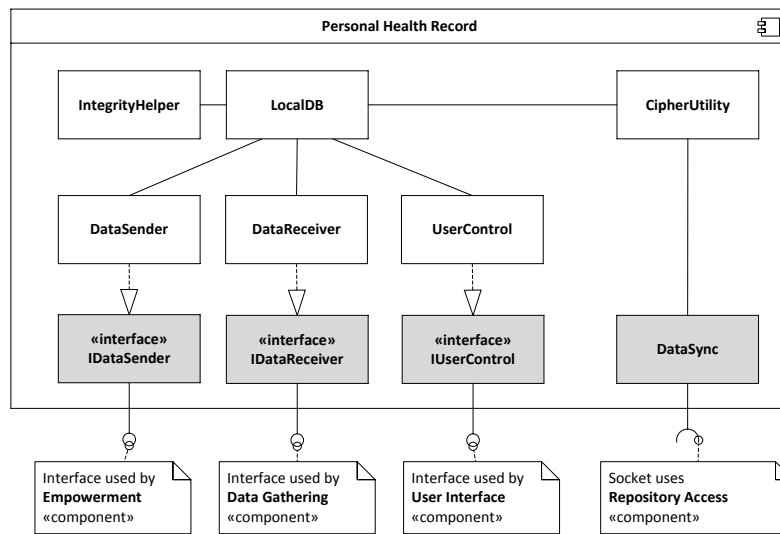


Figure 4.4: The class diagram draft of the *Personal Health Record* component.

**Cipher Utility** is a utility class which offers decryption and encryption methods.

**IDataSender** is an interface which is used by the *Empowerment* component in order to request data which is consequently provided by the *LocalDB* class and sent to the *Empowerment* component.

**DataSender** is an implementation of the *IDataSender* interface.

**IDataReceiver** is an interface which is used by the *Data Gathering* component, which uses it to submit gathered data to the PHR. As a result it is then persisted by the *LocalDB* class into the database.

**DataReceiver** is an implementation of the *IDataReceiver* interface.

**IUserControl** is an interface which is used by the *User Interface* component. It provides methods to see and alter contents of the local database and therefore is associated with the *LocalDB* class.

**UserControl** is an implementation of the *IUserControl* interface.

**DataSync** is a class which uses the `IDataControl` interface in the *Repository Access* component in order to store data in, or retrieve data from the *Repository System* via the *Repository Access* component.

#### 4.2.2 Repository Access

The class diagram of the *Repository Access* component is shown in figure 4.5.

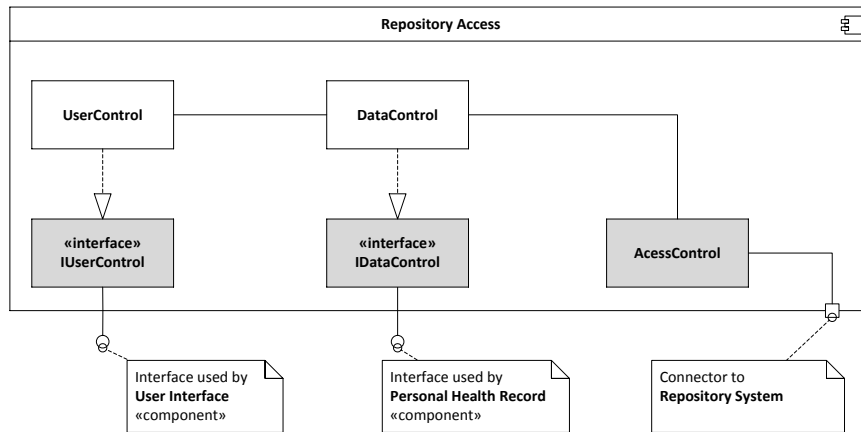


Figure 4.5: The class diagram draft of the *Repository Access* component.

**IUserControl** is an interface which is used by the *User Interface* component in order to change the used *Repository System*.

**UserControl** is an implementation of the *IUserControl* interface.

**IDataControl** is an interface which is used by the *Personal Health Record* component, which uses it for data exchange with the *Repository System* via the *AccessControl* class.

**DataControl** is a an implementation of the *IDataControl* interface.

**AccessControl** is a class which establishes the connection to the *Repository System* via SOAP/HTTPS in order to check the clients permission in the chosen repository and consequently initiate the data exchange.

### 4.2.3 Guest Access

The class diagram of the *Guest Access* component is shown in figure 4.6.

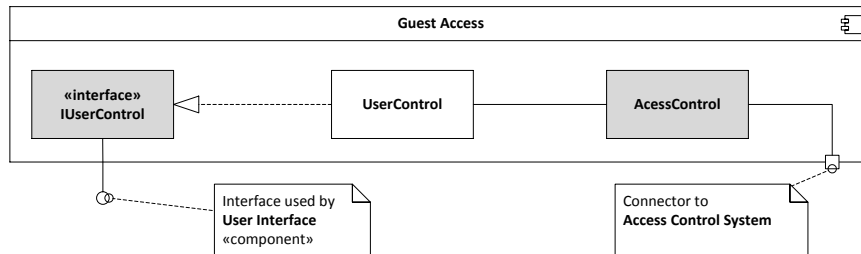


Figure 4.6: The class diagram draft of the *Guest Access* component.

**IUserControl** is an interface used by the *User Interface* component in order to allow the user to manage guest access in the *Access Control System* via the *AccessControl* class.

**UserControl** is an implementation of the *IUserControl* interface.

**AccessControl** is a class which establishes a connection to the *Access Control System* via SOAP/HTTPS. It then sends the requests of the *UserControl* class to the *Access Control System* in order to receive or alter current guest access permissions.

### 4.2.4 Data Gathering

The class diagram of the *Data Gathering* component is shown in figure 4.7.

**DataSet** is a class which represents a set of data, which has been collected and can be repacked by the *DataTransmitter* class in order to be sent to the *Personal Health Record* component. It includes contextual information, such as time and, if available, location, as well as the data items type, the data value, and the origin of the data.

**IDGPlugin** is an internal interface which is used to introduce new technologies and methods for data gathering. Every class implementing this interface will be loaded by the *PluginLoader* class and therefore, through the interfaces methods, has access to the *DataCollector* class and can be accessed by the *UserControl* class. The implementations may access external devices

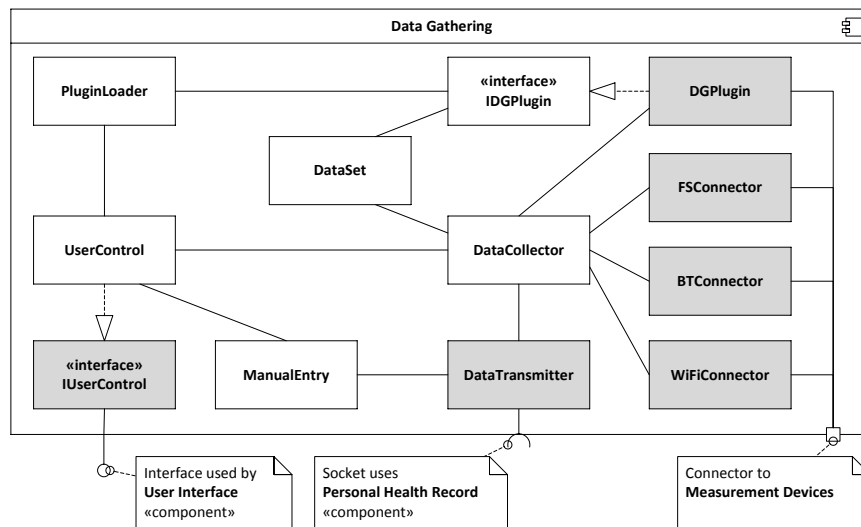


Figure 4.7: The class diagram draft of the *Data Gathering* component.

depending on the capabilities of the device which is running the client application. The following methods are executed by the *PluginLoader*:

```

1 // Executed on application start-up or first load of the plugin.
2 void init();
3
4 // Executed periodically. Used to collect and return data.
5 DataSet run();
6
7 // Executed when the plugin is activated.
8 void onActivate();
9
10 // Executed when the plugin is deactivated.
11 void onDeactivate();

```

**DGPlugin** is an exemplary implementation of the *IDGPlugin* interface.

**PluginLoader** is a loader class which is keeping a list of all installed plugins implementing the *IDGPlugin* interface and located in a certain area of the file system. It accesses the interfaces methods listed in the interfaces description.

**FSCollector** is a utility class which allows access to the file system.

**BTConnector** is a utility class which allows access to Bluetooth protocols and devices.

**WiFiConnector** is a utility class which allows access to WiFi (IEEE 802.11) protocols and devices.

**DataCollector** is a class which collects all gathered data from classes with collecting capabilities and submits them to the *DataTransmitter* class. Classes with collecting capabilities are *FSCollector*, *BTConnector*, *WiFiConnector*, as well as plugins whose data sets are collected and aggregated by the *PluginLoader* class. Additionally the *DataCollector* class adds missing contextual information to the *DataSet* objects.

**ManualEntry** is a class which represents a manual entry of the user. Manual entries do not require to be triggered for data collection and enable the user to add any data, objective or subjective, outside the boundaries of a *DataSet* and at any time. The user is further able to select the data source himself.

**DataTransmitter** is a class which repacks *ManualEntry* and *DataSet* objects and sends them to the *Personal Health Record* component via its *IDataReceiver* interface.

**IUserControl** is an interface which is used by the *User Interface* component. Firstly it is used to manage the data collection process. It provides methods to activate and deactivate the plugins and connector utility classes via the *PluginLoader*. Secondly it is used to add create and consequently add *ManualEntry* objects to the PHR.

**UserControl** is an implementation of the *IUserControl* interface.

### 4.2.5 Empowerment

The class diagram of the *Empowerment* component is shown in figure 4.8.

**OntologyHelper** is a class which attempts to retrieve semantic information out of retrieved data from *DSPlugin* classes and the *WebSourceParser* in order to support synonyms, layman's terms, and a possible reasoning engine. If the semantic information cannot be figured, the class simply forwards the original information to the *KnowledgeBase* for storage.



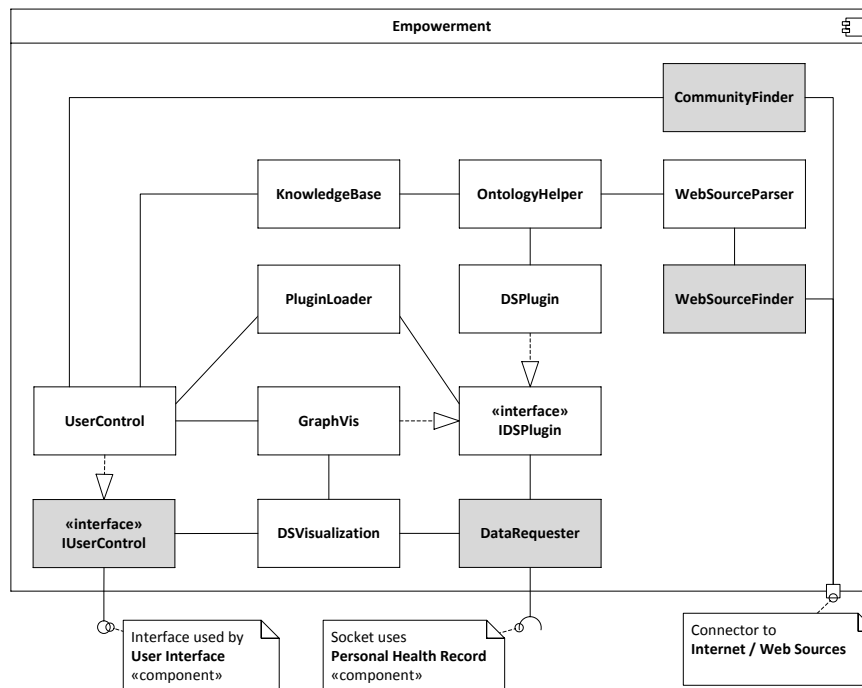


Figure 4.8: The class diagram draft of the *Empowerment* component.

**KnowledgeBase** is a class which stores knowledge information into a database. It therefore offers access to a DBMS and may be replaced at a later time in order to allow the deployment of different DBMS. It supports the storage of semantic information and plain knowledge as text as well.

**WebSourceParser** is a class which enables the parsing of web sources, mainly HTML and XML documents, in order to retrieve contained information and pass it on to the *OntologyHelper* class. It receives web sources from the *WebSourceFinder* class.

**WebSourceFinder** is a class which offers the retrieval of web sources. It supports the most common protocols, such as HTTP, HTTPS and FTP.

**CommunityFinder** is a class which uses HTTP and HTTPS in order to search for online communities. This class will be preset with certain search engine keywords, which can be changed by the *UserControl* class in order to specify or alter the search. The *UserControl* class also initiates and handles searches.

**IUserControl** is an interface which is used by the *User Interface* component, which enables the user to control the functions of the *Empowerment* component. It mainly allows the UI component to receive knowledge and data from the DS (Decision Support) Plugins via the *PluginLoader* in order to display it in the application. Further it allows the search for online sources and online communities, and manage saved knowledge information.

**UserControl** is an implementation of the *IUserControl* interface.

**DSVisualtization** is a class which is used to represent the results of decision support methods. It contains information for the *User Interface* component. This information can be pictures like info-graphics and diagrams, dynamic graph visualizations, and textual descriptions.

**IDSPlugin** is an internal interface which allows the implementation of new DS (Decision Support) methods. Every class implementing this interface will be loaded by the *PluginLoader* class. Implementing classes have access to the *DataRequester* and *OntologyHelper* classes, which ultimately allows them to retrieve information from the PHR as well as the knowledge base in order to use this information in their methods. The result of an execution should deliver a *DSVisualization* object. The following methods are executed by the *PluginLoader*:

```
1 // Executed on application start-up or first load of the plugin.
2 void init();
3
4 // Executed periodically. Used to get data and generate
5 // supportive messages and visualizations.
6 DSVisualization run();
7
8 // Executed when the plugin is activated.
9 void onActivate();
10
11 // Executed when the plugin is deactivated.
12 void onDeactivate();
```

**DSPlugin** is an implementation of the *IDSPlugin* interface.

**GraphVis** is a native implementation of the *IDSPlugin* interface. It uses data from the PHR in order to create a graphical diagram representation to be shown in the user interface.

**PluginLoader** is a loader class which is keeping a list of all installed plugins implementing the *IDSPugin* interface and located in a certain area of the file system. It accesses the interfaces methods listed in the interfaces description.

**DataRequester** is a class which uses the *IDataSender* interface of the *Personal Health Record* component in order to request stored information to be used by the decision support methods implemented in a *DSPlugin* class.

#### 4.2.6 User Interface

The class diagram of the *User Interface* component is shown in figure 4.9.

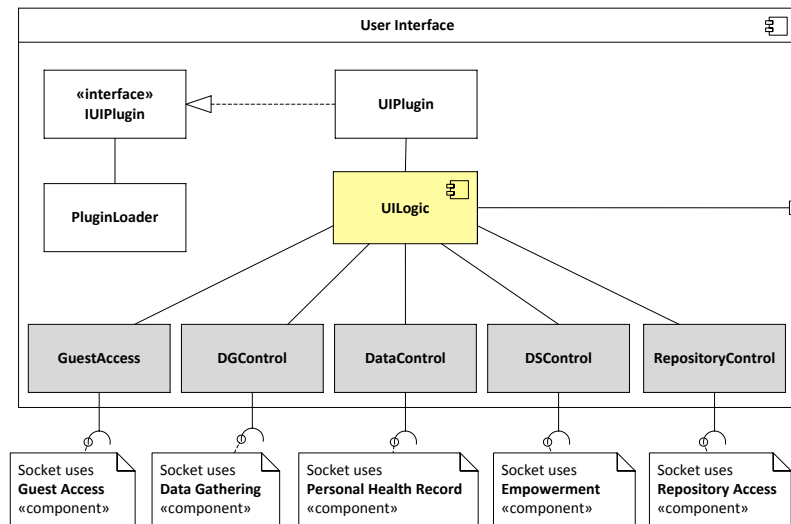


Figure 4.9: The class diagram draft of the *User Interface* component.

**GuestAccess** is a class which uses the *IUserControl* interface of the *Guest Access* component in order to add or alter guest access to PHR data in the *Access Control System*.

**DGControl** is a class which uses the *IUserControl* interface of the *DataGathering* component. It is used to manage data gathering functions, plugins and add manual entries.

**DataControl** is a class which uses the *IUserControl* interface of the *Personal Health Record* component in order to manage stored information in the PHR.

**DSControl** is a class which uses the *IUserControl* interface of the *Empowerment* component. It is used to manage decision support plugins, manage the knowledge base, and find online communities and sources.

**RepositoryControl** is a class which uses the *IUserControl* interface of the *Repository Access* component in order to change the used repository.

**IUIPlugin** is an interface which offers the possibility to change the appearance and behavior of the user-interface. All plugins implementing certain methods of the *IUIPlugin* interface are called by the *PluginLoader*.

**UIPlugin** is an implementation of the *IUIPlugin* interface.

**PluginLoader** is a class which keeps a list of all classes implementing the *IUIPlugin* interface and are stored in a certain area on the file system. It then calls the methods of the interface on every known and active implementation.

### UILogic

The *UILogic* sub-component contains the presentation and behavioral logic of the user interface. Since usability is a main requirement of the proposed system and is highly depending on the user-interfaces look and feel, the choice of the UI is left open for further research. For the time being a simple MVC (Model-View-Controller) pattern (see Burbeck, 1987) is used. This pattern is likely to be replaced by more sophisticated UI patterns in the future.

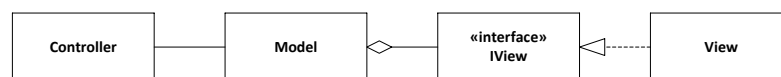


Figure 4.10: The MVC pattern as a temporary solution for the *UILogic* sub-component, based on the MVC pattern introduced for Smalltalk by Burbeck (1987)

**Model-View-Controller** is a design pattern for basic separation of concerns in regard of user interfaces in object-oriented programming (see figure 4.10). The *Controller* class dictates the behavior of the user interface. It accesses methods of the *Model* class, which contains information used to render the user interface. The *Model* class aggregates multiple *View* objects,

which all implement the *IView* interface for different visualizations. The *View* objects use the information of the *Model* which is used for the visual representation.

### 4.3 Repository System

In this section the proposed architecture for the Repository System is shown. In figure 4.11 the logical view is shown with its components, interfaces and connectors. The detailed description of the interfaces will be done in the components' respective subsections. However, a brief description of the components and their interactions are given in this part. Gray boxes indicate interconnections with other systems or components.

The *WebService Management* component manages the web services provided to the client systems. According to the methods requested from external systems it chooses to forward them to their respective destination component.

The *Persistence Management* component manages the storage of data in a database, dedicated to PHR information. It exchanges data with the *WebService Management* component in order to pass it to the client systems. Since the data stored on the Repository System is always encrypted, a de- or encryption part is not necessary.

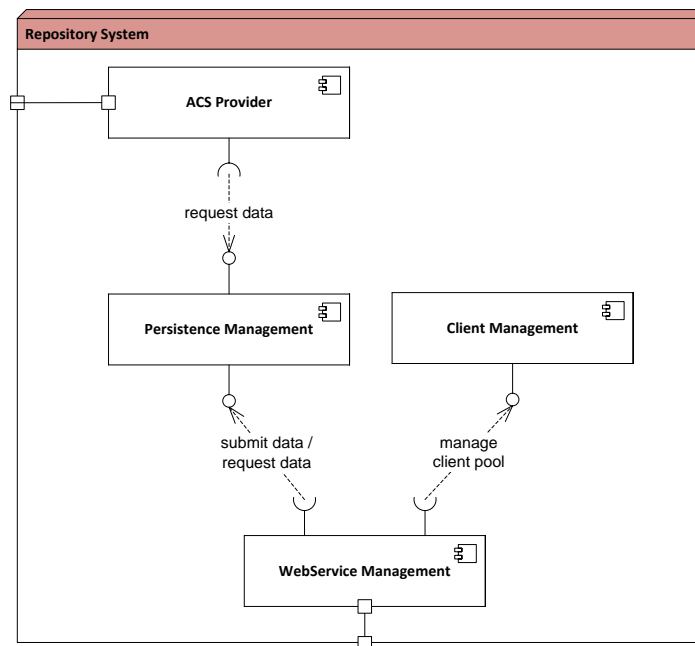


Figure 4.11: The main components of the *Repository System*.

The *Client Management* component keeps a list of trusted clients in order to ensure that only client systems authorized by the user are allowed to submit and receive data from the repository.

The *ACS Provider* component handles data requests from the Access Control System.

### 4.3.1 WebService Management

The class diagram of the *WebService Management* component is shown in figure 4.12.

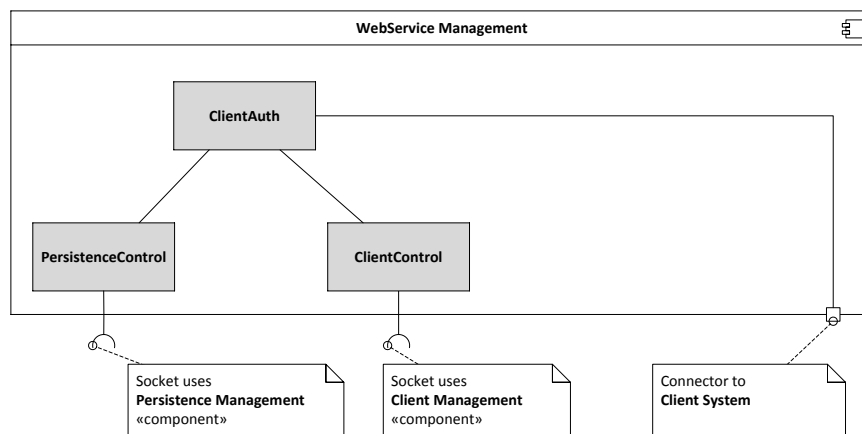


Figure 4.12: The class diagram draft of the *WebService Management* component (RS).

**PersistenceControl** is a class which uses the *IClientControl* interface of the *Persistence Management* component in order to access data stored in the database of the *Repository System*.

**ClientControl** is a class which uses the *IClientControl* of the *Client Management* component which is used to delegate management commands in order to alter the trusted client pool.

**ClientAuth** is a class which establishes the connection to the *Client System* via SOAP/HTTPS in order to authenticate the client and is subsequently used for data exchange between the *Repository System* and the *Client System*.

### 4.3.2 Persistence Management

The class diagram of the *Persistence Management* component is shown in figure 4.13.

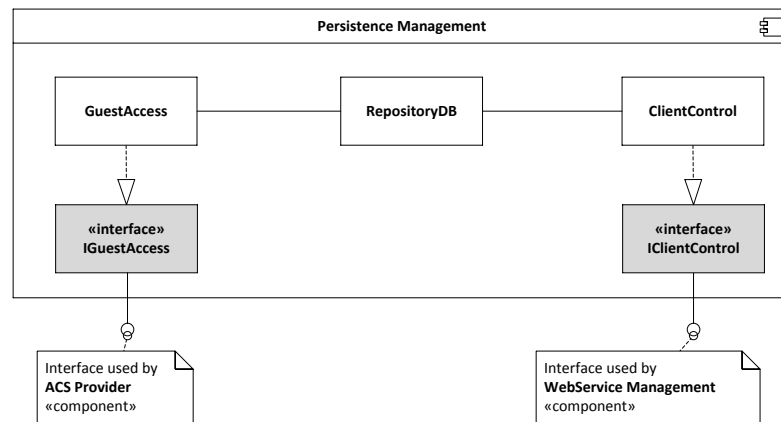


Figure 4.13: The class diagram draft of the *Persistence Management* component (RS).

**RepositoryDB** is the class which manages database access within the *Repository System*. It serves as the facade to the DBMS (Database Management System) and may be replaced at a later time in order to allow the deployment of different DBMS. Notably the data handled by this class and stored in the database is always encrypted.

**IGuestAccess** is an interface used by the *ACS Provider* component in order to provide data from the repository to the *Access Control System*.

**GuestAccess** is an implementation of the *IGuestAccess* interface.

**IClientControl** is an interface used by the *WebService Management* component in order to exchange data with the *Client System*.

**ClientControl** is an implementation of the *IClientControl* interface.

### 4.3.3 Client Management

The class diagram of the *Client Management* component is shown in figure 4.14.

**Client** is a class which represents a Client System. It may contains a hash value which uniquely identifies a Client System, based on immutable information from the client, sent as an alphanumerical identifier.

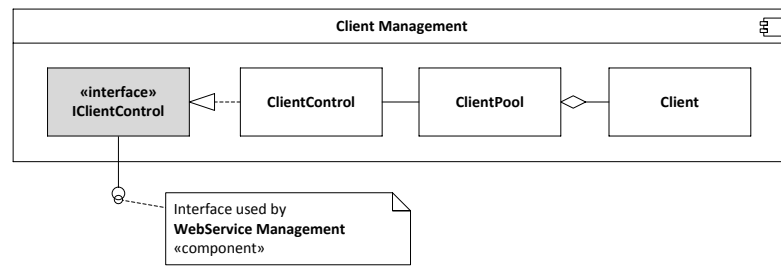


Figure 4.14: The class diagram draft of the *Client Management* component.

**ClientPool** is a class which aggregates several *Client* objects in order to offer a client search method:

```

1 // Returns true if the client is contained in the trusted
2 // client pool or false otherwise.
3 boolean containsClient(String clientId);

```

**IClientControl** is an interface used by the *WebService Management* component in order to authenticate a client to determine whether it has permissions to exchange data with the *Repository System*, or not.

**ClientControl** is an implementation of the *ClientControl* interface.

#### 4.3.4 ACS Provider

The class diagram of the *ACS Provider* component is shown in figure 4.15.

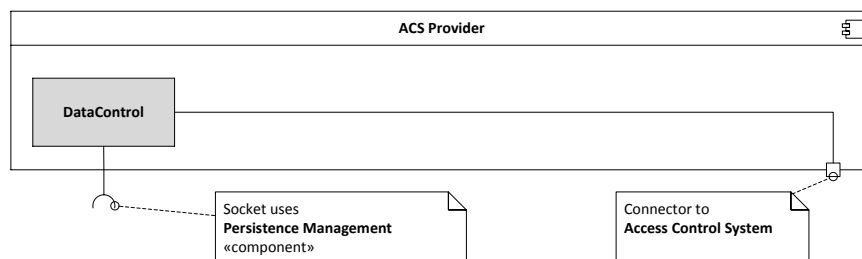


Figure 4.15: The class diagram draft of the *ACS Provider* component.



**DataControl** is a class which uses the *IGuestAccess* interface of the *Persistence Management* component in order to provide the *Access Control System* with data from the *Repository System*. The protocol used for data exchange is not yet defined.

#### 4.4 Access Control System

In this section the proposed architecture for the access control system is shown. In figure 4.16 the logical view is shown with its components, interfaces and connectors. As in previous sections, the detailed description of the interfaces will be done in the components' respective subsections. However, a brief description of the components and their interactions are given in this part. Gray boxes indicate interconnections with other systems or components.

The *WebService Management* component manages the web services provided to the client systems. Upon request it forwards the request to the Guest Access Management component.

The *Guest Access Management* component manages the data transfer between the *Persistence Management* component and, if necessary, the *Repository System*.

The *Persistence Management* component stores data for the time it is shared. It offers de-and encryption capabilities as well as data conversion methods.

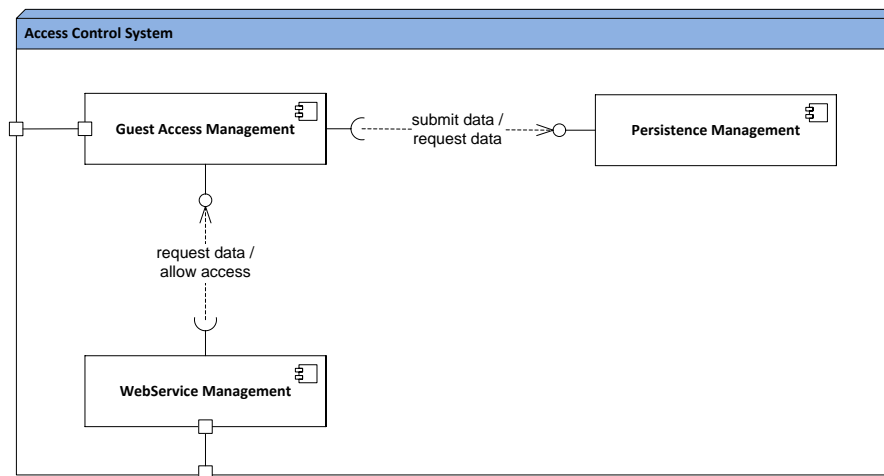


Figure 4.16: The main components of the *Access Control System*.

##### 4.4.1 Guest Access Management

The class diagram of the *Guest Access Management* component is shown in figure 4.17.

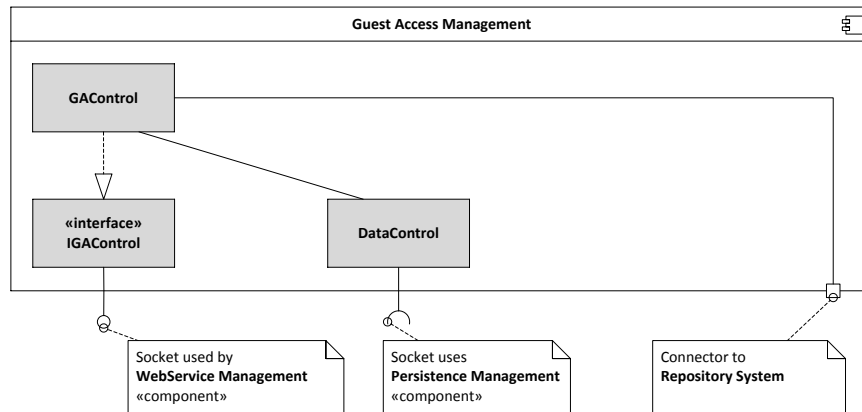


Figure 4.17: The class diagram draft of the *Guest Access Management* component.

**IGAControl** is an interface which is used by the *WebService Management* component in order to authenticate guest or client access and transfer data.

**GAControl** is an implementation of the *IGAControl* interface. It establishes a connection to the *Repository System* in order to request and receive data.

**DataControl** is a class which uses the *IDataSetControl* interface of the *Persistence Management* component in order to submit data previously received from the *Repository System*, or to receive data from the local database which will be sent to the third party system.

#### 4.4.2 Persistence Management

The class diagram of the *Persistence Management* component is shown in figure 4.18.

**DataSet** is a class which represents a set of information for a specific guest access. *DataSet* objects are encrypted using the *CipherUtility* and a key  $K_g$  which is automatically generated and unique for each authorized guest access.

**CipherUtility** is a utility class which is able to encrypt and decrypt *DataSet* objects.

**ConversionUtility** is a utility class which is able to convert *DataSet* objects into supported file types, such as HL7 CCD or CCR, and PDF reports.

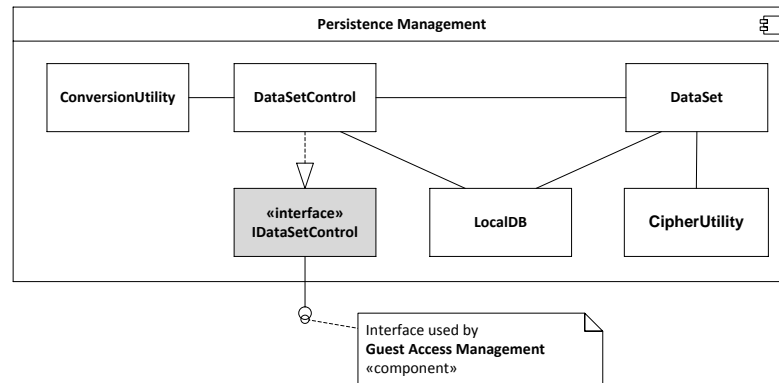


Figure 4.18: The class diagram draft of the *Persistence Management* component (ACS).

**LocalDB** is the class which manages database access within the *Access Control System*. It is used by the *DataSetControl* class in order to persist *DataSet* objects. It accesses the DBMS and may be replaced at a later time in order to allow the deployment of different DBMS.

**IDataSetControl** is an interface which provides methods to the *Guest Access Management* component.

Firstly, it receives data originating from the *Repository System* and the complementary decryption key from the *Client System*. These are used to decrypt the received data, which will then be re-packed for a specific guest access into a *DataSet* object and encrypted it via the *CipherUtility* class to store it in the local database.

Secondly it receives data and file format requests and complementary decryption keys which originate from third party systems. The requested data will be retrieved by the *LocalDB* class and decrypted using the *CipherUtility* class and the submitted decrypt key. The unencrypted data is then converted using the *ConversionUtility* class and sent to the *Guest Account Management* component in order to be ultimately sent to the third party system that requested the data.

Exemplary methods of this interface:

```

1 // Requests and, if successful, returns a set of data for the
2 //   guest account with the ID specified in guestId,
3 //   in the file format specified in fileType,
4 //   which will be decrypted using the key specified in accessKey.
5 byte[] requestData(String guestID, int fileType, String accessKey);
  
```

**DataSetControl** is an implementation of the *IDataSetControl* interface.

### 4.4.3 Webservice Management

The class diagram of the *Webservice Management* component is shown in figure 4.19.

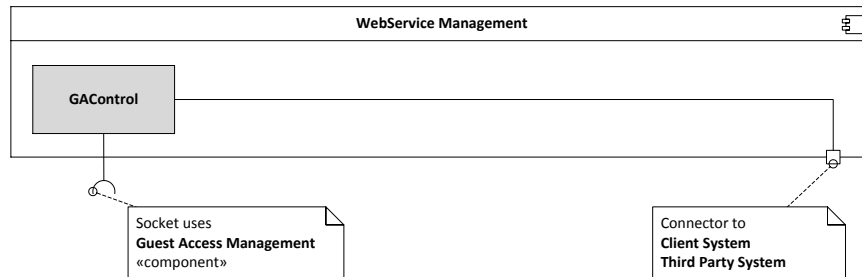


Figure 4.19: The class diagram draft of the *Webservice Management* component (ACS).

**GAControl** is a class which uses the *IGAControl* interface of the *Guest Access Management* component. It also establishes connections to the *Client System* or to a *Third Party System* (e.g. EMR/EHR system, a clinicians web-browser, etc.). It offers methods to the *Client System* in order to add or alter guest accesses and the implied storage of data. It further enables a *Third Party System* to request data previously stored in the local db for guest access. Both connections are done with SOAP via HTTPS.

## 4.5 Interactions & Use Cases

In this section the interactions between components and systems are shown on use case examples. The interactions are based in the architectural drafts shown in the previous sections. The notation is based on UML sequence diagrams and shows additionally the components and systems above the classes or objects, since the class names alone are highly ambiguous. The used method names in the message arrows are placeholders and are not yet part of the interfaces nor classes. The information below the method name annotates the transferred parameters.

These use cases are not complete and do therefore not qualify for a specification. They are merely a tool in order help to understand the presented architectural draft and concept. The chosen use case scenarios are explained informally step-by-step in the following sections. In the informal description it is often referred to classes, when in fact instantiated classes, or objects, are meant.

### 4.5.1 Retrieval and Local Storage of Measurement Data

The retrieval and storage of measurement data in the local database is used to automatically or semi-automatically retrieve data from different sources. In this use case the source is a file containing measurement values, stored on a file system. The process is semi-automatic and has to be initiated by the user. The sequence diagram is shown in figure 4.22.

The data gathering process in this example is semi-automatic and therefore started by the user in the user interface on the *Client System*. Firstly a list of possible gathering methods, so called gatherers, is requested and subsequently received from the *Data Collector* class in the *Data Gathering* component. The user then decides which gatherer he would like to use. In this case the user uses the file system and submits the path to the file to be gathered. A new request is sent to the *DataCollector* class, which then forwards the request according to the chosen gathering method to the *FSConnector* class. The *FSConnector* retrieves the file and extracts the measured values. Additionally it attempts to add information about the values origin, e.g. the measurement devices name. The valueList and origin information are sent back to the *DataCollector*, which creates a *DataSet* object *data* out of these information. This *data* object is then transmitted to the *Personal Health Record* component. The PHR component first checks the *data* for completeness with the *IntegrityHelper* class. If the data is complete, it is stored in the local database by the *LocalDB* class. Additionally the *LocalDB* class encrypts the new *data* using the *CipherUtility* class, which attempts to send the encrypted data *encData* to the *Repository System* in an asynchronous call. The result is a "*ok, sent*" information, which is

sent back all the way to the user interface, where the user gets confirmation, that his data has been stored in the local db and sent to the *Repository System*.

Another confirmation will be sent when the data has been saved in the *Repository System*, this process has been modeled in another use case in section 4.5.2.

Used variable names and there meanings are:

- *gathererList* is the list of available data gathering methods.
- *FS* is the identifier for the file system gathering method.
- *path* is the file path for the file that contains the values to be gathered.
- *valueList* contains the measured values extracted from the file.
- *origin* contains information about the origin of the values, e.g. device name.
- *data* is a *DataSet* object, containing the gathered information.
- *true* is a value of the primitive data type boolean, in this case verifying the completeness of submitted information.
- *encData* is the encrypted *data*.
- *ok, sent* is a verification message, that the data has been locally stored and *encData* has been sent to the *Repository System*.

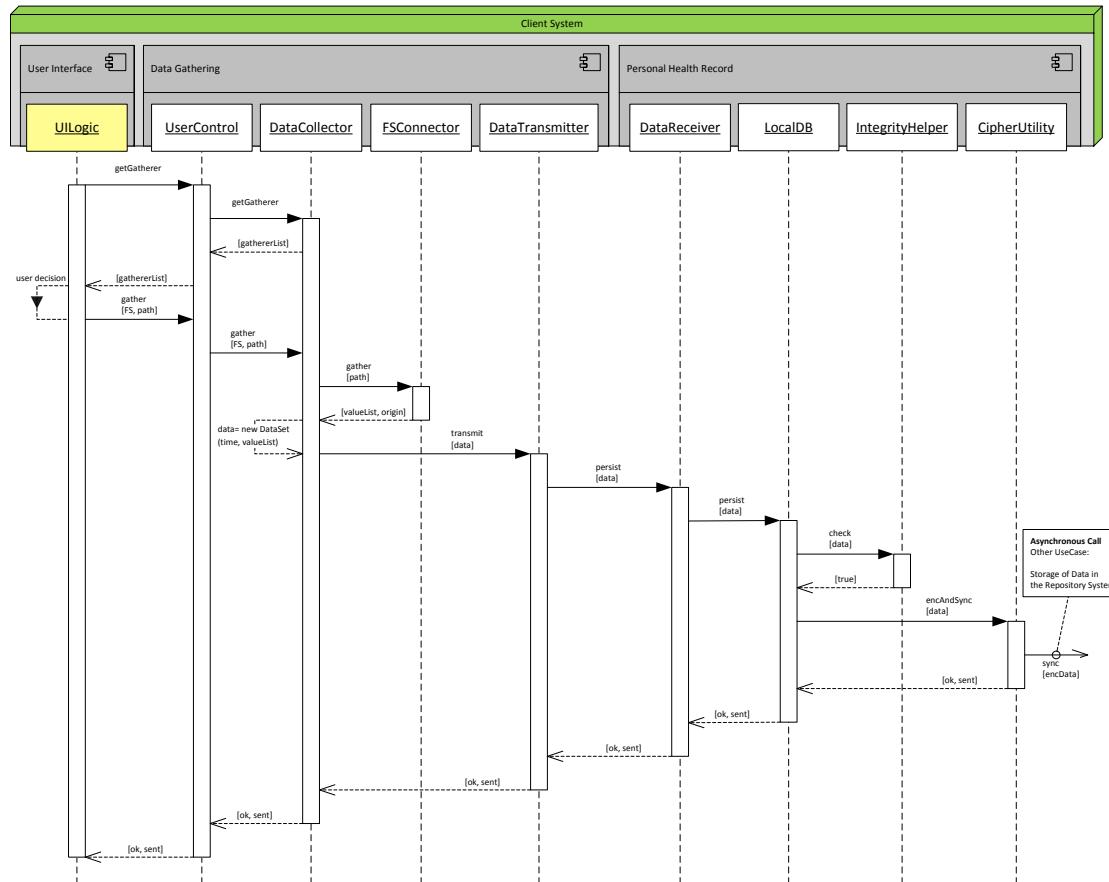


Figure 4.20: Sequence diagram for retrieval and local storage of data from the file system.

### 4.5.2 Storage of Data in the Repository System

The storage of previously received local data in the repository systems database is used to have a central but secure storage of all information, accessible by the clients. The sequence diagram is shown in figure 4.21.

The storage of data is exemplarily initiated by an asynchronous call from the *Personal Health Record* component in the *Client System*. The actions leading to this call are modeled in the use case shown in section 4.5.1.

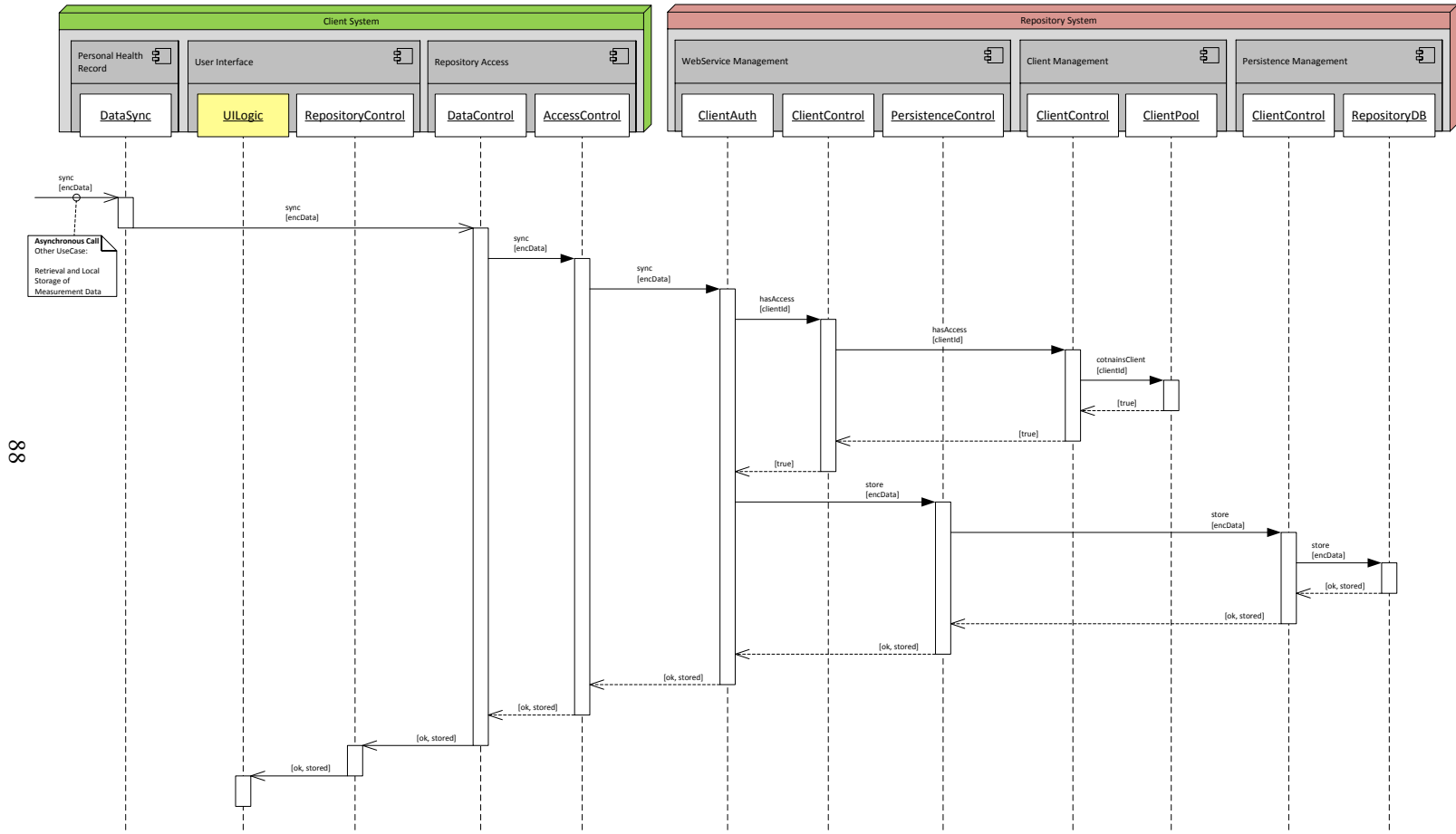
The asynchronous call is causing the *DataSync* class to forward the request for storing the encrypted data *encData* in the *Repository System* to the *Repository Access* component. Here a chain of method calls starts to access the *Repository System* via a web service. The client additionally sends a unique client identifier *clientId*. The first request is going to the *WebService Management* components *ClientAuth* class. The class asks the *ClientControl* class if the particular *Client System*, identified by its *clientId*, is allowed to access the repository. This is achieved by simply asking the *ClientPool*, if it contains the specified client. In case the client has access, a *true* value is returned to the *ClientAuth* class, which is then requesting the storage of the encrypted data *encData*. For this purpose it sends the request through the *PersistenceControl* class to the *Persistence Management* component. The *ClientControl* class handles the request originating from the client and stores the data in the repository database, using the *RepositoryDB* class.

The return value of this chain of actions is a *ok, stored* message, which is returned all the way back to the *Client System*, more specific the *Repository Access* component. From there it is sending an asynchronous "*ok, stored*" message to the user interface in order to inform the user that his data has been stored in the *Repository System*.

Used variable names and their meanings are:

- *encData* is the encrypted data to be stored.
- *clientId* is an identifier, which uniquely identifies a client system.
- *path* is the file path for the file that contains the values to be gathered.
- *ok, stored* is a verification message, that the data has been stored in the repository system.





88

Figure 4.21: Sequence diagram for storage data in the repository system.

### 4.5.3 Retrieval and Storage of Data for Guest Access

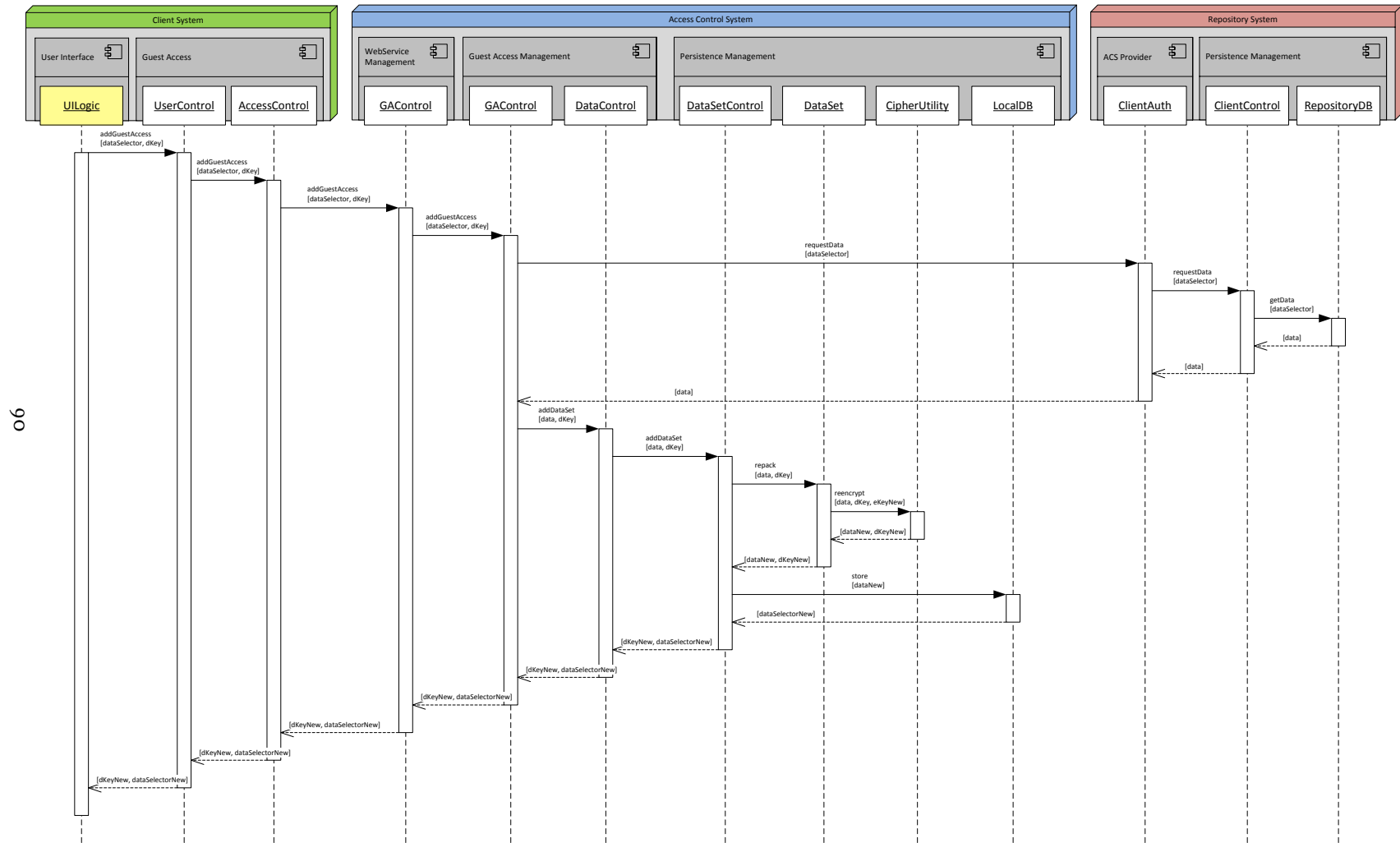
The retrieval and storage of data for guest access is used to share data contained in the *Repository System* with third party systems (e.g. clinicians, friends, family members) by creating guest access and store the data in the *Access Control System*. The sequence diagram is shown in figure 4.22.

The sharing process is initiated by the user in the *Client System* and serves the purpose to store specific data from the *Repository System* (RS) in the *Access Control System* (ACS) in order to share it with third party systems in a chosen file format. The separation of ACS and RS is done do detach the decryption of personal data from the *Repository System*, which stores all the users information, to minimize the risk of data exposure.

The user selects the data to be shared, which is identified by a *dataSelector* and sends his decryption key *dKey*. The user interface sends these parameters through the client systems *Guest Access* component to the *Access Control System*. The *Access Control System* then keeps the decryption key and passes the *dataSelector* on to the *Repository System* in order to receive the requested *data*. The *data* is then sent back and received by the ACS' *Guest Access Management* component, which attempts to add a data set to the local database. In order to do so, it firstly decrypts and then re-encrypts the received encrypted data with a new key *eKeyNew* in order to make it only readable by a specific decryption key *dKeyNew*. The newly encrypted *dataNew* is subsequently stored in the local database, which returns a new data selector *dataSelectorNew*. This *dataSelectorNew* and the decryption key *dKeyNew* can be used to access the shared data in the *Access Control System* later on, and are sent back to the *Client System*, which visualizes them in an appropriate way, e.g. as a download link for the clinician.

Used variable names and there meanings are:

- *dataSelector* is used to identify the data in the database.
- *dKey* is the decryption key used to decrypt the stored data.
- *data* is the encrypted data from the *Repository System*.
- *eKeyNew* is the newly generated encryption key, used to encrypt the data for the third party access.
- *dataNew* is the data encrypted with *eKeyNew*.
- *dKeyNew* is the complementary decryption key, used to decrypt *dataNew*.
- *dataSelectorNew* is the new data selector, used to identify data in the database.



90

Figure 4.22: Sequence diagram for retrieval and storage of data for guest access in the ACS.

### 4.5.4 Health Provider Access to Shared Data via a Web-Browser

The retrieval of data by a health provider is used by clinicians or healthcare professionals to gain access to previously shared data in the *Access Control System*. The sequence diagram is shown in figure 4.23.

The process is initiated by the web browser opening the address of the web service in the *Access Control System*. The *WbService Management* component then delegates the request through the *Guest Access Management* component, to the *Persistence Management* component. Then the *DataSetControl* class is retrieving the encrypted Data from the *LocalDB* class and therefore the local database. After the retrieval the encrypted data is sent along with the decryption key *dKey* to the *DataSet* object, which is using the *CipherUtility* in order to decrypt its *data*. The *data* is then returned to the *DataSetControl* class, which then determines the file format. Since in this example, the health provider did not specify a certain file format, it chooses to use a PDF (Portable Document Format) report, assuming that the health provider is able to open PDF files. The *DataSetControl* class uses the *Conversion Utility* to convert the *data* into a *PDF file* report and returns it. This *PDF file* will be returned all the way back, through the web service, to the health provider, which receives it with his web-browser.

The decryption process would fail, if an incorrect decryption key is provided, and consequently the data would not leaving the system. Another error would occur, if the *dataSelector* does not fit the submitted decryption key.

Used variable names and there meanings are:

- *dataSelector* is used to identify the data in the database.
- *dKey* is the decryption key used to decrypt the stored data.
- *userAgent* contains information about the client that sent the request, it is automatically sent by web-browsers.
- *encData* is the encrypted data from the database.
- *data* is the decrypted data from the database.
- *PDF* is an identifier for a PDF file format.
- *FilePDF* is the data contained in a PDF file report.

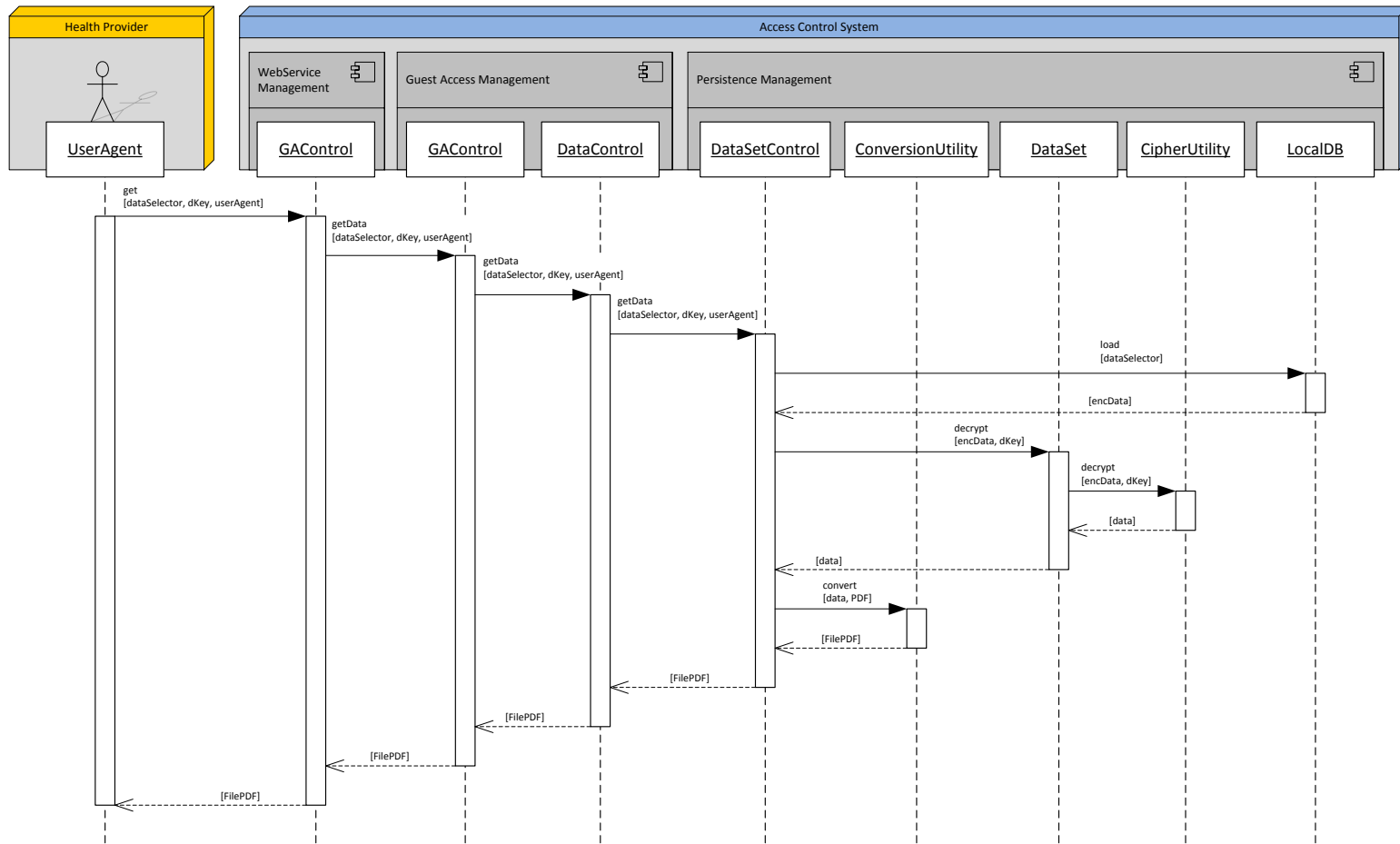


Figure 4.23: Sequence diagram for health provider access to shared data via a web-browser.

### 4.5.5 Visualization of PHR Data as a Graph

The visualization of data as a graph is a way to make data more comprehensive for the user and therefore part of the *Empowerment* component. The sequence diagram is shown in figure 4.24.

The visualization is started by the user in the user interface. The request for visualization is then sent to the *Empowerment* component, where the *UserControl* firstly receives it. It then forwards the identifier for the data to be visualized *dataSelector* to the *GraphVis* class, which is used for graph visualizations. The *GraphVis* class then requests the data to be visualized through the *DataRequester* class, from the *Personal Health Record* component. Subsequently the data is returned step-by-step to the *GraphVis* class, which creates a *DSVisualization* object, containing information about the graph generated with the previously received data. This *DSVisualization* object is returned all the way back to the user-interface, which visualizes the object.

Used variable names and there meanings are:

- *dataSelector* is used to identify the data in the database.
- *GRAPH* is an identifier for the kind of visualization that is requested.
- *DSVisualization* is the *DSVisualization* object, that contains the visualized graph information for the user interface.

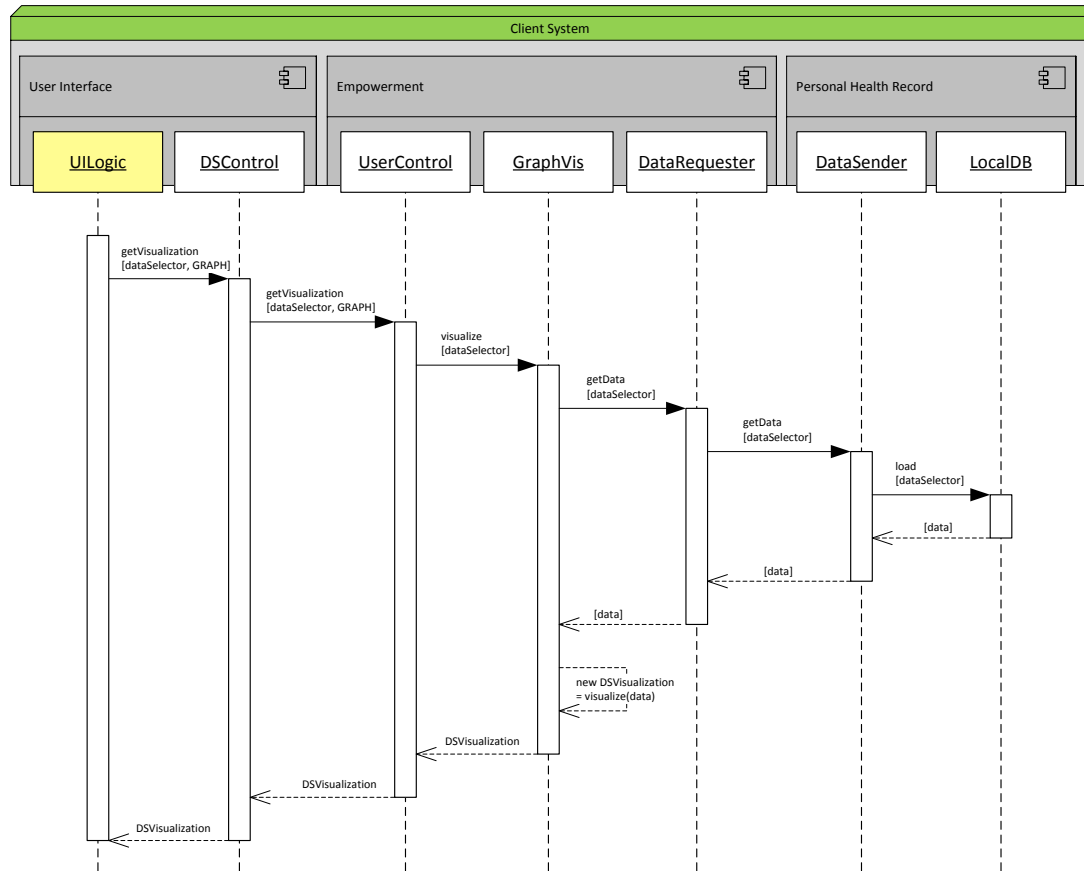


Figure 4.24: Sequence diagram for the visualization of PHR data as a graph.

### 4.5.6 Retrieval of Condition Information

The retrieval of information about a condition is used to increase a users health literacy and therefore part of the *Empowerment* component. The sequence diagram is shown in figure 4.25.

The process is initiated by the user in the user interface and sends a request with the particular *conditionName* to the *DSControl* class, which forwards the request to the *Empowerment* component. Within the *Empowerment* component, the *KnowledgeBase* firstly checks if the condition is already contained in the database. In this example it is not, and therefore a *false* is returned. Consequently, the information request is sent to the *OntologyHelper* class, which generates a list of search terms based on synonyms and abbreviations. This *searchList* is then passed on through the *WebSourceParser* to the *WebSourceFinder*, which is using specified search engines to find the defined search terms. The result is returned as markup, in this case HTML (Hypertext Markup Language), which is parsed by the *WebSourceParser* class. The resulting information is further returned as text to the *OntologyHelper* class, which interprets the information according to the previously submitted condition, and attempts to gather semantic meaning from it. The result is returned as a *knowledgeItem* to the *KnowledgeBase*, which stores it for future use and forwards it back to the *UserControl* class. The *UserControl* class creates a *DSVisualization* object *visualization* out of the returned information. The *visualization* object is then sent back to the user interface to be finally visualized.

Used variable names and there meanings are:

- *conditionName* is the name of the condition.
- *false* is a value of the primitive data type boolean, in this negating the containment of condition information in the knowledge base.
- "*condition*" is identifying the type of information that the system will look for, in this case a condition.
- *searchList* is a list of search terms, which are synonymous to the original search terms.
- *markupHTML* is the retrieved content as hypertext markup (HTML).
- *infoText* is the information in textual representation.
- *knowledgeItem* is the semantic information of the search result.
- *visualization* is the *DSVisualization* object which contains the knowledge information to be visualized in the user interface.



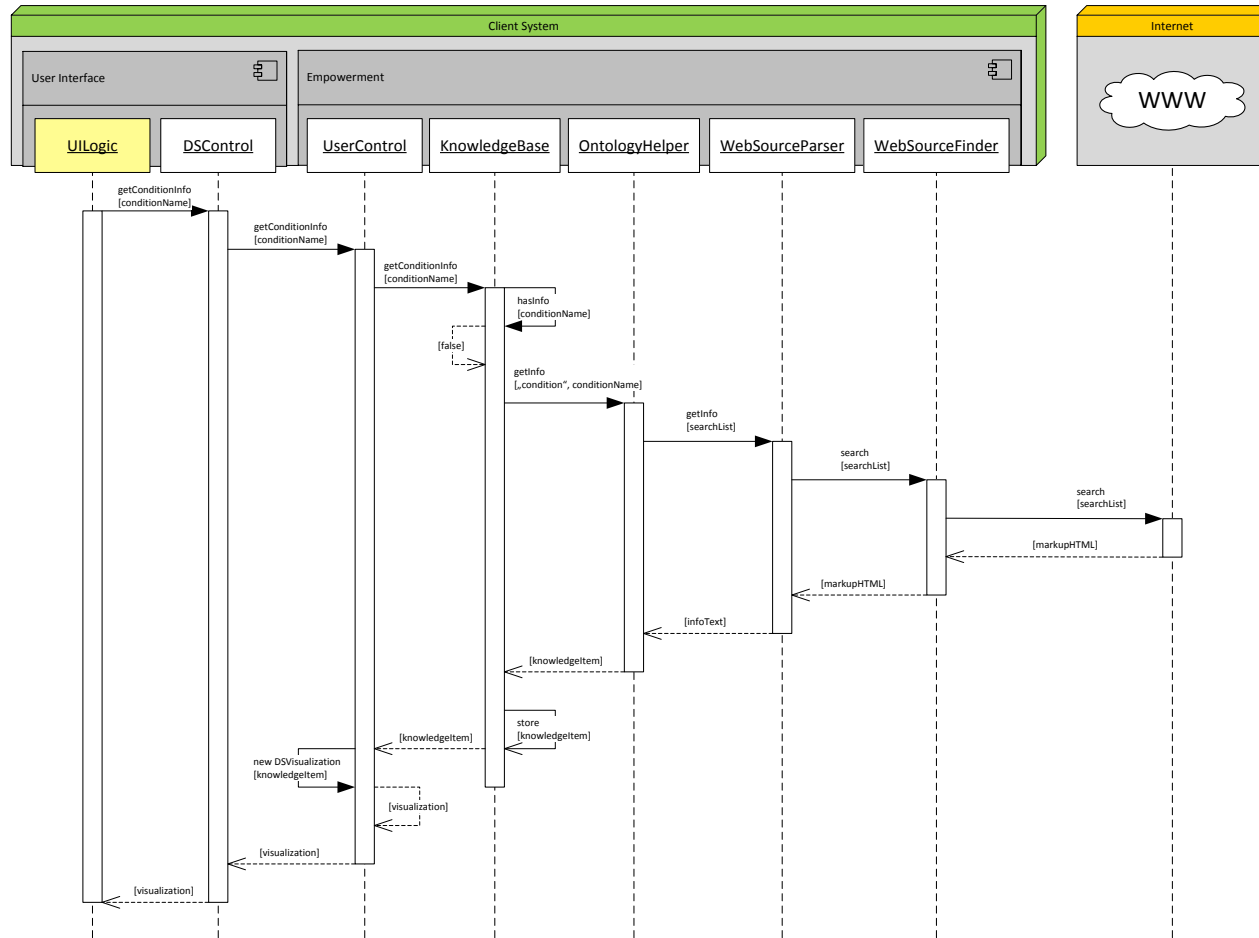


Figure 4.25: Sequence diagram for the retrieval of condition information.

## 4.6 Technological Considerations

### 4.6.1 Health Diary Entry and iPHR interface

Lähteenmäki et al. (2009) wrote an article about interoperability in PHRs. In this article they focused on the exchange of non-clinical PHR information and introduced the HDE (Health Diary Entry).

As a technical solution we propose a new data structure, Health Diary Entry document, which complements the existing standards in supporting transfer of non-clinical information. The data structure includes a mechanism for binding with external vocabularies and ontologies to achieve semantic interoperability. [...] The experiences have shown the suitability of the approach when information needs to be collected from different services into a common PHR database. (Lähteenmäki et al., 2009, p. 1729)

In the article the HDE was transferred using SOAP (Simple Object Access Protocol) and an iPHR interface which is placed between the PHR system and a trusted service which can submit or receive information. In the architectural draft proposed in this work the HDE can be used as a suitable data structure to contain information added by the user and data collected by measurement devices.

### 4.6.2 SNOMED CT

Regarding ontologies, there are several medical ontologies and terminology databases available. One of the most popular is SNOMED CT (SNOMED Clinical Terms), which is currently maintained by the International Health Terminology Standards Development Organisation (IHTSDO).

SNOMED CT provides the core general terminology for the electronic health record (EHR) and contains more than 311,000 active concepts with unique meanings and formal logic-based definitions organized into hierarchies. International Health Terminology Standards Development Organisation (30.08.2013)

This technology can be used in order to populate the knowledge base with names of conditions and their definitions. However, there is another advantage of this technology, which is based in its conformity with OWL (Web Ontology Language).

[..] SNOMED CT is conformant to OWL 2 EL. OWL stands for Web Ontology Language and is the standard of choice for semantic web. OWL 2 EL is subset [sic!] of the OWL language that intentionally restricts some semantic relationships (negation and disjunction specifically) that make querying extremely large data sets something that computers can do more easily. (Trotter and Uhlman, 2013, p. 160)

The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. W3C OWL Working Group (09.12.2012)

Consequently, these technologies can not only be used to populate knowledge bases, it can further be used in order to gather semantic information out of textual representations, and query information efficiently in the knowledge base. However, more work is needed to evaluate these technologies for use in the proposed system.

## 5 Concept Evaluation

In this chapter the proposed concept will be qualitatively evaluated.

The focus in this conception was lying on the separation of concerns regarding the core functions of record keeping, empowerment, and data gathering. All these functions were separated into modular components, yet all remained mostly within the *Client System*. External systems were only drafted for information sharing options (*Access Control System*), and centralized storage and backup (*Repository System*).

The system fulfilled most of the non-functional requirements. By using web services between the different systems, and very little classes across the components' boundaries within the systems, the principle of loose coupling made the easy replacement of whole components feasible.

On the other hand, some of the architectural requirements have not or not completely been fulfilled. Further some details about classes and data structures remain vague and unclear.

### 5.1 Requirement Review

In this section the architectural requirements defined in section 3.6.4 (Architectural Requirements) are reviewed and annotated depending on their fulfillment in the current state of the architectural draft previously presented. The reviewed requirements are shown in table 5.1.

There are three categories of fulfillment:

F *Fulfilled*: The requirement is fulfilled or fully considered in the current draft.

A *Approached*: The requirement is approached, all or most of its pre-requirements have been fulfilled, yet it cannot be considered fulfilled in the current draft.

N *Not-Fulfilled*: The requirement as well as its pre-requirements are not fulfilled.

According to the review there are 17 architectural requirements fulfilled, 19 are approached, and 5 are not fulfilled at all.

Table 5.1: List of architectural requirements and their coverage by the architectural draft.

| #    | Requirement  | * |
|------|--|---|
| ARo1 | The system shall be a distributed system consisting mainly of interconnected heterogeneous clients.  | F |
| ARo2 | The clients shall work on the following platforms and therefore require multiple implementations: Android, iOS, Linux, Windows, and Windows Phone.   | F |
| ARo3 | The system shall have an online repository that stores all the users encrypted data and serves as a backup for the clients.  | F |
| ARo4 | The repository shall store and transfer only encrypted data.   | F |
| ARo5 | The repository shall be managed on an online storage with logic for security and persistence only, by an authority yet to be determined.   | F |
| ARo6 | Every client shall be able to manage the clients (add, delete, or modify the trusted client pool). Any modification of the trusted client pool shall require the users master key.   | A |
| ARo7 | Every client within the trusted client pool shall be able to transfer encrypted data to and from the repository.   | F |
| ARo8 | Every client shall run exactly one client application.   | N |
| ARo9 | Every client application shall contain a user interface.   | A |
| AR10 | Every client application shall store user data depending on the client device capacity. Devices with less memory shall store only recent and often required data, while devices with higher capacity shall store more information. Not stored information shall be received from the repository on demand. | A |
| AR11 | Every client application shall offer access to the Personal Health Record data.  | F |
| AR12 | Every client application shall contain only logic which functions with the performance offered by the client device.   | N |
| AR13 | Besides the restriction due to the performance requirements, the user shall be able to add or delete modules to or from every client separately as he wishes.  | A |
| AR14 | All data shall be stored encrypted with an encryption key. A decryption key shall exist for decryption of the data.  | F |
| AR15 | Only trusted clients shall be able to submit data to, or retrieve data from the repository.  | F |
| AR16 | The PHR data shall be stored in a dedicated database.  | F |
| AR17 | The knowledge base shall be stored in a dedicated database.  | F |

| #    | Requirement   | * |
|------|---|---|
| AR18 | The persistence logic shall separate the DBMS and access methods in order to make the system independent from the DBMS in use.                        | A |
| AR19 | The persistence logic of the repository shall check the data to be persisted in the PHR for inconsistencies or conflicts with data already contained. | A |
| AR20 | Connectivity methods shall use the capabilities of the client devices.  | A |
| AR21 | The system shall natively support collection of data from files in a file system.   | F |
| AR22 | The system shall natively support collection of data via the Bluetooth protocol.  | A |
| AR23 | The system shall natively support collection of data via WiFi (IEEE 802.11) protocol.   | A |
| AR24 | The system shall natively support the collection of raw sensor data.  | A |
| AR25 | The system shall natively support the collection of measurement values.   | F |
| AR26 | The system shall natively support the collection of information from manual user entries.   | F |
| AR27 | Collected data shall always be automatically annotated with the source and timestamp.   | F |
| AR28 | Collected data shall be automatically annotated with location information, if possible.   | A |
| AR29 | The system shall natively support PHR data import from the file system.   | N |
| AR30 | The system shall natively support PHR data import upon request of HIS (e.g. EHR, PHR) via dedicated Interfaces.                                       | N |
| AR31 | The system shall natively support PHR data export to the file system.   | A |
| AR32 | The system shall natively support PHR data export to Health Information Systems (e.g. EHR, PHR) Interfaces.   | A |
| AR33 | The system shall support standardized file types for import and export (e.g. HL7 CCR/CCD)   | A |
| AR34 | The system shall natively support web protocols (e.g. HTTP/HTTPS, FTP) for knowledge retrieval.   | F |
| AR35 | The system shall include a mechanism to parse retrieved web elements for information.   | F |
| AR36 | The application logic shall be modular, where one module contains logic for a certain function.   | A |
| AR37 | The modules shall be independent from the clients platform.   | A |
| AR38 | The modules shall have to be approved by an authority yet to be determined.   | N |

| #    | Requirement  | * |
|------|--|---|
| AR39 | The system shall allow modules for the client applications user interface in order to change the appearance and behavior of the application.                                 | A |
| AR40 | The system shall allow modules for the empowerment / decision support component in order to be fit for future research findings, visualizations, communities, etc.           | A |
| AR41 | The system shall allow modules for the data collection component in order to be fit for future measurement and data gathering devices, such as ADL recognition systems, etc. | A |

## 5.2 Proposed Changes

This development of an architectural draft is an iterative process. Due to time restrictions of this thesis, only the first iterations could be done in order to present this work. In this section some changes are presented which should be done in the next iteration cycles.

The following changes should be considered for future development cycles:

- The *Access Control System* does not yet offer import capabilities. The introduction of such would require a more thorough and detailed data management within the clients PHR and the *Repository System*.
- It should be considered to completely detach the empowerment component into a separate system, due to the complexity of decision support and empowerment methods.
- It should be considered to detach the knowledge base as well, preferably conjoined with the mentioned dedicated empowerment system.
- Some work should go into the development of suitable abstract data types, which can be used inside components in order to benefit the systems modifiability. For interactions these should be kept as limited as possible to support the loose coupling principle.
- The classes and components require new names in order to remove ambiguous names. Ambiguous names may hinder an implementation.
- The usability is a huge area and should be done in a dedicated research since it is important for the user to understand the presented information correctly and accurately. As a consequence of this research a suitable user interface logic can be chosen or developed.

- 
- A suitable protocol for the communication between the *Access Control System* and the *Repository System* needs to be found. The protocol should be efficient in case the two systems are located on the same machine.



## **6 Conclusion & Future Work**

In this chapter the work done in this thesis is concluded and possible future work based on this thesis' work is described.

### **6.1 Conclusion**

The aim of this thesis was to draft a system concept or architecture for a comprehensive patient empowering and pervasive healthcare system. An extensive requirement analysis has been done based on relevant literature and covered a variety of different viewpoints regarding the proposed system and its projected components. This requirement analysis, it's results and later derived architectural requirements altogether, are a solid basis for a requirement list of a full specification.

The scope of the system has been quite vague in the beginning and narrowed down with every viewed resource that has been used for the requirement analysis. The vague requirements could not have been described more detailed, since the systems concept itself is encompassing a vast area of possible functions and features. Therefore, based on these requirements a system concept has been drafted using techniques from the area of software architecture. These were identifying architectural requirements and constructing logical views of the system. The logical views with their informal descriptions are closely related to most of the architectural requirements. These logical views may serve as a basis for a system specification as well. However, some architectural requirements have not been fulfilled by the presented architectural draft and therefore more work has to be done in this area.

### **6.2 Future Work**

Future work based on this thesis primarily includes the continued effort to complete the architectural draft in order to meet all architectural requirements. Further all the vague requirements have to be consolidated into a detailed and precise requirement list in order to serve for a full specification. Farther tasks based on a specification include a proof of concept implementation with basic use-case scenarios, followed by usability testing and evaluation.

## 6 Conclusion & Future Work

---

Future work further includes a market analysis in order to determine the financial value of the proposed system and to further find investors in order to finance the development. During the market analysis an empiric research should be undertaken to identify target audiences and the perceived usefulness of the system.

## Bibliography

- [Agarwal et al. 2013] AGARWAL, Ritu ; ANDERSON, Catherine ; ZARATE, Jesus ; WARD, Claudine: If We Offer it, Will They Accept? Factors Affecting Patient Use Intentions of Personal Health Records and Secure Messaging. In: *Journal of Medical Internet Research* 15 (2013), Nr. 2, p. e43. – ISSN 1438-8871
- [Alpay et al. 2010] ALPAY, Laurence L. ; BLANSON HENKEMANS, Olivier ; OTTEN, Wilma: E-health Applications and Services for Patient Empowerment: Directions for Best Practices in The Netherlands. In: *Telemedicine and e-Health* 16 (2010), Nr. 7, p. 787–791
- [Andrew et al. 2013] ANDREW, Adrienne H. ; EUSTICE, Kevin ; HICKL, Andy: Using location lifelogs to make meaning of food and physical activity behaviors. In: *7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013, 2013*, p. 408–411. – ISBN 978-1-936968-80-0
- [Arnrich et al. 2010] ARNRICH, Bert ; MAYORA, O. ; BARDRAM, Jakob ; TRÖSTER, G.: Pervasive Health - Paving the Way for a Pervasive, User-Centered and Preventive Healthcare Model. (2010)
- [ASTM 30.08.2013] ASTM: *ASTM E2369 - 12 Standard Specification for Continuity of Care Record (CCR)*. 30.08.2013. – URL <http://www.astm.org/Standards/E2369.htm>. – date visited: 04.09.2013
- [Baird et al. 2011] BAIRD, Aaron ; NORTH, Frederick ; RAGHU, T. S.: Personal Health Records (PHR) and the future of the physician-patient relationship. In: *the 2011 iConference, 2011*, p. 281–288. – ISBN 978-1-4503-0121-3
- [Bardram et al. 2007] BARDRAM, Jakob E. (Editor) ; MIHAILIDIS, Alex (Editor) ; WAN, Dadong (Editor): *Pervasive Computing in Healthcare*. Boca Raton : CRC Press, 2007. – ISBN 978-0849336218
- [Burbeck 1987] BURBECK, Steve: *Applications Programming in Smalltalk-80: How to use Model-View- Controller (MVC)*. Palo Alto and Calif : Softsmarts, Inc., 1987

## Bibliography

---

- [deBronkart 2010] DEBRONKART, Dave: *Laugh, sing, and eat like a pig: How an empowered patient beat stage IV cancer (and what healthcare can learn from it)*. Media and PA : Changing Outlook Press, 2010. – ISBN 0981650430
- [Fuji et al. 2012] FUJI, Kevin T. ; ABBOTT, Amy A. ; GALT, Kimberly A. ; DRINCIC, Andjela ; KRAFT, Mark ; KASHA, Ted: Standalone personal health records in the United States: meeting patient desires. In: *Health and Technology* 2 (2012), Nr. 3, p. 197–205. – ISSN 2190-7188
- [Gonzales and Tran 2012] GONZALES, Pedro ; TRAN, Binh Q.: Formal and de facto standards and guidelines for personal health records (PHRs). In: *38th Annual Northeast Bioengineering Conference (NEBEC), 2012, 2012*, p. 1–2
- [Hamari and Eranti 2011] HAMARI, Juho ; ERANTI, Veikko: Framework for Designing and Evaluating Game Achievements. In: *Think Design Play: The fifth international conference of the Digital Research Association (DIGRA)*, DiGRA/Utrecht School of the Arts, 2011
- [Health Level Seven International 03.09.2013] HEALTH LEVEL SEVEN INTERNATIONAL: *HL7*. 03.09.2013. – URL <http://www.hl7.org>. – date visited: 04.09.2013
- [International Health Terminology Standards Development Organisation 30.08.2013] INTERNATIONAL HEALTH TERMINOLOGY STANDARDS DEVELOPMENT ORGANISATION: *SNOMED CT: About*. 30.08.2013. – URL <http://www.ihtsdo.org/snomed-ct/snomed-ct0/>. – date visited: 30.08.2013
- [Kahn et al. 2009] KAHN, J. S. ; AULAKH, V. ; BOSWORTH, A.: What It Takes: Characteristics Of The Ideal Personal Health Record. In: *Health Affairs* 28 (2009), Nr. 2, p. 369–376. – ISSN 0278-2715
- [Lähteenmäki et al. ] LÄHTEENMÄKI, J. ; LEPPÄNEN, J. ; KAIJANRANTA, H.: Interoperability of personal health records. In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, p. 1726–1729
- [Lähteenmäki et al. 2009] LÄHTEENMÄKI, J. ; LEPPÄNEN, J. ; KAIJANRANTA, H.: Interoperability of personal health records. In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, p. 1726–1729
- [Ogbuji et al. 2011] OGBUJI, Chimezie ; GOMADAM, Karthik ; PETRIE, Charles: Web Technology and Architecture for Personal Health Records. In: *IEEE Internet Computing* 15 (2011), Nr. 4, p. 10–13. – ISSN 1089-7801

- [Puentes et al. 2013] PUENTES, John ; MONTAGNER, Julien ; LECORNU, Laurent ; LÄHTEENMÄKI, Jaakko: Quality Analysis of Sensors Data for Personal Health Records on Mobile Devices. In: BALI, Rajeev (Editor) ; TROSHANI, Indrit (Editor) ; GOLDBERG, Steve (Editor) ; WICKRAMASINGHE, Nilmini (Editor): *Pervasive Health Knowledge Management*. New York and NY : Springer New York, 2013, p. 103–133. – ISBN 978-1-4614-4513-5
- [Robison et al. 2012] ROBISON, James ; LI BAI ; MASTROGIANNIS, Dimitrios ; CHIU, C. ; TAN JIE WU: A Survey on PHR Technology. In: *14th International Conference on E-Health Networking, Applications and Services (Healthcom), IEEE 2012*. [S.l.] : [s.n.], 2012, p. 184–189. – ISBN 1457720396
- [Swan 2009] SWAN, Melanie: Emerging Patient-Driven Health Care Models: An Examination of Health Social Networks, Consumer Personalized Medicine and Quantified Self-Tracking. In: *International Journal of Environmental Research and Public Health* 6 (2009), Nr. 2, p. 492–525. – ISSN 1660-4601
- [Tang et al. 2006] TANG, P. C. ; ASH, J. S. ; BATES, D. W. ; OVERHAGE, J. M. ; SANDS, D. Z.: Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption. In: *Journal of the American Medical Informatics Association* 13 (2006), Nr. 2, p. 121–126. – ISSN 1067-5027
- [Trotter and Uhlman 2013] TROTTER, Fred ; UHLMAN, David: *Hacking Healthcare: A Guide to Standards, Workflows, and Meaningful Use*. Sebastopol and CA : O'Reilly Media, 2013. – ISBN 9781449305024
- [Varshney 2009] VARSHNEY, Upkar: *Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring*. Springer US, 2009. – ISBN 978-1-4419-0214-6
- [W3C OWL Working Group 09.12.2012] W3C OWL WORKING GROUP: *OWL 2 Web Ontology Language Document Overview (Second Edition)*. 09.12.2012. – URL <http://www.w3.org/TR/owl2-overview/>. – date visited: 30.08.2013

# Glossary

|             |  |                         |
|-------------|--|-------------------------|
| <b>AAL</b>  | <i>Ambient Assisted Living</i> is the usage of technologies in order to support and assist the elderly or impaired people in their daily lives.                          | 38                      |
| <b>ADL</b>  | <i>Activities of Daily Living</i> is a term for individuals' daily self care activities in healthcare.   | 38, 39, 61, 102         |
| <b>BME</b>  | <i>Biomedical Engineering</i> is a field which aims to combine engineering and medicine.   | 4                       |
| <b>CCD</b>  | The <i>Continuity of Care Document</i> is an XML-based health record standard and part of the CDA (Clinical Document Architecture) by HL7 (Health Level Seven).          | 20, 60, 81, 101         |
| <b>CCR</b>  | The <i>Continuity of Care Record</i> is a health record standard for patient data exchange by ASTM International.  | 20, 60, 81, 101         |
| <b>DBMS</b> | A <i>Database Management System</i> is a software system used to manage databases and its contents.  | 59, 66, 72, 78, 82, 101 |
| <b>DS</b>   | <i>Decision Support</i> aims to help individuals make decisions about specific circumstances.  | 40, 73                  |
| <b>ECCG</b> | An <i>Electrocardiogram</i> is the recording of heart activity over time.  | 25                      |
| <b>EHR</b>  | An <i>Electronic Health Record</i> is a record which contains health information about an individual from a healthcare institution or healthcare professionals.          | 5, 6                    |
| <b>EMR</b>  | An <i>Electronic Medical Record</i> is a collection of health records about a specific individual which can be distributed over several different systems and locations. | 6                       |
| <b>FTP</b>  | The <i>File Transfer Protocol</i> is a protocol used for the transfer of files over a computer network.  | 60, 72, 101             |

|                    |   |                                 |
|--------------------|---|---------------------------------|
| <b>HDE</b>         | <i>Health Diary Entry</i> is a proposed data structure for the exchange of non-clinical PHR information.  | 97                              |
| <b>HIS</b>         | A <i>Health Information System</i> is a system which assists in managing health related information. HIS encompass PHR and EHR/EMR, decision-support and data capture systems. The proposed system in this work is considered an HIS. | 60, 101                         |
| <b>HL7</b>         | <i>Health Level Seven</i> is a non-profit organization and their set of interoperability standards in healthcare informatics, e.g. HL7 v3.0, HL7 CDA, HL7 RIM.  | 60, 81, 101                     |
| <b>HTML</b>        | <i>Hypertext Markup Language</i> is a language which describes contents for the display in a web-browser.   | 72, 95                          |
| <b>HTTPS</b>       | The <i>Hypertext Transfer Protocol Secure</i> is the HTTP protocol on top of a security protocol layer (usually SSL/TLS).   | 60, 64, 68, 69, 72, 77, 83, 101 |
| <b>HTTP</b>        | The <i>Hypertext Transfer Protocol</i> is a protocol used to transmit structured text and other media over a computer network.  | 60, 72, 101                     |
| <b>Lifelogging</b> | <i>Lifelogging</i> is a term used for the logging or capture of any data about a persons life.  | 36                              |
| <b>MI</b>          | <i>Medical Informatics</i> is a field which aims to combine computer science or informatics, and medicine.  | 4                               |
| <b>MVC</b>         | <i>Model View Controller</i> is a design pattern for user interfaces.   | 75                              |
| <b>OTC</b>         | <i>Over-The-Counter</i> is a term which describes the purchase of products without professional notice, e.g. medication without the notice of a doctor.   | 24                              |
| <b>OWL</b>         | The <i>Web Ontology Language</i> is a set of knowledge representation languages for ontologies used in the semantic web movement.   | 97                              |
| <b>PCHR</b>        | A <i>Personally Controlled Health Record</i> or <i>Patient Controlled Health Record</i> are PHRs with an emphasis on being controlled by the individual.  | 6                               |

|                  |   |                        |
|------------------|---|------------------------|
| <b>PDF</b>       | The <i>Portable Document Format</i> is a file format used to render a document independently from the applications platform.  | 81                     |
| <b>PHR</b>       | A <i>Personal Health Record</i> is a record which contains health information about an individual, including information from the individual himself, and being controlled by the individual. Sometimes PHR refers to a whole PHR system, which utilizes the health record and offers various additional functions. | 5                      |
| <b>QST</b>       | <i>Quantified Self-tracking</i> is an application of the Quantified Self movement.  | 36                     |
| <b>QS</b>        | The <i>Quantified Self</i> is a movement which aims to collect data about a persons life via different technologies.  | 36                     |
| <b>Q&amp;A</b>   | Abbreviation for <i>Questions and Answers</i> .   | 44                     |
| <b>SNOMED CT</b> | <i>SNOMED Clinical Terms</i> is a comprehensive collection of clinical terms and their meanings.  | 97                     |
| <b>SOAP</b>      | Simple Object Access Protocol is a protocol specification used for the exchange of structured information via web services.   | 64, 68, 69, 77, 83, 97 |
| <b>UML</b>       | The <i>Unified Modeling Language</i> is a standardized modeling language used in software engineering and software architecture.  | 62, 84                 |
| <b>USB</b>       | A <i>Universal Serial Bus</i> is a standard for connectors and communication protocols used for the connection of devices to a computer system.   | 14                     |
| <b>XML</b>       | The <i>Extensible Markup Language</i> is a markup language which is used to structure data machine- and human-readable.   | 72                     |



*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 12. September 2013

---

Marco Johns