Diploma Thesis

A contextual eHealth terminal facilitating contactless technologies

Markus Falgenhauer

Institute for Genomics and Bioinformatics

Graz University of Technology

Evaluator:

Univ. Doz. DI Dr. techn. Günter Schreier, MSc

Supervisor:

Dipl. Ing. Kurt Edegger, AIT Austrian Institute of Technology GmbH

Graz, May 2011

This diploma thesis was established
in cooperation with

AIT Austrian Institute of Technology GmbH

Safety & Security Department

Head of the eHealth and AAL- research team:

Univ. Doz. DI Dr. techn. Günter Schreier, MSc

Reininghaussstraße 13/1
8020 Graz
Austria

DANKSAGUNG

Ich möchte mich an dieser Stelle bei all jenen danken, die mich bei der Durchführung meines Studiums und dieser abschließenden Arbeit unterstützt haben.

Besonderer Dank gilt meinen Eltern, die mir dieses Studium ermöglichten, mich bei all meinen Entscheidungen sowohl emotional als auch finanziell unterstützt haben und mir immer mit gutem Beispiel vorangegangen sind.

Weiterer Dank gilt all meinen Freunden und Freundinnen, die mich im Studium, in der Freizeit und auch bei sportlichen Aktivitäten unterstützt und begleitet haben, und mir somit einen Ausgleich zum Studienleben ermöglichten.

Ich möchte auch dem gesamten eHealth und AAL - Forschungsteam der AIT Austrian Institute of Technology GmbH einen herzlichen Dank aussprechen, da durch das freundliche Arbeitsklima und die persönliche Betreuung die Durchführung dieser Arbeit reibungslos möglich war.

ABSTRACT

Reducing the distance between users of health care applications and the applications themselves, by leveraging appropriate technologies, is mature issue in eHealth.

In course of this thesis a health care application on an Android tablet has been developed. This application admits several users, with different roles and permissions to interact with each other in different contexts. The authentication of the users is performed using NFC technology. For instance a patient is able to log in with his ID card and a screen for monitoring lifestyle and health parameters is shown. At the same application when a physician logs in with his ID card a screen appears, where he can perform configurations or view data of patients of who he has the permission to look at. If a patient ID card comes into the context when a physician is logged in, a screen for viewing the relevant data of the patient pops up. Furthermore, if a physician of another area of expertise and a patient come together, the application is able to change the context to show the relevant data for this particular physician.

This kind of “playing with cards” enables the development of extremely rich and innovative applications whereat these cards could represent user IDs as well as healthcare-, lifestyle-, or configuration data.

Furthermore, beside this concept an application was developed, where patients can track their nutrition intake, amount of drinking and physical activity for the resent day, including graphical feedback on the screen.

The main simplification for the users is performed by using the touchscreen as intuitive input device as well as NFC technology for the acquisition of data.

KURZFASSUNG

Bei der Entwicklung neuer Anwendungen in eHealth ist ein wichtiges Thema der wirksame Einsatz geeigneter Technologien, um den Nutzern die Interaktion mit diesen Anwendungen zu erleichtern.

Im Laufe dieser Diplomarbeit wurde eine eHealth Applikation für ein Android Tablet entwickelt. Diese Anwendung erlaubt mehreren Nutzern, mit verschiedenen Rollen und Berechtigungen, miteinander auf unterschiedliche Weise in Interaktion zu treten. Die Authentifizierung der Nutzer wird mithilfe von NFC Technologie durchgeführt. Als Beispiel: Ein Patient kann sich mit seiner RFID Karte am Tablet anmelden und erhält einen Bildschirm der es ihm ermöglicht Gesundheits- und Lifestyle- Daten einzugeben. In derselben Anwendung kann sich ein Arzt mit seiner ID Karte anmelden, wobei ein anderer Bildschirm dargestellt wird, auf dem er Konfigurationen ändern und Daten von seinen Patienten einsehen kann. Kommt die Karte eines Patienten hinzu während ein Arzt angemeldet ist, werden am Bildschirm die relevanten Daten des betreffenden Patienten angezeigt. Wenn die Karten eines Patienten und die eines Arztes anderen Fachgebietes zusammenkommen, können die für diesen Arzt relevanten Daten angezeigt werden.

Dieses „Spielen mit Karten“ ermöglicht die Entwicklung vereinfachender und innovativer Anwendungen, wobei die Karten, sowohl Benutzer- als auch Gesundheits- Lifestyle- oder Konfigurations- Daten enthalten können.

Neben diesem Konzept wurde eine Anwendung entwickelt, die es Patienten ermöglicht, ihre Nahrungsaufnahme, Menge an Getränken, und Bewegung eines Tages einzugeben und über grafische Rückmeldung mitzuverfolgen.

Die wesentlichen Vereinfachungen für die Nutzer ergeben sich aufgrund der Verwendung von Touchscreens als intuitive Eingabemöglichkeit und NFC Technologie zur Datenerfassung.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAL	**A**mbient **A**ssisted **L**iving
ADT	**A**ndroid **D**evelopment **T**ool
AHD	**A**pplication **H**osting **D**evice
AIT	**A**ustrian **I**nstitute of **T**echnology
API	**A**pplication **P**rogramming **I**nterface
CHA	**C**ontinua **H**ealth **A**llience
EDGE	**E**nhanced **D**ata Rates for **G**SM **E**volution
EPR	**E**lectronic **P**atient **R**ecord
GPRS	**G**eneral **P**acket **R**adio **S**ervice
GSM	**G**lobal **S**ystem for **M**obile **C**ommunication
GUI	**G**raphical **U**ser **I**nterface
HRN	**H**ealth **R**eporting **N**etwork
HSDPA	**H**igh-**S**peed **D**ownlink **P**acket **A**ccess
ID	**Id**entification
IDE	**I**ntegrated **D**evelopment **E**nvironment
LAN	**L**ocal **A**rea **N**etwork
NFC	**N**ear **F**ield **C**ommunication
PAN	**P**ersonal **A**rea **N**etwork
RFID	**R**adio **F**requency **I**dentification
SDK	**S**oftware **D**evelopment **K**it
SVN	**S**ub**v**ersio**n**
UMTS	**U**niversal **M**obile **T**elecommunication **S**ystem
URL	**U**niform **R**esource **L**ocator
USB	**U**niversal **S**erial **B**us

WAN	**W**ide **A**rea **N**etwork
WHO	**W**orld **H**ealth **O**rganisation
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
XML	e**X**tended **M**arkup **L**anguage

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.

Graz, am …………………………… ………………………………………………..

(Unterschrift)

STATUTORY DECLARATION

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 marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..

date (signature)

In order to improve readability, only the male form is used in the text, nevertheless all data apply to members of both gender.

1 Introduction

1.1 Motivation

The use of information and communication technology in healthcare brings along a lot of innovations. Due to the fact that innovations always come along with changes of previous workflows, challenges occur. These challenges affect all participants of the health care sector including patients as well as physicians. Therefore, an increase in usability of new technologies, with the ability to support users without confusing them too much, is an important issue.

Statistics show that, in general, elderly people require more support from the health care system than younger ones. Therefore, simplifications are fundamental requirements in the development of new concepts to make life easier for older patients. Based on a report of the world health organisation (*WHO, 2001*) the most common risk factors for cardiovascular and metabolism diseases are, among other things, cholesterol, overweight, too less consumption of fruits and vegetables as well as physical inactivity. Hence, a system giving feedback about nutrition, activity and so on could help people managing their lifestyle and maybe motivate them to change it to live a healthier life.

New technologies which are intuitive, well arranged, and easy to operate could be used for such a system. Additionally, giving elderly people the ability to interact with state of the art devices may help them to follow the fast increase of new technologies instead of just wondering what futuristic device the grandchild is playing with.

However, not only should patients be in the focus of easing life. Technologies should also be used to support providers of medical services in their daily work.

A concept including a prototype that combines all these thoughts is the topic I am focusing on in this thesis.

1.2 Contactless technologies in eHealth

Near Field Communication (NFC) is a contactless technology for information exchange over short distances (less than 4 centimetres)[1]. It can be either used for identification of people by bringing a user ID-card based on RFID technology near the reading device, or to acquire data, like personal health data from NFC enabled devices (e.g. body weight from an NFC enabled body weight scale, Morak et al., 2007). Also, a peer to peer communication between two NFC devices is a possibility for data exchange.

NFC is currently expected to be widely adopted a boom in near future due to the fact that buying tickets and performing payments can be done using mobile phones. There are already some pilot projects like a system from the *Mobilkom Austria* in connection with the *ÖBB* where tickets can be bought via NFC on two routes around Vienna[2]. Furthermore *Visa* and *Samsung* have announced to bring NFC payment to the Olympic Games in London 2012[3]. Therefore, the ability of devices supporting NFC should increase during the next years.

Bluetooth is an industry standard, for transmitting data wirelessly over a distance of several meters. It is characterised by low cost and low energy consumption. It is mainly used for portable devices to replace the cable for data transmission. There is also a Bluetooth Health Device Profile defined in the Continua Health Alliance (CHA) Design Guidelines (Version 1.5 April 2010), for personal area network (PAN) interfaces. There are already a lot of Bluetooth devices available, certified by CHA. Pulse Oximeters, Thermometers, Step Counters, Glucose Meters, just to mention some of them.

Another commonly used wireless technology is Wireless LAN (WLAN). It can be used to connect to the Internet as well as other WLAN devices to enable communication between them.

[1] http://www.nfc-forum.org/aboutnfc/nfc_and_contactless/

[2] http://www.nearfieldcommunicationsworld.com/2009/03/19/3878/mobilkom-austria-customers-to-use-nfc-tags-to-access-wap-pages/

[3] http://www.nearfieldcommunicationsworld.com/2011/03/31/36779/samsung-and-visa-to-bring-nfc-payments-to-the-london-2012-olympics/

1.3 Android Tablet-PC as an eHealth- terminal

Over the past few years, Tablets have become powerful and affordable devices that enable easy access to information and media. The interaction between humans and the devices can be performed very intuitively by touching items on the display with a finger. Furthermore, Tablets provide a wide range of functions and interfaces (access to the Internet, the ability to connect to other devices via Bluetooth or WIFI, taking a photo and so on).

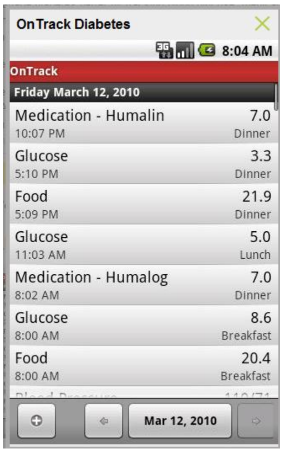
Android is a software stack (including an operating system) that enables easy access to all functions available on the tablet. With the use of the Android software development kit (SDK) developers are able to design new and innovative applications (apps) using all elements of the Android framework. These apps can be shared with others via the Android market for free, or for a certain fee.

The availability of NFC on Android systems is already on the road. The application programming interface (API) upon Android 2.3 provides access to NFC functionalities. Currently there are already Android smart phones including NFC available (e.g. Samsung Nexus S), tablet enabling NFC are already announced (NFC Android tablet from TazTag presented on the CeBIT 2011), hence the supply of devices should be guaranteed.

1.4 *Monitoring of personal health and lifestyle data*

There are already some applications for monitoring personal health- and lifestyle-data on the Internet and available on mobile devices. Web services like GoogleHealth[4] or Microsofts HealthVault[5] allow creating a personal health record, where the user can upload or insert personal health related data. Furthermore, this data can also be shared with external health care providers.

On Android devices there is the possibility to use web services like GoogleHealth. Additionally, on the Android market there are already a large number of applications available for collecting health data and giving feedback. One of them would be for instance "OnTrackDiabetes" available for free at the Android market[6]. With the help of this application, blood glucose, food, medication, blood pressure, exercise, weight and so on can be tracked (shown in Figure 1.1). The logged data can be sent per email to the physician, which may be a difficult workflow for elderly users.



Friday March 12, 2010

Medication - Humalin 7.0
10:07 PM Dinner

Glucose 3.3
5:10 PM Dinner

Food 21.9
5:09 PM Dinner

Glucose 5.0
11:03 AM Lunch

Medication - Humalog 7.0
8:02 AM Dinner

Glucose 8.6
8:00 AM Breakfast

Food 20.4
8:00 AM Breakfast

Figure 1.1: Screenshot of the OnTrack Diabetes application

[4] http://www.google.com/health

[5] http://www.healthvault.com/

[6] https://market.android.com/

Beside these stand-alone data collection systems which are available for everybody on the Internet, there are different telemonitoring systems available (e.g. ELICARD[7] Elisabethinen Hospital Linz) which allow the transmission of health data from a patient to a physician over a web system.

1.5 Simplifications in telemonitoring systems

A general schema of telemonitoring is shown in Figure 1.2. One actor is the patient entering health and lifestyle data into the system. Another actor is the physician, who has to have the possibility to view the data and giving feedback, hence closing the loop back to the patient. The third component is the telemonitoring service center holding the data and supporting the system.

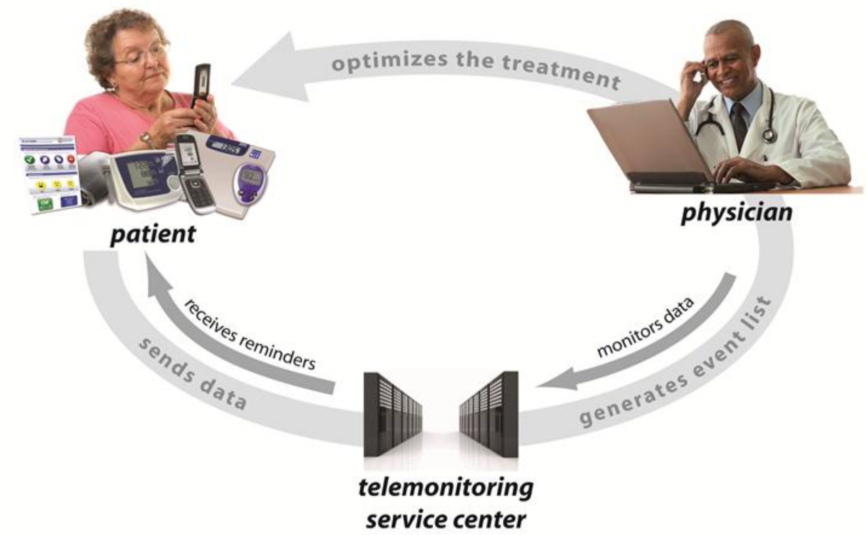


Figure 1.2: Telemonitoring scenario

[7] https://elicard.at/

To bring improvements for the participants of such a system, some simplifications can be performed:

Simplifications for patients

Using appropriate technologies which are easy and intuitive to deal with, can ease the interaction of patients with the telemonitoring system. These should especially support elderly people, which are the group of focus in this thesis.

Simplifications for physicians

Focusing the physician on relevant data by filtering or preselecting a given patient would simplify the workflow for the physician enormous. Currently, the Austrian Institute of Technology GmbH (AIT) operates telemonitoring systems for different indications. Within every system, the physician has to insert an URL into his web browser and enter his username and password for logging in. Then he has to search for the patient and after that for the data he is interested in. The more data is in the system the more the physician has to search.

This workflow is not unusual in a lot of different fields of applications. But it could be easier and more time efficient if the physician would log into the system with his ID card, and the patient's health data pops up at the moment the patient adds his card.

1.6 *"Playing with Cards" – the goal of the application*

The goal of this thesis is to develop a prototype of an application on an Android tablet used as a terminal, which allows multiple users with different roles and permissions to act in an eHealth scenario. Users should be able to log in at the terminal with their user ID cards, as indicated at Figure 1.3. Based on their role, they get different screens, where they are able to insert and view data. If another user comes into the context various things could happen. This depends on the role of both, the user who had- logged in already and the user who joins the context. The concept should be flexible and extensible to cover Ambient Assisted Living (AAL) scenarios as well as the support of patients suffering from chronic diseases.

Out of these thoughts the phrase "Playing with Cards" occurred. The NFC reader could be understood as a table, and different users put their cards on the table. Different contexts are created based on the combination of the cards.

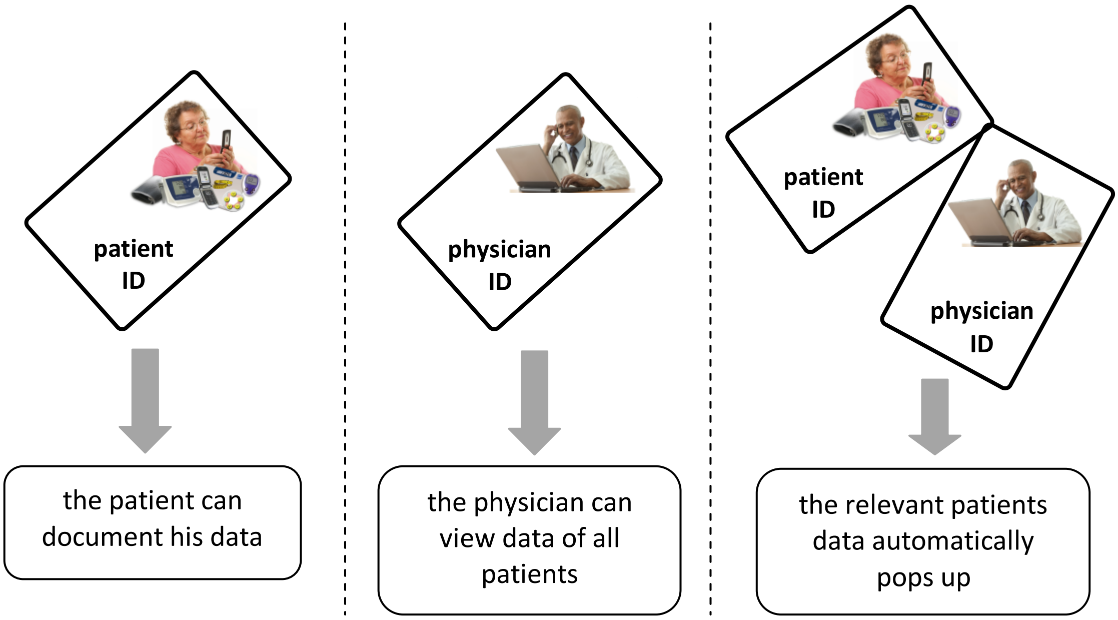

patient ID
the patient can document his data
physician ID
the physician can view data of all patients
patient ID
physician ID
the relevant patients data automatically pops up

Figure 1.3: The concept of "Playing with Cards"

A main advantage following out of this concept is, that the physician can obtain the data he is interested in based on the context. For example if a patient comes to a cardiologist heart relevant data is displayed whereas at a diabetologist metabolic relevant data is shown.

Two possible scenarios are described briefly:

1. The terminal is located in a nursing home. Patients are able to log in with their ID card and insert data every day (e.g. bodyweight, blood-pressure, nutrition, activity and so on). Attendants can log in with their ID card and have the ability to configure the program like setting thresholds, reminders and so forth. If an attendant is logged in and a patient comes to the context with his ID card, the attendant gets the permission to view the patient's data. If two patients put their card on the table simultaneously they can compare their data and so on.

2. The terminal is located at a medical practice. The physician logs in and is able to work within the application. When a patient arrives, he just has to put his ID card onto the NFC reader and all relevant data for that particular patient appears on the screen.

Therefore not only the information of who acts in the scenario, but also the location, where the interaction takes place may impact the context. On android there are several ways to implement location aware services using GPS or the availability of WLAN networks.

This approach of "Playing with Cards" promises a high ability for covering an enormous amount of different use cases. In this thesis a basic concept to cover these requirements is designed and prototypically implemented.

2 Methods

In this chapter, the concept for developing a multiuser patient terminal application is described. First, an overview over the software stack Android is given. After that, elements for the application are described and proper functions, available on Android including possible features, are outlined.

2.1 Android

Android is a software stack for mobile devices, which includes an operating system, middleware and some key applications[8]. It is based on a Linux-kernel 2.6 which provides services for security, memory management, process management, a network stack and drivers. Upon that kernel there are libraries and an Android runtime environment in which each application runs in its own Dalvik virtual machine. A set of core applications, like an email client, a browser, a calendar, a SMS program and some more, are already integrated. All applications are written in the Java programming language. The application framework provides an open development platform which allows developers to use all the advantages of components already installed on the device. These components can access the device hardware, store files, run services, set alarms, perform a call, or start other applications. It is also possible to use parts of other applications. For instance, if a scrolling list of images is needed, it does not have to be implemented, if another developer has already built one and provided it to others.

The application components are classified into four different types. These are activities, services, content providers and broadcast receivers. An application can contain one or more of them which are interacting with each other.

[8] http://developer.android.com

Activity

Activities represent visualisation and administration of graphical user interfaces. So a main task is the interaction with the user. This could be used to present elements on the display, like a menu or a list with touchable items, or to insert data, such as pushing a button or entering text. Android already provides some basic views for building activities, which are buttons, text fields, check boxes, menu items and so on.

The lifecycle of an activity is shown in Figure 2.1. When an activity is created its onCreate() method is called. In this method the static setup like creating views should be performed. It is followed by onStart() and onResume() which are also called after onPause() (because another activity came in front of this activity) and onStop() (because the activity was completely hidden by another activity). The onDestroy() method is either called if someone called the finish() method on it or it is called because the system killed the application while it was in the stop or pause state and the system needed resources.

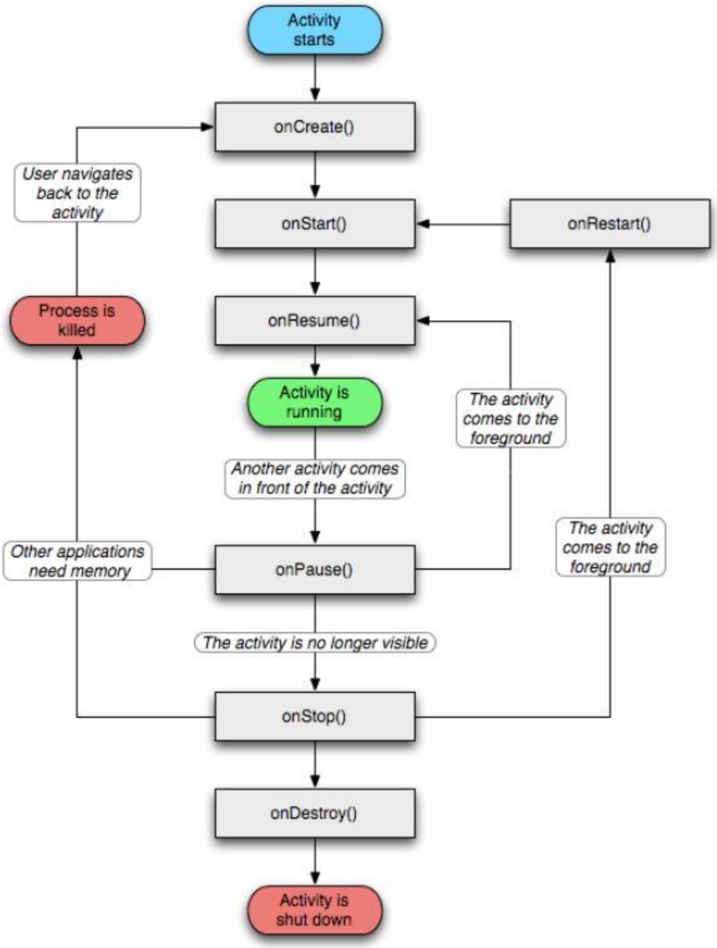

Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
The activity comes to the foreground
Other applications need memory
onPause()
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Figure 2.1: Lifecycle of an activity

Care has to be taken if changes of the configuration of the device are to be made. For instance when the screen orientation is changed from landscape- to portrait-orientation by rotating the device, the activity gets destroyed and a new instance of the activity will be created. For restoring the previous instance state, Android automatically saves the state generated from onSaveInstanceState(Bundle) called before onPause(). This Bundle savedInstanceState is given to the onCreate(Bundle savedInstanceState) method.

Service

Services are used for tasks which are running in the background. They have no visual user interface and can run for an unlimited period of time. A descriptive example for choosing a service is to play music. The music should be played even at times when the player application is not in the foreground and the user is dealing with other activities.

Content Provider

Content Providers administrate data and abstract the persistence layer. They can make data accessible to all applications. There are some providers for general data types like audio, video or contact information available where data can be added. Furthermore, there is also the possibility to build an own Content Provider.

Broadcast Receiver

Broadcast Receivers receive and respond to broadcast messages. These messages could have their origin in the system (e.g. low battery, bad network connection, time zone change) or an application can initiate a broadcast itself.

Intent

An intent is a basic concept of Android to enable applications or parts of applications to communicate with each other. Intent objects are a passive data structure holding the information of an operation, which should be performed (e.g. starting an activity or performing a call). Intents can contain a component name (e.g. the name of an activity or a service), an action (e.g. ACTION_CALL for initiating a phone call), data (e.g. a phone number), category (additional information about the kind of component for handling the intent), extras (key- value pairs of extra information) and flags (e.g. FLAG_ACTIVITY_NO_HISTORY for avoiding writing the activity to the history).

Intents can be resolved using two different approaches. The first is to address the target directly by its component name. This is called an explicit intent. These intents are mainly used for application internal messages. The second possibility is called implicit intent. This is used when the target name is unknown. In this case Android compares the contents of the intent objects to intent filters of all installed applications to find out which application can handle the intent best.

If an intent holds the description of something that has happened and should be announced it is called a broadcast. Broadcast Receivers can listen to them via intent filters.

2.2 Reference Topology of the Continua Health Alliance

The Continua Health Alliance (CHA) is an international, non-profit, open industry organization of more than 230 members. These are healthcare providers and technology companies joining for the improvement of interoperability in the healthcare sector.

In their Guidelines (*CHA Guidelines, 2010*) the CHA defined common concepts, basic structures, rules for refinement and extension of these structures, and association of guidelines with elements in this structure for the end-to-end architecture of the Continua ecosystem. The architecture distinguishes devices (which are physical entities) and components (which are logical entities), who interact with each other via interfaces.

In Figure 2.2 an overview over the Continua Reference Topology is shown. There is an Application Hosing Device (AHD) which can serve interfaces to Personal Area Network (PAN) Devices over a PAN interface, Local Area Networks (LAN) Devices over a LAN interface and Wide Area Network (WAN) Devices over a WAN interface. Across this WAN device the AHD can be connected to a Health Reporting Network (HRN) Device, holding Electronic Patient Records (EPR).

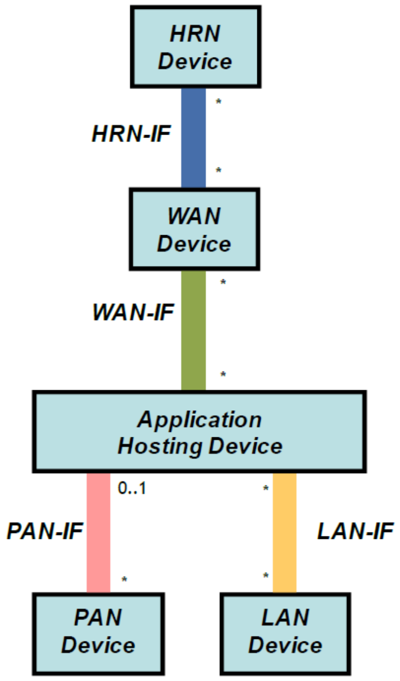


Figure 2.2: CHA Reference Topology

2.3 *Structure of the entire system*

The tablet has several possibilities to interact with its surrounding, as shown in **Fehler! Ungültiger Eigenverweis auf Textmarke.**. On the one hand there is the ability of acquiring data from PAN devices using Bluetooth, NFC, USB or any other connection depending on the specification of the tablet. Via these connections data can be acquired from Personal Health Devices or other data sources. Furthermore, with a Bluetooth paired NFC token data can also be collected from sources like RFID tags or NFC devices.

On the other hand there is the possibility to connect to a network like a LAN or a server via WLAN or 3G. Again it depends on the specification of the tablet, for instance depending on if a 3G module is built-in or not. If the tablet only supports WLAN there is the possibility to connect to the Internet via WLAN for example using a WLAN router connected with a 3G module.

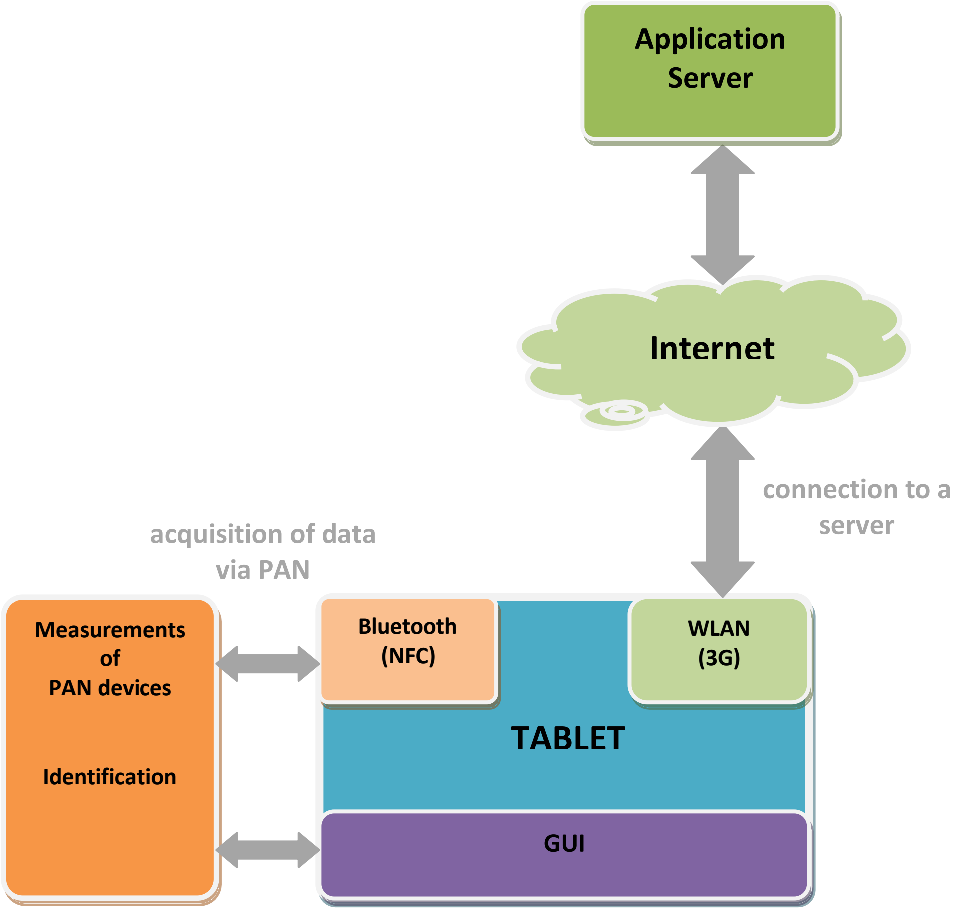


Figure 2.3: Structure of the entire system

2.4 Developing the application

The application was build based on state of the art software design. Attention at the design was turned on designing due to separation of concerns which leads to an extensible and maintainable application.

An overview over the main elements of the application is shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**. The application needs to have an element which is able to acquire data from different sources. There has to be a module to store persistent data. Another essential element is to generate a graphical interface for different users. Furthermore, an element has to decide who is interested in which data, there has to be an element which performs the authentication and another element has to control the workflow of the application.

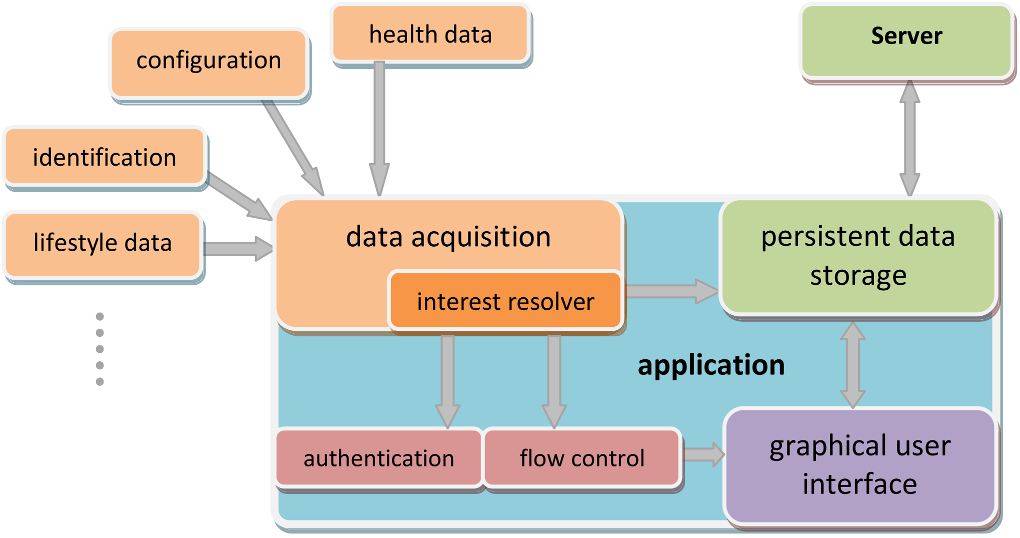


Figure 2.4: Basic architecture of the application

2.4.1 Acquisition of data

Acquisition of data is an essential key point for a monitoring system. This data could be medical data of a patient, data to identify users or even data for the configuration of the application. Additionally the ability of receiving data from a server like medication, messages, alarms, reminders, and so on, could be a requirement. The following selection describes some communication possibilities which the tablet provides to interact with other devices, users or the Internet.

Bluetooth

Bluetooth is a wireless standard for the transmission of data between devices over a short range (between 10 and 100m). A Bluetooth connection is build up ad hoc from one device to another without the need of a central point which manages the network. The CHA defined the Bluetooth Health Device Profile in their Guidelines as the standard for wireless personal area network (PAN) interface. They defined the interface between the Personal Healthcare Devices (e.g. a Bluetooth bodyweight scale like the UC-321PBT-C from A&D Medical) and the Application Hosting Device (e.g. the tablet).

So Bluetooth can be either used for acquiring medical data or for connecting to other devices (e.g. a token for collecting data from other sources).

USB

Using the universal serial bus (USB) allows it to connect the Android device to a personal computer. This allows to transfer data to the device by declaring it as a mass storage device or to debug applications using the Android debugging bridge (ADB).

USB is the wired PAN transmission standard for medical devices defined by the Continua Health Alliance, however, up to Android 2.3 USB host mode is not supported, so it can not be used as Application Hosting Device in terms of the CHA. Android 3.0 supports the USB host mode for keyboards and USB hubs.

Wireless LAN

WLAN is standardized in IEEE 802.11 and enables wireless communication with higher data rate (up to 100Mbps) than Bluetooth. It is generally used for a connection to the Internet via a WLAN router or to communicate directly to other WLAN devices (ad-hoc network) or over an access point. WLAN was designed to cover a distance for mobile computing within the range of a building.

3G

3G names the third generation of mobile communication services. It follows the first generation which was the Advanced Mobile Phone Service (AMPS) to transmit analog voice, and the second generation (2G) whose most common standard is the Global System for Mobile communication (GSM) which could not only transmit voice but also circuit-switched data with rates up to 14.4Kbps. A further step in cellular wireless technologies was GPRS (max. 114Kbps) widely considered as 2.5G and EDGE (max. 384Kbps). The third generation was developed to increase data transmission rates so as to allow the support of mobile multimedia, and other applications. It is based on packet-switched data including higher spectral efficiency at significantly higher speeds. This led to the following technologies: UMTS (max. 1.92Mbps downlink) and HSDPA (max. 14Mbps downlink)[9]. Technologies with higher transfer rate are planned to follow. The availability of 3G on the tablet enables connectivity to the Internet as well as using services like performing a call (e.g. to call a physician or helpdesk) sending a text message (SMS) or a multimedia message (MMS).

NFC

NFC is a standard for contactless interchange of data over a short range (below 4cm). The transmission can be either peer to peer (between two NFC devices) or between an initiator and a target (e.g. reading of passive RFID Tags). It can be used for the identification of users or, with the use of NFC-enabled medical devices, it is also possible to transmit medical data in the home environment (e.g. blood-pressure or bodyweight Morak et al., 2007).

[9] http://searchtelecom.techtarget.com/definition/3G

Camera

Mobile phones and tablets often have built-in cameras. They can be used to take pictures e.g. for a tele-dermatology system (Schreier et al., 2008) or even for scanning pharmacy packages to document which medicine was taken.

Graphical User Interface

Android offers a large amount of possibilities to acquire data using the graphical user interface (GUI) of applications. With buttons, image buttons, checkboxes, fields to insert text, lists which can be scrolled and where items can be selected, and so on, data can be entered into the application. Furthermore, also own elements can be designed, for instance an image button which increases the size of the image with every touch to simulate that the amount of something increases with every touch.

2.4.2 Identifying the purpose of acquired data

The acquired data can be medical data (should be stored to a certain user), data to identify and authenticate a user (should not be stored but authenticated) or configuration data (could perform changes in the configuration of the application). Because these different types of data have to be treated differently, there has to be some kind of filter to identify the purpose of the acquired data.

Therefore, some semantic rules have to be defined. On the basis of this rules it can be decided whether the data should be used for authentication, be stored in a datastorage or used for configuration of the application.

2.4.3 Authentication

Another important element is to provide a possibility for a user to authenticate with his ID card. Authentication can be performed with the use of a username and password stored on the ID card which can be compared with a username and password either stored in a database or on a server. The authentication is implemented as an interface with the method “authenticate” receiving a username and password, and returning true if the user is authorized, otherwise it returns false.

This approach allows an easy implementation of a log-in for a showcase application, but can also be replaced by another, more secure authentication module later on.

As the application should be able to handle multiple users with different roles, the concept of role based access control would seem to be the thing to use. In this concept, every user has a role, and every role has certain permissions. For instance patients can have the permission to write data into the database, and physicians can have the permission to read the data of a certain group of patients. The patients could also be divided into more defined groups like patients with a certain disease, with the permission to store lifestyle data as well as health data, and patients with other diseases only with the permission to store lifestyle data.

For further reading the concept of roles and permissions in hospitals is described in detail in a thesis of *S. Gerdes (2007)*.

2.4.4 Format of data sent to the application

The data acquired by the application should have a standardized format, to obtain interoperability. A suitable standard is Java Script Object Notation (JSON)[10].

JSON is a lightweight format to interchange data. It is human readable and for machines easy to parse and generate. A JSON structure may contain an object (which holds key value pairs), an array (which holds an ordered collection of values) or a value (which could be a string, a number, an object, an array, a boolean or null). Using this structure makes it possible to receive data from RFID tags (e.g. user identification), as well as from a web server (e.g. configuration data).

[10] http://www.json.org/

2.4.5 Persistent data storage

On Android platforms there are a couple of different possibilities to save data persistently. It can be stored in shared preferences, in the file system, in databases or with the help of content providers.

Shared Preferences

Shared preferences are a lightweight option to store primitive data in form of key value pairs. They are most commonly used to store settings or states of the application. As they are stored in a preference file, it is possible to use them in one application or, if the proper permission is set, also in other applications. Shared preferences can be predefined in the XML resources of the application and be changed during runtime[11]. A problem using shared preferences for multiple users exists due to the problem that it is difficult to attach user information to a key value pair. One possibility would be to store for instance the user ID inside the name of the key which is feasible but does not scale well.

Database

Databases are intended to store and access a huge and structured amount of data efficiently. In Android there is a full SQLite database engine included which supports most of the SQL commands. SQLite was optimized for the use in mobile platforms. It is a standalone database system, which means that it does not need a server. The data can be stored persistently, and it is also possible to access the data from several applications simultaneously. Android also provides helper classes to manage databases efficient.

An application in a context like this thesis has to store various data of different users. This leads to a huge number of different attributes. Furthermore, extensibility to new types of data should be possible with low investment. Traditional database systems have one column for one attribute. If for example cars are described in a table the attributes could be the vehicle identification number, brand, type, color and so on. These

[11] Android 2 Book, page 124-129

traditional database systems are typically used to describe items with a fixed number of attributes. However, clinical data has an enormous number of different attributes, where every row would be only filled with a few values leading to a sparse matrix (*Dinu, 2007*).

Therefore, a commonly used approach for storing health care data is the entity-attribute-value (EAV) model (*Eggebraaten, 2007*). In this model, every row of a table corresponds to an entity-attribute-value triplet, where each attribute-value pair is describing a fact of an entity. Each row in the table describes a single fact. The entity is the item that is described by the attribute and the value. The entity could be for example, a person, an item, a measuring device, even a group of persons and so on. The attribute could be something like the bodyweight, nutrition intake, as well as a vital-parameter calculated by the application. The list of attributes has to be defined in detail with local controlled vocabulary. Additionally, to the attribute metadata like allowed range, descriptions or how the attribute has to be displayed, can be stored. The value column of the table holds the actual value of the attribute.

This model is an extensible concept which in the basic specification consists of one table with three columns, an entity, an attribute and a value. However, it can also get a more complex structure, if for instance the entity is just described by a key which is mapped to another table which holds users with their user ID, name, addresse and so on.

Using this model allows to add new attributes without changing the database design by adding columns.

File System

Data can also be stored in the internal file system (device memory) or external (SD-card). The access can be performed directly with the use of java I/O methods (java.io). The internal file system creates a private path for the data during compilation, so this data can only be read and written from the corresponding application. Data can also be stored on the SD-card, where every application has access to it. For storing data internal and providing to other applications content providers have to be used.

Content Providers

Content Providers furnish an interface to store and retrieve data on a central point of the device memory where it is publicly accessible to all applications. Data could be the content of a database (e.g. personal contact information) as well as whole files (e.g. audio, video or images). Android already supplies some content providers for common data types, where data can be added and read out, but also allows to create specific content providers. Content providers only give access to the content of the data, but never to the whole database or file, which means that the source of the administrated data remains anonymous.

2.4.6 Synchronization of data with a server

A key element to avoid developing a stand-alone application is the ability to synchronize data with a server. The eHealth & AAL group within the Department of Safety & Security of the Austrian Institute of Technology already operates Medical Research Networks (Medizinisches Forschungsnetzwerk MFN). These Networks allow different telemedical applications to exchange data. Data can be uploaded and provided for patients as well as physicians, to close the circle between all involved parties.

Therefore, the data storage has to be implemented in a way which allows synchronization.

One possibility is to connect to the server via a hyper text transfer protocol (HTTP) or a secure HTTPS connection. In this case the tablet has to initiate the connection when for instance new data is available which should be synchronized.

2.5 User Interface

User interfaces on Android are provided by activities. They can be constructed programmatically or defined as an xml file in the resources of the application. This xml file can be set as the content view in the source code of the activity, where every item can be addressed via its ID.

Three common Android-specific elements to inform a user about something are Toast notifications, status bar notifications and dialog notifications[12]. Toast notifications are short messages that pop up on the screen just for some seconds. They do not allow a response by a user.

Status bar notifications add an icon to the status bar of the system. This bar is on the top of the screen and can be slided down for getting the whole notification. This is a good method for a service that runs in the background and which needs to notify the user about an event.

Dialog notifications can be used to inform about something or also to enter data. A dialog window appears in front of the application. The message can be read, accepted (with Ok) or allows to answer (with Yes, No, Cancel or even inserting values or texts).

Additionally, special elements can be created, like an activity with a red background and a message representing an alarm. Such an alarm could be confirmed by touching the screen twice within two seconds or something like that.

One possible method to display something like physical activity is to use a metaphor that represents the activity of the last week. *Consolvo et al.* uses the picture of a garden where flowers of different colors grow due to the different types of activity which are performed. If a weekly goal is achieved a large butterfly appears, and for recent goals small butterflies are displayed between the flowers. A similar possibility would be to display a pet which plays when the user is active, just lies and gets fat if the user is inactive during one week.

[12] http://developer.android.com/guide/topics/ui/notifiers/index.html

2.6 *Workflow*

To build a prototype application a workflow has to be described to show that the requirements can be met by a real system. Figure 2.5 shows the state diagram of the chosen workflow.

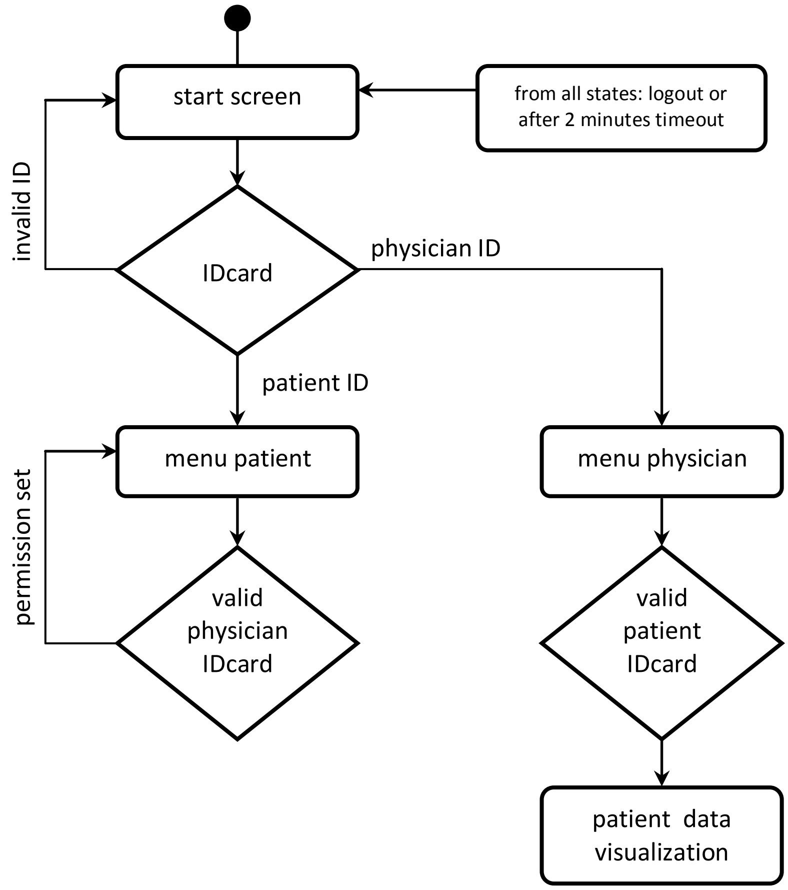


Figure 2.5: State diagram of the workflow

A user has to login with his valid ID card. Dependent of the user's role he gets to a certain menu, where another card can cause other things like a permission to be set, or a visualization of the patients lifestyle data popping up.

2.7 Acquiring data for a lifestyle management system

Different types of data have to be acquired for a lifestyle management system.

Nutrition

To abstract all the different types of food they can be categorized to the staple food types represented by the nutrition pyramid shown in Figure 2.6. These data have been taken from the nutrition suggestions (*Bundesministerium für Gesundheit, 2010*) from the Austrian Federal Ministry of Health. The basis are non-alcoholic drinks (at least 1.5 litres per day). The next important type is vegetables and fruit where five portions should be eaten per day. On top of them there are carbohydrates like crops, potatoes and rice of which 4 portions should be eaten per day. On the next level are dairy products like milk, cheese, and yoghurt of which 3 portions should be eaten per day. Above them there is fish, meat, and eggs which should be ate at least one to two times per week, but not too often. Fat and oil should be eaten just about two table spoons per day and with the lowest frequency sweets, fatty- and salty- food should be consumed.

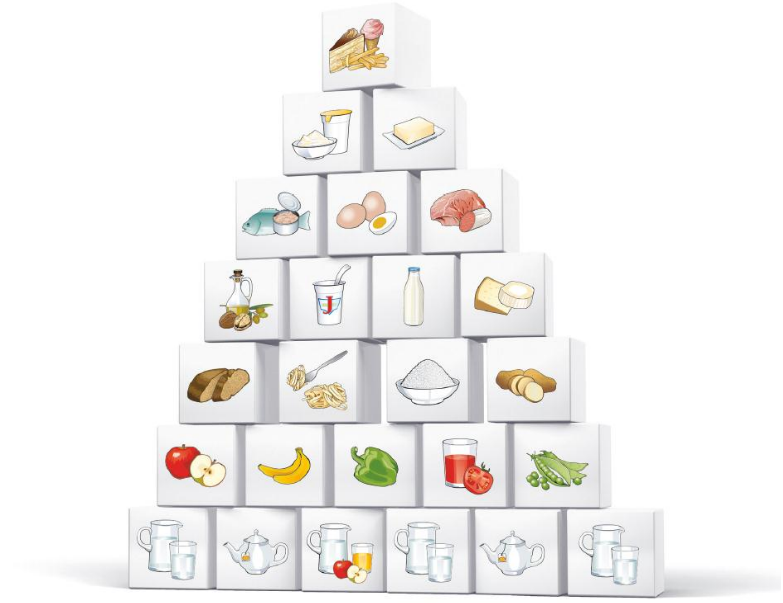


Figure 2.6: Nutrition pyramid of the Austrian Federal Ministry of Health

Activity

Beside a balanced nutrition a moderate amount of activity is also recommended by the Austrian Nutrition Report from 2008 for a healthy lifestyle. Based on the Austrian recommendations for health enhancing physical activity (*Titze et al., 2010*) published by the Federal Ministry of Health, elderly people should at least be physically active 150 minutes per week with medium intensity or 75 minutes per week with high intensity to enforce or sustain well-being.

Activities that are often performed by elderly people are going for a walk, shopping, gardening, household chore and sports.

Health data

Also health data can be acquired by a lifestyle management system. Bodyweight, blood pressure, blood glucose and so on can be useful additions for people suffering from different chronic diseases.

2.8 Software development environment

As a tool for developing the application, the software Eclipse 3.5[13] was chosen. Eclipse comprises an integrated development environment (IDE). It can be used to develop applications in Java or by using different plug-ins also in other programming languages. For developing applications on Android, the Android Software Development Kit (Android SDK) and the Android Development Tools (ADT) plug-in has to be installed. These components are needed for creating Android projects, virtual devices, for building, debugging, logging and so on.

Furthermore, an application can be packed and signed to an android application or .apk file, which can then be installed on an Android device.

As code repository a subversion system (SVN) was used. As plug-in for working on this repository as a team, for documentation and version control, Subversive was integrated to Eclipse. This tool allows a group of developers to work on the same project. If a developer changes something in the source code he commits it and other developers can update the state of the code.

A program that allows for the automatic testing of methods or parts of an application written in Java is JUnit. To use JUnit there exists also a plug-in for Eclipse. A test project with test cases has to be implemented, which can be run after changes have been made to check if the method or the part of the application still works correctly. The outcome is compared to the expected outcome so the result can only succeed or fail.

[13] http://www.eclipse.org/

2.9 Designing an applications for elderly users

An exact definition of the age of so called “elderly people” is both difficult and undefined. Based on a literature review of *guidelines for web access for older users*[14], different definitions starting from 50+ years up to 65+ years are mentioned. Some basic rules for designing an application for elderly people were taken from this literature review. There for instance is predicated, that technologies should support, but not dominate the daily routine of elderly people. They have individual knowledge and needs concerning technologies which have to be considered. Furthermore, GUIs have to be designed intuitively and for the needs of the user group because they are often not used to how information is processed in technical applications.

Applications can be designed very user friendly using appropriate technologies such as touch screens. According to *Greenstein and Arnaud (1988),* the most obvious advantage of touch screens is that the input device is also the output device. This allows direct eye-hand coordination leading to an easy operation of the system.

According to a study performed by *Holzinger (2002)* a suitable method for designing a system for disabled and elderly people is User-Centered Design, meaning that users are involved in the design process. The system has to be simple and easy to use, according to the proverb: *less is more*. The touchable items can be designed in a size, shape and color appropriate for a certain user group. With an interactive patient communications system *Holzinger* figured out that all people in his study who have never used a computer before found a touch screen interface easy to use.

Furthermore, giving elderly people the ability to interact with state of the art devices gives them the chance to keep up to date with technology.

[14] http://www.w3.org/TR/wai-age-literature/

3 Results

In this chapter the implementation of the prototype using an Android 2.1 tablet are presented. The modules which were described in the previous chapter were implemented to meet the requirements.

3.1 *The entire system*

The terminal consisted of the Android tablet Huawei S7 (*Huawei Technologies Co. Ltd, China, Shenzhen*) in combination with an RFID reader, as shown in Figure 3.1. As RFID reader a mobile phone with integrated NFC chip the Nokia 6212 Classic (*Nokia Co., Finland, Espoo*) was used. The advantages of using this phone as RFID reader is that it is an available NFC/RFID device, with a Bluetooth interface, that allows an easy, low cost connection to the tablet. Due to the fact that no other feature of the phone had to be used and no key had to be pressed it could also have been hidden in a black box. So in this setting it is just a device for reading RFID cards. Additionally, the reader can be picked up, acting as a token for collecting data from different NFC sources. If the token is out of Bluetooth connection range to the tablet, the collected NFC data is stored on the mobile phone and sent to the tablet later on.

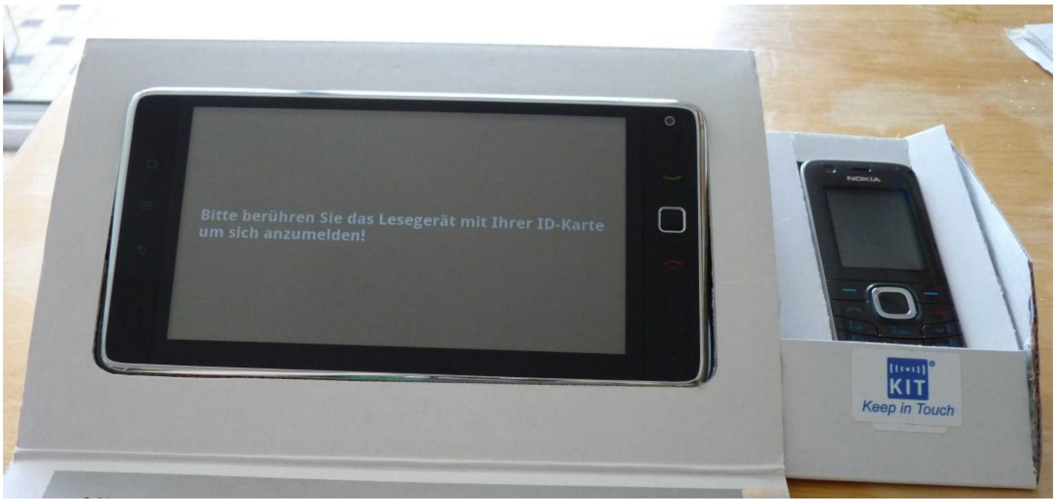

Bitte berühren Sie das Lesegerät mit Ihrer ID-Karte um sich anzumelden!
NOKIA
KIT
Keep in Touch

Figure 3.1: User terminal consisting of the tablet and an RFID reader

On the tablet the application with which the users interact was implemented. On the start screen the request for bringing the user ID card near the reader is displayed. Without ID card, there is no way to enter the application.

A block diagram of the entire implemented system is displayed in Figure 3.2. The tablet acquires data via the RFID reader paired with Bluetooth. Interaction with the user can be performed using the GUI which is displayed on the touchscreen. Synchronisation of the database with a server is not implemented so far, but there is the ability to add it later on.

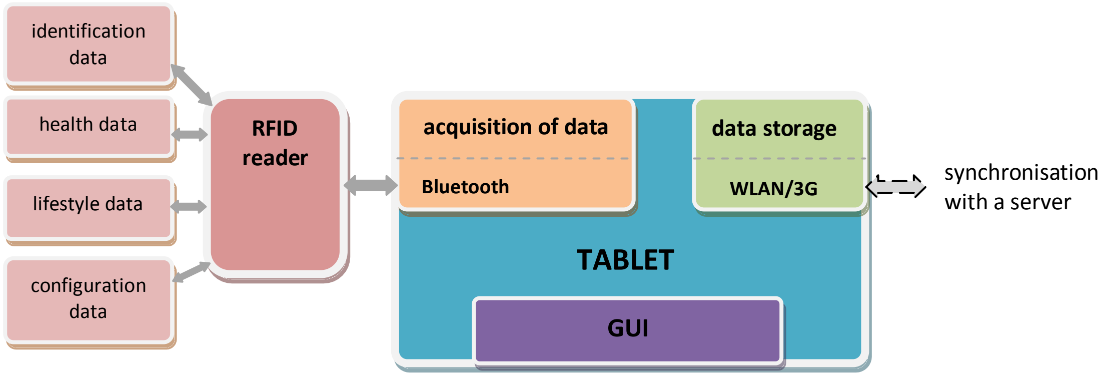


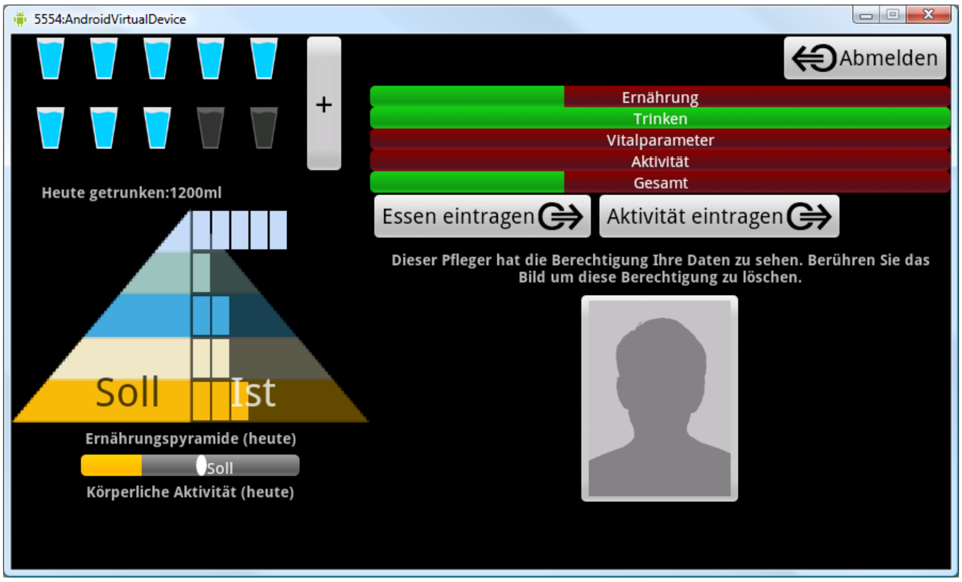
Figure 3.2: Block diagram of the entire system

The workflow using different cards will be described in the next chapter.

3.2 Workflow

The workflow like described in the methods (*2.6*) was implemented.

If a user logs in with a valid patient ID card, his own home screen appears as shown in Figure 3.3. There he can document his nutrition intake, the amount of the liquid he has drunk, and the activities he has performed. To give the user a feedback of his parameters during the current day there are some graphics on the screen. For instance there are some empty glasses that get filled if a certain amount of drinking is entered. So the user is able to follow if the daily amount of liquid has been okay so far, or if another glass of water would be the recommended amount and therefore could enforce his well-being. There is also a half nutrition pyramid shown, where the other half grows when the user enters data, so the user gets the response whether his nutrition is balanced or not. To display the current activity during the day, there is a bar, which displays the current amount compared to a target amount, to give the user the feedback if he has done enough exercise. Furthermore there are some bars that represent the balance calculated over the last 7 days.



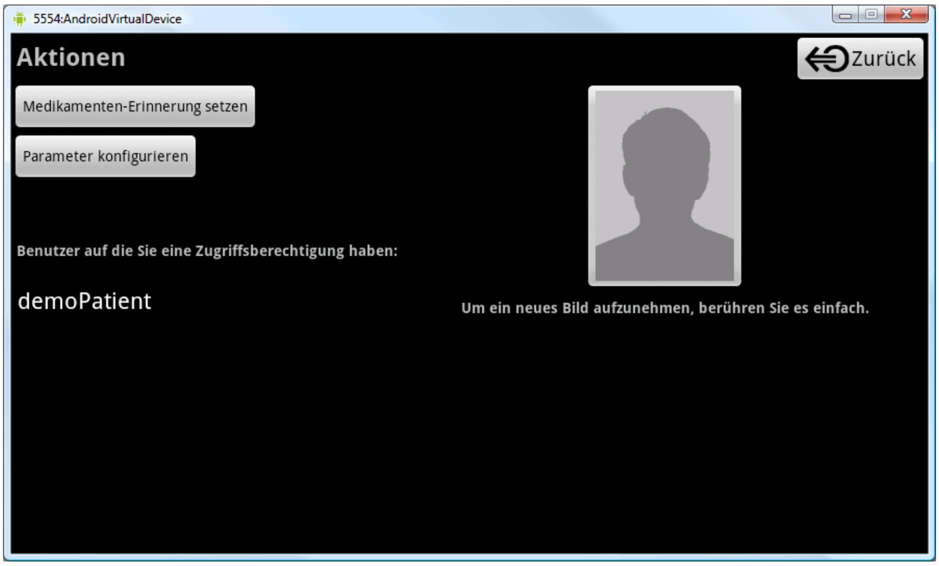
Dieser Pfleger hat die Berechtigung Ihre Daten zu sehen. Berühren Sie das Bild um diese Berechtigung zu löschen.

Figure 3.3: Home screen of a patient filled with some parameters

For documenting nutrition intake, the button “Essen eintragen” has to be touched, for documenting physical activity there is the button “Aktivität eintragen” and for documenting drinking the user has to touch the “+” on the right side of the glasses.

If a valid ID card of a physician comes into the context while the patient is logged in, the physician gets the permission to access the data of the patient. To make this connection transparent for the patient, the patient gets a photo of the authorized physician on his menu screen. With a touch onto the photo the authorization can be revoked. To logout the user can touch the logout button or he is logged out automatically after a certain time of no interaction with the application.

If a physician logs in with a valid ID card his own screen pops up, like shown in Figure 3.4. On this screen he is able to set alarms (e.g. for medication) set configuration parameters of the application (e.g. the target amount of activity) or take a photo of himself by touching the picture. This picture gets displayed on the patient's menu to show that he is authorized to access the patient's data. Furthermore, there is a list of the patients whose data he may access.



Benutzer auf die Sie eine Zugriffsberechtigung haben:

demoPatient

Um ein neues Bild aufzunehmen, berühren Sie es einfach.

Figure 3.4: Home screen of a physician

To access patient data the physician can touch the respective row of the patients list if he already has the permission. Another possibility is that a patient brings his ID card near the RFID reader, while the physician is logged in. After that, some graphics of the patient data emerge. These graphics represent the visualization of the data documented by the patient during the last 7 days. So the physician gets a survey of the amount and balance of drinking, nutrition and activity. Figure 3.5 shows an example of the patient's drinking amount during the last days.

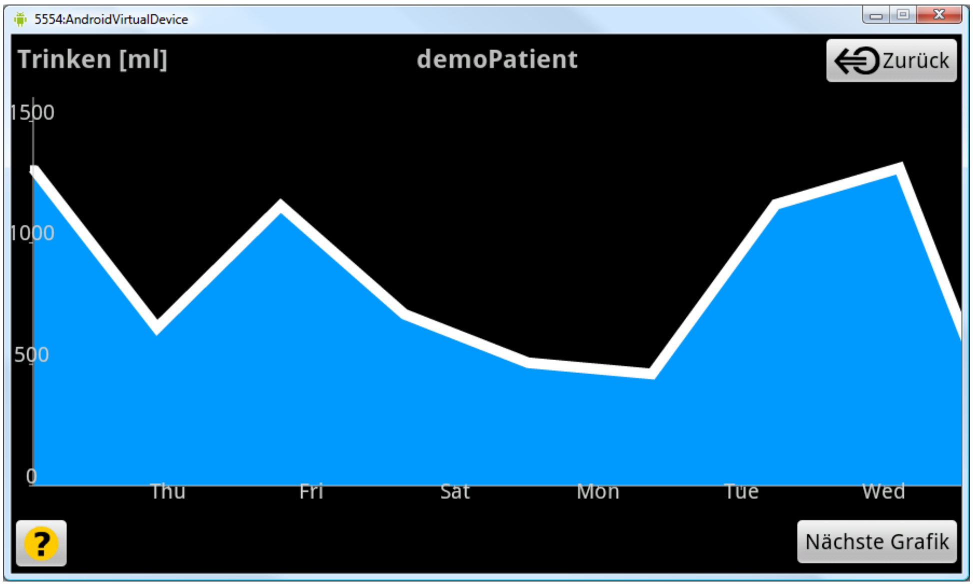


Figure 3.5: Survey of the amount of drinking during the last 7 days

In the next chapters some modules of the application will be described in more detail.

3.3 Data Acquisition

Acquiring data all over the time the application is running, is done by an Android service (2.1) which is said to be best practice by the Android developers guide.

Figure 3.6 shows an overview over the modules involved in acquiring data. The DataAcquisitionService handles all currently open connections. For the prototype of the application the connection to an NFC reader - paired via a Bluetooth channel was implemented. The NCF reader is a Nokia 6212 Classic mobile phone which is able to read RFID Tags. On this phone a java application is running, which reads the tags, transfers the content to a JSON string and sends it via Bluetooth to the tablet. If the Bluetooth connection gets lost, the data gets buffered in the phone, until the connection gets recovered automatically again.

This combination between the mobile phone and the tablet enables the application to read RFID tags similar like there would be a reader in the tablet integrated, because no interaction with the keypad or display of the phone is necessary. This workaround can be used until an Android tablet with integrated NFC reader is launched.

The acquired data is forwarded via an intent to the InterestResolver where it is routed to the desired destination or a broadcast is sent which every interested participant receives.

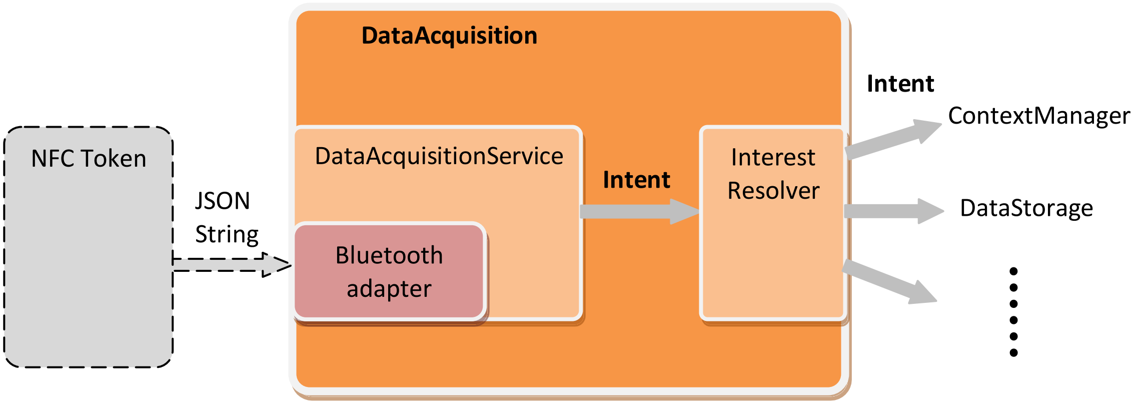


Figure 3.6: Data Acquisition

The DataAcquisitionService starts when the application is started and runs over the entire runtime. In the onStart()- method the Bluetooth connection is initialized and in the onDestroy()- method it is closed again.

The Bluetooth connection was implemented in a separate thread, because it is listening to a connection all the time while it is running. The Bluetooth-adapter is listening using RFCOMM (emulation of a serial port). The token is paired to the tablet and knows its MAC address. The token initiates the connection after it read an RFID tag and sends the data packed into a JSON string. When the connection is accepted by the tablet an input stream is created and written with the string. An onBluetoothReceivedListener (Observer Pattern[15]) in the DataAcquisitionService is notified of this event and forwards the string to the InterestResolver via an intent.

[15] http://www.tutorialized.com/view/tutorial/Observer-Pattern/39365

3.4 Interest Resolver

The InterestResolver dynamically determines which data is routed to a specific destination. To simplify, in this implementation all valid data types are known.

The InterestResolver is also a Service which starts every time it needs to resolve a message. The JSON String coming from the token looks like this:

```
{"measurement_uuid":{
    "values":{"index":{"key":key,"value":value,"unit":unit},...},
    "timestamp":ts,
    "client":{"agent":"RFIDToken","oid":"RT_1"}}}
```

- The measurement_uuid is a universally unique identifier of one measurement.
- Values consist of a key value unit triple.
- The timestamp is counted in milliseconds since 1 January 1970 and represents the time the RFID tag was read.
- The client is defined by the agent, which is the RFID Token and an object identifier.

An example for a transmitted string representing good wellbeing:

```
{"d8a733cf4ea487fb7e3bdda93ae85386":{
"values":{"0":{"key":"feeling","value":"good","unit":""},
    "1":{"key":"deviceid","value":"0123456789","unit":""},
    "2":{"key":"sessionid","value":"","unit":""}},
"timestamp":1284708132630,
"client":{"agent":"RFIDToken","oid":"RT_1"}}}
```

The InterestResolver unmarshals this string and checks which kind of data it contains. For instance, if a username and password is received, it sends an Intent containing this data to the ContextManager (Chapter 3.5) which is aware of the workflow states of the application. If the message contains data that should be stored in a database table it sends it, via intent, to the datastorage. To decide which data should be stored in the database, there is a list defined in an xml file which holds all valid attributes for the storage. In this prototype application data with the attributes: bwt (bodyweight), sbp, dbp (systolic and diastolic blood pressure), map (mean arterial pressure), and hrt (heart rate) are getting stored. These are the attributes coming from the KIT bodyweightscale and KIT bloodpressuremeter.

3.5 *Flow control*

To keep control of the workflow a separate service, named ContextManager, was implemented. This service gets started when the InterestResolver detects that incoming data contains username and password. This authentication data gets transmitted to the ContextManager via an intent where an AuthenticationManager gets asked if the combination of username and password is valid. In this case, it creates a user object that contains the role and permissions of the user. Furthermore, it stores the currently logged in user in the UserManager object handle the multiple users which may simultaneous be present. Figure 3.7 shows an overview of the flow control.

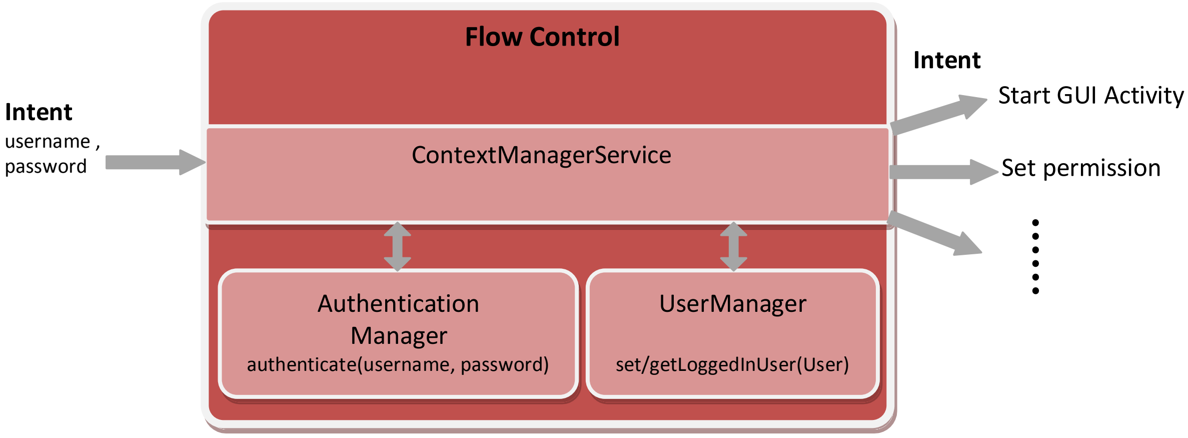


Figure 3.7: FlowControl

Based on the information of which user is already logged in and who currently logs in, different actions are performed or different activities are started. This prototype implements four different possibilities:

- If no user is logged in and a patient logs in the patient’s menu is displayed.
- If a patient is logged in and an ID card with the role of a physician comes into the context, the physician gets the permission to access the patient’s data.
- If no user is logged in and a physician logs in the physician menu pops up
- If a physician is logged in and a patient’s card joins the context, the survey of the patient’s last week data pops up.

3.6 Persistent data storage

Data which has to be stored persistently in the database could be just one single measurement (e.g. one value of the bodyweight) coming from the RFID token, a measurement representing lifestyle inserted to the application via the touchscreen, the role and permissions of a user, relationships between users or even pictures stored as byte-arrays and so on. As the amount of possible information can grow to a high number, several different tables, according to their concerns, which are related via mappings, were implemented. In this chapter some of them are described briefly:

Storage of measurements

For the implementation of the measurement storage, the entity-attribute-value model described in (2.4.5) was applied. Measurements may be bodyweight, blood pressure, nutrition intake, activity, drinking, and so on. These are stored as the attributes in a table that holds a measurement ID, the source of the data, attribute, value, unit and timestamp. The source of the data could be, for instance, an RFID tag, the touch screen (due to user interaction) or generated by the application itself (if the data was calculated from the application based on other measurements). The attribute may be bodyweight, systolic/diastolic blood pressure, heart rate, nutrition intake, activity or drinking. The value was chosen of type string because it is the least common denominator for the types of values, which could be integer, float, or string (e.g. little, medium, much). The string can be parsed to a number when it is read out. The corresponding unit could for example be kilograms for bodyweight, as well as the type of nutrition or activity (carbohydrates, vegetables, walking, sport and so on).

To store the measurements for multiple users the entity, which in this case is the user, has to be added. However the model also lets the possibility open that the entity could also be something else like a device or a group of users (if a group owns data). One option would be to add the data of this entity in an additional column of the measurement table. This is an applicable approach, but after thinking of “separation of concerns” (to build modular structures) the measurements and users are two different things. Furthermore, if other entities are added, new columns have to be added to the measurements table. So, the decision was made to store the measurements of all users in the measurements table, and in a second table the mapping form the measurement ID to the corresponding user ID (shown in Figure 3.8). This approach allows flexible enlargement of the entire database as well as keeping the overview.

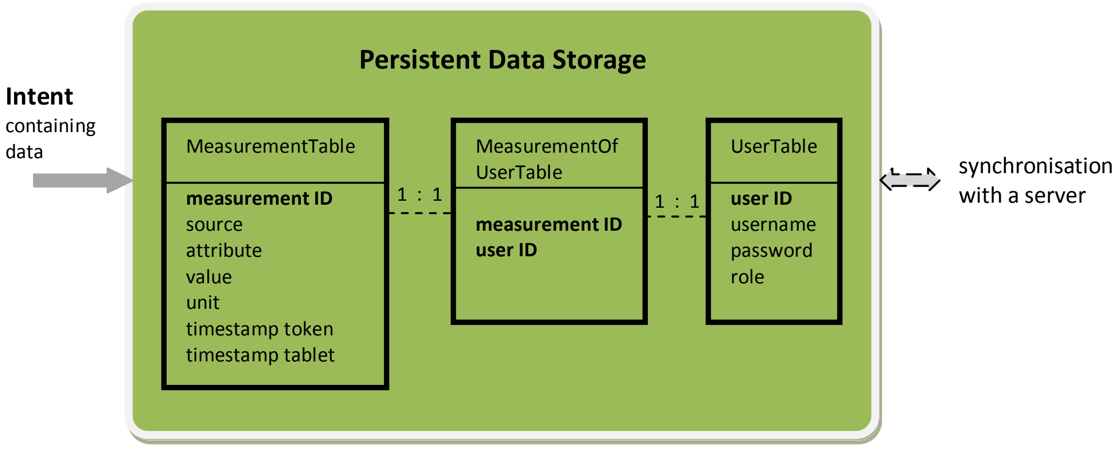


Figure 3.8: Mapping of a measurement to a user

Relation between Patients and Physicians

To remember that a relation between a patient and a physician exists these are also stored in a table to be persistent. Therefore, only a table with a column for patient IDs and one for physician IDs has to be created. Certainly it is also possible to delete these relations out of the database again.

Storage of a picture

In tables images can also be stored and accessed by converting them to a byte array. And again the same like for the connection of a measurement to a certain user an own table was build for the connection of a picture ID to a user ID.

3.6.1 Creating the database

Android already facilitates a SQLiteOpenHelper class, which has the ability to manage the database, like checking if the database exists, creating the database including the tables, or upgrading it. In the onCreate method of this class, the tables are created by executing SQL commands. In the prototype application six different tables are created.

- a MeasurementStorageTable holding all measurements
- a MeasurementOfUserStorageTable holding the connection between measurement ID and user ID
- a UserTable holding the user ID, username, password and role of the user
- a KeyValuesOfUserTable storing key- value- pairs of users, because in the first (single user) application Shared Preferences were used for storing daily amounts. As Shared Preferences are able to hold key value pairs, a key- value- user table can be used quite similar, but for multiple users.
- a PatientPhysicianTable to store relations between patients and physicians persistent
- a UserPictureTable to store pictures of users

3.6.2 Storing data into the databases

For writing data into the tables of the database, a connection to the database has to be opened. This can be done by creating a new DataStorage- object on which the method getWritableDatabase() can be called, and a SQLiteDatabase gets returned. On this database SQL statements can be executed, respectively special methods (SQLiteStatements) can be used to insert data into the tables. The advantage of SQLiteStatements is that reusable pre-compiled statement objects are created which allows easier and more secure handling. So they were used in the application.

For storing a measurement for a specific users two tables have to be written. First, the measurement is inserted into the MeasurementTable. This is done using a SQLiteStatement whose executeInsert()-method returns the ID of the entry that was written. This ID is then inserted as the measurement ID in combination with the currently logged- in user ID into the MeasurementOfUserTable.

3.6.3 Querying the datastorage

For querying the data storage the Method getReadableDatabase() has to be called. The query() method returns a so called “cursor” that points to the matching entries. With the query() method it is also possible to perform a join between two tables to treat them if they were one. This requires that one row of the first table matches to one row of the other table which it should be joined with.

So in case of the MeasurementTable and MeasurementOfUserTable, the measurementID is the value combining the two tables. After joining, the query can be performed for a certain measurement of a certain user.

In this application the query is performed in an own class. This class implements the method getValues to which the values, userID, fromTime, toTime, attribute, unit, orderBy and the application context, can be passed. The application context is needed to enable the class to find the database in the Android framework.

A very important point is that when a database- connection has been opened, it has to be closed again after performing all tasks. Also cursors have to be closed if they are not used any more.

3.6.4 Automatic testing

For automatic testing, if the data flow from the InterestResolver to the data storage works, a test case was written. The test (shown in Figure 3.9) sends an intent containing a JSON string with a randomized generated bodyweight to the InterestResolver. From there the application sends the data via intent to the StoreEavReceiver which handles to store the data into the table. Now the test waits for half a second to ensure that the data base had time to save the data. After that the test case queries the table and compares the last entry with the randomized data sent via the intent. If the value is equal the test succeeded.

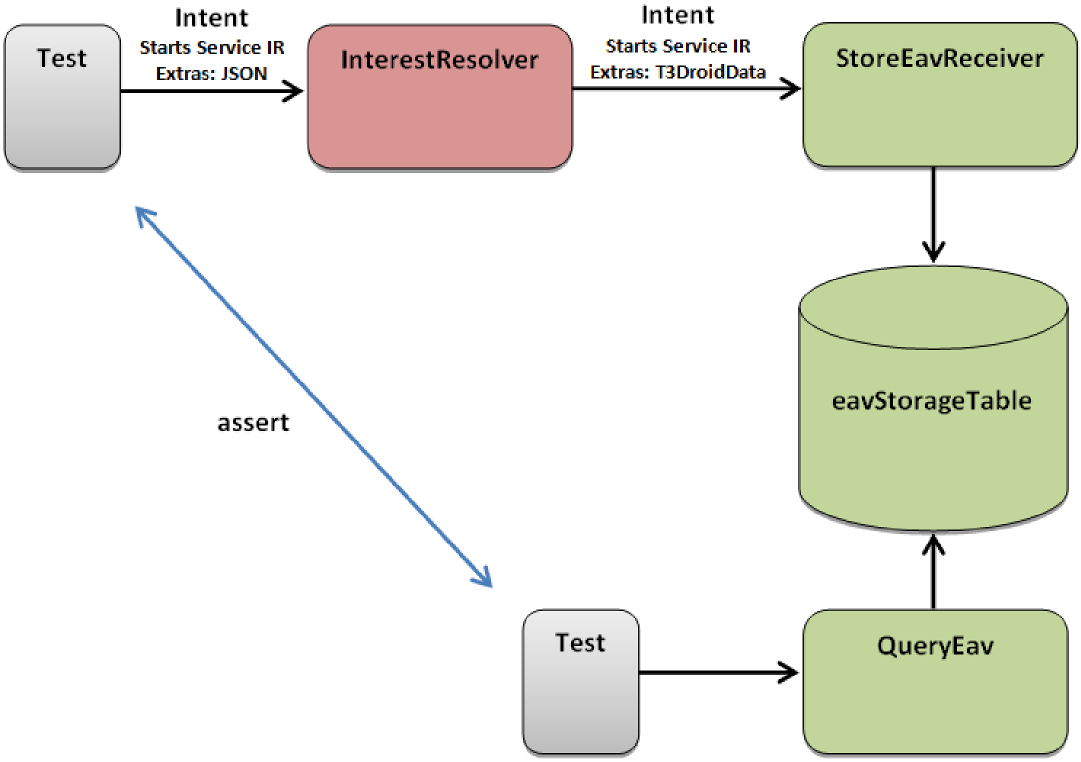


Figure 3.9: Test case for the data flow to the storage

3.7 Graphical User Interface

For developing this application the method User-Centered Design was applied. In the first stakeholder workshop it was established to monitor lifestyle data like, nutrition intake, activity, drinking and healthcare data like, bodyweight and bloodpressure. This application is a prototype of the application which will be used and evaluated by elderly users, to get response for improving it.

It was taken care at designing the parts of the GUI for the elderly users that there are at maximum 6 elements to touch per screen with large text sizes and different colors. Scrolling the screen as well as other special interaction methods like multi touch gestures were avoided, click and long click were joined to allow easier handling.

Toast notifications were avoided in the part for the older users because users because either they could overlook them or they could get annoyed if the message disappears before they were able to read it.

Also the status bar notifications were avoided, because the symbol in the status bar may be overlooked and the scrolling down of the status bar is a gesture to which a user has to be used to.

Dialog notifications were used in connection with pictures because, as the message is in front of the application, it can hardly be overseen. Furthermore the user has time to read the message until he quits it.

To enter the amount of drinking a dialog notification appears in front of the activity shown in Figure 3.10. In this showcase, 200ml which corresponds to a cup, 250ml which matches to a glass, or nothing can be entered.

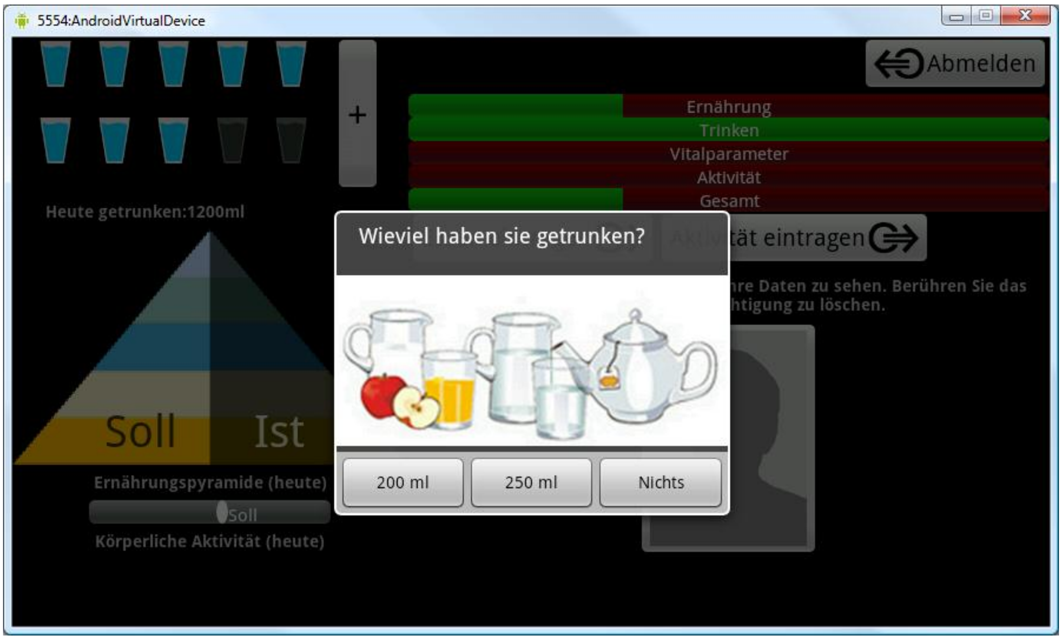

5554:AndroidVirtualDevice
+
Abmelden
Ernährung
Trinken
Vitalparameter
Aktivität
Gesamt
Heute getrunken:1200ml
Wieviel haben sie getrunken?
tät eintragen
hre Daten zu sehen. Berühren Sie das
htigung zu löschen.
Soll
Ist
Ernährungspyramide (heute)
Soll
Körperliche Aktivität (heute)
200 ml
250 ml
Nichts

Figure 3.10: Entering an amount of drinking to the application

After entering this amount, the glasses for giving feedback about the daily amount are getting filled. The ten glasses filled up would represent the recommended amount of 1.5l per day.

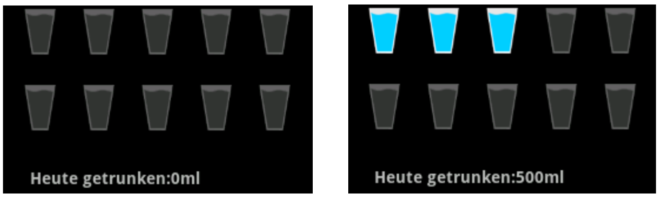
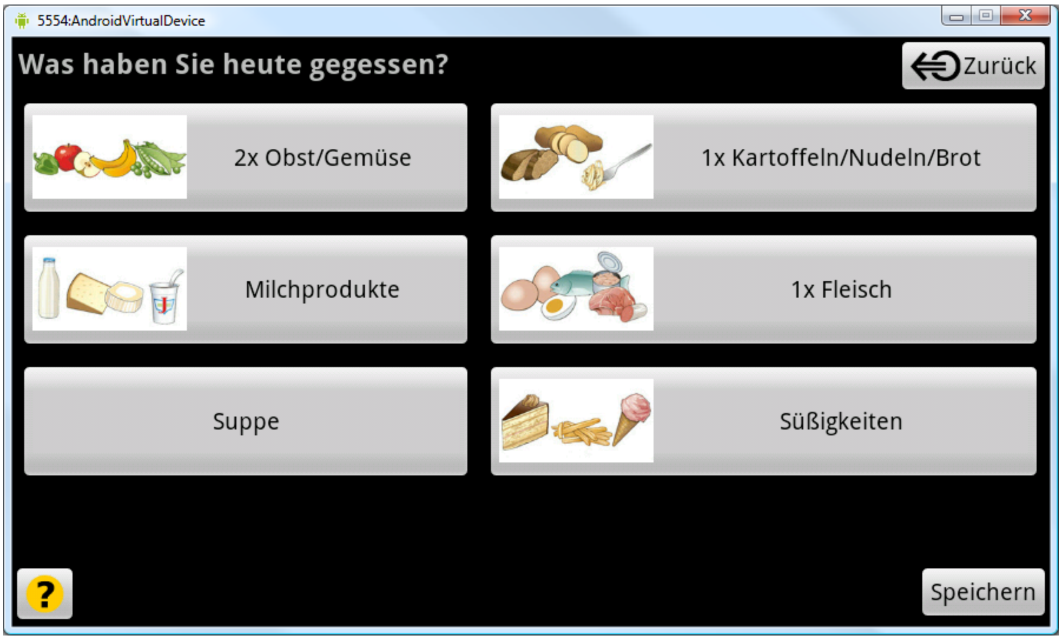


Figure 3.11: Empty glasses compared with three glasses drunken

To enter nutrition intake to the application, the activity shown in Figure 3.12 appears. If an item is touched, the number in front of the item increases its amount, so the user is able to enter more different types of food at once. If an item is touched five times it starts at zero again.



5554:AndroidVirtualDevice

Was haben Sie heute gegessen?

Figure 3.12: Entering nutrition intake to the application

To give feedback about the daily amount of food already eaten a picture of a nutrition pyramid can be shown where one half is nearly transparent. If a certain type of food has been eaten and documented with the application, the other half of the pyramid is filled (shown in Figure 3.13). This leads to the advantage that the user can easily follow how healthy his nutrition during this day was and may motivate to eat an apple in the evening instead of cookies.

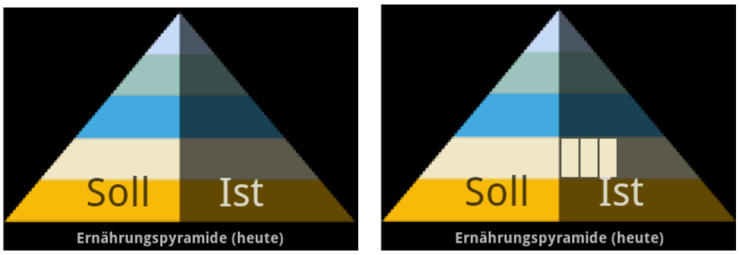


Figure 3.13: Nutrition pyramid empty compared with three portions of carbohydrates

The same was made for the amount of the activity where the actual amount is compared with a target amount.

3.8 *Calculation of vital parameters*

Some vital parameters are calculated by the application, for example, to give feedback to the user on his lifestyle and health data as entered during the last days, compared to the recommended values as a benchmark. In this application the values are represented by the bars shown in Figure 3.14, but they could also be used to control a metaphor like a garden of flowers (mentioned in 2.5). The calculation is performed daily at 4 o'clock in the morning and calculates parameters based on the data of the last three days.

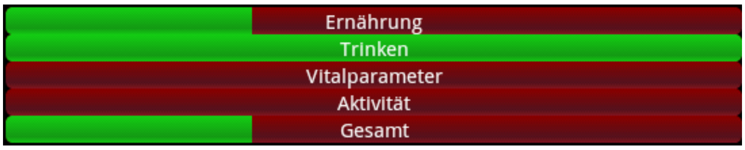


Figure 3.14: Scale of the vital parameters

Nutrition Vital Index

This index is calculated based on the sum of absolute variations of vegetables, carbohydrates, milk and meat from the ideal. As ideal a ratio vegetables:carbohydrates:milk:meat with 5:3:2:1 was recommended by a nutritionist.

The application is designed in a way that allows an easy change of the ideal ratio within a menu.

If the sum of the absolute variations is lower than:

- 10% the vital index of nutrition is 3 points
- 25% the vital index of nutrition is 2 points
- 50% the vital index of nutrition is 1 points
- 100% the vital index of nutrition is 0 points

Activity and Drinking Vital Index

This index is calculated based on the time an activity was performed multiplied by a factor of intensity. As and weekly duration of 150 minutes of medium intensity activity and 75 minutes of high intensity activity are recommended for elderly users (2.7), the factor for gardening, walking and shopping is set to 1, the factor of sport is set to 2. Also in this category points can be reached as well as for the amount of drinking.

If a certain amount of points per week is not reached in a dedicated category, advices can be displayed by the application like “Unsweetened herb tea is a proper alternative to drinking water”.

3.9 *First adoption of the system in a study*

A single user version of this prototype application was used for an application study of the project LiKeIT (Lifestyle Monitoring) of the AIT supported by the "programm benefit" of the Austrian Federal Ministry for Transport, Innovation and Technology. The setup for this study is shown in Figure 3.15. The RFID token, Samsung GT S5230N (*Samsung Group, South Korea, Seoul*), is able to read the data from a blood pressure meter, a body weight scale and the ID card. On the ID card the login data for the web system is stored, so the ID card has only to be touched once, as the application stores the login data.

The token is connected to the Archos 70 Internet Tablet (*Archos, France, Igny*) which is running the application. The tablet is able to communicate with the application server via a WLAN router (*3G-6200nL, EdiMax Technology Co., Ltd., China, Taiwan*) connected to a 3G USB modem (E173, *Huawei Technologies Co. Ltd, China, Shenzhen*).

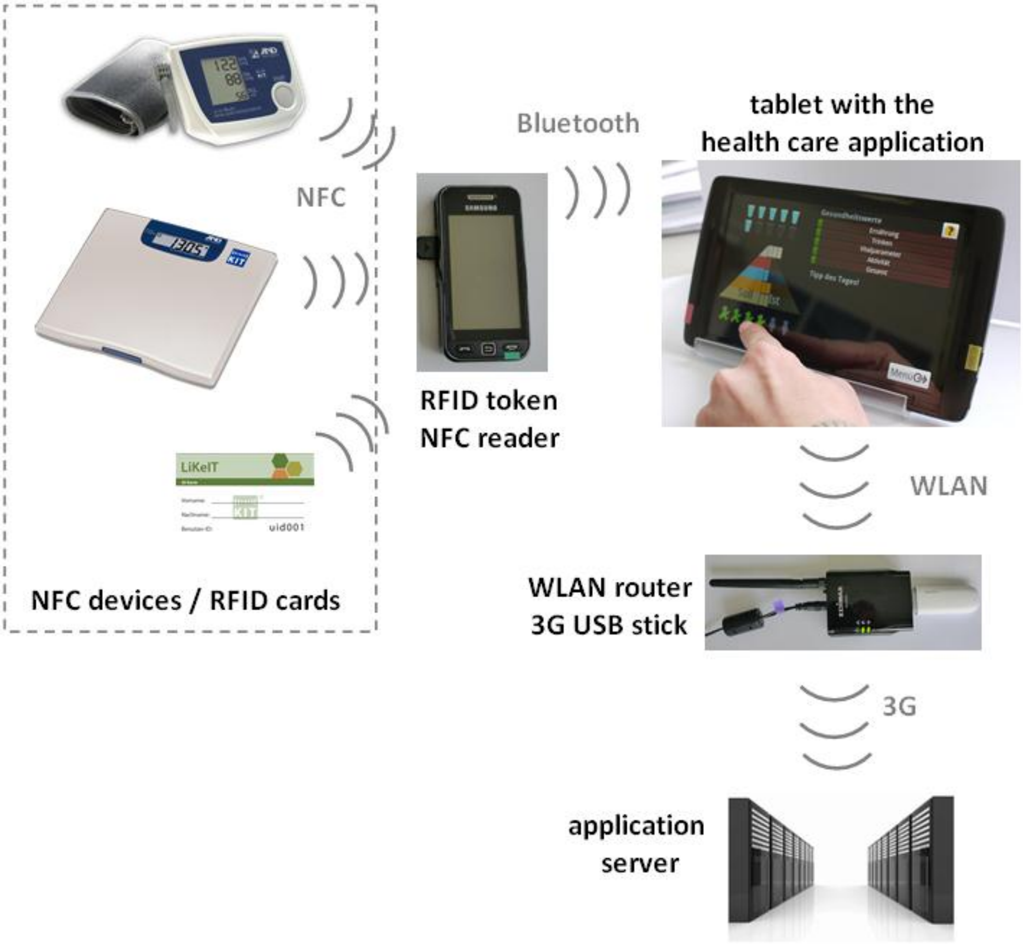


Figure 3.15: Setup of the study system

Ten people aged over 55 years were included in this study. They were told to daily monitor their nutrition intake, drinking amount and activity over a period of about six weeks. Furthermore, some of them also opted in to document their blood-pressure and body-weight, with the use of KIT devices read by the RFID token. As this study was not finished at the end of this thesis, there are no final results so far, but intermediate results will be outlined in the discussion section (*4.6*).

4 Discussion

A multiuser terminal for eHealth applications has been developed leveraging the concept of a given context when NFC cards are brought to the terminal in a timely correlating manner. These cards could either represent users (patients or physicians) or data (like medical data, lifestyle data or configuration data). With different combinations of cards, it is possible to change the context of the application. This context can be stored persistently or also be predefined for different users.

On the one hand it was dealt with the topic of providing the patients an easy way to enter data by using the touchscreen because it is an intuitive method for interacting with the application and an NFC reader for authentication as well as collecting sensor data. On the other hand the issue of easing the access to relevant patient data for a physician was worked out.

Furthermore, the application includes features for giving the patient visual feedback about his health and lifestyle parameters. Like the nutrition pyramid where one half grows due to the nutrition intake to show whether the nutrition is balanced. The amount of drinking and activity is compared to a recommended target amount. The problem of transparency of data has been dealt with by showing a picture of the physician who is allowed to access the data. So the patient knows who has the permission and can easily revoke it by touching the picture.

In the following a critical eye is cast on the current thesis.

4.1 Android

The use of Android as an operating system for health care applications comes along with advantages as well as disadvantages.

Android is a free and open mobile platform developed by the Open Handset Alliance[16], a group of 80 technology and mobile companies. There are a lot of different mobile operators (e.g. *Vodaphone, T-Mobile, China Mobile Communications Corporation,* etc.), handset manufacturers (e.g. *Dell, Huawei, HTC Corporation, Samsung Electronics* etc.), semiconductor companies (e.g. Intel Corporation, ARM, Gemalto, Nvidia, etc.), software companies (e.g. Google Inc., NXP Software, Access Co. Ltd., etc.), and commercialization companies (e.g. Borqs, Noser Engineering Inc., etc.), working together. Based on a study from Gartner Marketing Research[17] Android could become the most popular mobile operating system by the end of 2011 with a market share of 38.5 percent. So it should overhaul Symbian, which was the leader at the end of 2010 with 37.6 percent and is forecast to fall to 19.2 percent 2011. Based on this study Apples iOS will reach the second place by the end of 2011 with 19.4 percent.

As Android is supported by a lot of companies, and also marketing researchers predict a huge increase of market shares, the supply of a huge variety of different devices running on Android for reasonable prices should be guaranteed.

Another advantage of using Android is the possibility, that all features of the Android framework can be included into the application. This is shown in the prototype using the camera for taking a picture. Also all other features that are provided by Android can be included like calling the helpdesk or a physician by just touching one symbol on the screen. Or, there is also the opportunity to include environmental data into the application for example the current weather. Then an advice like “A little bit of walking could enforce your wellbeing!” could only be displayed when the sun is shining.

[16] http://www.openhandsetalliance.com/

[17] http://www.it-business.de/news/marktforschung/trends/studien/articles/310763/

This enables a huge potential of possible applications where some of them are hard to imagine. For instance there is an application called “Instance Heart Rate” available for free, where putting a finger onto the internal camera is used to detect the heart rate of a person. On devices without flashlight this has to be done in a well-lit environment, on devices with flashlight the finger is illuminated for a better detection. It is amazing how accurate this application works.

Although Linux is a multiuser system and Android is based on a Linux kernel, Android does not support the ability that multiple users can have their own account. So the topic of handling several users was implemented in the application.

4.2 Tablet PC as eHealth Terminal

For developing this application a tablet running on Android 2.1 without built in NFC has been chosen. That was, because it was the current version of Android and built in NFC was not available to that time. As new versions of Android are launched quite fast the current version for tablets is 3.0 and the first tablet with NFC was already presented on the CeBIT 2011 by *TazTag.*

Using a tablet as an eHealth terminal allows an intuitive interaction with the user as no keyboard or mouse has to be operated. Furthermore, it is a common device for a reasonable price which can be purchased in various specifications. So, if for instance for a project only WLAN and no 3G is needed with a screen size of 7 inches an appropriate tablet is available, if for another project HSDPA and a screen size of 10 inches is needed also a suitable device can be found.

4.3 Field of application

The concept could be applied for applications on a physician’s PC. A frequently used workflow, when a patient comes to a physician, is that the physician has to log into the right study system (sometimes there are more than one) with his username and password, and he has to query for the patient whose data he wants to look at. It would be a great alleviation if the patient comes in and puts his ID card on the table and the application data popped up on the screen.

Another possibility for this system would be, if a physician performs his ward round at hospital, carrying his tablet connected via WLAN to a central-data-storage (described in the article of the EHEALTHCOM magazine Nr. 2, 2010 from Bernd Greiss[18]). At every bedside there could be an ID card of the patient, and when touched, the current medical record appears on the tablet. An RFID wristband may also play the role of the patient ID card.

Furthermore, the terminal could be placed in a nursing home, where elderly people are able to insert data in an easy way, and the attendant has the ability to look at the data. This is actual the field this application focuses on, i.e. the domain of AAL.

[18] http://www.e-health-com.eu/details-news/mobil-digital/

4.4 Contactless technologies

NFC as contactless technology was used because the application was built into an already existing framework of NFC devices. Certainly data could also have been acquired only by using Bluetooth Personal Health Care Devices (defined by the CHA). However, one main advantage of NFC compared to Bluetooth is that the devices do not have to be paired which is an ease for the users.

Furthermore, NFC can be used to authenticate users as well as collecting sensor data. This has been shown in this thesis. However, the used RFID-reader does not support collision detection meaning that it is not possible to handle more cards simultaneously. So in this prototype application, the users have to put their cards close to the reader one after the other.

Furthermore, it is not possible to detect if a card is close to the reader for a longer time. This would enable the feature that a user logs out when takes away his card from the reader. But this would include the problem that a user may forget his card, which could not happen if the RFID tag was included in a wristband and the login performed by bringing the hand near the reader.

4.5 Quality attributes of the developed application

4.5.1 Extensibility

The application is developed in a way that allows expansion as well as modification. At the moment there is an authentication module implemented which has the authentication data stored as fixed values. However, as it is a separate module implementing only one method, authenticate(username, password), it can easily be replaced with another module which for example allows authentication via a network.

The same applies for the storage. The measurement data is not encrypted in this prototype application, but if it becomes a requirement to store the data cryptographically secure, the storage only has to be replaced by an encrypted database. So only one module, and not the whole application, has to be changed.

4.5.2 Usability

Especially when developing an application on a tablet for elderly people care has to be taken in terms of usability. It was not a requirement for this thesis to focus on usability but a few things that should be cared about or avoided were figured out. For instance, creating buttons in a size appropriate for elderly users, using text sizes a bit bigger than usual or avoiding that the users have to insert text with the on screen keyboard.

Another point that I suppose to be important are the different possibilities to interact with the touchscreen. For instance there is a difference between a short touch and a long touch to an element. A long touch is usually used for entering a menu. If the item has no menu or it is removed to another place, the long touch and short touch can in my opinion set to be the same, for easing the interaction of a user. Scrolling a screen by swiping or multi-touch gestures, like pinching for zooming, rotating and so on, which enables a user familiar to these powerful tools, but it has to be evaluated if they are suitable in applications for elderly people.

4.5.3 Testability

The modules which are started via an intent can easily be tested automatically using the *JUnitTest* plug-in in Eclipse. This is for instance done for testing the InterestResolver. Different intents containing data which could be sent to the InterestResolver are sent and the outcome whether the data is stored in the database or a user is authenticated can be checked.

If the acquisition of data works from the token to the terminal cannot be tested using JUnitTest. Therefore, some tests were created using ZetaTest-Tool[19] (version 1.0.2.98), including for instance if the transmission of data also works after the Bluetooth connection was lost and reconnected again.

[19] http://www.zeta-test.com/

4.5.4 Modifiability

Due to the modular layout of the application it is possible to modify or reuse parts of the application. It is also possible to add modules. For instance, a module for communicating to a server has already been added by another developer. He just added a Broadcast Receiver which is listening to an intent announcing that a measurement was stored. After this event a Service is started that performs the synchronization of the data with the server.

4.5.5 Portability

If an application is developed on Android, it is compatible with all devices that have the same or a higher version of Android (the same or higher API level). Furthermore, the features needed by the application have to be available on the device. For instance if the application uses a built in camera for taking pictures, the device has to have a built in camera. For this purpose, in the Android Manifest file it has to be declared that a camera is used by the application. At the installation, the user gets the notice, that the application needs access to the camera.

4.5.6 Interoperability

For the interoperability in health care applications mainly the Continua Health Alliance defines adequate standards in their guidelines.

In terms of these guidelines, the tablet is the Application Hosting Device and the token could be seen as a Personal Health Device. Unfortunately such kind of device, for collecting data is not defined in the guidelines so far. As standard for the transmission of data from the token to the tablet, the JSON format was chosen. For the implementation of a synchronisation with a server an adequate standard is the Health Level 7 (HL7), with which would have to be dealt with.

4.6 Actual status and forecast

As already mentioned in the results (*3.9*) a first prototype of the system is currently in use in an application study. Ten people aged over 55 years have been attending this study since 14th of April 2011. They daily monitor their nutrition intake, drinking amount and activity. Furthermore, some of them are also documenting their blood-pressure and body-weight, with the use of KIT devices read by the RFID token.

I assisted the enrolment and my first impression was that they are handling the application quite well. I was amazed by a 78 year old woman who had no problem dealing with the application, although her hand shivered pretty badly. All of the participants seemed enthusiastic about the system.

Due to the fact that the study is still running there are no final results so far. As temporarily results after four weeks, a look into the web system shows that the amount of transmitted data varies significantly. One user has a total number of transmissions of more than 2500. It can be seen in the web system that this is caused by transmitting data with the token several times after a measurement. This may be improved telling the patients that the acoustic signal on the token means that the data was transmitted at the scheduling. Furthermore, the token or the tablet could be modified to detect whether a value already was read out or not and a message telling that this data already was transmitted could be displayed as a visual feedback. Another four users transmitted each about 350 values during these four weeks. The terminals of the other users have problems with the transmission, either with the configuration or with the connection. Some of them transmitted data for some days or weeks, and two of them have no transmission so far.

On average there were about three phone calls with the helpdesk per person. All of them liked the system and declared to daily monitor their data. Although some of them have faults with the connection the application for them works as it should because the users can monitor their data and get feedback from the system also without the internet connection. When a connection will be built up somewhen, all data that is not transmitted so far will be transmitted.

After six weeks of monitoring the participants will return the devices and give feedback, so the system and especially the connection to the web system can be improved. Furthermore the data storage of the tablets will be read out and compared to the data in the web system. This will show whether missing data in the web-system has its origin in a fault of the connection or the patients did not track their data any more.

After that another ten persons will get the devices and evaluate it for additional improvement of the system.

BIBLIOGRAPHY

Bundesministerium für Gesundheit, 2010, Gesund genießen, *Kompetenzzentrum Ernährung & Prävention/AGES, Bundesministerium für Gesundheit, Österreich*, 1. Auflage Dezember 2010

CHA Guidelines, 2010, Continua Design Guidelines Version 1.5 April 8, 2010

Consolvo 2009: Consolvo, S., Landay, J., A., McDonald, D., W., (2009). Designing for Behavior Change in Everyday Life. *IEEE Computer Society* 0018-9162/09

Dinu 2007: Dinu, V., Nadkarni, P., (2007) Guidelines for the effective use of entity-attribute-value modeling for biomedical databases. *International Journal of Medical Informatics* 76, 769-779

Eggebraaten 2007: Eggebraaten, T., J., Tenner, J., W., Dubbels, J., C. (2007) A health-care data model based on the HL7 Reference Information Model. *IBM SYSTEMS JOURNAL*, V=L 46, NO 1, 2007

Gerdes 2007, Gerdes, S., Diploma Thesis: Rollenbasiertes Sicherheitskonzept für Krankenhäuser unter Berücksichtigung der aktuellen Entwicklung in der Gesundheitstelematik. *Universität Bremen*, Fachbereich 3, Mathematik und Informatik

Greenstein 1988: Greenstein, J.S., Arnaut, L.Y., In: *Handbook of human-computer interaction*. ed. by Helander, M., Amsterdam, 1997, 1318-1323

Haigh 2005: Karen Zita Haigh, Ph.D., Liana M. Kiff, Geoffrey Ho, Ph.D. (2005) The Independent LifeStyle AssistantTM (I.L.S.A.): *Lessons Learned*, Honeywell Laboratories

Holzinger 2002, Holzinger, A., User-Centered Interface Design for Disabled and Elderly People: First Experiences with Designing a Patient Communication System (PACOSY) Computers Helping People with Special Needs, *Lecture Notes in Computer Science*, 2002, Volume 2398/2002, 467-484, DOI: 10.1007/3-540-45491-8_8

Morak 2007: Morak, J., Kollmann, A.,& Schreier, G. (2007). Feasibility and usability of a home monitoring concept based on mobile phones and near field communication (NFC) technology. *Stud. Health Technol. Inform.*, pp. 112-6.

Schreier 2008: Schreier, G., Hayn, D., Kastner, P., Koller, S., Salmhofer, W., Hofmann-Wellenhof, R. A mobile-phone based teledermatology system to support self-management of patients suffering from psoriasis. *Conf Proc IEEE Eng Med Biol Soc*. 2008; 2008:5338-41.

Titze 2010: Titze, S., Ring-Dimitriou, S., Schober, P.H., Halbwachs, C., Samitz, G., Miko, H.C., Lercher, P., Stein, K.V., Gäbler, C., Bauer, R., Gollner, E., Windhaber, J., Bachl, N., Dorner, T.E. & Arbeitsgruppe Körperliche Aktivität/Bewegung/Sport der Österreichischen Gesellschaft für Public Health (2010). Bundesministerium für Gesundheit, Gesundheit Österreich GmbH, Geschäftsbereich Fonds Gesundes Österreich (Hrsg.). Österreichische Empfehlungen für gesundheitswirksame Bewegung. Wien: Eigenverlag.

WHO 2001: World Health Organization. Men Ageing and Health. Achieving health across the life span. Geneva : *World Health Organization*, 2001. 01WHO/NMH/ NPH 01.2.

PICTURE CREDITS

Figure 2.1: Lifecycle of an activity:
http://developer.android.com/reference/android/app/Activity.html

Figure 2.6: Nutrition pyramid of the Austrian Federal Ministry of Health:
http://www.bmg.gv.at/home/Schwerpunkte/Ernaehrung/Empfehlungen/DIE_OeSTERREICHISCHE_ERNAeHRUNGSPYRAMIDE

Figure 2.2: CHA Reference Topology:
CHA Reference Topology: Continua Design Guidelines Version 1.5 April 8, 2010, Figure 3-10 Reference Topology, p. 10