Expertise
and insight
for the future

Topi Penttilä

Diabetes self-management application for Apple Watch

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

1 November 2018

Author Title Number of Pages Date	Topi Penttilä Diabetes self-management application for Apple Watch 40 pages + 1 appendix 1 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Mobile Solutions
Instructors	Petri Vesikivi, Principal Lecturer

The objective of this thesis was to make a proposal of an optimal Apple Watch application for diabetes self-management and create a prototype of that. The system is indented to help diabetics in their everyday life and in the bigger picture, decrease the enormous cost of the care for individual and society. The research phase consisted of an online survey and an analysis on available solutions. The focus group of the survey was diabetics and health care professionals.

Design Thinking based approach was used as the design method. Once the first version of the application was made, face-to-face discussions with potential users were conducted. The final outcome of the thesis is a small proof-of-concept of an Apple Watch application but more importantly a proposal for a fully working system that could be implemented utilizing the ideas and techniques proposed by this paper.

There are many smartphone and smartwatch applications for diabetes self-management available. However, the majority of them do not match the criteria of diabetics and healthcare professionals and therefore are not used. The provided proposal is intended to match this criteria. Whether this really is the case, should be defined through a holistic testing and evaluation which was not possible to complete as part of this thesis.

Keywords	Diabetes, Apple Watch, healthtech

Tekijä Otsikko Sivumäärä Aika	Topi Penttilä Apple Watch -älykellosovellus diabeteksen omahoitoon 40 sivua + 1 liite 1.11.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Mobile Solutions
Ohjaaja	Lehtori Petri Vesikivi

Insinöörityön tarkoituksena oli tehdä ehdotus optimaalisesta Apple Watch -älykellosovelluksesta diabeteksen omahoidon tueksi ja toteuttaa sen demoversio. Järjestelmän tarkoituksena on helpottaa diabetesta sairastavia heidän jokapäiväisessä elämässään ja vähentää siitä yksilölle ja yhteiskunnalle koituvia kustannuksia. Taustatutkimusosa sisälsi internetkyselyn ja olemassa olevien ratkaisujen analysoinnin. Kyselyn kohderyhmää olivat sekä diabetesta sairastavat että terveydenhuollon ammattilaiset.

Kun ensimmäinen demoversio sovelluksesta oli valmis, käyttäjätutkimus jatkui henkilökohtaisilla keskusteluilla diabetesta sairastavien kanssa. Lopputuloksena syntyi pieni sovellusdemo Apple Watchille, mutta tärkeimpänä ehdotus kokonaisesta järjestelmästä, jonka voisi toteuttaa hyödyntäen tässä insinöörityössä esitettyjä ideoita ja teknologioita.

Markkinoilla on monia älypuhelin- ja älykellosovelluksia diabeteksen omahoitoon. Monet niistä eivät kuitenkaan ominaisuuksiltaan vastaa diabetesta sairastavien ja terveydenhuollon ammattilaisten tarpeisiin. Tässä opinnäytetyössä ehdotetun ratkaisun on tarkoitus vastata vähintään osaan näistä tarpeista. Toteutuuko tämä käytännössä, selviäisi kuitenkin vasta kokonaisvaltaisen testauksen kautta, jota ei tämän opinnäytetyön osana ollut mahdollista toteuttaa.

Avainsanat	sokeritauti, diabetes, Apple Watch, terveysteknologia

Contents

List of Abbreviations

API	Application Programming Interface
BLE	Bluetooth Low Energy
CBR	Case-Based Reasoning
CGM	Continuous Glucose Monitor
HTTP	Hypertext Transfer Protocol
HYPER	Hyperglycemia - High blood glucose level
HYPO	Hypoglycemia - Low blood glucose
IGT	Impaired Glucose Tolerance
IFG	impaired Fasting Glycaemia
JSON	JavaScript Object Notation
LTE	Long-Term Evolution – 4G mobile communications standard
NFC	Near-Field Communication
POC	Proof-Of-Concept
UI	User Interface
UX	User Experience
WBAN	Wireless Body Area Network
WLAN	Wireless Local Area Network

1 Introduction

Diabetes mellitus is a worldwide, increasing problem and the most common chronic illness there is [1]. With increasing number of diabetes patients and the lack of knowledge and health-care resources, many people suffer from the illness, partly in vain. With better knowledge and support, individuals with diabetes could live a better life.

Mobile Health (mHealth) interventions are a potential tool to manage the disease. Many times, diabetes mHealth solutions are seen as the most potential adaption in mHealth sector [2]. There is a lot of potential in using consumer electronics, such as smartphones, smartwatches and other wearables in diabetes self-management. The devices can provide fast and automated ways to gather user's health-related data, making it easily accessible. They can utilize Artificial and Clinical Intelligence as well as Machine Learning to analyse the data to educate, remind, alert and motivate the user to cope with the disease. Smart devices have the potential to really benefit daily lives of people with chronic illnesses. While being potential, currently the applications are not widely used as they do not fully match the criteria of diabetics and healthcare professionals.

As the research part of this thesis, the objective is to do a background research on the daily life of diabetics and the struggles they have in their everyday life. This research includes an online survey conducted together with Teemu Rytsölä, who is working on an Android Wear Diabetes self-management application. As part of the research, also some existing diabetes self-management applications on the market are compared, focusing on the ones that are available for Apple Watch. The outcome of the thesis is a proof-of-concept proposal of an optimal Apple Watch application for diabetic's daily use. The purpose of the application would be to make diabetics' life easier and decrease the enormous cost of Diabetes care worldwide.

2 Background

2.1 Diabetes as a worldwide disease

According to World Health Organization (WHO), diabetes or more precisely, diabetes mellitus, is a chronic disease where human body does not produce enough insulin or is unable to use the insulin effectively. Insulin is a natural human hormone which regulates the blood glucose level. Diabetes is a worldwide, increasing problem and the most common chronic illness in the world. Number of diabetes patients has raised from 108 million to 422 million between 1980 and 2014. [1]

There are different types of diabetes; type 1 diabetes, type 2 diabetes, gestational diabetes and different forms of prediabetes, to name a few. The cause of type 1 diabetes is not known and the disease cannot be prevented with lifestyle choices. It is many times childhood-onset. Diabetes may cause thirstiness, having to pee often, hunger, weight loss, fatigue and changes in eyesight. In type 1 diabetes, the body doesn't produce enough insulin and daily insulin intake is required. [1]

Type 2 diabetes is the most common form of diabetes worldwide. Type 2 diabetes is more commonly diagnosed on adults. To some extent, it can be prevented with living a healthy lifestyle while overweight and physical inactivity may result to type 2 diabetes. In type 2 diabetes, body is not able to use the insulin effectively enough. Symptoms are similar to type 1 diabetes. Type 2 diabetes can be hiding as a prediabetes several years after onset. Prediabetes, such as Impaired glucose tolerance (IGT) and impaired fasting glycaemia (IFG) many times precede type 2 diabetes. [1]

Diabetes is the major cause of many conditions, such as blindness, kidney failure, limb amputation, stroke and heart attacks. In 2015, around 1,6 million deaths were caused directly by diabetes. WHO projects that in 2030 Diabetes will be the 7th most common leading cause of death. [1]

As seen on image 1, the number of diabetes patients is increasing worldwide. In countries with low and mid income, the mortality rate is huge as the medications and treatments are too expensive for people. However, as smart devices are becoming cheaper and cheaper, they can help struggling people to gain more knowledge on how to prevent and treat the disease.

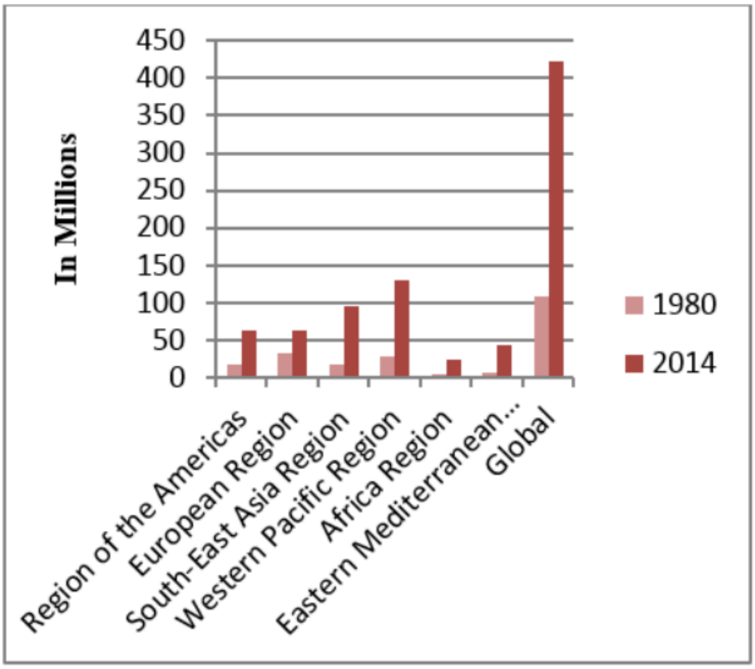


Image 1. Number of diabetics in millions by region. [3]

Living life with diabetes requires discipline. It affects the patient’s everyday life in many ways. Balancing with carbohydrates, insulin and lifestyle choices to keep the blood glucose levels in control is a time-consuming task and an art of its own. It requires a lot of knowledge as well as motivation.

The American Diabetes Association recommends Lifestyle Modification in order to prevent and treat type 2 diabetes. Basic principle of this Lifestyle Modification is that one’s energy intake is the same as energy output, or the carbohydrates ingested the same as the ones consumed by being physically active. It promotes healthy lifestyle and is the best way to prevent hyperglycaemia (high blood glucose level) and hypoglycaemia (low blood glucose level). [4]

The number of type 2 diabetes patients is constantly increasing. Alexander B et al. [2017: 283] proposed a behavioural sensing system utilizing smartwatch and sensors. Many times, recently diagnosed type 2 diabetics are lacking knowledge and motivation in treating the illness in a desirable manner. [4]

2.2 Usage of mHealth applications in diabetes self-management

Smart devices, such as smartphones and smartwatches can track and log health-related data. They can be used to track and log the most crucial data in diabetics' life; physical activity, blood glucose level and food intake. By monitoring the data, analysing it and learning from it, smart device can motivate and educate people to live a better life. [5]

There are hundreds of diabetes self-management mobile applications available but only a few of them have reported studies of the health outcomes. Haemoglobin A1c (HbA1c) is a mediocre-term average of blood glucose levels and a commonly recognized performance outcome of diabetes management. Some existing applications have been studied to significantly reduce the HbA1c. [6]

Based on the value that diabetes self-management applications have and the size of the market, they are constantly rated as to have the highest potential in mHealth sector. However, at the moment they are not being used that much as they do not meet the expectations the target group has. Diabetics might want to use these solutions as individuals but to provide a complete end-to-end solution, healthcare sector would need to be willing to take new technologies in use or integrate existing technologies with new ones. This is not cheap nor easy and thereby healthcare sector is slow at adopting these new technologies even though they could potentially decrease the costs in the long term. [2]

More research, evaluation and development is needed to provide the best possible applications and systems for diabetes care. This thesis will provide some potential solutions for some of the common problems.

3 Design and Research

This section will define the design thinking-based methodology utilized in the process and go through the research findings in designing a self-management application for Apple Watch. First it will glance the Apple Watch as a device and provide some basic design guidelines on how to get started with the design. Secondly, it will focus on some of the existing solutions for smart diabetes self-management, will dig into the field research that was carried out as part of the research and lastly will discuss about some considerations that need to be addressed when creating this type of system.

3.1 Design Thinking based approach

Design thinking is a solution-based approach in problem solving. It consists of 5 stages illustrated in image 2: [7]

- Empathize – Understand the users’ needs.
- Define – Define the core problems.
- Ideate – Create based on findings in the previous stages. Think outside the box.
- Prototype – Create the first scaled-down version of some of the requirements and carry out initial testing.
- Test – Develop a prototype and take that to users to get a better understanding of how the whole solution is experienced by them.

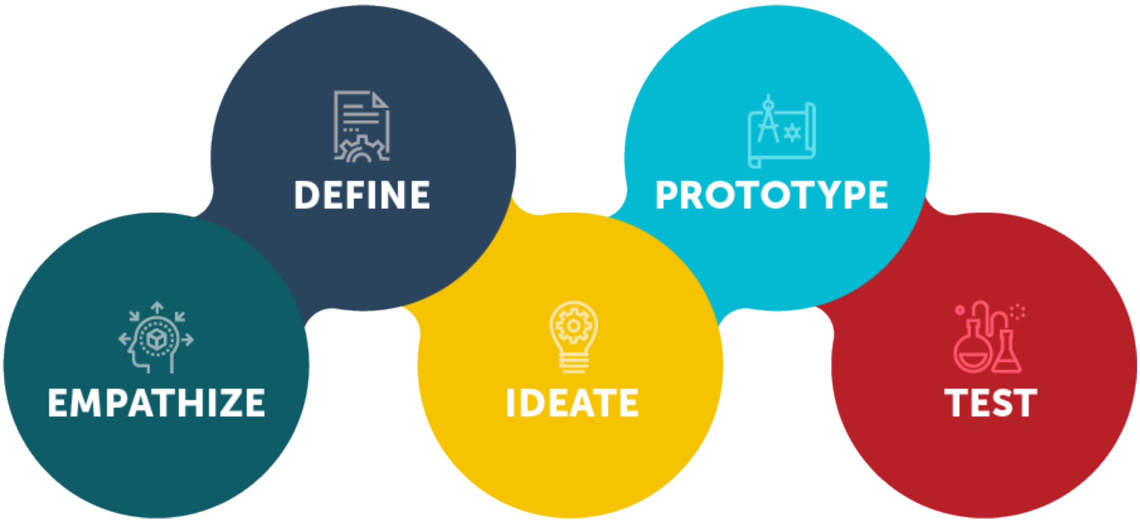


Image 2. Design thinking process. [8]

In this paper, the initial empathize stage included the evaluation on existing solutions and the online survey discussed in sections 3.3 and 3.4. After getting a fair understanding in the field, the core problem was defined and basic set of requirements were ideated. A prototype was made and brought for a small group of potential users. The next step would be to prototype the whole solution and go deeply into testing phase. The test phase is iterative and when re-empathizing, some problems would most likely be redefined [7].

3.2 Designing for Apple Watch

Apple watch is marketed to be a proactive health monitor, ultimate workout partner, comprehensive activity tracker as well as an easy way to connect. Sounds like a device that can do well in proactive diabetes self-care. [9]

Look and feel of Apple Watch

Apple Watch has a patented square design [10]. As seen on image 3, the watch has two physical controls both located on the right side of the device. On top there is a Digital Crown that rotates and presses in. Below the Crown, there is a large button.

Image 3. Apple Watch has two physical buttons. [11]

According to Apple’s Human Interface Guidelines, Apple Watch was designed based on three foundations: [12]

- Lightweight interactions - Focus is on the content that user cares about. Information is easily accessible and provides fast interactions.
- Holistic design - Becloud the boundaries between software and hardware.
- Personal communication - Watch is worn on a user’s wrist. The watch is highly personal and so should be the content.

When designing an Apple Watch app, the approach should be focused. Apple’s Human Interface Guidelines specify three focus themes for Apple Watch app design: [12]

- Glanceable - Relevant information is up-to-date and available to the user at any time on a glance
- Actionable - Like with glanceable approach, the information is relevant and up-to-date at any time. App has custom notifications that provide a possibility to make actions even without opening the app.
- Responsive - The information is up-to-date. Responsive app provides fast interactions and minimizes the time of launching new screens. App gives feedback to the user on what it is going to do and provides notifications about the progress.

Interactions with the watch happen in a matter of seconds so the design process requires hard focus on detail. According to the Planning a Great Apple Watch Experience speech given at the Apple Worldwide Developers Conference keynote in 2017, a great Apple Watch experience comes down to three qualities. [13]

First, the app is legible. The text on the screen should be readable with enough contrast. The screen on an Apple watch is small. However, the information on it should be readable in a glance. On Apple Watch design, usually legibility equals size. [13]

Secondly, the information should be concise. The watch should only show the information and allow the actions that are relevant and that matter to the user. This also relates strongly to the last point, timeliness. Shown information and allowed actions should be just right for the user at each given point of time. [13]

3.3 Available Diabetes self-management applications

On the Apple's App Store, there are several apps for Diabetes self-care available. This sections will go through some of the most popular ones and evaluate the design based on Apple's Human Interface Guidelines and personal experience on using these applications.

Sugar Sense

Sugar Sense has a nice-looking iOS application with a wide variety of features for health tracking, reminder support etc. There is also an Apple Watch application available. The Watch app is really limited showing only today's step count and the latest blood glucose measurement, as seen in image 4. Step count is calculated automatically by the app and blood glucose values can only be inputted from the iOS application. There are also no complications available for the watch faces.

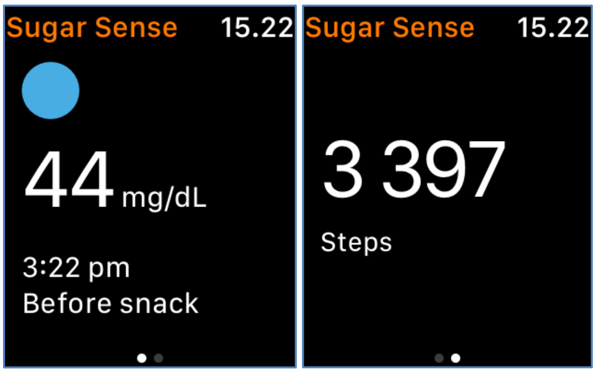


Image 4. Sugar Sense’s Apple watch application consist of only two views.

Legibility on the Sugar Sense’s watch app is on an adequate level. Everything shown in those two views are readable and they make sense to the user. Sugar Sense has taken the glanceable approach in design. The latest info is available on a glance in the app but can be argued whether this provides any value to the user. The information is timely but user cannot do any actions on the watch itself. Reminders are mirrored from phone to the watch but they do not provide any actions either.

Diabetes:M

As seen in image 5, Diabetes:M provides a fairly actionable and large watch application. User can input blood glucose, carbohydrates and insulin. The application also has an insulin calculator that tells user their active insulin levels based on previous data. This insulin level is also possible to see on watch face as a complication.

The page-based interface is not the most user-friendly. There is a vast number of objects on every view and inputting a value is not as fast as user would like the it to be on a watch.

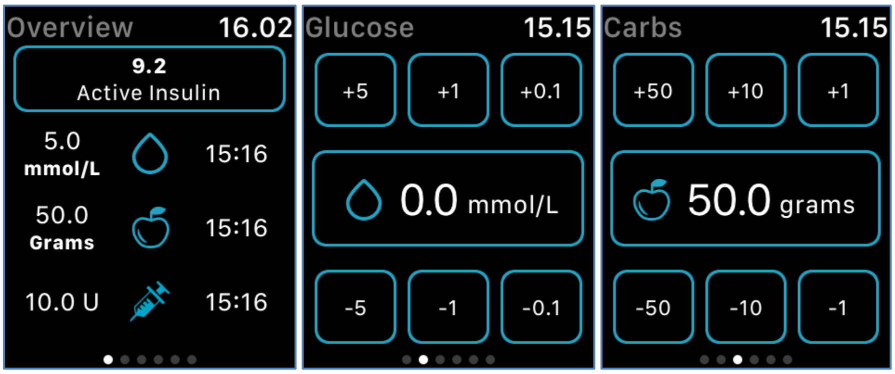


Image 5. Diabetes:M uses a confusing page-based interface.

Glanceable active insulin complication is a nice and pretty advanced feature. However, the overview in the app itself shows quite a bit of information and cannot really be called glanceable. The data input flow is confusing and takes too much time on a watch. The reminder notifications are not actionable. The information might be timely but can be argued whether it is concise.

The iOS application allows user to for example turn on pattern detection. It analyses the data and tries to find patterns to guide the user. Data can also be automatically imported from NightScout. NighScout is described in more details later in this section.

e-SMBG

e-SMBG's watch application uses page-based interface. As seen on image 6, first page shows a graph of blood glucose values for last 3 days, though the graph takes quite a while to load. On the second page user can make inputs, such as blood glucose and insulin. There is no carbohydrates input available on the watch application. User can also track events and see logbook of entries. The third page is interesting. It provides a more detailed chart on all the inputs for the past three days. The chart takes a long time to load and it is shown on a horizontal direction, which is not convenient for a watch. At least the last page's chart view is not concise nor legible.

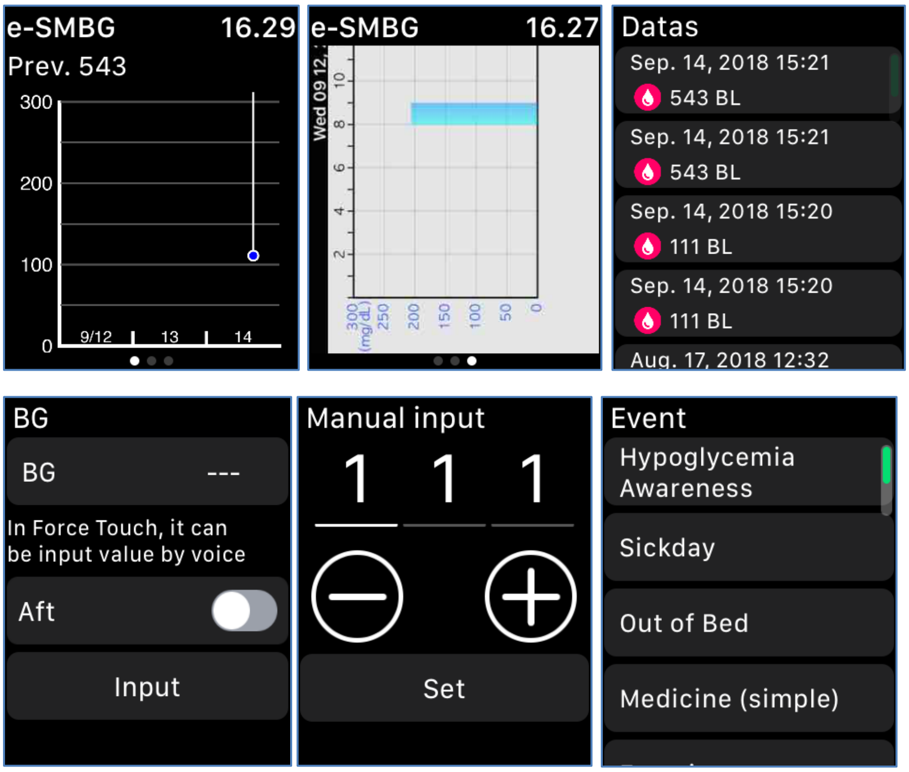


Image 6. e-SMBG provides many features on a smart watch app.

The input can be made manually and is somewhat user-friendly even though there is no indication on what data the user is inputting nor the unit in what the input is being made. Interestingly, the input can also be made spoken.

One Drop

One Drop's watch application is the nicest looking in all the apps tested. It is also fairly user-friendly. As seen on image 7, page-based interface consists of two views; input selection and a menu where user can select between different statistical views. User needs to dig quite deep before any relevant information is shown. This makes the app not that concise. The app is somewhat legible but list view on the second page contains some relatively small text. Also the only way to input new data is to use the whole-screen keyboard. This is not a native watchOS way to make inputs and might be hard at least for elderly people with big and shaky fingers. It is nice to have a lot of information on your wrist but are all these might not be relevant on a watch.

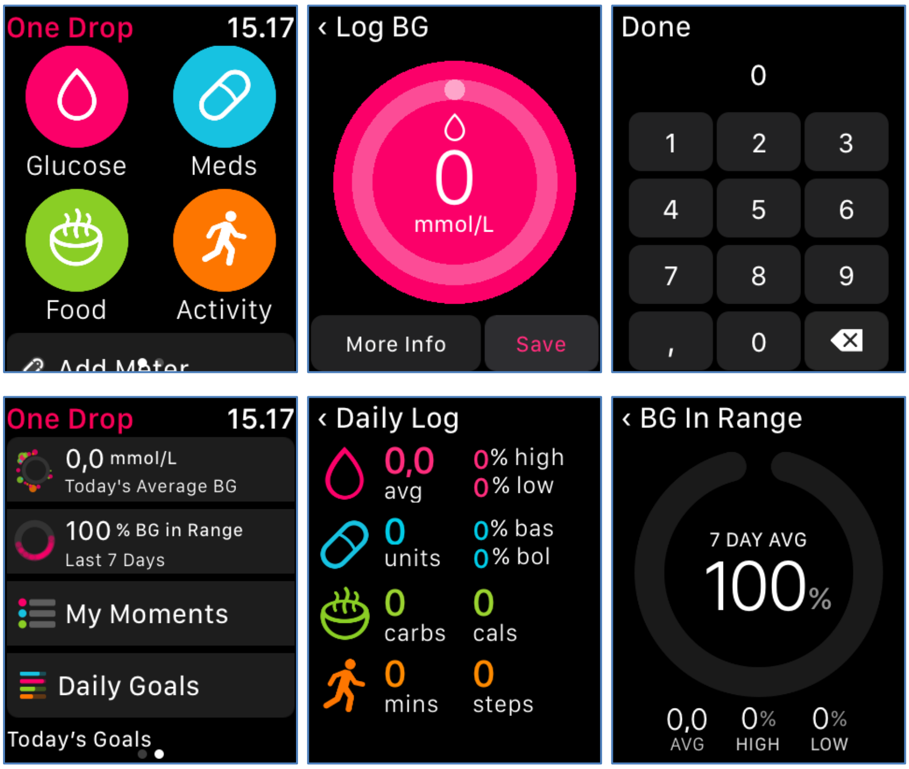


Image 7. One Drop Apple Watch app looks really good.

Application also comes with an average Blood Glucose complication. This increases the timeliness of the application. On the iOS app side, the application also supports Dexcom CGM sensor import from the API. The API is discussed in more detail in section 4.2.

Nightscout

Nightscout is not an Apple Watch app per say, but an interesting open source project. It is a system developed by type 1 diabetics. It allows diabetic with continuous glucose monitoring (CGM) sensor to get their measurement data in the cloud. User interfaces include smartphone and smartwatch applications along with a web app. As the data is in the cloud, it can also be monitored by for instance the relatives of the diabetic. Image 8 shows the different ways to show data on an Apple Watch. [14]

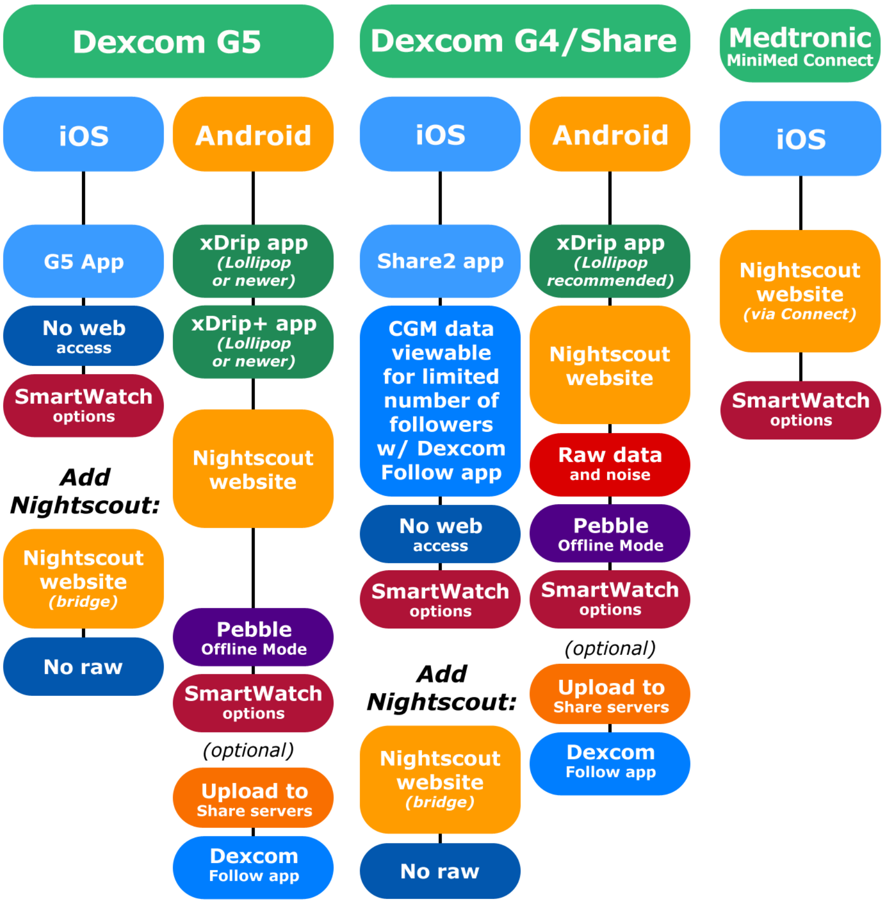


Image 8. Nighscout provides ways to get CGM measurements in the cloud. [14]

Setting up Nightscout is straight-forward and well documented but might feel tricky for a non-tech-savvy person. User needs to create a Github account and a copy the remote monitor repository, create a website and database, do some configuration and deploy the site. [15]

When personal Nightscout website has been set up, user can use for example an open source iOS and watchOS application called Nightguard to show the latest 30 values from the Nightscout server as seen on image 9. [15]

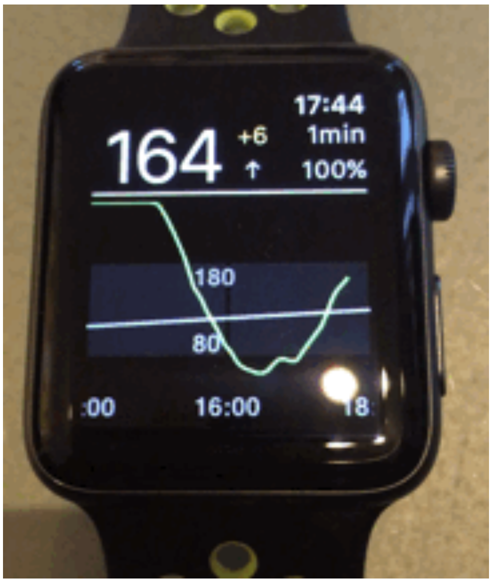


Image 9. User interface on Nightguard Apple watch application. [15]

Dexcom

One of the most popular CGM sensor manufacturers, Dexcom, has an iOS application that connects to the CGM sensor. The values can be seen on the app and also shared with others. There is also an Apple Watch extension for the application. The watch application shows a simple graph about the latest blood glucose values as seen in image 10. There is a complication that shows the latest blood glucose reading. The app is also able to send alert notifications about high and low blood glucose as well as about rapid increase or decrease. These alerts are configurable on the iOS app side. This makes the overall application's user experience quite timely and personal. [16]

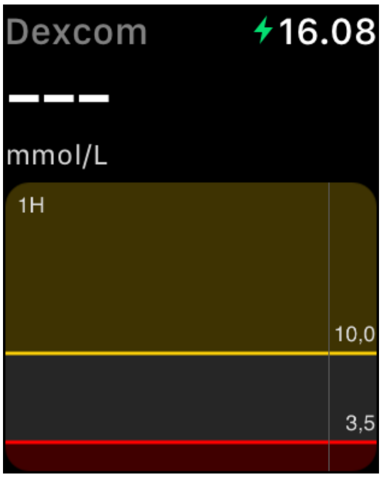


Image 10. Dexcom Apple Watch app user interface without data.

3.4 Field research

The purpose of the field research was to map out the hopes and needs of diabetics and to understand what it is like to live with diabetes. The research was partly conducted together with Teemu Rytsölä, who is working on an Android Wear application for Diabetes self-management. Field research was carried out with an online survey as well as discussions with type 1 diabetics at a diabetes conference in Helsinki, Finland. The online survey was carried out using Google Forms platform and the questions are listed in the appendix 1. The survey was shared on Facebook by diabetes clinic called Lääkärikeskus Neliapila and the Finnish Diabetes Association. Main focus groups were diabetics themselves and health-care professionals working with diabetes. Approximately 150 responses were received, from mainly type 1 diabetics. This section will go through the findings of the survey and the conference.

Age distribution of the respondents was quite even between the ages 15 and 65. Three quarters of the respondents were female. Also three quarters of the respondents were diabetics, the majorly suffered from type 1 and had had diabetes for more than 5 years.

The biggest challenges in living life with diabetes were queried as an open question. Still, the answers were quite consistent. The most challenging part is undoubtedly the glycaemic control, low blood glucose levels (especially at night time) and how stress, physical activity, insulin and consumed carbohydrates affect the blood glucose level. Finding balance requires discipline, planning and motivation. If you catch diabetes on your adult life, adapting to the new way of life might be tricky.

More than half of the respondents were familiar with at least some smart self-care applications for diabetes. Even bigger portion claimed to be ready to use smart applications if they realized that it really would have a positive impact on their everyday life. Based on this finding, there definitely seems to be a demand for this kind of solution.

The most desired features for the smartwatch application, regarding the survey, was to see real-time blood glucose reading. This would definitely increase the timeliness and personality of the application. For CGM sensor users, the app should alert on high (HYPER) and low (HYPO) blood glucose. Other valuable features would be the possibility to log carbohydrates and insulin. Physical activity should be tracked automatically. Also for diabetics who are taking insulin injections, a calculator that makes proposes on how much insulin should be injected and how much of it is left, would be valuable. All in all, the watch application should be accurate, reliable and easy to use. It could also remind user on taking blood glucose measurements and different medications. It should motivate user to be discipline. Machine learning and artificial intelligence could learn the daily routines of a diabetic and tailor the correct reminders as well as motivational notifications. The system could also contain clinical intelligence to guide user on how different things effect on each other.

The survey was only conducted in Finland which is generally a wealthy country with good access to health-care. This has an effect to the outcome of the survey and the needs would be rather different in a third-world country where people do not have access to the most advanced and modern ways to treat the disease. It is fair to believe, though, that some of the desired features would still remain. Smartwatches and smartphones are already relatively cheap and with the rapid evolution of technology, it is justifiable to believe that in a few years third-world countries have the access on the devices we have now.

3.5 Considerations

3.5.1 Security

As this system handles health-related and medical data, securing the information is essential as the data needs to be kept highly private and confidential. A system that utilizes Artificial Intelligence to suggest treatments for a patient has to ensure that the data provided is from the given patient, and that it is accurate. False data might cause the system to provide wrong assumptions which could potentially cause serious complications — or even death.

A typical architecture of an eHealthcare system consist of four layers which are illustrated in the image 11. Wearables, along with sensors and other smart devices such as smartphone, form together a Wireless Body Area Network, WBAN which is the layer 1 in the picture. The information between these nodes are transferred using short-area networks and then finally stored in cloud in a medical database. [17] Although the use of this technology has grown rapidly for the past few years, only a little research is done on the privacy of this kind of networks [18].

There can be multiple sensors and devices connected to the user interaction interface (layer 2). The User Interaction Interface can be a smartphone, computer or a smartwatch acting as a gateway. These devices collect the data from the first layer using for example BLE (Bluetooth Low Energy) or NFC (Near Field Communication) technologies sending it to the remote server using cellular connection LTE (Long Term Evaluation) or WLAN (Wireless LAN) for analysing [18]. Once analysed, server reports back to the User Interaction Layer providing the results to the user. The data on the server is usually also available for a team of health care professionals (layer 4). Each one of these layers is vulnerable for security attacks. [17]

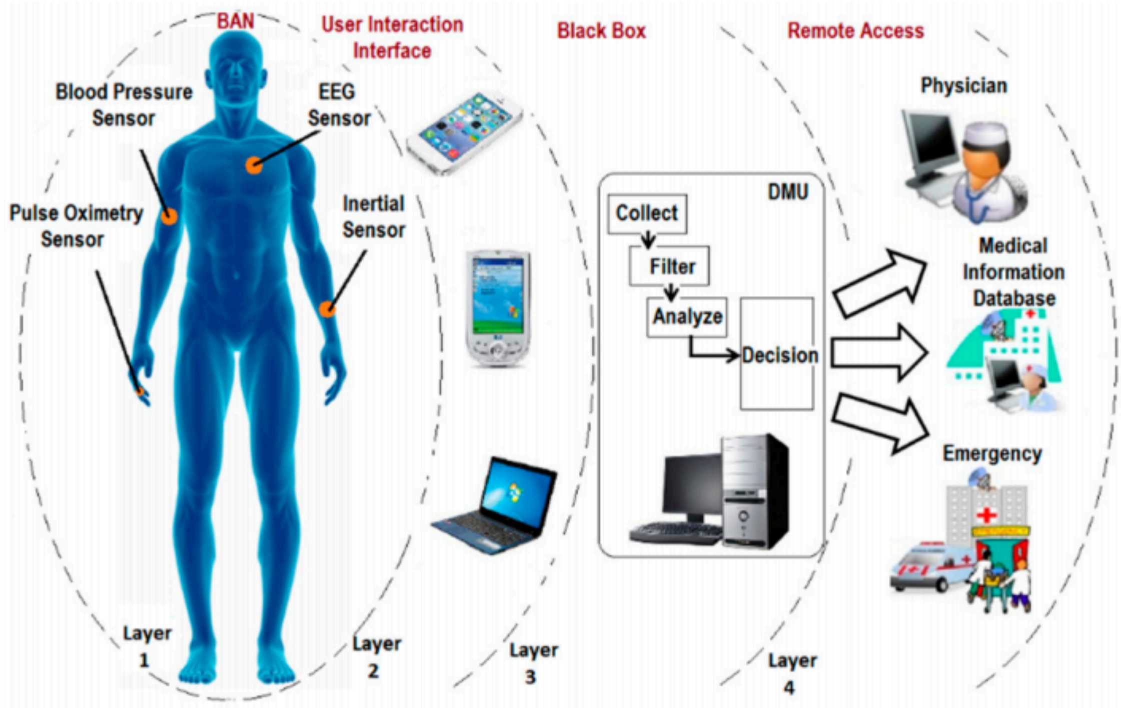


Image 11. A typical architecture of an eHealthcare system. [17]

Physical security of the watch

As we know on many smartphones, there is a possibility to set a passcode that user needs to input to unlock the device. Talking about Apple Watch, there is a possibility to provide a passcode also. For obvious usability reasons, this is not asked every time user starts using the watch. The passcode is prompted whenever user reboots the watch, takes the watch off the wrist and puts it back on or manually locks the watch. The watch can also be unlocked automatically when the paired iPhone is unlocked. [19: 37-38] However, all this configuration is up to user to set up so it might be a potential security risk. Even if all this is set up, if user is unconscious or sleeping, someone might be able to modify the data on the watch. On many smartphones there is also other unlicking methods, for instance face recognition (Face ID on iOS) and fingerprint sensors (Touch ID on iOS). None of this is available on Apple Watch as of writing this paper.

High-level privacy concerns

Once consented at the setup phase, smartphones and wearables automatically gather data of the user at all times. This helps the user to track users' health and user data but it also has privacy concerns. Many commercially Mobile Health (mHealth) applications are sending this data to third parties. In many cases, user does not know where the data is held and where it is sent. [20]

Many applications on the market have inadequate privacy policies. The market is unregulated on how consumers' privacy rights should be protected. Privacy policies often focus on protecting the developers from law suits but do not provide the application's information sharing policies. [20]

3.5.2 Licensing

Before entering commercial market, a solution like this would need to be licenced as medical device. For example, in the European Union, this requires CE marking and in the United States Food and Drug Administration's (FDA) approval. The CE marking process is generally easier and less time-consuming than FDA. However, while FDA approval means that the device is approved in all parts of the world, the CE mark might not even be enough for all the countries in the EU. The main difference between these two approaches is that CE marking process checks that the device does not cause any harm to its users or the environment, while on top of that, FDA checks that the device actually does what it is supposed to do. [21]

4 Proposed prototype

The initial implementation of the application was focusing to take the best parts of each application tested and combining them to one. The approach was quite engineering-based, rather than user-centred. In the implementation phase, everything felt clear, justifiable and user-friendly. Discussions and user-testing with the potential user group however changed the direction quite a bit. Users want an overall, reliable and automated solution that tells them the things they need to know without the need to dig deep. As discussed before, this approach is also the approach Apple wants the watch to deliver.

Based on research, a proposal for an optimal diabetes self-care Apple Watch application was made. This section goes through the most important features of the application. It also proposes different ways to implement the features. The focus group of the minimum viable product (MVP) proposal is type 1 diabetics but also other diabetics can certainly benefit from it. The MVP would include following features:

- Automatic activity tracking.
- Automatic gathering of CGM data.
- Manual data input for the most important values in diabetes management: blood glucose, carbohydrates, bolus insulin.
- Alerts about low and high blood glucose.
- Glanceable blood glucose trend indicator along with latest blood glucose value and active insulin.

4.1 Automatic activity tracking

Apple Watch automatically, out-of-the-box tracks activity and saves it to Apple Health application. Apple Watch also motivates user to live an active life.

A central repository for health and fitness data on iPhone is called HealthKit. Apple provides an API from where it is possible to request for example step count, calories burned, workout data, etc. Activity tracking is quite complex context as there is a fairly vast number of measurable things in physical activity. However, one of the most concrete, motivational and often measured activity data type is daily step count. For this proposal, that will be fetched from HealthKit. [22]

When HealthKit capability is added to the app, the app needs to ask permission to read the step count data. This is illustrated in listing 1.

```
if HKHealthStore.isHealthDataAvailable() {
          guard let stepCount = HKObjectType.quantityType(forIdentifier:
          .stepCount) else { return }
          let healthStore = HKHealthStore()
          healthStore.requestAuthorization(toShare: nil, read: [stepCount]) {
          (success, error) in
               if let error = error, !success {
                    print(error.localizedDescription)
               }
            }
 }
```

Listing 1. Ask for permission for HealthKit to read step data

When the permission has been granted, the application can automatically read the data from HealthKit every once in a while. For now, the daily steps are only persisted to the iOS application and are not shown to the user by this application. However, they can be seen on the Health application. In the future, the steps per day data could be used to send user motivational notifications. To some extent, Apple Watch already does this, though.

For prompting the daily step count from HealthKit, we use HKStatisticsCollectionQuery. Listing 2 shows how the HealthKit query anchor is set and the query is created. This query plots the step count from the current day and yesterday.

```
let calendar = Calendar(identifier: .gregorian)
var interval = DateComponents()
interval.day = 1

// Set the anchor date to midnight
let now = Date()
let startOfDayAnchor = calendar.startOfDay(for: now)

guard let stepsQuantityType = HKQuantityType.quantityType(forIdentifier:
.stepCount), let startDate = Calendar(identifier: .gregorian).date(byAdding:
.day, value: -1, to: now) else { return }

// Create the query
let query = HKStatisticsCollectionQuery(quantityType: stepsQuantityType,
                                                quantitySamplePredicate: nil,
                                                options: .cumulativeSum,
                                                anchorDate: startOfDayAnchor,
                                                intervalComponents: interval)
```

Listing 2. Setting the anchor and creating the HealthKit query.

When the query is created, we set the result handlers to the query. Initial result handler handles the initial query, while statisticsUpdateHandler is called in the background whenever something has changed in Health store.

Possible traveling and changes in time zone provide challenges in daily step count persisting. To tangle this problem, whenever the daily step value is saved or modified in the application, the local date is saved as an attribute. This attribute can then be used as an identifier to find the correct entry to modify as there should only be one daily step count entry per day.

4.2 Automatic gathering of CGM data

If user has CGM sensor in use, some options for automatic gathering of blood glucose data would include requesting data from Nightscout server as seen in section 3.2, using the application programming interfaces (APIs) provided by the sensor manufacturers or polling data from HealthKit. Things to consider are that Nighscout and remote APIs require constant internet access while HealthKit generally has some delay in delivering the values.

At the time writing this paper, there were basically three options of CGM available; Dexcom, Abbott and Medtronic, Dexcom being the most popular [23]. Dexcom exports data to HealthKit, though there is a three-hour delay in exporting the values. With HealthKit approach, the timeliness of the watch application would not be on a desired level as it would not provide any real-time information to the user. With Dexcom, even though constant internet access is required, the best way for now seems to be to use the API.

Dexcom API

In order to use Dexcom API, authentication needs to be made on the iOS application. Dexcom API uses industry standard authentication method called OAuth 2.0. The authentication flow is illustrated in the image 12. If user wants to authorize Dexcom data import, first the application needs to create an authentication URL where the user is taken. Next the user needs to provide their Dexcom account credentials to login. When user authorizes the import, an authorization code is returned as part of HTTP redirect. The partner application parses the code from the URL. This code can be used to obtain access token and refresh token. Access token is sent as part of the HTTP GET requests to the API and is used to authorize the requests. The token has a limited usage time and it needs to be refreshed every once in a while. If the data GET request returns error code 401, a new access token needs to be obtained using the refresh token. [24]

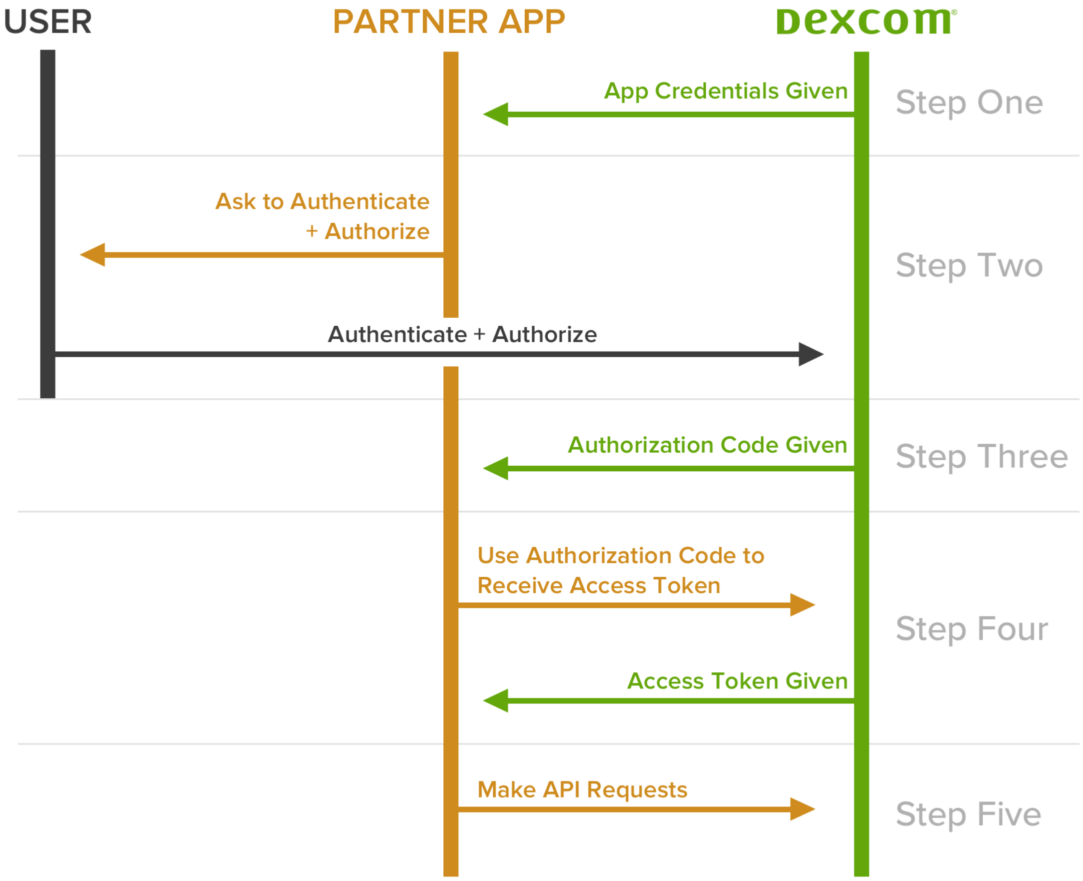


Image 12. Dexcom API authentication flow with OAuth 2.0. [24]

On our application, we use Keychain Services to store the access token and refresh token. Keychain allows us to store small amounts of data securely. After they are successfully saved on iOS app side, they need to be transferred to the counterpart watchOS app so that we can also perform HTTP requests on the watch side, when the phone is not reachable. The connection between devices is encrypted when using WatchConnectivity, so it is secure to transfer these values. The watch has its own Keychain, which is then used to store these values. [25]

Dexcom provides an API for third party developers with several data scopes: [26]

- Estimated Blood Glucose Levels
- Calibration Data
- Events Entry Data
- Device Details
- CGM Statistics

The Watch app could make use of Estimated Blood Glucose Levels that also provides information for the trend indicator. The example response JavaScript Object Notation (JSON) payload is shown in listing 3.

```
{
  "unit": "mg/dL",
  "rateUnit": "mg/dL/min",
  "egvs": [
    {
      "systemTime": "2018-02-06T09:12:35",
      "displayTime": "2018-02-06T01:12:35",
      "value": 122,
      "realtimeValue": 121,
      "smoothedValue": 122,
      "status": null,
      "trend": "flat",
      "trendRate": -0.5
    },
    {
      "systemTime": "2018-02-06T09:07:35",
      "displayTime": "2018-02-06T01:07:35",
      "value": 123,
      "realtimeValue": 124,
      "smoothedValue": 123,
      "status": null,
      "trend": "flat",
      "trendRate": -0.5
    }
  ]
}
```

Listing 3. Example of an API response of Estimated Blood Glucose Levels request. [26]

As discussed before, Dexcom has their own Apple Watch application available. If user has Dexcom CGM sensor, they probably also have the application installed. Thereby the importance of this integration can be questioned as Dexcom's own application does not require continuous network connection. However, to further development, this integration would be profitable. We need the data in order to analyse it.

4.3 Application's user interface

The app utilizes hierarchical navigation technique. The app always starts at the blood glucose scattered chart view which is illustrated on the image 13. The chart shows blood glucose data from the last three hours. Also the latest blood glucose reading is shown on the top of this view. The view is intended to be glanceable.

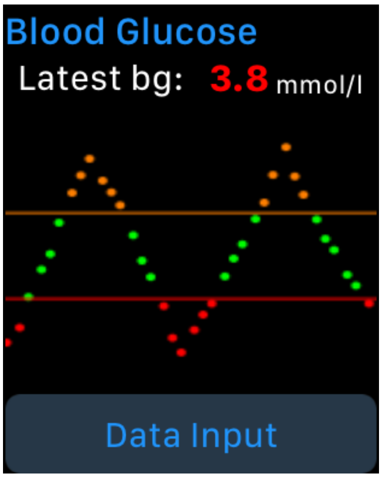


Image 13. Blood glucose scattered chart with latest blood glucose value.

By tapping Data Input button on the first view, user can navigate to data input selection view which is seen on image 14. On this view, user can select the type of data to input. Selection can be made between Blood Glucose, Carbohydrates or Insulin. If voice input is toggled on, the input can be made by voice.

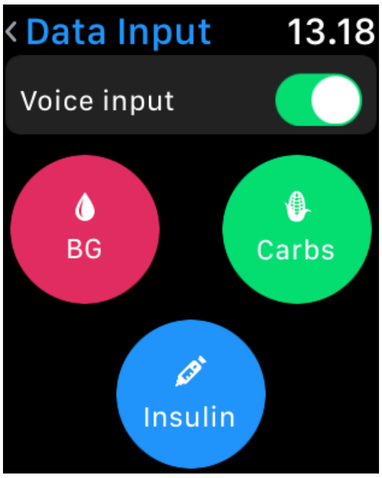


Image 14. Data input selection view.

Data input selection takes user to data input view which is illustrated on image 15. In case of voice input, the received voice payload is parsed and user is taken to the data input view to confirm the value. For decimal values, user can separately select the value from the left and right side of the decimal point. To select the values, user can either tap plus and minus buttons or use the digital crown to scroll between values.

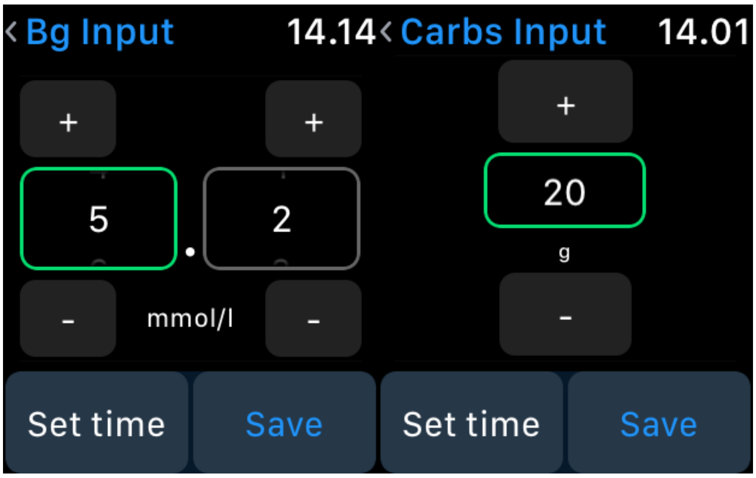


Image 15. Data input for blood glucose and carbohydrates.

The timestamp of the inputted value defaults to current time. If desired, user can also change the timestamp by navigating to time input view seen on image 16. Time input can be made similarly to the value input by using Digital Crown to browse between values.

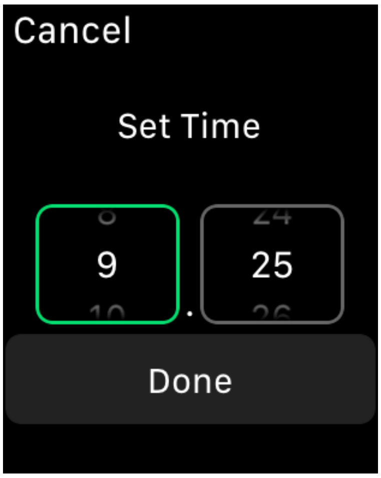


Image 16. Time input.

4.4 Alerts

If CGM is in use, the application can alert user of high and low blood glucose using predefined HYPER and HYPO limits. In case of HYPO, the app can remind user to eat. In case of HYPER, the app can remind user to take insulin or exercise. The data input could be made straight from the notification either by voice or input view. Alert should also be triggered when blood glucose value is decreasing or increasing rapidly. It should also be possible to turn these alerts off. This could possibly be done directly from the alert on the watch and also from the settings on the iOS application which is not implemented as part of this proof-of-concept.

4.5 Watch complications & snapshot

The most important, personal and timely information should be seen in a glance. On the watch face, user can select various complications that show relevant, personal and timely information about current state of the application. For different complication types, user can select following complications:

- Latest blood glucose value along with blood glucose trend indicator.
- Active insulin.

In different use cases, different types of complications make sense. If user is using CGM sensor, latest blood glucose value with trend indicator provides value. In this case, typically insulin pump is in use, so active insulin does not provide value. If user does not have insulin pump in use, active insulin provides value.

The available complications are illustrated in image 17. To some extent, user can modify where the complications are shown. For this proposal, two types of complications are supported, Utilitarian Small for the active insulin (top right corner on the image) and Utilitarian Large for the latest blood glucose value and the trend indicator (the bottom of the image). How the complications are updated is described briefly in section 4.6.4.

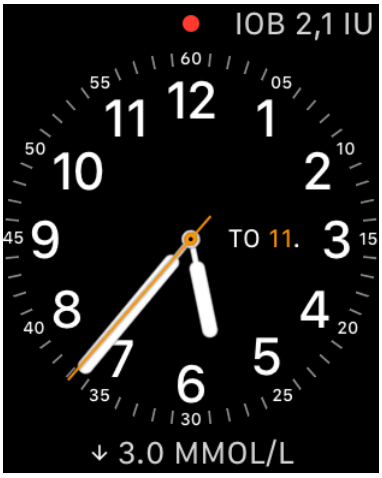
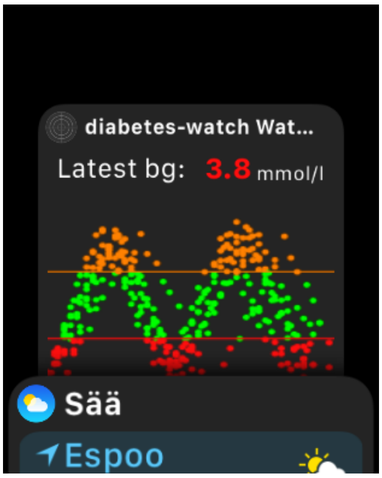


Image 17. Available complications.

Apple Watch also has a dock of application snapshots where the most used applications are shown, as seen in image 18. Apple considers that this is also a crucial part of the application that needs to be kept updated at all times. [27]



diabetes-watch Wat...

Latest bg: 3.8 mmol/l

Sää

Espoo

Image 18. Application snapshot.

4.6 Application architecture

A WatchApp consists of two bundles; WatchKit app and WatchKit extension. WatchKit app contains the storyboards of the app while the WatchKit extension contains the code. At the time of writing this paper, Apple still has not provided the possibility to create a standalone Apple Watch application. The application is always accompanied with an application installed on the user’s iPhone, although the Apple Watch application can be used even when the watch is not connected to the phone. [28]

4.6.1 Two-way communication between watchOS and iOS applications

Hence the standalone solution is not possible at the moment, this application uses iPhone as the data storage. The framework used in transferring data between WatchKit extension and the iOS app is called WatchConnectivity [29].

To setup WatchConnectivity, a WCSession object has to be created and configured in iOS app as well as in watchOS app. In this application, the iOS app and the WatchKit extension both have their own singleton classes called WatchSessionManager. This manager class inherits the WCSessionDelegate class and includes all methods needed for the WatchConnectivity. The WCSession is activated in application’s and extension’s delegate classes.

WCSession provides various ways to pass data between iOS app and WatchKit extension. For data sharing, we use the updateApplicationContext method. This ensures that the data will be transferred in the background even if the counterpart application is not running. If the counterpart app is not reachable at the moment, the method will be queued and the context is updated whenever the counterpart app is reachable again. Notable thing here is that if only one update request can be queued at once and new requests will override the old ones. [30]

The connection flow in transferring medical data between watch and phone is illustrated in the image 19. When new value is inputted or received on watchOS side, it is saved to the session and a request to update the context on iOS app is made. Whenever iOS app receives the context, new values will be persisted to Core Data. After this, the application context update request is made on iOS side. Whenever the application context is received on the watchOS, the current session is updated along with complications.

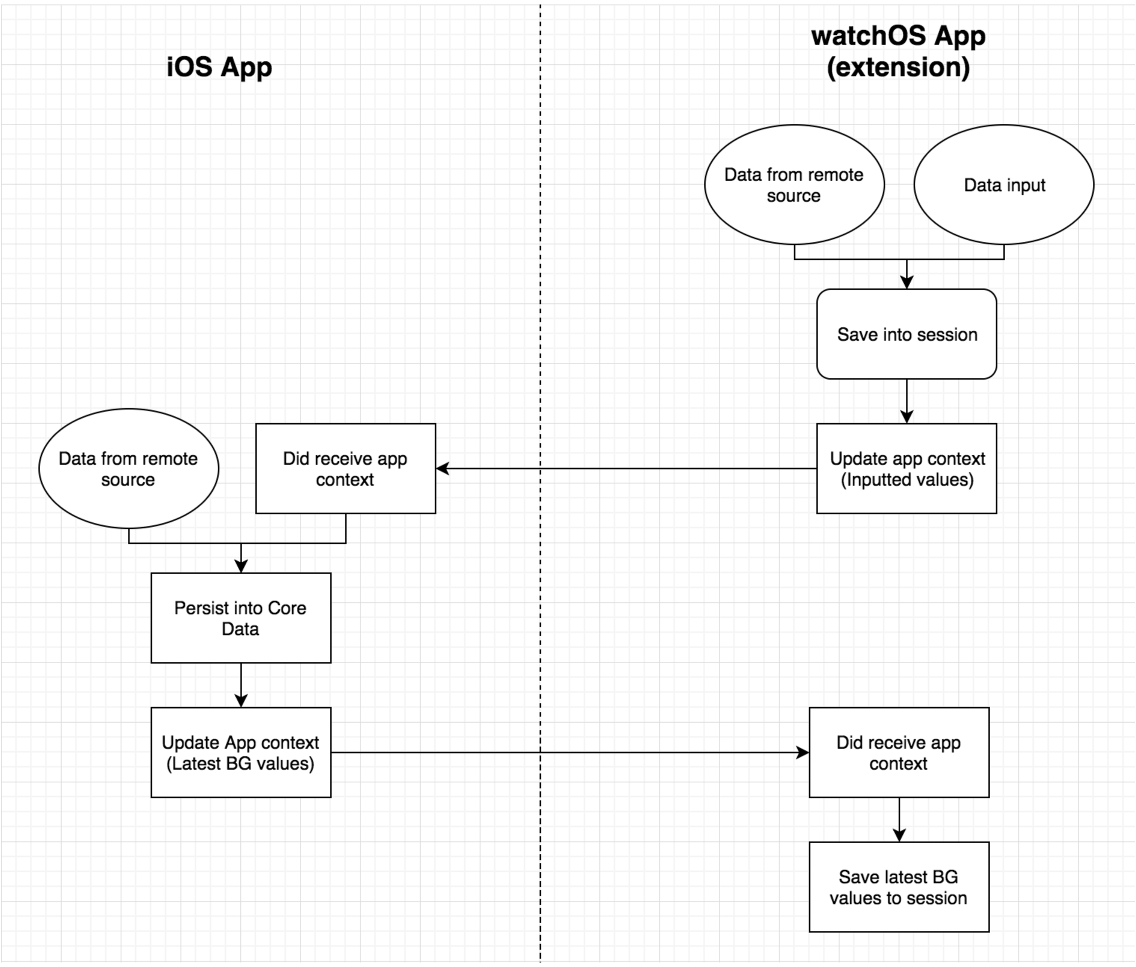


Image 19. Transferring data between iOS app and watchOS app.

4.6.2 Data persistence

On Apple development, Core Data is a native framework to manage the objects in model layer of the application. Core Data has many advantages compared to other model layer solutions, such as SQL. It integrates nicely to iOS tool chains, decreases the amount of code required and there is no need to write SQL queries, just associate an NSPredicate object with the fetch request. On this application, Core Data is the solution used for data persistence. [31]

Managed object model

Managed object model is created in the application’s bundle. The simple object model consists of following entities:

- DataEntry
- DataType
- EntryAttribute

This is a really simplified version of the data model needed for this kind of solution. The Entity-Relationship model is illustrated in the image 20. DataEntry has three attributes; timestamp, value and unit. DataTypes are created on the first launch of the app and DataEntry has one-to-one relationship to DataType. DataEntry has one-to-many relationship to EntryAttribute, meaning that there can be as many attributes on one DataEntry as needed. The keys and values on EntryAttribute can be freely defined.

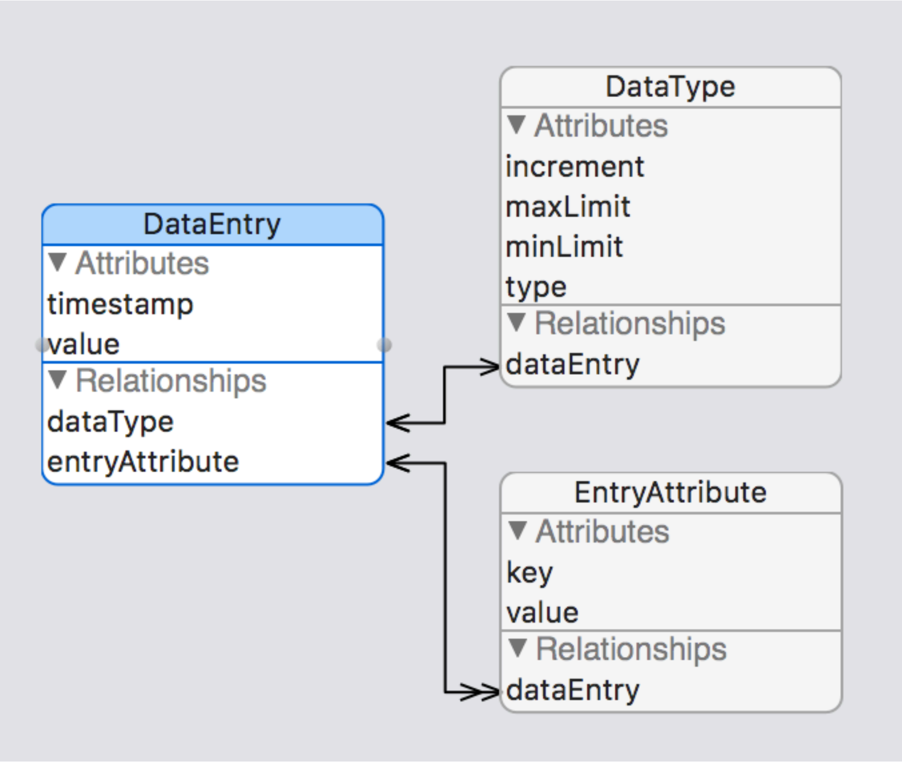


Image 20. Entity-Relationship model of the database.

Core Data persistent container is lazily initialized as seen in listing 4. Lazy initialization makes the container only available once it is needed.

```
lazy var persistentContainer: NSPersistentContainer = {
    let container = NSPersistentContainer(name: "DiabetesWatch")
      container.loadPersistentStores(completionHandler: { (description,
      error) in
      if let error = error {
           fatalError("Unable to load persistent stores: \(
           error.localizedDescription)")
        }
    })
    return container
}()
```

Listing 4. Initialization of NSPersistentContainer

4.6.3 Background refresh

Ideally, the watchOS app and the iOS app both could use background refresh to get latest values from external sources, such as Dexcom API. On watchOS, the WKApplicationRefreshBackgroundTask could be scheduled to happen every 15 minutes. Dexcom provides a new CGM reading every five minutes but to save resources, 15 minutes is the proposed interval. The task would include a URL task to download values from Dexcom API.

4.6.4 Complication updates

Complications are updated every time a change in the application context is received from the counterpart iOS app. They are also updated when new value is manually inputted or received from a remote source on the watchOS app.

5 Further work

In order to implement a fully working eHealthcare system, quite a bit of future work is needed. This paper has focused on creating an optimal watchOS extension to an iOS application. Only a proof-of-concept of this extension has been implemented for now. With only this extension, the system would not fully work. This section goes through some of the features that would be on the bucket list next. Some of these feature include:

- Insulin pump controlling.
- Backend and iOS application.
- Bolus insulin calculator that can make proposes and provide the active insulin value.
- Support for Bluetooth and NFC blood glucose meters.
- Smart reminders and assistants to guide the user to live a healthier and better life, keeping track of the risk factors.
- Automated assessment of carbohydrate intake.

5.1 Insulin pump controlling

Insulin pump is a popular, yet expensive device for diabetes management. Even though the pump makes managing diabetes easier and more automated, using the pump requires good knowledge of diabetes as a disease and some extra caution [32]. During the discussions with diabetics, many people claimed that it would be nice to be able to control the insulin pump from the watch. At the time of writing this paper, none of the most popular insulin pumps were even smartphone controlled. Hence this is a proposal that will be pushed to the future as it is not a straight-forward task with all the licensing and legal regulations.

5.2 Backend

Backend along with an application for healthcare professional is a crucial part of the complete system. A whole thesis or a few could be made of the backend itself and thereby we will only briefly touch the surface of what features the backend could have.

Backend would provide a remote data storage for the user’s data. This data could be then also available for healthcare professionals, friends and relatives through some kind of a web or mobile application, etc. Deploying a cloud-based service would also make possible and require a user authentication. To support scalability and provide better resource handling on the mobile clients, some smart assistants could also run on cloud and they could be modifiable by healthcare professionals. Upcoming 5G cellular mobile connections will bring completely new possibilities for the system.

5.3 iOS application

This thesis focused only to the implementation of the watchOS application. As the watchOS application is only an extension to the iOS application, the app on the phone should be able to do everything the watch can. On top of that, the iOS application should be used to handle the tasks required by the system that should not be done on the watch. For this POC, the iOS application handles the authorization for HealthKit and Dexcom API. On top of this, it should include at least the ability to set personal settings. The settings would need new database entities to be defined in Core Data. Some of the most important settings needed would include, but not limit to the following:

- Daily step target.
- Blood glucose target range with HYPO and HYPER limits.
- Settings needed for bolus calculation, for example insulin in use, insulin offset and acting times, insulin-carbohydrate ratio, insulin sensitivity factor, among others depending on the used algorithms.
- Reminder and notification settings.
- Bluetooth and NFC pairing.

5.4 Bolus insulin calculator

Many type 1 diabetics use insulin pump which automatically pumps insulin into diabetic’s body retaining blood sugar level at a desired level [33]. In this use case, the insulin calculation is done by the insulin pump.

Another use case is when the diabetic patient is injecting insulin dosages. In this case, when user is about to take rapid acting insulin (bolus insulin), the calculator tells how much insulin should be taken based on previous physical activity, carbohydrates, blood sugar levels, etc. When insulin has been inputted, the calculator can provide an active insulin value at any given time. A standard bolus calculation is displayed in formula 1.

$$B = \frac{CHO}{ICR} + \frac{G - G_T}{\mathrm{ISF}} - \mathrm{IOB} \qquad (1)$$

The standard insulin bolus calculation. B is the calculated Bolus, CHO the estimated carbohydrates consumed, ICR the insulin-carbohydrate ratio, G the current blood glucose, G_T the predefined glucose target, ISF the insulin sensitivity factor, IOB the insulin-on-board value. [33]

ISF depends on the individual’s insulin sensitivity. ICR represents how many grams of carbohydrates are covered off by a single unit of insulin. IOB is the remaining active insulin value at a given time.

The parameters of the calculation depend of various factors, such as physical activity, time of day, hormonal cycle, stress level, alcohol consumption and illnesses. There are different algorithms that have been proposed to adjust these factors, one of them is artificial intelligence technique called case-based reasoning (CBR) which is illustrated in image 21. There are four steps in the CBR cycle: [33]

- Retrieve the cases from the case base and select the one that is the closest to the current. This can be done using different similarity metrics, for example k-nearest neighbours.
- Reuse a solution of a retrieved case. If needed, the solution can be adjusted to match the current scenario better.
- Revise the glycaemic outcome if user has accepted the bolus proposal. If the outcome is dissatisfying, the solution of the retrieved case could be adjusted.
- Retain the case in the case base or create new if no previous similar case exists.

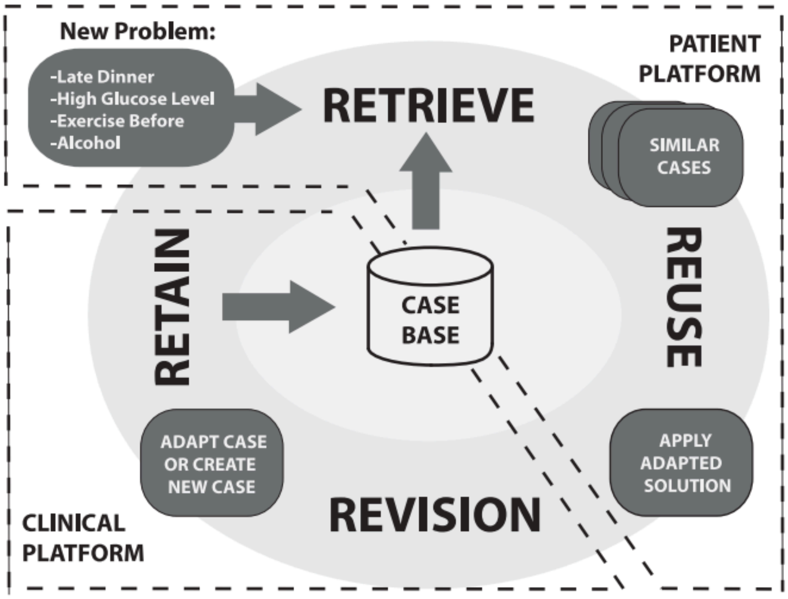


Image 21. A CBR cycle split into clinical and patient platforms. [33]

There are several commercially available iOS applications that already have implemented bolus calculator. One of them is Balansio. Balansio's bolus calculator is a CE certified Class 2b medical device [34]. By integrating the watch app to Balansio, it could be possible to utilize the ready-made algorithms.

5.5 Support for multiple sensors, Bluetooth and NFC meters

This proposal only includes the support for Dexcom CGM sensor. As stated before, other popular sensors include for instance Medtronic and Abbott Libre. The integration for these would also need to be made. Many applications on the market also support multiple wireless Bluetooth and NFC meters that are widely used. Integrations for these would be crucial to widen the target group.

There are several Bluetooth blood glucose meters on the market, which could be integrated to the application. Every time user manually takes blood glucose measurement, the value would automatically be transferred via Bluetooth.

5.6 Smart and modifiable reminders & assistants

Earlier we talked about the alert notifications that provide user with vital information on the current condition. These are the most important notifications but there could be more to help the user. Smart assistants are the most advanced feature discussed in this paper and could be seen as the real breakthrough. They have the real potential to reduce the need of a health-care professional.

Smart reminders and assistants could utilize machine learning principles to learn the lifestyle of the user. They could educate, motivate and guide user to live a healthier and more discipline life. An example of this could be an assistant that reminds and motivates user to be more active physically. As in many applications, this assistant could be turned on by default and user can then turn it off if they do not want to use it.

On the research, many diabetics also hoped for the possibility to set custom reminders. The reminder could remind user for example to measure blood glucose or take medications. Configuring these manually would happen on the iOS application side as we do not want to overpopulate the watchOS application. Potentially user could also set these reminders by voice on the watch side.

The smart assistants could analyse user’s behaviour and connect the data to future events to make the user to see the impact of different actions. The risk factors for diabetics, such as smoking and drinking, should also be taken into account. By applying Clinical Intelligence to the system, it could partly replace healthcare professionals.

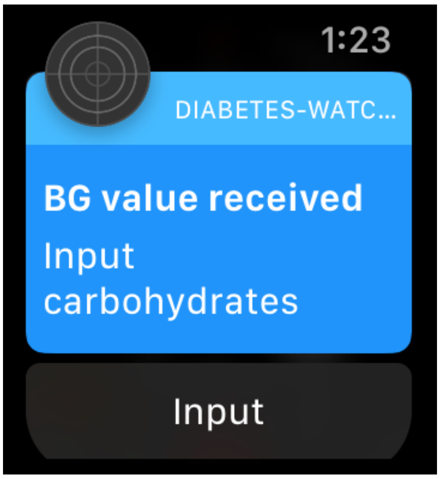
5.7 Automated estimation of carbohydrate intake

Carbohydrate intake assessment is an important task in diabetes self-management. However, it requires training, skills and work to correctly estimate the amount of carbohydrates in the food eaten. Solutions have been proposed to automatically estimate the amount of carbohydrates by using image recognition. One of these systems is GoCARB. User takes a picture of the plate of the food, the application segments the different food items from the picture, recognizes them and creates a 3D-model of those food items to estimate the volume of each item. The average error in GoCARB’s carbohydrates estimation is approximately 10%, which is quite impressive. [35]

In the future, we could create our own system for automated estimation of carbohydrates or integrate an existing system like GoCARB in the application to provide even more automation for the data input. With existing technology provided by Apple, all this would need to happen on the iOS application side as there is no camera on the Apple Watch.

5.8 Improvements for notifications and complications

If user is not using CGM and a blood glucose value is received for example via Bluetooth as described in the previous section, the application could show a notification and user would be able to input carbohydrates either by voice or using the data input view. An example of the notification is seen in image 22. Also, by utilizing bolus calculator, an insulin value could be then proposed to the user. This would make the user experience more personal.



BG value received

Input carbohydrates

Image 22. Example of an actionable notification.

This proof-of-concept was tested mainly on the first-generation Apple Watch that only supports a limited amount of different types of complications. As of watchOS 5, there are new types of complications to be supported.

6 Conclusion

The goal of this thesis was to propose an optimal Apple Watch application for diabetes self-care. The application is indented to make diabetic's life easier and in the bigger picture, decrease the cost of diabetes care. While there are many of these applications currently available in the market, they do not fully match the criteria of diabetics and healthcare professionals. This is one reason why they are not widely used.

Background research of this thesis was done with an online survey as well as by testing some commercially available applications. Some concerns were also raised. The system handles sensitive medical data, so the security and privacy are important. The application proposes treatments for diabetic – something that a healthcare professional could do – hence the application would need a medical device certificate.

Based on the research, first version of the application proof-of-concept was made. This first version was changed quite a bit after face-to-face discussions with the potential user group. The most important findings of the discussions were that users want a reliable system that is easy to use and requires as little user interaction as possible.

The final outcome of the thesis is a small proof-of-concept of an Apple Watch application but more importantly a proposal for a fully working system that could be implemented based on this thesis.

References

1 Diabetes. 2017. Internet Source. World Health Organization. <http://www.who.int/en/news-room/fact-sheets/detail/diabetes>. Read 30.8.2018.

2 Brahmbat, R et al. 2017. Diabetes mHealth Apps: Can they be effective?. Studies in health technology and informatics. Volume: 234, pages: 49-53. DOI: 10.3233/978-1-61499-742-9-49. Read 29.10.2018.

3 Islam, Alvi, Al Mamun. 2016. DiaHealth: A Smart App for Complete Diabetes Lifestyle Management. IEEE. <https://ieeexplore.ieee.org/document/7835396>. Read 30.8.2018.

4 Alexander, B et al. 2017. A behavioral sensing system that promotes positive lifestyle changes and improves metabolic control among adults with type 2 diabetes. IEEE. <https://ieeexplore.ieee.org//document/7937732>. Read 2.9.2018.

5 Baranyi, R; Willinger, R; Lederer, N; Walcher, F; Grechenig, T. 2018. DiaBeaThis — A gamified self-tracking portal to support people suffering from diabetes mellitus to control their blood glucose level. IEEE. < https://ieeexplore.ieee.org/document/8401352>. Read 2.9.2018.

6 Eden K et al. 2018. Mobile Applications for Self-Management of Diabetes. Technical Brief No. 31. Agency for Healthcare Research and Quality. DOI: 10.1007/s11606-018-4410-1. Read 29.10.2018.

7 Dam, R; Siang, T. 5 Stages in the Design Thinking Process. Internet Source. The Interaction Design Fundation ApS. <https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>. Read 25.10.2018.

8 Solutions Design & Consulting. Internet Source. DWT De Novo. <https://denovo.dwt.com/>. Read 25.10.2018.

9 Apple Watch. Internet Source. Apple Inc. <https://www.apple.com/lae/watch/>. Read 4.9.2018.

10 Reisinger, D. 2015. Apple Watch design wins patent on look and feel. Internet Source. CNET. < https://www.cnet.com/news/apple-watch-design-wins-patent-on-look-and-feel/>. Read 5.9.2018.

11 Gurpeen. 2017. How to use the Digital Crown and side button on the Apple Watch. Internet Source. TechGreatest. <https://www.cnet.com/news/apple-watch-design-wins-patent-on-look-and-feel/>. Read 5.9.2018.

12 watchOS Design Themes. Internet Source. Apple Developer. <https://developer.apple.com/design/human-interface-guidelines/watchos/overview/themes/>. Read 10.9.2018.

13 Planning a Great Apple Watch Experience. 2018. Video Source. Apple Developer WWDC. <https://developer.apple.com/videos/play/wwdc2017/808/>. Watched 8.8.2018.

14 Nightscout. Internet Source. Nightscout Project. <http://www.nightscout.info/>. Read 5.10.2018.

15 Nightguard. Internet Source. <https://github.com/nightscout/nightguard>. Read 5.10.2018.

16 Dexcom Apps. Internet Source. Dexcom. <https://www.dexcom.com/apps>. Read 6.10.2018.

17 Ghamari, M.; Janko, B.; Sherratt, R.S.; Harwin, W.; Piechockic, R.; Soltanpur, C.. 2016. A Survey on Wireless Body Area Networks for eHealthcare Systems in Residential Environments. Sensors. Volume: 16, Issue: 831. Read 6.9.2018.

18 Li, M; Lou, W; Ren, K. 2010. Data security and privacy in wireless body area networks. IEEE Wireless Communications. Volume: 17, Issue: 1, pages 51-58. Read 4.9.2018.

19 iOS Security Guide. 2018. Apple Inc. Read 12.10.2018.

20 Addonizzio, G. 2017. The Privacy Risks Surrounding Consumer Health and Fitness Apps, Associated Wearable Devices, and HIPAA's Limitations. Seton Hall University. Read 18.10.2018.

21 Conley, D. 2015. Two Paths for Medical Device Approval: FDA vs. CE. HealthManagement. Volume: 15, Issue: 2, 2015. Read 15.10.2018.

22 HKStatisticCollectionQuery. Internet Source. Apple Developer. <https://developer.apple.com/documentation/healthkit/hkstatisticscollectionquery>. Read 17.10.2018.

23 Scheiner, G. 2018. Choosing a CGM: 3 Heads are Better Than One. Internet Source. Integrated Diabetes Services. <https://integrateddiabetes.com/choosing-a-cgm-3-heads-are-better-than-one/>. Read 5.10.2018.

24 Authentication. Internet Source. Developer.dexcom. <https://developer.dexcom.com/authentication>. Read 5.10.2018

25 Storing Keys in the Keaychain. Internet Source. Apple Developer. <https://developer.apple.com/documentation/security/certificate_key_and_trust_services/keys/storing_keys_in_the_keychain>. Read 10.10.2018

26 Scopes & Access. Internet Source. developer.dexcom. <https://developer.dexcom.com/scopes-access>. Read 5.10.2018.

27 Keeping Your Watch App Up To Date. 2016. Video Source. <https://developer.apple.com/videos/play/wwdc2016/218/>. Watched 12.10.2018.

28 Developing for Apple Watch. Internet Source. Apple Developer. <https://developer.apple.com/library/archive/documentation/General/Conceptual/WatchKitProgrammingGuide/#//apple_ref/doc/uid/TP40014969-CH8-SW2>. Read 4.9.2018.

29 Watch Connectivity. Internet Source. Apple Developer. <https://developer.apple.com/documentation/watchconnectivity>. Read 4.9.2018.

30 WCSession. Internet Source. Apple Developer. <https://developer.apple.com/documentation/watchconnectivity/wcsession>. Read 13.9.2018

31 Core Data. Internet Source. Apple Developer. <https://developer.apple.com/documentation/coredata>. Read 4.9.2018.

32 Insulin Pumps. Internet Source. Diabète Québec. <https://www.diabete.qc.ca/en/living-with-diabetes/care-and-treatment/drugs-and-insulin/insulin-pumpsf>. Read 5.10.2018.

33 Pesl, P et al. An Advanced Bolus Calculator for Type 1 Diabetes: System Architecture and Usability Results. 2016. IEEE Journal of Biomedical and Health Informatics. Volume: 20, Issue: 1, pages 11-17. Read 9.10.2018.

34 Balansio. Internet Source. Quattro Folia. <https://www.balansio.com/individuals>. Read 10.10.2018.

35 Anthimopoulos M; Dehais J; Mougiakakou S. 2016. GoCARB: A Smartphone Application for Automatic Assessment of Carbohydrate Intake. MADiMa '16 Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management. Pages 91-91. DOI: 10.1145/2986035.2986046. Read 29.10.2018.

Online survey questions

1. Gender

- Male
- Female
- Other

2. Age

- 0-15
- 16 - 25
- 26 - 35
- 36 - 45
- 46 - 55
- 56 - 65
- More than 65

3. I am

- A diabetic
- Working with diabetes
- A diabetic working with diabetes
- Just interested in the subject

Basic information about your diabetes

1. Type of your diabetes

- Type I
- Type II
- Other:

2. For how long have you had diabetes?

- Less than 1 year
- 1 - 5 years
- More than 5 years

3. What do you see as the biggest challenges in living with diabetes? (open question)

Smart self-management

1. Do you use a smartphone? If you do, which operating system does it run on?

- Android (Samsung, Huawei, OnePlus, Sony, LG, etc.)
- iOS (Apple iPhone)
- Other
- I do not know
- I do not use a smartphone

2. Do you use smartwatch? If you do, which one?

- Android wear
- Apple watch
- Other smartwatch
- I do not use a smartwatch

Self-management tools

1. Have you familiarized yourself with some smart tools used in diabetes self-management, such as smartphone and smartwatch applications?

- Yes

- No

2. What kind of self-management tools do you use? (multiple choice)

- Insulin pump
- Bluetooth blood glucose meter
- Smartphone app for diabetes self-management
- Smartwatch applications for diabetes self-management
- Other:

3. 3. If you do not use a smartwatch but you would see a benefit from using it in your everyday life as a diabetic, would you use it?

- Yes
- No
- Maybe

Smartwatch

Smartwatch is a wrist watch that brings the features of a smartphone to your wrist. It can also measure different types of values more accurately than a smartphone. Examples of these being pulse, step count, energy consumption, etc.

1. Do you have experiences in smartwatch applications for diabetes self-management? What kind of experiences? What applications have you used? (open question)

2. What would be some important or valuable features of a smartwatch application for diabetes self-management? (open question)

3. Would you like that the watch reminds you of something? What would it be? (open question)

4. What would you like the watch to show you on a glance? (open question)

5. What would you like the watch or the application to track? (open question)