Ulm University | 89069 Ulm | Germany

Faculty for Engineering and Computer Science
Institute for Databases and Information Systems

Increasing User Motivation of a Mobile Health Application based on applying Operant Conditioning

Bachelor Thesis at Ulm University

Submitted by:
Michael Götze
michael.goetze@uni-ulm.de

Reviewer:
Prof. Dr. Manfred Reichert

Supervisor:
Dipl.-Inf. Marc Schickler

2015

Version of October 22, 2015

Satz: PDF-LaTeX 2_ε

Abstract

In times of demographic change, especially in western countries, it's getting more and more important to come up with cheap and efficient solutions to care for older people and their medical conditions. Now that smart phones are widely spread, they should be used to make life easier for their users. This characteristic can be achieved e.g. by developing mobile health applications for daily usage. In this bachelor thesis an approach is presented how the motivation of diabetes diseased people can be increased, to check for their own health, by applying concepts of operant conditioning. These concepts will be presented and how they result in a psychological impact on the user. Furthermore, an implementation of the designed concepts will be shown.

Acknowledgements

First I would like to thank my supervisor Marc Schickler who always supported and trusted me with my ideas. Additionally, I like to thank Johannes Schobel who gave me the possibility to introduce my concepts into his project, the BloodGlucose Diary. I would also like to thank my family, especially my sister Stephanie Götze, who proof-read this thesis. Last but not least I like to thank my lovely girlfriend Lina Gerigk who always supported me and took care of my messy time management.

Table of Contents

1 Introduction

In health care a lot of different mobile devices are providing the possibility to act as health adviser or measuring data such as blood sugar or pulse. These devices are suffering from the fact that they are only used for one interaction and can thereby be easily forgotten. Nowadays the possibilities of smart phones are increasing every year as well as their distribution. This provides an opportunity to pool the functionality of multiple medical devices into a single one. They are also providing a larger amount of technical capabilities to interact with their users or e.g. an emergency contact of a disease person. Because of the rising numbers of people falling sick with diabetes and the enlarging number of elderly people caused by the demographic change, it’s necessary to provide cheaper and more efficient solutions to be able to care for these people in the future [18]. The documentation of their blood glucose levels is important to identify the cause of blood glucose variations and thereby to develop a successful treatment. This effort requires a lot of discipline and motivation. Most diabetics document their blood glucose levels badly [36]. Therefore, it’s essential to develop a system that motivates users and also controls them if they miss checking regularly for their own welfare. A way achieving this objective is by the use of gamifications (see passage 2.2). According to S. Deterding, applying game elements in a non-game context can lead to a desired user behavior [19]. Such a system can be implemented on recent mobile platform such as Android and iOS, where already a huge amount of health applications exist, which are affording the potential to be improved. Since smart phones were invented to make life easier for their owners, it makes sense to particularly achieve this goal for old people as well. Especially people diseased with diabetes type 2 need a special treatment to improve their health and safety. Otherwise, this disease can lead to serious secondary diseases like blindness, heart attacks or amputations. A very big problem is that these people aren't used to handle their sickness on their own, because they weren't affected by diabetes for their whole life. Therefore, it's necessary to remind them to check for their own welfare. Due to the demographic change and it's resulting amount of elderly people, the costs for their medical treatments are rising massively [18]. It's helpful to lower costs for relatives and health insurances by improving the physical health of diseased people. Therefore, their independence to check for their own welfare needs to be increased, what results in secondary disease prevention.

1.1 Concrete Objective

This bachelor thesis deals with the increase of user motivation for a healthcare application. Therefore, different concepts, which are build on operant conditioning, to increase user motivation were developed and will be presented. These concepts are based on improving the treatment of diabetics by supporting and motivating them in their daily necessary tasks. The developed concepts will be implemented in an Android application, which allows diabetics to store their blood glucose records and display them. The implementation focuses on offering an intuitive and error preventive user interface.

1.2 Thesis Structure

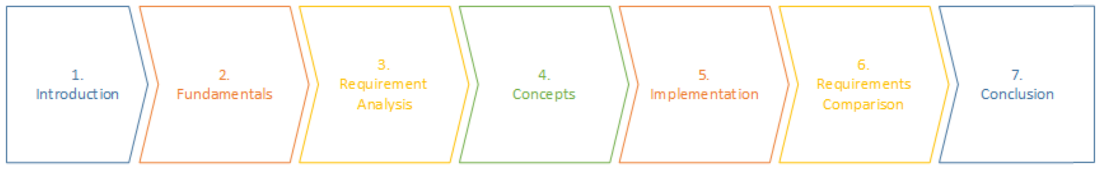


Figure 1.1: Thesis Structure

In *Fundamentals* (see chapter 2) the necessary information about the idea behind operant conditioning and how this is connected to the gamification concept are presented.

Finally, the basic knowledge of Diabetes is provided, to explain how the chosen and developed concepts are useful in this context.
Requirements (see chapter 3) defines the foundation and features of the exemplary implementation. The next chapter *Concepts* (see chapter 4) introduces the designed concepts and their desired impacts on the users. The resulting implementation of the designed concepts in a mobile application is shown in *Implementation* (see chapter 5). In *Requirements Comparison* the implementation result is evaluated regarding the initially defined requirements. Additionally, ideas for future extensions are presented.

2 Fundamentals

This chapter provides the basic information which are required to reproduce the thoughts of the developed concepts (see chapter 4). Hereby the psychological background of gamifications (operant conditioning) is explained, as well as the idea of gamification itself. Furthermore, the necessary knowledge about Diabetes is provided, to understand the reasons of the made decisions in the concept design.

2.1 Operant Conditioning

Operant Conditioning, also called *instrumental learning*, marks a method of learning in which a behavior is either intensified, when it's followed by reinforcement or weakened, when it's followed by punishment (see figure 2.1). Thereby the behavior will occur less frequently in case of punishment and more frequently in case of reinforcement.
Edward Lee Thorndike (1874–1949) discovered the "Law of effect" when he observed cats, which needed to pull a cord or push a pole to release themselves from a box [35]. By repeating this trial, the cats needed decreasingly time to release themselves. The basis of this learning method is the effect that the consequences of an action influence the behavior. Skinner extended Thorndike's "Law of effect" by the use of reinforcement. Therefore, he's known as the father of operant conditioning [29].
Operant conditioning is separated into two main categories of behavior influence (punishment and reinforcement). These main categories are also separated in each case into positive and negative shape, as shown in figure 2.1. Positive represents hereby the presence of a stimulus and negative the absence of a stimulus [35]. Here some examples to make the concept better comprehensible. A positive reinforcement could

be e.g. giving a star to a pupil when he/she gave a correct answer and a negative reinforcement could be e.g. turning of the sound of an alarm. According to this a positive punishment could be e.g. forcing a pupil to do an imposition when he/she did something wrong and a negative punishment could be e.g. to remove the pupil's toy.

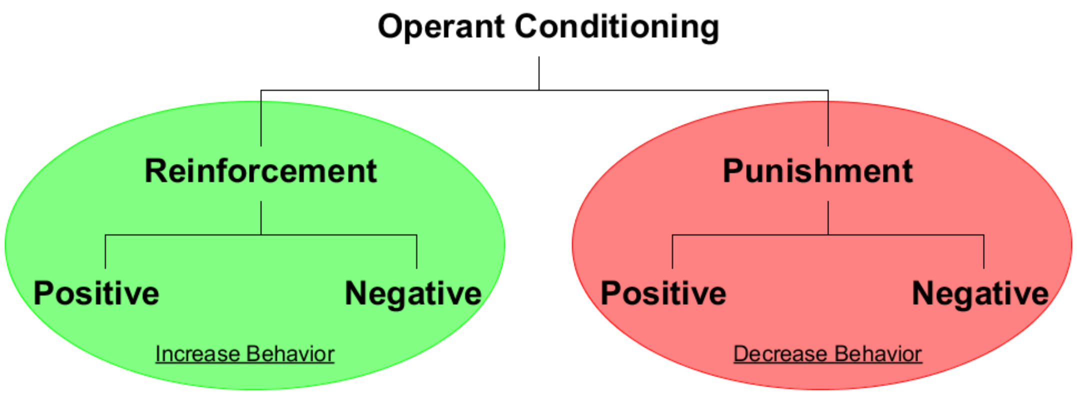


Figure 2.1: Operant conditioning

2.2 Gamifications

The term *gamification* primarily represents a process of an application service which increases user motivation and bindings. J. Hamari et al. defined the *gamification* concept as follows:

> "*A process of enhancing services with (motivational) affordances in order to invoke gameful experiences and further behavioral outcomes.*" [23]

Gamifications can be used to introduce gameful elements into a non-game context in order to influence behavioral outcomes. *Gamifications* consists of a motivational affordance and psychological or behavioral outcomes. In [23] J. Hamari et. al. presented a relatively large variety of different motivational affordances, which were included in applications in the context of different studies. Amongst others there are presented concepts such as Achievements, Feedback, Rewards, Progress or Challenges. They're either used in a learning or health context for the implemented application. These studies mostly investigate behavioral outcomes, what is aimed to produce positive effects regarding the user motivation [23]. Gamifications use the principle of operant conditioning (see chapter 2.1) to achieve the desired change in user behavior. This change can either reinforce or weaken a certain behavior [27]. In [32], the main key elements of gamifications are, according to Palmer, Lunceford and Patton [31], defined as:

a.) Progress paths

b.) Feedback and reward

c.) Social connection

d.) Interface and user experience

These elements also compose the foundation of health-related gamification concepts [32]. David Lenihan suggests that maximizing engagement, in health-related applications, can be achieved by introducing *c.) Social connection* [28]. The opportunity to be connected to family, friends or caregivers can result in a positive effect. The *b.) Feedback and reward* element can act as indicator of success, if it's executed at the right time, and is thereby an important feature for motivation. Rewards can be distributed in- and outside of the application. Achievements are a widespread implementation of gamifications and belong to the *a.) Progress paths* element. Achievements benefit from having clear goals/affordances which the user has to fulfill. Hamari et al. showed a definition and implementation of achievements in a health-related application [22].

2.2.1 Gamifications in Healthcare Context

The use of gamifications in healthcare applications is vast, since increasing user motivation for such applications results in an improvement of personal wellness and healthcare [23][32]. Medical treatment often contains repetitive, boring and/or awkward routines for the patient. These routines can be improved by sympathizing them with gamifications. A big amount of examples can be found at applications which are assisting the users with their reduction of weight. E.g. SlimKicker is a smart phone and computer application

where diet and fitness objectives are turned into a level-up game [15]. They introduced challenges and feedback to make a usual routine more pleasant.
The use of social sharing has spread widely over the last years. According to Lenihans theory, that sharing data of your current health status with family, friends or caregivers will have a positive effect of the users well-being, applications like FitBit [5], Atari Fit [2], Nike + Running [9], Map My Fitness [8] and many others, added the social sharing concept [28]. According to Hancock et al. an individual feedback, to the users of a health application, is essential to grant a positive user experience [24]. For the gamification design, it's also important to mind the main context of the application as well as the user's age [30].

2.3 Diabetes

Diabetes is a metabolic disease which results in a disturbance of carbohydrate reduction. The result of such a malfunction is an increased blood glucose value. Depending on the Diabetes type, this malfunction is caused by a different reason. [38][26]

Type 1 Diabetes mellitus is based on failure in the immune system, whereby it mistakenly regards the islet cells, which are producing insulin, as danger and destroys them. The insulin is a necessary hormone, which is required to convert carbohydrates into energy in order to transport it to body cells. Usually Type 1 Diabetes occurs in childhood and adolescence, the resulting treatment for the patients are lifelong insulin injections depending on their nutrition. The reason for the cell destruction is still unknown. [37][26]

Type 2 Diabetes mellitus usually occurs in adulthood and is caused by advancing age, obesity, unhealthy nutrition and the lack of physical activity. This type of diabetes suffers either from the pancreas doesn't produce enough insulin or the cells can't process the insulin properly. This results in an insulin resistance what can be treated, in the majority of cases, by improving nutrition, increasing physical activity or oral medication. [26]

These two types of diabetes are the most commons, whereas Type 2 Diabetes represents 90% of all diabetes cases worldwide. In 2014, 9% of the world population fell sick with diabetes what are more than 387 million people. The outcome of a bad treatment are permanently high blood glucose levels that can lead to serious diseases such as cardiovascular disease, blindness, kidney failure, and lower-limb amputation. More than 4.9 million people died from the consequences of diabetes in the year 2014. [26]

3 Requirements Analysis

3.1 Functional Requirements

Functional requirements cover the main functions of the demo application which should be offered for the user. The following table presents an overview of these functions based on the developed concepts to apply operant conditioning.

3.1.1 Reminder Alarms

i. Create *Reminder Alarms*:
The user needs the possibility to create reminder alarms. These consist of a start and end time. Within this time period a blood glucose record should be transferred.

ii. Create *Reminder Alarms*:
The user needs the possibility to create reminder alarms. These consist of a start and end time. Within this time period a blood glucose record should be transferred.

iii. Save *Reminder Alarms*:
All reminder alarms need to be saved on the device.

iv. Delete *Reminder Alarms*:
It needs to be possible to delete each reminder alarm.

v. Turn *Reminder Alarms* on and off:
It has to be possible to turn each reminder alarm on and off.

vi. Overview of all *Reminder Alarms*:
The user needs the possibility to see all existing reminder alarms and if they're active.

vii. *Reminder Alarms* notification:
Each alarm should only trigger if no blood glucose record was transferred within the time period and the reminder alarm is active. The user should be notified 30 min and 15 min before the end time of each reminder alarm. If the user didn't accomplish to add a blood glucose record within the time, the application should also notify the user. An Android notification should be displayed and a sound should be played.

3.1.2 Monitoring

i. Add *Emergency Contact* data:
For emergencies a phone number and email address can be set to notify an individual *Emergency Contact*.

ii. Notify *Emergency Contact* by SMS:
In an emergency the *Emergency Contact* can be notified, in the background, by SMS.

iii. Notify *Emergency Contact* by email:
In an emergency the *Emergency Contact* can be notified, in the background, by email.

iv. *Emergency Contact* notification in case of a critical blood glucose record:
The user can allow the application to notify the *Emergency Contact* if the transferred record is in the critical area.

v. *Emergency Contact* notification in case, no record was transferred within a reminder alarm period::
The user can allow the application to notify his/her *Emergency Contact* if he/she missed to transfer a blood glucose record within the time period of an active reminder alarm.

3.1.3 Achievements

i. Check for Achievements:
   If the user transfers a blood glucose record, the application needs to check if the next reachable Achievement is accomplished.

ii. Save reached Achievements:
   When an Achievement is accomplished it needs to be saved. Therefore, a JSON file has to be stored on the device.

iii. Load Achievements:
   All accomplished and unaccomplished Achievements need to be loaded for further processing from the JSON file.

iv. Achievement board:
   The Achievement board is necessary to display all so far reached Achievements as well as the next reachable Achievements.

v. Sort Achievement board:
   It should be possible to either display all Achievements or only one category of Achievements on the Achievement board.

vi. Display next reachable Achievements:
   The next reachable Achievement of each Achievement category should be displayed.

3.2 Non-Functional Requirements

Non-functional requirements cover characteristics and qualities which the application should have. The following table presents an overview of the non-functional requirements.

i. Technical Requirements:
   The demo application is developed in Java and should run on Android Version 4.0 and higher.

ii. Usability:
It's very important that the demo application is very easy and intuitively to use.

iii. Maintainability & Expandability:
The demo application should be expandable concerning the implemented features. Additionally, it should be good maintainable by a well documented code and documentation.

iv. Correctness & Reliability:
The demo application shouldn't suffer from failures and in cases of maloperation the application should work properly again after restart.

v. Look & Feel:
The user interface design should be based on Googles material design guidelines.

vi. User Friendliness:
The demo application should encourage user friendliness by preventing possible user input errors.

4 Concepts

In this chapter the elaborated concepts are presented, which are intended to increase the binding of the user to the application. Therefore, the different concepts are applying several learning methods of operant conditioning. These are either reinforcement or punishment to influence the favored behavior change. The impact of the concepts on the user can be reinforced by combining them with each other [28]. The motivational concepts are particular based on the disease pattern of the users, to simplify the handling of their diabetes disease.

4.1 Monitoring

The concept of monitoring the user consists of informing his/her emergency contact if the users medical condition is bad (e.g. low or high blood glucose level) or he/she misbehaved. A misbehavior is defined as not measuring his/her blood glucose at regular intervals. These intervals can be defined by several Reminder Alarms (see section 4.2). Thereby the monitoring aspect can act in two different ways of operant conditioning. On the one hand it can be a positive punishment, a punishment is added, whereas the punishment is e.g. a controlling phone call from the emergency contact. On the other hand it can be a negative reinforcement, a punishment is subducted, at which e.g. the previously explained punishment doesn't occur because the user acted accurately. According to the *Law of Effect* by Edward Lee Thorndike, applying this concept will result in a behavior change of the user in the course of time [35]. Lenihan also regards the concept of sharing personal information about the state of health has a positive effect on the user [28].

4.2 Reminder Alarms

A reminder alarm assists the user to avoid that he/she misses to measure his/her blood glucose. Most with diabetes diseased people usually have to measure their blood glucose in a certain time interval, e.g. between one and two hours after they had a meal. Therefore, each reminder alarm consists of a start and end time. Within this time period the user has to measure his/her blood glucose. If the user misses doing that, he/she'll be notified 30 and 15 min before the end of the reminder alarm. At the end of the time period the user will also be notified that he/she missed to measure his/her blood glucose.
A reminder alarm can act, just like the monitoring concept, as a positive punishment or a negative reinforcement and lead to the favored user behavior.
The concept of reminder alarms can be combined with the monitoring concept by informing the emergency contact if the user didn't measure his/her blood glucose within the time period. According to Lenihan, combining these two concepts will intensify the impact on the user [28].

4.3 Achievements

Achievements are concrete defined goals, which the user can complete. The achievements are defined by considering the importance of the concrete context to the main application [33] (Blood Glucose Diary) and the clarity for the user about the required affordances to complete each achievement [21]. The presented achievement concept is classified into three categories. Each category can be completed as *daily* and *one-time* achievement, thereby each of them has different requirements and feedback.

- **Category A**
  Requirement: counts the amount of transmitted blood glucose records
  Feedback:
  - *Daily:* Awesome! Today you've transmitted XY blood sugar values!
  - *One-time:* Awesome! You've transmitted XY blood sugar values!

- **Category B**
  Requirement: counts the amount of transmitted blood glucose records, at which the blood glucose level has to be in a noncritical area
  Feedback:
  - *Daily:* Perfect! Today you've transmitted XY blood sugar values within your boundaries!
  - *One-time:* Perfect! You've transmitted XY blood sugar values within your boundaries!
- **Category C**
  Requirement: counts the amount of transmitted blood glucose records, at which the record has to be transmitted within the time of reminder alarm period
  Feedback:
  - *Daily:* Nice! Today you've transmitted XY blood sugar values within your reminder interval!
  - *One-time:* Nice! You've transmitted XY blood sugar values within your reminder interval!

When the user completes an achievement, a positive feedback should motivate him/her for his/her effort. The *XY* in the feedback represents the count of transmitted records regarding the category, which are required to complete the achievement. The values differ between *daily* and *one-time* achievements and which is the last acquired one.

- *One-time:* [1,3,10,25,50,100,200,500,1000]
- *Daily:* [2,5,7,10,15,20,25,30]

Each achievement saves the date when it's completed to show the user when he/she completed the certain achievement. The design of the *daily* achievements allows the user an ongoing experience by completing these challenges every day. Thereby a repeating progress path is provided to the user, whereas the design of the *one-time* achievements provides an ultimate progress path. Therefore, the values of the *one-time* achievements are chosen higher than *daily* ones.

5 Implementation

The developed concepts were implemented in Java and integrated into an Android application at which the user can transfer his/her blood glucose records from a blood sugar monitoring device. These records can be displayed in a chart view. Furthermore, it’s possible to set warn and min/max boundaries, which are valuating the transferred records by a color (white, orange, red), shown in Figure 5.1. A major aspect of the implementation was to maximize the usability by preventing input failures and provide simple accessibility of the new features. Therefore, the user can access the achievements menu as well as the Reminder Alarm menu from the Navigation Bar in the main menu by tapping the trophy or the alarm clock (see Figure 5.1).

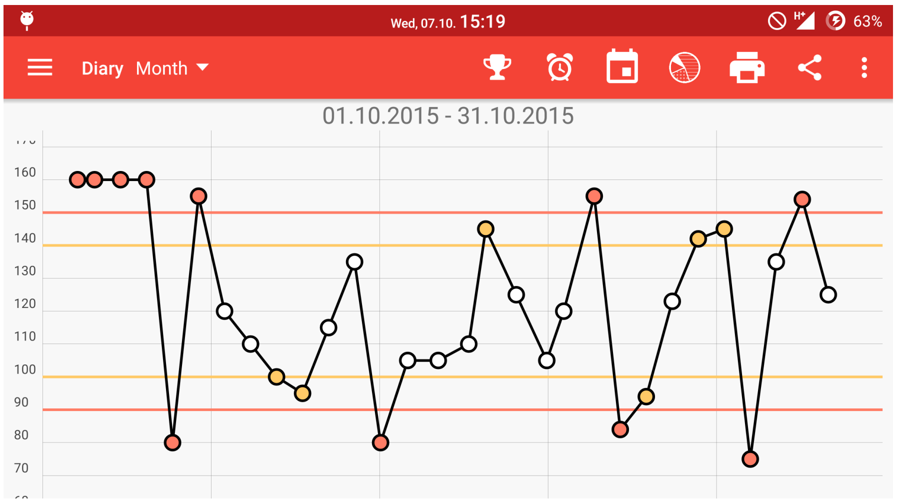


Figure 5.1: BloodGlucose Navigation Bar

5.1 Monitoring

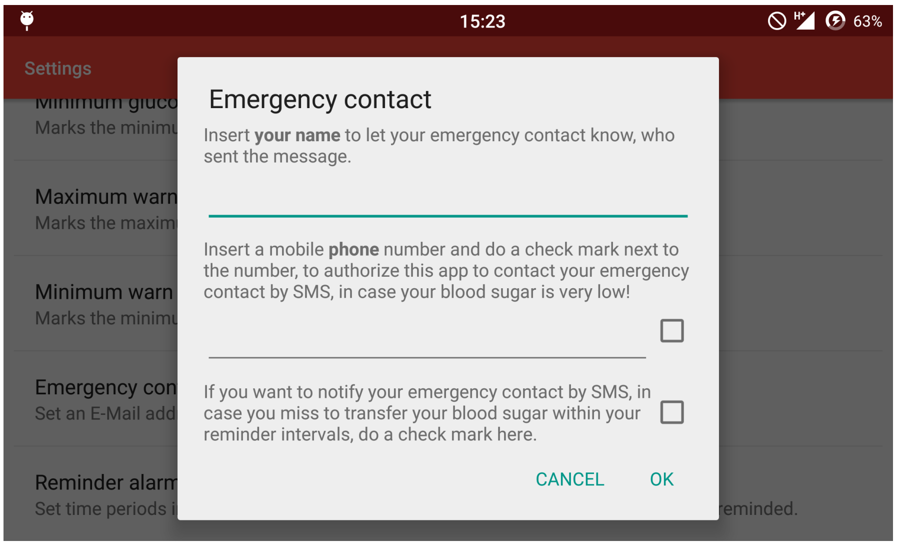
The *Monitoring* feature comprehends the authority for the application to contact the emergency contact of the user. In earlier state of the implementation it was planned to notify the emergency contact either by SMS or Email. Unfortunately it's only possible to contact the emergency contact by SMS, because it's only possible to start an Email application but the user still hast to send the Email by himself. Since the requirement for this feature is to act in the background and work without further user interaction, the notification of the emergency contact is only possible by SMS.
Therefore, a permission is required in the AndroidManifest.xml to authorize the application to send SMS. The user needs to declare the mobile phone number of his/her emergency contact (see Figure 5.2). This can be done from the PreferenceSettings which is accessible in the top right from the button with the three points (Android default). Furthermore, the user has to explicit allow the application to notify the emergency contact by enabling the checkbox next to each description (see Figure 5.2). Although the user already has to allow the application to send SMS by installing it, the double permission request was implemented to avoid unwanted costs.
For error prevention it's impossible to enable the checkbox unless a phone number is entered. If the user tries this anyway, a dialog is displayed to inform the user about the missing phone number. This accords to Schneiderman's design guidelines *3. Offer informative feedback* and *5. Prevent errors* [34].

The *Monitoring* implementation contains the review of the blood glucose records in two different contexts:

i. Critical blood glucose level

ii. End time of a *Reminder Alarm*

The first one takes place when the user transfers his/her blood glucose record(s) to the device. At this moment the boundaries for the critical blood glucose level need to be fetched. These boundaries can be adjusted from the *PreferenceSettings* [14], if the user doesn't adjust them, the default values are selected. The default lower boundary is at a blood glucose level of 90 and the upper one is set at 150. The transferred value is


15:23
63%
Settings

Minimum gluco
Marks the minimu

Maximum warn
Marks the maxim

Minimum warn
Marks the minimu

Emergency con
Set an E-Mail add

Reminder alarm
Set time periods i ... reminded.

Emergency contact

Insert **your name** to let your emergency contact know, who sent the message.

Insert a mobile **phone** number and do a check mark next to the number, to authorize this app to contact your emergency contact by SMS, in case your blood sugar is very low! ☐

If you want to notify your emergency contact by SMS, in case you miss to transfer your blood sugar within your reminder intervals, do a check mark here. ☐

CANCEL OK

Figure 5.2: BloodGlucose Emergency Contact Settings

then compared to the boundaries, if it's outside of them (less than lower boundary or higher than upper boundary) the application sends automatically (if enabled) a SMS to the emergency contact with the blood glucose level, timestamp of the record and the request to contact the user. To send the SMS the default *SmsManager* is picked and executed from the background. The second context when the *Monitoring* takes places is the end of a *Reminder Alarm*. If a *Reminder Alarm* is executed at its end time, no record was transferred within the time period. Then the application also sends automatically (if enabled) a SMS to the emergency contact about the missing record.

5.2 Reminder Alarms

For the implementation of the *Reminder Alarm*s, several components needed to be implemented concerning front- and backend of the application. The designed *Reminder Alarms* consists of a start and end time. The *Reminder Alarms* are schedules by a background service, which is started when the mobile device boots. An Android Service [13] needs special treatment, first it has to be declared in the AndroidManifest.xml just like an Activity. To restart the Service when the device boots the Application needs to declare additionally a BroadcastReceiver [3] and the *BOOT_ COMPLETED* permission in the AndroidManifest.xml. When the device is finished booting, the Android System sends a *BOOT_ COMPLETED* broadcast. This broadcast is received by the implemented ReminderManagerHelper.java class which extends *BroadcastReceiver* [3]. Considering the fact that a Service is part of an Application but it runs in the background and thereby works outside of the Application, it needs to refer to the Application about their Context to get access to the data and functions of the Application. A Service can be started by the command *Context.startService()* and runs until it's explicitly stopped by calling *Context.stopService()* [13]. Everytime, when a *Reminder Alarm* is created, deleted or executed, all existing *Reminder Alarm*s are stopped and rescheduled. In Figure 5.3 the decision process of the background service is diagrammed. Each *Reminder Alarm* needs to be rescheduled to a different time (30 min earlier, 15 min earlier or exact end time) depending on the actual time and end time of the *Reminder Alarm*. Each *Reminder Alarm* is represented by a *PendingIntent* [12]. PendingIntents are holding data outside of an application. These *PendingIntents* are scheduled by an *AlarmManager* [1] which executes at a certain time. Each *PendingIntent* needs to be recognized, this needs to be done by storing every data they consist of. Otherwise, the *PendingIntent* can't be addressed anymore and consequently never canceled. This can lead to uncontrollable and undesirable effects to the mobile device.

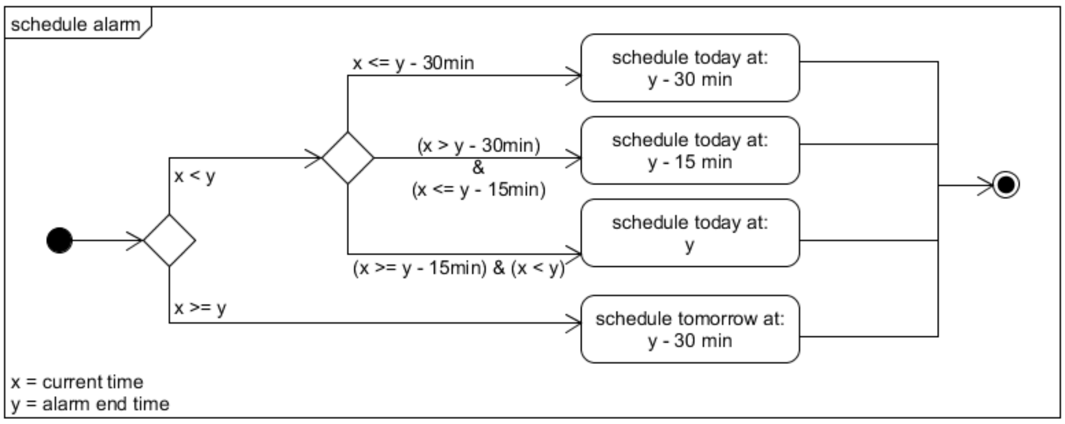
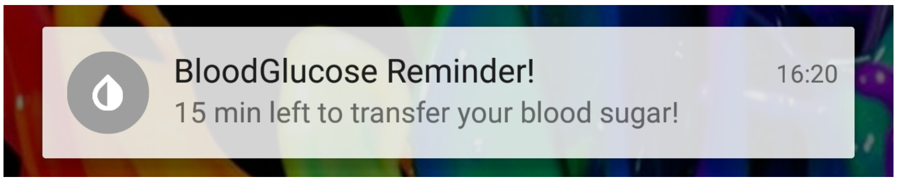

schedule alarm
x <= y - 30min
schedule today at: y - 30 min
x < y
(x > y - 30min) & (x <= y - 15min)
schedule today at: y - 15 min
schedule today at: y
(x >= y - 15min) & (x < y)
x >= y
schedule tomorrow at: y - 30 min
x = current time
y = alarm end time

Figure 5.3: Schedule *Reminder Alarm*

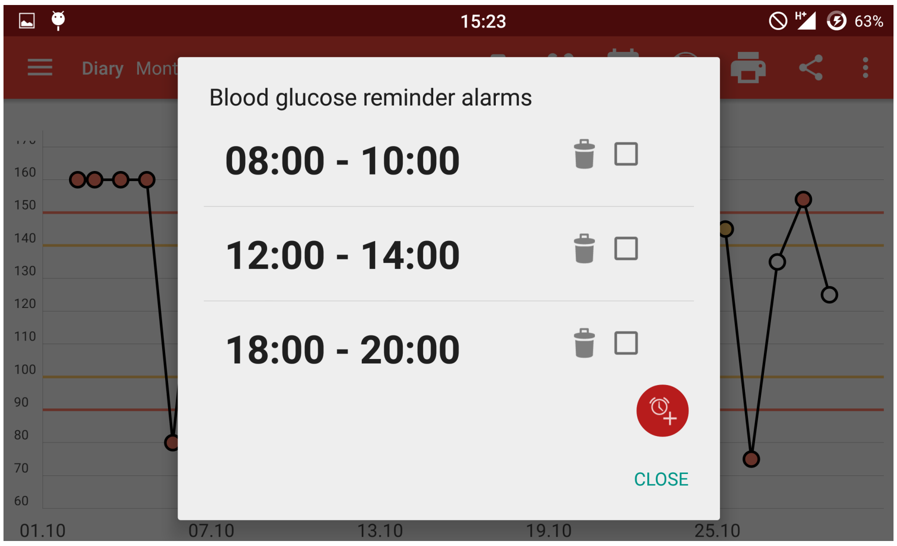
When a *Reminder Alarm* triggers, the *onStartCommand* method of the *ReminderService* is called. To notify the user about the current status, an Android Notification [10] is displayed together with the default Notification sound, a vibration and the default Notification light if possible, concerning the technical possibilities of the mobile device.



BloodGlucose Reminder! 16:20
15 min left to transfer your blood sugar!

Figure 5.4: BloodGlucose *Reminder Alarm*

A notification contains the information about the remaining time (30 min, 15 min or no time left) to transfer a blood glucose record to the device and the application icon (see Figure 5.4). According to the Google Design Patterns for Notifications, the notification icon in the system bar is especially white colored [11]. By tapping the notification, the user can directly enter the main menu of the application or dismiss it by swiping.



Blood glucose reminder alarms

08:00 - 10:00

12:00 - 14:00

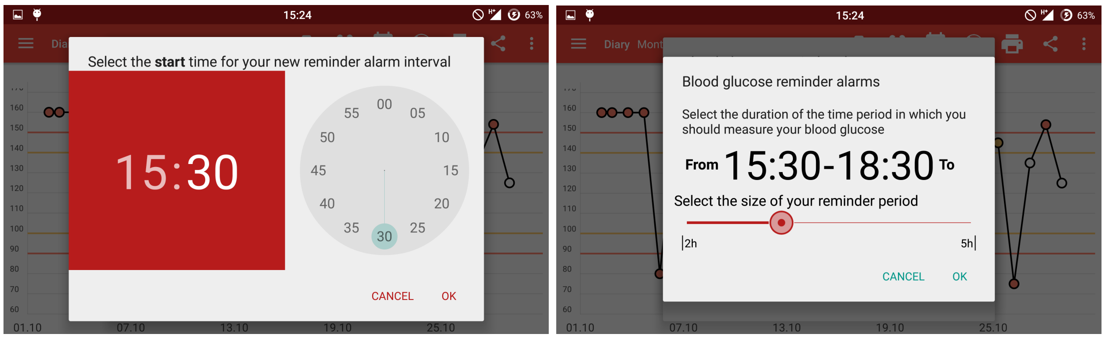
18:00 - 20:00

CLOSE

Figure 5.5: BloodGlucose *Reminder Alarms* Menu

The user can manage all *Reminder Alarms* from the *Reminder Alarm* menu which can be accessed by tapping the alarm clock button in the Navigation Bar. The *Reminder*

Alarm menu is displayed in an Android *Dialog* [6] on top of the main menu (see Figure 5.5). A list of all existing *Reminder Alarms* is displayed and each *Reminder Alarm* can be deleted by tapping the discard button and dis-/enabled it by tapping the checkbox. Instead of a checkbox, a toggle button could be chosen to dis/-enable each *Reminder Alarm* but the fact that the Android ToggleButtons suffer from incompatibility errors with earlier Android versions and also differ in the layout between these versions, the checkbox was regarded as the best option to grant the highest usability and reliability.
If the user wants to create a new *Reminder Alarm*, he/she has to tap the Floating action button at the bottom right of the Dialog. Design and positioning of the Floating action button corresponds to the Google design guidelines [4].


Select the **start** time for your new reminder alarm interval

15:30

Blood glucose reminder alarms

Select the duration of the time period in which you should measure your blood glucose

From 15:30-18:30 To

Select the size of your reminder period

(a) Select start time (b) Select duration

Figure 5.6: BloodGlucose create new *Reminder Alarm*

After tapping the Floating action button, a new Dialog appears at which the user has to select the start time of the new *Reminder Alarm* (see Figure 5.6 (a)). Therefore the default Android *TimePickerDialog* [16] was chosen and modified with the AppTheme which affects the colors of the *TimePickerDialog* to grant the best possible familiarity and usability for the time selection to the user. After selecting the time and confirming it, a new Dialog, to select the duration of the *Reminder Alarm* period, is displayed (see Figure 5.6 (b)). In the majority of cases the developers would display two TimePickerDialogs consecutively. One for the start and one for the end time of the *Reminder Alarm*. Due to the fact, that most users of the application will be elderly people and are more unskillful in mobile interaction then younger ones, an error preventive interface was developed, at which the user can only select the duration by a slider (*5. Prevent errors*). Additionally, the user can always see the current system change, concerning the selected duration and it's effect on the end time (*3. Offer informative feedback*). It's also possible to always reverses the procedure (*6. Permit easy reversal of actions*). These design aspects accord with the golden rules for designing user interfaces of Ben Shneiderman [34]. If the user confirms his/her input, all existing *Reminder Alarms* are stopped, the new *Reminder Alarm* is inserted into the local database and all *Reminder Alarms* (including the new one) are rescheduled.

5.3 Achievements

The achievements are stored in a, custom developed JSON structure (see Figure 5.7) to grant higher modularity, expandability and the possibility to introduce this developed concept into other applications. Thereby all achievements are stored in a JSONArray. Each achievement has a 'kind' and specified 'values'. The 'values' represent the required amount of existing records depending on the requirements of the categories. The amount of categories for each achievement is optional. This design implements the same amount for both kinds of achievements.
All categories can be identified by their name, which is an alphabetical order (A,B,C..). For storage efficiency the JSONArray 'counter_ dates' holds the dates, when an achievement is acquired, and thereby offers as well the position of the last acquired achievement. When the application checks which achievement is the next reachable one, it takes the length of the 'counter_ dates' JSONArray of each category and utilizes the value of the 'values' JSONArray at that certain position. Thereby it's possible to check which achievements were acquired, when they were acquired and which the next locked one is.

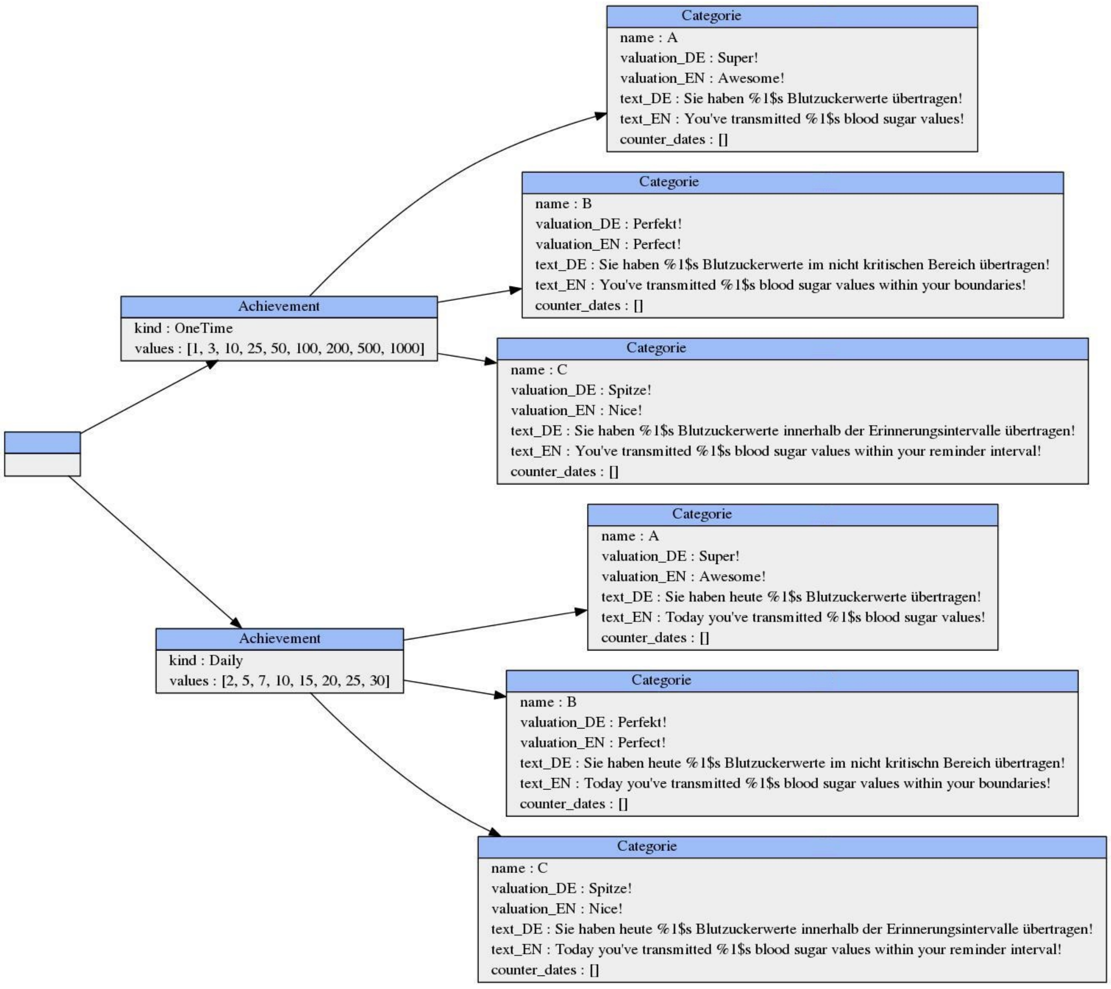

Categorie
name : A
valuation_DE : Super!
valuation_EN : Awesome!
text_DE : Sie haben %1$s Blutzuckerwerte übertragen!
text_EN : You've transmitted %1$s blood sugar values!
counter_dates : []
Categorie
name : B
valuation_DE : Perfekt!
valuation_EN : Perfect!
text_DE : Sie haben %1$s Blutzuckerwerte im nicht kritischen Bereich übertragen!
text_EN : You've transmitted %1$s blood sugar values within your boundaries!
counter_dates : []
Achievement
kind : OneTime
values : [1, 3, 10, 25, 50, 100, 200, 500, 1000]
Categorie
name : C
valuation_DE : Spitze!
valuation_EN : Nice!
text_DE : Sie haben %1$s Blutzuckerwerte innerhalb der Erinnerungsintervalle übertragen!
text_EN : You've transmitted %1$s blood sugar values within your reminder interval!
counter_dates : []
Categorie
name : A
valuation_DE : Super!
valuation_EN : Awesome!
text_DE : Sie haben heute %1$s Blutzuckerwerte übertragen!
text_EN : Today you've transmitted %1$s blood sugar values!
counter_dates : []
Achievement
kind : Daily
values : [2, 5, 7, 10, 15, 20, 25, 30]
Categorie
name : B
valuation_DE : Perfekt!
valuation_EN : Perfect!
text_DE : Sie haben heute %1$s Blutzuckerwerte im nicht kritischn Bereich übertragen!
text_EN : Today you've transmitted %1$s blood sugar values within your boundaries!
counter_dates : []
Categorie
name : C
valuation_DE : Spitze!
valuation_EN : Nice!
text_DE : Sie haben heute %1$s Blutzuckerwerte innerhalb der Erinnerungsintervalle übertragen!
text_EN : Today you've transmitted %1$s blood sugar values within your reminder interval!
counter_dates : []

Figure 5.7: Achievements UML schema diagram

Inside of the application the requirements of each achievement and their categories needs to be checked with the database entries. This is done every single time when a record is inserted. Since the requirements can't include a modular SQL statement for every existing SQL database it's necessary to implement the SQL statement in the application fitting to the particular database. In case of expanding the achievements and/or categories the statements to check for the new achievements have to be implemented. The JSON file is originally stored in the assets folder of the Android application and has to be copy to the local directory, otherwise the JSON file can't be modified. Thereby an unused version remains in the assets folder. Unfortunately it's not possible to solve this problem because of Android's restriction to modify data from the assets folder.

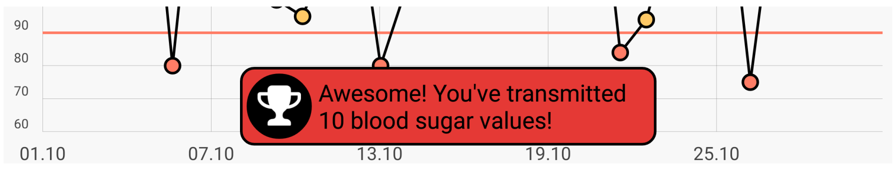
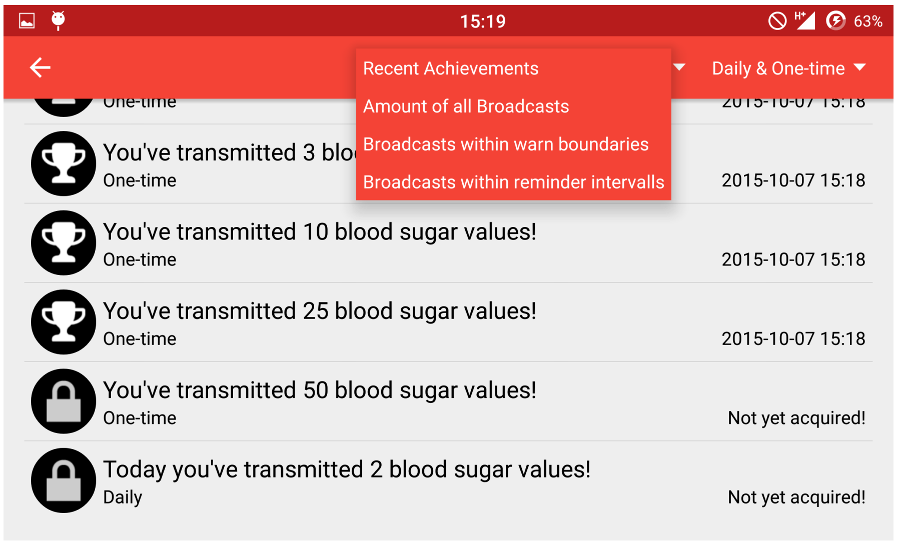

Awesome! You've transmitted 10 blood sugar values!

Figure 5.8: BloodGlucose achievement Toast

When an achievement is acquired, the user is notified by a custom designed Android Toast (see Figure 5.8). The Toast includes the message of the achievement and a trophy icon. For consistency the trophy icon represents the achievement in every illustration of the application according to Shneiderman's design guide line *1. Strive for consistency* [34]. After 3.5 seconds (Android's default duration for long delay) the Toast disappears. This duration was chosen to offer the user enough time to recognize and read the message.

The user can access the achievements menu from the trophy icon in the Navigation Bar of the main menu. The user will find an overview of the recently acquired achievements presented in the achievements menu. Additionally, he/she has the possibility to filter the displayed achievements by two filter options, each in a drop-down menu. These are arranged at the top right of the Navigation Bar (see Figure 5.9). In the first drop down menu the user can decide between the Categories of the achievements and the recently acquired achievements. The second one filters the *Kind* of the achievements (daily and/or one-time). Depending on the selected filter the acquired achievements are


15:19
63%
Recent Achievements
Amount of all Broadcasts
Broadcasts within warn boundaries
Broadcasts within reminder intervalls
Daily & One-time

One-time 2015-10-07 15:18

You've transmitted 3 blo
One-time 2015-10-07 15:18

You've transmitted 10 blood sugar values!
One-time 2015-10-07 15:18

You've transmitted 25 blood sugar values!
One-time 2015-10-07 15:18

You've transmitted 50 blood sugar values!
One-time Not yet acquired!

Today you've transmitted 2 blood sugar values!
Daily Not yet acquired!

Figure 5.9: BloodGlucose Achievements Menu

displayed as well as the next reachable ones. Acquired achievements have a trophy and the not yet reached ones have a lock icon (see Figure 5.9). The selected icons accord with the Google design guidelines for Icons [7]. The graphical user interface design for achievements is designed to offer the user the highest possible usability by simple and common used elements.

6 Requirements Comparison

In this chapter the defined requirements from chapter 3 are compared to the developed concepts and implementation. The implementation of each requirement is rated between **++** (requirement could be perfectly implemented) and **- -** (requirement couldn't be implemented).

6.1 Functional Requirements

6.1.1 Reminder Alarms

i. Create *Reminder Alarms*: **++**
   The user can create an arbitrary amount of *Reminder Alarms*.

ii. Save *Reminder Alarms*: **++**
   All *Reminder Alarms* can be saved in a local SQL-Lite database on the device.

iii. Delete *Reminder Alarms*: **++**
   It's no problem to delete *Reminder Alarms*.

iv. Turn *Reminder Alarms* on and off: **++**
   When a *Reminder Alarms* is turned off, the background service is canceled, when turned on the service is restarted.

v. Overview of all *Reminder Alarms*: **++**
   The user can see all existing *Reminder Alarms* and if they're active or inactive.

vi. Reminder alarm notification: **+**
   At the end of each *Reminder Alarm* a notification is displayed. Unfortunately if

the phone was shut down during a *Reminder Alarm*, afterwards no notification is displayed.

6.1.2 Emergency Contact

i. Add *Emergency Contact* data: **++**
The *Emergency Contact* data can be added from the settings.

ii. Notify *Emergency Contact* by SMS: **++**
It's possible to notify the *Emergency Contact* by SMS. This happens in the background of the application.

iii. Notify emergency contact by email: **- -**
It's not possible to notify the *Emergency Contact* by Email because the only possibility to do that would be by opening an Email application and the user still would has to send the Email by himself. Since this opportunity can't take place in the background this feature isn't implemented.

iv. *Emergency Contact* notification in case of a critical blood glucose record: **++**
If this feature is enabled, the *Emergency Contact* is notified by SMS.

v. Emergency contact notification in case, no record was transferred within a reminder alarm period: **++**
If this feature is enabled, the *Emergency Contact* is notified by SMS.

6.1.3 Achievements

i. Check for *Achievements*: **++**
Everytime when a blood glucose record is transferred to the device, the application checks if an *Achievement* is reached.

ii. Save reached *Achievements*: **++**
When an *Achievement* is acquired, the timestamp is saved in the JSON to the certain *Achievement*.

iii. Load *Achievements*: **++**
All acquired *Achievements* and the next reachable ones can be loaded from the JSON file.

iv. *Achievement* board: **++**
An *Achievement* board was implemented, it's accessible from the trophy icon in the Navigation Bar of the main menu.

v. Sort *Achievement* board: **++**
The *Achievement* board can be sorted by the kind (daily and/or one-time) and categories.

vi. Display next reachable *Achievements*: **+**
The next reachable *Achievement* is displayed, but it's not possible to see all unreached *Achievements*.

6.2 Non-Functional Requirements

Non-functional requirements cover characteristics and qualities which the application should have. The following table presents an overview of the non-functional requirements.

i. Technical Requirements: **++**
The application runs on Android 4.0 and higher without any known issues.

ii. Usability: **+**
The application is in nearly every case very easy to use, but for inexperienced users it's probably hard to find the *Emergency Contact* settings, due to the fact that most inexperienced users won't find the Android default settings menu by themselves.

iii. Maintainability & Expandability: **+**
The code is well documented but in case of expanding the *Achievement* feature, it's necessary to burrow into the existing SQL database to implement the SQL statements for the new *Achievements*.

iv. Correctness & Reliability: **+**
In some cases, with unknown cause, it happens that the application shuts down from the background. This happens from time to time after a restart of the mobile device.

v. Look & Feel: **+**
The user interface design is based on Googles material design guidelines. But in some Dialogs it wasn’t possible to apply the AppTheme. This results in the default button color in these Dialogs.

vi. User Friendliness: **++**
The application prevents any known possible input errors.

7 Conclusion

This chapter provides a final overview of the developed and implemented concepts. Additionally, possible extensions which can be implemented in the future to increase user motivation for mobile health applications are presented.
The implemented concepts are build on a psychological proven method, the operant conditioning. All concepts are focused on improving the health status of diabetics. Thereby the main goal was to increase user's motivation and binding to a health care application. This should be achieved by applying the concept of *Reminder Alarms*, *Monitoring* and *Achievements*. The concept of *Achievements* is already a widespread gamification method in fitness apps which is intended to increase user motivation. The concept of *Reminder Alarms* helps the user to not forget to measure his/her blood glucose level in regular intervals what increases the amount of information about the patient for the doctor which results in a better foundation of the treatment design. The *Monitoring* concept offers the possibility for family, friends and/or caregivers to take part at the life and treatment of the patient even if they aren't present. This could be an important factor in demographic change and rising costs of medical treatment. But the most important reason should be to improve thereby the medical condition of the patient and prevent long-term complications.

7.1 Future Extensions

The presented concepts will affect the user, in order to improve end extend these concepts, some ideas which where discovered during the implementation are here illustrated.

7.1.1 Rewards

The concept of rewarding someone for his behavior belongs to the positive reinforcement of operant conditioning [35]. To introduce real world rewards to the application, the user motivation and behavior could be improved even more and faster. Therefore, it could be possible to offer rewards e.g. coupons from different companies which would finance itself by the advertisement of the companies. Another possibility could be to offer the relatives or caregivers to create own rewards for acquiring certain Achievements.

7.1.2 Competition & Sharing

A known motivational concept, especially for young people [32], is competition. *Achievements* could be easily extended to be shared with a friend or relative. Additionally, a leader board could be design. The combination of sharing personal data combined with other gamifications leads, according to Lenihans [28] assumption, to a positive effect.

7.1.3 Server Integration

With a server integration the cooperation between doctors, health insurances and the patient could be improved by storing the relevant user data on the server. Thereby a doctor could easily check the information about his patient without visiting him directly or prepare himself for the next ward round. This could lower treatment costs. Additionally, the patient could use multiple devices, like a smart phone, tablet and/or computer to access his blood glucose records and wouldn't suffer anymore of the risk to lose his/her blood glucose records.

7.2 Closing Statement

The integration of the developed concepts into an Android application can be seen as a basic example to apply motivational improvement methods in concrete medical context of diabetes. Due to the fact that I'm affected myself with diabetes for the last six years, I'm very motivated to improve systems which can support and motivate diabetics with the accomplishment of their treatment. Hopefully this work is an inspiration to other developers to improve medical software for diabetes diseased people in order to improve their medical condition and wellnessS.

Bibliography

[1] AlarmManager - Android Developers. `http://developer.android.com/reference/android/app/AlarmManager.html`. Accessed: 2015-10-14.

[2] Atari Announces Atari Fit^{TM}, A Gamification Fitness App to Motivate Players with 150 Exercise Routines that Unlock Classic Atari Games. `http://www.prnewswire.com/news-releases/atari-announces-atari-fit-a-gamification-fitness-app-to-motivate-players-with-150-exercise-routines-that-unlock-classic-atari-games-300011137.html`. Accessed: 2015-10-06.

[3] BroadcastReceiver - Android Developers. `http://developer.android.com/reference/android/content/BroadcastReceiver.html`. Accessed: 2015-10-14.

[4] Buttons: Floating Action Button - Components - Google design guidelines. `https://www.google.de/design/spec/components/buttons-floating-action-button.html{#}buttons-floating-action-button-floating-action-button`. Accessed: 2015-10-09.

[5] Challenges | Fitbit Blog. `https://blog.fitbit.com/category/challenges/`. Accessed: 2015-10-06.

[6] Dialog - Android Developers. `http://developer.android.com/reference/android/app/Dialog.html`. Accessed: 2015-10-14.

[7] Icons - Style - Google design guidelines. `https://www.google.de/design/spec/style/icons.html{#}icons-product-icons`. Accessed: 2015-10-10.

[8] Map your routes. `http://www.mapmyfitness.com/workoutgames/{#}/`. Accessed: 2015-10-06.

[9] Nike+ Running GPS App for iPhone & Android. Nike.com. `http://www.nike.com/us/en{_}us/c/running/nikeplus/gps-app`. Accessed: 2015-10-06.

[10] Notifications - Android Developers. `http://developer.android.com/guide/topics/ui/notifiers/notifications.html`. Accessed: 2015-10-14.

[11] Notifications - Patterns - Google design guidelines. `https://www.google.com/design/spec/patterns/notifications.html{#}notifications-content`. Accessed: 2015-10-08.

[12] PendingIntent - Android Developers. `http://developer.android.com/reference/android/app/PendingIntent.html`. Accessed: 2015-10-14.

[13] Service - Android Developers. `http://developer.android.com/reference/android/app/Service.html`. Accessed: 2015-10-14.

[14] Settings - Android Developers. `http://developer.android.com/guide/topics/ui/settings.html`. Accessed: 2015-10-14.

[15] SlimKicker Calorie Counter and Fitness Tracker: Level Up Your Body. `http://www.slimkicker.com/`. Accessed: 2015-10-06.

[16] TimePickerDialog - Android Developers. `http://developer.android.com/reference/android/app/TimePickerDialog.html`. Accessed: 2015-10-14.

[17] Barton, Dr. Erin E. *Encyclopedia of Autism Spectrum Disorders*. Springer New York, New York, NY, 2013.

[18] Cassel, Dieter. Demographischer Wandel - Folgen für die gesetzliche Krankenversicherung. *Wirtschaftsdienst 81*, 2 (2001), 87–91.

[19] Deterding, Sebastian. Gamification: designing for motivation. *Interactions 19* (2012), 14–17.

[20] Gabe Zichermann, Christopher Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, Inc., 2011.

[21] Hamari, J. and Eranti, V. Framework for designing and evaluating game achievements. *Proc. DiGRA 2011: Think Design Play* (2011).

[22] Hamari, J. and Koivisto, J. Social Motivations To Use Gamification: An Empirical Study Of Gamifying Exercise. *ECIS* (2013).

[23] Hamari, Juho and Koivisto, Jonna and Sarsa, Harri. Does gamification work? - A literature review of empirical studies on gamification. *Proceedings of the Annual Hawaii International Conference on System Sciences*, JANUARY (2014), 3025–3034.

[24] Hancock, P. Hedonomics: The Power of Positive and. *Ergonomics in Design* (2005).

[25] Huotari, Kai and Hamari, Juho. Defining gamification - A Service Marketing Perspective. In *Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12*, ACM Press (New York, New York, USA, oct 2012), 17.

[26] International Diabetes Federation. *IDF Diabetes Atlas*, 6th ed. International Diabetes Federation, Brussels, Belgium: International Diabetes Federation, 2013.

[27] Kapp, Karl M. *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*, 1 ed. Pfeiffer & Company, may 2012.

[28] Lenihan, D. Health Games: A Key Component for the Evolution of Wellness Programs. *Games for Health Journal* (2012), 233–235.

[29] McLeod, S. A. B.F. Skinner - Operant Conditioning. `www.simplypsychology.org/operant-conditioning.html`, 2015. Accessed: 2015-10-13.

[30] Miller, Aaron S. and Cafazzo, Joseph A. and Seto, Emily. A game plan: Gamification design principles in mHealth applications for chronic disease management. *Health informatics journal* (jul 2014).

[31] Palmer, Doug and Lunceford, Steve and Patton, Aaron J. The Engagement Economy: How Gamification is Reshaping Businesses. *Deloite Review 11* (2012).

[32] Pereira, Pedro and Duarte, Emília and Rebelo, Francisco and Noriega, Paulo. A Review of Gamification for Health-Related Contexts. In *Proceedings of the Third International Conference on Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments*, vol. 8518 of *Lecture Notes in Computer Science*, Springer International Publishing (Cham, jun 2014), 742–753.

[33] Richards, Chad and Thompson, Craig W. and Graham, T. C. Nicholas. Beyond Designing for Motivation : The Importance of Context in Gamification. *CHI Play* (2014), 217–226.

[34] Shneiderman, Ben. *Designing the user interface: strategies for effective human-computer interaction*. University of Maryland, 1992.

[35] Thorndike, Edward L. and Bobertag, Otto. *Psychologie der Erziehung*. Darmstadt, 1970.

[36] Wittkugel, J. and Backhaus, N. Nutzerzentrierte Gestaltung einer Applikation im Diabetes-Kontext. *Mensch und Computer 2015– Workshopband* (2015).

[37] World Health Organization. About diabetes. `http://www.who.int/diabetes/action{_}online/basics/en/`, 2015. Accessed: 2015-09-25.

[38] World Health Organization. Diabetes. `http://www.who.int/mediacentre/factsheets/fs312/en/`, 2015. Accessed: 2015-09-25.

[39] Zhang, Ping. Technical opinion: Motivational affordances: reasons for ICT design and use. *Communications of the ACM 51*, 11 (nov 2008), 145.

List of Figures

Name: Michael Götze Matriculation number: 791370

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.

Ulm, ...

Michael Götze