MASTER'S THESIS | LUND UNIVERSITY 2015

System for activity tracking of patients with chronic kidney disease

Jens Gustafsson, Alfred Åkesson

Department of Computer Science
Faculty of Engineering LTH

ISSN 1650-2884
LU-CS-EX 2015-35

System for activity tracking of patients with chronic kidney disease

Jens Gustafsson
ada10jgu@student.lu.se

Alfred Åkesson
ada10aak@student.lu.se

July 28, 2015

Master's thesis work carried out at
the Department of Computer Science, Lund University.

Supervisor: Boris Magnusson, Boris.Magnusson@cs.lth.se

Examiner: Görel Hedin, gorel.hedin@cs.lth.se

Abstract

Many people suffering from chronic kidney disease are in need of a kidney transplant. A problem in the health care is that patients cannot undergo surgery if they have too much belly fat. Regular exercise and physical activity are therefore crucial for this group of people. A project group at the Skåne University Hospital has recently been established to help patients with chronic kidney disease to lose weight. A question they asked themselves was whether it was possible to use activity tracking devices in the project. The purpose of this thesis is to design and evaluate a system that can be used to track the physical activities of the patients. The system will be built using a Sony SmartBand, a wristband collecting and analyzing data about a user's daily activities.

Keywords: chronic kidney disease, activity tracking, lose weight, Sony Lifelog

Acknowledgements

We would like to thank Fredrik Nilsask and Mia Trondsen in the RSAT-group for all their feedback and support. We would also like to thank Boris Magnusson, our supervisor, and Görel Hedin, our examiner, for valuable input.

Contents

Chapter 1
Introduction

Many people suffering from chronic kidney disease are in need of a kidney transplant. A problem in the health care is that patients cannot undergo surgery if they have too much belly fat. It is very hard for this group of people to lose weight, as further described in section 2.2. Regular exercise and physical activity are therefore crucial for these patients.

A project group at the Skåne University Hospital has recently been established to help patients in need of a kidney transplant to lose weight. The group consists of a doctor, a psychotherapist, a dietitian and a welfare officer. One of their wishes was to use activity tracking devices in the project. For this reason another project group (the RSAT group) was established to investigate the possibilities of using such devices.

This thesis presents a system with the aim of helping over-weight patients with chronic kidney disease to lose weight. The system is called the Region Skåne Activity Tracking system, RSAT.

itACiH is a research project that investigates the possibilities of advanced health care at home. As a part of the itACiH project a planning-system was developed that is currently being used at the hospitals in Skåne. The planning system is built as a web application and can fetch data from sensors that are installed in a patient's home. RSAT is intended to be integrated with the planning system.

Sony Mobile has developed wearable devices that can track a user's daily activities. The data collected by these devices are saved online in their cloud and accessible through an API called the Lifelog API. Sony wants to investigate what different use cases that exists for their wearable devices and has chosen to sponsor this project with smart wristbands for the patients to wear.

1.1 The RSAT-group

RSAT has been developed in collaboration with a project group established at the Region Skåne University hospital. In this thesis this group is referred to as the RSAT-group. Sony

Mobile was also a member of this group and it was therefore chosen that RSAT should be built around their Lifelog platform.

1.2 Activity tracking health care

RSAT, the system to be developed is a tool intended to help physiotherapists and other clinicians to monitor a patient's physical activity. Its focus is to monitor over-weight patients with chronic kidney disease but it is assumed that it also could be used to monitor other groups of patients that for some reason needs to get more physical active.

The term *activity tracking health care* will be used throughout this thesis to describe this kinds of treatments.

1.3 Purpose and question formulation

The purpose of this thesis is to build a system that can be used in the activity tracking health care to help patients with chronic kidney disease to lose weight. The Sony Lifelog platform will be used as the foundation for the system. Therefore, an evaluation regarding how well suited the Sony Lifelog platform actually is for being used in the RSAT will be performed.

There exists many different activity tracking devices on the market. A goal with this thesis is therefore to investigate what different properties those have and to discuss what properties that an activity tracking device ideally should have in order to be used in the activity tracking health care.

The following questions will be used to fulfill the purpose:

- *How well suited is the Sony Lifelog platform to be used in the activity tracking health care from a technical standpoint?*
- *How well suited is RSAT, the system to be developed, to be used in the activity tracking health care?*
- *What properties should an activity tracking device ideally have in order to be used in the activity tracking health care?*

1.4 Method

Before the development of RSAT could start, an interview with a physiotherapist from the RSAT-group was held. The wishes from the physiotherapist was used as input when the requirements was created. Thereafter a proof-of-concept system was developed to verify that it was possible to use the Lifelog API.

Iterative prototyping was used to develop RSAT. For each iteration, the current requirements was identified and implemented. Each iteration resulted in a demo that the physiotherapist evaluated. Feedback from the physiotherapist was used as input to the next iteration. Sometimes the feedback resulted in new requirements on the system and if those were feasible, they were added to a development iteration.

In parallel with the development, a research investigating different wearable devices and different platforms used for activity tracking was performed. The experience from this investigation in combination with the experience from the development of the system created the foundation for the evaluation of both the Lifelog platform and RSAT.

1.5 Disposition

The thesis is organized into three different parts; an overview part, a development part and an evaluation part.

The overview part consists of chapters 2 and 3. It gives a medical background, describing why RSAT is needed and a technical background that describes the Lifelog platform.

The development part consists of chapters 4 and 5. This part contains information about the requirements on the system, what decisions that were made, why the decisions were made and how RSAT was implemented.

The evaluation part consists of chapters 6, 7 and 8. In chapter 6 RSAT is compared to system similar to it self. In the following chapters RSAT and Lifelog are disucssed and our conclusions are drawn.

1.6 Contribution statement

Most of the work in the thesis was done collectively by the two parties but in some cases one party was contributing more than the other. For the development part of this thesis Jens did most of the implementations in the Lifelog service and Alfred designed most of the databases and the database services.

For the report Alfred created all the illustrations and Jens created the introduction chapter. Most of the text in chapter two (About chronic kidney disease) was produced by Jens and most of the work in chapter 6 (Related work) was produced by Alfred. The rest of the thesis was created in collaboration. Most of the times, a first draft of a section was written by either Jens or Alfred and the other party proofread it and then created a second version of it.

Chapter 2
About chronic kidney disease

In this chapter an introduction to chronic kidney disease is given. The chapter starts with a short introduction to chronic conditions in general and is followed by a discussion about chronic kidney disease in particular.

2.1 Chronic conditions

The costs in the health care are ever increasing. A huge part of the health care and the economic costs associated with it addresses patients with chronic conditions. More than 60% of all deaths in the world is due to this types of complications. In Sweden 80% of all health care costs go to treating these conditions [18]. A study shows that three types of lifestyle behaviors leads to four chronic diseases. These diseases accounts for about 50% of all the deaths worldwide, see figure 2.1 [5].

Figure 2.1: Figure describing what lifestyle behaviors that increases the risk of death

Physical inactivity, as seen in figure 2.1, is one factor that could lead to a chronic condition. An increased amount of physical activity can therefore lower the risks of these conditions.

2.2 Chronic kidney disease

Kidneys have the responsibility of filtering waste products from the blood in the body [41]. Chronic kidney disease (CKD) is a gradual loss of kidney function over time [29]. There are no specific symptoms of the disease but illness, tiredness and a reduced appetite are all common symptoms [29]. As the kidney function is worsening, waste products build to high levels in the blood and may cause complications such as high blood pressure, muscle and bone mass loss [29]. The risk of getting other diseases such as cardiovascular disease, pericarditis and anemia is also growing. When the waste products in a patient's body becomes so high that the patient starts to get sick from it, the patient requires dialysis [29].

People with severe CKD are in need of a kidney transplant. Until a transplant can be performed, dialysis is required. Without dialysis, the patients would have a poor life expectancy [29]. Dialysis is a cumbersome treatment process for a patient as it takes up a lot of time and adds a lot of constraints to the patient's life. Dialysis is also expensive for the society. A rough estimate of the annual cost of treating one patient with dialysis in Sweden is 250 000 – 500 000 SEK [33].

Peritoneal dialysis and hemodialysis are dialysis that can be performed in the home of a CKD-patient. The principle behind peritoneal dialysis is that a sterile solution containing glucose is run into the patient's abdomen [6]. The peritoneum inside the patient acts as a membrane across which waste products from the blood is filtered. A downside with the peritoneal dialysis is that the calorie intake increases, since the sterile solution contains glucose. Hemodialysis is a kind of blood dialysis, where blood is filtered outside the body.

Despite that CKD is considered to be an irreversible condition, the progress of the disease can be slowed down. The National Kidney Disease Education Program (NKDEP) lists three strategies that can be used to slow down the progress [31]. One of those strategies is lifestyle interventions and addresses different health-promoting behaviour. One of the health-promoting behaviours they discuss is physical activity. NKDEP states that people suffering from CKD have the same recommendations for physical activity as the general population.

Eventually, a patient with CKD must undergo a kidney transplant. The patient must be in good physical shape, as the risk of infections and other diseases otherwise are too high. It is also essential that the patient does not have too much belly fat. Not only does the fat induce a greater risk for the patient's life, it also makes it harder for surgeons to practically perform the transplant.

Chapter 3
Wearable Technology

This chapter contains a description and discussion about wearable technology, describing SWR10(the smart band used to collect movement data from patients in this thesis) and the context it operates in. It also contains a discussion about platforms for saving data collected from wearable technology.

3.1 Smart wristbands in general

Wearable health and fitness devices are becoming more and more popular. Smart wristbands are the most popular devices in this category, though it is predicted that the smart watches will become favored over the wristbands in 2015 [13]. The difference between a smart wristband and a smartwatch is a bit of a grey area. A smart wristband and a smartwatch have more or less the same functionality. Smart watches in general tend to have more functionality than a wristband, but this is not always the case.

The focus in this section is on wristbands but the descriptions also holds for smart watches. For this thesis the SWR10 is the chosen activity tracker. This is done because SONY is a stakeholder in the project and has chosen that smart band for the project.

3.1.1 Accelerometer

Movement tracking and sleep tracking are two common features in the smart wristbands that exist on the market. According to Quantified Wellness, an article published by RGA Reinsurance Company [5], most of the data are collected through an accelerometer in the device. Many of the smart wristbands also takes advantages of the GPS in the smartphone they are connected to and thereby increases the accuracy of the measurements.

The Sony SmartBand SWR10 uses an accelerometer for recognizing different movements a user makes [37].

For use in the activity tracking health care it is essential that the patients and the clinician can trust the data gathered by the wristband to be able to draw conclusions from it. Data collected in this type of fitness devices has shown to differ a lot. For example an equally long walk could differ between activitys tracker from 8.9 km to 14.5 km [26]. Studies about the accuracy of the SWR10 has not been found in the research process of this thesis and due to other priorities a field study of this was not conducted in the thesis.

3.1.2 Other sensors

Other features that exist in wristbands on the market are the ability to track a user's heart rate, body temperature, blood oxygen level, respiration rate and galvanic skin response. The ways these sensors are used differ between different wristbands. In some wristbands, the user must activate a sensor in order to make a measurement and in some wristbands data are collected without any interaction with the user at all.

The possibility to see the patients heart rate is something the physiotherapist in the RSAT group expressed an interest for. If the heart rate was known, it would be easier to know how active each patient is during a day. The fact that the SWR10 does not have any heart rate sensor is therefore one of its major disadvantages.

3.1.3 Display

Some wristbands on the market are equipped with a display. The display can be used to deliver different kinds of intelligence feedback to the user and also to deliver different kinds of notifications such as if the user has been inactive for too long or reached an activity goal. Feedback and notifications are also delivered to the user through different kinds of vibrations and sounds in the wristbands. A display makes the smart bands more independent of the smartphone. A smartband that has a display only needs the smartphone for synchronizing data to the cloud. Real time update is not considered to be an important feature in RSAT meaning that synchronizing the smartband once a day would be acceptable. A smartband with a display would therefore eliminate the need for patients to carry the smartphone with them wherever they go as they could get sufficient feedback about their physical activities from the smartband instead.

3.1.4 Smartphone dependency

Most or all of the smart wristbands needs to be connected to a smartphone. The synchronization to the smartphone is performed using Bluetooth. Many of the wristband providers also synchronize the data to their own web servers, where the data sometimes is accessible through an API. Some of these API:s are free to use and for some API:s a licence is needed.

Most of the smart wristbands on the market are strongly tied to the manufacturer's own ecosystem, meaning that a user needs to use the manufacturers own applications to view the data collected by the wristband. This is also the case with SWR10. This application SWR10 is using is called Lifelog.

The Sony Lifelog application is an Android application which is built around the idea of logging a users lifestyle and daily activities [39]. The application collects various data about its users through different sensors in a smartphone and SmartWear-devices (Sony SmartWatch or a Sony Smartband) and displays it to the user. Examples of data it logs are information about what applications in the smartphone that have been used, what locations that have been visited and information about different physical activities performed during the day. This application collects the information from the SWR10.

An interesting company, which has chosen to not make there own application, is Angel. Angel has, according to themselves [2], developed the first smart wristband for health and fitness. It is based on an open platform, which includes connectivity, protocols, and SDK.

3.2 Different Activity Tracking Platforms

This section describes and evaluates the Sony Lifelog platform. It also compares Sony Lifelog to other platforms from a developers perspective.

3.2.1 Sony Lifelog API

All the data collected in the Sony Lifelog application are automatically uploaded to a central repository in the Sony cloud that stores the data. This data is globally accessible through a REST-based web-service API [39].

The API has three different endpoints; a profile endpoint, an activities endpoint and a location endpoint. The profile endpoint contains basic information about a user's name, age, height and weight and is entered by the user into the Lifelog application.

The activities endpoint contains the different data Lifelog has logged about performed activities throughout the day such as physical activity. All activities have a type and some also has a subtype.

The location endpoint provides a list of coordinates specifying what locations a user has been visiting during a certain time period. Activities and locations in the API are associated with timestamps. A response from the API always includes the time stamp of the earliest and latest included entry. The format of the timestamp follows the ISO 8601 standard.

Third parties can access the API to retrieve collected data about a user. Every request made to the API must be authorized and https must be the chosen communication protocol. The lifelog Web API uses OAuth 2.0 as its authorization mechanism [39]. To connect to the API, a user must login and give explicit permission for the application or the service to access to the data.

The response to a valid request made to the API contains a set of JSON objects. If the requested data is too large to fit in the size of one response message, the data is split up into several messages and a link-field is inserted into the response. This link-field contains a link pointing to the location from where the rest of the data can be fetched.

REST

The Sony Lifelog API claims to be a REST API. REST is an architectural style for designing API:s and are heavily used in modern web services. The concept is to represent resources as the analogy for the data endpoints and to have a small set of actions interacting with these resources. The REST architecture is based on the following principles according to Pautasso [34]:

- *Addressabilty* All resources must be addressable with a unique identification. On the web each resource uses a unique URI such that it becomes globally addressable.
- *Uniform Interface* The interface to the resources should according to REST be uniformed and simple. On the web the HTTP protocol is used and HTTP methods are used to interact with resources. The GET method is used to read a resource, the POST method is used to add a resource, the PUT method is used to to modify a resource and the DELETE method is used to delete a resource.
- *Stateless Interaction* This means that there is no session with multiple interactions. Instead every interaction is self-contained. No state is shared between a service and client after an interaction.
- *Self-Describing Messages* The requests and responses should contain both the data and the metadata about the format of the data. XML or JSON is usually used to represent data on the web. The Sony Lifelog API uses JSON to represent its data.
- *Hypermedia* Resources that contains a link to other resources uses hypermedia. If the data (the resource) to be sent in a response is too large to fit in a single message, the data can be organized in several messages, where each message contains a link (hypermedia) to the next part of the resource. Then, only the first part of the resource is sent. The receiver can later fetch the rest of the data with help from the hypermedia link in the message.

REST or not

Sony states that the Lifelog API is REST-based. In order to be a REST API the API shall fulfill the architectural principles that presented above. We have evaluated the API and concluded that it actually is a REST API. Below you find comments on why the API fulfils each principle:

- *Addressability* - Each request to the Sony Lifelog contains a time interval and a list of data types telling what you want your response to contain. The queries to the API are idempotent. Two queries containing the same request will always be responded with equal results. The addressability principle therefore holds, as you can address a resource using a unique URL.
- *Uniform interface* - The Sony Lifelog is a read-only API and only supports GET requests. It therefore has a uniform interface.
- *Stateless Interaction* - The communication with the API is stateless. The only exception to this is the authentication part of the communication. However, the authentication can be seen as something that happens before the connection to the API, which would make the API-communication completely stateless.

- *Self-describing Messages* - The strings used to query the API are self-describing and uses distinct parameters. The responses have a JSON-syntax, are self-describing and have distinct parameters.

- *Hypermedia* - The hypermedia principle is used by the API when the requested data is too large to fit in a single response. The API uses links that it sends with the first part of the response. These links contains the URL to the rest of the queried data.

OAuth 2.0

The Sony Lifelog API uses OAuth2.0 as its authorization mechanism. A user must log in and give explicit permission for access to their data. When that is done, an application can get OAuth2.0 authorization for API calls that are made on that user's behalf.

OAuth 2.0 is an authorization protocol developed by the Internet Engineering Task Force (IETF). It describes an open standard for authorization and enables third party applications to obtain limited access to a web-service on the behalf of a resource owner [19]. To describe the protocol, several key actors need to be described, these actors are the client, the resource owner, the authorization server and the resource server. IETF gives the following definitions to the roles:

- The resource owner: *"An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user."*

- The resource server: *"The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens."*

- The client: *"An application making protected resource requests on behalf of the resource owner and with its authorization. The term "client" does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices)."*

- The authorization server: *"The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization."*

The following is a general example of the protocol flow when a client is accessing protected data belonging to a user.

1. A client requests authentication from a resource owner.

2. The client receives an authorization grant credential from the resource owner.

3. The client requests an access token from the authorization server. The server verifies the authorization grant credential and returns an access token.

4. The client receives an access token, which can use to access the protected data in the resource server.

The Authorization Grant is a credential representing the resource owner's authorization. The IETF's OAuth 2.0 specification defines four different grant types; authorization code, implicit, resource owner password credentials and client credentials. The Sony Lifelog API uses the authorization code as its grant type and the other types will therefore not be further discussed [38]. When the authorization code grant type is used, the resource owner is redirected to an authentication server when the client requests authentication. When the resource owner has granted the request, the resource owner is redirected back to the client together with the authorization code, which is appended in the URL. The authentication code is short lived and single-use.

The access token is the only credential needed to access a user's protected resource and can be seen as an abstraction layer, since other credentials such as user-name and password must not be provided. An access token is valid until it expires or is revoked.

A refresh token can be issued together with an access token and, as the name suggests, is used to refresh expired or revoked access tokens. To refresh an access token, the refresh token is sent to the authorization server, which issues a new access token. A refresh token is meant to be valid for a longer time than an access token but that depends on the server implementing the OAuth-specification. A refresh token could in theory expire before an access token.

Limitations in the Lifelog API

There are two things in the Lifelog application that are missing in the API. First, goals the user set in the application cannot be read from the API. Second, in the Lifelog application the user can bookmark different moments. These moments can neither be read from the API. Activity and profile data on the other hand can easily be read from the API.

3.2.2 Google Fit

A competitor to Sony Lifelog is Google fit developed by Google. Google Fit is an open platform for both Android and web applications to share and store fitness data. An application collecting fitness data from a fitness tracker can store the data in Google Fit via the API. Other applications can also read other data from the API and use the data for different calculations. This makes it simple for an application to integrate fitness data collected by different fitness tracking devices. It also makes it convenient for a user to change fitness tracking device as the data the new device is collecting will have the same structure in the API [15].

Google fit is focused on fitness data and Google states that the data shall not be used for medical appliance. The terms state very clearly that the API is not for storing "medical or biometric data". The Google fit API may not be used in medical applications without the permission from Google [16].

The Google fit API has two interfaces; one Android interface and one REST interface. To get access to the API a Google account is needed, the user of the application also needs to give the application permission to access the data. The Google fit data is personal and tied to a Google account. OAuth is used for the REST authentication [15].

Three different types of data can be stored in the API [14]:

- *Public data types* are predefined by Google. These can be read and written by all applications. An example of this is the heart-rate data type, which is defined as `com.google.heart_rate.bpm`
- *Private custom data types* are defined by a specific application and those types can only be read and written by that specific application.
- *Shareable data types* are types of data that other developers can define and share on the Google Fit platform. All applications can read these data types but it is only the applications listed by the developer of the data types that can write data to them. `com.nike.NIKEFUEL` is an example of a shareable data type, which defines Nike's own metrics to measure activity.

Lifelog compared to Google fit

Google Fit and Lifelog both have a REST API that third party users can use to fetch activity data. It is possible to write data to the Google Fit API, something that is not possible in the Lifelog API. This makes it possible to integrate data from sensors of diffrent brands in the API. Both Google and Sony provides a uniform interface for different activity trackers. Although the Lifelog API only contains data from Sony products, all their different products collects data that has the same structure in the API. Google has stated that their API is not to be used in health care applications. This means that Google fit is not appropriate to be used in RSAT. But besides that, Google Fit has a lot of attributes that would make it useful in RSAT. It has all the capabilities that Lifelog has and also makes it possible to import data from other sensors into the API.

3.2.3 Apple Health

HealthKit is a developer framework, developed by Apple, that provides a common data structure for health and fitness applications on iOS. All the data stored in the HealthKit framework is stored in the *HealthKit Store*. This removes the need for developers to write complex and custom-made code to retrieve data from other applications. [3]

Apple Health is an application that makes it possible for the user to visualize the data stored in the HealthKit store. In the Health application users also have the ability to create emergency cards, e.g. medical IDs, in which they can insert medical information about themselves such as what their blood-type is or what allergies they have.

In the Health application the user explicitly controls which applications that are allowed to access the HealthKit store and what data they are allowed to read or write. When an application requests data from Apple Health and is not granted access it gets back an empty response [3]. This means that an application, which is denied access to some data cannot know whether it has been denied or if the data does not exist.

Apple does not use cloud storage for their HealthStore. Instead, all data is stored locally on the phone [3].

There is a wide amount of different types of data that can be stored in the HealthKit framework. A developer cannot create new types in the HealthKit, instead they must use the types provided by HealthKit.

Apple wants to make it as convenient as possible for both the developer and the user. From the user's perspective it shall be possible to retrieve information about health status without opening several different apps. From the developers perspective it shall be possible to use a single API to create applications, which uses data from many different sources. Apple claims that one of the great benefits of choosing to use the HealthKit framework is that it makes the sharing of data between different applications easy.

Diane J. Skiba, professor at University of Colorado (College of Nursing), writes in an article that there are many health record companies and different health care organizations in the US that are integrating their systems with Apple Health [36]. Skiba suggests that the university shall start introducing the concept of patient-generated health data in their health related courses [36].

Lifelog compared to Apple health

There are two major differences between Apple Health and Sony Lifelog. The first difference is that Apple Health does not have a cloud API. Apple, on the other hand, has an API that applications that users run on their smartphones, can use to insert and fetch data. The second difference is that Apple Health is marked to store both health and fitness data, while Sony Lifelog only seems to be marked for fitness data.

3.2.4 Jawbone UP platform

Jawbone is the manufacturer of several fitness trackers. Jawbone's first fitness tracker was released in 2011 [17]. They have created a developer platform called the UP platform. The platform consists of three different parts [21]:

- A REST API.
- An SDK for IOS and Android to consume the REST API.
- A way to communicate with the application using Bluetooth LE.

Developers can read and write data to the endpoints in the API. The API defines what data types that can be stored in the platform. The communication to the API is REST-based. The endpoints contains different data such as information about a user's body temperature and heart rate. Third party applications can also create custom data entries using the `generic_event` endpoint [20].

Lifelog compared to Jawbone UP Platform

The UP platform is more open than the Lifelog platform. The Jawbone API is both writable and readable. They also provide a low level API that different Bluetooth devices can use to write data to the platform. The Bluetooth API gives sensor manufacturers the possibility to create a sensor without the need for a companion application. This functionality is missing in the Lifelog platform.

Chapter 4

Prestudy

The requirements on RSAT were created in collaboration with a physiotherapist in the RSAT-group. Before the development of RSAT started, a discussion about what features that should exist in RSAT was held with the physiotherapist. A bunch of great ideas emerged that are described in section 4.1. All of the ideas were not possible to implement, at least not in this version, but they were used as input in order to create the requirements for RSAT. The requirements on RSAT are described in section 4.3.

A general design sketch of RSAT is given in chapter 4.5.

4.1 Wishes from the physiotherapist

What the physiotherapist considered most important was that RSAT should be able to track a patient's physical activity and automatically detect that a patient has been inactive for too long. RSAT should be able to detect different kinds of physical activity, both activities such as strength training and activities such as running or walking. Initially it was also wished that data such as pulse, blood pressure and blood glucose levels should exist for each patient in the system. Another feature that was regarded important was that it should be possible to track the weight of the patients. The patients should also be able to upload photos of what food they are eating.

The user-interface to be used by the physiotherapist should be easy to use. It was also expressed that there was a need for at least two kinds of views: (1) An overview view, where all patients can be listed and where it is easy to see the status of each patient. (2) A specific view where details about each patient can be viewed.

The physiotherapist also wanted that RSAT should be able to automatically create and send different kinds of intelligent messages to the patients. Those messages should be used to encourage the patients to workout more or to flatter patients that have been doing great progress.

4.2 Constraints on RSAT

Two things had to be taken into account when the requirements for RSAT were made. (1) Sony Lifelog was already the chosen data source. This meant that only data that could be tracked by Lifelog devices and that was accessible via the Lifelog API could be fetched into RSAT. (2) RSAT was to be integrated with a system that already existed (the itACiH planning system). This meant that the architecture of RSAT had to be designed such that an integration was possible.

The physiotherapist expressed an interest of being able to see data such as blood pressure and blood glucose levels in RSAT. This is data that is not possible to track with help of the Sony Lifelog platform and will therefore not exist in this version of RSAT.

All patients that are participating in the RSAT-project must have a smartphone and a wristband. This however, is nothing that the health care can demand their patients to buy and is therefore something that they must be provided with from the health care.

4.3 Requirements

4.3.1 Activity data

Lifelog can collect many kinds of data. All of the data that Lifelog can collect is not fitness- or health related. After considerations with the RSAT-group it was decided that the following data about a patient should be read from the Sony API and exist in RSAT:

- Hours slept per night.
- Steps taken per day.
- Hours walked per day.
- Hours run per day.
- Hours riding bike per day.
- Calories burned per day.
- Weight.

It was also decided that it should be possible to browse the history of all the collected data in RSAT. This function was considered important, since it would make it possible to see if patients tracked in RSAT makes any progress, i.e. if they increase their amount of physical activity or not.

4.3.2 User-interface

The physiotherapist wants a user-interface that is easy to use. The following requirements was set on the user-interface:

- RSAT shall be built as a system the can be reached from the Internet.
- The user-interface shall be easy to use.
- The data in RSAT shall be easy to access for the staff using it.
- It shall be possible to create specific goals for each patient in the system.
- It shall be possible to create goals for each datatype that exists in RSAT.
- It shall be possible for the clinician to have an overview of all patients, showing if they have completed their goals or not.
- It shall be possible for the clinicians to have an in depth view of each patient with history of all activity data.

4.3.3 Other requirements

- Each patient shall be able to view the goals that the clinician has created for him/her.
- It shall be possible to see if a patient has been inactive for too long.
- It shall be possible to integrate other wearable devices from other manufacturers in future versions of RSAT.
- A device/set of devices (smartphone and wristband) used by a patient shall be reusable. A new patient shall be able to be equipped with the same device/set of devices that have been used by another patient before.
- It shall be integrated with the itACiH planning system.

4.4 Initial idea

The initial idea was to develop a specific Android application that was installed on each patient's smartphone. This would have been the ultimate choice but this idea was thrown and it was chosen that the patients would use the Sony Lifelog Application instead. The two major reasons for this decision were: (1) Sony has invested a lot of resources to create the Lifelog application that runs on most newer Android devices. This application has almost all the functionality that the physiotherapist initially asked for. (2) That the schedule for building the RSAT was quite tight. If an Android application should be developed the time frame would have to be changed. It would also be hard to create an application that the patient's would prefer to use instead of the Lifelog application.

The Sony Lifelog API is a read-only API. Hence, it is not possible to write any data to the API. It is therefore not possible for the physiotherapist to set goals that are automatically updated in the patients Lifelog accounts. Another problem is that the goals are not accessible at all through the API. This means that in order to set goals, the physiotherapist must have a meeting with the patient where they together set the different goals for the patient. At this meeting, the physiotherapist inserts the goals in RSAT and the patient inserts the goals in the Lifelog application.

4.5 Final idea

It was decided that the patients tracked in RSAT will use the Lifelog application. It was also decided that a web application will be developed that the clinicians can use to track the activities of the patients.

Figure 4.1 shows an overview of the solution followed by a description of the steps for transferring the data from the patient to the physiotherapist.

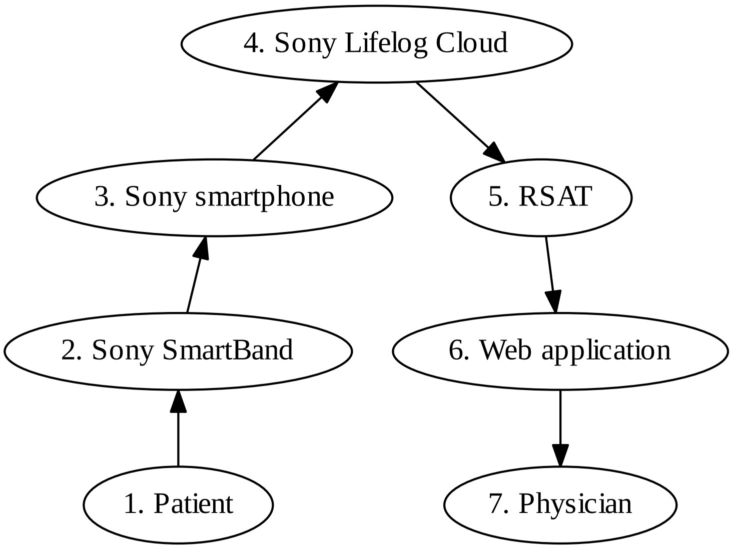


Figure 4.1: Graph of the proposed solution

1. Each patient added in RSAT has a smart Sony SmartBand wristband around his or her wrist.
2. The wristband collects data about each patient's movements and the data is categorized into different activities.
3. A Sony smartphone collects the data from the wristband.
4. The data is stored in the Sony Lifelog application on the smartphone and synchronized to Sony's cloud.
5. RSAT fetches data from each patient's RSAT-account in the Lifelog cloud, processes it and saves it to a database.
6. The RSAT web application retrieves the data from all patients in the database and visualizes it to the clinicians.
7. The clinicians can interact with RSAT and see the progress of each patient via the web application.

4.5.1 Process of adding a patient

Each patient in the program will be equipped with a Sony SmartBand and an Android smartphone that are borrowed from the hospital. The SmartBand must synchronize its data to a smartphone and that is the reason why the patients are also equipped with a smartphone and not only a SmartBand.

To make it easy for the clinicians and the patients to get started, each smartphone and SmartBand will be preconfigured by an administrator. The following steps are used to configure the devices and accounts for the patients:

1. A new Google account is created to be used in the Lifelog application.
2. All necessary applications are downloaded to the smartphone.
3. The SmartBand is connected with the smartphone.
4. The smartphone is added to RSAT and given a unique name.

When a new patient is added to RSAT, the clinician gives him/her the pre-configured set of devices. The clinician also links the patients profile to the smartphone's name in RSAT.

Chapter 5
Development

This chapter describes the development cycles of the RSAT. It also gives a brief introduction to Palcom, the different 3rd-party libraries and PalCom services that have been used to build RSAT.

5.1 PalCom

The planning system in the itACiH project use PalCom as their platform. PalCom is a platform that makes it possible to build software systems, which consists of small sub-systems [8]. These sub-systems are called *PalCom Services*. Different services are composed together with the help of assemblies. Some of the most common and important components in the PalCom platform are [8]:

- *PalCom service* - A PalCom service is a type of self-contained system. The service contains logic for both computations and interactions with the outside world. The implementation of services is made in Java and can therefore leverage the Java ecosystem. Every version of a service has a unique id called *serviceId* and every instance of a service has an unique id called *instance*.
- *PalCom command* - A PalCom command is what a PalCom Service uses to communicate with other services. A service defines a number of commands, either *in-commands* or *out-commands*. An in-command defines the format of a command sent to the service and an out command defines the format of a command sent from the service. A command consists of zero or more parameters. Each parameter has a name and a data type. The data type defines what type of data is sent in the parameter.
- *PalCom assemblies* - PalCom Assemblies are rules explaining how PalCom commands shall be routed in a PalCom network. An assembly connects a service out-commands with another service in-commands and vice-versa. This is what makes

it possible for different services to communicate with each other. An assembly connecting services with each other could have the following rule: “when service s1 sends command cmdOut, send it to command cmdIn on service s2". Assemblies are meant to make the interconnection between different services easy to implement.

- *PalCom device* - A PalCom device is a container that runs one or more services and has a unique id called *deviceId*.

- *PalCom Runtime Component* - Each service has a mechanism, which makes it possible to discover other services and to advertise its existence. This mechanism is called the PalCom Runtime Component. Different services can run on an application called theThing. theThing is a type of PalCom device. It has a JVM for running services and an interpreter for assemblies. It also handles all the network configuration and connections in a PalCom network.

5.1.1 PalCom services

The database service

The itACiH-project has developed a generic database service that can be extended to be used for database tasks on the PalCom platform. The database service has one *in-command* and one *out-command*. Each command to the database has 6 parameters. In this project only two of the parameters are used; the json-parameter and the classname-parameter.

- *json* - This parameter describes the parameters sent to the database in a JSON format.

- *classname* - This parameter tells the database service how to interpret the JSON-data sent in the json-parameter by providing the name of a predefined class in which the keys in the json-parameter exists as attributes.

Depending on what classname that is stated in a command received in the service, different database operations are performed. The operations to be performed are predefined in the service. A command sent to the database can look like this:

```
{
'json': '{"userId":3,"date":"2015-02-15T00:00:00","steps":30}',
'classname': 'databaseStep.RequestAddStepInterval'
}
```

When this command is received, the server converts the JSON-data in the json-parameter in the command to a Java object of the class "RequestAddStepInterval", which exists in the package "databaseStep". Then the service communicates with the actual database and performs the actions that are defined in a predefined statement in the service.

The WebServer service

The PalCom Web Bridge is used to connect web applications with PalCom services. The PalCom Web Bridge is therefore referred to as the web server in this thesis. The first version of the web server was developed in 2012 and was based on long-polling. It is described in the PalCom Web Bridge Design Document [35]. The current version of the web server makes use of web sockets instead.

A configuration object is needed to establish a connection between the web server and another PalCom-service. The configuration object is used to establish a web-socket and specifies device-id, service-id and instance name of the service its establishing the socket with. When the web-socket is established, in-commands can be sent to the service and out-commands can be listened to via the web service from a web-application.

5.2 3rd-party libraries and services

5.2.1 JavaScript libraries

Several different JavaScript libraries have been used to develop the web application of RSAT. The frameworks used the most are briefly described below:

- *Backbone.js* - A framework used to provide structure to web applications. It provides models, collections and views that can connect to existing API:s over a RESTful JSON interface [32].

- *Highcharts* - A library for making charts on the web using JavaScript[23].

- *Moment.js* - A JavaScript library used to parse, validate, manipulate, and display dates in JavaScript [28].

5.3 A proof of concept

As a first step in the development, a proof-of-concept system was created. The proof-of-concept was developed to assure that data from several different users could be fetched from the Sony Lifelog API and viewed in a user interface. The system consisted of two parts, (1) a web application that served as the user interface and (2) a server that fetched the data from the API and handled the authorizations. The two parts of the system used HTTP for communication. To make the implementation of the system as easy as possible, data about a user was stored to files.

The web application was developed in JavaScript, using Backbone as framework. Figure 5.2 shows a bar diagram that tells the number of steps user no. 15 has taken each day in a specific time interval.

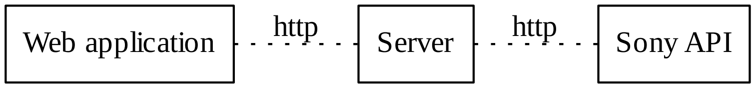


Figure 5.1: An overview of the proof-of-concept system.

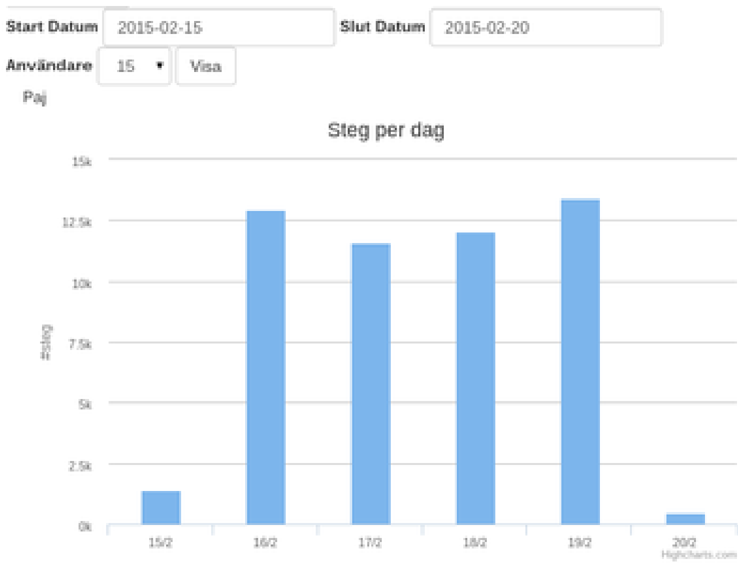


Figure 5.2: A printscreen of RSAT from prototype 1

5.4 First iteration

In the first iteration cycle of RSAT, an architecture based on the PalCom platform was introduced. It consisted of a web-application (that was more or less the same as the one developed in the proof-of-concept system) and four different PalCom-services. The version of RSAT from this iteration could fetch data about a user's steps per day and sleep behavior. This data was stored in a database and could be accessed and read via the web application. The decision to store the data in RSAT was due to long loading-times when fetching the data directly from Sony's cloud. An overview of the system and its different services are shown in figure 5.3.

5.4.1 The web application

The web application is the user-interface to be used by the clinicians. From here the clinicians can view the data and goals for each patient. This part of RSAT is written in JavaScript using Backbone as framework. Each time a clinician reads the activities of a patient, a request for the patient's data is sent to the database service via the web server. The webserver routes the request to the database service, which handles the request and responds with the requested data.

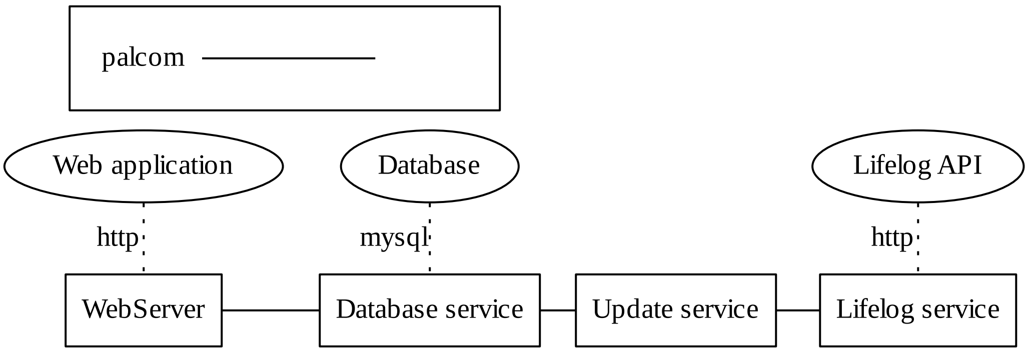


Figure 5.3: An overview of the system in the first iteration

5.4.2 The web server

The web application uses http as its communication protocol. In order for the web application to communicate with the PalCom-services in the system, a component that can translate the http requests in to PalCom-command and vice versa is therefore needed. This is what the web server does. The web server is described in section 5.1.1.

5.4.3 The Database service

The database service handles the communications with the database. The database service has been described in section 5.1.1. This service can input and query step and sleep data. When it receives a command, it performs different requests to the database. Depending on what type of command that was sent the service creates a response or responses to be sent back to the service making the request and/or to other services in the network.

5.4.4 The Update service

The update service has the responsibility of periodically asking the Lifelog service for data from the Lifelog API and to send it to the Database service. The service requests data from the Lifelog service in a predefined time interval. When the response from the Lifelog service arrives to the update service, the response is parsed and a command for adding the data in to the database is created and sent. In this iteration the update service has the responsibility of updating the data about how many steps a user has taken and for how long a user is sleeping during the night.

5.4.5 The Lifelog service

The Lifelog service is the gateway that connects the Sony Lifelog API to PalCom. Since the Lifelog API uses OAuth 2.0 as its authorization mechanism, the Lifelog service needs to have an access token to the account of every user that it shall retrieve data from.

The first time a patient is added to RSAT, the patient must grant RSAT access to his or her data in the API. When the patient is granting access, the Sony OAuth server sends a response containing an access token and a refresh token. Since the access token is only valid for a short time interval, the access token is not stored. Instead the refresh token is stored, which is valid for a unlimited time interval, or until the user revokes the access.

When the Lifelog Service requests data for a patient from the Sony API, it looks up the refresh token belonging to that patient in a database and uses that token to request a new access token. The response from the API contains an access token and a new refresh token. The new refresh token is stored instead of the old in the database and the access token is used to access the Lifelog API. The response from the Lifelog API has a JSON format. This data is parsed and returned in a PalCom-command.

5.5 Second iteration

Additions made in the second iteration of the RSAT were to include more data types. The following data types were added to the system:

- Walking time
- Running time
- Biking time
- Calories burned
- Weight

Another feature implemented during this iteration was a feature making it easy to add new patients to RSAT. In figure 5.4 an overview of the RSAT from the second iteration is shown. The services implemented and improved in this iteration are described in the subsections below.

5.5.1 Database services

The generic database service was used to create different types of database services. The following database services were created:

- *The database goal service*, which is used to store all goals of a user. When the clinician inputs a goal for an activity in to RSAT this service saves that goal to the database. The service also handles the fetching of these goals.
- *The database user service*, which handles the names of smartphones in the system by mapping the name on the phone to an ID number.
- *The database activity service*, which stores different activities and other data parsed from the Lifelog API. The data stored in the activities database service is parsed from the activities endpoint in the API.

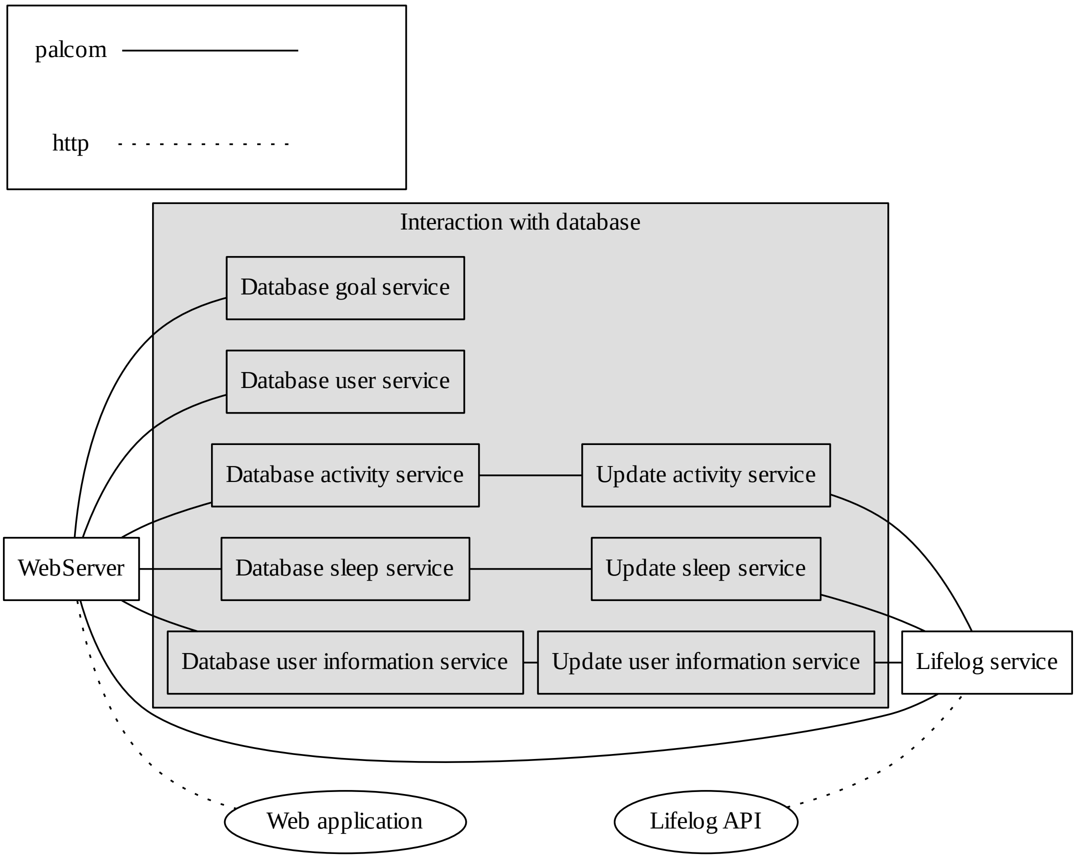


Figure 5.4: An overview of the system's architecture after the second iteration

- *The database user information*, which stores personal information about the users. An example is the weight is stored in this service. The data is parsed from the Lifelog API.

- *The database sleep service*, that stores information about a users sleep. The data is parsed from the activities endpoint in the Lifelog API.

5.5.2 Updater services

In this iteration the updater service from iteration 1 has been divided up into several updater services. Each service has the responsibility of updating a specific type of activity or a group of activities. Each updater service fetches a list from the database with information about the users ids such that it can query the Lifelog service for data. The task of updating the data of a user begins with the fetching of the date of the latest data point for the user. This date is sent to the Lifelog service so that the Lifelog service knows from when in time it shall start fetch data for the user. The service returns all data points from that user from the date returned from the database until current time. These data points are sent to the database service, which adds them to the database.

The following database services exists in the RSAT:

- *The Update activity service*, which triggers a signal to update activity data. Activity data includes the following: steps taken, walking time, running time, biking time and calories burned.
- *The Update sleep service*, which triggers a signal to update sleep data.
- *The Update user information service*, which triggers an update of user information.

5.5.3 The user interface

In the second iteration of the RSAT, two improvements of the user interface was made:

- The views in the graphs was fixed such that the data shown was showed for one week at the time.
- The ability to change an activity goal at the same place as the data was shown for the activity was added.

These improvements were based on feedback from the project group. Figure 5.5 shows the interface from iteration 2.

Figure 5.5: A printscreen of RSAT from prototype 2

Chapter 6
Related Work

Different smart wear manufacturers have started their own wellness programs, which are based on the idea about using the data collected by their wearable devices as input to measure the amount of physical activity performed by the people participating in the wellness program.

Some software systems used by health care providers are also integrating activity trackers in their system. Two examples are Fruit Street and Tactio. Fruit Street is an American company that are integrating Fitbit's activity trackers in their system and Tactio is an Canadian company integrating different activity trackers in their applications. In this chapter a comparison is made between these different systems and RSAT.

6.1 Wellness programs

Many corporations use wellness programs to encourage their employees to be healthy. The goals with wellness programs are in general to get people to eat more healthy food and to stay active.

According to Health Affairs, the leading journal of health policy, the interest in wellness programs is growing [4]. They state that more than 60% of the Americans get their health insurance from the company they are working at. It is therefore no surprise that workplace wellness programs can generate savings for a company.

Several smart wear manufacturers have recently (see figure 6.1) started their own wellness programs, among these Jawbone [22], Fitbit [7] and Garmin [10]. They have all built a system that focus on delivering the big picture about the employees' health and to trigger the employees to get active by social interactions and challenges in the program. They have all integrated their own products in their program.

The different wellness programs mentioned above all have a portal with a dashboard where all the participating employees can view their progress and compare it with their colleagues. The employees can also be organized into teams and specific team goals can be set for the teams. Administrators of a wellness program can measure the progress of the employees and their amounts of healthy activities through an administrative dashboard where statistics and summaries about the programs can be generated and viewed. This is similar to RSAT that is a dashboard for clinicians to administrate and view statistics of the patients.

Smart wear manufacturers creating their own wellness programs have full control over both hardware and software. This means that they can customize the hardware and software, as they want when implementing a feature. Garmin has taken advantage of this by creating the vivohub.

The vivohub collects data wirelessly from the Garmin activity trackers in it's surrounding [12]. The device is installed in the workspace where the wellness program is run. By placing vivohub in a busy place in the office, the hub can collect data from all employees wearing a wrist band. This eliminates the need for every employee to have a smartphone in order to send his or her data to the wellness program. A similar feature had been desirable for RSAT, to eliminate the need for patients to borrow a smartphone from the hospital.

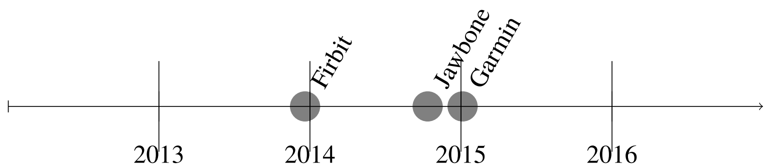


Figure 6.1: A timeline showing when three different smart wear manufacturers released their wellness programs [25] [27] [11].

6.2 Fruit Street

Fruit Street is a software system built to help health care providers connect with their patients and is made for the American market. Fruit Street makes it possible for the clinicians to have video chats and screen sharing sessions with their patients. Appointments can be booked and payments collected. Patients can upload pictures of what food they are eating and share it with the clinicians for nutrition purposes.

Another feature Fruit street has is the possibility for the patients to import data from their health trackers [9] into the system. (For now it is only possible to connect smartwear products manufactured by Fitbit). To add a patient in Fruit Street, the following steps are performed [24]:

1. The clinician sends an invite to the patient via e-mail.
2. The patient gets the invite and creates an account by filling out a form that is linked in the invitation.
3. The patient fills out a health questionnaire created by the clinician.
4. A video chat application is installed on the patient's device.

5. The patient connects his or her Fitbit account with the system by logging in to Fitbit via a redirect from the Fruit Street's web application.

This process is more complicated for the patient compared to RSAT where the clinicians add the user to the system and the patient do not have to do anything. In Fruit Street data is fetched from the patient's Fitbit account that the health care providers can access via the Fruit Street system. The health care providers can access both current- and historical data, which can be browsed in both weekly and monthly intervals [24].

Goals for a patient can also be created. Examples of goals that can be created are number of steps a patient shall take during a specific time period or how many hours a patient shall sleep during the night. The health care providers can also mark goals as completed. Compared to RSAT Fruit Street is more flexible in regards to how goals can be created for each patient. The only type of goal that is possible to create in RSAT is daily goals, where in Fruit Street a goal can have an arbitrary length, meaning that you can create a monthly or weekly goal.

6.3 Tactio

Tactio Health Group, a Canadian company, has developed a system called Tactio RPM [40]. Tactio RPM is meant to be used by both clinicians and patients. Its task is to to provide an overview of a patient's health for both the patient and the clinician and to provide a way for the two parts to remotely communicate with each other. The system consists of three parts; a smartphone application, an iPad application and a cloud platform [40]:

The smartphone application The smartphone application is meant to be used by the patients and can fetch data from various sensor devices such as body scales, activity monitors and heart rate monitors. Data can also be entered manually to the system through the application. The application can also show coaching messages from the clinicians to the patients.

The iPad application The iPad-application is used by the clinicians to monitor and manage the patients. The clinicians are able to see the status of different metrics such as BMI and blood pressure of all the patients in the system with help of different colour schemes used in the application. Health reports can also be generated for each patient.

The cloud platform The cloud platform is what stores all the sensor data that are fetched by the smartphone application. It is possible to get and post data from/to the platform via an API that uses OAuth-authentication. Thus, it is possible for third party applications to retrieve data or to put data into the system. The API can for example be used to interact with systems dealing with electronic health records [1] [40].

6.3.1 RSAT compared to Tactio

Patients participating in RSAT uses Sony's Lifelog application. Tactio has developed their own application for their participants to use. We have found three major advantages with

Tactio's approach. (1) They can decide by themselves what data shall be shown in the application. Data they consider irrelevant in the manufacturer's application can be ignored. (2) It makes it possible for Tactio to fetch data from sensors from several different manufacturers and to visualize the fetched data for the patients in a uniform way. This data can also be used as input to algorithms that can create new and more accurate data. (3) Custom made feedback and notification messages can be generated and sent to the applications from the clinicians. Different kinds of activity goals can also be generated by the clinicians and visualized in the application. One disadvantage with this approach is that they have to invest a lot of resources to create an application that can deliver a good user-experience for the patients. Since most manufacturers already have created an application, Tactio must create an application that is equally good or better to get the patients to use theirs instead. The clinicians using the Tactio system can only interact with the patients via an iPad application. RSAT is built as a web application and can be accessed through any web browser. This is both an advantage and a disadvantage at the same time. The advantage is that the clinicians only need to learn one platform and that the application will always have the exact same structure. The disadvantage is that Tactio data only can be accessed from an iPad where RSAT can be accessed from any normal web browser. Tactio's REST API can be used to integrate Tactio data to other systems. The same is true for RSAT but RSAT does not have an API, instead one has to implement a PalCom service that can communicate with RSAT. Therefore, unless the system to be integrated with RSAT is built on the PalCom platform, it is easier to integrate a system with Tactio than it it is to integrate it with RSAT.

Chapter 7

Evaluation

In this chapter RSAT is evaluated and the requirements that was stated before the development process began are discussed and reviewed. Pros and cons of using the Lifelog platform as a data source for RSAT are also discussed in this chapter.

7.1 Fulfillment of requirements

In this section all the requirements from section 4.3 are reviewed. Table 7.1 lists all the requirements for RSAT and tells whether they are fulfilled or not.

requirement	*fulfilled*
Show hours slept per night in web application	Yes
Show steps taken per day in web application	Yes
Show the walk, run and ride bike duration per day in web application	Yes
Show calories burned per day in web application	Yes
Show weight in web application	Yes
Show history of collected data in web application	Yes
Web application can be reached from the Internet	Yes

requirement	*fulfilled*
The user interface of the web application is easy to use	This is hard to verify, but a popular UI library was used to make RSAT look familiar for web-users.
It is easy to access the web application for the staff	Yes, just to follow a URL
It is possible to create goals for each patient	Yes
It is possible to create goals for each type of data	Yes
The web application can show an overview over all the patients and weather they have completed their goals	No, not implemented due to lack of time
The web application can show a view for each patient with all his data and history of that data	Yes
Each patient can see the goals created for him/her	No, the patient can not see the goal set by the clinician in RSAT. But if the patient set the same goal in the Lifelog application the goals can be seen there. Limitations in lifelog makes it impossible to synchronize goals between RSAT and the Lifelog application.
It is possible to see if a patient has been inactive for too long from the web application	No, it is not possible. It does not exist data about inactivity in the Lifelog API. If a user is inactive, no data is logged. This means that we cannot in real time know whether the user in inactive or if the user has lost its connection to the cloud or between the smartphone and the smartband.
Possible to integrate other wearable devices from other manufacturers	Yes. This can be done by implementing a service with commands similar to the Lifelog Service in RSAT and by changing the assembly
Devices are reusable	Yes, but a Lifelog account have to be created for each patient.
Integration with the planning system	No, due to lack of time but it can easily be done because both systems are built on the Palcom platform.

Table 7.1: Table showing whether a requirement was fulfilled or not.

As can be seen from the table, two of the requirements could not be fulfilled due to limitations in the Lifelog API and two could not be implemented due to a too tight time frame.

7.2 The Lifelog platform

In this section we will discuss the pros and cons of the Sony Lifelog platform from the perspective of its use in RSAT.

What is great about the Sony Lifelog platform is that its API is really easy to use. The API is REST-based, have a good documentation and all the data fetched from it have the same type of syntax. This makes it easy to both query the API and to parse its responses. However, even though the API is easy to use, it lacks some functionality, which, if existed, would have simplified the development of RSAT and also made RSAT more user friendly.

Three problems we found during the development of RSAT, and how they was solved, are described below:

- It is neither possible to read nor to write goals for a user via the Lifelog API. This was troublesome in the development as the requirements for RSAT stated that the clinicians should be able to both read and set the goals for their patients. The solution to this problem was to create a separate database in RSAT where the clinicians have to enter all the goals for their patients. These goals must also be manually added in each patient's Lifelog Application, such that the patients can see if they reach their goals or not. This creates a double maintenance problem, which increases the risk that some goals are incorrect. It also makes RSAT less easy to use for the clinicians.

- It is not possible to send any custom data to the Lifelog application. This made it impossible for the clinicians to send custom made feedback messages to their patients. The solution to this problem would have been to built a standalone application to be used on the smartphone instead of the Lifelog application but this was not done as it would have taken too much time.

- A user of the Lifelog platform must have a Sony or a Google account. This means that each time a patient is added to the RSAT-program, a new Sony or Google account has to be created. The alternative would be that the patient used his or hers own account, but we can neither make the assumption that all the patients have such an account nor that they are willing to create one.

We propose two alternative ways to solve the problems listed above.

The first solution would be if Sony creates a public low level API on their smartphones, which makes it possible to fetch data directly from the wristband in use. This solution would make it possible for us to create a custom application, which fetches data directly from the wristband without the need of connecting to the web API. This would eliminate the need for a user account since the application has direct access to the data in the wristband via the low level API. Having our own application for RSAT would also make it possible to build better functionality for goal handling and make it possible for the clinicians to send custom feedback to their patients.

The second solution would be that Sony changes their Lifelog API such that the goals created in the Lifelog application can be accessed and that it is possible to write data to it. Such a change would make it possible for us to implement functionality for the clinicians to get or set the goals for a patient directly via the web application in RSAT.

If it was possible to write data to the API, and if this data could be visualized in the Lifelog application, it would be possible for us to implement functionality such that the clinicians could send different kinds of feedback messages to their patients via the API.

To solve the third problem we listed above, Sony would have to change the API to have a device-focused approach instead of a user-focused approach. Instead of every user having an account, every device could have a unique ID and every device could be associated with one account that belonged to RSAT. RSAT would then only need one account to connect to all devices in the system. With this solution in use, the retrieval of data from a device would be easy. The only thing needed to be done would be to specify the ID of the device in the request and then the API would know which data is should respond with. This would make it easy to change the user of a phone because all notion of a user could be handled in RSAT.

7.3 RSAT

RSAT is built such that it shall be possible to integrate other activity trackers in future releases of it. However, since all data in RSAT is fetched from the Lifelog platform, adjustments has been made such that RSAT has a similar data structure of the data types used as those in the Lifelog platform. This means that if other wearables from other manufacturers shall be connected to RSAT, a standard describing the syntax of each data type must be implemented, such that all the data collected from different sources are uniform in the database.

This version of RSAT fetches data from Sony's Lifelog cloud and saves it to its own database. This might seem strange but the reason to this is to enhance the user experience. According to Nielsen, author of the book Designing Web Usability [30], the response time from a web service shall not exceed one second. In a test where we made 100 queries to the Lifelog API (see figure 7.1) we found out that the mean response time was 433 milliseconds, but the maximum response time was 3427 millisecond and the tenth largest response time was 1765 milliseconds. When we did the same test on our own database, the maximum response time was 23 milliseconds with a mean of 16 milliseconds. This was the reason why we chose to save the data in a database. Having our own database also enabled us to make large calculations of the activities for all users in one batch.

RSAT does not have as many features as similar systems, but it has a modular architecture, making it possible to add more features to it in the future. RSAT is also made to be integrated with an already existing system at Region Skåne. The possibility of making it tightly integrated with the already existing IT systems, is a thing that separates RSAT from the similar systems we have seen.

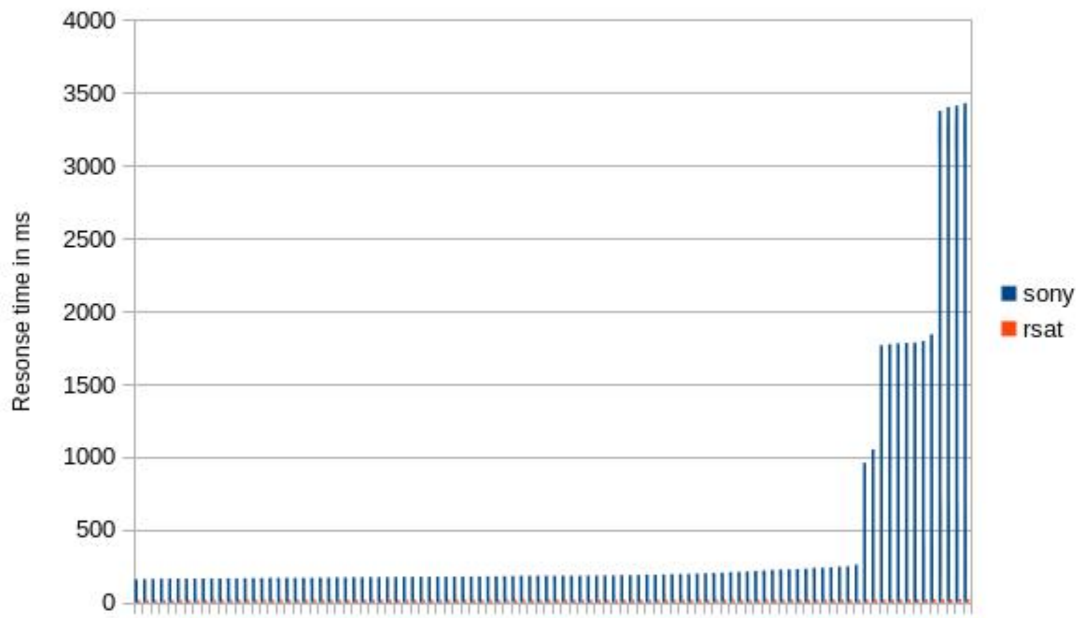


Figure 7.1: The response time for a query from the Sony Lifelog API compared to the response time for the same query to the database in RSAT.

Chapter 8
Conclusions

This chapter contains answers to the problem formulation stated in section 1.3 in the introduction chapter. It also contains a summary of the thesis and future work.

8.1 Research questions

Our problem formulation for this thesis (stated in section 1.3) consisted of three different questions. In this section, each of these questions are answered separately.

How well suited is the Sony Lifelog platform to be used in the activity tracking health care from a technical standpoint?

The Sony Lifelog platform has proven to work decently as a data source in this thesis and its smartphone application has a good-looking user interface and is easy to to use. For this two reasons we consider the platform to be suited to be used in the activity tracking health care. But we have also seen some disadvantages with the platform. One disadvantage, as discussed in section 7.2, is that the platform forces every patient to have a unique Lifelog account. Another disadvantage was that the Lifelog API did not contain all the data that was needed to implement two of the requirements stated on RSAT. Why these two requirements could not be implemented are briefly described below (for more information, see section 4.3 in the evaluation chapter):

- It was not not possible to read the activity goals created for a user from the Lifelog API and it was therefore not possible to make a solution that can guarantee that the correct goal is shown for the clinician.
- It was not possible to create a real time solution that could tell the whether a patient was inactive or not. In the API it is completely transparent whether a user is inactive or if the user have not synchronized its data with the Lifelog cloud.

We conclude that the Sony Lifelog platform can be used in the activity tracking health care. However, we also conclude that there is room for improvements for Sony to do on its platform in order to make it better suited for activity tracking health care.

How well suited is RSAT, the system to be developed, to be used in the activity tracking health care?

We believe that RSAT is well suited for activity tracking health care. The reason for this assumption is that the development has been performed in collaboration with a physiotherapist and that most of the requirements initially stated for RSAT are fulfilled.

What properties should an activity tracking device ideally have in order to be used in the activity tracking health care?

Some of the initial requirements/wishes from the physiotherapist on RSAT had to be removed due to limitations in the Lifelog API. This shows how important it is for an activity tracking device to be used in the activity tracking health care to have a rich API. A rich API makes it possible for the developers to create the optimal integration into the activity tracking health care. Ideally, the API should not only include the analyzed data, but also the raw data. If the raw data is available as well, it might be possible to find out more information about the physical activity performed and therefore implement more functionality.

Another property that a tracking device should have is a display such that a patient could get as much feedback as possible.

The smartband in use in RSAT has an accelerometer as its main (and only) sensor. If more sensors was included on the smartband more data, and more reliable data could have been calculated. We therefore conclude that an activity tracking device that shall be used in the activity tracking health care shall have as many sensors as possible.

8.2 Future work

Future work to be done in RSAT is to fulfill all the initial requirements that was setup in collaboration with the physiotherapist. There are two major requirements that needs to be fulfilled. (1) An overview view where the clinicians can see whether the patients have fulfilled their goals or not needs to be added to the system. (2) Integrate RSAT with the planning system that are in use at Region Skåne today.

Future work to be done beyond the fulfillments of the requirements is to integrate more activity trackers with more sensors into the system. Perhaps an activity tracker that is not as closely tied to its manufacturer's platform as the SWR10 could be integrated and used instead in the future. This would make it easier to enable the clinicians to send feedback messages to their patients and to handle their goals. This could also be accomplished if a separate application was developed and used instead of the Lifelog application on the patients' smartphones.

The ideal solution would be that a patient did not need to carry a smartphone in order to synchronize it's wristband's data. As a future work an investigation of using a so-called home hub instead of a smartphone could be performed. The home hub is developed as a part of the itACiH project and used in some other areas in the health care today. The home hub is used to connect different medical sensors in the home of the patients and it would therefore be a logical step to also connect activity trackers to the hub and use it in the same way as the vivohub by Garmin is used. In this case a smartband with a display as described in section 3.1.3 would be preferred.

To make it possible to fully use RSAT in the health care, security aspects of the system has to be further investigated. Also different legal aspects of the system have to be studied by lawyers. Last but not the least the medical effects of such systems has to be investigated and analyzed.

If RSAT proves to increase the physical activity of the patients, there is a chance that it could help other people with chronic diseases as well. This is an investigation to be performed as a future work of this thesis. Other studies can also include validating the accuracy of the sensor data.

8.3 Summary

Using the Lifelog platform we were able to build a web application that a clinician can use to track the physical activity of patients with chronic kidney disease. Some limitations in the Lifelog platform was found during the development and it was concluded that there exists other platforms that could be used instead of the Lifelog platform to make RSAT even better in the future. It was chosen that RSAT should save all the data queried from the Lifelog API in its own database. This increased the fetch time to be 28 times faster, which in one hand can be seen as a success for this thesis. On the other hand, all wishes from the physiotherapist could not be fulfilled, which could be seen as a failure. But hopefully this project will continue in the future and help patients with chronic kidney disease to attain the physical shape that is needed in order to undergo a kidney transplant. As this summary is written, RSAT are in use in a pilot program with real patients to study what effects the system might have on their weight loss. If the program is seen as a success, it is likely that it will be used in a bigger scale already next year.

Bibliography

[1] Tactio health groupe. *The Journal of mHealth*, 1(1):14–15, 2014.

[2] Angel. Faq | angel open sensor wristband. http://www.angelsensor.com/faq/. [Online; accessed 24-April-2015].

[3] Apple. The healthkit framework. https://developer.apple.com/library/ios/documentation/HealthKit/Reference/HealthKit_Framework/index.html#//apple_ref/doc/uid/TP40014707. [Online; accessed 10-February-2015].

[4] Katherine Baicker, David Cutler, and Zirui Song. Workplace wellness programs can generate savings. *Health affairs*, pages 10–1377, 2010.

[5] Julianne Callaway and Tim Rozar. Quantified wellness. *Research Bulletin*, 2015.

[6] Cecilia Sundqvist. Översikt , dialys, peritonealdialys. http://www.vardhandboken.se/Texter/Dialys-peritonealdialys/Oversikt/. [Online; accessed 06-May-2015].

[7] Fitbit. Fitbit for corporate wellness. https://www.fitbit.com/content/assets/wellness/FitbitWellness_InfoSheet.pdf. [Online; accessed 21-April-2015].

[8] David Svensson Fors. *Assemblies of Pervasive Services*. PhD thesis, 2009. http://www.ist-palcom.org/publications/files/AssembliesOfPervasiveServices.pdf.

[9] Fruit Street Health. Fruit street health launches new virtual lifestyle medicine clinic solution. http://www.reuters.com/article/2015/01/06/ca-fruitstreet-launch-idUSnPn3h2cBX+83+PRN20150106. [Online; accessed 22-April-2015].

[10] Garmin. Garmin | corporate wellness solutions. https://sites.garmin.com/en-US/wellness/. [Online; accessed 23-April-2015].

[11] Garmin. Garmin announces the garmin connect wellness portal. http://garmin.blogs.com/my_weblog/2015/01/garmin-announces-the-garmin-connect-wellness-portal-.html#.VUZhWfntlBc. [Online; accessed 06-May-2015].

[12] Garmin. vivohub. https://buy.garmin.com/en-US/US/shop-by-accessories/other/vivohub-/prod160901.html. [Online; accessed 06-May-2015].

[13] Gartner. Gartner says in 2015, 50 percent of people considering buying a smart wristband will choose a smartwatch instead. http://www.gartner.com/newsroom/id/2913318. [Online; accessed 24-April-2015, published 18-November-2015].

[14] Google fit. Fitness data types. https://developers.google.com/fit/android/data-types. [Online; accessed 04-April-2015Y].

[15] Google fit. Platform overview. https://developers.google.com/fit/overview. [Online; accessed 04-April-2015Y].

[16] Google fit. Terms and conditions. https://developers.google.com/fit/terms. [Online; accessed 04-April-2015Y].

[17] Fangfang Guo, Yu Li, Mohan S Kankanhalli, and Michael S Brown. An evaluation of wearable activity monitoring devices. In *Proceedings of the 1st ACM international workshop on Personal data meets distributed multimedia*, pages 31–34. ACM, 2013.

[18] IBM. Smarter care. http://www.ibm.com/smarterplanet/global/files/se__sv_se__smarter_healthcare_magasin_2014.pdf. [Online; accessed 21-April-2015].

[19] Internet Engineering Task Force (IETF). The oauth 2.0 authorization framework. https://tools.ietf.org/html/rfc6749. [Online; accessed 25-January-2015].

[20] Jawbone. Custom. https://jawbone.com/up/developer/endpoints/custom. [Online; accessed 04-April-2015Y].

[21] Jawbone. Get started building on up. https://jawbone.com/up/developer/. [Online; accessed 04-April-2015Y].

[22] Jawbone. Up for groups | corporate wellness by jawbone. https://groups.jawbone.com. [Online; accessed 23-April-2015].

[23] Joseph Kuan. *Learning Highcharts*. Packt Publishing Ltd, 2012.

[24] Laurence Girard. Fruit street virtual wellness clinic demo. https://www.youtube.com/watch?v=IAIRtOd6tRM. [Online; accessed 22-April-2015].

[25] Leanna K. Spreading the fitbit love to the workplace. http://blog.fitbit.com/spreading-the-fitbit-love-to-the-workplace/#i.18qhbt4seed9lu. [Online; accessed 06-May-2015].

[26] Lee Bell. Wearable technology cannot be trusted.

[27] Leslie Williams. Arianna huffington joins jawbone as advisor. `https://jawbone.com/blog/arianna-huffington/`. [Online; accessed 06-May-2015].

[28] Moment. Moment.js. `http://momentjs.com/`. [Online; accessed 10-May-2015].

[29] National kidney foundation. About chronic kidney disease. `https://www.kidney.org/kidneydisease/aboutckd`. [Online; accessed 06-May-2015].

[30] Jakob Nielsen. User interface directions for the web. *Communications of the ACM*, 42(1):65–72, 1999.

[31] NKDEP. Slow progression and reduce complications. `http://nkdep.nih.gov/identify-manage/manage-patients/slow-progression.shtml`. [Online; accessed 06-May-2015].

[32] Addy Osmani. *Developing Backbone. js Applications*. " O'Reilly Media, Inc.", 2013.

[33] P4 Stockholm. Pengar att spara på självdialys. `http://sverigesradio.se/sida/artikel.aspx?programid=103&artikel=1270709`. [Online; accessed 21-May-2015].

[34] Cesare Pautasso. Restful web services: principles, patterns, emerging technologies. In *Web Services Foundations*, pages 31–51. Springer, 2014.

[35] Thomas Sandholm, Boris Magnusson, and Björn A Johnsson. The palcom device web bridge. *Technical report, LU-CS-TR: 2012-251*, 2012.

[36] Diane J Skiba. The connected age: Digital tools for health. *Nursing Education Perspectives*, 35(6):415–417, 2014.

[37] Sony. Activity tracker android app from sony. `http://www.sonymobile.com/global-en/products/smartwear/smartband-swr10/`. [Online; accessed 12-March-2015].

[38] Sony. Authentication | sony developer world. `URL`. [Online; accessed 25-JANUARY-2015].

[39] Sony. Lifelog api | sony developer world. `https://developer.sony.com/develop/services/lifelog-api/lifelog-api-overview/`. [Online; accessed 12-March-2015].

[40] Tactio. Tactiorpm™ system. `http://www.tactiosoft.com/tactiorpm/`. [Online; accessed 17-April-2015].

[41] The kidney foundation of canada. Facing the facts. `http://www.kidney.ca/facing-the-facts`. [Online; accessed 06-May-2015].

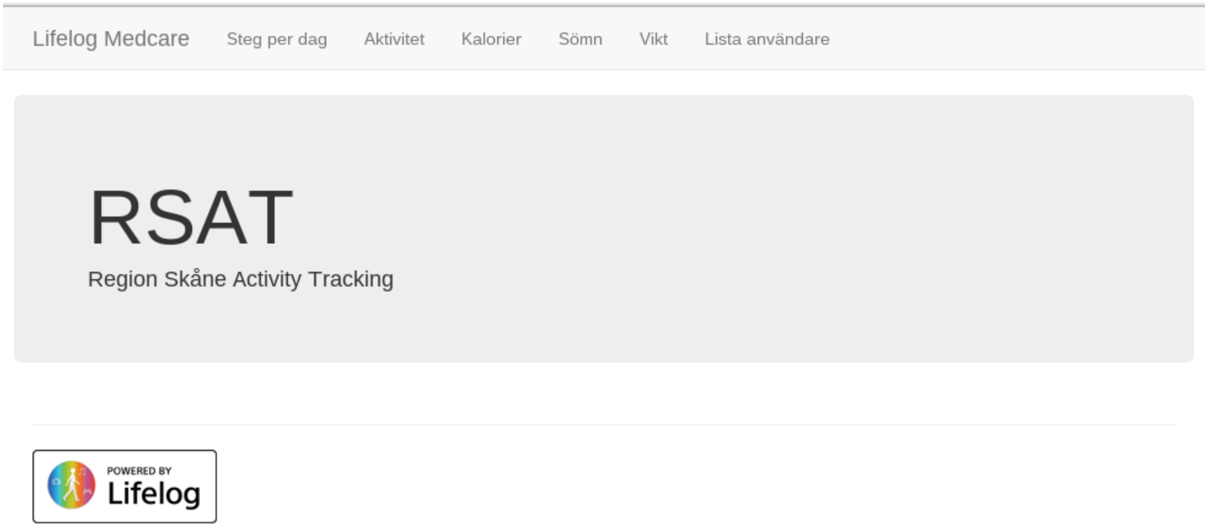
Appendices

Appendix A
Manual for RSAT

Manual för RSAT:s webbapplikation

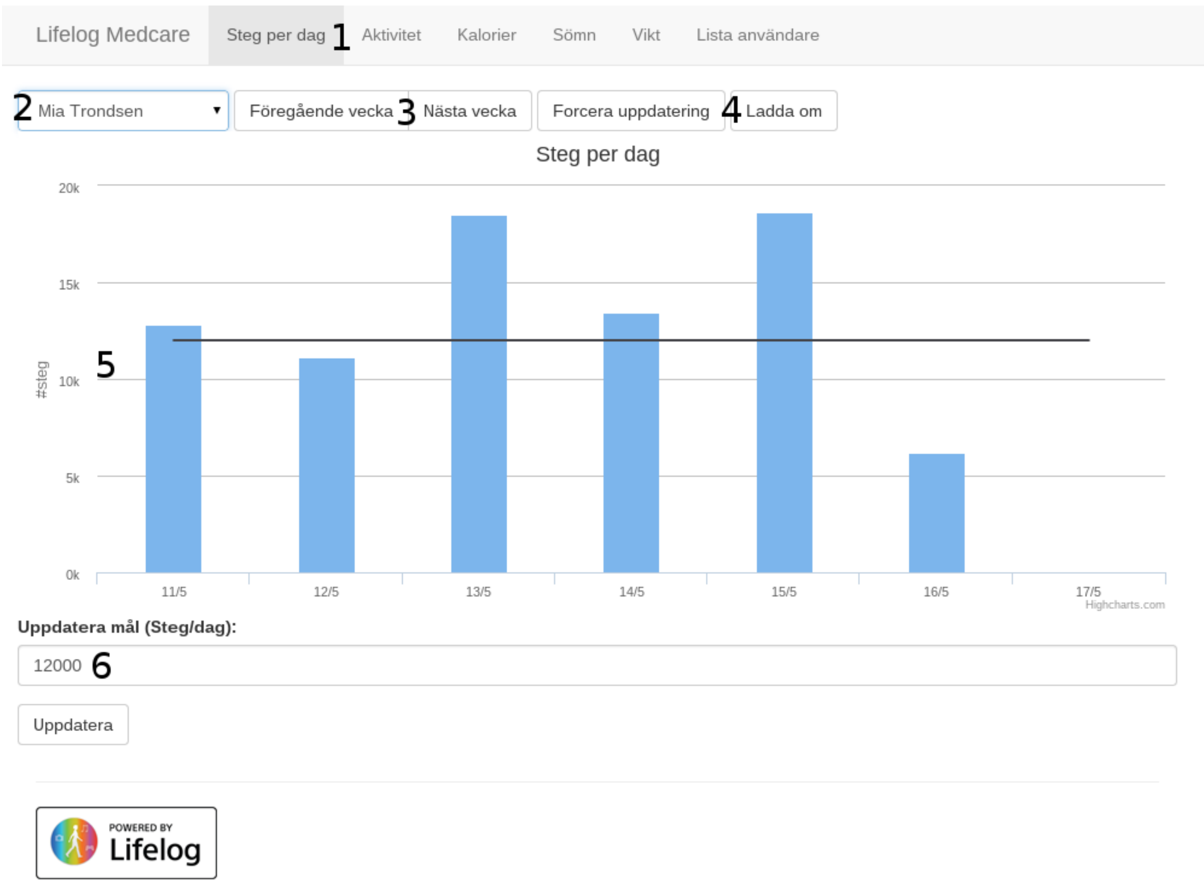
Observera att denna manual är gjord för stand-alone-applikationen som utvecklades innan RSAT integrerades i Region Skånes planeringsverktyg.

Startsida



Ovan visas startsidan, längst upp finns en meny där man kan välja mellan olika alternativ.

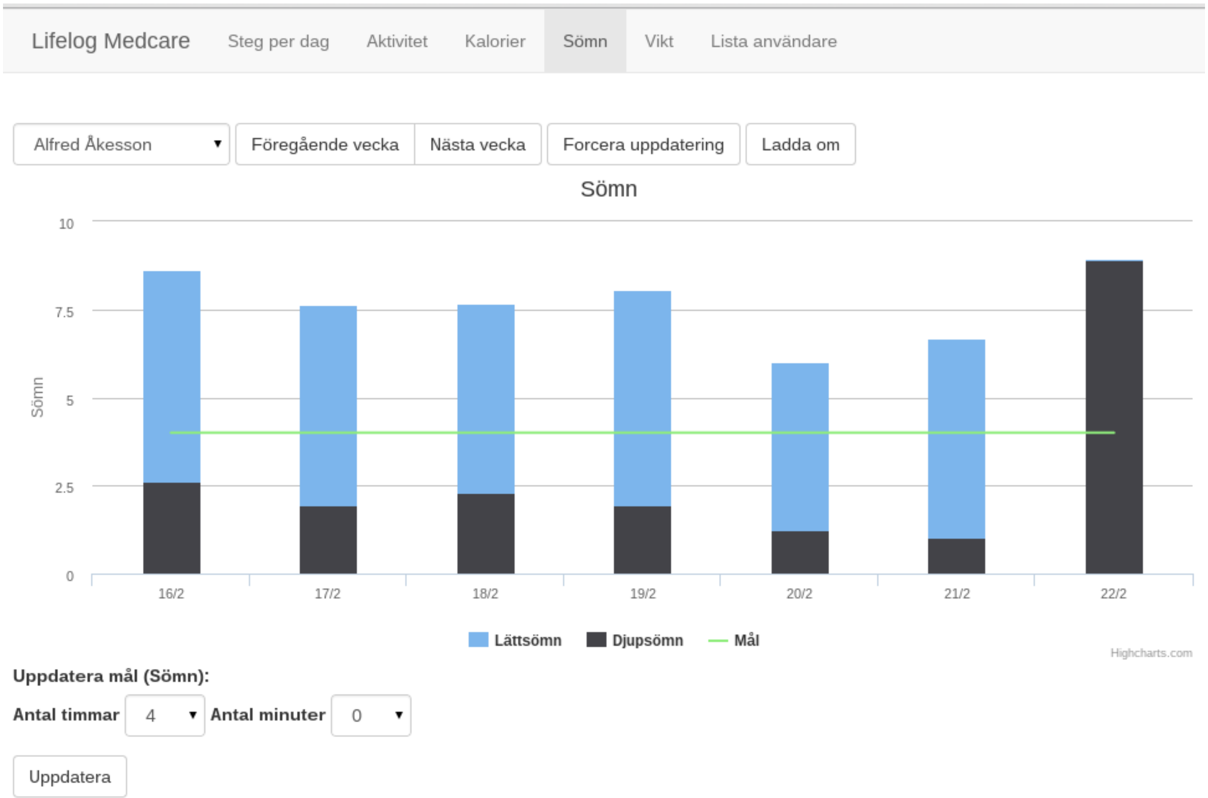
Stegdata



1. I menyn kan man välja vad man vill se på

2. Detta är en rullgardinsmenyn där man kan välja vilken användare man vill se.
3. Dess knapparna använder man för att gå en vecka fram eller tillbaka en vecka i diagrammet.
4. Knappen forcera uppdatering av datan i databasen (Senaste data i Lifelog-molnet hämtas ut och sparas i RSAT:s egen database). Genom att trycka ladda om kan man hämta den senaste datan från databasen och den visas i diagrammet i webbapplikationens vy.
5. Detta diagram visar hur många steg en användare har tagit varje dag i en vecka. Pelarana representerar antalet steg en användare har tagit under den dagen som syns under pelaren. Den svarta linjen är det mål som är satt i RSAT för hur många steg användaren ska gå om dagen.
6. Denna rutan används för att ändra målen, detta görs genom att skiva in det nya målet i textfältet och sedan trycka uppdatera.

Sömndata



Sömndiagrammet visa antalet timmar en användare har sovit under natten.
De blåa delen av pelarna visa andelen lättsömn och den svarta delen av pelaren visar andelen djupsömn. Den gröna linjen visa målet för hur mycket användaren ska sova under natten.

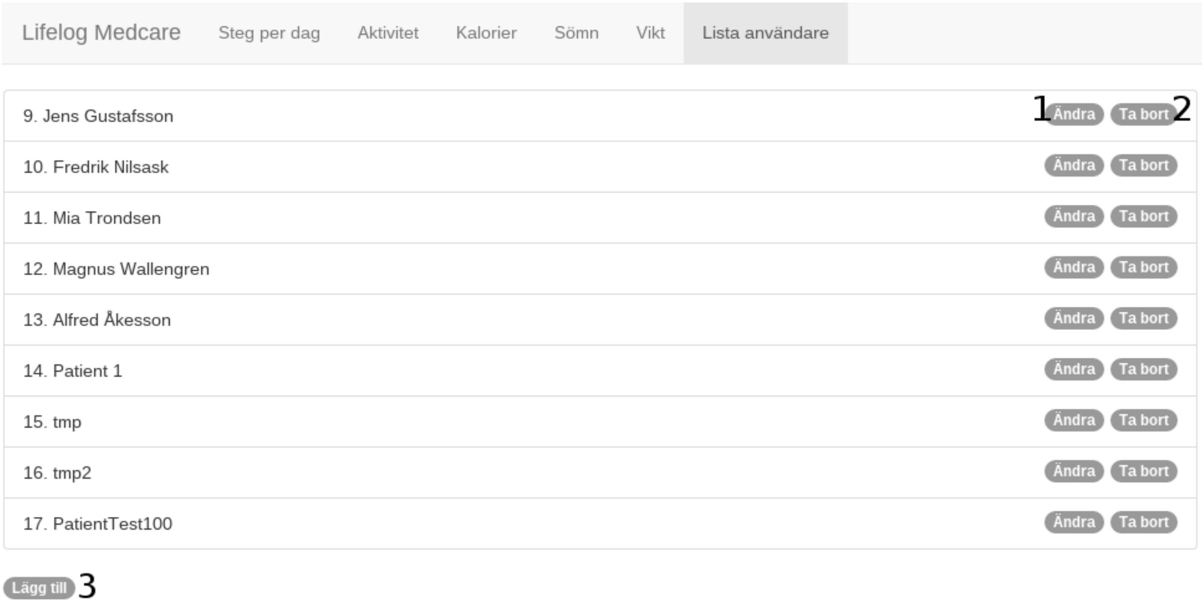
Aktivitetsdata

Det finns tre typer av aktivitetsdata för varje användare. Det är hur långt tid användaren går, cyklar och springer varje dag. Gångtig representeras av den gröna pelaren, löptid av den röda pelaren och cykeltid av den blå pelaren. Dessa aktiviteter har också mål som representeras av linjer i samma färg som stapeln för respektive aktivitets Mål setts för alla liknade sätt som för steg under diagrammet.

Övriga datatyper

I kalori finns ett diagram liknade som steg fast med antalet förbrända kalorier per dag. Vikt fliken visar en viktkurva.

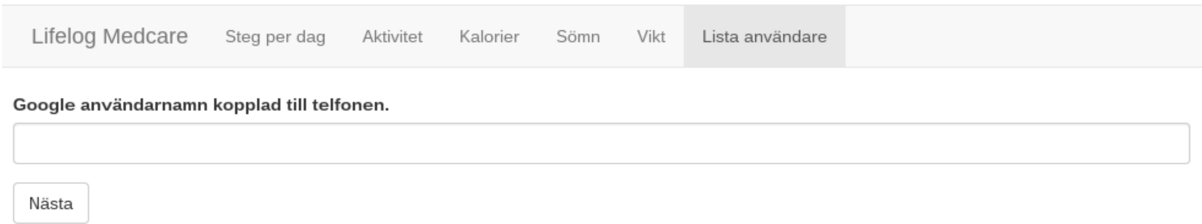
Lista användare



I Lista användare ser du en lista över alla användare i systemet.

1. Om du trycker på ändra knappen så kan du ändra namn på användaren på den raden på den raden.
2. Om du trycker på ta bort knappen tar du bort användaren för den raden.
3. Om trycker på lägg till kommer du till en sida där du kan lägga till nya användare.

Lägg till användare



Skriv in namnet på den användare du vill lägga till och tryckte på nästa. Nu kan du logga in på den användarens Lifelog-konto, antingen via ett Sony- eller Google-konto och däri ge RSAT tillgång till lifelog datan. Efter det blir du tillbakadirigerad till RSAT:s startsida och användaren är tillagd i listan som visas i vyn Lista användare.

EXAMENSARBETE System for activity tracking of patients with kidney failure
STUDENTER Jens Gustafsson, Alfred Åkesson
HANDLEDARE Boris Magnusson (LTH)
EXAMINATOR Görel Hedin (LTH)

Aktivitetsarmband i sjukvården

POPULÄRVETENSKAPLIG SAMMANFATTNING **Jens Gustafsson, Alfred Åkesson**

Aktivitetsarmband kan hjälpa patienter med njursvikt att gå ner i vikt. RSAT är ett system som visualiserar patienternas fysiska aktiviteter för deras vårdpersonal. Tack vare detta kan njursviktspatienter på ett effektivt sätt bli hjälpta att gå ner i vikt.

Många patienter med njursvikt är beroende av dialys för att klara av vardagen. Dialys är en omständlig process och tar mycket tid för patienten. Alternativet är njurtransplantation, men för att bli uppsatt på väntelistan för transplantation krävs det att patienten har en god fysik och att hen inte har för mycket bukfett. Risken att kroppen inte klarar av operationen är annars för stor. Bukfett innebär också att det blir svårt för läkaren att utföra transplantationen då bukfettet är i vägen.

Fram tills i år har det inte funnits någon hjälp att få för patienter som inte uppfyller kraven för att stå på väntelistan. Överviktiga patienter med dålig fysik har blivit instruerade att på egen hand gå ner i vikt och sedan ansöka på nytt om att bli uppsatta på listan. Patienter med njursvikt lider av orkeslöshet och har därför inte samma träningsmöjligheter som friska människor. Patienter som behandlas med påsdialys får dessutom i sig extra kalorier från dialysprocessen, något som gör det särskilt svårt att hålla vikten. En projektgrupp vid Skånes universitetssjukhus har därför bildats för att hjälpa denna grupp av patienter att gå ner i vikt. Som ett led i detta har RSAT utvecklats, ett system med uppgift att övervaka patienternas fysiska aktivitet.

RSAT använder Sony Lifelog. Sony Lifelog är en plattform som samlar in och sparar data om en användares dagliga aktiviteter. Genom att utrusta varje patient med ett Sony SmartBand (aktivitetsarmband) kan data om patienternas fysiska aktiviteter samlas in. Denna data synkroniseras till Sonys moln, varifrån RSAT hämtar data om varje användare. Uppgifterna visualiseras i ett webbgränssnitt för en sjukgymnast som kan följa patienternas framsteg. I dagsläget kan sjukgymnasten se veckodiagram för varje patients antal steg, förbrända kalorier, sömntid, gångtid, löptid och cykeltid. Detta hjälper sjukgymnasten att få insikt om vilka som gör framsteg och vilka som är i behov av mer hjälp för att lyckas med sin viktnedgång.

RSAT visar att det på ett relativt enkelt sätt går att bygga system som kan samla in aktivitetsdata om patienter. Fördelen med RSAT är dess arkitektur. Systemet består av flera olika minisystem (services). Varje service är helt oberoende av de andra och det är därför lätt att integrera fler källor att hämta data från i framtiden.

Under hösten kommer en pilotstudie att göras där RSAT används för att övervaka en grupp patienter med njursvikt. Systemet visualiserar data för sjukgymnasten som hen aldrig annars skulle känna till och medför en möjlighet att ordinera specifika träningsprogram. Genom att utnyttja den totala datan som RSAT samlar in kan sjukgymnasten få inblick i vilka träningsmetoder och ordinationer som ger bäst resultat för den enskilde patienten.