Markus Meyer, BSc

Development of a clinical decision support system for type 2 diabetes management used by health care professionals in domiciliary nursing care

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's Degree Programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Alexander Felfernig
Institute of Software Technology (IST)

Second Supervisor

Dipl.-Ing. Dr.techn. Klaus Donsa
JOANNEUM RESEARCH Forschungsgesellschaft mbH

Graz, December 2018

“Measuring programming progress by lines of code is like measuring aircraft building progress by weight.”

[Bill Gates]

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Date/Datum

Signature/Unterschrift

“I’ve finally learned what ’upward compatible’ means.
It means we get to keep all our old mistakes.”
[Dennie van Tassel]

Abstract

According to recent estimates, diabetes mellitus affects more than 425 million people in the world with continuing tendency to rise. To deal with the vast amount of affected people more and more technological aids are used which assist the physicians, nursing staff, and patients in the treatment of diabetes mellitus. Clinical Decision Support Systems (CDSSs) can help to find a personalized medication dosage for each patient by analyzing blood glucose values and medication administrations of the past few days.

In this work, we developed a tablet-based CDSS for the management of type 2 diabetes mellitus patients which is used by Health Care Professionals (HCPs) of domiciliary nursing care. As a team of software engineers, we developed Software as a Medical Device (SaMD) under a Quality Management System (QMS). The developed system – named GlucoTab@MobileCare – provides a task overview for each patient and can suggest a personalized medication dosage that should be administered to the patient. CDSSs on mobile devices allow HCPs to visit patients and manage their diabetes therapy directly at the point of care.

To enhance and simplify the usability of the system, a passwordless authentication method with Near Field Communication (NFC) tags based on the Universal Authentication Framework (UAF) was investigated. Moreover, an automated glucose measurement transmission protocol was developed to automatically transmit measurement values from a glucose meter to the CDSS and therefore eliminate the risk of wrongly entered data into the system. The transmission protocol was developed with the guidance of the Continua Design Guidelines and utilizes the standardized Bluetooth Generic Attribute Profile (GATT).

Overall, a prototype of an innovative CDSS was presented which is capable of calculating and delivering personalized medication dose suggestions and enables a seamless blood glucose measurement integration.

Kurzfassung

Aktuellen Abschätzungen zu Folge betrifft Diabetes Mellitus weltweit mehr als 425 Millionen Menschen und die Tendenz ist stark steigend. Um die enorme Anzahl an betroffenen Menschen bewältigen zu können werden mehr und mehr technologische Hilfsmittel eingesetzt, die Ärzte, Pflegepersonal und Patienten bei der Diabetestherapie unterstützen. Klinische Entscheidungsunterstützungssysteme (CDSSs) können dabei helfen eine individuelle Dosierung der Medikamente für jeden einzelnen Patienten zu finden, indem die Blutzucker Werte und verabreichten Medikamente der letzten Tage ausgewertet werden.

In dieser Arbeit wurde ein Tablet-basiertes CDSS für die Therapie von Typ 2 Diabetes Mellitus Patienten entwickelt, das vom Gesundheitspersonal (HCPs) der mobilen Hauskrankenpflege verwendet wird. Als ein Team aus Softwareentwicklern haben wir unter Anwendung eines Qualitätsmanagementsystems (QMS) Software als Medizinprodukt (SaMD) entwickelt. Das entwickelte System – namens GlucoTab@ MobileCare – bietet eine Aufgaben Übersicht für jeden Patienten und kann individuelle Dosierungen der Medikamente vorschlagen welche anschließend durch HCPs verabreicht werden können. Indem das CDSS auf einem mobilen Gerät läuft, ist es für HCPs möglich die Patienten Zuhause zu besuchen und ihre Diabetestherapie Patientennahe durchzuführen.

Um die Benutzerfreundlichkeit des Systems zu verbessern wurde eine kennwortfreie Authentisierungsmethode mittels Near Field Communication (NFC) Anhänger und dem Universal Authentication Framework (UAF) untersucht. Darüber hinaus wurde ein Protokoll zur automatisierten Blutzucker Messwerte Übertragung von einem Blutzucker Messgerät zum CDSS entwickelt um das Risiko von falsch übertragenen Werten zu eliminieren. Das Protokoll wurde Anhand der Continua Design Guidelines entwickelt und verwendet das standardisierte Bluetooth Generic Attribute Profile (GATT).

Insgesamt wurde ein Prototyp eines innovativen CDSS vorgestellt, welches in der Lage ist eine personalisierte Medikamentendosis vorzuschlagen und ermöglicht eine automatisierte Einbindung von gemessenen Blutzucker Werten.

Contents

List of Figures

List of Tables

List of Abbreviations

AD	Active Directory
AIDL	Android Interface Definition Language
API	Application Programming Interface
BLE	Bluetooth Low Energy
BR/EDR	Basic Rate/Enhanced Data Rate
CDSS	Clinical Decision Support System
DOM	Document Object Model
DSS	Decision Support System
GATT	Generic Attribute Profile
GP	General Practitioner
HCP	Health Care Professional
HIS	Hospital Information System
HL7	Health Level 7
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
IoT	Internet of Things
IP	Internet Protocol
IPC	Interprocess Communication
ISO	International Organization for Standardization
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
MDR	Medical Device Regulation
NFC	Near Field Communication
OAD	Oral Anti-Diabetic Drug
OS	Operating System

PHD	Personal Health Device
PHD-IF	Personal Health Device Interface
PHG	Personal Health Gateway
PIN	Personal Identification Number
QMS	Quality Management System
RACP	Record Access Control Point
RPC	Remote Procedure Call
SaMD	Software as a Medical Device
SOAP	Simple Object Access Protocol
SOP	Standard Operating Procedure
SOUP	Software of Unknown Provenance
SVN	Subversion
UAF	Universal Authentication Framework
UI	User Interface
USB	Universal Serial Bus
UUID	Universal Unique Identifier
VTD	Virtual Token Descriptor
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

Chapter 1

Introduction

On the one hand, computers outperform human capabilities of processing huge amounts of data without tiring or making computation errors. On the other hand, humans exceed a computer's capability of dealing with social, ethnic, and ethic issues as well as interpreting audio-visual data. There are many fields of work where a *combined* system of computer and human capabilities is used to benefit from the advantages of both areas. Medicine is an example for a field that greatly benefits from such systems [44].

GlucoTab® is an example of a tablet-based system that assists physicians or the nursing staff in the treatment of diabetes mellitus patients by analyzing numerous blood glucose measurements and suggesting a personalized medication dosage for each patient [45, 59]. Applications like GlucoTab® help to provide better health care for patients. However, these applications need to be reliable and secure since human lives are affected if something goes wrong. Therefore, they need to comply with several regulations and requirements as specified by the Medical Device Regulation (MDR) and therein referenced documents [24, 25, 26, 28, 29, 52].

1.1 Motivation

Diabetes mellitus affects more than 425 million people. Diabetes mellitus overcomes country borders, ethnic groups, and social classes and threatens millions of lives each year, leaving a reduced life expectancy, disabled people, fatal expenditures for vulnerable households, and a drained health care budget behind. Diabetes mellitus is a global issue that overwhelms health care systems [27].

However, the number of affected people is rising and predicted to reach 629 million in 2045 if nothing is done. Therefore, on the one hand, the International Diabetes Federation aims to raise awareness, education, and *prevention* of diabetes mellitus through training, mentoring, technical support, clinical leadership, policy, and protocols [27]. On the other hand, technological support systems on mobile devices are already used to assist physicians with the *management* of diabetes mellitus patients in hospitals [45] as well as in primary health care facilities [1, 3].

1.2 Goals

Primary goal of this work is to develop a mobile decision support system that is used by Health Care Professionals (HCPs) for the management of diabetes mellitus in the home care setting. More precisely, an existing decision support system for *hospitalized* patients with type 2 diabetes mellitus shall be further developed to satisfy the demands of the *domiciliary nursing care*.

Furthermore, research on innovative concepts to simplify and enhance the handling of this mobile decision support system shall be conducted. Therefore, a passwordless authentication as well as an automated glucose measurement transmission from glucose meters to mobile devices are investigated and prototypes shall be developed.

1.3 Executive Summary

As a result, a mobile decision support system for the treatment of patients with type 2 diabetes mellitus which is used by HCPs in the domiciliary nursing care was developed. The system assists HCPs on their daily routine of visiting patients at their homes to measure blood glucose levels, administer an appropriate amount of insulin, constantly adjust medication dosages, and manage other tasks related to diabetes mellitus.

Furthermore, a passwordless authentication method using Near Field Communication (NFC) tags instead of user name and password credentials was investigated and an implementation concept was developed. Due to unsatisfying results in preliminary tests with the NFC tag, the concept was not fully implemented. However, future work could revisit this concept and exchange the NFC tag with a Bluetooth device in order to mitigate the distance constraint as well as Android's discontinued support of trusted NFC devices.

Moreover, an automated glucose measurement transmission from a glucose meter to the mobile decision support system was investigated based on the Continua Design Guidelines and using standardized Bluetooth communication capabilities. Accordingly, a concept of an automated glucose measurement transmission protocol with an Android middleware was developed and a proof-of-concept prototype was implemented. Additionally, the protocol concept was documented in a scientific paper, presented at the eHealth 2018 conference in Vienna, and published by IOS Press [43].

Finally, usability tests of the developed system were conducted with HCPs which are experts in the field of diabetes mellitus. The tests provided insight on usability flaws whereupon suggestions for future work were inferred.

In conclusion, a glucose measurement transmission protocol was developed which eliminates the risk of miscopied or wrongly entered measurement data into a mobile decision support system. The decision support system is able to calculate a personalized medication dose which can then be used by HCPs in the domiciliary nursing care of type 2 diabetics.

1.4 Structure of this Document

The following three chapters summarize general background knowledge which should help the reader to understand all terminologies that are used throughout this work. Chapter 2 deals with diabetes mellitus, chapter 3 explains Clinical Decision Support Systems (CDSSs), and chapter 4 illustrates the development of Software as a Medical Device (SaMD).

Afterwards, chapter 5 explains the medical device GlucoTab® and discusses its further development towards the mobile home care setting. Three major concepts of GlucoTab@MobileCare (delegation, task management, and long-term management) are discussed in detail. Subsequently, chapter 6 presents a passwordless authentication workflow with NFC tags. Chapter 7 introduces an automated glucose measurement transmission protocol and a proof-of-concept implementation with GlucoTab@MobileCare.

Finally, a usability evaluation of the implemented software is summarized in chapter 8. Chapter 9 provides an outlook and suggests future work. A conclusion and summary is given in chapter 10.

Chapter 2

Diabetes Mellitus

The International Diabetes Federation [27] published the eighth edition of the *IDF Diabetes Atlas* in 2017, stating that diabetes mellitus currently affects more than 425 million people aged between 20 and 79 (see Figure 2.1). This corresponds to 1 in 11 adults suffering from diabetes mellitus and is expected to become 1 in 10 adults in the year 2045. Furthermore mentioned in the *IDF Diabetes Atlas*, more than 90% of all diabetics suffer from type 2 diabetes mellitus.

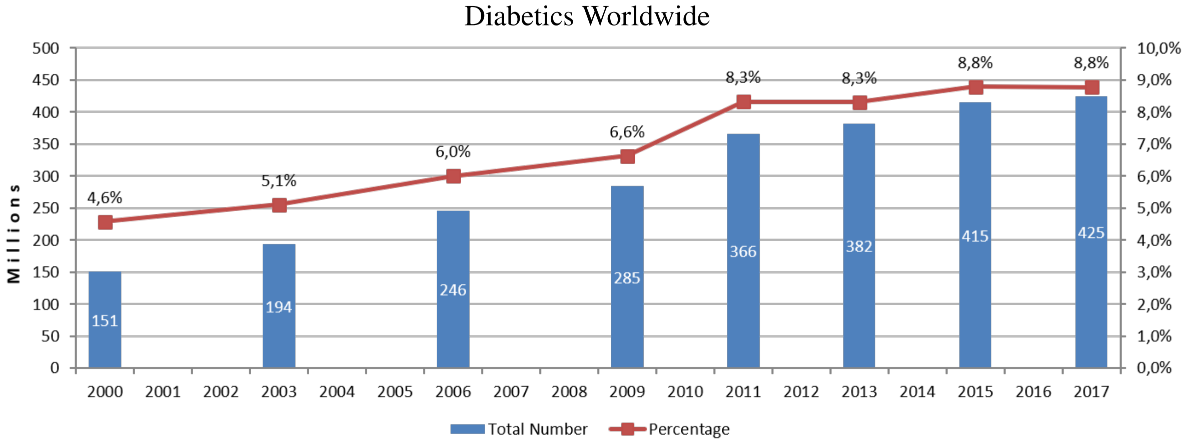


Figure 2.1: Adults (aged between 20 and 79) with diabetes mellitus.
[Created from the original data of the *IDF Diabetes Atlas* [27] with Excel]

Diabetes mellitus [8, 27, 51] is a medical condition leading to acute symptoms of excessive thirst, a dry mouth, frequent urination, lack of energy, extreme tiredness, or blurred vision. Source of these symptoms is an increased blood glucose concentration (hyperglycemia). There are different types of diabetes mellitus, depending on the reason of hyperglycemia to occur. The human body uses two hormones to regulate blood glucose concentration. Insulin is responsible for decreasing the blood glucose and glucagon is responsible for increasing the blood glucose. Both hormones are produced in the pancreas. In the case of *type 1 diabetes mellitus* an autoimmune reaction destroys the insulin-producing beta cells of the pancreas. Consequently, a lack of insulin emerges and leads to hyperglycemia which in turn leads to previously mentioned symptoms. People with *type 2 diabetes mellitus* on the other hand developed either an insulin resistance or a disorder of the insulin secretion. At some point the pancreas may not be able to produce sufficient insulin which again leads to previously mentioned symptoms.

Although Matthew Dobson already discovered the presence of sugar in the urine of diabetics in *1776* which can be used to identify hyperglycemia, the first self-monitoring product for blood glucose was created not earlier than *1965* by Ernest C. Adams. A paradigm shift occurred when glucose meters for physicians and the self-monitoring by patients became available. Measurements of the blood glucose level is meanwhile an essential parameter for the therapy of diabetes mellitus. However, blood glucose levels alternate rapidly throughout the day. Therefore, the HbA_{1C} is a second important parameter that reflects the long-term blood glucose level over a period of three to four months. A recommended HbA_{1C} target during type 2 diabetes therapy is a value below 7.0% or the equivalent of 53 $\frac{mmol}{mol}$ [8, 13, 51].

Poretsky [51] summarizes the findings of several studies on glycemic goals in the *Principles of Diabetes Mellitus* with a dedicated chapter. Conclusion of this chapter is the recommendation of a preprandial blood glucose level between 70 $\frac{mg}{dL}$ and 120 $\frac{mg}{dL}$ as well as a postprandial blood glucose concentration below 180 $\frac{mg}{dL}$. Preprandial blood glucose is measured *before* a meal and postprandial blood glucose is measured 1-2 hours *after* a meal. Achieving these recommended glucose levels has shown a HbA_{1C} reduction and a reduced risk for secondary diseases. However, practical target ranges may vary depending on the overall health status of the patient and for example target higher values with geriatric patients [45].

2.1 Treatment of Diabetes Mellitus Type 2

Diverse treatment options are applicable [8, 10, 13] in order to maintain blood glucose levels in a targeted range. First of all, lifestyle measures are taken (medical nutrition therapy, increased physical activity, weight reduction, and smoking cessation). If a change of lifestyle is not sufficient to achieve the glycemic goals, then Oral Anti-Diabetic Drugs (OADs) are administered. OADs commonly increase the insulin secretion of the pancreas or they increase the patient's insulin sensitivity and therefore result in lower blood glucose levels.

Finally, if glycemic goals are still not reached with lifestyle measures and OADs, then an insulin treatment is started in addition to the treatment with OADs. Insulin treatment is typically started with a basal (long-acting) insulin injection once a day and may be increased with an additional bolus (rapid/short-acting) insulin injection at meal times. An insulin treatment with one or more daily basal insulin injections as well as bolus insulin injections for meals and corrections of high blood glucose concentrations is called basal/bolus therapy. Alternatively, the basal/bolus insulin therapy may be exchanged with multiple injections of premixed short- and long-acting insulin [8, 10, 13].

2.2 Technological Therapy Aids

The first tool to measure blood glucose levels was a measurement strip and a lookup table that maps colors to glucose concentration values. The measurement strip slowly changed color when a drop of blood was placed on it and this color could then be compared to the lookup table to estimate glucose levels. Glucose meters nowadays still use a test strip that contains chemicals which react to the blood. However, the size of the device, the needed amount of blood, and the time to perform a measurement has reduced significantly whereas the measurement accuracy has improved [51].

Insulin pumps are used in the therapy of type 1 diabetes mellitus patients, which allows a more precise insulin supply with a continuous short-acting insulin administration throughout the day compared to only a few long-acting insulin injections. However, costs of insulin pumps are high and the use of pumps for type 2 diabetics in general is still debated [10]. Instead, systems to monitor and analyze long-term management of patients are used to assist the type 2 diabetes therapy.

Sim et al. [58] developed a diabetes dashboard that summarizes relevant laboratory results and reminds the user when repeated measurements need to be done. GlucoTab® [45, 59] is a mobile computerized decision support system that helps to prevent hypoglycemia and improves the glycemic management of hospitalized type 2 diabetes patients.

Other systems are developed to detect diseases or improve blood glucose control early on and try to prevent hospitalization [1, 3]. Smartphone applications[1] were developed to document and analyze blood glucose trends. Different applications[2] are embedded into the primary health care and allow a physician to remotely provide and adjust a personalized treatment plan.

[1] https://mysugr.com/

[2] http://insulia.com/

“If Java had true garbage collection,
most programs would delete themselves upon execution.”
[Robert Sewell]

Chapter 3

Clinical Decision Support System (CDSS)

Physicians need to consider huge data sets of possible diseases and treatments in the process of diagnosis and therapy. Therefore people started to incorporate computers into the decision making process early on. CDSSs assist the physician or patient by presenting patient-specific information in a well-structured format at relevant times to improve health care decisions and outcomes. Apart from showing filtered data lists, CDSSs might furthermore assist with alerts or reminders for specific events or situations. Furthermore, non-experts are able to perform best practice therapies with the support of CDSSs [44].

There are various aspects of CDSSs that can be used to classify them. One aspect that varies for different types of CDSSs is the time of support (before, during, or after a clinical decision is made). Another distinction is whether the support is active (alerts or reminders) or passive (suggestion or calculation as a result of a user input). However, a common classification of CDSSs is the differentiation between *knowledge-based* versus *non-knowledge-based* CDSSs [4].

3.1 Knowledge-based CDSS

An essential component of knowledge-based CDSSs is a *knowledge base*. The knowledge base contains the medical information that is required to infer decision support. Knowledge can be represented *declarative* (hard facts), *procedural* (actions or conclusions that can be derived), *graphical* (graph or network-based system), and *structured* (categorized knowledge). A second key component of knowledge-based CDSSs is the *reasoning engine*. Knowledge from the knowledge base is combined with user input and/or patient information and used in the reasoning engine to output decision suggestions (see Figure 3.1). The

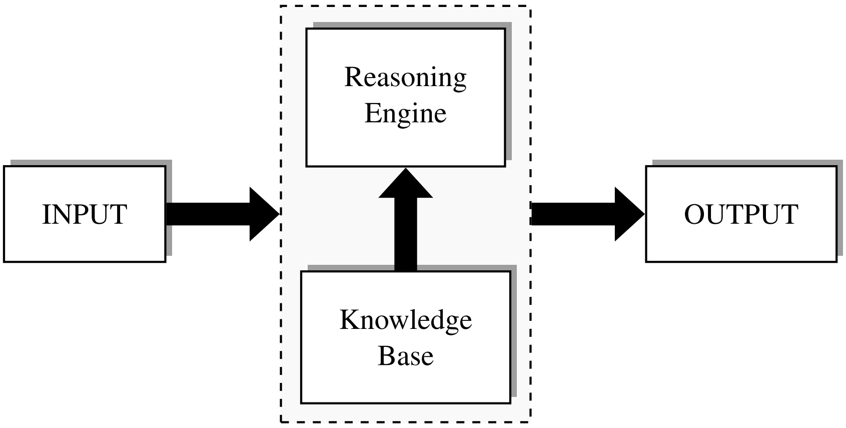


Figure 3.1: Components of a general knowledge-based CDSS [2].
[Redrawn from *Healthcare Data Analytics* [2]]

approach to derive these decisions can for instance be a rule-based system with complex combinations of if-then-else statements or a probabilistic model that estimates the likelihood of various outcomes using Bayes' theorem and conditional probability. A third important component of a knowledge-based CDSS is the *user communication method*. Input for CDSSs is in most cases provided by the user and output of a knowledge-based CDSS can for example be a list of diseases ranked by their probability [2, 4].

3.2 Non-knowledge-based CDSS

In contrast to knowledge-based CDSSs, a non-knowledge-based CDSS does not use a predefined knowledge base. Instead, the required knowledge is acquired with *machine learning*. This is a form of artificial intelligence that allows a computer system to *learn* from a large set of data samples or previous computations. A popular machine learning approach used for CDSSs is the artificial neural network [2].

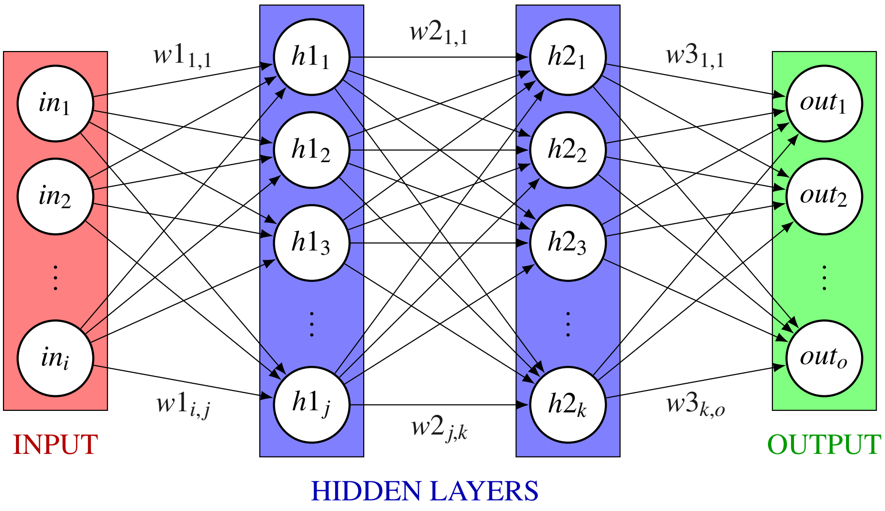


Figure 3.2: Components of an artificial neural network.

Artificial neural networks (see Figure 3.2) are inspired by nature and simulate a biological neural network. Neurons and their connections with synapses are modeled by a graph of nodes and weighted connections. The input nodes (in_1 to in_i) receive patient related data or symptoms and the output nodes (out_1 to out_o) represent a decision or diagnosis that is presented to the physician. All hidden layers and their nodes (hX_y) are further intermediate connections which allow a complex weight distribution between the input and output. All weights ($wX_{y,z}$) are learned by iterative *training* of the neural network with a large set of training data where the desired output for a given input vector is known [4].

The main advantage of such a non-knowledge-based CDSS over a knowledge-based CDSS is that no rules or expert knowledge needs to be manually written into the system. However, one disadvantage of CDSSs based on an artificial neural network is the long training time that is needed until accurate weights are obtained. Furthermore, the medical diagnosis or generally the output is presented by the network *without* any justification or explanation. Therefore, the reliability of those decisions is a problem when dealing with medical devices and the security of a patient's life [2].

Chapter 4

Software as a Medical Device (SaMD)

Standalone software that is used for medical diagnosis or to give decision support during the treatment of patients is categorized as a *medical device* [52]. Several directives and regulations regarding the development of SaMD exist in order to ensure the safety of patients during the use of such medical devices. The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) published standards which help developers to create software that is certified as a medical device. There are requirements to the usability [25], safety [26], documentation [24, 52], and life cycle processes [24] of software as well as requirements to the quality management system [28] and risk management [29] regarding medical devices.

4.1 Medical Device Regulation (MDR)

Among other things the Medical Device Regulation (MDR) [52] specifies a classification of medical devices and determines the contents of the technical documentation. The first classification step is the distinction between *a medical device* and ***not*** *a medical device*. According to the MDR, standalone software is classified as *a medical device* and needs to comply with the MDR if the intended use meets one or more of the following purposes regarding a disease, injury, or disability:

- alleviation
- compensation
- diagnosis
- monitoring
- prediction
- prevention
- prognosis
- treatment

The second classification step splits medical devices into three classes (I, II, and III) depending on the grade of damage it can deal to a patient. A higher classification of a medical device results in more extensive verification requirements and may require the inclusion of an independent notified body and a certified Quality Management System (QMS). However, the old regulations that were replaced by the MDR in 2017 are still applicable until May 2020 [52] and contain a slightly different classification rule. Therefore many SaMD products can still be brought to market as class I even though the new MDR would classify them as class II or III.

Further requirements defined in the MDR are a general description of the medical device, an intended purpose and a definition of the intended users, and an instruction for use. Documents to satisfy these requirements are incorporated into the technical documentation of the medical device. Moreover the MDR specifies requirements to establish and maintain a risk management system, a post-market surveillance system, and a QMS.

4.2 Quality Management System (QMS)

The ISO 13485 [28] is a standard for medical device manufacturing and specifies the requirements of QMSs. Apart from definitions and references to other standards, the ISO 13485 describes an approach of linked *processes* for the management of resources, responsibilities, and quality. Within the scope of QMSs, a process is an activity that takes inputs and creates outputs. Inputs might for example be requested features or sub components of the medical device and according outputs could be a design mockup or assembled components of the medical device.

Individual chapters of the ISO 13485 [28] focus on different topics for processes that should be established. Therefore, the resulting QMS might look like a collection of according Standard Operating Procedures (SOPs). The first topic to mention is a chapter addressing the review, analysis, and improvement of all established SOPs and the QMS effectiveness itself. Secondly, acquisition and maintenance of necessary tools, equipment, and facilities for the manufacturing of the medical device is described. Thirdly, the ISO 13485 demands to establish processes for the product realization. This includes the processes of retrieving customer requirements, planning and designing the medical device, actually developing and making the product, and finally installing the medical device for the customer. Furthermore, throughout product realization as well as all other QMS processes, risk management must be applied as defined in the referenced ISO 14971 standard.

4.3 Application of Risk Management to Medical Devices

Risk management is defined by ISO 14971 [29] with a process to *identify* hazards as well as to *estimate*, *evaluate*, and *control* associated risks. First, a risk management plan is created and criteria for risk acceptability are defined. Next, known and foreseeable hazards are *identified* and documented. For each identified hazard the probability of occurrence and severity of harm is *estimated*. Risks can then be *evaluated* with a risk evaluation matrix, which is a combination of the probability of occurrence and the severity of harm (see example in Table 4.1).

		Qualitative severity levels				
		Negligible	Minor	Serious	Critical	Catastrophic
Semi-quantitative probability levels	Frequent	▓	▓	▓	▓	▓
	Probable	▓ R_1	▓ R_2	▓	▓	▓
	Occasional	░	░ R_4	▓	▓ R_5	▓ R_6
	Remote	░	░	░	▓	▓
	Improbable	░	░	░ R_3	░	░

░ acceptable risk ▓ unacceptable risk

Table 4.1: Example of a risk evaluation matrix as illustrated in ISO 14971 [29].
[Redrawn from IEC 14971 Annex D [29]]

The risk evaluation shall help to identify hazards where risk reduction is required and therefore risk *control* measures need to be applied. ISO 14971 [29] specifies three risk control options with a given priority ranking. First, an “inherent safety by design” shall be targeted. The second option uses “protective measures in the medical device itself or the manufacturing process”, and as a third option “information for safety” can be composed and used to reduce risks. One or more risk control measures shall be applied in this order. However, if the residual risk is still unacceptable and no further risk control is practicable, then a risk/benefit analysis may be conducted. This analysis allows manufacturers to fabricate products where the medical benefit outweighs the residual risk.

Finally, all results and performed activities need to be documented in the *risk management file*. An overall residual risk acceptability is evaluated and a risk management report is created after reviewing that the risk management plan has been implemented appropriately.

4.4 Software Life Cycle Processes

During realization of SaMD products the IEC 62304 [24] needs to be followed. This standard deals with five major life cycle processes: software development, software maintenance, risk management, configuration management, and problem resolution.

Additional to the medical device classification given by the MDR, the IEC 62304 defines three software safety classes (A, B, C) determining that no harm, non-serious injury, and serious injury or death can arise from a hazardous situation. The resulting software safety class effects the requirements of the software development process.

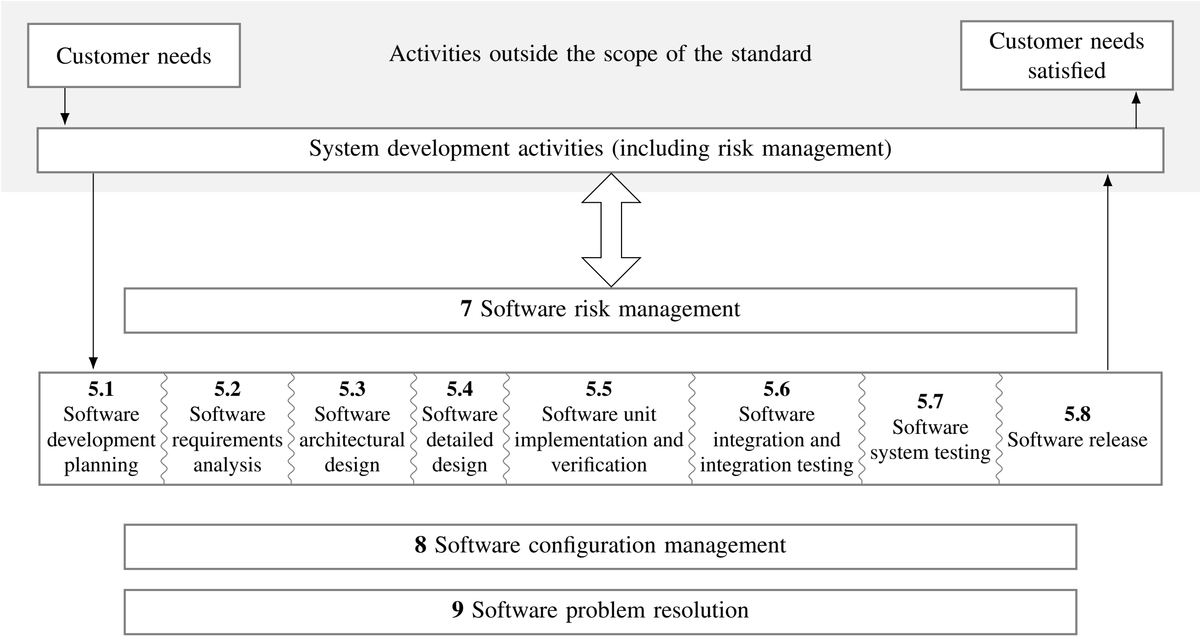

Customer needs
Activities outside the scope of the standard
Customer needs satisfied
System development activities (including risk management)
7 Software risk management
5.1 Software development planning
5.2 Software requirements analysis
5.3 Software architectural design
5.4 Software detailed design
5.5 Software unit implementation and verification
5.6 Software integration and integration testing
5.7 Software system testing
5.8 Software release
8 Software configuration management
9 Software problem resolution

Figure 4.1: Software live cycle processes defined in IEC 62304 [24].
[Redrawn from IEC 62304 [24]]

The software development process is subdivided into eight activities (see Figure 4.1) ranging from software development planning to software release. Software of safety class C requires a *detailed* documentation by performing *all* activities of the development process as well as the risk management, configuration management and the problem resolution. For software with a lower safety class some documentation requirements and activities may be omitted (e.g. 5.4, 5.6, and 5.7 are not required with class A) or reduced (e.g. not all sub tasks in 5.1 are required with class A nor B).

Risk management is applied in parallel throughout the entire software development process. Furthermore, the IEC 62304 [24] amends software-specific risk management requirements to the previously described ISO 14971. Software of Unknown Provenance (SOUP) is any library or external software

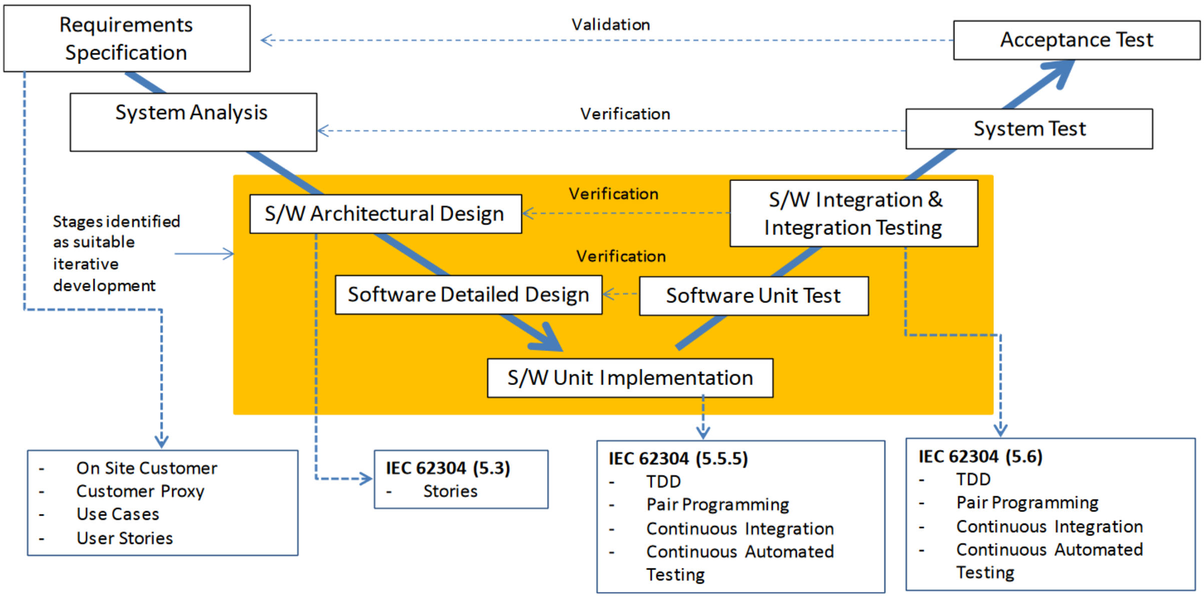


Figure 4.2: AV-Model (Agile V-Model) for Medical Device Software Development [21].
[Image by Hugh et al. [21]]

that is used within a SaMD. Since SOUPs are generally not developed for medical devices, additional evaluation of known issues and regular reviews of the SOUPs' issue tracking lists are necessary.

Commonly[1] used models to perform the activities of the IEC 62304 software development process are the waterfall model, V-model, or agile software development models. The V-model is similar to the waterfall model where tasks are performed in a linear sequence one after another. However, the V-model establishes a relation between development and testing at each level of detail which results in the characteristic V-shape (see Figure 4.2). Furthermore, agile software development practices such as Scrum[2] and extreme programming[3] can be beneficial during medical device development and are therefore incorporated by medical device organizations as well which may lead to a hybrid agile V-model approach [21, 24, 42].

4.5 Application of Usability Engineering to Medical Devices

Another standard that is used during development of SaMD is the IEC 62366 [25]. It defines a *usability engineering process* to evaluate and reduce risks during correct use as well as during use errors of the medical device. All results are documented in the *usability engineering file*.

Primary goal of the IEC 62366-1 is the User Interface (UI) *security*, followed by other goals such as *efficiency* and *user satisfaction* of the UI in IEC 62366-2. The usability engineering process determines a use specification to define a user and patient profile, the use environment, and the main operating functions of the medical device. Based on the use specification, use errors and related hazards can be identified and are subject to ISO 14971 risk management. Subsequently a usability evaluation plan is created with iterative, formative usability evaluations during the UI implementation to identify strengths and weaknesses of the UI as well as new use errors and enhance the UI design. The final UI is then evaluated with a summative usability evaluation to verify that the UI is *safe* to use.

[1] https://www.johner-institute.com/articles/software-iec-62304/software-lifecycle/

[2] `https://www.agilealliance.org/glossary/scrum`

[3] `https://www.agilealliance.org/glossary/xp`

Chapter 5

Further Development of GlucoTab®

This chapter describes the further development of GlucoTab® (a CDSS used in hospitals) to GlucoTab@ MobileCare (a CDSS used in home care). First, general information about GlucoTab® is given and some general modifications of GlucoTab@MobileCare are presented. Afterwards, three major modifications (delegation, task management, long-term management) are described in-depth in three separate sections.

5.1 General Information about GlucoTab®

GlucoTab® is a knowledge-based Clinical Decision Support System (CDSS) used by nurses for inpatient type 2 diabetes treatment. The development of GlucoTab® was started within the scope of European and Austrian research projects by JOANNEUM RESEARCH Forschungsgesellschaft mbH and the Medical University of Graz. Meanwhile, GlucoTab® is marketed by decide Clinical Software GmbH and operated in hospitals on a daily basis.

Assisting features of GlucoTab® include documentation and visualization of blood glucose measurements, nutrition, and medication doses. Furthermore, the system displays open tasks (e.g. blood glucose measurement task or insulin administration task) and provides *recommendations* for personalized insulin doses based on blood glucose measurements. Blood glucose is measured and documented four times a day (three preprandial and one bedtime value). Paper-based algorithms for diabetes treatment were adapted to match the requirements of hospital workflows and afterwards implemented in the mobile CDSS. The algorithm of GlucoTab® frequently adjusts the total daily insulin dose for each patient based on the preprandial glucose measurements of today and yesterday as well as the standardized recommendations of the incorporated paper-based algorithms [39, 45].

A major part of this thesis is the further development of GlucoTab® from the hospital environment to the *mobile care* environment. GlucoTab@MobileCare is therefore operated by HCPs who visit type 2 diabetes patients at their homes to measure blood glucose levels, administer drugs, and overall manage the diabetes treatment. The DiabetesTherapy@Home project made the collaborative development of GlucoTab@MobileCare by JOANNEUM RESEARCH Forschungsgesellschaft mbH, the University of Graz, and the Medical University of Graz possible.

GlucoTab® and GlucoTab@MobileCare are both software developed as a medical device within a QMS and following the MDR. The software is designed as a client-server model. Therefore, two main components are the Android front end and the back end (see Figure 5.1). The front-end software is running on an Android tablet and communicates with the back-end server via Simple Object Access Protocol (SOAP) web services. Communication between client and server is encrypted using Hypertext Transfer Protocol Secure (HTTPS). Additionally all client devices are registered in the back-end database during system setup. As a result, authorization of specific tablet devices can be revoked in case of theft.

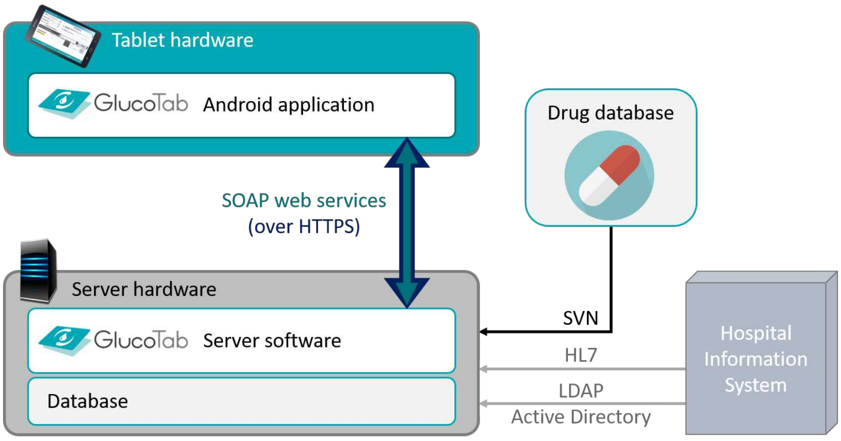
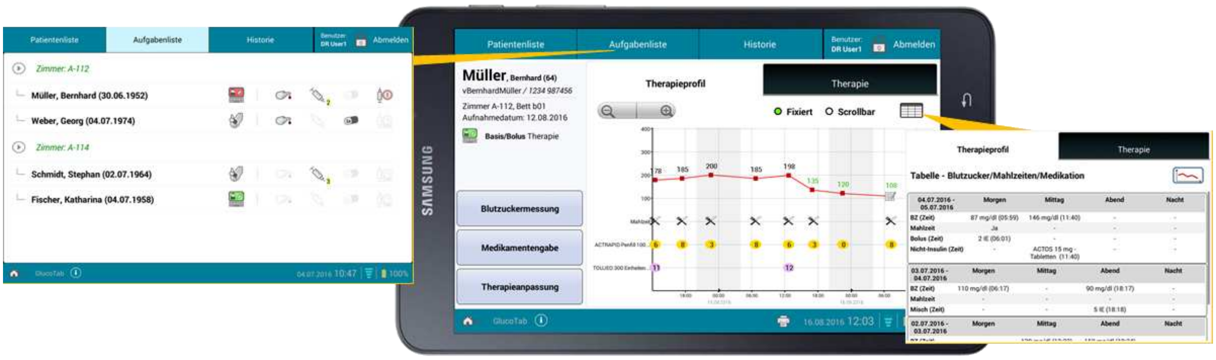


Figure 5.1: High level view of the system components in GlucoTab®.
[Image taken and adapted from the user manual of GlucoTab®]

In addition to a database, the backend implements a mapping to the Health Level 7 (HL7)[1] standard to receive patient data from the Hospital Information System (HIS). Moreover, the Lightweight Directory Access Protocol (LDAP) is utilized to reuse login credentials of the nursing staff which are provided by an Active Directory (AD) of the HIS. Finally, the back end also uses Subversion (SVN) to maintain an up-to-date drug list in the database.

The Android front end provides a structured list of patients to the user. A user can access patient details after authenticating with a valid user name and password. In the patient details view, general patient information such as name, age, and current type of therapy is shown. Furthermore, a chronological history of measured glucose concentrations and medication administrations is presented either as a chart or in a table view (see Figure 5.2).



Zimmer: A-112

Müller, Bernhard (30.06.1952)

Weber, Georg (04.07.1974)

Zimmer: A-114

Schmidt, Stephan (02.07.1964)

Fischer, Katharina (04.07.1958)

Müller, Bernhard (64)
vBernhardMüller / 1234 987456
Zimmer A-112, Bett b01
Aufnahmedatum: 12.08.2016
Basis/Bolus Therapie

Tabelle - Blutzucker/Mahlzeiten/Medikation

04.07.2016 - 05.07.2016	Morgen	Mittag	Abend	Nacht
BZ (Zeit)	87 mg/dl (05:59)	146 mg/dl (11:40)	-	-
Mahlzeit	Ja	-	-	-
Bolus (Zeit)	2 IE (06:01)	-	-	-
Nicht-Insulin (Zeit)	-	ACTOS 15 mg - Tabletten (11:40)	-	-

03.07.2016 - 04.07.2016	Morgen	Mittag	Abend	Nacht
BZ (Zeit)	110 mg/dl (06:17)	-	90 mg/dl (18:17)	-
Mahlzeit	-	-	-	-
Misch (Zeit)	-	-	5 IE (18:18)	-

02.07.2016 - 03.07.2016	Morgen	Mittag	Abend	Nacht

Figure 5.2: Task list (left), patient details chart view (middle) and table view (right) in GlucoTab®.
[Image based on figures taken from the user manual of GlucoTab®]

A kiosk mode (Samsung Knox[2]) prevents users from changing system settings or installing and launching unwanted applications. GlucoTab® is the only application users can operate and most hardware keys are disabled. The kiosk mode hides the default system status bar and runs GlucoTab® as full screen application. A custom status bar at the bottom of the screen is therefore used to display the current date and time as well as battery and network status to the user.

[1] https://www.hl7.org

[2] https://www.samsungknox.com

There are three main tasks that can be performed from the patient details view which are represented by the three buttons in the bottom left corner as shown in Figure 5.2. First of all there is the *blood glucose measurement*, which allows the user to submit a blood glucose concentration and a time of measurement with optional comments to the system. Secondly there is the *medication administration*, which follows right after the glucose measurement in an ordinary workflow of diabetes treatment. As a result, the medication dose can be fine-tuned according to the previously measured glucose level. Finally there is the *therapy adjustment*, where settings of the current therapy can be changed. Therapy adjustment tasks are automatically generated by the system whenever the underlying algorithm has detected a better treatment option for the patient.

GlucoTab@MobileCare was further developed based on the released version 5.0 of GlucoTab®. Due to time constraints and limited resources many parts of GlucoTab® were reused and therefore extended, adjusted or disabled to meet the requirements of GlucoTab@MobileCare. To identify changed user requirements for GlucoTab@MobileCare in the home care setting in comparison to the hospital environment of GlucoTab®, three meetings with HCPs from the Austrian Red Cross were conducted. The first meeting focused on delegation, the second meeting covered task management, and the third meeting discussed long-term management and PDF reports.

5.2 General Enhancements

The first and most essential change in GlucoTab@MobileCare towards a mobile working environment was the switch from a Wireless Local Area Network (WLAN) to a cellular network while communicating to the server. However, the Internet Protocol (IP), which is used by GlucoTab®, can be used with WLAN as well as with cellular networks. The SOAP communication is encrypted by using HTTPS. Therefore no changes to the source code were needed to allow a secure communication with cellular network data.

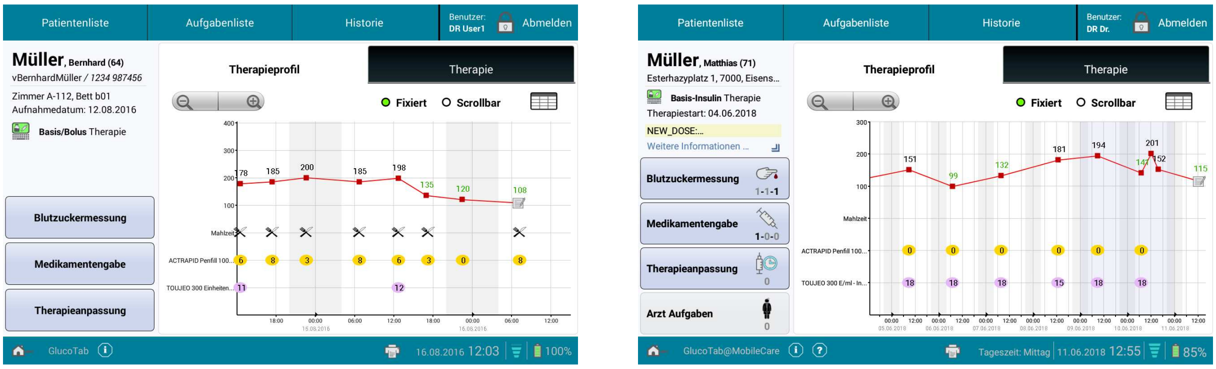
Android's system services [15] were used to display signal strength and connectivity changes of the cellular network in the status bar. On the one hand, Android's `ConnectivityManager` is used to receive broadcasts of a changed network connectivity or connectivity errors. On the other hand, Android's `TelephonyManager` is used to listen to signal strength changes. The signal strength is then presented in the status bar with 0 (▽) to 4 (▼) bars.

Secondly, the login dialog was extended with a *password recovery* label. The database schema was therefore extended with an additional token in the user table (could contain a random token which is sent to the user via mail or any other secret that is well known by the user). During password recovery the user is simply prompted for this token, the user name and a new password.

The next change to note is the details view of a patient. In GlucoTab® patients were organized by their location in a specific room and a specific bed. The view with a list of all patients offered a filter to show only patients in selected rooms. GlucoTab@MobileCare replaced this location information with the actual address of the patient. Furthermore, patients are assigned to a specific tour, by which patients can now be filtered in the list of all patients. HCPs can consequently set a filter to their tour in order to see all patients assigned to their work shift. As a matter of course the room and bed information was also replaced by address and tour information in the patient details view.

The three main buttons in the patient details view were extended by showing numbers according to the amount of tasks that need to be performed (see Figure 5.3). Additionally a fourth button for physician related tasks was added. Finally, the time axis of the chart view was adjusted to show a default time period of 7 days instead of 48 hours, since less events are expected to be documented in the domiciliary care.

Furthermore, concepts to increase usability and further support the HCP work flow were investigated. Since an erroneous documented blood glucose measurement severely increases the risk of hypoglycemia [11], one concept focused on an automated glucose measurement transmission protocol. The measured



(a) Patient detail view in GlucoTab® with 3 main action buttons and a chart history of 48 hours.

(b) Patient detail view in GlucoTab@ MobileCare with 4 detailed action buttons and a chart history of 7 days.

Figure 5.3: Comparison of patient detail views in (a) GlucoTab® and (b) GlucoTab@MobileCare. [Both images were taken as screen shots of the respective software]

blood glucose concentration is transmitted to the Android device immediately after measurement and thereby avoids the risk of erroneously copied values. However, this experimental concept was not developed within the SaMD and MDR requirements. Therefore it is not part of the medical device GlucoTab@ MobileCare and is separately discussed in chapter 7.

Another concept dealt with a simplified login mechanism using an NFC tag. Typing a user name and complex password takes time and may be impractical for HCPs during patient treatment. Therefore a login process with a personalized NFC tag was considered and results are separately summarized in chapter 6, since it is likewise not part of the medical device.

5.3 Delegation

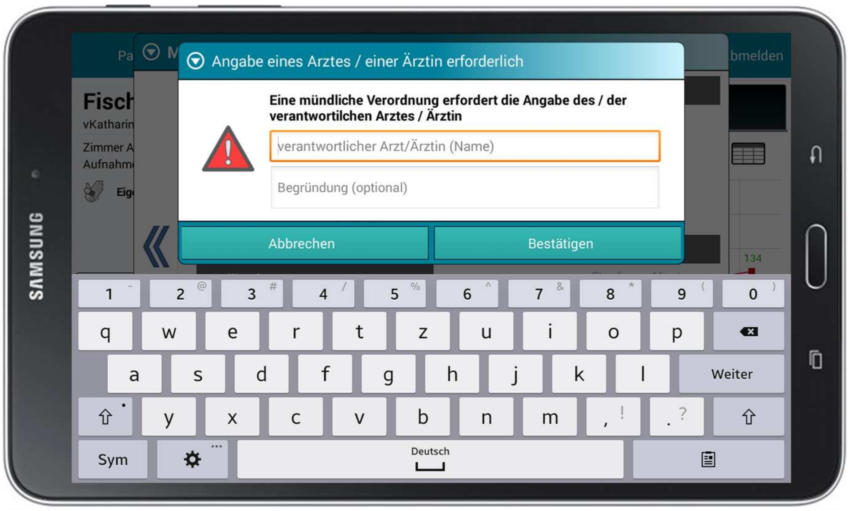
In the domiciliary care it is not feasible for the physician in charge to visit all his patients in person. Therefore HCPs take care of the regular visits and consult physicians only if necessary. However, the system must provide a way to document if a physician is consulted and a treatment is performed by the HCP with the approval of which physician.

5.3.1 Initial State

The initial version of GlucoTab® allowed HCPs to carry out one specific *delegated order*. When logged in as a nurse, the button for therapy adjustments changed to a delegated order button. By pressing this button, a single medication administration order could be added. Before finishing the delegated order, the user was asked to enter the ordering physician's name into a text box along with an optional explanatory statement (see Figure 5.4). Nevertheless it was **not** possible to adjust any therapy settings or interval prescriptions of medication as a HCP.

5.3.2 Planned Design

In the first Austrian Red Cross meeting the changed working environment from hospitals in GlucoTab® to patients' homes in GlucoTab@MobileCare was discussed. Since physicians are not nearby in the



Angabe eines Arztes / einer Ärztin erforderlich

Eine mündliche Verordnung erfordert die Angabe des / der verantwortlichen Arztes / Ärztin

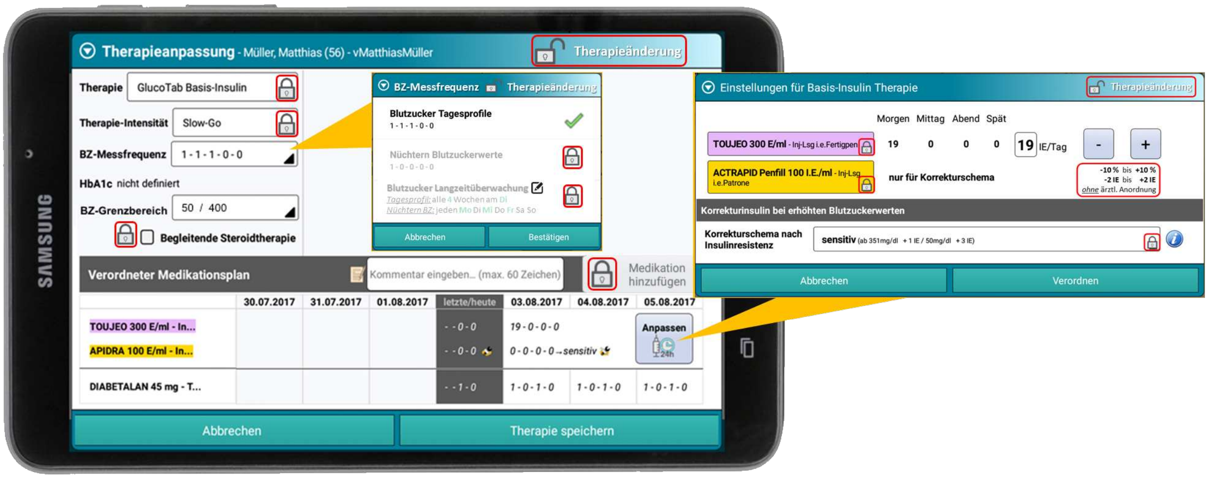
verantwortlicher Arzt/Ärztin (Name)

Begründung (optional)

Abbrechen | Bestätigen

Figure 5.4: Physician selection in GlucoTab® for delegated orders.
[Image taken from the user manual of GlucoTab®]

home care setting, HCPs should be able to adjust medication doses of interval prescriptions and change therapy settings on behalf of a physician. However, such delegated actions should only be possible after stating the name of the ordering physician who agreed on that action.



Therapieanpassung - Müller, Matthias (56) - vMatthiasMüller | Therapieänderung

Therapie: GlucoTab Basis-Insulin

Therapie-Intensität: Slow-Go

BZ-Messfrequenz: 1 - 1 - 1 - 0 - 0

HbA1c nicht definiert

BZ-Grenzbereich: 50 / 400

□ Begleitende Steroidtherapie

BZ-Messfrequenz | Therapieänderung

Blutzucker Tagesprofile 1 - 1 - 1 - 0 - 0 ✓

Nüchtern Blutzuckerwerte 1 - 0 - 0 - 0 - 0

Blutzucker Langzeitüberwachung
Tagesprofil: alle 4 Wochen am Di
Nüchtern BZ: jeden Mo Di Mi Do Fr Sa So

Abbrechen | Bestätigen

Verordneter Medikationsplan | Kommentar eingeben... (max. 60 Zeichen) | Medikation hinzufügen

	30.07.2017	31.07.2017	01.08.2017	letzte/heute	03.08.2017	04.08.2017	05.08.2017
TOUJEO 300 E/ml - In...				- - 0 - 0	19 - 0 - 0 - 0		Anpassen
APIDRA 100 E/ml - In...				- - 0 - 0	0 - 0 - 0 - 0 → sensitiv		
DIABETALAN 45 mg - T...				- - 1 - 0	1 - 0 - 1 - 0	1 - 0 - 1 - 0	1 - 0 - 1 - 0

Abbrechen | Therapie speichern

Einstellungen für Basis-Insulin Therapie | Therapieänderung

	Morgen	Mittag	Abend	Spät		
TOUJEO 300 E/ml - Inj-Lsg i.e.Fertigpen	19	0	0	0	19 IE/Tag	- +
ACTRAPID Penfill 100 I.E./ml - Inj-Lsg i.e.Patrone	nur für Korrekturschema					-10 % bis +10 % -2 IE bis +2 IE *ohne* ärztl. Anordnung

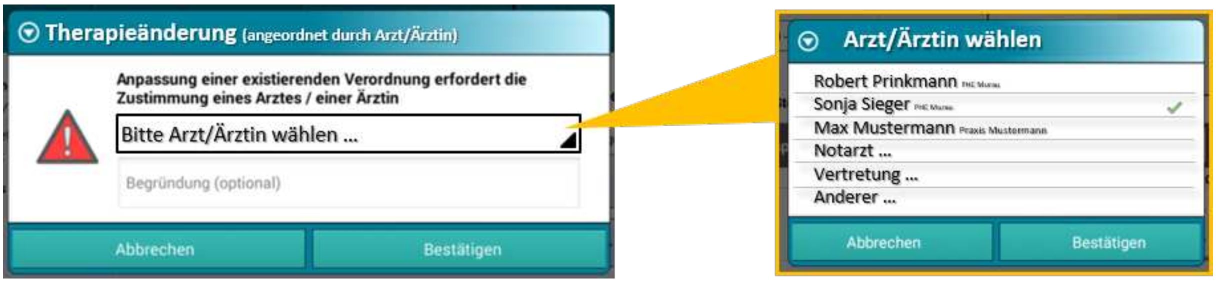
Korrekturinsulin bei erhöhten Blutzuckerwerten

Korrekturschema nach Insulinresistenz: sensitiv (ab 351mg/dl + 1 IE / 50mg/dl + 3 IE)

Abbrechen | Verordnen

Figure 5.5: Mockup of delegated therapy adjustments in GlucoTab@MobileCare.

If an action is only possible after approval of a physician, a closed lock symbol (see red marks in Figure 5.5) is shown. By pressing an unlock button, the user is prompted to select a physician in order to unlock the previously locked actions. The old text box should be replaced by a list of all known physicians and thereby simplify the physician selection for the user. Additionally, a generic *emergency physician*, *substitute physician* and *other physician* can be selected, whereupon the field for an explanatory statement becomes mandatory (see Figure 5.6). Furthermore, HCPs should be allowed to increase or decrease a medication dose by a configurable percentage *without* affirmation of a physician. If, for instance, the system suggests an insulin dose of 19 units and increasing or decreasing by 10% is allowed, then any value between 17 units and 21 units should be possible *without* delegation.



Therapieänderung (angeordnet durch Arzt/Ärztin)

Anpassung einer existierenden Verordnung erfordert die Zustimmung eines Arztes / einer Ärztin

Bitte Arzt/Ärztin wählen ...

Begründung (optional)

Arzt/Ärztin wählen

Robert Prinkmann PHC Murau
Sonja Sieger PHC Murau ✓
Max Mustermann Praxis Mustermann
Notarzt ...
Vertretung ...
Anderer ...

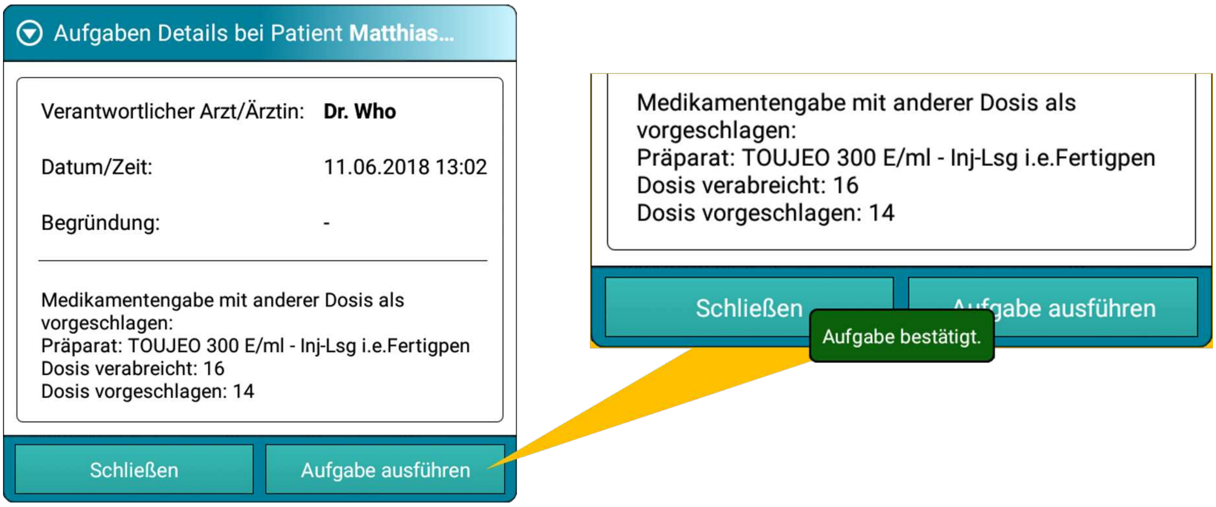
Figure 5.6: Mockup of a physician selection in GlucoTab@MobileCare for delegated orders.

5.3.3 Final Implementation

The actual implementation of delegations strongly suffered from a lack of time and resources. Many reused components of GlucoTab® would have need to be redesigned in order to allow a locked and unlocked state as previously shown with the lock symbols in Figure 5.5.

Therefore, a simplified approach was chosen which merely implemented the physician selection dialog box (see Figure 5.6) instead of the old text field. The Android front end sends the intended action request to the back end, assuming that no physician approval is required. Afterwards the back end performs the actual check whether a physician's approval is needed or not. If the action requires approval, a specific error code is send back to the front end. At the front end, the new dialog box is shown to prompt the user for the ordering physician. After selecting and confirming a physician, the Android front end will send the same action request again, except for an additional physician parameter which allows the back end to identify the request as a delegated order by the specified physician.

As soon as a delegated order is identified and executed, the back end generates a physician task to acknowledge the delegated order. The task details contain a summary of the performed action, a date and time, an optional explanatory statement and the name of the physician who authorized that action. Figure 5.7 shows the physician task of a delegated order where the suggested insulin dose of 14 units was overruled by the HCP with 16 units.



Aufgaben Details bei Patient **Matthias...**

Verantwortlicher Arzt/Ärztin: **Dr. Who**

Datum/Zeit: 11.06.2018 13:02

Begründung: -

Medikamentengabe mit anderer Dosis als vorgeschlagen:
Präparat: TOUJEO 300 E/ml - Inj-Lsg i.e.Fertigpen
Dosis verabreicht: 16
Dosis vorgeschlagen: 14

Medikamentengabe mit anderer Dosis als vorgeschlagen:
Präparat: TOUJEO 300 E/ml - Inj-Lsg i.e.Fertigpen
Dosis verabreicht: 16
Dosis vorgeschlagen: 14

Aufgabe bestätigt.

Figure 5.7: Dialog to show and confirm delegated orders in GlucoTab@MobileCare (left) and a toast message after a successful confirmation (right).

A major downside of the implemented design is the slight and late user feedback. At first glance users might assume that all actions are accessible without further permission. However, as soon as the actual request is send to the server, users may be asked to specify on whose behalf this action is performed. Additionally, the dialog to display and confirm delegated orders was derived from an already existing class which does not provide a method to remove or alter the rendered content. This restriction and the asynchronous nature of a web service request which causes a thread change and a loss of the dialog window handle would have meant numerous code modifications in order to provide a nice user feedback during confirmation.

Once again a simple approach with a toast message (see Figure 5.7) was chosen to report success or failure to the user. Although the chosen design fulfills the minimum requirements that were specified in the meeting with the Austrian Red Cross, the overall design of delegated orders should be enhanced in future releases.

5.4 Task Management

Diabetes treatment is essentially about keeping the blood glucose level in a target range. Blood glucose measurements can tell if the targeted range is achieved. Medication administrations are used to regulate and keep the patient's glucose level in the targeted range as good as possible.

A therapy in GlucoTab® primarily consists of *blood glucose measurement* tasks and *medication administration* tasks. Additionally an algorithm constantly analyses the blood glucose measurement values and creates *therapy adjustment* tasks to ensure that users adjust medication doses as well as the frequency of blood glucose measurements on a regular basis. *Medication administration* tasks are subdivided into interval order tasks and single order tasks. A single order task is only scheduled once at a specific time whereas an interval order task is scheduled repeatedly in a specified interval. In GlucoTab@ MobileCare three new task **activities** were introduced:

- **Blood glucose profile:** a blood glucose profile in GlucoTab@MobileCare is defined as a day with at least one blood glucose measurement in the morning, one measurement at midday and one measurement in the evening.
- **HbA$_{1C}$ measurement:** the HbA$_{1C}$ provides an approximation of the mean blood glucose concentration over the past three months [8]. Therefore GlucoTab@MobileCare reminds the user to check the patient's HbA$_{1C}$ on a regular basis with HbA$_{1C}$ measurement tasks.
- **Therapy check:** a therapy check task is generated to summarize an action or adjustment that has been performed after consulting a physician (delegated order). The task can afterwards be checked and confirmed by a physician as described in the previous section 5.3.

An overview of all available task activities with brief descriptions and a comparison of how they are illustrated in the task lists of GlucoTab® and GlucoTab@MobileCare is given in Table 5.1. Task activities marked as not available (N/A) are either not displayed in the task list or do not exist in the corresponding software.

Besides the distinction of different task *activities*, a task is also assigned a specific *state*. The four basic **states** of tasks in GlucoTab® are:

- **ACTIVE:** an *active* task can and should be executed as soon as possible, if the user has sufficient authorization to execute the task.
- **RESOLVED:** as soon as an *active* task is executed, the state changes to *resolved*.
- **EXPIRED:** if an *active* task is not executed in the specified time frame, the state changes to *expired*. Expired tasks can no longer be executed.

- **NEW:** the state *new* defines tasks that have already been created and are scheduled for the future, but cannot yet be executed.

Task activity	**Description**	**Illustration in**	
		GT	**GTMC**
MEASUREMENT	Blood glucose measurement tasks serve as a reminder for the user to perform a glucose measurement within a specific time frame.	🩸	🩸 1-1-1
MEDICATION_INTERVAL_ORDER	Medication administration tasks to remind the user of a *regularly scheduled* administration with a predefined medication dose.	💉	💉 1-0-0
MEDICATION_SINGLE_ORDER	*One-time* medication administration (common task as a corrective action after a high blood glucose measurement).	1x	N/A
THERAPY_ADJUSTMENT, THERAPY_CONFIRMATION	Tasks to suggest improved therapy settings to the user or to let the user keep and confirm the current therapy settings.	24h	🕒 1
THERAPY_REINITIALISATION	Tasks demanding a reinitialization of a neglected therapy and thereby a reactivation of the decision support system.	!	🕒 1
BG_DAY_PROFILE	Reminder that a blood glucose profile shall be performed for the patient.	N/A	📈 +4
MEASUREMENT_HBA1C	Reminder that a laboratory test of the HbA_{1C} shall be performed for the patient.	N/A	🧍 1
GP_THERAPY_CHECK	Tasks for physicians to check and confirm a delegated order.	N/A	🧍 3

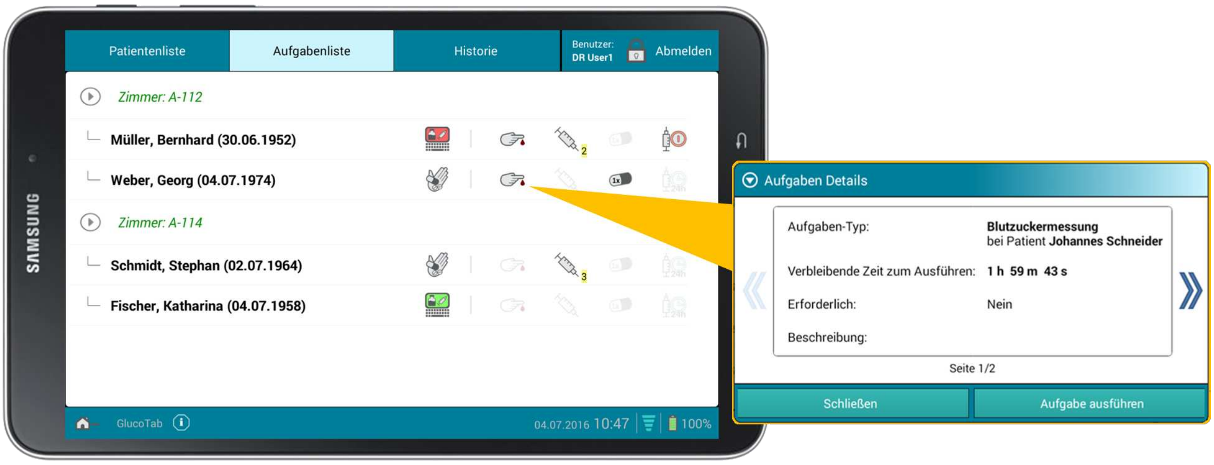
Table 5.1: Table of task activities and a comparison of their respective illustration in the task lists of GlucoTab® (GT) and GlucoTab@MobileCare (GTMC).

5.4.1 Initial State

A fundamental component of GlucoTab® is the task list. It provides a structured list of all enrolled patients and is grouped by the rooms where the patients are accommodated. Patients are represented by rows, which provide the patient's name and date of birth in the leftmost screen area. On the rightmost screen area there is an icon to indicate whether decision support is enabled or not and four further icons to display active tasks (see Figure 5.8). Active tasks are displayed in color, whereas inactive tasks are shown grayed out.

The task list of GlucoTab® displays tasks for glucose measurements (🩸), interval medication orders (💉), single medication orders (1x), and therapy adjustments (24h) and reinitializations (!). A tap gesture on an active icon will bring up a details dialog with a precise task description. Furthermore, a shortcut button to *execute* the task is shown at the bottom of the dialog, provided that the user has adequate permissions to perform the action.

GlucoTab® is designed for *inpatient* diabetes treatment. The nursing staff is therefore steadily engaged with the patient's diabetes therapy. Accordingly, the task list in GlucoTab® is designed to show



Zimmer: A-112

Müller, Bernhard (30.06.1952)

Weber, Georg (04.07.1974)

Zimmer: A-114

Schmidt, Stephan (02.07.1964)

Fischer, Katharina (04.07.1958)

Aufgaben Details

Aufgaben-Typ:	Blutzuckermessung bei Patient **Johannes Schneider**
Verbleibende Zeit zum Ausführen:	**1 h 59 m 43 s**
Erforderlich:	Nein
Beschreibung:	

Seite 1/2

Figure 5.8: Task list (left) and task details dialog (right) in GlucoTab®.
[Image based on images from the user manual of GlucoTab®]

only tasks that are *currently active* and thereby inform the nursing staff about their shortly upcoming duties.

Data that is displayed in the task list is retrieved from the back end via a SOAP web service request. To be entirely accurate, the `findPatients` web service method is called, which is originally designed for the patients list. However, the method is called with the additional `withTasksInResult` parameter set to true. Thereupon the back end answers with a list of all patients from the given ward.

Each eXtensible Markup Language (XML) element in the retrieved list contains basic information about the patient (e.g. first name, last name, date of birth) and another list of *relevant tasks*. These relevant tasks are then highlighted colorful in the task list (see Figure 5.8). At the same time, all task details are parsed and stored in a data structure which allows them to be displayed in the task details dialog. As a result of this eager loading method, the task details are parsed and processed even if they are not displayed at all. The resulting performance impact is negligible in GlucoTab® but will be further discussed in the implemented design of GlucoTab@MobileCare.

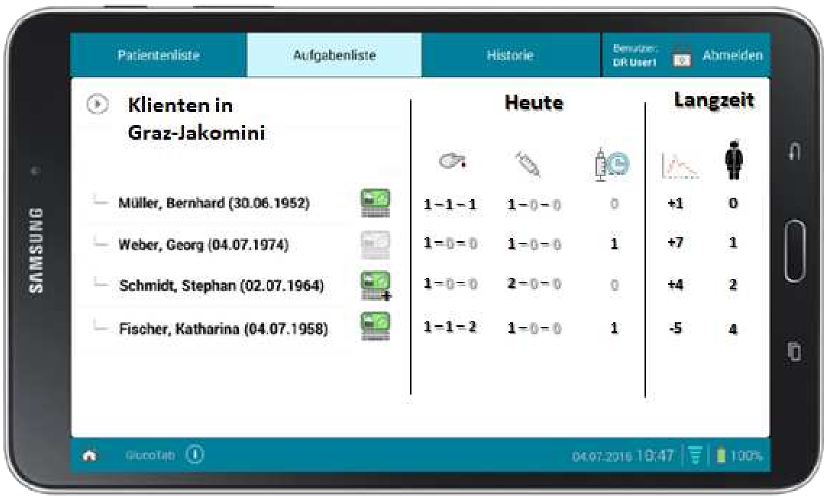
5.4.2 Planned Design

In comparison to the initial state of the task list from GlucoTab®, the new task list of GlucoTab@MobileCare shall display *all* (active, resolved, expired, and new) tasks of the current day. Furthermore, a new section with long-term tasks shall be introduced. Therefore the task list is divided into tasks of *today* and *long-term* tasks (see Figure 5.9). Daily tasks are blood glucose measurements (☞), medication administrations (✎), and therapy adjustments (⚙). Long-term tasks are blood glucose profiles (📈) and HbA$_{1C}$ measurements or physician related tasks (🚹).

Blood glucose measurements and medication administrations are displayed in a group with three digits. Each digit represents one time of the day in which a HCP might visit the patient to execute all necessary tasks (the first digit represents morning, the second represents midday and the third represents evening). The group 2-1-0 for example means that **2** tasks need to be performed in the morning, **1** task is scheduled at midday, and **0** tasks are scheduled for the evening.

The tint of a number (black or gray) provides information about the state of a task. Whenever further actions need to be performed, a solid black color is used. If a task is resolved or expired and therefore cannot be executed anymore, it is shown grayed out.

A therapy reinitialization is basically the same activity as a therapy adjustment, with the difference



Klienten in Graz-Jakomini	Heute			Langzeit	
Müller, Bernhard (30.06.1952)	1-1-1	1-0-0	0	+1	0
Weber, Georg (04.07.1974)	1-0-0	1-0-0	1	+7	1
Schmidt, Stephan (02.07.1964)	1-0-0	2-0-0	0	+4	2
Fischer, Katharina (04.07.1958)	1-1-2	1-0-0	1	-5	4

Figure 5.9: Mockup of the new task list in GlucoTab@MobileCare.

that the decision support is reactivated. Consequently, both task activities can be summarized under the generalized heading of therapy adjustments (). A simple numerical value shows the total number of tasks that are currently active. The same numerical representation is used for physician related tasks () in the long-term section.

Blood glucose profile tasks are not expected to be scheduled more often than *once a week*. Moreover, blood glucose profile tasks differ from other tasks as they are specified with two time frames instead of only one time frame where the task *can and should* be executed. The first time frame specifies a range when the task *can* be executed. The second time frame however specifies when the task is *intended* to be executed. As a result, blood glucose profile tasks are *intended* to be performed at a specific day, but *can* be performed some days before or afterwards and therefore provide more flexibility to the HCPs.

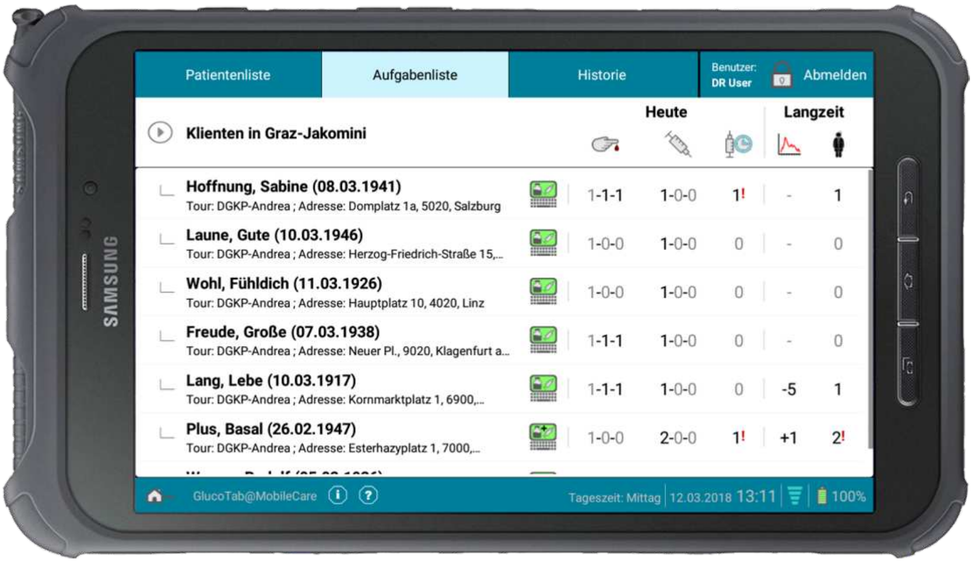
Hence, the number of days until the next blood glucose profile is shown in the task list with a *signed* numerical value. The signed numerical value is calculated by the date difference in days between the current date and the date when the task is *intended* to be executed. Consequently, a positive sign is shown if the intended time of execution is in the future and a negative sign is shown if the intended time of execution has already passed.

Another point to note is that the red and green icons, indicating if decision support is enabled or disabled, are replaced with colored and grayed out icons. This improves usability for users with a red-green color blindness since the difference in color and no color can be recognized. However, to further improve the usability, a color and contrast independent indication could be added [46].

5.4.3 Final Implementation

The task list of GlucoTab@MobileCare was implemented according to the planned design (see Figure 5.10). However, during development another enhancement related to the work flow in a mobile environment was suggested. As a matter of fact, HCPs need to *visit* the patient in order to execute the given tasks. Accordingly, the address of the patient is a relevant information which the HCP needs to read from the system. Therefore a second line of text was added beneath the patient name to display the address as well as the tour to which the patient is assigned.

Nonetheless, the implementation shows minor weaknesses. As pointed out in the initial state, the task data with all details is loaded eagerly with a single web service request. Not only due to the fact that active, resolved, expired, and new tasks are loaded at once, performance losses were noticed by the



Klienten in Graz-Jakomini	Heute			Langzeit	
Hoffnung, Sabine (08.03.1941) Tour: DGKP-Andrea ; Adresse: Domplatz 1a, 5020, Salzburg	1-1-1	1-0-0	1!	-	1
Laune, Gute (10.03.1946) Tour: DGKP-Andrea ; Adresse: Herzog-Friedrich-Straße 15,...	1-0-0	1-0-0	0	-	0
Wohl, Fühldich (11.03.1926) Tour: DGKP-Andrea ; Adresse: Hauptplatz 10, 4020, Linz	1-0-0	1-0-0	0	-	0
Freude, Große (07.03.1938) Tour: DGKP-Andrea ; Adresse: Neuer Pl., 9020, Klagenfurt a...	1-1-1	1-0-0	0	-	0
Lang, Lebe (10.03.1917) Tour: DGKP-Andrea ; Adresse: Kornmarktplatz 1, 6900,...	1-1-1	1-0-0	0	-5	1
Plus, Basal (26.02.1947) Tour: DGKP-Andrea ; Adresse: Esterhazyplatz 1, 7000,...	1-0-0	2-0-0	1!	+1	2!

Figure 5.10: Final task list in GlucoTab@MobileCare.

developer team regarding the time to load and display the task list. Concerning the response time of a system and adequate user feedback, Nielsen [46] concluded in the book *Usability Engineering* that:

> "1.0 second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data." Nielsen [46, page 135]

The actual time to load the task list in GlucoTab@MobileCare depends on several factors, such as the amount of patients in the list, the network latency from client to server, and the performance of the used hardware. However, since the development team clearly *lost the feeling of operating directly on the data* during development, some time measurements were added to the log output. These logs indicated that the time to load the task list frequently exceeded one second. Therefore, an empty list with the message "*Loading tasks ...*" is displayed to the user until the actual tasks are presented.

Yet, the cause of this response time issue was further investigated. First of all, the time from the web service request to the response was compared to the initial state. Since more tasks are loaded, a slightly increased latency was expected, but no major performance issues were identified at this stage.

Secondly, the time spent for parsing the XML response and writing the information into an internal data structure was measured and compared. Again, due to a larger data set to process, an increased time consumption was expected. The conversion to the internal data structure of the task list starts by generating an XML Document Object Model (DOM) from the response string. Next, nested for-loops iterate over the DOM to search for relevant XML elements via string matching. Once a relevant element is found, the content is transferred to the internal data structure.

However, there were minor inefficiencies identified in the source code responsible for loading the task list. Due to many consecutive `if`-statements instead of `else-if`-statements (see Figure 5.11), *all* element names were *always* compared during string matching. Moreover, repeated method invocations on the XML element to receive the element's name can be optimized by extracting a variable prior to the `if`-statements, which in fact might already be done by compiler optimization. Similarly, the `String.matches` method from Oracle [48] is implemented with regular expressions and should be replaced with `String.equals` for better performance.

```java
if (element.getLocalName().matches("firstField")) {
    ...
}
if (element.getLocalName().matches("secondField")) {
    ...
}
...
// this if-condition is always checked, even if firstField already matched!
if (element.getLocalName().matches("nthField")) {
    ...
}
```

(a) Initial state: many consecutive `if`-statements and repeated method invocations.

```java
// variable extraction
String name = element.getLocalName();
if (name.equals("firstField")) {
    ...
} else if (name.equals("secondField")) {
    ...
} ...
// else-if-statements are only checked if nothing matched before!
} else if (name.equals("nthField")) {
    ...
}
```

(b) Improved state: `else`-`if`-statements, extracted variable, and `String.equals` comparison.

Figure 5.11: Outline of minor performance issues during XML parsing found in GlucoTab®.

Although these statements do not have a noticeable performance impact on their own, they can sum up to a noteworthy performance impact when executed many times, such as in the nested `for`-loops during DOM tree traversal. As a result, a few small changes in the source code could improve the time consumption for XML parsing and data preparation by a reasonable amount. Nevertheless, according to Deshmukh and Bamnote [9], the performance could be improved even more with a different XML parser, such as Android's `XmlPullParser` or a Virtual Token Descriptor (VTD)-XML parser, compared to the DOM parser that is currently used.

Finally, the time to render and display the task list in the UI was measured. At this stage, a *significant* performance issue was determined. A detailed investigation of the source code revealed that additional computations are performed on the internal data structure to retrieve the final result for a single row in the task list. As pointed out by Manas and Grancini [40], it is the main thread's sole responsibility to maintain the UI in Android. Therefore, additional computations in the task list could be avoided altogether by changing the structure of the SOAP response from the server to a ready-to-display data structure. Another possibility for a performance improvement could be achieved if additional computations are performed asynchronous in a background thread. The main thread could initially display the general patient information along with a loading indication on the rightmost side. As soon as task details are computed, the background thread could then notify the main thread with the ready-to-display data structure.

Furthermore, excessive calls of Android's computationally expensive method `View.findViewById` could be avoided by recycling views with the ViewHolder pattern or Android's `RecyclerView` class, which enforces the ViewHolder pattern [15, 40, 53]. The ViewHolder pattern uses a separate class to hold layout references that were previously allocated by `View.findViewById`, which allows to reassign data to the layout and therefore reuse individual row layouts of a list during scrolling.

5.5 Long-term Management

The third significant enhancement in GlucoTab@MobileCare was the introduction of the *ongoing* state in the algorithm of the Decision Support System (DSS) during long-term management. Therefore, the algorithm of GlucoTab@MobileCare knows three states for the frequency of blood glucose measurements:

1. **Therapy determination** is the initial state of a therapy to determine the needed medication dosage for the patient. The algorithm prescribes *daily* blood glucose profiles in this state. After three consecutive blood glucose profiles are performed, the DSS creates a therapy adjustment task which can either suggest an alteration to the next therapy state or a modification of the medication dosage followed by another iteration in the therapy determination state.

2. **Fasting blood glucose for control** reduces the prescribed blood glucose measurements to one fasting blood glucose measurement each day. After at least three more days with fasting blood glucose measurements, the algorithm decides whether this therapy state is retained, a therapy determination is needed due to numerous glucose measurements outside the targeted range, or an alteration to the ongoing long-term management state is designated.

3. **Ongoing long-term management** is the final, targeted therapy state. In this state, the algorithm prescribes fasting blood glucose measurements only on chosen days of the week in accordance with blood glucose profiles in a chosen interval of one to nine weeks. Regular therapy confirmation tasks are scheduled which may suggest a preceding therapy state in case of unsatisfied therapy goals, such as blood glucose measurements outside the targeted range or significantly high HbA_{1C} results.

Moreover, HCPs using GlucoTab@MobileCare shall feel confident when following the suggestions from the DSS in a like manner as the HCPs feel confident carrying out the instructions of a physician in the hospital environment. Even though GlucoTab® is **not** a replacement for a physician, the suggested insulin doses of the algorithm are adhered by physicians in 97.5% of cases [45]. However, since the internal computations of the algorithm may not always be apparent to the user in the front end, exposing more algorithm related information may further enhance the user's insight and confidence in the DSS.

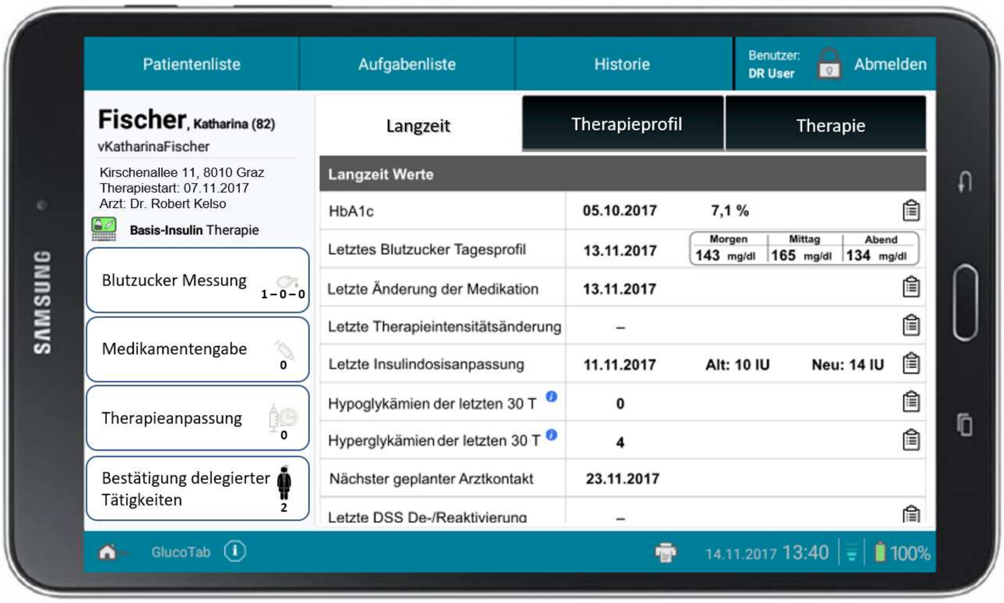
5.5.1 Planned Design

In the third meeting with the Austrian Red Cross, relevant values and numbers for the user in terms of long-term management were worked out. Besides the HbA_{1C}, the lower and upper limit of the targeted blood glucose range as well as the number of hypo- and hyperglycemias during patient treatment were considered as relevant and therefore be presented to the user. In the same way, the last performed blood glucose profile, last medication alteration, last medication dosage adjustment, and last DSS deactivation or reactivation were evaluated as relevant user information.

Some of these values are internally used by the algorithm in order to make decisions. Other values reflect the decisions made by the algorithm and could therefore deliver a deeper insight into the algorithm to the user. As a result, GlucoTab@MobileCare should be modified in order to display all relevant values to the user. On the one hand, the existing therapy tab should be extended with a lower limit that indicates hypoglycemias, an upper limit that indicates hyperglycemias, and the name of the patient's General Practitioner (GP). On the other hand, a new *long term* tab in the patient details view should be introduced (see Figure 5.12) and display the following values:

- dates and results of HbA_{1C} measurements

- date and measurement values of the last blood glucose profile
- dates of medication alterations
- dates of glycemic control modifications
- dates and the changed values of insulin dose adjustments
- number of hypoglycemias from the past 30 days
- number of hyperglycemias from the past 30 days
- date of the next planned GP consultation
- dates of DSS deactivations and reactivations



Fischer, Katharina (82)
vKatharinaFischer
Kirschenallee 11, 8010 Graz
Therapiestart: 07.11.2017
Arzt: Dr. Robert Kelso
Basis-Insulin Therapie

Langzeit Werte

HbA1c	05.10.2017	7,1 %	
Letztes Blutzucker Tagesprofil	13.11.2017	Morgen 143 mg/dl Mittag 165 mg/dl Abend 134 mg/dl	
Letzte Änderung der Medikation	13.11.2017		
Letzte Therapieintensitätsänderung	–		
Letzte Insulindosisanpassung	11.11.2017	Alt: 10 IU	Neu: 14 IU
Hypoglykämien der letzten 30 T	0		
Hyperglykämien der letzten 30 T	4		
Nächster geplanter Arztkontakt	23.11.2017		
Letzte DSS De-/Reaktivierung	–		

Figure 5.12: Mockup of the new long term tab in GlucoTab@MobileCare.

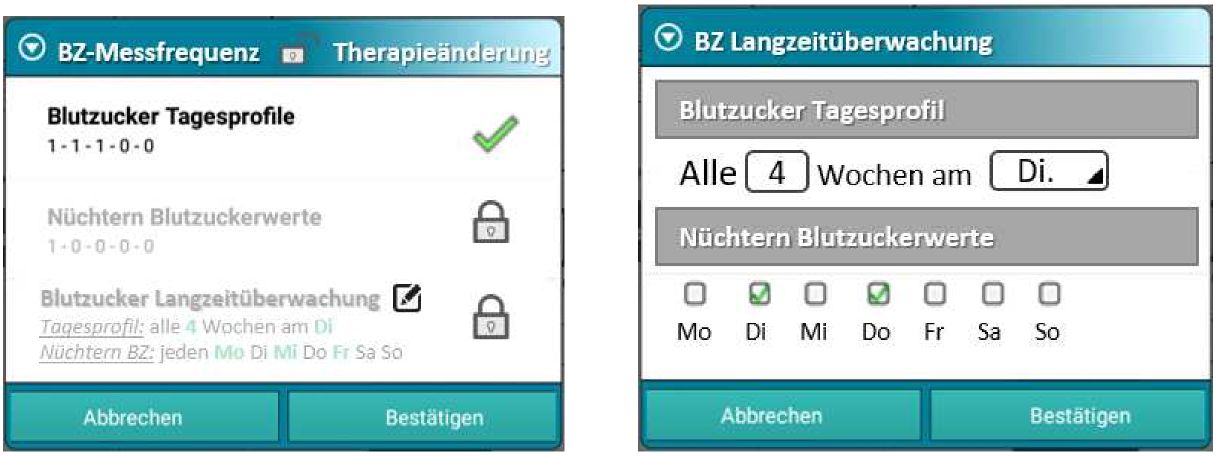
On the rightmost side of Figure 5.12 additional list icons (📋) are shown. Since the list only shows the most recent value, a tap on the corresponding icon should display a dialog with a simple history list of all previous dates and values.

To return to the subject of the new *ongoing* state, the selection of the blood glucose measurement frequency should be extended with a third option (see Figure 5.13a). HCPs should at any time be allowed to increase the frequency of measurements by downgrading the therapy state to a preceding one. However, decreasing the frequency of measurements by upgrading to a succeeding therapy state should only be possible with the approval of a physician (delegated order).

When selecting the *ongoing* state as the frequency for blood glucose measurements, further settings should be configured (see Figure 5.13b). It should be possible to schedule fasting blood glucose measurements for a chosen subset of weekdays, but still at least once a week. Moreover, an interval of weeks as well as a starting date for blood glucose profiles should be selected.

5.5.2 Final Implementation

The implementation of the *ongoing* state in the algorithm and therefore the back end had high priority. Conversely, displaying details of the algorithm and summaries of events to the user was considered as



BZ-Messfrequenz Therapieänderung

Blutzucker Tagesprofile
1 - 1 - 1 - 0 - 0 ✓

Nüchtern Blutzuckerwerte
1 - 0 - 0 - 0 - 0

Blutzucker Langzeitüberwachung
Tagesprofil: alle 4 Wochen am Di
Nüchtern BZ: jeden Mo Di Mi Do Fr Sa So

BZ Langzeitüberwachung

Blutzucker Tagesprofil

Alle 4 Wochen am Di.

Nüchtern Blutzuckerwerte

□ Mo ☑ Di □ Mi ☑ Do □ Fr □ Sa □ So

(a) Selection of the three frequency states for glucose measurements.

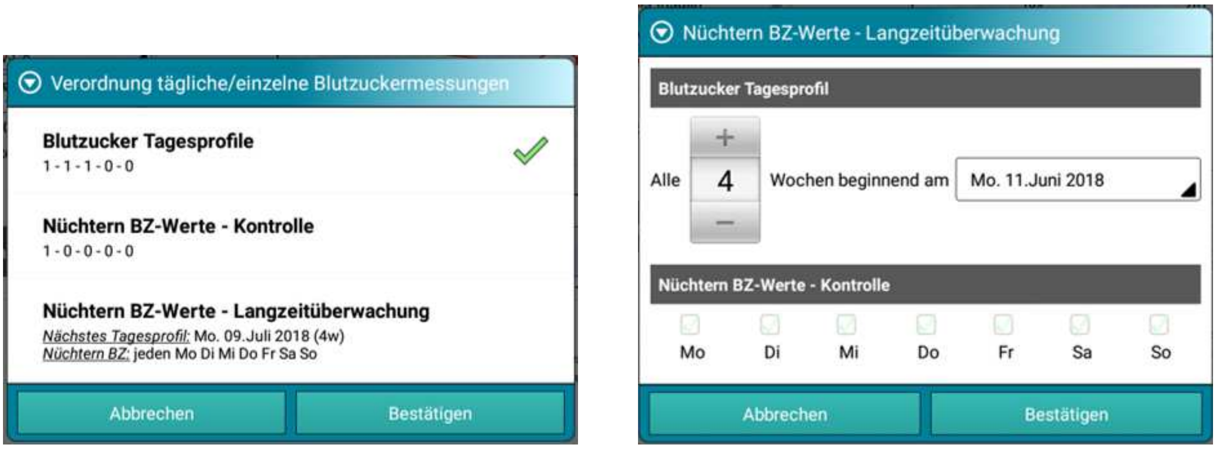
(b) Further settings of the new ongoing long-term management state.

Figure 5.13: Mockup of the new measurement frequency selection in GlucoTab@MobileCare.

nice to have.

Therefore, the *ongoing* state in the back end as well as the extended selection of the blood glucose measurement frequency in the front end were implemented first (see Figure 5.14a). Since delegation was only implemented in a simplified way, the planned locks to indicate that approval from a physician is needed are not part of the final dialog implementation. As described in section 5.3, the approval is only demanded after the event of selecting and submitting all changes to the back end.

Although the additional settings dialog for the new *ongoing* state was implemented as planned, the selection of individual weekdays was deactivated (see Figure 5.14b). The first release of GlucoTab@MobileCare (version 1.0) will be evaluated in a clinical trial. Purpose of the clinical trial is the verification of safety, efficacy, and usability of the medical device. Similarly, a clinical feasibility study has already been conducted for GlucoTab®, showing a good user acceptance and a significant reduction of hypoglycemia during diabetes management in comparison to a paper-based process [59].



Verordnung tägliche/einzelne Blutzuckermessungen

Blutzucker Tagesprofile
1 - 1 - 1 - 0 - 0 ✓

Nüchtern BZ-Werte - Kontrolle
1 - 0 - 0 - 0 - 0

Nüchtern BZ-Werte - Langzeitüberwachung
Nächstes Tagesprofil: Mo. 09.Juli 2018 (4w)
Nüchtern BZ: jeden Mo Di Mi Do Fr Sa So

Nüchtern BZ-Werte - Langzeitüberwachung

Blutzucker Tagesprofil

Alle 4 Wochen beginnend am Mo. 11.Juni 2018

Nüchtern BZ-Werte - Kontrolle

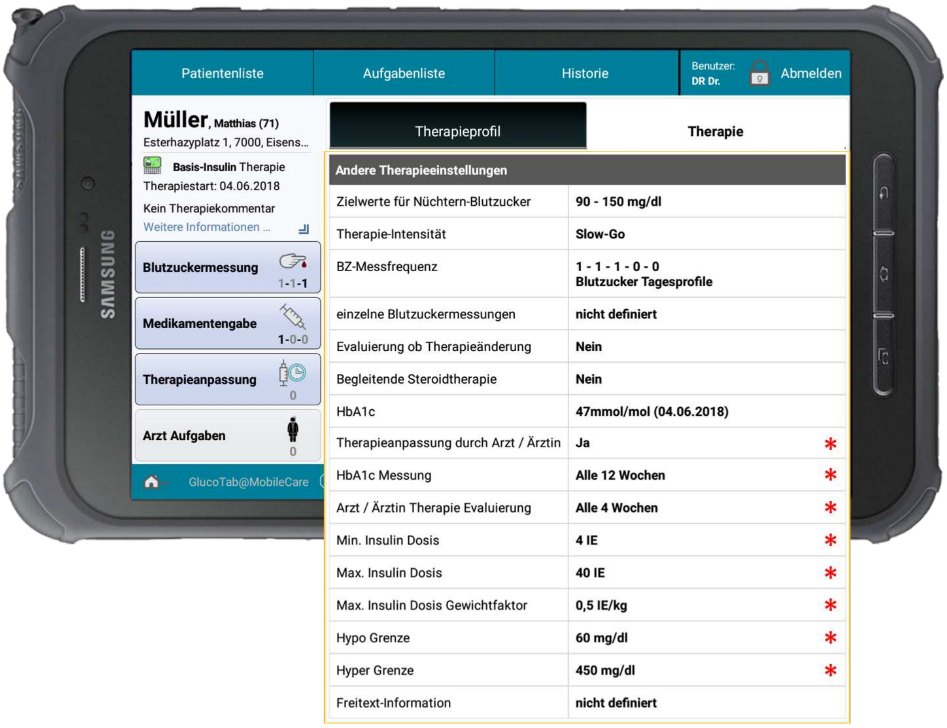
□ Mo □ Di □ Mi □ Do □ Fr □ Sa □ So

(a) Selection of the three frequency states for glucose measurements.

(b) Further settings of the new ongoing long-term management state.

Figure 5.14: Final measurement frequency selection in GlucoTab@MobileCare.

To collect more data in a shorter time and therefore reduce the costs of the clinical trial, the selection of weekdays is unalterably set to all days of the week. Apart from that, the selection of a weeks interval and starting date was implemented as intended. In retrospective, the design of deactivated check boxes does not clearly show the boxes in a *checked* state (see usability testing in chapter 8). Consequently, the distinction of the checked and unchecked state in the layout of deactivated check boxes should be improved in future releases.



Müller, Matthias (71)
Esterhazyplatz 1, 7000, Eisens...
Basis-Insulin Therapie
Therapiestart: 04.06.2018
Kein Therapiekommentar
Weitere Informationen ...

Andere Therapieeinstellungen		
Zielwerte für Nüchtern-Blutzucker	90 - 150 mg/dl	
Therapie-Intensität	Slow-Go	
BZ-Messfrequenz	1 - 1 - 1 - 0 - 0 Blutzucker Tagesprofile	
einzelne Blutzuckermessungen	nicht definiert	
Evaluierung ob Therapieänderung	Nein	
Begleitende Steroidtherapie	Nein	
HbA1c	47mmol/mol (04.06.2018)	
Therapieanpassung durch Arzt / Ärztin	Ja	*
HbA1c Messung	Alle 12 Wochen	*
Arzt / Ärztin Therapie Evaluierung	Alle 4 Wochen	*
Min. Insulin Dosis	4 IE	*
Max. Insulin Dosis	40 IE	*
Max. Insulin Dosis Gewichtfaktor	0,5 IE/kg	*
Hypo Grenze	60 mg/dl	*
Hyper Grenze	450 mg/dl	*
Freitext-Information	nicht definiert	

Figure 5.15: Final long-term information displayed in patient view of GlucoTab@MobileCare. Entries marked with a red asterisk (*) have been newly added.

Due to the low priority of the planned long-term tab, only the most important values which are anyway already used by the algorithm were chosen to be presented to the user. As a result, the introduction of a new long term tab was discarded and instead the tab showing general therapy settings was expanded with the following entries (see Figure 5.15):

- indication whether *a physician* performed the last therapy adjustment
- interval of weeks in which a HbA$_{1C}$ measurement task is scheduled
- interval of weeks in which a therapy confirmation by a physician is required
- factor used to calculate the upper insulin dosage limit based on the patient's weight
- lower/upper insulin dosage limit that can be suggested by the algorithm
- lower/upper blood glucose limit indicating a hypo/hyperglycemia

Chapter 6

Authentication with Near Field Communication (NFC) Tags

The first subject of investigation for a more convenient user experience in GlucoTab@MobileCare was the process of unlocking the device screen and logging in to the system. GlucoTab® is set up to use a Personal Identification Number (PIN) to unlock the tablet screen. This allows a user to *view* the task list, however to *perform* further actions, the user needs to authenticate against the server with a valid user name and password combination.

Regarding the lock screen on smartphones, Harbach et al. [17, 18] conducted studies showing that the time spend per day to unlock the smartphone is on average 2.6 minutes. Although the actual time to enter a correct PIN into the device comes down to only 1.9 seconds on average (much time is spend checking notifications on the lock screen before an actual unlock attempt), there exist other secure methods to unlock the lock screen in Android without entering any codes.

Biometric data is a secure alternative that can be used to unlock the screen if a fingerprint sensor or facial recognition software is available. However, the tablet that is used for GlucoTab@MobileCare does not fulfill the hardware requirements for biometric identification. Yet another option in Android is Smart Lock[1], which allows to register a trusted NFC or Bluetooth device. Whenever this trusted device is in range, the smartphone or tablet can be unlocked without the need to enter a code.

In regard to the authentication against the server, typing account credentials on a mobile device with a virtual keyboard bears no comparison to using a hardware keyboard. First of all, it is more time consuming due to altered modalities, such as an increased typing effort for numerical and special characters. Secondly, most people will only use two fingers and focus on the keyboard while typing, compared to typing with all fingers and without using the sense of sight. Thirdly, the usage of a virtual keyboard during authentication leads to weaker password choices by users, as shorter and less secure passwords are chosen. Finally, new security risks emerge, such as shoulder surfing: nearby observers detecting the user input by looking at the screen of the smartphone or tablet device [47, 55, 60].

Since authentication is a common requirement for online services and websites, the Fast IDentity Online (FIDO) Alliance proposed a *standards-based and interoperable* authentication ecosystem. The FIDO Universal Authentication Framework (UAF) [38] provides a *password-less* authentication method using public key cryptography. A FIDO UAF authenticator stores a private key which by design never leaves the device. The authenticator is used to solve authentication challenges with the aid of the private key. Communication to the authenticator may be supported over Universal Serial Bus (USB), NFC, or Bluetooth Low Energy (BLE). As a result, an authentication without a password is possible by simply attaching or approximating an authenticator to the mobile device.

[1] https://get.google.com/smartlock

To improve the user experience in GlucoTab@MobileCare, a conceptual authentication design using Smart Lock and FIDO UAF with an NFC tag was developed (see Figure 6.1). The workflow to unlock the tablet device and log in consists of three user interactions. Firstly, the user needs to press a button on the tablet to wake up the screen which will implicitly power on NFC capabilities of the device. Secondly, an NFC tag is registered as trusted device on the tablet and unlocks the tablet if it is held close to the NFC antenna. Finally, a confirmation or user interaction on the NFC tag may be necessary to approve the authentication process.

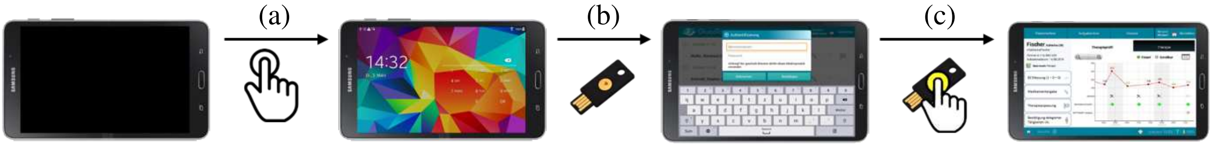


Figure 6.1: Prototype of an authentication process with an NFC tag: user interaction to wake up the device screen (a), followed by an unlocking of the screen with a trusted NFC device (b), and finally an authentication against the server based on the UAF (c).

However, preliminary tests of Smart Lock unlocking the tablet device that is used for GlucoTab@ MobileCare with an NFC tag turned out insufficient. The area and antenna range where NFC communication could be initiated was hard to find and demanded a precise alignment of the NFC tag on the back cover of the tablet.

Furthermore, the support of trusted NFC devices was discontinued in 2017, as reported by Google employees[2]. Although the used tablet still supports trusted NFC devices, future versions of GlucoTab@ MobileCare might be deployed on different hardware and therefore not support trusted NFC devices anymore.

Considering the preliminary test results, the long term support of trusted NFC devices, and the costs of FIDO Authenticators for each GlucoTab@MobileCare user, investigations on the NFC authentication were discontinued. Instead, a Bluetooth based protocol for an automated glucose measurement transmission from the glucose meter to the tablet device was proposed (see chapter 7).

[2] https://issuetracker.google.com/issues/65425413#comment24

Chapter 7

Automated Glucose Measurement Transmission Protocol

A study conducted by Selvan et al. [57] assessed the accuracy of logbook entries during self-monitoring of blood glucose. After 44 months the logbook entries were compared with the stored measurements in the glucose meter. Furthermore, the impact of inaccurate logbook entries on the long-term glycemic control was evaluated. As a result, 32.67% of logbooks contained errors, whereof 42.42% included omission, 27.27% involved fabrication, **18.18% were erroneous**, and 12.12% encompassed other errors. Long-term evaluation showed lower HbA_{1C} values for patients with accurate logbook entries compared to patients with inaccurate entries.

Another study [11] compared a paper-based protocol to a computerized diabetes management system. Besides the discovery of a significantly increased risk of hypoglycemic events after an insulin dosing error, overall **error rates** regarding blood glucose documentation **of 4.9%** for the paper-based protocol **and 4.0%** for the computerized system were identified. Moreover, a Point of Care Testing (POCT) device was used for blood glucose measurements. Although the device automatically transmitted the measurement values to a Laboratory Information System (LIS), the delay until these values could be used for medication dose calculations was too high:

> “The absence of instant automated transfer of BG measurements from POCT BG meters to GlucoTab® presents a potential risk. Together with hospital staff and the manufacturer we will search for a way to provide BG values in a timely manner because immediate availability and automated handling of BG values directly at the point of care can eliminate these errors.” Donsa et al. [11, page 64]

An automated glucose measurement transmission protocol was developed to *eliminate* the risk of erroneous documentation of blood glucose measurements as well as to allow a convenient and *instant* transfer to GlucoTab®. The protocol design is based on recommendations of the Continua Design Guidelines (see section 7.1). Furthermore, BLE is used for the transmission and communication between glucose meters and mobile devices receiving the health data. In particular, the *Glucose Profile* [22] is used on top of the Generic Attribute Profile (GATT) of Bluetooth (see section 7.2).

A background service running on the mobile device gathers the transferred health data and forwards the information to other applications that have registered as listeners and have implemented the required interface (see section 7.4). As a proof of concept, the designed protocol was integrated into GlucoTab@ MobileCare (see section 7.5) and evaluated during usability testing (see chapter 8).

The developed protocol design was published and presented as scientific paper at the eHealth 2018[1] conference [43]. Moreover, a scientific paper was presented by Frohner et al. [12] at the DSAI 2018[2] conference, where the protocol was used for a telemonitoring system of blood glucose (see Appendix A).

[1] http://www.ehealth2018.at

[2] http://www.dsai.ws/2018

Recent research shows great effort in implementing medical device communication based on the ISO/IEEE 11073 standards. The standards are constantly enhanced and adapted to meet the demands of the evolving Internet of Things (IoT) in the health care context [33, 54]. A special topic of interest are mobile devices (e.g. smartphones or tablets) which are capable of communicating with medical sensors and can be utilized by the patients themselves. Medical device communication implementations on mobile devices frequently use a middleware or adapter to support ISO/IEEE 11073 conform communication and provide a mapping for non standardized devices to the ISO/IEEE 11073 standards [6, 37, 41, 56].

Approaches were made to reduce power consumption by utilizing low energy transport technologies such as BLE and ZigBee [7, 34, 56]. However, since most smartphones and tablets rather support Bluetooth than ZigBee, a BLE based medical device communication seems more suitable for patients in the personal health care environment.

7.1 Continua Design Guidelines

The Continua Design Guidelines [49] are developed and maintained by the Personal Connected Health Alliance. A major intention of the guidelines is to provide a guidance for developers and manufacturers to exchange personal health data among different *entities* by specifying communication interfaces. Well known standards are incorporated into the Continua Design Guidelines to achieve a high interoperability of health care devices and health care applications. Four *entities* are defined for holding and processing health data and three high level **interfaces** for communication between these *entities* by the Continua Design Guidelines (see Figure 7.1):

1. **Personal Health Device Interface (PHD-IF)**: allows the transmission of health data from a *Personal Health Device (PHD)* to a *Personal Health Gateway (PHG)*. First, the *PHD* (e.g. measurement device for blood pressure, blood glucose, heart rate, pulse, or body weight) is used to generate personal health data. Next, the information is transmitted with a well known transport technology (e.g. USB, NFC, Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR), BLE, or ZigBee). Finally, health data is received and further processed by a *PHG* (e.g. mobile phone, tablet, laptop, or PC).

2. **Service Interface**: defines data exchange between a *PHG* located nearby the patient and a *Health & Fitness Service* running on a distant server or hosted in the cloud.

3. **Healthcare Information Service Interface**: specifies health data exchange between one *Health & Fitness Service* and either another *Health & Fitness Service* or a *Healthcare Information Service*.

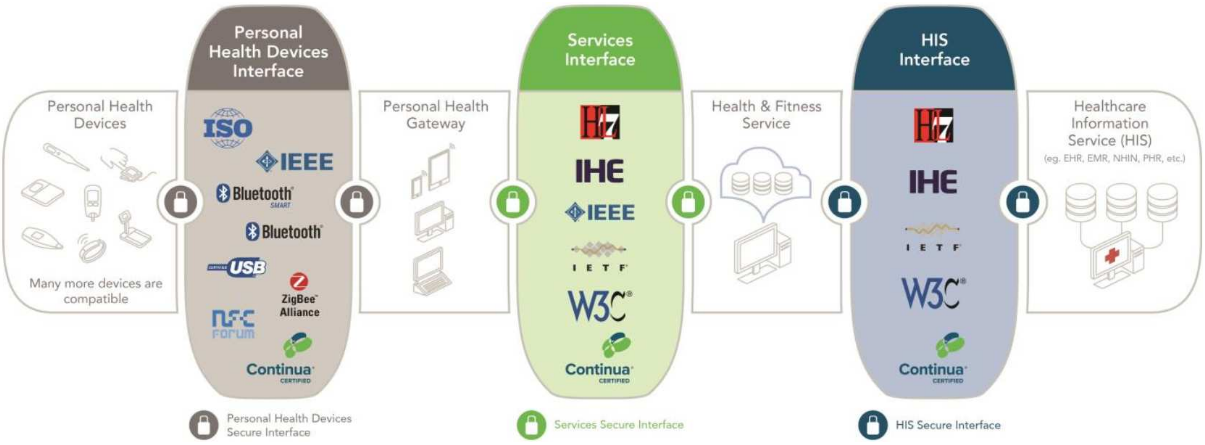


Figure 7.1: High level architecture of the Continua Design Guidelines.

The developed protocol of an automated glucose measurement transmission focuses specifically on the **PHD-IF** to transmit blood glucose measurements from a glucose meter to an Android tablet that is running GlucoTab®. However, the *H.811 Personal Health Devices Interface design guidelines* [50] define two sub interfaces in the PHD-IF related to their underlying transport technology:

1.a) USB, NFC, Bluetooth BR/EDR, and ZigBee belong to the **X73 sub interface**, which implements the ISO/IEEE 11073-20601 Optimized Exchange Protocol and consequently the therein referenced ISO/IEEE 11073-10101 Nomenclature as well.

1.b) In contrast, BLE belongs to a separate **Bluetooth LE sub interface**, which *does not* implement the ISO/IEEE 11073-20601 Optimized Exchange Protocol. Instead it uses compatible data types from the BLE protocol and can be transcoded according to the *Personal Health Devices Transcoding White Paper* [14] in order to correspond with the ISO/IEEE 11073 ecosystem.

The developed automated glucose measurement transmission protocol is based on the Bluetooth LE sub interface of the PHD-IF. Transcoding the BLE data types to the ISO/IEEE 11073-10101 Nomenclature is only an informative recommendation in the Bluetooth *Glucose Profile* [22, page 10] for compatibility with the ISO/IEEE 11073 ecosystem. However, transcoding is *required* in order to comply with the Continua Design Guidelines [50, page 116]. Therefore, regardless of which PHD-IF sub interface is used, a standardized nomenclature must be accomplished by the PHG.

ISO/IEEE 11073-10101 Nomenclature [30, 31] defines various terms used in the medical device communication (MDC) by assigning reference symbols and number codes in a structured way. Codes are 32-bit words consisting of a 16-bit code block number and a 16-bit term code. There are 12 code blocks defined in the nomenclature for a coarse classification of terms (see Table 7.1).

Symbol	Code Block Number	Comment
MDC_PART_UNSPEC	0	Unspecified
MDC_PART_OBJ	1	Object Infrastructure
MDC_PART_SCADA	2	Supervisory Control and Data Acquisition (SCADA)
MDC_PART_EVT	3	Event
MDC_PART_DIM	4	Dimension
MDC_PART_VATTR	5	Virtual Attribute
MDC_PART_PGRP	6	Parameter Group
MDC_PART_SITES	7	[Body] Site
MDC_PART_INFRA	8	Infrastructure
MDC_PART_FEF	9	File Exchange Format
MDC_PART_EXT_NOM	256	External Nomenclature
MDC_PART_PVT	1024	Private

Table 7.1: Table of partition (or code block) codes as specified in the ISO/IEEE 11073-10101 Nomenclature [30].

Terms (e.g. $\frac{mg}{dl}$ to describe a dimension) are defined in the ISO/IEEE 11073-10101 Nomenclature with a reference symbol that is assigned to a specific term code (e.g. MDC_DIM_MILLI_G_PER_DL = 2130). The nomenclature code is then derived by inserting the term code and the code block number (as specified in Table 7.1) into Equation 7.1. A more efficient computation of the nomenclature code, consistent with Equation 7.1, is achieved by *left shift («)* and *bitwise OR (|)* operations as shown in Equation 7.2 [30, 31].

$$NomenclatureCode = \left(CodeBlockNumber * 2^{16}\right) + TermCode \tag{7.1}$$

$$NomenclatureCode = (CodeBlockNumber \ll 16) \mid TermCode \tag{7.2}$$

Therefore, the nomenclature code is derived in an efficient and interoperable way and can be used to uniquely identify dimensions, events, medical devices, and much more during medical device communication [30]. The Continua Design Guidelines suggest to identify all components of the medical device communication by their assigned nomenclature code.

Finally, the ISO/IEEE 11073-10417 Glucose Meter Device Specialization [32] should be mentioned. It refines the ISO/IEEE 11073-20601 Optimized Exchange Protocol and extends the ISO/IEEE 11073-10101 Nomenclature with additional terms specific to *glucose meters*.

In conclusion, the Continua Design Guidelines specify a standardized and interoperable way of communication for devices used in personal health management and health care delivery [49].

7.2 Bluetooth Generic Attribute Profile (GATT)

The *Glucose Profile* [22] defines a client and a server role during the exchange of glucose measurement data. In terms of the an automated glucose measurement transmission protocol, the glucose meter incorporates the server role offering a measured glucose value and the mobile device incorporates the client role receiving and further processing the value. Bluetooth's Generic Attribute Profile (GATT) is the basis of the Glucose Profile and is in turn built on top of the Attribute Protocol (ATT) as specified in the *Bluetooth® Core Specification* [5].

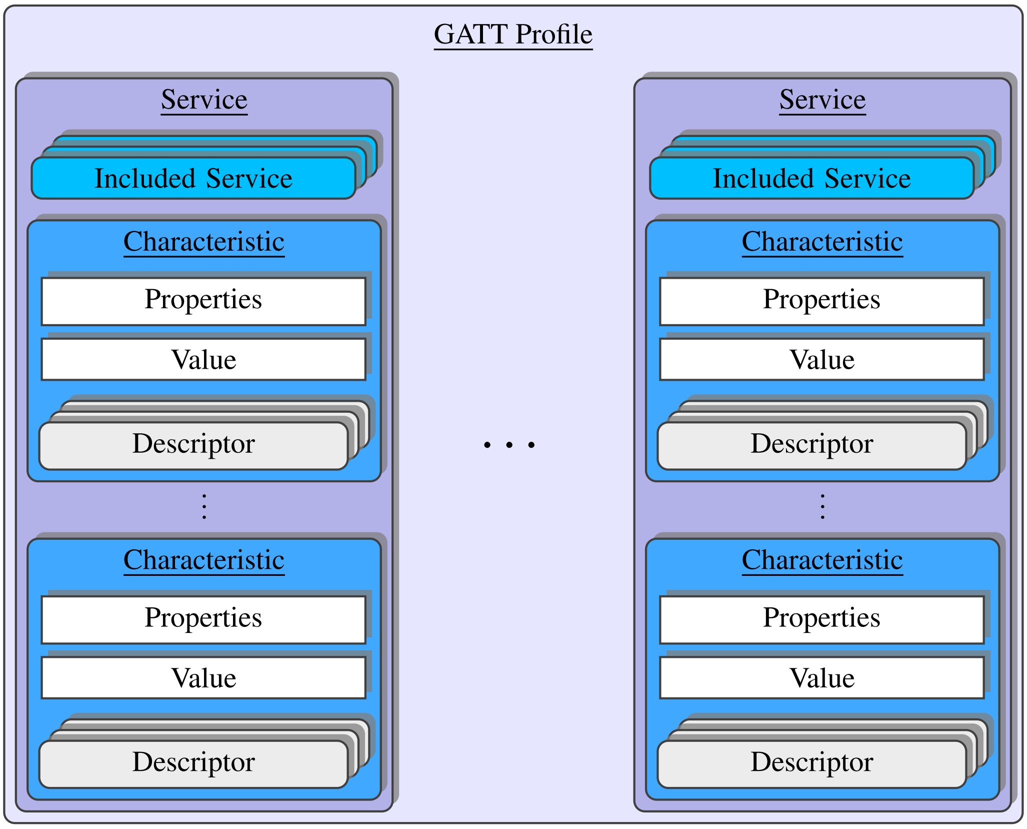


Figure 7.2: Hierarchical structure of the Generic Attribute Profile (GATT) as specified in the *Bluetooth® Core Specification* [5].

A GATT Profile [5] is a hierarchical structure definition of services, characteristics, and descriptors that are supplied by the server role (see Figure 7.2). Each profile contains at least one service, which facilitates an intended use case of the Bluetooth device. In the service definition, additional services may be referenced and are therefore included to the current service. Furthermore, zero or more characteristics are defined in a service. Characteristics hold the actual data value that can be read or written by a client. Moreover, optional descriptors may be specified to describe the value or allow specific configurations of the characteristic.

One particular descriptor defined in the *Bluetooth® Core Specification* [5] is the *Client Characteristic Configuration*. By writing certain values onto this descriptor, the server is asked to *notify* the client whenever the value of the characteristic changes. More precisely, a *notification* is used to send the changed value without acknowledgment to the client, as opposed to an *indication* where acknowledgment of the client is excepted by the server.

Universal Unique Identifiers (UUIDs) are used in GATT to reference services, characteristics, and descriptors. Since UUIDs are 128-bit values, the Bluetooth SIG defined a Bluetooth Base UUID with the value *00000000*-0000-1000-8000-00805F9B34FB. This allows standardized services to use a smaller 32-bit or 16-bit UUID which can be converted to a 128-bit UUID by filling up the *most significant bits* of the Bluetooth Base UUID [5].

The *Glucose Profile* [22] was used for the development of the automated glucose measurement transmission protocol. *Glucose Service* and *Device Information Service* are mandatory services that need to be implemented in the server role of the *Glucose Profile*.

In the *Device Information Service* [19], several characteristics are specified which provide a detailed description of the device. According to the specifications, all of the characteristics are *optional*. However, the *Glucose Profile* [22] overrules this specification and marks the **Manufacturer Name**, **Model Number**, and **System ID** characteristics as *mandatory*. A summary of all characteristics with respect to the *Glucose Profile* is given in Table 7.2.

DIS Characteristic	**Requirement**	**Description**
Manufacturer Name String	**M**andatory	Name of the manufacturer of the given device
Model Number String	**M**andatory	Model number of the device (assigned by vendor)
System ID	**M**andatory	An organizationally unique identifier followed by a manufacturer-defined identifier
Serial Number String	**O**ptional	Serial number of the device
Hardware Revision String	**O**ptional	Revision of the hardware within the device
Firmware Revision	**O**ptional	Revision of the firmware within the device
Software Revision String	**O**ptional	Revision of the software within the device
ISO/IEEE 11073-20601 Regulatory Certification Data List	**O**ptional	Regulatory and certification information for the device as defined in the ISO/IEEE 11073-20601 Optimized Exchange Protocol
PnP ID	**O**ptional	Combination of vendor ID, vendor ID source, product ID, and product version

Table 7.2: Table of Device Information Service (DIS) characteristics as specified by the *Device Information Service* [19] and overruled by the *Glucose Profile* [22].

The *Glucose Service* [23] defines the following four essential characteristics which are relevant during communication and data exchange with a glucose meter:

- **Glucose Measurement**: a characteristic that represents a measured glucose concentration. The value of this characteristic is a well defined bit string and contains at least a *flags* field, a *sequence number*, and a *base time*. The flags field indicates if further fields are present as well as in which base unit the glucose concentration is given ($\frac{kg}{L}$ or $\frac{mol}{L}$). One additional field that may be present is the *time offset*. The sum of the base time and time offset gives the actual time that is displayed to the user. Another optional field is the sensor status annunciation, which in turn is another bit field indicating detailed sensor status or errors (e.g. low battery, sensor temperature too low/high). Finally, there is one bit in the flags field that specifies whether a glucose concentration field (measured glucose concentration value), a sample location field (e.g. finger, earlobe, or an alternate site test), and a type field (e.g. capillary whole blood, venous plasma, or interstitial fluid) is present. Furthermore, the characteristic contains a client characteristic configuration *descriptor* which allows *notifications* of new glucose measurement values.
- **Glucose Measurement Context (optional)**: a characteristic that supplies additional context information for a glucose measurement. This characteristic is defined as *optional* in the *Glucose Service* specification. Similar to the glucose measurement characteristic, a *flags* field provides information on which other fields are present in the bit string value. Additional fields contain information about *carbohydrates*, *meal* relationships (e.g. fasting, preprandial, or postprandial), *testers* (e.g. self, HCP, or laboratory test), *health* conditions (e.g. under stress, during menses, or minor/major health issues), *exercise* intensity (expressed as a percentage) and duration, *medications*, and HbA$_{1C}$ levels. Again, the client characteristic configuration *descriptor* is available to enable *notifications*.
- **Glucose Feature**: a characteristic that provides a bit field indicating which features are supported by the glucose meter. Support of the sensor status annunciations given by the glucose measurement characteristic value are for example described by this characteristic's value.
- **Record Access Control Point (RACP)**: allows a client to query stored glucose measurement records from the server. A device implementing the *Glucose Service* **must** at least support the RACP operation to retrieve *all records* and the operation to retrieve records with a sequence number that is *greater than or equal* to a given number. Other *optional* RACP operations are for instance the retrieval of the first or last stored record, retrieval of records within a given range of sequence numbers or timestamps, and in a similar manner operations to delete stored records.

On the subject of an automated glucose measurement transmission protocol, notifications on **Glucose Measurement** characteristics and **Glucose Measurement Context** characteristics show great promise. Furthermore, the RACP offers a convenient method to synchronize measured glucose concentrations between a glucose meter and a mobile device.

7.3 Android's Interprocess Communication (IPC) and GATT support

The Android Application Programming Interface (API) [15] provides a comfortable method to establish a GATT connection with Bluetooth devices. Calling the `connectGatt` method of a `BluetoothDevice` object returns a new `BluetoothGatt` instance, which can be used to read and write Bluetooth GATT characteristics and descriptors. Furthermore, the `BluetoothGattCallback` class is used to register and receive callbacks on particular events. This allows an asynchronous way to handle connection state changes, GATT service discoveries, and read or write events of GATT characteristics.

If several applications on the same mobile device want to communicate with a glucose meter, it would be inconvenient to implement the same GATT communication and marshalling as well as unmarshalling of the glucose measurement characteristics in each application. Therefore, the developed protocol uses the concept of a middleware. The middleware is a service running in the background (not visible to the user) and is responsible for the management of connections to glucose meters and the communication between glucose meters and the mobile device. As a result, all applications that want to communicate with a glucose meter only need to implement communication with the middleware.

The *Android Developer Guides* [16] describe three commonly used methods for the communication between different applications within the same mobile device, also called Interprocess Communication (IPC). The first method sends small data packages from one process to another, which are called *Intents*. An Intent primarily consists of an *action* and *data* field. The *action* field roughly tells a receiving application what is supposed to happen with the *data*. For instance, an Intent containing the action ACTION_VIEW and the data `http://google.com` could be received by a browser application, which will then navigate and view the specified website. Whereas the same ACTION_VIEW with a data content of `file:///sdcard/image.jpg` could be interpreted by a photo gallery application to display the given image.

Secondly, the *Android Developer Guides* [16] mention methods utilizing Android's *binder* interface. The binder kernel module provides low level functionality to exchange data packages between processes and allows to perform Remote Procedure Calls (RPCs). To conveniently implement the complex binder interface, Android provides an abstraction wrapper with the Android Interface Definition Language (AIDL).

As pointed out by Manas and Grancini [40] in their description of IPC, two processes are in general not allowed to access each others memory space. Therefore, communication is performed with RPCs, which allows to call predefined methods of another process as if it was called in a local context. The definition of RPC methods is done with AIDL, which looks similar to definitions in the Java programming language. However, during compilation of the AIDL file, several components for the IPC will be generated by the compiler. A `Proxy` and a `Stub` inner class are generated for the defined interface, which represent the client side (calling the RPC) and server side (executing the method), respectively. Developers only need to implement the methods that were defined with AIDL in the inner Stub class and the Android Operating System (OS) will handle the marshalling and unmarshalling of primitive data types from the server side to the client side (see Figure 7.3). However, if more complex data objects are used during RPCs, those objects need to implement the `Parcelable` interface which defines marshalling and unmarshalling of the given object.

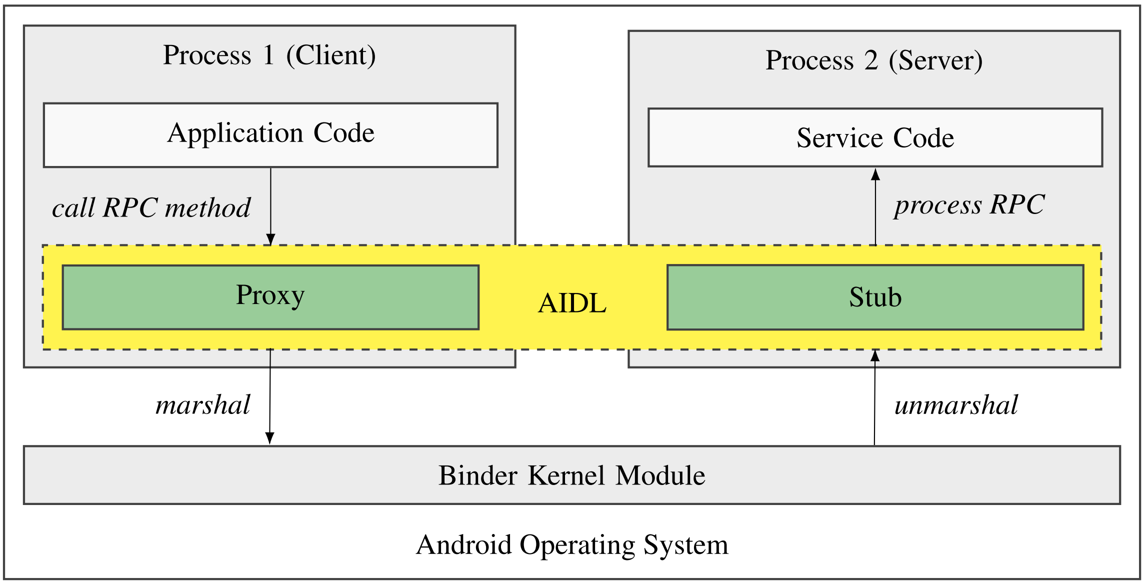


Figure 7.3: Interprocess Communication (IPC) using the Android Interface Definition Language (AIDL) and Android's binder interface.

A third method for IPC described in the *Android Developer Guides* [16] are *content providers*. Similar to a database, a content provider offers methods to **C**reate, **R**etrieve, **U**pdate, and **D**elete (CRUD) the stored data. Furthermore, it is possible to grant other applications and processes permissions to read and write the data of the content provider.

In terms of an automated glucose measurement transmission protocol the method of using `Intents` for IPC provides a loose coupling between middleware and the target application. However, a tight coupling could help to sustain the well defined and standardized data structures along the entire path of medical device communication. Furthermore, a performance analysis showed significantly less latency and CPU usage with Android's *binder* or *content provider* compared to the communication with `Intents` [20].

All glucose measurements could be synchronized and stored with a *content provider*. Android's `ContentObserver` would enable applications to be notified if new measurements are received and stored by the middleware. However, since the measurement data contains sensible patient data and several people might have access to one and the same device, it may be undesirable to permanently store the glucose measurements on the mobile device. Moreover, a glucose meter does not only act as a data storage of measurement records and might offer various other GATT services apart from the Glucose Service (e.g. Current Time Service).

AIDL provides a tight coupling of applications and allows the exchange of complex data structures. Operations of the RACP could be performed with RPCs and would be executed in an asynchronous manner. Callback listeners could be used to notify applications whenever new glucose measurements are received.

7.4 Protocol Design

The developed automated glucose measurement transmission protocol is based on the Continua Design Guidelines and uses the Glucose Profile to communicate with BLE devices. A service is running on the mobile device in the background to manage the BLE communication with glucose meters (previously also described as middleware). Interaction with the background service is specified via AIDL and the `IGlucoseMeterManager` interface. Applications that want to communicate with this glucose meter manager need to implement the `IGlucoseMeterManagerListener` interface. The connection to the service is currently established with Android's `bindService` method using an *explicit* component name to identify the service. Afterwards, the connection state is monitored using an implementation of the `ServiceConnection` interface.

The primary method of the `IGlucoseMeterManager` interface is the `registerListener` function. Applications can register themselves as listeners of a glucose meter with a given Media Access Control (MAC) address (see Figure 7.4). A callback listener is passed as an `IBinder` reference and must implement the `IGlucoseMeterManagerListener` interface. As a result, the glucose meter manager returns an `InvocationResult` object which yields success or failure information and an additional error code and error message in case of failure.

Behind the scenes, the middleware creates a BLE connection to the Bluetooth device with the given MAC address. Subsequently a GATT connection is initiated and GATT services are discovered. If the connected device does not support the Glucose Service (UUID 0x1808) or if the glucose measurement and RACP characteristics (UUID 0x2a18 and 0x2a52) are not available, then an exception is thrown and the connection is closed. Otherwise, GATT *notifications* for the glucose measurement and glucose measurement context characteristics as well as *indications* for the RACP characteristic are enabled.

As soon as a listener implementing the `IGlucoseMeterManagerListener` interface is successfully registered at the manager, `GlucoseMeasurement` objects can be received over the `onReceiveData` call-

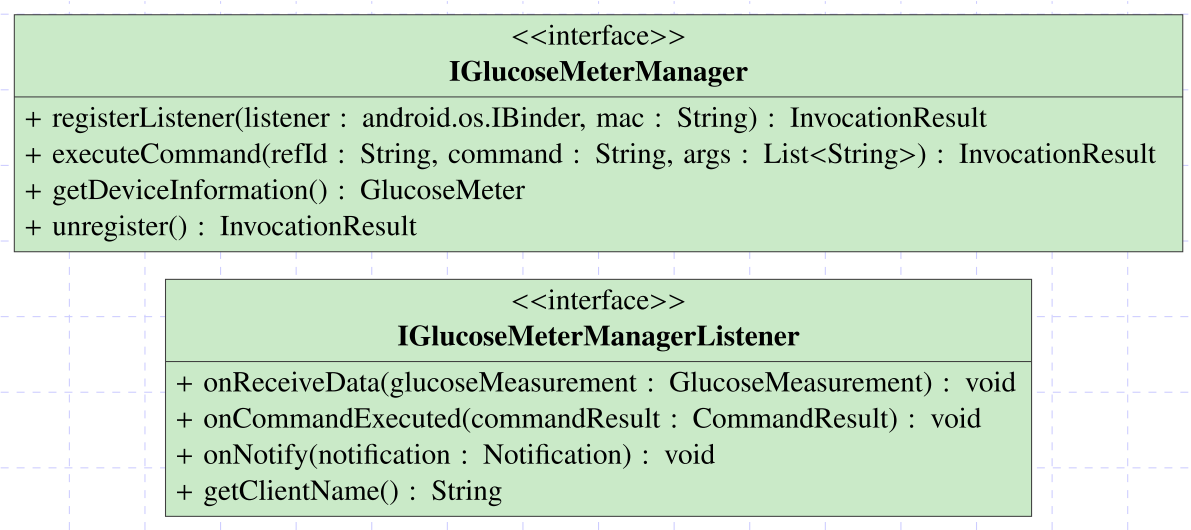

<<interface>>
IGlucoseMeterManager
+ registerListener(listener : android.os.IBinder, mac : String) : InvocationResult
+ executeCommand(refId : String, command : String, args : List<String>) : InvocationResult
+ getDeviceInformation() : GlucoseMeter
+ unregister() : InvocationResult
<<interface>>
IGlucoseMeterManagerListener
+ onReceiveData(glucoseMeasurement : GlucoseMeasurement) : void
+ onCommandExecuted(commandResult : CommandResult) : void
+ onNotify(notification : Notification) : void
+ getClientName() : String

Figure 7.4: Interface definition of the `IGlucoseMeterManager` and `IGlucoseMeterManagerListener`.

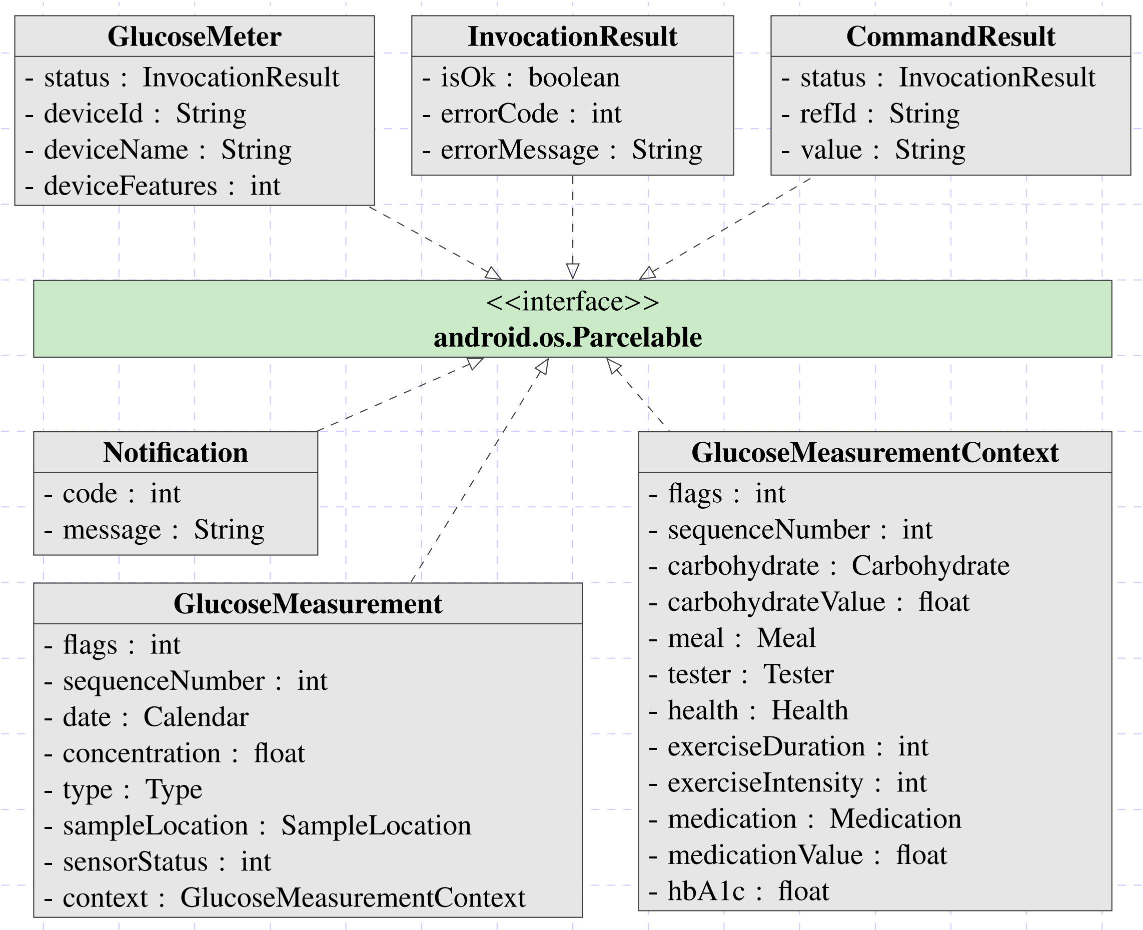


Figure 7.5: Unified Modeling Language (UML) class diagram of complex data structures implementing the `android.os.Parcelable` interface which are used during IPC.

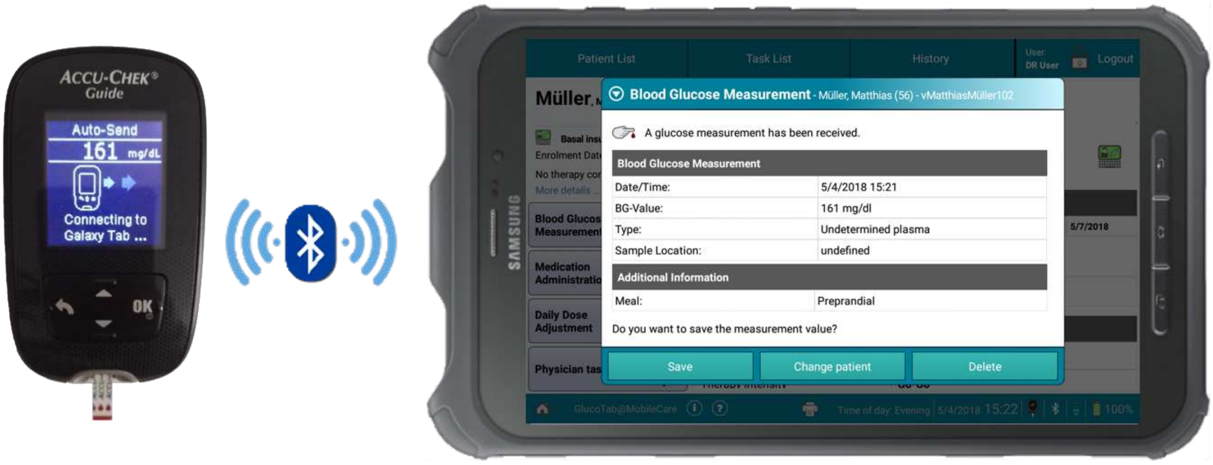
back. In order to send complex data structures such as the `GlucoseMeasurement` or `InvocationResult` from one process to the other (IPC), they need to implement the `android.os.Parcelable` interface (see Figure 7.5) which defines the marshalling and unmarshalling of the information into primitive data types that are supported by Android's binder module (e.g. `int`, `long`, `char`, or `boolean`).

In order to invoke RACP operations on the glucose meter, the `executeCommand` method is exposed by the `IGlucoseMeterManager` interface. A reference ID is given as the first argument of this method to link command results to the initiated requests in the asynchronous workflow. Further arguments are a String constant representing the actual command to execute followed by optional parameters for the command. An `InvocationResult` is returned, indicating if a connection to the glucose meter could be established and whether the command was successfully initiated over the BLE protocol stack. As a result, glucose measurement records are delivered through the `onReceiveData` callback, followed by a `CommandResult` object received through the `onCommandExecuted` callback.

The remaining two methods of the `IGlucoseMeterManager` interface are used to retrieve detailed device information of a glucose meter (`getDeviceInformation`) and to `unregister` a previously registered listener. On the other end, the `IGlucoseMeterManagerListener` interface provides the `getClientName` method that can be used to identify listeners. Furthermore, the `onNotify` callback can be implemented to receive additional event notifications from the glucose meter manager, such as the connection state to BLE devices.

7.5 Showcase Integration into GlucoTab®

As proof of concept, the developed protocol for an automated glucose measurement transmission was integrated into GlucoTab@MobileCare. Therefore, a local service was added to GlucoTab@MobileCare which binds to the glucose meter manager service that is running in another process. On the one hand, the local service updates the status bar of GlucoTab@MobileCare with the current connection state to the BLE enabled glucose meter and a special icon is displayed if a new glucose measurement is received. On the other hand, the service *temporarily* stores the glucose measurements until the HCP is ready to deposit the records into the GlucoTab® system.



Blood Glucose Measurement - Müller, Matthias (56) - vMatthiasMüller102

A glucose measurement has been received.

Blood Glucose Measurement	
Date/Time:	5/4/2018 15:21
BG-Value:	161 mg/dl
Type:	Undetermined plasma
Sample Location:	undefined

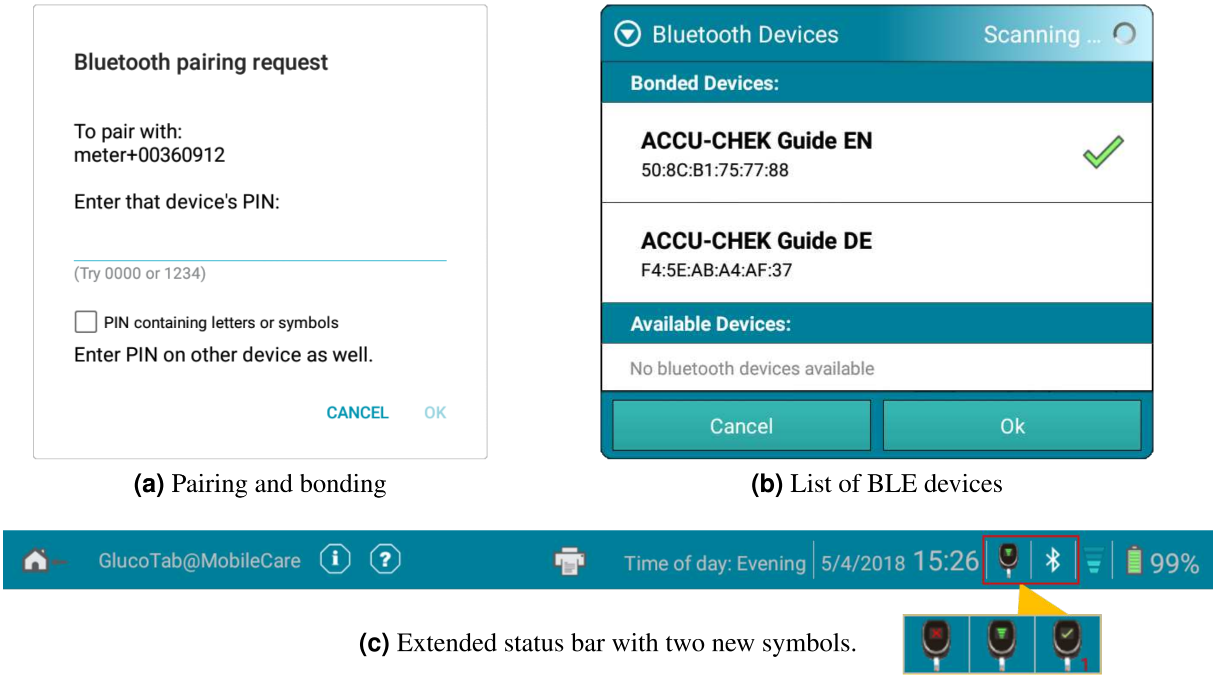
Additional Information	
Meal:	Preprandial

Do you want to save the measurement value?

Figure 7.6: Automated glucose measurement transmission from an Accu-Chek® Guide glucose meter to the GlucoTab@MobileCare Android application.

The Accu-Chek® Guide glucose meter is continua certified[3] and can therefore be used to measure a glucose concentration and transmit the measurement value to an Android tablet with the developed protocol. GlucoTab@MobileCare automatically receives the glucose concentration along with additional measurement context information as soon as the measurement is completed and subsequently displays a summary dialog to the user (see Figure 7.6). Precondition for the automated glucose measurement transmission protocol to work is a *bonding* between glucose meter and the mobile device. In terms of BLE, *bonding* is a method of storing cryptographic keys in order to quickly reestablish a secure connection with the so called *pairing* process [5]. The process of pairing and bonding only needs to be performed *once* before the first glucose measurement transmission is done.

In order to manage and view BLE enabled glucose meters within GlucoTab@MobileCare, a new Bluetooth symbol was added to the status bar (see Figure 7.7). A simple tab gesture on the icon shows a list of already bonded devices and allows the user to scan for further available devices that are advertising to support the glucose service. The current implementation only allows one active GATT connection at a time, which is why users need to select one of the glucose meters in the bonded devices list (see tick in Figure 7.7b). Likewise, another symbol was added to the status bar that indicates the current connection state and availability of new glucose measurement records to store with regards to the selected glucose meter. Possible states indicated by this symbol are *disconnected*, *connected*, and *measurements available*.



Bluetooth pairing request

To pair with:
meter+00360912

Enter that device's PIN:

(Try 0000 or 1234)

□ PIN containing letters or symbols

Enter PIN on other device as well.

CANCEL OK

(a) Pairing and bonding

Bluetooth Devices — Scanning ...

Bonded Devices:

ACCU-CHEK Guide EN ✓
50:8C:B1:75:77:88

ACCU-CHEK Guide DE
F4:5E:AB:A4:AF:37

Available Devices:

No bluetooth devices available

Cancel Ok

(b) List of BLE devices

GlucoTab@MobileCare — Time of day: Evening | 5/4/2018 15:26 — 99%

(c) Extended status bar with two new symbols.

Figure 7.7: New status bar symbols (c) were added, showing the current connection state to the glucose meter. A list of BLE devices (b) is shown when pressing the Bluetooth symbol. BLE devices can be permanently bonded (a) from the list of available devices.

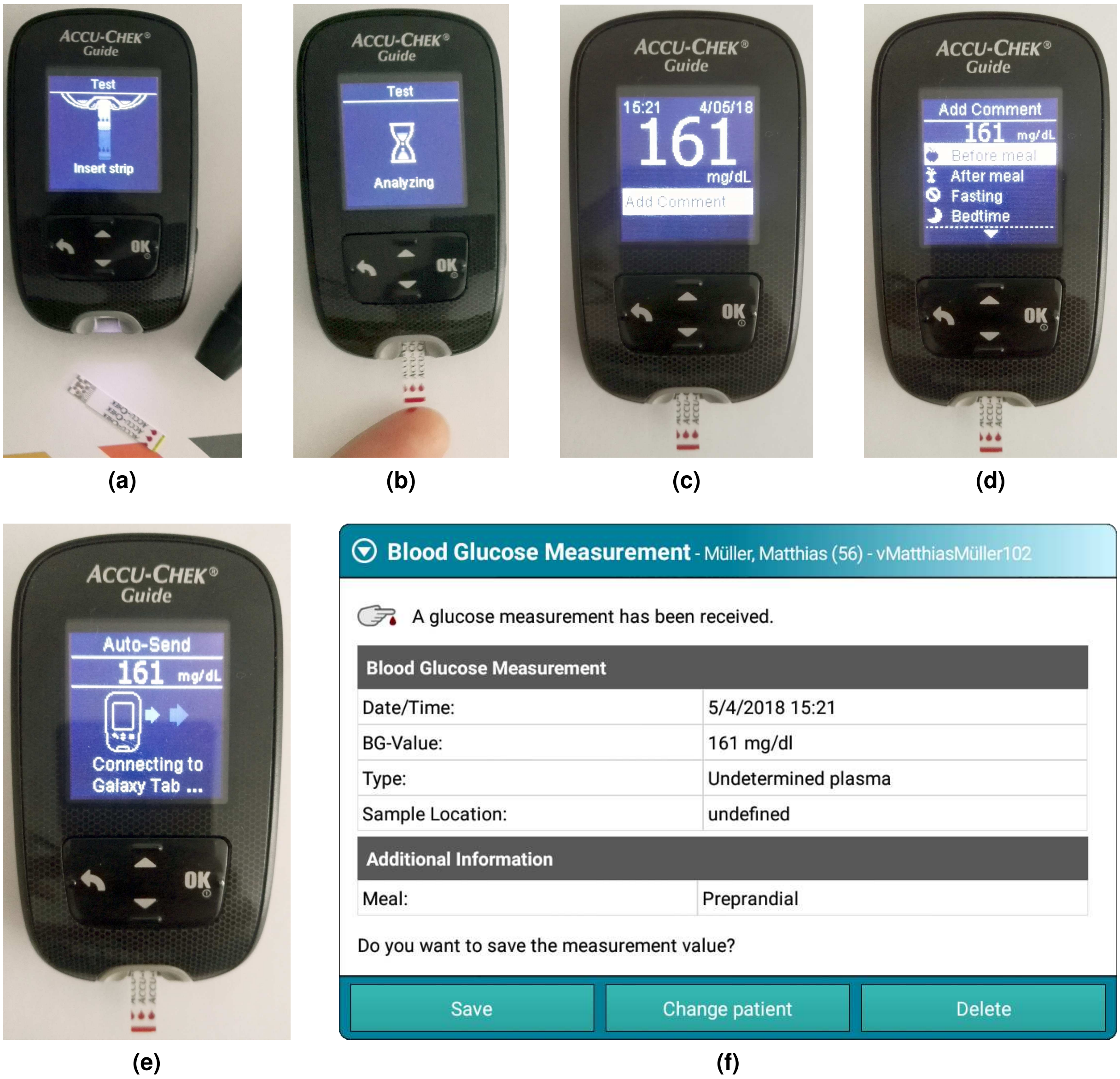
The Accu-Chek® Guide[4] glucose meter allows a bonding with up to five different BLE devices simultaneously. However, only one of the bonded devices can be selected for the automatic data transfer after measurements. Moreover, the glucose meter will only append a glucose measurement context

[3] `https://www.pchalliance.org/product-showcase`

[4] `https://www.accu-chek.com/meters/guide-meter`

characteristic to a glucose measurement if the context information is added by the user immediately after performing the measurement (see Figure 7.8c and Figure 7.8d). If 90 seconds expire without any interaction of the user, the Accu-Chek® Guide will initiate an automatic power off and start the BLE transmission *without* context information.

As soon as a glucose measurement record arrives at GlucoTab@MobileCare, the status bar symbol is updated and displays the number of records that were received during the transmission. Furthermore, either a short toast message or the detailed summary dialog is shown. Which of those two display options is performed depends on the current state and context of the application. If the user is currently performing any tasks within the application, only a short toast message including the measured glucose concentration is shown. However, if the user is not in the middle of a workflow, the detailed summary dialog with all the glucose measurement context information is displayed (see Figure 7.8f).



Blood Glucose Measurement - Müller, Matthias (56) - vMatthiasMüller102

A glucose measurement has been received.

Blood Glucose Measurement	
Date/Time:	5/4/2018 15:21
BG-Value:	161 mg/dl
Type:	Undetermined plasma
Sample Location:	undefined

Additional Information	
Meal:	Preprandial

Do you want to save the measurement value?

Figure 7.8: Workflow of performing (a, b) and annotating (c, d) a glucose measurement followed by an automated glucose measurement transmission to GlucoTab@MobileCare (e, f).

Finally, the available buttons at the bottom of the summary dialog vary depending on the application state and user context. If the user is *not logged in*, only a close button is available. In case that the user is *logged in, but no patient is selected*, a button to select a patient as well as a button to delete the glucose measurement record are shown. Finally, if the user is *logged in and a patient was previously selected* then three buttons are available allowing the user to change the patient, save the record, or delete the record.

7.6 Protocol and Implementation Remarks

The core component of the implemented protocol is the middleware. Currently, only the *Glucose Profile* was implemented and design decisions were made concentrating on the detailed glucose profile specification. Therefore the interface definition is not as generic as initially intended and would need to be adapted to support other health device profiles.

Compared to the blood pressure or weight scale profiles for instance, there are several similarities in the structure of a blood glucose measurement characteristic, a blood pressure measurement characteristic, and a weight measurement characteristic. Consequently, the `onReceiveData` callback in the listener interface could have been designed to receive a more generic and abstract `MeasurementRecord` instead of a specific `GlucoseMeasurement` object. Likewise, the `GlucoseMeter` could have been abstracted to a `HealthDevice` to indicate the general device information and the supported features as specified in the particular `GlucoseFeature`, `BloodPressureFeature`, and `WeightScaleFeature` characteristics.

The reliability of a measured blood glucose concentration steadily decreases as the elapsed time since the measurement increases. Therefore, an accurate system time is essential for glucose meters which are used by a DSS to calculate medication doses. To ensure a reasonable time accuracy the middleware was initially designed to implement the *Current Time Service* [35]. Consequently, the clocks of glucose meter and mobile device running the middleware could be synchronized at every encounter. However, the current implementation of the middleware does not support the Current Time Service.

Furthermore, the current implementation does not support a simultaneous connection of multiple glucose meters and all registered listeners receive the same data callbacks. This means that an application A might send a RACP request to the middleware, and an application B will likewise receive the response although application B did not issue any commands.

Admittedly, the protocol design is currently not entirely compliant to the Continua Design Guidelines, since a transcoding into the ISO/IEEE 11073-10101 Nomenclature is missing. Regarding the proof-of-concept showcase, a transcoding into the data structure used by GlucoTab@MobileCare was necessary either way and is therefore implemented on the listener side. However, glucose concentrations are forwarded by the middleware with the base units that are used in the Bluetooth specifications ($\frac{kg}{L} or \frac{mol}{L}$) instead of commonly used units in diabetes management ($\frac{mg}{dL} and \frac{mmol}{L}$). The *Glucose Service* [23] explicitly specifies the usage of *kilogram_per_litre* and *mole_per_litre*, although the more applicable units *milligram_per_decilitre* and *millimole_per_litre* are defined in the Bluetooth Assigned Numbers[5]. The Personal Connected Health Alliance [50] mentioned that currently the guidance of the communication *between different devices* has main priority and not communication between applications *within a single device*. Nevertheless, future versions of the guidelines might discuss a common middleware and interfaces for IPC.

Last but not least, some experiences with the Accu-Chek® Guide and the Android GATT API are worth mentioning. First of all, the glucose meter is generally made for glucose measurements at the finger and does transmit a *sample location*, however the value is *always undefined*. Secondly, measurements from a control solution are not shown in the logbook of the device, and yet they are considered and

[5] `https://www.bluetooth.com/specifications/assigned-numbers/units`

transferred by RACP commands. Thirdly, if 90 seconds elapse after a glucose measurement without additional user interaction, the Accu-Chek® Guide meter will dispatch the record via the auto-send functionality before powering off and it will not be possible to add a measurement context information afterwards anymore. Finally, attempts to pair and bond the Accu-Chek® Guide from Android's settings menu may result in an unexpected behavior: the glucose meter will show an unspecific connection error whereas the Android OS will report a successful connection. Turns out that the Accu-Chek® Guide will only accept pairing requests that are initiated with the `connectGatt` method of the GATT API.

On the subject of Android's GATT API, insight into a fragile, low-level communication framework with many pitfalls was gained. One and the same middleware implementation used with two different Accu-Chek® Guide devices resulted in a *reliable* glucose measurement characteristic notification on one device, and *no* glucose measurement characteristic notifications at all on the other device. The issue causing this behavior could not be identified for sure, but is likely to be explained by the asynchronous GATT communication in a multithreaded middleware software that is utilizing the Bluetooth stack via system services.

Chapter 8

Usability Testing

The thinking aloud method is a commonly used usability engineering method [36, 46]. Test users are asked to execute given tasks using the application that is to be tested. Along the way of solving the tasks, they shall speak their thoughts out loud. As a result, usability engineers can follow the user's chain of thoughts and identify problems and misconceptions in the application.

A summative usability evaluation of the medical device *GlucoTab@MobileCare* will be performed in a separate clinical study and is not within the scope of this work. However, usability testing with a thinking aloud test and a feedback questionnaire was performed to evaluate user satisfaction of the developed glucose measurement transmission protocol. This opportunity was taken to evaluate some of the modifications that were applied to GlucoTab@MobileCare (see chapter 5) as well.

8.1 Performed Usability Evaluation

The number of test users that are needed to efficiently find a majority of usability flaws is subject to a great deal of discussion in literature [36]. Since the proper number of users to at least identify a specified percentage of flaws strongly depends on the size of the application and on the actual number of flaws that can be found, there might be no *correct* answer to this question. Moreover, it can not be assured that all, not even most, of the flaws in an application will be found. Therefore, Lazar et al. [36] recommend to rather ask the questions of “how many users can we afford?”, “how many users can we get?”, and “how many users do we have time for?”.

Test users for the thinking aloud test were required to have a profound knowledge in the subject of diabetes mellitus and preferably already have experience with GlucoTab®. Due to these additional requirements, two HCPs and two GPs were kindly asked to participate in the usability tests. All four participants are professionals in the treatment of diabetics and had previous knowledge of GlucoTab®. Therefore, only a short explanation and training of the new features in GlucoTab@MobileCare was given before the actual thinking aloud test.

Three tasks were specified to test the usability of GlucoTab@MobileCare in combination with the developed automated glucose measurement transmission protocol:

1. In the first task, usability of the new task list layout was tested. User were told that they just started their working shift and the tablet was handed over from a colleague. Therefore, users were asked to use the GlucoTab@MobileCare application to identify the current time of the day (morning, midday, or evening) and name all patients with active tasks in the current time of the day. Finally, users should find out details of selected tasks which required them to navigate the task list.
2. Secondly, users were asked to log in as a nurse with given credentials and to perform a blood glucose measurement with the Accu-Chek® Guide glucose meter for a particular patient. A control solution

was provided in order to avoid the need of pricking with a lancet to extract blood. Subsequently, a medication administration should be carried out and on behalf of a given physician, the suggested insulin dose should be increased by four additional units (delegated order).

3. The last task concentrated on advanced functions that are only available for physicians, including the acknowledgement of a delegated order and an alteration of the frequency of blood glucose measurements.

After the thinking aloud test, a short feedback questionnaire was completed by the users. The questionnaire contained nine closed-ended questions, each with a Likert scale from 1 to 5 (*strongly agree* to *strongly disagree*). Furthermore, a blank space for textual justifications or remarks was provided after each question.

8.2 Results and Remarks

First of all, GlucoTab® users are required to complete a training before they may use the system on real patients in order to prevent faulty operation which could endanger patients. All four test users had at least some experience with the UI and workflow of GlucoTab®. However, a well-founded training on the new GlucoTab@MobileCare system was not feasible within the limited time of the test users. Consequently, some of the following usability flaws may not arise in a real world application with well trained and more experienced users.

Concerning the task list, one test user experienced minor difficulties when asked to read task details (first tried to tap on the patient name, then on the list heading symbol, and finally on the digits group). Another user revealed that it was not obvious how to open the task details dialog without prior instruction. Moreover, 3 out of 4 users struggled when they were asked to find out whether a specific task was carried out successfully or not; users were first unsure if the gray color means successful or unsuccessful and were afterwards unsure on how to interpret the remaining time in the task details dialog. Given these points, it would be advisable to further highlight *expired* tasks as well as *overdue* blood glucose profile tasks with red color or symbols in the task list.

Another usability flaw was observed by all four test users during the confirmation of delegated orders. All unconfirmed delegated orders are loaded and displayed together in a single dialog window. As already mentioned in subsection 5.3.3, a simplified implementation was used that doesn't make it possible to remove or add content entries without reloading the entire dialog. Furthermore, the dialog window handle is lost during the asynchronous web service request and would need to be stored in a variable that is accessible from each location where the task dialog is used. As a result, all four test users were confused when confirming a delegated order and only a toast message confirmation was shown as feedback instead of updating or closing the dialog.

The next point to note is that 3 out of 4 users had difficulties to spot where the blood glucose measurement frequency could be altered to the ongoing state. Popular assumed locations were the *blood glucose measurement* and the *physician tasks* menu items. However, users with a more in-depth knowledge of the algorithm used by the DSS should have no problem to associate a blood glucose measurement frequency with a therapy setting and therefore the *therapy adjustment* menu item.

As a final point on GlucoTab@MobileCare, users discovered one UI inconsistency and one design flaw. A dialog to select a start date for regular blood glucose profiles is provided by the Android OS with an inconsistent design (see Figure 8.1a). Moreover, Android's `DatePickerDialog` utilized on the used tablet displays the confirm button on the left side of the dialog, whereas all affirmation actions in GlucoTab® are consistently aligned to the right side. A further design consideration was outlined by a user in regard to the deactivated but selected check boxes (see Figure 8.1b). During *thinking aloud*, a user mentioned to "just leave all days *unchecked* as they are", although all check boxes were actually in the *checked* state.

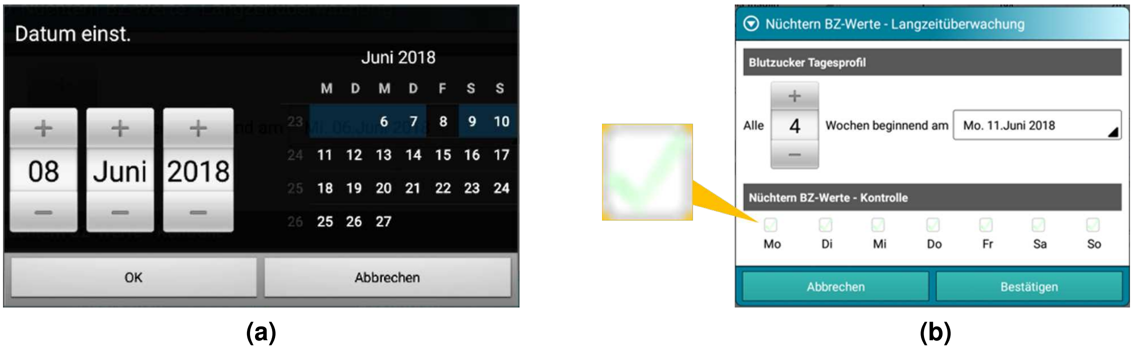


Figure 8.1: Design flaws and inconsistencies observed in GlucoTab@MobileCare.

Evaluation of the automated glucose measurement transmission protocol showed that 3 out of 4 test users would prefer (strongly agree) an *automated* over a *manual* measurement transmission in the future. The fourth user partly agreed and partly disagreed, with the reasoning that falling back to a manual transmission in case of a connection or setup error of the Bluetooth connection may lead to an interrupted workflow and consequently a loss of time. However, during usability test, *all* measurement transmissions that were initiated from the glucose meter were also successfully received by the Android application.

Nevertheless, the concern of a misconfigured Bluetooth connection is not completely unjustified, since the configuration of an auto send feature (glucose measurement characteristic notification) may vary on diverse glucose meters from different manufacturers. However, medical devices are required to include an *instruction for use* as specified by the European medical device regulation [52]. Therefore, a proper Bluetooth setup and configuration of the auto send feature should be reasonable for users with a capable glucose meter by following the instructions of the user manual.

During the second task, one user preferred to perform an actual *blood* glucose measurement instead of using the provided control solution. Since the process of disposing the used lancet and cleaning the puncture wound took longer than 90 seconds, insights on how the Accu-Chek® Guide glucose meter acts in such situations was gained. The glucose meter restrained further measurement context input and started to transmit the basic measurement record just before an automatic power off. This behavior leads to a single glucose measurement characteristic that is received by the middleware. Furthermore, no context information can be added afterwards from the Accu-Chek® Guide device. Despite a lack of information, this restriction makes the need to update an already transferred measurement record with retrospectively added context information unnecessary and no synchronization problems arise.

A further point to mention is that only 1 out of 4 users *strongly agreed* with the questionnaire statement that there were experienced *no difficulties or inconveniences* on initiating a measurement transmission from the Accu-Chek® Guide to the tablet. Although there were no critical difficulties, some of the reasons for a slightly declined user satisfaction were the previously described automatic power off as well as the fact that users were not familiar with the process of starting the Bluetooth transmission by pressing the OK button on the glucose meter. However, all test users agreed that they would not face the same difficulties again after once familiarizing with the workflow and the glucose meter.

“Good programmers use their brains,
but good guidelines save us having to think out every case.”
[Francis Glassborow]

Chapter 9

Outlook and Future Work

This chapter discusses general trends in CDSSs and the development of SaMD. Subsequently, ideas for future work on GlucoTab® and the automated glucose measurement transmission protocol are proposed.

9.1 General Trends

Early visions for computer-based systems in medicine consisted of decision-making systems that would *replace* physicians. However, the general trend quickly moved towards decision support systems to *assist* physicians. Human abilities to handle unexpected situations, process audio-visual data, incorporate personal priorities of a patient, or deal with social and ethic issues are remarkable. Currently there is no evidence that machines will ever possess the same capabilities that are used by physicians to make medical decisions. Nonetheless, CDSSs benefit from the increasing amount and availability of digital health data through electronic health records, mobile health devices, and the HIS. A *data-driven trend* is recognizable which incorporates machine learning capabilities and non-knowledge-based CDSSs [44]. Furthermore, mobile CDSSs that are used by non-clinicians or patients themselves are emerging due to a large availability of smartphones. Similarly, cloud computing enables new architectures and applications of CDSSs [4].

The Continua Design Guidelines[1] are applied by more and more medical device manufacturing companies. As a result, the general trend of glucose meters might move towards a standardized communication protocol and an interoperable communication method for mobile health applications. This could further increase the trend towards patient controlled, mobile CDSSs.

Security and reliability are crucial for medical devices. As discussed in chapter 4, the MDR comes into force in May 2020 and legislates requirements for risk and quality management. However, SaMD experts[2] have criticized a new classification rule which will likely result in most medical software to be classified as class II or class III. Accordingly, a certified QMS and the involvement of a notified body is required which will make it difficult for small companies and start ups to develop SaMD.

9.2 Ideas for Future Work

The algorithm of GlucoTab@MobileCare assumes that all blood glucose concentrations are fasting or preprandial measurements. However, in practice it will not always be possible to measure a preprandial blood glucose since HCPs might for example visit a patient just *after* a meal. This imposes the risk of

[1] https://www.pchalliance.org/product-showcase

[2] https://www.johner-institute.com/articles/regulatory-affairs/medical-device-regulation

a wrong dosage adjustment since the blood glucose concentration can be significantly affected [32] by a meal.

Therefore, a first suggestion for future work is to add an input field for the meal type (fasting, preprandial, postprandial, bedtime, and casual) into the blood glucose measurement view of GlucoTab@MobileCare. Subsequently a simple filter could be used to only consider fasting or preprandial measurements for the algorithm and therefore eliminate the risk of wrong dosage adjustments due to high values from postprandial blood glucose measurements.

Furthermore, the task list of GlucoTab@MobileCare could be reworked with a ViewHolder pattern (see subsection 5.4.3) to increase performance. Moreover, the web service method that is used to load the task list could be extended to only transmit *changes* in subsequent requests to refresh the task list. Aside from performance enhancements, the task list could use highlighting of expired tasks and overdue blood glucose profiles with colors or symbols as concluded from the usability evaluation.

Another recommendation based on the usability evaluation results is to rework the dialog that is used to confirm delegated orders. Users expected the dialog to update after confirmation of a delegated order as well as the dialog to close if no more orders need confirmation. Additionally, design consistency of button placements in dialog windows should be enforced and the distinction of unchecked and checked states of checkbox elements could be emphasized (revisit Figure 8.1).

In chapter 5 the planned design for delegation and long-term management was presented. However, both concepts were not implemented as planned to the whole extent. Therefore, future work could provide two distinct UI states for HCPs with and without physician approval during delegation (see subsection 5.3.2). Currently users receive an error message upon input submission telling them that a physician needs to be consulted. A client side UI state could be implemented such that users can only configure therapy settings which they are allowed to change. By selecting a physician for a delegated order, all input elements could then be unlocked. Similarly, the long-term tab could be implemented as planned (revisit subsection 5.5.1) in order to provide history lists for patient and therapy related values instead of just the current value.

Following general trends of mobile CDSSs that are used by patients, future work could consider a version of GlucoTab® that is used solely by the patient without the help of a clinician. Many patients could benefit from a personalized around-the-clock diabetes assistant on their smartphone providing medical decision support.

As another idea, the presented concept of an NFC authentication could be modified to a BLE authentication process. Android Smart Lock discontinued the support of trusted NFC devices but supports trusted *Bluetooth* devices to automatically unlock an Android device. Moreover, the FIDO UAF supports communication over BLE.

Finally, there are some enhancements that could be applied to the Android middleware of the automated glucose measurement transmission protocol as previously mentioned in section 7.6. First of all, a transcoding to the ISO/IEEE 11073-10101 Nomenclature should be added to ensure interoperability with arbitrary PHGs. Furthermore, implementing the *Current Time Service* would allow glucose meters to synchronize their clock regularly. Last but not least, a support of multiple, simultaneous glucose meter connections could be implemented as well as all optional commands of the RACP that can be used to retrieve measurement records.

Chapter 10

Conclusion

This work presented GlucoTab@MobileCare, a CDSS for the treatment of patients with type 2 diabetes mellitus that is used by HCPs. GlucoTab@MobileCare is a further development of GlucoTab® and moved from an inpatient care in hospitals to the *domiciliary nursing care* environment. Studies on GlucoTab® showed a significant reduction of hypoglycemia compared to paper-based processes [59] with a well accepted and efficacious system to maintain glycemic control [39, 45]. We are looking forward to achieving similar results with GlucoTab@MobileCare in the domiciliary nursing care.

An NFC based authentication method based on the UAF was investigated as a first improvement of the user experience in GlucoTab®. Therefore, an NFC tag is used to unlock the mobile device and a private key on the authenticator replaces an authentication with user name and password. This allows HCPs to focus on their therapy workflows without wasting time on recalling user name and password combinations as well as typing these credentials into the mobile device. However, detailed examination of NFC support by the Android OS revealed that trusted NFC devices to unlock the mobile device are not longer supported and trusted BLE devices are recommended instead. Moreover, preliminary tests showed that a very precise positioning of the NFC tag towards the reader was necessary with the user tablet in order to unlock the device. Therefore, the presented NFC authentication was not implemented. However, future work could adjust and implement the authentication process with BLE authenticators.

As a main result, an automated glucose measurement transmission protocol was presented and published as conference paper [43]. An Android middleware was implemented to handle communication between glucose meters and the Android device. Subsequently, applications can register as listeners for glucose measurement records and specify a callback function that is executed whenever a new glucose measurement is performed. Transmitted measurement records contain basic measurement information such as glucose concentration with a corresponding unit and a timestamp. Additionally, optional context information may be transmitted which contains information such as an amount of carbohydrates that have been consumed or whether the measurement was performed before or after a meal.

The automated glucose measurement transmission protocol was implemented based on the Continua Design Guidelines using the standardized Bluetooth GATT and is therefore compatible with any Continua certified glucose meter that supports Bluetooth. As a proof on concept, the protocol was integrated into GlucoTab@MobileCare and tested with the Continua certified Accu-Chek® Guide glucose meter. Despite minor problems, due to a neglected confirmation of the blood glucose measurement on the Accu-Chek® Guide glucose meter which starts the wireless communication, the performed usability evaluation showed that test users were satisfied and would in general prefer an automated over a manual measurement transmission. As a result, the risk of miscopied or misread glucose measurement values that are used for the calculation of medication doses could be eliminated.

“Any fool can write code that a computer can understand.
Good programmers write code that humans can understand.”
[Martin Fowler]

Appendix A

Conference Paper

A scientific paper that describes the automated glucose measurement transmission protocol was written and submitted by Meyer et al. [43] to the eHealth 2018 conference which had the convenient motto "Biomedical meets eHealth - From Sensors to Decisions". After acceptance of the manuscript, the work was presented at the conference in Vienna and published by the IOS Press. The published conference paper is included as appendix on the following pages.

Moreover, a second conference paper was composed by Frohner et al. [12] which describes a system of blood glucose telemonitoring using the automated glucose measurement transmission protocol. The manuscript was submitted to the DSAI 2018 conference and was accepted. However, at the time of writing the conference proceedings are not yet published by ACM and therefore the work could *not* be included as an appendix.

Health Informatics Meets eHealth
G. Schreier and D. Hayn (Eds.)

doi:10.3233/978-1-61499-858-7-132

Development of a Protocol for Automated Glucose Measurement Transmission Used in Clinical Decision Support Systems Based on the Continua Design Guidelines

Markus MEYER[a], Klaus DONSA[a,1], Thomas TRUSKALLER[a], Matthias FROHNER[b], Birgit POHN[b], Alexander FELFERNIG[c], Frank SINNER[a] and Thomas PIEBER[a,d]
[a]HEALTH - JOANNEUM RESEARCH Forschungsgesellschaft mbH, Graz, Austria
[b]University of Applied Sciences Technikum Wien, Vienna, Austria
[c]Graz University of Technology, Graz, Austria
[d]Medical University of Graz, Graz, Austria

Abstract Background: A fast and accurate data transmission from glucose meter to clinical decision support systems (CDSSs) is crucial for the management of type 2 diabetes mellitus since almost all therapeutic interventions are derived from glucose measurements. Objectives: Aim was to develop a prototype of an automated glucose measurement transmission protocol based on the Continua Design Guidelines and to embed the protocol into a CDSS used by healthcare professionals. Methods: A literature and market research was performed to analyze the state-of-the-art and thereupon develop, integrate and validate an automated glucose measurement transmission protocol in an iterative process. Results: Findings from literature and market research guided towards the development of a standardized glucose measurement transmission protocol using a middleware. The interface description to communicate with the glucose meter was illustrated and embedded into a CDSS. Conclusion: A prototype of an interoperable transmission of glucose measurements was developed and implemented in a CDSS presenting a promising way to reduce medication errors and improve user satisfaction.

Keywords. Mobile Health, Standardization, Clinical Decision Support Systems, Type 2 Diabetes Mellitus, Medication Errors.

1. Introduction

425 million people are suffering from diabetes mellitus worldwide and 58 million people in Europe with around 90% of type 2 diabetes mellitus (T2DM) according to recent estimates of the International Diabetes Federation (IDF) [1]. Glucose measurements and medication administration are important components of T2DM therapy.

Each patient needs a different amount of insulin which depends on many internal and external factors. General Practitioners (GPs) need comprehensive diabetes knowledge and experience to set a personalized insulin dosage for a patient by analyzing the logged glucose measurements. Since it is a time-consuming task to identify the cause of every Hypo- and Hyperglycemia, GPs can usually only focus on the most recent ones. A clinical decision support system (CDSS) can analyze large datasets of measurements

[1] Corresponding Author: Klaus Donsa, HEALTH - JOANNEUM RESEARCH Forschungsgesellschaft mbH, Neue Stiftingtalstraße 2, 8010 Graz, E-Mail: healthca@joanneum.at.

in a short time and can therefore assist GPs by suggesting a personalized medication dosage derived from all logged glucose measurements.

GlucoTab® is a CDSS which is already used in hospitals for the therapy of T2DM. Medication dosage suggestions are derived from glucose measurements as well as personalized therapy settings and follow a rule-based algorithm. The algorithm in place frequently updates the therapy settings based on new input data to fit the personal needs of the patient. GlucoTab® is currently operated by healthcare professionals (HCPs), however with the limitation that the glucose measurements are transferred manually into the CDSS.

The most important variable of a rule-based algorithm to calculate insulin dosages is the blood glucose level. A previously performed study [2] showed an error rate of 5% during manual transfer of measured glucose concentrations from the glucose meter in a paper-based workflow and an error rate of 4% in a computerized workflow. The study further revealed an increased probability of a hypoglycemic event following an insulin dosing error (odds ratio 3.1). Severe hypoglycemia is an indicator for poor patient outcomes and higher mortality risk and should therefore be avoided [3]. Preventing errors by the transfer of glucose measurements with an automated transmission protocol can therefore reduce the mortality rate.

Almost all therapeutic actions are derived from the measured glucose concentrations and therefore an accurate transmission of glucose measurements to other health-related devices such as CDSSs is desirable. Aim was the development of a prototype of an automated glucose measurement transmission protocol based on the Continua Design Guidelines. The protocol will be embedded into a mobile CDSS which will be used by healthcare professionals for the treatment of T2DM patients in the home care setting.

2. Methods

We performed a structured literature and market research on measurement transmission protocols to retrieve state-of-the-art implementations and to develop a prototype of an automated glucose measurement transmission protocol.

2.1. Literature and market research

The query “IEEE 11073 (Medical OR Health) Device” was used to search for publications about protocols, systems and devices which use the personal health device communication standard as defined by ISO/IEEE 11073. The query is applied to IEEE Xplore, ACM and PubMed Digital Library with a total of 99 distinct results. Based on title and abstract, 57 publications were identified as non-relevant for the research because they did not comprise the topic of a personal health device. Titles and abstracts of the 42 remaining relevant publications were examined and rated on a scale from 1 to 10 according to their relevance. This resulted in 21 relevant papers with a ranking of 5 or higher.

We used the google search engine and google play store to identify the state-of-the-art of glucose meters and smartphone applications related to automated measurement transmissions. Glucose meters are categorized by their data transport type (Bluetooth/ZigBee/USB/NFC) and whether they are Continua certified or not.

Table 1. Glucose meters, data transport and standardization status.

Glucose Meter	Data Transport	Continua Certified
Accu-Chek® Guide	Bluetooth/USB	Yes
Accu-Chek® Instant	Bluetooth/USB	Yes
Contour® Next/Plus ONE	Bluetooth/USB	Yes
FORA® D40	Bluetooth/USB	Yes
Accu-Chek® Active	USB	Yes
Accu-Chek® Mobile	USB	Yes
Accu-Chek® Aviva/Performa Insight	USB	Yes
Abbott FreeStyle Libre	NFC	No
Accu-Chek® Aviva/Performa Connect	Bluetooth/USB	No
AgaMatrix Jazz Wireless 2	Bluetooth	No
Beurer GL 50 evo	Bluetooth/USB	No
BodyTel® GlucoTel	Bluetooth	No
Dexcom G5	Bluetooth	No
FORA® TN'G / TN'G Voice	Bluetooth	No
GlucoMen Areo / Areo 2K	Bluetooth/NFC/USB	No
MediTouch® 2 connect	Bluetooth/USB	No
Medtronic Enlite® Sensor	Bluetooth	No
OneTouch Verio Flex®	Bluetooth/USB	No

Applications are categorized by features like reminders and bolus calculators, as well as whether they are using a standardized or a proprietary communication protocol.

2.2. Development of a prototype of an automated glucose measurement transmission protocol

Results from the literature and market research were the basis for the development of an automated glucose measurement transmission protocol from glucose meters to CDSSs. Literature research highlighted the benefits of standardized communication protocols which confirmed the development following the Continua Design Guidelines. Market research revealed a lack of glucose meters using a standardized communication protocol and was resolved by a middleware for the communication with glucose meters which can be extended to translate non-standardized messages into standardized messages.

The system was designed to transfer the measurements from an Accu-Chek® Guide glucose meter via Bluetooth Low Energy to a middleware running on Android. The middleware shall then provide an interface to other applications and allow them to receive and read measured glucose concentrations. The CDSS GlucoTab® implements the interface to the middleware and is thereby able to make therapy decisions from the sensor readings in real-time.

3. Results

Results from literature and market research were used as basis for development of a standardized glucose measurement transmission protocol using an extendable middleware to allow the support of a broad range of glucose meters.

3.1. Literature and market research

Market research revealed that at least 18 glucose meters which can transfer a measured glucose value to another device for further examination exist. However, only 7 out of the

18 listed devices use the standardized ISO/IEEE 11073-20601 PHD exchange protocol [4] and ISO/IEEE 11073-10417 glucose meter device specialization [5], as suggested by the Continua Design Guidelines (Table 1). Moreover only 4 of them support a wireless Bluetooth communication which increases usability in a mobile healthcare setting.

Market research on Android applications which are used to receive measurement values from a glucose meter showed a lack of standardization likewise in software applications and in personal health devices. Four out of 12 listed applications support the standardized protocol as defined by the Continua Design Guidelines (Table 2). Market research revealed further that most applications can only receive, store and visualize the measurement values, but they neither help the user with the medication dosage calculation nor with reminders for glucose measurements or medication administration.

Accu-Chek® Connect and mySugr provide a bolus calculator. The application thereby suggests an insulin dosage for a meal, based on carbohydrates, measured blood glucose and some personalized therapy settings, such as the amount of insulin needed per gram carbohydrate. GlucoTab® takes this approach one step further and manages the entire diabetes therapy. The application tells the user when and how often a glucose measurement should be performed and calculates an appropriate medication dosage several times per day.

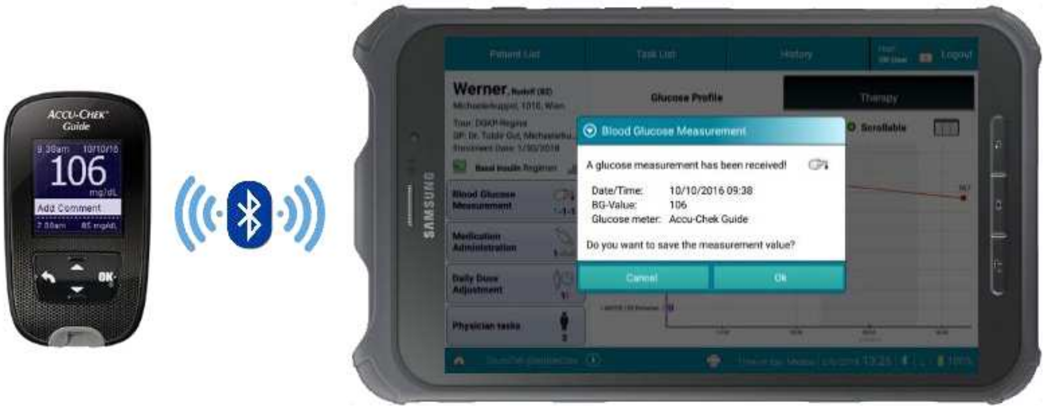
3.2. Development of a prototype of an automated glucose measurement transmission protocol

The developed glucose measurement transmission protocol was designed and implemented according to the Continua Design Guidelines [6]. These guidelines define, beside other things, what has to be considered when implementing a Bluetooth Low Energy interface between a personal health device (the Accu-Chek® glucose meter) and a personal health gateway (the tablet running GlucoTab®). This specification is strongly aligned with the specification from the Bluetooth Special Interest Group. An end use application of the protocol was embedded into GlucoTab® (Figure 1).

GlucoTab® is installed together with a middleware on the used tablet. The middleware was implemented as a background service (not having an own graphical user interface) and provides the needed functionalities to connect and communicate with the used Accu-Chek® glucose meter on the one and GlucoTab® on the other hand.

Table 2. Feature, protocol and user comparison of diabetes related android applications.

Application	Logbook	Medication Calculator	Therapy Adjustment	Reminder	Continua Protocol	User
GlucoTab®	Yes	Yes	Yes	Yes	Yes	HCP
mySugr	Yes	Yes	No	Yes	Yes	P
Accu-Chek® Connect	Yes	Yes	No	No	Yes	P
Contour® Diabetes	Yes	No	No	Yes	Yes	P
AgaMatrix Diabetes Manager	Yes	No	No	Yes	No	P
Beurer HealthManager	Yes	No	No	No	No	P
BodyTel Blutzucker	Yes	No	No	No	No	P
Glucolog Lite/Mobile	Yes	No	No	No	No	P
iFORA Diabetes Manager	Yes	No	No	No	No	P
LibreLink	Yes	No	No	No	No	P
OneTouch Reveal®	Yes	No	No	No	No	P
VitaDock+	Yes	No	No	No	No	P



Blood Glucose Measurement

A glucose measurement has been received!

Date/Time: 10/10/2016 09:38
BG-Value: 106
Glucose meter: Accu-Chek Guide

Do you want to save the measurement value?

Figure 1. Transmission from Accu-Chek® Guide glucose meter to GlucoTab® CDSS.

Communication between middleware and GlucoTab® were defined through Android Interface Definition Language (AIDL). For the information that is exchanged based on the data objects (Figure 2) stated in the interfaces' definition, further efforts towards standardization were made by using the terminology defined in ISO/IEEE 11073-10417[5], i.e. the terms and/or code as stated in this standard were used to identify the measured parameters and meta data. Following this standard, information like measuring the glucose concentration based on an capillary whole-blood sample or meta information like that measurement has been taken pre-prandial was coded as the integer values 23112 and 29260 respectively.

The middleware, acting as a glucose meter manager, implements an IGlucoseMeterManager interface (Figure 3) and handles communication with the glucose meter as well as with GlucoTab®. Android's build-in platform support for BLE, which is available since Android 4.3, is used to read the services provided by a remote BLE device.

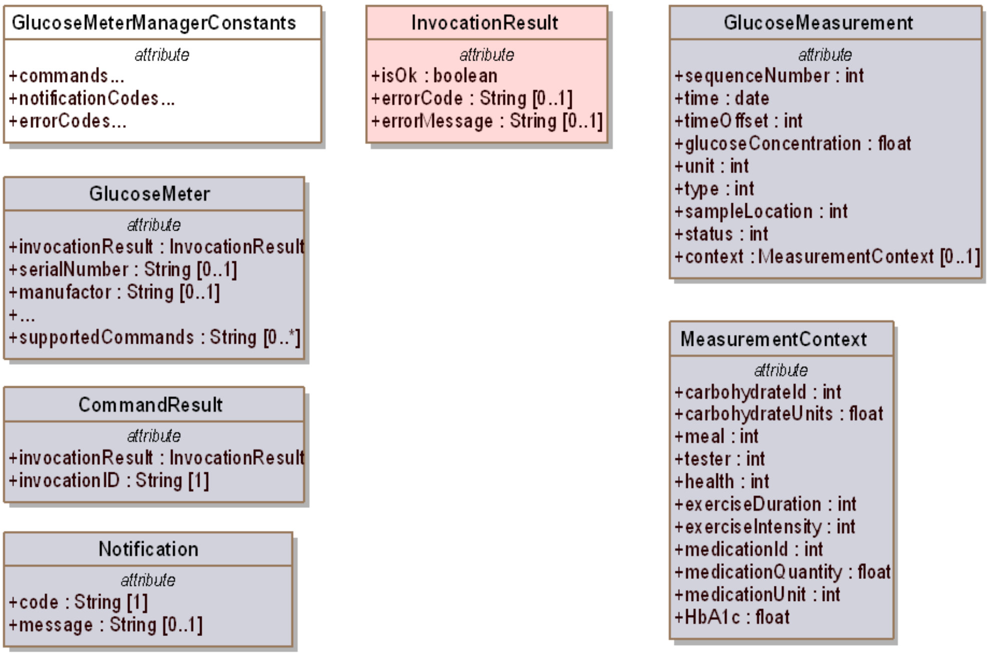


Figure 2. Data objects exchanged by middleware and GlucoTab®.

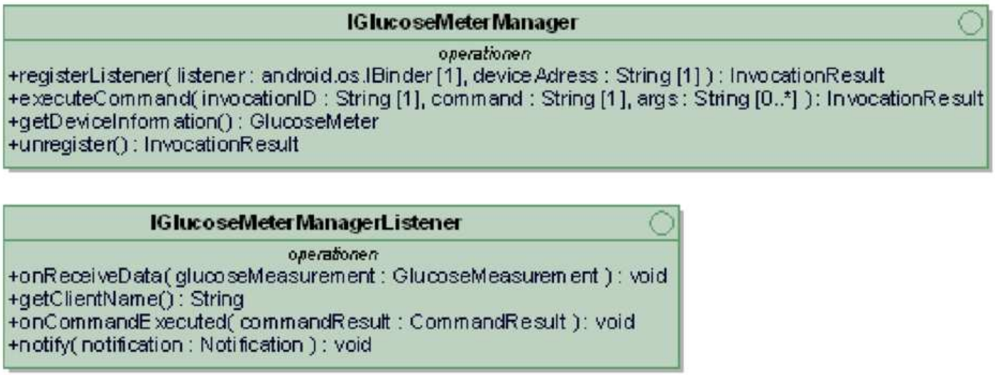

IGlucoseMeterManager
operationen
+registerListener(listener : android.os.IBinder [1], deviceAdress : String [1]) : InvocationResult
+executeCommand(invocationID : String [1], command : String [1], args : String [0..*]) : InvocationResult
+getDeviceInformation() : GlucoseMeter
+unregister() : InvocationResult
IGlucoseMeterManagerListener
operationen
+onReceiveData(glucoseMeasurement : GlucoseMeasurement) : void
+getClientName() : String
+onCommandExecuted(commandResult : CommandResult) : void
+notify(notification : Notification) : void

Figure 3. AIDL interface definition between middleware and GlucoTab®.

GlucoTab® implements the IGlucoseMeterManagerListener interface and registers as listener at the middleware by calling the registerListener method. This method holds the unique MAC address of the Bluetooth device to which the connection should be established. After the middleware has established the connection the "Record Access Control Point" service characteristic is used to query actively for stored values. This method also enables the listener to delete stored measurement values on the glucose meter. Methods of the IGlucoseMeterManager interface return a lightweight InvocationResult object to identify errors and supply error details with an error code and a short error message.

The GlucoseMeter class returned by the getDeviceInformation method contains general information about a connected glucose meter and additionally provides information about supported commands by the health device. GlucoseMeasurement is used to provide general information of a glucose measurement, such as the date and time and the glucose concentration. MeasurementContext can be used to give further details about a glucose measurement, such as exercise, meal and medication information related to the measurement when supported by the used blood glucose meter.

4. Discussion

Literature and market research substantiate the need for standardization in communication protocols used by personal health devices. As mentioned in [7-15], a standardized communication protocol for personal health devices enables a seamless plug and play compatibility of various sensors from different manufacturers. However, as mentioned by [8,11,13,15], manufacturers use their own software and communication protocols, building proprietary solutions that can only work alone or inside a single-vendor system. Proprietary protocol solutions eliminate the communication between devices of different manufacturers, leading to an interoperability problem.

One reason for the lack of standardization is the gap between current regulations as well as health policies of medical devices and stakeholders manufacturing the personal health technology. Insufficient or over regulation of health standards can both significantly delay the market adoption [12]. A further reason is the vast amount of pages in standard documents that need to be observed [13] and the overly complex design of the protocols [16]. Nevertheless, advantages of a standardized protocol are the interoperability of devices from different manufacturers, lower healthcare costs and a better patient treatment [11-13]. Moreover, there are tools and frameworks provided to help overcome the difficulties of implementing a standardized communication protocol [13,17].

This article presents a standardized implementation of an automated glucose measurement transmission protocol from glucose meters to the CDSS GlucoTab®. By following the Continua Design Guidelines, future Continua certified glucose meters will be able to communicate out of the box with the presented implementation. Beside the implemented feature to query data using the "Record Access Control Point" characteristic another approach would be to have the data transferred automatically using the indication service, as described by the "Glucose Measurements" service characteristic specification [18]. Concerning the different possibilities how to acquire the data from the Bluetooth device, initiated by the device versus initiated by the middleware or GlucoTab®, at this point in time the decision was made for the latter.

At the first glance an automatic transfer of the data seems more desirable, but for the workflow of HCPs it seems more usable when they can actively pull the values. Therefore the "Record Access Control Point" service characteristic has been implemented and used to get and to delete data from the Bluetooth device. However, another possible way for the same user experience would be that the middleware stores and forwards the data received from the glucose meter. This approach has the disadvantage that data might end up in the middleware and might not be requested from GlucoTab®.

By embedding the automated glucose measurement transmission protocol into GlucoTab® and thereby eliminating the risk of miscopied or misread glucose measurement values, the treatment of T2DM patients by healthcare professionals was further improved. After measuring the glucose concentration of a patient, the value will automatically be transmitted to GlucoTab®. The clinical decision support system can immediately feed the measured value into an algorithm to calculate an adapted medication dosage based on the current glucose level.

Further research including a summative usability assessment of the automated transmission protocol using GlucoTab® in a working environment with healthcare professionals is already planned.

In conclusion a prototype of an automated glucose measurement transmission protocol was developed and embedded into GlucoTab®. Since the glucose measurement is the source of the algorithm behind GlucoTab® a reliable and automated transmission of this data helps to reduce medication errors and to assist HCPs on routine tasks.

References

[1] I. D. Federation, IDF Diabetes Atlas, Eighth edition 2017, 8th ed. 2017.

[2] K. Donsa et al., "Impact of errors in paper-based and computerized diabetes management with decision support for hospitalized patients with type 2 diabetes. A post-hoc analysis of a before and after study," Int. J. Med. Inform., vol. published, no. http://dx.doi.org/10.1016/j.ijmedinf.2016.03.007, 2016.

[3] A. K. Lee et al., "The Association of Severe Hypoglycemia With Incident Cardiovascular Events and Mortality in Adults With Type 2 Diabetes.," Diabetes Care, vol. 41, no. 1, pp. 104–111, Jan. 2018.

[4] IEEE 11073 Committee, IEEE 20601 - Health informatics—Personal health device communication: Application profile — Optimized Exchange Protocol, vol. 2014. 2014.

[5] I. 11073TM S. Committee, "IEEE 11073-10417-2015 - Health informatics -- Personal health device communication Part 10417: Device Specialization -- Glucose Meter." p. 78, 2015.

[6] Personal Connected Health Alliance, "H.811 Personal Health Devices Interface Design Guidelines," p. 114, 2016.

[7] H. S. Park, H. Cho, and H. S. Kim, “Development of cell phone application for blood glucose self-monitoring based on ISO/IEEE 11073 and HL7 CCD,” Healthc. Inform. Res., vol. 21, no. 2, pp. 83–94, 2015.

[8] J. H. Lim, C. Park, and S. J. Park, “Home healthcare settop-box for senior chronic care using ISO/IEEE 11073 PHD standard,” 2010 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC’10, pp. 216–219, 2010.

[9] M. Clarke, S. Member, J. De Folter, C. Palmer, and V. Verma, “Building Point of Care Health Technologies on the,” pp. 16–18, 2013.

[10] A. Aragüés et al., “Trends and challenges of the emerging technologies toward interoperability and standardization in e-health communications,” IEEE Commun. Mag., vol. 49, no. 11, pp. 182–188, 2011.

[11] I. Martińez et al., “Standard-based middleware platform for medical sensor networks and u-health,” Proc. - Int. Conf. Comput. Commun. Networks, ICCCN, pp. 714–719, 2008.

[12] D. Zhong, M. J. Kirwan, and X. Duan, “Regulatory barriers blocking standardization of interoperability,” J. Med. Internet Res., vol. 15, no. 7, 2013.

[13] J. H. Lim, C. Park, S. J. Park, and K. C. Lee, “ISO/IEEE 11073 PHD message generation toolkit to standardize healthcare device,” Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, pp. 1161–1164, 2011.

[14] H. L. Li, Z. Bin Duan, J. Z. Cui, and Z. W. Chen, “A design of general medical data adapter based on ISO/IEEE 11073 standards,” 2015 12th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. ICCWAMTIP 2015, pp. 404–407, 2016.

[15] J. C. Nam, W. K. Seo, J. S. Bae, and Y. Z. Cho, “Design and development of a u-Health system based on the ISO/IEEE 11073 PHD standards,” 17th Asia-Pacific Conf. Commun. APCC 2011, no. October, pp. 789–793, 2011.

[16] M. Clarke, J. de Folter, V. Verma, and H. Gokalp, “Interoperable End-to-End Remote Patient Monitoring Platform based on IEEE 11073 PHD and ZigBee Health Care Profile,” IEEE Trans. Biomed. Eng., vol. 1, no. c, 2017.

[17] L. Yu, Y. Lei, R. N. Kacker, D. R. Kuhn, R. D. Sriram, and K. Brady, “A general conformance testing framework for IEEE 11073 PHD’s communication model,” Proc. 6th Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA ’13, pp. 1–8, 2013.

[18] B. SIG, “Bluetooth GATT Characteristic: Glucose Measurement,” 2012. [Online]. Available: https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.characteristic.glucose_measurement.xml. [Accessed: 25-Jan-2018].

“Good code is its own best documentation.”

[Steve McConnell]

Bibliography

[1] V. S. Ajay, D. Jindal, A. Roy, V. Venugopal, R. Sharma, A. Pawar, S. Kinra, N. Tandon, and D. Prabhakaran. "Development of a Smartphone-Enabled Hypertension and Diabetes Mellitus Management Package to Facilitate Evidence-Based Care Delivery in Primary Healthcare Facilities in India: The mPower Heart Project". In: *Journal of the American Heart Association: Cardiovascular and Cerebrovascular Disease* 5 (12 Dec. 2016). DOI: 10.1161/JAHA.116.004343 (cited on pages 1, 5).

[2] M. Alther and C. K. Reddy. "Clinical Decision Support Systems". In: *Healthcare Data Analytics*. Chapman & Hall/CRC, 2015. Chapter 19, pages 625–656. ISBN: 978-1-482-23211-0 (cited on pages 7–8).

[3] L. M. Beltrán, A. Olmo-Fernández, J. R. Banegas, and J. García-Puig. "Electronic clinical decision support system and multifactorial risk factor control in patients with type 2 diabetes in primary health care". In: *European Journal of Internal Medicine* (Oct. 2017), pages 35–37. ISSN: 0953-6205. DOI: 10.1016/j.ejim.2017.07.024 (cited on pages 1, 5).

[4] E. S. Berner. *Clinical Decision Support Systems: Theory and Practice*. 3rd edition. Springer International Publishing, 2016. ISBN: 978-3-319-31913-1. DOI: 10.1007/978-3-319-31913-1 (cited on pages 7–8, 49).

[5] Bluetooth SIG. *Bluetooth® Core Specification*. Version 4.2. Dec. 2, 2014. `https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043` (cited on pages 34–35, 41).

[6] X. Chen, X. Bao, Z. Fang, and S. Xia. "Design and development of a ubiquitous healthcare monitoring system based on Android platform and ISO/IEEE 11073 standards". In: *2016 IEEE International Conference on Information and Automation (ICIA)*. Aug. 2016, pages 1165–1168. DOI: 10.1109/ICInfA.2016.7831995 (cited on page 32).

[7] M. Clarke, J. de Folter, V. Verma, and H. Gokalp. "Interoperable End-to-End Remote Patient Monitoring Platform Based on IEEE 11073 PHD and ZigBee Health Care Profile". In: *IEEE Transactions on Biomedical Engineering* 65.5 (May 2018), pages 1014–1025. ISSN: 0018-9294. DOI: 10.1109/TBME.2017.2732501 (cited on page 32).

[8] R. DeFronzo, E. Ferrannini, P. Zimmet, and G. Alberti. *International Textbook of Diabetes Mellitus*. Wiley, 2015. ISBN: 978-1-1183-8767-2 (cited on pages 3–4, 19).

[9] V. M. Deshmukh and G. R. Bamnote. "An Empirical Study of XML Parsers across Applications". In: *2015 International Conference on Computing Communication Control and Automation*. Feb. 2015, pages 396–401. DOI: 10.1109/ICCUBEA.2015.83 (cited on page 24).

[10] "Diabetes mellitus - Anleitungen für die Praxis". In: *Wiener klinische Wochenschrift* 128.2 (Apr. 2016), pages 37–228. ISSN: 1613-7671 (cited on page 4).

[11] K. Donsa, P. Beck, B. Höll, J. K. Mader, L. Schaupp, J. Plank, K. M. Neubauer, C. Baumgartner, and T. R. Pieber. "Impact of errors in paper-based and computerized diabetes management with decision support for hospitalized patients with type 2 diabetes. A post-hoc analysis of a before and

after study". In: *International Journal of Medical Informatics* 90 (June 2016), pages 58–67. ISSN: 1386-5056. DOI: 10.1016/j.ijmedinf.2016.03.007 (cited on pages 15, 31).

[12] M. Frohner, M. Meyer, K. Donsa, P. Urbauer, V. David, and S. Sauermann. "Telemonitoring of Blood Glucose". In: *Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*. DSAI 2018. New York, NY, USA: ACM, 2018. ISBN: 978-1-4503-6467-6 (cited on pages 31, 53).

[13] "Global Guideline for Type 2 Diabetes". In: *Diabetes Research and Clinical Practice* 104.1 (Apr. 2014), pages 1–52. ISSN: 0168-8227. DOI: 10.1016/j.diabres.2012.10.001 (cited on page 4).

[14] R. G. Gómez, R. D. Hughes, D. Bogia, K. Shingala, L. Kermes, M. Lima, R. Herbster, E. Pfützenreuter, L. Ott, J. R. Barr, G. Schatz, R. Strickland, B. Reinhold, M. Clarke, N. Hamming, M. Yeung, W. Heck, and J. Hartmann. *Personal Health Devices Transcoding White Paper*. Version 1.6. Oct. 28, 2015. https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=310657 (cited on page 33).

[15] Google. *Android API Level 22 Reference*. May 2018. https://developer.android.com/reference (cited on pages 15, 24, 36).

[16] Google. *Android Developer Guides*. May 2018. https://developer.android.com/guide (cited on pages 37–38).

[17] M. Harbach, A. De Luca, and S. Egelman. "The Anatomy of Smartphone Unlocking: A Field Study of Android Lock Screens". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pages 4806–4817. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858267 (cited on page 29).

[18] M. Harbach, E. von Zezschwitz, A. Fichtner, A. D. Luca, and M. Smith. "It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception". In: *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. Menlo Park, CA: USENIX Association, 2014, pages 213–230. ISBN: 978-1-931971-13-3 (cited on page 29).

[19] R. Heydon, R. Hughes, K. Shingala, M. Lima, J. Hillyard, and L. Ott. *Device Information Service*. Version 1.1. Nov. 29, 2011. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=244369 (cited on page 35).

[20] C.-K. Hsieh, H. Falaki, N. Ramanathan, H. Tangmunarunkit, and D. Estrin. "Performance Evaluation of Android IPC for Continuous Sensing Applications". In: *SIGMOBILE Mobile Computing and Communications Review* 16.4 (Feb. 2013), pages 6–7. ISSN: 1559-1662. DOI: 10.1145/2436196.2436200 (cited on page 38).

[21] M. M. Hugh, O. Cawley, F. McCaffcry, I. Richardson, and X. Wang. "An agile V-model for medical device software development to overcome the challenges with plan-driven software development lifecycles". In: *2013 5th International Workshop on Software Engineering in Health Care (SEHC)*. May 2013, pages 12–19. DOI: 10.1109/SEHC.2013.6602471 (cited on page 12).

[22] R. Hughes, B. Krishnamoorthy, R. Strickland, and T. Cook. *Glucose Profile*. Version 1.0. Apr. 3, 2012. https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=248025 (cited on pages 31, 33–35, 43).

[23] R. Hughes, B. Krishnamoorthy, R. Strickland, and T. Cook. *Glucose Service*. Version 1.0. Apr. 3, 2012. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=248026 (cited on pages 35–36, 43).

[24] *IEC 62304:2006/Amd 1:2015 Medical Device Software — Software Life Cycle Processes*. June 2015. https://www.iso.org/standard/64686.html (cited on pages 1, 9, 11–12).

[25] *IEC 62366-1:2015 Medical Devices — Part 1: Application of Usability Engineering to Medical Devices*. Feb. 2015. https://www.iso.org/standard/63179.html (cited on pages 1, 9, 12).

[26] *IEC 82304-1:2016 Health software — Part 1: General Requirements for Product Safety*. Oct. 2016. https://www.iso.org/standard/59543.html (cited on pages 1, 9).

[27] International Diabetes Federation. *IDF Diabetes Atlas*. 8th edition. Brussels, Belgium: International Diabetes Federation, 2017. http://www.diabetesatlas.org (cited on pages 1, 3).

[28] *ISO 13485:2016 Medical Devices — Quality Management Systems — Requirements for Regulatory Purposes*. Mar. 2016. https://www.iso.org/standard/59752.html (cited on pages 1, 9–10).

[29] *ISO 14971:2007 Medical Devices — Application of Risk Management to Medical Devices*. Mar. 2007. https://www.iso.org/standard/38193.html (cited on pages 1, 9–10).

[30] *ISO/IEEE 11073-10101: Health Informatics — Point-Of-Care Medical Device Communication - Part 10101: Nomenclature*. 2004. DOI: 10.1109/IEEESTD.2004.95741 (cited on pages 33–34).

[31] *ISO/IEEE 11073-10101a: Health Informatics — Point-Of-Care Medical Device Communication - Part 10101: Nomenclature – Amendment 1: Additional Definitions*. Dec. 2015. DOI: 10.1109/IEEESTD.2015.7350198 (cited on page 33).

[32] *ISO/IEEE 11073-10417: Health Informatics — Personal Health Device Communication - Part 10417: Device Specialization – Glucose Meter*. July 2015. DOI: 10.1109/IEEESTD.2015.7134655 (cited on pages 34, 50).

[33] H. W. Kang, C. M. Kim, and S. J. Koh. "ISO/IEEE 11073-Based Healthcare Services over IoT Platform Using 6LoWPAN and BLE: Architecture and Experimentation". In: *2016 International Conference on Networking and Network Applications (NaNA)*. July 2016, pages 313–318. DOI: 10.1109/NaNA.2016.26 (cited on page 32).

[34] J. Kim and O. Song. "ISO/IEEE 11073 interoperability for personal health devices based on ZigBee healthcare service". In: *2015 IEEE International Conference on Consumer Electronics (ICCE)*. Jan. 2015, pages 263–264. DOI: 10.1109/ICCE.2015.7066406 (cited on page 32).

[35] M. Kirwan, S. Michitsuta, S. Nagashima, N. Fukushima, D. Matsuoh, T. Arai, R. Heydon, E. Fleury, R. Galli, T. Kimura, S. Koyama, S. Oshiyama, R. Hughes, A. Kelur, K. Shingala, D. Sadler, K. Jachim, K. Kerai, J. Salokannel, F. Berntsen, N. Granquist, B. Redding, and G. Goyal. *Current Time Service*. Version 1.1. Oct. 7, 2014. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=292957 (cited on page 43).

[36] J. Lazar, J. H. Feng, and H. Hochheiser. *Research Methods in Human Computer Interaction*. 2nd edition. Boston, MA, USA: Morgan Kaufmann, 2017. ISBN: 978-0-12-805390-4. DOI: 10.1016/B978-0-12-805390-4.00010-8 (cited on page 45).

[37] H.-L. Li, Z.-B. Duan, J.-Z. Cui, and Z.-W. Chen. "A design of general medical data adapter based on ISO/IEEE 11073 standards". In: *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. Dec. 2015, pages 404–407. DOI: 10.1109/ICCWAMTIP.2015.7494019 (cited on page 32).

[38] S. Machani, R. Philpott, S. Srinivas, J. Kemp, and J. Hodges. *FIDO UAF Architectural Overview*. Version 1.1. Feb. 2, 2017. https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/fido-uaf-overview-v1.1-ps-20170202.pdf (cited on page 29).

[39] J. K. Mader, K. M. Neubauer, L. Schaupp, T. Augustin, P. Beck, S. Spat, B. Höll, G. M. Treiber, F. M. Fruhwald, T. R. Pieber, and J. Plank. "Efficacy, usability and sequence of operations of a workflow-integrated algorithm for basal–bolus insulin therapy in hospitalized type 2 diabetes

patients". In: *Diabetes, Obesity and Metabolism* 16.2 (), pages 137–146. DOI: 10.1111/dom.12186 (cited on pages 13, 51).

[40] E. Manas and D. Grancini. *Android High Performance Programming*. Packt Publishing, 2016. ISBN: 978-1-7852-8338-3. `https://books.google.at/books?id=aaXWDQAAQBAJ` (cited on pages 24, 37).

[41] I. Martinez, J. Escayola, M. Martinez-Espronceda, L. Serrano, J. Trigo, S. Led, and J. Garcia. "Standard-Based Middleware Platform for Medical Sensor Networks and u-Health". In: *2008 Proceedings of 17th International Conference on Computer Communications and Networks*. Aug. 2008, pages 1–6. DOI: 10.1109/ICCCN.2008.ECP.135 (cited on page 32).

[42] M. McHugh, F. McCaffery, and V. Casey. "Barriers to Adopting Agile Practices When Developing Medical Device Software". In: *Software Process Improvement and Capability Determination*. Edited by A. Mas, A. Mesquida, T. Rout, R. V. O'Connor, and A. Dorling. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pages 141–147. ISBN: 978-3-642-30439-2 (cited on page 12).

[43] M. Meyer, K. Donsa, T. Truskaller, M. Frohner, B. Pohn, A. Felfernig, F. Sinner, and T. R. Pieber. "Development of a Protocol for Automated Glucose Measurement Transmission Used in Clinical Decision Support Systems Based on the Continua Design Guidelines". In: *Health Informatics Meets eHealth*. Volume 248. Studies in Health Technology and Informatics. 2018, pages 132–139. DOI: 10.3233/978-1-61499-858-7-132 (cited on pages 2, 31, 51, 53).

[44] M. A. Musen, B. Middleton, and R. A. Greenes. "Clinical Decision-Support Systems". In: *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. Edited by E. H. Shortliffe and J. J. Cimino. London: Springer London, 2014, pages 643–674. ISBN: 978-1-4471-4474-8. DOI: 10.1007/978-1-4471-4474-8_22 (cited on pages 1, 7, 49).

[45] K. M. Neubauer, J. K. Mader, B. Höll, F. Aberer, K. Donsa, T. Augustin, L. Schaupp, S. Spat, P. Beck, F. M. Fruhwald, C. Schnedl, A. R. Rosenkranz, D. B. Lumenta, L. P. Kamolz, J. Plank, and T. R. Pieber. "Standardized Glycemic Management with a Computerized Workflow and Decision Support System for Hospitalized Patients with Type 2 Diabetes on Different Wards". In: *Diabetes Technology & Therapeutics* 17.10 (Oct. 2015), pages 685–692. ISSN: 1557-8593. DOI: 10.1089/dia.2015.0027 (cited on pages 1, 4, 13, 25, 51).

[46] J. Nielsen. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 0-125-18405-0 (cited on pages 22–23, 45).

[47] J. Nielsen and R. Budiu. *Mobile Usability*. Berkeley, CA, USA: New Riders, Oct. 24, 2012. ISBN: 978-0-3218-8448-0 (cited on page 29).

[48] Oracle. *Java™ Platform Standard Edition 7 API Specification*. June 25, 2018. `https://docs.oracle.com/javase/7/docs` (cited on page 23).

[49] Personal Connected Health Alliance. *H.810 Introduction – Interoperability Design Guidelines for Personal Connected Health Systems*. Dec. 29, 2017. `https://www.pchalliance.org/continua-design-guidelines` (cited on pages 32, 34).

[50] Personal Connected Health Alliance. *H.811 Personal Health Devices Interface design guidelines*. Dec. 29, 2017. `https://www.pchalliance.org/continua-design-guidelines` (cited on pages 33, 43).

[51] L. Poretsky. *Principles of Diabetes Mellitus*. Principles of Diabetes Mellitus. Springer International Publishing, 2017. ISBN: 978-3-3191-8740-2 (cited on pages 3–4).

[52] "Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC". In: *Official*

Journal of the European Union L 117 (May 5, 2017), pages 1–175. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017R0745 (cited on pages 1, 9, 47).

[53] A. Ruiz. *Mastering Android Application Development*. Packt Publishing, 2015. ISBN: 978-1-7858-8762-8. https://books.google.at/books?id=TgKACwAAQBAJ (cited on page 24).

[54] D. F. S. Santos, M. A. M. Lima, J. L. Nascimento, A. Perkusich, and H. O. Almeida. "Enabling the first step for IoT health systems using Antidote and IEEE 11073". In: *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. Apr. 2018, pages 308–313. DOI: 10.1109/WCNCW.2018.8369020 (cited on page 32).

[55] F. Schaub, R. Deyhle, and M. Weber. "Password Entry Usability and Shoulder Surfing Susceptibility on Different Smartphone Platforms". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. MUM '12. Ulm, Germany: ACM, 2012, 13:1–13:10. ISBN: 978-1-4503-1815-0. DOI: 10.1145/2406367.2406384 (cited on page 29).

[56] M. Schmidt and R. Obermaisser. "Middleware for the integration of Bluetooth LE devices based on MQTT and ISO/IEEE 11073". In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pages 1–4. DOI: 10.1109/CCECE.2017.7946779 (cited on page 32).

[57] C. Selvan, A. Thukral, D. Dutta, S. Ghosh, and S. Chowdhury. "Impact of Self-monitoring of Blood Glucose Log Reliability on Long-term Glycemic Outcomes in Children with Type 1 Diabetes". In: *Indian Journal of Endocrinology and Metabolism* 21.3 (2017), pages 382–386 (cited on page 31).

[58] L. L. W. Sim, K. H. K. Ban, T. W. Tan, S. K. Sethi, and T. P. Loh. "Development of a clinical decision support system for diabetes care: A pilot study". In: *PLOS ONE* 12.2 (Feb. 2017), pages 1–15. DOI: 10.1371/journal.pone.0173021 (cited on page 4).

[59] S. Spat, K. Donsa, P. Beck, B. Höll, J. K. Mader, L. Schaupp, T. Augustin, F. Chiarugi, K. M. Lichtenegger, J. Plank, and T. R. Pieber. "A Mobile Computerized Decision Support System to Prevent Hypoglycemia in Hospitalized Patients With Type 2 Diabetes Mellitus: Lessons Learned From a Clinical Feasibility Study." In: *Journal of Diabetes Science and Technology* 11.1 (Nov. 2017), pages 20–28. ISSN: 1932-2968. DOI: 10.1177/1932296816676501 (cited on pages 1, 4, 27, 51).

[60] E. von Zezschwitz, A. De Luca, and H. Hussmann. "Honey, I Shrunk the Keys: Influences of Mobile Devices on Password Composition and Authentication Performance". In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. NordiCHI '14. Helsinki, Finland: ACM, 2014, pages 461–470. ISBN: 978-1-4503-2542-4. DOI: 10.1145/2639189.2639218 (cited on page 29).