Motivational Strategies to Improve Self Management Rehabilitation

Silas Wittrock

Bachelor Student Creative Technology

Client	ZiekenhuisGroep Twente
Supervisors	Prof. Dr. M.M.R Vollenbroek-Hutten
	PHd. Dieuwke van Dartel
Critical Observer	Dr. A.M. Schaafstal

Abstract

A hip fracture injury is one of the most common injuries in The Netherlands among elderly people. As the Dutch population is aging, the amount of hip fracture injuries will increase. After surgery, the patients need rehabilitation. In the current situation, patients receive 30 minute sessions of physiotherapy a day. There is not enough labor capacity to increase the amount of therapy sessions, thus the focus has shifted towards self-management rehabilitation. This paper researches how to design a system that is able to bring older adults into action by themselves. The prototype is able to track the activity level of a patient by using a sensor mat in the chair. The more active the patient is, the more a flower starts to bloom. If the patient is inactive for too long, the flower will shrink until exercises are executed. The patient can receive assistance by the system via speech explanation and the patient can answer to the system via speech as well. The tests showed that participants are motivated because of the speech interaction with the system. Besides that, the participants understood the exercises and when they had to do the exercises. The system shows potential and further tests with the potential end users should point out if they accept such a system.

Acknowledgements

I am thankful for all the support and guidance that everyone gave me during the project. First of all, I would like to thank my supervisors, Miriam Vollenbroek-Hutten and Dieuwke van Dartel. They were very supportive and provided interesting insights and feedback during the different phases of the project. They were able to arrange expert review sessions which contributed the final project as it is. In addition, I would like to thank my critical observer, Alma Schaafstal, who supported me with different views on the problem. I also acknowledge ZorgAccent and their physiotherapists for giving a tour around the nursing home. Finally, I would like to thank my fellow graduate students who were very supportive and had interesting critiques during the ideation and the testing.

Table of Contents

1 Introduction

Hip fracture injury is one of the most common injuries in The Netherlands. According to the National Institution for Public Health and the Environment (n.d.), yearly 18.000 people are treated for a hip fracture injury. This number keeps growing due to the aging of the population. Three quarters of the hip fracture injuries concern patients over the age of 75. Because of the patients older age, they experience functional deterioration and prolonged hospital stays. Besides that, the patients need rehabilitation in a nursing home.

In the current situation, patients receive daily physiotherapy sessions for half an hour. In some nursing homes the patients can attend an additional group session. Traditional rehabilitation shows that the more you train, the more you gain. Unfortunately, there is not enough labor capacity to increase the amount of physiotherapy sessions. Thus the focus is shifting towards increasing self management rehabilitation by use of technology. However, the patients lack the knowledge and the skills to execute the rehabilitation exercises on their own as stated in Wildenbos G.A. et al. (2019). Without instructions from a nurse or physiotherapist, the patients are unable to maintain their rehabilitation outside therapy sessions.

Therefore, a system should be developed that is able to bring older adults into action by themselves. Using the system, patients should be able to execute the exercises for rehabilitation without the help of a nurse or physiotherapist. The aim of this paper is to find out how to design a system that is able to bring older adults into action by themselves. This can be achieved by exploring strategies that are used by existing technologies and finding a way to apply those to the final system.

To reach the goal, literature will be researched as well as existing systems that relate to the requirements of this paper. This paper will search an answer for the following questions ‘How to design a system that is able to bring older adults into action by themselves?’, ‘What strategies are used by existing systems to bring older adults into action?’ and ‘How to apply those strategies into one system?’.

To answer those questions, a literature research relating to self-management rehabilitation among older adults will be conducted. These findings result in a list of requirements. On top of that, research to existing systems is used to answer one of the questions. This information will be used to generate ideas for the prototype that in the end will be evaluated and tested.

2 Background Research

Over the past five decades the percentage of adults (18+) of the population in The Netherlands increased with 14%. According to CBS (2014), the population is aging more and faster. As 75% of the hip fracture patients are at an older age, the amount of hip fracture injuries will increase as well. In order to find ways to make patients rehabilitate without the need for help of a physiotherapist, strategies in self-management rehabilitation relating to older adults should be explored.

2.1 Education

One of the problems is that patients often lack the knowledge to do the exercises by themselves. During rehabilitation sessions the patient often depends on the instructions of a physiotherapist or nurse. The system should be able to take the place of the physiotherapist while the patient is doing the exercises by themselves. The barrier to start with exercises is lower if the patients knows how to do the exercises and what to expect. The system should be able to provide education about how to execute the exercises in the right way and raise awareness to the importance and benefits of those exercises. According to Varnfield et al. (2014), instructional videos can provide the knowledge to the patients without the need of a physiotherapist. On the other hand, Escolar-Reina et al. (2010) argues that written instructions are often performed incorrect. They state that the way information is brought to the patient has an influence on the patients endurance of self-management rehabilitation. During the ideation phase, methods to distribute the knowledge to the patient should be explored.

To increase the patients interest in rehabilitation, goal setting can be used. Being able to set goals is important for patients to maintain their rehabilitation schedule. Only one in five older adults achieve the advocated amount of physical activity. In consequence, most patients need to make some behavioral changes in order to be more active. This can be achieved by introducing small goals in addition to the overarching goal of returning home as soon as possible. As Bongartz et al. (2017)states, when patients have clear goals for themselves, they are more likely to maintain their rehabilitation schedule. Examples of those goals are achieving a set amount of exercises each day or increasing their walking distance to a certain distance.

Those goals can be adjusted along the way which give patients the feeling they are progressing faster if they achieve their set goals.

In order to sustain the improvements of self-management rehabilitation there should be some sort of intervention. According to Du et al. (2017), self-management rehabilitation has moderate short-term improvements and small, but significant, long-term improvements. The goals can be structured in short-term goals and long-term goals as well. Meng et al. (2018) point out, in 4 order to sustain the short-term benefits of self-management rehabilitation for the long-term there is need for intervention.

2.2 Reminders

It is common that older adults, because of their age, experience cognitive deterioration. Due to the cognitive impairments the patients might forget to do exercise or the importance of the exercises. The system should be able to remind the patient about doing exercises. This can be well combined with education about the importance of the exercises. In addition, as Escolar-Reina et al. (2010) point out, patient often find it difficult to fit the rehabilitation exercises in their daily routine. To overcome this problem exercises should be designed to minimize the interruption of the patients daily routine as well as using reminders. The exercises should be simple in order for patients with cognitive impairments to execute them as well.

2.3 Feedback

During the rehabilitation sessions with an physiotherapist, patients receive feedback about how they are progressing in their rehabilitation schedule and if they are executing those exercises in the right way. Escolar-Reina et al. (2010) state that proper and personal feedback increases the patients endurance of rehabilitation exercises. Feedback should be included in the system to inform patients about their progress and their performance of the exercises. This could be combined with education. If the system is able to track if exercises are executed incorrectly, feedback could be in the form of instruction about the exercises. According to Peel et al. (2016), patients who do receive feedback are significantly more active compared to the patients who do not receive feedback. Showing progress compared to other days or weeks during rehabilitation can increase patients motivation.

This concludes that self-management rehabilitation is feasible. The user should receive the education the patient needs in order to be able to execute the exercises without physiotherapist or nurse. The system should incorporate functions like goal-setting to keep patients motivated to do rehabilitation exercises on their own. The goals should be able to change according to the patients skills and abilities. Using sensors to measure progress and give feedback based on the goals and the progress of the patients. The feedback needs to be useful and personal. Reminder should be send to users when they forget to be active on their own. The system assists in the self-management rehabilitation and further research should point out if such a system is able to sustain the short-term benefits for the long-term.

2.4 Motivation for rehabilitation

The following motivators have proven to be successful in self-management rehabilitation. First, using barriers that keep them from exercising as a motivator for becoming active. As Schutzer & Graves (2004) point out, barriers for being active are highly related with motivators for being active. To turn those barriers into a motivator, patients should be given knowledge about the benefits of certain exercises which might spark their intrinsic motivation. Peel et al. (2016) support this by stating when patients receive information about their performance, their activity level increases. With the knowledge about benefits of exercises, the patients know what exercises to do in order improve their current situation. Thus this system should include patient education about the importance and benefits of the exercises. Increasing the knowledge about the importance can spark the motivation of patients.

3 Method

The Creative Technology Design Process by Mader and Eggink (2014) is used during this project. The aim of this method is first generating a lot of ideas in different directions. Thereafter the requirements and research is used to narrow down on the generated ideas until there is a desired idea. This is achieved via multiple phases.

During the Ideation phase ideas will be generated. Brainstorms will help to generate ideas in different directions. With the help of the requirements and the research these ideas will be narrowed down to potential solutions. Those ideas will eventually be narrowed down to the final idea.

During the Specification phase, the different components will be evaluated and chosen based on the requirements. Parts of the prototype are build to test and tweak the prototype.

During the Realization phase the information and parts of the previous phases are combined to create a final prototype that is able to be evaluated.

The Evaluation phase is used to evaluate the prototype and perform user tests. The results of the user test can give valuable insights in to what extend the requirements are met. Furthermore, the feedback can be used to improve the prototype.

4 Ideation

This chapter is focuses on the ideation phase. In order for the ideation to be effective, system requirements should be constructed. The list of requirements are created with caregivers, physiotherapists and potential users. Besides that, the state of the art is researched to find issues with the created list of requirements.

4.1 Requirements

Reminders

The system must remind the user to do exercises. When the reminders show should be based on the time of the day and if the patient is active or not. The reminders need to to be easily understood by the patient.

Explain exercises

The system must be able to explain the exercises to the patient. The system should be able to help the patient without the need of a nurse or physiotherapist.

Safety

The exercises should be constructed in a way that the odds of a patient falling is as low as possible. Because the system is used without help of a nurse or physiotherapist, fall prevention is a must.

Rewarding

The system needs to be rewarding for the user. After doing exercises, the user should be able to feel rewarded to increase motivation.

Comfortable activity detection

The system should be able to detect the amount of inactivity of the patient. This can be achieved in various ways and should be unintrusive for the user.

4.2 Related Work

There are several related technologies that all try to assist in the rehabilitation or being active in general. Because this topic is relatively new, most of the new emerging products have not yet proven to be successful or are still experiments. Nevertheless, these technologies show interesting concepts which might be applied to the system in this project.

4.2.1 Activity Trackers

In order to give feedback, some way of tracking the user is necessary. One way to do this is an activity tracker. Activity trackers are growing in popularity. An example of a wearable activity tracker is the Fitbit . Quoting “Fitbit Official Site for Activity Trackers & More” (n.d.), “Fitbit motivates you to reach your health and fitness goals by tracking your activity, exercise, sleep, weight and more.”. The functions of most activity trackers include goal-setting and daily feedback of the users activity. Fitbit uses sensors to gather the data about the users activity in order to compute personal feedback related to the users goals. If the system should be able to give feedback to the patient, the system should include sensors that are able to track progress of the patients. The problem with activity trackers is that the patient has to wear it in order to track activity. The patient might forget wearing the activity tracker.

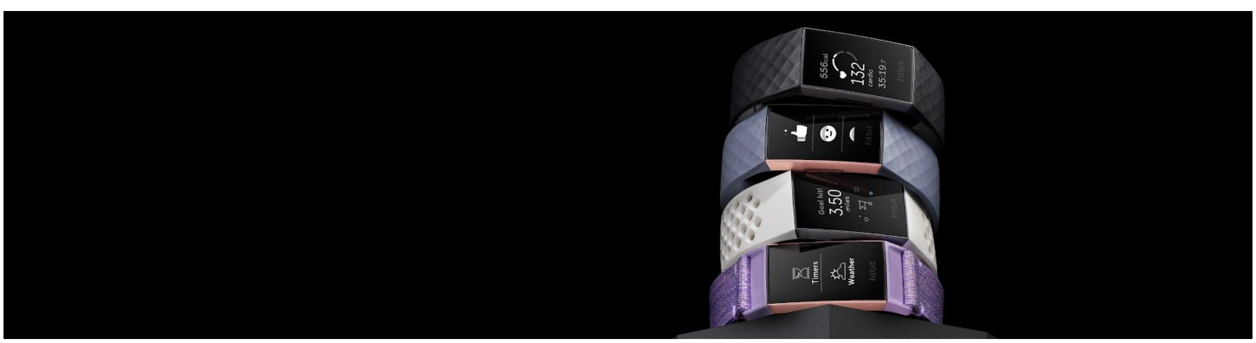


Figure 1: Fitbit Activity Tracker; Source [1]

4.2.2 Active Video Games

Besides the activity tracker, a variety of the found technologies are gamifying the rehabilitation process in order to make it more fun. There are several ways to include Active Video Games (AVG’s), also known as exergames, in the rehabilitation program. Zeng, Pope, Lee, and Gao (2017) define AVG's as a game where users have to physically interact with on-screen avatars and this definition will be used in this paper. The Nintendo Wii and Xbox Kinect are examples of AVG's. According to Zeng, Pope, Lee, and Gao (2017) older adults enjoy being physically active using an AVG. More important, improvements during rehabilitation using AVG’s are possible compared with normal therapy. Research points out that AVG’s have no negative effects compared to normal therapy. Hence for the patients, AVG’s can be beneficial during rehabilitation. Besides improvements during rehabilitation, patients perceived improvements in their physical, social and psychological well-being, as stated in Peng, Crouse, and Lin (2013). The perceived improvements and the possible improvements during rehabilitation are favorable for an increase in motivation of patients.

There are smartphone games that promote physical activity and actually need physical activity in order to earn in-game rewards. For example, Run An Empire where one can go for a run create an empire based on your running route. While this is probably not fitting for older adults, it shows the popularity of gamification related to health and activity.

SilverFit is an example of a system that uses AVG’s in order to assist in self-management rehabilitation. SilverFit consists of mini games that are specifically designed for older adults and their rehabilitation exercises. The users can play by themselves or with others to create a competitive surrounding. According to Skjæret-Maroni et al. (2016), the exergames used for rehabilitation should be selected with care. Taking into account which types of movements are necessary for the recovery of the patient and the difficulty of the exercises to keep the motivation high. SilverFit educates the patient and gives feedback in the form of receiving a higher score if the exercises are executed well. However, there is no evidence found that the Silverfit is able to send reminders to their users.

In general, the acceptance of games in rehabilitation of older adults is positive. According to Martini et al. (2019), older adults used the games on regular basis if they are available. Therefore, the concept of AVG’s can be used to motivate and possibly improve rehabilitation. Additionally, Martini et al. (2019) state that these games can be used in self-management rehabilitation.

Figure 2: SilverFit Application; Source [2]

4.2.3 Social Assistive Robots

Social Assistive Robots (SARs) are robots that are created to prevent loneliness. However, these robots are also able to assist during rehabilitation therapy. According to Kellmeyer and Müller (2018), for SARs to be active, the robot and the patient should have the same goals. Besides that, the robot should be trustworthy and reliable. The robot will help the patient with doing exercises and give feedback based on the performance. If the user completes an exercise, the robot gives a compliment to the user. However, there should be more research in the acceptance of such a robot by older adults.

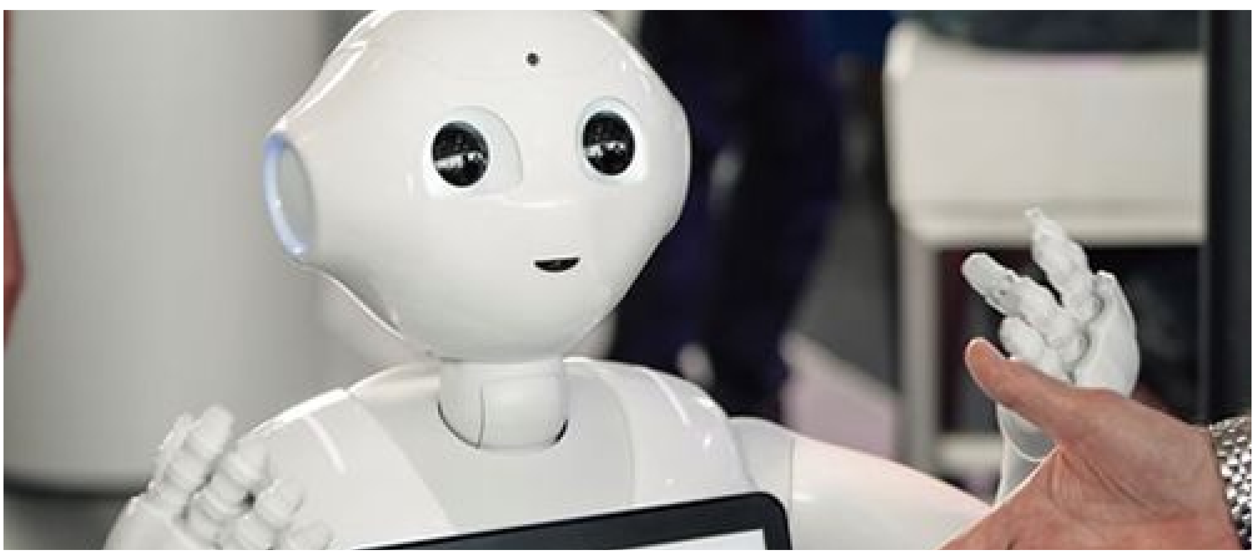


Figure 3: Pepper, the social robot; Source [3]

4.2.4 Conclusion

The aim of this research is to find strategies that are able to support patients in self management rehabilitation. This research concludes insightful information for the ideation phase. Interesting findings are the trend of gamifying rehabilitation processes and a social robot that assists the patient during rehabilitation. During the ideation phase it would be interesting to see what concepts could be applied to the system.

The researched showed that feedback is an important feature to enhance the motivation of the patient. Activity Trackers such as Fitbit already include this with success in their products. During the ideation phase different methods to give feedback to the user should be explored that will show the patients if they are on the recommended activity level for that day.

One part of the AVG’s that stood out are the rewards. When progress is made, rewards are given to the user. This could be included to give people an incentive to start doing exercises and receive a reward. Besides that, the social robots show that interaction could be useful for transfer knowledge to the user. The motivation also has benefits from having a mutual interaction.

Those findings combined with the research lead to interesting and useful insights that will be used during the brainstorm sessions. During the ideation those individual elements can be elaborated to find how to implement those into the final system.

4.3 Ideas

The Using the requirements and the research, brainstorm sessions were conducted. This resulted in a variety of ideas which are more elaborated and described in this chapter.

4.3.1 Social Robots

The social robot Pepper is developed with the intention to make people happy. The robot is able to have conversations with people and read emotions. The creators of Pepper allow other developers to create content and use with other intentions, for example rehabilitation.

The idea is using existing sensors like the Fitbit and Mox to gather data about the activity level of the patient. If a patient activity level decreases below a certain level, the system should start encouraging the patient to do exercises. The Pepper robot can be used to do exercises together with the patient. When patient X's activity level is below the threshold and in their room, the robot is notified and will drive to the room of patient X. The robot can show instructional videos to assist the patient with the exercises. Because this idea is not the most innovative solution and the Pepper robot is an expensive solution, the idea was not elaborated more.

Figure 4: Robot dogs in nursing home; Source [4]

4.3.2 Sensor Mat

For the system to know if the user is doing exercises, the movement of the user has to be tracked in a way. This could be part of the final product. During the brainstorm the idea of a pressure mat was generated.

The pressure mat is located in front of the chair. If the user is seated, their feet are resting on the mat. The sensors in the mat know that the patient is sitting on the chair. The exercises that the patients have to do are all using the patient's legs. This means that all exercises can be measured using this mat. The system also knows if the exercises are executed in the right way. This makes it easier to give personal feedback to the patient during the exercise session.

Figure 5: Sketch of the sensor mat

4.3.3 Remind Wristband

After the visit to the nursery home ZorgAccent where patients are rehabilitating it became clear that reminding patients to do their exercises is very important. Most patients were not doing exercises besides the half hour therapy sessions. During a brainstorm session, the idea of a wristband that is able to remind patients to do exercises when they are inactive. The wristband is prepared with sensors that measure movement and a form of reminding. This could be vibration, sound, light or a combination of those.

This only solves the requirement of reminding the patient. There is still need to assist the patient with the exercises. Therefore the previous idea with the mat with pressure sensors can be attached to the chair in the room of the patient. This mat is able to sense the feet of the patient during the exercises. As all exercises are using the legs, all exercises can be measured with the mat. If the patient executes the exercises in the wrong way, the mat will know this and show an explanatory video of the exercise.

To get valuable feedback of the generated ideas, the idea was pitched to experts. The experts included were multiple physiotherapists and nurses. The feedback of that session was used to combine the good elements of the previous ideas with new generated ideas.

4.3.4 Rehabilitating Flower

During the expert feedback session, new ideas arose and were elaborated to the final idea. The sensor mat on the floor is not the optimal solution, because a lot of the patients use a wheelchair to get around. If the mat is located on the floor, the user has to go to that spot with the wheelchair before the system is able to do the exercises. Therefore the mat should be changed in order to be placeable in a chair. The location should be easily adjustable to move the mat to a wheelchair when the patient is using that. The sensor mat is used to measure how long the patient is inactive while sitting on the chair. When a threshold is met, the system should remind the user of doing exercises.

A visit to the nursing home was arranged to see what exercises the patients are doing during the physiotherapy sessions. In addition, the visit could be used to find a suitable spot for the system to be effective. The place the patients are spending the most time inactive is within their own room. Outside their own room they have the physiotherapy session, the social environment where they are talking with others or eating. The system should be located in a place where the patient is inactive already, in order for the system to remind the patient when that is happening.

Another part of the feedback of the expert feedback session included that the patients are already wearing sensors for research. Having a system without the need of sensors attached to the user would be preferable, because without something attached to the body of the user, the user or a nurse does not have to charge the sensors every now and then. Besides that, if the user does not have to wear a sensor, the user can not forget to put it on or to accidentally take it off.

Another way to remind patients is to have a visible representation of their activity level. The representation should change when the patient is less active to remind the patient that it is time for exercises. The system should still be able to give instructions to the user if the patient forgot how to do exercises. Because the system is located in the room of the patient, the brainstorm focussed on objects that could be already in the room and are able to represent an amount of activity. Almost all rooms have flowers and a flower is able to represent activity level in a good way. The more the flower is dangling down, the higher the inactivity level of the patient.

To explain the idea, a possible scenario would be like: ‘The user is sitting inactive on the chair for a while now. The flower is slowly starting to dangle down. The user sees the flower dangling down and starts to do exercises. Slowly the flower raises back to its full blooming state. The user now knows that they are at the right activity level.‘. The flower represents that doing exercises is good for you and when you do the exercises often, the recovery will go faster.

However, there is still need for instructions of exercises to be brought to the user via the system. If the flower dangles down and the user is not starting the exercises by themselves, they need help with the exercises. The system needs to be able to transfer knowledge to the patient in some way.

5 Specification

In this chapter the final idea will be elaborated to a functional prototype. Different options will be considered and therefore design choices will be made.

5.1 Sensor Mat

There are several ways to detect if a person is seated on the mat. A pressure sensor or pressure switch is the first sensor that comes to mind. However, a LDR (light dependent resistor) could be used as well such that the sitting person block light coming in the sensor. One of the requirements is that the system should be comfortable for the patient. Therefore, the sensor mat should not have parts that are sticking out.

The pressure sensor is very thin and can be installed under the mat and still being able to measure the pressure. Because it is very thin, the user does not feel a difference when sitting on the sensor. The pressure sensor was tested by using a mat on top of the sensor. The sensor was able to distinguish a presence of a person on the chair all the time. This concludes that one pressure sensor is enough to detect if someone is in the chair.

5.2 Flower

The flower represents the activity level of the user. The initial idea let the flower dangle down more when the activity level is low. There was no solution found using the components that were available to achieve something like that. Thus the representation method had to be changed. Another way to represent the activity level is using the height of the flower. It is mechanically easier to raise and lower the flower to represent the activity level. The lower the flower gets, the more it will be hidden within the vase. The flower controller is discussed in Chapter 5.3.

The flower itself should be clearly visible from all angles, such that the patient can place the flower anywhere in the room and still see the flower. The flower needs to have a stem in order to rise above the leaves and hide within a vase.

5.3 Flower Controller

The flower is controlled by an Arduino microcontroller to be able to fit in a small space. An Arduino Uno is most suitable as this microcontroller is of reasonable size. In addition, the Arduino Uno supports Pulse-Width Modulation which is necessary for controlling a servo motor. The Arduino is able to communicate with other application, making it easy to add applications to the system that receive data from the Arduino as well.

The flower should be able to move up and down. A servo motor can drive a gear that in turn drives a gear rack. This converts the circular motion to vertical motion. The flower can be attached to the gear rack to move up or down. The servo needs to be controlled precisely in order to prevent overspinning. If the gear drives the rack for to long, the gear and the gear rack will disconnect. Therefore, the servo motor should be selected with this in mind.

There are different types of servo motors available. A continuous servo motor is able to rotate infinite in both directions with the ability to control the spin direction and speed. A 180 degree servo motor is able to spin back and forth over 180 degrees with the ability to control the angle the servo motor should be at. This makes the 180 degrees servo motor more suitable, because it can be controlled using degrees instead of a speed. The precision makes it nearly impossible for the gear and the gear rack to disconnect that would otherwise result in a system failure.

5.4 Speech and Recognition

In order to transfer the knowledge about how to execute the exercises to the patient, a variety of methods were explored. First video was considered, however with video there is need for a screen. To keep the system easy to fit within a vase, audio is chosen. The system should also be able to listen to the patients answers and act accordingly.

In order to achieve that, a P5 library is used that has access to Google’s speech and recognition software. Using this library, it was easy to test the exercise sentences because the library has text to speech software. This could potentially be very easy to implement new sentences without the need to record the voice. However, the sound of the voice was very unnatural. Therefore all the spoken sentences are pre-recorded to ensure a voice that is more natural and pleasurable to listen to or interact with.

Another reason this library is chosen is because the speech recognition is also included. With the functions, the system can start listening using a microphone. The system triggers when something is said and the speech recognition software turns the speech into text. The text can be used within the program to determine what is said and what to do next.

Finally, this library is open source and free to use. Other alternatives like an Arduino voice recognition module cost ranges between €20,00 and €50,00. Besides the high costs, this module understands a limited amount of commands while the library is able to recognize all words. In theory, using the library there are unlimited command possibilities. Therefore there is the possibility for the system to be expanded with more functions and commands instead of only yes or no questions.

6 Realization

After most of the design choices are made, the prototype needs to be realized. The hardware needs to be installed and the coding of the Arduino and spoken exercises has to be put together. This chapter covers the installation process of the hardware of the prototype.

6.1 Components

The system uses multiple components that are connected. This section discusses the realization of each component.

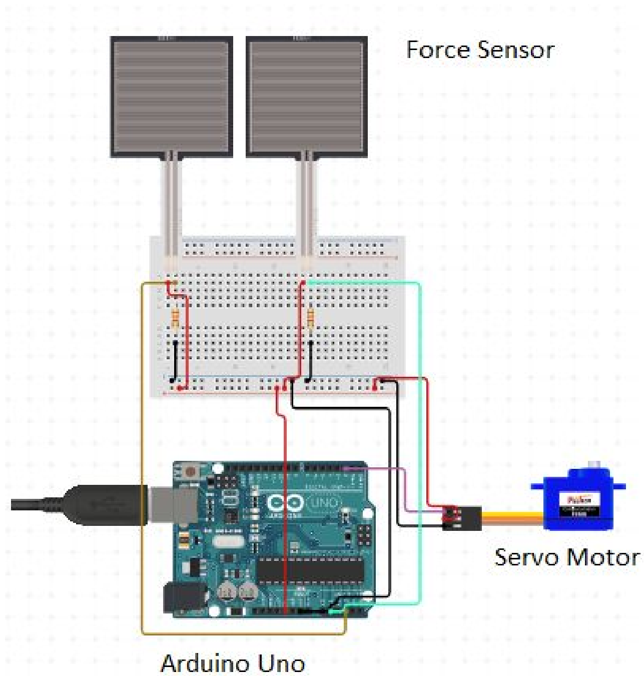
6.1.1 Flower

A plastic flower in a bright red color is used in the final prototype. The bright red color differentiates a lot from the green leaves. The fake flower had to be hidden when the patient is inactive. Therefore more greenery was added in which the flower can subtract to be hidden for the user while at the same time the greenery covered the electronic system to give it a more attractive look. The first prototype used a tube in which the flower could be hidden. Eventually the tube is removed to lower the friction and making it easier for the servo motor to raise the flower.

Figure 6: The flower used in the prototype covered with greenery

6.1.2 Flower Controller

The system is controlled using an Arduino Uno. This microcontroller is able to execute the required tasks. Sensors are attached to measure if the patient is sitting in the chair for a long time. A servo motor is used to change the appearance of the flower. Figure 7 shows a schematic of the systems circuit. Figure 8 shows the gear and rack that are laser cut. The flower is attached to the gear rack that is driven by the gear to go up and down. The code that is used to control the system can be found in Appendix A.



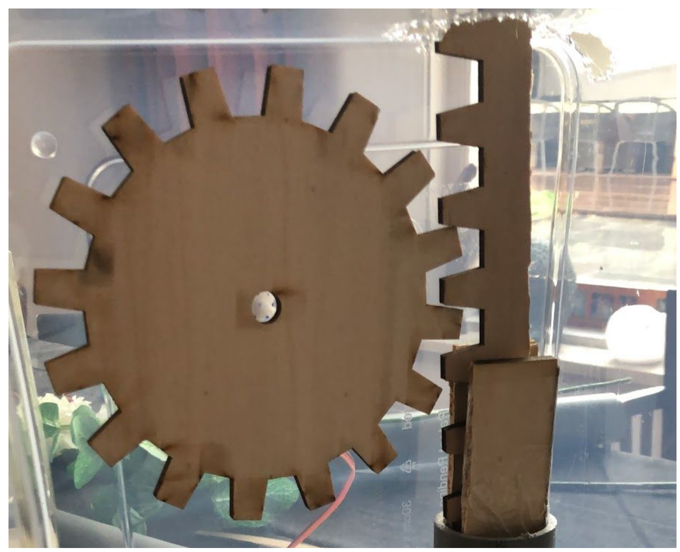


Figure 7: Circuit of the Arduino

Figure 8: Gear and rack

6.1.3 Exercise Controller

The exercises are created with help of the physiotherapists. They provided a few exercises that patients are allowed to do by themselves. The exercises are explained using the speech and recognition library. The system will guide the patient through the exercises with step by step explanations. A small example shows how the exercises guide the patient step by step: “Let’s start with the first exercise. Please ensure that you are sitting straight in the chair with your feet flat on the ground. Now we slowly lift our toes until only our heels touch the ground. Slowly roll your feet from the heels towards your toes until only your toes touch the ground…”.

After each exercise, the patient is asked if the patient managed to complete the exercise. After the patient answers, the system continues with the next exercise or explains the previous exercise depending on the answer. The sensor mat is able to detect differences between being inactive and doing exercises in the chair. When someone is doing the exercises, the input data fluctuates. This data could be used to confirm that the patient is being busy with the exercises or not.

The prototype includes three exercises that are provided by the physiotherapists. The patients normally receive those exercises on a flyer. These exercises are chosen because patients are able to do the exercises while seated. This means there is no risk of falling making the exercises more safe for the patients. The code that drives the exercises can be found in Appendix B.

6.1.4 Speech and Recognition

The most desirable way for patients to use the system is by using their voice as patients do not have to touch or walk to a spot before they are able to interact with the system. The system can deduct what the patients are saying by use of speech recognition software. An existing library for P5 is used that connects with Google’s services. This gives the system access to one of the most accurate speech recognition software available. The drawback of using this speech recognition is that the system needs to be connected to the internet in order to use the speech recognition.

The speech recognition is able to recognize almost everything a person says, however the prototype is limited to listen only to yes and no. If something else is said, the system asks the person to answer with yes or no followed by the previously asked question. The prototype is designed to listen to the Dutch language, because the test participants will be Dutch. The prototypes language can be adjusted by changing the sentences in the code. The speech recognition is able to understand this as long as Google's speech recognition includes that language.

6.2 Costs

The costs are computed using the price of parts when bought individually. This means that when the product goes into production on a larger scale, the price can be brought down significantly. The parts can be bought in bulk, lowering the price per piece and some parts could be custom made to remove all obsolete parts that come with parts that are bought.

Component	**Price**
Arduino Uno	€13,00
Servo Motor	€3,75
Mechanical Parts	< €1,00
Flower and greenery	€2,50
Cables and other components	€2,00
Total	**€22,25**

6.3 Prototype

The final prototype combines all components. Figure 9 shows the final prototype with all parts connected. The measurements of the sensor mat are processed by the Arduino. When inactivity is detected the Arduino drives the servo motor to change the height of the flower. When the flower is at its lower state, the Arduino starts the exercise application. The exercise application controls when a pre-recorded message should start. In addition, the application listens to the user and uses the data to decide whether the system should go on to the next exercise or if the system should explain the exercise more. If there is no response, the system tries to do the exercises later. The system stores a boolean value (true or false) which is false at the beginning of the program. When a patient is failing to do the exercises three times in a row, this boolean turns true. This could be used to inform caregivers about patients having difficulties with exercises or the system.

Figure 9: The final prototype

7 Evaluation

7.1 Functional Evaluation

In this section, the functional requirements will be evaluated and discusses to what extend the requirements are met.

7.1.1 Flower Controller

The system is able to read and use the incoming data from the sensors. This means that the system is able to check if the patient is sitting in the chair. When the patient is inactive for to long, the system triggers the servo motor to slowly hide the flower. This happens in stages such that the patient is able to see the effect of being inactive. For test purposes, the prototype was measuring the pressure sensor every 20 seconds and the inactivity threshold was set at 3 periods. All periods are adjustable such that the intensity and the amount of exercises can change according to the stage of the rehabilitation process of the patient.

If the person starts to be active by themselves, the flower starts to rise again. If the flower is completely gone, meaning that the patient is inactive for too long, the system sends a signal to the exercise controller to start the exercises with the patient.

7.1.2 Exercise Controller

The exercise controller will start when the flower controller sends the trigger that the flower is completely hidden. The exercise controller is able to read and use the trigger and connect with the patient using the recorded voice messages and the speech recognition.

Between every question the system asks to the patient, the system will wait for an answer consisting of ‘ja’ or ‘nee’, meaning ‘yes’ and ‘no’. Three different situations can occur during that waiting period and the system can handle all situations well. First, the system can be ignored by the patient. If this happens, the system will ask the question again after a few seconds. When the patient still ignores this question, the system informs the patient that it stops trying and the system will try it again at a later time. When the exercises are ignored three times in a row, the caregiver can be notified to inform why the patient is not able to do the exercises. This way the nurse knows that the patient is either not doing the exercises or is not understanding the system and needs explanation.

Second, the patient could say ‘ja’ or ‘nee’. The system will either ask if the patient is sure when the patient said no or go on to the next exercise if the patient said yes. When the system proceeds to the following exercise, a trigger will be send to the flower controller to return a part of the flower until all exercises are done and the flower is completely visible.

Finally, the patient could say something else. This means that the patient is responding, however not in a way the system understands yet. If this happens, the system asks the patient ‘could you answer the question with yes or no’ following by the last question asked. To keep the system from asking the question when, for example, the patient is speaking to someone, the system will stop and try again later when the patient is not responding with yes or no within three times.

7.1.3 Speech and Recognition

The voice recordings can be easily added in the application. After recording the voice the file should be placed in the sounds folder. The pre-recorded messages have a slow speaking speed to make it easier to understand. The system does not yet have an option where the patient is able to customize the sound of the speaker voice.

The system listens to the user after a question is asked and waits until the answer is given. Testing showed that the speech recognition is very accurate. It could identify most sentences within one try. This means that other speech commands could be implemented to make the system able to have more complex conversations with the patient. However, the prototype is only able to listen to yes and no. Everything else will be returned to the exercise controller as ‘different’.

7.2 User Evaluation

This section will discuss the user tests and the outcomes. The functional prototype is used during the user tests. The aim of the first test is to discover if the system works properly when the system is used. After the test the participants are asked to answer questions which would result in feedback and possible flaws in the system. The participants will be informed about the prototype in the same way how the product would be introduced to the end users. The test plan can be found in the Appendix D.

7.2.1 Test I

The first user test includes six students aged between 21 and 26. The age of the participants is not important as the aim of the test is to see if the functionality of the prototype is working as intended. Especially to find out if the sensor mat is receiving the right values and if the inactivity threshold is configured in the right way. Besides that, the test will point out if the flower mechanism works reliably. Finally, the test will point out if the exercises flow is correct and the speech recognition is able to detect the voices of all participants.

Results I

During all tests the mechanical parts were working as intended. The sensor mat data did show some incorrect data every now and then. However, the system was programmed to take an average of the measuring period to take in account that the sensor might fail to send data to the Arduino sometimes. Therefore, this caused no problems with raising or lowering the flower. The system was able to initiate the exercise program at the correct time in all the tests as well.

During the test, it became clear that the exercises could be improved. The user feedback also said that some of the exercises did not run smoothly. The exercises needed to be more instructive and at some moments, the pause between each exercise was not long enough. The observations and the feedback were combined to improve the exercise messages and these were recorded for the improved prototype.

7.2.2 Test II

During the second user test, the improved prototype is used. The participants of the second test were aged between 45 and 54. The aim of the test was to see if the exercises were understandable. In addition, the test could give some valuable insights in how people react to the prototype.

Results II

During the second test some problems occurred with the servo motor. The motor lost some power due to unknown reasons and was not able to lift the flower consistently. Therefore a new servo motor was used and in addition the prototype was reworked. A less heavy fake flower was used in combination with more sturdy mechanical systems to improve the consistency of the flower.

The participants provided some interesting feedback as well. First, the voice assistance during the exercises was found to be motivating and instructive. This also gave some participants the feeling that someone is helping them. This motivated them to go on with the exercise. Second, the participants showed that the ability to interact with the system can be beneficial for the motivation as well. However, because the prototype is only able to answer to yes or no answers, for some participants the conversations felt stiff. This can be solved by adding more commands that will ensure more smooth conversations.

Another part of the feedback is the distinctiveness of the flower. The flower used in the prototype was from some angles still visible when in the lowest state. Therefore, the difference between very active and not active was difficult to distinguish.

8 Conclusion

The research found several key elements for self-management rehabilitation. Some of these elements are also used by existing products related to motivation and rehabilitation. A prototype was developed which was able to combine those key elements into one system. During the user tests the concept showed potential. The users were motivated doing the exercises, because there was an interaction between the user and the system. Additional user tests with the potential end users should be used to find out about the acceptance of such a system.

Results of the user tests showed that the requirements are realised in the prototype. First of all, the prototype explains the exercises, educating the patient how to do the exercises. Secondly, the prototype uses the flower to give feedback and remind the user about their activity level. The patients are able to see the flower grow when exercises are done. Finally, the prototype is safe as the exercises are all in sitting posture limiting the risk of falling during an exercise.

Although all requirements are realised, some of those could be improved. For example the distinctiveness of the flower could be improved when a stronger servo motor is used. The prototype could not use a big distinctive flower because the motor was not able to lift the flower in the air. Another improvement could be the way a patient is reminded. In the current prototype, the patient could completely ignore the flower. However, there would still be some a reminder when the system starts doing the exercises together with the patient.

Furthermore, the evaluation provided some interesting insights in possible future developments regarding this concept. Possible future improvements are covered in the next chapter.

8.1 Future Works

The prototype showed potential for future developments. The first recommendation is to extend the amount of recognized commands with the speech recognition. The user tests showed the benefits of interaction with the system, this can be further improved by having a more fluid conversation. The speech recognition has an important role in the smoothness of the conversation. The final prototype is able to understand ‘yes’ and ‘no’ which limits the conversation the system can have. The system could be extended with more speech commands to ensure a more fluid conversation. The ability to ask questions during the exercises should be looked into as well. If a patient does not understand what is happening and ask a question, the system should be able to answer that. Not only between the exercises, but also during the exercises. For example if the patient asks during an exercise for some more explanation, the system should give more information to the user before continuing.

Furthermore, adding an interface for caregivers is recommended. During the project was found that the exercises will change during the course of the rehabilitation. In the beginning, the patient is able to do simple and not too intense exercises while in a later stage the patient might want to do some more intense exercises. Having an interface for the caregivers will improve the system. The caregivers should be able to add and remove exercises for a patient. The interface could also showcase information about how often the patient uses the system and if the patient skipped some exercises. The caregiver then knows if the patient needs help or if the exercises are on the right level for the current stage of the rehabilitation. Besides that, the caregivers could use such an interface to change the desired amount of exercises each day.

Another recommendation is combining the system with a social robot. When the user does not understand the exercises explained via audio, the social robot could come and assist the conversation with video explained exercises. Social robots are already capable of having conversations, this might be useful when the user gets stuck during an exercise. In addition, this might give the patient the feeling a caregiver is helping them.

References

Bongartz, M., Kiss, R., Ullrich, P., Eckert, T., Bauer, J., & Hauer, K. (2017). Development of a homebased training program for post-ward geriatric rehabilitation patients with cognitive impairment: study protocol of a randomized-controlled trail. BMC Geriatrics, 17(1), 214. https://doi.org/10.1186/s12877-017-0615-0

CBS. (2014). Half of Dutch adults will be over 50 in 2019. Retrieved November 2, 2018, from https://www.cbs.nl/nl-nl/nieuws/2014/38/in-2019-helft-van-volwassenen-50-plusser

Du, S., Hu, L., Dong, J., Xu, G., Chen, X., Jin, S., … Yin, H. (2017). Self-management program for chronic low back pain: A systematic review and meta-analysis. Patient Education and Counseling, 100(1), 37–49. https://doi.org/10.1016/j.pec.2016.07.029

Escolar-Reina, P., Medina-Mirapeix, F., Gascán-Cánovas, J. J., Montilla-Herrador, J., Jimeno-Serrano, F. J., De Oliveira Sousa, S. L., … Lomas-Vega, R. (2010). How do care-provider and home exercise program characteristics affect patient adherence in chronic neck and back pain: A qualitative study. BMC Health Services Research, 10(1), 60. https://doi.org/10.1186/1472-6963-10-60

Fitbit Inc. (2017). Fitbit Official Site for Activity Trackers & More. Retrieved October 21, 2018, from https://www.fitbit.com/no/home%0Ahttp://www.fitbit.com/#i.1mxltfgou4fqdt

Martini, L., Vannetti, F., Fabbri, L., Gerli, F., Mosca, I., Pazzi, S., … Savazzi, F. (2019). GOAL (games for olders active life): A web-application for cognitive impairment tele-rehabilitation. In IFMBE Proceedings (Vol. 68, pp. 177–182). https://doi.org/10.1007/978-981-10-9023-3_32

Meng, K., Reusch, A., Musekamp, G., Seekatz, B., Zietz, B., Steimann, G., … Faller, H. (2018). Selfmanagement education for rehabilitation inpatients: A cluster-randomized controlled trial. Patient Education and Counseling, 101(9), 1630–1638. https://doi.org/10.1016/j.pec.2018.03.027

National Institution for Public Health and the Environment. (n.d.). Nationaal Kompas Volksgezondheid - RIVM. Retrieved November 4, 2018, from https://www.rivm.nl/Documenten_en_publicaties/Wetenschappelijk/Databases/Preventie_Ziekte_Zorg/Nationaal_Kompas_Volksgezondheid

Peel, N. M., Paul, S. K., Cameron, I. D., Crotty, M., Kurrle, S. E., & Gray, L. C. (2016). Promoting activity in geriatric rehabilitation: A randomized controlled trial of accelerometry. PLoS ONE, 11(8), e0160906. https://doi.org/10.1371/journal.pone.0160906

Peng, W., Crouse, J. C., & Lin, J.-H. (2013). Using Active Video Games for Physical Activity Promotion. Health Education & Behavior, 40(2), 171–192. https://doi.org/10.1177/1090198112444956

Philipp Kellmeyer and Oliver Müller. (2018). Robots as tools and partners in rehabilitation. Retrieved November 6, 2018, from www.sciencedaily.com/releases/2018/08/180817125404.htm

Schutzer, K. A., & Graves, B. S. (2004). Barriers and motivations to exercise in older adults. Preventive Medicine, 39(5), 1056–1061. https://doi.org/10.1016/j.ypmed.2004.04.003

Shelly Levy-Tzedek (n.d.). A robot congratulates a patient for correctly sorting the colored beakers. Retrieved from https://www.sciencedaily.com/images/2018/08/180817125404_1_540x360.jpg

Skjæret-Maroni, N., Vonstad, E. K., Ihlen, E. A. F., Tan, X. C., Helbostad, J. L., & Vereijken, B. (2016). Exergaming in older adults: Movement characteristics while playing stepping games. Frontiers in Psychology, 7(JUN), 964. https://doi.org/10.3389/fpsyg.2016.00964

Varnfield, M., Karunanithi, M., Lee, C. K., Honeyman, E., Arnold, D., Ding, H., … Walters, D. L. (2014). Smartphone-based home care model improved use of cardiac rehabilitation in postmyocardial infarction patients: Results from a randomised controlled trial. Heart, 100(22), 1770–1779. https://doi.org/10.1136/heartjnl-2014-30578311

Wildenbos G.A., Jaspers M.W.M., Schijven M.P., Dusseljee-Peute L.W. (2019). Mobile health for older adult patients: Using an aging barriers framework to classify usability problems https://doi.org/10.1016/j.ijmedinf.2019.01.006

Zeng, N., Pope, Z., Lee, J. E., & Gao, Z. (2017). A systematic review of active video games on rehabilitative outcomes among older patients. Journal of Sport and Health Science, 6(1), 33–43. https://doi.org/10.1016/j.jshs.2016.12.002

[1] https://www.fitbit.com/home
[2] https://silverfit.com/nl/toepassingen/fysiotherapie
[3] https://www.robotzorg.nl/product/pepper-robot/
[4]https://www.independent.co.uk/arts-entertainment/photography/japan-robot-elderly-care-ageing-population-exercises-movement-a8295706.html

Appendices

Appendix A - Arduino Code

```
/*
	The code to retrieve data from the pressure sensor and drive the servo motor. Serial communication is used to communicate with the exercise application.
*/

#include <Servo.h>
unsigned long startMillis;  //some global variables available anywhere in the program
unsigned long currentMillis;
const unsigned long period = 3000;  //the value is a number of milliseconds

String inputString = "";         // a String to hold incoming data
bool stringComplete = false;  // whether the string is complete

int statePin = 8;
int ledPin = 9;
int potPin = A0;

int pressurePin = A2;

Servo flowerServo;

int exerciseCount = 0;

boolean measureTime; // true -> BUSY WITH MEASURING TIME
boolean exerciseTime; // true -> BUSY WITH EXERCISING
int tryCounter; // POSSIBLE NOT IN ARDUINO BUT IN PROCESSING OR P5
```

```
int value; // VALUE

int flowerState; // CURRENT FLOWER STATE
const int flowerInit = 2; // MAX HEIGHT STAGE

int inactiveCounter; // COUNTS INACTIVE TIME PERIODS
int activeCounter; // COUNTS ACTIVE TIME PERIODS

void setup() {
  // initialize pins.
  pinMode(statePin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(potPin,INPUT);
  pinMode(pressurePin, INPUT);

  flowerServo.attach(10);



  // initialize serial.
  Serial.begin(9600);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);

  // initialize variables.
  startMillis = millis();
  exerciseTime = false;
  measureTime = true;
  tryCounter = 0;
  inactiveCounter = 0;
  activeCounter = 0;

  flowerState = flowerInit;
  for (int i = 0; i<flowerState; i++){
```

```
    changeServo(changeFlowerState(1));
  }
}

void loop() {

  if (measureTime){
    currentMillis = millis();
    if(currentMillis-startMillis >= period) {

        if (flowerState != 0){
//          Serial.println("Current millis at period.");
        //Do Stuff
        int avg = getAverage(pressurePin,5,4);
//          Serial.println(avg);

        // if average is above 50 -> inactive
        if (avg >= 70){
          inactiveCounter++;
          if (activeCounter == 0){
          changeServo(changeFlowerState(-1));
          } else if (activeCounter > 0){
            activeCounter--;
          }
        } else if ( avg < 70 && activeCounter == 3){ // if average is below 50 and been for more
than 3 times -> certainly active thus reset
          inactiveCounter = 0;
          activeCounter = 0;
          changeServo(changeFlowerState(1));

          // Average below 50 so increase active counter
        } else {
          activeCounter++;
```

```
        }
//        Serial.print("Inactive level: ");
//        Serial.println(inactiveCounter);
//        Serial.print("Active level: ");
//        Serial.println(activeCounter);
//        Serial.println("Average retreived.");

      } else {
        Serial.write("S");
        // SEND message to the exercise program
        measureTime = false;
        exerciseTime = true;
      }
      startMillis = millis();
    }
  }

  if (exerciseTime) {
      if (stringComplete){
//    Serial.println("String Complete in loop");

    // RECEIVE STATUS UPDATES FROM exercise program
    // CHANGE FLOWER STATE ACCORDINGLY

    if (inputString == "done"){ // should reset all values and start again
      measureTime = true;
      exerciseTime = false;
      exerciseCount++;
      flowerState = flowerInit;
//      Serial.println("MeasureTime = true");
    } else if (inputString == "flower") {
      changeServo(changeFlowerState(1));
    } else if (inputString == "tryagain"){
```

```
      flowerState = flowerInit/2;
      measureTime = true;
      exerciseTime = false;
    }
    inputString ="";
    stringComplete = false;

  }
 }

//  value = analogRead(potPin);
//  value = map(value, 0, 1023, 0, 255);
//  analogWrite(ledPin,value);
//  //Serial.println(value);
//  delay(100);

}

void changeServo(int state){
  int angle = map(state, 0,flowerInit,0,80);
  flowerServo.write(angle);
//  Serial.print("Servo angle: ");
//  Serial.println(angle);
}

int changeFlowerState(int state){
  if (state == 1 && flowerState < flowerInit){
    flowerState++;
  } else if (state == -1 && flowerState > 0){
    flowerState--;
  }
//  Serial.print("Change flower state to: ");
//  Serial.println(flowerState);
```

```
  return flowerState;
}

int getAverage(int pin, int seconds, int timespersecond){
  int temp = 0;
  digitalWrite(statePin, HIGH);
  for (int i = 0; i<seconds*timespersecond; i++){

    int data = analogRead(pin);
    data = map(data,0,1023,0,100);
    temp+= data;
    int delayTime = 1000 / timespersecond;
    delay(delayTime);

    Serial.print("Tussendata: ");
    Serial.println(data);
  }
  digitalWrite(statePin, LOW);
  temp = temp / (seconds * timespersecond);
  return temp;
}

/*
  SerialEvent occurs whenever a new data comes in the hardware serial RX. This
  routine is run between each time loop() runs, so using delay inside loop can
  delay response. Multiple bytes of data may be available.
*/
void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
```

```
    // if the incoming character is a newline, set a flag so the main loop can
    // do something about it:
    if (inChar == '\n') {
      stringComplete = true;
//      Serial.println("String Complete");
    } else {
      inputString += inChar;
    }
  }
}
```

Appendix B - Exercise code

```
<html>
<head>
	<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.4.5/p5.min.js"></script>
	<script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.4.5/addons/p5.dom.js"></script>
	<script src="../lib/p5.speech.js"></script>
	<script src="../lib/p5.serialport.js"></script>
	<script src="../lib/p5.sound.js"></script>

	<script>

	var serial;
	var portName = 'COM3';
	var inData;

	var listenTo = new p5.SpeechRec();
	var speakTo = new p5.Speech();

	var listenButton;

	var loadSoundButton;

	var listenBoolean = false;

	var resetProgram = false ;
	var programState = false;

	var recentWord;

	var wrongCounter;
```

```
var idleCounter;

var stageCounter;
var perStageCounter;

var currentTime;

var flowerState;

var timerIntervalStart;
var timerIntervalPeriodOne;
var timerIntervalPeriodTwo;
var timerIntervalPeriodThree;

var timerSpeechStart;
var timerSpeechPeriod = 6000;

var helloMr,helloMs,exerciseOne, exerciseOneDone, exerciseOneBreak;

function preload (){
	helloMr = loadSound('hallomeneer.mp3');
	helloMs = loadSound('hallo-mevrouw.m4a');
	exerciseOneStart = loadSound('oef1-beginnen.m4a');
	exerciseOneDone = loadSound('oef1-done.m4a');
	exerciseOneBreak = loadSound('oef1-break.m4a');
}

function setup() {
	createCanvas(800,400);
	background(255,255,255);
	listenTo.onResult = getResult;
	listenTo.onEnd = afterListen;
	speakTo.setRate(0.55);
```

```
		loadSoundButton = createButton('Load sounds');
		loadSoundButton.position(400,100);
		//loadSoundButton.mousePressed();

		listenButton = createButton('Speak');
		listenButton.position(20, 100);
		listenButton.mousePressed(doSpeak);



		timerSpeechStart = millis();

		stageCounter = 0;
		flowerState = 0;
		perStageCounter = 0;
		wrongCounter = 0;
		idleCounter = 0;
		recentWord = 0;

		//set up communication port
serial = new p5.SerialPort();       // make a new instance of the serialport library
serial.on('list', printList);  // set a callback function for the serialport list event
serial.on('connected', serverConnected); // callback for connecting to the server
serial.on('open', portOpen);        // callback for the port opening
serial.on('data', serialEvent);     // callback for when new data arrives
serial.on('error', serialError);    // callback for errors
serial.on('close', portClose);      // callback for the port closing

serial.list();                      // list the serial ports
serial.open(portName);              // open a serial port
	}

	function doSpeak(){
```

```
		listenTo.start();
		// serial.write("start\n");
		// console.log("start to arduino");
	}

	function draw() {
		currentTime = millis();
		//console.log(millis());


		if (resetProgram) {

			flowerState = 0;
			wrongCounter = 0;
			idleCounter = 0;
			stageCounter = 0;
			perStageCounter = 0;

			timerSpeechStart = millis();
			resetProgram = false;
			programState = true;

		} else if (programState) {

			if (recentWord != 0 && stageCounter%3 == 1){
				if (recentWord == 1){
					stageCounter++;
					recentWord = 0;
					idleCounter=0;
					timerSpeechStart = millis();
					timerIntervalStart = millis();
				} else if (recentWord == 2) {
					recentWord = 0;
```

```
                    speakTo.speak("Oke. Jammer, we proberen het later nog
een keer.");
                    serial.write("tryagain\n");
                    programState = false;

                } else {
                    speakTo.speak("Probeer met ja of nee te antwoorden.")
                    recentWord = 0;
                    stageCounter--;
                    wrongCounter++;
                    console.log("No yes or no answer Counter:
"+wrongCounter);
                }
            } else if (recentWord == 0 && !listenBoolean){
                listenBoolean = true;
                listenTo.start();
            }

            if (currentTime - timerSpeechStart >= timerSpeechPeriod){
                console.log("SpeakPeriod");
                if (stageCounter%3 == 1) {
                    idleCounter++;
                    console.log("Idle Counter: "+idleCounter);
                }
                if (wrongCounter > 1 || idleCounter > 10) {
                    speakTo.speak("We gaan het later opnieuw proberen om
oefeningen te doen.");
                    serial.write("tryagain\n");
                    programState = false;
                } if (idleCounter == 5){
                    stageCounter--;
                }
                // Stage 0 -> welkom
```

```
                    // Stage 1 -> luisteren naar ja
                    // stage 2 -> oefening
                    // stage 3 -> vraag of we door kunnen
                    // stage 4 -> luisteren naar ja
                    // stage 5 -> Pauze en door met oefening 2
                    // stage 6 -> vraag of we door kunnen
                    // stage 7 -> luisteren naar ja
                    // stage 8 -> Pauze en door met oefening 3
                    // stage 9 -> vraag of het is gelukt
                    // stage 10 -> luisteren naar ja
                    // stage 11 -> klaar
                    if (programState){
                        stages();
                    }



                    timerSpeechStart = millis();
                }
            }
        }

        function stages(){
            if (stageCounter == 0){
                helloMr.play();
                // speakTo.speak("Hallo meneer Wittrock, doet u mee met de
oefeningen?");
                listenNow();
                stageCounter++;
            } else if (stageCounter == 2) {
                if (perStageCounter == 0){
                    exerciseOneStart.play();
                    //speakTo.speak("Fijn. Ga rechtop zitten in de stoel.");
```

```
                perStageCounter++;
            } else if (perStageCounter == 1){
                //speakTo.speak("Zet uw voeten plat op de grond.");
                perStageCounter++;
            } else if (perStageCounter == 2){
                //speakTo.speak("Til langzaam de tenen omhoog zodat alleen de
hak van uw voet de grond raakt.");
                perStageCounter++;
         } else if (perStageCounter == 3){
                //speakTo.speak("En u kan weer ontspannen. Dat was de
warming up.");
                perStageCounter = 0;
                stageCounter++;
            }
        } else if (stageCounter == 3){
            //speakTo.speak("Is de oefening gelukt?");
            exerciseOneDone.play();
            listenNow()
            stageCounter++;
        } else if (stageCounter == 5){
            serial.write("flower\n");
            flowerState = 1;

            if (perStageCounter == 0){
                //speakTo.speak("Even tijd voor een korte pauze. Over een paar
seconde gaan we weer verder met de volgende oefening.");
                exerciseOneBreak.play();
                perStageCounter++;
            } else if (perStageCounter > 0 && perStageCounter < 5){
                // speakTo.speak("");
            //  perStageCounter++; PAUZEEEE
            } else if (perStageCounter == 5){
                speakTo.speak("Laten we weer verder gaan.");
```

```
                    perStageCounter++;
                } else if (perStageCounter == 6){
                    speakTo.speak("De volgende oefening is voor de
bloedsomloop.");
                    perStageCounter++;
                } else if (perStageCounter == 7){
                    speakTo.speak("Ga weer rechtop zitten met uw armen langs uw
zij");
                    perStageCounter++;
                } else if (perStageCounter == 8){
                    speakTo.speak("Strek uw armen recht voor u uit");
                    perStageCounter++;
                } else if (perStageCounter == 9){
                    speakTo.speak("En laat uw armen maar weer zakken");
                    perStageCounter++;
                } else if (perStageCounter == 10){
                    speakTo.speak("Armen gestrekt voor u uit");
                    perStageCounter++;
                } else if (perStageCounter == 11){
                    speakTo.speak("En weer laten zakken");
                    perStageCounter++;
                } else if (perStageCounter == 12){
                    speakTo.speak("Armen gestrekt");
                    perStageCounter++;
                } else if (perStageCounter == 13) {
                    speakTo.speak("En u kunt weer ontspannen.");
                    perStageCounter = 0;
                    stageCounter++;
                }
            } else if (stageCounter == 6) {
                speakTo.speak("Is de oefening gelukt?");
                listenNow()
                stageCounter++;
```

```
		} else if (stageCounter == 8){
			flowerState = 2;
			serial.write("flower\n");

			speakTo.speak("Dan is het weer tijd voor een korte pauze. Over een paar
seconde beginnen we aan de laatste oefening.");
			stageCounter++;
		} else if (stageCounter == 9) {
			speakTo.speak("Is de oefening gelukt?");
			stageCounter++;
		} else if (stageCounter == 11) {
			speakTo.speak("U bent goed bezig geweest! De bloem bloeit weer in
volle glorie.");
			stageCounter++;
		} else if (stageCounter == 12) {
			// RESET PROGRAM
			serial.write("done\n");
		}
	}

	function listenNow(){
		if (!listenBoolean){
			listenBoolean = true;
			listenTo.start();
		}
	}

function afterListen(){
	listenBoolean = false;
}

	function getResult() {
		console.log("getResult");
```

```
            listenBoolean = false;
            var positiveString = "ja";
            var negativeString = "nee";
            var seeString = "zien";
            var resetString = "opnieuw";
            if (listenTo.resultValue == true) {
                if (listenTo.resultString == positiveString) {
                    // YES
                    recentWord = 1;
                } else if (listenTo.resultString == negativeString) {
                    // NO
                    recentWord = 2;
                } else if (listenTo.resultString == seeString){
                    console.log(stageCounter);
                } else if (listenTo.resultString == resetString){
                    resetProgram = true;
                }

                 else {
                    recentWord = 3;
                }

                console.log(listenTo.resultString);
            }
        }


        function printList(portList) {
// portList is an array of serial port names
for (var i = 0; i < portList.length; i++) {
// Display the list the console:
print(i + " " + portList[i]);
}
```

```
}

function serverConnected() {
  print('connected to server.');
}

function portOpen() {
  print('the serial port opened.')
}

function serialEvent() {
	inData = serial.read();
	console.log("Normal "+inData);
	console.log(inData == 83);

	if (inData == 83){
		console.log("inData == 83");
		resetProgram=true;
		inData = 0;
	} else if (inData == '83'){

		console.log("inData == '83'");
		resetProgram=true;
		inData = 0;
	} else {
		console.log("nothing happened")
	}
}

function serialError(err) {
  print('Something went wrong with the serial port. ' + err);
}
```

```
function portClose() {
  print('The serial port closed.');
}
```

```
//
//
//
//
//
//
//
// var myRec = new p5.SpeechRec(); // new P5.SpeechRec object
// var myVoice = new p5.Speech(); // new P5.Speech object
//
// myRec.continuous = true; // do continuous recognition
// myRec.interimResults = false; // allow partial recognition (faster, less accurate)
//
// var listenButton;
// var wrongReactionCounter = 0;
//
// var mostrecentword;
// var wordState;
//
// var currentTime;
// var startTime;
// var askPeriod = 1000;
//
// var speakIntervalStart;
// var speakIntervalPeriod = 10000;
```

```
        //
        // var checkStartTime;
        // var checkPeriod = 2000;
        //
        // var logExerciseStateStart;
        // var logExerciseStatePeriod = 10000;
        //
        // var startExercises = true;
        //
        // var exerciseState;
        //
        // function setup()
        // {
        //      exerciseState = 0;
        //      startTime = millis();
        //      checkStartTime = millis();
        //      logExerciseStateStart = millis();
        //      // graphics stuff:
        //      createCanvas(800, 400);
        //      background(255, 255, 255);
        // //   listenButton = createButton('Listen');
//      //      listenButton.position(width/2, height/3);
//      //      listenButton.mousePressed(doListen);
        //      fill(0, 0, 0, 255);
        //      // instructions:
        //      textSize(32);
        //      textAlign(CENTER);
        //      text("say something", width/2, height/2);
        //      myRec.onResult = showResult;
        //      myRec.start();
        // }
        //
        // function draw()
```

```
// {
//	currentTime=millis();
//
//	if (currentTime - logExerciseStateStart >= logExerciseStatePeriod){
//		console.log(exerciseState);
//
//		logExerciseStateStart = millis();
//	}
//
//	if (startExercises){
//		if (currentTime - checkStartTime >= checkPeriod){
//
//			doExercise();
//
//			checkStartTime = millis();
//		}
//	}
//	//systemAsks();
//	// why draw when you can talk?
// }
//
// function doExercise(){
//	console.log("Word State: "+wordState);
//	if (exerciseState%2 == 0){
//		wordState = 0;
//		myRec.resultString = "";
//	} else if (exerciseState%2 == 1) {
//		if (currentTime - speakIntervalStart >= speakIntervalPeriod){
//			if (listenToPerson() == 1){
//				exerciseState++;
//			} else if (listenToPerson() == 2){
//				// oops person said no
//			} else if (listenToPerson() == 3){
```

```
//				// Something else is said
//				exerciseState--;
//			} else if (listenToPerson() == 0){
//				// Nothing is said
//
//			}
//			speakIntervalStart = millis();
//		}
//	}
//
//	if (exerciseState == 0) {
//		myVoice.speak("Hallo mevrouw, doet u mee met de oefeningen?");
//		mostrecentword = "";
//		exerciseState++;
//		wordState = 0;
//		speakIntervalStart = millis();
//	} else if (exerciseState == 2 ) {
//		myVoice.speak("Fijn. Ga rechtop zitten in de stoel.");
//
//		myVoice.speak("Zet uw voeten plat op de grond. Til langzaam de tenen
omhoog zodat alleen de hak van uw voet de grond raakt.");
//		delay(5000);
//		myVoice.speak("Nu rolt u langzaam uw voet naar voren totdat alleen uw
tenen de grond aanraken.")
//
//		mostrecentword = "";
//		exerciseState++;
//		wordState = 0;
//		speakIntervalStart = millis();
//	} else if (exerciseState == 4 ) {
//		myVoice.speak("Dan gaan we door naar oefening twee blablabla. Zullen
we naar de laatste oefening gaan?");
//		mostrecentword = "";
```

```
//			exerciseState++;
//			wordState = 0;
//			speakIntervalStart = millis();
//		} else if (exerciseState == 6 ) {
//			myVoice.speak("En nu zijn we bij de laatste oefening op en neer
hoplakee klaar. Zie de bloem groot zijn.");
//			mostrecentword = "";
//			exerciseState++;
//			wordState = 0;
//			speakIntervalStart = millis();
//		}
// }
//
// function listenToPerson(){
//		return wordState;
// }
//
// function systemAsks(){
//		if (currentTime - startTime >= askPeriod){
//			doListen();
//			startTime = millis();
//		}
// }
//
// function doListen()
// {
//		myRec.start();
// }
//
// function showResult()
// {
//		if(myRec.resultValue==true) {
//			var positiveString = "ja";
```

```
//                  var negativeString = "nee";
//                  if(positiveString == myRec.resultString) { // SAID YES
//                            background(192,255,192); // Green
//                            wordState = 1;
//                            // wrongReactionCounter = 0;
//                            // exerciseState++;
//                            // background(255, 192, 192); // Red
//                  } else if(negativeString == myRec.resultString) {
//                                      background(255, 192, 192); // Red
//                                      wordState = 2;
//                                      // wrongReactionCounter = 0;
//                                     // background(192,255,192); // Green
//                  } else {
//                            background(192,192,255); // blue
//                            wordState = 3;
//                            // wrongReactionCounter++;
//                  }
//
//                  text(myRec.resultString, width/2, height/2);
//                  console.log(myRec.resultString);
//        }
// }
//
// function parseResult()
// {
//        // recognition system will often append words into phrases.
//        // so hack here is to only use the last word:
//  mostrecentword = myRec.resultString.split(' ').pop();
//
//  console.log("resultstring " + myRec.resultString);
//  console.log("testje: "+myRec.resultString.split(' ').pop());
//        if(mostrecentword.indexOf("ja")!==-1) {
//                  background(192,255,192); // Green
```

```
//			if (exerciseState%2 == 1){
//				exerciseState++;
//			}
//
//		} else if (mostrecentword.indexOf("nee")!==-1){
//			background(255, 192, 192); // Red
//		 // background(192,255,192); // Green
//	} else {
//		 background(192,192,255);
//	}
//	text(mostrecentword, width/2, height/2);
//		console.log(mostrecentword);
// }

</script>
</head>
<body>
</body>
</html>
```

Appendix C - Statement of Consent Form

Test administrator: ______________________

Participant: ______________________

Number: ______________________

The aim of this test is to evaluate the functional prototype that is used for improving self-management rehabilitation for older adults with a hip fracture.

During the test the system is observed to find potential shortcomings and improvements. The information gathered during the test session may be used in the reports. We will not include your name.

You are allowed to stop participation in the test at all times. There is a test administrator next to you if you have any questions.

Statement of Informed Consent

I hereby declare that I have read the description of the test and my rights as participant. I agree to voluntarily participate in the test.

Name: ______________________

Signature: ______________________

Date: ______________________

Appendix D - User test plan

Achtergrond

Ouderen die na een heupfractuur moeten revalideren komen vaak in een verpleeghuis terecht. De motivatie voor het revalideren is wel aanwezig, de patiënten willen zo snel mogelijk naar huis. Toch doet een groot deel van de patiënten geen oefeningen uit zichzelf, iets wat juist essentieel is voor de snelheid van herstel.

Dit systeem is ontwikkeld om bij zo'n patiënt op de kamer geïnstalleerd te worden met als doel om de ouderen te herinneren en te ondersteunen bij het doen van de oefeningen zonder dat daar een fysiotherapeut of een verpleegster bij aanwezig moet zijn.

Doelstelling

Met deze test probeer ik er achter te komen in hoeverre het systeem werkt zoals het zou moeten werken. Door het systeem te laten gebruiken komen alle functies aanbod en wordt zo duidelijk of alle taken uitgevoerd kunnen worden.Ook is te zien aan de interactie met het systeem of het op alle momenten duidelijk is wat er gevraagd wordt.

Doelgroep

Aangezien het doel is om te kijken of alle functies werken is het niet van belang om uitsluitend ouderen met een heupfractuur te includeren. Om betrouwbare resultaten te halen worden er minimaal 5 deelnemers gevraagd om het systeem te testen.

Taakomschrijving

Het systeem wordt twee keer doorgelopen. De deelnemer wordt bij de eerste keer geacht om inactief op de stoel te zitten totdat het systeem een herinnering stuurt. Daarna kan de gebruiker vrij reagereren op wat er gebeurt.

Tijdens de tweede doorloop moet de gebruiker de bloem laten verdwijnen en daarna de bloem weer terug laten komen door het systeem te gebruiken.

Data

De data die wordt verzameld tijdens de test

Aantal fouten in het systeem	
Aantal fouten tijdens spraakherkenning	
Aantal gestelde vragen	
Bloem verdwijnt zoals gewenst	
Bloem komt terug zoals gewenst	
Druksensor meet juiste waarden	
Eventuele opmerkingen	

Na het testen volgt een vragenlijst die de deelnemer mag invullen.

Locatie

De test zal plaatsvinden in een ruimte waar de persoon zich samen met de testafnemer bevindt. Hier staat een stoel en een tafel waarop de bloem geplaatst is.