Design and implementation of a web application for rehabilitation planning for stroke patients with milder disabilities

Master Thesis

Sebastian Stocker
183112

Health Informatics Center
Karolinska Institutet, Stockholm
Supervisor: Maria Hägglund, PhD
Co-Supervisor: Nadia Davoody, MSc
Examiner: Sabine Koch, Professor

Medical Informatics
Heilbronn University & University of Heidelberg
Co-Supervisor: Martin Haag, Professor

Abstract

Background: Stroke rehabilitation is a complex process that requires collaboration between stroke patients and various health professionals. One important component of the rehabilitation is to set goals collaboratively with health professionals. The goal setting process can be time-consuming. In many cases, it is complicated for the patient and difficult to track for the health professionals. A simple user interface that supports patients, their family members and health professionals can help both sides to make the goal setting and attainment process easier.

Objectives: The aim is to design and develop a software for the goal attainment process of stroke patients with milder disabilities that facilitates goal setting process and the traceability of the goal progress for patients and health professionals.

Methods: Based on previous evaluated results, the web interface was developed and improved. Using this knowledge, a goal setting interface was added. To analyze the the goal setting process, goal attainment scaling (GAS) was included as well as parts of the International Classification of Functioning, Disability and Health (ICF) core set for stroke. The results were discussed afterwards in focus groups and evaluated based on two stroke patients, one family member and health professionals.

Results: We developed an interactive prototype, that can aid the rehabilitation at home by inserting problems with ICF codes and different kinds of goals, creating new activities and tracing goal progress by reviewing the different goals. With the help of the GAS the outcome of the patient's goals are visualized by a line chart presenting the positive or negative outcomes of the stroke rehabilitation.

Conclusion: The interactive prototype showed that it can support stroke patients during their rehabilitation at home. A usability test indicated that the goal setting and attainment process was perceived as useful for patients and their family members. Small improvements have to be made to simplify use and error handling. For health professionals, the prototype could also simplify the documentation process by using ICF in the prototype, and also improving collaboration when using the tool for coordination.

Keywords: stroke, rehabilitation planning, goal setting, design science, ICF, web application

Acknowledges

I would like to express my appreciation to my supervisors Nadia and Maria. Thank you for taking time out of your busy schedule to support me and to establish this thesis.

Thanks also to Thomas Wetter, who helped me establish contact with Stockholm, and to Sabine Koch, who gave me that opportunity to be part of her research team.

Thank you to the members of HIC and LIME for the great working atmosphere and the wonderful conversations during lunch and all the fikas.

A big thank you to the Lions Club Heilbronn, which helped me finance the exchange with Sweden.

Finally, a big thank you to my family for always supporting me and encouraging me when I needed the most.

Contents

List of Figures

List of Tables

List of Abbreviations

GAS	Goal attainment score
SMART	Specific, Measurable, Achievable, Realistic/Relevant and Timed
NICE	National Institute for Health and Care Excellence
ICF	International Classification of Functioning, Disability and Health
GWT	Google Web Toolkit
RIA	Rich internet application
JSNI	JavaScript Native Interface
RPC	Remote procedure call

1 Introduction

1.1 Motivation

For decades, stroke has been the major cause of long-term disabilities and the third reason for deaths worldwide. In 2012, 6.7 million people died from a stroke, which presents 11,9 percent of global deaths. By this count, stroke is the second leading cause of death [World Health Organization, 2014]. Despite a constant decline in stroke cases over the last three decades, risk factors, like obesity and high blood pressure, which reflect modern lifestyle changes, still lead to a high rate of stroke cases [Kinlay, 2011]. Aging populations also see higher stroke rates, as the risk of getting a stroke rises considerably with age [Sacco et al., 1997]. But younger persons are also not protected against strokes. Stroke has long been known as a disease of the elderly. However, over the last few decades, we are seeing an increase in the number of young people aged between 20 and 64 years, who suffer a stroke [Feigin et al., 2014]. Many young people get a stroke at the most productive period of their life, which hits them hard because they need to have jobs to support their families.

Regardless of age, a stroke can lead to different kinds and severities of disabilities. This is why the consequent stroke therapy is a challenge for every stroke unit. It necessitates high levels of effort from different health professionals, like physiotherapists, speech therapists or psychological therapists to identify a customized therapy for every patient that suffers from a stroke. These different health professionals should collaborate closely, during the therapy to know about the activities and progress of the patient, as they often complement each other [Norrving, 2007]. An important aspect of the rehabilitation of the patient is setting goals together with health professionals. These goals could either be essential for physical rehabilitation or wishes from the patient himself [Scobbie et al., 2011].

During the therapy, patients should also be involved in the rehabilitation process [Wade, 1992]. Most patients seek understandable information about the current status of their rehabilitation process [Wiles et al., 1998]. In this regard, a special software can improve this situation. On the one hand, this could support the patient during the rehabilitation with information about his care process. On the other hand, health professionals could then gather daily information from

the patient about his current activities, which were created together with other health professionals.

1.2 Objective and research question

A software that supports patients and health professionals during their stroke rehabilitation can be realized in many different ways. But a prototype, that can be handled by stroke patients, especially those with different disabilities, is a challenge. Hence, the resulting research questions are:

1. How can a software be designed to support stroke patients and health professionals during rehabilitation ?
2. How can the goal setting process be included in an application ?
3. Is the prototype usable for stroke patients ?

1.3 Previous work

This thesis is based on the work of further studies and theses that are all part of the PhD project "Improving collaboration and self-management in stroke care through ICT". In previous studies, the information needs of stroke survivors were identified to facilitate self-care [Woldemariam, 2013], and paper and interactive prototypes designed with a focus on appearance and usability for use in stroke patients [Vis, 2012] [Perlich, 2012].
As this thesis is based on the above-mentioned studies, parts of these studies have been used or complemented for this prototype.

2 Background

2.1 Stroke

A stroke is caused by cutting off the flow of blood and oxygen to the brain [World Health Organization, 2013]. There are two types of stroke: the ischemic stroke and hemorrhagic stroke. The ischemic stroke appears when an artery is blocked by either blood clots or plaque. 87 percent of all strokes are classified as an ischemic stroke. The hemorrhagic stroke is caused by a broken blood vessel and the subsequent blood that leaks into the brain. Although the hemorrhagic stroke accounts for just 13 percent of all strokes, it is responsible for 30 percent of all stroke deaths [National Stroke Association, 2014]. The risk for a stroke depends on several factors. To some degree, everyone has a slightly higher risk of getting a stroke. Lifestyle diseases, which are caused by an unhealthy way of living, play a big role in the risk to getting a stroke. High blood pressure, which is often caused by smoking, alcohol consumption or a sedentary lifestyle, leads to a higher risk [Bongers et al., 2006]. One important factor after a stroke attack is time. Two million brain cells die every minute during a stroke as long as the patient does not receive medical aid. Every second without aid increases the risk of permanent brain damage, disability or death [National Stroke Association, 2014]. A doctor should be called at the first suspicion, which can manifest itself in vision disorders, dizziness, brackishness, numbness, confusion, etc. . Since a stroke can occur in every area of the brain, the aftermath for each person cannot be generalized. Moreover, the duration of the undersupply of blood to the brain becomes important in matters of disabilities. The range of disabilities after a stroke could include disorientations, speech problems or paralysis of entire body parts. Overall, the impairments could be classified into physical, cognitive or psychological disabilities [Adams et al., 2007].

2.2 Stroke rehabilitation

2.2.1 General stroke rehabilitation

Since the impact of the stroke can influence every bodily area, the rehabilitation process cannot be generalized. Every stroke patient needs a different care plan tailored to the patient's demands. The actual rehabilitation of stroke begins in the acute phase, once the patient is in a stable condition in a stroke unit. The primary therapy targets maximizing the physical and cognitive abilities, and to reducing the amount of disabilities, so that a patient is able to return to a normal daily routine. In many cases, basic activities have to be learned again as other parts of the brain have to substitute the damaged brain areas. Since the impairments can appear in different bodily areas e.g. paralysis or speech disabilities the therapy has to be created in an interdisciplinary manner, with the involvement of different health providers and professionals [Duncan et al., 2005]. The Helsingborg Consensus Conference on Stroke Management, which took place in 1995 and 2006, examined the latest evidence-based knowledge in the management of stroke. For stroke rehabilitation, they identified the following involved professionals:

- Stroke physician
- Trained nurses
- Physiotherapists
- Speech therapist
- Occupational therapists

[Norrving, 2007]

The patient's family should also be involved in the rehabilitation process by learning the basic principles of rehabilitation.
Stroke rehabilitation is a continuing process. When the patient is discharged from the stroke unit and hospital, active rehabilitation is continued at home by a mobile rehabilitation team or by trained family members. Community based health services can bring a stroke patient back to an independent daily life. With respect to collaboration between health professionals, it is important that every member of the team is aware of the activities for the rehabilitation plan and supports these activities. Together with these different health professionals, goals should be formulated and linked with dedicated activities [Duncan et al., 2005].

2.2.2 Stroke rehabilitation in Stockholm

As this study was performed in Stockholm (Sweden), the results were worked out on the basis of local rehabilitation processes. This involves the rehabilitation process of the Jakobsberg primary care centre in Stockholm. Particular consideration has been placed on the rehabilitation at home. The local rehabilitation process does not differ much from process described already. After the acute phase, further rehabilitation is continued at home. This stage involves the stroke team from Jakobsberg, which consists of speech therapists, occupational therapists, a physiotherapist and a counsellor. These health professionals visit the patients at home to conduct exercises with them. The outcomes of the exercises are provided by the therapist orally to the patient, the documentation by the therapists is completed afterwards at home. Home rehabilitation is provided for several years after the stroke; afterwards, the patients are left on their own.

2.3 Goal setting in stroke care

Goal setting after a stroke is an essential part of stroke rehabilitation. It has been shown that goal setting has a positive outcome and is used to enhance patient motivation, adherence and autonomy as well as improve satisfaction with the rehabilitation, task performance, team work and evaluation progress [Sugavanam et al., 2013]. However, there is no *best way* how to handle an entire goal setting process. The National Institute for Health and Care Excellence (NICE) recommends that goals have to be meaningful and relevant to the patient, that they should focus on activity and participation, that the goals should be challenging but achievable and include long- and short-term goals [Swain et al., 2008]. Goals could be set in different kinds of areas. The main areas are physical rehabilitation, recreational/social, family/community and personal. In the best cases one goal is set for each of this these areas. Breaking each goal into smaller parts by creating short-term goals makes it easer for the patient to reach the goal.

To set a goal, four stages are needed to identify the four main parts of goal setting: problems, goals, activities and outcome.

1. Goal negotiation: The patient considers his current situation and identifies the main problems.
2. Goal setting: The identified problems(s) is refined into a specific, challenging rehabilitation goal.
3. Action planning: This details what the patient has to do to meet the goal
   Coping planing: This outlines the strategies to be activated if barriers hinder action plan

attainment.

4. Appraisal and feedback stage: This involves prompt progress review and getting or giving feedback as well as outcome measurement.

Those four steps should be created collaboratively involving all the people that are involved in the care process - namely the patient himself, the different health professionals and the patient's family. Here, it is important that the patient expresses his personal wishes and expectations, and discusses these with the health professionals and the family to prevent unrealistic goals. Non-attainment of goals is often inevitable, which can end in disappointment and frustration; hence, it must be regulated earlier. However, it can be helpful for the patient to realize his limitations and to reduce goals, breaking them down into smaller steps or even abandoning them altogether [Scobbie et al., 2011].

3 Methods and material

3.1 Design science

Johannesson and Perjons defined design science as the "scientific study and creation of artifacts as they are developed and used by people with the goal of solving practical problems of general interest" [Johannesson and Perjons, 2012], which emphasizes two things: solving problems and artefacts. Since the term 'artefact' is not self-explanatory, Johannesson and Perjons added a definition, describing it as an "object made by humans with the intention to be used for addressing a practical problem". They are also comparable to an architect's plan for a building, which has an inner construction, an outside, which is represented by the environment in which it is located, and an interface between the inner side and the outside, which provides the functionality. To find a solution, which is nothing other than an artifact, for a practical problem, different steps are necessary:

1. Problem investigation: explicate/describe the problem, explain it, predict what would happen if nothing else is done about it.
2. Solution design/outline of artifact: (no solution will be found): the means/requirements to reach a solution for the explicated problem
3. Design and develop artifact: create an artifact that addresses the explicated proof
4. Demonstrate artifact: use it in an illustrative or real-life case
5. Artifact evaluation: fulfills the artifact the requirements; brings the specified design closer to the goal; three-questions to answer

[Wieringa, 2009]

The method to solve a problem and to create an artifact is can be solved with the user-centered design method. It is based on the understanding of users, their tasks and environment. As the main focus is on the user, he is involved throughout the design and development process. Result are evaluated using usability test. Afterwards, the resulting problems are discussed and improvements are derived from it [Jokela et al., 2003]. The whole user-centered design is an iterative process (cf. Figure 3.1).

As this thesis is part of a project, it does not cover the whole process. The requirement as well as parts of the design have already been made in previous theses, which resulted in paper-based and static prototypes. However, as this thesis discusses new topics, the different steps of the user-centered design model are included and supplemented by new findings of this thesis.

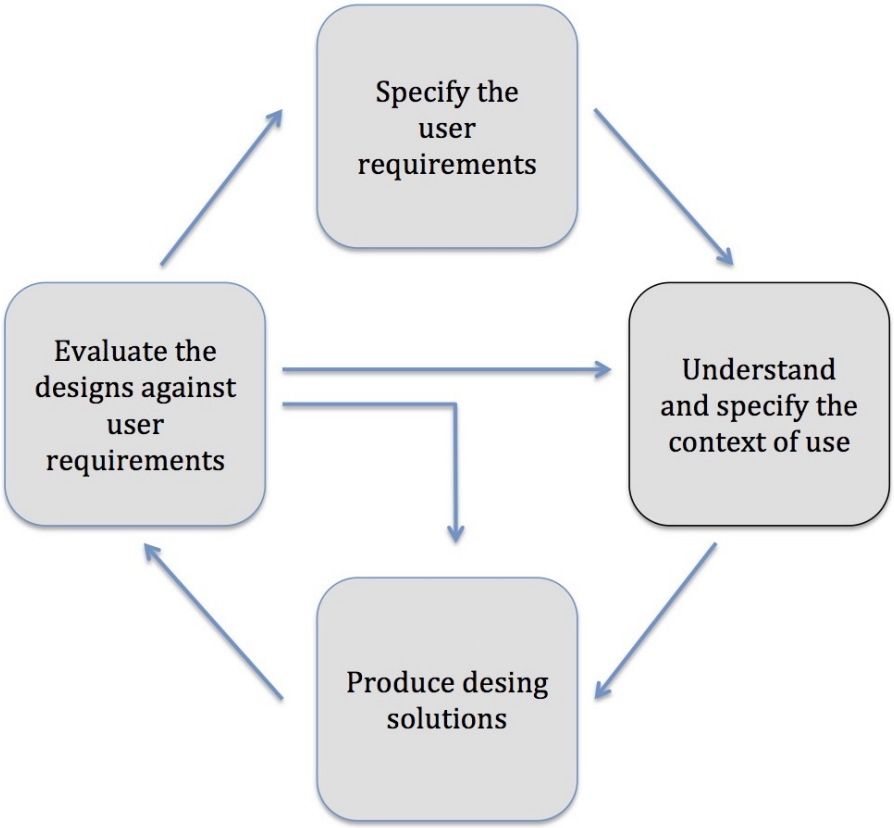


Figure 3.1: User-centered design model [Jokela et al., 2003]

3.2 Care plan / CONTsys / OLD@HOME

The structure for the goal setting process consisting of a problem, goal, activity and outcome was described in the OLD@HOME project for a shared care plan. It was applied from the standardized CONTsys and the reference model of openEHR to a shared care plan that fulfills the requirements to support collaboration among different actors in home-care of the elderly [Hägglund et al., 2011]. As this application might support stroke patients after they being discharged from the hospital and support collaboration among different health professionals by sharing the care plan, this model was reused to model the care plan of stroke patients.

3.3 ICF core set for stroke

The ICF core set for stroke is the International Classification of Functioning, Disability and Health released by the WHO in the year 2001. It was developed to create a general agreement between different guidelines, to facilitate the comparability of research findings and improve stroke care. The list contains different characteristics of stroke survivors by describing the bodily functions, body structure, activities and participation, and environmental factors. Every classification has its own ICF code and is mainly used for the documentimg a stroke patient in care. The core set contains 41 categories for the component body functions, five for body structure, 51 activities and participations, and 33 environmental factors. [Geyh et al., 2004]
Although the use of ICF is widespread, the handling to find the belonging code can take much time. Including the use of ICF codes directly in an electronic rehabilitation plan would simplify the documentation process.

3.4 Goal attainment scaling (GAS)

The measurement of GAS was developed by Kirusek and Sherman [Kiresuk and Sherman, 1968]. It was developed to measure the success of a goal attainment process. At several review points, the goal attainment process is reviewed and evaluated with inputs from the patient and the different health professionals. For GAS to be used, the goal must be measurable. Therefore, the patient has to maintain different things. A good way to specify a goal was developed by Schut [Schut and Stam, 1994]. He figured that goals have to be specific, measurable, achievable, realistic/relevant and timed (SMART). SMART goals can then be used for the GAS method to evaluate the patient's progress and decide on the further steps of the patient's rehabilitation plan. To start with the goal setting process, five steps are recommended:

1. Defining the expected goals:
   In these steps, the patient himself, or together with a health professional, starts to define her/his personal goals. Therefore, the main problems of the patient have to be identified and described as a SMART goal.

2. Weighting the goal
   To evaluate the goal, each goal needs two different weights: the importance and the difficulty. To rate the importance and difficulty, a four-point scale, as shown in Table 3.2, is used.

Importance	Difficulty
0 = not at all (important)	0 = not at all (difficult)
1 = a little (important)	1 = a little (difficult)
2 = moderately (important)	2 = moderately (difficult)
3 = very (important)	3 = very (difficult)

Table 3.1: Four point scale of importance and difficulty

3. Definition of the expected outcome:
   In this part, the health professional defines the probable outcome after the treatment. This has to be done objectively and based on observation to inhibit unrealistic expectations of the patient [Turner-Stokes, 2009].

4. Score baseline:
   The baseline is needed to set the conditions of the patient before he starts to train for the goal. Usually, the baseline is rated with -1 if the patient is already able to do some things. If the patient is not able to do anything, then the baseline is set to -2.

5. Goal attainment scoring and GAS calculation:
   In the last step, the outcome is rated and the GAS score is calculated. On previously defined review points, the progress of the patient is evaluated in most cases with the help of a health professional. To assess the goal progress, a five-point scale is used to arrive at the degree of attainment of the goal:

   - If the patient achieves the expected level, the score is 0.
   - If the patient achieves better than the expected outcome, the score is +1 for somewhat better or +2 for much better.
   - If the patient achieves worse than the expected outcome, the score is -1 for somewhat worse or -2 for much worse.

   [Turner-Stokes, 2009]

After the evaluation and the previously set values, the overall GAS score can be evaluated using the following formula:

$$GAS = 50 + \frac{10 \sum(w_i x_i)}{\sqrt{((1-\rho)\sum w_i^2 + \rho(\sum w_i)^2)}} \tag{3.1}$$

where
w_i is the weight assigned to the i-th goal (importance $\times$ difficulty).
x_i is the value of the assessed goal (between -2 and +2).
ρ is the expected correlation of the goals, which is set to 0.3, in accordance with Kirusek and Sherman [Kiresuk and Sherman, 1968].

3.5 Evaluation methods

3.5.1 Focus group

The first part of the evaluation was a focus group meeting (cf. [Berg et al., 2004]). This was carried out over three focus group meetings with the neurology team from the Jakobsberg primary care centre. The stroke team consists of six persons, two physiotherapists, two speech therapists , one counselor and one occupational therapist. Purposive sampling was used to select the stroke Jakobsberg stroke team, since this team fulfills the criteria to get specialist knowledge for this project [Jupp, 2006]. The meetings were held in Swedish, protocoled by researchers and transcribed verbatim afterwards. Additionally the interviews were recorded on tape. Each meeting lasted one hour and started with a brief introduction on the topic by presenting the previous work and printed-out prototypes, followed by the explanation of the functionality of the actual interactive prototype. Thereafter, the neurology team was interviewed and asked for their opinion based on prepared questions.

3.5.2 Usability testing

Usability testing is part of the user-centered interaction design. It is used to evaluate prototypes to find problems that users have to control through functions of the software. Therefore, it is essential for the evaluation part of the design process (cf. section 3.1). A usability test can be performed in different ways, depending on the development progress of the software, or the target group. For this evaluation, a formative evaluation scenario has been selected, as described by

[Burmester et al., 2010]. The basis of this scenario is the observation of a patient while he executes different tasks and deriving usability problems from the context of use. To gather the problems, different methods can be used, like a written protocol, recording of the screen and the statements of the patient via the think-aloud method [Nielsen, 1994], and standardized questionnaires.

For evaluating this prototype two stroke patients and one family member were given 10 tasks, that they were to complete in 30 minutes. To get information about the usability of the prototype, the whole screen along with all mouse movements and clicks were recorded using the free open-source software *CamStudio* [1] during the test. Additionally, the patients were asked to think aloud, about the different steps that they performed during the tasks. The statements during the test and the questions after the test were also recorded with an audio recorder.
After the test procedure, the results of the tasks were summarized and usability problems identified. Additionally, the severity of each usability problem was determined, according to the scale of Nielsen [Nielsen, 1995].

Rating	Description
0	I don't agree that this is a usability problem at all
1	Cosmetic problem only: need not be fixed unless extra time is available on project
2	Minor usability problem: fixing this should be given low priority
3	Major usability problem: important to fix, so should be given high priority
4	Usability catastrophe: imperative to fix this before product can be released

Table 3.2: Four-point scale of importance and difficulty

As the target group of this web application are stroke patient's with milder disabilities, the test-subject were gathered from a stroke organization in Stockholm that had already cooperated in different projects.

Person No.	Gender	Age	Time since first stroke	Use of electronic devices
1	male	61 - 70	2 years or more	computers
2	male	61 - 70	2 years or more	computers
3	male	61 - 70	- (family member)	computer, smart phones, tablet

Table 3.3: List of different test-subjects

1 Link: `http://camstudio.org/` (Last accessed 25 November 2014)

In order to achieve the same preconditions without influencing the test subject and to get a usable result, every moment of the 45-minutes test procedure was planned in different steps:

1. Short description
2. Introduction to the usability test
3. Introductory video explaining the prototype
4. Execution of the 10 different tasks
5. Answering the questionnaire containing six questions about the test subject
6. Answering eight questions regarding the prototype

The detailed test-procedure can be found in the Appendix section A.3.

3.5.3 Ethical consideration

All participants of the usability tests and focus group meetings were volunteers. Before the usability test and meetings, the participants received an information letter informing them about the study and the planned procedure. Before the actual usability tests and meetings, all participants signed a consent form to give a written agreement to use the data for this study. The interviews and meeting were recorded on tape. The verbatim transcription and the subsequent evaluation was made without using names to insure confidentially.

4 Implementation

4.1 Implementation environment

4.1.1 Development kits

The web-application was created with the help of the following development kits:

- Java 7 JDK from Oracle [1]
- Google Web Toolkit (GWT) SDK Version 2.6.0 from Google [2]

GWT 2.6.0 is compatible with only Java Version 7. Hence, Java Version 8 was not used.

4.1.2 Frameworks

- Google Web Toolkit (GWT):
  The GWT is an open-source framework for developing rich internet application (RIAs) and was developed by Google ™. RIAs can be compared with desktop applications and offer a similar range of functionality. This is accomplished through interactive elements, which can asynchronously reload data from a web server and present them by changing parts of the current view of a webpage. The actual application is developed in the programming language Java and is later compiled into pure JavaScript code. Hence, the web application can be used in standard web browser like Firefox, Chrome, Safari or any other web browser that is compatible with JavaScript. The development in Java also allows the developer to debug the code in Java, which makes the development process easier. The structure of the GWT framework consists of four main components:

1 Link: `http://www.oracle.com/technetwork/java/` (Last accessed 15 October 2014)
2 Link: `http://www.gwtproject.org/` (Last accessed 15 October 2014)

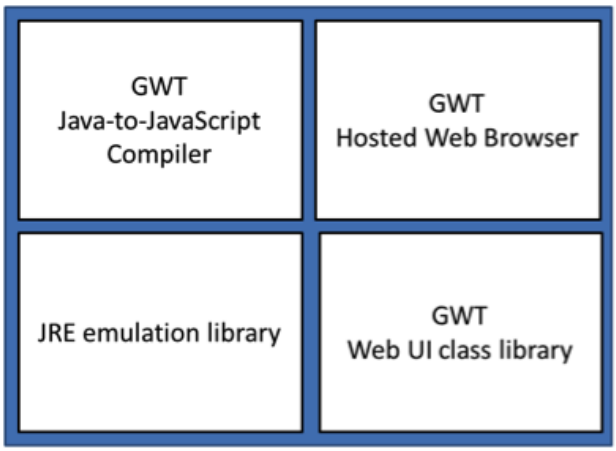


Figure 4.1: GWT system architecture

- The Java-to-JavaScript compiler translates the Java code to native JavaScript code.
- The hosted web browser allows the fast execution of the web application during the development without compiling the entire code in the so-called *hosted mode*. Thereby, the Java code is not translated into JavaScript code. During this process, the code runs as a Java application in a virtual machine and is presented in the browser via a browser plugin.
- The JRE emulation library contains all Java runtime classes that were translated into JavaScript and, therefore, can be used for a GWT application.
- The Web UI class library contains all the elements that can be used in the layout and design of the web application interface, like different layout panels, buttons or text boxes.

A GWT application has a strict package structure for the development. Every folder or package has its own function, which is considered by the compiler Figure 4.2.

- **src**: The src-folder is the overall folder for the different code of an application.
- **client**: In the client package is the code that is translated by the Java-to-JavaScript compiler into native JavaScript code and then used by the web browser.
- **server**: This code is not compiled by the Java-to-JavaScript compiler. It remains as native Java-code and is used by a server like tomcat or jetty.
- **shared**: The code in the shared-package is used by the server and the client and is, therefore, translated into JavaScript-code. Hence, domain classes can be used from the server and the client.
- **test**: The test folder contains classes - e.g. for JUnit - to test the application.
- **war**: The war folder contains the actual web application. It contains the entry HTML-file and the related CSS-files. Additionally this folder contains the compiled JavaScript code.

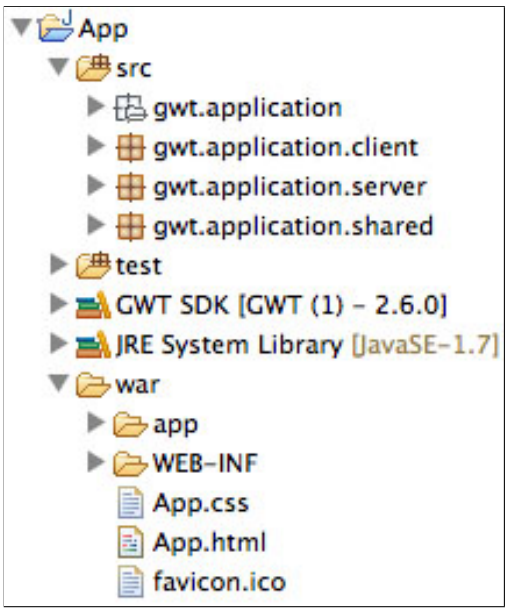


Figure 4.2: GWT folder structure

To create a webpage, GWT offers a basic quantity of web elements, like layout elements, buttons or text boxes. These elements can be composed via an XML file to a frame without functionality, which is added in the actual Java code. To give the page a unique look, the different elements can be designed using a cascading style sheet (CSS).

Although the major development is in Java, the GWT offers the possibility to use native JavaScript within Java Code. For this purpose, the Java method that contains JavaScript code is declared *native* and the JavaScript code is included in a special formatted comment block. An example is presented in Code 4.1.

```
1    public static native void alert(String msg) /*-{
2      //native JavaScript Code
3      $wnd.alert(msg);
4    }-*/;
```

Code 4.1: JavaScript Native Interface example

- GWT-Bootstrap:
  GWT-Bootstrap is based on the open-source JavaScript and CSS library Bootstrap developed by Twitter. GWT-Bootstrap uses this JavaScript library and creates, with the help of JSNI, Java classes around the JavaScript code that can be used with the GWT. As mentioned above, GWT provides only a basis of elements to create a webpage. GWT-Bootstrap offers, in addition, more elements and a modern design.
- GWT-chartJS:
  GWT-chartJS can be used to represent data in different chart types. It is based on a JavaScript library and was also adapted via JSNI for use with GWT. In this thesis, it was used to display the calculated GAS values in a line chart.

4.1.3 Additional tools

Apart from the above-mentioned development kits and frameworks, some other tools were used during the development.

- Apache Maven:
  Apache Maven is a build management tool based on Java and developed by the Apache Software Foundation. So-called archetypes are used to set up a software project. Archetypes are templates for different software projects - e.g. for a web application or a simple Java application - and contain the folder structure and library dependencies for the specific software project. A *pom* file is created for the specific software project, where additional libraries can be added, which are considered by the compiler during the building process. With different goal types - e.g. an assembly goal - the compiled code is packed in a file so that it can be used on different platforms.
- Git:
  Git is a version management tool that allows different developers to merge their programmed code and to manage the version that results from changes and conflicts. The code is uploaded to a local and a central repository, so that everyone is able to access the new code and code changes. In this work, Git was mainly used as a backup tool to store the code on a central server and prevent loss of data.

- Eclipse:
  All tools and frameworks were combined in the integrated development environment (IDE) Eclipse Version 4.3 Kepler [3]. To use all tools in Eclipse, the following plugins have been installed:
  - Google Plugin for Eclipse 4.3 [4]
  - m2e - Maven Integration for Eclipse [5]
  - EGit - Team Provider 3.4.1 [6]

4.2 Implementation details

The use of remote procedure calls (RPCs) in GWT can be realized in different ways. The common way is to use the GWT RPC framework, which is part of GWT. With the GWT RPC, it is possible to exchange Java objects over HTTP as the framework serializes every Java object. The implementation is based on the Java servlet architecture. However, GWT RPC are not the same as web services, like SOAP or REST. Therefore, a proprietary protocol though makes the implementation process easier as the domain classes can be used on the client and the server side. An exemplary architecture for the GoalService implementation is in Figure 4.3.

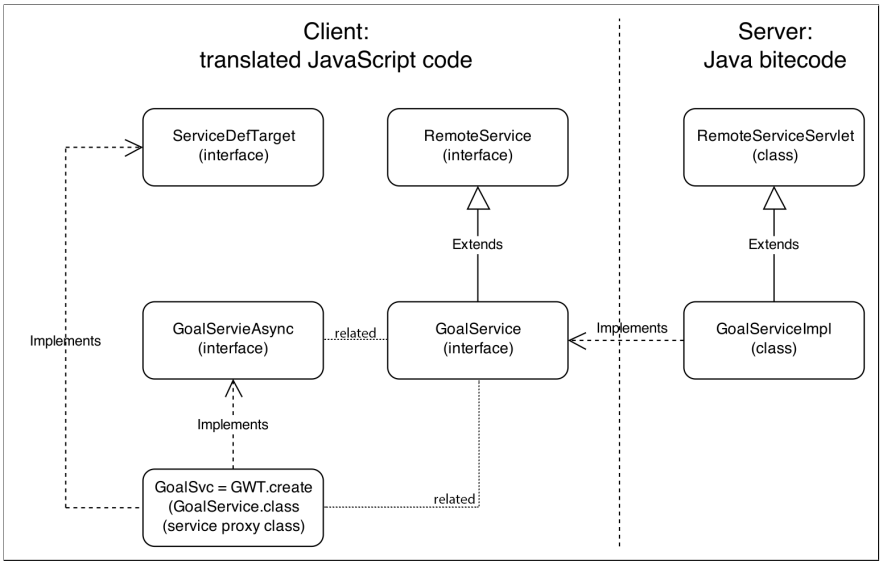


Figure 4.3: GWT-RPC architecture

3 https://www.eclipse.org/
4 https://dl.google.com/eclipse/plugin/4.3
5 https://www.eclipse.org/m2e/
6 http://www.eclipse.org/egit/

4.2.1 Client implementation

The client implementation consists of the main client package and the two additional subpackages, *events* and *views*. The subpackage *views* consists again of three subpackages: *dummy*, *modal* and *widget*. The full package structure is presented in Figure 4.4.

- Client-package:
  The main *client* package contains the overall structure of the applications' user interface, which consists of the Menu at the top and content at the bottom. The latter portion is organized from the Goals class and is the entry-point for the whole application. The entry-point class loads at the beginning of every class that is important to run the application. This includes the factory class, which manages every single view of the whole application and ensures that every view object is created just once.

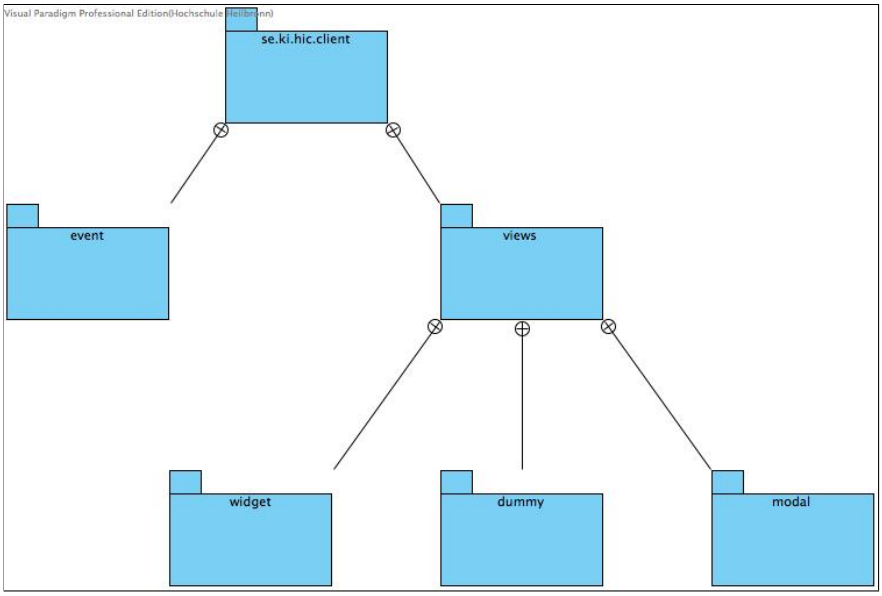


Figure 4.4: Client package structure

In collaboration with the static history class from the GWT framework, the controller class is responsible for loading the different views. According to the history tokens, which are collected in the enum class *RedirectString*, the controller class sets the selected menu link and replaces the content frame with the selected frame from the factory class.

The helper class is responsible for setting the font size in the whole application. With the help of three different buttons, the font size can be changed.

Additionally, the main client package contains the four service interfaces, which define methods that are implemented by the server classes and used from the asynchronous service in the client package (cf. Figure 4.3).

- Views subpackge:
  The subpackage *views* contains the different views, that can be selected from the menu bar. As the menu bar consists of five menu items, the package has five different classes. The different elements of the view were first described in an UIbinder class with XML and the functionality was added later with the help of the appropriate Java class. The functionality of every view is explained in the results chapter.

- Dummy subpackage:
  The dummy subpackage expands the actual view package with other views. In contrast to the *view* package the *dummy* package contains only static pages without functionality to demonstrate examples for data that were used in earlier prototypes but do not form the content of this work.

- Modal subpackage:
  This subpackge contains different modal windows, which are mostly used as a wizard to create/edit/delete problems, goals or activities. The windows were first described in a UIBinder class in XML and were completed with functionality in the appropriate Java class.

- Widget subpackage:
  In the widget subpackage are all element compositions that serve, in conjunction, a specific functionality. Every class inherits from the GWT class composite and can, therefore, be used in every class that offers the method addWidget(Widget widget). Every widget was first composed in a UIBinder class and extended with functionality with the proper Java class (cf. Figure 4.5).

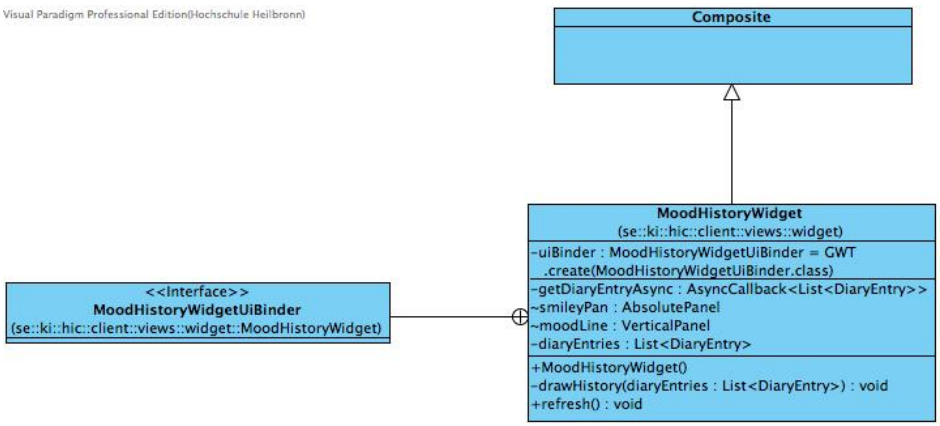


Figure 4.5: Class diagram example for one widget

- Event subpackage:
  As the different classes of views and widgets do not have direct dependencies, it would not be possible to have interaction among these classes. To handle this, the subpackage *event* was created. It contains classes to handle different events that occur between those widgets or views. The singleton class *EventBus* operates as a central connection, which is used to fire the specific events to the proper class (cf. Figure 4.6). The actual events are defined in the appropriate event class, where further methods can be implemented, which are necessary for the event, like transmitting information between the widgets. The *handler* interface is implemented in the receiving class, which contains one method that is executed after the event was fired. There are five implemented events in this package:
  - CalendarEvent: used to transmit a date between the calendar and the daily view
  - DiaryEvent: used to view the selected diary entry after selecting a date
  - OutcomteViewEvent: used to show selected parts of the outcome view
  - RefreshEvent: used to refresh pages after data has changed
  - ViewEvent: used to highlight goals or problems after selecting one in the overview

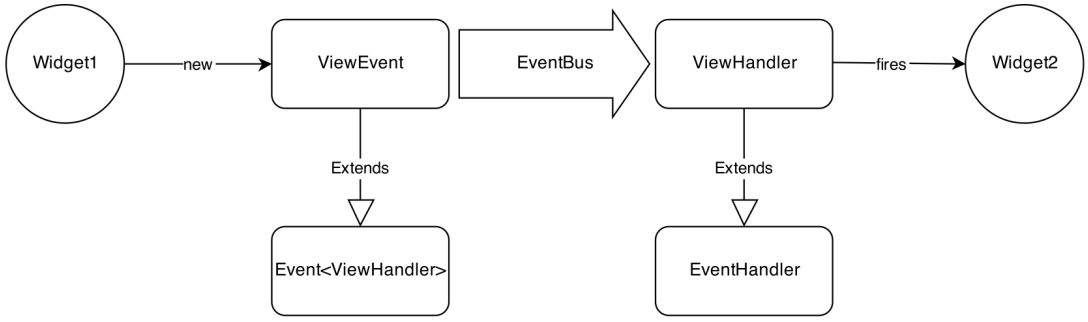


Figure 4.6: Scheme of event handling between two widgets

4.2.2 Server implementation

The server package consists of four Java classes. These four classes are all remote-service classes, which are used from the different client views.

GoalServiceImpl:
The GoalServiceImpl class is the implementation of the interface GoalService. It contains all methods that are linked to the domain classes Goal and ReviewPoint. It calculates the GAS

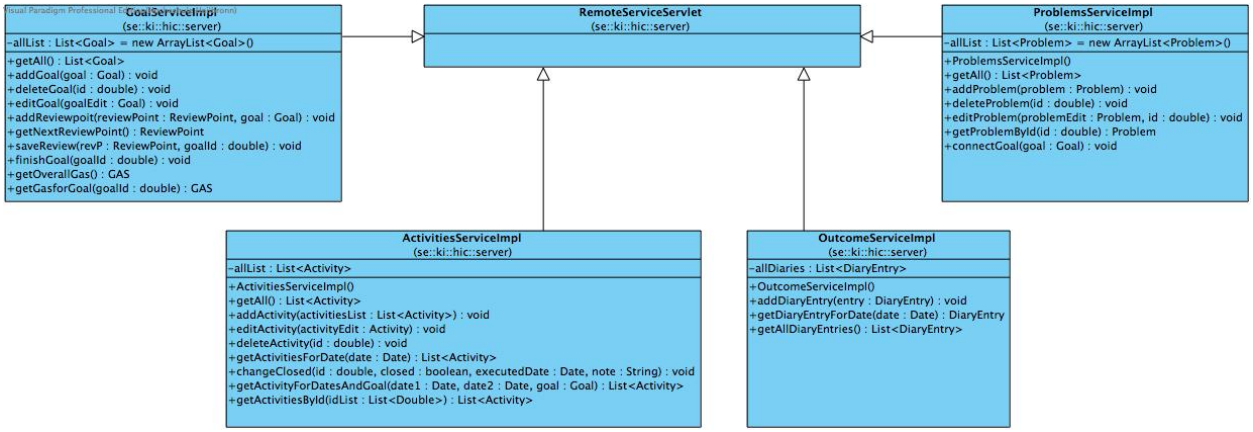


Figure 4.7: Server package

value for every single goal and also the overall GAS, which includes any goal (see example code in Appendix subsection A.2.1). These methods are used as software interfaces to supply the user interface with data. The return values are received by an asynchronous service in the client package. The entire Table A.1 documentation of the software interface is in the Appendix chapter.

ActivitiesServiceImpl:
The ActivitiesServiceImpl class is the implementation of the interface ActivitiesService. It is responsible for the management of all activities that are related to a goal. This includes standard methods like get, add, edit and delete, just as other methods to close a activity and get activities for a particular date or ID as documented in Table A.2

OutcomeServiceImpl:
This class contains mainly those methods that are used to manage the diary entries from the outcome page. Hence, it contains only the standard methods for adding a diary entry, which also includes to changing or deleting an entry, and two get methods - one to get all diary entries and another to get a diary entry for a particular date. See Table A.3.

ProblemsServiceImpl:
The ProblemServiceImpl class contains all methods for managing the domain class Problem. It contains all problems in one list, which can be requested with the getAll() method. Additionally, a problem can be requested by a certain problem ID or connected to an assigned goal. Furthermore, the class also contains also the standard methods to add, delete or edit a problem as described in Table A.4.

4.2.3 Domain implementation

The domain classes are localized in the *shared* package of the GWT package-structure (cf. Figure 4.2). Hence, these classes can be used from the client as well as from the server, and classes are translated into JavaScript code. Every domain class contains an ID that results from random number generator.

The domain classes reflect the four main areas that are used for the goal setting process and derived from the concept of the shared care plan based on CONTsys: problem, goal, activity and outcome. The Outcome class is represented by the *GAS* class, which contains the calculated GAS value from the different goals. Additionally, there are two further classes. The *DiaryEntry* class contains a string of a diary entries and the patient's mood, encoded as an integer value, of the patient. This *ReviewPoint* class is used to store the intermediate status of every goal. Therefore, the class structure was realized like in Figure 4.8.

- Problem:
  This class represents any problem that can occur from a stroke. Therefore, the fields *problemText* and *problemDescription* are used to describe the goal. As the problem can be closed after solving it, the Boolean value *open* can be used to close a problem. To color the problem in the user interface, the string value *colorCode* is used to store the rgb string. To connect the problem with different goals, the problem class holds a list with the connected goals. The relevant getter and setter methods complete the class.

- Goal:
  The *Goal* class represents the two different kinds of goals, simple and SMART, in one class. Therefore, the Boolean value *smartGoal* can be set to distinguish between those two types. The same applies to setting the goal, personal with the Boolean value *personalGoal*. Furthermore, the class contains the basic values that are necessary to define the goal, like the goal-title, a start- and an end-date and a color-code, which is used to color the relevant widget in the user interface. To define a SMART goal additional values that are needed include the integer values baseline, importance, difficulty and the weight. Every goal class also has a field to the linked problem and a list with the relevant review points. As the fields are all private, the corresponding getter and setter methods were added as well as a compareTo method from the interface *comparable* to sort different goals by IDs.

- Activity:
  The domain class *Activity* is the implementation of an activity that is related to any goal. To show the connection between an activity and a goal in the user interface, the class also contains a color code. The actual *Activity* class contains fields for a description, a planned and an executed date, and a specific ID. When the activity is achieved, the Boolean value, open,

can be set to false. Additionally, the class contains two optional string-fields: The URL field is provided to hold a link to a webpage that is used as a tool to achieve the activity while the note field that can be used to write anything regarding the achieved activity.

- ReviewPoint:
  The *ReviewPoint*-class represents a review point that is related to a specific goal. Since there are two kinds of goals, the review point differs as well, but is not separated into two classes. The differences are the fields that are used for GAS. The integer field *gasCode* saves the rated outcome and is used to calculate the actual GAS value that is saved in this class in the field of the same name. Apart from these fields the *ReviewPoint* class contains a date value for the day of the review point, a status string to save the reached status of the review point, a note string and also a Boolean value to see if a review point is open or closed. Like any other domain class, this class has also an ID field. In order to be able to sort the review point by dates, this class contains, in addition to the getter and setter methods, a *compareTo()* method.

- GAS:
  The *GAS* class was modeled to hold a sequence of the calculated GAS values with the relevant label. It is mainly used as a data transfer object (DTO) and holds the objects that are used from the chartJS classes.

- DiaryEntry:
  The *DiaryEntry* class models a diary entry, which consists of the actual entry string, the date of the entry, and a value for the mood that represents the three different moods, good (1), bad (-1), and so-so (0) as an integer value.

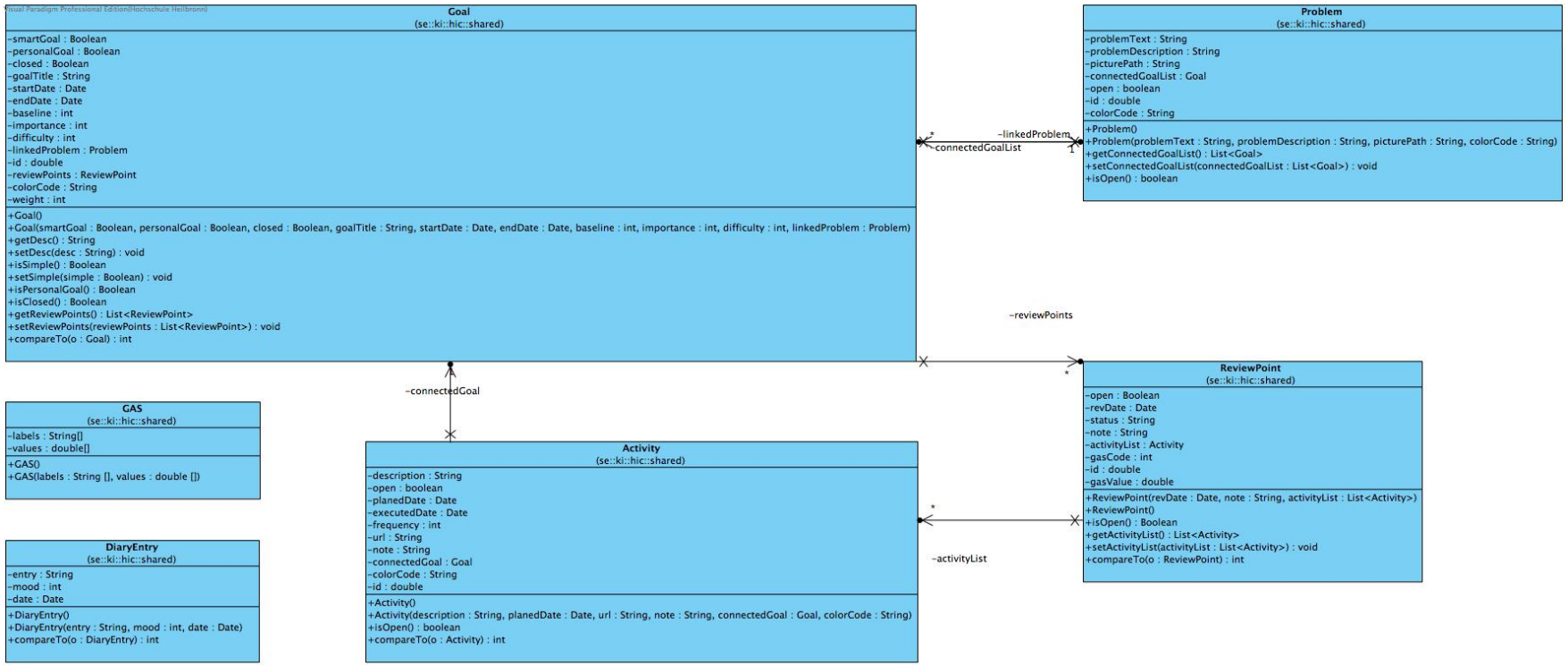


Figure 4.8: Class diagram of the domain classes

4.2.4 Database implementation

Since the web application is just a prototype and, therefor, not supposed to be used in long-term work, a database was not attached to the back-end. To test the prototype, the data were stored temporarily in the server classes.

5 Results

5.1 Interactive prototype

The prototype was realized as a web application that uses JavaScript. It is compatible with standard web browsers, like Chrome, Firefox, Safari and the Internet Explorer Version 9 or later, and can, therefore, be used by most of the users who have a computer at home. The webpage was optimized for a screen resolution not smaller than 1024x 720 pixels and can also be used on screens which an aspect ratio of 4:3.

5.1.1 Structure

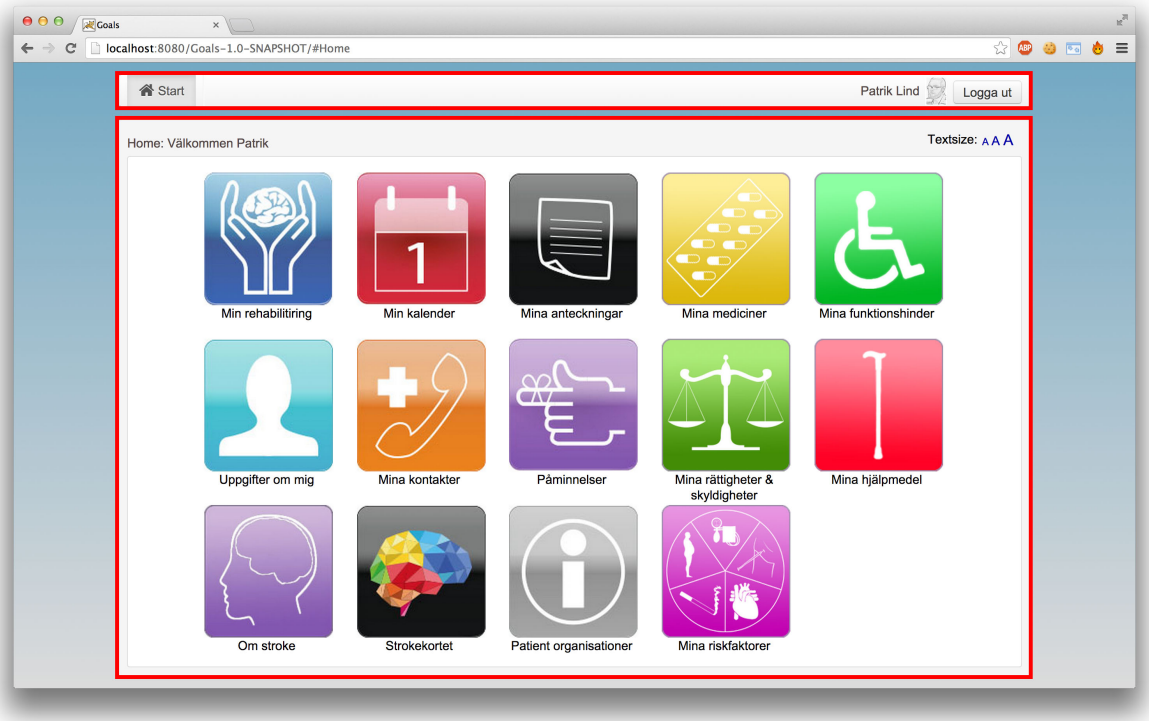


Figure 5.1: Structure of the web-application

The structure of the web application consists of two main areas: the menu bar on the top and the content in the middle of the page (cf. Figure 5.1).

- Menu-bar:
  The menu bar contains different menu items, which changes depending on which menu icon from the home screen is selected. The *start* button is always present, as it has the function of bringing the user back to the home screen, where he can select from the different icons. Also the patient's picture and the logout button are always visible to the right of the menu bar.
- Content:
  The content itself consists of two parts: the upper portion and the actual content. The top part shows the title of the selected page on the left; on the right are three different-sized As, which are used to resize the text globally (cf. Figure 5.2). The resize buttons can be accessed from every page. The second part is the actual content, which changes according to the selected item.
- Widgets:
  On the pages behind the *Min rehabilitering* icon the content pages consist of different widgets. Widgets are components that are self-contained by offering their own functionality. They can be embedded independently in the different pages. One example would be the the text resize buttons, which collectively offer one functionality and are used on different pages.

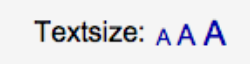


Figure 5.2: Textsize widget

5.1.2 Home screen

The application starts with the home screen. The view contains different icons, which represent different categories, that were partially identified in the master thesis of Yamrotsow Woldemariam [Woldemariam, 2013]. The icons were adopted from the prototype of Sara Lyckstedt master's thesis [Vis, 2012] and extended by five further icons (cf. Figure 5.3).

Overall, the home screen consists of 14 categories.

Besides the *My rehabilitation* page the other pages were realized as *dummy* pages with static content. Some were adopted from previous prototypes or adapted slightly to this prototype. The following chapter provides a short overview of these pages.

Figure 5.3: Homescreen icons - red frame shows newly added icons

5.1.3 Dummy pages

- My rehabilitation (Min rehabilitering):
  My rehabilitation was fully implemented using dynamic pages and is explained in the next chapter.
- My calendar (Min kalender):
  My calendar contains the screenshot of the calendar developed by Sara Lyckstedt [Vis, 2012], which is used in her prototype. It contains all appointments that are not related to the actual rehabilitation plan.
- My notes (Mina anteckningar):
  My notes contains a list of notes that could be created either by writing or voice recording.

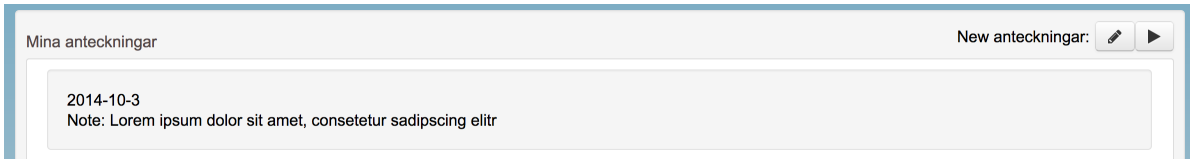


Figure 5.4: Screenshot of the *Mina anteckningar* page

- Information about me (Uppgifter om mig):
  This page provides information about the patient, such as name and address as well as a picture of the patient.
- My contact (Mina kontakter):
  My contact contains the addresses of different important contacts, like family members or the patient's caregivers.
- Reminders (Påminnelser):
  On this page the patient can set simple reminders with a date and a time, as per her/his needs.
- About stroke (Om stroke):
  The *About stroke* page gives the patient general information about stroke.
- Stroke card (Strokekortet):
  This site contains a stroke card, which explains how to recognize symptoms of a stroke. The stroke card is provided by the Swedish National Stroke Association. [1]
- Patient organisations (Patient organisationer):
  This page embeds the overview page of the Swedish National Stroke Association, where a patient can find the nearest patient organization [2]. The page was included as an iframe.

Newly added in this prototype:

- My medicine (Mina mediciner):
  Min medicine contains a list of drugs that the patient has to take. It outlines what each drug looks like as well as when, how much, how often and in which way the patient should take the drugs. The patient can also add new drugs to his list.

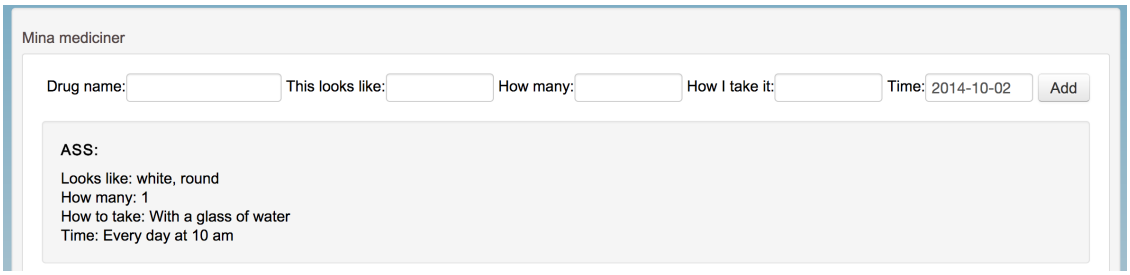


Figure 5.5: Screenshot of the *My medicine*-page

1 Link: `http://www.strokeforbundet.se` (Last accessed 15 October 2014)
2 Link: `http://www.strokeforbundet.se/show.asp?si=480&go=Föreningarna` (Last accessed 15 October 2014)

- My disabilities (Mina funktionshinder):
  This page provides an overview of the patient's disabilities.
- My rights & obligations (Mina rättigheter & skyldigheter):
  On this page the patient gets a brief overview of his rights and obligations after suffering a stroke. To provide full overview the page offers a link to the Swedish care guide site 1177.se [3] [4]
- My tools (Mina hjälpmedel):
  This page embeds a overview page of different tools, that can be used during during the rehabilitation. This, too, is from the Swedish care guide site 1177.se [5] . The page was included using an iframe.
- My risk factors (Mina riskfaktorer):
  My risk factors contains a list with descriptions of the different risk factors that can lead to a stroke. The major risk factors are also presented in the icon smoking, obesity, high blood pressure, diabetes and heart disease.

3 Link: `http://www.1177.se/Stockholm/Tema/Stroke/Hjalp-och-stod/Lagen-om-stod-och-service-till-vissa-funktionshindrade---LSS-` (Last accessed 15 October 2014)

4 Link: `http://www.1177.se/Stockholm/Tema/Stroke/Hjalp-och-stod/Om-man-inte-ar-nojd-med-varden` (Last accessed 15 October 2014)

5 Link: `http://www.1177.se/Stockholm/Tema/Hjalpmedel/` (Last accessed 15 October 2014)

5.1.4 My rehabilitation (Mina rehabilitering)

By clicking on the home screen on the *My rehabilitation* icon the patient can access to his rehabilitation plan. It consists of five main areas, which are shown in the menu bar to the right of the *Start* item: overview, problem, goals, activities and outcome.

Figure 5.6: Screenshot of the menu, when clicking on the rehabilitation icon

- Overview (Översikt):
  At the beginning, the user reaches the overview page. The page illustrates the overall connection between a problem and the connected goals and activities as an overview for the patient. For this representation, every problem and goal is listed. In order to preserve clarity, only the amount of open activities are presented in this view. To clarify the connection between goals and activities, the listed elements of the overview are displayed in the same color. With the attached *view* button, the user can reach the appropriate problem, goal, which get highlighted on the proper page. By clicking on the *view* button in an activity box the page changes also to the *activities* view but shows there all activities that are related to one goal. The *view* button in the *GAS* or the *Mood History* box changes to the *Outcome* view and shows there only the appropriate content by fading out the inappropriate content. If the patient has an open review point, he also gets a red colored hint at the top of this page.

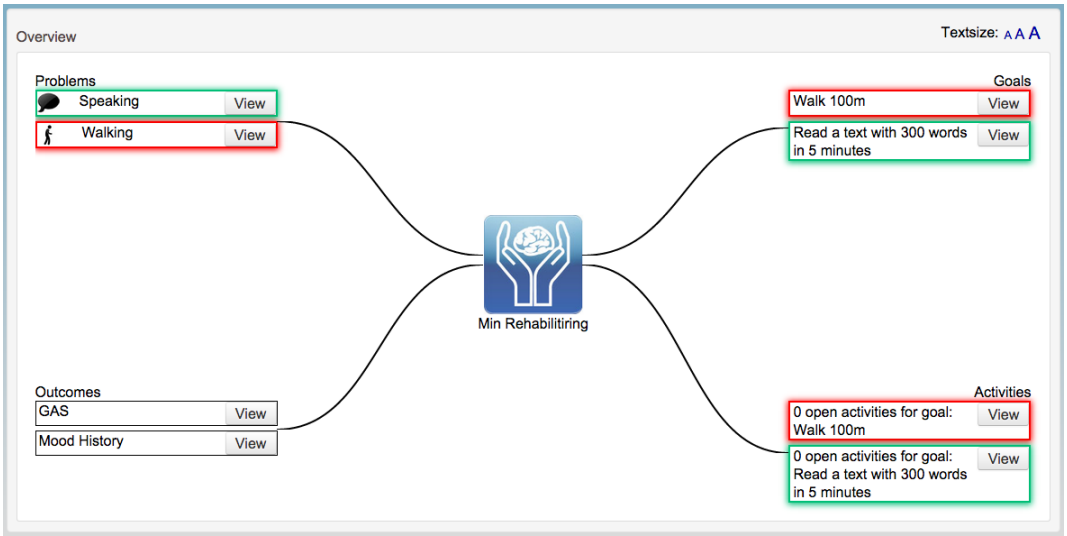


Figure 5.7: Screenshot of the overview page with colored problems and the appropriate goals and activities

- Problems view (Problemet vy):
  The problems view is separated into two parts: "View open problems" and "View closed problems", which are accessible from two *tabs* located on the top of the content page. A new problem can be added by clicking on the the "Add problem" - button that is located on the upper right side of the content page. Clicking this opens a modal window that contains a wizard to set up a new problem (cf. Figure 5.9).

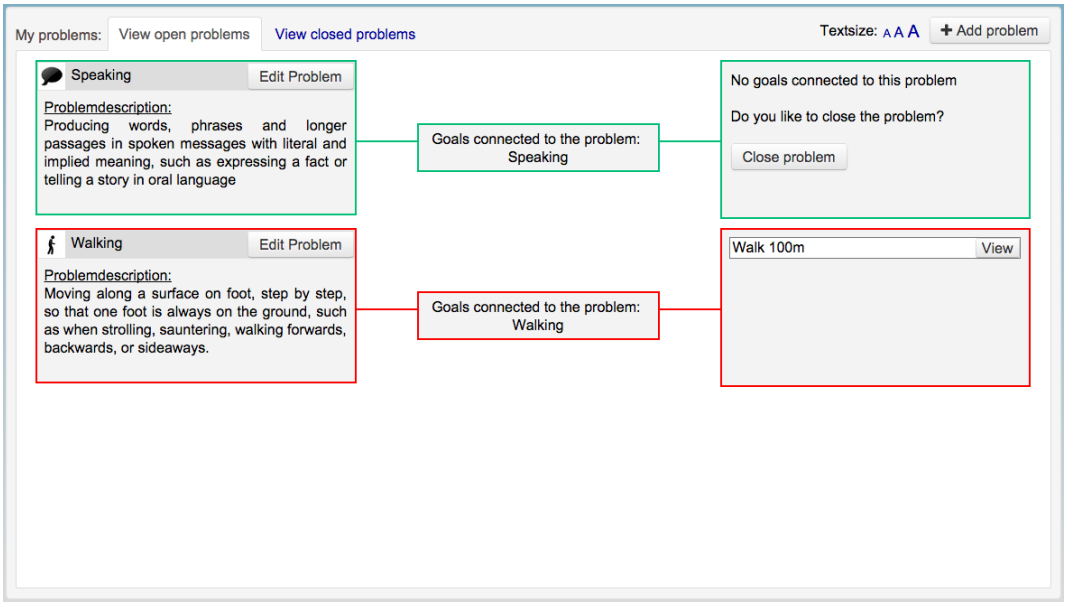


Figure 5.8: Screenshot of the problem view with two differently colored problem widgets that represent two ICF problems

  - Problem wizard:
    The problem wizard (see Figure 5.9) offers two ways to set up a new problem. With the *Predefined ICF problems* drop-down menu, it is possible to fill out all fields with default information from the *Comprehensive ICF core set for stroke*. In the prototype, three different ICF problems were added to this drop-down menu. The color can still be selected at this point. Therefore, the four different therapies offer different color options:

Therapy	Color code
Logopedi (Speech therapy)	green
Arbetsterapeut (Occupational therapy)	yellow
Sjukgymnast (Physiotherapy)	blue
Kurator (Counselor)	red

Table 5.1: Color coding for different therapies

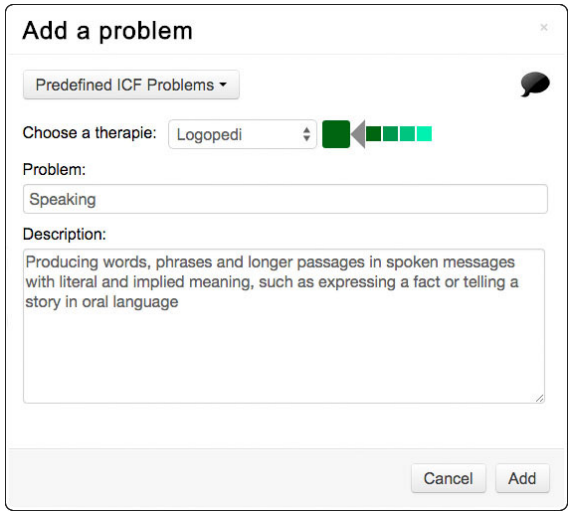


Figure 5.9: Screenshot of the problem wizard to add a new problem

– Problem widget:
After the problem is created using the problem wizard a problem widget appears in the *View open problems* view (see Figure 5.8). The problem widget consists of three main areas. The left side contains an icon representing one of the three implemented ICF problems, and the corresponding heading of the ICF category. Beneath that is the problem description, which was also used from the ICF core set for stroke.
The right side shows all open goals that are connected to the problem. With the proper *View* button one reaches the connected goal, which will be highlighted with the appropriate color. In case all goals are closed or no goals are connected to the problem, the user is able to close the problem by clicking the *Close problem* button which moves the the widget to the "View closed problems" page.
The rectangle between those areas contains a label to clarify the connection between the problem and the different goals.

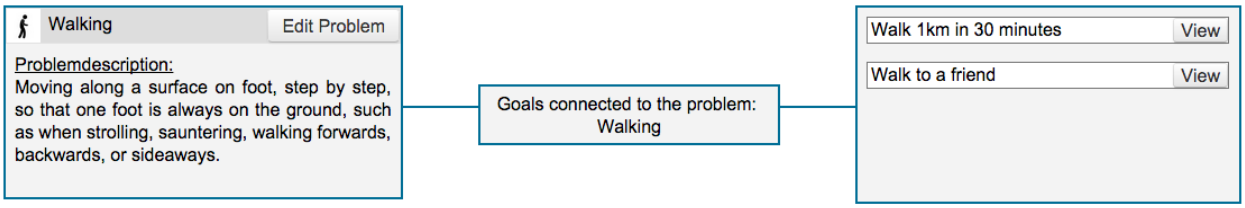


Figure 5.10: Screenshot of the problem widget

- Goals View (Mål vy):
  The goals view has three different sections that can be reached over the proper tabs at the top of the goals content page. Two sections are for the different kinds of goals - the simple goal ("View simple goals) and the SMART goal ("View SMART goals") - and one section for closed goals ("View Archive"), which contains simple as well as SMART goals. At the top right corner, the view contains the *Add goal* button, which starts the wizard to set up a new goal.

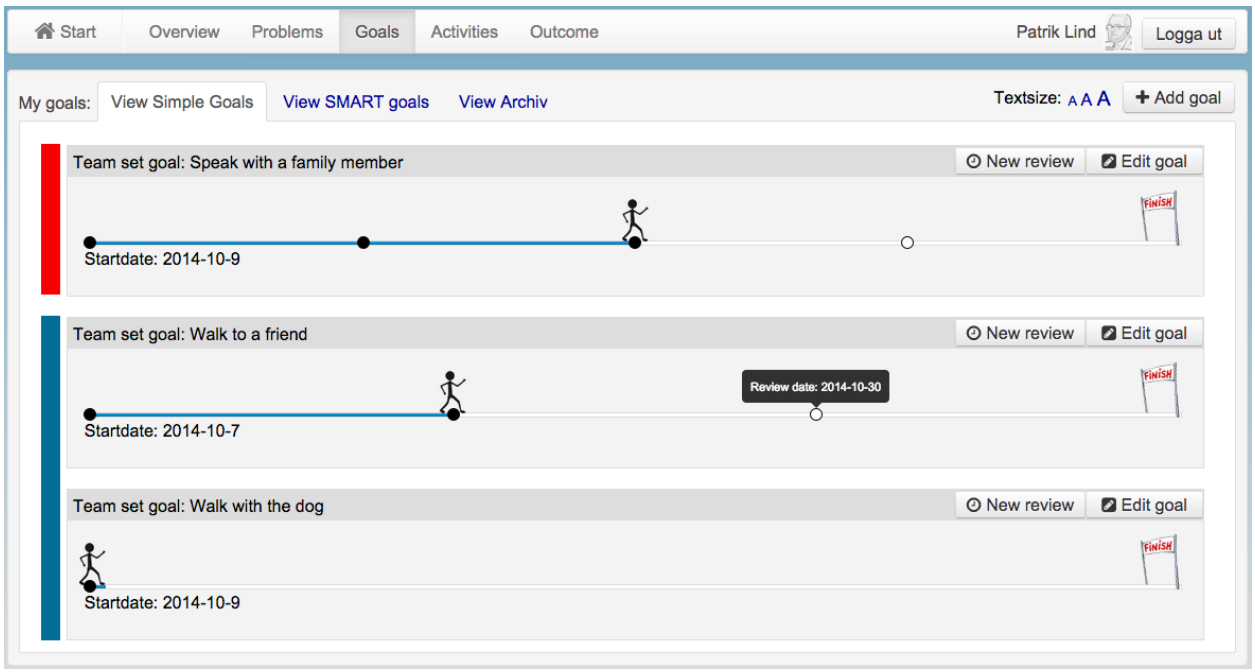


Figure 5.11: Screenshot of the goals view with three simple goals, whereof two belong to the same problem.

  - Goal wizard:
    The goal wizard was implemented as a modal window and overlays the current goals view. It guides the user through the steps to set up either a simple goal or a SMART goal. The following steps can be seen:

    1. The first step of the wizard is the same for both goal types. In this step, the user is asked which kind of goal he likes to set. Additionally, if the goal is not set together with the rehabilitation team, the user can select the *personal goal* check box. After selecting the type of the goal, the topic shows how many steps the wizard contains.

    Steps for a simple goal:

2. In the second step of setting up a simple goal, the user can add a description for his goal and link the goal to the appropriate problem.

Steps for a SMART goal:

2. In the second step of setting up a SMART goal the user has to describe the goal. The goal should match the definition of a SMART goal, as described in section 3.4. A text with the definition assists the user while he writes the description. As in the simple goal the user can link the goal with the appropriate problem.
3. In the third step the user has to set the baseline. For this, he can choose between the two options "Some function" and "None - as bad as can be", which are presented as radio buttons.
4. Step four is used to set the importance and difficulty based on the SMART goal. As from section 3.4, the user has to choose in each case between three options via radio buttons.

Figure 5.12: Steps to set up a simple or SMART goal

– Goal widget:
The goal widgets of a SMART goal and a simple goal differ only minimally. Hence, the overall setting is the same for both goals. On the left side, you see in both cases a color bar, which shows the color from the connected problem. The actual goal is described in the bigger box, which occupies the remaining space. It has two parts: The top part contains the description of the goal as well as the terms to see if the goal is either set with the team or a personal goal, which is shown in front of the description. On the upper right are different buttons. Here, the two versions differ according to the type of goal. Both have a "New review"-button and an "Edit goal" button. The SMART goal box has additionally a "GAS History"-button.

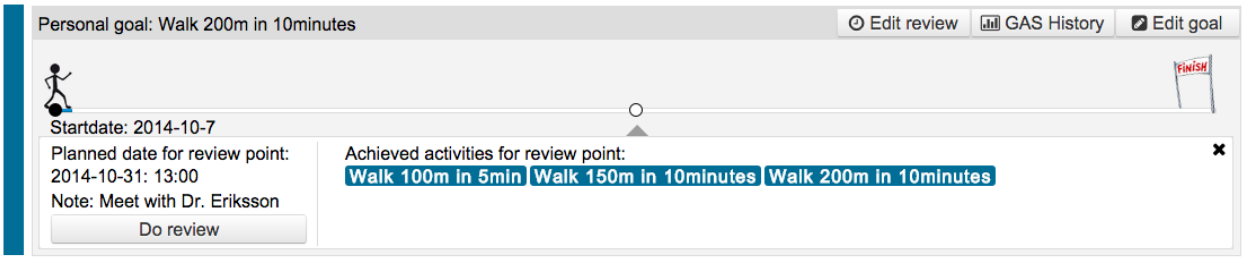


Figure 5.13: Screenshot of the expanded goal widget from the SMART goal "Walk 200m in 10 minutes". Three activities have been achieved for this goal.

The part below contains the progress bar, which represents the progress of the patient's goal. The small figure on top of the bar shows the current status of the patient. When the patient reaches the goal, he moves through the goal line at the end of the progress bar. To reach the goal the patient can set review points with the button "New review". In this modal window, he can set a date and a note for the review point - e.g. a note about which doctor or place to visit. After saving this form a new, white dot appears on the progress bar.

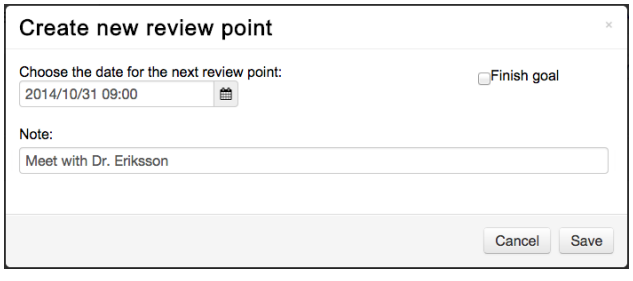


Figure 5.14: Modal window to add a new review point.

It is now possible to move the mouse over the dot to see the review date or to click on the dot. Upon clicking on the dot, the widget expands and shows the details of this review point, as have been entered in the form. The "New review"-button on the top changes then to a "Edit review", allowing the user to change the details of the review point. In the expanded state, the user can also see activities he has already achieved for this goal. When the user has reached the date for the review point, he can do the review by either following the hints from the overview (described above) or by clicking on the "Do review" button in the expanded version of the goal widget. The user then sees a new modal window that asks him about his current status or, provided that it is a SMART goal, he can assess the outcome for this review point by clicking on the button that describes his reached outcome best. Once review is completed the white dot on the progress line changes to black.

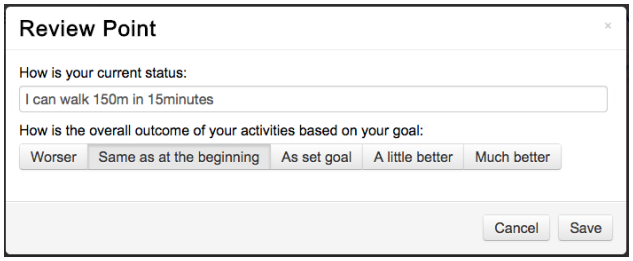


Figure 5.15: Modal window to do the review point.

The same procedure occurs when the patient wants to finish the goal. Here, he has to add a new review point by clicking on the "New review" button. In this modal window, the user can finish the goal by clicking the checkbox "Finish goal". The modal window then allows the user to chose a date and additionally set his current status. Provided that it is a SMART goal, he can also do a review by assessing the outcome of the goal. After assessing the goal, the goal widget disappears from the appropriate view and can be then be found in the archive tab. To distinguish between simple and SMART goals in the archive tab, the respective name is added to the title of the goal widget. The end-date of the goal also appears at the end of the progress bar.

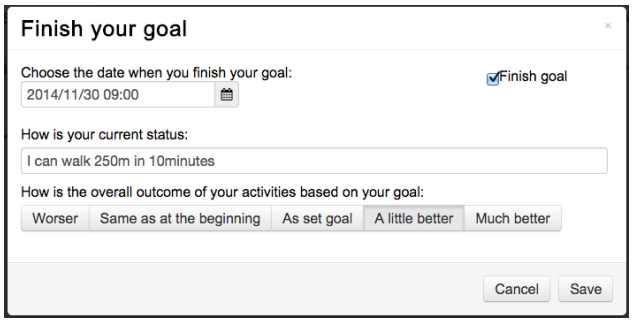


Figure 5.16: Modal window to complete a SMART goal.

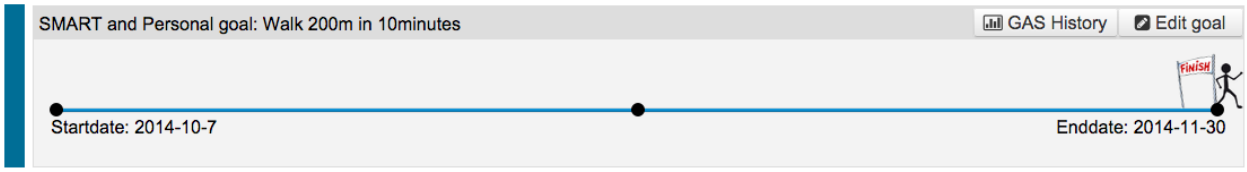


Figure 5.17: Screenshot of the goal widget. It shows a finished goal with one closed review point.

After doing one review, the program is able to calculate the GAS value, which can be seen by clicking on the "GAS history" button on the top of the goal widget. A new modal window opens after the click. This contains a line diagram of the calculated GAS values, starting on the left with the previous determined baseline.

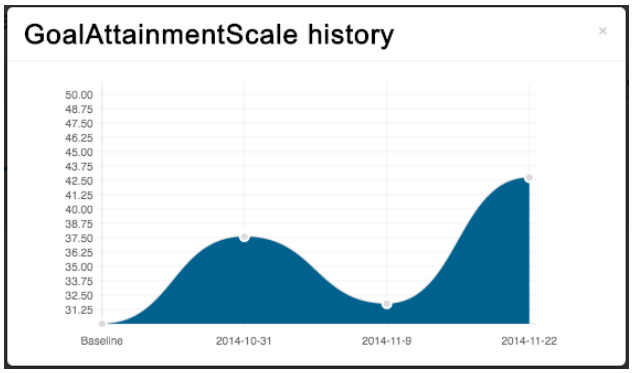


Figure 5.18: Line chart of the calculated GAS

- Activities view (Aktiviteter vy):

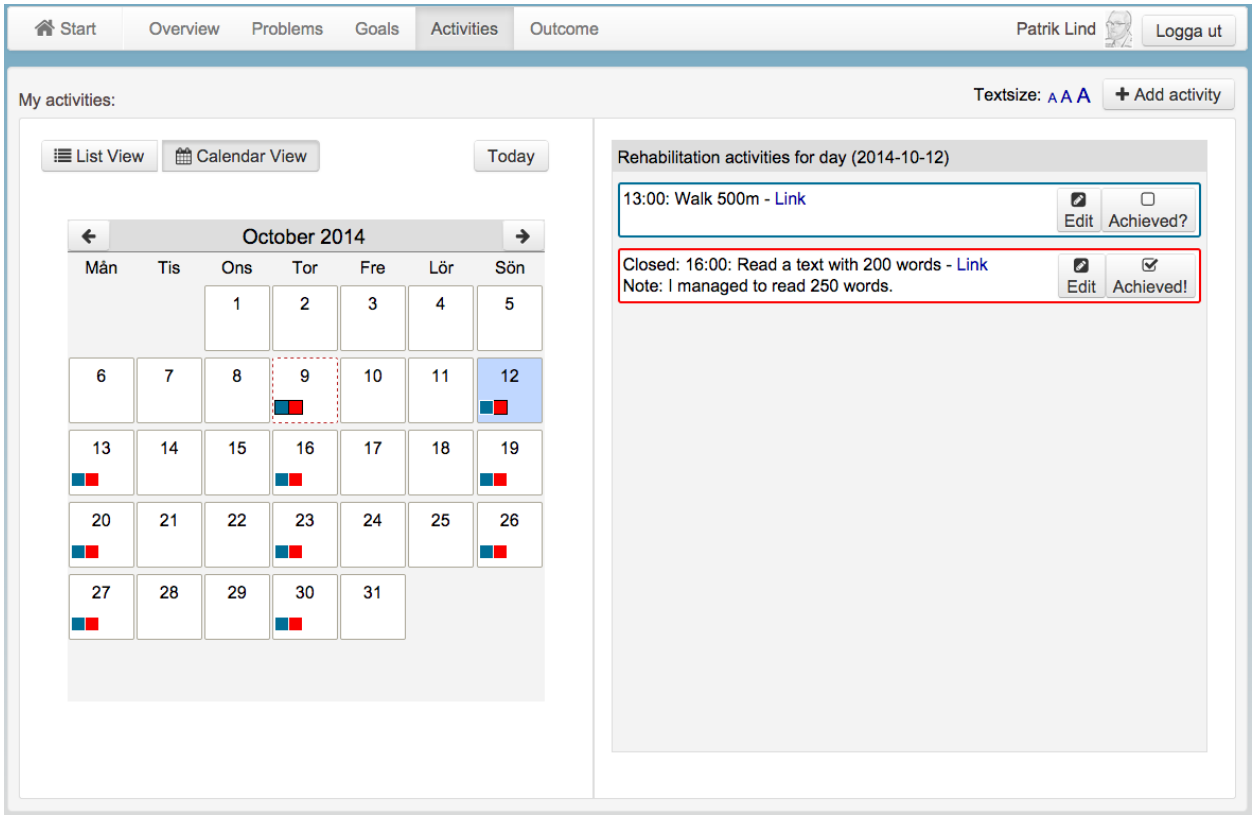


Figure 5.19: Screenshot of the activities view with the classic calendar on the left side

The content portion of the activities view is also split in two parts. The left side contains two kinds of calendars, which show the activities either in a classic calendar or a list view. The particular view can be accessed by the two buttons located at the top of the calendars. There is also a today button on the right side of the buttons, which shows and highlights in both views the current day, although the current day tile is also highlighted with a red-dashed frame. As default, the classic calendar view is shown at the beginning.

- Classic calendar:
  The classic calendar offers a monthly view and has tiles for every day. Every tile can contain little squares, which represent the actual activity and are of the same color as the connected goal or problem. With the arrows on the top of the classic calendar, the month can be switched.

– List calendar:
The *list* view consists of a list of the days to come, starting with the current date. The list was implemented as an infinite list, which loads new days when the scroll bar reaches the end of the list. Every day from the list view can contain different activity elements. These are marked using a smart color bar at the beginning, which is of the same color as the connected goal. Additionally, the activity element contains the planned time for the activity and the activity description. On the right side of the activity element is also a picture of a checkbox, which shows if the activity has already been achieved or not.

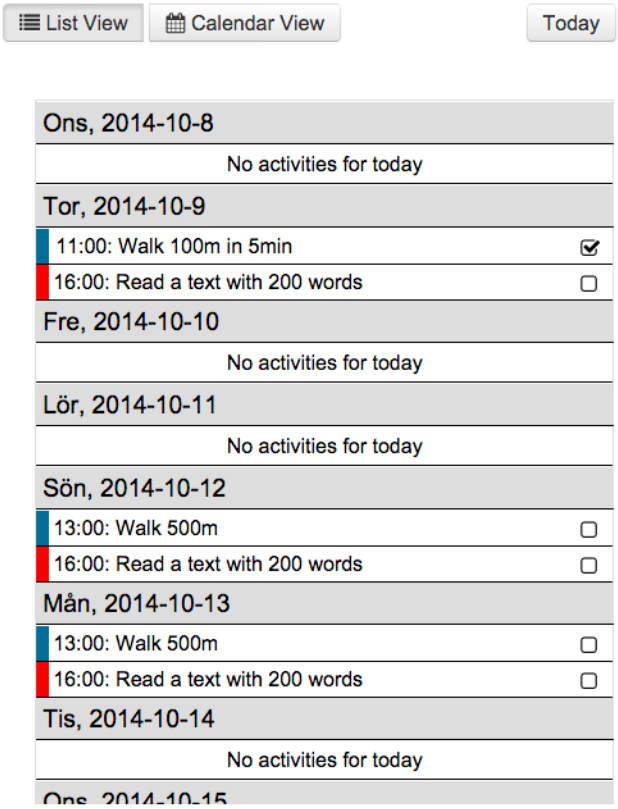


Figure 5.20: Screenshot of the calendar as a list.

Every date or activity element of the classic calendar or the list view can be marked with a mouse click, which is shown by changing the background of the respective tile or activity element from white to a light blue. Furthermore, the right side of the activity view changes and shows every activity of the selected day.

– Activity widget:

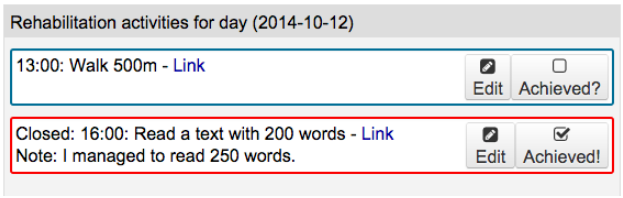


Figure 5.21: Screenshot of the activity widget with one open and one closed activity.

The right side of the activities view consists of the activity widgets, which are embedded in a frame that shows the selected calendar date at the top. The different dates for an activity are presented in a colored frame, which shows the information - like the time and the description - of the activity. If the patient added a link for a tool, a hyperlink appears at the end of the description. With the two buttons on the right side, the user can either edit the activity or change the status to "achieved", after accomplishing the activity. For this purpose, a modal window appears where the user is asked if he has achieved the activity (see Figure 5.22).

Figure 5.22: Modal window to set the activity to achieved.

By changing the field to *yes* an input field appears, wherein the patient can add a note related to that specific performance of the activity. This note appears then in the actual activity widget below the description text. In order to ensure that the activity is achieved, the text "closed" appears in front of the description, the label of the achieved button changes from "achieved?" to "achieved!" and the icon changes from an empty checkbox to a checked checkbox.

– Activity-wizard:
By clicking on the "Add activity"-button on the top right of the activities view, a new modal window opens (see Screenshot Figure 5.23). This activity wizard offers the user the possibility to connect an activity to a goal and add one or more repetitions of one activity to the calendar of the activities view. Therefor, the user has to select a goal for his activity, add a description and provide a date for starting with the activity. Alternatively it is possible to add a Url to the activity, which links, e.g. to a tool that the patient uses for her/his activity. Thereafter the user can decide whether he wants to repeat the activity. By selecting the radio button to yes the window expands, so that the user is able to set the repetitions. To keep things as simple as possible, the user can here select the specific days of the week, when he prefers to repeat the activity and as well as the date when the repetitions should end. The right side of the activity wizard is used to keep track of the amount of repetitions that result from the selected days and the end-date. It summarizes the duration, consisting of the start- and the end-dates, and also the overall amount of repetitions, the repetitions per week and the number of weeks. By clicking on the "Save"-button, the amount of activities is added automatically to the calendar and shown as small squares in the classic calendar view or as list elements in the list view (cf. Figure 5.19).

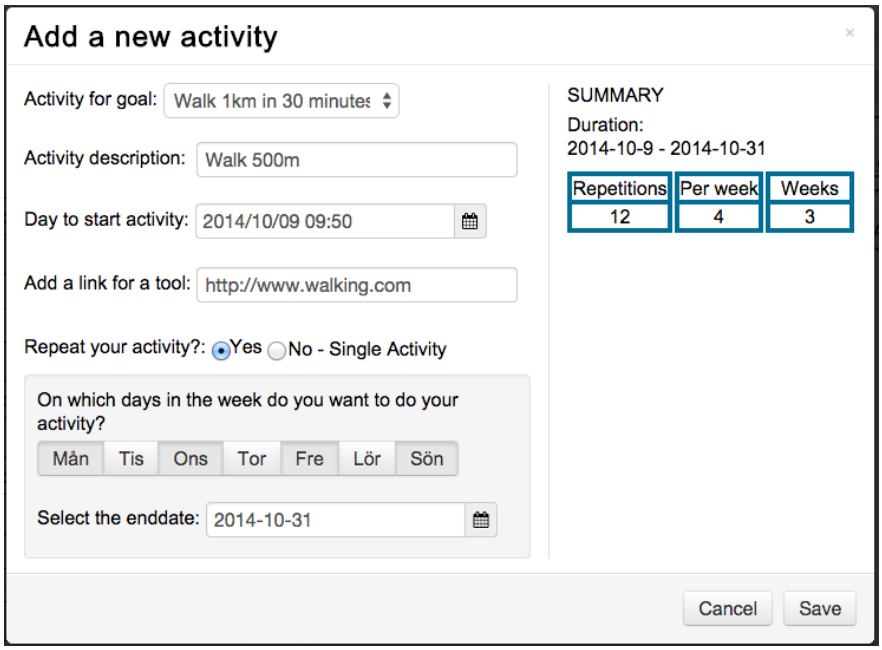


Figure 5.23: Screenshot of the activity wizard.

- Outcome view(Utfal vy):
  The *outcome* view is also divided into two parts. The left side contains the overall calculated GAS, which includes every review point from any goal, that has already been evaluated. It is displayed as a line chart that begins on the left with the oldest review point. As those review points are from different goals, the first assessed review point represents the baseline. hence, the chart could differ from the GAS, which is calculated for a single goal.
  Beneath the graph is a small statistical field, which gives the patient a short overview of past achievements. Therefore, it shows the amount of achieved problems, goals and activities.

Figure 5.24: Screenshot of outcome view.

On the right side, the outcome view offers the patient a diary that allows her/him to add the current mood, as a smiley, to the diary entry. For this purpose, a button group with three different smileys is available, reflecting the three moods, good, bad and so-so. By entering the outcome view, the diary entry of the current day is presented in the diarybox. On the upper right of this box, the user can choose among the different dates to see, edit or create entries from earlier dates. All diary entries are also presented with the selected smiley in the *Mood History* widget, which is located under the diary box. With this widget, the user can see the course of his mood at a glance. By clicking on the smiley, the diary entry also appears in the diary box.

5.2 Prototype evaluation

The prototype was evaluated in two parts. The first question concerned the practical use and terms of the prototype together with health professionals of a focus group while the second focused on the interactive prototype together with stroke patients.

5.2.1 Focus group evaluation

The results of the evaluation of the focus group meetings are structured in 10 parts in the same way that health professionals of the focus groups were questioned about the prototype.

1. Content/Functionality
   The first spontaneous impression after the introduction to the prototype was a bit reserved. It is possibly that, given the wide range of functions and the short presentation, they faced problems in using the full functionality. They had some concerns about the rehabilitation in conjunction with the software.

   > " ... and who shall guide the user?"

   With regard to the content of the information, the opinion was that the patient gets enough information to continue their rehabilitation at home.

   > "Home care services can see who is doing what, and can schedule times ... '

   However, they were not sure, if the reference to risk factors might concern some patients.

   > "Risk factors - can cause concern. But it can be very individual; some may not need to be reminded"

   Additionally, thought was given to adding information about prophylaxis in order to prevent additional strokes.

2. Welcome page
   The welcome page made a good impression on the health professionals.

   > "The first icons are accessible and useful."

3. Rehabilitation plan - overview page
   The use of colors to mark therapists on the overview page received a very positive evaluation, since the therapists already used different colors for different training programs so that patients could distinguish these easily in their schedule. For this purpose, they proposed the color that they already used, which then were directly implemented into the prototype (see Figure 5.9).

The idea of presenting an overview that shows the connection between a problem, goal, activity and outcome was also easy to understand:

> 'The connections are clear."

4. Identify problems
   In the section that is used to identify problems, the simple use of ICF terms was welcomed, as the therapists were already using ICF terms in their journals. However, they used paper-based catalogues, which makes it difficult for them to use it in a rapid way, as they are forced to look up every single term. Hence, the use of an electronic catalogue during the identification of a problem, which would ease the documentation process, was evaluated positively.
   The connection between a problem and a goal was also considered positive.

5. Goal setting
   The therapists had some concerns over whether patients might understand the difference between simple and SMART goals. They use SMART goals in their therapy, although only paper-based in their journal. Seeing past records also received positive comments, as being able to view the timeline prior achievements can be motivating for patients.

   > "It becomes partly clear. One can see that it's a little way to go."

   They were also unsure about wether the patient can see already accomplished goals, as the finished goals disappear from the simple or SMART goal screen and appear in the archive page.

6. Activities and training
   The focus of the activities and training section was on the two different calendars. This was necessary because it might be hard for stroke patients to understand why there are two calendars that look different and manage different things, the first concentrating on rehabilitation activities and the second on other activities. Therefore, they had concerns, since the patients often have a lot of activities scheduled.

   > "They may find it difficult to understand who is responsible for what and who will do what activity when."

7. Outcome
   For the outcome page, it was proposed to add more specific rating scales for the diary - like for headaches, pains or fatigue - rather than just three smileys for different moods.

8. Usability
   For the welcome page the term "Mina funktionshinder" (My disabilities), was criticized as stroke patients often have hidden disabilities like brain fatigue, which cannot really described. Therefore, the term "Mina svårigheter" (My difficulties) was proposed as an alternative term.

For the full interface, it was proposed to add short explanations that make it easier to understand the different functions.

> "Short simple explanations that can be used to involve relatives, home care and so on."

Additionally it was proposed to add two more sections to the welcome page:

- Movements: things that can change over time.
- Health conditions: possibility to wash hands, and so on.

9. Influence on work practices
With regard to the influence of the software in their daily work, time was a big factor, as the therapists have only 30 minutes per session with the patient. In most cases during the rehabilitation at home, the time is used for the patient. Even the documentation is carried out afterwards and only a few things get documented for the patient. Most aspects during the rehabilitation session are communicated verbally.

> "I'm afraid to get stuck in it - in how to use the program itself, so that it does not become an obstacle to the exercise."

However, it is considered to be a tool to plan and improve cooperation between different actors involved in patient care and rehabilitation, when it is used for coordination. It is also used to avoid inquiries between the actors.

> "Maybe avoid having to call a contact, ensuring that they have an appointment later that day ..."

The conclusion indicates that there will not be time to use the tool.

10. Acceptance
Support for the software are the main concerns in terms of acceptance. Especially when there are problems with the software, the patient or therapists should have the possibility to call for support.
The actual level of acceptance may differ based on the patient's computer skills, as well as the different disabilities that might make it impossible for the patient to handle any computer.

5.2.2 Usability test results

The usability test was performed as described in the methods subsection 3.5.2. The results of the usability test are the following identified problems together with the potential improvements and the severity of each problem.

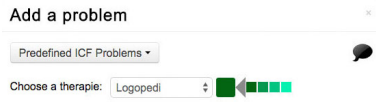
Problem: User has problems with the term ICF.	
Task:	3
Position:	Problem wizard
Problem description:	During Task 3 the user should use an predefined ICF code but has never heard of ICF before.
Potential reason:	No hint or description about the meaning and use of ICF.
Screenshot:	
Severity:	2 - minor usability problem
Potential solution:	Add a description to abbreviations or loan words.

Table 5.2: First usability problem

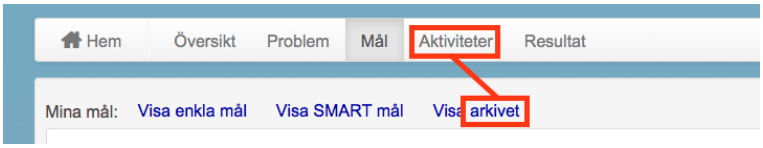
Problem: The words "arkivet" and "aktivitet" are easily interchanged.	
Task:	8
Position:	Main menu bar and menu bar of goal view.
Problem description:	Several users interchanged the words "arkivet" and "aktivitet" by mistake when searching for one of the two words.
Potential reason:	Stroke patient find it hard to differentiate among small differences.
Screenshot:	
Severity:	2 - minor usability problem
Potential solution:	Increase font size or add icons to identify the different functions.

Table 5.3: Second usability problem

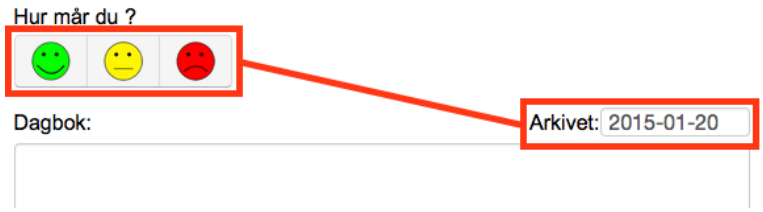
Problem: Program unselects already selected smiley when the date is changed.	
Task:	9
Position:	Activity view
Problem description:	The user tried to add a new smiley for his mood history. So, he selected first smiley for his mood and changed the date afterwards. After changing the date, the smiley was unselected by the program.
Potential reason:	Wrong routine in programming code.
Screenshot:	
Severity:	3 - major usability problem
Potential solution:	Remove deselection from programming code or add a hint about the order in which the mood history should be used.

Table 5.4: Third usability problem

Problem: There is a tendency to drag and drop the mood history smileys even though they are buttons.	
Task:	9
Position:	Activity view
Problem description:	The user tried to drag and drop the smiley in the diary text box even though they are buttons.
Potential reason:	Not clear that smileys are buttons.
Screenshot:	
Severity:	1 - cosmetic problem only
Potential solution:	Add a description of how to use the mood selection.

Table 5.5: Fourth usability problem

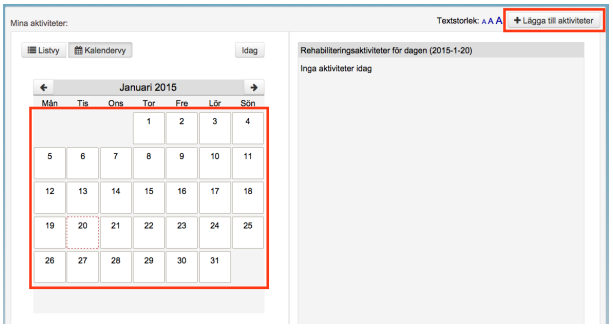
Problem: Adding an activity directly in the calendar.	
Task:	8
Position:	Activity view
Problem description:	The user tried to add a new activity directly in the calendar of the activity view even though there is a button for this in the upper right corner.
Potential reason:	IOther software commonly allow you to add a new date directly in a calendar.
Screenshot:	
Severity:	1 - cosmetic problem only
Potential solution:	Add a description on how to add an activity or include the function to add an activity directly into the calendar.

Table 5.6: Fifth usability problem

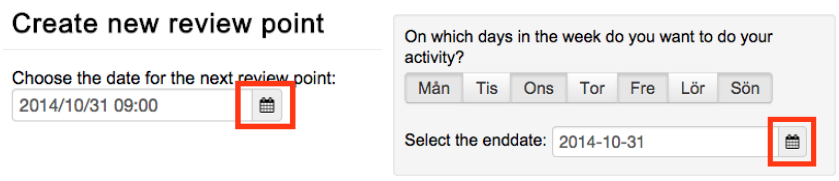
Problem: Clicking on the calendar icon does not show the actual calendar.	
Task:	7 and 8
Position:	Add goal wizard, Add activity wizard
Problem description:	The user did not get access to set a date for the review point or activity by clicking on the calendar icon. Only the input field on the left of the icon gives access to set a date.
Potential reason:	The user expects a calendar when he clicks on a calendar icon.
Screenshot:	
Severity:	3 - major usability problem
Potential solution:	Make calendar icon clickable.

Table 5.7: Sixth usability problem

Problem: Error help messages in wizards are missing, only the text becomes red.	
Position:	Wizards
Problem description:	The user got stuck when he tried to fill out the problem wizard and forgot to check a box, as the text just became red.
Potential reason:	No help messages in the wizards .
Screenshot:	
Severity:	2 - minor usability problem
Potential solution:	Add error help messages.

Table 5.8: Seventh usability problem

Problems, that were mentioned in the questionnaire:

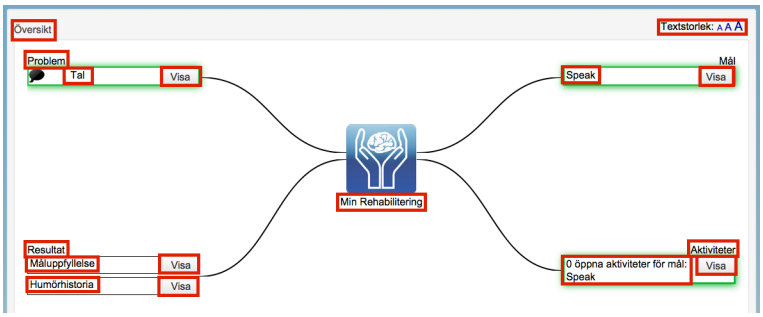
Problem: Headlines in the overview page are too small in relation to the other text.	
Position:	Overview page
Problem description:	It was difficult for users to distinguish between the headlines and the rest of the text on the overview page.
Potential reason:	Text size is the same throughout the whole overview page.
Screenshot:	
Severity:	1 - cosmetic problem only
Potential solution:	Change text size for headlines (bigger) and for the boxes (smaller or the same size).

Table 5.9: Eighth usability problem

Problem: The color red, which is used for the color coding on the overview page, is assumed as an error.	
Position:	Overview page, problem page, goal page
Problem description:	The user assumed an error when he saw the color red on the overview page, although this is actually used to color code a problem/goal/activity.
Potential reason:	The color red is connected to an error.
Screenshot:	
Severity:	1 - cosmetic problem only
Potential solution:	Use of another color or even a lighter or darker red.

Table 5.10: Ninth usability problem

5.2.3 Questionare summary

1. Do you have any suggestions for improving the prototype?
   One user proposed that the user interface should be closer to the user interface of Windows, as this has become a standard interface and most users are familiar with it.

2. Do you understand the difference between a simple and a SMART goal?
   None of the users could explain the difference between a simple and a SMART goal, right away. Most of them realized that there is a difference, but were unable to put it into words.

3. What do you think about GAS?
   The GAS was commonly seen as a helpful tool that supports the patient's progress throughout the rehabilitation. In particular, the graphical representation on the basis of a line chart facilitates its comprehension. The labelling of the diagram with the calculated GAS values did not contribute toward understand the GAS and led to confusion. Some users were confused by the term itself, as they did not know its meaning. The replacement of the abbreviation was proposed.
   To see the progress of the different goals, a chart displaying all goals was suggested an addition to the outcome view.
   Negative effects for the patients could follow from a sub-par GAS chart.

4. What is your opinion on mood history?
   The mood history was also seen as a helpful tool to follow the patient's mood in a simple way. The connection to the diary was also seen as a good way to analyze why one might be in a bad mood on a given day. A more detailed scale spanning five to 10 degrees was proposed.

5. Do you think that you can/want to use the care and rehabilitation plan as a supportive tool for tracking your goals and activities?
   All of the users replied in affirmative to this answer. Some users would also use it without any rehabilitation programme, just for themselves.

6. What do you think about the first page and overview page?
   The first page, due the use of icons for the different functions, appeared modern and useful to the user. The users did not miss any further information or functions.
   Regarding the overview page, the representation of different elements using colors was criticized. The use of the color red confused patients, who viewed it as an error in the program. Also the use of the colors as a frame met with varied reactions It was proposed to use the color for the whole element, including the background. For people who are colorblind or having trouble to distinguish colors, another visualization to see connected goals was proposed - e.g. with a small number in a box.

7. What is your overall opinion about the prototype?
   Overall opinion regarding the prototype was evaluated differed. Opinions ranged from the prototype being uncomplicated to it being complicated. The general agreement was that repeated use would make the prototype simpler. The introduction video also helped the patients to find their way through the prototype.

6 Discussion

6.1 Discussion of the results

With the prototype, stroke patients are now able to set their own goals with the help of a web application either in collaboration with a health professional or by themselves. How patients can use the prototype was one of the research questions for this thesis. However, even during the design process and the following implementation, it became clear that the result might lead to a complex user interface. The use of the OLD@HOME structure - consisting of the components problem, goals, activities and outcome as well as the appropriate connection between those components - makes it essential for the user has to create and link those components in the user interface. In spite of this assumption, the results of the usability tests indicated that the prototype is feasible for stroke patients. Most of the patients announced that the prototype seems complicated but admitted that the software will become easier after they have used it a few times. The use of the same color for connected components made it simpler for patients to understand the relation between the different OLD@HOME components. The input of the components via the user interface did not become a major difficulty. Most of the operating elements were located properly and the input of the component with the help of the different wizards were achieved after a certain amount of time. Small fonts or similar button captions ("arkivet" and "aktivitet") resulted in problems. The use of different icons and font sizes for control elements could avoid confusion [Wiedenbeck, 1999].
With the use of a visualization for the goal progress via the GAS through a line chart, the patient is now able to see his achievements directly, and not just through the reports of his therapists, which could result in further motivation for the patient. However, the evaluation showed that all patients encountered problems in distinguishing between a simple and a SMART goal at the first go. To avoid this, a pure presentation that shows just the chart, with hints at the progress, might be a better solution.

Health professionals now have the opportunity to gain insights into the different activities of their patient, not just through the rehabilitation that they provide but also from other health professionals and additional private activities, which they have added to the system. This information could help them to improve their collaboration by using the software for coordination, although

further research would be necessary to see the impact on clinical practice. By adding predefined ICF problems, the documentation of the rehabilitation could be eased, as most health professionals already use ICF codes, but only occasionally. An additional module in the application that provides a summary for the health professionals would also help in this regard.

The focus group evaluation indicated that the time factor might be an obstacle in the use of the application during rehabilitation. As the therapists have only 30 minutes per session with the patient, they will not be able to use the software together with the patient or support the patient when the latter has problems. The training period for the therapists to become familiar with the software would also be a time factor. Thus, it is up to individual patients to decide whether or not to use the software.

In summary, the prototype can be a helpful tool for patients or health professionals. However, it still needs some improvements - e.g. in the form of more explanations to help patients while they are using the application or when they got stuck. One improvement could also be to omit features in order ti simplify it further.

6.2 Discussion of the methods

The user-centered design to develop a software artifact, was used for this thesis. The main focus of this thesis was the construction phase, which includes the implementation and the usability testing of the application. However, it was necessary to supplement and adjust previous prototypes as several parts of the interface did not exist. The actual implementation was done by using object-orientated methods. Part of the design process is the evaluation of the interface by user, which are represented in this thesis by two stroke patients and one family member. The quantity of stroke patients conforms to the minimal number of participants needed for a usability test. Nielson [Nielsen and Landauer, 1993] concluded in his mathematical model that "three users are enough to get an idea of the diversity in user behavior and insight into what's unique and what can be generalized", which corresponds to 65% of the usability problems. However, he states the number five as the optimum and maximum number of testers to achieve 85% of the usability problems. Unfortunately, two additional persons could not be recruited due to lack of time.

Although the development of the prototype was targeted to patient needs, lots of information was given in the focus group meetings to facilitate understanding of organizational process of the rehabilitation. Three such meetings were held. These involved two to three different health professionals in each case; all of the therapists, who are involved in the rehabilitation process were

asked to provide an insight into their daily work and to derive the information for use it in the prototype.

7 Conclusion and future work

This thesis used an interactive prototype to show how rehabilitation for stroke patients can be improved by a web application. The prototype supports stroke patients with milder disabilities, who set their own goals and can follow their own progress during their rehabilitation at home. It also supports health professionals to see and follow the activities of stroke patients, activities that where created with different therapists and to improve collaboration between them. Focus group meetings were used to examine the requirements from the perspective of health professionals perspective, who treat stroke patients in their everyday clinical practice. To gain an understanding of the stroke patients' perspective usability, tests were performed to examine the operability of the prototype and to find problems that occurred during the handling.

The result is a prototype that was welcomed by all of the patients and that could help them in their rehabilitation process. However, the prototype still needs slight improvements like explanatory notes or hints, to prevent patients from getting lost in the application. One task for the future will also be the way in which the application can be used, especially how it can be used in stroke rehabilitation without affecting the actual care process. This is necessary because stroke teams have only a limited time with for the rehabilitation session and cannot use it to give extensive explanations or support to the patient. Family members can play a major role in helping the patient with the application.

Porting the software so that it can be used with a tablet or a mobile could also help simplify the access to the application as patients use smartphones and would be able to get reminders directly on their screen. Furthermore, including the data from already existing training applications, which are already installed on many smartphones could facilitate acceptance and use the software.

A Appendix

A.1 Software-interface documentation

A.1.1 Goalservice interface

Method	Parameters	Return Value	Description
getAll	-	List<Goal>	Returns all goals in a list
addGoal	Goal	void	Adds a new goal to the list
deleteGoal	Id	void	Deletes the goal with the associated ID
editGoal	Goal	void	Edits a goal
addReviewpoint	Reviewpoint, Goal	void	Adds a new reviewpoint to the assigned goal
getNextReviewPoint	-	Reviewpoint	Returns the reviewpoint, which is next
saveReview	Reviewpoint, GoalId	void	Saves the reviewpoint
finishGoal	GoalId	void	Closes the goal with the associated ID
getOverallGas	-	GAS	Calculates the overall GAS values for all goals
getGasforGoal	GoalId	GAS	Calculates the GAS for the assigned goalID

Table A.1: Methods in GoalServiceImpl()

A.1.2 Activityservice interface

Method	Parameters	Return Value	Description
getAll	-	List<Activity>	Returns all activities in a list
addActivity	List<Activity>	void	Adds new activities to the list
editActivity	Activity	void	Edits activity with the ID
deleteActivity	ActivityId	void	Deletes the activity with the assigned ID
getActivitiesForDate	Date	List<Activity>	Returns a list of activities for the specific date
changeClosed	ActivityId, ExecutedDate, Note	void	Changes the Status(closed/open) for a activity
getActivityForDatesAndGoal	Date1, Date2, Goal	List<Activity>	Returns all goals between Date1 and Date2, which are associated to the assigned goal
getActivitiesById	List<GoalId>	List<Activity>	Returns a list that are linked to the assigned goal IDs

Table A.2: Methods in ActivityServiceImpl()

A.1.3 Outcome-service interface

Method	Parameters	Return Value	Description
addDiaryEntry	DiaryEntry	void	Adds new diary entry to the list
getDiaryEntryForDate	Date	DiaryEntry	Returns the diary entry for a date
getAllDiaryEntries	-	List<DiaryEntry>	Returns all diary entries

Table A.3: Methods in OutcomeServiceImpl()

A.1.4 Problem-service interface

Method	Parameters	Return Value	Description
getAll	-	List<Problem>	Returns all problems in a list
addProblem	Problem	void	Adds new problem to the list
editProblem	Problem, double ID	void	Edits a problem with the ID
deleteProblem	double ID	void	Deletes the problem with the assigned ID
getProblemById	double ID	Problem	Returns a problem with the proper ID
connectGoal	Goal	void	Connects a problem to the proper goal

Table A.4: Methods in ProblemServiceImpl()

A.2 Source-code example

A.2.1 GAS calculation source-code

```
public GAS getGasforGoal(double goalId) {
    String[] dates;
    double[] values;
    List<ReviewPoint> tempRevplist = new ArrayList<ReviewPoint>();
    int sumWeightOutcome = 0;
    double sumWeightSquare = 0;
    double sumWeight = 0;
    double gas = 0;
    for (Goal goal : allList) {
      if (!goal.isSimple()) {
        for (ReviewPoint revP : goal.getReviewPoints()) {
          //summation of the weight outcome
          sumWeightOutcome = sumWeightOutcome + (revP.getGasCode() * goal.getWeight());
          //summation of the square of the weight
          sumWeightSquare = sumWeightSquare + Math.pow(goal.getWeight(), 2);
          //summation of the weight
          sumWeight = sumWeight + goal.getWeight();
          //actual gas calculation
          gas = (10 * sumWeight_Outcome) / Math.sqrt((0.7 * sumWeightSquare) + (0.3 * Math.pow(sumWeight, 2))) + 50;
          revP.setGasValue(gas);
          tempRevplist.add(revP);
        }
      }
    }

    //conversion to string values for chart representation
    dates = new String[tempRevplist.size()];
    values = new double[tempRevplist.size()];
    for (int i = 0; i < tempRevplist.size(); i++) {
      if (i == 0 || tempRevplist.get(i).getRevDate() == null) {
        dates[i] = "Baseline";
      } else {
        dates[i] = (tempRevplist.get(i).getRevDate().getYear() + 1900)
            + "-"
            + (tempRevplist.get(i).getRevDate().getMonth() + 1)
            + "-" + tempRevplist.get(i).getRevDate().getDate();
      }
      values[i] = tempRevplist.get(i).getGasValue();
    }
    return new GAS(dates, values);

  }
```

A.3 Usability test documentation

Tasks:

1. Find out which primary care centre you are listed at.
2. Find out if you have any problem in your rehabilitation plan.
3. Add a new problem regarding your walking. Use the pre-defined ICF codes. Link it to occupational therapy.
4. Add a new problem for you stress without using ICF codes. Link it to counseling.
5. Add a simple personal goal. Add a description and link it to your stress problem.
6. Add a new SMART goal and link it to your walking problem. Choose 'as bad as it can be' in step 3 of 4. And then choose 'difficult' and 'very important' in step 4 of 4.
7. Decide a new review point for your SMART goal for the 31 October. Choose 13:00.
8. Add a new activity for you SMART goal. The activity should start on 27 of November 15:00 and be repeated every Thursday until the 18 of December.
9. Add your mood history for dates 30 October 2014 and 15 November 2014.
10. Do the review for your SMART goal. Choose 'Mycket bättre'.

Follow up questions after the test:

1. Do you have any suggestions for improving the prototype ?
2. Rationales for performance (what the participant says about the prototype)
3. Do you understand the difference between simple and smart goal ?
4. What do you think about Goal attainment scale ?
5. What is your opinion on mood history ?
6. Do you think that you can/want to use the care and rehabilitation plan as a supportive tool for tracking your goals and activities?
7. What do you think about the home page and overall page ?
8. What is your overall opinion about the prototype ?

List of References

[Adams et al., 2007] Adams, H. P., Del Zoppo, G., Alberts, M. J., Bhatt, D. L., Brass, L., Furlan, A., Grubb, R. L., Higashida, R. T., Jauch, E. C., Kidwell, C., et al. (2007). Guidelines for the early management of adults with ischemic stroke - a guideline. *Circulation*, 115(20):e478–e534.

[Berg et al., 2004] Berg, B. L., Lune, H., and Lune, H. (2004). *Qualitative research methods for the social sciences*, volume 5. Pearson Boston, MA.

[Bongers et al., 2006] Bongers, T. N., de Maat, M. P. M., van Goor, M.-L. P. J., Bhagwanbali, V., van Vliet, H. H. D. M., Gómez García, E. B., Dippel, D. W. J., and Leebeek, F. W. G. (2006). High von willebrand factor levels increase the risk of first ischemic stroke: influence of adamts13, inflammation, and genetic variability. *Stroke*, 37(11):2672–7.

[Burmester et al., 2010] Burmester, M., Jäger, K., Mast, M., Peissner, M., and Sproll, S. (2010). Design verstehen–formative evaluation der user experience. *Usability Professionals*, pages 206–214.

[Duncan et al., 2005] Duncan, P. W., Zorowitz, R., Bates, B., Choi, J. Y., Glasberg, J. J., Graham, G. D., Katz, R. C., Lamberty, K., and Reker, D. (2005). Management of adult stroke rehabilitation care a clinical practice guideline. *Stroke*, 36(9):e100–e143.

[Feigin et al., 2014] Feigin, V. L., Forouzanfar, M. H., Krishnamurthi, R., Mensah, G. A., Connor, M., Bennett, D. A., Moran, A. E., Sacco, R. L., Anderson, L., Truelsen, T., O'Donnell, M., Venketasubramanian, N., Barker-Collo, S., Lawes, C. M. M., Wang, W., Shinohara, Y., Witt, E., Ezzati, M., Naghavi, M., and Murray, C. (2014). Global and regional burden of stroke during 1990-2010: findings from the global burden of disease study 2010. *Lancet*, 383(9913):245–54.

[Geyh et al., 2004] Geyh, S., Cieza, A., Schouten, J., Dickson, H., Frommelt, P., Omar, Z., Kostanjsek, N., Ring, H., and Stucki, G. (2004). Icf core sets for stroke. *J Rehabil Med*, pages 135–41.

[Hägglund et al., 2011] Hägglund, M., Chen, R., and Koch, S. (2011). Modeling shared care plans using contsys and openehr to support shared homecare of the elderly. *J Am Med Inform Assoc*, 18(1):66–9.

[Johannesson and Perjons, 2012] Johannesson, P. and Perjons, E. (2012). A design science primer. *Unpublished Manuscript, February*, 25.

[Jokela et al., 2003] Jokela, T., Iivari, N., Matero, J., and Karukka, M. (2003). The standard of user-centered design and the standard definition of usability: analyzing iso 13407 against iso 9241-11. In *Proceedings of the Latin American conference on Human-computer interaction*, pages 53–60. ACM.

[Jupp, 2006] Jupp, V. (2006). *The Sage dictionary of social research methods*. Sage.

[Kinlay, 2011] Kinlay, S. (2011). Changes in stroke epidemiology, prevention, and treatment. *Circulation*, 124(19):e494–e496.

[Kiresuk and Sherman, 1968] Kiresuk, T. J. and Sherman, R. E. (1968). Goal attainment scaling: A general method for evaluating comprehensive community mental health programs. *Community Ment Health J*, 4(6):443–53.

[National Stroke Association, 2014] National Stroke Association (2014). Stroke 101: Fast facts on stroke (online; accessed 2 october 2014). `http://www.stroke.org/site/DocServer/STROKE_101_Fact_Sheet.pdf`.

[Nielsen, 1994] Nielsen, J. (1994). *Usability engineering*. Elsevier.

[Nielsen, 1995] Nielsen, J. (1995). Severity ratings for usability problems. *Papers and Essays*, page 54.

[Nielsen and Landauer, 1993] Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 206–213, New York, NY, USA. ACM.

[Norrving, 2007] Norrving, B. (2007). The 2006 helsingborg consensus conference on european stroke strategies. *International Journal of Stroke*, 2(2):139–143.

[Perlich, 2012] Perlich, A. (2012). Designing an e-service for stroke patients — how can visualization support the management of the individual care process? Master's thesis, Karolinska Institutet Stockholm.

[Sacco et al., 1997] Sacco, R. L., Benjamin, E. J., Broderick, J. P., Dyken, M., Easton, J. D., Feinberg, W. M., Goldstein, L. B., Gorelick, P. B., Howard, G., Kittner, S. J., et al. (1997). Risk factors. *Stroke*, 28(7):1507–1517.

[Schut and Stam, 1994] Schut, H. A. and Stam, H. J. (1994). Goals in rehabilitation teamwork. *Disabil Rehabil*, 16(4):223–6.

[Scobbie et al., 2011] Scobbie, L., Dixon, D., and Wyke, S. (2011). Goal setting and action planning in the rehabilitation setting: development of a theoretically informed practice framework. *Clinical Rehabilitation*, 25(5):468–482.

[Sugavanam et al., 2013] Sugavanam, T., Mead, G., Bulley, C., Donaghy, M., and van Wijck, F. (2013). The effects and experiences of goal setting in stroke rehabilitation - a systematic review. *Disabil Rehabil*, 35(3):177–90.

[Swain et al., 2008] Swain, S., Turner, C., Tyrrell, P., Rudd, A., et al. (2008). Diagnosis and initial management of acute stroke and transient ischaemic attack: summary of nice guidance. *BMJ*, 337.

[Turner-Stokes, 2009] Turner-Stokes, L. (2009). Goal attainment scaling (gas) in rehabilitation: a practical guide. *Clin Rehabil*, 23(4):362–70.

[Vis, 2012] Vis, S. L. (2012). Användarcentrerad utveckling - design av rehabiliteringshjälpmedel för strokepatienter. Master's thesis, KTH Industriell teknik och management, Stockholm.

[Wade, 1992] Wade, D. T. (1992). Stroke: rehabilitation and long-term care. *Lancet*, 339(8796):791–3.

[Wiedenbeck, 1999] Wiedenbeck, S. (1999). The use of icons and labels in an end user application program: an empirical study of learning and retention. *Behaviour & Information Technology*, 18(2):68–82.

[Wieringa, 2009] Wieringa, R. J. (2009). Design science as nested problem solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia*, pages 1–12, New York. ACM.

[Wiles et al., 1998] Wiles, R., Pain, H., Buckland, S., and McLellan, L. (1998). Providing appropriate information to patients and carers following a stroke. *Journal of Advanced Nursing*, 28(4):794–801.

[Woldemariam, 2013] Woldemariam, Y. (2013). Exploring of stroke survivors' information needs for an information and communication technology based home stroke rehabilitation plan to facilitate self-care. Master's thesis, Karolinska Institutet Stockholm.

[World Health Organization, 2013] World Health Organization (2013). Stroke, cerebrovascular accident (online; accessed 15 october 2014). `http://www.who.int/topics/cerebrovascular_accident/en/`.

[World Health Organization, 2014] World Health Organization (2014). The top 10 causes of death (online; accessed 15 october 2014). `http://www.who.int/mediacentre/factsheets/fs310/en/`.

Affirmation

I hereby declare that the master's thesis I have submitted is my own work. Where the work of others has been quoted or reproduced, the source is always given. This thesis or parts have not yet been presented to an university as part of an examination or degree.

Heilbronn, 17.02.2015 Sebastian Stocker