Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt

2017•2018

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Ontwerp en implementatie van een revalidatie platform gebruik makende van de leap motion sensor

PROMOTOR :
De heer Thijs VANDENRYT
Prof. dr. ir. Ronald THOELEN

PROMOTOR :
prof.dr. Raf MEESEN

Ruben Debien, Jonas Vanstraelen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

2017•2018

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Ontwerp en implementatie van een revalidatie platform gebruik makende van de leap motion sensor

PROMOTOR :

De heer Thijs VANDENRYT

Prof. dr. ir. Ronald THOELEN

PROMOTOR :

prof.dr. Raf MEESEN

Ruben Debien, Jonas Vanstraelen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Preface

The last assignment before graduating as an industrial engineer, is writing a master's thesis, this combines 4 years of theoretical knowledge into one large project. We were provided with the opportunity to learn new things. Not only technical knowledge was gained but also working together as a team on a larger product. One of the more special parts of this project was the collaboration with other master's theses from the UHasselt department of Rehabilitation Sciences and Physiotherapy, which gave us a more realistic view of a real project as the "customer" was somewhat inexperienced.

The success of this project would not have been possible without the unrelenting support and vast experience of others. We would like to thank our supervisors, family and friends.

We would like to thank Prof. dr. Raf Meesen, our external promoter, who provided us with the opportunity to join in on this project and explained the different rehabilitation techniques and exercises. This information made it easier for us to get a deeper understanding of the subject we were working on.

Dr. ir. Ronald Thoelen and ing. Thijs Vandenryt, our internal promoters, who provided technical knowledge in certain specific subjects and helped steer the project in the right direction.

Marijn Lemmens, who provided us with the groundwork for our project and helped us get started with the platform.

And finally, we would like to thank the entire Biomedical device engineering group, led by Prof. dr. ir. Ronald Thoelen, for the opportunity to practice the defense of the master's thesis.

Table of contents

List of figures

Abstract

This thesis describes the making of a medical platform, which will be used by the research group of the university of Hasselt named REVAL. This platform comes in the form of a website, where accessibility is the most prevalent benefit. Through this website, patients won’t have to come to REVAL and researchers do not have to visit the patient to gather data. With this platform, a large datapool can be assembled to gather data of regular people to later use as a comparison for patients.

Several experiments are implemented on the platform and more can be implemented later. Two pursuit tasks, one with an accelerating target and another with a disappearing target, and a bimanual motor tracking task (BMTT) are already developed and can be customized by researchers to fit in their research. The former involves following a target, which moves in a circular trajectory, with the cursor and the latter uses two stepper motors and Mechaduinos as controllers. The goal of the BMTT is to replicate a linear movement in a random direction by twisting the stepper motors.

To further digitalize the work of a physiotherapist, the leap motion sensor was tested for its viability as sensor for medical research. The leap motion is a sensor designed to track hands and fingers which makes it ideal for a finger pinching detector. After testing, it turned out that the leap motion was not accurate enough to be reliable as a medical sensor.

Abstract Nederlands

Deze thesis beschrijft de ontwikkeling van een medisch platform ,dat gebruikt zal worden door een onderzoeksgroep van de universiteit Hasselt genaamd REVAL en komt in de vorm van een website, wat de toegankelijkheid vergroot. Patiënten kunnen dan testen thuis afnemen zonder de aanwezigheid van een onderzoeker. Met dit platform kan een grote hoeveelheid data verzameld worden van gezonde mensen zodat deze data later gebruikt kan worden om patiënten mee te vergelijken.

Op dit platform zijn al enkele experimenten uitgewerkt en meerdere kunnen later toegevoegd worden. Twee achtervolgings taken, een met een versnellend doel en een met een verdwijnend doel, en een bimanuele motorische achtervolgings taak (BMAT) zijn al ontwikkeld en kunnen aangepast worden door onderzoekers zodat de taken beter passen bij hun onderzoek. Een achtervolgings taak bestaat uit een doel, dat een cirkelvormige beweging maakt, dat de patiënt moet volgen met de muis. De BMAT maakt gebruik van stepper motors en mechaduinos als controllers. Met behulp van deze moet de patiënt een willekeurige lineaire beweging maken.

Om het werk van een kinesist verder te digitaliseren, is er onderzoek gedaan naar het gebruik van de leap motion sensor als een controller voor enkel experimenten. De leap motion is een sensor gespecialiseerd in het volgen van vingers en handen. Deze sensor zou ideaal zijn voor het detecteren van knijpbewegingen van de vinger. Na testen bleek dat de leap motion in zijn huidige toestand niet accuraat genoeg was en deze niet gebruikt kon worden voor zulke testen.

1 Introduction

1.1 Background

In a world where Big Data is so prevalent, data collection on a large scale has not been introduced into the sector of rehabilitation techniques. Currently most tasks are performed either at the practice of a physiotherapist or at the home of the patient. Providing a platform where researchers can acquire data indirectly from their patients, would enable the collection at a larger scale and with a centralized data pool. From this data pool, people’s performance could be compared to that of their respective age group. By comparing this data researchers can distinguish how different people’s brain behaves and ages.

This platform was commissioned by the research group REVAL of UHasselt university, which researches different techniques in different fields of rehabilitation research. This platform would enable them to connect with other research groups or individuals worldwide and also collaborate on different researches with them. REVAL was also interested in looking into rehabilitation using a game controller, the Leap Motion.

Besides a research platform REVAL wanted to experiment with new test equipment to perform certain experiments or to improve upon existing setups. In this paper, the viability of the Leap Motion sensor as a tool for measuring the dexterity of people, is tested.

This project is a continuation of a bachelor's thesis done last year by Sander Denorme, where a simple pursuit task could be performed, and data could be collected. But the platform lacked some fundamental features like creating different tasks with different parameters and the website could not be moderated.

1.2 Problem Definition

A platform had to be made where researchers could contact a multitude of people in different age groups and have them perform tasks with specific settings or parameters. From these tasks data would be stored and different parameters would be calculated, so that researchers can analyze it afterwards.

To create a system like this the researcher has to be able to create a research in which different experiments can be added, these experiments are different versions of a premade template where certain parameters can be changed. To simplify the data collection process, these tasks also have to be a part of the platform.

The tasks that needed to be implemented include: two more complicated versions of the pursuit task and a bimanual motor tracking. Two master’s theses from the department of Rehabilitation Sciences and Physiotherapy, required the platform as their data collection tool, in which they use the two pursuit tasks as experiments. The Leap Motion is a hand tracking device that was meant for 3D virtual reality, which could possibly be used as a sensor for different task that require specific hand movements.

1.3 Objectives

The main objective is to create a platform or website where researchers can log on, create researches and easily reach people without having direct contact with them. Admins should be able to moderate and authenticate these researches and researchers, this is to prevent unqualified users from becoming a researcher and to prevent duplicate or invalid researches. An invalid research can be one without a proper scientific basis or one with nonsensical parameter

Developing the tasks for the master’s theses had the highest priority because they had to start collecting data at the start of the second semester. Not only did these tasks have to be developed, the system also had to be deployed so that it could be used in the field.

The Bimanual Motor Tracking Task was a project that started as an eHealth project but was unfinished and had to be expanded upon and incorporated into the platform. This task provided a challenge because two USB devices have to be connected to the website.

The Leap Motion as game controller features a device that enables developers to detect certain hand and finger movements. The goal of the Leap Motion part of this project is to see if the controller could be used as a scientific sensor that returns representable data and is not flawed. By looking into tasks for which the Leap Motion could be used and testing it in different environments the limits of the Leap Motion could be measured and compared to existing sensors.

1.4 Method and Materials

The design and base of the website were originally part of a bachelor thesis, where the original platform was made by Marijn Lemmens, a PhD student of the research group IMO-IMOMEC. The platform featured a website made with Laravel and a data processing server made with the Spring Framework. The used frameworks did not change but the entire data structure and most of the services had to be revised to reach the objectives of this project.

1.5 Outline

Chapter 2
This chapter gives more technical insight into the technologies that were used to create the website.

Chapter 3
Explains how the platform works and contains some decisions that were made in its development.

Chapter 4
Focuses more on the different tasks that were implemented and the problems that were faced.

Chapter 5
Discusses the Leap Motion use for research purposes and its limitations.

Chapter 6
Finally, chapter 6 concludes the achievements of this master’s thesis.

2 Technical Background

This chapter provides a small overview of the frameworks, technologies and devices that were used to create the platform.

2.1 Laravel

Laravel is a PHP web application framework designed to make the job of a developer easier. This is done by simplifying the implementation of functionalities that are used in almost all applications. It contains the model-view-controller design pattern and is currently the most used PHP framework [1].

2.1.1 Authentication

With Laravel come several pre-built authentication controllers. These controllers take care of: user login and registration, password reset and handles e-mailing links to reset the password when it is forgotten. Besides these controllers, Laravel also gives the option to create basic login and registration views.

2.1.2 Blade

Blade is a template engine used by Laravel. One of the biggest strengths of blade compared to other PHP templating engines, is that it allows PHP to be used in the views. The blade view files have the blade.php extension meaning that all view files are compiled into PHP code. After compilation the view files are cached until they are modified which means that basically no overhead is added to the application.

Template inheritance and sections

Most website have a similar layout over multiple pages, for example: a top navigation bar, a sidebar and a content area in the middle. This layout can be made by having one main view file that contains several sections marked by @section and @yield tags. An @section tag defines a section of content and an @yield directive is used to display the contents of a section. This main view file acts as a parent file.

A child view can inherit the layout from the parent view with an @extends directive. The child can then modify the @yield directives from the parent view with its own @section directives. An advantage of the section and inheritance features from blade is added structure to the filesystem of the application. Rather than having multiple big files that all have a significant portion identical to each other, the blade template makes it possible to remove this redundancy and to improve readability of the code by splitting it into multiple smaller files that whom are specialized for one certain page.

Boilerplate code reduction

Besides the sections and inheritance features Blade is very effective in reducing boilerplate code and improving the readability of the code. Now it is easy to display or echo PHP variables into the HTML or JavaScript code. This can be achieved without opening a PHP tag but rather with double curly braces ({{ }}). On top of this it also adds security to XSS attacks or cross-site scripting attacks. Common control structures in PHP, such as if statements and for loops, can be called upon with ‘@’ before the structure, for example: @if and @endif.

2.1.3 Routing

Every Laravel project contains a routes folder with PHP files that handle routes to different webpages or to different API’s. In these route files new routes can be defined by specifying the type of HTTP request (POST, GET, etc) and then binding a URL extension to a function of a certain controller. If only a view needs to be returned a special view route can be used instead of a GET request to remove the need of adding a function to an existing controller or to add a totally new controller. Parameters can be added to routes, for example a user id to load a user specific file. These parameters can then be exposed to constraints if the application requires it.

In case certain routes share the same requirements considering middleware (see middleware in Security) or namespaces, they can be placed in the same route group.

2.1.4 Security

Encryption and Hashing

The framework has certain functions that can encrypt or hash data. The encrypter that Laravel supplies uses OpenSSL to provide AES-256 or AES-128 encryption. All the encrypted data is signed with a MAC or message authentication code to protect the message’s data integrity and authenticity.

User passwords are hashed via Bcrypt or Argon2 hashing, with Bcrypt as the default. Bcrypt includes a 128-bit salt which defends against dictionary attacks. Dictionary attacks is a technique of finding passwords by matching hashed data to hashes of words in for example a dictionary. By adding a salt, regular passwords that are known words, are randomized to make it impossible to find a match.

Middleware

Middleware adds another layer of security to the website in the form of an HTTP request filter. There are different types of middleware that act in different circumstances. A common form of middleware is authentication middleware. With this form HTTP requests can only be made by authenticated users, otherwise they are redirected to another webpage like the homepage for example. Middleware can restrict HTTP requests on a different level than authentication, combined with Zizaco entrust for example, middleware can restrict HTTP request based on roles and permissions. This makes it possible to restrict users without adequate rights, from performing sensitive requests.

Middleware can be assigned in different ways: globally, via routes or in groups. Global middleware run with every HTTP request while route middlewares are only done to requests to a certain URL. Middleware groups are several middlewares combined under one key. The web middleware group is assigned

CSRF protection

CSRF attacks or cross-site request forgery is a type of exploit used to make an authenticated user execute unwanted actions, for example: making a user change his/her email address via a ‘unsuspicious’ link that does not mention such intentions. To protect the user against these types of attacks, Laravel generates CSRF tokens for each session. This token verifies that the user is the one making the request. The verifyCsrfToken middleware automatically checks the validity of CSRF tokens.

2.1.5 Database

Query builder

Laravel provides a query builder which can perform most database operations, such as: insert, delete, select and update. The query builder makes use of PDO (PHP Data Objects) parameter bindings, a way of binding a parameter to a certain data type, which protects against SQL injection attacks. An SQL injection attack is a technique for tampering with databases by submitting malicious input in a form which then alternates the SQL statements that handles the form.

Eloquent ORM

The eloquent ORM or object-relational mapping is a strong feature that allows the developer to define relation between database objects or tables. Every database table corresponds with a model. The name, key and columns of the table are defined in these models. Eloquent will assume names when they are not specifically set in the model, for example: the table name will be the name of the model with an ‘s’ appended and all in lowercase and the primary key will be named ‘id’. Eloquent also automatically adds a ‘created_at’ and ‘updated_at’ column which stores the timestamp of the time the record was added and last updated respectively. This can be disabled but will have to be specified in the model.

Each eloquent model acts as a query builder that can be used to access the table it describes. All query builder functions work as functions of that model and will return all the records of the table that adhere to these constraints. Eloquent adds short functions that help in retrieving single records. The ‘find’ and ‘first’ function are then used instead of ‘all’ or ‘get’ which return multiple records.

Relations between tables come in 3 different forms: one to one, one to many and many to many. These relations have to be defined in both models for it to work correctly. These relationships are defined by functions that combine two models via their keys and foreign keys. These functions return objects of the linked records, for example: two tables, A and B, are linked, when you have an object of A you have access to all the columns belonging to it. Now if you want to find the contents of a certain column from table B you can simply get the object of B that belong to the object A by calling the relation function. Then you can get all contents of the columns from table B. Without eloquent you would first have to write a query that joins to tables based on the foreign key before being able to access the contents of table B.

2.2 Hibernate ORM

Hibernate is a framework made for java to provide an alternative to write queries to access an SQL database. The goal of Hibernate ORM [2] or Object Relational Mapping is to create persistence data. And persistence data means that objects that are created in java can be stored and accessed even after an application is shut down. And be used beyond the scope of the java virtual machine.

Object relational mapping enables a server to match an object model with a relational database. By configuring and linking the objects together, like in an Entity Relationship(ER) model, the objects can request data from each other and acquire additional information, without using SQL INNER JOINs to connect 2 tables of object with each other.

For example, there are multiple trees that hold different leaves. When this must be stored in the database, a table “trees” is made that stores information on the trees and the table “leaves” is added that stores the information about the leaves. Leaves only belong to a single tree but a tree can have multiple leaves. Java and SQL have different interpretations on how this is implemented. In java the Tree object will have a list of leaves objects. While a leaf will have a reference to a single tree object. In SQL these 2 tables are connected with what is called a one to many relationship, this relationship is created by adding a column to the leaf that contains the unique identifier of the tree. The difference in structure can be seen in Figure 1.

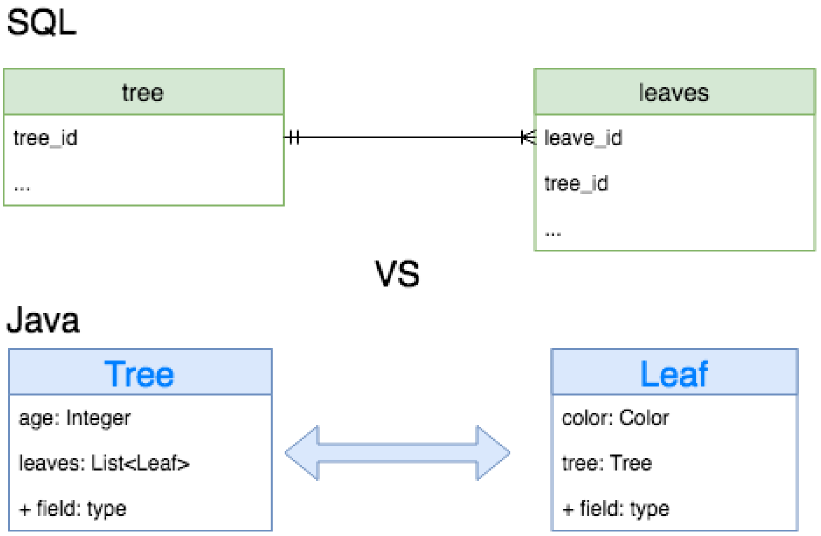


Figure 1: The different representations of the same object in java and SQL

The Hibernate ORM places these connections between 2 objects on top of each other and maps the java object to the ER object. Hibernate does not do this by itself, it must be configured. This is done by changing XML files to represent the connection between the java object and the databases table. When there are changes to be made to the database, these XML also must be changed accordingly. This allows the user to push new additions to the database without the need to change Java code.

When this connection is made, Hibernate or the Java server can interpret the database but there is still no structure to insert, update, remove or request different objects. These Functionalities are added through Business Objects(BO) and Data Access Objects(DAO). The DAO contains all the functionalities required to change the database with basic queries like insert, update, ... or more advanced queries that request data from the database. A BO represents a data client which means that it implements the DAO for a specific Java object of the application.

What makes hibernate stand out from other ORM frameworks is, that it enables fetching. Fetching makes it possible to request data from the database and not only get the object that was demanded but also, if needed, get the objects related to the requested object. Continuing with the tree and leaves example, for instance if a tree was requested, the leaves would, when asked, automatically be added to the leaves list and no additional querying is needed. This functionality greatly improves performance, if for instance multiple trees are needed, hibernate will put every single tree into a list and automatically insert for every tree all the leaves that belong to that tree. This is all requested in one query, which reduces the strain on the SQL server and Java application.

2.3 REST API

REST or representational state transfer API or application programming interface is a way for clients to request data from a server. It uses the basic HTTP methods of requesting pages, for example: POST and GET. The data is usually returned as JSON as it is the most used data format on the web and are instantly readable by PHP and JavaScript, which are the most used web development languages [3]. Other data formats are XML and PSON, but these are less common. REST API enables servers to return stored data back to clients in a specific data format, without the server having to store the data in that specific format.

2.4 Spring

The Spring Framework provides a developer with the ability to tie together different objects, without having to write an overarching module that manages all these connections. This module is made by spring instead and is configured by using XML files. The module is also responsible for managing the lifecycle of an object, which means: creating the object, calling the initialization functions and linking these objects together. These managed objects are also called beans and must be configured by either adding annotations to the java classes or by creating an XML file for each Bean or both can be applied as well, but this makes the programming less comprehensible and should be avoided.

Beans can be obtained by other java objects through utilizing either dependency lookup or dependency injection. In dependency lookup the Object directly demands the bean with a specific name from the overarching spring module. With dependency injection the module provides the beans to the object that require them, either in a constructor method or in a factory method.

Spring also provides a Model View Controller (MVC) solution to create a request-based framework. This solution introduces different interfaces that each have their own task. These interfaces are all separated from each other and have their own tasks to perform, when a request is sent to the java application it is first handled by a front controller which in this case is the DispatcherServlet. This controller hands the control to certain interfaces during HTTP request’s executions. Each of these interfaces performs their separate task and are implemented by default after the installation of the Spring framework into the Application.

Spring boot is spring’s solution to create a stand-alone application that can be run without the need of configuring every detail. It generally makes the life of the developer easier by automatically generating the base structure for a project and automatically starting tomcat server instead of having to deploy a WAR file. Tomcat is a server container that runs in a shell environment. A WAR or Web Application Resource file is a distributed collection of different components that are used in web application, such as java classes, JAR files, XML files, etc.

2.5 RethinkDB

RethinkDB is a scalable JSON database that works differently from normal SQL databases. Instead of having to poll the data, which can still be done, RethinkDB allows an application to constantly receive data in real-time. The advantage of using a JSON database instead of using an SQL database is that data can be stored that has the same overarching structure, but the underlying structure does not have to be the same. This enables developers to think outside of the box and create applications that are more flexible in data storage.

2.6 Mechaduino

2.6.1 Hardware

The Mechaduino contains several chips that enables it to control and measure the position of the stepper motor:

- A4954,
- AS5047D,
- ARM M0+ SAM D21,

The A4954 chip is a dual H-bridge, that enables the Mechaduino to control the stepper motor. This A4954 also enables the use of micro-stepping which is important for the accurate control of the stepper motor and in turn the position of the controller.

The angle of the axis is measured by the AS5047D chip which is a 14-bit magnetic rotary position sensor. By placing a magnet on the back of the stepper motor axis and centrally aligning the sensor with this magnet, the angle of the magnetic field can be measured. This enables the sensor to determine the angle of the axis, however the Mechaduino requires the specific angle of the axis so this sensor needs to be calibrated at purchase.

The ARM M0+ SAM D21 chip functions as the main controller. The Arduino-zero bootloader is installed by default, which means that it can be easily configured as an Arduino. A basic program is also provided by the creators of the Mechaduino which sets up the basic connection to the other chips on the board. It provides the developer with a basic controller which can be expanded when necessary.

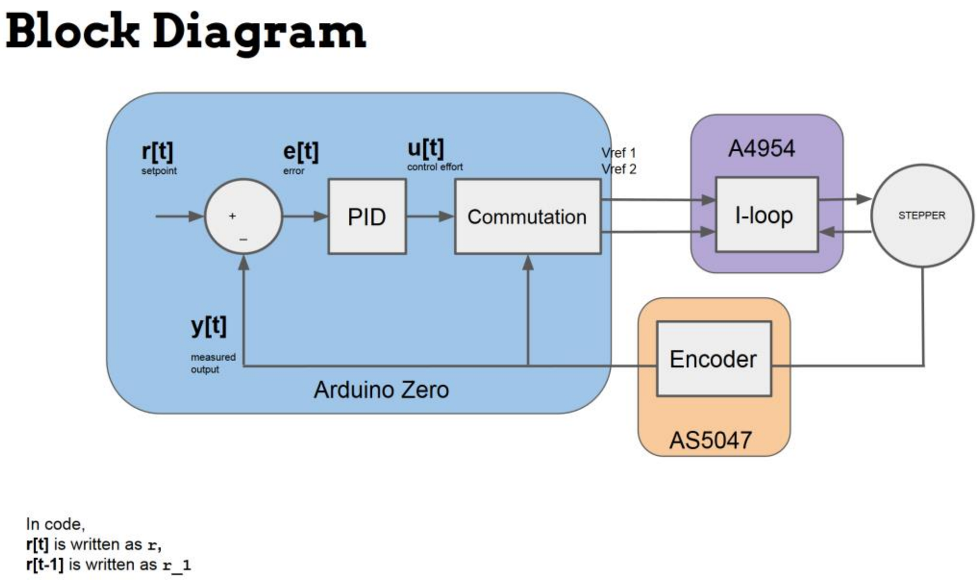
2.6.2 Software

The software as mentioned is written in the Arduino IDE and therefore uses C++ as the base programing language. Because the base program provided by the Mechaduino-team was inadequate, changes had to be made to facilitate all the required functions of the controller.

The base controller is equipped with 4 different modes of operation:

- position,
- velocity,
- torque,
- non-feedback,

The position and velocity modes use PID controllers to change the motor angle to the setpoint value. The values of these PID controller ideally must be configured per Mechaduino because the stepper motor properties are not always identical between each Mechaduino.
With the position mode enabled, the Mechaduino does everything in its power to go to a specific angle. This angle can exceed 360 degrees, but the controller counts the rotations that have been made and works out the correct angle. This mode provides a strong counteractive force and makes it very difficult to move the disk.



In code,
r[t] is written as `r`,
r[t-1] is written as `r_1`

Figure 2: Explanation of Closed loop block diagram [4]

Torque mode applies a constant torque to the stepper. This is done by applying a constant current to the coils of the stepper motor. This current is dependent on the setpoint given by the user. The base torque mode also keeps turning until the force applied to stop the stepper motor is greater than the setpoint.

The non-feedback mode does not use the encoder but uses the base functionality of the stepper motor. This mode is mainly used for testing and calibration purposes.

The encoder requires some reference point to return an accurate angle of the current position of the stepper motor. This reference point is given by a lookup-table that contains the angle for each possible value given by the encoder. Because the AS5047D is 14-bit, the size of the table must be 16384 float values which is 65536 bytes. That makes this table too large to fit into the SRAM of the SAM D21 which is only 32KB. So, this value is saved as a constant on the flash memory and is therefore unchangeable by the program.

To calibrate the encoder the stepper does a full rotation, 400 steps, and saves the encoder readings for these 400 angles and then starts interpolating between these values to print the lookup-table into the serial monitor. The user mustg copy and paste this table in to the code.

3 Platform

This master's thesis is a continuation of a bachelor thesis that was made last year and included a platform with a single task: the pursuit task. The platform was changed to add new functionalities and add different kinds of users. These changes are more prominent in the backbone and in the database as a multitude of tables were added. The original backbone featured 2 different databases: an SQL database and a NoSQL database RethinkDB. The SQL database was used to create structure between the individual data elements and RethinkDB was used to store the data. The frontend was made with Laravel and the backend with Spring Boot.

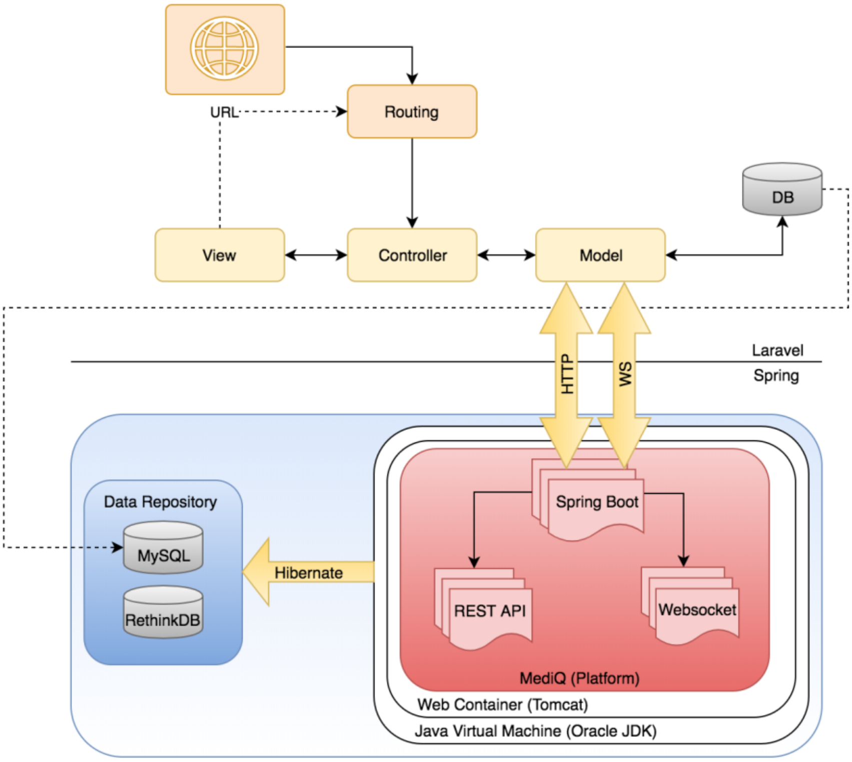


Figure 3: The structure of the platform

Figure 3 shows the structure of the platform. The upper part of the figure shows the structure of a basic Laravel project with the model-view-controller structure. The model connects with the MySQL-database through eloquent. When the site must connect to the spring platform, it uses either REST API for requesting data results and webSockets to push data.

3.1 Overview

Figure 4 shows the structure of how the website is currently implemented. The platform is centered around a research, which is a combination of experiments and when experiments are performed, measurements are created and linked to the experiments that they originate from. Different user groups have been defined so that users can be given different permissions and restrict access to different pages on the site. There are 3 different user groups: Admin, researcher and patient. Patients can join experiments and can perform experiments which are then saved as measurements. Researchers can create a research and define experiments with different parameters within this research. They can also view and download the measurements of their respective researches. Admin safeguard the platform and deny access to uncertified researchers and must approve a research before it is made accessible to patients and fellow researchers. Figure 5 shows the entity-relationship diagram implementation of the database; certain design decisions will be explained further on and a bigger version of the diagram will also be included as an attachment.

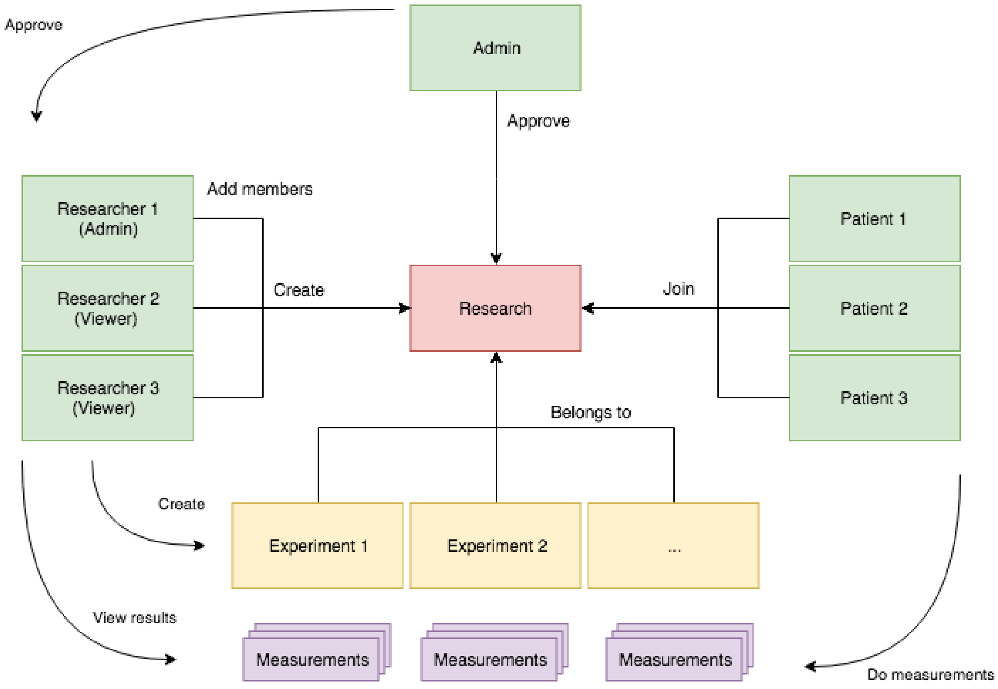


Figure 4: The overview of the hierarchy of the website

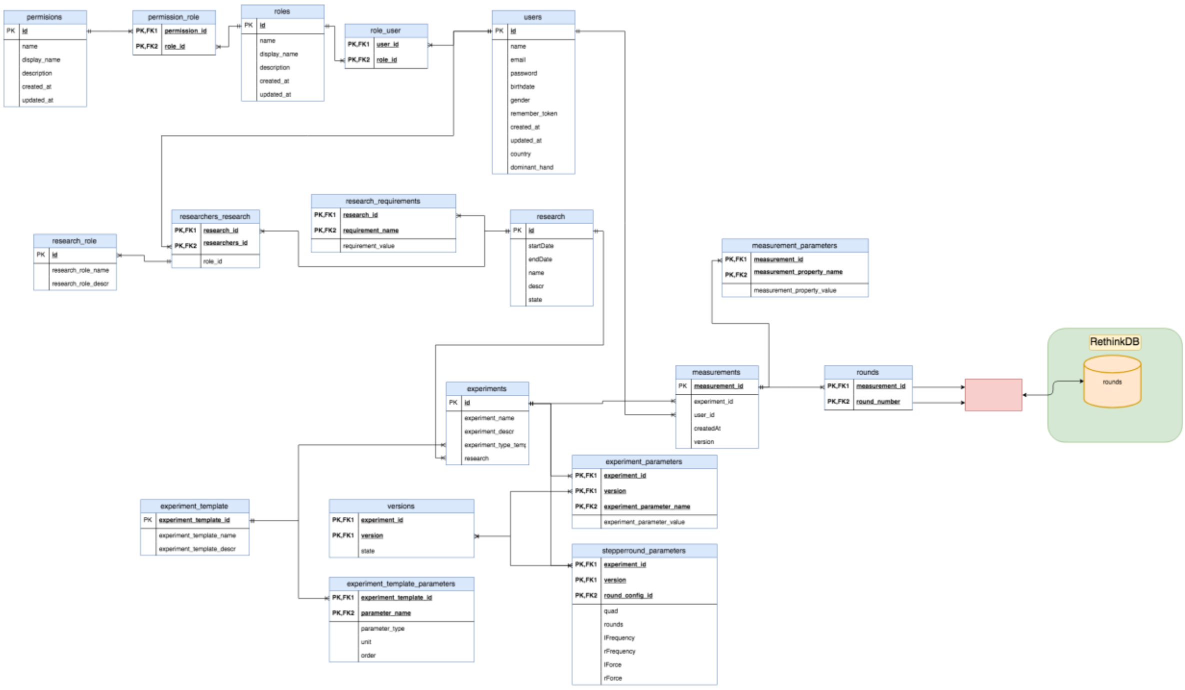


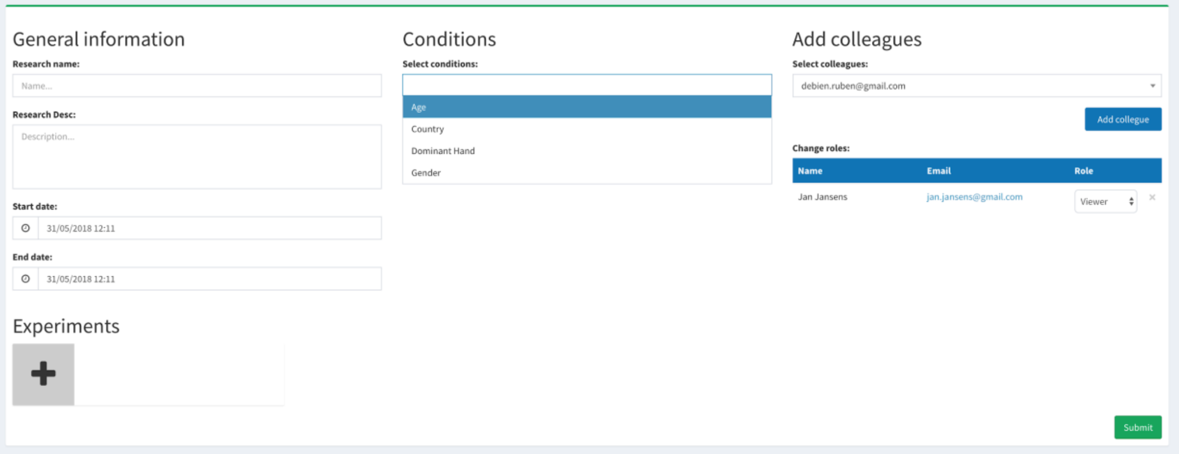
Figure 5: Entity-relationship diagram of the database

3.2 Research

A research is what binds different experiments together but also allows researchers to connect with patients. Researchers cannot invite patients to join their research the initiative has to come from the patient, this is to prevent spam requests.

A research is created on the bottom of the research page. Figure 6 shows the HTML-element, where researches are created. There are four main parameter groups: general information, conditions, colleagues and experiments. General information includes the name, description, end and start date of a research. The conditions allow the researcher to control which users can join a research. The different criteria are: age, home country, the dominant hand and their gender. Experiments can also be added at the creation of the research.

To enable collaboration in a research, other researchers can be added to a research. These researchers can each be given a specific role within a research, of which there are currently two: admin and viewer. Admins within a research can change parameters of experiments and change the general information of a research, such as the name and the end date. The roles can later be changed by an admin user and users could also be removed from the research by an admin. Viewers of a research can only view the measurements of each experiment.


General information

Research name: Name...

Research Desc: Description...

Start date: 31/05/2018 12:11

End date: 31/05/2018 12:11

Experiments

Conditions

Select conditions:

- Age
- Country
- Dominant Hand
- Gender

Add colleagues

Select colleagues: debien.ruben@gmail.com

Add collegue

Change roles:

Name	Email	Role
Jan Jansens	jan.jansens@gmail.com	Viewer

Submit

Figure 6: Element that allows researchers to create researches

3.3 Experiments

Experiments are different versions of a specific task, where the parameters are chosen by the researchers. A single experiment only belongs to a single research, this is to keep the data of the experiment within its designated research. Each experiment must be validated by an admin from the platform. The three different tasks each have different parameters, they are, however, still stored within the same SQL- table. By doing so the platform becomes more scalable. Otherwise when a new task must be added to the website, it would require that a table for that specific task had to be made. This is implemented by adding a separate table, ”experiments_parameters”, where the parameters that are different for each task are stored. The parameters that are similar for each task are stored in the “experiments” table. One issue that was encountered with this method is that when a task requires complex parameters, like the “round config” which will be discussed in another chapter, the table does not suffice. In this task there are multiple rounds and each of them has a specific set of parameters per round. To store this data in the experiments parameters table, a compromise had to be made.

By saving the data as JSON it could be stored as a string in the table itself, doing so would only have one benefit being that it would be easier to implement and read. There are, however, some serious downsides to this method. The first being that the table would load slower because when a query is executed, and the parameter value is requested in this query, it must go through every single byte of string characters. The second downside of this method is that the size of this JSON is unknown and could be much larger than was originally anticipated, however, the size of a field in an SQL-database table is set at the creation of the table and cannot be changed on the spot by the platform to accommodate the bigger parameter. Changing the size of the parameter value field, would also mean that the other parameters that are not as big would take up more space, for example: the round config, would also take up more space and would take up useless disk space.

The JSON could be directly saved outside of the table which would solve the first problem of not having to query through the table, but it then would be slower to query these parameters, because they are not stored in the table format, the table merely contains a reference to this JSON object.

So instead a table was created that stores this information for the specific tasks, it is not the most elegant solution, but it is the more efficient method in terms of storage capacity and data query speed. The ER-schema of these tables can be seen on figure 7.

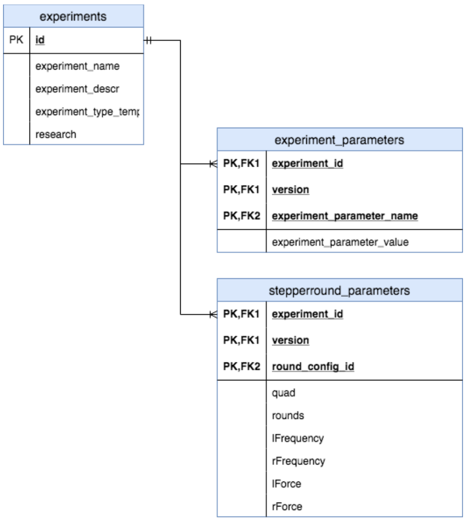


Figure 7: Entity-relationship diagram implementation for experiments

To add a functionality where researchers can change the parameters of an experiment, when this happens, the data that was previously collected for this experiment has essentially become unusable and must be deleted. To prevent this loss of data a version system was developed, where each of the parameters have a version tag and each experiment has a version table where the version id is stored and if it has been accepted or not by the admin of the platform. So, when a researcher wants to view the data from a previous accepted version he can by simply going to the research page where the previous versions are indicated.

3.4 Measurements

When an experiment is performed, a measurement is created. This measurement is stored in the “measurements” table and can contain some parameters which are stored in the “measurement_parameters”. These tables only store the Metadata and not the actual datasets. The datasets are not stored in an SQL-table for the same reason as with experiments, JSON storage is inadvisable in SQL, certainly because the datasets can become quite large. To store the data of a measurement RethinkDB was used, which makes it possible to store JSON objects.

To create a connection between the SQL-database and the RethinkDB database, a separate SQL table was used. Because most experiments include different rounds where a subject must do the task multiple times. This table includes the measurement identifier and the round number and can be used by the hibernate to request data in the backbone and eloquent to generate the front end. This structure is shown in figure 8.

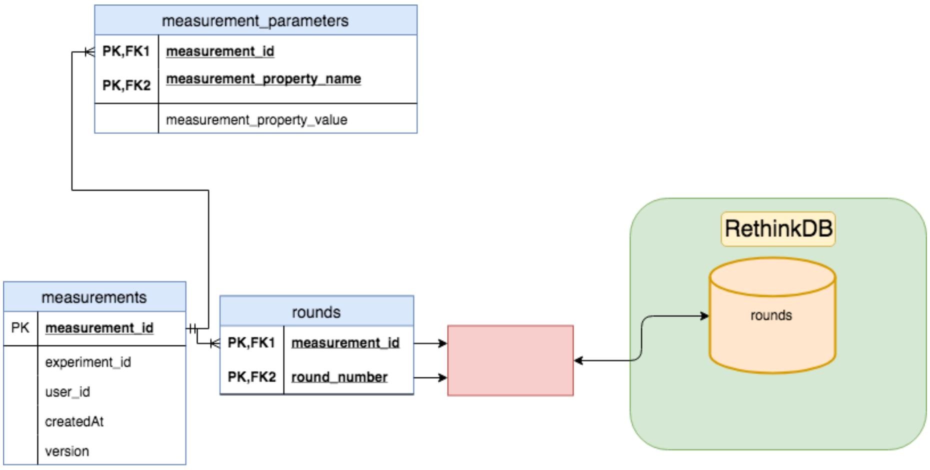


Figure 8: The RethinkDB connection in the entity-relationship diagram

When data is requested from the server, it polls the data from the measurement round per round and calculates the required information, which is different for each task. This data is then sent to the frontend using a REST response. A more optimal solution would be to store the calculated information alongside the original recorded datasets when the information is first requested. Any subsequent request for that specific measurement would just have to read the data from the RethinkDB and would require less processing power, however, this would generally double the required storage capacity for storing one measurement.

3.5 User Roles and permissions

By defining roles and permissions, different types of users can be created. The Zizaco entrust Laravel plugin enables this functionality. It provides the tables that must be added to the database and also provides middleware to restrict access to certain pages for specific user groups. It provides user with functions to check for roles or permissions they might have. There are 3 different user groups: admin, researcher and patient. The permission system works like a hierarchy, a user group inherits the permissions from the group below it. The researcher can do everything a patient can do but can also perform additional actions.

3.5.1 Patient

The patient’s homepage shows him the researches, he participated in and the ones he could join. For each subscribed research, the patient can see the duration and the description of the research. Underneath this information each of the experiments linked to the research are also displayed with links to their respective data collection page. Figure 9 indicates what a single research might look like on a patient’s homepage.

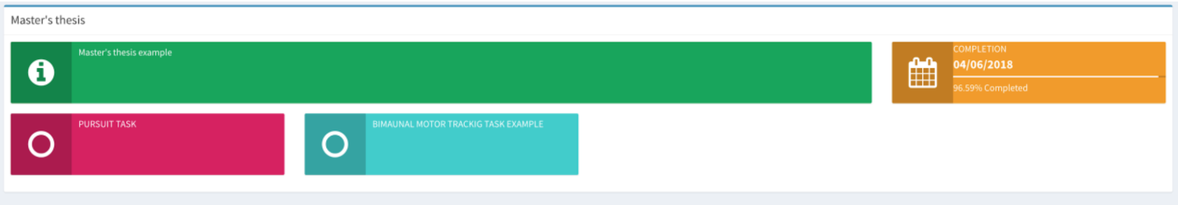


Figure 9: Research information on homepage for patients

On the top of the homepage the patient can see the different researches he could join. These researches are presented in a scrolling bar ordered by expiring date. Ongoing researches are prioritized. On this scrollbar the patient can then directly subscribe to the research or click the info button where he gets redirected to the “research info” page. This is where all the available researches are listed for the current patient. The info looks the same as on figure 9.

3.5.2 Researchers

Researchers have the same homepage as a patient where they can participate in researches. Under the scrolling bar, researchers can find a link to the research page. This page allows the researcher to create and manage researches. To join the platform as a researcher a document must be uploaded, that signifies that researcher is valid. When declined the researcher can retry to upload the document. Until the researcher is validated, he can log on to the platform but will only have the rights or permissions of a patient.

3.5.3 Admins

Admins are researchers with more control. They can see every research that is currently being conducted on the platform. They are also the gatekeepers of the platform, every researcher that wants to join must have his document reviewed by the admins and every research and experiment has to be validated by admins. Other than these extra rights, the admin has the same permissions as a researcher, so besides being able to view every research, they can also create their own researches.

3.6 Plugins and extensions

3.6.1 Zizaco entrust

Zizaco entrust was used to provide permissions and roles in the platform. It also adds blade template function to generate web pages more easily. And it also adds middleware that enables the restriction of certain users to a specific page.

3.6.2 Hashids

To perform a specific experiment, a page must be loaded. This page is requested with a GET HTTP request, that means that in the URL the request is visible and so is the identifier of the experiment. Showing the identifier would show the users of the website the structure of the database and this information could be used to damage the database. The alternative would have been to request the page with POST, so that the request would not be visible in the URL. But when the user reloads that page, he would get a popup that notifies him that he is resending a POST request which is annoying. That is why the Laravel plugin Hashid was used, it turns a string or number in to a string with a specified length. Normally this is called hashing, hence the name Hashid, but it is not true hashing as the original value can still be recalculated. It uses a salt with which the original value can be recalculated. The same principle is used in multiple sites to mask data identification, for example: youtube’s video link.

3.6.3 JsRender

JsRender is a JavaScript plugin that has the same functionality as the blade templates in Laravel. It allows developer to generate HTML structures from templates, using JSON data structures as input for the functions. The alternative would be to generate string from JSON objects, but this would take much longer and is extremely taxing for the local pc as it must process more elements. Whereas JsRender loads these templates at page load and only the missing JSON elements must be filled in, which is less demanding for the local pc.

3.6.4 Graph.js

The data review pages on the platform include some graphs where researchers can see some general information on the performed measurements or more specific information on a single measurement. These graphs were created using the Graph.js which is a JavaScript plugin that uses generic HTML canvases and turns them into graphs. These graphs can be configured with color and specific datasets, which makes it easier to compare data

3.6.5 Beautification extensions and theme

The main theme of the website was AdminLTE, which is a bootstrap theme that has been configured for easy web development. The following scripts were used to create a more streamlined and appealing platform:

- Slimscroll,
- Smoothscroll,
- Pace,
- Select2,

3.7 Master’s theses Rehabilitation Sciences and Physiotherapy

As part of this master’s thesis, the platform had to be deployed and be usable at the end of the first semester. The platform was to be used as data collection software and was going to be tested in the field. Due to time constraints, we implemented a much more simplified version of the platform. The students did not need to create new researches and experiments, the student groups each needed a single experiment which were both variations of the pursuit task that will be explained later. Each test subject was only going to be examined once, so types of users were put aside, and all users were made researchers. To add patients to the measurements, a special table was added where all the data on the test subjects was saved. This data included the initials of the test subject, the date of birth, the gender and the dominant hand.

4 Tasks

4.1 Pursuit tasks

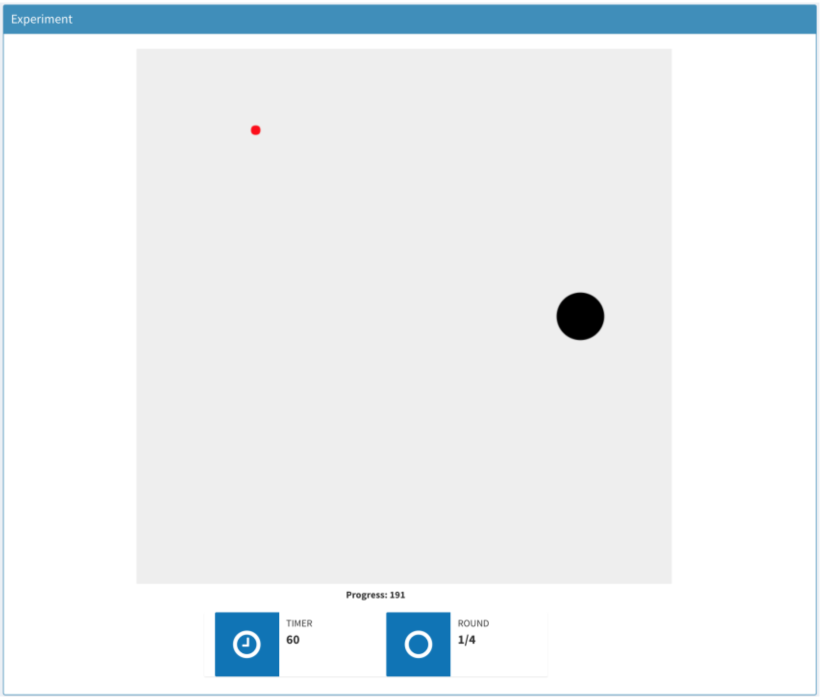
4.1.1 Introduction

A pursuit task involves the user following a target and copying its movement as close as possible. There are two types of pursuit tasks implemented on the platform named: acceleration and disappearing. Acceleration involves an accelerating target and disappearing involves the target turning invisible for a couple of seconds and then reappearing. Both are based on an original design made by Sander Denorme in his bachelor thesis.

4.1.2 Purpose

Master students of rehabilitation sciences and physiology incorporate pursuit tasks in their master’s theses. The basic concept of the pursuit task is that a person controls a circle (user object) with a mouse and must follow or track another circle (target object). This target object moves in a circular trajectory for the duration of a round which is 60 seconds. There are 2 versions of this test, one with an acceleration of the target object and another where the target object disappears for certain time intervals. The goal of the former is to test children in their ability to use predictive motor control. In other words, being able to accurately predict the location of the target object knowing that its speed will increase. The latter will test the ability of children to learn and become better in using their feedforward control in these visuomanual tasks [5].

4.1.3 Specification



Progress: 191

TIMER 60

ROUND 1/4

Figure 10: User interface Pursuit Tasks

Both pursuit tasks consist of 2 objects: the target object (black circle) and the cursor (red circle), with the position of their respective centers as important data. This data is sent to the backbone every 0.1 seconds in packages of 10 samples resulting in a sample frequency of 100 Hertz (Hz). Out of this data the distance between the cursor and target object can be calculated per frame or sample. This in turn makes it possible to find other parameters such as: the amount of times the cursor is outside of the target; the speed of the cursor and the average distance between cursor and target.

The size of the cursor, target object, the radius of its trajectory and the angular speed of the target are all template parameters that can be defined when a new experiment is created. This way different levels of difficulty can be emulated to test different age groups for example.

4.1.4 Problems

The original pursuit task, implemented by Sander Denorme, updated the canvas using intervals. Intervals are JavaScript functions that periodically call upon other JavaScript functions after a predefined time interval has passed. The problem that arose with this approach is that the drawing of the circle becomes very shaky. The problem was caused by the interval function, which is not prioritized by the browser. This meant that the redrawing of the circle didn't always happen at the exact time the interval ran out resulting in an irregular refresh rate of the canvas. The position of the target object is incrementally determined meaning that after every interval the previous position is incremented by a certain value depending on the angular velocity of the target object. If the interval time differs from what was expected it means that fewer intervals happen during the length of the round resulting in fewer position increments. This means that the target objects end position was always different for every round and that there would be fewer samples than there should be with 100 Hz as sample frequency.

To resolve this, animation frames were used [6],which are handled by the browser and are called with a high priority. The browser will always try to ensure these functions are updated at the frame rate of the computer screen and produce a smooth movement. The faulty position is not solved by this functionality and has been resolved by using the Unix Timestamp. At the start of the test, the application requests the number of milliseconds since the first of January 1970. Then at each call of the "updateCanvas" function the Timestamp is requested again and compared to the one at the start. Knowing the angular velocity of the target, it is possible to calculate the position of the target out of the elapsed time since the round started. Another advantage of this approach is when the user exits and reenters the browser tab, the circle will immediately move to the appropriate position.

4.1.5 Deployment

For the research of the master students, for whom these tasks were initially designed, certain parameters had to be specified for it to be a valid research. The size of the target and cursor and the trajectory radius of the latter are all fixed parameters specified in centimeters. The size of objects on a computer screen are described in pixels. To match pixels to centimeters, the number of pixels per centimeter had to be found/calculated. This was achieved by adding a new settings page shown in Figure 11.

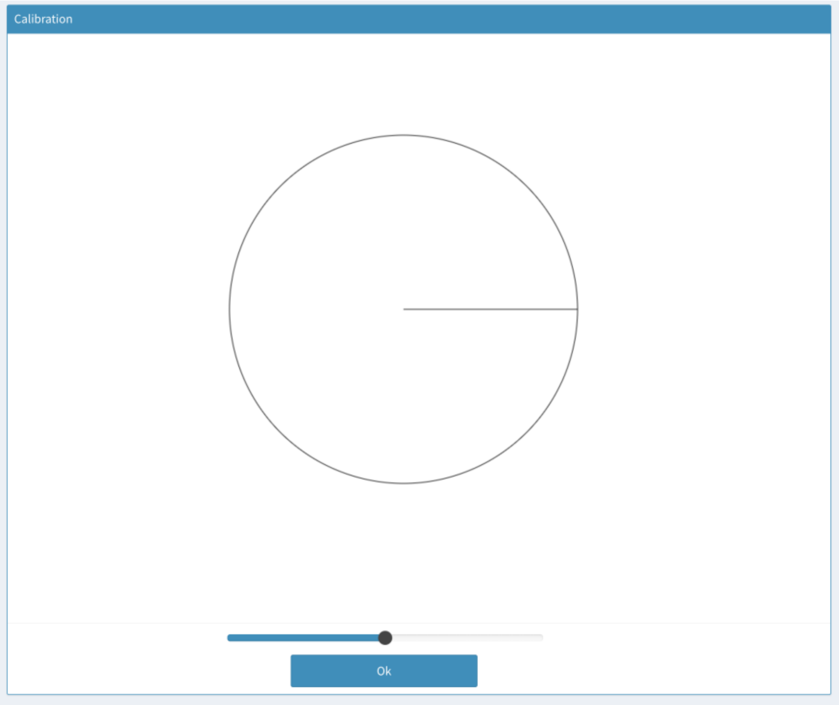


Figure 11: Size adjustment page where user measures the screen

As seen in the picture above, the new settings page contains a circle which emulates the trajectory of the target. The radius of this circle can be adjusted with the slider. Users then must physically measure the radius of the circle or the length of the line. When the line is 5cm long the user must press ok to confirm. This requirement was first set by the master students as they wanted a trajectory of that size.

The number of pixels can then be calculated by dividing the pixel radius of the circle by 5cm. This value is stored as a cookie, so the user only has to do this once. When the cookie is not set on a device, the user is automatically redirected to the configuration page before he/she can perform the task.

4.1.6 Backbone changes

The research done with these tasks was meant for children between the age of 6 and 11. The master students contacted schools to find test subjects for their research. To speed up the data acquisition phase, some changes were made to the backbone.

The first of these changes was removing admin users and allowing regular users to view results from tasks. This made it possible for the master students to review their data without having to create an admin account.

Besides this another table was made in the database named test subjects. This table stored all the personal data of the children that did these tests. Personal data contains: First and last name, birth date, gender and the dominant hand of the user. The benefit of having test subjects was removing the need to create accounts for every child that had to undergo these tests and so speeding up the succession between children. At the start of a new task a test subject could be made, or a previously registered test subject could be reselected.

Compared to the original design of the pursuit task, these tasks have more than one round. To resolve this the rounds table was added in the database. Here all the round ID’s and test ID’s are stored for each completed round for each measurement. The actual data from these rounds is stored in the RethinkDB database with a composite key of testID and roundId. So, in SQL all rounds per measurement are defined with a link to RethinkDB where all the actual data is stored.

The last big adaptation to the original experiment was in its sample frequency. The original had a sampling frequency of 10 Hz. To have more representative data, the data-rate was increased to 200Hz. But older computers could not keep up with this frequency, so it was later adjusted to 100Hz. The computers on the client side did not have any problems with this change of frequency but the server side of the application started to experience some crashes. By sending data in sets of 10 it was possible to reduce the number of packages received by the backbone. This did not solve the problem. Later a problem was discovered with the communication between frontend and backbone that originated from the previous design. Every time the frontend sent data to the backbone a new thread was created but never closed. This problem was detected with a MacBook Pro, were only 2542 threads [7] could be created for each process. The program would crash when it would try to exceed the maximum number of threads. This fatal defect was discovered during our project, as a result of the increase in sample frequency resulting in more threads being opened and because of the harsh restriction of the maximum amount of threads allowed by OSX. By ensuring that the threads were terminated after the backbone received the data it was resolved.

4.2 Bimanual motor tracking task

Students from the course E-Health [8] made a controller to help the Revalidation Research Center speed up the rehabilitation process of multiple sclerosis patients. The new controller seemed very useful not only for MS patients but also for people not suffering from this disease. So, the Revalidation Research Center saw an opportunity and wanted to add it to the platform.

4.2.1 Overview

The test subject must control a circle (user object) with 2 controllers. By turning the right controller, the user controls the X-movement and with the left the user controls the Y-moment of this object. The goal of the test is to follow another circle (target object), this target moves in a line at a random angle. The target object always moves a set distance of 360 pixels, but its speed is determined by the duration of one round. To give the test subject visual feedback a trail that follows the user object is added.

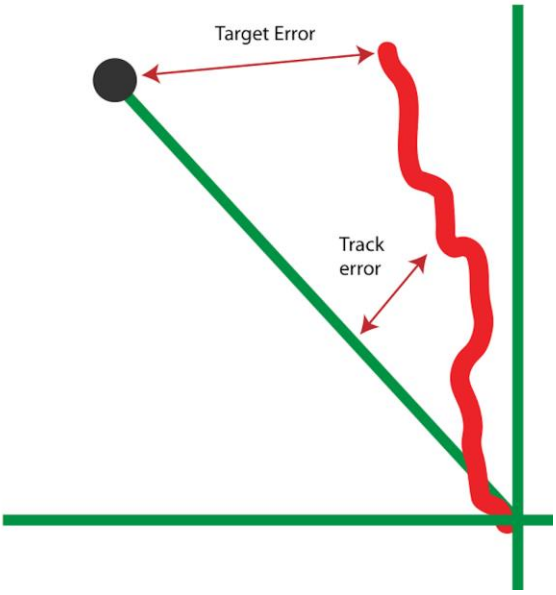


Figure 12: Shows the data that is calculated for the Bimanual Motor Tracking Task

The result of the experiment is a dataset of distances between the user object and target object throughout the measurement, this is called the target error. And the perpendicular distances from the target object track to the user object is the track error, this can also be viewed as the track that the test subject followed during the test. These 2 datasets can be calculated using the x and y position of both the user object and the target object. The visual representation of the results can be seen on figure 12.

4.2.2 Mechaduino

The main functionality of the controller is provided by the Mechaduino, which is an Arduino-Compatible board that fits perfectly on the back of a Nema 17 stepper motor. This Mechaduino provides a rotational feedback to the platform where the angle is transformed into a x or y axis movement of a target. The original design mounted the Mechaduino in a Plexiglas frame with changeable angles.

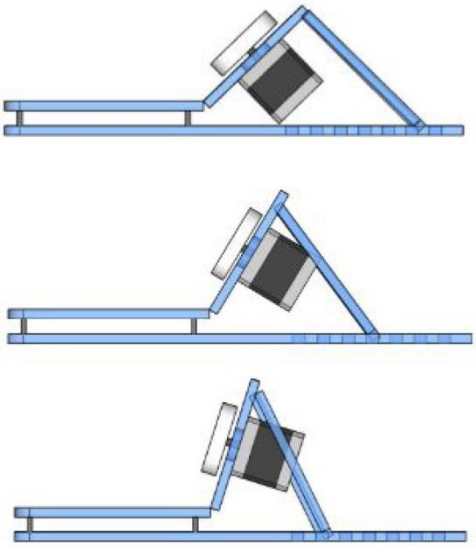


Figure 13: Design made during the eHealth project [8]

However, because the controllers were not connected and thus were 2 separate devices, they would move independently and influence the test. Another disadvantage are the hinges to make the angle adjustable, these cause the controller to feel shaky and unsteady.

To solve these problems, the angle was chosen to be fixed at 70° and the 2 controllers were put on a base plate. The distance between the controllers was chosen at 25cm, this is the distance that was used in the original tests. The distance between the controllers can be increased by an interval of 6 centimeters, this is to further increase the functionality of the test setup and to possibly make it easier for people with wider shoulders to do the test more easily. The maximum distance is now 51 centimeters. The design can be seen on figure 14 and includes raiseable armrests, that should make it more comfortable for users to perform the task.

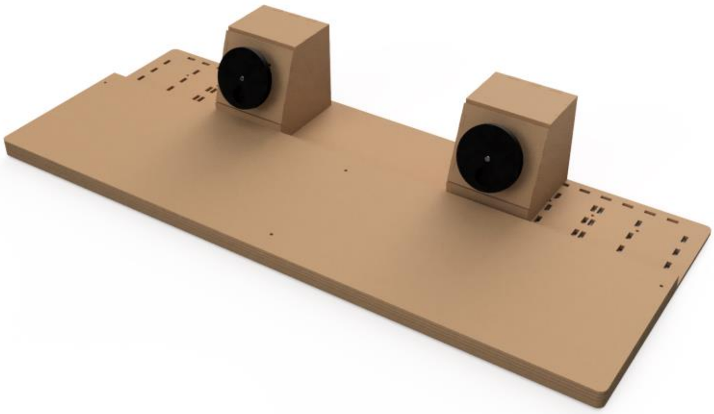


Figure 14: Current Design in MDF (Rendered in Fusion 360)

4.2.3 Changes to the Mechaduino

WebUSB

This controller will be used to control a target on a web browser, therefore we need a connection between the Mechaduino and a browser. By default, due to safety precautions a web browser such as Google Chrome or Firefox cannot access USB devices. Google, however, included an API in the Chrome 61 release, which is called "WebUSB". This API can be used to connect to several USB devices. It was originally designed to connect browsers to several peripheral devices such as printers and USB storage devices.

A special Arduino library was based on the USB serial library. The USB interface is not marked as USB CDC-ACM, which normally causes the operating system to automatically claim it, and does not allow the browser to easily access the Arduino. Instead it enables for a WebUSB-compatible interface with vendor-specific options. Vendor-specific selection enables the user to select for example only devices that were made by a specific manufacturer such as Arduino. This vendor system provides a filtering system to remove unwanted devices from the selection for the user.

Controller

The goal, to further expand the basic assignment to provide a opposite force to the user using the controller, had some setbacks. Torque mode seemed the best starting point, the first problem with this mode is that it keeps turning in a direction at a constant speed. By removing this change in angle, the stepper motor seemed to apply a constant force. But when the required torque became too large, the stepper motor automatically started returning to its zero-angle point of the encoder. This is undesirable and was probably caused by the small change in the calibration of the encoder.

By combining the position and torque mode, a system can be created that still counteracts the user’s movement and holds the disk in place when needed. The previously adjusted torque mode is used when the user is applying a force that is greater than the requested counterforce given by the user. The position mode is used when the disk is not being touched or the user is applying less force than the setpoint.

However, the position mode requires a PID controller to work properly. After experimenting with different values for the P, I and D components, the conclusion was drawn that the most stable and error free system was given by just a proportional or P controller, removing the Integral and Differential components.

4.2.4 Webpage

WebUSB

The computer side of this experiment is handled by JavaScript running in the browser, Google Chrome. Unlike other browsers, Google Chrome offers the opportunity to not use an external application to give USB devices a connection to the browser but does this directly. However, there are some security concerns that must be resolved before the API can be used. The restrictions given by Google Chrome are:

- user gesture,
- HTTPS,
- feature policy,

The function to request all the connected USB devices must be called by a user gesture, such as a button or any other input tag. This is to ensure that the browser cannot access unauthorized USB devices. When the user clicks the "add device" button the function `navigator.usb.requestDevice` is called, this function displays a list of valid USB devices at the top left of the browser. The user then gives the website permission to access to the USB device.

HTTPS is required because the peripheral device could be a recording device such as a camera or a microphone. HTTPS encrypts data that must be send back to the server, therefor making it more secure. The WebUSB feature is not enabled by default. Either the user must enable it manually or the developer can use an iframe to enable it on page load, for that page only. The WebUSB API is still an experimental feature so it is not enabled by default. Because WebUSB is vendor specific, USB devices other than the Mechaduino can be filtered out using the vendor ID 0x2341. After the connection is made and the port is setup. This setup includes claiming the interface and selecting the channels to be used.

User interface

The user is first asked to connect 2 devices to the website and then calibrate them by turning the right controller 180° in any direction. The Serial number of both the controllers is saved as a session variable so the configuration does not have to be redone at the start of each measurement. This session variable expires when the browser closes or after 3 hours of not using the website.

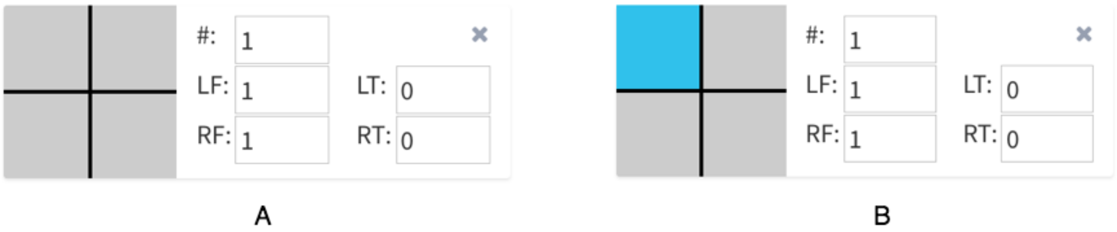
The measurement interface is based on the original model, it has identical color patterns and produces a sound when the user has to start each round of the measurement. The parameters that can be configured for this experiment are:

- audio feedback, the beep sound at start of round;
- duration, the duration of the experiment;
- reaction delay, the seconds of visual feedback;
- units per rotation, how many pixels the target moves with one rotation;
- short and long start time, a random delay before the round starts;
- config rounds, the configuration given for a set of rounds;

Config round is a parameter that describes how each round of an experiment is configured. In the UI component, a researcher can select the right and left frequency (LF and RF). Which is how many rotations the controllers must make to achieve the units per rotation. So, when this value is set to 2, the controller must be turned twice to move the target by the set number of units, specified in the units per rotation.

The right and left torque resistance are the settings that control the resistance of the motors. There are 10 levels where 0 is no resistance and 10 is the maximum resistance where a person can still actively move the disk. These values do not represent the actual force required to move the disk, they have no unit at all, so they are dimensionless numbers.

The number of rounds and the quadrant in which the line must be drawn can also be selected in this element. A working example can be seen on Figure 15 Many of these elements can be chained to form a series of tests in one measurement.



#: 1
LF: 1 LT: 0
RF: 1 RT: 0

#: 1
LF: 1 LT: 0
RF: 1 RT: 0

Figure 15: (A) Settings box with no parameters entered. (B) Quadrant selected and number of rounds entered

5 Leap Motion

5.1 Specifications

The Leap Motion is a sensor that is specifically designed to track hands and fingers. It is equipped with 2 cameras and 3 infrared LEDs. The interaction space is an 8 ft^3 which equals $0.277m^3$, inverted pyramid with a maximum range of 80 cm.

The Leap Motion takes pictures and its USB controller adjusts their resolution in the local memory and then streams them as grayscale stereo images (images of left and right camera) to the computer where it is received by the Leap Motion service. Here the image gets processed so that hands and fingers are distinguished from the background. Objects are created that relate to each other as shown in figure 16 and placed in a JSON array. These objects can be accessed via the LeapJS plugin. This plugin sets up a websocket connection on port 6347. Via this socket JSON messages are sent to the client.

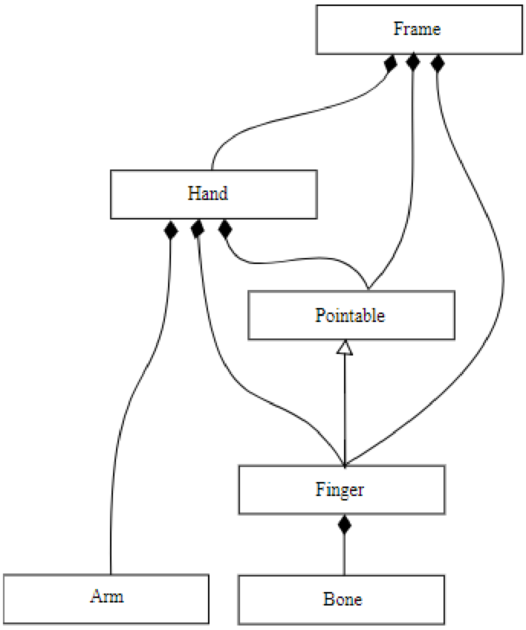


Figure 16: Diagram of relations of objects created by the leap motion

5.2 Purpose

To be able to test a patient's dexterity without having to send a physiotherapist, a new solution had to be found. The leap motion sensor with its finger tracking ability might be the solution. A new task that detects finger pinches (fingers touching the thumb) has to be created. The speed with which patients can do the right finger pinch in a random pattern, different for both hands, determines how good their brains functions.

5.3 Finger pinching measurement

Finger pinches can be detected by checking the position of all valid fingers of all valid hands in the frame. Sometimes hands or fingers are invalid if the leap motion was unable to find enough data of that object in the frame. The strength of a hand pinch can be measured by the leap motion and is returned as a number from 0 to 1. So that when a pinch is detected that is strong enough, the distance between all the fingers to the thumb is calculated with the shortest distance being the pinching finger if this distance was closer than a predetermined distance.

For example, the predetermined distance is 5 cm then the distance to all fingers is calculated and compared to the predetermined or ‘closest’ distance. If that distance was closer, then that distance would be the new closest distance. This predetermined closest distance could be used to fine tune the sensor. The greater the closest distance is the sooner a finger would be the closest thus making the sensor more sensitive.

The finger pinching task involves 8 circles, emulating all fingers except for the thumbs. When a circle turns green it means that the user should pinch with the corresponding finger. Both hands pinch in parallel and the pinching fingers varies between both hands.

5.4 Problems

Testing the viability of the leap motion as a finger pinch detector gave a negative result. Although the leap motion is an impressive sensor that works great in certain aspects such as using a finger as a cursor and recognizing gestures, it lacks accuracy in determining finger pinches. It has the ability to do this task but has a big chance of missing or misinterpreting finger pinches.

The distance between thumb and the pinching finger was always at least 15 mm according to the leap motion while in real-life they were touching. Even holding the pinch for a couple of seconds resulted in the same inaccuracy. This meant the predetermined closest distance that could be set had to be higher as to not miss all the pinches. The closest distance was set to 5 cm, but this makes it possible the cheat with these tests because your fingers only need to be approximately 5 cm apart for it to be a pinch.

Another problem was that the leap motion has only one viewpoint. Obscuring certain fingers from the camera by accident meant that the sensor could not determine where that finger was and so it would estimate or guess where it was. This would result in wrong determinations of pinches. This problem become prevalent when the little finger was the pinching finger as this finger often would obscure the ring finger. The leap motion would then assume that the ring finger was standing next to the little finger and so would also be touching the thumb. This meant that often the ring finger was seen as the pinching finger instead of the little finger.

To accurately determine the inaccuracy of the leap motion a conducting glove was used. With this glove it would be possible to detect the time of pinches and the amount and compare it to what the leap motion detected. This could quantify the inaccuracy of the leap motion but the glove itself turned out to be faulty. The glove was already of a certain age and its conductivity might have deteriorated making it impossible to use as a comparison tool.

A medical application has no room for error or misinterpretation, every time the leap motion makes a mistake a wrong diagnosis might be made. This means that the leap motion, in its current state, is not accurate enough to be used for this application.

6 Conclusion

At the end of this master's thesis a working platform is created where patients and researchers can sign in to. Patients can then subscribe to researches that are created by researchers and approved by admins. Patients can then perform experiments in their subscribed researches.

Researchers have to upload a PDF file, that needs to be verified by the admin, that proves that they are a valid researcher. They can then start their own research and add new experiments from the existing templates.

Admin privileges can only be granted through the database. Admins have to approve new researchers as well as their researches. Changes in experiment parameters also have to be checked by the admin and approved before they will take effect.

This platform combines multiple programming languages including: JavaScript, Java, HTML, CSS, C++ and PHP. In the bimanual motor tracking experiment a connection is made with an external USB device that sends its data directly to the browser.

The leap motion turned out to be too inaccurate to use in a medical application in its current state. It was unable to detect finger pinches without misinterpretations or missed pinches. This made it not reliable for a medical application where one mistake might result in a wrong diagnosis.

References

[1] „11 Best PHP Frameworks for Modern Web Developers in 2018,” [Online]. Available: https://coderseye.com/best-php-frameworks-for-web-developers/.

[2] T. point, „Hibernate: Java persisntace framework,” Tutorials Point (I) Pvt. Ltd., 2015.

[3] Y. Shiotsu, „Web Development 101: Top Web Development Languages to Learn in 2018,” [Online]. Available: https://www.upwork.com/blog/2017/11/top-web-development-languages-2018/. [Geopend 10 May 2018].

[4] "Tropical Labs - mechaduino," [Online]. Available: http://tropical-labs.com/index.php/mechaduino. [Accessed 25 may 2018].

[5] A. Luyten en L. Peeters, „de ontwikkeling van feedforward controle bij typische ontwikkelende kinderen tijdens de uitvoering van een visuomanuele tracking taak,” 2018.

[6] V. Nieuwinckel, „window.requestAnimationFrame(),” [Online]. Available: https://developer.mozilla.org/nl/docs/Web/API/Window/requestAnimationFrame. [Geopend 15 May 2018].

[7] f. unwelcome, „Can't get past 2542 Threads in Java on 4GB iMac OSX 10.6.3 Snow Leopard,” [Online]. Available: https://stackoverflow.com/questions/2860889/cant-get-past-2542-threads-in-java-on-4gb-imac-osx-10-6-3-snow-leopard-32bit. [Geopend 14 November 2017].

[8] M. T. &. V. T. Macoye Lorenz, „Development of an Interactive Game Controller to Enhance the Revalidation of MS Patients,” Faculty of Computer Sciences, University of Hasselt, 2017.

[9] ivanakimov, „ivanakimov/hashids,” [Online]. Available: https://github.com/ivanakimov/hashids.php. [Geopend 2018 May 25].

[10] Zizaco, „Zizaco/entrust,” [Online]. Available: https://github.com/Zizaco/entrust. [Geopend 24 May 2018].

[11] B. Moore, „JsViews,” [Online]. Available: https://www.jsviews.com/. [Geopend 15 May 2018].

[12] „Chart.js,” [Online]. Available: https://www.chartjs.org/. [Geopend 23 May 2018].

[13] Rochal, „jQuery slimScroll,” [Online]. Available: http://rocha.la/jQuery-slimScroll. [Geopend 23 May 2018].

[14] cferdinandi, "cferdinandi/smooth-scroll," [Online]. Available: https://github.com/cferdinandi/smooth-scroll. [Accessed 25 May 2018].

[15] „PACE Automatic page load progress bar,” [Online]. Available: http://github.hubspot.com/pace/docs/welcome/. [Geopend 23 May 2018].

[16] K. Brown, „Select2," [Online]. Available: https://select2.org/. [Geopend 23 May 2018].

[17] J. H. A. A. C. S. R. H. T. R. D. D. D. K. M. P. T. T. E. V. P. T. B. H. A. C. J. L. C. L. M. F. S. B. Rod Johnson, „Spring java/j2ee Application Framework," 2005-2008.

[18] S. F. Docs, „REST API Developer Guide," salesforce.com, inc, 2016.

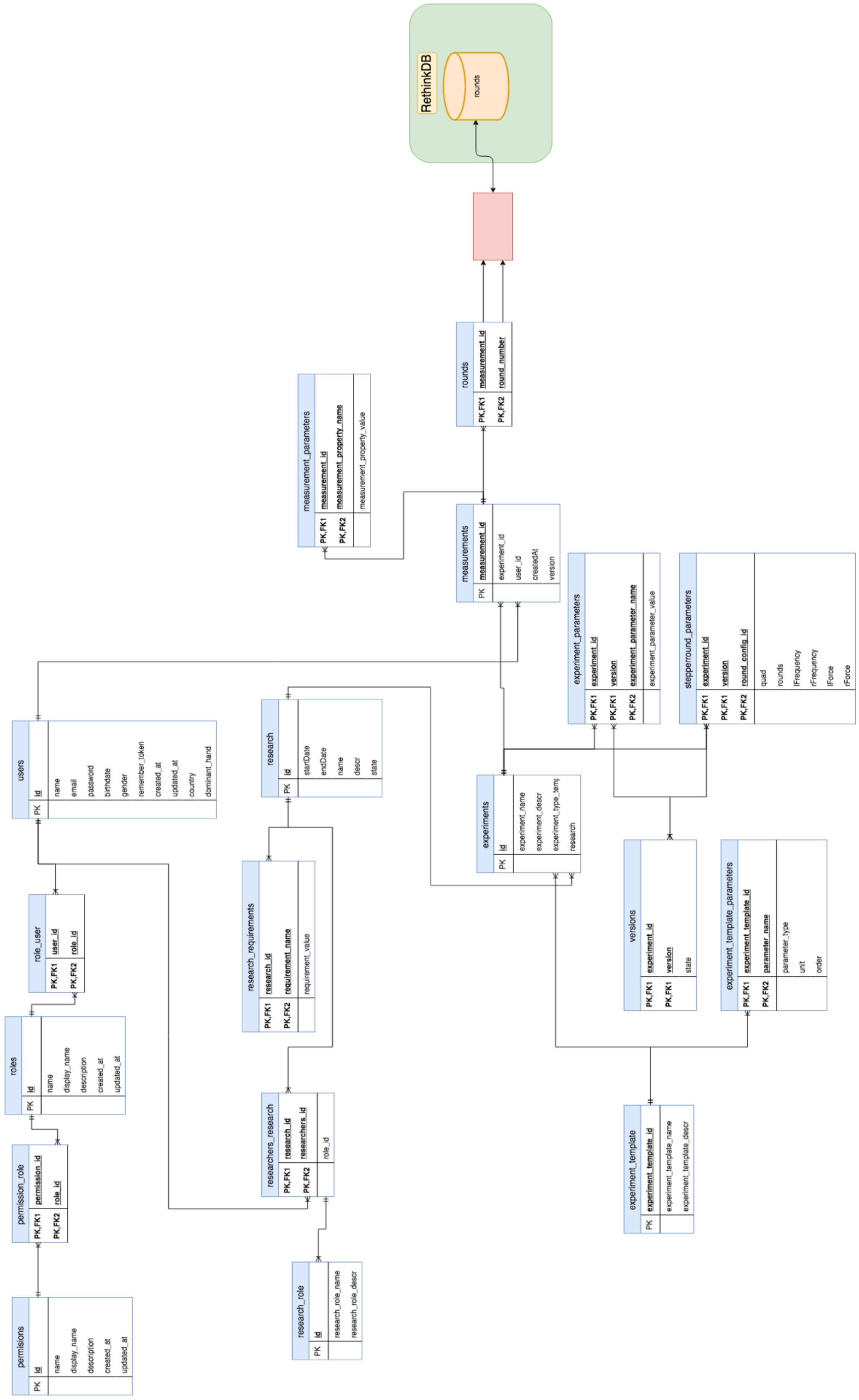
[19] „The open-source database for the realtime web, RethinkDB," [Online]. Available: https://www.rethinkdb.com/. [Geopend 25 May 2018].

[20] „Laravel documentation," [Online]. Available: https://laravel.com/docs/master.

[21] „Leap Motion documentation," [Online]. Available: https://developer.leapmotion.com/documentation/java/index.html.

Attachments

Attachment A: Entity-relationship diagram



Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Ontwerp en implementatie van een revalidatie platform gebruik makende van de leap motion sensor

Richting: **master in de industriële wetenschappen: elektronica-ICT**
Jaar: **2018**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Debien, Ruben **Vanstraelen, Jonas**

Datum: **4/06/2018**