UiT THE ARCTIC UNIVERSITY OF NORWAY

Faculty of Science and Technology
Department of Computer Science

Internet of Things Mini Display-based Motivation, Notification and Warning System for Groups of People with Diabetes Type 1

—

Martin Haugen Mikalsen

Master thesis in INF-3981 June 2018

“Simplicity is prerequisite for reliability.”
–Edsger Dijkstra

“One of my most productive days was
throwing away 1,000 lines of code”
–Ken Thompson

Abstract

Diabetes is a complex and time-consuming affair for the people that are afflicted by the disease. A strict self-management regime needs to be followed to avoid short- and long-term complications, which requires a great deal of motivation and support from others. Low- or high-blood glucose levels can cause short- and long-term consequences, ranging from headaches and thirst to coma. In the last few years, new technology has made it easier for diabetics to control their blood glucose levels. In this thesis. we explore the possibility and methods of integration data from Continues Glucose Meters and fitness trackers, into small social network groups. Here they can share their blood glucose levels with each other's and discuss their condition in a safe environment. In addition, to prevent high- and low- blood glucose values, there is also an alarm function implemented, which will share contact information to group members in case of an emergency. Motivation plays a large part in the self-management routine of people with diabetes, therefore several motivation factors were included in the thesis project. By interviewing people with diabetes and working with motivation in earlier work, we found features that should be included in an application that can support people with diabetes. The result of the thesis is a mobile application, and an Internet of Things device, both of which can show text-messages and Blood glucose levels of group members. The Internet of Things device has an attached electronic-ink display and the idea behind it is that these can be placed around the house, office and so on. To verify the usability of the system a small user-study was conducted, the results suggest a good potential for the system.

Acknowledgements

I would like to start by thanking my supervisors Gunnar Hartvigsen, Miroslav Munzy and Eirik Årsand for their support, guidance, suggestions and general help throughout the project duration. I've would also thank the participants of both the feature and user-study, for their time.

The idea of the project came from the *Crazy new ideas* workshop held in the spring of 2017 with the Medical Information & Telemedicine (MI&T) group. The idea was first suggested by Hartvigsen as a simple cube that could show the Blood glucose levels of family or friends in a traffic light method (Green for ok, Yellow for warning and Red for a severe event). With discussion, this idea was developed further, with the suggestion from the author and Munzy we changed the idea to use an E-ink screen instead of lights and attach that to a small microcontroller. The idea was explored in a Capstone project, which resulted in an early prototype of the device that could display messages and blood glucose values, together with a working backend service that facilitated making groups, message and so on. For this master thesis, we developed the idea further, with a focus on diabetes alarms, notifications, and motivation in addition to general improvements to the system.

Martin Haugen Mikalsen
01.06.2018

Contents

List of Figures

List of Tables

1

Introduction

Diabetes Mellitus is one of the most common chronic metabolic disorders, affecting over 422 million persons worldwide, and is suspected to more than double by 2030.[5] The disorder required patients to follow a strict and time-consuming routine every single day. Without proper self-management and glycemic control, the patient can risk both short-term and long-term complications. Too high or low blood sugar can cause hypoglycemia and hyperglycemia respectively, a severe hypoglycemia event can cause the person to go into an induce seizure or coma and be life-treating without assistance from another person. [1] With the rapid development of technology, new devices have become available for diabetics to help and improve their lives. One of these is Continuous Glucose Monitoring Sensors, which is a small sensor which is placed on the skin, and read the blood glucose levels, warn about hypo and hyper events, and upload the data to the cloud.[2] Diabetes self-management is an important aspect of the daily life of people with diabetes, and the average person spends around 2 hours a day on this. But there is evidence that less than 50% of diabetics meet the glycemic goals recommended by the American Diabetes Association (ADA)[3], even less for adolescence[4]. Motivation is an important factor in allowing persons to adhere to regimes of self-management.

1.1 Goals and Limitations

The main goal of this project is to develop a motivation, notification and warning system for small groups of people with diabetes using Internet of Things (IoT) mini displays. The idea is to let the users/group members, or their Diabetes Diary app, NightScout and/or medical sensors, communicate through sending messages etc. to mini displays that might be located anywhere .In this way, each member of the group will know that in case of a life-threatening hypoglycemia situation there are people that can take action, either themselves or through contacting emergency personnel. In addition, small group also represents a private forum/social network where the disorder can be discussed away from the public. The main feature of the application is the notification/warning features on low- and high blood glucose, which are collected from their own personal NightScout backend, which supplies data from continuous glucose monitoring sensors. To display the messages and blood glucose readings, there are two implemented frontend applications, one for smartphones and one small inexpensive IoT-enabled device with an electric-ink display. For prolonged use and motivation to not only use the application but also improve their diabetic management we have incorporated motivational mechanisms in our application.

We limit yourself to patients with Type 1 Diabetes, but there is nothing that stops this application to be useful for persons with Type 2 diabetes, other chronic condition. We also not intend to be a replacement for other tools, we simply try to add a platform where groups of people can monitor, talk about their disorder and motivate each other.

1.2 Methods

We developed a prototype, based on input from diabetics we interviewed, and from the knowledge gained by working together with a group trying to find motivational factors for prolonged used in mobile health applications.

Input of the general design and functionalities underway was provided by input done by supervisors and family/friends. At the end of the project, we conducted a small user-study to gain knowledge about their thoughts on the general design, usability and motivational aspect of the application.

1.3 Organization

The rest of the thesis is structured as follows:

- Chapter 2 - goes into the theoretical background of Diabetes and Motivation.
- Chapter 3 - explains some of the major technical theoretical background, which include Internet of Things and Serverless architecture.
- Chapter 4 - explains the methods and materials, which include the research methods, and present the protocol and systems used to build the system.
- Chapter 5 - Present the requirements found through interviews and research, we also present a scenario which involves following three persons with diabetes for 24 hours, and how they can utilize the system.
- Chapter 6 - Related work present some of the products that have a similar function to the system presented in this text.
- Chapter 5 - Present the design and implementation of the system, which includes the Server and the two frontends.
- Chapter 6 - discusses the scenario in light on how this might help persons with diabetes. In addition, we discuss the design choices and alternatives related to the different components of the system. Finally, the results from user-study interviews are presented and discussed.
- Chapter 7 - suggestions for improvements and/or new features that could be added to the system in the future.
- Chapter 8 - Conclusion remarks for this project.

2

Background

2.1 Diabetes Mellitus

Diabetes Mellitus is one of the most common chronic metabolic disorders defined by the insufficient or inefficient insulin delivery by the pancreatic β-cells, causing chronic *hyperglycemia*. Without diagnose and treatment patients risk long-term complications such as heart disease, stroke, vascular disease, blindness, nerve damage, amputation and kidney disease. World Health Organization (WHO) estimated that in 2014 422 million people have diabetes, which is a steep rise from 108 million reported in 1980. This number is likely to more than double by 2030 without any action taken. The disorder is rising more rapidly in middle- and low-income countries[5]. In Norway, about 245 000 people (4.7%) have a diabetes diagnosis, with additional undiagnosed cases. Of these 28 000 have type 1 diabetes and 216 000 have type 2 diabetes[6].

2.1.1 Types of Diabetes

Diabetes is categorized into types: type 1, type 2 and gestational diabetes which are a temperately condition that occurs during pregnancy[7]. There is also suggested to be a type 3 diabetes which is linked to Alzheimer's disease and is termed as a form of diabetes that selectively involves the brain and has features that overlap both type 1 and type 2 diabetes. [8]

Type 1 Diabetes

Type 1 Diabetes(T1D) is the result of pancreatic β-cell destruction, which results in that the person can't produce insulin and require constant exogenous insulin therapy and blood glucose monitoring. The goal of the treatment is optimal BG control, which involves insulin replacement therapy, extensive education, and disease self-management[9].

Type 2

Type 2 Diabetes(T2D) is the more frequent form of diabetes and occur when the body does not use insulin properly and is frequently called *insulin resistance*, declining insulin production and eventual pancreatic β-cell failure[10]. Overweight[11], hypertension and dyslipidaemia (prolonged elevation of insulin levels) are T2D risk factors[12]

2.1.2 Hypoglycemia

The European Medicines Agency define a symptomatic hypoglycemia as: *An event during which typical symptoms of hypoglycemia are accompanied by a measured plasma glucose concentration less than or equal to 3.9 mmol/L (70 mg/dL)*, while a severe hypoglycemia is defined as: *An event requiring assistance of another person to actively administer carbohydrate, glucagon, or other resuscitative actions. These episodes may be associated with sufficient neuroglycopenia(shortage of glucose in the brain) to induce seizure or coma.* [9] Hypoglycemia is a feared condition for diabetics. According to Cryer the average T1D diabetic has 1-2 occurrences of hypoglycemia each week-, and 10 to 25% suffer at least one episode of severe hypoglycemia each month[13]. Hypoglycemia can be caused by a number of factors like high or ill-timed insulin doses, overnight fasts, excessive physical activity and alcohol ingestion, table 2.1 summarizes the causes of hypoglycemia[14]. Hypoglycemia may be associated with a range of symptoms progressing from sweating and hunger to cognitive dysfunction, seizures, coma and even sudden cardiac death.

Nocturnal Hypoglycemia , as the name suggests, is a hypoglycemia situation which happens when the subject is sleeping. Severe hypoglycemic episodes often occur during sleep and can be caused by insufficient food intake and/or inappropriate insulin dosage the previous night. Nocturnal hypoglycemia is rather common with up to 50% of adults and 78% of children and can last for multiple hours[15]. The condition has been attributed to sudden nocturnal deaths (known as "dead in bed" syndrome), which account for 5%–6% of all

Causes	Example
Incorrect insulin administration	Insulin is taken in excess or at the wrong time relative to food intake and/or physical activity; incorrect type of insulin taken
Insufficient exogenous carbohydrate	Delayed or missed meals or overnight fast
Decreased endogenous glucose production	Excess alcohol consumption
Increased utilization of carbohydrate/depletion of hepatic glycogen stores	Exercise or weight loss
Increased insulin sensitivity	During the night, exercise, weight loss
Delayed gastric emptying	Conditions such as gastroparesis
Decreased insulin clearance	Conditions such as progressive renal failure

Table 2.1: Causes of hypoglycemia

death among young people < 40years with T1D [16]. And is maybe the most feared condition diabetics face in their daily life.

Hypoglycemia Unawareness , a portion of persons which has had diabetes for a long time can develop an immunity to the warning symptoms and therefore not act on it (drink/eat) to prevent the onset of hypoglycemia[13], and thus the risk of severe episodes is increased by more than 5 times compared to other diabetics [17]. This can be not only dangerous for the subject, but also to others because of sudden unexpected blackouts, for example, while operating a motor vehicle[18].

Quality of life impact , the fear of having a hypoglycemia situation is increased after experiencing a non-severe event, and the patient often tends to monitor their glucose more carefully and reduce their normal number of doses or units of insulin. In addition, reports of increased tiredness, sleeping difficulties and absent from work. 37.8% T1D and 29.9% T2D diabetics, reported an increased fear of future hypoglycemia following a non-severe hypoglycemic event, while a larger proportion of patients (63.6% and 84.2%, respectively) reported increased fear following a severe hypoglycemic event[19]. The same study also reports reduction to the Health-Related Quality of Life (HRQoL), with greater negative effect for severe hypoglycemic events than non-severe events, and nocturnal events compared to daytime events.

2.1.3 Hyperglycemia

Hyperglycemia, or high blood sugar and occur when the body has too little insulin for processing the intake of glucose. Hyperglycemia can lead to serious problems when it goes untreated for a longer period of time, the most serious is ketoacidosis (diabetic coma). This condition can occur when your body does not have enough insulin, and the body can't use glucose for fuel, so the body starts breaking down fats to fuel instead. The waste product of this process is ketones, which the body can't tolerate a large amount of, so it's discarded through urine. When the body can't discard all the waste product it builds up and can eventually cause ketoacidosis. [20]

Other complication includes:

- Heart and blood vessel diseases, which can increase the risk of heart attack, stroke and peripheral artery disease.
- Kidney function reduction, which may lead to kidney failure.
- Nerve damage.
- Eye diseases, which can lead to reduced or even blindness.

These complications often occur after a long period of time with bad frequent hyperglycemia and general poor glucose control.

Early warnings signs of hyperglycemia include a high BGL of over 8 mmol/l, high sugar content in urine, frequent urination and increase thirst. There are two main treatments for this condition 1) exercise 2) insulin treatment.

2.1.4 Technological Support in Treatment of Diabetes

Technology can help ease the burden of diabetes, some of these technologies are Continuous glucose monitoring sensors, insulin pumps, and electronic diaries.

Continuous glucose monitoring sensors

Continuous glucose monitoring (CGM) sensors (see figure2.1) are portable and small devices that allow for measuring the blood glucose levels in real time continuous for several days without the need for changing the sensor. CGM devices are made up of three components[2]:

Figure 2.1: A Dexcom G4 CGM device (source: www.dexcom.com)

- Needle-based sensor, which is usually inserted in the abdominal subcutis and measures an electrical signal proportional to the glucose concentration present in the interstitial fluid.

- Transmitter, often applied over the sensor.

- Portable device, which can receive the signals from the transmitter (often a mobile phone), and a visual application that shows the data to the user.

The benefits of CGM devices are that instead of self-monitoring of blood glucose (SMBG, which is done through pricking the finger, extracting a drop of blood before testing it in a small device), is that the process is automatic, continues and provides both hypoglycemia warnings and visual feedback through the mobile application or a separate receiver. Deiss et al. showed that patients with CGM's reported making adjust their treatment and/or changes to their lifestyle and food intake, but could not determine a link between real-time CGM and the improvement in glycemic control because of no registering of information about their self-management of diabetes therapy on a daily basis[21]. Tamborlane et al. found a strong correlation between CGM benefits and age. People over 25 years old displayed a substantially tighter glycemic control, while patients between 8 to 14-year-old showed less benefit and who were 15 to 24 years of age saw no benefit at all[22]. A suspected reason for this is parental involvement in the treatment, and the transition from parental assistance to self-management in adolescents, which has been shown to deteriorate the glycemic control[23]. There are also studies that CGM is associated with a significant reduction in HbA1c percentage, with the greatest reduction in those with the highest HbA1c and the ones that most frequently used the sensor. The reduction of hypoglycemia is also shown to be reduced by wearing a CGM. [24][25]

Even though CGM's is useful, improves the glycemic control in most cases and makes the life of a diabetics easier, the officially approved guidelines set by the regulatory authorities does not recommend to use the data for determining insulin doses because of the fact that CGM readings often does not correspond exactly to SMBG measurement results taken at the same time—especially during fast changes in BG[26][27]. CGMs measure glucose levels in interstitial fluid (ISF) and not directly in blood as SMBG, do not instantly transfer between the two compartments which result in different measurements, before equalize after some minutes. This time-delay has been estimated to be around 5-10 minutes [28][29][26], but there is also a strong correlation between the time delays and the patients carbohydrate-to-insulin-ratio and the total daily dose of insulin, as shown by Reiterer et al.[30]. In addition to this delay there is also delay associated with sampling, transfer, and processing of the data by the CGM sensor itself, this substantially lower than the delay discussed above, and sensor-specific, according to Barry et al.[31] the delay is in terms of 1-2 minutes.

Insulin Pumps

Continuous subcutaneous insulin infusion, or simply insulin pumps is an important and evolving form of insulin delivery, which is mainly used for people with type 1 diabetes. The pump is a portable electromechanical pump which mimics a none-diabetics insulin delivery[32]. Pairing a insulin pump and an CGM is often used for patients that has problems regulation their BGL, and can function as a artificiality patricians or closed loop system, an example of this is OpenAPS[1], powered and developed by NightScout.

Diabetes Diaries

Diaries are a crucial part of diabetes self-management, they usually list the BGL, insulin injections, food intake and physical activity. In the past these diaries were paper-based, which made it tedious to record and review parameters, few patients use paper diaries over longer periods, even if it gave the person an improved disease management[33]. Today, several different electronic diaries exist, which are available for a number of devices, including the two major platforms iOS and Android. One of these which are the *Diabetes Diary*[2] developed by Nasjonalt senter for e-helseforskning (NSE).

1. https://openaps.org/
2. http://www.diabetesdagboka.no/nb/

2.2 Motivation

Diabetes, like most chronic disorders, is complex and time-consuming, where the completion of several regimen components is required. For example, it requires the frequent administration of insulin, checking blood glucose levels, incorporating physical activity, and counting carbohydrates to be able to cope well with the disorder. Failure to adequately complete these daily-tasks might lead to high or low blood sugar which can result into both short and long-term diabetes complications[1]. The recommended self-care would require more than 2 hours daily, with more for elderly newly diagnosed patients, while diet and exercise are the most time-consuming tasks[34].

Motivation to adhere to the strict and necessary regime is a challenge for most diabetics, for example studies on T2D indicate that less than 50% of patients[3] achieve the glycemic goals recommended by the American Diabetes Association (ADA), while adolescents with T1D have been reported to only have a 21% adherence to the same goals[4]. There is evidence that diabetes has an increased rate of depression [35] and depression symptoms are suggested to be associated with worse diabetes self-care and poorer diabetes control [36].

2.2.1 Self-Efficacy

Social Cognitive Theory, is the view that people learn by watching others and explains the personality in terms of how a person thinks about and responds to one's social environment. Albert Bandura is the pioneer of this theory.He has argued that when people see someone else awarded for behaviour, they tend to behave the same way to attain an award. People are also more likely to imitate those with whom they identify.

A key concept of social-cognitive theory is *Self-Efficacy*, which is defined as people's beliefs in his/her innate ability to achieve goals. Bandura defines it as: *people's beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives. Self-efficacy beliefs determine how people feel, think, motivate themselves and behave.* [37]. Self-efficacy contributes to motivation in multiple ways [38]:

- shaping aspirations and goals
- determining the amount of effort and determination a person will put into reaching that goal
- shaping the result expected from one's efforts

People who conceive themselves highly will except good results, while those who do not will be less confident in themselves will envision/expect lesser results. in addition, progressive mastery of an activity leads to satisfaction, which in turn serves as an ongoing motivational factor.

2.2.2 Self-Determination Theory

Self-determination Theory (SDT) is built around the distinction between motivations that are autonomous vs. those who are controlled. Autonomous motivation is tied to the notation behaviour where the person themselves feel enjoyment for the behaviour. Behaviour that is controlled can be tied to doing something to get a reward or avoid some punishment, it could also be external pressure, which makes a person do something. [39] STD beliefs that all of humanity has a set of personal physiological needs which are needed for optimal well-begin and performance: [40]

- **Competence** - That the person feels confident, effective with relation to the activity.
- **Relativeness** - To be cared for, care for others and belonging in various groups that are important to the person.
- **Autonomy** - Feeling joy, pleasure, and joy about what the activity which are performed.

Without these needs met, there will be negative physiological consequences.[41]

Further, there are three concepts that need to be explained: the first is intrinsic motivations, which means that a behaviour or activity is performed for themselves, in order to experience pleasure, joy, and satisfaction from the activity. This can be seen as an inner drive that influences the individual to succeed. Extrinsic motivation, on the other hand, involves performing a behaviour due to external factors, like receiving rewards or avoiding punishment. The last one is amotivation referees to the lack or absence of motivation, for example not feeling joy or not doing it because of the promise of reward or avoiding punishment.[42] There are two types of autonomous motivation[40], the first is intrinsic while the other is a sub-type of extrinsic motivation, called integrated regulation. Integrated regulation is where the behaviour is emitted out of choice, and the person sees that the behaviour is valued and judged as important for the individual, even when if the activity is not pleasant. [42] Deci explains this by saying: *The individual internalizes extrinsic motivation understand the value of the activity and integrate it, so it becomes a part of themselves. By doing this they will be autonomously motivated*[41]. This could be for example a diabetic person, who has struggled with hypoglycemia in the past, and now has started to study and improve their adherence to self-management and glucose control, to avoid the risk of short- and long-term complications.

Senécal et al. show that both self-efficient and STD were associated with adherence and life satisfaction associated with adherence. [38] While Willam et al. showed that diabetics with autonomy supportive health care providers patients become more motivated to regulate their BGL, feel more able to regulate and showed an improvement in their HbA_1c values. [39]

2.2.3 Gamification

Gamification is defined as the implementation of game elements in real-world contexts and foster personal motivation and performance in regard to the activity the elements are applied to. This will enchant services/application with motivational mechanisms in order to invoke gameful experiences and further behaviour outcomes[43]. The general idea is that gamification is just adding game mechanics into a service which in turn automatically becomes more engaging and attains a better retention of customers[44]. This can be used in eHealth applications or serious games to motivate/engage the users to keep using the application for a longer period of time, and thus increasing the health benefits. Some of the most used gamification methods are:

- Badges
- Leaderboards
- Points
- Levels
- Story/Theme
- Goals
- Feedback
- Rewards
- Progress
- Challenges

Hamari et al. did a systematic review of gamification effects, and they found that a majority of the reviewed studies showed a psychological or behavioural effect/benefit. They reviewed papers reported positive experiences(e.g. on engagement and enjoyment) from all the studies, but some of the studies also found that the positive results decreased over long-term usage, the cause might be the loss of interest in badges/achievements unless new challenges and mechanisms were added[45].

3

State Of The Art

The Internet of Things(IoT) is relatively new and novel paradigm that is in rapid growth. The basic idea is of IoT is due to the variety of object- such as RFID, NFC, Sensors, smartphone, etc. which can interact with each other by having a distinct address. These objects are transformed from being dumb to smart by manipulating its underlying technologies such as omnipresent and pervasive computing, embedded devices, communication technologies, sensor networks, protocols, and applications. [46] The term was first defined by Kevin Ashton in 1999. The original concept describes uniquely identifiable interoperable connected objects with radio-frequency identification (RFID) technology [47]. Since it was defined in the late 90s new wireless technologies have become more frequently used, which can be used to implement the modern idea of IoT[48]. Some of the technologies that make modern IoT device possible are:

- Smaller, cheaper and more powerful microchips
- The Worldwide Web
- Increase coverage of cheap and rather precise sensors
- Increased AI/machine-learning research and development
- Wireless communication protocols like:

- WiFi 802.11
- Sigfox
- Bluetooth
- LoRaWAN
- LTE/4G

The main strength of the IoT idea is the high impact it will have on several aspects of everyday-life and behaviour of potential users. From the point of view of a private user, the most obvious effects of the IoT introduction will be visible in both working and domestic fields. In this context, Home automation (domotics), assisted living, e-health, enhanced learning are only a few examples of possible application scenarios in which the new paradigm will play a leading role in the near future. [46]

3.1 Serverless architecture

Datacenters are the big motivator for improving how applications are built and developed. In the early days, every application ran on its own physical machine, which caused too many of the machines was underutilized, in addition to the huge cost in operation and new hardware. This lead to the development of hardware support for Virtualization(Intel's VT-x and AMD's AMD-V), which enables consolidation of multiple services onto single servers, which reduce the cost, improving manageability and underutilized. But hardware-visualization is not free, where the performance is reduced compare running on bare-metal hardware, in addition, it involves running a fully operational system in each virtual machine, which increases the memory usage. A light-weight alternative has arisen from these problems, the leading of these alternatives are containers. [49] Containers can be traced back to the Linux command *chroot* and FreeBSD's *jails*, the technology was developed further and LXC[1] become the de-facto standard for containerization. With containers, applications share the same host machines OS which results in that the size is reduced compared to if the same applications were running as a hypervised deployment.[50]. In figure 3.1 we showcase the differences between applications running on a hypervisor vs in a container.

1. https://linuxcontainers.org/lxc/

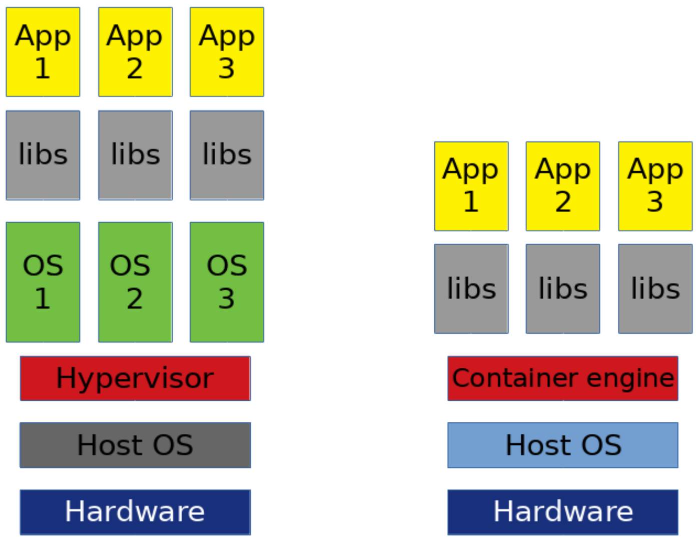

Figure 3.1: A illustration of applications running on a hypervisor and in containers

3.1.1 Docker

Docker is open source containerization technology for developers and system administrators to develop, ship, and run applications. "*We can have a quick idea about Docker containers by associating it with a container ship where its containers are loaded on to the ship and then shipped and unloaded to different locations.* " [51] Docker lets you quickly assemble applications from components and eliminates the friction that can come when shipping code. Docker provides the ability to package and run an application in a loosely isolated environment called a docker container. [52]

Docker builds on LXC with a kernel- and application-level API that together allows the processes to run in isolation. Docker uses namespaces to completely isolate the application's vire of the underlying operating environment, which include processes, network, users and file systems. [53] A Docker container is created by using base images, made by other contributors. These images can consist of the just the bare OS, or include a prebuilt application which is ready for deployment. Building new or modifying the images, forms a new layer above the old image, this could be done by installing new software or adding configuration to an image. The formation of new images can be done through manual input or by automatic scripts which are added to a so-called Dockerfile.

Docker swarm

To scale docker deployed applications we can use Docker swarm[2], which is a cluster management and orchestration and is embedded into the docker engine. A swarm consists of multiple docker hosts, each can be a manager (to manage membership and delegation) and workers (which run swarm services). The swarm manager uses ingress load balancing.

3.1.2 Microservices

Microservices are a prime example of this approach. Instead of designing and developing a single logical executable (*monolith*) server application, the application is separated into smaller independently applications. With the adaption of cloud hosting there has become evident that monolith application is lackluster, for the following reasons[54]:

- Lack of fault isolation - Large applications are often complex, and with smaller changes to the software stack, may cause the whole system to crash. Since monolith is built as a single large executable the crash of a single operation will affect the whole system.

- Hard to Scale - scaling large application is difficult and requires a lot of resources and time. In most cases, the only way to scale is to deploy multiple instances for a whole monolith application, even if there is only one part of the application are the reason for the need of scaling. For example, this could be that the user authorization is stressed by the number of requests, but since there is no way to extract this part of the application a whole new instance of the monolith needs to be deployed. This would require deploying additional instances, and set up load balancers to offload the stressed instance of the application.

- Deployments are hard - With rapid development the deploy service needs frequent updates and restarts.

- Technology lock-down - When developing a single monolith application, the development is locked into one single technology stack. Layers are often coupled together with function calls, and often developed with the same stack for the sake of interoperability. When the program grows, the monolith grows thus making the switch/experimentation with new technology difficult or time-consuming.

2. https://docs.docker.com/engine/swarm/key-concepts/

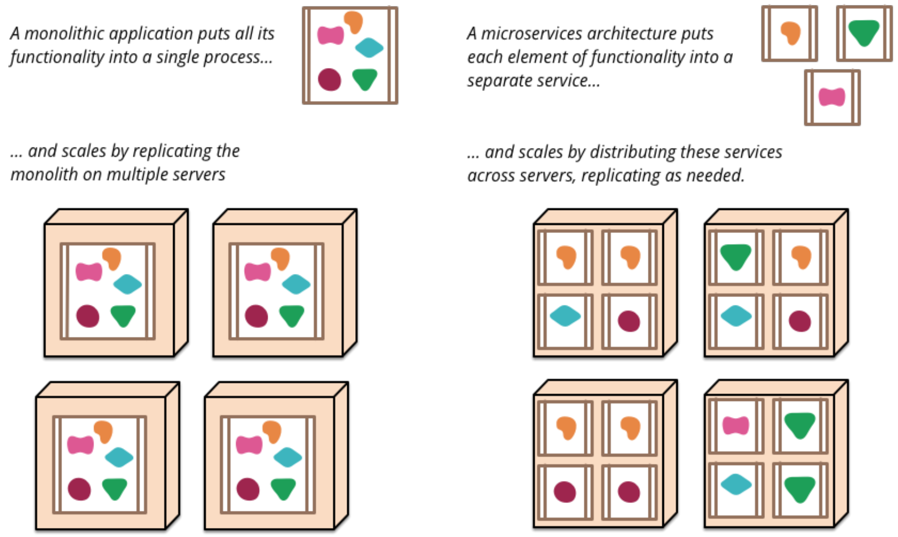

Figure 3.2: Monoliths and Microservices (source: [55])

This brings us to microservices, which aims to correct some of these problems. Most of the large-scale websites, including Netflix, Amazon, and Uber has changed from a monolithic architecture to a microservice architecture.[56] The basic concepts of microservices are [51]:

- **Small and focused** - Each service should be treated as an independent application
- **Loosely coupled** - Each service should not be dependent or with the need for coordination with other services
- **Language-neutral** Each service can be built on the technology most suited for the task, without adding complexity to the service as a whole.

Using a microservice approach is not without its downsides, it will induce some operation overhead, because of the increased number of deployments. It also requires the developers to be able to pay attention to distributed application development (e.g., independent data models, resilient communication between microservices, eventual consistency, and operational complexity).[54]

A common way to look at the concept of microservice is to view it in the light of the UNIX philosophy and the concept of DOTADIW, or "Do One Thing and Do It Well." [57].

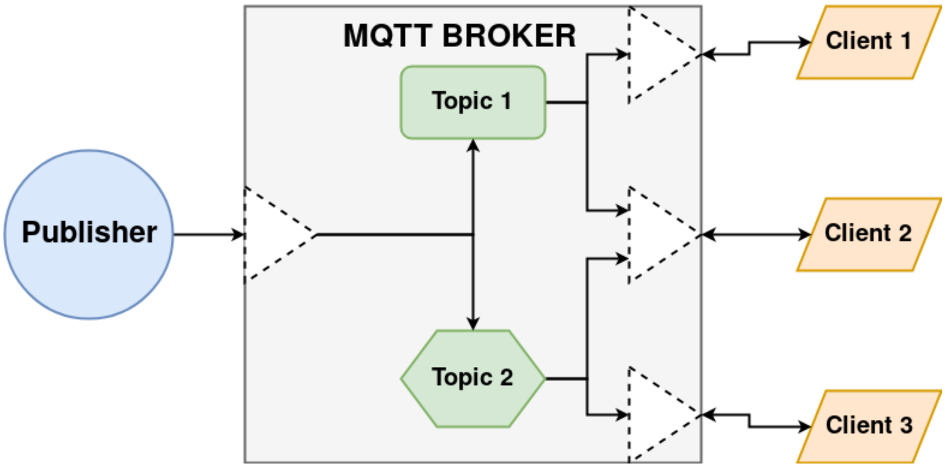

Figure 3.3: Sample MQTT Publish-Subscribe illustration

3.2 Push messages

Push messages can be seen as a permission-based mobile communications channel. Unlike normal HTTP traffic the client doesn't need to ask for a update, instead they usually register with the provider or server that they are interested in updates to a spesific topic or general message from other clients or a server. Once a new message is created, it's pushed out to the clients that are interested in that message. The messages are provider in near real-time and can even be buffered for devices that are not able to receive them.

3.2.1 Message Queuing Telemetry Transport

MQTT (Message Queuing Telemetry Transport) was originally developed by IBM for telemetric application, and has since been open sourced, as an Eclipse project[3]. It's a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport [58], allowing thousands of clients to connect to a single server. MQTT message are exchanged through a MQTT broker, and clients can publish or subscribe to spesific MQTT topics. A illustration of is shown in figure 3.3. These characteristics make it ideal for use in constrained environments or low-bandwidth networks with limited processing capabilities, small memory capacities, battery concerns and high latency. The Quality of Service (QoS) feature allows the protocol to provide traditional messaging qualities of service, you can think of it as a postage without confirmation of delivery and an courier service with ensured delivery. There are three different options for

3. https://mosquitto.org/

QOS[59]:

- **QoS 0 - At most once** The message is delivered according to the best effort of the underlying network. A response is not excepted, this offer the best performance but for clients that disconnect unexpectedly the message will not be cached and therefor not delivered to that client.
- **QoS 1 - At least once** The MQTT broker will try to deliver the message atleast once, therefor there might be duplicates delivered to the clients.
- **QoS 2 - Exactly once** The highest level of Quality of Service, Additional protocol features ensure that duplicate messages are not delivered to the receiving application. The message is delivered once and only once. Many IoT libraries not not support this level of Quality of Service.

Retain messages allows the MQTT broker to store messages, when newly connected subscribers receive the retain messages immediately when connected. [60]

3.2.2 Firebase Cloud Messages

Firebase Cloud Messages(FCM)[4] formally called Google Cloud Messages(GCM) is a part of Google Cloud Platform. It provide a push service to developers that can easily be integrated into applications, with support and examples for IOS, Android, web and servers. It can utilize two different methods of transfers, HTTP and XMPP, both providing upstream and downstream messages. Another important aspect of FCM is the topic feature, which is similar to the MQTT topics. There are some benefits in choosing FCM over other options like MQTT, 1) It's a service, so there is no need to setup a broker 2) Native support for Android and IOS, where the message can be delivered even if the application that uses FCM is not running.

4. https://firebase.google.com/docs/cloud-messaging/

4

Methods and Materials

4.1 Methods

For gathering information about what kind of features people with diabetes look for in an application that can enable the monitoring, warning, and motivation of people with diabetes. a feature interview was conducted. Additional through discussion with the supervisors, we found a few features that we think is beneficial to diabetics. For the motivational aspect, the main requirements came from the research the author was part of with several members of the Medical informatics Group from the Department of the Computer Science. The goal was to identify the major motivational factors that facilitate prolonged used of medical mobile applications (Mhealth), with focus on three different patient groups (Diabetes, Sickle cell, and physical activity). In total 16 participants were interviewed, 10 had sickle cell disease, 2 had type 1 diabetes and 4 had no chronic diseases. The lessons learned from the interviews and discussion with the group is utilized throughout the project and was of great importance when the requirements and features of the project were developed. The whole article is available in the appendix, due to the fact that it's not published yet (presented at the phealth-2018 conference [1]).

In addition, we also did a limited user-study with 3 people (1 without diabetes, and two with) at the end of the project duration, to get feedback from potential users of the system we present in the text.

1. https://www.ntnu.edu/phealth2018/program/

The development was approached with regards to the incremental build model. Since the requirements were laid out and the different elements of the system were already planned (not in details, more of an idea of how they should function). Many of the different modules are depended on each other, and therefore the baseline was first developed before the other modular was built. Each new modular was thoroughly tested by manual or automatic mechanisms, before going on to the next module. Of course, unexpected things happen, and you have to go back and change some of the previous modules.

4.1.1 Critique of Methods

Even with requirements and user-study interviews the users should have been more involved in the process, this could have worked out as workshops or focus groups throughout the project duration to get continues feedback on the features and design. The interview sample-size is also a concern, and might not give a good indication of the general population's opinion and thoughts about the usefulness of the project.

4.2 Materials

The experimental part of the project consists of developing a prototype which incorporates all the features set out the requirements. To do these different devices, each using their own SDKs and programming languages.

4.2.1 Emerging Technology

We use a number of technologies, protocols, and concepts in the project; therefore an explanation of the different components/protocols used in the project is warranted.

MongoDB

MongoDB is a schema-less document-oriented database (NoSQL)[2], released in 2009 as an open source under the GNU Affero General Public License and the Apache License. It's written in C++ and is intended to be scalable. The fundamental idea is that the *row* in traditional SQL databases are replaced by JSON-like *documents*, with no enforced schema. The documents can contain

2. https://www.mongodb.com/

any data, like lists or string, number, and even embedded documents, where the fields may vary from document to document. It supplies Ad-hoc queries, indexing, and real-time aggregation, with added MapReduce functionalities. [61]

MongoDB provides high performance, high operability, high availability and easy scalability. [62]

4.2.2 NightScout

[63] (also called CGM in the Cloud) is an open source, DIY (do it yourself) initiative which enables real-time monitoring with selected Continuous Glucose meters (CGM). The data from the CGM is uploaded to a self-hosted server and can be accessed through a REST API. The project was started by tech-savvy parents of children with type 1 diabetes who were tired of waiting for a solution from the manufacturers of CGM devices, and thus started reverse-engineering, and implementing the NightScout core-services. [64]

4.2.3 Dexcom API

In 2017 the company Dexcom[3], which is one of the leading companies supplying Continuous glucose meters, announced that they were to allow some patient data to be available to third parties via an API. The API is secured using Oauth2 and provide the access to the users' BG levels, events (like exercise and food intake) which are logged using the Dexcom mobile application and CGM device.

4.2.4 IoT Device

Device

The IoT device utilizing a ESP8266[65] (shown in figure 4.1) is a WIFI-enabled microcontroller developed by Espressif inc. The microcontroller has a Tensilica Xtensa Diamond Standard 106Micro based microprocessor running at a frequency between 40-160MHz (depending on device). It also includes 80 KiB user data RAM, 16 GPIO pins (supporting I^2C, UART, and SPI) and 514 KiB to 16 MiB external flash which are used for the compiled binary and could be used for storage of small files used by the programming running on the device.

3. https://www.dexcom.com

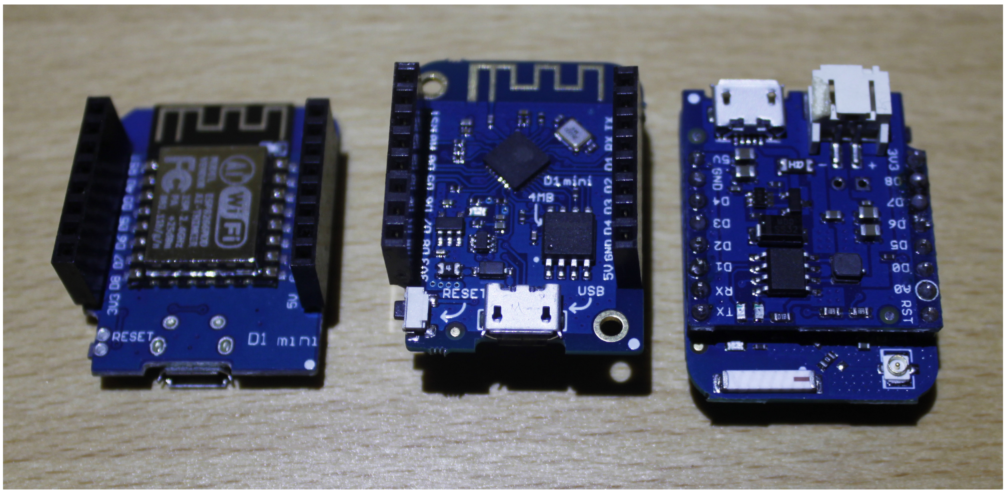

Figure 4.1: Three different ESP8266 devices from Wemos.

Connected sensors and display

Connected to the microcontroller is lithium ion battery cell, battery charge/control circuit and an Electronic Paper Display (EPD) to display the Blood Glucose measurements and general messages.

Electronic Paper Display (EPD) or sometimes also called E-ink, smart paper or e-paper is an electronic ink display technology. EPD is designed to be thin, lightweight, power-efficient and has the same appearance as normal ink on paper[66]. The display is built up of many small capsules with two or three types of plastic spheres inside, a clear liquid and two or more transparent electrodes. The spheres are electrically charged and are attracted or repel, and by applying a voltage to one of the spheres the polarity is changed and the capsule (or pixel) changes color. [67]

EPD does not require power to maintain an image, it's only when the polarity needs to be changed (to change picture) power is required. Unlike most display technologies, EPD screens do not require any power to maintain a static image, which means that they only use power on updates.[68]

The total cost

The cost for all the components are around 30 USD,

- Wemos D1 Mini - 3-5$

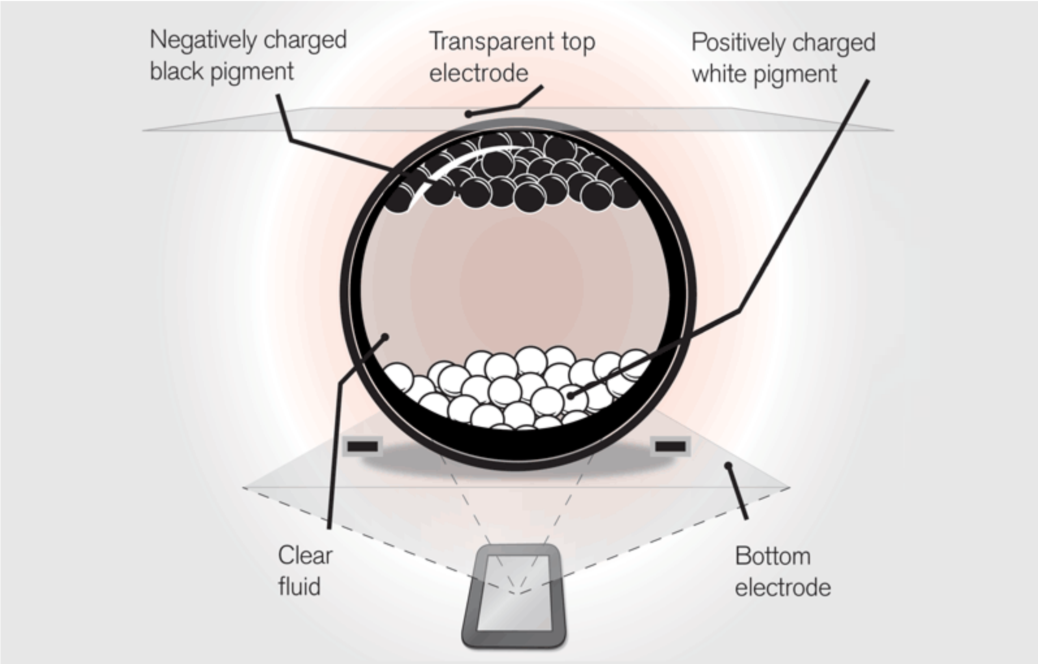

Figure 4.2: Capsules filled with negatively and positively charged plastic spheres colour the surface of an Electronic Paper Display when an electric charge is applied (source: http://www.eink.com/)

- Waveshare 2.9inch E-ink Display - 17-20$
- Battery shield for Wemos D1 mini - 2$
- 1000mAh Li-ion battery - Around 6$

Note that this is the normal prices, it's possible that if one would buy a larger number of devices, one could get a rather good discount from the sellers. One could also skip the battery shield and battery and run it of a power bank or from a USB-charger, which would reduce the price. Another possibility is to buy a different sized display, WaveShare has variants ranging from 1.5inch to 7inches and in the last few months they have added displays with three colors (red, black and white) or grey-scales for the more expensive models. Not that this is a development device, a production model would probably have a custom PCB with the ESP WROOM chip (MCU package) and some kind of battery charge and protection chip only, and the price would be decreased even further.

Software development kit (SDK)

There is a number of different SDKs that could be utilized for programming the ESP8266.

- Espressif's official SDK - C based on the GCC toolchain.
- NodeMCU - Lua-based.
- Arduino - C++ based.
- PlatformIO - cross-platform IDE and unified debugger which sits on top of other SDKs and uses their libraries.
- ESP-Open-SDK - Free and open (as much as possible) integrated SDK for ESP8266.
- ESP-Open-RTOS - Open source FreeRTOS-based ESP8266 software framework.

Since the author had some prior experience in the Arduino SDK the IoT is developed with this SDK together with PlatformIO. There the added benefit of the Arduino SDK, that there is a number of third-party add-ons/libraries, which can be utilized instead of writing it from scratch.

4.2.5 Smartphone Application

Since the IoT device can't interact with the content, e.g. make new messages, groups, replies, change settings etc. there needs to be an application to let users do this. To achieve this there is two viable solution that can reach most of the users, a website or mobile application.

There are two major SDKs for mobile application, namely IOS and Android, which totally dominate the mobile marked with 12.1% and 87.7% respectively[4]. A concept application was developed using the Android SDK, but since it would be beneficial to support both major frameworks, we switch halfway through the project to NativeScript with Angular.

4. https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

NativeScript

NativeScript is an open-source mobile framework made available under the Apache 2 license. It's actively developed and supported by Telerik[5]. The Framework enables building native applications for both platforms from a single code-base. The codebase can be written using TypeScript(TS) or Modern (ES6) JavaScript(JS), and one can choose to use web-frameworks like Angular or Vue.js. NativeScript will compile the TS/JS down to code specific to each platform with its native rendering engine and elements. The result is an application with native-like performance and user experience.

The core concept of NativeScript is very similar to progressive web-frameworks, the application consists of multiple pages, where each page represents a separate application screen (activities in the Android SDK). The logic of the page is contained in a TS or JS file, while the layout is often separated into another HTML or XML file, with additional styling defined in CSS (or SCSS) files.

Nativescript is powered by node.js, and the package manager NPM is available (in addition to the built-in package manager for NativeScript, TNS), making integrating and managing third-party modules and libraries easy.

Angular Angular is an open-source web framework developed by Google, for use in web or mobile application[6]. It's licensed under the MIT license and was released in 2010, and currently is one of the most influential and used frameworks with big sites like The Guardian, Gmail, PayPal as users.

4.2.6 Server

The Server is written in Go, a relatively new programming from Google. it's a compiled, statically typed language in the tradition of C and similar languages, with a garbage collection, limited structural typing, memory safety features and with CSP-style concurrent programming features. it was first release in 2009 and was created by Robert Griesemer, Rob Pike, and Ken Thompson. The current version is 1.10.2.

To handle the RESTFUL HTTP API traffic the minimalistic open-source web framework Gin is utilized, which features a martini-like API and good performance due to the usage of httprouter (a faster implementation of HTTP handling than the one found in the golang standard library).

5. https://www.telerik.com/
6. https://angularjs.org/

4.2.7 Physical Activity

Under the project duration, the author was involved in a study to examine the availability of wrist-worn fitness wearables and analyze the availability of relevant fitness sensors from 2011 to 2017[69]. From the literature search, we found the Xiaomi Mi Band, which has been shown to be the best package compared to its price in a study comparing 17 devices with regards to steps and heart-rate measurements [70]. The Mi Band is a small cheap fitness-band, by the Chinese company Xiaomi and comes in four versions which all have IP67 certification and an activity sensor, and can detect sleep, steps and raw activity-levels:

1. Mi Band - original with colored leds - discontinued
2. Mi Band 1a - white leds, no HR sensor - discontinued
3. Mi Band 1s - white leds, HR sensor - 14$
4. Mi Band 2 - with display, button, HR sensor - 20$

Figure 4.3: The Xiaomi Mi Band 2, used for tracking the activity for users (source: digitaltrends)

The device used in the project is the Mi Band 2, because of the small OLED screen, which shows the time, the number of steps, distance, heart-rate, and calories burned. Throughout the testing, it has shown to have a long battery life of around 3 weeks with heart-rate measurements every 30 min and continuously throughout sleep. Xiaomi has its own application called Mi Fit[7] which is standard application that can be linked to the device, this will upload the data to the cloud, which might be a concern for some users. The application support exporting data to Google Fit, which makes it possible for other application to utilize the data collected from the fitness-tracker.

GadgetBridge

GadgetBridge[8] is a mobile application that replaces the vendor's closed source application, without the need to create an account and transmit any of your data to the cloud or through a network. It supports a number of smartwatches and fitness-bands, like the Pebble, Xiaomi Mi Band, AmazFit, and HPlus. The application supplies mostly the same features as the closed source alternative, like vibration on notification, alarms, step goals, inactivity warnings and device settings (like heart-rate intervals). The most important part of a smartwatch/fitness-tracker is the physical data which are connected by the sensors in the device itself, Gadgetbridge fetches the activity data from the device and show it in graphs (see fig. 4.3), the data can then be manually or automatically exported into an SQLite3 database for other application to utilize. The application is only available for Android, and through F-Droid[9] (a Free and Open Source(FOSS) Software repository, similar to the Google Playstore/IOS AppStore). GadgetBridge[10] is a mobile application that replaces the vendor's closed source application, without the need to create an account and transmit any of your data to the cloud or through a network. It support a number of smartwatches and fitness-bands, like the Pebble, Xiaomi Mi Band, AmazFit and HPlus. The application supplies mostly the same features as the closed source alternative, like vibration on notification, alarms, step goals, inactivity warnings and device settings (like heart-rate intervals). The most important part of a smartwatch/fitnesstracker is the physical data which are connected by the sensors in the device itself, Gadgetbridge fetches the activity data from the device and show it in graphs (see fig.), the data can then be manually or automatically exported into a SQLite3 database for other application to utilize. The application is only available for Android, and through F-Droid[11] (a Free and Open Source(FOSS) Software repository, similar to the Google Playstore/IOS AppStore).

7. https://play.google.com/store/apps/details?id=com.xiaomi.hm.health&hl=en_US
8. https://github.com/Freeyourgadget/Gadgetbridge/
9. https://f-droid.org/
10. https://github.com/Freeyourgadget/Gadgetbridge/
11. https://f-droid.org/

5

Requirements

To figure out what kind of features users want in an application that has it focuses on warning/alarm and motivation 2 diabetics was interviewed, in addition to the research that was conducted for the Mhealth motivational paper. The following is a description of the features combined from both of this.

Small private group Many of the Mhealth interviewees states that didn't enjoy having a large social circle around them when discussing personal disease subjects, instead they discussed in person, or in small groups online. Based on this input the decision was made isolated communication into small groups, where 2-6 person could talk about their problems, motivate and watch out for each other.

Internal Group Competitions Some people like to compete, while others hate it. But the thought is that if the groups are small and the participants do know and trust each other, it might be more attractive for more of the participants, like an internal not so serious competition or challenge. There was a wish for an opt-in functionality so that those who want to compete can do so, while the other not need to. Since physical activity is important for the self-management of diabetes we choose to focus on this, there was also a suggestion to compete with regards to HbA_1c, but some of the interviewees state that this could be demoralizing for those who not have the ability to have a strict diabetics control.

A platform for discussing their problems privately/securely with liked minded people Like said in the first feature, the main communication channel is contained inside groups. This gives the members a safe place to discuss their condition and ask for advice or tips regarding it, and get answers from other people with the same condition or the knowledge about how to manage. This is one of the main features that the interviewees mentioned. So, the system needs to have some kind of way to communicate between the members in text form, with replies to that message.

Achievements/Badges Achievements/badges are rewarded to users based on their actions, like creating x posts, connecting to a CGM and physical activity. These concepts come from gamification and can be motivational for the users if executed right.

Warning on high and low blood glucose levels The most important factor from the feature interviews was the warning on both high and low BGL, this would make the person more secure that if something would happen, another person would be able to know and do the appropriate action. Another key-feature was that the alarm should have an ability to be canceled by the diabetics him-/herself and that they should be the first to get a warning before it was sent to the rest of the group members.

Integration to exciting applications A concern for people with diabetes had was that this application should occupy them too much, and there was just another application that would require them to add details about their daily life. Therefore the application has no way to act as a diabetes diary or a substitute for one, the data that is used for diabetes comes from other sources, so this is a platform for warning others and discussing their condition in a safe and fun environment.

5.1 Scenario

The following scenario presents an example of how a group of people with type 1 diabetes may support and watch out for each other using our new Internet of Things Mini Display-based Motivation, Notification and Warning System.

Paul, Miriam, and Clark are three friends which all have diabetes, a CGM, a smart band and live in different locations. They have agreed to watch over each other's blood glucose levels, try to motivate each other, and decided to try this the application for this purpose. Paul has ordered six sets of microcontroller, screen and batteries, two of these have been sent to both Miriam and Clark with instruction on how to wire them up properly. Paul has one IoT device on his desk in the bedroom and one in the office, Miriam has one in the kitchen and one at the break room of her job and Clark has one in the bedroom and one in the living room.

They all make an account, and connect their NightScout server, added contact persons and configured their IoT device through their smartphone. Paul made a group called *The Golden Three*, which Miriam and Clark applied to join and Paul accepted their request, and now they are sharing their CGM data with each other. We will follow these three throughout a day, to see what kind of features and possibilities the application has and what benefits it could give to people with diabetes.

07:00AM Paul wakes up, turns on the light and checks the IoT device for new messages and the groups BGL levels, there are no new messages and the BGL looks fine for the rest of the group, but his own is high. He has been experiencing this often in the last few weeks, his doctor told him it was *The Dawn Phenomenon*, which basically is a natural rise in blood sugar in the morning and effect some diabetics, he takes a shot of fast-acting insulin to get it down.

It seems like Miriam was the most active person yesterday with 12 000 steps, Paul is concern that he might lose the weekly competition so he finds his smartphone and opens the mobile application. It seems like he still got a 2000 step lead over Miriam, it's rather close so he needs to plan for a hike later today. While he has the application open he creates a message, which gratulate Miriam with yesterday's victory, while wishing them all a good day. He goes to the bathroom, checks his BGL (it's fine now) and start making himself breakfast before going to work he adds the carbs into his NightScout application.

07:30AM Miriam alarm rings, still half-awake she stumbles to the kitchen and sets on the kettle before going to the bathroom. After a hot shower, she comes back to the kitchen, makes a sandwich and coffee before sitting down, adding the breakfast to her app, it's fast because this is something he always has in the morning and is saved in the application. The IoT-device in front of her shows the message that Paul sent half an hour before, and she also sees that his blood sugar is climbing which is a good sign, he ate before going to work.

09:23AM Clark's blood sugar level has been declined all night, which is normal. But Clark is a late sleeper and has had multiple nocturnal hypoglycemia events in the past and now his BGL is reaching a dangerous level. Paul has just arrived at work and is sitting in his office when he notices this and sends a message asking if Clark is awake.

09:37AM Clark is still asleep and hasn't noticed the message Paul sent him, so Paul calls Clark and ask him to wake up and check his BGL (which is updated every 5th min, by the server). Clark answers the phone and checks the IoT-device which is situated right at his bed, the BGL reading is 3.98 mmol/l, he gets out of bed and into the kitchen, where he gets a glass of orange juice to hopefully get him to an acceptable level. It's his day of work today, so he will take it easy, maybe go for a walk later but first, he will watch some tv-series.

10:00AM Clark's BGL has risen to 4.5mmol/l which is good, he replies to Paul message, thanking him for the call. He walks out of his home and starts his daily commute to his work in the city. Miriam has been watching Clark's blood sugar rise on her smartphone and is pleased, it's really good that Paul detected it before he got a hypo again she thinks to herself.

11:00AM Before Paul takes an early lunch, he looks over to the IoT-device, Clark's BGL has stabilized around 7.2mmol/l, and Miriam is not very high but nothing to worry about yet. He walks down to the cafeteria next door to his office, today they have spicy Mexican soup special, he orders but when he asks for the carbs the clerk just shrugs her shoulders. "I guess I'll have to guess," he thinks to himself, "it's not optimal but it smells so nice". After he's done with the soup he starts a calorie counting application on his phone, and types in soup, *1dl of soup = 10carbs*, but that's not Mexican soup. Paul takes 1.5x the insulin to make up for the difference in soup carbs.

11:25AM Paul's BGL is declining in a fast tempo, he took too much insulin. He's not aware since after lunch he had to attend a meeting. Clark notices that Paul's BGL is dropping fast and that he's in a meeting, calling him now would not be the right thing to do. He tried to send him a message via the mobile application, hopefully, he will notice it in time to avoid hypoglycemia.

11:32AM Paul did not notice the message, and thus his BGL has triggered a warning from the server. It sends him a notification, which makes a sound. He checks it and sends a feedback to the server so that the other not get a notification too, before excusing himself from the meeting, and heads to the bathroom for a small injection of insulin. He checks the mobile application, and sees that Clark warn him about it, replies a thank you back before heading to meeting again.

12:00PM Both Miriam and Clark have lunch around the same time. Miriam brought pre-packed lunch from home, so she knows the number of carbs and the insulin she has to take, Clark is at home so she makes a few sandwiches. Clark checks up on Paul while eating, and notices that his BGL is stabilizing.

2:00PM Paul's BGL is still fine when he gets off work early, he had planned to hike today to still ensure that he was in the lead. So, he walks home from work, he still has the near miss with hypo in mind so he grabs a chocolate bar from a supermarket on his way out from the office. Paul, on the other hand, has not been very active today, which is not normal for him, he eats the same as he normally would do and his BGL is rising.

3:00PM Paul just got in the door and checks his smart band for the number of steps he has achieved today; 15 000 steps. "That's pretty good, bet that Miriam won't be able to beat me today," he thinks to himself. Before checking his BGL, it's a bit low but he just ate that chocolate bar so it should be ok, he just has to keep an eye on it. He notices that Paul has not been active today and that his BGL has risen and is close to a hyper event, he sends him a message trying to tell him that he maybe should try to walk for an hour or something, adding that watching TV all day isn't the best way to spend his day out of work.

4:00PM Miriam isn't that competition driven, she likes to win some days but it's mostly a bonus added, she thinks that maybe Paul takes it to far sometimes. But as long as he has control over his BGL it doesn't concern her. Her day is almost over, but her work has a social event today, this often involves food and drinks; it's not that she doesn't enjoy it, but it makes it harder to control her BGL. She sends a message to Paul and Clark, telling them that she's going, and ask them if they could have an eye on her, it makes her less anxious just knowing that others can keep an eye while she enjoys other things, and try not to focus so much on her diabetes all the time.

5:00PM Paul hasn't seen the message, and his blood sugar is spiking he now has a mild hyperglycemia, his smartphone beeps and tells him that he might want to do something about it. He gets up from the coach and checks his BGL, and it's over the limit. He takes a small dose of insulin, and since he has been inactive all day he might as well go for walk. So, he gets his snickers and goes out, he might as well work on his new achievement, it's only 10 000 steps away. Miriam has just got the food she ordered, the server informed her about the carbs content so she thinks it will be ok, just have to add the wine she's having with the food. Paul, on the other hand, is making dinner for his family. Today its spaghetti Carbonara and he carefully makes a note of everything and adds it to his Mobile application so he can take the appropriate amount of insulin after the meal.

6:20PM Paul comes back from his walk, he went to a friend for a coffee, his smart band tells him that he walked around 6km, or 5500steps, that's pretty good. He might have a new achievement tomorrow if he not has a busy day at work. His BGL has been lowered, but that's normal after light physical activity and the fact that he took insulin before going out. Now he needs some dinner, "I think I'll just have a frozen pizza", he thinks to himself and turns on the stove.

8:00PM Miriam has had a few drinks by now, but now the social event is done. Before the Uber comes to pick her up, she goes to the bathroom and injects insulin. She has kept count on the number of drinks, so it should be fine. Clark has kept an eye on her BGL, and it seems to be going fine, so he makes comments on her original message that she's doing great, and hope she had a great time.

10:00PM The day is coming to an end, Paul has been enjoying himself reading a book, Clark, on the other hand, is touching upon some notes for tomorrows work, while Miriam is relaxing in front of the TV. They all receive their daily personalized feedback from the system at the same time. Paul's feedback states that he should watch and maybe make notes on what he's eating so he can a take a more appropriate amount of insulin, other than that everything is good. He's been thinking about that and asks the other two about what he should do when the cafeteria does not know the carbs content of a dish. Miriam replies that he should bring lunch from home, so he knows what's in it. Clark, on the other hand, says that he experienced the same thing, and stopped going to the specific cafeteria and tried another one and that has worked for him. Miriam feedback is a motivation for her, she did well today and she should keep it up, maybe try to more physically active, but she's doing really good. Clark knows that he has been inactive today, so he's feedback states that he should move around a bit more, or at least watch his BGL levels when he's inactive, but it was really good that he took a walk right after he realized the high BGL.

11:30PM Paul is going to bed, and send a goodnight message to other two, telling them to keep up the good work. Miriam and Clark reply, telling him goodnight in return and compliments him on the physical activity and the work he's doing.

So, this was a day in three diabetics life, it might be simplified but highlights the use cases that the system is built for. There is not serious hypoglycemia, where one person does not answer the alarm, but that is detailed in the design/implementation part.

6

Related Work

There isn't any application that tried achieve the same as this project, to the best of our knowledge, that being sharing BGL values, in a social media setting with messages and groups. We did a extensive literature and general search and found somewhat similar applications that are relevant to this project.

6.1 Hypoband

Hypoband[1] is a smart band on that is placed on the wrist which monitors the skin's perspiration and temperature which can detect hypoglycemia events. If a hypoglycemic event is detected an alarm goes off, and if the person needs medical attention there is a panic button which has the ability to send an SMS/call/alarm to caregivers. The price for the Hyperband is £74.9 with additional subscriptions fees.

1. http://hypoband.com/i

6.2 GlucoNightWatch

The GlucoNightWatch [2] is rather interesting because it uses the FreeStyle Libre CGM sensor, for measuring the BG values. It's formed as a sports bracelet which is wrapper around the sensors transferring data over NFC and then the data is transferred to a mobile phone via Bluetooth and allows for remote alarms via SMS or Call to parents or a care person with customizable thresholds. Currently, the product is only a prototype and is crowd-funded on Kickstarter, and the proposed price is set to 219$ after the campaign ends.

6.3 Alert

Alert[3] is an application that allows for users to call for help by sending notifications to three emergency contacts, with location and even allows for a conference call between you and the contacts. The application support Apple's Healthkit and can access blood glucose reading captured by any other applications, and notify in case of out-of-range results. The application is only available for Apple IOS, and has a premium offering which involves unlimited voice calls (free version only allow for three free calls), and has a price of $99.99.

6.4 Dexcom Share

If the patient is using the Dexcom G5 mobile application there is a function where the user can opt-in to share their BGL values in near real-time with up to 5 persons using another application called Dexcom Follower. [4].

6.5 Guardian Connect

Medtronic's Guardian Connect[5] is mobile application that connects the a CGM and/or insulin pump. It's only available for IOS, but an Android application is in the works according to Medtronic. The data from the CGM device is uploaded to their CareLink®cloud solution, where up to 5 *care persons* can view the data.

2. https://gluconightwatch.com
3. https://helparound.co/alert/
4. https://www.dexcom.com/dexcom-share-0
5. https://www.medtronic-diabetes.com.au/products/guardian-connect

In addition, SMS can be sent out to these persons in case of an alarm.[6]

6.6 Eversense Now

Eversense[7] Now is newly launched application, approved in Europe, the Middle East, and Africa. It allows users of a CGM to select up to five friends or family members to view their glucose data, which are updated every 5th min. The feature is only available for users of a Eversense CGM sensor and IOS.

These examples are all propitiatory solutions, the first two requires extra hardware and is subscription based, the others require the user to be locked into a specific CGM device. NightScout, on the other hand, is open-sourced, in active development, support a number of CGM's from different manufacturer and is used by thousands of people around the world; therefore it's natural to implement a social alarm system on top of the already proven project. The cost of the different applications is also a concern for low-income patients, our system only requires a NightScout compatible CGM sensor, smart-phone, and a server. The server can easily be deployed to the cloud, thanks to Docker or even ran on a single board computer like a Raspberry Pi. None of these application/devices has the feature to show the create groups for discussing the condition or checking up on each other's BG levels outside of alarms. In addition to this, these application does not have a social network feature, which could be used for more lightweight communication, questions, motivational comments and other motivational factors like badges and small competitions.

6. https://www.medtronic-diabetes.com.au/support/frequently-asked-questions/guardian-connect
7. https://eversensediabetes.com/products/eversense-mobile-app/

7

Design and Implementation

To fulfill the requirements set out, a system was designed and implemented. It consists of three different parts. In the middle is a server which serves the frontend devices (IoT and mobile applications). The IoT device receives updates through MQTT, while the mobile application utilizes the RestAPI and FCM for notification purposes, a simplified illustration of the system is shown in figure 7.1.

7.1 Backend server

The server's task is to supply the frontend devices with communication, store the data and fetch Blood Glucose Levels from the different sources (That being NightScout or Dexcom). It uses the microservices architecture, which involves the separation of endpoints to individual docker containers, meaning that they are isolated and independent of each other. To manage external HTTP traffic, Nginx is used as a reverse proxy to redirect the calls from the users to the correct container, additionally, Nginx is responsible for encrypting the traffic with certificates supplied from Let's Encrypt.

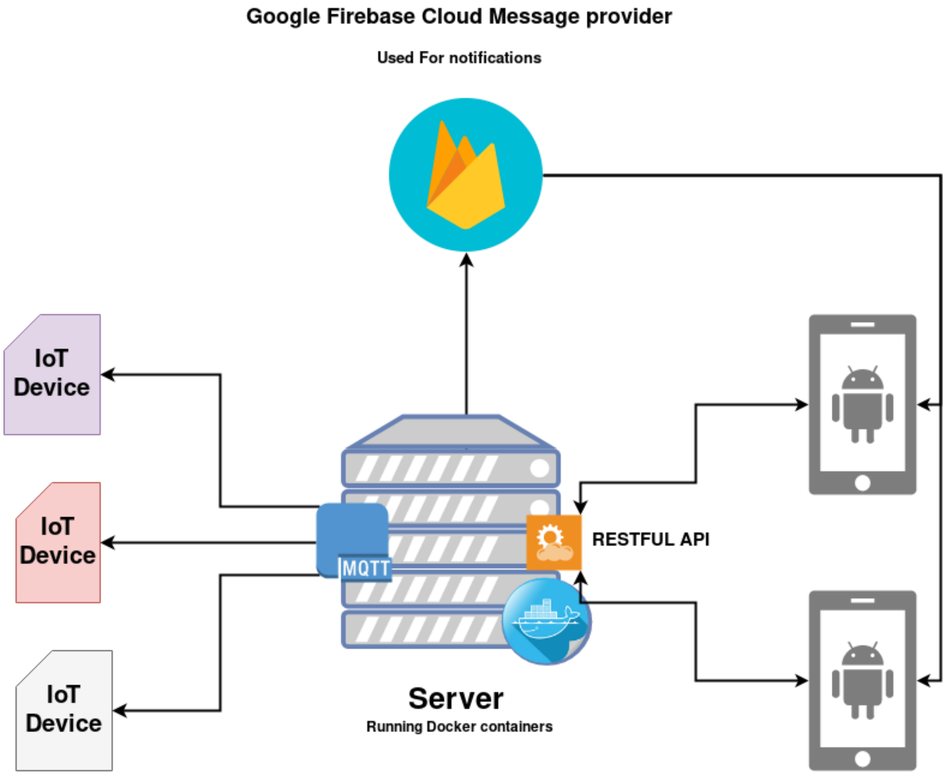

Figure 7.1: A simplified overview of the components and communication.

7.1.1 RESTAPI

The web server offers a REST API for clients to utilize, the endpoints can be found in the Swagger document located at the root of the server source code. Most of the endpoints are for basic CRUD (Create, read, update and delete) operations, but there are a few out outlier for simplicity of supplying data to the clients. For authentication, we use JWT (JSON Web Tokens) tokens, which are added to the request headers and identify the user, which are the same token used in Oauth2. The token has a limited lifetime, once the token has expired a new token has to be requested.

7.1.2 MQTT

The IoT devices aren't always able to receive new updated which are either sent into the server from the mobile application (messages) or BG measurements fetched from the server, this might be because the device is not turn on or is in a power-saving mode. Therefore, a cached snapshot of the messages for each topic needs to be stored in the retain-message of the MQTT server, which is supplied by the servers MQTT container on startup, this message contains the latest 10 messages for each group, which are updated when new messages are created. In addition to the retain-message, there is also a live channel, where single, new messages are sent to the connected clients. For blood glucose measurements messages, we only use the live channel, this is due to the size of the messages from all the group members will most likely exceed the maximum supported parse size, the IoT device can handle.

For authentication and authorization, we use the Mosquitto Auth Plugin with a MongoDB preference. This enables the MQTT server to independently verify the login credentials and the channels the users have access to.

7.1.3 Database

The central part of any web-server is the database, we use MongoDB which is a single application in a separate container. A more reliable option is to use a third-party cloud-provider or even multiple containers on separate computers. This would enable sharding(horizontal partitioning), and redundancy, but for our case, it's enough with a single container. The models are defined server-side since MongoDB documents are schema-less. Following is a short describes the models used:

User

The user model is the primary model and describes the user and her/his settings. The model contains the hashed password (using PBKDF2[71]), their username, avatar (user image), statistics, Firebase Cloud Message(FCM) token and embedded structures which are user settings.

Token embedded structure contains the JWT token, and it's expires time. This is done because of the fact that the authorization method does not allow for distributed authentication without HTTP calls. So instead of calling the auth container for each call, we store it in the database and fetch it from the container which is responsible for the request, saving time and resources.

NightScout links the user to his/her own NightScout server and contains a URL and the groups which are allowed to use the information. When a user adds a NightScout config through the REST API, the URL is verified by an HTTP call. If it succeeds in this HTTP fetch, and the returned body matches the NightScout API-specification, the server returns an HTTP Accepted status back, but it can't verify it will return an HTTP not valid instead. This is done to

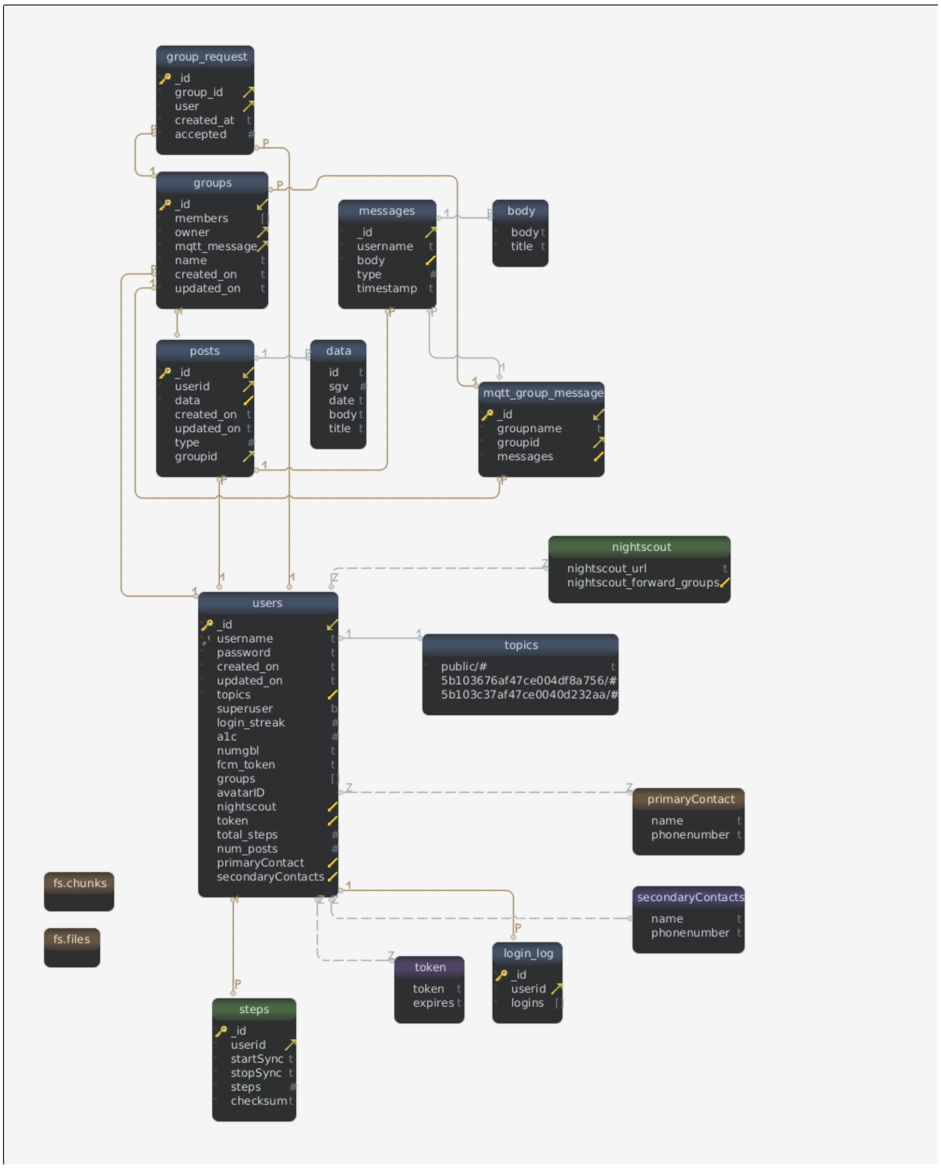

Figure 7.2: The database visualized by the the application dbschema

make the end-user (via the mobile application aware if their input was correct or not).

Topics Used for the ACL authorization for the MQTT server, it contains a list of group ids and the allowed action that a user can do (read and/or write).

Steps To record the number of steps of each user on a daily interval, we have a separate document which contains the number of steps, the start

Group

The group model is the structure that facilitates the group functionality, it contains the group's name and its members. It's main functionality to link the members to the group and the messages (posts, see next section).

Post

The post model is the basic message model, it holds metadata which includes the group-id, the owner of the post, replies, and timestamps. The data of the model is universal, meaning that the data can contain any structure, and is decoded down the actual data on the server/frontend side, to help with the decoding a type variable is also included. As of now, we support three types:

- Message - basic text message, containing a Body and Title.
- Blood Glucose Message - contains an array of BGL measurements, with value, id and a timestamp
- Appointment - contains an appointment fetched from Outlook, message and a message.

For text messages we also have the functionality to add replies by the other group members, these replies are implemented as an array of embedded structures in the post model.

7.1.4 Components

Since we're using the microservices architecture, we have multiple Docker containers, which combined makes up the backend-services. Each of the containers can be built and started by a docker-compose script, which can be found in the root of the source-code. Docker-compose makes it possible to have links to other containers (so for example, the API-handler get a link to the database) and dependencies (wait for the database and Nginx before starting the API-handler containers). All the API-handling containers follow the same guidelines or code sequence, where the API endpoints are defined in the main code, and specific model-based operation are imported from the model-package (which is included in the source code). This makes the management, testing, and cleanliness of the code better. The following is a short summary of all the containers:

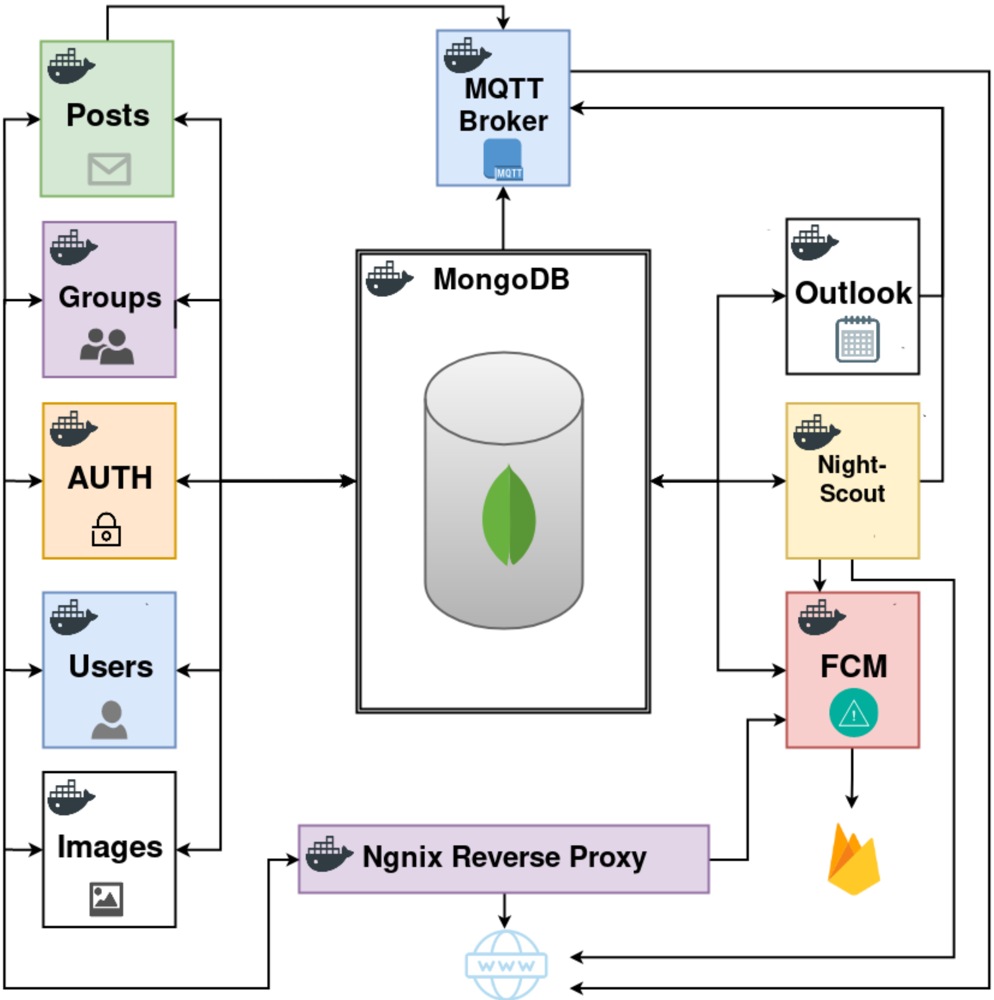

Figure 7.3: Overview of the internal containers composing the server

Nginx

The Nginx container contains the reverse-proxy, it handles the incoming traffic (to port 443) and redirects it to the correct handler (based on the URL).

It wraps/unwraps the HTTPS communication, this means that inter-communication is un-encrypted. This is not a concern as long as the communication is limited to a single computer/WAN network (protected by a firewall), but would have to be changed if the containers would be distributed to multiple computers another approach would have to be considered, instead of the current implementation.

MongoDB

Contains the database, and is linked to all containers (except for nginx).

User-API

The user-API container manage the user associated endpoints, that begin the signup, settings and so on, and is linked to the Database and Nginx container for data and communication purposes.

Group-API

The group-API container handles the group-based API endpoints, this includes the creation of a new group, the invitation and acceptation/ rejection of new members, and data which is associated with the group (get posts and steps from the group). When a group has been created, the group is visual for other users through the search endpoint, users which has a wish to join that group sends in a request to join, which the owner of the group can approve or disapprove. Once the user has joined a group, he/her will get access to the group information and the associated data, messages, steps and so on.

Post-API

This container contains the endpoints to create, update and delete messages and replies. A post (message), is always tied to a specific group and only members can view and comment on that post. A message part of the post contains two fields - title, body, and has a required number of characters. When a message is created and the requirements are met, it's stored in the database appended to retain message and sent to the live channel to devices that already online. Users can reply to messages, these replies are embedded into the message structure.

Image-API

The user can store an avatar (profile picture) to his account and a group which they are the owner of. Images are resized into thumbnails by the endpoints in the user and group handler and stored in MongoDB's GridFS in two versions, the original size, and the downsized thumbnail. The Image Container has endpoints for the application to fetch the images based on the ID, which is stored in the User or group model.

MQTT-Server

For communication with the IoT-devices, we use an MQTT-server which is located in the container, to allow for authentication and authorization we use the Mosquitto-auth plugin. Since there is no docker-container which contains this, we had to build it based on a plain Mosquitto image, and supply a script which is run when the container is built. The script will download the auth-plugin and compile it, change the configuration to enable the direct authorization and authentication to MongoDB.

NightScout-Fetcher

When a user has added a NightScout endpoint via the mobile application, a method of fetching the data has to be implemented. When a user has added and the server has verified the NightScout endpoint the URL will be added to the users' model. The URL is used by the NightScout-fetcher container, which periodically will call the URL and extract the events which have happened after the last sync (last time the events were gathered). The results are appended to the data of a post which are flagged as a post containing BGL measurements for this user. The event is checked and compared to thresholds which can be changed in the configure file for the server if the event's value exceeds or are beneath the corresponding threshold, the event is forwarded to the FCM-notification container, which handles the notifications to the user and group members. In addition to the notification, HbA_1c level is calculated and the event is published to the appropriated groups via the MQTT protocol.

FCM notification

To enable alarm and notifications for the end user the different options had to be explored since we carry around a smartphone at almost all the time this option was chosen. Notification to smartphones are rather restrictive since the application most of the time is not running, killed by the operating system or even not open by the end-user. Therefore we can't use MQTT like we utilize for the IoT-devices, and use a form of push-notification which are supported by the mobile operation-systems, the most known and utilized is Firebase Cloud Messages (FCM), but there is a few alternatives like Pushy[1]. This ensures that the end-user will receive the notification in a timely manner, as long the mobile phone is operational and has network connectivity.

The goal is to enable notifications for hypo and hyper events, where the BGL val-

1. https://pushy.me/

ues are supplied by the external source (NightScout) For the two events, there is a distinct difference in the procedure of notifying the user. For Hypoglycemia, the alarm is first sent to the source of the event (user with the low-blood sugar) and this is user gets a period to answer, by clicking on the notification. The notification data contains the event and a random integer which needs to be added to the header of the return call, which will cancel the next action. If the user does not answer in the period of time allocated, the notification will be sent to the members of the groups the user is a member of, this notification will contain the same data, with an added field which contains the contact-persons the user has set up. The thought is that the members can contact these persons and let them know that something might be wrong and take the appropriate action. Since Hyperglycemia isn't as critical, the notification is only sent to the source of the event. A diagram of the process can be seen in figure 7.4

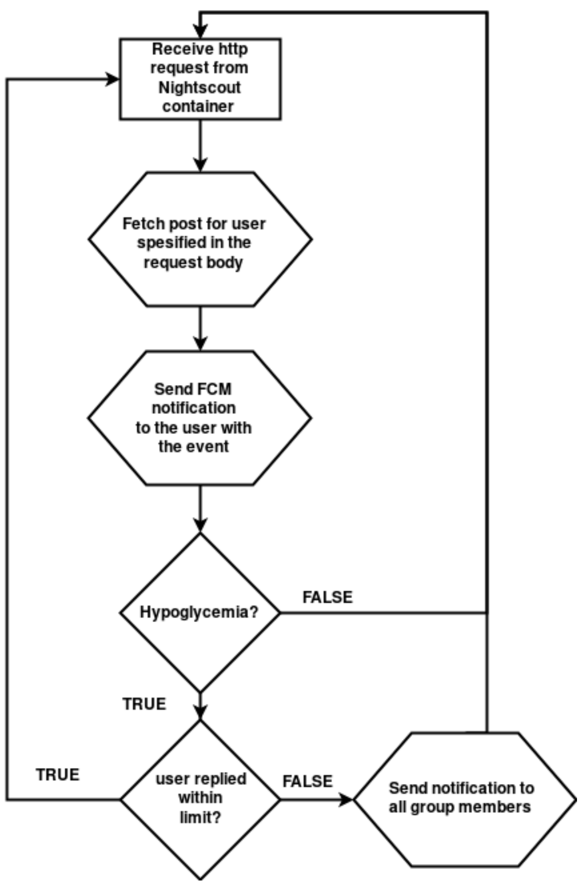

Figure 7.4: Diagram of FCM notifications

Outlook

The plan was to enable the extraction of appointments and meeting and show them in application and IoT device. But at the end of the project duration, the University changed from a local outlook implementation to Office365, which is not compatible with the implementation. The implementation is using a Python script with pyexchange, which implements the calls to a 2010 Exchange server, running is script is a Go module, which has the same sequence as the NightScout-fetcher. This means that the Golang module will look for users with an outlook config every 5min, start the python script for every user, extracting the appointments which are within 2 hours from now. These are returned to the golang and published to the groups live channels.

Dexcom

The Dexcom is not available for the EU-market, therefore this is just a proof-of-concept implementation and only tested with a sandbox user (a user account with simulated user data). The implementation is a proof of concept only, and currently only makes the Oauth2 tokens which are needed for the fetching of data. See the discussion on why this didn't get implemented further, that begin said it fully possible to develop this further which would allow users to import their data from Dexcom in the future.

7.1.5 Testings the Application

To develop the server application we used test-driven development, the test is implemented for the main endpoints. In addition to this we also manually tested the routes with the Postman[2], a commonly used API development test application. In addition, we also tested the server when developing the mobile application and IoT-device to see if it would act as expected.

7.2 Mobile application

To let users interact with the system, a mobile application was implemented. First, a native Android application was design, but due to concern about cross-platform support and the relative simplicity of the alternative, a NativeScript application was created instead. The android application was used as a proof of concept and a guideline for the design of the new application.

2. https://www.getpostman.com/

7.2.1 Concept Android Application

The Capstone project had an android application, which was planned to be developed further. But there were major shortcomings in the code and this caused problems when trying to develop it further, there was a number of compiler errors which seems to pop up at every new feature added. The strategy of using fragments instead of activities in the application also gave major headaches due to paging and problems with the back stack. Therefore it was decided to start fresh with a new application, which could serve as the main user interaction with the system. The experience learned from the Android application was incorporated into the new application, together with the general design. Pictures can be found in figure7.5, 7.6 and 7.7.

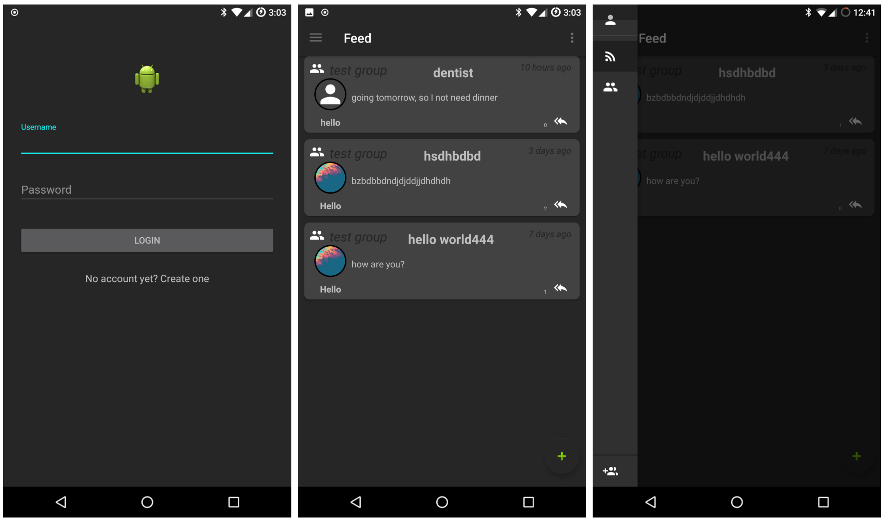

Figure 7.5: The Android applications screenshots, for the left: login-screen, feed-screen and the side-bar navigation bar

7.2.2 NativeScript Application

The application is built around the materialized design principles as defined by Google. The main navigation is done through the Navigation Drawer, which is collapsed while not in used and can be displayed by clicking the hamburger icon or by drawing from the left side. It contains a link to the pages/actions that can be accessed by the user and a user profile at the top.

When the application starts the user is presented by a login screen(figure 7.8, first picture), where the user can use his/her login details or sign-up if he/she does not already have an account. Once the login credentials are verified by the server, and the token has been returned, the application will call a number

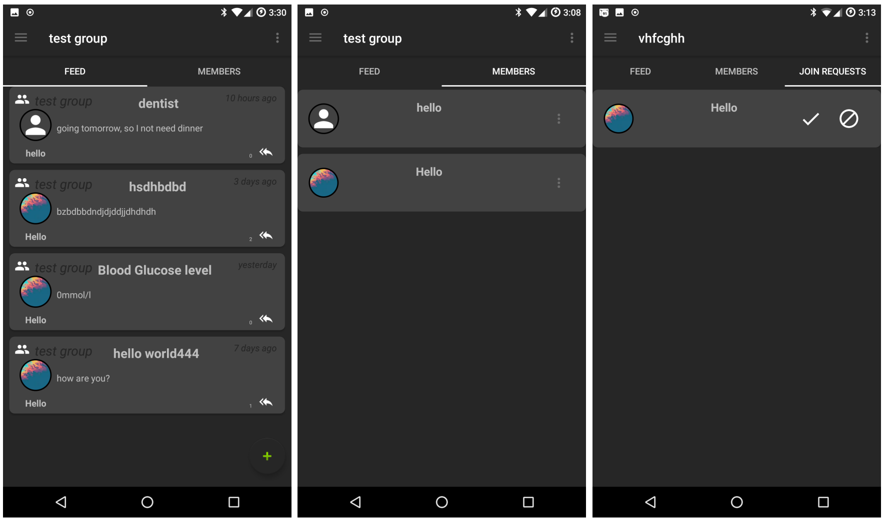

Figure 7.6: The Android applications screenshots, for the left: post from a specific group, group members and outstanding join-requests from a user

of HTTP calls, to fetch the user's groups, the groups users, and the feed.

Pages

The Feed (figure 7.8, second picture) is a collection of the latest text messages from all the user's groups and is presented once the HTTP calls are complete. The different messages are presented in a list, where each element in the list has a card, with the title, text, group name, username, users avatar and a time since the message was created. By pressing on the message-card a page will be displayed, with more details and replies to the message.

The group section shows the users groups, by clicking one of the cards will bring the user to the group page. In addition to this, the user also has the ability to create new groups, by clicking the Floating Action Button, which will make a dialog pop up. This page contains all the posts by users to that specific group, in addition to the BG messages from the members who have activated them. The page also contains the group members, and a chart of the number of steps each member has walked the last day, week and month (more on how to gather this later). The owner/creator of the group has special abilities/tabs which enables him/her to accept/decline requests to join the group, and some settings (like adding a group-picture). The BG message can be clicked like the normal messages, it will bring up a line-chart of the reported data for the last 24 hours.

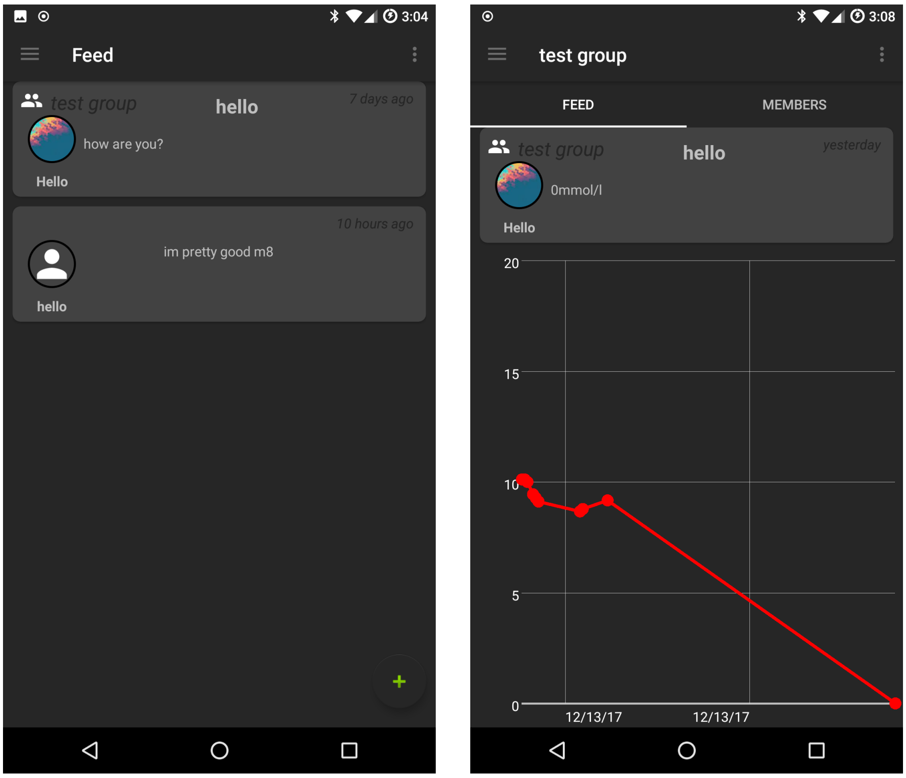

Figure 7.7: The Android applications screenshots, for the left: details about a specific post, with replies and a sample blood glucose graph

Search is a way for users to find new groups, by pressing a none green group (green indicated that the user is already a member), will create a request to join the group. Which the owner needs to approve, on the group page.

My measurements allows the user to view his/her own blood glucose level, activity, and Heart-rate in a line-chart. The activity and heart-rate are imported from the Gadgetbridge database if enabled in the settings. Google health and Apple Health import is also available, but only in the form of heart-rate. We use Chart.js in a webview, because of bugs/problem with the NativeScript-UI-chart library (see issue [3,4]), where intervals of hour could not be used, it also had multiple other problems with points not able to be more than 5 hours apart and the transition between days would crash the chart. Therefore we had to pick another option. There were two different choices: 1) was to incorporate a native android and IOS chart library, but would mean creating two different implementations. 2) to use a JavaScript library, in a normal webpage and load the page using the mobile OS webview, and use bidirectional communication

3. https://github.com/telerik/nativescript-ui-feedback/issues/136
4. https://github.com/telerik/nativescript-ui-feedback/issues/321

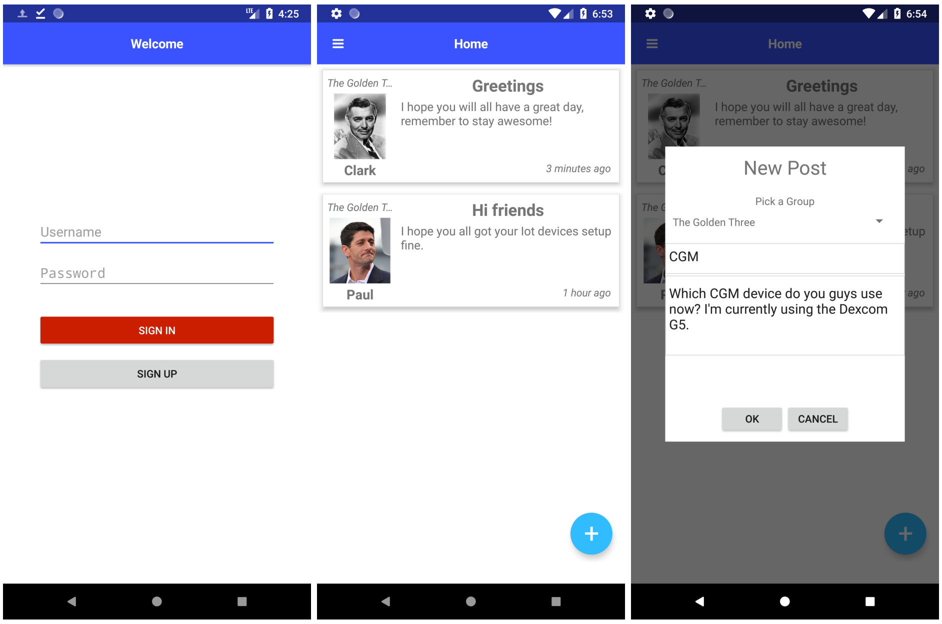

Figure 7.8: For the left: login screen, feed and new post dialog

between the mobile application and the webpage to transfer the data. To reduce the amount of work that had to be done, we chose the latter options and used chart.js to implement the chart. This might reduce the performance, but we get away from implementing different versions for each the mobile OS's. Since we support both importing the data from the GadgetBridge database and Google fit/Apple HealthKit we support a number of different fitness devices.

In the setting page , the user can configure his user account, adding a NightScout URL, Dexcom API, GadgetBridge Database, google fit/apple health and setting up their IoT device. The NightScout can be added, by supplying the URL, and the groups the user want to share the BGL within a dialog (if no group is picked, it will share with all the user's groups), the URL will be verified at the server side to ensure that it's valid. To enable the Dexcom API the user needs to sign in using Oauth2, which is implemented in the server portion of the system. The application will call the endpoint on the server, which will redirect to the Dexcom login service, the user logs in and the server gets the needed authentication details, as per the Oauth2 protocol. To close the webview (which we use to show the login-page), the application monitors the URL, and once it reaches the return-endpoint it will close the webview. GadgetBridge can auto-export the activity-logs at user specific intervals since the user also can pick the filename and location we need to let the user pick that file, so we can use it in the application. A file picker was implemented in a dialog so that the user can do exactly that. This file is then used in the charts

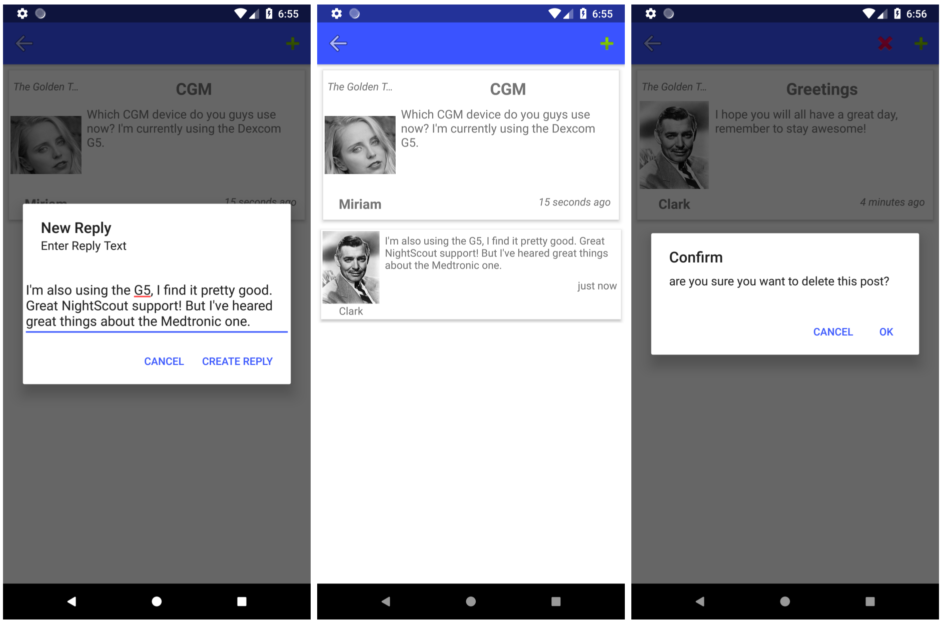

Figure 7.9: for the left: new reply dialog, post-detail and delete confirmation dialog

and the step-counting mechanism.

To enable the competition of the number of steps between the users, we need a mechanism to upload this data to the server, a time of the last time the user synced the activity data is also necessary. When a user wants to sync her/his data to the server, a worker thread will be launched so as to not interrupt any other action the user is doing, and not halting the main thread. This worker will first contact the server, and get the last sync time, and then query the Gadgetbridge database for the all rows from that time to now, which are aggregated by days, and sent then sent to the server which stores the information, and uses it in the competition and achievements for the user. A problem was encounter due to the fact that time zone is not inherited by the host machine to the docker instances, therefore the start of the day on the server was UTC while on the mobile application this was the local time. Since we use UNIX-time, in every part of the application this was rather hard to identify and fix. For Google Health/Apple Fit we just query the API with the start-date and aggregation options, which are then sent to the server, in the same way as the GadgetBridge data.

For users to view statistics, set up contact persons, upload an avatar (user profile picture) and view achievements a separate page was implemented, namely the user profile page. The user's avatar can be changed by clicking on the image, which brings up the native-media dialog, and the image can be picked from

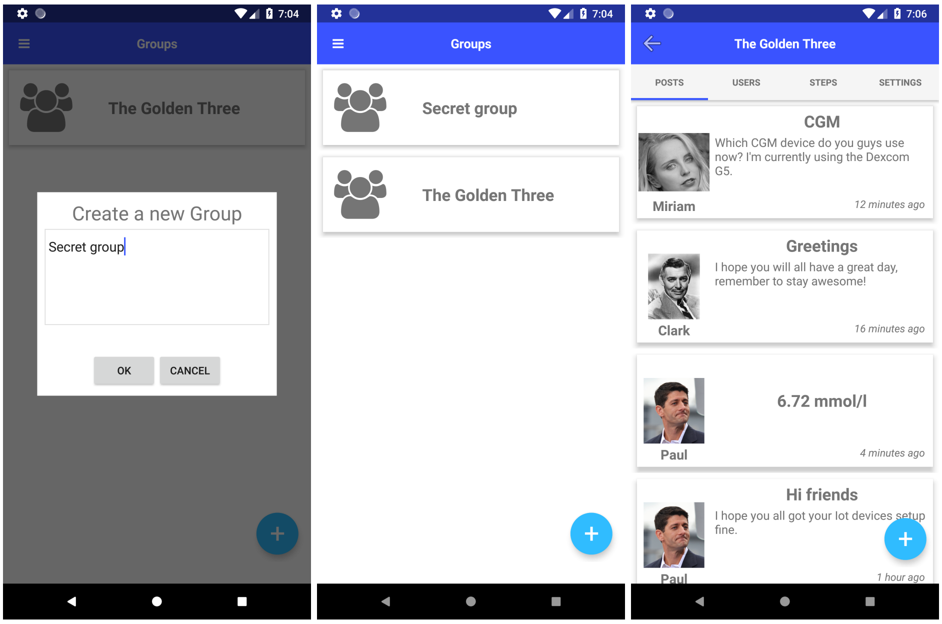

Figure 7.10: for the left: new group dialog, the groups the user is part of, and Group details page (showing the messages from that group)

within that. After the user has selected a new image, this will be sent to the web-server which will in compress it and store it, in addition, to update the users model to reflect the changed avatar.

Handling BGL events is a major feature of the application, like said in the server section when a low or high BGL level is detected an FCM message is first sent to the user which has the event to inform him/her of the occurrence. Since high levels aren't deadly (in the worst-case scenario), we do not send that event to the rest of the group members, it's more an alert to the person. For low blood sugar, we send a warning to the person like for high-levels, but if the person does not answer in a period of time (5 min) through the mobile application, a notification is sent to every member in the groups the user has allowed to see the NightScout data (forward-groups). The notification contains the latest BGL event, and a list of contact persons, which member can try to contact. Since we're using FCM message, these notification have two different effects depending on if the mobile application is open (and running), firstly if the application is not running a notification is created, and once press will launch the application and bring the user to a dialog with the previously described data. If the application is running and in focus, the same dialog will be displayed to the user without the need for a notification.

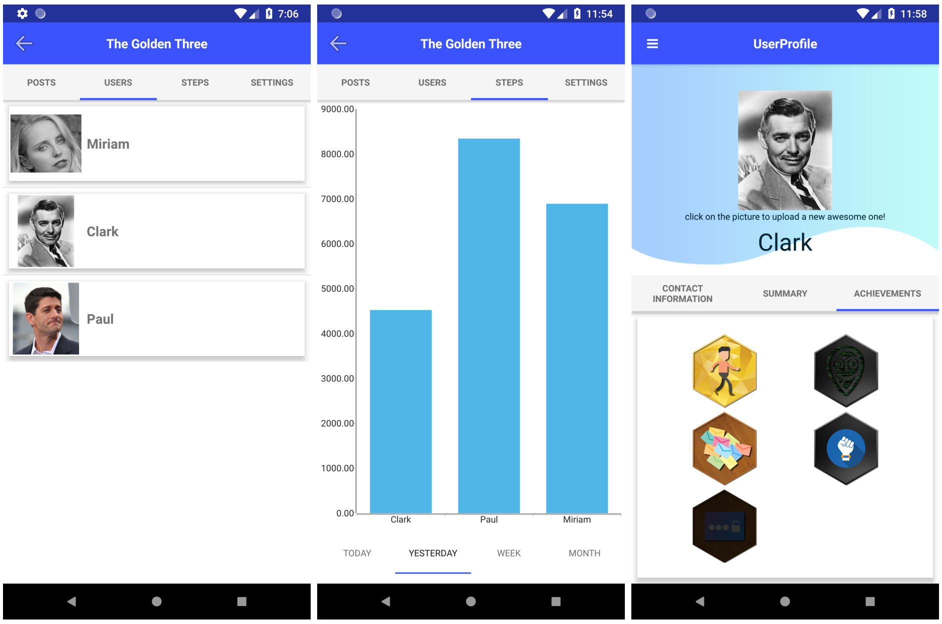

Figure 7.11: for the left: Group members, step competition and badges

Achievements/badges The achievements look and implementation are influenced by the mobile augmented-reality game Ingress, developed by Niantic. The achievements are represented as badges, with different rarity (wood, silver, gold, and graphene) based on the badge requirement, for example, the number of steps, where the badge are granted after 10000, 25000, 50000 and 100000 km. The badges are displayed in a grid with the images of the badges when pressed a dialog is launched which display the requirements rarities and the current progress to the next badge. Currently, we have 5 badges: daily logins, number of steps, number of posts and adding a CGM endpoint and allowing the application to read from a gadgetbridge database. The allocation of achievements is managed by the mobile application, and thus the server only contains the raw data.

The application uses session and local-storage to cache already fetched data, like groups and user model, in addition to the logged in user's model, settings and token. Be caching the results we reduce the amount of data which are downloaded for each new page reduced, by a significant amount, thus reducing the rendering speed by the pages. Since updates happen we fetch always fetch the newest data for example when the feed or the group-page is rendered.

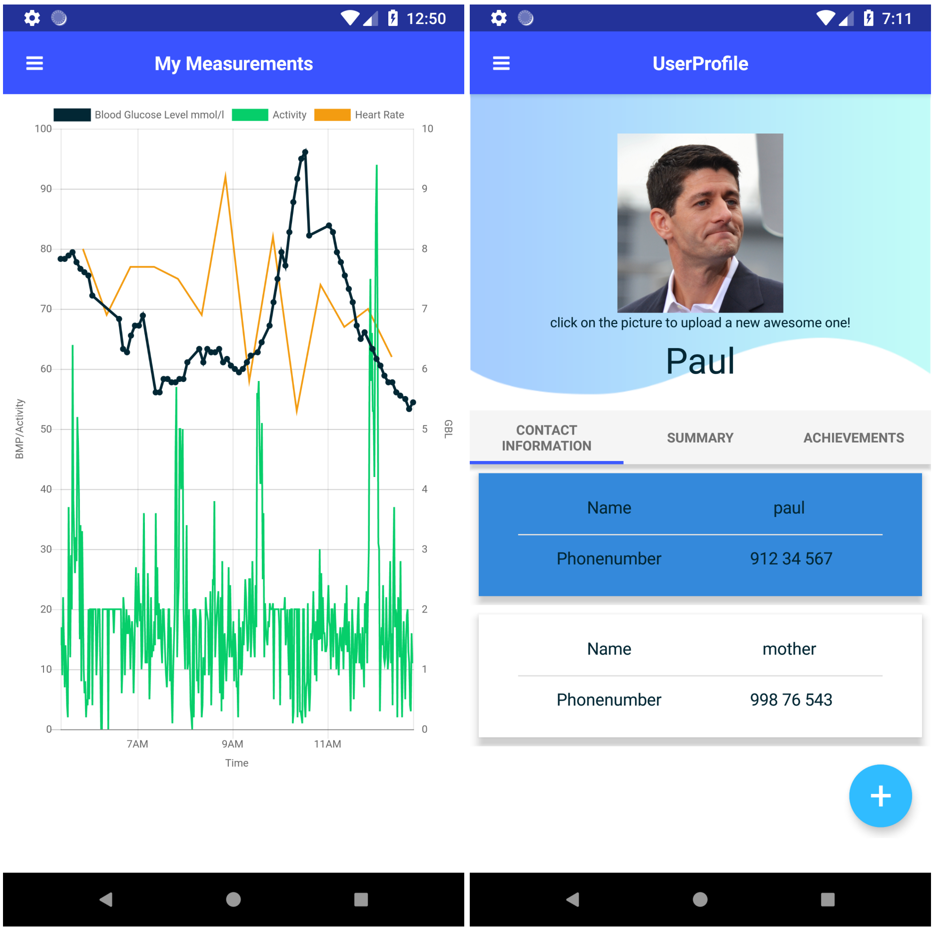

Figure 7.12: Personal BGL with physical activity, the different layers can be disabled and user-profile showing the contacts.

Testing

The development was done with contiguous testing of each element in the application since the author not has access to an Apple computer with Xcode, only the Android version was tested. To make it run on IOS it would most likely involve some minor fixes and improvement to some of the code (including the different external modules, like firebase, health kit and so on).

7.3 IoT Device

The IoT device is a frontend that is thought to be a small, power-efficient device that could be placed in commonly visited areas, like the kitchen, work-place

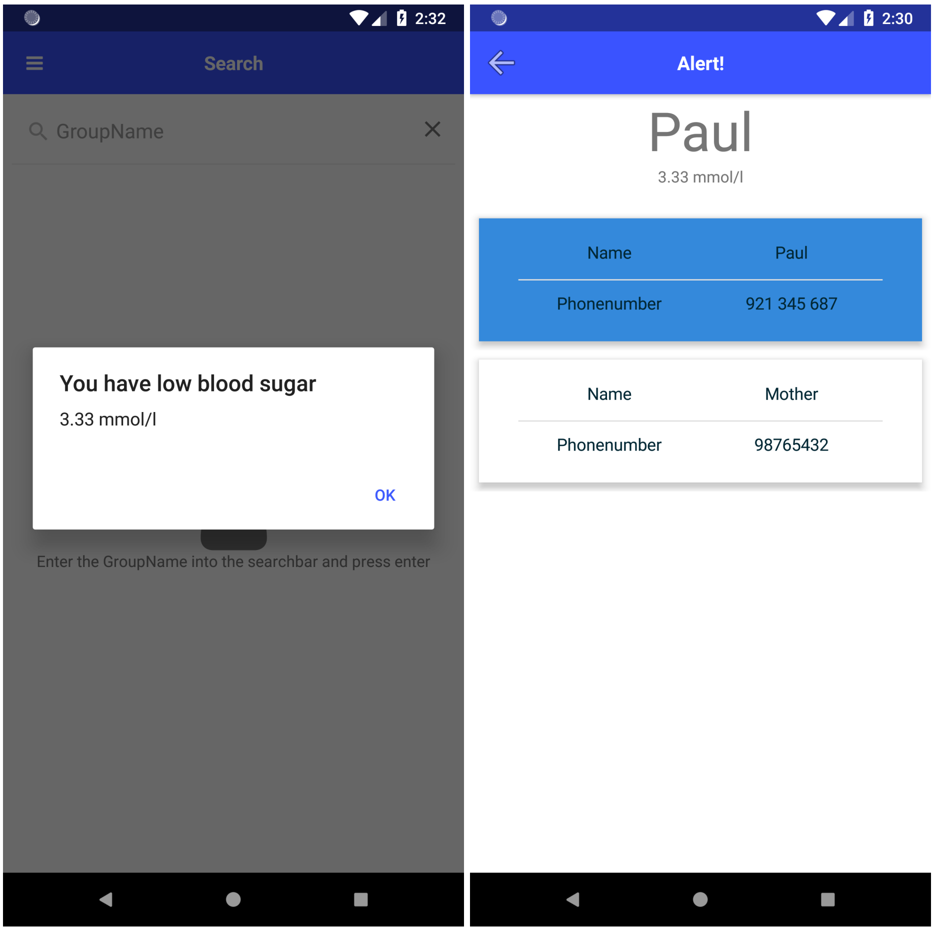

Figure 7.13: Dismissible notification, and alarm with contact information from a alarm which was ignored by the source of the alarm.

and so on. It shows the latest messages which are sent by members of the user's group and their blood glucose levels on the included EPD display.

The EPD display is 2.9inches and has a resolution of 296x128, so naturally, there is a limited amount of information that could be shown of it and make it readable; therefore we do not show replies to messages. The limited amount of memory is also a concern, and therefore we limit the number of glucose measurements and messages, but this is handled on the server side mostly (except for new messages, where the device itself handles the deletion of older messages). For displaying messages on the screen, we use the GxEPD library which is dependent on top of the Adafruit GFX Library, from these libraries we get primitive drawing functions which are used to create the different screens/images on the screen. These primitives include fonts, xbitmap drawing, partial updates and simple geometric figures (lines and circles).

Figure 7.14: The IoT-device with a BGL from NightScout

Validating the HTTPS certificate is not a trivial operation on the ESP8266, this is due to the fact that the SDK does not include a root certificate, and can't internally verify the signature of an HTTPS connection. The ESP8266 does support TLS1.2, but since the root CA is not included, we need to supply the signed CA for the supplier (Let's Encrypt). But the device does not support the CA as plaintext, so firstly the following procedure was found to work[5]:

1. Export the **supplier CA** in DER format from the website (for let's encrypt this is DST Root CA X3 from Digital Signature Trust Co.)

2. use the UNIX command *xxd* to do a hexdump of the DER CA to a C include file style file

3. Include the newly created output file in the source code of the IoT device

4. Additionally, the certificate need to be validated in the firmware:

   (a) Synchronize time using a Network Time Protocol server - this is needed to validate the TLS certificates offered by the server are currently valid

   (b) Lastly one can validate the certificates, using the function *verifyCertChain*

For communication we mainly use MQTT. The messages are sent from the server to topics, to which the device is subscribed to. These messages are

5. see https://github.com/esp8266/Arduino/issues/1851, for discussion

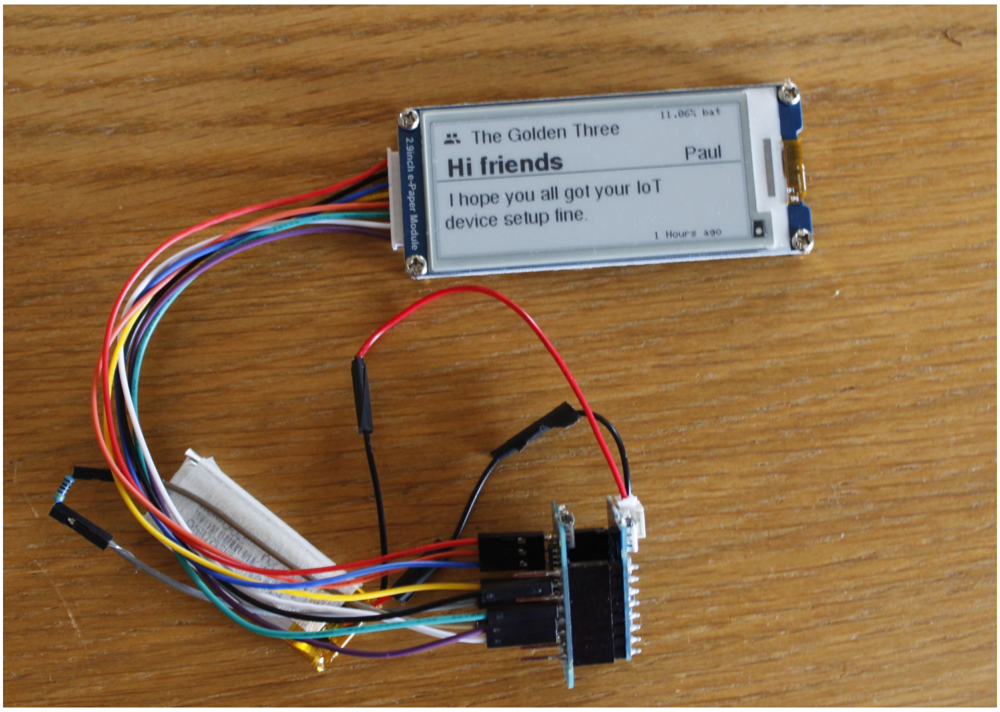

Figure 7.15: The IoT-device showing a sample text message sent to a group

cached by the MQTT server and ensure that they are delivered at least once to each of the IoT devices (through the use of QOS 1).

To setup/configure the device, a simple web server is running on the device. If there already exists a config file (stored in the internal flash storage), the web server is only active for 5 mins, before it's turn off giving the user an opportunity to change the settings. If the device has no config file, the device will go into station mode, where the user can connect their mobile phone to the WiFi channel the device sets up, and config can be made. The user can change the username/password, wireless authentication and the groups the device should follow.

To reduce the power-usage the device goes into a low powered sleep mode, called light-sleep which should (according to Espressif) only draw 0.4mA, but this might be higher since this only count the MCU itself, and not the serial to USB and other chips on the WEMOS development board and battery-shield. Further discussion on power-usage and alternatives are in the discussion section.

A diagram of the process can be viewed in figure 7.16.

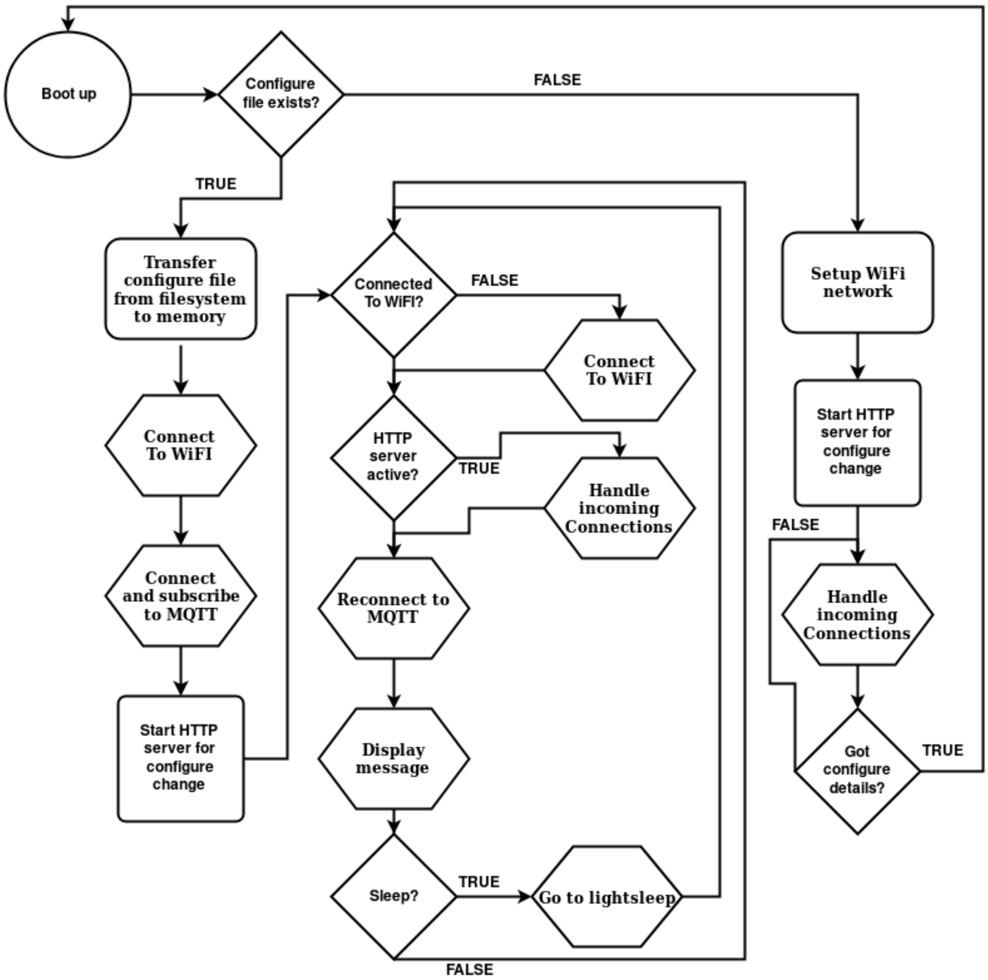

Figure 7.16: Diagram of the runtime cycle for the IoT device

8

Discussion

8.1 Scenario

Coming back to the scenario in the feature chapter we will go through what has been implemented, and how they could impact the life of diabetics and their family, friends, and care-persons.

8.1.1 Alarm functionality

Firstly the alarm system which was used multiple times in the example is implemented and works, this is the one requirement that was the most requested by the interviewees. The basic function of the alarm is described in the design/implementation chapter. But this isn't that novel, since multiple CGM companies (see related works chapter) provide the same service, e.g. sending a notification to contact persons on an event. But the things that differentiate our system with these, are that the BGL values are available for the group members at all timed, and in near real-time. Like we saw in the scenario, Paul saw that Clark had a very low BGL in the morning, and it didn't improve. He first tried to contact him via the application and eventually called him and told him before he got a hypo event.

8.1.2 Social Interaction

The social interaction is also an important part, small groups are something that was brought up a number of time in the Mhealth motivation interviews. Some of the interviewed persons said that didn't feel comfortable to share personal, and preferred small groups, where everyone knew each other from real life or from other online communities. In the scenario, Paul asked the others what to do when he couldn't get the nutritional information from a cafeteria, and the other told him their experiences and tips on how to handle this kind of events. This might not be that private, but it might be that if he posted it on, for example, a public social network, he would get answers from persons, that not necessarily know or be informed about his condition. Instead, he can post in a place he knows that he will get reasonable answers from persons with the same condition and know how he thinks/behaves.

Going back to self-determination theory one of the person's physiological needs of every human being needed for optimal well-begin and performance was *Relativeness*. In a group setting like this you are cared about, can care about other and feel you belong to a group. Patient support groups are a good example of how a group of persons with a disease can learn and help other to coup, and improve their condition. Patient groups are commonly used for chronic diseases, like diabetes. They help persons to find their own individualized solutions through sharing experiences amount peer-patients, Huh et al. found three main factors that help the persons in a patient support group[72]:

- Patients operationalize strategies and daily routines to a form that could easily be adopted, tried out, and tweaked. this includes set-by-step instructions, bullet-points of lessons, blog, book etc.

- Illness trajectories alignment work, where the patients through comparing and contrasting each other's personalized experiences, other gains a personal knowledge about the diseases. This could be that an experienced patient shares how they coup with a newly diagnosed patient which are struggling to come to grips with the condition. This could involve offering warnings, reassurances about the future etc.

- Sharing operationalizes and illness trajectories result in that the patient group members build a common understanding of their chronic disease, which helps each individual member towards a better self-management routine.

Since the group members are meant to be interacting with each other on regular basis some effects can be drawn from collaborative learning in for example small student groups. Slavin[73] distinguishes three different effects of collaborative learning on student groups:

- **Motivation** - Group motivates the groups' members to exert maximum effort because each member of the group can only attain his/her own personal goal if the group does.

- **Cohesiveness** - The group develop a team spirit through their interaction and tried to help each other so that the group could succeed.

- **Developmental perspective** - The interaction with other group member makes it possible to discuss, argue, present and hear another person's viewpoint, which might contribute to an increase in the personal knowledge of the topic.

For this application the last two are the most influential, where they help each other with warnings, tips, suggestions, discussing different methods to improve their self-management and so on. With activity over a longer period-of-time, a team spirit would form between the members in the group, and thus prolong the usage of the application. In addition to discussing their problem in a private and secure place, they can increase their knowledge by sharing or reading experiences. The first effect could be interesting to explore, where the group as a collective might, for example, compete for a place on a leaderboard, this could be the average number of steps, HbA_1c or similar things. But this would have to require the agreement of every member of a group before that group could join.

There is already a number of platforms for discussing chronic disease management, like face-to-face patient support groups, but in the last few years online communities have popped up as an alternative. These are a rapidly growing platform for discussing health-related issues, with reports of up to 90% of older adults using social media to find and share health-related topics. The result also showed that the eHealth literacy is increased in those who use these social networks for their illness [74]. The creation of groups on Facebook, can be open or closed to the public, but even if the group is closed the data is utilized by Facebook for various reasons (notable to personalize advertisement), which make some worried about sharing their personal information in such groups, especially after the recent Cambridge Analytica Scandal, which had a huge impact in the public trust of Facebook Inc. [1]. In our application the group are private, but the data is open on the server. But since the source code is available for anyone, so anyone can fork the application and run it for themselves and a few friends, this would solve many of the problems with came forth from the

1. http://nordic.businessinsider.com/facebook-trust-collapses-after-cambridge-analytica-data-scandal-2018-4?r=US&IR=T

mHealth interviews regarding privacy concerns about their own data, and how it's used.

8.1.3 Competition

Some persons find competitions motivational, while others don't. We saw this trend in the Mhealth interviews where some were very competitive, while others were not. Franken and Brown argue that this because of the distinction between ego- and task-orientated persons, where task-orientated can be motivated by the task itself, rather than the reward or feeling of winning. It follows that their improvement in performance is the driving force behind their motivation to compete. For some winning is the key, and for some competition is motivation is a driving force to put forth the effort which might improve their performance in the future. They also found that males are more likely to be competitive than women[75]. Malhotra also makes a clear distinction between "competitive motivation" and the "desire to win" [76]. Paul is definitely a "desire to win" person, while Miriam gets "competitive motivation" from competing.

8.1.4 The knowledge of someone else watching over you

From the requirements interviews, we heard that the fact that someone else could watch over your BGL values, while the person was doing something else would give some relief from the constant worrying about their condition. We saw this in the scenario when Miriam went out for dinner with her company, and she asked the others to keep an eye out for her. Even though every patient has their condition in the back of their mind at all times, it helps them having "time off", where they can concentrate on other things, without having a constant worry about their condition. Of course, this shouldn't be a system that makes other take care of you at all times, but it could be a support for events like this. This also ties in to STD's *relativeness* concept.

8.1.5 Badges

Badges/achievements are a well-known gamification mechanism, used in many settings from real games, software stores, mobile application to mhealth applications. Badges seem to increase the positive affect competence need satisfaction, as well as perceived task meaningfulness [43]. In the scenario Paul walked, to lower his BGL values, but as an extra benefit, he made some progress on his next badge, which might contribute that he went a bit further than he otherwise would have done.

All of the features described in the scenario is implemented, except the personalized comment/tips/suggestions which they received at the end of the day. Diabetes is a complex disorder, and a system that could analyze the performance of a person's data throughout a whole day is way out of scope for a master thesis. But we added it to illustrate the usefulness and possibilities that collecting data might have in the future, through advance Machine learning/AI.

8.2 Dexcom

We chose to add a proof-of-concept login to Dexcom since not everyone will have the opportunity to use NightScout with their CGM sensor. But the API provided by Dexcom is not usable for alarm/notification part of the application, this is due to the choice by Dexcom to apply a three-hour delay for retrieving the data reported by the CGM, with the explanation:

Currently, the Dexcom API offers EGV data on a three-hour delay. The Dexcom API is intended for software developers to retrieve retrospective CGM data and CGM-derived metrics for use in software applications. Real-time treatment decisions should not be made based on the data displayed to the end user, and the delay helps clarify this. Dexcom settled on the three-hour delay after consultation with FDA. [2]

8.3 Notification/alarms

The current implementation does not try to predict glucose levels, it only reacts to the current values reported by NightScout. A more optimal option would be a prediction of the BG level, instead of relying on the current values. Another side of the issue is that CGM sensors often have built-in predictions so maybe a prediction might not be needed, there is not a risk to health before the levels extend the recommended threshold, and thus the person does not need any external attention.

8.3.1 Thresholds

The thresholds are based on the American Diabetes Association guidelines for Hypo and Hyper, 3.9mmol/l and 7.7 mmol/l. These parameters differ

2. https://developer.dexcom.com/content/frequently-asked-questions

from person-to-person, therefore it should not be as static as it currently is implemented, in the future this might be changed to allow the users themselves to pick the threshold values. Another possible solution is to use machine learning to detect the onset of hypo/hyper based on data from the users' CGM, carb intake and physical activity, but that's a task which would require a great deal of development time.

8.3.2 Alarm Fatigue

Too many alarms is a concern that might be relevant for members which make larger groups or join too many groups. Where they become desensitized by too many alarms over a period-of-time, this effect is called alarm fatigue and makes the alarm sound become *background noise*. This effect is evident in for example clinicians, where alarms are commonplace and many false or clinically irrelevant. [77]

A possible limitation to the number of members in a group and the number of groups a member can join might partially solve this concern. But additionally the current implementation right now will make a new notification every single time a new BGL value is fetched, a method to reduce the number of notification is to implement a silence period between each time a notification is sent. For example, when the source (the person with the event) click on the notification (and disables the sending to others in the group), there should be at least 10 minutes until the next notification is sent.

8.4 IoT-Device

The device is meant to be something that could be put multiple places in the house, office, breakroom etc. as a low-cost, low-maintenance method of viewing messages, and BGL values from the user group members. It can't be interacted with and receives updates continuously. One of the main features of the device is that it should be low-cost and low-maintenance, meaning that it should run for days if not weeks without the need for charging or other action taken by the user.

8.4.1 BGL message

The BGL message is made up by the latest 8 measurements taken from the source of the values, the limited amount of points is due to both the JSON decoders limit and the memory concern of the IoT-device. If a user is a member of many groups, and the device is set to show all of these it might run out of memory which is a real concern. This could easily be increased in the future and is basically just limited in the backend part of the application.

8.4.2 Power Usage

The IoT device is an ESP8266 which has a limited amount of memory and processing power, but compared to other MCU it's rather powerful but draws a substantial amount of power. To reduce the power requirements processing time, wireless communication and runtime should be reduced to a minimum.

This can be achieved first and foremost by implementing sleep cycles into the sequence of operations, in the implementation it was mentioned that we use light-sleep. table 8.1 shows the different sleep-modes that the ESP is capable of using.

Item		Modem-sleep		Light-sleep		Deep-sleep
		Automatic	Forced	Automatic	Forced	Forced
Wi-Fi connectivity		Connected	Disconnected	Connected	Disconnected	Disconnected
GPIO state		Unchanged		Unchanged		Unchanged (2 µA)
Wi-Fi		OFF		OFF		OFF
System clock		ON		OFF		OFF
RTC		ON		ON		ON
CPU		ON		Pending		OFF
Substrate current		15 mA		0.4 mA		~ 20 µA
Average current	DTIM = 1	16.2 mA		1.8 mA		-
	DTIM = 3	15.4 mA		0.9 mA		
	DTIM = 10	15.2 mA		0.55 mA		

Table 8.1: Difference between the three sleep modes of the ESP8266 (source: [78])

Deep Sleep is the best solution for power saving. It turns off all sections of the MCU, except for the Real Time Clock (RTC), this also involves removing all data in the memory, including the Program Counter. Saving states in this mode is rather problematic because it basically comes down to three options, the first is to store it in the very small 512 bytes RTC memory, which is not enough to store the messages and associated data which is needed for the operation. One option would be to rely on retain message only, but this would increase the amount of data the device need to fetch for each sleep cycle, another problem would be the BGL messages which are only sent through the live channel, this could be modified to be stored in retain message, with a channel for each group member. Another option is to store data in the internal flash memory with the SPLIFF file system, this is already used for the configure file that is used to change the WiFi-, MQTT- credentials, and groups the device will subscribe to. The problem with this approach is that the flash memory has a limited read/writes before the EEPROM (around 100,000 erase/write cycles [79]) starts to fail. The last option requires additional hardware in form of an SDcard reader and a card, this has a higher write count tolerance and would last longer than the EEPROM.

LightSleep , the second-best sleep mode. This mode turns off WiFi, system-clock and suspends the CPU operation. It holds the data in memory, which is helpful for shorter sleep cycles.

ModemSleep , the standard sleep mode, here the only thing that is turned off is the WiFi-module. It supplies a small power reduction useful for very short delays.

Optimal sleep-cycle The optimal solution would involve deep and light sleep. Deep sleep for long periods of time, this could be controlled by using a motion detection sensor, which interrupts the sleep and starts the device when it detects movement. After the device has exited the deep sleep mode it could run for a period of time in the light-sleep mode before checking if a movement is detected, if yes it should continue with light-sleep and if not, the device should go back to deep sleep. The problem is that the ESP8266 deep-sleep wakeup procedure is rather constructive. The RST pin and D0 need to be connected to even wake up from a timed sleep, the RST will send a small signal to the D0 pin to essentially reset the device. There are known workarounds for this (see https://github.com/rgrokett/ESP8266_PIRv2/blob/master/ESP8266_PIRv2.pdf), but this approach requires an NPN transistor which was ordered but did not arrive before the delivery deadline of the project.

Current power usage Without an oscilloscope the absolute power usage is hard to measure, therefore this will just be an approximation of the power usage of the device. Multiple ways of trying to measure the power-draw but to no success, it might seem like the power meter has a too high resistor and makes the ESP brownout (not enough power to boot), so we will use the findings from two sources. Firstly, we the usage based on Espressif and the E-paper display documentation. The second is from our own testing, basing our result on the batteries voltage.

$$Operation = 80\ mA \times 10\ sec = 800\ mAsec$$

$Sleep \; = \; 6 \; mA \; \times \; 60 \; sec \; = \; 36 \; mAsec$

$Number \; of \; cycles \; = \; \frac{1000 \; mA*3600 \; sec}{800+36}$
$= \; 4306 \; cycles \; = \; 71 \; hours = \sim 3days$

We first tested a with one device running with a 700mAh battery, to measure the voltage drop we connected a jumper on the battery shield and read the values via the Analog to Digital pin, and calculated the raw data to voltage. The voltage was read every 15 min and published to MQTT channel which was read and stored in the database for later analyses. The first experiment gave a surprise, the light-sleep implementation could only run for 8,4hours, which was disappointing. This data can be found in figure 8.1 (green line), and gives the indication that there is something wrong with our assumption, I see two possible causes:

- Light Sleep does not work properly
- Battery Shield draws a constant amount of ampere disregarding needed for the MCU.

With regards to light-sleep almost everything was tried which is listed in the light-sleep thread on github[3], but one of two things happen: 1) exception 2) did not wake up from sleep. The only working implementation is the one which is included in the source code.

To investigate further we made a stripped-down version of source code, this time with deep-sleep, to see if the drainage was improved. This version does only rely on retail messages, so there are no BGL updates for the time being. We utilize the RTC memory for the most important variables, what group and message that should be displayed next, and the boot count. Since the RTC is just a system clock, the time needs to be fetched for every cycle. The current implementation only uses 24 bytes of RTC memory, so in the future, it might be possible to remove the need for reads from the EEPROM, to fetch WiFi-info, server authentication and the groups the device are subscribed to. By Espressif specification with deep sleep, we should see a battery life of:

$Operation \; = \; 80 \; mA \; \times \; 5 \; sec \; = \; 400 \; mAsec$

$Sleep \; = \; 0.02 \; mA \; \times \; 60 \; sec \; = \; 1.2 \; mAsec$

$Number \; of \; cycles \; = \; \frac{1000 \; mA*3600 \; sec}{400+1.2}$
$= \; 8973 \; cycles \; = \; 149 \; hours = \sim 6.2days$

3. https://github.com/esp8266/Arduino/issues/1381

As seen in figure 8.1,there is something wrong with the light-sleep functionalities. To see if the device even went to sleep we conducted another experiment, this time with modem-sleep instead of light-sleep. The result was that the device ran for 18hours, so over twice the duration of the light-sleep implementation. it eventuates that the implementation of the light-sleep functionality is the cause of the lackluster battery performance. But this could be fixed by a possible firmware update in the future. Both of these are included in the source code, where the light-sleep is commented out. The deep-sleep concept implementation is also included.

Generally, the battery time could be even extended by using long sleep cycles. Another possibility for light and modem sleep is to not connect to WiFi and just serve the text messages, and just occasionally connect to the WiFi and MQTT to fetch any new messages. But for further development, the most likely candidate is to utilize deep-sleep and try to implement some kind of system for devices to fetch BGL values instead of relying on the live ones. An idea is to have a field in the group retain for the number of group members that have added NightScout URLs, and create new channels like *group_id/BGL/number*, and then the device can just subscribe to *group_id/BGL*, and get all the BGL values. But this involves a great deal of network traffic, which is not very attractive in terms of power-consumption (there are well-known spikes to 200mA when the ESP is connected to WiFi).

Problems with the ESP8266 , The last segment illustrated some of the problems with the ESP8266, namely the lack of documentation and debug tools. It's mostly used by the hobbyists in a sense and observe fashion, where the device has a sensor which it reads and sends that data to another device, and the source code is often very simple and minimalistic compared to the source code of the device used in this project. Throughout the project duration there were new problems with the ESP, random crashes, sleep modes not working etc. and the stack exceptions 8.2 is almost impossible to decode without an external tool[4], some even won't be recognised by the tool.

8.4.3 Alternatives

There aren't many alternative MCU's that could be used for this project, the ones that were found are listed in table 8.2 the table we can see that there is really just 3 options for the MCU, ESP8266, ESP32 and RTL8710.

The ESP32[80] is basically the big brother of the ESP8266, but have some additional features like Bluetooth LE, a dual-core processor run at twice the

4. https://github.com/me-no-dev/EspExceptionDecoder

Manufacturer	Productname	Processor	Price
Espressif	ESP8266	Tensilica Xtensa L106 (80/160 MHz)	$3
Espressif	ESP32	Tensilica Xtensa LX6 (240 MHz)	$5-6
Realtek	RTL8710	ARM Cortex-M3 (166 MHz)	$6
Texas Instruments	CC3220S-LAUNCHXL	ARM Cortex-M4 (80 MHz)	$40 for devboard
MediaTek	MT7688	MIPS24KE (580 MHz)	$25

Table 8.2: MCU with WIFI source: https://en.wikipedia.org/wiki/List_of_Wi-Fi_microcontrollers, with prices found on Aliexpress/google.

speed, more storage, and memory. Andreas Spiess[5][6] has compared popular development boards with regards to their power consumption [7] and the results are very different depending on the board used, but the general result is that they use around 10 times more power in deep sleep and 2x in idle compared to the ESP8266. The ESP32 is rather exciting in term of the possibilities that could be implemented with regards to memory, processing power and Bluetooth. Porting from the ESP8266 to the ESP32 is rather simple, and just requires some modifications to the sleep code. Additionally, the idle sleep cycled discussed above could be implemented on it without the NPN transistor since it supports deep sleep wakeup from a GPIO pin.

The Realtek RTL8710[81] uses an ARM Cortex-M3 Processor 166MHz and is comparative to the ESP32, but does not support Bluetooth. It runs the FreeRTOS as default operating system, as is supported by Arduino. The power consumption is listed as 140-180mA with WiFi and 10 μA deep sleep, but this is the MCU itself and may vary in the development boards as seen in the ESP32 (and most other MCU).

If power consumption is not a concern, aka if the plan is to leave plugged into a USB charger. A single-board computer like the Raspberry Pi zero w or any of the many clones could be used instead. The benefit would be that you could use it for this application, but also other things like a small web-. web-server or even a DNS resolver (see PiHole).

5. http://www.sensorsiot.org/
6. https://www.youtube.com/channel/UCu7_D0048KbfhpEohoP7YSQ
7. https://docs.google.com/spreadsheets/d/1Mu-bNwpnkiNUiM7f2dx8-gPnIAFMibsC2hMlWhIHbPQ/edit#gid=0

8.4.4 Display

For this prototype, we use a 2.9inch black and white screen from Waveshare, but there are many other options OLED, LCD and color/gray tone EPD displays. The strengths of EPD are that they require very little power to change the image and holds the image even when there's no power, which none of the others are capable of doing. They have a good visibility (like ink on paper) in light environments but does not have an internal light source, which makes them not visual in darkness.

8.5 Server

The server as said in the design/implementation chapter is separated into small containers which make the deployment and optionally scaling easy. The only thing needed to run the application is docker, docker-compose and let's encrypt installed on a host machine and everything else is fetching and built automatically.

Docker Docker is a very useful and powerful containerization application and makes it easy to deploy larger and more complex application without the need for user configuration, it's all included in the docker file and thus the end-user only needs to build the application to get up and running. The configure took some time to get into, but it made the development of server backend smoother. A benefit with using smaller programs instead of a big one is that single containers can be restarted once a change has been done, so for example when a small portion of the NightScout-fetcher container was changed, the whole system didn't need to be restarted. Of course, there are some challenges with cross-communication, but this only necessary for two of the containers, the NightScout-fetcher and the FCM-notification container, and was done through internal HTTP communication. An alternative for Docker is Kubernetes which is design by Google and now maintained by the Cloud Native Computing Foundation.

Alternatives , There is a number of different languages that could be used instead of Go, the most popular for web server application are Node.js and Java with Spring. The choice of Golang has a number of reasons, the most important one is that the syntax is very C-like, it's a compiled language and it not run in a VM. In addition, the author has experience in the language from earlier, and before the project started was involved in building a RESTFUL API using Ruby-on-rails, and had researched the possibility to utilize Go instead, for that project. Overall Golang is still a young language but has matured in last few years with more developers and packages.

Alternatives for the microservice architecture is, of course, a monolithic one, which would have worked just fine for the limited number of users currently using the application, but with regards to the future possible increase in usage, it's a good way to be able to scale. Another important point is the development of new features, where those could we worked on without disturbing/interacting the other containers. In a monolithic application, there is often two branched deployed one for testing new features/development and one for operation, this would not be needed in a microservice application.

8.6 Mobile-application

The choice of start over from scratch wasn't easy, the first option was to create a new Native Android application, but since this would not give us cross-platform capability a number of platforms were researched. There is a number of new platforms that could provide what we're looking for, namely easy development, cross-platform capability and a relative native experience, from this we started to investigate the alternatives:

- Flutter - new Android SDK developed mainly for the upcoming Google Mobile OS Fuchsia, written in Dart.
- React-Native - a cross-platform SDK which uses React.js developed by Facebook.
- Xamarin - created by Windows, written in C#.
- Ionic - uses Cordova to build/deploy as native applications. Uses webview instead of native-elements, which reduce the performance, but allows for code (without modifications) to be deployed as a website.
- NativeScript - an open source cross-platform SDK from Telerik built using JavaScript.

Since the author has some limited experience in JavaScript, Flutter and Xamarin were ruled out. After some simple experimenting Ionic was also ruled out, because of the lack of performance compared to the remaining candidates. Lastly, React-Native was ruled out because of an ideological reason, React is developed by Facebook Inc. and have a history of *BSD + patents license* in the past. This license stated that you could not was use React if you were direct or indirect competition with Facebook [8], this triggered the Apache project to ban every open source projects from Facebook in July of 2017[9]. Facebook later re-licenses React under MIT license [10]. Another reason why NativeScript was chosen is that you can use Angular, which is one of the most used JavaScript frameworks, and would give the author some experience in the framework.

Generally, the application is in a useful state, there are of course some areas that the application could be polished and improved, but the main functionalities are in place. Some of the minor features like allowing other to see a persons user-profile and group-management are not in place of delivery, but this is due to that time ran out. An application takes a long time to implement and the different pages need to be planned out, designed and then implemented, which takes a significant amount of time.

8.7 User-Study

To figure out the usability of our product/system we conducted a small user-study with 3 people, 2 with type 1 diabetes(one long-term and one who got it in a late age) and one without. We explained the system, use-cases and it's implemented featured. Lastly, we let the users' play around with the system before asking them questions about the general design, what benefits they believed a system like this could have for a person with diabetes and about the motivational factors implemented.

The general consensus was that the project concept was very good, and could be useful for them. With regards to the design, it seemed like they liked it, one interviewee said: "*I was surprised when I opened the (mobile)application, it looks like an application, which comes from Samsung or Google.*". The navigation and general design seem to be appalling to the people we interviewed, even with some minor bugs that were present. When asked about the benefits with regards to diabetes, they answered that the small groups were a good concept and that they could see themselves using it, with other family members or friends. Alerts when someone had a low or high BGL, was the main feature they appreciated. The social part some was unsure what to talk about, but they saw the benefits of discussing diabetes-related topics. The badges were well received, and would regarding all the interviewees help them keep motivated to improve themselves. Competition, on the other hand, had a mix reception,

8. https://github.com/facebook/react/blob/b8ba8c83f318b84e42933f6928f231dc0918f864/PATENTS
9. https://www.theregister.co.uk/2017/07/17/apache_says_no_to_facebook_code_libraries/
10. https://code.facebook.com/posts/300798627056246/relicensing-react-jest-flow-and-immutable-js/

which was expected with regards to that some like it, while other not. All the participants brought up the idea about HbA_1c competition, as a possible new feature.

The integration of physical activity was also very well received, one said: "*I've not seen an application that does this before. Physical activity is very important for us with diabetes. You keep everything in one place, BGL, chart on how it affects each other etc. This is something very many is struggling with, they do some workout and the BGL is plummeting. Here you can see the relationship between the two.*". One of the participants brought up concerns about thresholds of the alarms, he suggested that 1) it should be able to be modified by the user themselves 2) Long periods of high-blood sugar should be alerted to others. Privacy concerns about who could view the personal BGL was also brought up several times by one of the participants.

With the IoT-device it seemed like the subjects were also very positive, but it should be in an enclosure like a box or something similar. The pages were reported to be very readable, and simple to understand. With regards to the Diabetes BGL's one participant said that the chart was not needed, he argued that the most important factor was the BG value itself, and if he would just glance at the device, he would not pay attention to the chart at all. He also suggested that the BG value should be bigger, and maybe an arrow could indicate the direction of the BG could be implemented. Another said that the chart needs some more labels, so you can know what you're looking at. He also suggested allowing users to choose if they wanted a chart or an arrow/difference between the last measurements. Another important thing that came up was: *Those who have had diabetes for a long time, are depend on things should be simple, and easy to see.*. The notion to include more information to the IoT device was mixed, one was citing that the device should be something that could be glanced at instantly tell them about their own or the group members BG value, another suggested adding more screen like achievement notifications and physical activity.

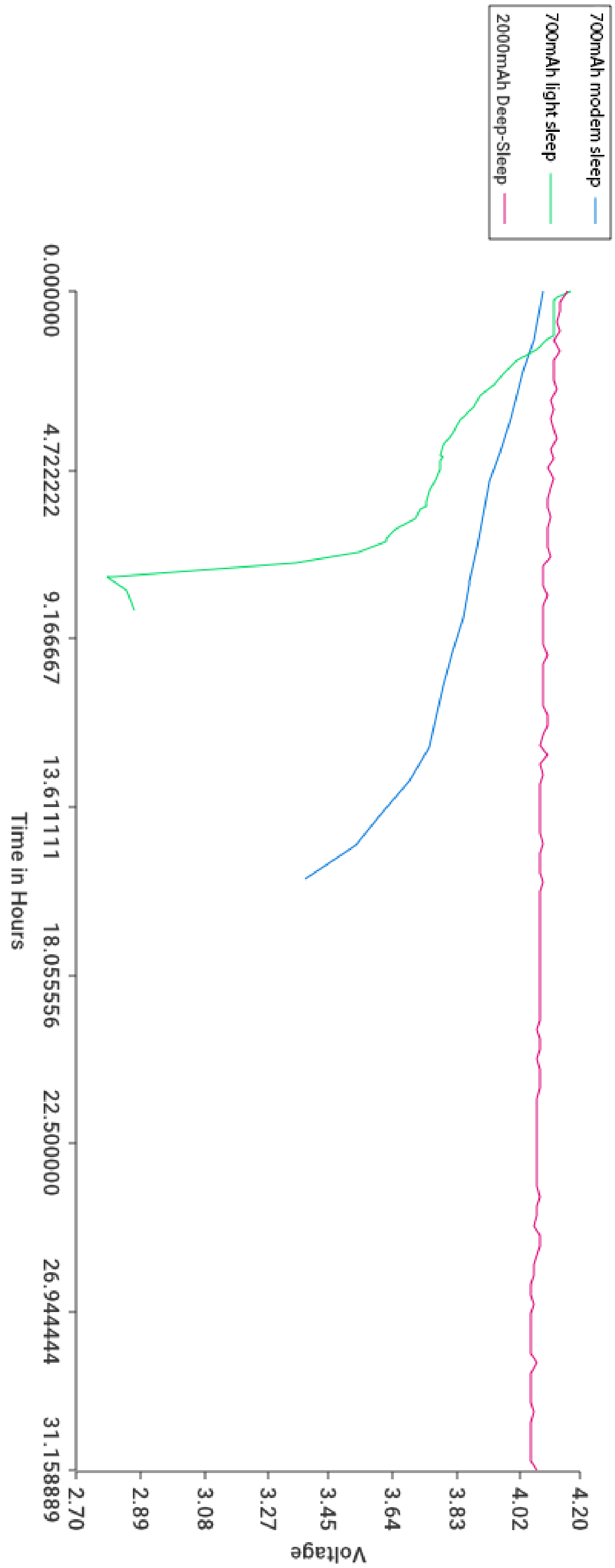

Figure 8.1: ESP8266 Discharge graph

```
Exception (29):
epc1=0x4000e1c3 epc2=0x00000000 epc3=0x00000000
excvaddr=0x00000018 depc=0x00000000

ctx: cont
sp: 3fff09e0 end: 3fff0e10 offset: 01a0

>>>stack>>>
3fff0b80:  3fff2050 00000182 40233fb0 4020bc0a
3fff0b90:  40212ced 3fff2010 00000000 ffffff80
3fff0ba0:  65736724 4020e1db 3fff2010 3fff1a90
3fff0bb0:  40212f2e 3fff2010 3fff1a90 00000000
...
3fff0da0:  3ffefba0 00000000 00000001 feefeffe
3fff0db0:  feefeffe feefeffe 3ffefd1c 3ffefddc
3fff0dc0:  3fffdad0 3ffefba0 3ffefd1c 4020238c
3fff0dd0:  feefeffe feefeffe feefeffe feefeffe
3fff0de0:  feefeffe feefeffe feefeffe 3ffefddc
3fff0df0:  3fffdad0 00000000 3ffefdd5 40206790
3fff0e00:  feefeffe feefeffe 3ffefdf0 40100114
<<<stack<<<
```

Figure 8.2: Example Stack Exception on the esp8266

9 Further Work

Throughout the project, there were new ideas, that could be implemented. Which would be useful or simply exciting and fun for the end-users. The following is some of these features/ideas, and some suggestions from the requirement interviews we conducted. Some of them are easy and could be implemented rather quickly while others require a lot of data/time to get working. In addition to these new ideas, there is also some polishing that should be done to the system as a whole. One of the main things is trying to get the all the features ported to the deep-sleep IoT implementation, but there wasn't enough time for this, but there is a plan in place to be able the achieve this. Another thing is the polish on the mobile application and the implementation of new features like mentioned in the discussion and from the feedback we got from the user-study interviews.

9.1 Common room IoT functionality

Think about an office or large common room where multiple persons which have an account and a device which can uniquely identify the user (a Bluetooth Low Energy (BLE) enable device sound very fitting for this purpose), with an installed device. Once a person comes into the room (or in range of the device to pick up the personal device) a computer will pick up the person and show the status of that person on a display which can be viewed by other. The status could be the BGL levels, the number of steps or some future addition to the system.

To make this possible first of all the screen needs to be bigger, a prime candidate would be a big screen TV which can be viewed from most of the room, therefore an MCU is mostly out of the question since they generally lack the support to output HDMI signal to drive the TV. Single-board computer (SBC) like the Raspberry Pi could be used for this application, the only requirement is that they the possible to output an image and detect devices that are in the approximation of itself. Detecting the persons could be done, like mentioned by the use of Bluetooth from for example a cheap BLE beacon or smart band/watch, where the device BLE MAC address is registered by the user and stored in the server. One can think that it's possible to use a mobile phone for the identification, but the problem is that modern smart-phones have a privacy feature which randomizes the BLE mac addresses (LE Privacy)[82], which makes it harder to track and identify the owner/wearer. To display the person's data on the screen one can design a web-interface or some application that runs on the device and cycles through the persons in the area, this should be rather simple since the server REST API is available. Another idea would be to make a public competition with leaderboards (like we have one the mobile application), we can think about many different features that could be added with this feature, like daily or weekly declaring of a winner, achievements popup when a goal is reached and so on.

9.2 BGL-goal competition

From the user-study interview, we had some good suggestion on what could be implemented to increase the motivations of the users, one of these was a competition on BGL-levels. With just a suggestion we can make a few drafts about the implementation of such a competition. Firstly, the HbA_1c measurements could be used, but these are long-time measurements, a more limited competition would be daily goals, for example, *I would like to stay under a specific value, and not have any Hypoglycemic events.* or *I would like to keep an average around x, for a whole day*. These goals could be private or shared with groups, and award badges or points on completion. The purpose would be like the step compensation to create some reward system for a healthy activity, which would not only motivate the user but also improve their general health.

9.3 Personalized suggestions/tips

Personalized suggestions and messages from the system could be motivational for the user. Let us say if a person has had a good period for a number of days in a row the system could send a motivation notification, this could be something like *Your Doing Great! Keep up the good work!*. Another example would be if the person is doing poorly, the system could send some suggestion that would make the person improve their management. This would require machine learning and/or artificial intelligence and more data to analyze, but the option is not out of reach and would be beneficial for the users.

9.4 Integration with other applications

The feature interviews revealed a need to integrate data from other applications, like the diabetes diaries, calorie-counting, and physical activity apps. There was a concern that yet another application would just increase the burden on the patient and would not be used as much. One solution to this is to use more of the NightScout functionalities since they support adding insulin, food, and physical-activity, another possibility is to interact with *diabetes dagboken*. The additional NightScout functionalities were not tested in this application because of the test-persons didn't add their food intake or log any physical activity in the app. But this is absolutely something that could be looked more into in the future.

9.5 Additional screen/images on the IoT-device

We could easily additional screen or messages types to the IoT-device, like a chart of physical activity between group-members, notifications when someone is granted a badges/achievement. There is really just the limited amount of screen space and memory. Another idea is to add a weather station screen, with inside and outside temperature, humility and so on.

9.6 Alarms

9.6.1 Mobile Application

The current alarm will appear as a normal notification with the standard sound, changing sound was tried multiple times without any success. Another approach to using SMS, to notify group members of a serious event, or even calling. Google just released Google Duplex[1], a natural sounding AI voice assistant which are able to make phone calls.

9.6.2 IoT-device

There is limited functionality to notify on the IoT-device, one option is to flash the screen when a hypo or hyper event occurs. With extra modules/parts this could be extended, for example, RGB LEDs like a traffic light (red for critical, orange for danger and green for ok) and speakers.

1. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html

10

Conclusion

We presented a working product of a system that fulfills the goals set out at the beginning of the project. With interviews and lessons learned by writing a paper on motivational factors for Mhealth apps, requirements for the project was created. These were then used to implement a working system, which is presented in this thesis. All the requirements set out was fulfilled in the system, which enables diabetics to make small groups, have competition based on physical activity, read/write messages with replies and earn badges. In addition, we also integrated CGM data into the application using NightScout, this gave the opportunity to implement alarm/notification on high- and low blood glucose values and to share these values to group-members in real-time. Motivational factors have also been included in the resulting product, which includes badges and competition. From the limited user-studies conducted we got good feedback, where they saw the real-life usability of the system. We got some feedback regarding possible improvements, modification and new features that could incorporate into the application to improve it. The end result suggests that it are attractive to potential users, with loads of opportunities to add new features and future development.

Bibliography

[1] American Diabetes Association. *American Diabetes Association Complete Guide to Diabetes*. Vol. 4. Alexandria, VA: Victor Graphics, 2005.

[2] Andrea Facchinetti. "Continuous Glucose Monitoring Sensors: Past, Present and Future Algorithmic Challenges." In: *Sensors (Basel)* 16.12 (Dec. 2016). Ed. by W. Rudolf Seitz. 27941663[pmid], p. 2093. ISSN: 1424-8220. DOI: 10.3390/s16122093. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5191073/.

[3] Bailey C. J. and Kodack M. "Patient adherence to medication requirements for therapy of type 2 diabetes." In: *International Journal of Clinical Practice* 65.3 (), pp. 314–322. DOI: 10.1111/j.1742-1241.2010.02544.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1742-1241.2010.02544.x.

[4] Wood et al. "Most Youth With Type 1 Diabetes in the T1D Exchange Clinic Registry Do Not Meet American Diabetes Association or International Society for Pediatric and Adolescent Diabetes Clinical Guidelines." In: *Diabetes Care* 36.7 (2013), pp. 2035–2037. ISSN: 0149-5992. DOI: 10.2337/dc12-1959. URL: http://care.diabetesjournals.org/content/36/7/2035.

[5] World health organization. *Diabetes – Factsheet*. 2017. URL: http://www.who.int/mediacentre/factsheets/fs312/en/index.html.

[6] Norwegian Institute of Public Health. *Diabetes in Norway*. Accessed - 10.Feb.2018. Aug. 2017. URL: https://www.fhi.no/en/op/hin/health--disease/diabetes-in-norway---public-health-/.

[7] Thomas A. Buchanan and Anny H. Xiang. "Gestational diabetes mellitus." In: *The Journal of Clinical Investigation* 115.3 (Mar. 2005), pp. 485–491. DOI: 10.1172/JCI24531. URL: https://doi.org/10.1172/JCI24531.

[8] Suzanne M. de la Monte and Jack R. Wands. "Alzheimer's Disease Is Type 3 Diabetes-Evidence Reviewed." In: *J Diabetes Sci Technol* 2.6 (Nov. 2008). 19885299[pmid], pp. 1101–1113. ISSN: 1932-2968. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2769828/.

[9] European Medicines Agency. "Guideline on clinical investigation of medicinal products in the treatment or prevention of diabetes mellitus."

In: (May 2012). URL: http://www.ema.europa.eu/docs/en_GB/document_library/Scientific_guideline/2012/06/WC500129256.pdf.

[10] Abdulfatai B. Olokoba, Olusegun A. Obateru, and Lateefat B. Olokoba. “Type 2 Diabetes Mellitus: A Review of Current Trends.” In: *Oman Med J* 27.4 (July 2012). 23071876[pmid], pp. 269–273. ISSN: 1999-768X. DOI: 10.5001/omj.2012.68. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3464757/.

[11] Centers for Disease Control and Prevention (CDC). “Prevalence of overweight and obesity among adults with diagnosed diabetes–United States, 1988-1994 and 1999-2002.” In: *Morbidity and Mortality Weekly Report* (Nov. 2004).

[12] National Institute of Diabetes, Digestive, and Kidney Diseases. *Risk Factors for Type 2 Diabetes*. Access - 14.05.2018. Nov. 2016. URL: https://www.niddk.nih.gov/health-information/diabetes/overview/risk-factors-type-2-diabetes.

[13] Philip E Cryer. “Iatrogenic Hypoglycemia as a Cause of Hypoglycemia-Associated Autonomic Failure in IDDM: A Vicious Cycle.” In: *Diabetes* 41.3 (1992), pp. 255–260. ISSN: 0012-1797. DOI: 10.2337/diab.41.3.255. URL: http://diabetes.diabetesjournals.org/content/41/3/255.

[14] Nitil Kedia. “Treatment of severe diabetic hypoglycemia with glucagon: an underutilized therapeutic approach.” In: *Diabetes Metab Syndr Obes* 4 (Sept. 2011). 21969805[pmid], pp. 337–346. ISSN: 1178-7007. DOI: 10.2147/DMSO.S20633. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3180523/.

[15] Lawrence C. Perlmuter et al. “Glycemic Control and Hypoglycemia: Is the loser the winner?” In: *Diabetes Care* 31.10 (Oct. 2008). 18820231[pmid], pp. 2072–2076. ISSN: 0149-5992. DOI: 10.2337/dc08-1441. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2551657/.

[16] Koltin Dror and Daneman Denis. “‘Dead-in-bed’ syndrome – a diabetes nightmare.” In: *Pediatric Diabetes* 9.5 (), pp. 504–507. DOI: 10.1111/j.1399-5448.2008.00404.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1399-5448.2008.00404.x.

[17] Sandra E. Olsen et al. “Impaired Awareness of Hypoglycemia in Adults With Type 1 Diabetes Is Not Associated With Autonomic Dysfunction or Peripheral Neuropathy.” In: *Diabetes Care* 39.3 (2016), pp. 426–433. ISSN: 0149-5992. DOI: 10.2337/dc15-1469. URL: http://care.diabetesjournals.org/content/39/3/426.

[18] Thomas J. Songer and Rashida R. Dorsey. “High Risk Characteristics for Motor Vehicle Crashes in Persons with Diabetes by Age.” In: *Annu Proc Assoc Adv Automot Med* 50 (2006). 16968646[pmid], pp. 335–351. ISSN: 1540-0360. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3217477/.

[19] Stewart Harris et al. “The Effect of Hypoglycemia on Health-Related Quality of Life: Canadian Results from a Multinational Time Trade-off

Survey." In: *Canadian Journal of Diabetes* 38.1 (Feb. 2014), pp. 45–52. ISSN: 1499-2671. DOI: 10.1016/j.jcjd.2013.09.001. URL: http://dx.doi.org/10.1016/j.jcjd.2013.09.001.

[20] American Diabetes Association. *Hyperglycemia (High Blood Glucose)*. Accessed 10.05.2018. URL: http://www.diabetes.org/living-with-diabetes/treatment-and-care/blood-glucose-control/hyperglycemia.html.

[21] Dorothee Deiss et al. "Improved Glycemic Control in Poorly Controlled Patients with Type 1 Diabetes Using Real-Time Continuous Glucose Monitoring." In: *Diabetes Care* 29.12 (2006), pp. 2730–2732. ISSN: 0149-5992. DOI: 10.2337/dc06-1134. URL: http://care.diabetesjournals.org/content/29/12/2730.

[22] Tamborlane et al. "Continuous Glucose Monitoring and Intensive Treatment of Type 1 Diabetes." In: *New England Journal of Medicine* 359.14 (2008). PMID: 18779236, pp. 1464–1476. DOI: 10.1056/NEJMoa0805017. URL: https://doi.org/10.1056/NEJMoa0805017.

[23] Insabella Glendessa et al. "The transition to young adulthood in youth with type 1 diabetes on intensive treatment." In: *Pediatric Diabetes* 8.4 (), pp. 228–234. DOI: 10.1111/j.1399-5448.2007.00266.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1399-5448.2007.00266.x.

[24] John C Pickup, Suzanne C Freeman, and Alex J Sutton. "Glycaemic control in type 1 diabetes during real time continuous glucose monitoring compared with self monitoring of blood glucose: meta-analysis of randomised controlled trials using individual patient data." In: *BMJ* 343 (2011). ISSN: 0959-8138. DOI: 10.1136/bmj.d3805. URL: https://www.bmj.com/content/343/bmj.d3805.

[25] Davidson MB. "Continuous glucose monitoring in patients with type 1 diabetes taking insulin injections." In: *JAMA* 317.4 (2017), pp. 363–364. DOI: 10.1001/jama.2016.20327. URL: +%20http://dx.doi.org/10.1001/jama.2016.20327.

[26] Günther Schmelzeisen-Redeker et al. "Time Delay of CGM Sensors: Relevance, Causes, and Countermeasures." In: *J Diabetes Sci Technol* 9.5 (Sept. 2015). 26243773[pmid], pp. 1006–1015. ISSN: 1932-2968. DOI: 10.1177/1932296815590154. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4667340/.

[27] Eray Kulcu et al. "Physiological Differences Between Interstitial Glucose and Blood Glucose Measured in Human Subjects." In: *Diabetes Care* 26.8 (2003), pp. 2405–2409. ISSN: 0149-5992. DOI: 10.2337/diacare.26.8.2405. URL: http://care.diabetesjournals.org/content/26/8/2405.

[28] Ananda Basu et al. "Time Lag of Glucose From Intravascular to Interstitial Compartment in Humans." In: *Diabetes* 62.12 (Dec. 2013). 24009261[pmid], pp. 4083–4087. ISSN: 0012-1797. DOI: 10.2337/db13-1132. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3837059/.

[29] Ananda Basu et al. "Time Lag of Glucose From Intravascular to Interstitial Compartment in Type 1 Diabetes." In: *J Diabetes Sci Technol* 9.1 (Jan. 2015). 25305282[pmid], pp. 63–68. ISSN: 1932-2968. DOI: 10.1177/1932296814554797. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4495531/.

[30] Florian Reiterer et al. "Identification of CGM Time Delays and Implications for BG Control in T1DM." In: *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016*. Ed. by Efthyvoulos Kyriacou, Stelios Christofides, and Constantinos S. Pattichis. Cham: Springer International Publishing, 2016, pp. 190–195. ISBN: 978-3-319-32703-7.

[31] D. Barry Keenan et al. "Delays in Minimally Invasive Continuous Glucose Monitoring Devices: A Review of Current Technology." In: *J Diabetes Sci Technol* 3.5 (Sept. 2009). 20144438[pmid], pp. 1207–1214. ISSN: 1932-2968. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2769894/.

[32] J. C. Pickup et al. "Continuous subcutaneous insulin infusion: an approach to achieving normoglycaemia." In: *Br Med J* 1.6107 (Jan. 1978). 340000[pmid], pp. 204–207. ISSN: 0007-1447. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1602534/.

[33] Eirik Arsand. "The few touch digital diabetes diary : user-involved design of mobile self-help tools for people with diabetes." In: (June 2009).

[34] Louise B Russell, Dong-Churl Suh, and Monika Safford. "Time requirements for diabetes self-management: Too much for many?" In: 54 (Feb. 2005), pp. 52–6.

[35] Ryan J. Anderson et al. "The Prevalence of Comorbid Depression in Adults With Diabetes." In: *Diabetes Care* 24.6 (2001), pp. 1069–1078. ISSN: 0149-5992. DOI: 10.2337/diacare.24.6.1069. URL: http://care.diabetesjournals.org/content/24/6/1069.

[36] Lawrence Fisher et al. "Clinical Depression Versus Distress Among Patients With Type 2 Diabetes." In: *Diabetes Care* 30.3 (2007), pp. 542–548. ISSN: 0149-5992. DOI: 10.2337/dc06-1614. URL: http://care.diabetesjournals.org/content/30/3/542.

[37] Albert Bandura. "Self-Efficacy." In: *Encyclopedia of human behavior* (1994), pp. 71–81.

[38] Caroline Senécal and David Nouwen Arieand White. "Motivation and dietary self-care in adults with diabetes: Are self-efficacy and autonomous self-regulation complementary or competing constructs?" In: *Health Psychology* 19.5 (2000), pp. 452–457. DOI: 10.1037/0278-6133.19.5.452.

[39] Geoffrey Williams, Z R Freedman, and Edward Deci. "Supporting Autonomy to Motivate Patients With Diabetes for Glucose Control." In: 21 (Oct. 1998), pp. 1644–51.

[40] Richard M. Ryan and Edward L. Deci. "Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development and Well-

Begin." In: *American Psychologist* 55.1 (2000), pp. 68–78. DOI: 10.1037//0003-066X,55.1.68.

[41] Edward Deci. *Self-Determination Theory*. Accessed - 24.05.2018. URL: https://www.youtube.com/watch?v=m6fm1gt5YAM.

[42] Robert J. Vallerand. "TOWARD A HIERARCHICAL MODEL OF INTRINSIC AND EXTRINSIC MOTIVATION." In: *Advances in Experimental Social Psychology* 29 (1997).

[43] Michael Sailer et al. "How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction." In: *Computers in Human Behavior* 69 (2017), pp. 371–380. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2016.12.033. URL: http://www.sciencedirect.com/science/article/pii/S074756321630855X.

[44] Juho Hamari. "Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service." In: *Electronic Commerce Research and Applications* 12.4 (2013), pp. 236–245. ISSN: 1567-4223. URL: http://www.sciencedirect.com/science/article/pii/S1567422313000112.

[45] Juho Hamari, Jonna Koivisto, and Harri Sarsa. "Does Gamification Work? — A Literature Review of Empirical Studies on Gamification." In: *Proceedings of the Annual Hawaii International Conference on System Sciences* (Jan. 2014).

[46] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A Survey." In: *Comput. Netw.* 54.15 (Oct. 2010), pp. 2787–2805. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010. URL: http://dx.doi.org/10.1016/j.comnet.2010.05.010.

[47] Kevin Ashton. *That 'Internet of Things' Thing. In the real world, things matter more than ideas.* Ed. by rfidjournal. [Accessed Nov. 8, 2017]. June 2009. URL: http://www.rfidjournal.com/articles/view?4986.

[48] S. H. Shah and I. Yaqoob. "A survey: Internet of Things (IOT) technologies, applications and challenges." In: *2016 IEEE Smart Energy Grid Engineering (SEGE)*. Aug. 2016, pp. 381–385. DOI: 10.1109/SEGE.2016.7589556.

[49] Scott Hendrickson et al. "Serverless Computation with openLambda." In: *Proceedings of the 8th USENIX Conference on Hot Topics in Cloud Computing*. HotCloud'16. Denver, CO: USENIX Association, 2016, pp. 33–39. URL: http://dl.acm.org/citation.cfm?id=3027041.3027047.

[50] D. Bernstein. "Containers and Cloud: From LXC to Docker to Kubernetes." In: *IEEE Cloud Computing* 1.3 (Sept. 2014), pp. 81–84. ISSN: 2325-6095. DOI: 10.1109/MCC.2014.51.

[51] D. Jaramillo, D. V. Nguyen, and R. Smart. "Leveraging microservices architecture by using Docker technology." In: *SoutheastCon 2016*. Mar. 2016, pp. 1–5. DOI: 10.1109/SECON.2016.7506647.

[52] Docker Inc. *About Docker Engine*. [Accessed Nov. 30, 2017]. URL: https://docs.docker.com/engine/#docker-user-guide.

[53] Docker Inc. *Docker Overview*. [Accessed May. 21, 2017]. URL: https://docs.docker.com/engine/docker-overview/.

[54] Andy Wu. "Taking the Cloud-Native Approach with Microservices." In: (2017). URL: https://cloud.google.com/files/Cloud-native-approach-with-microservices.pdf.

[55] James Lewis and Martin Fowler. *Microservices, a definition of this new architectural term*. [Accessed Nov. 29, 2017]. Mar. 2014. URL: https://martinfowler.com/articles/microservices.html.

[56] Chris Richardson. *Pattern: Microservice Architecture*. Accessed - 24.05.2018. 2017. URL: http://microservices.io/patterns/microservices.html.

[57] Eric S. Raymond. *The Art of UNIX Programming*. Pearson Education, 2003. ISBN: 0131429019.

[58] MQTT.org. *HomeSite*. [Accessed Nov. 29, 2017]. URL: http://mqtt.org/.

[59] Valerie Lampkin. *What is MQTT and how does it work with WebSphere MQ?* Mar. 2012. URL: https://www.ibm.com/developerworks/mydeveloperworks/blogs/aimsupport/entry/what_is_mqtt_and_how_does_it_work_with_websphere_mq?lang=en.

[60] HiveMQ. *MQTT Essentials Part 8: Retained Messages*. [Accessed Nov. 30, 2017]. URL: https://www.hivemq.com/blog/mqtt-essentials-part-8-retained-messages.

[61] MongoDB inc. *MongoDB Architecture*. Accessed April.04.2018. URL: https://www.mongodb.com/mongodb-architecture.

[62] C. Győrödi et al. "A comparative study: MongoDB vs. MySQL." In: *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*. June 2015, pp. 1–6. DOI: 10.1109/EMES.2015.7158433.

[63] Nightscout Project. *Nightscout - we are not waiting*. [Accessed Nov. 28, 2017]. URL: http://www.nightscout.info/.

[64] Lee JM, Hirschfeld E, and Wedding J. "A patient-designed do-it-yourself mobile technology system for diabetes: Promise and challenges for a new era in medicine." In: *JAMA* 315.14 (2016), pp. 1447–1448. DOI: 10.1001/jama.2016.1903. URL: +%20http://dx.doi.org/10.1001/jama.2016.1903.

[65] *ESP8266EX DataSheet*. [Accessed Nov. 12, 2017]. Espressif inc. URL: https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.

[66] R.Jayalakshmi Adithya. Potu and Dr.K.Umpathy. "Smart Paper Technology a Review Based On Concepts of E-Paper Technology." In: *IOSR Journal of Electronics and Communication Engineering(IOSR-JECE)* 1 (Jan. 2017). ISSN: 2278-8735. URL: http://www.iosrjournals.org/iosr-jece/papers/Vol.%2011%20Issue%201/Version-1/G011114246.pdf.

[67] E Ink Holdings Inc. *Electronic Ink*. [Accessed Des. 8, 2017]. URL: E%20Ink%20Holdings%20Inc.

[68] Peng Fei Bai et al. "Review of paper-like display technologies (invited review)." In: *Progress In Electromagnetics Research* 147 (2014), pp. 95–116.

[69] André Henriksen et al. "Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables." In: *J Med Internet Res* 20.3 (Mar. 2018). Ed. by Gunther Eysenbach. 29567635[pmid], e110. ISSN: 1439-4456. DOI: 10.2196/jmir.9157. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5887043/.

[70] Fatema El-Amrawy and Mohamed Ismail Nounou. "Are Currently Available Wearable Devices for Activity Tracking and Heart Rate Monitoring Accurate, Precise, and Medically Beneficial?" In: *Healthc Inform Res* 21.4 (Oct. 2015). 26618039[pmid], pp. 315–320. ISSN: 2093-3681. DOI: 10.4258/hir.2015.21.4.315. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4659890/.

[71] B. Kaliski. *PKCS #5: Password-Based Cryptography Specification Version 2.0*. United States, 2000.

[72] Jina Huh and Mark S. Ackerman. "Collaborative Help in Chronic Disease Management: Supporting Individualized Problems." In: *CSCW* 2012 (2012). 25360442[pmid], pp. 853–862. DOI: 10.1145/2145204.2145331. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4211623/.

[73] Robert E. Slavin. "Research on cooperative learning and achievement: What we know, what we need to know." In: *Contemporary Educational Psychology* 21.1 (1996), pp. 43–69. ISSN: 0361-476X(Print). DOI: 10.1006/ceps.1996.0004.

[74] Bethany Tennant et al. "eHealth Literacy and Web 2.0 Health Information Seeking Behaviors Among Baby Boomers and Older Adults." In: *J Med Internet Res* 17.3 (Mar. 2015), e70. DOI: 10.2196/jmir.3992. URL: http://www.ncbi.nlm.nih.gov/pubmed/25783036.

[75] Robert E. Franken and Douglas J. Brown. "Why do people like competition? The motivation for winning, putting forth effort, improving one's performance, performing well, being instrumental, and expressing forceful/aggressive behavior." In: *Personality and Individual Differences* 19.2 (1995), pp. 175–184. ISSN: 0191-8869. DOI: https://doi.org/10.1016/0191-8869(95)00035-5. URL: http://www.sciencedirect.com/science/article/pii/0191886995000355.

[76] Deepak Malhotra. "The desire to win: The effects of competitive arousal on motivation and behavior." In: *Organizational Behavior and Human Decision Processes* 111.2 (2010), pp. 139–146. ISSN: 0749-5978. DOI: https://doi.org/10.1016/j.obhdp.2009.11.005. URL: http://www.sciencedirect.com/science/article/pii/S074959780900106X.

[77] Barbara J. Drew et al. "Insights into the Problem of Alarm Fatigue with Physiologic Monitor Devices: A Comprehensive Observational Study of Consecutive Intensive Care Unit Patients." In: *PLOS ONE* 9.10 (Oct.

2014), pp. 1–23. DOI: 10.1371/journal.pone.0110274. URL: https://doi.org/10.1371/journal.pone.0110274.

[78] *ESP8266 Low-Power Solutions*. [Accessed Nov. 12, 2017]. Espressif inc. URL: https://espressif.com/sites/default/files/documentation/9b-esp8266-low_power_solutions_en.pdf.

[79] *SpiFlash. 8M-BIT, 16M-BIT AND 32M-BIT SERIAL FLASH MEMORY WITH DUAL AND QUAD SPI*. [Accessed Nov. 12, 2017]. Winbond. URL: https://elinux.org/images/f/f5/Winbond-w25q32.pdf.

[80] *ESP32 Datasheet*. [Accessed Des. 8, 2017]|. Espressif Systems inc. Dec. 2017. URL: http://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[81] Ltd. Zhongshan beidoustar Network Communication Technology Co. *RTL8710 wifi module specification V2.1*. Accessed 10.05.2018. Nov. 2016. URL: https://github.com/SeeedDocument/Bazaar_doc/raw/master/113990294/res/RTL8710%C2%A0wifi%C2%A0module%C2%A0specification%C2%A0V2.1%C2%A0.pdf.

[82] Scott Lester and Paul Stone. *Bluetooth LE - Increasingly popular, but still not very private*. Accessed April.29.2018. Apr. 2016. URL: https://www.contextis.com/blog/bluetooth-le-increasingly-popular-still-not-very-private.

Appendices

Motivational factors for user engagement with mHealth apps

Ashenafi Zebene WOLDAREGAY[a,*], David-Zacharie ISSOM[b,*], André HENRIKSEN[c,*,1], Henna MARTTILA[d], Martin MIKALSEN[a], Gerit PFUHL[e], Keiichi SATO[a,f], Christian LOVIS[b,g], and Gunnar HARTVIGSEN[a]

[a] *Department of Computer Science, UiT The Arctic University of Norway*

[b] *Geneva University Hospitals, Division of Medical Information Sciences, Switzerland*

[c] *Department of Community Medicine, UiT The Arctic University of Norway*

[d] *University of Lapland, Faculty of Art and Design, Rovaniemi, Finland*

[e] *Department of Psychology, UiT The Arctic University of Norway*

[f] *Institute of Design, Illinois Institute of Technology, USA*

[g] *University of Geneva, Switzerland*

Abstract. The widespread adoption of smartphones creates an enormous potential to improve healthcare services. Numerous apps, sensors, and devices are developed for health self-management purposes. However, adoption rates remain low and long-term user engagement is a major issue. The goal of this study is to identify major motivational factors that can facilitate prolonged use of mobile health systems. To this end, we conducted 16 interviews with representatives of various cultural backgrounds, disease history, age, and gender. Participants' experiences indicated that existing systems were unable to answer their self-management needs properly. People with a disease history favored learning from data, as well as from others via social media integration. People without chronic disease felt more reserved about social media integration. In conclusion, systems that collect and share personal data should have a clear opt-in or opt-out option to motivate usage. Additionally, researchers and mobile health system developers could achieve long-term adoption by giving clear answers to privacy and trust issues, while offering people strong added value according to their individual needs.

Keywords. Wearables, self-management, diabetes, sickle-cell disease, physical activity, user engagement, motivation

1. Introduction

The widespread adoption of smartphones has created an enormous potential to improve healthcare services. Currently**,** mobile health (mHealth) apps created by researchers, the health industry, ICT device manufacturers, and private citizens, are surging on the market. However, adoption rates of these technologies appear to be low. Presumably, this is due to a lack of motivation from most users to use these apps for a prolonged period [1]. Some studies indicate that lack of personalization can reduce long-term engagement [2, 3]. According to Eccles & Wigfield, user expectations and values could directly affect

* Authors contributed equally

[1] Corresponding author, André Henriksen, UiT The Arctic University of Norway, Postboks 6050 Langnes, 9037 Tromsø, Norway; E-mail: andre.henriksen@uit.no.

individual performance, persistence, and task selection [4]. By considering appropriate factors of motivation, such systems could be aligned with users' individual health behaviors, needs, and goals [5]. In this study, we will focus on people without chronic disease, people with type 1 diabetes, and people with sickle cell disease (SCD).

Diabetes mellitus is a chronic disease that affects blood glucose metabolism resulting in lack of blood glucose (BG) control. People with diabetes must maintain an optimal BG level to avoid long-term complications [10]. Adherence to these routines is not always easy, even though various technological solutions exist to empower them to better self-manage. However, many people seem to lack motivation in using these solutions for a prolonged time [11]. SCD is the most neglected [6], yet very deadly non-communicable disease [7, 8]. SCD is one of the leading causes of death worldwide and is characterized by severe complications. However, simple health interventions [12, 13] (e.g. early diagnosis, antibiotic prophylaxis, and comprehensive care), and the conduct of a healthy lifestyle (e.g. avoid extreme heat and cold, pay attention to diet, stay hydrated, take vitamins supplements, exercise moderately, rest sufficiently, practice moderate physical activity, take adequate medication, and avoid stress), can largely improve health outcomes by reducing morbidity and mortality [9]. People with SCD could benefit from such self-management technologies [14]. Moderate physical activity (MPA) play an important role in reducing adverse effects of many chronic diseases, as well as support a healthy lifestyle, e.g. helps with depression [15].

Motivational challenges regarding mHealth usage can be viewed from the user perspectives (i.e. unmet needs) and the way these technologies are developed (e.g. simplicity, usability, better functionalities, and others) [16]. Mendiola et al. [17] pointed out general mHealth features that contributed positively to user ratings, including features that save time, are intuitive and straightforward, provides specific instructions for better management, and can share data with designated individuals.

In regards to diabetes, Jung et al. [16] identified the following features as the key unmet needs in existing mHealth apps: information feedback, simple and usable reports, and communication capability. Moreover, Scheibe et al. [18] identified lack of additional benefits, individually tailored features, and ease of use as crucial discouragement factors. Furthermore, Lithgow et al. [19] found that the lack of glucometer or insulin pump integration with existing mHealth apps, is the leading factors for dissatisfaction among people with type 1 diabetes. Regarding SCD, there are few apps and studies available, and as far as we are aware, this will be the first SCD related qualitative study on mHealth motivation. Concerning physical activity, Clemens et.al [20] found a correlation between health app usage and motivation, when the app is used for physical activity planning, monitoring, and feedback. A recent meta-analysis supports these findings, as they found monitoring to be an effective motivational factor [21].

The goal of this study was to investigate motivational factors in the following scenarios: 1) wearables and sensors, 2) data sharing, 3) data integration, and 4) social media and entertainment. We also aimed to identify major motivational factors that lead to enhanced long-term user engagement, and to motivate users to share privately collected health data on a cloud repository for secondary usage. In this paper, we present the results from our qualitative study, which will be a basis for a larger study with more participants. This paper is organized as follows; section 2 describes the experimental framework and the methodology used to evaluate responses. Section 3 gives details about the obtained results, and is divided into three parts according to patient's health status. Section 4 discusses the results and interprets it in light of the selected theory. Finally, section 5 present our conclusions.

2. Method

The four key motivational factors proposed by Locke [22] are goals and intentions, needs, values, and emotions. To assess the user motivations regarding these scenarios, we used Locke's indicators of motivation to identify participants' needs, values (knowledge and experience) and expectations. Furthermore, we interpreted the results using Eccles' [4] theory of motivation, using task-value beliefs and expectancy-related factors.

We conducted in-person interviews with 16 participants. Ten participants had SCD, 4 females and 6 males, aged between 21 and 55. Two participants had type 1 diabetes, 1 female and 1 male, aged between 30 and 32. Four participants had no chronic disease, 3 females and 1 male, aged between 26 and 50. Participants were from Norway, Finland, and Switzerland. Participants were recruited using convenience sampling. However, we took care to recruit people with diverse age, gender, cultural background, and disease status.

We developed a common interview guide, which was divided into five themes: 1) goals, attitude and expectations, 2) wearables and sensors, 3) data sharing, 4) data integration, and 5) social media and entertainment factors. We used inductive thematic analysis to categorize participants' responses into three sub-themes: 1) knowledge, 2) experience, and 3) expectations. We extracted keywords from the interviews and used them to categorize responses into themes and sub-themes. Finally, we identified the main motivational factors for each participants' group and identified similarities and differences among them. The Norwegian Regional Committees for Medical and Health Research Ethics reviewed the study protocol and concluded that no ethical approval was necessary. All participants gave informed consent.

3. Results

3.1. Diabetes

Regarding relevant *knowledge*, participants showed lack of proficiency concerning usability and functionality of diabetes-related technologies, including apps, sensors, and wearables, e.g., "*There are many apps and wearables, but not conscious of which brand and apps so I need someone to tell me what is good for diabetes, why and how to use it".* Regarding data sharing and integration, participants reported lack of trust and fear of data breaches as their main concerns.

Regarding hands-on *experiences*, participants reported lack of regular use of self-management apps, sensors, and wearable devices. As such, one participant said, *"I found it difficult to choose among the available apps, and I would need recommendations".* Sensors and wearables equipped with automatic logging were favored. However, manual logging is dependent on the frequency and time of execution and must be easy and quick to do, e.g., *"What is missing in most of diabetes app is automatic indication or integration of sensors that automatically collect data".* The participants positively evaluated social media experiences, i.e. ability to share experiences (tips and tricks), getting knowledge inaccessible from other places.

The participants indicated that diabetes should not affect their life too much, and need relief from their self-management responsibilities, emphasizing automatic and intelligent systems that could take over their tasks. One participant said, *"I think the* *biggest problem is probably food and how to get it right, something that could track how different kind of foods affects blood sugar and give recommendations would be nice".* Generally, expectations were about ease of use, quality information, intelligence, and indiscernibility of self-management systems for diabetes. Sensors and wearables should provide timely feedbacks, with frequencies based on their necessities. Participants reported willingness to send their data to shared public repositories for further analysis purposes. However, this was conditional that confidentiality is guaranteed and that data are useful for them as well as for the public. Expectations are about services, such as data analysis and integrated views (e.g. infection detection capability, depth, and relevance), immediate feedback and notifications, and decision-support features. In this respect, one participant said, *"Sometimes your blood glucose might be too high but you don't know what exactly is wrong. So, it would be nice if I could get information about what is wrong and then I can do something about that."* Regarding social media and entertainment factors, they preferred to chat and discuss some issues along with the capability of user-specified sharing of fitness and health data. In this regard, one participant said, *"It would be nice if I can have features on diabetes mobile app that I can share data with my friends so that we can chat and discuss issues"*. Furthermore, participants preferred features like anonymous data sharing to learn from each other, sharing tips and features that could monitor and compare average BG levels with peers in the community. As to this one participants said, *"Something that monitors average blood glucose values, and shows us how we do compared to others, e.g. comparison game, would actually be interesting and motivating".*

3.2. Sickle Cell Disease

Regarding participants' *knowledge*, all reported needing help for their self-management related tasks. However, they would prefer if an invisible technology gave this help. As a participant expressed, "*Not too intrusive sensors that can help to see when something is going bad, before it's too late and before I'd need to go to hospital"*. They wished to learn more about how to self-care, e.g. *"It should release my health-related tasks, accompany me with the disease, the pain, the risks I take day-to-day, so prevention".* Few participants knew relevant mHealth apps or wearable devices. Three people reported using sensors for self-monitoring purposes. In contrast, all participants showed interest in learning about the potential triggers of health deterioration and found the possibility of tracking hematological data very important. Some participants described privacy issues as central, mostly regarding access and circulation of data, e.g. *"I want to be able to choose where data go".*

Participants had limited *experience* using mHealth systems. However, they shared the difficulty of living with their disease and managing the limitations it imposes on their life. They highlighted the inability of existing systems to support their needs, i.e. *"Everything is adapted to normal people, nothing is adapted to people with SCD needs".* In their opinion, current mHealth platforms were not giving them relevant and tailored feedback. They found existing systems unable to help them during their daily activities and not personalized enough. As one participant said, *"It must be very personalized, it's easy to find things on the Internet, but it's mostly for normal people"*. Indeed, most interviewees were not using mHealth apps. They stated having no knowledge of relevant apps for managing their disease or judged them irrelevant. All participants reported using social media for information sharing, communication and entertainment with people at remote locations. Only one participant used social media for talking about his health.

Others considered information on social networks as not always accurate, badly structured, and felt that discussions could be violent, while lacking consideration for their privacy. This was well illustrated by two participants' statements, e.g. *"Sadly it's often submerged by information and chaotic.",* and *I see also privacy problems, some voyeurism"*.

All participants had very clear *expectations* and showed similar motivational factors. They highlighted the wish for using a tool capable of supporting their health-related tasks, e.g. "*Anything that helps to anticipate health deterioration is good"* and *"My motivation is the usefulness of the system only. But why not a quiz for improving my health literacy"*. Participants were also very clear about the importance of getting trustworthy information and timely self-care information. By this, they also showed their motivation for using a system that responds to their needs and *"adds a clear benefit"*.

3.3. Physical activity

Knowledge about digital solutions (e.g. wearables and mobile apps) in the PA group varied significantly. However, the consensus was that PA solutions were easy to use, but how the collected data should be used, was not always clear. The two main topics emerging from participants were concerns about privacy and data ownership. Privacy issues were mostly related to data storage and a feeling of being under constant surveillance. They still used these systems, mostly because they add value for them. Several participants showed strong concerns about the usage of their data by service providers without their control, e.g., "*I am the source of their income (data), but I usually don't get anything in return. I don't control how it is used*".

PA apps can collect steps and show basic analysis, but the *experience* from participants was that data analysis is less than optimal, e.g. "*I have a lot of data, but should have some app that allowed me to better analyze this".* Another issue was the wasting of time related to social media, and that oversharing should be considered before integrating social media features in any application.

Expectations towards future apps were high among participants. Most expect some gamification features. However, the main findings indicated two major sources of motivation and frustration: 1) issues regarding data security, anonymity, and how others use the data, and 2) the limited quality of what they get in return from sharing personal data. Both issues were raised several times during the interviews.

4. Discussion

According to Eccles' theory of motivations, tasks-values and outcome expectations towards any system, are the main factors of motivation for long time engagement. The task-values are mainly categorized as cost, utility value, attainment value, and intrinsic value.

We identified several *cost-related* values, where the primary motivational issues were the amount of time needed to learn and use new mHealth systems. Moreover, trust and privacy issues, along with the non-indiscernibility of the system, and lack of optimal data analysis, were equally important factors. Furthermore, regarding social media usage, inaccurate and poorly structured information, along with oversharing and wasting time, were also main concerns.

Regarding *utility* values, mHealth systems that provide easy ways to learn better self-care and avoid health deterioration are favored most. In addition, concerning social media usage, capabilities such as sharing information, experiences, and advice, are also important driving factors for prolonged engagement.

Expectations could directly affect individual's motivation to engage with mHealth systems. In this regard, the main *expectations* are that the system should be beneficial for them and can provide high-quality data and an integrated overview. Additionally, it must provide timely self-management information and feedback. Furthermore, participant willingness to engage with a system is affected by its trustworthiness, ability to provide data security, and anonymity of data sharing. Expectations towards various features included tailored features support, gamification, and decision support.

The major limitation of the study is the uneven distributions of the sample size among each group of study participants, which might somewhat bias the comparisons of motivational factors drawn among the groups.

5. Conclusion

Despite the interviewees' different backgrounds (i.e. age, gender, disease, ethnicity, and culture), *cost* related values such as privacy concerns and data access issues were crucial for most of them. Moreover, they also criticized the imprecision of data analyses (i.e. depth and relevance). *Utility* related values such as entertainment factors, e.g. comparison with peers and quizzes, were favored. Moreover, responding to daily self-management needs was a crucial requirement for all groups. People with diabetes or SCD (i.e. chronic disease) found social media integration to be an important motivation factor, while people without chronic disease did not. Furthermore, the former groups seemed to value learning from data, as well as comparing their data and their knowledge with other people. However, they criticized inconsistency regarding the quality of information.

Some important factors of motivation stood out from these preliminary results. To gain long-term adoption, it appears that mHealth systems must give clear answers to privacy and trust concerns, offer clear benefits, target individual needs and provide very high-quality information (i.e. trustworthiness, data analysis, organization, and timeliness). To increase adoption rates, researchers and developers should put efforts in meeting these criteria, for instance by systematically asking people about their needs and having easy opt-in, opt-out options for various functionalities of the mHealth app, respectively.

References

1. Krebs, P. and D.T. Duncan, *Health App Use Among US Mobile Phone Owners: A National Survey.* JMIR Mhealth Uhealth, 2015. **3**(4): p. e101.
2. Azhar, F.A.B. and J.S. Dhillon. *A systematic review of factors influencing the effective use of mHealth apps for self-care.* in *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. 2016.
3. Iacoviello, M.B., et al., *Clickotine, A Personalized Smartphone App for Smoking Cessation: Initial Evaluation.* JMIR Mhealth Uhealth, 2017. **5**(4): p. e56.
4. Eccles, J.S. and A. Wigfield, *Motivational beliefs, values, and goals.* Annu Rev Psychol, 2002. **53**: p. 109-32.
5. Anderson, K., O. Burford, and L. Emmerton, *Mobile Health Apps to Facilitate Self-Care: A Qualitative Study of User Experiences.* PLOS ONE, 2016. **11**(5): p. e0156164.

6. Piel, F.B., et al., *Global distribution of the sickle cell gene and geographical confirmation of the malaria hypothesis.* Nat Commun, 2010. **1**: p. 104.
7. Lopez, A.D., et al., *Remembering the forgotten non-communicable diseases.* BMC Medicine, 2014. **12**(1): p. 200.
8. Chakravorty, S. and T.N. Williams, *Sickle cell disease: a neglected chronic disease of increasing global health importance.* Archives of Disease in Childhood, 2014.
9. Matthie, N., C. Jenerette, and S. McMillan, *The Role of Self-Care in Sickle Cell Disease.* Pain management nursing : official journal of the American Society of Pain Management Nurses, 2015. **16**(3): p. 257-266.
10. Arsand, E., et al., *Mobile health applications to assist patients with diabetes: lessons learned and design implications.* J Diabetes Sci Technol, 2012. **6**(5): p. 1197-1206.
11. Shibuta, T., et al., *Willingness of patients with diabetes to use an ICT-based self-management tool: a cross-sectional study.* BMJ Open Diabetes Res Care, 2017. **5**(1): p. e000322.
12. Tanabe, P., et al., *A qualitative analysis of best self-management practices: sickle cell disease.* J Natl Med Assoc, 2010. **102**(11): p. 1033-41.
13. Vacca, V.M.J. and L. Blank, *Sickle cell disease: Where are we now?* Nursing2017, 2017. **47**(4): p. 26-34.
14. Chakravorty, S. and T.N. Williams, *Sickle cell disease: a neglected chronic disease of increasing global health importance.* Archives of Disease in Childhood, 2014. **100**(1): p. 48.
15. Kvam, S., et al., *Exercise as a treatment for depression: A meta-analysis.* J Affect Disord, 2016. **202**: p. 67-86.
16. Jung, M., et al. *Understanding Patients' Needs in Diabetes for Mobile Health -- A Case Study*. in *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*. 2016.
17. Mendiola, F.M., M. Kalnicki, and S. Lindenauer, *Valuable Features in Mobile Health Apps for Patients and Consumers: Content Analysis of Apps and User Ratings.* JMIR mHealth uHealth, 2015. **3**(2): p. e40.
18. Scheibe, M., et al., *Acceptance factors of mobile apps for diabetes by patients aged 50 or older: a qualitative study.* Med 2 0, 2015. **4**(1): p. e1.
19. Lithgow, K., A. Edwards, and D. Rabi, *Smartphone App Use for Diabetes Management: Evaluating Patient Perspectives.* JMIR Diabetes, 2017. **2**(1): p. e2.
20. Ernsting, C., et al., *Using Smartphones and Health Apps to Change and Manage Health Behaviors: A Population-Based Survey.* J Med Internet Res, 2017. **19**(4): p. e101.
21. Harkin, B., et al., *Does monitoring goal progress promote goal attainment? A meta-analysis of the experimental evidence.* Psychological Bulletin, 2016. **142**(2): p. 198-229.
22. Locke, E., *Motivation, Cognition, and Action: An Analysis of Studies of Task Goals and Knowledge.* Applied Psychology, 2000. **49**(3): p. 408-429.

UiT THE ARCTIC UNIVERSITY OF NORWAY

Extended Glucose Monitoring Through The Use Of Group-Based Internet Of Things Mini Displays

Martin Haugen Mikalsen[1], Miroslav Muzny[1,4], Eirik Årsand[2,3], Gunnar Hartvigsen[1,2]

[1]UiT The Arctic University of Norway, Department of Computer Science, Tromsø, Norway
[2]Norwegian Centre for E-health research (NSE), University hospital of North Norway (UNN)
[3]UiT The Arctic University of Norway, Department of Clinical Medicine, Tromsø, Norway
[4]Charles University in Prague, 1st Faculty of Medicine, Prague, Czech Republic

/ ABSTRACT

Social media, mobile technology and Internet of Things (IoT) have enabled new ways of interacting between people with diabetes. However, there are still empty spots on the map that should be explored. One of these is motivational technology (persuasive technology) that can be used by different groups of people with diabetes to motivate each other to be optimal regulated or to simply get continues status updates and reminders on diabetes-related health parameters.

/ METHODS

We are developing a system for group-based IoT-enabled mini displays that will enable extended glucose monitoring of people with diabetes. In this way, users or their sensors can communicate through sending messages and images to mini displays that might be located anywhere. However, we expect that most of them will be located in people's homes or in their professional surroundings. CGM data used by the system is gathered by users sensors, and uploaded to their own NightScout[1] Database. This database exposes an API which is in turn used by a server application to pull the data periodically and push it to the IoT device which show it on the display.

An mobile application is also developed for interaction and mananging the different aspects of the system, sending new text-messages, creating new groups, accepting new members, managing group settings and so on.

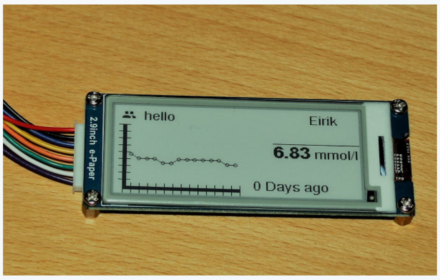

Figure 1: blood glucose levels from a CGM device with a graph over previous measurements

/ MATERIALS

IoT devices often are small, low-cost, and low-energy devices that can run for a long time without the need of changing or recharging batteries. For this project we have chosen the WiFi enabled microcontroller ESP8266[2] and a Electronic Paper Display (EPD) display[3].

EPD are designed to provide a similar appearance of an ordinary ink on paper. Unlike most display technologies, EPD screens do not require any power to maintain a static image, which means that they only consume power on updates of the picture.[4]

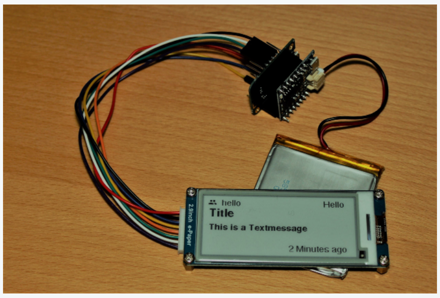

Figure 2: The IoT device with a text message sent from the mobile application, and pushed to the IoT device

/ RESULTS

A Working Prototype has been developed with the ability to show text/messages from multiple users on the IoT device, in addition to the transfer of CGM data status with current and past values. In addition a backend server has been developed to support the storage and communication from the mobile application and to the IoT device.

References:

[1] NightScout Project. NightScout - We are not waiting. url: http://www.nightscout.info/
[2] Espressif Systems. ESP8266 overview. url: http://espressif.com/en/products/hardware/esp8266ex/overview
[3] R.Jayalakshmi Adithya. Potu and Dr.K.Umpathy. "Smart Paper Technology a Review Based On Concepts of E-Paper Technology". In: IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 1 (Jan. 2017). issn: 2278-8735. url: http://www.iosrjournals. org/iosr-jece/papers/Vol.%2011%20Issue%201/Version-1/ G011114246.pdf.
[4] Peng Fei Bai et al. "Review of paper-like display technologies (invited review)". In: Progress In Electro magnetics Research 147(2014), pp.95– 116.

For more information, please contact:

Martin Haugen Mikalsen, e-mail: martin.mikalsen@outlook.com
Gunnar Hartvigsen, e-mail: gunnar.hartvigsen@uit.no

UNIVERSITY OF TROMSØ